Sudip K. Mazumder
Editor

Wireless Networking Based Control

Springer

Wireless Networking Based Control

Sudip K. Mazumder
Editor

Wireless Networking Based Control

Springer

Editor
Sudip K. Mazumder
University of Illinois, Chicago
Dept. Electrical and Computer Eng.
851 South Morgan Street
1020 Science and Eng. Offices
Chicago Illinois 60607
USA
mazumder@ece.uic.edu
sudipkumarmazumder@gmail.com

ISBN 978-1-4419-7392-4 e-ISBN 978-1-4419-7393-1
DOI 10.1007/978-1-4419-7393-1
Springer New York Dordrecht Heidelberg London

Preface

Wireless networking is gaining significant momentum in several areas of application due to advantages encompassing mobility, reconfigurability, easy commissioning, and spatio-temporal sensing. While initial focus of wireless networking has been on communication and sensing alone, a new field now has emerged that uses the same communication channel for enabling network control. This leads to several interesting issues and possibilities that are not typically encountered in traditional wire-based network control. This book will emphasize on and outline some of those issues from the standpoints of both theory and applications with focus on the core theme of control using wireless network and control of the information exchanged over the wireless network. Broadly, the topics covered in this book encompass the following:

- Robust stabilization of wireless network control systems in the presence of delays, packet drop out, fading
- State estimation over wireless network under random measurement delay
- Distributed optimization algorithm for addressing feedback delay and network-throughput tradeoff in wireless control-communication network
- Cyber-physical control over wireless sensor and actuator networks
- Estimation of a dynamical system over a wireless fading channel using Kalman filter
- Control over wireless multi-hop networks based on time-delay and finite spectrum assignment
- Position localization in wireless sensor networks
- Cross-layer optimized-based protocols for control over wireless sensor networks
- Rendezvous problem and consensus protocols for application in control of distributed mobile wireless networks
- Redeployment control of mobile sensors for enhancing wireless network quality and channel capacity
- Design of IEEE 802.15.4-based performance-metrics-optimizing distributed and adaptive algorithms and protocols for wireless control and monitoring applications
- Coordinated control over low-frequency-radio-based ad-hoc underwater wireless communication network

It is my sincere hope that, by providing an overview on important, interesting, and relevant issues related to wireless network-based control, this book, which represents the work of motivated researchers, will provide a great service to the community and create greater interest in this rapidly growing field.

Chicago, Illinois, USA Sudip K. Mazumder

Contents

Contributors

Carlos H. Caicedo-Núñez Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA, ccaicedo@princeton.edu

Jian Chen Intelligent Systems and Control, School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, County Antrim, UK, jchen04@qub.ac.uk

Carlo Fischione ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, carlofi@ee.kth.se

Alireza Ghaffarkhah Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87113, USA, alinem@ece.unm.edu

Vijay Gupta University of Notre Dame, Notre Dame, IN, USA, Vijay.Gupta.21@nd.edu

João P. Hespanha University of California, Santa Barbara, CA, USA, hespanha@ece.ucsb.edu

Steve Huseth Honeywell ACS Labs, Golden Valley, MN, USA, steve.huseth@honeywell.com

George W. Irwin Intelligent Systems and Control, School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, County Antrim, UK, g.irwin@qub.ac.uk

Karl Henrik Johansson ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, kallej@ee.kth.se

Mikael Johansson ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, mikaelj@ee.kth.se

Daniel J. Klein Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, USA, djklein@ece.ucsb.edu

Soumitri Kolavennu Honeywell ACS Labs, Golden Valley, MN, USA, soumitri.kolavennu@honeywell.com

Xiangjie Kong School of Software, Dalian University of Technology, Dalian 116620, China, xjkong@dlut.edu.cn

Feng-Li Lian Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, fengli@ntu.edu.tw

Yi-Chun Lin Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, d96921002@ntu.edu.tw

Piergiuseppe Di Marco ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, pidm@ee.kth.se

Sudip K. Mazumder Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA, mazumder@ece.uic.edu

Adrian McKernan Intelligent Systems and Control, School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, County Antrim, UK, amckernan01@qub.ac.uk

Kristi A. Morgansen University of Washington, Seattle, WA, USA, morgansen@aa.washington.edu

Yasamin Mostofi Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87113, USA, ymostofi@ece.unm.edu

Richard M. Murray Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91106, USA, murray@cds.caltech.edu

Payam Naghshtabrizi Ford Motor Company, Dearborn, MI, USA, pnaghsht@ford.com

Pangun Park ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, pgpark@ee.kth.se

Ling Shi Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, eesling@ust.hk

Muhammad Tahir Department of Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan, mtahir@uet.edu.pk

Ko-Hsin Tsai Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, ncslab@cc.ee.ntu.edu.tw

Emmanuel Witrant UJF GIPSA-lab, Department of Control Systems, University of Grenoble, Saint Martin d'Hères, France, emmanuel.witrant@gipsa-lab.inpg.fr

Feng Xia School of Software, Dalian University of Technology, Dalian 116620, China, f.xia@ieee.org

Lihua Xie The School of Electrical and Electrical Engineering, Nanyang Technological University, Singapore, elhxie@ntu.edu.sg

Zhenzhen Xu School of Software, Dalian University of Technology, Dalian 116620, China, xzz@dlut.edu.cn

Miloš Žefran Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA, mzefran@uic.edu

Chapter 1
Implementation Considerations For Wireless Networked Control Systems

Payam Naghshtabrizi and João P. Hespanha

Abstract We show that delay impulsive systems are a natural framework to model wired and wireless NCSs with variable sampling intervals and delays as well as packet dropouts. Then, we employ discontinuous Lyapunov functionals to characterize admissible sampling intervals and delays such that exponential stability of NCS is guaranteed. These results allow us to determine requirements needed to establish exponential stability, which is the most basic Quality of Performance (QoP) required by the application layer. Then we focus on the question of whether or not the Quality of Service (QoS) provided by the wireless network suffices to fulfill the required QoP. To answer this question, we employ results from real-time scheduling and provide a set of conditions under which the desired QoP can be achieved.

Keywords Network control system · Quality of service · Quality of performance · Lyapunov functional · Delay · Scheduling · Control · Sampling · System

1.1 Introduction

Network Control Systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators, and controllers occurs through a shared band-limited digital communication network, as shown in Fig. 1.1. There are two types of NCSs in terms of medium used at the physical layer: wired and wireless. Wired NCSs have been used widely in automotive and aerospace industry [14] to reduce weight and cost, increase reliability and connectivity. Particularly drive-by-wire and fly-by-wire systems have shown a high penetration rate in these industries. In wireless NCSs, the communication relies on the wireless technology and it has been finding applications in a broad range of areas that in which it is difficult or expensive to install wire, such as mobile sensor networks [17], HVAC systems [1], automated highway, and unmanned aerial vehicles [18].

P. Naghshtabrizi (✉)
Ford Motor Company, Dearborn, MI, USA
e-mail: pnaghsht@ford.com

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_1,

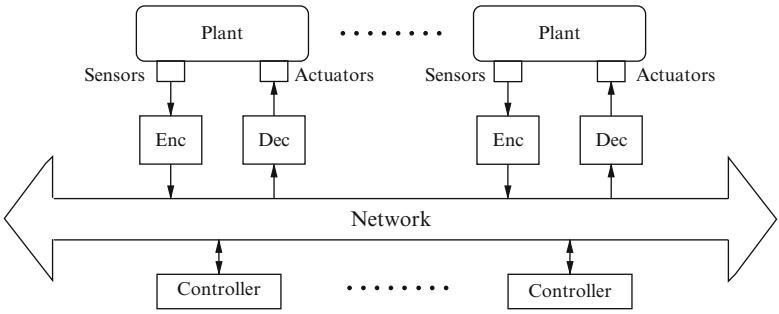


Fig. 1.1 General NCS architecture

In both the wired and wireless domains, use of a shared network – in contrast to using several dedicated independent connections – introduces new challenges to NCSs [7]. However, the reduced channel reliability and limited bandwidth that characterize the wireless domain require special care. In this paper, we mainly focus on wireless NCSs, although most of the results presented are also applicable to wired NCSs. Traditional control theory assumes the feedback data to be accurate, timely, and lossless. These assumptions do not hold for wireless NCSs and the following factors have to be considered:

Sampling and Delay. To transmit a continuous-time signal over a network, the signal must be sampled, encoded in a digital format, transmitted over the network, and finally the data must be decoded at the receiver side. This process is significantly different from the usual periodic sampling in digital control. The overall *delay* between sampling and eventual decoding at the receiverside/end can be highly variable because both the network access delays (i.e., the time it takes for a shared network to accept data) and the transmission delays (i.e., the time during which data are in transit inside the network) depend on highly variable network conditions, such as congestion and channel quality. In some NCSs, the data transmitted are time-stamped, which means that the receiver may have an estimate of the delay's duration and take appropriate corrective action.

Packet dropout. Another significant difference between NCSs and standard digital control is the possibility that data may be lost while in transit through the network. Typically, *packet dropouts* result from transmission errors in physical network links (which is far more common in wireless than in wired networks) or from buffer overflows due to congestion. Long transmission delays sometimes result in packet reordering, which essentially amounts to a packet dropout if the receiver discards "outdated" arrivals.

Systems architecture. Figure 1.1 shows the general architecture of an NCS. In this figure, the *encoder* blocks map measurements into streams of "symbols" that can be transmitted across the network. Encoders serve two purposes: they decide *when* to sample a continuous-time signal for transmission and *what* to send through the network. Conversely, the *decoder* blocks perform the task of mapping the streams of symbols received from the network into continuous actuation signals. One could

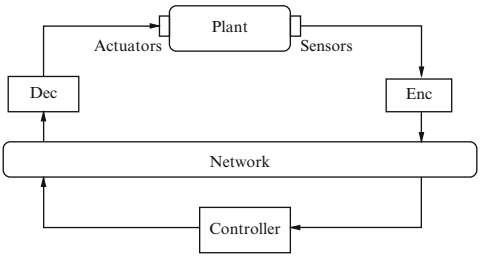


Fig. 1.2 Single-loop NCS

also include in Fig. 1.1 encoding/decoding blocks to mediate the controllers' access to the network. Throughout this paper, the encoder is simply a sampler and the decoder is a hold. However, in Sect. 1.3.1.3 we will consider more sophisticated encoder/decoder pairs.

Most of the research on NCSs considers structures simpler than the general one depicted in Fig. 1.1. For example, some controllers may be collocated (and therefore can communicate directly) with the corresponding actuators. It is also often common to consider a single feedback loop as in Fig. 1.2. Although considerably simpler than the system shown in Fig. 1.1, this architecture still captures many important characteristics of NCSs, such as bandwidth limitations, variable communication delays, and packet dropouts. In this paper, we only consider linear plants and controllers; however, some of the results can be extended to nonlinear systems [11].

In Sects. 1.2 and 1.3, we show that delay impulsive systems provide a natural framework to model (wireless) NCSs with variable sampling intervals and delays as well as packet dropouts. Then, we employ discontinuous Lyapunov functionals to derive a condition that can be used to guarantee stability of the closed-loop NCS. This condition is expressed in the form of a set of LMIs that can be solved numerically using software packages such as MATLAB. By solving these LMIs, one can characterize admissible sampling intervals and delays for which exponential stability of the NCS is guaranteed.

From a networking perspective, the NCS is implemented using the usual layered architecture consists of application layer, network layer, MAC layer and physical layer [10]. From this perspective, our goal is to determine under what conditions the network can provide stabilization, which is the most basic form of Quality of Performance (QoP). In essence, the stability conditions place requirements on the Quality of Service (QoS) that the lower layers need to provide to the application layer to obtain the desired QoP.

Section 1.4 is focused precisely on determining conditions under which the network provides a level of QoS that permits the desired application layer QoP. We review different real-time scheduling policies and identify the ones implementable on wireless NCSs. Among the discussed policies, the most desirable is Earliest Deadline First (EDF) because it has the advantage of being a dynamic algorithm that can adapt to changes in the wireless network. For EDF scheduling, we provide a set of conditions, often known as scheduling tests in real-time literature, that when satisfied, ensures the desired QoS of wireless NCS. Finally, in Sect. 1.5, we address the question of how to implement EDF scheduling policy on LAN wireless NCSs.

Notation. We denote the transpose of a matrix P by P'. We write $P > 0$ (or $P < 0$) when P is a symmetric positive (or negative) definite matrix and we write a symmetric matrix $\begin{bmatrix} A & B \\ B' & C \end{bmatrix}$ as $\begin{bmatrix} A & B \\ * & C \end{bmatrix}$. We denote the limit from below of a signal $x(t)$ by $x^-(t)$, i.e., $x^-(t) := \lim_{\tau \uparrow t} x(\tau)$. Given an interval $I \subset \mathbb{R}$, $B(I, \mathbb{R}^n)$ denotes the space of real functions from I to $\mathbb{R}^n$ with norm $\|\phi\| := \sup_{t \in I} |\phi(t)|, \phi \in B(I, \mathbb{R}^n)$, where $|.|$ denotes any one of the equivalent norms in $\mathbb{R}^n$. x_t denotes the function $x_t : [-r, 0] \to \mathbb{R}^n$ defined by $x_t(\theta) = x(t + \theta)$, and r is a fixed positive constant.

1.1.1 Related Work

To reduce network traffic in NCSs, significant work has been devoted to finding maximum allowable transmission interval τ_{MATI} that are not overly conservative (see [7] and references therein). First, we review the related work in which there is no delay in the control loop. In [21], τ_{MATI} is computed for linear and nonlinear systems with Round-Robin (static) or Try-Once-Discard (TOD) (dynamic) protocols. Nesic et al. [15, 16] study the input–output stability properties of nonlinear NCSs based on a small gain theorem to find τ_{MATI} for NCSs. Fridman et al. [6], Naghshtabrizi et al. [13], Yue et al. [24] consider linear NCSs and formulate the problem of finding τ_{MATI} by solving LMIs. In the presence of variable delays in the control loop, [5, 12, 25] show that for a given lower bound $\tau_{\min}$ on the delay in the control loop, stability can be guaranteed for a less conservative τ_{MATI} than in the absence of the lower bound.

Ye et al. [23] introduced prioritized Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for mixed wireless traffic, in which some of the network capacity is devoted to real-time control and monitoring. They introduced and validated several new algorithms for dynamically scheduling the traffic of wireless NCSs. We use a similar MAC protocol for the wireless network (more precisely wireless LAN networks). Liu and Goldsmith [10] presented a cross layer codesign of network and distributed controllers and addressed the tradeoff between communication and controller performance. The designed controller is robust and adaptive to the communication faults, such as random delays and packet losses, while the network should be designed with the goal of optimizing the end-to-end control performance. Tabbara et al. [19] defined the notion of persistently exciting scheduling protocols and showed that it is a natural property to demand, especially for the design of wireless NCSs. Xia et al. [22] developed a cross layer adaptive feedback scheduling scheme to codesign control and wireless communications. The authors identified that the Deadline Miss Ratio (DMR) is an important factor to determine the sampling intervals. Consequently, the authors proposed a sampling algorithm that is the minimum of a function of DMR and maximum sampling period.

1.2 Delay Impulsive Systems: A Model For NCSs With Variable Sampling And Delay, SISO Case

Consider the system depicted in Fig. 1.3. The LTI process has a state space model of the form $\dot{x}(t) = Ax(t) + Bu(t)$, where x, u are the state and input of the process. At the sampling time $s_k, k \in \mathbb{N}$ the process state, $x(s_k)$ is sent to update the process input to be used as soon as it arrives and it should be kept constant until the next control command update. We denote the *k-th input update time* by t_k, which is the time instant at which the k-th sample arrives at the destination. In particular, denoting by τ_k the total delay that the k-th sample experiences in the loop, then $t_k := s_k + \tau_k$. The resulting closed-loop system can be written as

$$\dot{x}(t) = Ax(t) + Bx(s_k), \qquad t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1}),\ k \in \mathbb{N}. \tag{1.1}$$

We write the resulting closed-loop system (1.1) as an impulsive system of the form

$$\dot{\xi}(t) = F\xi(t), \quad t \neq t_k,\ \forall k \in \mathbb{N} \tag{1.2a}$$

$$\xi(t_k) = \begin{bmatrix} x^-(t_k) \\ x(s_k) \end{bmatrix}, \quad t = t_k, \forall k \in \mathbb{N}, \tag{1.2b}$$

where

$$F := \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \quad \xi(t) := \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}.$$

The overall state of the system ξ is composed of the process state, x, and the *hold state*, z where $z(t) := x(s_k),\ t \in [t_k, t_{k+1})$.

1.2.1 NCSs Modeled By Impulsive Systems

Equations (1.2) or (1.1) can be used to model NCSs in which a linear plant $\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t)$ where $x_p \in \mathbb{R}^n, u_p \in \mathbb{R}^m$ are the state and the input of the plant, respectively, is in feedback with a static feedback gain K. At time $s_k, k \in \mathbb{N}$ the plant's state, $x(s_k)$, is sent to the controller and the control command $Kx(s_k)$ is sent back to the plant to be used as soon as it arrives and it should be kept constant

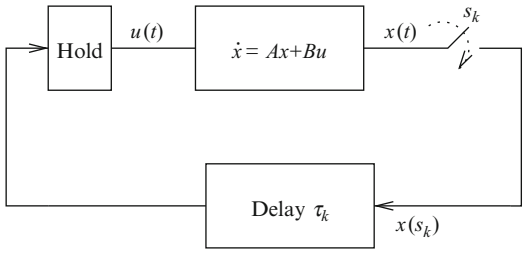


Fig. 1.3 An abstract system with delay τ_k, where $u(t) = x(s_k)$ for $t \in [s_k + \tau_k, s_{k+1} + \tau_{k+1})$

until the next control command update. In particular, denoting by τ_k the total delay that the k-th sample experiences in the loop, then $t_k := s_k + \tau_k$. Then the closed-loop system can be written as (1.2) with $x := x_p,\ A := A_p,\ B := B_p K$.

Remark 1. Note that we only index the samples that reach the destination, which enables us to capture sample drops [24]. Consequently, even if the sampling intervals are constant, because of the sample drops the closed-loop should still be seen as a system with variable sampling intervals.

1.2.2 Exponential Stability Of SISO NCSs

In this section, we provide conditions in terms of LMIs to guarantee exponential stability of the linear delay impulsive system in (1.2) which models the NCS described in Sect. 1.2.1. The system (1.2) is said to be (globally uniformly) exponentially stable over a given set $\mathscr{S}$ of sampling-delay sequences, if there exist $c, \lambda > 0$ such that for every $(\{s_k\}, \{\tau_k\}) \in \mathscr{S}$ and every initial condition x_{t_0} the solution to (1.2) satisfies $|x(t)| \leq c\|x_{t_0}\| e^{-\lambda(t-t_0)},\ \forall t \geq t_0$.

In this paper, we are mostly interested in class $\mathscr{S}$ of admissible sampling-delay sequences characterized by three parameters: The maximum interval of time τ_{MATI} between a signal is sampled and the *following* sample arrives at the destination; the minimum delay $\tau_{\min}$; and the maximum delay $\tau_{\max}$. Specifically, to be consistent with the results in [12, 25], and [5], we characterize the admissible set $\mathscr{S}$ of sampling-delay sequences $(\{s_k\}, \{\tau_k\})$ such that

$$s_{k+1} + \tau_{k+1} - s_k \leq \tau_{\text{MATI}}, \qquad \tau_{\min} \leq \tau_k \leq \tau_{\max}. \tag{1.3}$$

Although we adopt the above characterization, (1.3) is not in a convenient form to provide the sampling rule. Another characterization is the admissible set $\bar{\mathscr{S}}$ of sampling-delay sequences $(\{s_k\}, \{\tau_k\})$ such that

$$s_{k+1} - s_k \leq \gamma_{\max}, \qquad \tau_{\min} \leq \tau_k \leq \tau_{\max}, \tag{1.4}$$

which provides an explicit bound on the sampling intervals. Note that if any sampling-delay sequence belongs to $\bar{\mathscr{S}}$, it necessarily belongs to $\mathscr{S}$ provided that $\gamma_{\max} := \tau_{\text{MATI}} - \tau_{\max}$.

The following theorem was proved in [11] based on the Lyapunov functional

$$\begin{aligned} V :=& x' P x + \int_{t-\rho_1}^{t} (\rho_{1\max} - t + s)\dot{x}'(s) R_1 \dot{x}(s) \mathrm{d}s \\ &+ \int_{t-\rho_2}^{t} (\rho_{2\max} - t + s)\dot{x}'(s) R_2 \dot{x}(s) \mathrm{d}s + \int_{t-\tau_{\min}}^{t} (\tau_{\min} - t + s)\dot{x}'(s) R_3 \dot{x}(s) \mathrm{d}s \\ &+ \int_{t-\rho_1}^{t-\tau_{\min}} (\rho_{1\max} - t + s)\dot{x}'(s) R_4 \dot{x}(s) \mathrm{d}s + (\rho_{1\max} - \tau_{\min}) \int_{t-\tau_{\min}}^{t} \dot{x}'(s) R_4 \dot{x}(s) \mathrm{d}s \\ &+ \int_{t-\tau_{\min}}^{t} x'(s) Z x(s) \mathrm{d}s + (\rho_{1\max} - \rho_1)(x - w)' X (x - w), \end{aligned} \tag{1.5}$$

with $P, X, Z, R_i, i = 1, .., 4$ positive definite matrices and

$$w(t) := x(t_k), \qquad \rho_1(t) := t - s_k, \quad \rho_2(t) := t - t_k, \quad t_k \le t < t_{k+1}, \tag{1.6}$$

$$\rho_{1\max} := \sup_{t\ge 0} \rho_1(t), \qquad \rho_{2\max} := \sup_{t\ge 0} \rho_2(t). \tag{1.7}$$

Theorem 1. *The system (1.2) is exponentially stable over $\mathscr{S}$ defined by (1.3), if there exist positive definite matrices $P, X, Z, R_i, i = 1, \ldots, 4$ and (not necessarily symmetric) matrices $N_i, i = 1, \ldots, 4$ that satisfy the following LMIs:*

$$\begin{bmatrix} M_1 + \tau_{\mathrm{MATI}}(M_2 + M_3) & \tau_{\max} N_1 & \tau_{\min} N_3 \\ * & -\tau_{\max} R_1 & 0 \\ * & * & -\tau_{\min} R_3 \end{bmatrix} < 0, \tag{1.8a}$$

$$\begin{bmatrix} M_1 + \tau_{\mathrm{MATI}} M_2 & \tau_{\max} N_1 & \tau_{\min} N_3 & \tau_{\mathrm{MATI}}(N_1 + N_2) & \tau_{\mathrm{MATI}} N_4 \\ * & -\tau_{\max} R_1 & 0 & 0 & 0 \\ * & * & -\tau_{\min} R_3 & 0 & 0 \\ * & * & * & -\tau_{\mathrm{MATI}}(R_1 + R_2) & 0 \\ * & * & * & * & -\tau_{\mathrm{MATI}} R_4 \end{bmatrix} < 0, \tag{1.8b}$$

where

$$\begin{aligned} M_1 := {} & \bar{F}' \begin{bmatrix} P & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} P \\ 0 \\ 0 \\ 0 \end{bmatrix} \bar{F} + \tau_{\min} F'(R_1 + R_3) F - \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} X \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix}' \\ & + \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix} Z \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}' - \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix} Z \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix}' - N_1 \begin{bmatrix} I & -I & 0 & 0 \end{bmatrix} - \begin{bmatrix} I \\ -I \\ 0 \\ 0 \end{bmatrix} N_1' - N_2 \begin{bmatrix} I & 0 & -I & 0 \end{bmatrix} \\ & - \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} N_2' - N_3 \begin{bmatrix} I & 0 & 0 & -I \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ 0 \\ -I \end{bmatrix} N_3' - N_4 \begin{bmatrix} 0 & -I & 0 & I \end{bmatrix} - \begin{bmatrix} 0 \\ -I \\ 0 \\ I \end{bmatrix} N_4', \end{aligned}$$

$$M_2 := \bar{F}'(R_1 + R_2 + R_4)\bar{F}, \qquad M_3 := \begin{bmatrix} I \\ 0 \\ -I \\ 0 \end{bmatrix} X \bar{F} + \bar{F}' X \begin{bmatrix} I & 0 & -I & 0 \end{bmatrix},$$

with $\bar{F} := \begin{bmatrix} A & B & 0 & 0 \end{bmatrix}$. □

If the LMIs in (1.8) are feasible for given τ_{MATI}, $\tau_{\min}$, and $\tau_{\max}$, then there exists a $d_3 > 0$ such that $\frac{\mathrm{d}V(x_t,t)}{\mathrm{d}t} \leq -d_3|x(t)|^2$. It is straightforward to show that the Lyapunov functional (1.5) satisfies the remaining conditions of Theorem 15 in [11] that provides sufficient conditions for exponential stability of delay impulsive systems. Hence, the NCS modeled by the delay impulsive system (1.2) is (globally uniformly) exponentially stable over $\mathscr{S}$ given by (1.3).

When the load in the network is low and the computation delays are small, the total end-to-end delay in the loop is dominated by transmission and propagation delays, which can be small. This motivates a closer examination of the case $\tau_{\min} = 0$, which is the subject of the following result. The conditions in the theorem that follows can be derived from the conditions in Theorem 1 for the case when $\tau_{\min} \to 0$ or they can be directly derived employing the following Lyapunov functional

$$V := x'Px + \int_{t-\rho_1}^{t} (\rho_{1\max} - t + s)\dot{x}'(s)R_1\dot{x}(s)\mathrm{d}s + \int_{t-\rho_2}^{t} (\rho_{2\max} - t + s)\dot{x}'(s)R_2\dot{x}(s)\mathrm{d}s + (\rho_{1\max} - \rho_1)(x-w)'X(x-w).$$

Theorem 2. *The system (1.2) is exponentially stable over $\mathscr{S}$ defined by (1.3) with $\tau_{\min} = 0$, if there exist positive definite matrices P, X, R_1, R_2 and (not necessarily symmetric) matrices N_1, N_2 that satisfy the following LMIs:*

$$\begin{bmatrix} M_1 + \tau_{\text{MATI}}(M_2 + M_3) & \tau_{\max}N_1 \\ * & -\tau_{\max}R_1 \end{bmatrix} < 0, \tag{1.9a}$$

$$\begin{bmatrix} M_1 + \tau_{\text{MATI}}M_2 & \tau_{\max}N_1 & \tau_{\text{MATI}}(N_1 + N_2) \\ * & -\tau_{\max}R_1 & 0 \\ * & * & -\tau_{\text{MATI}}(R_1 + R_2) \end{bmatrix} < 0, \tag{1.9b}$$

where

$$\begin{aligned} M_1 :=& \bar{F}'\begin{bmatrix} P & 0 & 0 \end{bmatrix} + \begin{bmatrix} P \\ 0 \\ 0 \end{bmatrix}\bar{F} - \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} X \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix}' - N_1\begin{bmatrix} I & -I & 0 \end{bmatrix} - \begin{bmatrix} I \\ -I \\ 0 \end{bmatrix} N_1' \\ & - N_2\begin{bmatrix} I & 0 & -I \end{bmatrix} - \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} N_2' \\ M_2 :=& \bar{F}'(R_1 + R_2 + R_4)\bar{F}, \\ M_3 :=& \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} X\bar{F} + \bar{F}'X\begin{bmatrix} I & 0 & -I \end{bmatrix}, \end{aligned} \tag{1.10}$$

with $\bar{F} := \begin{bmatrix} A & B & 0 \end{bmatrix}$. □

1.2.3 Example

Consider the state space plant model

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -0.1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 0.1\end{bmatrix}u,$$

with state feedback gain $K = -\begin{bmatrix}3.75 & 11.5\end{bmatrix}$, for which we have [2]

$$A = \begin{bmatrix}0 & 1\\ 0 & -0.1\end{bmatrix}, \qquad B = -\begin{bmatrix}0\\ 0.1\end{bmatrix} \times \begin{bmatrix}3.75 & 11.5\end{bmatrix}.$$

By checking the condition $eig\left(\begin{bmatrix}I & 0\\ I & 0\end{bmatrix} e^{Fh}\right) < 0$ on a tight grid of h, we can show that the closed-loop system remains stable for any *constant* sampling interval smaller than 1.7, and becomes unstable for larger constant sampling intervals. On the other hand, when the sampling interval approaches zero, the system is essentially described by a Delay Differential Equation (DDE) and we can find the maximum constant delay for which stability is guaranteed by looking at the roots of the characteristic function $\det(sI - A - B\mathrm{e}^{-\tau_0 s})$. Using the Pade approximation $\mathrm{e}^{-\tau_0 s} = \frac{1-s\tau_0/2}{1+s\tau_0/2}$ to compute the determinant polynomial, we conclude by the Routh–Hurwitz test that the system is stable for any constant delay smaller than 1.36. Comparing these numbers with the maximum variable sampling interval 1.1137 and the maximum variable delay 1.0744 both obtained using Theorem 1 (see below) reveals that this result is not very conservative.

No-delay and variable sampling. When there is no delay but the sampling intervals are variable, τ_{MATI} determines an upper bound on the variable sampling intervals $s_{k+1} - s_k$. The upper bound given by [6, 12, 24] (when $\tau_{\min} = 0$) is 0.8696 which is improved to 0.8871 in [25]. Theorem 1 and [13] give the upper bound equal to 1.1137.

Variable-delay and sampling. Figure 1.4(a) shows the value of τ_{MATI} obtained from Theorem 1, as a function of $\tau_{\min}$ for different values of $\tau_{\max}$. The dashed curves in Figs. 1.4(a) and 1.4(b) are the same and correspond to the largest τ_{MATI} for different values of $\tau_{\max}$. Figure 1.4(b) shows τ_{MATI} versus $\tau_{\min}$ where the results from [12, 25] are shown by $+$, $\times$, respectively. The values of τ_{MATI} given by [5] lie between the "$+$" and "$\times$" in Fig. 1.4(b) and we do not show them. In Theorem 1, τ_{MATI} is a function of $\tau_{\min}$ and $\tau_{\max}$. To be able to compare our result with the others, we consider two values for $\tau_{\max}$ and we obtain τ_{MATI} as a function of $\tau_{\min}$ based on Theorem 1. First we consider $\tau_{\max} = \tau_{\min}$, which is the case that the delay is constant and equal to the value of $\tau_{\min}$. The largest τ_{MATI} for a given $\tau_{\min}$ provided by Theorem 1 is shown using an "o" in Fig. 1.4(b). The second case is when $\tau_{\max} = \tau_{\mathrm{MATI}}$, which is the case where there can be very large delays in the loop in comparison with the sampling intervals. The largest τ_{MATI} for a given $\tau_{\min}$ for this case provided by

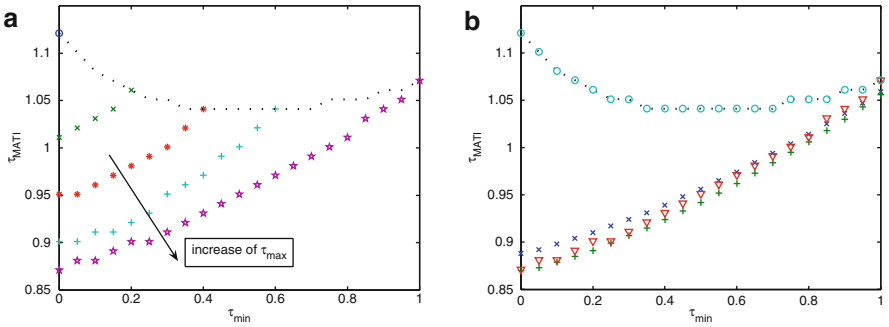


Fig. 1.4 Figure 1.4(**a**) shows the plot of τ_{MATI} versus $\tau_{\min}$ for $\tau_{\max}$ equal to 0, .2, .4, .6, 1 based on Theorem 1. The dashed line is the same as the one in Fig. 1.4(**b**). Figure 1.4(**b**) shows the plot τ_{MATI} versus $\tau_{\min}$ where $\tau_{\max} = \tau_{\min}$ from [12] ('+') and [25] ('×'), the worse case where $\tau_{\max} = \tau_{\text{MATI}}$ ('∇') and the best case where $\tau_{\max} = \tau_{\min}$ ('o') from Theorem 1

Theorem 1 is shown using a "∇" in Fig. 1.4(b). One can observe that when the delays in the control loop are small, our method shows a good improvement in comparison with the other results in the literature.

1.3 Delay Impulsive Systems: A Model for NCSs with Variable Sampling and Delay, MIMO Case

We now consider the MIMO system depicted in Fig. 1.5. The input is partitioned as $u := \begin{bmatrix} u_1' & \cdots & u_m' \end{bmatrix}'$ where $u_i \in \mathbb{R}^{q_i}$, $i \in \{1, \cdots, m\}$ and $\sum_{i=1}^{m} q_i = q$ and the output is partitioned similarly with $y := \begin{bmatrix} y_1' & \cdots & y_m' \end{bmatrix}'$ where $y_i \in \mathbb{R}^{q_i}$, $i \in \{1, \cdots, m\}$. At time s_k^i, $i \in \{1, \cdots, m\}, k \in \mathbb{N}$ the i-th output of the system, $y_i(t)$ is sampled and $y_i(s_k^i)$ is sent through the network to update u_i, to be used as soon as it arrives until the next update arrives. The total delay in the control loop that the k-th sample of y_i experiences is denoted by τ_k^i where $\tau_{i\min} \leq \tau_k^i \leq \tau_{i\max}$, $\forall k \in \mathbb{N}, i \in \{1, \cdots, m\}$. We define $t_k^i := s_k^i + \tau_k^i$, which is the time instant that the value of u_i is updated. The overall system can be written as an impulsive system of the form

$$\dot{\xi}(t) = F\xi(t), \qquad t \neq t_k^i,\ \forall k \in \mathbb{N}, i \in \{1, \cdots, m\} \tag{1.11a}$$

$$\xi(t_k) = \begin{bmatrix} x^-(t_k^i) \\ \cdots\cdots\cdots \\ z_1^-(t_k^i) \\ \vdots \\ y_i(s_k^i) \\ \vdots \\ z_m^-(t_k^i) \end{bmatrix}, \qquad t = t_k^i,\ \forall k \in \mathbb{N}, i \in \{1, \cdots, m\}, \tag{1.11b}$$

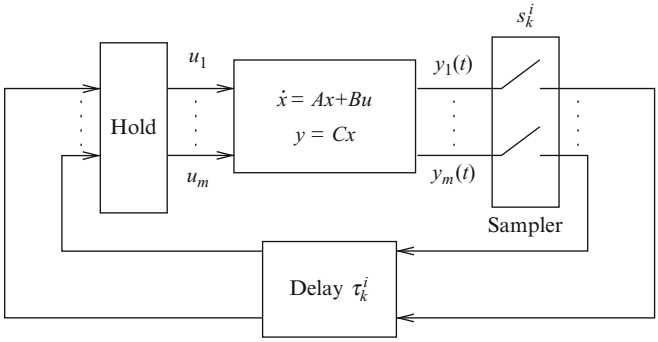


Fig. 1.5 MIMO system with variable sampling intervals and delays, where $u_i(t) = y_i(s_k^i)$ for $t \in [t_k^i, t_{k+1}^i), \forall i \in \{1, \cdots, m\}$ where $t_k^i := s_k^i + \tau_k^i$

where

$$F := \begin{bmatrix} A & \vdots & B \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix}, \qquad \xi(t) := \begin{bmatrix} x(t) \\ \cdots\cdots \\ z(t) \end{bmatrix}, \qquad z(t) := \begin{bmatrix} z_1(t) \\ \vdots \\ z_m(t) \end{bmatrix},$$

so we have $z_i(t) := y_i(s_k^i), t \in [t_k^i, t_{k+1}^i)$.

1.3.1 NCSs Modeled By MIMO System

The impulsive system (1.11) can be used to represent the distributed sensors/actuators configurations shown in Figs. 1.6 and 1.7. We now consider an LTI plant

$$\dot{x}_p(t) = A_p x_p(t) + B_p u_p(t), \qquad y_p(t) = C_p x_p(t), \tag{1.12}$$

where $x_p \in \mathbb{R}^{n_p}, u_p := \begin{bmatrix} u'_{p1} \cdots u'_{pm_c} \end{bmatrix}' \in \mathbb{R}^{m_c}$, and $y_p := \begin{bmatrix} y'_{p1} \cdots y'_{pm_p} \end{bmatrix}' \in \mathbb{R}^{m_p}$ are the state, input, and output of the plant and matrices, respectively. At time $s_k^i, i \in \{1, \cdots, m_p\}$, sensor i sends $y_{pi}(s_k^i)$ to the controller, which arrives at the destination at time t_k^i. When a new measurement of the sensor i arrives at the controller side, the corresponding value at the hold block, z_i, is updated and held constant until another measurement of the sensor i arrives (all other hold values remain unchanged when the value of hold i is updated). Hence, $u_{ci}(t) = z_i(t) = y_{pi}(s_k^i), t \in [t_k^i, t_k^{i+1})$ for $\forall i \in \{1, \cdots, m_p\}$. An output feedback controller (or a state feedback controller) uses the measurements to construct the control signal. The controller has the state space of the form

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t), \qquad y_c(t) = C_c x_c(t) + D_c u_c(t), \tag{1.13}$$

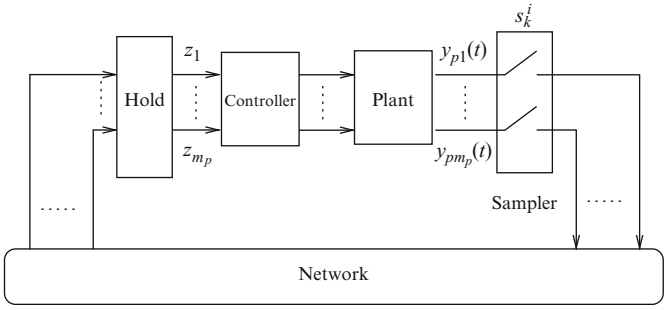


Fig. 1.6 One channel NCSs with the plant (1.12) and the controller (1.13)

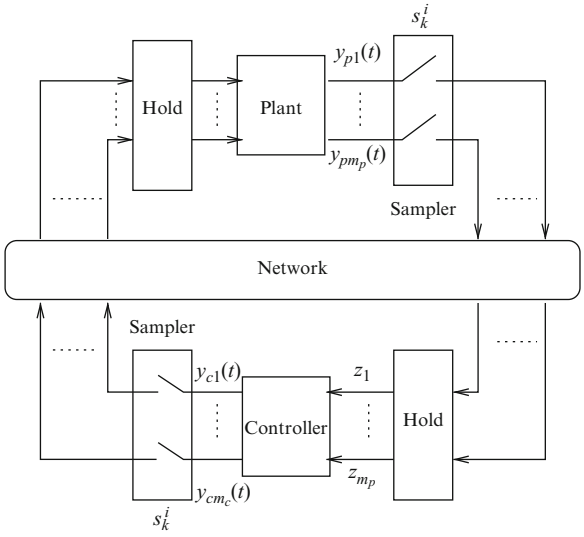


Fig. 1.7 Two channel NCS with the plant (1.12) and anticipative or nonanticipative controller (1.13)

where $x_c \in \mathbb{R}^{n_c}, u_c := \begin{bmatrix} u'_{c1} \cdots u'_{cm_p} \end{bmatrix}' \in \mathbb{R}^{m_p}, y_c := \begin{bmatrix} y'_{c1} \cdots y'_{cm_c} \end{bmatrix}' \in \mathbb{R}^{m_c}$ are the state, input, and output of the controller, respectively. The main difference between the NCS configurations discussed below is related to the control signal construction.

1.3.1.1 One-Channel NCS with Dynamic Feedback Controller

Figure 1.6 shows a one-channel NCS consisting of a plant (1.12) in feedback with a dynamic output controller (1.13). In this one-channel NCS, the controller is directly connected to the plant and only the measurements of the plant are sent through

the network. To match the system in Fig. 1.6 with the system in Fig. 1.5 $y_i(t) := y_{pi}(t)$, $m := m_p$, and $x := \begin{bmatrix} x_p \\ x_c \end{bmatrix}$. This closed-loop system can be written as the impulsive system (1.11), where

$$A := \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad B := \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix}, \quad C := \begin{bmatrix} C_p & 0 \end{bmatrix}. \tag{1.14}$$

One-channel NCSs may look artificial since the controller and the plant appear to be collocated and one may suggest that there is no need to send the output of the plant through the network. It turns out that there are important cases of NCSs that can be modeled as a one-channel NCSs. One example is controlling a robot using cameras that are not mounted on the robot, which provide the global image of the field. In this case, position of the robot is "measured" by cameras and the measurements are sent through the network to the robot to be used to compute the control commands by the local controller of the robot.

1.3.1.2 Two-Channel NCS with Nonanticipative Controller

Figure 1.7 shows a two-channel NCS consisting of a plant (1.12) in feedback with a *nonanticipative* controller (1.13) where $D_c = 0$. Nonanticipative controllers are simply output–feedback controllers for which a single value control command is calculated. Now the controller is located away from the actuators and the control commands also need to be sent through the network. The control signal for the actuator i, $y_{ci}(t)$, is sampled at times s_k^i, $i \in [m_p + 1, \cdots, m_p + m_c]$, and the samples get to the actuator i at times $t_k^i := s_k^i + \tau_k^i$. The nonanticipative control unit sends a single-value control command to be applied to the actuator i of the plant and held until the next control update of the actuator i arrives (all other actuator values remain unchanged while the value of actuator i is updated). Hence, $u_{pi}(t) = z_i(t) = y_{ci}(s_k^i)$, $t \in [t_k^i, t_{k+1}^i)$ for $m_p + 1 \le i \le m_p + m_c$. So, in this case we can model the closed loop as the impulsive system (1.11) in Fig. 1.5, with

$$y_i := \begin{cases} y_{pi}, & 1 \le i \le m_p \\ y_{ci}, & m_p + 1 \le i \le m_p + m_c \end{cases},$$

$m := m_p + m_c$, $x := \begin{bmatrix} x_p \\ x_c \end{bmatrix}$ and

$$A := \begin{bmatrix} A_p & 0 \\ 0 & A_c \end{bmatrix}, \qquad B := \begin{bmatrix} 0 & B_p \\ B_c & 0 \end{bmatrix}, \qquad C := \begin{bmatrix} C_p & 0 \\ 0 & C_c \end{bmatrix}. \tag{1.15}$$

1.3.1.3 Two-Channel NCS with Anticipative Controller

Figure 1.7 can also represent a two-channel NCS consisting of a plant with the state-space (1.12) in feedback with an *anticipative* controller with state-space (1.13). Anticipative controller attempts to compensate the sampling and delay introduced by the actuation channels. For simplicity, we assume that the actuation channels are sampled with constant sampling intervals $h = s^i_{k+1} - s^i_k$, and that the delay in the actuation channels is constant and equal to $\tau = \tau^i_k \; \forall k \in \mathbb{N}, i \in \{m_p + 1, \cdots, m_p + m_c\}$. At each sampling time $s^i_k, i \in [m_p + 1, \cdots, m_p + m_c]$ the controller sends a time-varying control signal $y_{ci}(\cdot)$ to the actuator i. This control signal should be used from the time $s^i_k + \tau$ at which it arrives until the time $s^i_k + h + \tau$ at which the next control update of the actuator i will arrive. This leads to $u_{pi}(t) = y_{ci}(t), \forall t \in \left[s^i_k + \tau, s^i_k + h + \tau\right)$. However, the prediction of control signal $y_{ci}(t)$ needed in the interval $\left[s^i_k + \tau, s^i_k + h + \tau\right)$ must be available at the transmission time s^i_k, which requires the control unit to calculate the control signal up to $h + \tau$ time units into the future.

Anticipative controllers send actuation signals to be used during time intervals of duration h, therefore the sample and hold blocks in Fig. 1.7 should be understood in a broad sense. In practice, the sample block would send over the network some parametric form of the control signal $u_{ci}(\cdot)$ (e.g., the coefficients of a polynomial approximation to this signal).

An estimate $\hat{x}_c(t)$ of $x_c(t + h + \tau)$ can be constructed as $\dot{\hat{x}}_c(t) = A_c \hat{x}_c(t) + B_c u_c(t)$, where

$$u_{ci}(t) = y_{pi}\left(s^i_k\right), \; \forall t \in \left[s^i_k + \tau, s^i_k + \tau + h\right), \tag{1.16}$$

for $i \in \{1, \cdots, m_p\}$. To compensate for time-varying delays and sampling intervals in the actuation channels, $\hat{x}_c$ would have to be calculated further into the future. Hence, the assumptions of constant delay and sampling interval for actuation channel can be relaxed by predicting x_c further into the future.

With the controller state prediction (1.16), the signal $y_{ci}(t)$ sent at times s^i_k, to be used in the interval $\left[s^i_k + \tau, s^i_k + h + \tau\right)$ is then given by

$$y_{ci}(t) = C_{ci} \hat{x}_c(t - h - \tau) + D_c u_{ci}(t), \; \forall t \in \left[s^i_k + \tau, s^i_k + h + \tau\right), \tag{1.17}$$

which only requires the knowledge of $\hat{x}_c(.)$ in the interval $t \in \left[s^i_k - h, s^i_k\right)$, available at the transmission times s^i_k. The closed-loop system can be written as

$$\begin{bmatrix} \dot{\hat{x}}_p(t) \\ \dot{\hat{x}}_c(t) \end{bmatrix} = \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} \hat{x}_p(t) \\ \hat{x}_c(t) \end{bmatrix} + \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix} u_c(t), \tag{1.18}$$

where $\hat{x}_p(t) := x_p(t + h + \tau)$ and the elements of $u_c(t)$ are defined by (1.16). Hence in this case $y_i(t) := y_{pi}(t), \quad m := m_p$ and the closed-loop system with

state $x := \begin{bmatrix} \hat{x}_p \\ \hat{x}_c \end{bmatrix}$ can be written as the impulsive system (1.11), where

$$A := \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad B := \begin{bmatrix} B_p D_c \\ B_c \end{bmatrix}, \quad C := \begin{bmatrix} C_p & 0 \end{bmatrix}. \tag{1.19}$$

Equation (1.14), which represents a one-channel NCS with dynamic output feedback, is similar to (1.19) that represents two-cannel NCS with anticipative controller. Consequently, for purpose of analysis, one can model a two-cannel NCS with anticipative controller as a one-channel NCS with *"fictitious"* delays equal to the sum of the delay in the sensor to actuator channels, the delay in the actuator to sensor channels, and the sampling of the actuator channel.

Remark 2. Anticipative controllers are similar to model predictive controllers in the sense that both calculate future control actions. However, in model predictive controllers only the most recent control prediction is applied. Anticipative controllers are predictive controllers that send a control prediction for a certain duration. At the expense of sending more information to the actuators in each packet, one expects that fewer packets need to be transmitted to stabilize the system.

1.3.2 Exponential Stability of MIMO NCSs

In this section, we provide exponential stability conditions for the linear impulsive system (1.11) that can model both one-channel and two-channel NCSs with anticipative or nonanticipative controller, as described in Sect. 1.3.1.

Since the minimum of delay in the network is typically small, for simplicity of derivations, we assume that $\tau_{i\,\min} = 0$, $1 \leq i \leq m$. We now present two theorems for the stability of the system (1.11). These stability tests are generalizations of the result in Theorem 1. The first theorem is less conservative; however, the number of LMIs grows exponentially with m. The second stability condition is based on the feasibility of a single LMI, but its size grows only linearly with m. For small m, the first stability test is more adequate because it leads to less conservative results, but the second stability test is more desirable for large m. We present our results for $m = 2$, but deriving the stability conditions for other values of m is straightforward by following the same steps.

Inspired by the Lyapunov functional (1.5), we employ the Lyapunov functional

$$V := V_1 + V_2 + V_3 + V_4, \tag{1.20}$$

$$V_1 := x' P x,$$

$$V_2 := \sum_{i=1}^{2} \int_{t-\rho_i}^{t} (\rho_{i\,\max} - t + s) \dot{y}_i'(s) R_{1i} \dot{y}_i(s) \mathrm{d}s,$$

$$V_3 := \sum_{i=1}^{2} \int_{t-\sigma_i}^{t} (\sigma_{i\max} - t + s)\dot{y}_i'(s) R_{2i} \dot{y}_i(s) \mathrm{d}s,$$

$$V_4 := \sum_{i=1}^{2} (\rho_{i\max} - \rho_i)(y_i - w_i)' X_i (y_i - w_i),$$

with P, R_{1i}, R_{2i}, X_i symmetric positive definite matrices and

$$\rho_i(t) := t - s_k^i,\ \sigma_i(t) := t - t_k^i,\ w_i(t) := y_i(t_k^i), \quad t \in \left[t_k^i, t_{k+1}^i\right),$$
$$\rho_{i\max} := \sup_{t\geq 0} \rho_i(t), \quad \sigma_{i\max} := \sup_{t\geq 0} \sigma_i(t),$$

for $i = 1, 2$. The next theorem characterizes the admissible set $\mathscr{S}_i, i = 1, 2$, of sampling-delay sequences $(\{s_k^i\}, \{\tau_k^i\})$ such that

$$s_{k+1}^i + \tau_{k+1}^i - s_k^i \leq \rho_{i\max}, \qquad 0 \leq \tau_k^i \leq \tau_{i\max}, \quad \forall k \in \mathbb{N}. \tag{1.21}$$

Theorem 3. *The system* (1.11) *is exponentially stable over* $\mathscr{S}$ *defined by* (1.21)*, if there exist symmetric positive definite matrices* P, R_{1i}, R_{2i}, X_i *and (not necessarily symmetric) matrices* N_{1i}, N_{2i} *that satisfy the following LMIs:*

$$\begin{bmatrix} M_1 + \rho_{1\max}(M_{21} + M_{31}) + \rho_{2\max}(M_{22} + M_{32}) & \tau_{1\max} N_{11} & \tau_{2\max} N_{12} \\ * & -\tau_{1\max} R_{11} & 0_{q_1 q_2} \\ * & * & -\tau_{2\max} R_{12} \end{bmatrix} < 0, \tag{1.22a}$$

$$\begin{bmatrix} M_1 + \rho_{1\max} M_{21} + \rho_{2\max}(M_{22} + M_{32}) & \tau_{1\max} N_{11} & \tau_{2\max} N_{12} & G_{11} \\ * & -\tau_{1\max} R_{11} & 0 q_1 q_2 & 0_{q_1 q_1} \\ * & * & -\tau_{2\max} R_{12} & 0_{q_2 q_1} \\ * & * & * & G_{21} \end{bmatrix} < 0, \tag{1.22b}$$

$$\begin{bmatrix} M_1 + \rho_{1\max}(M_{21} + M_{31}) + \rho_{2\max} M_{22} & \tau_{1\max} N_{11} & \tau_{2\max} N_{12} & G_{12} \\ * & -\tau_{1\max} R_{11} & 0_{q_1 q_2} & 0_{q_1 q_2} \\ * & * & -\tau_{2\max} R_{12} & 0_{q_2 q_2} \\ * & * & * & G_{22} \end{bmatrix} < 0, \tag{1.22c}$$

$$\begin{bmatrix} M_1 + \rho_{1\max} M_{21} + \rho_{2\max} M_{22} & \tau_{1\max} N_{11} & \tau_{2\max} N_{12} & G_{11} & G_{12} \\ * & -\tau_{1\max} R_{11} & 0_{q_1 q_2} & 0_{q_1 q_1} & 0_{q_1 q_2} \\ * & * & -\tau_{2\max} R_{12} & 0_{q_2 q_1} & 0_{q_2 q_2} \\ * & * & * & G_{21} & 0_{q_1 q_2} \\ * & * & * & * & G_{22} \end{bmatrix} < 0, \tag{1.22d}$$

where $\bar{F} := \begin{bmatrix} A & B & 0_{nq} \end{bmatrix}$ *and*

$$\begin{aligned} M_1 &:= \bar{F}' \begin{bmatrix} P & 0_{nq} & 0_{nq} \end{bmatrix} + \begin{bmatrix} P \\ 0_{qn} \\ 0_{qn} \end{bmatrix} \bar{F} - T_1' X_1 T_1 - T_2' X_2 T_2 - N_{11} T_3 \\ &\quad - T_3' N_{11}' - N_{21} T_1 - T_1' N_{21}' - N_{12} T_4 - T_4' N_{12}' - N_{22} T_2 - T_2' N_{22}', \\ M_{2i} &:= \bar{F}' C_i' (R_{1i} + R_{2i}) C_i \bar{F}, \qquad M_{3i} := T_i' X_i C_i \bar{F} + \bar{F}' C_i' X_i T_i, \\ G_{1i} &:= \rho_{i\max} (N_{1i} + N_{2i}), \qquad G_{2i} := \rho_{i\max} (R_{1i} + R_{2i}) \end{aligned} \tag{1.23}$$

with

$$\begin{aligned} &C_1 := \begin{bmatrix} I_{q_1} & 0_{q_1 q_2} \end{bmatrix} C, \quad C_2 := \begin{bmatrix} 0_{q_2 q_1} & I_{q_2} \end{bmatrix} C, \quad T_1 := \begin{bmatrix} C_1 & 0_{q_1 q} & -\bar{I}_1 \end{bmatrix}, \\ &T_2 := \begin{bmatrix} C_2 & 0_{q_2 q} & -\bar{I}_2 \end{bmatrix}, \quad T_3 := \begin{bmatrix} C_1 & -\bar{I}_1 & 0_{q_1 q} \end{bmatrix}, \quad T_4 := \begin{bmatrix} C_2 & -\bar{I}_2 & 0_{q_2 q} \end{bmatrix}, \\ &\bar{I}_1 := \begin{bmatrix} I_{q_1} & 0_{q_1 q_2} \end{bmatrix}, \quad \bar{I}_2 := \begin{bmatrix} 0_{q_2 q_1} & I_{q_2} \end{bmatrix}. \end{aligned} \tag{1.24}$$

□

It is possible to generalize Theorem 3 for an arbitrary m. However, the number of LMIs that one needs to solve equal to 2^m and the size of LMIs and the number of scalar variables increases linearly, which limits the application of this result for large values of m. The next theorem, also provided in [11], presents another stability test, which is more conservative, but only requires the solution of a single LMI.

Theorem 4. *The system* (1.11) *is exponentially stable over* $\mathscr{S}_i$, $i = 1, 2$, *defined by* (1.21)*, provided that there exist symmetric positive definite matrices* P, R_{1i}, R_{2i} *and (not necessarily symmetric) matrices* N_{1i}, N_{2i} *that satisfy the following LMIs:*

$$\begin{bmatrix} \bar{M}_1 + \rho_{1\max} \bar{M}_{21} + \rho_{2\max} \bar{M}_{22} & \tau_{1\max} N_{11} & \tau_{2\max} N_{12} & G_{11} & G_{12} \\ * & -\tau_{1\max} R_{11} & 0_{q_1 q_2} & 0_{q_1 q_1} & 0_{q_1 q_2} \\ * & * & -\tau_{2\max} R_{12} & 0_{q_2 q_1} & 0_{q_2 q_2} \\ * & * & * & G_{21} & 0_{q_1 q_2} \\ * & * & * & * & G_{22} \end{bmatrix} < 0, \tag{1.25}$$

where $\bar{F} := \begin{bmatrix} A & B & 0_{nq} \end{bmatrix}$ *and*

$$\begin{aligned} \bar{M}_1 &:= \bar{F}' \begin{bmatrix} P & 0_{nq} & 0_{nq} \end{bmatrix} + \begin{bmatrix} P \\ 0_{qn} \\ 0_{qn} \end{bmatrix} \bar{F} - N_{11} T_3 - T_3' N_{11}' - N_{21} T_1 \\ &\quad - T_1' N_{21}' - N_{12} T_4 - T_4' N_{12}' - N_{22} T_2 - T_2' N_{22}', \\ \bar{M}_{2i} &:= \bar{F}' C_i' (R_{1i} + R_{2i}) C_i \bar{F}, \quad G_{1i} := \rho_{i\max} (N_{1i} + N_{2i}), \\ G_{2i} &:= \rho_{i\max} (R_{1i} + R_{2i}), \end{aligned}$$

with the matrix variables defined in (1.24)*.* □

By solving the LMIs in Theorems 3 or 4, one can find positive constants $\rho_{i\,\max}, i = 1, 2$ that determine upper bounds between the *consecutive samples of channel i* for which the stability of the closed-loop system is guaranteed, for a given lower and upper bound on the total delay in the loop i.

Most of the work in the literature has been devoted to finding a single constant τ_{MATI} ([7, 15, 16, 20, 21] and references therein) that provides an upper bound between *any consecutive sampling instances* for which the stability of the closed-loop system is guaranteed. It is thus not surprising that having m distinct constants $\rho_{i\,\max}$, $1 \leq i \leq m$ instead of one single constant τ_{MATI}, reveals more information about the system and permits less conservative results.

1.3.3 Example

This example appeared in [8,15,19,21] and considers a linearized model of the form (1.12) for a two-input, two-output batch reactor, where

$$A_p := \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}, \; B_p := \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix},$$

$$C_p := \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

This system is controlled by a PI controller of the form (1.13), where

$$A_c := \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B_c := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C_c := \begin{bmatrix} -2 & 0 \\ 0 & 8 \end{bmatrix}, \quad D_C := \begin{bmatrix} 0 & -2 \\ 5 & 0 \end{bmatrix}.$$

Following the assumptions of [8, 15, 19, 21], we assume that only the outputs of the plant are transmitted over the network, there are no dropouts and the outputs are sent in a round-robin fashion and consecutively. We compare τ_{MATI} of this example given by the stability results in [8, 15, 19, 21], where the effect of the delay was ignored. From Theorem 3, we compute $\rho_{1\,\max} = 0.081, \rho_{2\,\max} = 0.113$ when there is no delay. We can also show that if the upper bound between any consecutive sampling, τ_{MATI}, is smaller than $\frac{1}{2}\min(\rho_{1\,\max}, \rho_{2\,\max}) = .0405$, then the upper bound between the samples of y_{p1} or y_{p2} are smaller than $\min(\rho_{1\,\max}, \rho_{2\,\max})$ and the system is stable. Table 1.1 shows the less conservative results in the literature and our τ_{MATI} for comparison. Note that the value of τ_{MATI} for a (stochastic) uniform intersampling time distribution given by [8] is less conservative than τ_{MATI} given by Theorem 3. However, for a fair comparison our result should be compared to the stochastic arbitrary intersampling time distribution given by [8]. If we can send the measurements of y_{1p} and y_{2p} in one packet, then $\tau_{\text{MATI}} = \min(\rho_{1\,\max}, \rho_{2\,\max}) = 0.081$, because

Table 1.1 Comparison of τ_{MATI} for example 1.3.3 when there is no delay

	No drop
Maximum deterministic time interval between samples from [19]	0.0123
Maximum stochastic arbitrary intersampling time distribution from [8]	0.0279
Maximum stochastic uniform intersampling time distribution from [8]	0.0517
Maximum deterministic time interval between samples from Theorem 3	0.0405
Maximum deterministic time interval between samples from Theorem 4	0.029

the requirement for the stability that the consecutive samplings of y_{1p} and y_{2p} are smaller than $\rho_{1\,\max}$ and $\rho_{2\,\max}$, respectively. As expected, in the presence of delays the value of τ_{MATI} decreases and, for example, when the maximum delay is 0.03, then $\rho_{1\,\max} = 0.058$ and $\rho_{2\,\max} = 0.087$.

1.4 Wireless NCS QoS

The framework that we have developed so far provides conditions to guarantee QoP of the control system in terms of exponential stability. We now focus on the QoS that the network should provide to obtain the desired QoP at the application layer.

Consider again the system depicted in Fig. 1.1. We assume that the overall system consists of ℓ connections, where a *connection* is a path between a sampler and its corresponding hold. From the results in previous sections, we can find constants $\gamma_{i\,\max}, \tau_{i\,\max}, i \in \{1, 2, ...\}$ such that[1]

$$s^i_{k+1} - s^i_k \leq \gamma_{i\,\max}, \quad \tau^i_k \leq \tau_{i\,\max}, \tag{1.26}$$

for which exponential stability of all subsystems is guaranteed. Suppose now that the upper bounds on the sampling intervals of all connections, $\gamma_{i\,\max}, i \in \{1, \cdots \ell\}$, are given, then using Theorem 1 or Theorem 2 for the SISO case and Theorem 3 or 4 for the MIMO case (with $\tau_{i\,\min} = 0$), we can find upper bounds $\tau_{i\,\max}$ for the total delay in each connection so that exponential stability can be guaranteed. The main question addressed in this section is whether or not the network can deliver all packets for all connections before their *deadlines* $\tau_{i\,\max}$.

To answer this question, we will employ results in real-time scheduling [4]. In real-time scheduling, different jobs are released periodically or lower bounds between release times are given. In the most basic setting, one shared resource services different jobs and servicing a job takes a certain amount of time. Each job should be completed before a deadline and if all the timing requirements can be met, then the set of jobs is said to be *schedulable*. In the context of real-time computation,

[1] The results in this chapter require explicit bounds on the sampling intervals. Hence, we use the definition of sampling-delay sequences as in (1.4) for SISO and MIMO cases. Note that $\rho_{i\,\max} = \gamma_{i\,\max} + \tau_{i\,\max}$. Moreover, for simplicity we assume the lower bound on the delay is zero.

typically a processor is a shared computation resource and jobs are corresponds to computing tasks. While computation resources can also be shared in NCSs; for example, a processor can be used to implement two controllers [9], this will not pursue here and we assume that the computation delay is negligible. In the context of NCSs, the shared resource typically is a network shared between different nodes.

Job i refers to transmitting a packet from the source to the destination of connection i and the time required to service a job is the time needed to transfer a packet, which we call the transmission time. Suppose that $\gamma_{i\,\max}, \tau_{i\,\max}$ are given such that the stability LMIs are satisfied. If the set of "jobs" are schedulable with deadlines $\tau_{i\,\max}$, and release times greater than $\gamma_{i\,\min}$ for every $i \in \{1, \cdots, \ell\}$, then the completion of the job i is guaranteed before $\tau_{i\,\max}$. Hence, the corresponding sampling-delay sequence satisfies (1.26) and consequently stability of all subsystems connected to the network is guaranteed.

1.4.1 Types Of Real-Time Scheduling

Two main types of scheduling can be found in the literature: nonpreemptive and preemptive. In nonpreemptive scheduling, a running service will not be interrupted to service a higher priority job. On the contrary, in preemptive scheduling, as soon as a job with a higher priority is released, the shared resource is allocated to the higher priority job and the current job with lower priority is interrupted. Preemptive scheduling is suitable for computation resource sharing, but cannot be used on communication networks for which access to the network cannot be granted to a new transmission until the current transmission is completed.

There are two main priority assignment schemes: static and dynamic. A static priority is fixed and set a priori, so it can be stored in a table. Static scheduling is simple, yet it is very inflexible to changes, failures, and often it underutilizes the shared resources [4]. When scheduling decisions are based on the current decision variables, we have what is called a dynamic scheduling. Dynamic priority assignment is more difficult to implement because priorities change over time and they should be computed online; however, a dynamic priority assignment is generally more flexible and efficient. In the following, we summarize the most common scheduling policies and we refer the readers to [4] for more details.

First-Come First-Serve (FCFS) scheduling. This policy serves the oldest request first so that resource allocation is based on the order of request arrivals. This policy is generally not suitable for control application because it may serve a job with longer deadline over a job with shorter deadline.

Round-Robin (RR) scheduling. This is a static algorithm in which a fixed time slot is dedicated to each node. This policy is simple and effective when: all nodes are synchronized, they have data most of the time, and the network structure is fixed so that no new node joins after the time slots are assigned to the nodes. If, for any reason, a node loses its turn, no matter how close its deadline is, it should wait until its next allocated slot.

Deadline Monotonic (DM) scheduling. This *static* policy allocates the resource to nodes according to their deadlines. A task with the *shortest deadline*, (smallest $\tau_{i\,\max}$) is assigned the highest priority. For example if $\tau_{1\,\max} = 3$ and $\tau_{2\,\max} = 4$, then jobs of source one will always have higher priority over jobs of source two.

Earliest Deadline First (EDF) scheduling. EDF is a dynamic algorithm that assigns priorities to jobs according to their *absolute deadlines*, which are the times remaining to miss the deadline. A job with the earliest absolute deadline, $(t_{il} + \tau_{i\,\max} - t)$ will have the highest priority, where t_{il} is the last sampling time of connection i and t is the current time. Again, consider job one with $\tau_{1\,\max} = 3, t_{1l} = 2$ and job two with $\tau_{2\,\max} = 4, t_{2l} = 0$, which means that job one must be completed before time 5 and job two must be completed before time 4. In this case, if both nodes have a job ready to be service at time 3, then job two will be served because its absolute deadline is smaller (in spite of the fact that its $\tau_{i\,\max}$ is larger).

Each of these scheduling policies can be easily implemented on computation resources, but scheduling policies for shared communication resources depend on the specific network. FCFS is not easily implementable on wireless networks; however, RR, DM, and EDF are implementable on particular wireless networks as we shall see in Sect. 1.5. Of all the policies discussed, the most desirable policy is EDF because, as a dynamic algorithm, it is most adaptable to network conditions. The disadvantage of EDF is that the priorities are a function of time and should be updated periodically, which require spending additional computation [11].

1.4.2 Scheduling Test To Guarantee QoS

The core of scheduling analysis is a scheduling test, which determines whether a particular scheduling policy can guarantee that the tasks will be serviced, even under the worst case condition. When this happens, we say that the tasks are *schedulable under the policy*. Our focus here will be on EDF scheduling. The deadline to finish job $i \in \{1, \cdots, n\}$ is denoted by D_i, the lower bound between consecutive job release times is denoted by T_i, and the time to service job i is denoted by C_i. If the conditions in the next theorem hold for a given set of jobs, then the set of jobs is schedulable under EDF policy. This means that the maximum delay experienced by job i, from the time it is released to the resource until the time that it is serviced, is always smaller than its deadline D_i. This delay consists of the wait time until the jobs gets service plus the service time. Note that the wait time depends greatly on the scheduling policy.

Theorem 5 ([26]). *A set of connections* $(T_i, C_i, D_i), i \in \{1, \cdots, n\}$ *is schedulable over a network under a (nonpreemptive) EDF scheduling policy if and only if the following inequalities hold:*

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1, \tag{1.27}$$

$$\sum_{i=1}^{n} \left\lfloor \frac{t - D_i}{T_i} \right\rfloor^{+} C_i + C_{\max} \leq t, \qquad \forall t \in \cup_{i=1}^{n} S_i, \tag{1.28}$$

where $C_{\max} := \max_i C_i$

$$S_i := \left\{ D_i + hT_i : h = 0, 1, \cdots, \left\lfloor \frac{d_{\max} - d_i}{T_i} \right\rfloor \right\},$$
$$d_{\max} := \max \left\{ D_1, \cdots, D_n, \frac{\sum_{i=1}^{n} (1 - D_i/T_i) C_i + C_{\max}}{1 - \sum_{i=1}^{n} C_i/T_i} \right\},$$

$\lfloor . \rfloor$ *is a floor function,* $\lfloor x \rfloor^{+} := \lfloor x + 1 \rfloor$ *for* $x \geq 0$ *and zero otherwise.* □

To apply the results in Sects. 1.2 or 1.3 to our problem, we associate a job i with a packet transmission in connection i. We set $D_i = \tau_{i\,\max}$ to be the deadline for the packets in connection i, $T_i = \gamma_{i\,\min}$ to be the minimum time between consecutive transmissions, and C_i to be the transmission time in that same connection.

Corollary 1. *Assume that the sampling intervals satisfy* $\gamma_{i\,\min} \leq s^i_{k+1} - s^i_k \leq \gamma_{i\,\max}$, *and* (1.27)–(1.28) *hold for* $D_i = \tau_{i\,\max}$, $T_i = \gamma_{i\,\min}$, *and* C_i *equal to the transmission time of the connection* i. *If* $\gamma_{i\,\max}$ *and* $\tau_{i\,\max}$ *satisfy the stability conditions in Sect. 1.2 or 1.3 (where* $\rho_{i\,\max} = \gamma_{i\,\max} + \tau_{i\,\max}$*), then all subsystems connected to the network are exponentially stable.*

This corollary formally verifies the design specification that end-to-end delays must be smaller than $\tau_{i\,\max}$. Without this type of scheduling analysis, one has to rely on extensive testing to find rare events that may destabilize the system.

1.5 Implementation Considerations

In this section, we discuss the implementation of EDF scheduling on wireless NCSs. In particular, we consider wireless LAN networks governed by the IEEE 802.11 set of standards. These standards use a distributed coordination function (DCF) or a point coordination function (PCF) for Medium Access Protocol (MAC). Based on CSMA/CA protocol, DCF uses random backoff in the event of a collision. In wireless NCSs, short and periodic packets are sent frequently, so DCF is not suitable since its throughput is high only with light bursty traffic. However, PCF can be implemented to ensure high throughput by a polling mechanism to eliminate collisions. A node called the Access Point (AP), grants permission to nodes to transmit. There are three priority levels of mixed traffic [23]:

- **Level 1.** Contains time critical aperiodic data that is bursty and cannot bear any loss. Retransmission is required when the former transmission is unsuccessful.
- **Level 2.** Includes time critical periodic data that can tolerate some loss.

- **Level 3.** Consists of noncritical data, which require data integrity, i.e., no loss is allowed. Retransmission is always implemented when there is data loss.

Level 1 data has the highest and level 3 has the lowest priority. NCS measurement and control packets are level 2 data and the focus of this paper. For simplicity, we assume that all data is level 2; however, the extension to the general case is straightforward. For implementation, Ye et al. [23] proposed a Centrally Maintained Polling Table (CMPT) to poll stations to schedule the level 2 data. The table maintains globally known network induced errors[2] for each connection, which is defined as $e_i := z_i(t) - y_i(t)$. The CMPT knows $z_i(t)$ since it was broadcasted on the network, but not $y_i(t)$. The authors proposed to estimate $y_i(t)$ if $t \neq s_k^i$, $\forall k \in \mathbb{N}$ and otherwise update it with the true broadcasted value. Ye et al. [23] proposed several algorithms to handle scheduling and grant node permission to send data.

The implementation of our method can be similar to the method proposed by Ye et al. [23]; however, instead of error, our method employs *timers*. For each connection i, we employ a timer r_i that keeps track of how much time has elapsed since the last time permission was granted to node i and transmission was successful. Note that since the receiver node confirms a successful transmission by sending acknowledgment (ACK), the AP node is aware of possible packet dropout. As soon as the AP node grants permission to a node, its corresponding timer is reset to zero. The timer values are maintained in CMPT and are globally known. When the AP senses that the wireless network is idle, it gives permission to the node whose timer is closest to its deadline $\gamma_{i\,\max}$ (if $r_i \geq \gamma_{i\,\min}$).

In case of packet dropout, the AP does not get an ACK packet from the destination node, and again grants the permission to the same node. The only required assumption is that the network designer must know an upper bound on the maximum number of consecutive packet dropouts in the network. With that knowledge, it is possible to find the upper bound for the sampling intervals (see Remark 1 and also the example in Sect. 1.5.1).

1.5.1 Example

We consider the example of a motion control system for sheet control in a printer paper path from [3]. The system consists of several pinches or rollers, driven by motors, to move papers through the printer. Motor controllers are implemented on the AP node and the position and velocity measurements are sent through a wireless LAN network. The motors are directly connected to the AP node. Each subsystem (a single motor-roller pair) can be modeled as $\ddot{x}_s = \frac{nr_P}{J_M + n^2 J_P} u$, where $J_M = 1.95 \times 10^5$ kg/m^2 is the inertia of the motor, $J_P = 6.5 \times 10^5$ kg/m^2 is the inertia of

[2] In Sect. 1.3, we defined $z_i := y_i(s_k^i)$, $t \in [t_k^i, t_{k+1}^i)$, but since Ye et al. [23] assume that the delay is negligible, this would correspond to $z_i(t) := y_i(s_k^i)$, $t \in [s_k^i, s_{k+1}^i)$.

the pinch, $r_P = 14 \times 10^{-3}$ m is the radius of the pinch, $n = 0.2$ is the transmission ratio between motor and pinch and u is the motor torque. Each subsystem can be presented with the state space of the form $\dot{x} = Ax + B_u u$ with

$$x = \begin{bmatrix} x_s \\ \dot{x}_s \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ \frac{nr_P}{J_M + n^2 J_P} \end{bmatrix}. \tag{1.29}$$

We use the state feedback control $K = -\begin{bmatrix} 50 & 1.18 \end{bmatrix}$ to control the motors. We assume that for each subsystem, the AP node needs 0.1 ms to read a measurement packet from its buffer, decode the data, calculate the control command, and apply it to the motor. Moreover, we assume that it takes 1 ms between the AP consecutive sent permissions to nodes (for this example, sending ACK is not necessary because the receiver of all measurement packets is the AP node itself and it knows if the data was dropped or corrupted).

In traditional control system design, one often ignores the effect of network delays and selects a constant sampling times that are sufficiently fast so that sampling can be ignored. By checking the condition

$$eig\left(\begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix} \mathrm{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} h} \right) < 1, \quad B := B_u \times K, \tag{1.30}$$

on a tight grid of h, we can show that the closed-loop system remains stable for any *constant* sampling interval smaller than 48 ms, and becomes unstable for larger constant sampling intervals. So a designer who follows traditional design guidelines may choose the sampling interval equal 12 ms, which is 4 times faster than the threshold beyond which stability is lost. A key question at this point is: how many motors can be controlled given this architecture. The answer to this question depends on the designer experience and judgment. A conservative designer would choose $n = 6$ to guarantee a total bus load equal to 50% $\left(\text{the total bus load is defined as } \sum_i \frac{C_i}{T_i} \times 100\%\right)$, whereas an aggressive designer would choose up to $n = 11$ so that the bus load remains strictly lower than 100% (91.7% in this case).

In the following, we present our proposed systematic design process. The closed loop subsystems can be modeled as a SISO delay impulsive system given by (1.2) with A, B defined in (1.29) and (1.30). Then we determine the set of pairs $(\gamma_{i\,\max}, \tau_{i\,\max})$ for which the system would be exponentially stable, based on Theorem 1. This set is shown in Fig. 1.8. To compare with the first approach, we choose the sampling times constant and equal to 12 ms.

We consider two scenarios. First, we assume there is no packet dropout so $\gamma_{i\,\min} = \gamma_{i\,\max} = 12$ ms. Based on Fig. 1.8, for this choice, stability of the subsystems are guaranteed for delays smaller than 10 ms. At the scheduling level, we determine how many subsystems can share the network so that the total delay in each

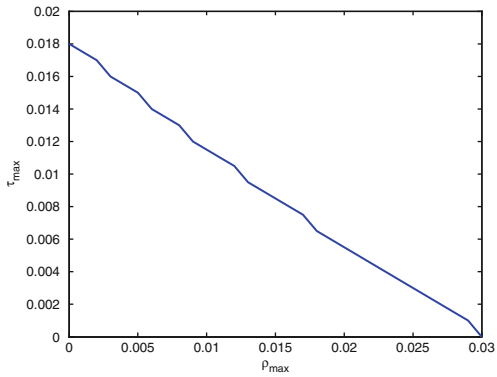


Fig. 1.8 Admissible set of variable sampling-delay sequences for a single motor-pinch subsystem is any sampling interval and delay sequence that belong to the triangle consists of the horizontal and vertical axis and the blue line

loop remains smaller than 10 ms. To do so, we test the conditions in Theorem 5 with $T_i = 12, C_i = 1, D_i = 10-0.1 = 9.9$ for different values of n. It turns out that the conditions are satisfied for up to $n = 9$. This result indicates that 9 pinch-motor subsystems can share the network while the stability of all subsystems is guaranteed. Note that for $n = 10, 11$ the delay can be larger than 10 ms for some corner cases that may not be easily captured by simulation and testing (the worse case delay occurs when all the sensors send data at the same time). By following the proposed design procedure, we can avoid very conservative choices (e.g., $n = 6$) or choices that lead to unsafe behavior of the subsystems.

In a second scenario, we assume that at most 3 consecutive packet dropout are possible. For this case $\gamma_{i\,\min} = 12$, and $\gamma_{i\,\max} = 15$. Based on the analysis results depicted in Fig. 1.8, for this choice, stability of the subsystems are guaranteed for delays smaller than 8 ms. By testing the conditions in Theorem 5 with $T_i = 15$, $C_i = 1, D_i = 8-0.1 = 7.9$ for different values of n, it turns out that the conditions are satisfied for up to $n = 7$.

1.6 Conclusions and Future Work

We showed that delay impulsive systems are a natural framework to model wired or wireless NCSs with variable sampling intervals and delays and possible packet dropouts. We employed discontinuous Lyapunov functionals to characterize admissible sampling intervals and delays such that exponential stability of wireless NCS is guaranteed. We defined exponential stability as a minimal QoP, and we found the requirements to guarantee QoP at the application level of wireless NCSs. Then we provided a set of conditions for EDF scheduling, that if satisfied, ensures the desired QoS for the wireless network required to provide QoP. We also discussed considerations to implement EDF scheduling (or other dynamic scheduling policies), on a wireless network.

An important topic for future research is the controller and network codesign. The framework we developed for the analysis of wireless NCS can provide the foundation for the codesign of the network and controller. It is also important to consider other QoP metrics such as robust design as measured by the H_∞ norm or the H_2 norm. In general, faster sampling improves QoP of the control systems; however, QoS of the network may decrease due to higher traffic of the network. The tradeoff between the performance of the control systems at the application layer and the behavior at other layers of wireless NCS is another important topic.

References

1. T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proc. of the 13th Mediterranean Conf. on Control and Automation*, volume 3, pages 719–724, 2005.
2. M. S. Branicky, S. M. Phillips, and W. Zhang. Stability of networked control systems: Explicit analysis of delay. In *Proc. of the 2000 Am. Contr. Conf.*, volume 4, pages 2352–2357, June 2000.
3. M. Cloosterman, N. van de Wouw, W. Heemels, and H. Nijmeijer. Robust stability of networked control systems with time-varying network-induced delays. In *Proc. of the 45th Conf. on Decision and Contr.*, Dec. 2006.
4. F. Cottet, J. Delacroix, C. Kaiser, and Z. Mammeri. *Scheduling in real-time systems*. Willy, 2002.
5. E. Fridman. A new Lyapunov technique for robust control of systems with uncertain non-small delays. *J. Math. Contr. Info.*, 22(3), Nov. 2005.
6. E. Fridman, A. Seuret, and J. P. Richard. Robust sampled-data stabilization of linear systems: An input delay approach. *Automatica*, 40(8):1441–1446, Aug. 2004.
7. J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. of the IEEE*, 95(1):138–162, Jan. 2007.
8. J. P. Hespanha and A. R. Teel. Stochastic impulsive systems driven by renewal processes. *Proc. of the 17th Int. Symp. on the Mathematical Theory of Networks and Syst.*, pages 606–618, July 2006.
9. F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks. *IEEE Contr. Syst. Mag.,* 21(1):66–83, Feb. 2001.
10. X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proc. of the 43th Conf. on Decision and Contr.*, volume 3, pages 2823– 2829, Dec. 2005.
11. P. Naghshtabrizi. *Delay impulsive systems: A framework for modeling networked control systems*. PhD thesis, University of California at Santa Barbara, 2007.
12. P. Naghshtabrizi and J. P. Hespanha. Designing observer-based controller for network control system. In *Proc. of the 44th Conf. on Decision and Contr.*, volume 4, pages 2876–2880, June 2005.
13. P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel. On the robust stability and stabilization of sampled-data systems: A hybrid system approach. In *Proc. of the 45th Conf. on Decision and Contr.*, pages 4873–4878, 2006.
14. N. Navet, Y. Song, F. Simonot-Lion, and C. Wilvert. Trends in automotive communication systems. *Proc. of the IEEE*, 93(6):1204–1223, June 2005.
15. D. Nesic and A. R. Teel. Input-output stability properties of networked control systems. *IEEE Trans. on Automat. Contr.*, 49(10):1650–1667, Oct. 2004.
16. D. Nesic and A. R. Teel. Input-to-state stability of networked control systems. *Automatica*, 40 (12):2121–2128, Dec. 2004.

17. P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. Automat. Contr.*, 49 (8):1292–1302, Aug. 2004.
18. P. Seiler and R. Sengupta. An H_∞ approach to networked control. *IEEE Trans. Automat. Contr.*, 50(3):356–364, Mar. 2005.
19. M. Tabbara, D. Nesic, and A. R. Teel. Input- output stability of wireless networked control systems. In *Proc. of the 44th Conf. on Decision and Contr.*, pages 209–214, Dec. 2005.
20. G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. In *Proc. of the 1999 Am. Contr. Conf.*, volume 4, pages 2876–2880, June 1999.
21. G. C. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Trans. Contr. Syst. Tech.*, 10(3):438–446, May 2002.
22. F. Xia, L. Ma, C. Peng, Y. Sun, and J. Dong. Cross-layer adaptive feedback scheduling of wireless control systems. 8(7):4265–4281, 2008.
23. H. Ye, G. C. Walsh, and L. G. Bushnell. Real-time mixed -traffic wireless networks. *IEEE Trans. Industrial Electronics.*, 48(5):883–890, Oct. 2001.
24. D. Yue, Q.-L. Han, and C. Peng. State feedback controller design for networked control systems. *IEEE Trans. Automat. Contr.*, 51(11):640–644, Nov. 2004.
25. D. Yue, Q. L. Han, and J. Lam. Network-based robust H_∞ control of systems with uncertainty. *Automatica*, 41(6):640–644, June 2005.
26. Q. Zheng and K. G. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Trans. Communications*, 42(2/3/4):1096–1105, Feb. 1994.

Chapter 2
State Estimation Over an Unreliable Network

Ling Shi, Lihua Xie, and Richard M. Murray

Abstract In this chapter, we consider Kalman filtering over a packet-delaying network. Given the probability distribution of the delay, we can completely characterize the filter performance via a probabilistic approach. We assume the estimator maintains a buffer of length D so that at each time k, the estimator is able to retrieve all available data packets up to time $k - D + 1$. Both the cases of sensor with and without necessary computation capability for filter updates are considered. When the sensor has no computation capability, for a given D, we give lower and upper bounds on the probability for which the estimation error covariance is within a prescribed bound. When the sensor has computation capability, we show that the previously derived lower and upper bounds are equal to each other. An approach for determining the minimum buffer length for a required performance in probability is given and an evaluation on the number of expected filter updates is provided.

Keywords Kalman filter · Networked control systems · Packet-delaying networks · Estimation theory · Probabilistic performance

2.1 Introduction

The Kalman filter has played a central role in systems theory and has found wide applications in many fields, such as control, signal processing, and communications. In the standard Kalman filter, it is assumed that sensor data are transmitted along perfect communication channels and are available to the estimator either instantaneously or with some fixed delays, and no interaction between communication and control is considered. This abstraction has been adopted until recently when networks, especially wireless networks, are used in sensing and control systems for

L. Shi (✉)
Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
e-mail: eesling@ust.hk

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_2,

transmitting, data from sensor to controller and/or from controller to actuator. While having many advantages such as low cost and flexibility, networks also induce many new issues due to their limited capabilities and uncertainties, such as limited bandwidth, packet losses, and latency. On the other hand, in wireless sensor networks, sensor nodes also have limited computation capability in addition to their limitations in communications. These constraints undoubtedly affect system performance or even stability and cannot be neglected when designing estimation and control algorithms, which has inspired a lot of research in control with communication constraints; see the survey [3] and the references therein.

In recent years, networked control problems have gained much interest. In particular, the state estimation problem over a network has been widely studied. The problem of state estimation and stabilization of a linear time-invariant system over a digital communication channel, which has a finite bandwidth capacity was introduced by Wong and Brockett [19, 20] and further pursued by others (e.g., in [1, 7, 10, 17]). Sinopoli et al. [15] discussed how packet loss can affect state estimation. They showed there exists a certain threshold of the packet loss rate above which the state estimation error diverges in the expected sense, i.e., the expected value of the error covariance matrix becomes unbounded as time goes to infinity. They also provided lower and upper bounds of the threshold value. Following the spirit of [15], Liu and Goldsmith [5] extended the idea to the case, where there are multiple sensors and the packets arriving from different sensors are dropped independently. They provided similar bounds on the packet loss rate for a stable estimate, again in the expected sense. Huang and Dey [4], Xie and Xie [21] characterize packet losses as a Markov chain and give some sufficient and necessary stability conditions under the notion of peak covariance stability. The drawback of using mean covariance matrix as a stability measure is that it may conceal the fact that events with arbitrarily low probability may make the mean value diverge. Different from [4, 15, 21], Shi et al. [13] investigate the stability of the Kalman filter through a probabilistic approach.

The problem of state estimation and control with delayed measurements is not new and has been studied even before the emergence of networked control [11, 22]. Nilsson [9] analyzed delays that are either fixed or random according to a Markov chain. He solves the LQG optimal control problem for the different delay models. It has been well known that discrete-time systems with constant or known time-varying bounded measurement delays may be handled by state augmentation in conjunction with the standard Kalman filtering or by the reorganized innovation approach in [23, 24]. Although sensor data are usually time-stamped and thus transmission delays are known to the filter, the delays in networked systems are random in nature. For example, the ZigBee/IEEE 802.15.4 protocol is widely used in sensor network and wireless control applications [25]. When multiple sensor nodes simultaneously access the channel, a random waiting time is generated by the CSMA/CA algorithm for each node before they try to access the channel again. Thus, the experienced delay for data measurement is typically random.

For the problem of randomly delayed measurements, Ray et al. [11] present a modification of the conventional minimum variance state estimator to accommodate

the effects of the random arrival of measurements, whereas a suboptimal filter in the least mean square sense is given in [22]. In [6], a recursive minimum variance state estimator is presented for linear discrete-time partially observed systems, where the observations are transmitted by communication channels with randomly independent delays. Using covariance information, recursive least-squares linear estimators for signals with random delays are studied in [8]. Furthermore, the filtering problems with random delays and missing measurements have been investigated in [12, 16, 18] via the linear matrix inequality and the Riccati equation approaches, respectively. Note that most of the aforementioned work is concerned with the optimal or suboptimal average design, where the mean filtering error covariance is taken with respect to a random i.i.d. variable that characterizes the random delay in addition to the process and measurement noises and the initial state. Thus, the derived filter is in fact suboptimal when the delay is known online. There has been no systematic analysis on the performance of the Kalman filter, which offers the optimal filtering performance for systems with random measurement delay available online.

The goal of this chapter is to study the performance of Kalman filter under random measurement delay. We assume that the probability distribution of the delay is given and aim to give a complete characterization of filter performance. Due to the limited computation capability of the filtering center and also in consideration of the fact that a late arriving measurement related to the system state in the far past may not contribute much to the improvement of the accuracy of the current estimate, it is practically important to determine a proper buffer length for measurement data within which a measurement will be used to update the current state and beyond which the data will be discarded. The buffer provides a tradeoff between performance and computational load. In the chapter, for a given buffer length, we shall give lower and upper bounds for the probability at which the filtering error covariance is within a prescribed bound, i.e., $\mathbf{Pr}[P_k \leq M]$ for some given M. The upper and lower bounds can be easily evaluated by the probability distribution of the delay and the system dynamics. An approach for determining the minimum buffer length for a required performance in probability is given and an evaluation on the number of expected filter updates is provided. Both the cases of sensor with and without necessary computation capability for filter updates are considered.

2.2 Problem Setup

We consider the problem of state estimation over a packet-delaying network as seen from Fig. 2.1. The process dynamics and sensor measurement equation are given as follows:

$$x_{k+1} = Ax_k + w_k, \tag{2.2.1}$$

$$y_k = Cx_k + v_k. \tag{2.2.2}$$

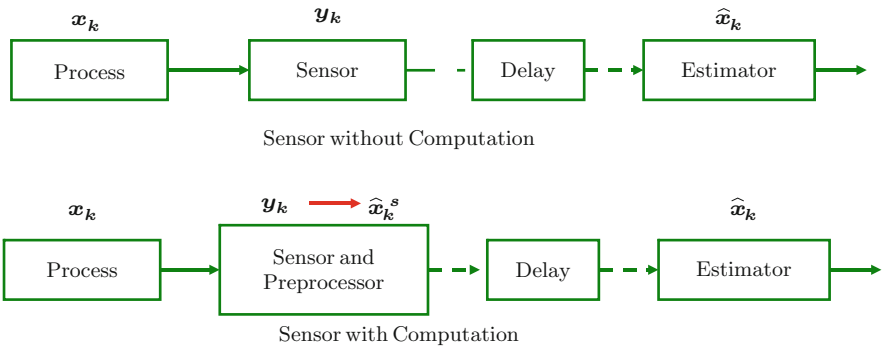


Fig. 2.1 System block diagram

In the above equations, $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^m$ is the observation vector, $w_{k-1} \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are zero mean white Gaussian random vectors with $\mathbb{E}[w_k w_j{}'] = \delta_{kj} Q$, $Q \geq 0$, $\mathbb{E}[v_k v_j{}'] = \delta_{kj} R$, $R > 0$, $\mathbb{E}[w_k v_j{}'] = 0 \; \forall j, k$, where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ otherwise. We assume that the pair (A, C) is observable, and $(A, \sqrt{Q})$ is controllable.

Depending on its computational capability, the sensor can either send y_k or preprocess y_k and send $\hat{x}_k^s$ to the remote estimator, where $\hat{x}_k^s$ is defined at the sensor as

$$\hat{x}_k^s \triangleq \mathbb{E}[x_k | y_1, \ldots, y_k].$$

The two cases correspond to the two scenarios in Fig. 2.1, i.e., sensor without/with computation capability.

After taking a measurement at time k, the sensor sends y_k (or $\hat{x}_k^s$) to a remote estimator for generating the state estimate. We assume that the measurement data packets from the sensor are to be sent across a packet-delaying network, with negligible quantization effects, to the estimator. Each y_k (or $\hat{x}_k^s$) is delayed by d_k times, where d_k is a random variable described by a probability mass function f, i.e.,

$$f(j) = \mathbf{Pr}[d_k = j], \; j = 0, 1, \ldots \tag{2.2.3}$$

For simplicity, we assume d_{k_1} and d_{k_2} are independent if $k_1 \neq k_2$. We further assume that d_k carries no information about the state, e.g., d_k is independent of w_k, v_k and the initial state x_0. Similar to [14], we can use a Markov chain to model consecutive data packet delays, and the results extend straightforward to that case. Note that the i.i.d packet drop with drop rate $1 - \gamma$ considered in the literature can be treated as a special case here, i.e.,

$$f(0) = \gamma, \; f(\infty) = 1 - \gamma, \; f(j) = 0, 1 \leq j < \infty.$$

Thus, the theory developed here includes the packet drop analysis as well.

Define the following state estimate and other quantities at the remote estimator

$$\begin{aligned}
\hat{x}_k^- &\triangleq \mathbb{E}[x_k | \text{all data packets up to } k-1], \\
\hat{x}_k &\triangleq \mathbb{E}[x_k | \text{all data packets up to } k], \\
P_k^- &\triangleq \mathbb{E}\left[\left(x_k - \hat{x}_k^-\right) \left(x_k - \hat{x}_k^-\right)' | \text{all data packets up to } k-1 \right], \\
P_k &\triangleq \mathbb{E}\left[\left(x_k - \hat{x}_k\right) \left(x_k - \hat{x}_k\right)' | \text{all data packets up to } k \right].
\end{aligned}$$

Assume the estimator discards any data y_k (or $\hat{x}_k^s$) that are delayed by D times or more. Given the system and the network delay models in (2.2.1)–(2.2.3), we are interested in the following problems.

1. How should $\hat{x}_k$ be computed?
2. What is the relationship between P_k and D?
3. For a given $M \geq 0$ and $\epsilon \in [0, 1]$, what is the minimum D such that

$$\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon.$$

In the rest of the chapter, we provide solutions to the above three problems for each of the two scenarios in Fig. 2.1.

The following terms that are frequently used in subsequent sections are defined below. It is assumed that (A, C, Q, R) are the same as they appear in Sect. 2.2; $\lambda_i(A)$ is the ith eigenvalue of the matrix A; $X \in \mathbb{S}_+^n$ where $\mathbb{S}_+^n$ is the set of n by n positive semidefinite matrices; $h, g : \mathbb{S}_+^n \to \mathbb{S}_+^n$ are functions defined below; Y_i is a random variable, where the underlying sample spaces will be clear from its context.

$$\begin{aligned}
h(X) &\triangleq AXA' + Q \\
g(X) &\triangleq h(X) - AXC'[CXC' + R]^{-1}CXA' \\
\tilde{g}(X) &\triangleq X - XC'[CXC' + R]^{-1}CX \\
h \circ g(X) &\triangleq h(g(X)) \\
h^t(X) &\triangleq \underbrace{h \circ \cdots \circ h}_{t \text{ times}}(X) \\
\mathbf{Pr}[Y_1 | Y_2] &\triangleq \mathbf{Pr}[Y_1] \text{ given } Y_2
\end{aligned}$$

2.3 Sensor Without Computation Capability

In this section, we consider the first scenario in Fig. 2.1, i.e., the sensor has no computation and sends y_k to the remote estimator. We assume C is full rank, and without loss of generality, we assume C^{-1} exists. The general C case will be considered in Sect. 2.4.1.

2.3.1 Modified Kalman Filtering

Let γ_t^k be the indicator function for y_t at time $k, t \leq k$, which is defined as follows.

$$\gamma_t^k = \begin{cases} 1, y_t \text{ received at time } k, \\ 0, \text{otherwise.} \end{cases}$$

Further define $\gamma_{k-i} \triangleq \sum_{j=0}^{i} \gamma_{k-i}^{k-j}$, i.e., γ_{k-i} indicates whether y_{k-i} is received by the estimator at or before k.

Assume $\hat{x}_{k-1}$ is optimal. Depending on whether y_k is received or not, i.e., $\gamma_k^k = 1$ or 0, $(\hat{x}_k, P_k)$ is known to be computed by a modified Kalman filter (**MKF**) [15]. We write $(\hat{x}_k, P_k)$ in compact form as follows.

$$(\hat{x}_k, P_k) = \mathbf{MKF}(\hat{x}_{k-1}, P_{k-1}, \gamma_k^k y_k),$$

which represents the following set of equations:

$$\begin{cases} \hat{x}_k^- &= A\hat{x}_{k-1}, \\ P_k^- &= AP_{k-1}A' + Q, \\ K_k &= P_k^- C'[CP_k^- C' + R]^{-1}, \\ \hat{x}_k &= A\hat{x}_{k-1} + \gamma_k^k K_k(y_k - CA\hat{x}_{k-1}), \\ P_k &= (I - \gamma_k^k K_k C)P_k^-. \end{cases}$$

Assume $\gamma_k^k = 1$ for all k, then **MKF** reduces to the standard Kalman filter. In this case, P_k^- and P_k can be shown to satisfy

$$P_k^- = g(P_{k-1}^-), \quad P_k = \tilde{g} \circ h(P_{k-1}).$$

Let P^* be the unique positive semidefinite solution[1] to $g(X) = X$, i.e., $P^* = g(P^*)$. Define $\overline{P}$ as $\overline{P} \triangleq \tilde{g}(P^*)$. Then we have

$$\tilde{g} \circ h(\overline{P}) = \tilde{g} \circ h \circ \tilde{g}(P^*) = \tilde{g} \circ g(P^*) = \tilde{g}(P^*) = \overline{P},$$

where we use the fact that $h \circ \tilde{g} = g$. In other words,

$$P^* = \lim_{k\to\infty} P_k^-, \quad \overline{P} = \lim_{k\to\infty} P_k.$$

[1] Since (A, C) is assumed to be observable and $(A, \sqrt{Q})$ controllable, from standard Kalman filtering analysis, P^* exists.

2.3.2 Optimal Estimation with Delayed Measurements

As y_{k-i} may arrive at time k due to the delays introduced by the network, we can improve the estimation quality by recalculating $\hat{x}_{k-i}$ utilizing the new available measurement y_{k-i}. Once $\hat{x}_{k-i}$ is updated, we can update $\hat{x}_{k-i+1}$ in a similar fashion. This is the basic idea contained in the flow diagram in Fig. 2.2. Proposition 2.3.1 summarizes the main estimation result.

Proposition 2.3.1. *Let $y_{k-i}, i \in [0, D-1]$ be the oldest measurement received by the estimator at time k. Then $\hat{x}_k$ is computed by $i+1$ **MKF**s as*

$$\begin{aligned}
(\hat{x}_{k-i}, P_{k-i}) &= \boldsymbol{MKF}(\hat{x}_{k-i-1}, P_{k-i-1}, y_{k-i}) \\
(\hat{x}_{k-i+1}, P_{k-i+1}) &= \boldsymbol{MKF}(\hat{x}_{k-i}, P_{k-i}, \gamma_{k-i+1} y_{k-i+1}) \\
&\vdots \\
(\hat{x}_{k-1}, P_{k-1}) &= \boldsymbol{MKF}(\hat{x}_{k-2}, P_{k-2}, \gamma_{k-1} y_{k-1}) \\
(\hat{x}_k, P_k) &= \boldsymbol{MKF}(\hat{x}_{k-1}, P_{k-1}, \gamma_k y_k).
\end{aligned}$$

*Furthermore, $\hat{D}$, the average number of **MKF** used at each time k is given by*

$$\hat{D} = \prod_{i=1}^{D-1} \big(1 - f(i)\big) + \sum_{j=2}^{D} \prod_{i=j}^{D-1} \big(1 - f(i)\big) f(j-1)\, j, \tag{2.3.1}$$

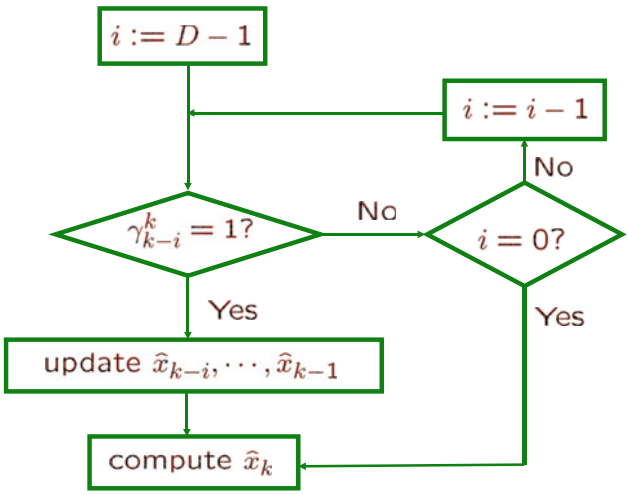


Fig. 2.2 Optimal estimation: sensor without computation capability

where

$$\prod_{i=D}^{D-1} \big(1 - f(i)\big) \triangleq 1.$$

Proof. We know that the estimate $\hat{x}_k$ is generated from the estimate of $\hat{x}_{k-1}$ together with $\gamma_k y_k$ at time k through an **MKF**. Similarly, the estimate $\hat{x}_{k-1}$ is generated from the estimate of $\hat{x}_{k-2}$ together with $\gamma_{k-1} y_{k-1}$ at time k through an **MKF**, etc. This recursion for $i+1$ steps corresponds to the $i+1$ **MKF**s stated in the theorem. Let $\hat{D}_k$ be the number of **MKF** used at each time k. Notice that $1 \le \hat{D}_k \le D$. Thus,

$$\hat{D} = \sum_{j=1}^{D} j \mathbf{Pr}[\hat{D}_k = j].$$

Consider $\mathbf{Pr}[\hat{D}_k = 1]$. Since $\hat{D}_k = 1$ iff $\gamma_{k-i}^k = 0$ for all $1 \le i \le D-1$, we have

$$\mathbf{Pr}[\hat{D}_k = 1] = \mathbf{Pr}[\gamma_{k-i}^k = 0, 1 \le i \le D-1].$$

As $\mathbf{Pr}[\gamma_{k-i}^k = 0] = 1 - f(i)$, we obtain

$$\mathbf{Pr}[\hat{D}_k = 1] = \prod_{i=1}^{D-1} \big(1 - f(i)\big).$$

Similarly, for $2 \le j \le D-1$, we have

$$\mathbf{Pr}[\hat{D}_k = j] = \prod_{i=j}^{D-1} \big(1 - f(i)\big) f(j-1)$$

and when $j = D$,

$$\mathbf{Pr}[\hat{D}_k = j] = f(D-1).$$

Therefore, we obtain $\hat{D}$ in (2.3.1). □

Remark 2.3.2. Similar results on estimation with delayed packets can be derived in other ways (e.g. [6]) but the result in Proposition 2.3.1 fits our intended uses later in the paper.

Remark 2.3.3. Although this section focuses on when C is invertible, proposition 2.3.1 does not need to assume C to be invertible. Depending on whether (A, C) is detectable or not, the error covariance matrix may diverge. However, the algorithm always produces the optimal estimate at each time.

Notice that in the first **MKF**, $\gamma_{k-i}^k = 1$ and hence $\gamma_{k-i} = 1$. As a result, we simply write $\gamma_{k-i} y_{k-i} = y_{k-i}$.

2.3.3 Lower and Upper Bounds of $\mathbf{Pr}[P_k \leq M]$

Since d_k is random and described by the probability mass function f, γ_{k-i}^k $(i = 0, \ldots, D-1)$ is also random. As a consequence, P_k computed as in Proposition 2.3.1 is a random variable. Define $\hat{\gamma}_i(D)$ as

$$\hat{\gamma}_i(D) \triangleq \begin{cases} \sum_{j=0}^{i} f(j), \text{if } 0 \leq i < D, \\ \sum_{j=0}^{D-1} f(j), \text{if } i \geq D. \end{cases}$$

Recall that γ_{k-i} indicates whether y_{k-i} is received by the estimator at or before k, so it is easy to verify that

$$\mathbf{Pr}[\gamma_{k-i} = 1] = \hat{\gamma}_i(D). \tag{2.3.2}$$

Define $\overline{M} \triangleq C^{-1}RC^{-1'}$. Then we have the following result that shows the relationship between P_k and $\overline{M}$.

Lemma 2.3.4. *For any $k \geq 1$, if $\gamma_k = 1$, then $P_k \leq \overline{M}$.*

Proof. As $\gamma_k = 1$, we have $P_k = \tilde{g} \circ h(P_{k-1}) \leq \overline{M}$, where the inequality is from Lemma 2.A.2 in Appendix 2.A. □

Remark 2.3.5. We can also interpret Lemma 2.3.4 as follows. One way to obtain an estimate $\tilde{x}_k$ when $\gamma_k = 1$ is simply by inverting the measurement, i.e., $\tilde{x}_k = C^{-1}y_k$. Therefore,

$$\tilde{e}_k = C^{-1}v_k \text{ and } \tilde{P}_k = \mathbb{E}[\tilde{e}_k\tilde{e}_k'] = C^{-1}RC^{-1'} = \overline{M}.$$

Since Kalman filter is optimal among the set of all linear filters, we must have $P_k \leq \tilde{P}_k = \overline{M}$.

Recall that $\overline{P}$ is defined in Sect. 2.3.1 as the steady-state error covariance for the Kalman filter. For $M \geq \overline{M}$, let us define $k_1(M)$ and $k_2(M)$ as follows:

$$k_1(M) \triangleq \min\{t \geq 1 : h^t(\overline{M}) \not\leq M\}, \tag{2.3.3}$$

$$k_2(M) \triangleq \min\{t \geq 1 : h^t(\overline{P}) \not\leq M\}. \tag{2.3.4}$$

We sometimes write $k_i(M)$ as $k_i, i = 1, 2$ for simplicity for the rest of the chapter. The following lemma shows the relationship between $\overline{P}$ and $\overline{M}$ as well as k_1 and k_2.

Lemma 2.3.6. *(1) $\overline{P} \leq \overline{M}$; (2) $k_1 \leq k_2$ whenever either k_i is finite, $i = 1, 2$.*

Proof. 1. $\overline{P} = \tilde{g}(P^*) \leq \overline{M}$ where the inequality is from Lemma 2.A.2 in Appendix 2.A.

2. Without loss of generality, we assume k_2 is finite. If k_1 is finite, and $k_1 > k_2$, then according to their definitions, we must have

$$M \geq h^{k_1-1}(\overline{M}) \geq h^{k_1-1}(\bar{P}) \geq h^{k_2}(\bar{P}),$$

which violates the definition of k_2. Notice that we use the property that h is nondecreasing as well as $h(\bar{P}) \geq \bar{P}$ from Lemma 2.A.1 and 2.A.3 in Sect. 2.A in the Appendix. Similarly, we can show that k_1 cannot be infinite. Therefore, we must have $k_1 \leq k_2$. □

Lemma 2.3.7. *Assume $P_0 \geq \overline{P}$. Then $P_k \geq \overline{P}$ for all $k \geq 0$.*

Proof. Since **MKF** is used at each time k,

$$P_k = \hat{f}_k^k \circ \hat{f}_{k-1}^k \cdots \hat{f}_1^k(P_0) \geq \overline{P},$$

where $\hat{f}_{k-i}^k = h$ or $\hat{f}_{k-i}^k = \tilde{g} \circ h$ depending on the packet arrival sequence[2]. The inequality is from Lemma 2.A.1 in Appendix 2.A. □

Define N_k as the number of consecutive packets not received by k, i.e.,

$$N_k \triangleq \min\{t \geq 0 : \gamma_{k-t} = 1\}. \tag{2.3.5}$$

Define

$$\theta(k_i, D) \triangleq \prod_{j=0}^{k_i-1} \big(1 - \hat{\gamma}_j(D)\big). \tag{2.3.6}$$

It is easy to see that

$$\theta(k_1, D) \geq \theta(k_2, D).$$

Lemma 2.3.8. *Let k_1, k_2, and N_k be defined according to (2.3.3)–(2.3.5). Then*

$$\mathbf{Pr}[N_k \geq k_i] = \theta(k_i, D), i = 1, 2. \tag{2.3.7}$$

Proof.

$$\mathbf{Pr}[N_k \geq k_i] = \mathbf{Pr}[\gamma_{k-i} = 0, 0 \leq i \leq k_i - 1] = \theta(k_i, D).$$

□

Theorem 2.3.9. *Assume $\overline{P} \leq P_0 \leq \overline{M}$. For any $M \geq \overline{M}$, we have*

$$1 - \theta(k_1, D) \leq \mathbf{Pr}[P_k \leq M] \leq 1 - \theta(k_2, D). \tag{2.3.8}$$

Proof. We divide the proof into two parts. For the rest of the proof, all probabilities are conditioned on the given D.

[2] Note that we use the superscript k in $\hat{f}_{k-i}^k$ to emphasize that it depends on the current time, k. For example, if $d_{k-i} = i + 1$, i.e., $\gamma_{k-i} = 0$ and $\gamma_{k-i}^{k+1} = 1$, then $\hat{f}_{k-i}^k = h$ and $\hat{f}_{k-i}^{k+1} = \tilde{g} \circ h$.

1. Let us first prove $1-\theta(k_1, D) \leq \mathbf{Pr}[P_k \leq M]$, or in other words,

$$1-\mathbf{Pr}[N_k \geq k_1] \leq \mathbf{Pr}[P_k \leq M].$$

As $\gamma_k = 1$ or 0, there are in total 2^k possible realizations of γ_1 to γ_k as seen from Fig. 2.3. Let Σ_1 denote those packet arrival sequences of γ_1 to γ_k such that $N_k \geq k_1$. Similarly, let Σ_2 denote those packet arrival sequences such that $N_k < k_1$. Let $P_k(\sigma_i)$ be the error covariance at time k when the underlying packet arrival sequence is σ_i, where $\sigma_i \in \Sigma_i, i = 1, 2$. Consider a particular $\sigma_2 \in \Sigma_2$. As $\gamma_{k-k_1+1} = 1$, from Lemma 2.3.4, $P_{k-k_1+1} \leq \overline{M}$. Therefore, we have

$$P_k(\sigma_2) \leq h^{k_1-1}(P_{k-k_1+1}) \leq h^{k_1-1}(\overline{M}) \leq M,$$

where the first and second inequalities are from Lemma 2.A.1 in Appendix 2.A and the last inequality is from the definition of k_1. In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma_2] = 1.$$

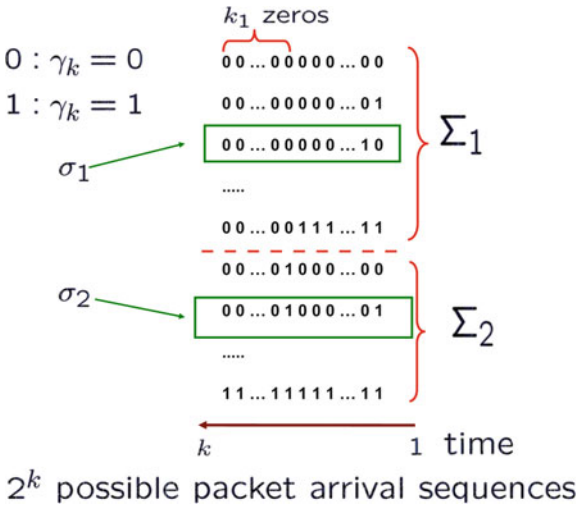


Fig. 2.3 $N_k \geq k_1$

Therefore,

$$\begin{aligned}\mathbf{Pr}[P_k \le M] &= \sum_{\sigma \in \Sigma_1 \cup \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma]\mathbf{Pr}(\sigma) \\ &= \sum_{\sigma_1 \in \Sigma_1} \mathbf{Pr}[P_k \le M|\sigma_1]\mathbf{Pr}(\sigma_1) + \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma_2]\mathbf{Pr}(\sigma_2) \\ &\ge \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma_2]\mathbf{Pr}(\sigma_2) \\ &= \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}(\sigma_2) = \mathbf{Pr}(\Sigma_2) = 1 - \mathbf{Pr}(\Sigma_1) = 1 - \mathbf{Pr}[N_k \ge k_1],\end{aligned}$$

where the first equality is from the Total Probability Theorem, the second equality holds as Σ_1 and Σ_2 are disjoint, and the third inequality holds as the first sum is nonnegative. The rest equalities are easy to see.

2. We now prove $\mathbf{Pr}[P_k \le M] \le 1 - \theta(k_2, D)$, or in other words

$$\mathbf{Pr}[P_k \le M] \le 1 - \mathbf{Pr}[N_k \ge k_2].$$

Let Σ_1' denote those packet arrival sequences of γ_1 to γ_k such that $N_k \ge k_2$ and Σ_2' denote those packet arrival sequences such that $N_k < k_2$ (Fig. 2.4). Consider $\sigma_1' \in \Sigma_1'$. Let

$$s(\sigma_1') = \min\{t \ge 1 : \gamma_{k-t} = 1|\sigma_1'\}.$$

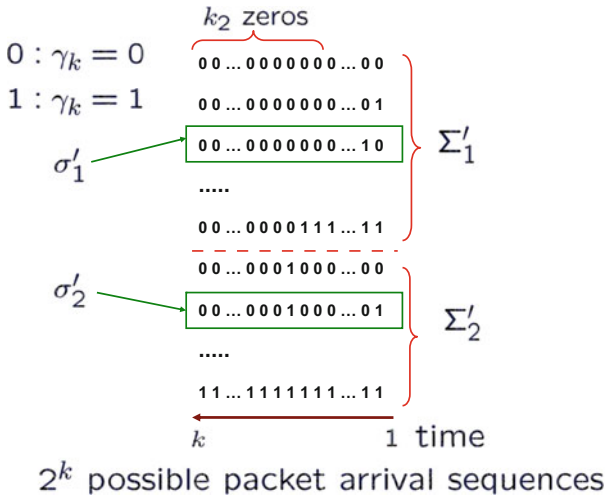


Fig. 2.4 $N_k \ge k_2$

As $\sigma_1' \in \Sigma_1'$, we must have $s \geq k_2$. Consequently,

$$P_k(\sigma_1') = h^{s(\sigma_1')}\left(P_{k-s(\sigma_1')}\right) \geq h^{s(\sigma_1')}(\overline{P}),$$

where the inequality is from Lemma 2.3.7. Therefore we conclude that

$$P_k(\sigma_1') \not\leq M.$$

Otherwise,

$$h^{s(\sigma_1')}(\overline{P}) \leq P_k(\sigma_1') \leq M,$$

which violates the definition of k_2. In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma_1'] = 0.$$

Therefore, we have

$$\begin{aligned}
\mathbf{Pr}[P_k \leq M] &= \sum_{\sigma \in \Sigma_1 \cup \Sigma_2} \mathbf{Pr}[P_k \leq M | \sigma] \mathbf{Pr}(\sigma) \\
&= \sum_{\sigma_1' \in \Sigma_1'} \mathbf{Pr}[P_k \leq M | \sigma_1'] \mathbf{Pr}(\sigma_1') + \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M | \sigma_2'] \mathbf{Pr}(\sigma_2') \\
&= \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M | \sigma_2'] \mathbf{Pr}(\sigma_2') \\
&\leq \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}(\sigma_2') = \mathbf{Pr}(\Sigma_2') = 1 - \mathbf{Pr}(\Sigma_1') = 1 - \mathbf{Pr}[N_k \geq k_2],
\end{aligned}$$

where the inequality is from the fact that

$$\mathbf{Pr}[P_k \leq M | \sigma_2'] \leq 1 \text{ for any } \sigma_2' \in \Sigma_2'.$$

□

2.3.4 Computing the Minimum D

Assume we require that

$$\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon, \tag{2.3.9}$$

then according to (2.3.8), a sufficient condition is that

$$\theta(k_1, D) \leq \epsilon. \tag{2.3.10}$$

And a necessary condition is that

$$\theta(k_2, D) \leq \epsilon. \tag{2.3.11}$$

For a given M, define

$$\epsilon_1(M) \triangleq \theta(k_1, k_1 - 1), \tag{2.3.12}$$

$$\epsilon_2(M) \triangleq \theta(k_2, k_2 - 1). \tag{2.3.13}$$

2.3.4.1 Sufficient Minimum D

Notice that $\theta(k_1, D)$ is decreasing when $1 \leq D \leq k_1 - 1$ and remains constant when $D \geq k_1$. Hence if $\epsilon < \epsilon_1(M)$, no matter how large D is, there is *no guarantee* that $\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon$. If $\epsilon \geq \epsilon_1(M)$, then the minimum D_s such that *guarantees* $\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon$ is given by

$$D_s = \min\{D : \theta(k_1, D) \leq \epsilon, 1 \leq D \leq k_1 - 1\}. \tag{2.3.14}$$

2.3.4.2 Necessary Minimum D

Similarly, $\theta(k_2, D)$ is decreasing when $1 \leq D \leq k_2 - 1$ and remains constant when $D \geq k_2$. Hence if $\epsilon < \epsilon_2(M)$, no matter how large D is, it is guaranteed that $\mathbf{Pr}[P_k \leq M] > 1 - \epsilon$. If $\epsilon \geq \epsilon_1(M)$, then the minimum D_s such that it is *possible* that $\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon$ is given by

$$D_s = \min\{D : \theta(k_2, D) \leq \epsilon, 1 \leq D \leq k_2 - 1\}. \tag{2.3.15}$$

Example 2.3.10. Consider (2.2.1) and (2.2.2) with

$$A = 1.4, C = 1, Q = 0.2, R = 0.5.$$

We model the packet delay as a Poisson distribution with mean d, i.e., the probability density function $f(i)$ satisfies

$$f(i) = \frac{d^i \mathrm{e}^{-d}}{i!}, i = 0, 1, \ldots,$$

where $d = \mathbb{E}[d_k]$ denotes the mean value of the packet delay.

When $M = 50$ (Fig. 2.5), it is calculated that $k_1(M) = k_2(M) = 7$, hence $\theta(k_1, D) = \theta(k_2, D)$ and $\theta(7, 6) = 0.0313$. Thus, we can find the minimum D that guarantees $\mathbf{Pr}[P_k \leq 50] \geq 1 - \epsilon$ for any $\epsilon \geq 0.0313$. For any $\epsilon < 0.0313$, no matter how large D is, $\mathbf{Pr}[P_k \leq 50] < 1 - \epsilon$.

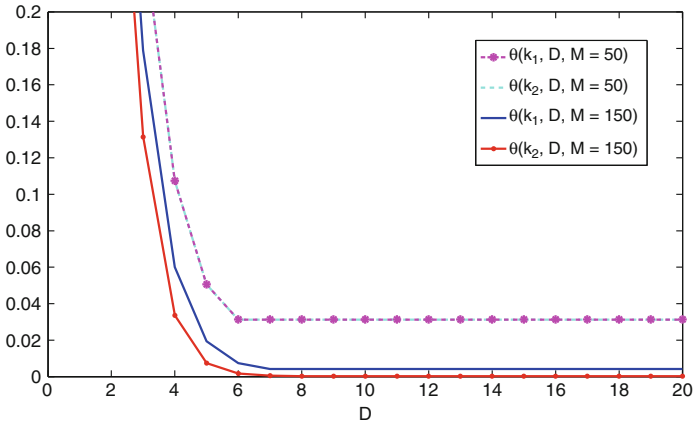


Fig. 2.5 $\theta(k_i, D)$ for different M

When $M = 150$, it is calculated that $k_1(M) = 8$ and $k_2(M) = 9$, hence $\theta(k_1, D) > \theta(k_2, D)$. We also find that $\theta(8, 7) = 0.0042$ and $\theta(9, 8) = 0.0003$. Therefore if $\epsilon > 0.0042$, we can find minimum D that guarantees $\mathbf{Pr}[P_k \leq 150] \geq 1 - \epsilon$; if $\epsilon < 0.0003$, no matter how large D is, $\mathbf{Pr}[P_k \leq 150] > 1 - \epsilon$.

Remark 2.3.11. We find the minimum D that gives the desired filter performance, i.e., $\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon$ for a given M and ϵ. We can also find the minimum D when requiring $\mathbb{E}[P_k]$ to be stable, i.e., $\lim_{k\to\infty} \mathbb{E}[P_k] < \infty$, and we provide the detailed analysis in Appendix 2.B using the theory developed in this section.

2.4 Sensor with Computation Capability

In this section, we consider the second scenario in Fig. 2.1, i.e., the sensor has necessary computation capability and sends $\hat{x}_k^s$ to the remote estimator. We assume all the variables in this section, e.g., γ_t^k, γ_k, etc. are the same as they are defined in Sect. 2.3 unless they are explicitly defined.

Consider the case when k is sufficiently large so that the Kalman filter enters steady state at the sensor side, i.e., $P_k^s = \overline{P}$. It is clear that the optimal estimation at the remote estimator is as follows. If $\gamma_k = 1$, then $\hat{x}_k = \hat{x}_k^s$ and $P_k = P_k^s = \overline{P}$. If $\gamma_k = 0$ and $\gamma_{k-1} = 1$, then $\hat{x}_k = A\hat{x}_{k-1}^s$ and $P_k = h(\overline{P})$. This is repeated until we examine γ_{k-D+1}. The full optimal estimation algorithm is presented in Fig. 2.6.

Note that in the first scenario (Fig. 2.2), i.e., sensor has no computation capability, we examine the sequence from γ_{k-D+1}^k to γ_k^k, while in the second scenario, (Fig. 2.6), we examine the sequence from γ_k to γ_{k-D+1}.

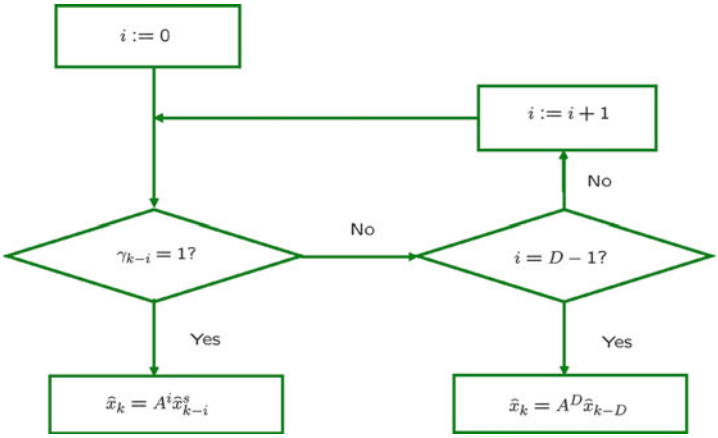


Fig. 2.6 Optimal estimation: sensor with computation capability

.

Theorem 2.4.1. *Assume k is sufficiently large such that $P^s_k = \overline{P}$. For any $M \geq \overline{P}$, we have*

$$\mathbf{Pr}[P_k \leq M] = 1 - \theta(k_2, D). \tag{2.4.1}$$

Proof. For the rest of the proof, all probabilities are conditioned on the given D. Let σ'_i and $\Sigma'_i, i = 1, 2$ be defined in the same way as in the proof of Theorem 2.3.9 (see Fig. 2.4). Clearly for any $\sigma'_2 \in \Sigma'_2$,

$$P_k(\sigma'_2) \leq h^{k_2-1}(\overline{P}) \leq M.$$

The first inequality is from the fact that $\gamma_{k-k_2+1} = 1$ and hence $P_{k-k_2+1} = \overline{P}$. The second inequality is from the definition of k_2. In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma'_2] = 1.$$

Similar to the proof of Theorem 2.3.9, for $\sigma'_1 \in \Sigma'_1$, let us define

$$s = s(\sigma'_1) \triangleq \min\{t \geq 1 : \gamma_{k-t} = 1 | \sigma'_1\}.$$

As $\sigma'_1 \in \Sigma'_1$, $s \geq k_2$. Therefore

$$P_k(\sigma'_1) = h^s(\overline{P}) \not\leq M.$$

In other words, $\mathbf{Pr}[P_k \leq M|\sigma_1'] = 0$. Therefore

$$\begin{aligned}\mathbf{Pr}[P_k \leq M] &= \sum_{\sigma' \in \Sigma_1' \cup \Sigma_2'} \mathbf{Pr}[P_k \leq M|\sigma']\mathbf{Pr}(\sigma') \\ &= \sum_{\sigma_1' \in \Sigma_1'} \mathbf{Pr}[P_k \leq M|\sigma_1']\mathbf{Pr}(\sigma_1') + \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M|\sigma_2']\mathbf{Pr}(\sigma_2') \\ &= \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M|\sigma_2']\mathbf{Pr}(\sigma_2') \\ &= \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}(\sigma_2') = \mathbf{Pr}(\Sigma_2') = 1 - \mathbf{Pr}(\Sigma_1') = 1 - \mathbf{Pr}[N_k \geq k_2].\end{aligned}$$

□

Computing $\mathbf{Pr}[N_k \geq k_2]$ follows exactly the same way as in Sect. 2.3.4. Since we have a strict equality in (2.4.1), in order that

$$\mathbf{Pr}[P_k \leq M] \geq 1 - \epsilon$$

a necessary and sufficient condition is that

$$\mathbf{Pr}[N_k \geq k_2] \leq \epsilon. \tag{2.4.2}$$

Therefore, the minimum D^* that guarantees (2.4.2) to hold is given by

$$D^* = \min\{D : \theta(k_2, D) \leq \epsilon, 1 \leq D \leq k_2 - 1\}. \tag{2.4.3}$$

Note that since $\theta(k_2, D) \geq \theta(k_2, k_2 - 1) = \epsilon_2(M)$, D^* from the above equation exists if and only if $\epsilon \geq \epsilon_2(M)$.

2.4.1 When C Is Not Full Rank

We use Theorem 2.4.1 to tackle the case when C is not full rank for the first scenario, i.e., sensor without computation capability. Since (A, C) is observable, there exists r $(2 \leq r \leq n)$ such that

$$\begin{bmatrix} C \\ CA \\ \cdots \\ CA^{r-1} \end{bmatrix}$$

is full rank. In this section, we consider the special case when $r = 2$, and in particular, we assume $\begin{bmatrix} C \\ CA \end{bmatrix}^{-1}$ exists. The idea readily extends to the general case.

Unlike the case when C^{-1} exists, and y_k is sent across the network, here we assume that the previous measurement y_{k-1} is sent along with y_k. This only requires that the sensor has a buffer that stores y_{k-1}. Then if $\gamma_k = 1$, both y_k and y_{k-1} are received. Thus, we can use the following linear estimator to generate $\hat{x}_k$

$$\hat{x}_k = A \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} y_k \\ y_{k-1} \end{bmatrix}.$$

The corresponding error covariance can be calculated as

$$P_k = AM_1A' + Q - A \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} CQ \\ 0 \end{bmatrix} - \begin{bmatrix} CQ \\ 0 \end{bmatrix}' \begin{bmatrix} CA \\ C \end{bmatrix}^{-1'} A',$$

where

$$M_1 = \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} CQC' + R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} CA \\ C \end{bmatrix}^{-1'}.$$

Therefore once the packet for time k is received, i.e., $\gamma_k = 1$, we have

$$P_k = AM_1A' + Q - A \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} CQ \\ 0 \end{bmatrix} - \begin{bmatrix} CQ \\ 0 \end{bmatrix}' \begin{bmatrix} CA \\ C \end{bmatrix}^{-1'} A' \triangleq \tilde{P}.$$

Now, if we treat $\tilde{P}$ as the steady-state error covariance at the sensor side, i.e., by letting $P_k^s = \tilde{P}$, and define

$$k_v \triangleq \min\{t \geq 1 : h^t(\tilde{P}) \not\leq M\},$$

we immediately obtain

$$\mathbf{Pr}[P_k \leq M] = 1 - \theta(k_v, D). \tag{2.4.4}$$

Remark 2.4.2. Although we give the exact expression of $\mathbf{Pr}[P_k \leq M]$ in (2.4.4), we have to point out that $\theta(k_v, D) \geq \theta(k_2, D)$, as $\tilde{P} \geq \overline{P}$ due to the optimality of Kalman filter. Thus, the case that sensor has computation capability leads to better filter performance, which is illustrated from the vector system example in the next section.

2.5 Examples

2.5.1 Scalar System

Consider the same parameters as in Example 2.3.10, i.e.,

$$A = 1.4, C = 1, Q = 0.2, R = 0.5$$

and

$$f(i) = \frac{d^i \mathrm{e}^{-d}}{i!}, i = 0, 1, \ldots$$

Figure 2.7 shows the values of $f(i)$ for $0 \leq i \leq 20$ for $d = 3$ and 5, respectively.

2.5.1.1 Sensor Without Computation Capability

We run a Monte Carlo simulation for different parameters. Figures 2.8–2.10 show the results when D and d take different values. From Fig. 2.11, we can see that both smaller d and larger D lead to larger $\mathbf{Pr}[P_k \leq M]$, which confirms the theory developed in this chapter. We also notice that when $d = 3$, the filter's performances using $D = 10$ and $D = 5$ only differ slightly (though the former one is better than the latter one), which confirms that using a large buffer may not improve the filter performance drastically.

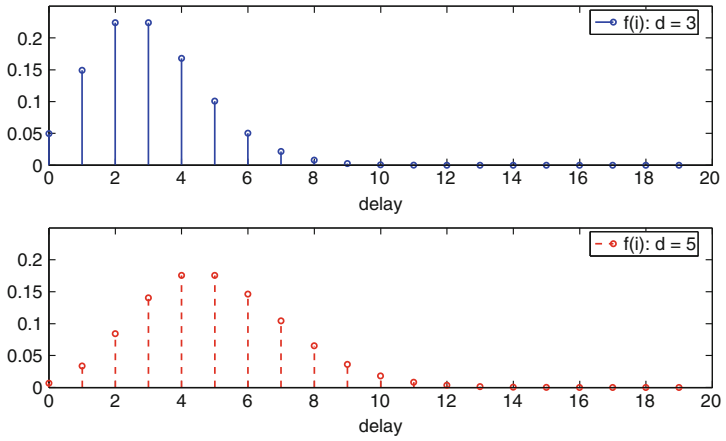


Fig. 2.7 Poisson distribution with $d = 3$ and $d = 5$

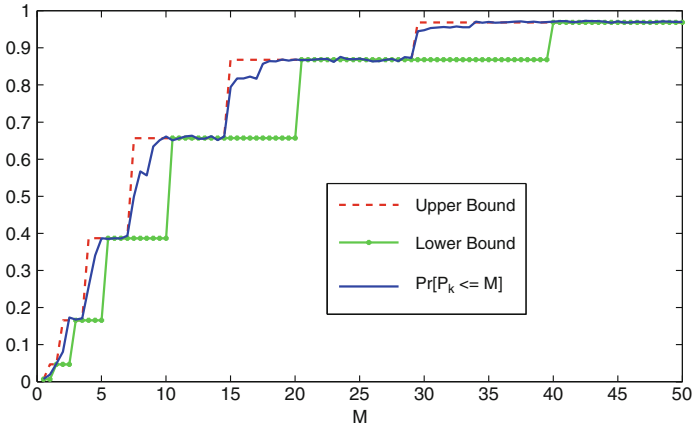


Fig. 2.8 $\mathbf{Pr}[P_k \leq M = 10], d = 5$

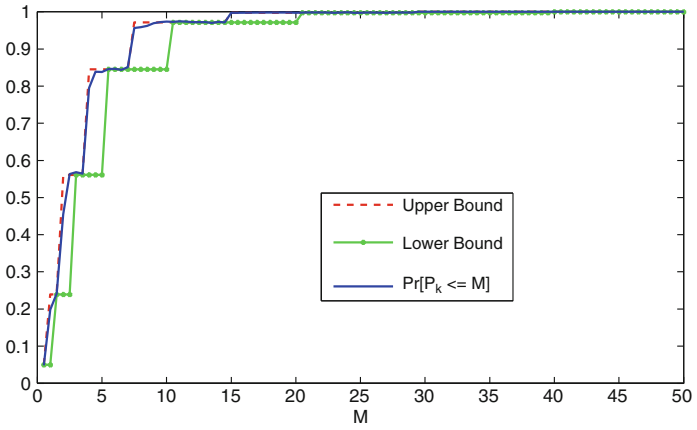


Fig. 2.9 $\mathbf{Pr}[P_k \leq M = 10], d = 3$

2.5.1.2 Sensor with Computation Capability

We run a Monte Carlo simulation for the case when the sensor has computation capability. Figure 2.12 shows the result when $D = 10$ and $d = 5$. As we can see, the predicted value of $\mathbf{Pr}[P_k \leq M]$ from (2.4.1) matches well with the actual value.

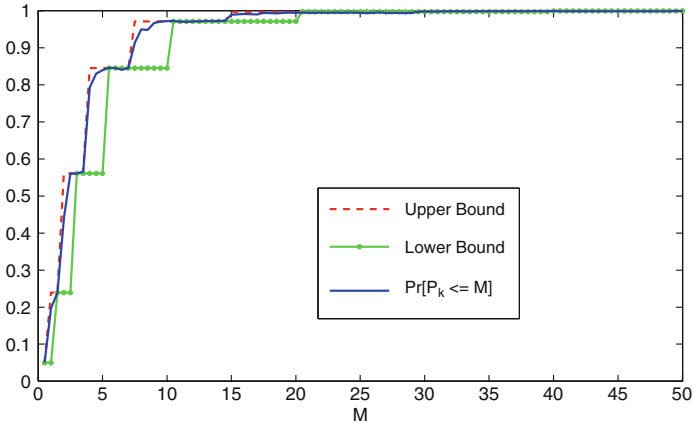


Fig. 2.10 $\mathbf{Pr}[P_k \leq M = 5], d = 3$

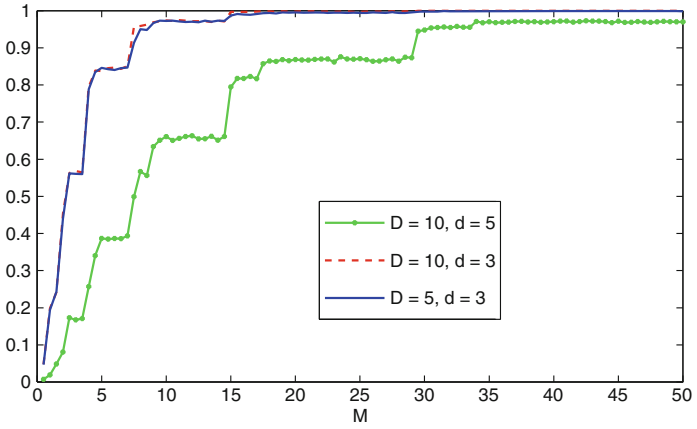


Fig. 2.11 Comparison of the three simulations

2.5.2 Vector System

Consider a vehicle moving in a two-dimensional space according to the standard constant acceleration model, which assumes that the vehicle has zero acceleration except for a small perturbation. The state of the vehicle consists of its x and y positions as well as velocities. Assume a sensor measures the positions of the vehicle

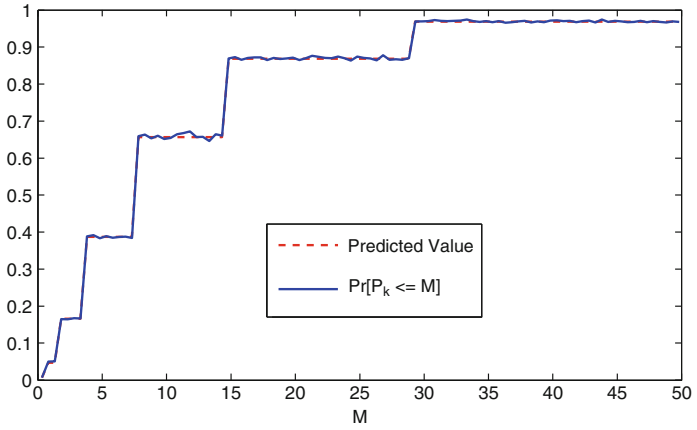


Fig. 2.12 $\mathbf{Pr}[P_k \leq M = 10], d = 5$

and sends the measurements to a remote estimator over a packet-delaying network. The system parameters are given according to (2.2.1)–(2.2.2) as follows:

$$A = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The process and measurement noise covariances are $Q = \mathbf{diag}(0.01, 0.01, 0.01, 0.01)$ and $R = \mathbf{diag}(0.001, 0.001)$. We assume the same delay profile as in the scalar system example with $D = 5$ and $d = 3$.

We run a Monte Carlo simulation for both cases when the sensor has or has not computation capability. As we can see from Fig. 2.13, the predicted values of $\mathbf{Pr}[P_k \leq M]$ from (2.4.1) and (2.4.4) match well with the actual values. We also notice that when sensor has computation capability, the actual filter performance is better than when sensor has no computation capability, as stated in Remark 2.4.2. In Fig. 2.13, the M in the x-axis means $M \times I_4$, where I_4 is the identity matrix of dimension 4.

2.A Supporting Lemmas

Lemma 2.A.1. *For any* $0 \leq X \leq Y$,

$$\begin{aligned} &h(X) \leq h(Y), \quad g(X) \leq g(Y), \\ &\tilde{g}(X) \leq \tilde{g}(Y), \quad \tilde{g}(X) \leq X, \\ &h \circ \tilde{g}(X) = g(X), \quad g(X) \leq h(X). \end{aligned}$$

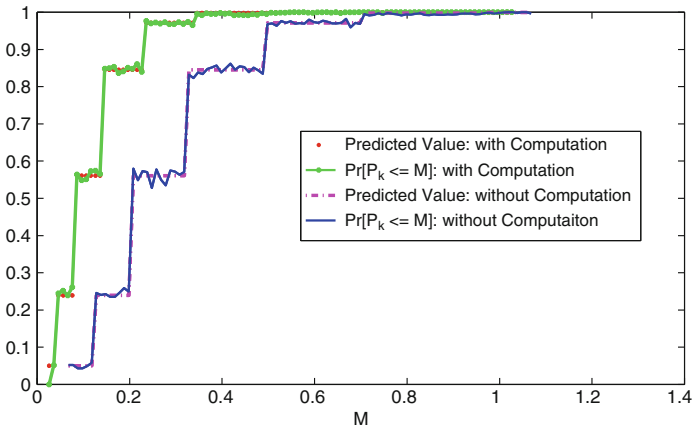


Fig. 2.13 $\mathbf{Pr}[P_k \leq M = 5], d = 3$

Proof. $h(X) \leq h(Y)$ holds as $h(X)$ is affine in X. Proof for $g(X) \leq g(Y)$ can be found in Lemma 1-c in [15]. As $\tilde{g}$ is a special form of g by setting $A = I$ and $Q = 0$, we immediately obtain $\tilde{g}(X) \leq \tilde{g}(Y)$. Next, we have

$$\tilde{g}(X) = X - XC'[CXC' + R]^{-1}CX \leq X$$

and

$$\begin{aligned} h \circ \tilde{g}(X) &= h(X - XC'[CXC' + R]^{-1}CX) \\ &= A(X - XC'[CXC' + R]^{-1}CX)A' + Q \\ &= g(X). \end{aligned}$$

Finally, we have

$$g(X) = h(X) - AXC'[CXC' + R]^{-1}CXA' \leq h(X).$$

□

Lemma 2.A.2. *For any* $X \geq 0, \tilde{g}(X) \leq \overline{M}$.

Proof. For any $t > 0$, we have

$$\tilde{g}(t\overline{M}) = \frac{t}{t+1}\overline{M} \leq \overline{M}.$$

For all $X \geq 0$, since $\overline{M} > 0$, it is clear that there exists $t_1 > 0$ such that $t_1\overline{M} > X$. Therefore,

$$\tilde{g}(X) \leq \tilde{g}(t_1\overline{M}) \leq \overline{M}.$$

□

Lemma 2.A.3. $\overline{P} \leq h(\overline{P})$.

Proof.

$$h(\overline{P}) = h \circ \tilde{g}(P^*) = g(P^*) = P^* \geq \tilde{g}(P^*) = \overline{P},$$

where the first and the last equality are from the definition of $\overline{P}$, the third equality is from the definition of P^*. The rest equality and inequality are from Lemma 2.A.1. □

Lemma 2.A.4. *Let X be a continuous random variable defined on $[0, \infty)$ and let $F(x) = \mathbf{Pr}[X \leq x]$. Then*

$$\mathbb{E}[X] = \int_0^\infty [1 - F(x)]\mathrm{d}x.$$

Proof. See Lemma (4) in [2], page 93. □

2.B Evaluate $\mathbb{E}[P_k]$ and Its Stability

Consider (2.2.1) and (2.2.2) with

$$A = a > 1, Q = q > 0, C = c > 0, R = r > 0.$$

For scalar systems, $\mathbf{Pr}[P_k \leq M]$ is the cumulative distribution function of the random variable P_k (for a given D). Therefore, we can find $\mathbb{E}[P_k]$ from Theorem 2.3.9 or Theorem 2.4.1 by using Lemma 2.A.4 in Appendix 2.A, i.e., we write $\mathbb{E}[P_k]$ as

$$\begin{aligned}\mathbb{E}[P_k] &= \int_0^\infty (1 - \mathbf{Pr}[P_k \leq M])\mathrm{d}M \\ &= \int_0^{\overline{M}} (1 - \mathbf{Pr}[P_k \leq M])\mathrm{d}M + \int_{\overline{M}}^\infty (1 - \mathbf{Pr}[P_k \leq M])\mathrm{d}M.\end{aligned}$$

Let us consider the case when sensor has no computation capability. The following results extends trivially to the case when sensor has computation capability. Using the fact

$$0 \leq \mathbf{Pr}[P_k \leq M] \leq 1,$$

we have

$$\mathbb{E}[P_k] \leq \overline{M} + \int_{\overline{M}}^\infty (1 - \mathbf{Pr}[P_k \leq M])\mathrm{d}M \leq \overline{M} + \int_{\overline{M}}^\infty \theta(k_1, D)\mathrm{d}M,$$

and

$$\mathbb{E}[P_k] \geq \int_{\overline{M}}^{\infty} (1 - \mathbf{Pr}[P_k \leq M])\mathrm{d}M \geq \int_{\overline{M}}^{\infty} \theta(k_2, D)\mathrm{d}M.$$

Recall that $k_1(M) = \min\{t \geq 1 : h^t(\overline{M}) \not\leq M\}$ and

$$\begin{aligned} h^t(\overline{M}) &= a^{2t}\overline{M} + q(1 + a^2 + \cdots + a^{2t-2}) = \left(\overline{M} + \frac{q}{a^2 - 1}\right) a^{2t} - \frac{q}{a^2 - 1} \\ &= c_1 a^{2t} - c_2 \end{aligned}$$

where

$$c_1 = \overline{M} + \frac{q}{a^2 - 1}, c_2 = \frac{q}{a^2 - 1},$$

therefore for any $t \geq 1$,

$$k_1(M) = t, \text{ if } c_1 a^{2t-2} - c_2 \leq M < c_1 a^{2t} - c_2.$$

Therefore,

$$\mathbb{E}[P_k] \leq \overline{M} + \int_{\overline{M}}^{\infty} \theta(k_1, D)\mathrm{d}M = \overline{M} + \sum_{t=1}^{\infty} (c_1 a^{2t} - c_1 a^{2t-2})\theta(t, D), \quad (2.B.1)$$

and

$$\mathbb{E}[P_k] \geq \int_{\overline{M}}^{\infty} \theta(k_2, D)\mathrm{d}M = \sum_{t=1}^{\infty} (c_1' a^{2t} - c_1' a^{2t-2})\theta(t, D), \quad (2.B.2)$$

where $c_1' = \overline{P} + \frac{q}{a^2 - 1}$.

Lemma 2.B.1. *The minimum D that guarantees*

$$\lim_{k \to \infty} \mathbb{E}[P_k] < \infty$$

is given by

$$D_{\min} = \min \left\{ d : \sum_{i=0}^{d-1} f(i) > 1 - \frac{1}{a^2} \right\}. \quad (2.B.3)$$

Any other $D < D_{\min}$ leads to

$$\lim_{k \to \infty} \mathbb{E}[P_k] = \infty.$$

Proof. Recall that $\theta(k_i, D)$ is defined in (2.3.6) as

$$\theta(k_i, D) = \prod_{j=0}^{k_i-1} \big(1 - \hat{\gamma}_j(D)\big),$$

and $\hat{\gamma}_i(D) = \hat{\gamma}_D$ for any $i \geq D$. Thus from (2.B.1), in order that $\mathbb{E}[P_k] < \infty$, it is sufficient that

$$1 - \hat{\gamma}_D < \frac{1}{a^2},$$

or in other words,

$$\sum_{i=0}^{D-1} f(i) > 1 - \frac{1}{a^2}.$$

Similarly, if $D < D_{\min}$, i.e.,

$$1 - \hat{\gamma}_D \geq \frac{1}{a^2},$$

then from (2.B.2),

$$\lim_{k\to\infty} \mathbb{E}[P_k] = \infty.$$

□

References

1. R. W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45, July 2000.
2. Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 3 edition, 2001.
3. J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems, volume 95. Proceedings of the IEEE, January 2007.
4. M. Huang and S. Dey. Stability of kalman filtering with markovian packet losses. *Automatica*, 43:598–607, 2007.
5. Xiangheng Liu and Andrea Goldsmith. Kalman filtering with partial observation losses. pages 4180– 4186. In Proceedings of the IEEE Conference on Decision and Control, 2004.
6. A. S. Matveev and A. V. Savkin. The problem of state estimation via asynchronous communication channels with irregular transmission times. *IEEE Transactions on Automatic Control*, 48:670–676, 2003.
7. G. N. Nair and R. J. Evans. Communication-limited stabilization of linear systems. In *Proceedings of the 39th Conf. on Decision and Contr.*, volume 1, pages 1005–1010, Dec 2000.
8. S. Nakamori, R. Caballero-Aguila, A. Hermoso-Carazo, and J. Linares-Perez. Recursive estimators of signals from measurements with stochastic delays using covariance information. *Applied Mathematics and Computation*, 162:65–79, 2005.
9. J. Nilsson. *Real Time Control Systems with Delays*. PhD thesis, Lund Institute of Technology, Lund, Sweden, 1998.
10. I. R. Petersen and A. V. Savkin. Multi-rate stabilization of multivariable discrete-time linear systems via a limited capacity communication channel. In *Proceedings of the 40th Conf. on Decision and Contr.*, volume 1, pages 304–309, Dec 2001.

11. A. Ray, L. W. Liou, and J. H. Shen. State estimation using randomly delayed measurements. *J. Dyn. Syst., Measurement Contr.*, 115:19–26, 1993.
12. M. Sahebsara, T. Chen, and S. L. Shah. Optimal h2 filtering with random sensor delay, multiple packet dropout and uncertain observations. *Int. J. Control*, 80:292–301, 2007.
13. L. Shi, M. Epstein, A. Tiwari, and R. M. Murray. Estimation with information loss: Asymptotic analysis and error bounds. In *Proceedings of IEEE Conf. on Decision and Control*, pages 1215–1221, Dec 2005.
14. L. Shi, M. Epstein, A. Tiwari, and R. M. Murray. Kalman filtering over a packet dropping network: a probabilistic approach. In *Tenth International Conference on Control, Automation, Robotics and Vision, Dec 2008, Hanoi, Vietnam.*
15. B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.
16. S. Sun, L. Xie, W. Xiao, and Y. C. Soh. Optimal linear estimation for systems with multiple packet dropouts. *Automatica*, 44(5):1333–1342, May 2008.
17. S. C. Tatikonda. *Control Under Communication Constraints*. PhD thesis, Massachusetts Institute of Technology, 2000.
18. Z. Wang, D. W. C. Ho, and X. Liu. Robust filtering under randomly varying sensor delay with variance constraints. *IEEE Trans. Circuits Systems-II: Express briefs*, 51(6):320–326, 2004.
19. W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth-part i: State estimation problems. *IEEE Transactions on Automatic Control*, 42, Sept 1997.
20. W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth-part ii: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 44, May 1999.
21. L. Xie and L. Xie. Stability of a random riccati equation with markovian binary switching. *IEEE Transactions on Automatic Control*, 53(7):1759–1764, 2008.
22. E. Yaz and A. Ray. Linear unbiased state estimation for random models with sensor delay. In *Proceedings of IEEE Conf. on Decision and Control*, pages 47–52, Dec 1996.
23. H. Zhang and L. Xie. *Control and Estimation of Systems with Input/Output Delays*. Springer, 2007.
24. H. Zhang, L. Xie, D. Zhang, and Y. C. Soh. A re-organized innovation approach to linear estimation. *IEEE Transactions on Automatic Control*, 49(10):1810–1814, 2004.
25. ZigBee Alliance. http://www.zigbee.org/

Chapter 3
Distributed Optimal Delay Robustness and Network Throughput Tradeoff in Control-Communication Networks

Muhammad Tahir and Sudip K. Mazumder

Abstract This chapter presents a resource optimization framework for wireless networks used for information exchange among the nodes of a distributed control system. In particular, the proposed framework achieves an optimal tradeoff between end-to-end delay robustness and network throughput for given delay thresholds imposed by the control application layer. To select an objective function for delay robustness, which can provide the desired tradeoff optimally, we employ sensitivity analysis. For maximizing the network throughput, an effective link transmission rate based power control problem is solved. The proposed resource optimization framework is then extended using an iterative suboptimal cross-layer algorithm to improve the link congestion fairness. Performance evaluation results show that a small compromise in the network throughput can provide a large delay robustness depending on the operating point. An improvement in the link congestion fairness performance at the cost of reduced network throughput is also studied.

Keywords Resource optimization · Delay · Network throughput · Robustness · Sensitivity analysis · Transmission · Link-efficiency function · Distributed algorithm · Power control · Rate allocation · Iterative cross-layer algorithm

3.1 Introduction

Wireless networks are becoming popular for distributed networked control applications, ranging from sensor and actuator networks to distributed robotics as well as from building automation to medical services. Employing wireless networks for distributed control applications allows flexible system installation in addition to mobile operation, while reducing system maintenance costs. Design and development of such a distributed control system empowered by a wireless communication

S.K. Mazumder (✉)
Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA
e-mail: mazumder@ece.uic.edu

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_3,

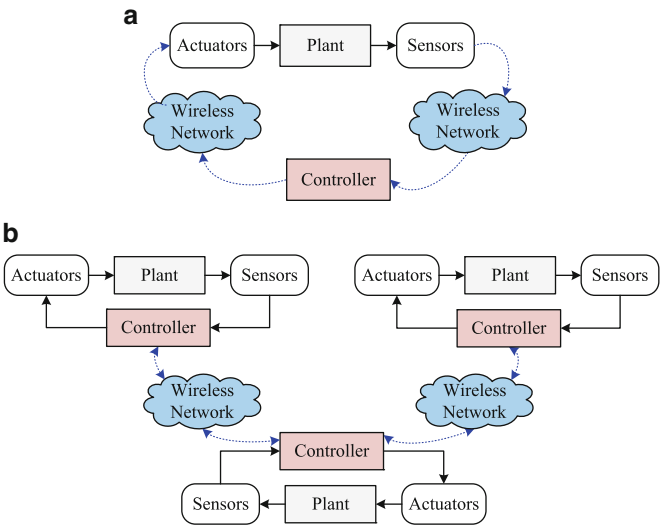


Fig. 3.1 Two cases of wireless communication network integration with distributed control system. (**a**) Control loop closed over the wireless network. (**b**) Different controllers coordinating using wireless communication network to achieve an overall objective of the distributed control system

network for information exchange is a challenging task, which requires new approaches to be explored. Wireless communication networks can be integrated with distributed control systems at different levels. Two such levels of integration are descried pictorially in Fig. 3.1.

Employing wireless networks for information exchange requires meeting the end-to-end delay threshold requirements imposed by the distributed control application layer [1–4]. Network control systems [1], hierarchical interactive communication-control networks [2,3], wireless multimedia sensor networks [4], to name a few, require delay guarantees from the underlying communication network to meet, not only the performance requirements of the application, but also ensure a stable system operation. For example, in case of distributed implementation of model predictive control [5], network performance is dependent on coordination among node controllers using internode information exchange. In this case, higher rate of information exchange improves the performance of distributed control system. In contrast, a decentralization approach as discussed in [6] aims to reduce communication overhead to achieve system robustness against communication delays at the expense of degraded performance.

Authors in [7] study the key components of time delays, in a networked distributed control system, to develop guidelines for obtaining the optimal sampling rates, which can ensure quality of service for communication network and quality of performance for distributed control system. The sampling rate of control system from wireless network standpoint is a parameter of application layer and determines the minimum rate requirements to be met by wireless communication network. Delay dependent performance of a distributed-control system [8] poses new

challenges requiring an optimal utilization of the underlying communication network resources [9]. For a given sampling rate and corresponding minimum-rate requirements by the application layer, the objective of a wireless network design is to maximize network resource utilization while meeting the end-to-end delay thresholds. But, an optimal resource utilization problem with proportional fairness as discussed in [10, 11] can provide a solution with end-to-end delays approaching delay thresholds for active delay constraints corresponding to smaller values of delay thresholds [9]. This results in optimal network throughput at the expense of vulnerable distributed control-system, which is prone to performance degradation and/or control-system instability due to the possibility of delay threshold violations for some active delay constraints.

To prevent such an event from happening, we have proposed a resource optimization framework, which provides delay margin (a measure of the gap between optimal end-to-end delay and the corresponding delay threshold) by introducing the end-to-end *delay robustness* parameter ε (will be simply called robustness parameter). The introduction of ε leads to delay margin at the expense of slightly degraded throughput performance of the wireless communication network. The parameter ε achieves an optimal tradeoff between network-throughput and delay robustness for a given delay threshold imposed by the application layer. In contrast to the decentralization approach in [6], which achieves robustness through reduced communication overhead, our approach provides robustness at the expense of slightly degraded communication network performance. The proposed approach can improve convergence performance of distributed control implementation by allowing faster communication among the nodes without compromising delay robustness.

To realize the above-mentioned tradeoff, we have formulated a resource optimization problem, which captures the delay robustness in the end-to-end delay constraints through parameter ε and the price for that robustness is penalized in the objective function for each transmission session between a source-destination pair. For a given throughput objective, we have used *sensitivity analysis* to define the robustness objective function leading to an optimal tradeoff between contending robustness and network-throughput parameters. In a relevant work, the authors in [12] have used an energy-robustness tradeoff while performing the distributed network power control. Our distributed resource optimization algorithm (DROA) achieves an optimal tradeoff between network-throughput and the delay robustness for any source-destination pair independently in contrast to the framework proposed in [12], which achieves the tradeoff at the network level by considering the total power. We also provide an efficient distributed power control based on the *link-efficiency function* [13]. When compared to link capacity-based power control [14], the proposed power control provides better end-to-end delay guarantees at the cost of degradation in the network-throughput performance.

The DROA is extended to an integrated cross-layer framework, which incorporates the effect of queuing delays, while evaluating the link weights, to achieve link congestion fairness. This results in further robustness exploiting the cross coupling between the network and the physical layers of the protocol stack [15]. Recently, the problem of joint power control at the physical layer and power-aware routing at the network layer is discussed in [15, 16]. The iterative solution proposed in [15]

adapts the routes after computing optimal powers without considering the effect of route switching on scheduling. On the other hand, the framework proposed in [16] is based on the idea of flow splitting by assigning the rates to the links based on their transmission power levels. The proposed iterative cross-layer algorithm (ICLA) is different from [15] as it is based on iterative updates of both the routes and transmission schedules after achieving convergence of DROA. In contrast to [16], we do not allow flow splitting to reduce the implementation complexity. In the proposed solution, once DROA is converged, we use the optimal power allocation as well as the current congestion price (in the form of queuing delay) to update the routes leading to an improved congestion fairness. The proposed solution is not limited to distributed control systems, rather can easily be extended for the case of wireless multimedia networks, where the quality of the multimedia stream as well as the buffer size requirements on the receiver node [4] is a function of end-to-end delay performance of the wireless communication network.

We provide a network model and construct the resource optimization problem incorporating the robustness tradeoff in detail in Sect. 3.2. In Sect. 3.3, the objective function for robustness is obtained first using sensitivity analysis, which is followed by the development of DROA for solving the resource optimization problem distributively, for a given set of routes and transmission schedules. Using the optimal parameters obtained from DROA, we next provide ICLA for joint routing, scheduling and resource allocation in Sect. 3.4. Numerical results for optimal tradeoff and convergence performance are provided in Sect. 3.5. Finally, we conclude our contributions and discuss some of the possible future research directions in Sect. 3.6.

3.2 System Model and Problem Formulation

In this section, we outline the network model and develop the constraint set and the multi-objective function leading to the network resource-optimization problem formulation.

3.2.1 Network Model

We model the wireless network as a *weighted directed graph* with N and L representing the sets of nodes and links, respectively. Communication from node n_j to node n_i is marked by the presence of link l_{ij} and will be abbreviated as l to simplify the notation. In a multi-hop wireless network, the transmission sessions are denoted by set S, where each transmission session $s_i \in S$ represents an ongoing transmission between a source-destination pair through the intermediate nodes. Each transmission session s_i is characterized by the following attributes:

- A shortest directed route consisting of a subset of links $L(s_i) \subseteq L \ \forall s_i$. Since routing is not the focus here, we use Dijkstra's algorithm [17] to determine the routes. The assignment of link weights for evaluating routes is explained later in this section.

- An end-to-end session rate $r_{s_i} \in \mathbf{r}$, where $\mathbf{r}$ is the set of rates for active transmission sessions.
- The minimum rate, $R_{\min}(s_i)$, and end-to-end delay threshold, $D_{\max}(s_i)\ \forall s_i$, requirement imposed by application layer (in our case given by performance/stability criteria of distributed system).

Different minimum data rates $R_{\min}(s_i)$ and maximum end-to-end delays $D_{\max}(s_i)$ required by different s_i constitute heterogeneous wireless network traffic. At the medium access control (MAC) layer, we define *transmission cycle* H to be the set of *transmission schedules* $h_i \in H$. Since multiple nodes can transmit concurrently, each h_i has an associated subset of simultaneously transmitting links $L(h_i) \subseteq L$. We allow simultaneous transmission between node pairs when Euclidean distance between the transmitter of link j and the receiver of link k, Δ_{kj}, satisfies $\nu \leq \frac{\Delta_{kj}}{\Delta_{kk}}$, where Δ_{kk} is Euclidean distance between the transmitter and receiver of link k and the choice of parameter ν, typically $2 \leq \nu \leq 4$, [18], is based on acceptable interference level. The inequality $\nu \leq \frac{\Delta_{kj}}{\Delta_{kk}}$ ensures that transmission range is less than interference range and depends on the value of parameter ν.

We define average transmission rate $\overline{R_l}$ in one transmission cycle as $\overline{R_l} = \sum_{h_i} I_l(h_i) R_l / |H|$. In the expression for $\overline{R_l}$, $|H|$ is the number of transmission schedules in one transmission cycle, R_l is the instantaneous transmission rate and $I_l(h_i)$ is an indicator function defined by

$$I_l(h_i) = \begin{cases} 1 & l \in L(h_i) \\ 0 & \text{otherwise} \end{cases}. \tag{3.1}$$

Packet success rate (PSR) at link l is a function of received signal-to-interference-and-noise-ratio (SINR) given by $\gamma_l(\mathbf{P}) = \frac{G_{ll} P_l}{z_l + \sum_{m \neq l} G_{lm} P_m}$, where G_{lm} represents the channel-gain from the transmitter of link m to the receiver of link l and P_l and z_l are, respectively, the transmitter power and the additive noise for link l. We require P_l to satisfy $0 \leq P_l \leq P_{\max} \forall P_l \in \mathbf{P}$, where $\mathbf{P}$ is the vector of link powers. We obtain PSR from *link-efficiency function* $\eta_l(\gamma_l(\mathbf{P}))$ [13, 19], which is an increasing, continuous and S-shaped (sigmoidal [20]) function with $\eta_l(\infty) = 1$. The link-efficiency function for narrow-band modulation also satisfies $\eta_l(0) = 0$ and is valid for many practical cases [13]. Now, the effective data rate at link l is obtained by scaling average data transmission rate $\overline{R_l}$ with PSR and is $\overline{R_l}\eta_l(\gamma_l(\mathbf{P}))$.

To construct network topology link weights are required, which can be chosen using channel gains for some fixed link transmission power level P_l initially. This choice is meant to construct the network topology based on the physical distribution of the nodes. A better choice for link weights would have been based on the link SINR but that leads to an almost intractable situation. For instance, the presence or absence of a link in the network topology based on link SINR will have a two-way coupling, i.e., when a link is added it will affect the SINR of existing links and if SINR of some of the existing links become infeasible they will be removed from the topology and all other links need to be reevaluated. This will take long time and

will not ensure a fixed network topology. Due to this reason we define link weight, w_l, for evaluating the routes, using the receiver power as follows

$$w_l = \begin{cases} \frac{1}{G_{ll} P_l} & 1/\left(G_{ll} P_l\right) \leq \theta_l \\ \infty & \text{otherwise} \end{cases} \quad \forall l. \tag{3.2}$$

The parameter θ_l is user defined threshold based on the physical proximity of the nodes, which controls the range within which a node can communicate directly to other nodes. A possible initialization for w_l in (3.2) can be obtained using $P_l = P_{\max}/2$.

3.2.2 Link-Efficiency Function

To obtain an expression for link-efficiency function, which is an effective measure of the PSR, we first quantify link packet-error rate (PER). The PER, for many modulation and channel-coding schemes, can be well approximated by the following family of functions [19]:

$$\text{PER}(\gamma_l(\mathbf{P})) = \left(1 + \mathrm{e}^{b(\gamma_l^{(dB)}(\mathbf{P}) - \sigma)}\right)^{-1}. \tag{3.3}$$

In (3.3), $\gamma_l^{(dB)}(\mathbf{P}) = 10\log_{10}(\gamma_l(\mathbf{P}))$ and b and σ are fitting parameters, which mainly depend on the modulation type, channel-coding scheme and the packet length used, and can be obtained offline. Figure 3.2 shows an example packet-error rate curve as a function of link SINR γ_l for $b = 0.8$ and $\sigma = 5$. It should be noted that the PER function is convex in γ_l for $\gamma_l(\mathbf{P}) > \gamma_{\min}(l)$ but not in P_l. When $\text{PER}(\gamma_l(\mathbf{P}))$ is small, we can approximate (3.3) using

$$\begin{aligned} \text{PER}(\gamma_l(\mathbf{P})) &\approx \mathrm{e}^{-b(\gamma_l^{(dB)}(\mathbf{P}) - \sigma)}, \\ &= \mathrm{e}^{b\sigma}\left(\gamma_l\left(\mathbf{P}\right)\right)^{-mb}, \end{aligned} \tag{3.4}$$

where the constant m is $10(\log_e 10)^{-1}$. Now the link-efficiency function is obtained as

$$\eta_l(\gamma_l(\mathbf{P})) \approx 1 - \mathrm{e}^{b\sigma}\left(\gamma_l\left(\mathbf{P}\right)\right)^{-mb}. \tag{3.5}$$

3.2.3 End-to-End Delay Model

For delay model, we first focus on a single link. Average delay at link l due to queuing and transmission, when using M/D/1 queuing model [17] and packet length of μ_{s_i} for s_i, is obtained as

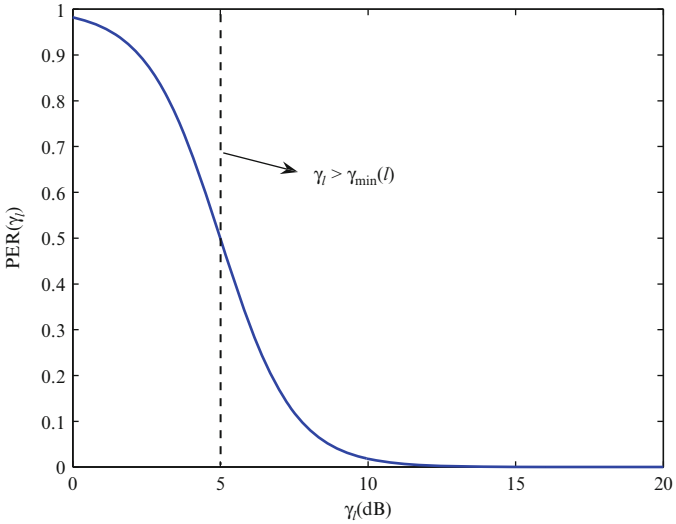


Fig. 3.2 PER as a function of link SINR. The dashed line marks the region for which link SINR is larger than a certain threshold and as a consequence ensures the convexity of the PER in (3.3) for the region $\gamma_l > \gamma_{\min}(l)$.

$$\frac{\mu_{s_i}}{2}\left\{\frac{1}{\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))-\sum_{s_i:\, l\in L(s_i)} r_{s_i}}+\frac{1}{\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))}\right\}, \tag{3.6}$$

where $\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))$ and $\sum_{s_i:\, l\in L(s_i)} r_{s_i}$ represent the effective transmission and average arrival rates, respectively, for link l. Our link delay formulation is different from the one proposed in [10], where the authors have modeled the average link-transmission-rate using the link capacity. It is well known that capacity achieving codes require large word lengths and complex processing and hence cannot provide delay guarantees. On the other hand, our formulation models average link-transmission-rate as the multiplication of $\overline{R}_l$ and $\eta_l(\gamma_l(\mathbf{P}))$ to obtain average link delay. Using (3.6), we obtain end-to-end delay bounded by $D_{\max}(s_i)$ for transmission session s_i by accumulating the link delays along the shortest route $L(s_i)$ as

$$\frac{1}{2}\sum_{l\in L(s_i)}\left\{\frac{\mu_{s_i}}{\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))-\sum_{s_i:\, l\in L(s_i)} r_{s_i}}+\frac{\mu_{s_i}}{\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))}\right\}\leq D_{\max}(s_i). \tag{3.7}$$

In (3.7), first term approximately measures the waiting time in the queue and second term corresponds to the transmission delay. The possibility of an individual link being member of different shortest routes requires a systematic procedure to adapt $D_{\max}(s_i)$ by the application. This is achieved by first perturbing $D_{\max}(s_i)$ for some s_i and then performing sensitivity analysis leading to the selection of an optimal delay robustness objective function.

3.2.4 Resource Optimization Problem

To formulate the wireless-network resource-optimization problem, we introduce the link transmission $d_l^{(t)} \in \mathbf{d}^{(t)}$ and queuing $d_l^{(q)} \in \mathbf{d}^{(q)}$ delay auxiliary variables to decompose the inequality in (3.7) into the following two inequalities

$$\sum_{l \in L(s_i)} \left(d_l^{(t)} + d_l^{(q)} \right) \leq D_{\max}(s_i), \tag{3.8}$$

$$\frac{\mu_{s_i}/2}{\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \sum_{s_i:\, l \in L(s_i)} r_{s_i}} \leq d_l^{(q)}, \quad \frac{\mu_{s_i}/2}{\overline{R}_l \eta_l(\gamma_l(\mathbf{P}))} \leq d_l^{(t)}. \tag{3.9}$$

We now introduce the robustness parameter $\varepsilon_{s_i} \in \boldsymbol{\Upsilon}$, and $\varepsilon_{s_i} \in [0, 1)$ for distributed system application layer by modifying the delay constraint in (3.8) as $\sum_{l \in L(s_i)} (d_l^{(t)} + d_l^{(q)}) \leq (1 - \varepsilon_{s_i}) D_{\max}(s_i)$. The parameter ε_{s_i} is upper bounded by 1 due to the fact that the transmission rate cannot be increased arbitrarily and it is not allowed to take on negative values to avoid system instability due to delay threshold violation. In contrast to the approaches providing delay guarantees [21], the robustness parameter provides delay margin against wireless communication network performance fluctuations mainly due to time-varying channel gains, link congestion, route switching and allows the distributed system to respond to these fluctuations. Next, we formulate constrained resource optimization problem to achieve the tradeoff between robustness and network-throughput:

$$\textbf{maximize} \quad J = \sum_{s_i} \left\{ \alpha_{s_i} U(r_{s_i}) + (1 - \alpha_{s_i}) \phi(\varepsilon_{s_i}) \right\}, \tag{3.10}$$

$$\textbf{s.t.} \quad \frac{1}{D_{\max}(s_i)} \sum_{l \in L(s_i)} \left(d_l^{(t)} + d_l^{(q)} \right) \leq (1 - \varepsilon_{s_i}), \quad \forall s_i \tag{3.11}$$

$$\frac{\mu_{s_i}}{2 d_l^{(q)}} \leq \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \sum_{s_i:\, l \in L(s_i)} r_{s_i} \right), \quad \frac{\mu_{s_i}}{2 d_l^{(t)}} \leq \overline{R}_l \eta_l(\gamma_l(\mathbf{P})), \quad \forall l \tag{3.12}$$

$$0 \leq d_l^{(t)}, d_l^{(q)}, \quad \gamma_{\min}(l) \leq \gamma_l(\mathbf{P}), \quad 0 \leq P_l \leq P_{\max}, \quad \forall l \tag{3.13}$$

$$0 \leq \varepsilon_{s_i} < 1, \quad R_{\min}(s_i) \leq r_{s_i} \quad \forall s_i. \tag{3.14}$$

In (3.10), J is the overall objective function, $U(.)$ and $\phi(.)$ are concave functions of their respective parameters, α_{s_i} is the user-defined tradeoff parameter to achieve a desired level of robustness. In (3.13), $\gamma_{\min}(l)$ is the received SINR threshold at link l to ensure the convexity of PER. For a given $U(r_{s_i})$, the objective function $\phi(\varepsilon_{s_i})$ will be used to modulate the robustness-throughput tradeoff by adjusting ε_{s_i}. We will derive an expression for $\phi(\varepsilon_{s_i})$ using sensitivity analysis, which will

achieve an optimal robustness-throughput tradeoff. It is worth mentioning that setting $\alpha_{s_i} = 1$ and replacing the link-efficiency function with fixed link capacity leads to the simplified version of the problem discussed in [10]. The resource-optimization problem discussed in [10], and its extended version with power control in [11], lead to an optimal solution where $\sum_{l \in L(s_i)} (d_l^{(t)} + d_l^{(q)}) \to D_{\max}(s_i)$ for some s_i. This leaves negligible margin for the application layer to respond to any network topological variations or time-varying channel gains and is one of the motivations leading to the robustness-throughput optimal tradeoff problem in (3.10)–(3.14).

3.3 Solution Approach and Distributed Algorithm

The resource optimization problem in (3.10)–(3.14) is nonlinear due to the link-efficiency function in (3.5), the choice of the objective functions, and the structure of the link delay inequalities in (3.12). A distributed solution to the resource optimization problem will be achieved by first decomposing the problem into subproblems and then solving the individual subproblems coupled through the dual variables. The introduction of auxiliary variables and usage of 'log' transformation will translate each of the nonlinear subproblems to an equivalent convex form. To obtain this, we first associate dual variables $\lambda_l \in \mathbf{\Lambda}$, $\xi_l \in \mathbf{\Xi}$ and $\psi_{s_i} \in \mathbf{\Psi}$ with end-to-end, queuing and transmission delays in (3.11)–(3.12), respectively, to form the Lagrangian given by

$$\begin{aligned}
&\textbf{maximize} \qquad L(\mathbf{r}, \mathbf{P}, \mathbf{d}^{(q)}, \mathbf{d}^{(t)}, \mathbf{\Upsilon}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi}) \\
&= \Bigg\{ \sum_{s_i} \left(\alpha_{s_i} U(r_{s_i}) + (1-\alpha_{s_i})\phi(\varepsilon_{s_i})\right) + \sum_{s_i} \psi_{s_i} \Bigg((1-\varepsilon_{s_i}) - \frac{1}{D_{\max}(s_i)} \\
&\qquad \sum_{l \in L(s_i)} \left(d_l^{(t)} + d_l^{(q)}\right) \Bigg) + \sum_l \xi_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \frac{\mu_{s_i}}{2d_l^{(t)}} \right) \\
&\qquad + \sum_l \lambda_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \sum_{s_i:\, l \in L(s_i)} r_{s_i} - \frac{\mu_{s_i}}{2d_l^{(q)}} \right) \Bigg\} \mid \gamma_{\min}(l) \le \gamma_l(\mathbf{P}), \\
&0 \le d_l^{(t)}, d_l^{(q)},\ 0 \le P_l \le P_{\max},\ 0 \le \varepsilon_{s_i} < 1,\ R_{\min}(s_i) \le r_{s_i}. \qquad (3.15) \\
&= \Bigg[\Bigg\{ \sum_{s_i} \left(\alpha_{s_i} U(r_{s_i}) - \sum_{l \in L(s_i)} \lambda_l r_{s_i} \right) \mid R_{\min}(s_i) \le r_{s_i} \Bigg\}
\end{aligned}$$

$$+\left\{\sum_{s_i}\left((1-\alpha_{s_i})\phi(\varepsilon_{s_i})+\psi_{s_i}(1-\varepsilon_{s_i})\right)\,\middle|\,0\le\varepsilon_{s_i}<1\right\}$$
$$-\left\{\sum_{l}\left(\frac{\lambda_l\mu_{s_i}}{2d_l^{(q)}}+\frac{\xi_l\mu_{s_i}}{2d_l^{(t)}}+\sum_{s_i:\,l\in L(s_i)}\psi_{s_i}\frac{d_l^{(t)}+d_l^{(q)}}{D_{\max}(s_i)}\right)\,\middle|\,0\le d_l^{(t)},d_l^{(q)}\right\}$$
$$+\left\{\sum_{l}\left((\lambda_l+\xi_l)\overline{R}_l\eta_l(\gamma_l(\mathbf{P}))\right)\,|\,\gamma_{\min}(l)\le\gamma_l(\mathbf{P}),\ 0\le P_l\le P_{\max}\right\}\Bigg]. \quad (3.16)$$

The maximization problem in (3.16) is decomposable into rate $r_{s_i}\in\mathbf{r}$, delay $d_l^{(t)}\in\mathbf{d}^{(t)}$ and $d_l^{(q)}\in\mathbf{d}^{(q)}$, the robustness control $\varepsilon_{s_i}\in\boldsymbol{\Upsilon}$ and the link transmitter power control $P_l\in\mathbf{P}$ subproblems. The associated dual problem is

$$\textbf{minimize}\quad g(\boldsymbol{\Lambda},\boldsymbol{\Xi},\boldsymbol{\Psi})\quad \textbf{s.t}\quad \lambda_l,\xi_l,\psi_{s_i}\ge 0\quad\forall\, l,\ s_i. \quad (3.17)$$

In (3.17), $g(\boldsymbol{\Lambda},\boldsymbol{\Xi},\boldsymbol{\Psi})=L(\mathbf{r}^*,\mathbf{P}^*,\mathbf{d}^{*(q)},\mathbf{d}^{*(t)},\boldsymbol{\Upsilon}^*,\boldsymbol{\Lambda},\boldsymbol{\Xi},\boldsymbol{\Psi})=$ **maximize** $L(\mathbf{r},\mathbf{P},\mathbf{d}^{(q)},\mathbf{d}^{(t)},\boldsymbol{\Upsilon},\boldsymbol{\Lambda},\boldsymbol{\Xi},\boldsymbol{\Psi})$ and $\mathbf{r}^*,\ \mathbf{P}^*,\ \mathbf{d}^{*(q)},\mathbf{d}^{*(t)}$ and $\boldsymbol{\Upsilon}^*$ are the optimal primal variables obtained by solving (3.16). The block diagram representation in Fig. 3.3 shows a possible realization of DROA by decomposing the original problem into subproblems using dual decomposition. The distributed realization for DROA can be

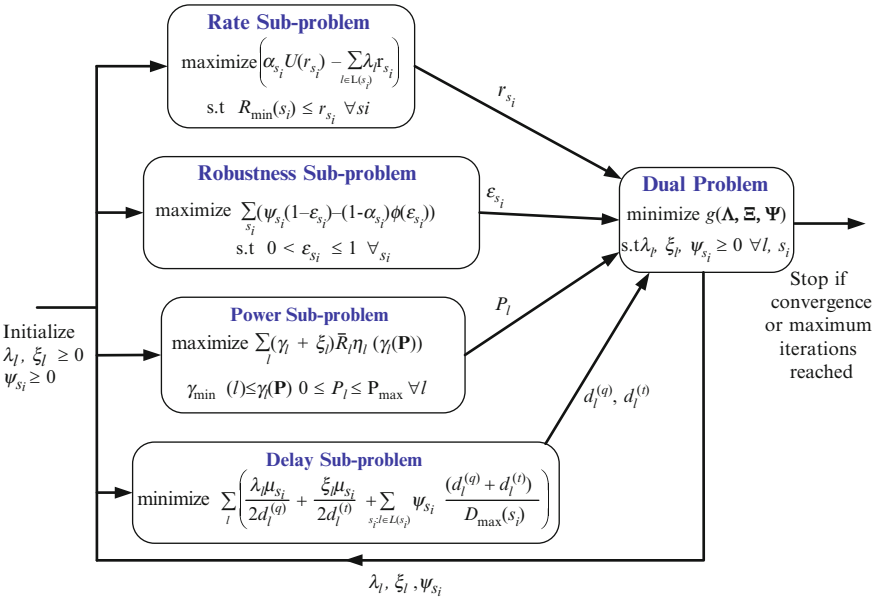


Fig. 3.3 Block diagram representation for distributed implementation of the network resource-optimization problem using dual decomposition

initialized by constructing the network topology using $P_l = P_{\max}/2$ along with an associated feasible transmission schedule $h_i \in H$. Next, we will discuss the solution approaches for these subproblems and their distributed implementation. The proof of convergence for the distributed realization of DROA is provided in Appendix A.

3.3.1 Robustness Subproblem

The robustness subproblem from (3.16) is described by

$$\begin{aligned} \textbf{maximize} \quad & \sum_{s_i} \left((1-\alpha_{s_i})\phi(\varepsilon_{s_i}) + \psi_{s_i}(1-\varepsilon_{s_i})\right) \\ \textbf{s.t.} \quad & 0 \leq \varepsilon_{s_i} < 1. \end{aligned} \tag{3.18}$$

As pointed out earlier, for a given $U(r_{s_i})$, which is $\log(r_{s_i})$ in our case to achieve proportional throughput fairness [14], the objective function $\phi(\varepsilon_{s_i})$ is responsible for modulating the robustness-throughput tradeoff. To achieve this tradeoff optimally, we need to choose an appropriate $\phi(\varepsilon_{s_i})$. For that purpose, as a first step we fix $\alpha_{s_i}=1\ \forall s_i$ in (3.10)–(3.14) and use sensitivity analysis to study the effect of perturbing the end-to-end delay threshold $D_{\max}(s_i)$ on the optimal network-throughput.

Step 1: Sensitivity Analysis

We perturb end-to-end delay constraint for s_i^{th} session by $u_{s_i} \in \mathbf{u}$ and observe its effect on optimal network-throughput $\rho^*(\mathbf{u})$ by solving the following perturbed optimization problem:

$$\begin{aligned} \rho^*(\mathbf{u}) = \textbf{maximize} \quad & \sum_{s_i} U(r_{s_i}) \frac{1}{D_{\max}(s_i)} \sum_{l \in L(s_i)} \left(d_l^{(t)} + d_l^{(q)}\right) \leq u_{s_i}, \quad \forall s_i \\ & \text{and constraints (3.12)} - \text{(3.14)}. \end{aligned} \tag{3.19}$$

The $u_{s_i}=1\ \forall s_i$ represents the unperturbed system, while $0<u_{s_i}\leq 1$ and $u_{s_i}>1$, respectively, correspond to tightening (throughput reduction) and loosening (throughput increment) the s_i^{th} constraint. If $\psi_{s_i}^*$, corresponding to $\rho^*(\mathbf{1})$, represent the optimal value of the Lagrange multipliers associated with the unperturbed end-to-end delay constraints, then the fractional change in the optimal network-throughput due to s_i^{th} constraint perturbation is obtained as

$$\begin{aligned} \frac{\rho^*(\mathbf{u}) - \rho^*(\mathbf{1})}{\rho^*(\mathbf{1})} &= \frac{\rho^*(u_{s_i}\mathbf{e}_{s_i}) - \rho^*(\mathbf{1})}{\rho^*(\mathbf{1})}, \\ &= (u_{s_i}-1)\frac{\partial\rho^*(\mathbf{1})/\partial u_{s_i}}{\rho^*(\mathbf{1})} + o(u_{s_i}), \\ &= (u_{s_i}-1)(\psi_{s_i}^*/\rho^*(\mathbf{1})) + o(u_{s_i}) \approx (u_{s_i}-1)\frac{\psi_{s_i}^*}{\rho^*(\mathbf{1})}. \end{aligned} \tag{3.20}$$

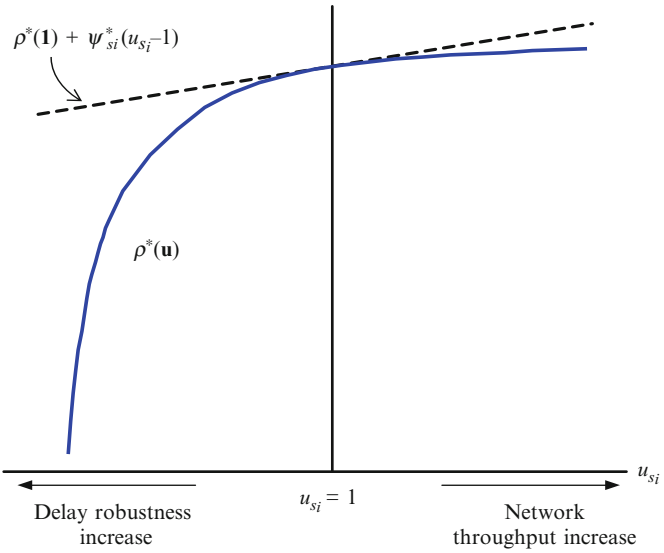


Fig. 3.4 Robustness-throughput tradeoff using sensitivity analysis. The optimal value $\rho^*(\mathbf{1})$ correspond to $u_{s_i} = 1\ \forall s_i$

In the first equality of (3.20), $\mathbf{e}_{s_i}$ is a vector with all its entries equal to 0 with the exception of s_i^{th} entry, which is 1. The second equality follows from the Taylor series expansion and in the third equality we have used the fact that $\partial\rho^*(\mathbf{1})/\partial u_{s_i} = \psi^*_{s_i}$ [20], followed by the first-order approximation. Figure 3.4 shows the effect of varying u_{s_i}, for session s_i, from its nominal value on the optimal network-throughput performance. The supporting hyper-plane $\rho^*(\mathbf{1}) + \psi^*_{s_i}(u_{s_i} - 1)$ shown in Fig. 3.4 at $\rho^*(\mathbf{1})$ with gradient $\psi^*_{s_i}$ illustrates the result in (3.20).

Step 2: Choice of Delay Tradeoff Objective Function

Differentiating the objective function in (3.18) with respect to ε_{s_i}, setting it equal to zero and evaluating at optimal point, we obtain

$$\partial\phi(\varepsilon_{s_i})/\partial\varepsilon_{s_i}\ |_{\varepsilon_{s_i}=\varepsilon^*_{s_i}} = -\psi^*_{s_i}/(1-\alpha_{s_i}), \tag{3.21}$$

which gives $\psi^*_{s_i} = -(1-\alpha_{s_i})\partial\phi(\varepsilon_{s_i})/\partial\varepsilon_{s_i}|_{\varepsilon_{s_i}=\varepsilon^*_{s_i}}$. Now replacing u_{s_i} with $(1-\varepsilon_{s_i})$ (for $u_{s_i} \in (0, 1]$) in the expression for fractional change in throughput in (3.20) and using $\psi^*_{s_i} = -(1-\alpha_{s_i})\partial\phi(\varepsilon_{s_i})/\partial\varepsilon_{s_i}|_{\varepsilon_{s_i}=\varepsilon^*_{s_i}}$, we have

$$(1-\alpha_{s_i})\left(\frac{\varepsilon_{s_i}}{\rho^*(\mathbf{0})}\right)\frac{\partial\phi(\varepsilon_{s_i})}{\partial\varepsilon_{s_i}}\Big|_{\varepsilon_{s_i}=\varepsilon^*_{s_i}}, \tag{3.22}$$

where $\rho^*(\mathbf{0})$ is obtained by mapping $\rho^*(\mathbf{1})$ from u_{s_i} to ε_{s_i} domain. The reason for mapping u_{s_i} to ε_{s_i} in the interval $(0, 1]$ is to ensure that $D_{\max}(s_i)$, defined by the application layer, is not violated. If the maximum throughput fraction available for tradeoff with the end-to-end delay, for s_i^{th} session is denoted by δ_{s_i}, then equating (3.22) to this maximum available throughput fraction results in

$$\frac{\partial \phi(\varepsilon_{s_i})}{\partial \varepsilon_{s_i}} = \frac{\rho^*(\mathbf{0})}{(1-\alpha_{s_i})} \frac{\delta_{s_i}}{\varepsilon_{s_i}}, \tag{3.23}$$

and integrating (3.23) gives

$$\phi(\varepsilon_{s_i}) = \frac{\rho^*(\mathbf{0})}{(1-\alpha_{s_i})} \delta_{s_i} \log(\varepsilon_{s_i}). \tag{3.24}$$

For the unperturbed problem, the optimal value of the objective function in (3.19) is $\rho^*(\mathbf{1}) = \sum_{s_i} U(r_{s_i}^*)$ and for the s_i^{th} session the optimal throughput is $r_{s_i}^*$. Now if we define the maximum throughput fraction, δ_{s_i}, as $\delta_{s_i} = a r_{s_i}^*$ where the normalizing as well as scaling constant, a is given as $\frac{0.1}{R_{\min}(s)} < a < \frac{0.9}{R_{\min}(s)}$. Now since the magnitude of $\rho^*(\mathbf{0})$ is same as $\rho^*(\mathbf{1})$, we have $\phi(\varepsilon_{s_i}) = \frac{\rho^*(\mathbf{1})}{1-\alpha_{s_i}} a r_{s_i}^* \log(\varepsilon_{s_i})$. For the parameter values given in Sect. 3.5, $\rho^*(\mathbf{0}) = 23.4$. Now each session s_i will have a corresponding optimal value $r_{s_i}^*$ for unperturbed optimization case, which can be used to determine $\phi(\varepsilon_{s_i})$ to solve the robustness subproblem. Using (3.24), the robustness-throughput tradeoff subproblem becomes

$$\textbf{maximize} \sum_{s_i} (\rho^*(\mathbf{0}) \delta_{s_i} \log(\varepsilon_{s_i}) + \psi_{s_i}(1 - \varepsilon_{s_i})) \ \textbf{s.t.}\ 0 \leq \varepsilon_{s_i} < 1.$$

This problem can be solved distributively in ε_{s_i} using efficient algorithms for convex optimization [20]. It should be noted that we require the optimal throughput without delay robustness, $\rho^*(\mathbf{0})$ to evaluate the objective of the robustness-throughput tradeoff subproblem.

3.3.2 Power Control Subproblem

The objective of maximizing accumulated link transmission rates for all active links in an interference-limited wireless-network leads to the following coupled and constrained power control problem:

$$\begin{aligned} \textbf{maximize} \quad & \sum_{l} \left((\lambda_l + \xi_l) \overline{R}_l \eta_l(\gamma_l(\mathbf{P})) \right) \\ \textbf{s.t.} \quad & \gamma_{\min}(l) \leq \gamma_l(\mathbf{P}),\ 0 \leq P_l \leq P_{\max} \quad \forall l \end{aligned} \tag{3.25}$$

From (3.5), the maximization problem in (3.25) is equivalent to the following minimization problem

$$\begin{aligned} \textbf{minimize} \quad & \sum_{l} \left((\lambda_l + \xi_l) \overline{R}_l \mathrm{e}^{b\sigma} (\gamma_l(\mathbf{P}))^{-mb} \right) \\ \textbf{s.t.} \quad & \gamma_{\min}(l) \leq \gamma_l(\mathbf{P}),\ 0 \leq P_l \leq P_{\max} \ \ \forall l \end{aligned} \tag{3.26}$$

The problem in (3.26) is not convex optimization problem but can be transformed into a convex problem as explained in the sequel. By introducing the auxiliary variables t_l, we rewrite the problem in (3.26) in *epigraph form* [20] as

$$\begin{aligned} \textbf{minimize} \quad & \sum_{l} (\lambda_l + \xi_l) \overline{R}_l \mathrm{e}^{b\sigma} t_l \\ \textbf{s.t.} \quad & (\gamma_l(\mathbf{P}))^{-mb} \leq t_l, \ \ \gamma_{\min}(l) \leq \gamma_l(\mathbf{P}), \ \ 0 \leq P_l \leq P_{\max} \ \ \forall l. \end{aligned} \tag{3.27}$$

Next, we first apply 'log' transformation to the inequality constraint $(\gamma_l(\mathbf{P}))^{-mb} \leq t_l, \ \forall l$, and then define new variables $\tilde{t}_l = \log(t_l)$ and $\tilde{\gamma}_{\min}(l) = \log(\gamma_{\min}(l)) \ \forall l$. Now, the problem in (3.27) can be rewritten as

$$\begin{aligned} \textbf{minimize} \quad & \sum_{l} (\lambda_l + \xi_l) \overline{R}_l \mathrm{e}^{b\sigma} \mathrm{e}^{\tilde{t}_l} \\ \textbf{s.t.} \quad & -\frac{\tilde{t}_l}{mb} \leq \log(\gamma_l(\mathbf{P})), \ \ \tilde{\gamma}_{\min}(l) \leq \log(\gamma_l(\mathbf{P})), \ \ 0 \leq P_l \leq P_{\max} \ \forall l. \end{aligned} \tag{3.28}$$

The constraint $-\frac{\tilde{t}_l}{mb} \leq \log(\gamma_l(\mathbf{P}))$ is tight near optimal solution. This is true because $t_l \ll 1$ for high SINR (see 3.27) and hence $\tilde{t}_l \ll 0$ when the objective function of the form $\mathrm{e}^{\tilde{t}_l}$ will be minimized. Now, the feasibility of the problem in (3.28) will require $-\frac{\tilde{t}_l}{mb} \geq \tilde{\gamma}_{\min}(l)$ due to the tightness of the constraint $-\frac{\tilde{t}_l}{mb} \leq \log(\gamma_l(\mathbf{P}))$ compared to the box constraint $\tilde{\gamma}_{\min}(l) \leq \log(\gamma_l(\mathbf{P}))$. This allows us to combine the two constraints $-\frac{\tilde{t}_l}{mb} \leq \log(\gamma_l(\mathbf{P}))$ and $\tilde{\gamma}_{\min}(l) \leq \log(\gamma_l(\mathbf{P}))$ as $\log(\gamma_l(\mathbf{P})) \geq -\frac{\tilde{t}_l}{mb} \geq \tilde{\gamma}_{\min}(l)$ and can be rewritten as $\log(\gamma_l(\mathbf{P})) \geq -\frac{\tilde{t}_l}{mb}$ and $\tilde{t}_l \leq -(mb)\tilde{\gamma}_{\min}(l)$. Next, we assign Lagrange multipliers $\pi_l \in \mathbf{\Pi} \ \forall l$ to the coupling constraints $-\frac{\tilde{t}_l}{mb} \leq \log(\gamma_l(\mathbf{P})), \ \forall \ l$ to obtain

$$\begin{aligned} \textbf{minimize} \quad & L_{\mathbf{P}}(\tilde{t}_l, \mathbf{P}) = \sum_{l} (\lambda_l + \xi_l) \overline{R}_l \mathrm{e}^{b\sigma} \mathrm{e}^{\tilde{t}_l} - \sum_{l} \pi_l \left(\log(\gamma_l(\mathbf{P})) + \frac{\tilde{t}_l}{mb} \right) \\ \textbf{s.t.} \quad & \tilde{t}_l \leq -(mb)\tilde{\gamma}_{\min}(l),\ 0 \leq P_l \leq P_{\max} \quad \forall l. \end{aligned} \tag{3.29}$$

The dual problem corresponding to the minimization problem in (3.29) is given by **maximize** $g_{\mathbf{P}}(\mathbf{\Pi})$, **s.t.** $\pi_l \geq 0 \ \forall l$, where $g_{\mathbf{P}}(\mathbf{\Pi}) = L_{\mathbf{P}}(\tilde{t}_l^*, \mathbf{P}^*)$ and $\tilde{t}_l^*$ and $\mathbf{P}^*$ are optimal values of the corresponding variables. The problem in (3.29) is decomposable into two subproblems in primal variables $\tilde{t}_l$ and $P_l \ \forall l$ as

$$\textbf{minimize} \quad \sum_{l} \left((\lambda_l + \xi_l) \overline{R}_l \mathrm{e}^{b\sigma} \mathrm{e}^{\tilde{t}_l} - \pi_l \frac{\tilde{t}_l}{mb} \right) \quad \textbf{s.t.} \quad \tilde{t}_l \leq -(mb)\tilde{\gamma}_{\min}(l) \quad \forall l, \tag{3.30}$$

$$\textbf{maximize} \quad \sum_{l} \pi_l \log(\gamma_l(\mathbf{P})) \quad \textbf{s.t.} \quad 0 \leq P_l \leq P_{\max} \quad \forall l. \tag{3.31}$$

For a given value of the dual variables π_l, the two subproblems in (3.30) and (3.31) are disjoint in primal variables $\tilde{t}_l$ and P_l $\forall l$, and can be solved independently. At each update of π_l through projected subgradient, algorithm regulates the two subproblems in (3.30) and (3.31) toward an overall optimization given in (3.29).

Alternatively, one can also construct an unconstrained optimization problem corresponding to the one in (3.29) by assigning two additional dual variables to the constraints $\tilde{t}_l \leq -(mb)\tilde{\gamma}_{\min}(l)$ and $0 \leq P_l \leq P_{\max}$ $\forall l$, which can be decomposed again into two unconstrained subproblems in two primal variables $\tilde{t}_l$ and P_l $\forall l$. The unconstrained subproblems constructed this way can be solved independently and feasibility of the problem will ensure its convergence. For further reading on dual decomposition, we refer the reader to ([20], Chap. 5), [23].

The convexity of the subproblem in (3.30) can be verified by the condition $-\frac{1}{\tilde{t}_l}\partial f(\tilde{t}_l)/\partial \tilde{t}_l \leq \partial^2 f(\tilde{t}_l)/\partial \tilde{t}_l^2$, where $f(\tilde{t}_l)$ is the objective function in (3.30). The subproblem in (3.31) is not convex but can be transformed into an equivalent convex form by 'log' transformation of power vector $\mathbf{P}$ [14]. For that, we first define $\hat{P}_l = \log(P_l)$ $\forall P_l \in \mathbf{P}$ and then evaluate the Hessian of (3.31). Once the convexity is ensured, the gradient of (3.31) is used to update P_l $\forall l$. The associated dual problem, **maximize** $g_{\mathbf{P}}(\mathbf{\Pi})$, **s.t.** $\pi_l \geq 0$ $\forall l$, is solved by using the following subgradient update

$$\pi(k+1) = \left[\pi(k) - \beta_\pi(k) \left(\log(\gamma_l(\mathbf{P})) + \frac{\tilde{t}_l}{mb} \right) \right]^+. \tag{3.32}$$

In (3.32), $\beta_\pi(k)$ is the step size at k^{th} step and $[x]^+$ is defined as $\max\{0, x\}$. How to make an appropriate choice for the step size is discussed in Sect. 3.3.5. A proof of convergence for the distributed power control can be drawn on the same lines as given in Appendix A.

3.3.3 Rate Allocation Subproblem

The rate allocation subproblem from (3.16) is given by

$$\textbf{maximize} \quad \sum_{s_i} \left(\alpha_{s_i} U(r_{s_i}) - \sum_{l \in L(s_i)} \lambda_l r_{s_i} \right) \quad \textbf{s.t.} \quad R_{\min}(s_i) \leq r_{s_i}. \tag{3.33}$$

The objective function $U(r_{s_i})$ is defined to be concave for proportional fairness among different transmission sessions and is $U(r_{s_i}) = \log(r_{s_i})$. The rate subproblem in (3.33) is separable in r_{s_i} and can be solved for each r_{s_i} separately. It turns out that a closed-form solution for this problem is possible as discussed in the sequel. Introducing the multipliers ϑ_{s_i} for the rate constraint $R_{\min}(s_i) \leq r_{s_i}$, we obtain the KKT conditions for the subproblem in (3.33) as follows:

$$\begin{aligned} R_{\min}(s_i) \leq r^*_{s_i}, \quad 0 \leq \vartheta^*_{s_i} \quad \forall s_i, \\ \left(r^*_{s_i} - R_{\min}(s_i)\right) \vartheta^*_{s_i} = 0 \quad \forall s_i, \\ \frac{\alpha_{s_i}}{r^*_{s_i}} - \sum_{l \in L(s_i)} \lambda_l + \vartheta^*_{s_i} = 0 \quad \forall s_i. \end{aligned} \tag{3.34}$$

In (3.34), x^* shows the optimal value of the variable x. It is observed that $\vartheta^*_{s_i}$ $\forall s_i$ are slack variables and can be eliminated, reducing (3.34) to

$$R_{\min}(s_i) \leq r^*_{s_i}, \quad \frac{\alpha_{s_i}}{r^*_{s_i}} \leq \sum_{l \in L(s_i)} \lambda_l \quad \forall s_i, \tag{3.35}$$

$$\left(\sum_{l \in L(s_i)} \lambda_l - \frac{\alpha_{s_i}}{r^*_{s_i}} \right) \left(r^*_{s_i} - R_{\min}(s_i)\right) = 0 \quad \forall s_i. \tag{3.36}$$

Now if $R_{\min}(s_i) \geq \left(\alpha_{s_i} / (\sum_{l \in L(s_i)} \lambda_l)\right)$, then the condition in (3.36) will hold for $r^*_{s_i} = R_{\min}(s_i)$; and for the opposite case it will hold for $r^*_{s_i} = \alpha_{s_i} \left(\sum_{l \in L(s_i)} \lambda_l\right)^{-1}$. So we have a closed-form solution, for $r^*_{s_i}$, given by

$$r^*_{s_i} = \max \left\{ R_{\min}(s_i), \frac{\alpha_{s_i}}{\sum_{l \in L(s_i)} \lambda_l} \right\}. \tag{3.37}$$

The result in (3.37) ensures the feasibility of the solution for rate variables r_{s_i} by not allowing it to take values below $R_{\min}(s_i)$. Since the dual variables λ_l $\forall l$ are updated at each iteration of the algorithm and due to strong duality, optimal values of the dual variables are obtained when the algorithm is terminated after meeting the convergence criterion set by the error tolerance as discussed in Sect. 3.3.5. For the optimal values of the dual variables, the unique solution for $r^*_{s_i}$ is given by

$$r^*_{s_i} = \max \left\{ R_{\min}(s_i), \frac{\alpha_{s_i}}{\sum_{l \in L(s_i)} \lambda^*_l} \right\}. \tag{3.38}$$

3.3.4 Link Delay Subproblem

The link delay minimization subproblem from (3.16) is given by

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{d}^{(t)}, \mathbf{d}^{(q)}) = \sum_{l} \left(\frac{\lambda_l \mu_{s_i}}{2d_l^{(q)}} + \frac{\xi_l \mu_{s_i}}{2d_l^{(t)}} + \sum_{s_i:\, l \in L(s_i)} \psi_{s_i} \frac{d_l^{(t)} + d_l^{(q)}}{D_{\max}(s_i)} \right) \\ \textbf{s.t.} \quad & 0 \le d_l^{(t)}, d_l^{(q)} \quad \forall l. \end{aligned} \tag{3.39}$$

The optimization subproblem in (3.39) is convex if the Hessian of the objective function is positive semidefinite. Since the objective function is not coupled in the link delay variables, the convexity for $f(\mathbf{d}^{(t)}, \mathbf{d}^{(q)})$ can also be verified by using the inequality

$$-\frac{1}{d_l^{(i)}} \frac{\partial f(\mathbf{d}^{(t)}, \mathbf{d}^{(q)})}{\partial d_l^{(i)}} \le \frac{\partial^2 f(\mathbf{d}^{(t)}, \mathbf{d}^{(q)})}{\left(\partial d_l^{(i)}\right)^2},$$

where $i \in \{q, t\}, \ \forall i$. The delay subproblem is then solved by gradient projection method [24].

3.3.5 Dual Problem

The dual problem in (3.17) associated with the original problem can be solved using a subgradient method [25]. The iterative updates for λ_l and ξ_l link dual variables are given by

$$\lambda_l(k+1) = \left[\lambda_l(k) - \beta_\lambda(k) \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \sum_{s_i: l \in L(s_i)} r_{s_i} - \frac{\mu_{s_i}}{2d_l^{(q)}} \right) \right]^+, \tag{3.40}$$

$$\xi_l(k+1) = \left[\xi_l(k) - \beta_\xi(k) \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P})) - \frac{\mu_{s_i}}{2d_l^{(t)}} \right) \right]^+, \tag{3.41}$$

and the subgradient updates for the ψ_{s_i} dual variables, associated with robustness-parameter, are obtained as

$$\psi_{s_i}(k+1) = \left[\psi_{s_i}(k) - \beta_\psi(k) \left((1 - \varepsilon_{s_i}) - \frac{1}{D_{\max}(s_i)} \sum_{l \in L(s_i)} \left(d_l^{(t)} + d_l^{(q)} \right) \right) \right]^+. \tag{3.42}$$

The variables $\beta_\lambda(k)$, $\beta_\xi(k)$, and $\beta_\psi(k)$ in (3.40)–(3.42) are the respective step sizes for the dual updates. How to make an appropriate choice of these step sizes is discussed below.

For a convex function f, a vector φ is said to be its subgradient at $\mathbf{x} \in domf$ if $f(\mathbf{x}') \geq f(\mathbf{x}) + \varphi^T(\mathbf{x}' - \mathbf{x}) \ \ \forall \mathbf{x}'$. Now, the minimization problem is defined over the closed and convex set $\mathcal{C}$ as

$$\min_{\mathbf{x} \in C} f(x). \tag{3.43}$$

The subgradient method to solve (3.43) uses the iteration $\mathbf{x}(k+1) = P\left[\mathbf{x}(k) - \beta(k)\ \varphi^{(k)}\right]$, where $\mathbf{x}(k)$ is the k^{th} iteration, $\varphi^{(k)}$ is the subgradient of f at $\mathbf{x}(k)$, $\beta(k) > 0$ is the k^{th} step size, and P is the projection on $\mathcal{C}$. Assuming $\mathbf{x}^*$ is a minimizer of (3.43) and there exists G such that $\| \varphi^{(k)} \|_2 \leq G, \forall k$, [26, 27], then

$$\min_{i=1,2,\cdots,k} f(x^{(i)}) - f^* \leq \frac{\| \mathbf{x}^{(1)} - \mathbf{x}^* \|_2^2 + G^2 \sum_{i=1}^{k} \beta^2(i)}{2 \sum_{i=1}^{k} \beta(i)}. \tag{3.44}$$

Using the expression in (3.44), the following convergence results can be obtained readily. For constant step size rule i.e., $\beta(k) = \beta$, (3.44) converges to $\min_{i=1,2,\cdots,k} f(x^{(i)}) - f^* \leq \frac{G^2\beta}{2}$ as $k \to \infty$. For the case of diminishing step size rule, i.e., $\lim_{k\to\infty} \beta(k) = 0$, $\sum_{k=1}^{\infty} \beta(k) = \infty$, the expression on the right hand side of (3.44) converges to zero. An error tolerance of ϵ, such that $\frac{\|\mathbf{x}^{(1)} - \mathbf{x}^*\|_2^2 + G^2 \sum_{i=1}^{k} \beta^2(i)}{2 \sum_{i=1}^{k} \beta(i)} \leq \epsilon$ is used for the termination of the subgradient algorithm. For a given value of ϵ, the subgradient algorithm will require $k = (GR/\epsilon)^2$ steps in case of diminishing step size rule and the corresponding step size is determined by $\beta(k) = \frac{R}{G\sqrt{k}}$, where R is a number that satisfies $R \geq \| \mathbf{x}^{(1)} - \mathbf{x}^* \|_2$. From the expression for $\beta(k)$, the parameter R is required for its evaluation. But there is no simple method for finding a value of R, which is justifiable. There are two possibilities, one assuming an arbitrary upper bound from the partial knowledge about the optimal set $\mathbf{x}^*$, while in the second case an optimum should be found first and will require the global information.

3.4 Iterative Cross-Layer Algorithm

In the previous section, we proposed a distributed resource optimization algorithm, leading to an optimal tradeoff between the network-throughput and corresponding delay robustness. The convergence of DROA for a feasible set of fixed routes $L(s_i)\ \forall s_i$ and transmission schedules $h_i \in H$ can result in congestion on some of the highly loaded links. This can be avoided by diverting information flow away from the congested links, which can be achieved by incorporating the queuing delay into the link weights as

$$w_l = \begin{cases} \frac{1}{G_{ll} P_l} + ad_l^{(q)} & 1/\left(G_{ll} P_l + ad_l^{(q)}\right) \leq \theta_l \\ \infty & \text{otherwise} \end{cases} \quad \forall l, \tag{3.45}$$

Algorithm 1 Iterative Cross-Layer Algorithm (ICLA)

Require: Choose feasible r_{s_i}, P_l, h_i and $L(s_i)$. Set $flag = 1$
  compute $r^*_{s_i}$, $d^{*(t)}_l$, $d^{*(q)}_l$, P^*_l, J^* and set $J^*_{new} = J^*$
  while $flag \neq 0$ **do**
    if $J^* \leq J^*_{new}$ **or** $|(J^*_{new} - J^*)/J^*| \geq \chi$ **then**
      update w_l, $L(s_i)$, h_i and set $J^* = J^*_{new}$
      if $|H|_{new} \leq |H|$ **then**
        recompute $r^*_{s_i}$, $d^{*(t)}_l$, $d^{*(q)}_l$, P^*_l and J^*_{new}
      else
        $flag = 0$
      end if
    else
      $flag = 0$
    end if
  end while

where a is the normalizing constant. Once DROA is converged, we update the link weights, w_l $\forall l$, using (3.45) based on the optimal link power and queuing delay components. Using updated link weights, we reevaluate the shortest routes $L(s_i)$ $\forall s_i$ with the possibility of route switching. The route switching is attributed to higher link congestion due to the incorporation of $d^{(q)}_l$ in the expression for w_l.

Finding an optimal transmission schedule for the updated routes is NP hard [28]. Using an interference aware link scheduling approach [18] and updated routes, we provide a suboptimal ICLA outlined in Algorithm 1. The proposed ICLA achieves convergence in a small number of iterations (less than 10 iterations for a network of 50 nodes). The algorithm alternates between DROA and route and schedule updates until further improvement in the performance cannot be achieved. Below, we describe the key steps in ICLA to achieve convergence:

1. For a given initial set of routes and transmission schedules, when DROA has converged, we update w_l for all existing links according to (3.45) using P^*_l, $d^{*(t)}_l$ and $d^{*(q)}_l$. An existing link l is removed from the network topology if $\frac{1}{G_{ll}P^*_l} + ad^{*(q)}_l > \theta_l$. This can happen if the transmitting node of that link is highly congested or the channel is in deep fade.
2. Using updated w_l $\forall l$, the shortest routes $L(s_i)$ $\forall s_i$ and the corresponding transmission schedules $h_i \in H$ $\forall i$ are computed again. To ensure convergence and avoid unnecessary iterations of ICLA, we require $J^* \leq J^*_{new}$ and $|H|_{new} \leq |H|$, where $|H|_{new}$ is updated transmission cycle consisting of recomputed transmission schedules and J^*_{new} is the recomputed optimal objective value for DROA.
3. The ICLA is terminated if one of the following conditions is met:
   - Objective condition $J^* \leq J^*_{new}$ for DROA is violated.
   - The transmission cycle constraint $|H|_{new} \leq |H|$ is violated.
   - For a predefined tolerance χ, the convergence of ICLA is achieved.

In the performance evaluation of ICLA, we will study the tradeoff between network throughput and the resulting link queuing delay fairness. For that purpose, we define max-min link queuing delay fairness as

$$\text{Link Queuing Delay Fairness} = \frac{\min\{d_l^{(q)} \mid d_l^{(q)} \in \mathbf{d}^{(q)}\}}{\max\{d_l^{(q)} \mid d_l^{(q)} \in \mathbf{d}^{(q)}\}}, \tag{3.46}$$

3.5 Results

For the performance analysis of the distributed algorithm, we use the following network parameters. A constant packet size of 50 bytes and $P_{\max}$ of 10 dBm are used. For simultaneous transmissions, distance threshold ν=2 and the channel-gain $G_{ij} = \Delta_{ij}^{-3}$ are used. Since there can be a large number of possible combinations for α_{s_i}, $D_{\max}(s_i)$, $R_{\min}(s_i)$ and resulting ε_{s_i} for different s_i, we use $\alpha_{s_i} = \alpha_s$, $D_{\max}(s_i) = D_{\max}(s)$, $R_{\min}(s_i) = R_{\min}(s)$, and $\varepsilon_{s_i} = \varepsilon_s$ without any loss of generality.

3.5.1 Throughput-Robustness Tradeoff

To study the robustness-throughput tradeoff characteristics of DROA for different values of α_{s_i}, we use the example network shown in Fig. 3.5. We keep the routes $L(s_i)$ $\forall s_i$ and the corresponding transmission schedules fixed in studying this

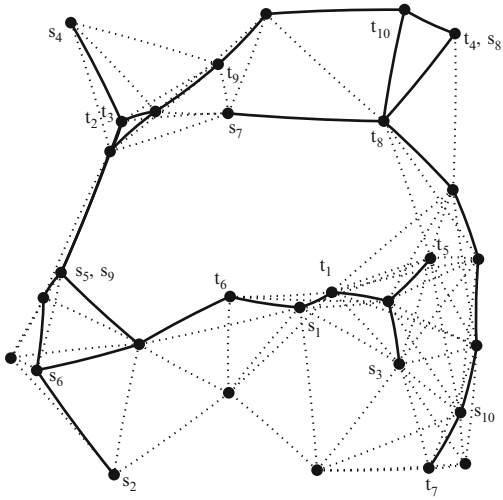


Fig. 3.5 An example network used in performance evaluation with solid lines representing the selected links participating in scheduled transmission, while dotted lines represent the links satisfying (3.2). The nodes marked by s_i and t_i represent the starting and ending nodes for transmission session i

tradeoff. For performance analysis, an optimal network-throughput (Ω_{η_l}), using the link-efficiency function (η_l) based transmission rate, is defined as follows

$$\Omega_{\eta_l} = \left\{ \sum_s r_s^* \;\middle|\; \frac{\mu_s/2}{\overline{R}_l \eta_l(\mathbf{P}) - \sum_{s:\, l \in L(s)} r_s} \leq d_l^{*(q)},\ \frac{\mu_s/2}{\overline{R}_l \eta_l(\mathbf{P})} \leq d_l^{*(t)} \right\}. \quad (3.47)$$

The optimal network throughput performance, as a function of end-to-end delay threshold $D_{\max}(s)$, is shown in Fig. 3.6 for different values of α_s. Increasing α_s when it is small (for instance $\alpha_s < 0.3$) gives a considerable throughput performance gain while providing moderate gain for $\alpha_s > 0.6$. The corresponding delay robustness performance for different values of parameter α_s is shown in Fig. 3.7. The results in Figs. 3.6 and 3.7 show that for $\alpha_s > 0.5$ a small compromise in the optimal network throughput Ω_{η_l} can provide a significant delay robustness in the form of delay margin[1]. For example, by changing α_s from 0.9 to 0.7 at $D_{\max}(s) = 20$ ms, the optimal network-throughput decreases by only 3% but provides an increase of 30% in the delay margin or equivalently a decrease of 44% in the optimal end-to-end delay d_s^*.

The variation in parameter ε_s provides an insight into the optimal robustness-throughput tradeoff. Figure 3.8 plots ε_s as function of $D_{\max}(s)$ and parameter α_s. Reducing α_s is equivalent to compromising the throughput for an improvement in

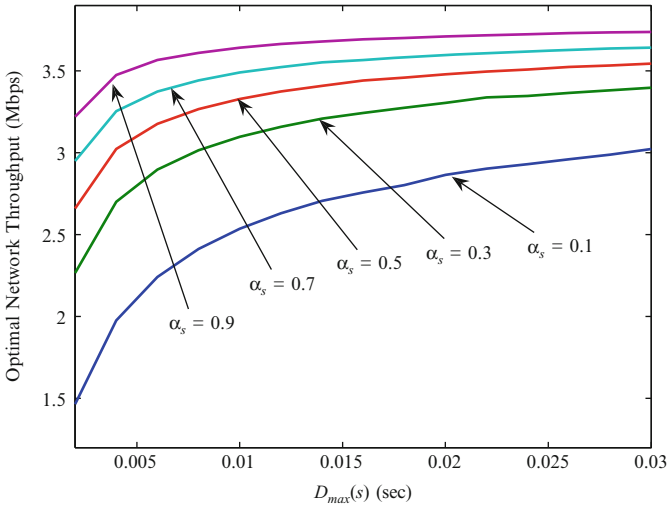


Fig. 3.6 Optimal network-throughput $\Omega_{\eta l}$ for different α_s values as a function of maximum end-to-end delay threshold ($D_{\max}(s)$). $R_{\min}(s)$ of 100 Kbps is used

[1] We have defined the delay margin equivalent to $D_{\max}(s) - d_s^*$, where d_s^* is the optimal end-to-end delay given by $d_s^* = \sum_{l \in L(s_i)} \left(d_l^{*(q)} + d_l^{*(t)} \right)$.

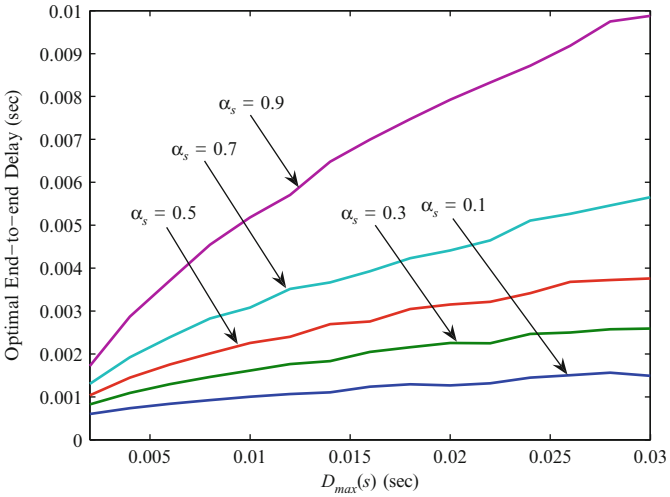


Fig. 3.7 Optimal end-to-end delay performance corresponding to an arbitrarily chosen transmission session for different values of α_s as a function of maximum end-to-end delay threshold ($D_{\max}(s)$). $R_{\min}(s)$ of 100 Kbps is used

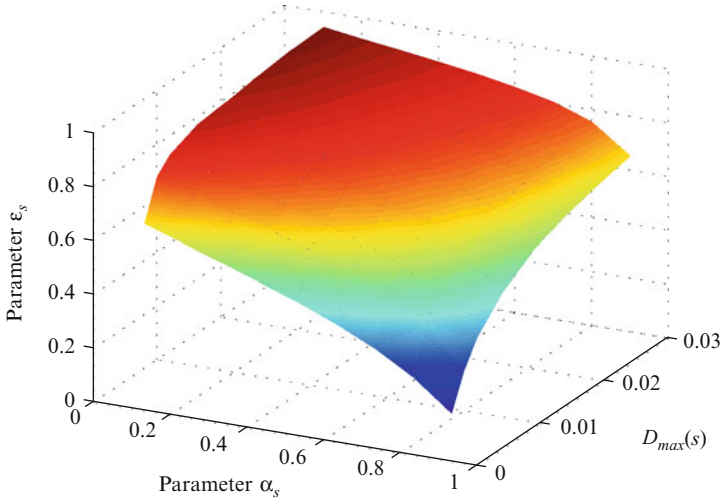


Fig. 3.8 The robustness parameter variation as a function of end-to-end delay threshold and optimization objective weighting parameter α_s

the delay margin and is achieved by an increase in ε_s for a given delay threshold $D_{\max}(s)$ as observed in Fig. 3.8. In other words, for fixed $D_{\max}(s)$ an increase in ε_s forces $d_l^{*(q)}$ and/or $d_l^{*(t)}$ to decrease to satisfy (3.11) and hence improving the delay margin. On the other hand, an improvement in delay margin with an increase

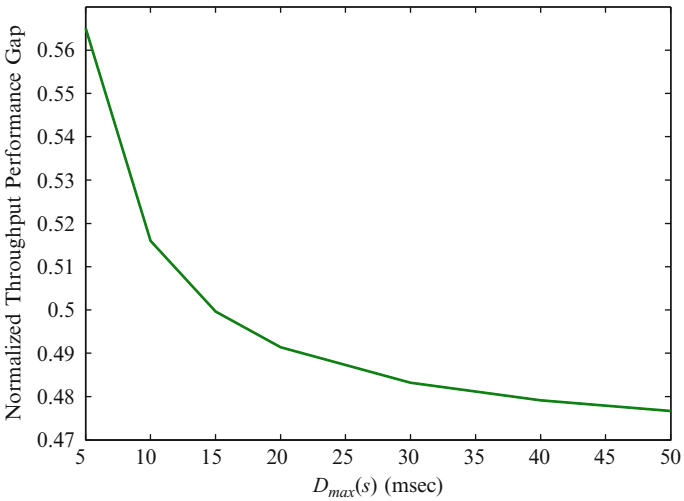


Fig. 3.9 Optimal network-throughput performance comparison of the link-efficiency function (3.5) based effective transmission rate model with the link capacity based effective transmission rate model of [10, 11] using the example network of Fig. 3.5

in $D_{\max}(s)$ will depend on the choice of α_s. A lower value of parameter α_s will result in higher delay margin with an increase in $D_{\max}(s)$ and vice versa.

We next compare the throughput Ω_{η_l} (computed for effective link-transmission-rate) with the link capacity based throughput measure Ω_{c_l} employing the transmission model of [10, 11]. For that purpose, Ω_{c_l} is computed as follows

$$\Omega_{c_l} = \left\{ \sum_s r_s^* \;\middle|\; \frac{\mu_s/2}{c_l(\mathbf{P}) - \sum_{s:\, l \in L(s)} r_s} \leq d_l^{*(q)},\ \frac{\mu_s/2}{c_l(\mathbf{P})} \leq d_l^{*(t)} \right\}. \quad (3.48)$$

Using the throughput definitions in (3.47) and (3.48), the normalized throughput performance gap, obtained as $\Omega_{c_l} - \Omega_{\eta_l}/\Omega_{c_l}$, is shown in Fig. 3.9. The superior throughput performance in case of link capacity is attributed to the flexibility in the choice of any capacity-achieving channel-coding techniques and may require higher $D_{\max}(s)$ to be viable. On the other hand, the effective-transmission-rate based model, not limited by capacity-achieving channel codes, will perform better in situations with stringent requirements on $D_{\max}(s)$.

3.5.2 Performance Evaluation of ICLA

Next, we study the convergence performance of ICLA by updating the routes and the transmission schedule once the convergence of DROA is achieved. The network-throughput Ω_{η_l} as a function of the number of ICLA iterations is shown in Fig. 3.10

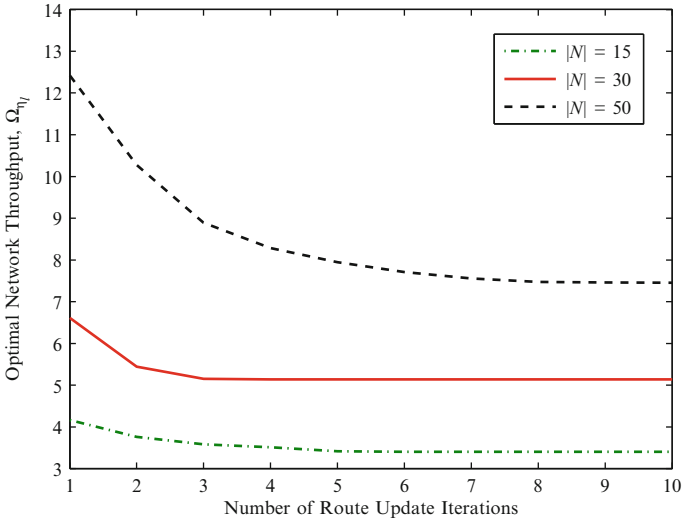


Fig. 3.10 Optimal network throughput as a function of route update iterations. The parameters $\alpha_s = 0.8$, $D_{\max}(s) = 20$ ms, and $R_{\min}(s)$ of 100 Kbps are used for this result.

for different number of nodes. It can be seen from Fig. 3.10 that the iterative algorithm converges in less than ten iterations for an arbitrary network of 50 nodes. The reduction in the network-throughput during convergence is mainly due to the fact that the link queuing delay at the start of the algorithm is initialized as $d_l^{(q)} = 0$. The reduction in the ICLA throughput results in an improved link queuing delay fairness and is shown in Fig. A.1 as a result of convergence of ICLA. Figure A.1 also shows the percentage reduction in the network-throughput Ω_{η_l}. The result in Fig. A.1 shows the tradeoff between throughput optimality and the resulting link congestion fairness and can be used in tuning the network parameters for the desired performance.

3.6 Summary and Conclusions

A distributed resource optimization framework for delay robustness and network-throughput tradeoff using sensitivity analysis is proposed. Network-throughput maximization is achieved by solving an effective link-transmission-rate based power control problem. The proposed resource optimization algorithm is extended to an iterative cross-layer algorithm by solving the resource allocation, routing and scheduling problems iteratively. Our results show that a small compromise in the optimal network-throughput can provide large delay robustness. Higher degradation of network-throughput, while improving link queuing delay fairness, suggests that an arbitrary scaling of the network is not possible when delay fairness is of interest. A possible future research is to explore how the clustering can be used as a possible solution for network scaling.

Appendix A

Convergence of DROA

The first step in the proof of convergence is to show the convexity of the problem. This is verified for each of the subproblems either by evaluating the Hessian or by validating (in case of single variable functions) the inequality $-\frac{1}{x}\frac{\partial f(x)}{\partial x} \leq \frac{\partial^2 f(x)}{(\partial x)^2}$ for a given function f. Once the convexity of the problem in (3.10)–(3.14) is verified, the dual function can be defined as

$$g(\mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi}) = \max_{\mathbf{r},\mathbf{P},\mathbf{d}^{(q)},\mathbf{d}^{(t)},\mathbf{\Upsilon}} L(\mathbf{r}, \mathbf{P}, \mathbf{d}^{(q)}, \mathbf{d}^{(t)}, \mathbf{\Upsilon}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi}) \tag{A.1}$$

The maximization in (A.1), as discussed in Sect. 3.3, is achieved by solving the robustness, power, rate, and delay subproblems. The solution to those subproblems allows to evaluate $g(\mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi})$. By strong duality [20], the overall resource optimization problem is solved by minimizing the dual as in (3.17). The key step in dual minimization is to show that the updates in (3.40)–(3.42) indeed solve the dual minimization problem. We next show that the update in (3.40) is indeed a subgradient update for dual variables $\lambda_l \ \forall l$. For given $\lambda_l \ \forall l$, let $\mathbf{r}^*$, $\mathbf{P}^*$, $\mathbf{d}^{*(q)}$, $\mathbf{d}^{*(t)}$, and $\mathbf{\Upsilon}^*$ are optimal solutions for the respective variables, then

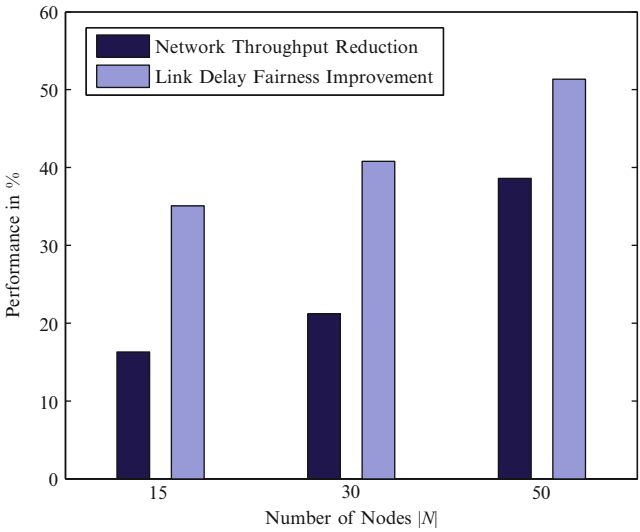


Fig. A.1 Average percentage performance improvement in link delay fairness leading to congestion avoidance. The price to avoid the congestion in terms of reduction in network-throughput $\Omega_{\eta l}$ is also shown

$$g(\mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi}) = \Bigg\{ \sum_{s_i} \left(\alpha_{s_i} U(r^*_{s_i}) + (1-\alpha_{s_i})\phi(\varepsilon^*_{s_i})\right) + \sum_l \xi_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \frac{\mu_{s_i}}{2d_l^{*(t)}} \right) + \sum_{s_i} \psi_{s_i} \left((1-\varepsilon^*_{s_i}) - \frac{1}{D_{\max}(s_i)} \sum_{l \in L(s_i)} \left(d_l^{*(t)} + d_l^{*(q)} \right) \right) + \sum_l \lambda_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \sum_{s_i:\, l \in L(s_i)} r^*_{s_i} - \frac{\mu_{s_i}}{2d_l^{*(q)}} \right) \Bigg\}, \tag{A.2}$$

and for some arbitrary $\lambda'_l \in \mathbf{\Lambda}'\ \forall l$, we have

$$g(\Lambda', \mathbf{\Xi}, \mathbf{\Psi}) \geq \Bigg\{ \sum_{s_i} \left(\alpha_{s_i} U(r^*_{s_i}) + (1-\alpha_{s_i})\phi(\varepsilon^*_{s_i})\right) + \sum_l \xi_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \frac{\mu_{s_i}}{2d_l^{*(t)}} \right) + \sum_{s_i} \psi_{s_i} \left((1-\varepsilon^*_{s_i}) - \frac{1}{D_{\max}(s_i)} \sum_{l \in L(s_i)} \left(d_l^{*(t)} + d_l^{*(q)} \right) \right) + \sum_l \lambda'_l \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \sum_{s_i:\, l \in L(s_i)} r^*_{s_i} - \frac{\mu_{s_i}}{2d_l^{*(q)}} \right) \Bigg\}. \tag{A.3}$$

Subtracting (A.3) from (A.2), we have

$$g(\mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi}) - g(\Lambda', \mathbf{\Xi}, \mathbf{\Psi}) \leq \sum_l (\lambda_l - \lambda'_l) \left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \sum_{s_i:\, l \in L(s_i)} r^*_{s_i} - \frac{\mu_{s_i}}{2d_l^{*(q)}} \right). \tag{A.4}$$

Using the definition of subgradient [25], we can verify that $\left(\overline{R}_l \eta_l(\gamma_l(\mathbf{P}^*)) - \sum_{s_i:\, l \in L(s_i)} r^*_{s_i} - \frac{\mu_{s_i}}{2d_l^{*(q)}}\right)$ is the subgradient of $g(\mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Psi})$. A similar procedure can be used for verifying the subgradients for ξ_l and ψ_{s_i} in (3.41) and (3.42), respectively. An appropriate choice of the step sizes $\beta_\lambda(k), \beta_\xi(k)$, and $\beta_\psi(k)$, as discussed in Sect. 3.3.5 [25], will result in the convergence of subgradient updates within a predefined error tolerance ϵ of the optimal value as explained below. Once the optimal dual variables are found, the corresponding primal variables can be obtained by solving their respective subproblems. And due to strong duality, primal variables must be global optimum providing unique solution.

References

1. W. Zhang, M. Branicky, and S. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.
2. X. Liu and A. Goldsmith, "Wireless communication tradeoffs in distributed control," *IEEE Conference on Decision and Control*, 2003, pp. 688–694.
3. S. Mazumder, M. Tahir, and K. Acharya, "Master-slave current-sharing control of a parallel DC-DC converter system over an RF communication interface," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 59–66, 2008.
4. I. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
5. A. Bemporad, S. Di Cairano, E. Henriksson, and K. Johansson, "Hybrid model predictive control based on wireless sensor feedback: An experimental study," *IEEE Conference on Decision and Control*, 2007, pp. 5062–5067.
6. J. Lavaei, A. Momeni, and A. Aghdam, "A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 268–278, 2008.
7. F. Lian, J. Moyne, D. Tilbury *et al.*, "Network design consideration for distributed control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 2, pp. 297–307, 2002.
8. X. Liu and A. Goldsmith, "Wireless network design for distributed control," *IEEE American Control Conference*, 2004, pp. 2823–2829.
9. S. Mazumder, K. Acharya, and M. Tahir, "Joint optimization of control performance and network resource utilization in homogeneous power networks," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1736–1745, 2009.
10. M. Saad and A. Yu, "Optimal network rate allocation under end-to-end quality-of-service requirements," *IEEE Transactions on Network and Service Management*, vol. 4, no. 3, pp. 40–49, 2007.
11. M. Tahir and S. Mazumder, "Delay constrained optimal resource utilization of wireless networks for distributed control systems," *IEEE Communications Letters*, vol. 12, no. 4, pp. 289–291, 2008.
12. C. Tan, D. Palomar, and M. Chiang, "Energy-robustness tradeoff in cellular network power control," *To appear IEEE/ACM Transactions on Networking*, 2009.
13. F. Meshkati, H. Poor, S. Schwartz, and N. Mandayam, "An energy-efficient approach to power control and receiver design in wireless data networks," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1885–1894, 2005.
14. M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, 2005.
15. C. Comaniciu and H. Poor, "On energy efficient hierarchical cross-layer design: joint power control and routing for ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. Article ID 60707, p. 9, 2007.
16. L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, 2004.
17. D. Bertsekas and R. Gallager, *Data Networks.* Prentice-Hall, NJ, USA, 1991.
18. W. Wang, Y. Wang, X. Li, W. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proc. of International Conference on Mobile Computing and Networking*, 2006, pp. 262–273.
19. T. Ren and R. La, "Power control algorithm for providing packet error rate guarantees in ad hoc networks," *IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 6040–6045.
20. S. Boyd and L. Vandenberghe, *Convex Optimization*. DCambridge University Press, 2004.
21. A. Orda, "Routing with end-to-end QoS guarantees in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 365–374, 1999.

22. K. Jalaleddini, K. Moezzi, A. Aghdam, M. Alasti, and V. Tarokh, "Controller Design for Rate Assignment in Wireless Networks," *IEEE International Conference on Communications*, 2009, pp. 1–6.
23. D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.
24. R. Haftka and Z. Gürdal, *Elements of Structural Optimization*. Kluwer, 1992.
25. N. Shor, K. Kiwiel, and A. Ruszcaynski, *Minimization Methods for Non-differentiable Functions*. Springer, New York, 1985.
26. D. Bertsekas, *Nonlinear programming*. Athena Scientific, Belmont, MA, 1999.
27. S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.
28. J. Yeo, H. Lee, and S. Kim, "An efficient broadcast scheduling algorithm for TDMA ad hoc networks," *Computers and Operations Research*, vol. 29, no. 13, pp. 1793–1806, 2002.

Chapter 4
Cyber-Physical Control Over Wireless Sensor and Actuator Networks with Packet Loss

Feng Xia, Xiangjie Kong, and Zhenzhen Xu

Abstract There is a growing interest in design and implementation of cyber-physical control systems over wireless sensor and actuator networks (WSANs). Thanks to the use of wireless communications and distributed architectures, these systems encompass many advantages as compared to traditional networked control systems using hard wirelines. While WSANs are enabling a new generation of control systems, they also introduce considerable challenges to quality-of-service (QoS) provisioning. In this chapter, we examine some of the major QoS challenges raised by WSANs, including resource constraints, platform heterogeneity, dynamic network topology, and mixed traffic. These challenges make it difficult to fulfill the requirements of cyber-physical control in terms of reliability and real time. The focus of this chapter is on addressing the problem of network reliability. Specifically, we analyze the behavior of wireless channels via simulations based on a realistic link-layer model. Packet loss rate (PLR) is taken as a major metric for the analysis. The results confirm the unreliability of wireless communications and the uncertainty of packet loss over WSANs. To tackle packet loss, we present a simple solution that can take advantage of existing prediction algorithms. Simulations are conducted to evaluate the performance of several classical prediction algorithms used for loss compensation. The results give some insights into how to deal with packet loss in cyber-physical control systems over unreliable WSANs.

Keywords Wireless sensor and actuator networks · Cyber-physical systems · Wireless control · Quality of service · Packet loss

4.1 Introduction

A wireless sensor and actuator network (WSAN) [1–4] is a networked system of geographically distributed sensor and actuator nodes, as shown in Fig. 4.1. These nodes are interconnected via wireless links. The scale of the network depends highly

F. Xia (✉)
School of Software, Dalian University of Technology, Dalian 116620, China
e-mail: f.xia@ieee.org.

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_4,

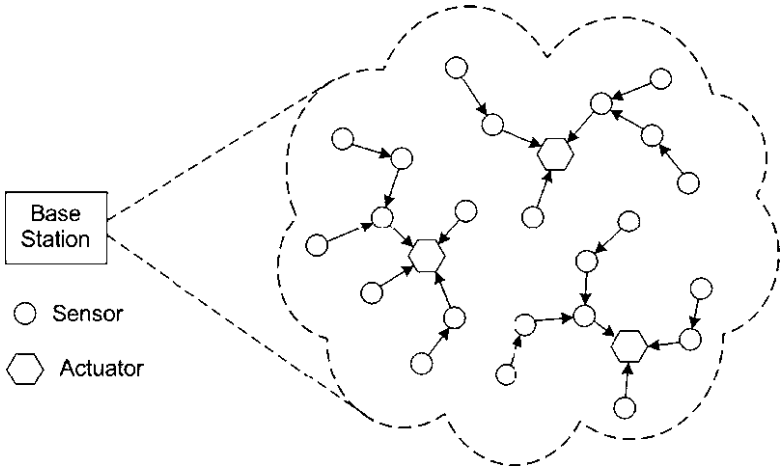


Fig. 4.1 A wireless sensor and actuator network

on the target application. In general, both sensor and actuator nodes are equipped with some data processing and wireless communication capabilities, as well as power supply. Over the past years, a number of prototype and commercial wireless sensor nodes have been made available by research institutions and companies from around the world. Typical examples include BTnode, FireFly, IMote2, MicaZ, SunSPOT, TinyNode584, Tmote Sky, etc. Sensors gather information about the state of physical world and transmit the collected data to actuators through single-hop or multi-hop communications over the radio channel. Upon receipt of the required information, the actuators make the decision about how to react to this information and perform corresponding actions to change the behavior of the physical environment. As such, a closed loop is formed integrating the cyber and physical worlds. In addition to sensor and actuator nodes, there is commonly a base station in the WSAN, which is principally responsible for monitoring and managing the overall network through communicating with sensors and actuators.

General wireless sensor networks (WSNs) are used for information gathering in applications such as habitat monitoring, military surveillance, agriculture and environmental sensing, and health monitoring. The primary functionality of such WSNs is to sense and monitor the state of the physical world. In most cases, they are unable to *affect* the physical environment. In many applications, however, it is not sufficient to just observe the state of the physical system; but it is also necessary to respond to the sensed events/data by performing corresponding actions upon the physical system. For instance, in a fire handling system, it is necessary for the actuators to turn on the water sprinklers upon receipt of a report of fire. WSANs can satisfy such requirements by enabling the application systems to sense, interact, and change the physical world, e.g., to monitor and manipulate the lighting in a smart office or the speed of a mobile robot.

By closing the loop involving both the cyber and the physical worlds, WSANs will become one of the most critical technologies for building future cyber-physical

systems (CPSs) [5, 6], which promise to revolutionize the way we interact with the physical world. Such systems can be deployed in many applications such as health care, home automation, assisted living, intelligent building, intelligent transportation, disaster relief, planet exploration, and industrial control. In particular, WSANs exploit the methodology of feedback, which has been recognized as the central element of control systems. The advent of WSANs has the potential to revolutionarily promote existing control applications by enabling an unprecedented degree of distributed cyber-physical control.

Today's control systems are usually built upon hard wirelines. In contrast, control over WSANs exploits the potential of wireless communications, which deliver many advantages [7, 8]. For instance, various difficulties related to the installation and maintenance of the large number of cables are completely eliminated. Consequently, the flexibility and expandability of the system can be further enhanced. At the same time, system maintenance and update become easier, and the cost will of course be reduced. In some harsh environments, it is forbidden or unfavorable to use cables due to constraints concerning, e.g., physical environments and production conditions. This is especially the case when deleterious chemicals, severe vibrations, and high temperatures are present that could potentially damage any sort of cabling. For such situations wireless technologies offer a much better choice for achieving connectivity. In addition, wireless control satisfies the requirements of mobile systems, enabling closed-loop control of mobile objectives, such as automated guided vehicles, mobile robots, and unmanned aerial vehicles.

However, the use of wireless networks in connecting spatially distributed sensors, controllers, and actuators raises new challenges for control systems design [3, 7]. Wireless channels have adverse properties, such as path loss, multi-path fading, adjacent channel interference, Doppler shifts, and half-duplex operations. Consequently, transmitting radio signals over wireless channels can be affected by many factors, such as ambient noise, physical obstacles, node movement, environmental changes, and transmission power, to mention just a few. The inherent openness of wireless connections may potentially cause the operating environment of the system to be highly dynamic and unpredictable, since the wireless channel might be used by other coexisting devices. Wireless communications are much less dependable than wirelines in that the bit error rate of a wireless channel is usually several times that of a wired connection [9]. This is especially true for WSANs that feature low-power communications. As a consequence, constructing cyber-physical control systems over WSANs is challenging because the network quality-of-service (QoS) cannot always be guaranteed. Particularly, the control performance might be sacrificed due to unpredictable packet loss, which could even cause system instability in extreme cases.

This chapter aims to develop a better understanding of how to realize cyber-physical control over unreliable WSANs. For this purpose, major challenges with respect to QoS provisioning will be outlined. Special attention is given to the packet loss problem arising in the context of WSAN. Using a realistic link-layer model, we analyze the characteristics of the wireless channel in terms of packet loss rate (PLR). The results confirm the unreliability and uncertainty of wireless communications

over WSANs. To cope with packet loss, we present a simple solution that can take advantage of existing prediction algorithms, such as time series forecasting. Simulations are conducted to evaluate the performance of several classical prediction algorithms used for packet loss compensation.

The rest of the chapter is organized as follows. Section 4.2 summarizes some state-of-the-art work related to this chapter. In Sect. 4.3, we describe the architecture of cyber-physical control systems exploiting WSANs. Major QoS challenges are discussed in Sect. 4.4. The behavior of wireless channel is analyzed via simulations in Sect. 4.5, where we focus on PLR with respect to communication distance and transmission power. A simple approach to packet loss compensation is presented and evaluated in Sects. 4.6 and 4.7, respectively. Section 4.8 concludes the chapter.

4.2 Related Work

CPSs [5, 10] are integrations of computation, networking, and physical dynamics, in which embedded devices such as sensors and actuators are (wirelessly) networked to sense, monitor, and control the physical world. It is believed in both the academic and the industrial communities that CPS will have great technical, economic, and societal impact in the future. The CPS of tomorrow will far exceed those of today in terms of both performance and efficiency. The realm of CPS is opening up unprecedented opportunities for research and development in numerous disciplines, e.g., computing, communications, and control. In recent years, CPS has been attracting attention from a rapidly increasing number of researchers and engineers. To fully exploit the potential of CPS, however, many challenges must be overcome.

WSANs play an essential role in cyber-physical control systems, since they are the *bridge* between the cyber and physical worlds. Akyildiz and Kasimoglu [11] described research challenges for coordination and communication problems in WSANs. Melodia et al. [12] further studied these problems in WSANs with mobile actuators. Rezgui and Eltoweissy [13] discussed the opportunities and challenges for service-oriented sensor and actuator networks. Sikka et al. [14] deployed a large heterogeneous WSAN on a working farm to explore sensor network applications that can help manage large-scale farming systems. In comparison with the field of general WSN in which significant progress has been made over the years, WSAN is a relatively new research area yet to be explored.

In particular, there is only limited work in the WSAN area targeting cyber-physical control applications. For example, Li [15] prototyped a light monitoring and control system as a case study of WSANs. Oh et al. [16] illustrated the main challenges in developing real-time control systems for pursuit-evasion games using a large-scale sensor network. Bosch et al. [17] reported the application of WSANs in distributed movement control and coordination of autonomous vehicles. Korber et al. [18] dealt with some of the design issues of a highly modular and scalable implementation of a WSAN for factory automation applications. Nikolakopoulos et al. [19] developed a gain scheduler for wirelessly networked control systems to

cope with time-varying delay induced by multi-hop communications in sensor networks. Despite growing interest, the impact of packet loss, as a result of unreliable communications in WSANs, on the performance of cyber-physical control remains open for further research.

In the control community, significant effort has been made for packet loss compensation. A survey on this topic can be found in [20]. Despite their differences, most of the existing packet loss compensation methods share the following features. First, they depend heavily on the knowledge about the accurate models of the physical systems to be controlled, and, possibly, the controller design. Second, the relevant algorithms are computationally intensive. For these reasons, they are impractical for real systems lacking well-established mathematical models. In addition, they are usually not desirable solutions for resource-constrained WSANs because of overly large computational overheads.

This chapter is closely related to our previous work [2–4, 6]. In [3], Xia et al. proposed an application-level design methodology for WSANs in mobile control applications. The unreliability of wireless links within sensor networks was also studied experimentally. QoS challenges and opportunities for WSANs were discussed in [2, 6]. A QoS management scheme using fuzzy logic control technique and feedback scheduling concept was presented in [4, 6].

4.3 System Architecture

Feedback cyber-physical control deals with the regulation of the characteristics of a CPS. The main idea of feedback control is to exploit measurements of the system's outputs to determine the control commands that yield the desired system behavior. As shown in Fig. 4.2, a controller, together with some sensors and actuators, is usually used to sense the operation of the physical system, compare it against the desired behavior, compute control commands, and perform actions onto the system

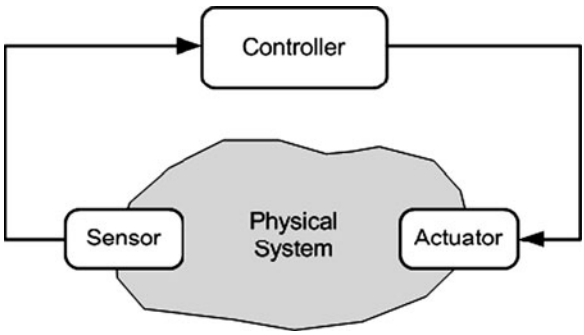


Fig. 4.2 A cyber-physical control system

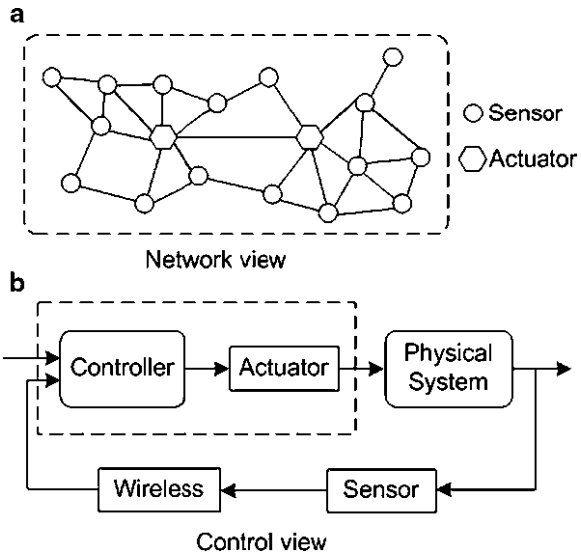


Fig. 4.3 WSAN architecture without explicit controllers: (**a**) network view; (**b**) control view

to effect the desired change. This feedback architecture of a cyber-physical control system is also called *closed loop*, implying that the cyber space and the physical system are able to affect each other.

As mentioned previously, there are three essential components in a WSAN: sensors, actuators, and a base station. Depending on whether there are explicit controller entities within the network, two types of system architectures of WSANs for cyber-physical control can be distinguished, as shown in Figs. 4.3 and 4.4, respectively [3]. These architectures are called automated architecture and semiautomated architecture, respectively in [11].

In the first type of architecture as shown in Fig. 4.3a, there is no explicit controller entity in the WSAN. In this case, controllers are embedded into the actuators and control algorithms for making decisions on what actions should be performed upon the physical systems will be executed on the actuator nodes. The data gathered by sensors will be transmitted directly to the corresponding actuators via single-hop or multi-hop communications. The actuators then process all incoming data by executing pre-designed control algorithms and perform appropriate actions. From the control perspective, the actuator nodes serve as not only the actuators but also the controllers in control loops. From a high-level view, wireless communications over WSANs are involved only in transmitting the sensed data from sensors to actuators. Control commands do not need to experience any wireless transmission because the controllers and the actuators are integrated, as shown in Fig. 4.3b. In this chapter, we consider cyber-physical control systems with this architecture.

Figure 4.4a shows an alternative type of architecture, in which one or more controller entities explicitly exist in the WSAN. The controller entities could be

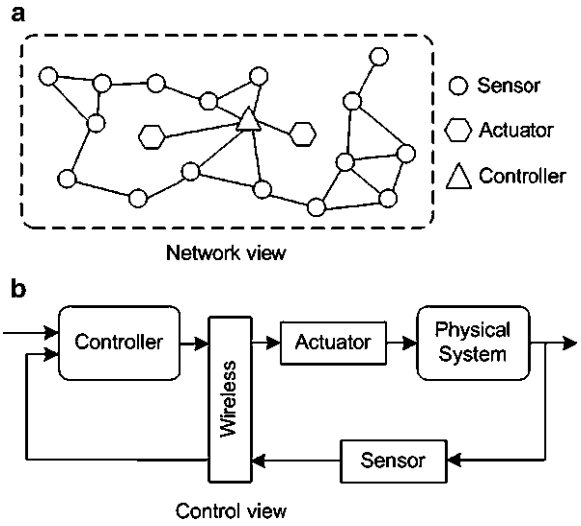


Fig. 4.4 WSAN architecture with explicit controllers: (**a**) network view; (**b**) control view

functional modules embedded in the base stations or separated nodes equipped with sufficient computation and communication capacities. With this architecture, sensors send the collected data to the controller entities. The controller entities then execute certain control algorithms to produce control commands and send them to actuators. Finally, the actuators perform the actions. In this context, both the sensor data and control commands need to be transmitted wirelessly in a single-hop or multi-hop fashion. A high-level view of the applications of this architecture is depicted in Fig. 4.4b.

4.4 QoS Challenges

Cyber-physical control imposes considerable QoS requirements on WSANs. For instance, in a fire handling system built upon a WSAN, sensors need to report the occurrence of a fire to actuators in a timely and reliable fashion; then the actuators equipped with water sprinklers will react by a certain deadline so that the situation will not become uncontrollable. Depending on the type of application, QoS in WSANs can be characterized by reliability, timeliness, robustness, availability, and security, among others.

WSAN in nature is a special category of wireless networks, which has its own characteristics besides the previously mentioned properties of wireless channels. These unique characteristics make it quite difficult to provide QoS support in control systems over WSANs. Some major challenges in this context are described in the following [2].

4.4.1 Resource Constraints

Wireless sensor nodes are usually low-cost, low-power, small devices that are equipped with only limited data processing capability, transmission rate, battery energy, and memory. For example, the MICAz mote from Crossbow is based on the Atmel ATmega128L 8-bit microcontroller that provides only up to 8 MHz clock frequency, 128-KB flash program memory, and 4-KB EEPROM; the transmit data rate is limited to 250 Kbps. Due to the limitation on transmission power, the available bandwidth and the radio range of the wireless channel are often limited. In particular, energy conservation is critically important for extending the lifetime of the network, because it is often infeasible or undesirable to recharge or replace the batteries attached to sensor nodes once they are deployed. Actuator nodes typically have stronger computation and communication capabilities and more energy budget relative to sensors. Despite this fact, resource constraints apply to both sensors and actuators.

In the presence of resource constraints, the network QoS may suffer from the unavailability of computing and/or communication resources. For instance, a number of nodes that want to transmit messages over the same WSAN have to compete for the limited bandwidth that the network is able to provide. As a consequence, some data transmissions will possibly experience large delays, resulting in low level of QoS. Due to the limited memory size, data packets may be dropped before the nodes successfully send them to the destination. Therefore, it is of critical importance to use the available resources in WSANs in a very efficient way.

4.4.2 Platform Heterogeneity

Sensors and actuators do not share the same level of resource constraints, as mentioned above. Possibly designed using different technologies and with different goals, they are different from each other in many aspects, such as computing/communication capabilities, functionality, and number. In a large-scale system of systems, the hardware and networking technologies used in the underlying WSANs may differ from one subsystem to another. This is true because of the lack of relevant standards dedicated to WSANs and hence commercially available products often have disparate features. This platform heterogeneity makes it very difficult to make full use of the resources available in the integrated system. Consequently, resource efficiency cannot be maximized in many situations. In addition, the platform heterogeneity also makes it challenging to achieve real-time and reliable communication between different nodes.

4.4.3 Dynamic Network Topology

Unlike WSNs where (sensor) nodes are typically stationary, the actuators in WSANs may be mobile [12]. In fact, node mobility is an intrinsic nature of many applications such as, among others, intelligent transportation, assisted living, urban warfare, planetary exploration, and animal control. During runtime, new sensor/actuator nodes may be added; the state of a node is possibly changed to or from sleeping mode by the employed power management mechanism; some nodes may even die due to exhausted battery energy. All of these factors might potentially cause the network topologies of WSANs to change dynamically.

Dealing with the inherent dynamics of WSANs requires QoS mechanisms to work in dynamic and even unpredictable environments. In this context, QoS adaptation becomes necessary; that is, WSANs must be adaptive and flexible at runtime with respect to changes in available resources. For example, when an intermediate node dies, the network should still be able to guarantee real-time and reliable communication by exploiting appropriate protocols and algorithms.

4.4.4 Mixed Traffic

In many situations, diverse applications need to share the same WSAN, inducing both periodic and aperiodic data. This feature will become increasingly evident as the scale of WSANs grows. Some sensors may be used to create the measurements of certain physical variables in a periodic manner for the purpose of monitoring and/or control. Meanwhile, some others may be deployed to detect critical events. For instance, in a smart home, some sensors are used to sense the temperature and lighting, while some others are responsible for reporting events such as the entering or leaving of a person. Furthermore, disparate sensors for different kinds of physical variables, e.g., temperature, humidity, location, and speed, generate traffic flows with different characteristics (e.g., message size and sampling rate). This feature of WSANs necessitates the support of service differentiation in QoS management.

4.5 Wireless Channel Characterization

In the following, we focus our attention on the problem of packet loss. In control systems, packet loss degrades control performance and even causes system instability. Because real-life control applications can only tolerate occasional packet losses with a certain upper bound of allowable PLR, WSAN design should minimize the occurrence of packet losses as much as possible. Ideally, every packet should be transmitted successfully from the source to the destination without loss. However, due to many factors such as low-power radio communication, variable transmission power, multi-hop transmission, noise, radio interference, and node mobility, packet loss cannot be completely avoided in WSANs.

To elaborate on this issue, it is necessary to understand the characteristics of wireless channels used by WSANs. More specifically, we need to capture the wireless link quality in terms of PLR. For this purpose, we perform simulations based on a realistic WSN link-layer model developed by Zuniga and Krishnamachari [21]. Some of the simulation results will be presented in this section. In the literature, experiments have been conducted for analysis of the link quality in WSNs, e.g., [22–25].

Simulation parameters used in the channel model are set according to the profile of MICA2 motes, as used in [21]. In this work, we examine the impact of two major factors: communication distance (i.e., the physical distance between the transmitter and the receiver, denoted d) and transmission power (of the transmitter), though there are many other factors that could induce packet loss. For simplicity, the effect of medium access contention between different nodes is not taken into consideration.

Figure 4.5 depicts the relationship between PLR and the transmitter–receiver distance. In this case, the transmission power is set to 0 dBm (a high level). For every distance ranging from 1 to 15 m with steps of 1 m, 80 independent measures are given.

As we can see from Fig. 4.5, the link quality is highly related to the transmitter–receiver distance. Depending on the link quality, the whole area can be divided into three regions: connected region, transitional region, and disconnected region [21]. According to Fig. 4.5, the connected region in this case corresponds to the distances between 0 and 3 m. In this region, all packets sent by the transmitter will be received

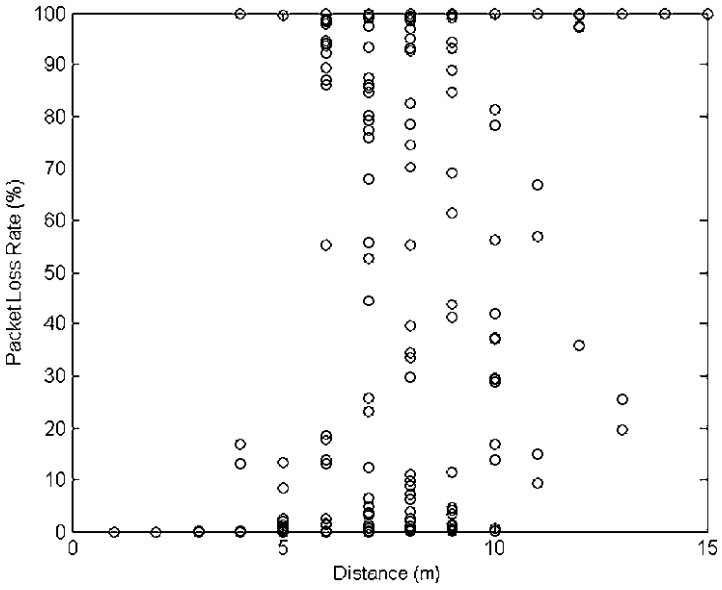


Fig. 4.5 Packet loss rate with respect to distance (transmission power: 0 dBm)

successfully by the destination node (i.e., the receiver), implying a PLR of zero. In contrast, when the receiver resides within the disconnected region (corresponding to $d \geq 14$ m), no packet will be received. Approximately, the radio range is around 13 m. These observations are very easy to understand since the strength of an electromagnetic signal decays with respect to distance during propagation. The packet cannot be received if the received signal strength is below the receive sensitivity of the destination node.

The transitional region deserves special attention, which corresponds to the distances between 4 and 13 m. Within this region, the PLR associated with the link could vary drastically. This is true even for a given distance. For instance, for distances between 4 and 10 m, the PLR could vary between 0% and 100%. This demonstrates the uncertainty as well as unreliability of the wireless link. The uncertainty of PLR for a given distance can be explained by the fact that the signal strength is random and often log-normally distributed about the mean distance-dependent value [21].

To examine the influence of transmission power, we change the transmission power to a low level, i.e., −10 dBm, and re-run the simulations. The results are reported in Fig. 4.6. It can be seen that the radio range is significantly reduced at a low level of transmission power. When the distance reaches 8 m, the link becomes disconnected, with a PLR of 100%. In comparison with Fig. 4.5, the size of the connected region shrinks in Fig. 4.6. Within the transitional region (3 m $\leq d \leq$ 7 m) in Fig. 4.6, the PLR is highly uncertain, which is similar to the case of high transmission power (Fig. 4.5).

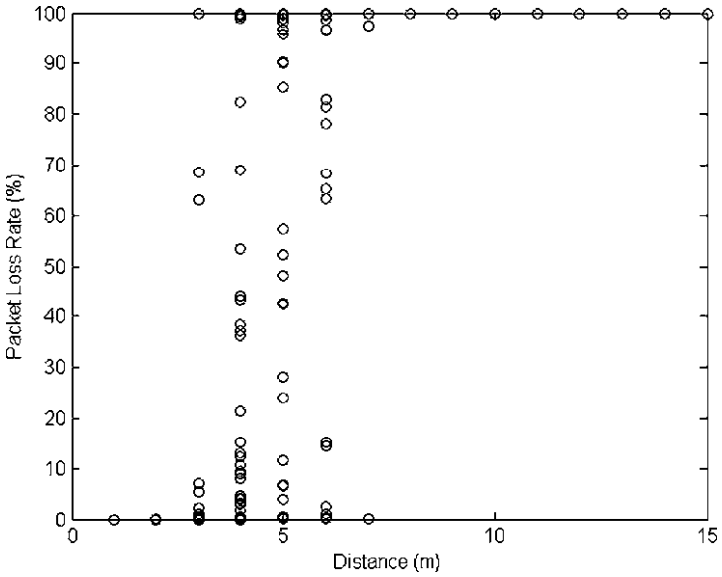


Fig. 4.6 Packet loss rate with respect to distance (transmission power: −10 dBm)

In summary, the PLR of wireless channels could vary significantly with respect to distance and transmission power. The radio range and the size of connected region (corresponding to reliable links) depend heavily on the transmission power. Generally speaking, they increase with the level of transmission power. In the transitional region, the wireless link becomes unreliable, featuring uncertain PLR. In the context of cyber-physical control, this unreliability of wireless links and the uncertainty of PLR raise crucial challenges.

4.6 Packet Loss Compensation

In this section, we present a simple solution that can be used in cyber-physical control systems over WSANs to cope with packet loss. Due to the above-analyzed characteristics of wireless channels, it is important to develop a platform-independent paradigm to enhance the reliability of WSANs under lossy conditions. A desirable solution should be widely applicable to diverse application scenarios with different system and environment setups.

Based on these observations, we attempt to develop an approach for packet loss compensation, which conforms to the following principles: (1) modifying only the application layer of the networks without exploiting any application-specific (lower layer) network protocols, (2) not using any statistic information about the distribution of PLR in any specific WSAN, and (3) not using the knowledge about the models of the controlled physical systems and the controller design of the applications.

4.6.1 A Simple Solution

In this work, we employ a simple method on the actuator nodes to cope with packet loss occurring in WSANs. The basic idea is whenever a sensor data packet is lost, the actuator will produce an estimate of the sensed value and compute the control command (usually called control input in control terms) based on this value. Note that in our previous work [3], we proposed a method that predicts directly the control command based on previous control command values, which is different from the method used here.

Let y denote the controlled variable, i.e., measurement of system output. Suppose that the kth sensed data, i.e., $y(k)$, is lost. From a control perspective, k corresponds to sampling instance in discrete time. The actuator will calculate an estimate of $y(k)$, denoted $\hat{y}(k)$, using a predication algorithm, say $f(g)$. Accordingly, we have:

$$\hat{y}(k) = f(y(k-1), y(k-2), \ldots, y(k-m)) \tag{4.1}$$

where m represents the number of history data that are stored temporarily for the purpose of prediction. Intuitively, m is an important design parameter of the algorithm, which determines the overhead in terms of memory requirements.

For simplicity, we do not consider the effect of delay. More specifically, we assume zero transmission delay in this work. Using (4.1), the actuator predicts $y(k)$ based on the previous m consecutive measurements (which are also possibly predicted values) in the case of packet loss. $\hat{y}(k)$ will then be used to compute the control command. Given that the accuracy of the prediction is sufficiently high, proper actions will be performed on the controlled physical system regardless of the loss of the sensed data. In this way, the effect of packet loss on the performance of the control applications can be substantially reduced. From the application's viewpoint, the reliability of the WSAN is improved. It is worth noting that the value of $\hat{y}(k)$ will be stored (as $y(k)$) when $y(k)$ is lost, and this value will then be used as $y(k)$ whenever necessary (e.g., if a later packet is lost).

The work flow of the actuator (running at every sampling instance) can be illustrated as follows:

Input: Sensed data
Output: Control command
Begin
  If the sensed data $y(k)$ is lost then
    Compute $\hat{y}(k)$ using (4.1)
    Set $y(k) = \hat{y}(k)$
  End if
  Produce control command with respect to $y(k)$ (through executing control algorithm)
  Store $y(k)$ into memory
  Discard $y(k-m)$ in the memory
  Perform actions corresponding to the control command
End

It is clear that this solution is quite simple. The major overhead is a small fraction of memory for temporarily storing the previous m measurements. Despite this, it does not depend on any knowledge about the underlying platform, environment, link quality characteristics, models of the controlled systems, or controller design.

4.6.2 Prediction Algorithms

It is intuitive that the performance of the above solution is closely related to the prediction accuracy of the algorithm employed, i.e., $f(g)$. Therefore, the design/choice of the prediction algorithm is important in this context. Furthermore, resource-constrained sensor and actuator nodes favor simple algorithms that yield small computational overheads.

Many existing prediction algorithms could be employed here. In this work, we explore three types of classic prediction algorithms, which are detailed below.

Algorithm 1

This algorithm is based on the assumption that the state of the physical system does not change during the last sampling period. It can be formulated as follows:

$$\hat{y}(k) = y(k-1) \tag{4.2}$$

Algorithm 2

The second algorithm computes a moving average of the previous m samples. This average is then used as the predicted value. Accordingly, we have:

$$\hat{y}(k) = \frac{1}{m}\sum_{i=1}^{m} y(k-i) \tag{4.3}$$

Algorithm 3

A property of Algorithm 2 is that it treats every previous measurement equally. Algorithm 3 represents an alternative method by giving different weights to previous measurements. More specifically, this algorithm is given by:

$$\hat{y}(k) = \alpha \times y(k-1) + (1-\alpha) \times y(k-2), \tag{4.4}$$

where α is a design parameter that commonly satisfies $0 < \alpha < 1$.

4.7 Simulation Results

In this section, we conduct simulations using Matlab to evaluate the performance of the solution and algorithms presented in the previous section. The objective is to examine their potential in coping with packet loss in cyber-physical control systems over WSANs. We first describe the simulation settings, and then analyze the simulation results.

4.7.1 Setup Overview

Consider a physical system that can be modeled in transfer function as follows:

$$G(s) = \frac{1000}{s^2 + s} \tag{4.5}$$

The controller uses the PID (proportional-integral-derivative) control law, the most popular control law in practical control applications. Controller parameters are chosen according to [26]. The sampling period of the sensor is set to 10 ms.

The integral of absolute error (IAE, a widely used performance metric in control community) is recorded to measure the performance of the control application. IAE is calculated as:

$$\mathrm{IAE}(t) = \int_0^t |r(\tau) - y(\tau)| \, \mathrm{d}\tau, \tag{4.6}$$

where t denotes (simulation) time and $r(t)$ the desired system output. Note that, in general, larger IAE values imply worse performance.

To examine the effects of different levels of packet loss, we consider the following values of PLRs: 0, 20, and 40%, respectively. To reflect the statistical effect of random packet loss, each simulation runs 100 s, which is equal to ten thousand sampling periods. For Algorithm 2, we set m to 3. For Algorithm 3, $\alpha = 0.7$.

4.7.2 Results and Analysis

Figure 4.7 depicts a piece of the system output when there is no packet loss (i.e., PLR $= 0$). Although there is no need of loss recovery, this case will serve as a baseline for studying the impact of packet loss. It can be seen that the control performance is quite good when no packet is lost (i.e., all packets are successfully transmitted).

When the PLR becomes 20%, the (accumulated) IAE values corresponding to the three algorithms presented in Sect. 4.6.2 are given in Fig. 4.8, along with the

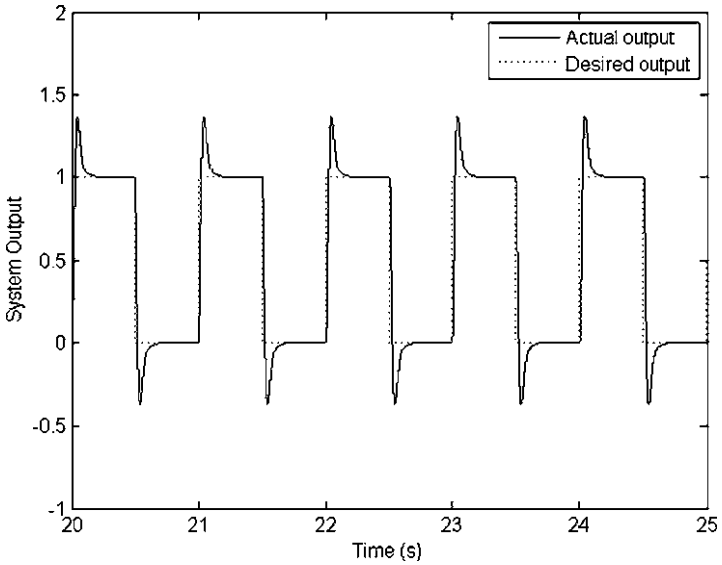


Fig. 4.7 System output in the case of no packet loss

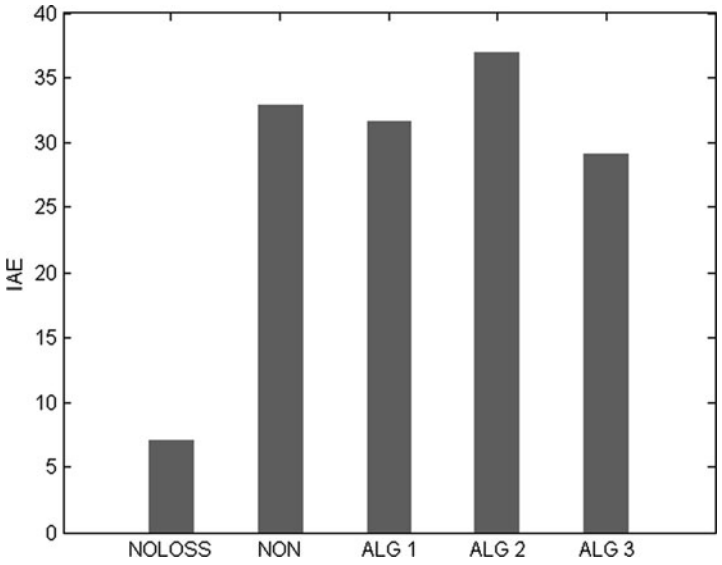


Fig. 4.8 Accumulated IAE values with a PLR of 20%

IAE value in the case of no compensation (denoted NON). In addition, the IAE value for the case of no packet loss (denoted NOLOSS) is also given for the purpose of comparison. As we can see, when the PLR changes from 0 to 20%, the IAE value increases from 7.1 to 32.9 if no compensation method for packet loss is employed. This confirms the fact that packet loss deteriorates control performance. In this case, the three algorithms yield comparable control performance which makes insignificant difference from the case of no compensation.

Figure 4.9 depicts the accumulated IAE values associated with the case in which PLR is 40%. Due to increase in PLR, the IAE values in this case are much larger than those in Fig. 4.8, implying that the control performance become worse. For example, for the solution that does not use any compensation method, the IAE value reaches 169.6 when PLR is 40%, while this value is 32.9 with a PLR of 20%. In the case of a PLR of 40%, all of the three algorithms result in better control performance than the no compensation solution, which is indicated by the relatively smaller IAE values. For instance, the IAE value associated with Algorithm 3 is 100.6, which is less than 60% of that of the no compensation solution.

It is noteworthy that different results might possibly be obtained for different simulation runs due to the uncertainty of packet loss, though every simulation run in our experimentation lasts a considerably large number of (i.e., 10,000) sampling periods. Figures 4.8 and 4.9 are only some results that are representative of many other results we have obtained. From these results, we could make the remark that the prediction algorithm needs to be designed very carefully to effectively tackle the problem of unpredictable packet loss, and, furthermore, this is often difficult.

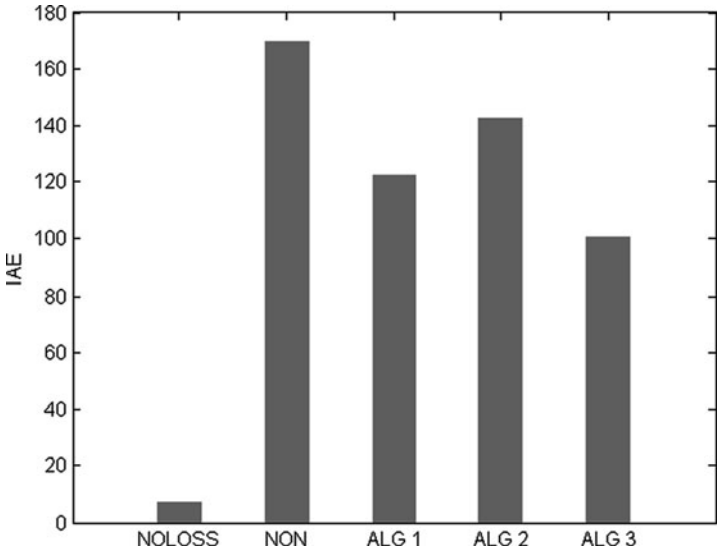


Fig. 4.9 Accumulated IAE values with a PLR of 40%

4.8 Conclusion

This chapter has dealt with the topic of how to construct cyber-physical control systems over WSANs that are unreliable. We have examined the system architecture and relevant QoS challenges. The behavior of wireless channels in terms of PLR has been captured by means of simulations using a realistic link-layer model. We presented a simple solution for addressing the problem of packet loss. This solution can be used as a generic framework in which many existing prediction algorithms are applicable. We have also conducted simulations to evaluate the performance of three simple algorithms. The results give some interesting insights useful for control over lossy WSANs. However, it remains open to devise simple yet efficient prediction algorithms for packet loss compensation.

Acknowledgments This work is partially supported by Natural Science Foundation of China under Grant No. 60903153.

References

1. J.A. Stankovic, When Sensor and Actuator Networks Cover the World, ETRI Journal, 30(5), 627–633, 2008
2. F. Xia, QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks", Sensors, 8(2), 1099–1110, 2008
3. F. Xia, Y.-C. Tian, Y. Li, Y. Sun, Wireless Sensor/Actuator Network Design for Mobile Control Applications, Sensors, 7(10), 2157–2173, 2007

4. F. Xia, W. Zhao, Y. Sun, Y.-C. Tian, Fuzzy Logic Control Based QoS Management in Wireless Sensor/Actuator Networks, Sensors, 7(12), 3179–3191, 2007
5. NSF Workshop on Cyber-Physical Systems, http://varma.ece.cmu.edu/cps/, Oct. 2006
6. F. Xia, L. Ma, J. Dong, Y. Sun, Network QoS Management in Cyber-Physical Systems, The Int.Conf. on Embedded Software and Systems (ICESS), IEEE, Chengdu, China, pp: 302–307, July 2008
7. A. Willig, K. Matheus, A. Wolisz, Wireless technology in industrial networks, Proceedings of the IEEE, 93(6), 1130–1151, 2005
8. F. Xia, W. Zhao, Flexible Time-Triggered Sampling in Smart Sensor-Based Wireless Control Systems, Sensors, 7(11), 2548–2564, 2007
9. N. Ploplys, P. Kawka, A. Alleyne, Closed-Loop Control over Wireless Networks, IEEE Control Systems Magazine, 24(3), 58–71, 2004
10. W. Wolf, Cyber-physical Systems, IEEE Computer, 42(3), 88–89, 2009
11. I.F. Akyildiz, I.H. Kasimoglu, Wireless sensor and actor networks: research challenges, Ad Hoc Networks, 2(4), 351–367, 2004
12. T. Melodia, D. Pompili, I.F. Akyildiz, Handling Mobility in Wireless Sensor and Actor Networks, IEEE Transaction on Mobile Computing, 9(2), 160–173, 2010
13. A. Rezgui, M. Eltoweissy, Service-Oriented Sensor-Actuator Networks, IEEE Communications Magazine, 45(12), 92–100, 2007
14. P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, G. Bishop-Hurley, Wireless Ad-hoc Sensor and Actuator Networks on the Farm", Proc. of IPSN'06, April 2006, Nashville, Tennessee, USA
15. S.-F. Li, Wireless sensor actuator network for light monitoring and control application, Proc. 3rd IEEE Consumer Communications and Networking Conf., 2006, pp. 974–978
16. S. Oh, L. Schenato, P. Chen, S. Sastry, Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments, Proceedings of the IEEE, 95(1), 234–254, 2007
17. S. Bosch, M. Marin-Perianu, R. Marin-Perianu, H. Scholten, P. Havinga, FollowMe! Mobile Team Coordination in Wireless Sensor and Actuator Networks, IEEE Int. Conf. on Pervasive Computing and Communications (PerCom), Mar 2009, pp. 1–11
18. H.-J. Korber, H. Wattar, G. Scholl, Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications, IEEE Transaction on Industrial Informatics, 3(2), 111–119, 2007
19. G. Nikolakopoulos, A. Panousopoulou, A. Tzes, J. Lygeros, Multi-hopping induced gain scheduling for wireless networked control systems, Proc. of the 44th IEEE Conf. on Decision and Control, and the European Control Conf., Seville, Spain, Dec 2005
20. J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A Survey of Recent Results in Networked Control Systems, Proceedings of the IEEE, 95, 1, 138–162, 2007
21. M. Zuniga, B. Krishnamachari, Analyzing the transitional region in low power wireless links, In: Proc. of the IEEE 1st Annual Conf. on Sensor and Ad Hoc Communications and Networks (SECON), 2004, pp. 517–526
22. A. Cerpa, N. Busek, D. Estrin, SCALE: A Tool for Simple Connectivity Assessment in Lossy Environments, Tech. Rep. 0021, Center for Embedded Networked Sensing, UCLA, 2003
23. A. Woo, T. Tong, D. Culler, Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks, Proc. 1st ACM Conf. Embedded Network Sensor Systems (SenSys), Nov 2003, Los Angeles, California, USA.
24. L. Tang, K.-C. Wang, Y. Huang, F. Gu, Channel Characterization and Link Quality Assessment of IEEE 802.15.4 Compliant Radio for Factory Environments, IEEE Transaction on Industrial Informatics, 3(2), 99–110, 2007
25. J. Zhao, R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, Proc. 1st ACM Conf. Embedded Network Sensor Systems (SenSys), Nov 2003, Los Angeles, California, USA
26. F. Xia, G. Tian, Y. Sun, Feedback Scheduling: An Event-Driven Paradigm, ACM SIGPLAN Notices, 42(12), 7–14, 2007

Chapter 5
Kalman Filtering Over Wireless Fading Channels

Yasamin Mostofi and Alireza Ghaffarkhah

Abstract In this chapter, we consider the estimation of a dynamical system over a wireless fading channel using a Kalman filter. We develop a framework for understanding the impact of stochastic communication noise, packet drop and the knowledge available on the link qualities on Kalman filtering over fading channels. We consider three cases of "full-knowledge", "no-knowledge" and "partial-knowledge", based on the knowledge available on the communication quality. We characterize the dynamics of these scenarios and establish the necessary and sufficient conditions to ensure stability. We then propose new ways of optimizing the packet drop to minimize the average estimation error variance of the Kalman filter. We consider both adaptive and non-adaptive optimization of the packet drop. Our results show that considerable performance improvement can be achieved by using the proposed framework.

Keywords Network control systems · Estimation over realistic wireless links · Fading channels · Kalman filtering

5.1 Introduction

The unprecedented growth of sensing, communication and computation, in the past few years, has created the possibility of exploring and controlling the environment in ways not possible before. A wide range of sensor network applications have emerged, from environmental monitoring, emergency response, smart homes and factories to surveillance, security and military applications [1, 2]. The vision of a multi-agent mobile network cooperatively learning and adapting in harsh unknown environments to achieve a common goal is closer than ever. In such networks, each node has limited capabilities and needs to cooperate with others to achieve the task. Therefore, there can be cases that an agent senses a parameter of interest and sends

Y. Mostofi (✉)
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87113, USA
e-mail: ymostofi@ece.unm.edu

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_5,

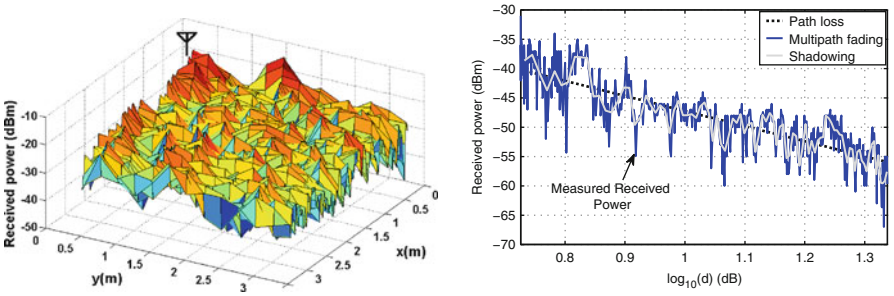


Fig. 5.1 Channel measurement (*left*) in the Cooperative Network Lab and (*right*) along a hallway in the basement of the ECE building at the University of New Mexico

its measurement wirelessly to a remote node, which will be in charge of estimation and possibly producing a control command. Wireless communication then plays a key role in the overall performance of such cooperative networks as the agents share sensor measurements and receive control commands over wireless links.

In a realistic communication setting, such as an urban area or indoor environment, Line-Of-Sight (LOS) communication may not be possible due to the existence of several objects that can attenuate, reflect, diffract or block the transmitted signal. The received signal power typically experiences considerable variations and can change drastically in even a small distance. Figure 5.1 (left and solid dark curve of right), for instance, shows examples of channel measurements in the Electrical and Computer Engineering (ECE) building at the University of New Mexico. It can be seen that channel can change drastically with a small movement of an agent. In general, communication between sensor nodes can be degraded due to factors such as fading, shadowing or distance-dependent path loss [3, 4]. As a result, the received Signal-to-Noise Ratio (SNR) of some of the receptions can be too low resulting in a packet drop. Furthermore, poor link quality can result in bit flips and therefore the packets that are kept are not necessarily free of error. As a result, sensing and control can not be considered independent of communication issues. An integrative approach is needed, where sensing, communication and control issues are jointly considered in the design of these systems.

Considering the impact of communication on networked estimation and control is an emerging research area, which has received considerable attention in the past few years. Among the uncertainties introduced by communication, impact of quantization on networked estimation and control has been studied extensively [5, 6]. Stabilization of linear dynamical systems in the presence of quantization noise has been characterized [7–9] and trade-offs between information rate and convergence time is formulated [10]. To address the inadequacy of the classical definition of capacity for networked control applications, anytime capacity was introduced and utilized for stabilization of linear systems [11]. Disturbance rejection and the corresponding required extra rate in these systems were considered in [12]. Nilsson studied the impact of random delays on stability and optimal control of networked control systems [13].

In estimation and control over a wireless link, poor link qualities are the main performance degradation factors. Along this line, Micheli et al. investigated the impact of packet loss on estimation by considering random sampling of a dynamical system [14]. This is followed by the work of Sinopoli et al. which derived bounds for the maximum tolerable probability of packet loss to maintain stability [15]. Several other papers have appeared since then that consider the impact of packet loss or random delays on networked control systems [16–21].

The current framework in the literature, however, does not consider the impact of stochastic communication errors on estimation and control over wireless links. Transmission over a wireless fading channel, as shown in Fig. 5.1, can result in a time-varying packet drop. Furthermore, the packets that are kept are not necessarily free of noise. The variance of this noise is also stochastic due to its dependency on the SNR. Therefore, we need to consider the impact of both packet drop and stochastic communication noise on the performance. Moreover, the knowledge on the communication reception quality may not be fully available to the estimator, resulting in different forms of Kalman filtering. In [22,23], we considered the impact of communication errors on estimation over wireless links. We showed that dropping all the erroneous packets may not be the optimum strategy for estimation of a rapidly-changing dynamical system over a wireless link. In this chapter, we extend our previous work and characterize the impact of both packet drop and communication errors on Kalman filtering over wireless fading channels. We are interested in characterizing the impact of the knowledge available on the link qualities on Kalman filtering over wireless channels. We consider three cases of "full-knowledge", "no-knowledge" and "partial-knowledge", based on the knowledge available on the communication quality. We characterize the dynamics of these scenarios and establish necessary and sufficient conditions to ensure stability. We then show how to optimize the packet drop in the physical layer to minimize the average estimation error variance of the Kalman filter. Our approach is truly integrative as the available knowledge on link qualities is utilized in the estimator and current quality of estimation is used in some of our proposed packet drop designs.

We conclude this section with an overview of the chapter. In Sect. 5.2, we formulate the problem and summarize key features of a transmission over a wireless fading channel. In Sect. 5.3, we consider and characterize three possible Kalman filtering scenarios, depending on the knowledge available on the communication error variance in the estimator. In Sect. 5.4, we then optimize the packet drop mechanism. We show how dropping all the erroneous packets (or keeping them all) may not be the optimum strategy and that packet drop should be designed and adapted based on the knowledge available on the link and estimation quality as well as the dynamics of the system under estimation. We conclude in Sect. 5.5.

5.2 System Model

Consider a mobile sensor observing a system with a linear dynamics as follows:

$$x[k+1] = Ax[k] + w[k] \qquad \text{and} \qquad y[k] = Cx[k] + v[k], \tag{5.1}$$

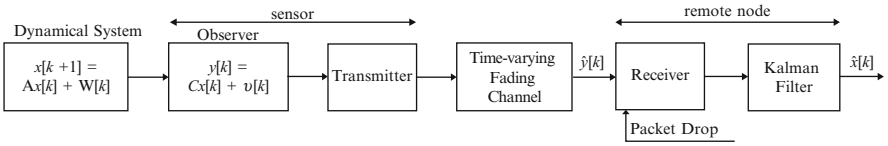


Fig. 5.2 A schematic of estimation over a wireless fading channel

where $x[k] \in \mathbb{R}^N$ and $y[k] \in \mathbb{R}^M$ represent the state and observation, respectively. $w[k] \in \mathbb{R}^N$ and $v[k] \in \mathbb{R}^M$ represent zero-mean Gaussian process and observation noise vectors with variances of $Q \succcurlyeq 0$ and $R_s \succ 0$, respectively. In this chapter, we take $M = N$ and C invertible to focus on the impact of time-varying fading channels and the resulting communication error on Kalman filtering. We are interested in estimating unstable dynamical systems. Therefore, matrix A has one or more of its eigenvalues outside the unit circle. It should, however, be noted that the proposed optimization framework of this chapter is also applicable to the case of a stable A. Furthermore, we take the pair $(A, Q^{1/2})$ to be controllable. The sensor then transmits its observation over a *time-varying fading channel* to a remote node, which is in charge of estimation. Figure 5.2 shows the high-level schematic of the considered problem. In this chapter, we consider and optimize such problems, i.e. Kalman filtering in the presence of packet drop and stochastic communication noise.

5.2.1 An Overview of Wireless Communications [3, 4, 24]

In this section, we briefly describe the key factors needed for modelling the impact of a time-varying fading channel on Kalman filtering.

5.2.1.1 Received Signal-to-Noise Ratio

A fundamental parameter that characterizes the performance of a communication channel is the received Signal-to-Noise Ratio (SNR). Received SNR is defined as the ratio of the received signal power divided by the receiver thermal noise power. SNR is a key factor in determining whether a received packet will be kept and used in the receiving node or not. Furthermore, it determines how well the received bits and, as a result, the received measurement can be retrieved. Let $\Upsilon[k]$ represent the instantaneous received SNR at the kth transmission. We will have $\Upsilon[k] = \frac{|h[k]|^2 \sigma_s^2}{\sigma_T^2}$, where $h[k] \in \mathbb{C}$ represents the time-varying coefficient of the baseband equivalent channel during the transmission of $x[k]$, $\sigma_s^2 = \mathbb{E}(|s|^2)$ is the transmitted signal power and $\sigma_T^2 = \mathbb{E}(|n_{\text{thermal}}|^2)$ is the power of the receiver thermal noise. As the sensor moves, the remote node will experience different channels and therefore different SNRs.

5.2.1.2 Probabilistic Characterization of SNR

In wireless communications, it is common to model the channel (and as a result the received SNR) probabilistically, with the goal of capturing its underlying dynamics. The utilized probabilistic models are the results of analyzing several empirical data over the years. In general, a communication channel between two nodes can be modelled as a multi-scale dynamical system with three major dynamics: *multi-path fading*, *shadowing* and *path loss*. Figure 5.1 (right) shows the received signal power across a route in the basement of ECE building at UNM [24]. The three main dynamics of the received signal power are marked on the figure. When a wireless transmission occurs, replicas of the transmitted signal will arrive at the receiver due to phenomena such as reflection and scattering. Different replicas can be added constructively or destructively depending on the phase terms of individual ones. As a result, with a small movement of a node, the phase terms can change drastically, resulting in the rapid variations of the channel. Such rapid variations are referred to as multi-path fading and can be seen from Fig. 5.1 (right, solid dark curve). The higher the number of reflectors and scatterers in the environment, the more severe small-scale variations could be. By spatially averaging the received signal locally and over distances that channel can still be considered stationary, a slower dynamic emerges, which is called shadowing (light gray curve of Fig. 5.1 right). Shadowing is the result of the transmitted signal being possibly blocked by a number of obstacles before reaching the receiver. Empirical data has shown lognormal to be a good match for the distribution of shadowing. Finally, by averaging over the variations of shadowing, a distance-dependent trend is seen, which is, in dB, an affine function of the log of the distance between the transmitter and receiver.

Since SNR is proportional to the received signal power, it has similar dynamics. Therefore, if the movement of the transmitting node is confined to a small area, then $\Upsilon[k]$ can be considered a stationary stochastic process. On the other hand, if the node is moving fast and over larger distances during the estimation process, then the non-stationary nature of the channel should also be considered by taking into account shadowing. Finally, for movements over even larger areas, non-stationary behaviour of shadowing should also be considered by taking into account the underlying path loss trend. In this chapter, we consider the multi-path fading, i.e. we model the SNR as a stochastic but stationary process. Our results are easily extendable to the cases of non-stationary channels.

5.2.1.3 Distribution of Fading

In this chapter, we use the term fading to refer to multi-path fading. Over small enough distances where channel (or equivalently SNR) can be considered stationary, it can be mathematically shown that Rayleigh distribution is a good match for the distribution of the square root of SNR if there is no LOS path while Rician provides

a better match if an LOS exists. A more general distribution is Nakagami [25], which has the following pdf for $\Upsilon_{\text{sqrt}} \triangleq \sqrt{\Upsilon}$:

$$\chi(\Upsilon_{\text{sqrt}}) = \frac{2m^m \Upsilon_{\text{sqrt}}^{2m-1}}{\Gamma(m)\Upsilon_{\text{ave}}^m} \exp\left(\frac{-m\Upsilon_{\text{sqrt}}^2}{\Upsilon_{\text{ave}}}\right) \tag{5.2}$$

for $m \geq 0.5$, where m is the fading parameter, Υ_{ave} denotes the average of SNR and $\Gamma(.)$ is the Gamma function. If $m = 1$, this distribution becomes Rayleigh whereas for $m = \frac{(m'+1)^2}{2m'+1}$, it is reduced to a Rician distribution with parameter m'. These distributions also match several empirical data. In this chapter, we do not make any assumption on the probability distribution of Υ. Only when we want to provide an example, we will take Υ to be exponentially distributed (i.e. the square root is Rayleigh), which is a common model for outdoor fading channels with no LOS path. We also take the SNR to be uncorrelated from one transmission to the next.

5.2.1.4 Stochastic Communication Noise Variance

The sensor node of Fig. 5.2 quantizes the observation, $y[k]$, transforms it into a packet of bits and transmits it over a fading channel. The remote node will receive a noisy version of the transmitted data due to bit flip. Let $\hat{y}[k]$ represent the received signal, as shown in Fig. 5.2. $\hat{y}[k]$ is what the second node assumes the kth transmitted observation was. Let $n[k]$ represent the difference between the transmitted observation and the received one:

$$n[k] = \hat{y}[k] - y[k]. \tag{5.3}$$

We refer to $n[k]$ as communication error (or communication noise) throughout the chapter. This is the error that is caused by poor link quality and the resulting flipping of some of the transmitted bits during the kth transmission.[1] The variance of $n[k]$ at the kth transmission is given by

$$R_c[k] = \mathbb{E}\big(n[k]n^T[k]|\Upsilon[k]\big) = \sigma_c^2\big(\Upsilon[k]\big)I, \tag{5.4}$$

where I represents the identity matrix throughout this chapter and $\sigma_c^2\big(\Upsilon[k]\big)$ is a non-increasing function of SNR that depends on the transmitter and receiver design principles, such as modulation and coding, as well as the transmission environment. It can be seen that the communication noise has a stochastic variance as the variance is a function of SNR. Throughout the chapter, we use $\mathbb{E}(z)$ to denote average of the variable z. Note that we took the communication error in the transmission of different elements of vector $y[k]$ to be uncorrelated due to fast fading. This assumption is

[1] Note that for transmissions over a fading channel, the impact of quantization is typically negligible as compared to the communication errors. Therefore, in this chapter, we ignore the impact of quantization as it has also been heavily explored in the context of networked control systems.

mainly utilized in Sect. 5.4 for the optimization of packet drop. To keep our analysis general, in this chapter we do not make any assumption on the form of $\sigma_c^2(\Upsilon[k])$ as a function of $\Upsilon[k]$. By considering both the communication and observation errors, we have

$$\hat{y}[k] = Cx[k] + \underbrace{v[k] + n[k]}_{r[k]}, \tag{5.5}$$

where $r[k]$ represents the overall error that enters the estimation process at the kth time-step and has the variance of

$$R[k] = \mathbb{E}\big(r[k]r^T[k]|\Upsilon[k]\big) = R_s + R_c[k]. \tag{5.6}$$

$R[k]$ is therefore stochastic due to its dependency on the SNR. Since our emphasis is on the communication noise, throughout the chapter, we refer to $R[k]$ as the "overall communication noise variance".

5.2.1.5 Packet Drop Probability

Let $\mu[k]$ indicate if the receiver drops the kth packet, i.e. $\mu[k] = 1$ means that the kth packet is dropped while $\mu[k] = 0$ denotes otherwise. $\mu[k]$ can also be represented as a function of $\Upsilon[k]$: $\mu[k] = G\big(\Upsilon[k]\big)$. It should be noted that the receiver may not decide on dropping packets directly based on the instantaneous received SNR. However, since any other utilized measure is a function of $\Upsilon[k]$, we find it useful to express μ as a function of this fundamental parameter. Experimental results have shown G to be well approximated as follows [26]:

$$\mu[k] = \begin{cases} 0 & \Upsilon[k] \geq \Upsilon_T \\ 1 & \text{else} \end{cases} \tag{5.7}$$

This means that the receiver keeps those packets with the received instantaneous SNR above a designated threshold Υ_T. In this chapter, we show how the threshold Υ_T can be optimized to control the amount of information loss and communication error that enters the estimation process.

5.2.2 *Estimation Using a Kalman Filter*

The remote node estimates the state based on the received observations using a Kalman filter [27]. Let $\hat{x}[k] = \hat{x}[k|k-1]$ represent the estimate of $x[k]$ using all the received observations up to and including time $k-1$. Then $P[k]$ represents the corresponding estimation error variance given $\Upsilon[0], \cdots, \Upsilon[k-1]$:

$$P[k] = P[k \mid k-1] = \mathbb{E}\Big[\Big(x[k] - \hat{x}[k]\Big)\Big(x[k] - \hat{x}[k]\Big)^T \;\Big|\; \Upsilon[0], \cdots, \Upsilon[k-1]\Big]. \tag{5.8}$$

As can be seen, (5.8) is different from the traditional form of Kalman filter since $P[k]$ is a stochastic function due to its dependency on $\Upsilon[0], \Upsilon[1], \cdots, \Upsilon[k-1]$. Therefore, to obtain $\mathbb{E}\big(P[k]\big)$, $P[k]$ should be averaged over the joint distribution of $\Upsilon[0], \Upsilon[1], \cdots, \Upsilon[k-1]$.

5.2.3 *Impact of Packet Drop and Stochastic Communication Error on Kalman Filtering*

A transmission over a fading channel can result in the dropping of some of the packets. Furthermore, the packets that are kept may not be free of noise, depending on the packet drop threshold Υ_{T}. It is possible to increase the threshold such that the packets that are kept are free of error. However, this increases packet drop probability and can result in instability or poor performance. Therefore, we need to consider the impact of both packet drop and communication noise on the performance. The communication noise variance is also stochastic due to its dependency on SNR. As a result, Kalman filtering over a fading channel will not have its traditional form.

Furthermore, Kalman filtering is done in the application layer, whereas the physical layer is in charge of wireless communication. An estimate of SNR is typically available in the physical layer. Such an estimate can be translated to an assessment of the communication noise variance if a mathematical characterization for σ_c^2 exists. However, this knowledge may or may not be available in the application layer, depending on the system design. There can also be cases where an exact assessment of the communication noise variance and the resulting reception quality is only partially available in the physical layer. As a result, full-knowledge of the communication noise variance may not be available at the estimator. Depending on the available knowledge on the communication noise variance, the Kalman filter will have different forms. In this chapter, we are interested in understanding the impact of the knowledge available on the link qualities on Kalman filtering over wireless channels. We consider three cases of "full-knowledge", "no-knowledge" and "partial-knowledge", based on the knowledge available on the communication quality in the estimator. In the next section, we characterize the dynamics of these scenarios and establish necessary and sufficient conditions to ensure stability. Then, in Sect. 5.4, we show how to optimize packet drop threshold in the physical layer to minimize the average estimation error variance of the Kalman filter.

5.3 Stability Analysis

Consider the traditional form of Kalman filtering with no packet drop and full-knowledge on the reception quality. We have, $\hat{x}[k+1] = A\hat{x}[k] + K_f[k]\big(\hat{y}[k] - C\hat{x}[k]\big)$ and $P[k+1] = Q + \big(A - K_f[k]C\big)P[k]\big(A - K_f[k]C\big)^T + K_f[k]R[k]K_f^T[k]$, where $K_f[k] = AP[k]C^T\big(CP[k]C^T + R[k]\big)^{-1}$ is the

optimal gain [27]. However, as discussed in the previous section, the received packet is dropped if $\Upsilon[k] < \Upsilon_{\mathrm{T}}[k]$. Furthermore, full knowledge on the variance of the stochastic communication noise might not be available. Then, the traditional form of Kalman filter needs to be modified to reflect packet drop and the available information at the estimator. Let $0 \preceq \tilde{R}[k] \preceq R[k]$ denote the part of $R[k]$ that is known to the estimator. $\tilde{R}[k]$ can be considered what the estimator perceives $R[k]$ to be. Then, $\tilde{R}[k]$ is used to calculate the filtering gain which, when considered jointly with the packet drop, results in the following recursion for $P[k]$:

$$P[k+1] = \mu[k]AP[k]A^T + Q + \big(1-\mu[k]\big)\Big[\big(A-\tilde{K}_f[k]C\big)P[k]\big(A-\tilde{K}_f[k]C\big)^T + \tilde{K}_f[k]R[k]\tilde{K}_f^T[k]\Big], \tag{5.9}$$

where $\tilde{K}_f[k] = AP[k]C^T\big(CP[k]C^T + \tilde{R}[k]\big)^{-1}$ and $\mu[k]$ is defined in (5.7). We assume that the initial error variance, $P[0]$, is positive definite. Then, using the fact that $(A, Q^{1/2})$ is controllable, one can easily verify that $P[k] \succ 0$ for $k \geq 0$, which we utilize later in our derivations. By inserting $\tilde{K}_f$ in (5.9), we obtain

$$P[k+1] = AP[k]A^T + Q - \big(1-\mu[k]\big)AP[k]C^T\big(CP[k]C^T + \tilde{R}[k]\big)^{-1} \times \big(CP[k]C^T + 2\tilde{R}[k] - R[k]\big)\big(CP[k]C^T + \tilde{R}[k]\big)^{-1}CP[k]A^T. \tag{5.10}$$

In this part, we consider three different cases, based on the knowledge available on the link quality at the estimator:

1. Estimator has no knowledge on the reception quality: $\tilde{R}[k] = 0$
2. Estimator has full knowledge on the reception quality: $\tilde{R}[k] = R[k]$
3. Estimator has partial knowledge on the reception quality: $0 \prec \tilde{R}[k] \prec R[k]$

In the following, we derive the necessary and sufficient condition for ensuring the stability of these cases and show them to have the same stability condition. Before doing so, we introduce a few basic definitions and lemmas that will be used throughout the chapter. Note that all the variables used in this chapter are real.

Definition 5.1. We consider the estimation process stable as long as the average estimation error variance, $\mathbb{E}\big(P[k]\big)$, stays bounded.

Definition 5.2. Let $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$ represent a symmetric matrix-valued function. Then f is non-decreasing with respect to its symmetric input matrix variable if $\Pi_1 \succeq \Pi_2 \Rightarrow f\big(\Pi_1\big) \succeq f\big(\Pi_2\big)$, for any symmetric Π_1 and $\Pi_2 \in \mathbb{R}^{N\times N}$. Similarly, f is increasing if $\Pi_1 \succ \Pi_2 \Rightarrow f\big(\Pi_1\big) \succ f\big(\Pi_2\big)$.

Definition 5.3. Let f represent a symmetric matrix-valued function, $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$. Function f is convex with respect to matrix inequality if $f(\theta\Pi_1 + (1-\theta)\Pi_2) \preceq \theta f(\Pi_1) + (1-\theta) f(\Pi_2)$, for arbitrary Π_1 and $\Pi_2 \in \mathbb{R}^{N\times N}$ and $\theta \in [0,1]$. Similarly, f is concave if $-f$ is convex. See [28] for more details.

Lemma 5.1. *We have the following:*

1. *Let $\Pi_1 \in \mathbb{R}^{N\times N}$ and $\Pi_2 \in \mathbb{R}^{N\times N}$ represent two symmetric positive definite matrices. Then $\Pi_1 \preceq \Pi_2$ if and only if $\Pi_1^{-1} \succeq \Pi_2^{-1}$.*
2. *If $\Pi_1 \preceq \Pi_2$ for symmetric Π_1 and Π_2, then $\Psi\Pi_1\Psi^T \preceq \Psi\Pi_2\Psi^T$ for an arbitrary $\Psi \in \mathbb{R}^{N'\times N}$.*
3. *Let $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$ represent a symmetric function. Let $\Pi_1 \in \mathbb{R}^{N\times N}$ be symmetric. If f is an increasing (non-decreasing) function of Π_1, then $f\big(\Pi_1 + \Pi_2\big)$ is also increasing (non-decreasing) with respect to Π_1 for a constant Π_2.*
4. *If $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$ is symmetric and a non-decreasing function of symmetric Π_1, then $\Psi f\big(\Pi_1\big)\Psi^T$ is also non-decreasing as a function of Π_1 and for an arbitrary $\Psi \in \mathbb{R}^{N''\times N'}$.*
5. *Let $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ represent matrix inverse function: $f(\Pi_1) = \Pi_1^{-1}$ for a symmetric positive definite Π_1. Then f is a convex function of Π_1.*
6. *If $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$ is symmetric and a convex function of Π_1, then $f\big(\Pi_1 + \Pi_2\big)$ is also convex with respect to Π_1 and for a constant Π_2.*
7. *If $f : \mathbb{R}^{N\times N} \to \mathbb{R}^{N'\times N'}$ is symmetric and a convex function of Π_1, then $\Psi f\big(\Pi_1\big)\Psi^T$ is also convex for an arbitrary matrix $\Psi \in \mathbb{R}^{N''\times N'}$.*
8. *Let $f(\Pi_1) = \Pi_1\big(\Pi_2 + \Pi_1\big)^{-1}\Pi_1$, where Π_1 and Π_2 are symmetric positive definite matrices. f is a convex function of Π_1.*

Proof. see [29, 30]. □

Lemma 5.2. *Consider the following Lyapunov equation with Δ Hermitian: $\Sigma = \Pi\Sigma\Pi^T + \Delta$. Then the following holds:*

1. *If Π is a stable matrix (spectral radius less than one), Σ will be unique and Hermitian and can be expressed as follows:*

$$\Sigma = \sum_{i=0}^{\infty} \Pi^i \Delta\big(\Pi^T\big)^i. \tag{5.11}$$

2. *If $\big(\Pi, \Delta^{1/2}\big)$ is controllable and $\Delta \succeq 0$, then Σ will be Hermitian, unique and positive definite iff Π is stable.*

Proof. see [27]. □

5.3.1 Noise-Free Case

Consider the case where $\tilde{R}[k] = R[k] = 0$ if a packet is kept. We refer to this scenario as the noise-free case. Comparison with the dynamics of this case will help us characterize the dynamics of the three aforementioned scenarios. In this case, (5.10) is simplified to $P[k+1] = \mu[k]AP[k]A^T + Q \triangleq \Phi_{\text{perf}}(P[k], \mu[k])$, with the following average dynamics:

$$\mathbb{E}\big(P[k+1]\big) = \mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)A\mathbb{E}\big(P[k]\big)A^T + Q, \tag{5.12}$$

where $\mu_{\text{ave}}(\Upsilon_{\text{T}})$ is the average probability of packet drop:

$$\mu_{\text{ave}}(\Upsilon_{\text{T}}) = \mathbb{E}\big(\mu[k]\big) = \int_0^{\Upsilon_{\text{T}}} \chi(\Upsilon)\mathrm{d}\Upsilon, \tag{5.13}$$

with χ representing the probability density function of Υ. Let $\rho_{\max}(A)$ represent the spectral radius of matrix A. The average dynamical system of (5.12) is stable iff

$$\mu_{\text{ave}}(\Upsilon_{\text{T}}) < \rho_{\max}^{-2}(A). \tag{5.14}$$

5.3.2 Estimator has No Knowledge on the Reception Quality

From (5.10), we have the following for the case where no information on the reception quality is available at the estimator:

$$\begin{aligned} P[k+1] &= AP[k]A^T + Q - \big(1-\mu[k]\big)AP[k]C^T\big(CP[k]C^T\big)^{-1} \\ &\quad \times \big(CP[k]C^T - R[k]\big)\big(CP[k]C^T\big)^{-1}CP[k]A^T \\ &= \mu[k]AP[k]A^T + Q + (1-\mu[k])AC^{-1}R[k]\big(C^T\big)^{-1}A^T \\ &\triangleq \Phi_{\text{no}}\big(P[k], \mu[k], R[k]\big). \end{aligned} \tag{5.15}$$

The average of $P[k]$ is then given by

$$\mathbb{E}\big(P[k+1]\big) = \mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)A\mathbb{E}\big(P[k]\big)A^T + Q + AC^{-1}R_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\big(C^T\big)^{-1}A^T, \tag{5.16}$$

where $\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)$ is defined in (5.13) and $R_{\text{ave}}\big(\Upsilon_{\text{T}}\big)$ is the average noise variance that enters the estimation process:

$$R_{\text{ave}}(\Upsilon_{\text{T}}) = \mathbb{E}\big(R[k]\big) = \Big(1-\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\Big)R_{\text{s}} + \left[\int_{\Upsilon_{\text{T}}}^{\infty} \sigma_{\text{c}}^2(\Upsilon)\chi(\Upsilon)\mathrm{d}\Upsilon\right] I \tag{5.17}$$

with I denoting the identity matrix of the appropriate size.

Lemma 5.3. *Let $\rho_{\max}(A)$ represent the spectral radius of matrix A. The average dynamical system of (5.16) is stable if and only if $\mu_{ave}(\Upsilon_{\text{T}}) < \rho_{\max}^{-2}(A)$ or equivalently $\Upsilon_{\text{T}} < \Upsilon_{\text{T,c}}$, where $\Upsilon_{\text{T,c}}$ is the unique solution of the following equation:*

$$\int_0^{\Upsilon_{\text{T,c}}} \chi(\Upsilon)\mathrm{d}\Upsilon = \rho_{\max}^{-2}(A). \tag{5.18}$$

Proof. See [27]. □

Note that $\Upsilon_{\mathrm{T,c}}$ is often referred to as the *critical threshold*. In the next sections, we use the dynamics of the "noise-free" and "no-knowledge" cases to understand the dynamics of the "full" and "partial" knowledge scenarios.

5.3.3 Estimator has Full Knowledge on the Reception Quality

Consider (5.10) with $\tilde{R}[k] = R[k]$. We have

$$\begin{aligned} P[k+1] &= AP[k]A^T + Q - \big(1-\mu[k]\big)AP[k]C^T\big(CP[k]C^T + R[k]\big)^{-1}CP[k]A^T \\ &= Q + AP[k]A^T - AP[k]\left(P[k] + \Pi_z[k]\right)^{-1}P[k]A^T \\ &= Q + A\Pi_z[k]A^T - A\Pi_z[k]\big(P[k] + \Pi_z[k]\big)^{-1}\Pi_z[k]A^T \\ &\triangleq \Phi_{\mathrm{full}}\big(P[k], \mu[k], R[k]\big), \end{aligned} \tag{5.19}$$

where $\Sigma_z[k] = \begin{cases} R[k] & \mu[k] = 0 \\ \infty & \text{otherwise} \end{cases}$, $\Pi_z[k] = C^{-1}\Sigma_z[k]\big(C^T\big)^{-1}$ and the third line is written using matrix inversion lemma [29]. The following set of lemmas relate the dynamics of this case to those of the "no-knowledge" and "noise-free" cases.

Lemma 5.4. *Consider any symmetric P_1, P_2 and P_3 such that $P_1 \succeq P_2 \succeq P_3 \succ 0$. For any $0 \le \mu \le 1$ and $R \succ 0$, we have*

$$\Phi_{no}\big(P_1, \mu, R\big) \succeq \Phi_{full}\big(P_2, \mu, R\big) \succeq \Phi_{perf}\big(P_3, \mu\big). \tag{5.20}$$

Proof. For any $P \succ 0$, we have $\big(CPC^T + R\big)^{-1} = \big(CPC^T\big)^{-1} - \big(CPC^T\big)^{-1}\Big[\big(CPC^T\big)^{-1} + R^{-1}\Big]^{-1}\big(CPC^T\big)^{-1}$. Therefore, using Lemma 5.1,

$$\begin{aligned} R \succeq \Big[\big(CPC^T\big)^{-1} + R^{-1}\Big]^{-1} &\Rightarrow \big(CPC^T\big)^{-1}R\big(CPC^T\big)^{-1} \\ &\succeq \big(CPC^T\big)^{-1}\Big[\big(CPC^T\big)^{-1} + R^{-1}\Big]^{-1}\big(CPC^T\big)^{-1} \\ &\Rightarrow \big(CPC^T\big)^{-1}\big(CPC^T - R\big)\big(CPC^T\big)^{-1} \\ &\preceq \big(CPC^T + R\big)^{-1} \Rightarrow \Phi_{\mathrm{no}}\big(P, \mu, R\big) \succeq \Phi_{\mathrm{full}}\big(P, \mu, R\big). \end{aligned} \tag{5.21}$$

Furthermore, for any $\tilde{K}_f$, $\big(A - \tilde{K}_f C\big)P\big(A - \tilde{K}_f C\big)^T \succeq 0$ and $\tilde{K}_f R \tilde{K}_f^T \succeq 0$. Then from (5.9),

$$\Phi_{\mathrm{full}}\big(P, \mu, R\big) \succeq \mu APA^T + Q = \Phi_{\mathrm{perf}}\big(P, \mu\big). \tag{5.22}$$

From the third line of (5.19), it can be seen that $\Phi_{\text{full}}\big(P, \mu, R\big)$ is a non-decreasing functions of $P \succ 0$. Moreover, $\Phi_{\text{perf}}\big(P, \mu\big)$ is a non-decreasing function of $P \succ 0$. Then for $P_1 \succeq P_2 \succeq P_3 \succ 0$, we obtain

$$\Phi_{\text{no}}\big(P_1, \mu, R\big) \succeq \Phi_{\text{full}}\big(P_1, \mu, R\big) \succeq \Phi_{\text{full}}\big(P_2, \mu, R\big) \succeq \Phi_{\text{perf}}\big(P_2, \mu\big) \succeq \Phi_{\text{perf}}\big(P_3, \mu\big). \tag{5.23}$$

□

Lemma 5.5. *Let* $P_1[k+1] = \Phi_{no}\big(P_1[k], \mu[k], R[k]\big)$, $P_2[k+1] = \Phi_{full}\big(P_2[k], \mu[k], R[k]\big)$ *and* $P_3[k+1] = \Phi_{perf}\big(P_3[k], \mu[k]\big)$. *Then starting from the same initial error covariance* P_0, *we have*

$$\mathbb{E}\big(P_1[k]\big) \succeq \mathbb{E}\big(P_2[k]\big) \succeq \mathbb{E}\big(P_3[k]\big), \qquad \text{for } k \geq 0. \tag{5.24}$$

Proof. At $k = 0$, $P_1[0] = P_2[0] = P_3[0] = P_0$. Assume that at time k, $P_1[k] \succeq P_2[k] \succeq P_3[k]$. Then, from Lemma 5.4 we have,

$$\begin{aligned} &\Phi_{\text{no}}\big(P_1[k], \mu[k], R[k]\big) \succeq \Phi_{\text{full}}\big(P_2[k], \mu[k], R[k]\big) \succeq \Phi_{\text{perf}}\big(P_3[k], \mu[k]\big) \\ &\quad \Rightarrow P_1[k+1] \succeq P_2[k+1] \succeq P_3[k+1]. \end{aligned} \tag{5.25}$$

This proves that starting from the same P_0, $P_1[k] \succeq P_2[k] \succeq P_3[k]$ for $k \geq 0$ (by using mathematical induction). Furthermore, at any time k, $P_1[k]$, $P_2[k]$ and $P_3[k]$ can be expressed as functions of P_0 and the sequence $\Upsilon[0], \cdots, \Upsilon[k-1]$: $P_1[k] = \Lambda_{\text{no}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big)$, $P_2[k] = \Lambda_{\text{full}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big)$ and $P_3[k] = \Lambda_{\text{perf}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big)$. Therefore

$$\mathbb{E}\big(P_1[k]\big) = \int_0^\infty \cdots \int_0^\infty \Lambda_{\text{no}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big) \Big[\prod_{j=0}^{k-1} \chi(\Upsilon_j)\Big] \mathrm{d}\Upsilon[0] \cdots \mathrm{d}\Upsilon[k-1]. \tag{5.26}$$

Similar expressions can be written for $\mathbb{E}\big(P_2[k]\big)$ and $\mathbb{E}\big(P_3[k]\big)$. Since $\chi(\Upsilon_j)$ is non-negative for $j = 0, \cdots, k-1$ and $\Lambda_{\text{no}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big) \succeq \Lambda_{\text{full}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big) \succeq \Lambda_{\text{perf}}\big(P_0, \Upsilon[0], \cdots, \Upsilon[k-1]\big)$, we have $\mathbb{E}\big(P_1[k]\big) \succeq \mathbb{E}\big(P_2[k]\big) \succeq \mathbb{E}\big(P_3[k]\big)$. □

The next theorem shows that the stability regions of the full-knowledge and no-knowledge cases are the same.

Theorem 5.1. *The dynamical system of (5.19) is stable if and only if* $\mu_{ave}\big(\Upsilon_{\text{T}}\big) < \rho_{\max}^{-2}(A)$ *or equivalently* $\Upsilon_{\text{T}} < \Upsilon_{\text{T,c}}$, *where* $\rho_{\max}(A)$ *and* $\Upsilon_{\text{T,c}}$ *are as defined in Lemma 5.3.*

Proof. From Lemma 5.5, we know that the average estimation error variance of the full-knowledge case is upper bounded by that of the no-knowledge case and lower bounded by that of the noise-free case. Therefore, from Lemma 5.3 and (5.14) for the stability of the noise-free case, we can see that $\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big) < \rho_{\max}^{-2}(A)$ is

the necessary and sufficient condition for ensuring the stability of the case of full-knowledge. □

It is also possible to deduce the same final result by directly relating the average of the dynamics of the full-knowledge case to those of noise-free and no-knowledge cases. We show this alternative approach in the appendix.

5.3.4 Estimator has Partial Knowledge on the Reception Quality

Consider (5.10) with $0 \prec \tilde{R}[k] \prec R[k]$:

$$\begin{aligned} P[k+1] &= AP[k]A^T + Q - \big(1-\mu[k]\big)AP[k]C^T\big(CP[k]C^T + \tilde{R}[k]\big)^{-1} \\ &\quad \times \big(CP[k]C^T + 2\tilde{R}[k] - R[k]\big)\big(CP[k]C^T + \tilde{R}[k]\big)^{-1}CP[k]A^T \\ &\triangleq \Phi_{\text{par}}\big(P[k], \mu[k], R[k], \tilde{R}[k]\big). \end{aligned} \tag{5.27}$$

To facilitate mathematical derivations of the partial-knowledge case, we assume diagonal $\tilde{R}$ and R in this part: $\tilde{R}[k] = \tilde{\sigma}_n^2\big(\Upsilon[k]\big)I$ and $R[k] = \sigma_n^2\big(\Upsilon[k]\big)I$.

Lemma 5.6. *Assume* $\Pi \succ 0$ *and* $0 \le \xi_2 \le \xi_1 \le \xi_{\max}$. *Then* $\big(\Pi + \xi_1 I\big)^{-1}\big(\Pi + 2\xi_1 I - \xi_{\max} I\big)\big(\Pi + \xi_1 I\big)^{-1} \succeq \big(\Pi + \xi_2 I\big)^{-1}\big(\Pi + 2\xi_2 I - \xi_{\max} I\big)\big(\Pi + \xi_2 I\big)^{-1}$.

Proof. Define $\xi_{\text{diff}} \triangleq \xi_1 - \xi_2 \ge 0$ and $\tilde{\Pi} \triangleq \Pi + \xi_2 I$. Since $\xi_2 - \xi_{\max} \le 0$ and $\xi_1 + \xi_2 - 2\xi_{\max} \le 0$, we have

$$\begin{aligned} \tilde{\Pi}^3 + (\xi_2 - \xi_{\max} + 2\xi_{\text{diff}})\tilde{\Pi}^2 &\succeq \tilde{\Pi}^3 + (\xi_2 - \xi_{\max} + 2\xi_{\text{diff}})\tilde{\Pi}^2 \\ &\quad + \xi_{\text{diff}}^2(\xi_2 - \xi_{\max})I + \xi_{\text{diff}}\underbrace{(2\xi_2 - 2\xi_{\max} + \xi_{\text{diff}})}_{\xi_1+\xi_2-2\xi_{\max}}\tilde{\Pi} \\ &= \big(\tilde{\Pi}^2 + \xi_{\text{diff}}\tilde{\Pi}\big)\big(\tilde{\Pi}^{-1} + (\xi_2 - \xi_{\max})\tilde{\Pi}^{-2}\big)\big(\tilde{\Pi}^2 + \xi_{\text{diff}}\tilde{\Pi}\big) \end{aligned} \tag{5.28}$$

or equivalently $\tilde{\Pi}\big(\tilde{\Pi} + (\xi_2 + 2\xi_{\text{diff}} - \xi_{\max})I\big)\tilde{\Pi} \succeq \tilde{\Pi}\big(\tilde{\Pi} + \xi_{\text{diff}} I\big)\tilde{\Pi}^{-1}\big(\tilde{\Pi} + (\xi_2 - \xi_{\max})I\big)\tilde{\Pi}^{-1}\big(\tilde{\Pi} + \xi_{\text{diff}} I\big)\tilde{\Pi}$. Let $\Psi = \big(\Pi + \xi_1 I\big)^{-1}\big(\Pi + \xi_2 I\big)^{-1}$. Then, using Lemma 5.1, we have,

$$\begin{aligned} &\Psi\tilde{\Pi}\big(\tilde{\Pi} + (\xi_2 + 2\xi_{\text{diff}} - \xi_{\max})I\big)\tilde{\Pi}\Psi^T \\ &\quad \succeq \Psi\tilde{\Pi}\big(\tilde{\Pi} + \xi_{\text{diff}} I\big)\tilde{\Pi}^{-1}\big(\tilde{\Pi} + (\xi_2 - \xi_{\max})I\big)\tilde{\Pi}^{-1}\big(\tilde{\Pi} + \xi_{\text{diff}} I\big)\tilde{\Pi}\Psi^T, \end{aligned} \tag{5.29}$$

which can be easily verified to be the same as the desired inequality. □

Lemma 5.7. *Consider* $0 \le \mu \le 1$, $\tilde{R} = \tilde{\sigma}_n^2 I$, $R = \sigma_n^2 I$ *and* $P \succ 0$. *Then,* $\Phi_{par}\big(P, \mu, R, \tilde{R}\big)$ *is a non-increasing function of* $\tilde{\sigma}_n^2 \in [0, \sigma_n^2]$.

Proof. For $\tilde{R} = \tilde{\sigma}_n^2 I$ and $R = \sigma_n^2 I$, $\Phi_{\text{par}}\big(P, \mu, R, \tilde{R}\big)$ can be written as

$$\Phi_{\text{par}}\big(P, \mu, R, \tilde{R}\big) = APA^T + Q - (1-\mu)APC^T\big(CPC^T + \tilde{\sigma}_n^2 I\big)^{-1} \times \big(CPC^T + 2\tilde{\sigma}_n^2 I - \sigma_n^2 I\big)\big(CPC^T + \tilde{\sigma}_n^2 I\big)^{-1} CPA^T. \quad (5.30)$$

Then Lemma 5.6 implies that $\big(CPC^T + \tilde{\sigma}_n^2 I\big)^{-1}\big(CPC^T + 2\tilde{\sigma}_n^2 I - \sigma_n^2 I\big)\big(CPC^T + \tilde{\sigma}_n^2 I\big)^{-1}$ is a non-decreasing function of $\tilde{\sigma}_n^2$ and as a result $\Phi_{\text{par}}\big(P, \mu, R, \tilde{R}\big)$ is a non-increasing function of $\tilde{\sigma}_n^2$. □

Lemma 5.7 indicates that as the partial knowledge increases $\left(\frac{\tilde{\sigma}_n^2}{\sigma_n^2} \to 1\right)$, the error covariance $P[k+1]$ decreases, given the same $P[k]$. Also one can easily confirm, using Lemma 5.7, that for diagonal R and $\tilde{R}$:

$$\Phi_{\text{no}}\big(P, \mu, R\big) \succeq \Phi_{\text{par}}\big(P, \mu, R, \tilde{R}\big) \succeq \Phi_{\text{full}}\big(P, \mu, R\big) \succeq \Phi_{\text{perf}}\big(P, \mu\big). \quad (5.31)$$

Since characterizing the necessary and sufficient stability condition for the partial case is considerably challenging, we next make an approximation by assuming that $\tilde{R}$ is small. This approximation will then help us establish that $\Phi_{\text{par}}\big(P, \mu, R, \tilde{R}\big)$ is non-decreasing as a function of $P \succ 0$, which will then help us characterize the stability region.

Lemma 5.8. *Let $\rho_{min}(\Pi)$ represent the minimum eigenvalue of symmetric $\Pi \succ 0$. If $\Pi \succ 0$, $\xi_2 \geq 0$ and $0 \leq \xi_1 \ll \rho_{min}(\Pi)$, then we have the following approximation by only keeping the first-order terms: $\big(\Pi + \xi_1 I\big)^{-1}\big(\Pi + 2\xi_1 I - \xi_2 I\big)\big(\Pi + \xi_1 I\big)^{-1} \approx \Pi^{-1}\Big(\Pi - \xi_2 I + 2\xi_1\xi_2\Pi^{-1}\Big)\Pi^{-1}$.*

Proof. We have $\big(\Pi + \xi_1 I\big)\big(\Pi^{-1} - \xi_1\Pi^{-2}\big) = I - \xi_1^2\Pi^{-2}$. Therefore, for $0 \leq \xi_1 \ll \rho_{\min}(\Pi)$, we have $\|\xi_1^2\Pi^{-2}\| \ll 1$, which results in the following approximations by only keeping the first-order terms as a function of ξ_1:

$$\big(\Pi + \xi_1 I\big)^{-1} \approx \Pi^{-1} - \xi_1\Pi^{-2} \text{ and } \big(\Pi + \xi_1 I\big)^{-2} \approx \Pi^{-2} - 2\xi_1\Pi^{-3}. \quad (5.32)$$

Then, we have $\big(\Pi + \xi_1 I\big)^{-1}\big(\Pi + 2\xi_1 I - \xi_2 I\big)\big(\Pi + \xi_1 I\big)^{-1} = \big(\Pi + \xi_1 I\big)^{-1} - (\xi_2 - \xi_1)\big(\Pi + \xi_1 I\big)^{-2}$. But

$$\begin{aligned}\big(\Pi + \xi_1 I\big)^{-1} - (\xi_2 - \xi_1)\big(\Pi + \xi_1 I\big)^{-2} &\approx \Pi^{-1} - \xi_1\Pi^{-2} - (\xi_2 - \xi_1)\big(\Pi^{-2} - 2\xi_1\Pi^{-3}\big) \\ &= \Pi^{-1}\Big(\underbrace{I - 2\xi_1^2\Pi^{-2}}_{\approx I} - \xi_2\Pi^{-1} + 2\xi_1\xi_2\Pi^{-2}\Big) \\ &\approx \Pi^{-1} - \xi_2\Pi^{-2} + 2\xi_1\xi_2\Pi^{-3} \\ &= \Pi^{-1}\Big(\Pi - \xi_2 I + 2\xi_1\xi_2\Pi^{-1}\Big)\Pi^{-1}.\end{aligned} \quad (5.33)$$

□

Lemma 5.9. *Let $\tilde{R} = \tilde{\sigma}_n^2 I$, $R = \sigma_n^2 I$ and assume that $\tilde{\sigma}_n^2 \ll \rho_{min}(CPC^T)$. Then, the first-order approximation of $\Phi_{par}(P, \mu, R, \tilde{R})$ is a non-decreasing function of $P \succ 0$.*

Proof. By using Lemma 5.8, we can approximate $\Phi_{\text{par}}(P, \mu, R, \tilde{R})$ as follows

$$\begin{aligned}\Phi_{\text{par}}(P, \mu, R, \tilde{R}) \approx \mu APA^T + Q + (1-\mu)\sigma_n^2 A\big(C^T C\big)^{-1} A^T \\ - 2(1-\mu)\tilde{\sigma}_n^2 \sigma_n^2 A\big(C^T C\big)^{-1} P^{-1}\big(C^T C\big)^{-1} A^T, \end{aligned} \tag{5.34}$$

which is a non-decreasing function of P. □

Lemma 5.10. *Let $\tilde{R}[k] = \tilde{\sigma}_n^2(\Upsilon[k])I$, $R[k] = \sigma_n^2(\Upsilon[k])I$ and $\tilde{\sigma}_n^2(\Upsilon[k]) \ll \rho_{min}(CP[k]C^T)$ for all k. Let $P_1[k+1] = \Phi_{no}(P_1[k], \mu[k], R[k])$, $P_2[k+1] = \Phi_{par}(P_2[k], \mu[k], R[k], \tilde{R}[k])$, $P_3[k+1] = \Phi_{full}(P_3[k], \mu[k], R[k])$ and $P_4[k+1] = \Phi_{perf}(P_4[k], \mu[k])$. Then starting from the same initial P_0, we have*

$$\mathbb{E}\big(P_1[k]\big) \succeq \mathbb{E}\big(P_2[k]\big) \succeq \mathbb{E}\big(P_3[k]\big) \succeq \mathbb{E}\big(P_4[k]\big), \qquad k \geq 0. \tag{5.35}$$

Proof. From (5.31), we have the following for diagonal $R[k]$ and $\tilde{R}[k]$ and any $P[k] \succ 0$:

$$\begin{aligned}\Phi_{\text{no}}\big(P[k], \mu[k], R[k]\big) \succeq \Phi_{\text{par}}\big(P[k], \mu[k], R[k], \tilde{R}[k]\big) \\ \succeq \Phi_{\text{full}}\big(P[k], \mu[k], R[k]\big) \succeq \Phi_{\text{perf}}\big(P[k], \mu[k]\big). \end{aligned} \tag{5.36}$$

For $\tilde{\sigma}_n^2(\Upsilon[k]) \ll \rho_{\min}(CP[k]C^T)$, Φ_{par} is a non-decreasing function of P according to Lemma 5.9. Therefore, we have the following for $P_1[k] \succeq P_2[k] \succeq P_3[k] \succeq P_4[k] \succ 0$,

$$\begin{aligned}&\Phi_{\text{no}}\big(P_1[k], \mu[k], R[k]\big) \succeq \Phi_{\text{par}}\big(P_1[k], \mu[k], R[k], \tilde{R}[k]\big) \\ &\quad\succeq \Phi_{\text{par}}\big(P_2[k], \mu[k], R[k], \tilde{R}[k]\big) \succeq \Phi_{\text{full}}\big(P_2[k], \mu[k], R[k]\big) \\ &\quad\succeq \Phi_{\text{full}}\big(P_3[k], \mu[k], R[k]\big) \succeq \Phi_{\text{perf}}\big(P_3[k], \mu[k]\big) \succeq \Phi_{\text{perf}}\big(P_4[k], \mu[k]\big). \end{aligned} \tag{5.37}$$

Then, (5.35) can be easily proved similar to Lemma 5.5. □

The next theorem shows that, under the assumptions we made, the stability region of the partial-knowledge case is the same as the full-knowledge and no-knowledge cases.

Theorem 5.2. *Assume $\tilde{R}[k] = \tilde{\sigma}_n^2(\Upsilon[k])I$, $R[k] = \sigma_n^2(\Upsilon[k])I$ and for all k we have $\tilde{\sigma}_n^2(\Upsilon[k]) \ll \rho_{min}(CP[k]C^T)$. Then the dynamical system of (5.27) is stable if and only if $\mu_{ave}(\Upsilon_{\text{T}}) < \rho_{\max}^{-2}(A)$ or equivalently $\Upsilon_{\text{T}} < \Upsilon_{\text{T,c}}$, where $\rho_{\max}(A)$ and $\Upsilon_{\text{T,c}}$ are as defined in Lemma 5.3.*

Proof. By utilizing Lemma 5.10, this can be proved similar to Theorem 5.1. □

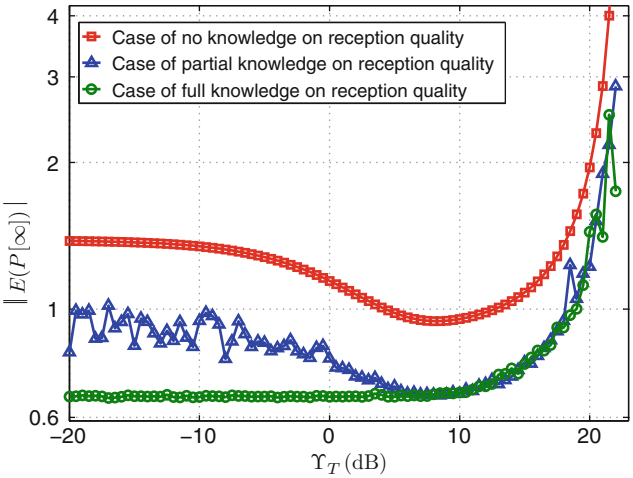


Fig. 5.3 Impact of the knowledge available on the link qualities on the performance and stability of Kalman filtering over fading channels

Figure 5.3 shows the norm of the asymptotic average estimation error variance for the three cases of full-knowledge, partial-knowledge and no-knowledge, and as a function of Υ_{T}. For this case, SNR is exponentially distributed with the average of 30dB, $\sigma_n^2(\Upsilon) = 0.50 + 533.3 \times \Omega\left(\sqrt{\Upsilon}\right)$, where $\Omega(d) = \frac{1}{\sqrt{2\pi}} \int_d^\infty \mathrm{e}^{-t^2/2} \mathrm{d}t$ for an arbitrary d and $\tilde{\sigma}_n^2(\Upsilon) = 0.5$. This is the variance of the communication noise for a binary modulation system that utilizes gray coding [31]. The following parameters are chosen for this example: $A = \begin{pmatrix} 2 & 0.3 & 0.45 \\ 0.4 & 0.2 & 0.5 \\ 1.5 & 0.6 & 0.34 \end{pmatrix}$, $Q = 0.001I$ and $C = 2I$, which results in $\Upsilon_{\mathrm{T,c}} = 22.53$ dB. As can be seen, all the three curves have the same critical stability point, indicated by $\Upsilon_{\mathrm{T,c}}$. As expected, the more is known about the link qualities, the better the performance is. The figure also shows that not all $\Upsilon_{\mathrm{T}} < \Upsilon_{\mathrm{T,c}}$ result in the same performance and that there is an optimum threshold that minimizes the average estimation error variance. Υ_{T} impacts both the packet drop and the amount of communication noise that enters the estimation process. Therefore, optimizing it can properly control the overall performance, as we shall show in the next section.

5.4 Optimization of Packet Drop in the Physical Layer

In this part, we consider the optimization of packet drop in the physical layer. In (5.10), the threshold Υ_{T} impacts both the amount of information loss ($\mu[k]$) and the quality of those packets that are kept ($R[k]$). If this threshold is chosen very high,

the information loss rate could become considerably high, depending on the quality of the link. However, once a packet is kept, the communication error that enters the estimation process will be small. On the other hand, for low thresholds, information loss rate will be lower at the cost of more noisy receptions. In this section, we are interested in characterizing the optimum threshold for the two cases that the estimator has "full knowledge" and "no knowledge" on the communication quality. Since the optimization of the threshold occurs in the physical layer, in this section we assume that the physical layer has full knowledge of SNR and the corresponding communication error variance function. For instance, for the case where the estimator has no knowledge on the reception quality, the physical layer would still have link quality information, based on which it will optimize the threshold. However, the information on the link quality is not transferred to the application layer, which is in charge of estimation. In other words, the physical layer controls the impact of the communication links on the estimation process. For the case where the estimator has full knowledge on the reception quality, on the other hand, the physical layer constantly passes on the information on the link qualities to the estimator. Therefore, both the physical and application layers utilize this knowledge. The framework of this section can then be extended to other scenarios, for instance to the case where the physical layer only has partial knowledge on the link qualities (such as an upper bound). While in the previous section, we took the threshold to be constant during the estimation process, in this section we consider both adaptive (time-varying) and non-adaptive thresholding. In the adaptive case, we show how the threshold can be optimized at every time-step to minimize the average estimation error variance of the next time. For the non-adaptive thresholding, we find the optimum threshold that minimizes the asymptotic average estimation error variance. There are interesting trade-offs between the two approaches, as discussed in this section.

5.4.1 Estimator has No Knowledge on the Reception Quality

In this part, we consider the case where the estimator has no information on the reception quality. We consider both scenarios with fixed and adaptive thresholding. From (5.15), we have $P[k+1] = \mu[k]AP[k]A^T + Q + (1-\mu[k])AC^{-1}R[k]\big(C^T\big)^{-1}A^T$. Both $\mu[k]$ and $R[k]$ are functions of $\Upsilon_{\mathrm{T}}[k]$, which is taken to be time-varying to allow for adaptive thresholding. Since the stability analysis of the previous section was carried out assuming that Υ_{T} is time-invariant, we need to first establish the stability of the time-varying case. For the derivations of this section, we take $R[k]$ to be diagonal: $R[k] = \sigma_n^2(\Upsilon[k])I$. We have

$$P[k+1] = \mu[k]AP[k]A^T + Q + \underbrace{(1-\mu[k])\sigma_n^2\big(\Upsilon[k]\big)}_{\beta[k]} A\big(C^TC\big)^{-1}A^T, \quad (5.38)$$

with the following solution as a function of the initial variance P_0:

$$P[k] = \left(\prod_{i=0}^{k-1} \mu[i]\right) A^k P_0 \left(A^T\right)^k + \sum_{i=0}^{k-1} \left(\prod_{j=i+1}^{k-1} \mu[j]\right) A^{k-i-1} \left(Q + \beta[i] A \left(C^T C\right)^{-1} A^T\right) \left(A^T\right)^{k-i-1}. \tag{5.39}$$

At the kth time-step, $\mu[k]$ and $\beta[k]$ are functions of the instantaneous received SNR, $\Upsilon[k]$. Since $\Upsilon[k_1]$ and $\Upsilon[k_2]$, for $k_1 \neq k_2$, are independent, $\mu[k_1]$ and $\mu[k_2]$ as well as $\mu[k_1]$ and $\beta[k_2]$ are also independent for $k_1 \neq k_2$. Therefore,

$$\mathbb{E}\left(P[k]\right) = \left(\prod_{i=0}^{k-1} \mu_{\text{ave}}\left(\Upsilon_{\text{T}}[i]\right)\right) A^k P_0 \left(A^T\right)^k + \sum_{i=0}^{k-1} \left(\prod_{j=i+1}^{k-1} \mu_{\text{ave}}\left(\Upsilon_{\text{T}}[j]\right)\right) \times A^{k-i-1} \left(Q + \sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[i]\right) A \left(C^T C\right)^{-1} A^T\right) \left(A^T\right)^{k-i-1}, \tag{5.40}$$

where $\mu_{\text{ave}}\left(\Upsilon_{\text{T}}[k]\right)$ and $\sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[k]\right)$ represent time-varying average probability of packet loss (spatial averaging over fading) and average noise variance that entered the estimation process, respectively:

$$\mu_{\text{ave}}\left(\Upsilon_{\text{T}}[k]\right) = \mathbb{E}\left(\mu[k]\right) = \int_0^{\Upsilon_{\text{T}}[k]} \chi(\Upsilon) \mathrm{d}\Upsilon \tag{5.41}$$

and

$$\sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[k]\right) = \mathbb{E}\left(\beta[k]\right) = \int_{\Upsilon_{\text{T}}[k]}^{\infty} \sigma_n^2(\Upsilon) \chi(\Upsilon) \mathrm{d}\Upsilon, \tag{5.42}$$

where χ represents the probability density function of Υ. Then (5.40) is the solution to the following average dynamical system:

$$\mathbb{E}\left(P[k+1]\right) = \mu_{\text{ave}}\left(\Upsilon_{\text{T}}[k]\right) A \mathbb{E}\left(P[k]\right) A^T + Q + \sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[k]\right) A (C^T C)^{-1} A^T. \tag{5.43}$$

It can be seen that for a time-varying $\Upsilon_{\text{T}}[k]$, we have a time-varying average dynamical system, for which we have to ensure stability.

Lemma 5.11. *Let $\rho_{\max}(A)$ denote the spectral radius of matrix A. If for an arbitrary small $\varepsilon > 0$, $\mu_{ave}(\Upsilon_{\text{T}}[k]) \leq \underbrace{(1-\varepsilon)\rho^{-2}_{\max}(A)}_{\mu_{ave,max}}$, then the average dynamical system of (5.43) will be stable.*

Proof. Let $\sigma^2_{n,\text{ave,max}} = \max_{\Upsilon_{\text{T}}[k]} \sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[k]\right) = \int_0^{\infty} \sigma_n^2(\Upsilon) \chi(\Upsilon) \mathrm{d}\Upsilon$. Then,

$$\mathbb{E}\left(P[k+1]\right) \preceq \mu_{\text{max,ave}} A \mathbb{E}\left(P[k]\right) A^T + Q + \sigma^2_{n,\text{ave,max}} A \left(C^T C\right)^{-1} A^T. \tag{5.44}$$

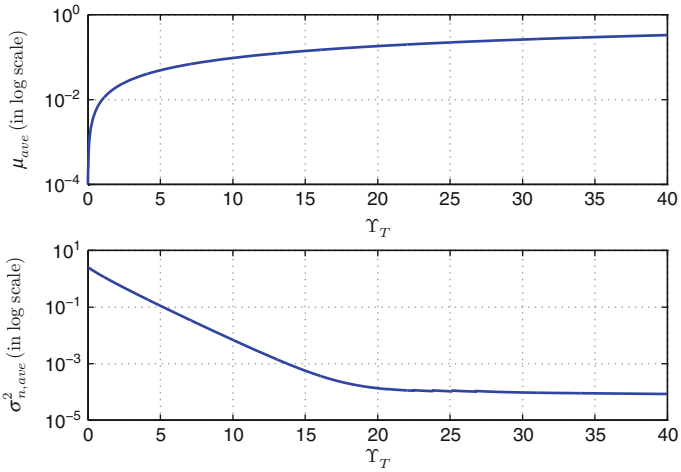


Fig. 5.4 Examples of the average probability of packet drop and average overall communication noise variance as a function of Υ_{T}. As the threshold increases, μ_{ave} increases while communication noise variance $\sigma^2_{n,\mathrm{ave}}$ decreases

Starting from the same initial condition, $\mathbb{E}\big(P[k]\big)$ is upper-bounded by the solution of $\mathbb{E}\big(P_m[k+1]\big) = \mu_{\mathrm{ave,max}} A\mathbb{E}\big(P_m[k]\big)A^T + Q + \sigma^2_{n,\mathrm{ave,max}} A\big(C^T C\big)^{-1}A^T$, which is stable [27]. This shows that $\mu_{\mathrm{ave}}\big(\Upsilon_{\mathrm{T}}[k]\big) \leq \mu_{\mathrm{ave,max}}$ is a sufficient condition for ensuring the stability of (5.43). □

Note that $\mu_{\mathrm{ave}}(\Upsilon_{\mathrm{T}}[k]) \leq \mu_{\mathrm{ave,max}}$ is equivalent to $\Upsilon_{\mathrm{T}}[k] \leq \Upsilon^{\varepsilon}_{\mathrm{T,c}}$, where $\Upsilon^{\varepsilon}_{\mathrm{T,c}}$ is such that $\int_0^{\Upsilon^{\varepsilon}_{\mathrm{T,c}}} \chi(\Upsilon)\mathrm{d}\Upsilon = (1-\varepsilon)\rho^{-2}_{\max}(A)$ for an unstable A. Next, we develop a foundation for the optimization of the threshold. Figure 5.4 shows examples of μ_{ave} and $\sigma^2_{n,\mathrm{ave}}$ as a function of the threshold. It can be seen that the optimum threshold should properly control and balance the amount of information loss and communication error that enters the estimation process, as we shall characterize in this section.

5.4.1.1 Adaptive Thresholding

In this part, we consider the case where $\Upsilon_{\mathrm{T}}[k]$ is time-varying and is optimized at every time-step based on the current estimation error variance. Consider (5.38). We have the following conditional average for the estimation error variance at time $k+1$:

$$\mathbb{E}\Big(P[k+1]\,\Big|\,P[k]\Big) = AP[k]A^T + Q + \int_{\Upsilon_{\mathrm{T}}[k]}^{\infty}\Big[\sigma_n^2(\Upsilon)A\big(C^TC\big)^{-1}A^T - AP[k]A^T\Big]\chi(\Upsilon)\mathrm{d}\Upsilon. \tag{5.45}$$

Then at the kth time-step, our goal is to find $\Upsilon_{\mathrm{T}}[k]$ such that $\mathbb{E}\Big(P[k+1] \;\Big|\; P[k]\Big)$ is minimized.

Minimization of Trace:

Consider minimizing $\mathrm{tr}\Big(\mathbb{E}\Big(P[k+1] \;\Big|\; P[k]\Big)\Big)$, where tr(.) denotes the trace of the argument. We have

$$\mathrm{tr}\Big(\mathbb{E}\Big(P[k+1]\Big|P[k]\Big)\Big) = \mathrm{tr}\Big(AP[k]A^T + Q\Big) + \int_{\Upsilon_{\mathrm{T}}[k]}^{\infty} \Big[\sigma_n^2(\Upsilon)\mathrm{tr}\Big(A\big(C^TC\big)^{-1}A^T\Big) - \mathrm{tr}\Big(AP[k]A^T\Big)\Big]\chi(\Upsilon)\mathrm{d}\Upsilon, \tag{5.46}$$

and the following optimization problem:

$$\begin{aligned}&\Upsilon^*_{T,\mathrm{trace,adp}}[k]\\ &\quad= \arg\min_{\Upsilon_{\mathrm{T}}[k]} \int_{\Upsilon_{\mathrm{T}}[k]}^{\infty} \underbrace{\Big[\sigma_n^2(\Upsilon)\mathrm{tr}\Big(A\big(C^TC\big)^{-1}A^T\Big) - \mathrm{tr}\Big(AP[k]A^T\Big)\Big]}_{\Theta\left(\Upsilon,P[k]\right)}\chi(\Upsilon)\mathrm{d}\Upsilon,\\ &\text{subject to } 0 \le \Upsilon_{\mathrm{T}}[k] \le \Upsilon^{\varepsilon}_{\mathrm{T,c}}\end{aligned} \tag{5.47}$$

with $\Upsilon^*_{T,\mathrm{trace,adp}}[k]$ denoting the optimum threshold at time-step k, when minimizing the trace in the adaptive case.

Note that $\sigma_n^2\big(\Upsilon\big)$ and, as a result $\Theta\big(\Upsilon, P[k]\big)$, are a non-increasing function of Υ. Furthermore, $\chi(\Upsilon)$ is always non-negative, which results in $\int_{\Upsilon_{\mathrm{T}}[k]}^{\infty}\Theta\big(\Upsilon, P[k]\big)\chi(\Upsilon)\mathrm{d}\Upsilon$ minimized when $\Theta\big(\Upsilon, P[k]\big)$ changes sign. Thus, three different cases can happen:

1. $\lim_{\Upsilon\to 0}\Theta\big(\Upsilon, P[k]\big) \ge 0$ and $\lim_{\Upsilon\to\infty}\Theta\big(\Upsilon, P[k]\big) \ge 0$: In this case, $\Theta\big(\Upsilon, P[k]\big)$ is non-negative as a function of Υ. Therefore, $\Upsilon^*_{T,\mathrm{trace,adp}}[k] = \Upsilon^{\varepsilon}_{\mathrm{T,c}}$. One possible scenario that results in non-negative $\Theta\big(\Upsilon, P[k]\big)$ is when $\mathrm{tr}\Big(AP[k]A^T\Big)$ is considerably low.
2. $\lim_{\Upsilon\to 0}\Theta\big(\Upsilon, P[k]\big) \le 0$ and $\lim_{\Upsilon\to\infty}\Theta\big(\Upsilon, P[k]\big) < 0$: In this case $\Theta\big(\Upsilon, P[k]\big)$ is negative as a function of Υ. Therefore, $\Upsilon^*_{T,\mathrm{trace,adp}}[k] = 0$, i.e. the next packet is kept independent of its quality. For instance, if $\mathrm{tr}\Big(AP[k]A^T\Big)$ is considerably high, it can result in a negative $\Theta\big(\Upsilon, P[k]\big)$.
3. $\lim_{\Upsilon\to 0}\Theta\big(\Upsilon, P[k]\big) > 0$ and $\lim_{\Upsilon\to\infty}\Theta\big(\Upsilon, P[k]\big) < 0$: In this case, the optimum threshold is $\Upsilon^*_{T,\mathrm{trace,adp}}[k] = \min\big\{\Upsilon^{\varepsilon}_{\mathrm{T,c}}, \gamma^*_{T,\mathrm{trace,adp}}\big\}$, where $\gamma^*_{T,\mathrm{trace,adp}}$ is the unique solution to the following equation:

$$\Theta\big(\gamma^*_{T,\mathrm{trace,adp}}, P[k]\big) = 0 \;\Rightarrow\; \sigma_n^2\big(\gamma^*_{T,\mathrm{trace,adp}}\big)\,\mathrm{tr}\Big(A\big(C^TC\big)^{-1}A^T\Big) = \mathrm{tr}\Big(AP[k]A^T\Big). \tag{5.48}$$

Minimization of Determinant:

Consider (5.45), which can be written as:

$$\mathbb{E}\Big(P[k+1]\,\Big|\,P[k]\Big) = \mu_{\text{ave}}(\Upsilon_{\text{T}}[k])AP[k]A^T + \sigma^2_{n,\text{ave}}(\Upsilon_{\text{T}}[k])A\big(C^TC\big)^{-1}A^T + Q. \tag{5.49}$$

Lemma 5.12. *Consider the case where Q is negligible in (5.49). The optimum adaptive threshold, $\Upsilon^*_{T,det,adp}[k]$, that minimizes the determinant of $\mathbb{E}\Big(P[k+1]\,\Big|\,P[k]\Big)$ is then the solution to the following optimization problem:*

$$\Upsilon^*_{T,det,adp}[k] = argmin_{\Upsilon_{\text{T}}[k]} \prod_i \Big(\mu_{ave}(\Upsilon_{\text{T}}[k]) + \zeta_i \sigma^2_{n,ave}(\Upsilon_{\text{T}}[k])\Big)$$

$$\text{subject to } 0 \le \Upsilon_{\text{T}}[k] \le \Upsilon^{\varepsilon}_{\text{T,c}}, \tag{5.50}$$

where $\Xi[k] = P^{-1/2}[k](C^TC)^{-1}P^{-1/2}[k]$ and ζ_is for $1 \le i \le N$ are the eigenvalues of $\Xi[k]$.

Proof. Since $P[k]$ is positive definite, there exists a unique positive definite $P^{1/2}[k]$ such that $P[k] = P^{1/2}[k]P^{1/2}[k]$. We then have the following for negligible Q:

$$\mathbb{E}\Big(P[k+1]\,\Big|\,P[k]\Big) = AP^{1/2}[k]\Big(\mu_{\text{ave}}\big(\Upsilon_{\text{T}}[k]\big)I + \sigma^2_{n,\text{ave}}\big(\Upsilon_{\text{T}}[k]\big)\Xi[k]\Big)P^{1/2}[k]A^T, \tag{5.51}$$

where $\Xi[k]$ is as denoted earlier in the lemma with the following diagonalization $\Xi[k] = F\zeta_{\text{diag}}F^T$, with $FF^T = I$ and $\zeta_{\text{diag}} = \text{diag}\{\zeta_1, \zeta_2, \ldots, \zeta_N\}$. Then,

$$\begin{aligned}\det\Big(\mathbb{E}\Big(P[k+1]\,\Big|\,P[k]\Big)\Big) &= \det\big(AP[k]A^T\big)\det\Big(\mu_{\text{ave}}\big(\Upsilon_{\text{T}}[k]\big)I + \sigma^2_{n,\text{ave}}\big(\Upsilon_{\text{T}}[k]\big)\zeta_{\text{diag}}\Big)\\ &= \det\big(AP[k]A^T\big)\prod_i\Big(\mu_{\text{ave}}\big(\Upsilon_{\text{T}}[k]\big) + \zeta_i\sigma^2_{n,\text{ave}}\big(\Upsilon_{\text{T}}[k]\big)\Big).\end{aligned} \tag{5.52}$$

Therefore, $\Upsilon^*_{T,\text{det,adp}}[k]$ is the solution to the optimization problem of (5.50). □

We know that the optimum threshold is either at the boundaries of the feasible set ($\Upsilon_{\text{T}}[k] = 0$ and $\Upsilon_{\text{T}}[k] = \Upsilon^{\varepsilon}_{\text{T,c}}$) or is where the derivative is zero: $\frac{\partial\det\left(\mathbb{E}\left(P[k+1]\,\middle|\,P[k]\right)\right)}{\partial\Upsilon_{\text{T}}[k]} = 0 \Rightarrow \sum_i \frac{\chi(\Upsilon_{\text{T}}[k])\left(1-\zeta_i\sigma^2_n\left(\Upsilon_{\text{T}}[k]\right)\right)}{\mu_{\text{ave}}\left(\Upsilon_{\text{T}}[k]\right)+\zeta_i\sigma^2_{n,\text{ave}}\left(\Upsilon_{\text{T}}[k]\right)} = 0$. Among these points, the one that minimizes (5.52) is the optimum threshold. Therefore in practice, based on the given parameters such as the shape of the overall communication

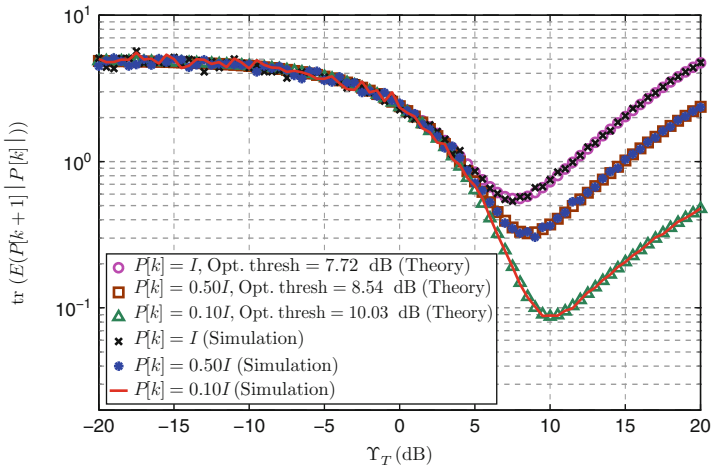


Fig. 5.5 Optimization of packet drop through one-step adaptive thresholding for three different $P[k]$s and average SNR of 20 dB. This is for the case where knowledge of reception quality is not available at the estimator

noise variance (σ_n^2) and the pdf of SNR, the optimum threshold can be identified for this case. Figure 5.5 shows the performance of adaptive thresholding at time $k+1$ given three different $P[k]$s. For this example, Υ is taken to have an exponential distribution with the average of 20 dB and the overall communication noise variance is taken as follows: $\sigma_n^2(\Upsilon) = 1.27 \times 10^{-4} + 533.3 \times \Omega(\sqrt{\Upsilon})$, where $\Omega(d) = \frac{1}{\sqrt{2\pi}} \int_d^\infty \mathrm{e}^{-t^2/2} \mathrm{d}t$ for an arbitrary d. This is the variance of the communication noise for a binary modulation system that utilizes gray coding [31] and corresponds to 10 bits per sample and quantization step size of 0.0391. The rest of the parameters are the same as for Fig. 5.3. The optimum thresholds can be seen from the figure. Furthermore, the theoretical curves are confirmed with simulation results. As can be seen, the higher $\mathrm{tr}(AP[k]A^T)$ is (which corresponds to a higher $P[k]$ for this example), the lower the threshold will be. This can also be seen from the solution of the optimization problem of (5.4.1.1). For instance, in (3), the higher $\mathrm{tr}(AP[k]A^T)$ is, the lower the optimum threshold will be. This also makes intuitive sense as for higher $P[k]$s, the overall noise variance is more tolerable and the threshold should be lowered to avoid more information loss.

5.4.1.2 Non-Adaptive Thresholding Through Minimization of $\mathbb{E}\big(P[\infty]\big)$

In this part, we consider the case where a non-adaptive threshold is used throughout the estimation process. We show how to optimize this threshold such that $\mathbb{E}\big(P[\infty]\big)$ is minimized [22]. We consider minimization of both norm and determinant of $\mathbb{E}\big(P[\infty]\big)$. The derivations can be similarly carried out to minimize the trace of

$\mathbb{E}\big(P[\infty]\big)$. Consider (5.43). In this case, Υ_{T} will be time invariant. Then, using Lemma 5.2, we will have the following expression for $\mathbb{E}\big(P[\infty]\big)$:

$$\mathbb{E}\big(P[\infty]\big) = \sum_{i=0}^{\infty} \mu_{\mathrm{ave}}^{i}(\Upsilon_{\mathrm{T}}) A^{i} Q \big(A^{T}\big)^{i} + \sigma_{n,\mathrm{ave}}^{2}(\Upsilon_{\mathrm{T}}) \sum_{i=0}^{\infty} \mu_{\mathrm{ave}}^{i}(\Upsilon_{\mathrm{T}}) A^{i+1} (C^{T}C)^{-1} \big(A^{T}\big)^{i+1}. \quad (5.53)$$

Let $\Upsilon^{*}_{T,\mathrm{norm,fixed}}$ represent the optimum way of dropping packets, which will minimize the spectral norm of the asymptotic average estimation error variance for a non-adaptive threshold: $\Upsilon^{*}_{T,\mathrm{norm,fixed}} = \operatorname{argmin} \big\|\mathbb{E}\big(P[\infty]\big)\big\|$, for $\Upsilon_{\mathrm{T}} < \Upsilon_{\mathrm{T,c}}$. Also, let $\Upsilon^{*}_{T,\mathrm{det,fixed}}$ represent the optimum way of dropping packets, which will minimize the determinant of the asymptotic average estimation error variance: $\Upsilon^{*}_{T,\mathrm{det,fixed}} = \operatorname{argmin} \det\Big(\mathbb{E}\big(P[\infty]\big)\Big)$, for $\Upsilon_{\mathrm{T}} < \Upsilon_{\mathrm{T,c}}$. Since we need to find an exact expression for the norm and determinant of $\mathbb{E}\big(P[\infty]\big)$, we need to make a few simplifying assumptions in this part. More specifically, we assume that $C = \varsigma I$ and $Q = qI$. We furthermore take $A = A_{\mathrm{s}}$, where A_{s} is a symmetric matrix, i.e. $A_{\mathrm{s}} = A_{\mathrm{s}}^{T}$. Under these assumptions, (5.53) can be rewritten as

$$\mathbb{E}\big(P[\infty]\big) = \varsigma^{-2}\sigma_{n,\mathrm{ave}}^{2}(\Upsilon_{\mathrm{T}}) \sum_{i=0}^{\infty} \mu_{\mathrm{ave}}^{i}(\Upsilon_{\mathrm{T}}) \big(A_{\mathrm{s}}\big)^{2i+2} + q \sum_{i=0}^{\infty} \mu_{\mathrm{ave}}^{i}(\Upsilon_{\mathrm{T}}) \big(A_{\mathrm{s}}\big)^{2i}. \quad (5.54)$$

Theorem 5.3. *(Balance of Information Loss & Communication Noise) Consider the average system dynamics of (5.54) with an unstable A_{s}. Then $\Upsilon^{*}_{T,norm,fixed}$ will be as follows:*

$$\Upsilon^{*}_{T,norm,fixed} = \begin{cases} \gamma^{*}_{T,norm,fixed} & \gamma^{*}_{T,norm,fixed} \geq 0 \\ 0 & otherwise \end{cases}, \quad (5.55)$$

where $\gamma^{*}_{T,norm,fixed}$ is the unique solution to the following equation:

$$\underbrace{\mu_{ave}(\gamma^{*}_{T,norm,fixed})}_{information\quad loss} + \underbrace{\sigma^{2}_{n,norm}(\gamma^{*}_{T,norm,fixed})}_{overall\quad comm.\quad noise} + \frac{\varsigma^{2} q}{\rho^{2}_{\max}\sigma^{2}_{n}(\Upsilon = \gamma^{*}_{T,norm,fixed})} = \rho^{-2}_{\max}, \quad (5.56)$$

with $\sigma^{2}_{n,norm}$ denoting the normalized average overall communication noise variance: $\sigma^{2}_{n,norm}(\gamma^{*}_{T,norm,fixed}) = \frac{\sigma^{2}_{n,ave}(\gamma^{*}_{T,norm,fixed})}{\sigma^{2}_{n}(\Upsilon=\gamma^{*}_{T,norm,fixed})}$. Furthermore, $\Upsilon^{*}_{T,det,fixed}$ is determined as follows:

$$\Upsilon^{*}_{T,det,fixed} = \begin{cases} \gamma^{*}_{T,det,fixed} & \gamma^{*}_{T,det,fixed} \geq 0 \\ 0 & otherwise \end{cases} \quad (5.57)$$

where $\gamma^*_{T,det,fixed}$ *is the unique solution to the following equation:*

$$\sum_{i=1}^{N} \frac{\rho_i^2}{1-\rho_i^2 \mu_{ave}(\gamma^*_{T,det,fixed})} = \sum_{i=1}^{N} \frac{1}{\sigma^2_{n,norm}(\gamma^*_{T,det,fixed}) + \frac{q\varsigma^2}{\sigma_n^2(\Upsilon=\gamma^*_{T,det,fixed})\rho_i^2}}, \quad (5.58)$$

with $\rho_1, \rho_2, \ldots, \rho_N$ *representing the ordered eigenvalues of matrix A:* $|\rho_1| \geq |\rho_2| \geq \ldots \geq |\rho_N|$ *and* $\rho_{\max} = |\rho_1|$.

Proof. Consider the diagonalization of matrix A_s: $A_s = LDL^T$, where $L^{TL} = I$ and $D = \text{diag}\{\rho_1, \rho_2, \ldots, \rho_N\}$. The following can be easily confirmed from (5.54): $\mathbb{E}(P[\infty]) = L\text{diag}\left\{\frac{q+\varsigma^{-2}\rho_1^2\sigma^2_{n,\text{ave}}(\Upsilon_\text{T})}{1-\rho_1^2\mu_\text{ave}(\Upsilon_\text{T})}, \ldots, \frac{q+\varsigma^{-2}\rho_N^2\sigma^2_{n,\text{ave}}(\Upsilon_\text{T})}{1-\rho_N^2\mu_\text{ave}(\Upsilon_\text{T})}\right\} L^T$, which results in

$$\left\|\mathbb{E}(P[\infty])\right\| = \frac{q+\varsigma^{-2}\rho_1^2\sigma^2_{n,\text{ave}}(\Upsilon_\text{T})}{1-\rho_1^2\mu_\text{ave}(\Upsilon_\text{T})}. \quad (5.59)$$

Let $\gamma^*_{T,\text{norm,fixed}}$ represent any solution to (5.56). It can be easily verified that $\frac{\partial\|\mathbb{E}(P[\infty])\|}{\partial\Upsilon_\text{T}}$ is only zero at $\gamma^*_{T,\text{norm,fixed}}$. Next we show that (5.56) has a unique solution. Assume that (5.56) has two solutions: $\gamma^*_{T,\text{norm,fixed,1}}$ and $\gamma^*_{T,\text{norm,fixed,2}} > \gamma^*_{T,\text{norm,fixed,1}}$. Since σ_n^2 is a non-increasing function of Υ, we will have the following:

$$\begin{aligned}
&\mu_\text{ave}(\gamma^*_{T,\text{norm,fixed,1}}) + \sigma^2_{n,\text{norm}}(\gamma^*_{T,\text{norm,fixed,1}}) + \frac{\varsigma^2 q}{\rho^2_{\max}\sigma_n^2(\gamma^*_{T,\text{norm,fixed,1}})} \\
&\quad - \left[\mu_\text{ave}(\gamma^*_{T,\text{norm,fixed,2}}) + \sigma^2_{n,\text{norm}}(\gamma^*_{T,\text{norm,fixed,2}}) + \frac{\varsigma^2 q}{\rho^2_{\max}\sigma_n^2(\gamma^*_{T,\text{norm,fixed,2}})}\right] \\
&= \underbrace{\int_{\gamma^*_{T,\text{norm,fixed,2}}}^{\gamma^*_{T,\text{norm,fixed,1}}} \chi(\Upsilon)\mathrm{d}\Upsilon + \int_{\gamma^*_{T,\text{norm,fixed,1}}}^{\gamma^*_{T,\text{norm,fixed,2}}} \frac{\sigma_n^2(\Upsilon)\chi(\Upsilon)}{\sigma_n^2(\Upsilon=\gamma^*_{T,\text{norm,fixed,1}})}\mathrm{d}\Upsilon}_{<0} \\
&\quad + \underbrace{\left(\frac{1}{\sigma_n^2(\Upsilon=\gamma^*_{T,\text{norm,fixed,1}})} - \frac{1}{\sigma_n^2(\Upsilon=\gamma^*_{T,\text{norm,fixed,2}})}\right)\int_{\gamma^*_{T,\text{norm,fixed,2}}}^{\infty} \sigma_n^2(\Upsilon)\chi(\Upsilon)\mathrm{d}\Upsilon}_{<0} \\
&\quad + \underbrace{\frac{\varsigma^2 q}{\rho^2_{\max}}\left(\frac{1}{\sigma_n^2(\Upsilon=\gamma^*_{T,\text{norm,fixed,1}})} - \frac{1}{\sigma_n^2(\Upsilon=\gamma^*_{T,\text{norm,fixed,2}})}\right)}_{<0} < 0. \quad (5.60)
\end{aligned}$$

Therefore[2], $\gamma^*_{T,\text{norm,fixed,1}} = \gamma^*_{T,\text{norm,fixed,2}}$. Let $\Upsilon_{\text{T,c}}$ be the critical stability threshold defined in the previous section: $1-\rho^2_{\max}\mu_\text{ave}(\Upsilon_{\text{T,c}}) = 0$. We have $\gamma^*_{T,\text{norm,fixed}} < \Upsilon_{\text{T,c}}$. Consider those cases where there exists a non-negative solution to (5.56).

[2] Note that $\frac{\partial\sigma_n^2(\Upsilon)}{\partial\Upsilon}$ is taken to be zero only asymptotically.

Then using the fact that $\lim_{\Upsilon_{\mathrm{T}} \to \Upsilon_{\mathrm{T,c}}} \mathbb{E}\big(P[\infty]\big) \to \infty$ shows that $\gamma^*_{T,\mathrm{norm,fixed}}$ corresponds to the unique minimum of $\big\|\mathbb{E}\big(P[\infty]\big)\|$, i.e. $\Upsilon^*_{T,\mathrm{norm,fixed}} = \gamma^*_{T,\mathrm{norm,fixed}}$. If the process noise is the dominant noise, compared to the communication noise, there may be no positive solution to (5.56). It can be easily seen that, in such cases, $\big\|\mathbb{E}\big(P[\infty]\big)\big\|$ will be an increasing function for $\Upsilon_{\mathrm{T}} \geq 0$, resulting in $\Upsilon^*_{T,\mathrm{norm,fixed}} = 0$.

Next, we will find $\Upsilon^*_{T,\mathrm{det,fixed}}$. We will have $\det\Big(\mathbb{E}\big(P[\infty]\big)\Big) = \prod_{i=1}^{N} \frac{\rho_i^2 \varsigma^{-2}\sigma^2_{n,\mathrm{ave}}(\Upsilon_{\mathrm{T}})+q}{1-\rho_i^2\mu_{\mathrm{ave}}(\Upsilon_{\mathrm{T}})}$. It can be easily confirmed that

$$\frac{\partial \det\ \mathbb{E}(P[\infty])}{\partial \Upsilon_{\mathrm{T}}} = \chi(\Upsilon_{\mathrm{T}}) \frac{\prod_{i=1}^{N}\left(\varsigma^{-2}\sigma^2_{n,\mathrm{ave}}(\Upsilon_{\mathrm{T}})\rho_i^2+q\right)}{\prod_{i=1}^{N}\left(1-\rho_i^2\mu_{\mathrm{ave}}(\Upsilon_{\mathrm{T}})\right)} \Bigg[\sum_{j=1}^{N} \frac{\rho_j^2}{1-\rho_j^2\mu_{\mathrm{ave}}(\Upsilon_{\mathrm{T}})} - \sum_{j=1}^{N} \frac{\varsigma^{-2}\sigma^2_n(\Upsilon=\Upsilon_{\mathrm{T}})\rho_j^2}{\varsigma^{-2}\sigma^2_{n,\mathrm{ave}}(\Upsilon_{\mathrm{T}})\rho_j^2+q}\Bigg]. \tag{5.61}$$

Therefore, $\frac{\partial \det\left(\mathbb{E}\left(P[\infty]\right)\right)}{\partial \Upsilon_{\mathrm{T}}}|_{\Upsilon_{\mathrm{T}}=\gamma^*_{T,\mathrm{det,fixed}}} = 0$ will result in (5.58).

In a similar manner, it can be easily confirmed that (5.58) has a unique solution and that $\Upsilon^*_{T,\mathrm{det,fixed}}$ corresponds to the global minimum of the determinant of the asymptotic average estimation error variance. □

Theorem 5.3 shows that in this case the optimum way of dropping packets is the one that provides a balance between information loss (μ_{ave}) and overall communication noise ($\sigma^2_{n,\mathrm{ave}}$). Equation (5.56) (and (5.58)) may not have a positive solution if process noise is the dominant noise compared to the overall communication noise (the third term on the left hand side of (5.56), for instance, will then get considerably high values). In such cases, the receiver should keep all the packets as communication noise is not the bottleneck. However, as long as process noise is not the dominant noise, the optimum way of dropping packets is the one that provides a balance between information loss and communication noise as indicated by (5.56) (and (5.58)).

Figure 5.6 shows the asymptotic average estimation error variance as a function of a non-adaptive threshold, for different average SNRs and for the same parameters of Fig. 5.5. It can be seen that if Υ_{T} is too low, estimation performance degrades due to excessive communication noise. On the other hand, having Υ_{T} too high will result in the loss of information, which will degrade the performance. The optimum Υ_{T} (as predicted by Theorem 5.3) provides the necessary balance between loss of information and communication noise, reaching the minimum of the estimation error curves. As Υ_{T} increases, the estimation will approach the instability regions, predicted by (5.18) due to high information loss. Note that while (5.56) is derived for symmetric A matrices, Fig. 5.6 is plotted for a non-symmetric one. Yet the minima of the curves satisfy (5.56). This suggests that a similar expression could be valid for the general case.

Figure 5.7 (left) shows the trace of the average estimation error variance as a function of time and for both adaptive and non-adaptive approaches. The average SNR is 18 dB for this figure and the rest of the parameters are the same as in Fig. 5.5.

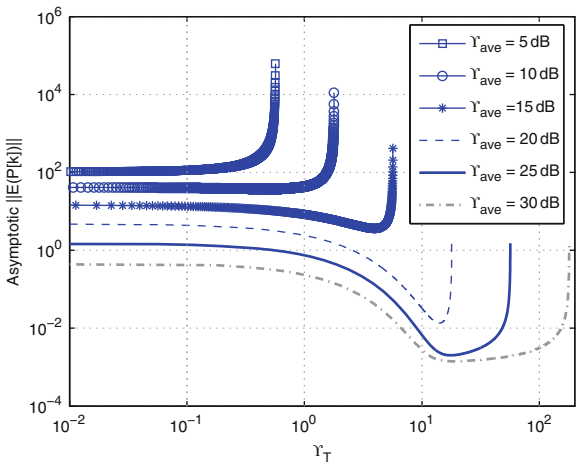


Fig. 5.6 Minimums of the curves indicate the optimum packet drop threshold for the non-adaptive case, when the knowledge of reception quality is not available at the estimator

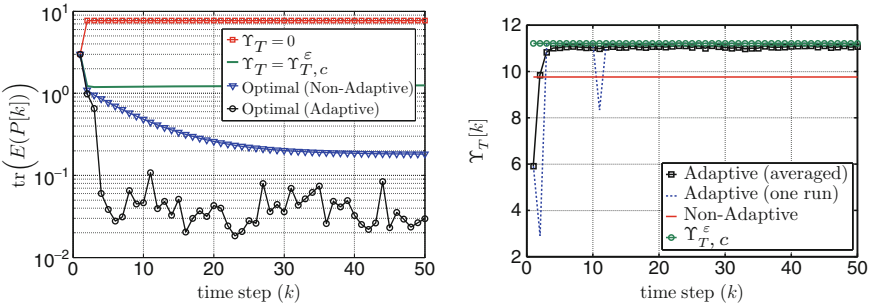


Fig. 5.7 (*Left*) Packet drop threshold optimization for the case where the estimator has no-knowledge of the reception quality and $\Upsilon_{\text{ave}} = 18$ dB. (*Right*) The corresponding thresholds for the left figure

Performances for the cases of $\Upsilon_{\text{T}} = 0$ and $\Upsilon_{\text{T}} = \Upsilon^{\varepsilon}_{\text{T,c}}$ are also plotted for comparison which show that they perform poorly in this case. Figure 5.7 (right) shows the corresponding thresholds for this figure. It can be seen that the adaptive approach chooses lower thresholds (on average) at the beginning. However, as the estimation error variance decreases, it can afford to choose higher thresholds. As compared with the non-adaptive approach, it can be seen that the adaptive approach performs better, as expected. By monitoring the current estimation error variance constantly, the adaptive approach can optimize the threshold better. However, to do so, it (the physical layer) requires constant knowledge of the estimation error variance from the application layer. Furthermore, the optimum threshold for the adaptive case can take values close to the critical threshold more often to optimize the performance. However, this comes at the cost of decreasing the stability margin. Thus, there are interesting trade-offs between the two approaches. Figure 5.8 shows similar curves for the case of average SNR of 25 dB. Similar trends can be seen.

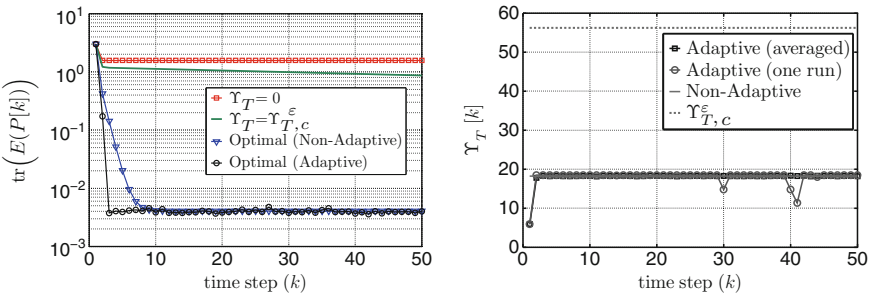


Fig. 5.8 (*Left*) Packet drop threshold optimization for the case, where the estimator has no-knowledge of the reception quality and $\Upsilon_{\text{ave}} = 25\,\text{dB}$. (*Right*) The corresponding thresholds for the left figure

5.4.2 *Estimator has Full Knowledge on the Reception Quality*

In this part, we consider the case where the full-knowledge of reception quality (R) is available at the estimator. Consider (5.19) (second and third lines). First, we characterize the one-step optimization of packet drop. Let $\Upsilon_{T_1}[k]$ and $\Upsilon_{T_2}[k]$ represent two possible thresholds at time-step k, where $\Upsilon_{T_1}[k] < \Upsilon_{T_2}[k]$. Note that $R\big(\Upsilon[k]\big) = R_{\text{s}} + \sigma_{\text{c}}^2(\Upsilon[k])I$. Then, given $P[k]$, it can be easily confirm (using (5.19)) that if $\Upsilon_{T_1}[k] < \Upsilon[k] < \Upsilon_{T_2}[k]$, then $P_1[k+1] \preceq P_2[k+1]$. Otherwise, $P_1[k+1] = P_2[k+1]$. Therefore, for any SNR at time k, we have $P_1[k+1] \preceq P_2[k+1]$. This means that the design with a smaller threshold will lower the next step estimation error variance. Therefore, the optimum threshold will be zero: $\Upsilon_{\text{T}}^* = 0$. Next, we consider non-adaptive optimization of Υ_{T}.

Theorem 5.4. *Consider (5.19). Keeping all the packets, i.e.* $\Upsilon_{\text{T}} = 0$*, will minimize the estimation error covariance at any time and therefore its average too.*

Proof. Let P_1 and P_2 represent estimation error covariance matrices of two estimators using fixed thresholds of Υ_{T1} and Υ_{T2}, where $\Upsilon_{T1} < \Upsilon_{T2}$. Then, $\Pi_{z,1}[k] \preceq \Pi_{z,2}[k]$. Assume that $P_1[0] = P_2[0]$. It is easy to see that $P_1[1] \preceq P_2[1]$ for any $\Upsilon[0]$. Consider the case where $P_1[k] \preceq P_2[k]$. Then, we have

$$\begin{aligned} P_1[k+1] &\preceq AP_1[k]A^T + Q - AP_1[k]\left(P_1[k] + \Pi_{z,2}[k]\right)^{-1} P_1[k]A^T \\ &= Q + A\Pi_{z,2}[k]A^T - A\Pi_{z,2}[k](P_1[k] + \Pi_{z,2}[k])^{-1}\Pi_{z,2}[k]A^T \\ &\preceq Q + A\Pi_{z,2}[k]A^T - A\Pi_{z,2}[k](P_2[k] + \Pi_{z,2}[k])^{-1}\Pi_{z,2}[k]A^T \\ &\preceq P_2[k+1]. \end{aligned} \tag{5.62}$$

Therefore, $\Upsilon_{\text{T}} = 0$ minimizes the estimation error variance at any time as well as its average. □

Figure 5.9 shows the performance of threshold optimization for both cases where the estimator has "full knowledge" and "no knowledge" on the reception quality.

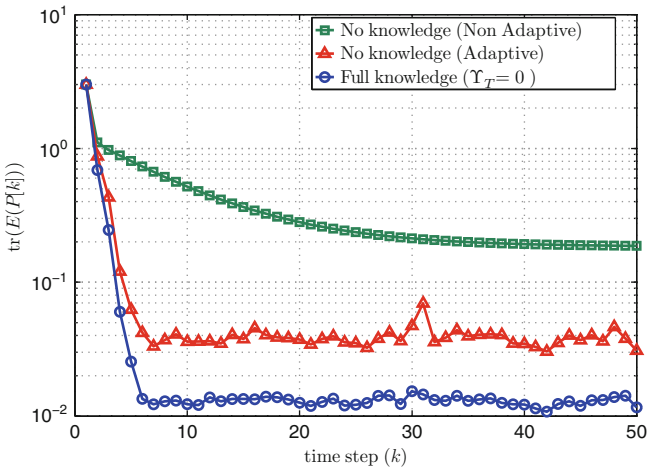


Fig. 5.9 Packet drop threshold optimization for both cases, where the estimator has "full-knowledge" and "no-knowledge" on the reception quality and $\Upsilon_{\text{ave}} = 18$ dB

It can be seen that the case with full knowledge (optimum threshold is zero in this case) performs the best, as expected. It can also be seen that the performance of the adaptive approach, when channel knowledge is not available at the estimator, is close to the case of full knowledge.

5.5 Conclusions and Further Extensions

In this chapter, we considered the impact of both stochastic communication noise and packet drop on the estimation of a dynamical system over a wireless fading channel. We characterized the impact of the knowledge available on the link qualities by considering the stability and performance of three cases of Kalman filtering with "full knowledge", "no knowledge" and "partial knowledge". We then proposed adaptive and non-adaptive ways of optimizing the packet drop to minimize the average estimation error variance of the Kalman filter. Our results showed that considerable performance can be gained by using our proposed optimization framework. There are several ways of extending our results, as we discussed throughout the chapter. We assumed an invertible C matrix. Furthermore, the derivations of the partial-knowledge case were carried out assuming that the known part of stochastic noise variance is small. Our proposed optimization framework of Sect. 5.4 can also be extended to scenarios where the physical layer only has partial information on the link quality, such as an upper bound. Mathematically characterizing the performance of such cases and properly optimizing the threshold are among possible extensions of this work. Finally, we assumed stochastic stationary channels in this chapter. Our framework can be easily extended to the case of non-stationary channels.

Appendix

Let $P_1[k]$ and $P_2[k]$ represent the estimation error covariance matrices of the noise-free and full-knowledge cases, respectively. From (5.22), we know that $\mathbb{E}\big(P_2[k+1]\big) \succeq \mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)A\mathbb{E}\big(P_2[k]\big)A^T + Q$. Therefore, we can easily establish that $\mathbb{E}\big(P_2[k]\big) \succeq \mathbb{E}\big(P_1[k]\big) \Rightarrow \mathbb{E}\big(P_2[k+1]\big) \succeq \mathbb{E}\big(P_1[k+1]\big)$. Next, we compare the average dynamics of the full-knowledge case with that of no knowledge one. Let $P_1[k]$ and $P_2[k]$ represent the estimation error covariance matrices of full-knowledge and no-knowledge cases, respectively. We have

$$\begin{aligned}\mathbb{E}\big(P_1[k+1]\big|P_1[k]\big) = \big(1-\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\big)\mathbb{E}\big(P_1[k+1]\big|P_1[k],\Upsilon[k]>\Upsilon_{\text{T}}\big)\\ +\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\mathbb{E}\big(P_1[k+1]\big|P_1[k],\Upsilon[k]\leq\Upsilon_{\text{T}}\big).\end{aligned} \tag{5.63}$$

Using Lemma 5.1, it can be easily confirmed that $P_1[k+1]$ is a concave function of $\Sigma_z\big(\Upsilon[k]\big)$ in (5.19). Therefore, using conditional Jensen's inequality, we have $\mathbb{E}\big(P_1[k+1]\big|P_1[k],\Upsilon[k]>\Upsilon_{\text{T}}\big) \preceq AP_1[k]A^T + Q - f\big(P_1[k]\big)$, where

$$f\big(P_1[k]\big) = AP_1[k]\Big(P_1[k] + C^{-1}\mathbb{E}\big(\Sigma_z[k]\big|\Upsilon[k]>\Upsilon_{\text{T}}\big)C^{-T}\Big)^{-1}P_1[k]A^T, \tag{5.64}$$

and $\mathbb{E}\big(P_1[k+1]\big|P_1[k]\big) \preceq AP_1[k]A^T + Q - \big(1-\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\big)f\big(P_1[k]\big)$. It can be seen, using Lemma 5.1, that f is a convex function of $P_1[k]$. Therefore by applying Jensen's inequality, $\mathbb{E}\big(P_1[k+1]\big) \preceq A\mathbb{E}\big(P_1[k]\big)A^T + Q - \big(1-\mu_{\text{ave}}\big(\Upsilon_{\text{T}}\big)\big)f\big(\mathbb{E}\big(P_1[k]\big)\big)$. By noting that $\mathbb{E}\big(\Sigma_z\big(\Upsilon[k]\big)\big|\Upsilon[k]>\Upsilon_{\text{T}}\big) = \frac{R_{\text{ave}}(\Upsilon_{\text{T}})}{1-\mu_{\text{ave}}(\Upsilon_{\text{T}})}$, it can be confirmed, after a few lines of derivations using (5.16), that $\mathbb{E}\big(P_2[k]\big) \succeq \mathbb{E}\big(P_1[k]\big) \Rightarrow \mathbb{E}\big(P_2[k+1]\big) \succeq \mathbb{E}\big(P_1[k+1]\big)$. Then, the stability condition for the full-knowledge case can be established in a similar manner to Theorem 5.1.

Acknowledgements This work is supported in part by NSF award # 0812338.

References

1. C. Chong and S. Kumar, "Sensor networks: evolution, opportunities and challenges," *Proceedings of the IEEE*, vol. 91, pp. 1247–1256, Aug. 2003.
2. B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, vol. 91, pp. 1235–1246, Aug. 2003.
3. W. Jakes, *Microwave Mobile Communications*. IEEE Press, New York, 1974.
4. T. S. Rappaport, *Wireless Communications, Principles and Practice*. Prentice-Hall, New Jersey, July 1999.
5. W. Wong and R. Brockett, "Systems with Finite Communication Bandwidth Constraints–Part I: State Estimation Problems," *IEEE Transactions on Automatic Control*, vol. 42, Sept. 1997.
6. W. Wong and R. Brockett, "Systems with Finite Communication Bandwidth Constraints–II: Stabilization with Limited Information Feedback," *IEEE Transactions on Automatic Control*, vol. 44, May 1999.

7. S. Tatikonda, *Control under Communication Constraints*. PhD thesis, MIT, 2000.
8. J. Hespanha, A. Ortega, and L. Vasudevan, "Towards the control of linear systems with minimum bit-rate," in *In Proc. of the Int. Symp. on the Mathematical Theory of Networks and Syst.*, vol. 1, Aug. 2002.
9. N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46, Sept. 2001.
10. F. Faganini and S. Zampieri, "Stability analysis and synthesis for scalar linear systems with a quantized feedback," *IEEE Transactions on Automatic Control*, vol. 48, Sept. 2003.
11. A. Sahai and S. Mitter, "The Necessity and Sufficiency of Anytime Capacity for Stabilization of a Linear System over a Noisy Communication Link, Part I: Scalar Systems," *IEEE Transactions on Information Theory*, vol. 52, pp. 3369–3395, Aug. 2006.
12. N. Martins and M. Dahleh, "Fundamental limitations of performance in the presence of finite capacity feedback," in *Proceedings of 2005 American Control Conference*, vol. 1, pp. 79–86, June 2005.
13. J. Nilsson, *Real-time Control Systems with Delays*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, 1998.
14. M. Micheli and M. I. Jordan, "Random sampling of a continuous-time dynamical systems," in *Proceedings of 15th International Symposium on the Mathematical Theory of Networks and Systems (MTNS)*, (University of Notre Dame, South Bend, Indiana), 2002.
15. B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.
16. X. Liu and A. Goldsmith, "Kalman filtering with partial observation losses," in *Proceedings of the 43rd IEEE Conf. Decision and Control*, vol. 4, (Atlantis, Bahamas), pp. 4180–4186, Dec. 2004.
17. L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1562–1572, Sept. 2004.
18. M. Chow, Z. Sun, and H. Li, "Optimal stabilizing gain selection for networked control systems with time delays and packet losses," *IEEE Transactions on Control Systems Technology*, vol. 17, pp. 1154–1162, Sept. 2009.
19. O. C. Imer, S. Yuksel, and T. Basar, "Optimal control of LTI systems over communication networks," *Automatica*, vol. 42, pp. 1429–1440, Sept. 2006.
20. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, pp. 163–187, Jan. 2007.
21. J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, pp. 138–162, Jan. 2007.
22. Y. Mostofi and R. Murray, "To Drop or Not to Drop: Design Principles for Kalman Filtering over Wireless Fading Channels," *IEEE Transactions on Automatic Control*, vol. 54, pp. 376–381, February 2009.
23. Y. Mostofi and R. Murray, "Kalman filtering over wireless fading channels - How to handle packet drop," *International Journal of Robust and Nonlinear Control*, vol. 19, pp. 1993–2015, February 2009.
24. Y. Mostofi, A. Gonzales-Ruiz, A. Ghaffarkhah, and D. Li, "Characterization and Modeling of Wireless Channels for Networked Robotic and Control Systems – A Comprehensive Overview," in *Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (St. Louis, MO), October 2009.
25. A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
26. D. Son, B. Krishnamachari, and J. Heidemann, "Experimental Analysis of Concurrent Packet Transmissions in Low-Power Wireless Networks," USC-ISI Technical Report, ISI-TR-2005-609, November 2005.
27. T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall information and system sciences series, 2000.
28. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

29. R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, Cambridge, 1999.
30. A. W. Marshal and I. Olkin, *Inequalities: Theory of Majorization and its Applications*. Academic, New York, 1979.
31. Y. Mostofi and R. Murray, "Effect of time-varying fading channels on the control performance of a mobile sensor node," in *Proceedings of IEEE 1st International Conference on Sensor and Adhoc Communications and Networks (SECON)*, (Santa Clara, California), Oct. 2004.

Chapter 6
Time-Delay Estimation and Finite-Spectrum Assignment for Control Over Multi-Hop WSN

Emmanuel Witrant, Pangun Park, and Mikael Johansson

Abstract Recent technological developments and the need for global control strategies to meet with stringent performance requirements rendered the use of wireless sensor networks (WSNs) of prime importance for today's automation. Such communication devices heavily affect the information transport, especially when a multi-hop configuration is introduced to minimize the energy consumption of the network. The associated time-delays are strongly varying and necessitate the design of a dedicated control approach for remote regulation, as proposed in this chapter. First, the communication process is analyzed and illustrated with the Breath protocol, which results in a description of the delay in the frequency domain. Such analysis motivates the use of a CUMSUM Kalman filter to estimate the delay from round-trip-time (RTT) measurements while selecting the desired frequencies. This estimation then allows for the setup of a state-predictor that ensures a finite spectrum assignment (FSA) on the closed-loop spectrum. The efficiency of this approach and the impact of packet losses is finally investigated on an inverted pendulum using experimental network measurements.

Keywords Network controlled systems · Wireless networks · Multi-hop networks · Time-delay estimation · Predictive control · Finite spectrum assignment

6.1 Introduction

There are major advantages in terms of increased productivity and reduced installation costs in the use of wireless communication technology in industrial control systems [1, 2]. Several wireless technologies are currently reaching the plant and

E. Witrant (✉)
UJF GIPSA-lab, Department of Control Systems, University of Grenoble,
Saint Martin d'Hères, France
e-mail: emmanuel.witrant@gipsa-lab.inpg.fr

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_6,

workshop floor addressing a range of applications. A multi-hop wireless network is one such technology that offers not only great flexibility, but also challenges when it comes to predictability.

Wireless sensor networks (WSNs) are used in emerging applications in industrial automation and dedicated algorithms, such as randomized multi-hop routing protocols [3], were recently shown to be useful in these applications. From the perspective of designing control systems utilizing wireless multi-hop networks, one of the most significant characteristics is the need of hiding the system complexity through suitable abstraction of the networks and their nodes. One simple but important way of doing this is by interpreting the network as a communication channel with time-varying delay.

Experimental results on telerobotics over TCP/IP networks illustrate the importance of varying transmission delays in the control loop [4], where the authors combine a generic estimation/prediction control scheme with a buffering of the received packets, which compensates for the delays jitter. While the buffer may decrease the transient performances of the closed-loop system [5], this work motivates the use of estimation/prediction schemes to achieve robust and efficient control over networks. Another interesting experimental result is provided in [6], where a peer-to-peer wireless network (802.11b) and a UDP protocol are used to transmit the sensors and actuators signals. An inverted pendulum is controlled through this network using LQR techniques without compensating the delay (classical state feedback), which results in relatively large oscillations in the system response (complex poles induced by the delay) even for small latencies (average value less then 3.6 ms). The packet losses (significant effect above 15%) are efficiently compensated using a dynamic adjustment of the sampling interval. These experimental results illustrate the fact that when the transmissions are fairly deterministic and the control/sensors signals have a significant effect on the network load, simple solutions can be used to compensate for the packet losses and timing issues while the delay effects need a more careful consideration.

The main focus of this chapter is a new scheme for control over wireless multi-hop networks based on time-delay and finite spectrum assignment (FSA). Time-varying delays for the transmission of sensor and control data over the wireless network are caused by a randomized multi-hop routing protocol called Breath [3]. Breath is designed for control applications using WSN and it ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. We also consider the modification of Breath by using an Automatic Repeat reQuest (ARQ) mechanism [7] to improve the packet reception rate (reliability). The characteristics of the routing protocol together with lower-layer network mechanisms give rise to a delay process with high variance and stepwise changing mean. Estimation of the transport delay over multi-hop networks is particularly difficult, since detailed modeling from first principles is prevented by the high complexity and multitude of traffics over the network. Here, we use the low-order model proposed in [8] to set a predictive control scheme with a delay estimator. The estimator is based on a Kalman filter with a change detection algorithm [8]. It is able to track the delay mean changes but efficiently attenuate

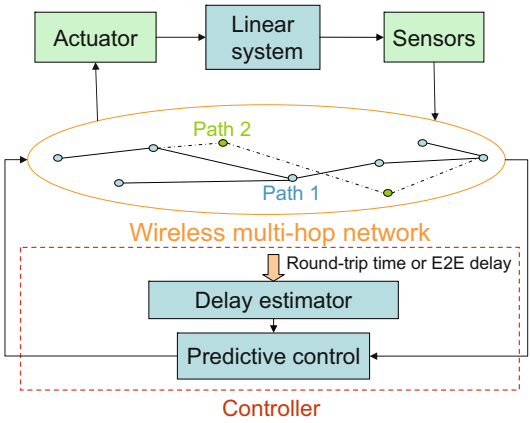


Fig. 6.1 Overview of the system and control setup

the high frequency jitter. Compared to similar works such as [9], where an output feedback scheme uses the maximal value of the delay and GPS synchronization for remote control over Internet, we *explicitly* compensate for the time-varying delay effect. The control scheme is analyzed and its implementation detailed. Network data from an experimental setup, where the randomized multi-hop routing protocol [3] is implemented on sensor nodes, are used to illustrate the efficiency of the approach. This work is an extended version of [10], with additional insights on the delay estimation and WSN aspects.

The outline of the paper is as follows. Section 6.2 defines the considered problem of control over a wireless multi-hop network. The wireless network and its randomized routing protocol is described in Sect. 6.3. The time-delay estimator is presented in Sect. 6.4. Section 6.5 presents the controller and its delay compensator. Experimental results are given in Sect. 6.6, where the tradeoff between lossless communications and time-delays is discussed.

6.2 Problem Description

The problem considered is depicted in Fig. 6.1, where a linear system is remotely controlled over a wireless multi-hop network. We suppose that round-trip time or end-to-end delay measurements (i.e., between the sensors and controller, which also implies a proper clock synchronization) are available. A predictive control law uses the filtered measurements to compensate for variations induced by the multi-hop network, which induces communication delays (time-varying average latency and jitter), packet losses and communication bandwidth constraints. More precisely, we consider the remote control of linear systems that write as:

$$\dot{x}(t) = Ax(t) + Bv(t), \quad x(0) = x_0 \tag{6.1}$$

$$y(t) = Cx(t), \tag{6.2}$$

where $x \in R^n$ is the internal state, $v \in R$ is the system input, $y \in R^m$ is the system output, and A, B, C are matrices of appropriate dimensions. The pairs (A, B) and (A, C) are assumed to be controllable and observable, respectively, but no assumption is made on the stability of A. The delays introduced by the network dynamics are defined as:

- $\tau_1(t)$: delay between the sensors and the controller
- $\tau_2(t)$: delay between the controller and the actuator

The actuator is event-driven and uses the last received signal. Packet losses are included in this description by considering that the last received and most recent controller output $u(\cdot)$ is taken as the input for the system, which is then:

$$\begin{cases} w(t_k) = u(t_k - \tau_2(t_k)) & \text{if packet } k \text{ is received} \\ w(t_k) = w(t_{k-1}) & \text{else} \end{cases} \tag{6.3}$$

where $w(\cdot)$ is the signal received by the actuator from the network, $u(\cdot)$ is the controller output and $v(t) = w(t_k)\ \forall\, t \in [t_k, t_{k+1})$. The index k is used to differentiate the discrete data packets from the continuous physical variables. The chronology of the signals is indicated in the transmitted data to reconstruct their history on the receiver site. The receivers consider only the most recent information, meaning that a packet is also considered as lost if a more recent one is received first.

6.3 Wireless Multi-Hop Networks

In many scenarios of relevant interest, wireless multi-hop networks do not have fixed communication paths [11], but the end-to-end path followed by packets happens according to a dynamic selection of hops [12, 13]. Indeed, it is often impossible to build fixed routing tables for these networks due to the time-varying communication channel and network topology. In such networks, a packet is routed to a particular next-hop node as long as the node selected for a hop ensures progress toward destination. Furthermore, a major technique to ensure power savings and longer network lifetime is to turn off a node whenever its presence is not strictly required for the correct operation of the network [2, 3, 13, 14]. In this case, each node goes to sleep for a random amount of time depending on traffic and network conditions, which means that the network topology is changing randomly.

Consider the multi-hop network reported in Fig. 6.1, where the transmitter to receiver path is composed of a time-varying number of hops $h(t)$. Therefore, we can consider the following sources of delay:

- The time α_i to wait before sending a data packet, which is typically a random variable
- The time F to forward a data packet once the connection with the next node is set up, which is constant and contains the propagation and transmission delay
- Possibly the time β_i induced by an ARQ mechanism, which retransmits a packet for a predefined number of time if no acknowledgement is received

The end-to-end delay can then be expressed as follows:

$$\tau(t) = h(t)F + \sum_{i=1}^{h(t)} (\alpha_i + \beta_i). \tag{6.4}$$

For the Breath protocol [3] used to set the experimental setup, the routing, medium access control, and sleeping discipline are randomized in such a way that the energy consumption of the communication network is minimized while meeting the constraints on packet delay and reliability. Since it is representative of a class of protocol for wireless automation, we take it as a reference for the experimental results. With this protocol, α_i depends on the exponentially distributed wake-up rate, whose average value $\mu_{c,i}$ is constrained by a function of the traffic rate, channel condition, and application requirements. Note that since $h(t)$, α_i and β_i are random variables, the end-to-end delay clearly results in a random variable.

If one looks at the outcomes of (6.4), there are three possible behaviors:

1. Low-frequency delays, due to changes in the number of hops $h(t)$. The end-to-end number of hops is expected to change when an obstacle changes the channel conditions for long time (i.e., the shadow fading of the wireless propagation). Since shadowing changes with low-frequency, this causes low frequency changes in the delay.
2. Middle frequency delays, which depend on the random selection of the next awaken hop for a fixed number of clusters (related to $\mu_{c,i}$ in the example considered). This is induced by the fact that a transmitting node has to wait for some node to wake up in the forwarding region. For example, in the randomized protocol, this phenomenon is directly caused by the wake-up rate $\mu_{c,i}$, which is higher than the rate with which the shadow fading changes.
3. High-frequency delays, related to the transmissions between nodes and packet loss probability. Since the wireless transmission may be lossy due to fast fading (which causes the execution of the ARQ), there might be highly time-varying delays. Hence, this delay is directly related to the packet loss probability. Note that the frequency of the fast fading is very high in comparison with the shadow fading.

Example 1. Considering the WSN multi-hop network presented in Sect. 6.6, which describes the specific experimental setup, time-delay measurements are obtained for a three-hop configuration. The power spectral density of these delays is obtained thanks to the periodograms computation and depicted in Fig. 6.2. We can clearly notice the dominent low-frequency component of the delays ($0\ dB/rad/sample$), followed by the medium (two peaks above $-15\,dB/rad/sample$) and high (below $-18\,dB/rad/sample$, close to white noise) frequency behaviors.

The measurements can also be used to investigate the class of models that can be associated with such delays. As an example, a spectral model (empirical transfer function estimate (ETFE)) and a time-domain model (autoregressive moving average – ARMA) are presented in Fig. 6.3. The ARMA model is set with a second-order polynomial on the delay and a first-order polynomial on the white noise. While

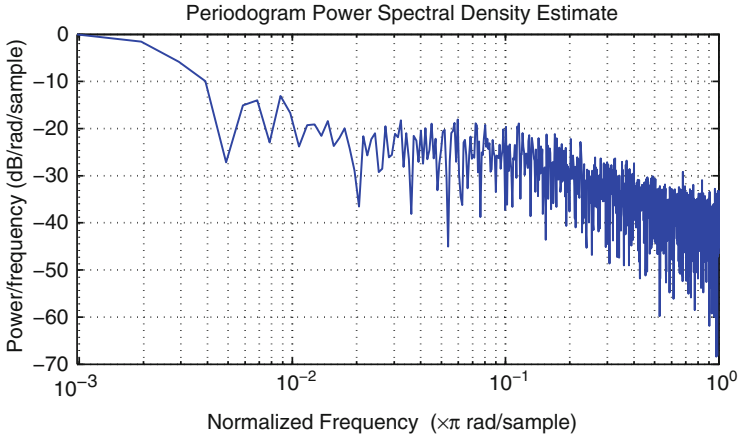


Fig. 6.2 Power spectral density of the delays induced by a three-hops network

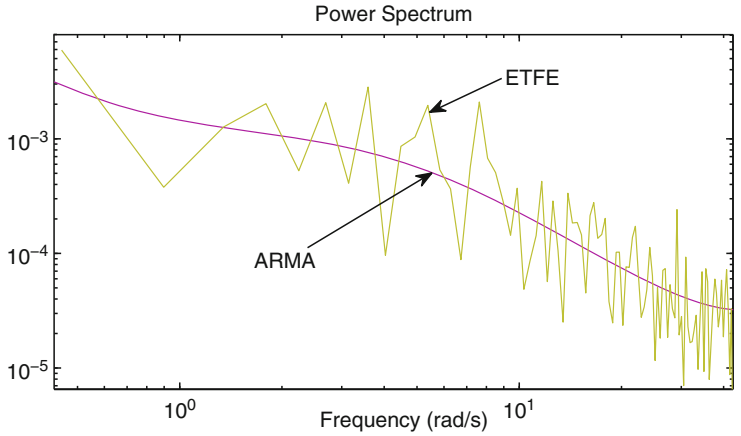


Fig. 6.3 Empirical transfer function estimate (ETFE) and auto regressive moving average (ARMA) model of the network delays

providing for a rough approximation of the signal, we can see that the ARMA model gives a reasonable approximation of the amplitudes at the three frequency levels. This motivates the structure of the delay estimation presented in the next section.

6.4 Time-Delay Estimator

A critical component of the proposed control strategy is the delay estimator, which output is used to set the time-varying horizon of the predictive controller. Examining the real delay traces from our network (introduced in the previous example), shown in dashed lines in Fig. 6.4, we see that the delay exhibits a large variability and

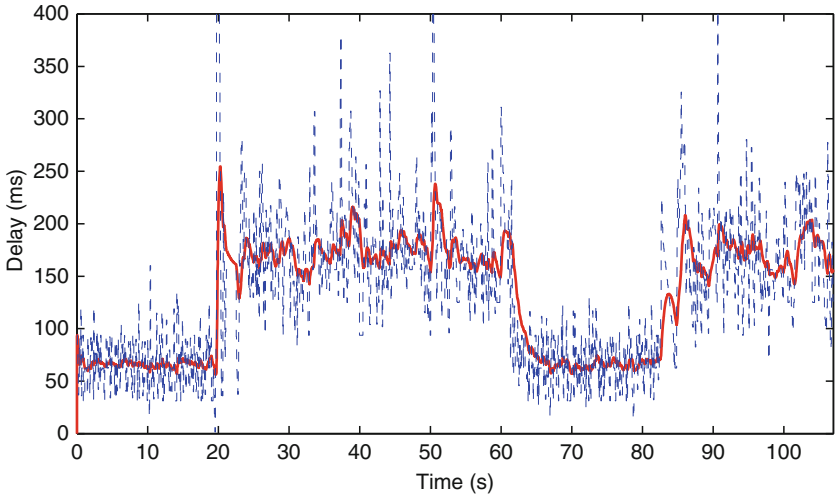


Fig. 6.4 Delay traces and delay estimates computed with the combined Kalman-filter and CUMSUM detector

persistent changes in the mean. Although these changes can be traced back to a particular feature of our protocol (discussed in the previous section), similar effects have been reported for a wide range of network quantities and network technologies [15]. Since the controller-actuator latency is not immediately available to the controller, but has to be deduced from the (delayed information about the) controller-actuator round-trip time, we will look for a delay estimator that allows to smooth out the fast variations while being able to react quickly to the changes in mean.

To this end, we propose to use a combination of Kalman filter and CUMSUM change-detection (cf., [8, 16]). The underlying signal model is:

$$\begin{aligned} \tau_{k+1} &= \tau_k + v_k \\ r_k &= c\,\tau_k + w_k, \end{aligned}$$

where τ_k is the one-way latency, r_k is the measured time, c is the number of delays included in r_k (i.e., 1 if the end-to-end delay of the channel is measured and 2 if it is the round trip time), w_k is a white-noise sequence with covariance R_k and:

$$v_k = \begin{cases} 0 & \text{with probability } 1-q, \\ v & \text{with probability } q, \text{ where } \mathrm{Cov}(v_k) = q^{-1} Q_k, \end{cases}$$

models the abrupt changes in the mean. The associated Kalman filter updates are given by:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + K_k(r_k - c\hat{\tau}_k), \text{ with}$$
$$K_k = \frac{c P_{k-1}}{R_k + c^2 P_{k-1}}, \quad P_k = P_{k-1} - \frac{c^2 P_{k-1}^2}{R_k + c^2 P_{k-1}} + Q_k.$$

The tracking ability of the filter is proportional to the variance of the process noise Q_k, but increasing Q_k also decreases the smoothing effect that we are looking for.

To improve performance, we run a CUMSUM detector in parallel with the Kalman filter. The detector looks at the prediction errors $\varepsilon_k = r_k - c\hat{\tau}_k$ sums up the negative and positive prediction errors:

$$g_k^+ = \max(g_{k-1} + \varepsilon_k - \kappa, 0)$$
$$g_k^- = \max(g_{k-1} - \varepsilon_k - \kappa, 0)$$

and monitors if g_k^+ or g_k^- exceeds a threshold g_{thr}. Here, the parameter κ is a design parameter that specifies the negative drift (see [16] for design guidelines). At these threshold crossings, Q_k is set to a large value, g^+ and g^- are reset to zero, $\hat{\tau}_k$ is set to $c^{-1}r_k$ and the Kalman filter iterations resumed. The combined Kalman-filter and CUMSUM test allows to achieve both fast tracking of the changes in the mean and good suppression of the high-frequency variations, as illustrated in Fig. 6.4 (solid line).

6.5 Predictive Control Approach

The control approach proposed here is based on the state predictor with a time-varying horizon first proposed in [17]. The main advantage of such method compared to other delay compensation strategies [18] is to allow for a *finite spectrum assignment* on systems with time-varying delays [19, 20]. An overview on the use of state predictors in networked control systems is provided in [5], where it is compared with more classical control approaches.

6.5.1 Ideal Delay Compensation

We use the results established in [21] and subsequent works, where the delay induced by the network dynamics is explicitly taken into account in the control setup. For simplicity and to focus on the network effect on the closed-loop system, we suppose that the full delayed state[1] $x(t - \tau_1)$ is available to set the control law. This hypothesis can be easily removed using an observer-based control setup as the one proposed in [22]. When there is no packet loss and the delay is fully known (and can be predicted), the main theorem in [21] can be extended to the case, where both communication channels experience a delay as follows.

[1] The time dependency of the variables $\delta(t)$, $\tau_1(t)$ and $\tau_2(t)$ are omitted in the notations from this point.

Theorem 1. *Consider the system:*

$$\dot{x}(t) = Ax(t) + Bu(t - \tau_2), \quad x(0) = x_0 \tag{6.5}$$

with (A, B) *a controllable pair. Assume that the network dynamics is such that the following holds for* τ_1 *and* τ_2*:*

$(A1)$ $\tau_{1,2} \in \mathcal{C}(R^+, [0, \tau_{\max}]) \quad \forall t,$

$(A2)$ $\dot{\tau}_{1,2} < 1 \quad \forall t \geq t_0.$

Then the feedback control law:

$$u(t) = -K\mathrm{e}^{A\delta}\left[\mathrm{e}^{A\tau_1}x(t - \tau_1) + \mathrm{e}^{At}\int_{t-\tau_1}^{t+\delta}\mathrm{e}^{-A\theta}Bu(\theta - \tau_2(\theta))\mathrm{d}\theta\right] \tag{6.6}$$

$$\dot{\delta}(t) = -\frac{\lambda}{1 - \mathrm{d}\tau_2(\zeta)/\mathrm{d}\zeta}\delta + \frac{\mathrm{d}\tau_2(\zeta)/\mathrm{d}\zeta + \lambda\tau_2(\zeta)}{1 - \mathrm{d}\tau_2(\zeta)/\mathrm{d}\zeta} \tag{6.7}$$

with $\zeta = t + \delta$*,* λ *a positive constant,* $\delta(0) = \delta_0$*, ensures that the system trajectories* $x(t)$ *converge to zero.*

Proof. (Outline) First note that (6.6) is obtained by setting the feedback law as:

$$u(t) = -Kx(t + \delta(t)),$$

where $x(t + \delta(t)$ is obtained from the fundamental of (6.5) as:

$$x(t + \delta) = \mathrm{e}^{A\delta}\left[\mathrm{e}^{A\tau_1}x(t - \tau_1) + \mathrm{e}^{At}\int_{t-\tau_1}^{t+\delta}\mathrm{e}^{-A\theta}Bu(\theta - \tau_2(\theta))\mathrm{d}\theta\right]$$

Noting that with $(A1) - (A2)$ the dynamics (6.7) ensures the exponential convergence toward zero of $\delta(t) - \tau(t + \delta(t))$ the closed-loop dynamics is set as:

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t + \delta) = (A - BK)x(t + \delta).$$

Such dynamics is stable if $(A - BK)$ is Hurwitz and the convergence of $x(t)$ is a direct consequence of the one of $x(t + \delta)$. □

If the communication delay is perfectly known and satisfies A1–A2 (which correspond to a lossless network description), then the closed-loop system behavior can be inferred from the *time-shifted* dynamics:

$$\frac{\mathrm{d}x}{\mathrm{d}\zeta} = (A - BK)x(\zeta)$$

and its spectrum coincides with the spectrum of the matrix:

$$A - \left| \frac{1+\dot{\delta}}{\dot{\tau}(t+\delta)} \right| \mathrm{e}^{-A\delta} BK.$$

Considering the worst-case delay properties to set the closed-loop spectrum, which correspond to a maximum delay $\tau_{\max}$ and a maximum delay variation $\dot{\tau}_{\max}$, the feedback gain has to be chosen such that the matrix:

$$A - \left| \frac{1}{1-\dot{\tau}_{\max}} \right| \mathrm{e}^{-A\tau_{\max}} BK \tag{6.8}$$

is Hurwitz.

6.5.2 Use of the Delay Estimator

The control law is set using the delay estimator described in Sect. 6.4. Both delays τ_1 and τ_2 are then replaced by $\hat{\tau}$ in the control setup description introduced in Theorem 1. This is motivated by the fact that the predictor architecture is used to compensate for the time-variations of the latency (i.e., the slow and medium dynamics of the delay), which are estimated with the CUMSUM Kalman filter. It is reasonable to suppose that the average values of the delays in both communication channels are close and provided by the estimator. The control law is then given as:

$$u(t) = -K\mathrm{e}^{A\delta} \left[\mathrm{e}^{A\hat{\tau}} x(t-\tau_1) + \mathrm{e}^{At} \int_{t-\hat{\tau}}^{t+\delta} \mathrm{e}^{-A\theta} Bu(\theta - \hat{\tau}(\theta)) \mathrm{d}\theta \right]. \tag{6.9}$$

The predictor horizon is computed using the actual values of the estimated delay and delay variation directly instead of the predicted ones (at time ζ). We then approximate δ with $\hat{\delta}$, which dynamics is given by:

$$\dot{\hat{\delta}}(t) = \frac{\dot{\hat{\tau}}(t) + \lambda(\hat{\tau}(t) - \hat{\delta}(t))}{1 - \dot{\hat{\tau}}(t)} \tag{6.10}$$

6.5.3 Control Algorithm

Special care has to be taken in the computation of the control law (6.9)–(6.10), especially concerning the integral term used in the predictor's part. Indeed, the discretization of the integral may introduce some numerical instabilities, as illustrated

in [23] and explained in [24]. The effect of these instabilities is further increased if the communication channel experiences packet losses and bandwidth limitation. A solution to this problem is to include the discretized integral in a dynamic control law, as described in this section.

First, considering the integral term:

$$\mathcal{I}(t) \doteq \mathrm{e}^{At} \int_{t-\hat{\tau}}^{t+\hat{\delta}} \mathrm{e}^{-A\theta} B u(\theta - \hat{\tau}(\theta)) \mathrm{d}\theta$$

computed at the time instant denoted by k, we have

$$\mathcal{I}_k = \mathrm{e}^{A\hat{\tau}_k} \int_{0}^{\hat{\tau}_k+\hat{\delta}_k} \mathrm{e}^{-A t} B u(\mu + t_k - \hat{\tau}_k - \hat{\tau}(\mu + t_k - \hat{\tau}_k)) \mathrm{d}\mu.$$

Introducing the function $f(\mu) \doteq \mathrm{e}^{-A\mu} B u(\mu + t_k - \hat{\tau}_k - \hat{\tau}(\mu + t_k - \hat{\tau}_k))$ and considering that the delay evolution cannot be predicted, the integral term is computed with:

$$f(\mu) = \mathrm{e}^{-A\mu} B u(\min(\mu + t_k - \hat{\tau}_k - \hat{\tau}(\min(\mu + t_k - \hat{\tau}_k, t_k)), t_k))$$

to ensure the causality of the predictor computation.

The second step is to discretize the integral term, which is done according to the guidelines proposed in [24] for the single delay case. More precisely, the integral term is discretized with a trapezoidal rule and included into the dynamic controller:

$$\begin{cases} \dot{z}(t) = -az(t) + bK\mathrm{e}^{A(\hat{\delta}_k + \hat{\tau}_k)} \left[x(t - \tau_1) + \dfrac{t_\mathrm{s}}{2} \displaystyle\sum_{i=1}^{n_k} f((i-1)t_\mathrm{s}) + f(i t_\mathrm{s}) \right], \\ u_k = z(t_k) \end{cases} \tag{6.11}$$

where $n_k \doteq (\hat{\delta}_k + \hat{\tau}_k)/t_\mathrm{s}$, a and b are some positive design parameters, t_s is the sampling time set by the controller clock, and n_k is the number of steps used to discretize the integral, which is timevarying. Note that we chose the controller sampling time as the integrand step, since it sets the buffering of the past estimated delays and controller outputs.

Finally, an upper bound $\dot{\bar{\tau}}_\mathrm{thr}$ is set on the maximum estimated delay variation that can be used for the computation of $\delta(t)$ since the delay estimator may have some fast variations, especially when it is triggered by the alarm signal. The dynamics of the predictor horizon is then set with:

$$\dot{\hat{\delta}}(t) = \frac{\dot{\bar{\tau}}_k + \lambda(\hat{\tau}_k - \hat{\delta}(t))}{1 - \dot{\bar{\tau}}_k}, \tag{6.12}$$

where

$$\begin{cases} \dot{\hat{\tau}}_k = \dfrac{\hat{\tau}_k - \hat{\tau}_{k-1}}{t_s} & \text{if } \dot{\hat{\tau}} < \dot{\hat{\tau}}_{\text{thr}} \\ \dot{\hat{\tau}}_k = \dot{\hat{\tau}}_{\text{thr}} & \text{else} \end{cases}$$

The issues considered in this section are all critical for a safe implementation of the predictor with a time-varying horizon, especially to control a system over the network considered. They are summarized with the following result.

Result 6.5.1 *Using the delay estimation provided by the CUMSUM Kalman filter described in Sect. 6.4, the proposed predictive control approach is set with the dynamic controller (6.11), where the predictor horizon is set by the dynamics (6.12) and the controller gain is such that the closed-loop matrix (6.8) is Hurwitz.*

6.6 Experimental and Simulation Results

In this section, the network setup is presented along with some experimental results, which are used to estimate the efficiency of the proposed control setup using appropriate simulation tools. More details on the experimental setup and protocol specifications can be found in [3].

6.6.1 Communication Protocol

We use the Breath protocol proposed in [3] and briefly discussed in Sect. 6.3. Breath is designed for WSNs where source nodes attached to a plant must transmit information via multi-hop routing to a sink. Breath ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. The design approach relies on a constrained optimization problem, whereby the objective function is the energy consumption and the constraints are the packet delay and reliability. The optimal working point of the protocol is achieved by a simple algorithm, which adapts to traffic variations and channel conditions with negligible overhead. The protocol has been implemented and experimentally evaluated on a test-bed with off-the-shelf wireless sensor nodes [25], and it has been compared with an IEEE 802.15.4 standard solution [26]. Analytical and experimental results show that Breath is tunable and meets delay and reliability requirements. Breath exhibits a good distribution of the working load, thus ensuring a long lifetime of the network. In addition, since this protocol does not include any specific error control mechanism that can ensure a reliable communication, we also introduce a stop-and-wait ARQ algorithm to provide for a more reliable data transfer service. ARQ combines error detection and retransmission to ensure that the data is delivered accurately to the user, despite the errors that may occur during transmission [7].

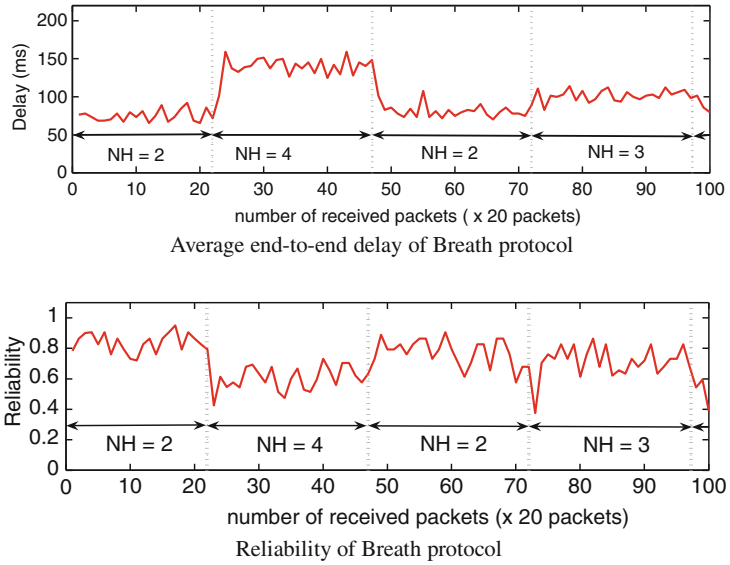


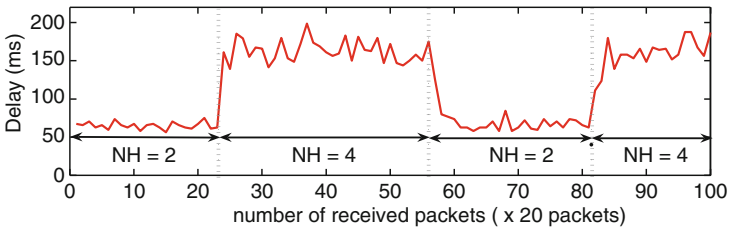
Fig. 6.5 Average end-to-end network-induced delay and reliability of Breath protocol

However, there is packet loss due to the limited number of retransmissions with ARQ mechanism (2 in our experiment). In the following, we consider the Breath protocol without and with ARQ mechanism.

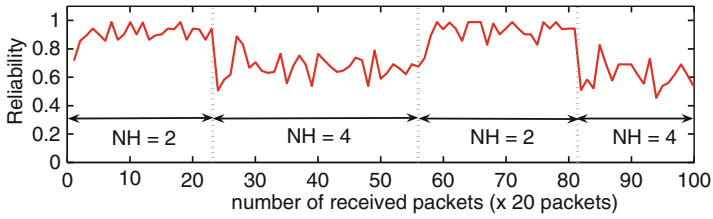
6.6.2 Network Setup and Experimental Measurements

Our experiment is based on the Breath protocol using Tmote nodes [25], which are equipped with the radio controller Chipcon CC2420 operating at 2.4 GHz and supporting the IEEE 802.15.4 standard. The source is located at 30 m from the destination node, which is connected to the computer and generates 20 pckts/s. Eighteen nodes are placed at regular intervals (1.58 m) along a straight line without obstacles. Surrounding objects are static, with minimal time-varying changes in the wireless channel due to multi-path fading effects. At each location, the sensor node is placed in the same orientation to avoid nonlinearities in the transmission pattern.

Figures 6.5 and 6.6 show the average end-to-end delay and reliability of the Breath protocol without and with ARQ, respectively. In the figure, note that "NH" refers to the number of hops of the network. We recall that the number of hops is set depending on the channel condition, traffic load, and application requirements of the network. It is clear to observe that both the average end-to-end delay and the reliability increase as the number of hops increases. In addition, ARQ mechanism increases the end-to-end delay because of the waiting time to receive an acknowledgement. We remark here that there is tradeoff between the packet delay and reliability [2]. Furthermore, increasing the number of hops gives an outstandingly bad reliability if more than four hops are used.



Average end-to-end delay of Breath protocol with ARQ



Reliability of Breath protocol with ARQ

Fig. 6.6 Average end-to-end network-induced delay and reliability of Breath protocol with ARQ

6.6.3 Physical System and Control Setup

The closed-loop system is presented as a block diagram in Fig. 6.7, which depicts the plant, network, controller, and their interconnection. As an example, the linear system considered is similar to the T-shape ECP inverted pendulum presented in [21], an unstable open-loop system with no-minimum phase. Its dynamics is set with the state-space matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.80 & 0 & 12.56 & 0 \\ 0 & 0 & 0 & 1 \\ -2.42 & 0 & -8.33 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 4.57 \\ 0 \\ 0.38 \end{bmatrix}$$

The eigenvalues of A are $\lambda_{1,2} = 0.74 \pm 2.08i$, $\lambda_{3,4} = -0.74 \pm 2.08i$. The choice of an unstable open-loop plant emphasizes the efficiency of the controller to compensate for the transmission perturbations and the performance limitations.

Experimental delay measurements provide for τ_1 and τ_2 by splitting the initial measured data into two sets, one every other sample being attributed to each set. This means that the channels are not symmetric but belong to the same network (same global configuration). Only the last received packets are considered and the packet losses are handled as described in (6.3). The bandwidth limitation is set by zero-order hold with a 50 ms sampling time. The controller includes:

- The time-delay estimator (based on end-to-end delay measurements)
- A reference trajectory block that sets the control objective
- The predictive control setup

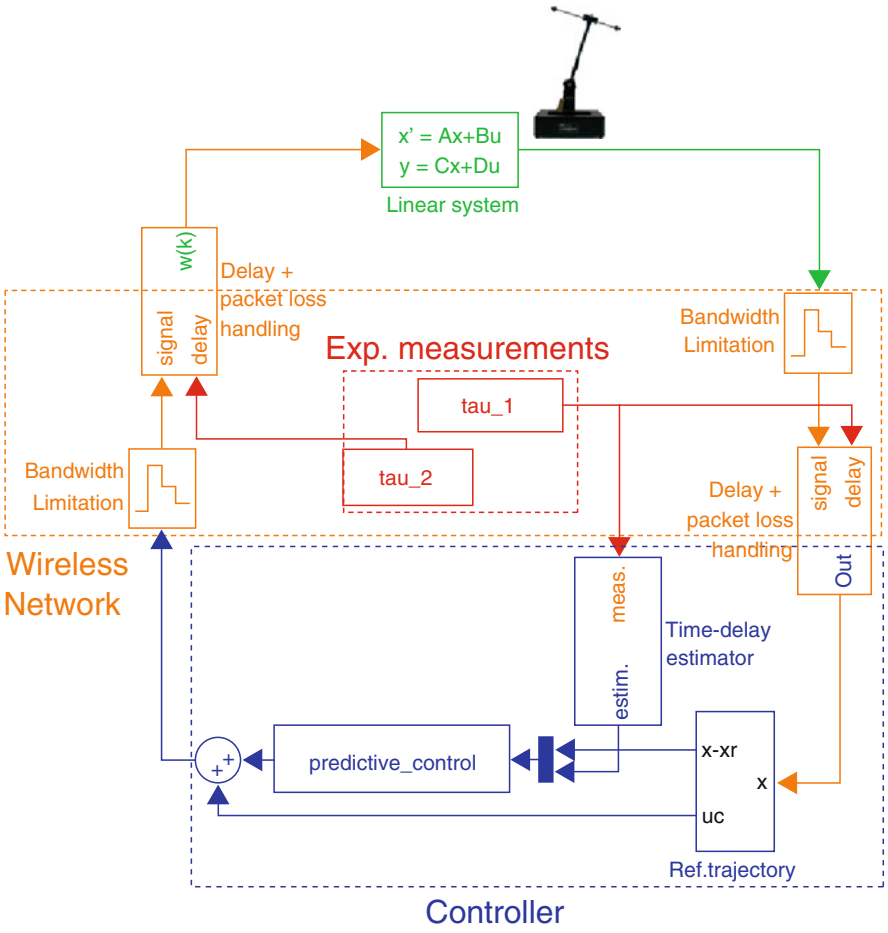


Fig. 6.7 Block diagram of the closed loop system with plant, network, measurements and controller

The CUMSUM Kalman filter is set with $R = 10\,000$, $Q = 1$, $\nu = 0.01$ and $g_{\text{thr}} = 35$. The reference trajectory, for the pendulum angle $x(3)$, is a filtered square signal. The predictive control is set with $\lambda = 10, a = b = 100$ (the introduced controller dynamics is equivalent to a low-pass filter) and $K = [0.48\ 0.36\ 2.20\ 0.57]$ sets the poles of (6.8) to $(-3 + 0.1i; -3 - 0.1i; -4; -4)$.

6.6.4 Simulation Results

Figures 6.8 and 6.9 (without and with ARQ) present the actual and estimated time-delays (top), the reference and the system output (middle), and the control output (bottom). In both cases, the transmissions are very noisy, in terms of jitter and packet

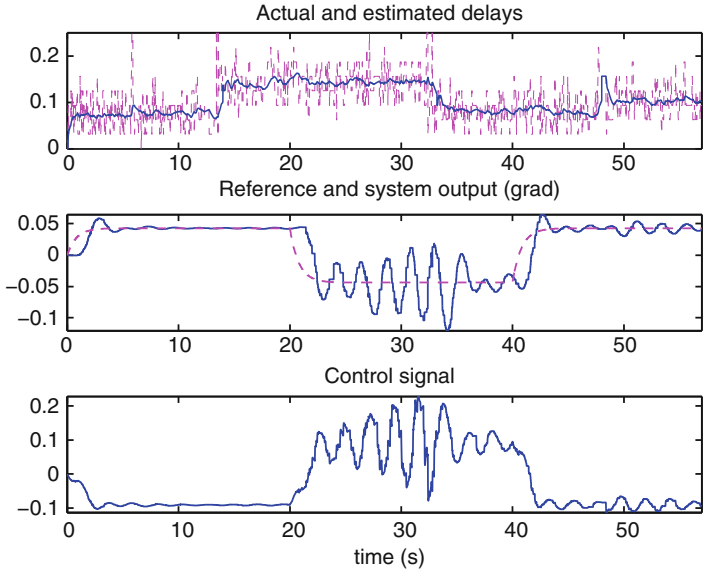


Fig. 6.8 Delays, system output and control signal without ARQ

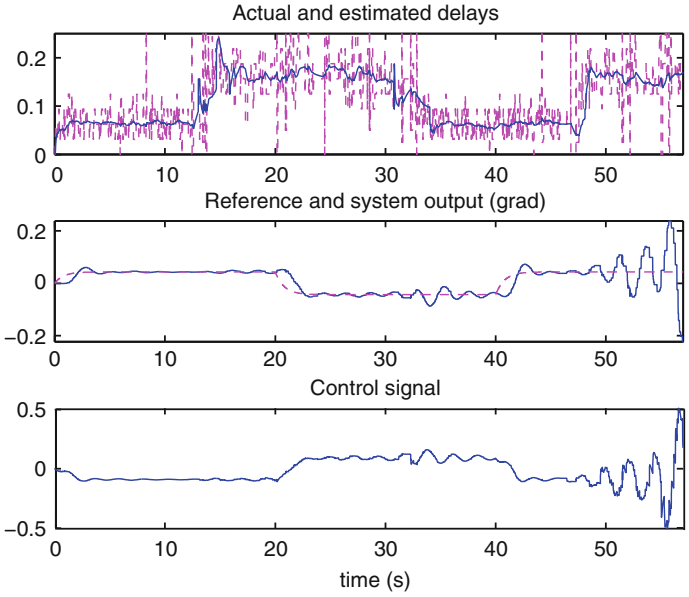


Fig. 6.9 Delays, system output and control signal with ARQ

losses (up to 60% of the signal). These figures illustrate the controller efficiency and the proper handling of the high-frequency noise induced by the numerical integration problem. The effect of the number of hops on the system response clearly

appears in both cases, particularly when a change in the reference sign occurs when there are four hops. Indeed Fig. 6.8 depicts a situation where the bad quality of the communication channel introduces some oscillations, which are reduced as soon as the number of hops is set back to two. The benefit of ARQ appears at the beginning of the simulations but is discussable when the decrease in the average reliability is not compensated by the communication algorithm. Comparing the different simulation results, it appears that the system oscillations are induced by the low reliability rather than high end-to-end delay (i.e., comparing the different 4-hops cases). This is due to the fact that the control setup explicitly compensate the delay effects, while the robustness with respect to packet losses is not specifically addressed in the feedback design. These simulations also show that an increase of the number of hops has few effects on the system response when it is established in a steady state. A control-oriented multi-hop wireless network protocol could efficiently use this information to minimize the emission energy (increase the number of hops) according to some measurements of the physical state while ensuring satisfying performances.

6.7 Conclusions

This chapter addressed the problem of remote control over a multi-hop wireless network. The constraints of such a network on the communication channel, such as time-varying delays (jitter), packet losses, and bandwidth limitations are considered. The time-delay induced by the network is estimated thanks to a CUMSUM Kalman filter that provides for a fast tracking of the mean and good suppression of the high frequency variations of the delay (jitter). This estimator is used to design a dynamic predictive control law that compensates for the estimated part of the delay explicitly. The causality of this controller is set according to the estimator and specific considerations on safe implementation are detailed. A necessary condition on the assigned closed-loop spectrum is provided but the particular poles location is not specified, which provides for a degree of freedom for optimal or robust control approaches. Based on experimental delay and packet loss, some simulations illustrate the proposed control scheme and the choice of a communication protocol.

Acknowledgement This work was supported by the EU project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation, and the Swedish Governmental Agency for Innovation System

References

1. A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, pp. 1130–1151, June 2005.
2. P. Park, "Protocol design for control applications using wireless sensor networks," Licentiate thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2009.
3. P. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-adapting protocol for wireless sensor networks in control and automation," in *IEEE SECON*, 2008.

4. A. Lelevé, P. Fraisse, and P. Dauchez, "Telerobotics over IP networks: towards a low-level real-time architecture," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, Hawai, USA, 2001, pp. 643–648.
5. E. Witrant, C. Canudas-de-Wit, D. Georges, and M. Alamir, "On the use of state predictors in networked control systems," in *Applications of Time-Delay Systems*, ser. Lecture Notes in Control and Information Sciences, J. Chiasson and J.-J. Loiseau, Eds. Springer, 2007, vol. 352.
6. N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireles networks," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 58–71, 2004.
7. A. Leon-Garcia and I. Widjaja, *Counmmunication Netorks*. McGraw Hill, 2004.
8. K. Jacobsson, N. Möller, K. H. Johansson, and H. Hjalmarsson, "Some modeling and estimation issues in traffic control of heterogeneous networks," in *Proc. of the 16th Int. Symp. on Math. Th. of Net. and Syst.*, Leuven, Belgium, July 2004.
9. A. Seuret, M. Termens-Ballester, A. Toguyeni, S. E. Khattabi, and J.-P. Richard, "Implementation of an internet-controlled system under variable delays," in *Proc. of the IEEE Conf. on Emerging Technologies and Factory Automation*, Sept. 2006, pp. 675–680.
10. E. Witrant, P. Park, M. Johansson, C. Fischione, and K. Johansson, "Control over wireless multi-hop networks," in *IEEE Conference on Control Applications*, Singapore, Oct. 2007.
11. J. N. A. Karaki and A. E. Kamal, "Routing technques in wireless sensor networks: A survey," *IEEE Wireless Communication Magazine*, vol. 1, pp. 6–28, 2004.
12. J. V. Greuen, D. Petrović, A. Bonivento, J. Rabaey, K. Ramchandran, and A. Sangiovanni-Vincentelli, "Adaptive sleep discipline for energy conservation and robustness in dense sensor networks," *IEEE ICC*, vol. 1, 2004.
13. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," *ACM/IEEE MobiCom*, vol. 1, pp. 70–84, 2001.
14. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM/IEEE MobiCom*, vol. 1, 2001.
15. M. Kim and B. D. Noble, "Mobile network estimation." in *ACM/IEEE MobiCom*, 2001, pp. 298–309.
16. F. Gustafsson, *Adaptive Filtering and Change Detection*. Wiley, 2000.
17. A. W. Olbrot, "On controllability of linear systems with time delays in the control," *IEEE Transactions on Automatic Control*, vol. ac-16, pp. 664–666, 1972.
18. J. P. Richard, "Time delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, Oct. 2003.
19. A. Manitius and A. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE Transactions on Automatic Control*, vol. 24, pp. 541–552, Aug. 1979.
20. Z. Artstein, "Linear systems with delayed control: a reduction," *IEEE Transactions on Automatic Control*, vol. ac-27, no. 4, pp. 869–879, Aug. 1982.
21. E. Witrant, C. Canudas-de-Wit, D. Georges, and M. Alamir, "Remote stabilization via time-varying communication network delays: Application to TCP networks," in *Proc. of the IEEE Conference on Control Applications*, Taipei, Taiwan, Sep. 2004.
22. E. Witrant, C. Canudas-de-Wit, and D. Georges, "Remote output stabilization under two channels time-varying delays," in *Proc. of the* 4^{th} *IFAC Workshop on Time Delay Systems*, Rocquencourt, France, Sep. 2003.
23. V. Van Assche, M. Dambrine, J.-F. Lafay, and J.-P. Richard, "Some problems arising in the implementation of distributed-delay control laws," in *Proc. of the 38th Conference on Decision and Control*, Phoenix, Arizona (USA), Dec. 1999.
24. S. Mondié and W. Michiels, "Finite spectrum assignment of unstable time-delay systems with a safe implementation," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2207–2212, Dec. 2003.
25. *Tmote Sky Data Sheet*, Moteiv, San Francisco, CA, 2006. [Online]. Available: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf.
26. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2006. [Online]. Available: http://www.ieee802.org.

Chapter 7
Localization in Wireless Sensor Networks

Steve Huseth and Soumitri Kolavennu

Abstract Within wireless sensor networks, the location of the sensor can be a critical attribute of the data sensed. A device knowing its relative position to the other devices in the wireless network or its absolute position with respect to a map increases the value of the data sensed. Sensing the precise location of wireless devices has been a topic of serious study for some time. A number of methods and techniques have been suggested, using various physical attributes (IR, acoustic, time of flight); however, we have chosen to emphasize received signal strength (RSS), which is simple to implement and uses the signal strength sampling capability used by the network layer of a wireless network. Using algorithms such as *fingerprinting* and *multilateration*, a reasonably accurate position estimate can be established depending on the density of the network devices and the environment in which they must operate. We discuss the previous work in RSS localization and describe the limitations of these algorithms from a practical and an experimental perspective. We will conclude with a discussion of our work using these algorithms in 3 wireless network test beds constructed in an industrial refinery and an office environment.

Keywords Signal strength localization · RSSI-based localization · Multilateration · Signal strength propagation

7.1 Introduction

A wireless sensor network consists of a number of low-power, low-cost sensor nodes, tethered together using a wireless network. Sensors collect information about the environment and exchange information with neighbors, and the information is relayed ultimately to external sink nodes. In addition to the data sensed by the wireless sensor device, its location in the wireless network can also be a valuable attribute. This location can be either *relative* or *absolute*. Relative location refers to devices positioning themselves with respect to neighbors one or more hops away.

S. Huseth (✉)
Honeywell ACS Labs, Golden Valley, Minnesota, USA
e-mail: steve.huseth@honeywell.com

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_7,

In terms of the OSI communication model, relative location can be used at multiple layers. With the link layer and physical layer, localization and distance information can be used for transmit power control [20]. At the routing layer, sensors aware of their relative position can improve routing efficiency by selective flooding or forwarding packets in the physical direction of the device that is closer to the destination [3, 12, 26]. Within the network layer, shortest path routes require relative position of the devices in the network [4, 11, 17]. In addition, the localization and positioning of router nodes in mobile ad hoc networks is useful in maintaining connectivity and increasing robustness of the network.

Absolute location refers to positioning each device with respect to fixed reference points or external coordinate systems such as a map. Knowing the absolute location of the device further enhances the value of the sensor data available within the network. Logical and physical security can be linked through precise localization information with respect to physical barriers [1]. A wireless sensor network for monitoring ambient temperature in an agricultural field created through a random dispersal of devices can only sense temperature values over the entire field identifying maximum, minimum, and average temperature. If each sensor also knows its absolute position with respect to a map, excessively hot or cold spots can be identified. A map containing temperature gradients can be created allowing treatment of specific sections nearing maximum or minimum temperatures rather than the entire field.

The application layer contains some of the most widely recognized uses of absolute location in wireless sensor networks. This includes identifying soldiers in a field, locating equipment in factories, and tracking people in buildings [9]. Self-positioning devices in factories or industrial plants reduce the total cost of ownership through enhanced commissioning based on installed position and simplified determination of where a sensor that requires repair or replacement is located. For example, within an HVAC system, auto-locating wireless thermostats are able to determine their respective control zones, simplifying installation and maintenance. Sensor failures can be handled by deferring to the next nearest thermostat enhancing fault tolerance of the entire system.

7.2 Localization Technologies and Approaches

Technologies for determining the precise location of wireless devices have been a topic of serious study for some time resulting in a number of methods and techniques being suggested. These techniques employ one or more physical attributes of the wireless medium, such as received signal strength (RSS) [15], angle of arrival (AoA) [6, 19, 22], and time of flight (TOF) [13]. Each technique offers differing levels of additional hardware cost and consistent accuracy. Many non-RF approaches have been suggested such as IR and acoustic and have demonstrated significant levels of performance.

The *Active Badge* system [33] describes a location tracking system using an IR badge worn by a person that emits an IR signal containing a unique ID. IR receivers placed at known locations collect the beacon message and forward the ID to a centralized location engine. The system achieves a coarse level localization of locating a person or asset within a room or vicinity that is within range of an IR sensor. Although the location technology is available at very low cost, it requires installation of a costly IR location infrastructure separate from the wireless sensor network. Also other physical challenges remain of scaling poorly due to limited IR range and poor performance in the presence of direct sunlight.

Active Bat [8] introduces ultrasound and the ability to measure the acoustic beacon time of flight. Tags consist of an RF receiver and an ultrasonic transducer. At periodic intervals, a base station transmits an RF message containing the unique identifier for a single tag. The tag receives the message and emits an ultrasonic pulse. Ultrasonic receivers in the vicinity note the time of the RF message from the base station and the time-of-arrival of the ultrasonic pulse from the tag. Using the speed of sound in air (adjusted for the ambient temperature), the distance to each ultrasonic receiver can be estimated and forwarded to the base station. With limited hardware, the performance of the system is impressive achieving 9 cm accuracy for 95% of the samples taken. The principal challenge of the system is again the infrastructure cost of setting up the ultrasonic receivers. Since ultrasound does not pass through solid structures such as walls and floors, multiple receivers must be placed in each room to enable the trilateration calculation from at least three receivers.

GPS receivers are probably the most common method for locating people or devices. This localization approach is based on a pre-existing infrastructure of deployed satellites. This essentially makes the cost for the absolute positioning infrastructure to be free. However, integrating GPS receiving capability into low-cost wireless sensors can still be cost prohibitive. Furthermore, GPS signals are not dependable inside buildings, under dense foliage, or in highly metallic environments of plants and industrial facilities.

Angle of arrival (AoA) uses an antenna array and the ability to precisely measure the RF wave front being received at each antenna element [6, 19, 22]. Due to distortions in the signal caused by multipath reflections and interference, a precise angle of inflection can be very difficult to determine. A measurement error of a few degrees is magnified as the separation distance increases. However, combining AoA with a distance measurement capability allows a position estimate from a single receiver reducing the amount of infrastructure that must be deployed.

Time-of-flight techniques such as time of arrival (TOA) and time difference of arrival (TDOA) are perhaps the most widely used location techniques in general communication networks. However, their applicability to wireless sensor networks suffers similar disadvantages as AoA; the nodes require very fast and accurate clocks and tight time synchronization resulting in costly hardware additions to the wireless device [13]. Both techniques require at least three fixed receivers at precise known locations to determine an accurate position of the device at the unknown position. It should be noted, however, that sensor networks physical layers based on ultra wide band (UWB) are though expensive becoming popular and UWB-based sensor networks can localize nodes using TOA and TDOA techniques [5].

Due to the high hardware cost of these approaches and need for a separate dedicated location infrastructure to the wireless network, we will not be covering these approaches further in this discussion.

The most accessible and lowest cost method for localization in wireless sensor networks is sensing RSS. All wireless networks are dependent on measuring the signal strength exchanged between devices to develop network routes and maintain connectivity within the network. Hence, the RSS is a readily available measurement in every wireless sensor node. This same mechanism forms the basis of RSS-based localization.

RSS-based approaches are well suited for use with wireless sensor networks. The location approach used is subject to a number of similar constraints shared by wireless sensor networks in general. The approach must be power efficient supporting battery powered operation for extended intervals. It must be robust to local interference and packet drops to maintain the integrity of the location system. Finally, it must provide sufficient repeatable accuracy for the application domain being supported. As the cost of wireless devices is reduced, the labor to commission each device with its associated location becomes unacceptable requiring devices to automatically identify their own position. In wireless networks, a small number of select nodes in the network can have known fixed positions and provide the reference infrastructure for all the other devices with unknown positions. As we will discuss, using the signal strength sensing capability of any wireless device that is able to route messages in a wireless network, simple location estimates can be produced with little additional cost. However, for applications requiring greater accuracy or different location sensing technology, a separate infrastructure may be required. In such cases, the cost of the infrastructure can be the dominate contributor to the overall cost of the location system.

7.3 Signal Strength Localization

A wireless network capable of determining the position of its members consists of a combination of devices at known positions called *anchor nodes* and other devices at unknown positions referred to as *blind nodes*. Each blind node listens for periodic transmissions from anchor nodes in the vicinity and uses the RSS from three or more to calculate its position in 2D. The process can be repeated with the blind nodes now knowing their position and becoming anchor nodes supporting the positioning of other blind nodes. As we will discuss, each position estimate contains some error. When the process is repeated, this error is accumulated in the subsequent localization operations.

RSS-based location algorithms generally fall into one of two categories of *multilateration* and *fingerprinting*. These techniques differ in the effort required to set up and commission the system and the ability to be adapted to complex and changing environments.

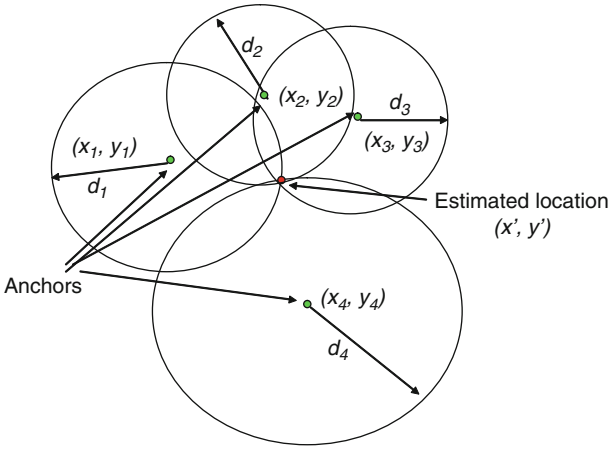


Fig. 7.1 Multilateration

Multilateration (an extension to trilateration utilizing three or more measurements) is simple to implement and operates well for many applications. Within a wireless network, the distance to three or more nodes at known positions can be used to estimate the position of the node with the unknown position. In two dimensions, each blind node listens for periodic transmissions from three or more anchor nodes calculating the distance d_i to each anchor as shown in Fig. 7.1. Assuming all the distances are accurate, the locus of the position of the blind node is the intersection of the circles.

The difficulty of this approach is calculating the precise distance d_i from the RSS. The calculation reduces to finding the most probable location of the blind node given the distance estimate from a set of known points (anchors) to the blind node. The distance estimate could be obtained from a number of sources including TOF or RSS. The distance can also be estimated from RF signal propagation models or through experimental sampling of the environment to record the RF signal strength at various distances. The signal strength path loss follows the relationship $L(d) \propto d^n$, where $L(d)$ is the path loss at distance d. The path loss will increase exponentially according to n. In the free space model $n = 2$, however, as we will see, n can vary significantly based on the properties of the environment. The free space propagation model must be adjusted for many aspects of the environment, such as the specific building materials used and the number of walls the wireless network is operating in. These calculations can be exceedingly complex. Fading and interference, accuracy of the transmit power being emitted, the RSS being measured, and the orientation of the transmitting and receiving antenna are all factors that affect this measurement.

Determining a single value of n can be possible for environments with open or uniform topologies, where the RF propagation is relatively consistent throughout the building or facility. The actual position estimation calculation can be mathematically cast as a maximum likelihood estimate (MLE) or a minimum mean squared

error estimate (MMSE) [19, 21]. Given the uncertainty in the distance estimate and determining the best value for n, it is possible to also estimate the best case accuracy that can be expected through the Cramér-Rao lower bound (CRLB) calculation [21] creating uncertainty bounds for the position estimate rather than a single point.

The fingerprinting approach [10] addresses some of the variability caused by the environment by replacing the continuous distance estimation with discreet sampling at selected sample points. Sample points are selected that are representative of the RF propagation within the environment and nominally will include the distortions due to multipath fading, interference, and radio hardware limitations. During a "training phase," signal strength measurements from each anchor at each sample point are collected along with a truth value of the actual location. This creates a database of signal strength measurements expected at each sample point. Assuming limited changes to the physical environment, the RF signature, i.e., the "fingerprint," should be the same over time such that when a blind node is at or near the sample point position, it will receive the same signal strength values from each of the anchors matching the fingerprint. In reality, physical environments are constantly changing by time of day, movement of people in the vicinity, and relocation of equipment and furniture. Capturing a large database of sample points can be labor intensive and must be repeated as a result of significant changes to the topology of the building.

In this paper, we will discuss these RSS-based location approaches and various hybrid strategies blending the concepts of fingerprinting and multilateration. We will conclude with a discussion our own experiments in several different physical environments and using different radio networks.

7.4 Related Work and Background

Location sensing using signal strength has been widely suggested as a location sensing method. It is the method of choice over other techniques due to its simplicity and dependence on minimal hardware support making it cost effective for many applications.

Shrivastave [29] uses a minimalistic approach to signal strength location, where a very simple model of RSS-based location tracking is proposed. Shrivastave shows that location can be established using an array of binary signal strength sensors as location anchor nodes. Each sensor outputs a 1 when the blind node is within its sensing range and a 0 when it is not. Using the trajectory of the blind node and probabilistic analysis, the performance of such a system can be shown to follow the formula of $1/\rho R$, where R is the sensing radius and ρ is the sensor density per unit area. R is dependent on the signal propagation model for the environment, while ρ represents the discrete sampling of signal strength. These same concepts of sensing radius and sensor density reappear as a measure of accuracy in all the other signal strength techniques we will discuss.

Bahl and Padmanabhan [2] performed some early work, comparing several empirical and model-based signal strength approaches using the RADAR system. An empirical fingerprinting approach is described, which consists of an offline sample point collection, followed by several matching approaches. During the offline phase, each sample point consists of collecting an RSS n-tuple (where n is the number of anchor points) of averaged RSS measurements along with the sample point location. During the online phase, the RSS n-tuple is recorded for an unknown location and a technique referred to as *nearest neighbor in signal space* (NNSS) selects the closest reference point by computing the Euclidean distance measure between current n-tuple's RSS values and the recorded values from the reference point. In collecting the sample point data, variability in the data was noted particularly with respect to antenna orientation. To address this, multiple RSS n-tuples were collected at each sample point corresponding to each orientation and included as separate sample points.

Recognizing the labor involved in sample point collection, Bahl also describes a model-based approach, where the offline sample point collection phase is replaced with an automated sample point generation approach. A path-loss model is used to calculate signal strength values for intersection points on a grid that are used as reference points. The same sample point matching approach is then used. Bahl found the empirical approach to be superior and was able to demonstrate a 2–3 m accuracy within a typical office environment and an approximately 30% improvement in the location resolution over the model-based approach. It was noted that while the empirical method was superior in terms of accuracy, the model-based approach was significantly easier to deploy. Similar to the results noted by Shrivastave, the accuracy is determined by a combination of the fidelity of the signal propagation model and the sensing density.

Collection of a large set of sample points for a large network can be cost-prohibitive from a labor standpoint, leading to hybrid approaches. Gwon and Jain [7] outline an approach for using the signal strength measurements between anchors as a mechanism for developing the signal strength to distance mapping algorithm without additional sampling. Using this approach, the system takes into account the placement and performance characteristics of each anchor to generate a different mapping function for each one. Li et al. [14] suggest another hybrid approach of using a two-step process of locating the room using fingerprinting, then using multilateration to find the mobile within the room. The assumption is that fingerprinting can perform the coarse localization with a smaller number of sample points to obtain position down to the room level. Multilateration is better behaved in open rooms with limited obstacles and can be used to perform the fine-grained localization.

Savvides et al. [25] suggest an iterative approach using a signal strength model to define a relative position map of the nodes in the network called *collaborative multilateration*. The approach uses a distributed algorithm for determining relative location among each node in the network. Nodes calculate relative distances to each of their neighbors and establish a rough connectivity graph. As graphs are shared among neighbors, a larger graph of the entire system is created showing connectivity among all the devices in the network. The algorithm is repeated with

each node adjusting its calculated position based on inputs from neighbors. The algorithm continues to repeat until some minimum position delta is achieved by each node. This algorithm assumes a reasonably static network, where nodes have minimal movement that would inhibit convergence of the algorithm. Developing a complete topology of all the devices requires reasonably accurate distance measurements and reasonably high connectivity among the nodes in the network to avoid graph ambiguities such as inversions and reflections.

7.5 Accuracy and Performance in RSS-Based Localization

Location error can be described as a combination of *intrinsic* and *extrinsic* errors [24]. The intrinsic component consists of variability in the radio design, antenna orientation, and accuracy of the signal strength measurement by the receiver. Extrinsic errors are a result of the physical environment resulting in multipath fading, interference, and shadowing. In experimental environments, a number of measures can be taken to minimize the impact of both intrinsic and extrinsic errors however; such control is often not realistic in fielded systems. Systems will be installed in environments that cannot be controlled or fully anticipated by the network designers. Consequently, operational systems must be designed with as much automatic self-configuration as possible. Control of intrinsic and extrinsic errors can be improved using two approaches: (1) building improved models of the signal propagation characteristics within the environment and (2) increasing the sensor density. Increasing sensor density is costly; however, it essentially mitigates errors in the signal propagation model by providing a larger number of discrete reference points for making the location estimation. We will discuss each of these factors in the following sections.

7.5.1 Multilateration and the Signal Strength Propagation Model

A number of approaches have been used to develop accurate signal propagation models. These range from continuous models, which estimate signal propagation to discrete models, which use curve fitting against a set of calibration sample points.

Using analytical means to estimate the distance of the mobile node significantly reduces the effort required to deploy operational systems. Both analytic and experimental approaches have been suggested for deriving this model. Analytical approaches such as ray-tracing [32] or detailed radiosignal propagation models [30] have been suggested. Typically, these approaches attempt to include material absorption and signal reflections, which work well for large objects, such as walls and machinery, but do not take into account smaller features, such as furniture [28]. It is nearly impossible to include all the dimensions of variability in the channel that

must not only include variations in the physical space but must also account for antenna orientation and radiation patterns, and uniformity of transmit power and receive sensitivity that can differ from radio to radio.

In its simplest form in free space, the signal strength from an RF emitter degrades exponentially with distance following the physics of wave propagation according to $(Pr/Pt) = G_t G_r (\lambda/4\pi d)^2$, where P_t is the transmission power of a signal at the transmitter and P_r is the received signal strength (RSS) at the receiver, G_t and G_r are the gains on the antennas at the transmitter and receiver, λ is the signal wavelength in meters, and d is the distance in meters separating the transmitter and receiver. Converting this to decibels (dB) and assuming no antenna gain we have the path loss described as

$$P_t(\text{dBm}) - P_r(\text{dBm}) = L\ (\text{dB}) = 2 * \left(10 \log_{10}\left(\frac{4\pi}{\lambda}\right) + 10 \log_{10}(d)\right) \quad (7.1)$$

Additional factors of attenuation from the signal passing through walls and floors, and scattering due to reflections need to also be included. Interference, scattering, and reflections increase the path loss exponentially over the distance traveled requiring a path loss exponent different from the factor of 2 shown. In corridors, the wave guide effect might result in an exponential value lower than 2. More typically, a cluttered factory or office environment will have an exponential value much greater than 2. Seidel and Rappaport [27] calculate the mean path loss exponent for several building environments showing this value varies from 1.8 to 5.0.

Seidel also suggests adding a constant path loss due to attenuation through significant obstacles, such as floors adding a *Floor Attenuation Factor* (FAF). Bahl and Padmanabhan [2] extend this concept adding the *Wall Attenuation Factor* (WAF) and is able to reduce the variability in the data by accounting for each wall. This produces improved results up to a threshold where the exponential attenuation loss becomes the dominant element. Within a location system, this also assumes some level of a priori knowledge of the location of the blind device to calculate the number of walls and floors the signal is passing through. Other intrinsic losses can also be added, such as losses due to the receiver or transmitter antenna, physical enclosures, and mounting orientation. Combining both intrinsic and extrinsic factors for the RSS and assuming a transmit power P_t of 0 dBm we have the formula:

$$P_r\ (\text{dBm}) = \alpha + \beta \log_e(d) \quad (7.2)$$

The two key parameters α and β represent the extrinsic factors of the environment and intrinsic factors of a radio. These values must be estimated for the model to be customized to an environment and radio design used. Note that the RSS in dBm is usually available as a measure at each radio.

Texas Instruments, Inc defines these terms in the CC2431 System-On-Chip (SOC) [31], which includes a built in location engine. The chip must be programmed with the location of up to 16 anchor nodes in the wireless network, which are used in the position calculation of the blind node. Texas Instruments, Inc suggests that the

values for α and β be calculated empirically. α is calculated as the mean RSS at 1 m distance. β is determined by collecting a number of RSS samples each at various known distances in the environment. The sample points should be collected from various distances and various locations around the site. The slope of a least squares fit of a logarithmic curve provides a reasonable estimate for the value of β.

Assuming some of the nodes in the network have fixed known positions, the position of blind nodes can be determined through minimizing the weighted mean squared error between the Euclidean distance between the anchor nodes and the blind node. The distance is calculated from the path loss formula (2) and solving for d resulting in:

$$d_i = \beta e^{-\alpha S_i}, \tag{7.3}$$

where S_i is the measured signal strength in dBm between the fixed node and the blind node.

Estimating the position of the blind node is equivalent to solving the following weighted least squares optimization problem.

$$\underset{Xm,Ym,Zm}{\arg\min} \sum_{i=0}^{n} w_i(s_i) \left(\sqrt{(X_m - X_i)^2 + (Y_m - Y_i)^2 + (Z_m - Z_i)^2} - d_i \right)^2, \tag{7.4}$$

where (X_m, Y_m, Z_m) is the location of the blind node to be calculated, (X_i, Y_i, Z_i) are the positions of each of the anchor nodes and d_i is the estimated distance to that anchor. We have added the term w_i as a weight on each term. Since less distortion is expected from signals coming from nearer anchors over anchor further away, this weighting should also be proportional to the signal strength received.

α and β in (7.3) determine the environment of signal propagation. The values used for these variables are averaged as they are typically not uniform throughout a given environment. Hallways and open foyers will have different values than a highly cluttered environment with a number of obstructions. However with sufficient sample collection, even a building with a large number of walls and offices may behave satisfactorily if the environment is relatively uniform.

Figure 7.2 shows a sampling of an environment from increasing separation distances between the transmitter and receiver and fitting an exponential curve on the resulting data. Although this does not address differences in signal propagation in sections of the environment not sampled or differences in the transmitters and receivers, it provides a simple mechanism to develop the initial mapping function.

The highest performance of multilateration-based algorithms will typically be in the center of the space being sensed where beacons will be received from most of the anchors. Significant errors exist at the boundaries of the area particularly when calculating the position of a blind node outside the polygon created by the anchors referred to as the *convex hull*. Performance of the system falls off rapidly outside the convex hull, particularly where minimization of the distance estimates are used, causing the estimate to drift to the center of the strongest anchor measurements. The fingerprinting approach has a distinct advantage to address these boundary conditions if sample points are created outside the convex hull.

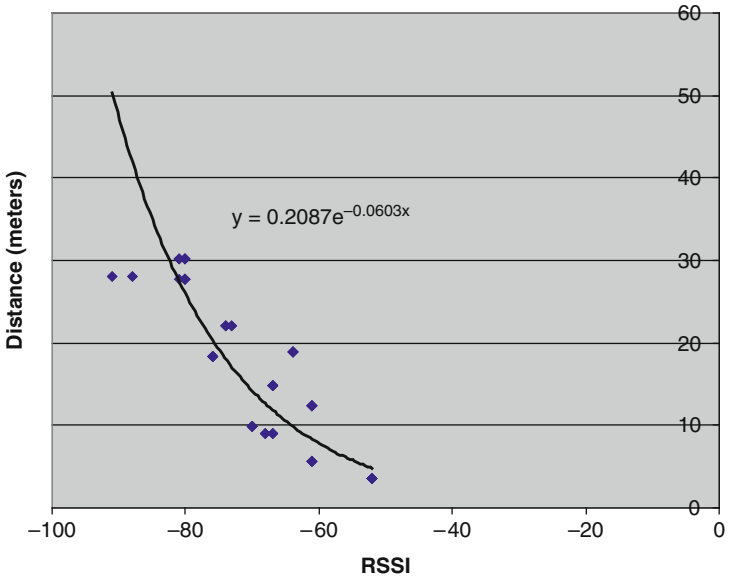


Fig. 7.2 Curve fitting of signal strength vs. distance

7.5.2 Fingerprinting Approaches

Where multilateration depends on a continuous function for estimating the signal strength behavior, fingerprinting addresses the nonlinear behavior by using a number of sample points to measure the RF behavior at specific locations throughout the environment. A number of algorithms have been suggested [2, 15] for matching the received RSS values with the RSS values in the database. The generalized distance between the two vectors is determined by:

$$Z = \sqrt{\sum_{i=0}^{n} w_i \, (S_{obs} - s_i)^2} \frac{1}{c}, \tag{7.5}$$

where S_{obs} is the vector of observed signal strength values from each anchor, s_i is the corresponding signal strength values at the sample point collected during the training phase, and c is the number of anchor signal strength values received. As in multilateration, a weighting w_i [16, 23] can be applied that is proportional to the expected signal strength value s_i. This formula corresponds to the Euclidean signal distance between the collected signal strength data measured and the sample point data in the database. Minimizing the signal distance produces the sample point with the closest match and is referred to as the nearest neighbor algorithm (NN) [2]. Other improvements have been proposed to average the position of K of the closest sample points, where $K > 1$ referred to as the K nearest neighbors (KNN). Although this

can improve the results where the density of the sample points is relatively high, averaging a small number of sample points that are distant from the actual position can increase the error in the position estimate. In such cases, the NN approach will have the best performance.

A characteristic of the fingerprinting algorithm is the dependence on the availability of the same anchor nodes during both the online and offline phases. The loss of an anchor due to failure or interference will affect the matching algorithm. Since the lack of a signal from an anchor can be due to a number of nonlocation dependent factors such as node failure or temporary interference, it is difficult to include that information into the fingerprint matching algorithm. Instead, we have chosen to factor the algorithm by the number of anchors responding. This is generally successful when a small number of beacon messages are lost. However, in environments with a large number of anchors, the set of anchor beacon messages received may change over time, which will not match the fingerprint captured. In such cases, even though messages from a number of anchors have been heard, the lack of a fingerprint match will result in significant location errors. In this case, the multilateration algorithm will be more successful as it is able to use whatever anchors are available.

7.5.3 Location System Setup and Configuration Challenges

In fielded systems, the most common failure mode is incorrect or insufficient setup and configuration. Poor placement of the anchor nodes or misalignment of the antennas can have a significant effect on the performance of the system. The placement of the tag on the person or equipment also can attenuate the RSS in unpredictable ways. Lin et al. [18] suggest that antenna orientation alone can affect the localization error by 1–3 m.

As we have discussed, some level of site survey is required to characterize the environment, which consists of either sampling the environment at different anchor distances to develop the values for α and β or creating the database of fingerprint sample points.

Also required as input into either algorithm are the positions (x, y, z) of the anchor points and reference points. RSS-based localization provides a coarse localization and as such the precision in positioning the anchors should be at least 10–20% of the accuracy desired for the blind nodes. A significant area of research that still remains is validating the correctness of the site survey data to achieve this precision. Typically, the fixed anchor devices are placed by hand on an X–Y grid with respect to a digital map. It is assumed the digital map is accurate with respect to the accuracy desired and has not been distorted or stretched resulting in placement errors. Also the maps are reasonably current with sufficient detail to allow accurate placement of the anchors or sample points. Marking the exact location of an anchor device on a map where the device is placed in a large room can be quite error prone. Without careful measurement of each reference point or the ability to place the device with respect to specific map features, significant placement errors can

result which propagate into each location solution. Finally, placement of anchors on multiple floors of a building requires careful registration of each floor with respect to a common origin. Maps for different floors may not be constructed with the same scale or may do not completely overlap the floor below or above. Determining exact position of the anchors and sample points on overlaid maps with a high degree of precision can be challenging without the benefit of a more complete 3D model.

7.5.4 Blind Node Centric vs. Anchor Node Centric Architecture

Location algorithms can be implemented either centrally where the blind node transmits the beacon message or distributively where anchors send the beacon message. In the centralized approach, the beacon from the blind node is received by the anchors that in turn forward the signal strength information to a central location server. The centralized architecture is most common among commercial applications which are able to support a large central location server and typically require low-cost blind nodes. It is the easiest to implement and control as it allows the blind node to be as simple as possible transmitting the beacon at a periodic rate. It is possible that this may cause overloading of the location server in the case of large numbers of blind nodes and taxing the communication infrastructure. This may be addressed by providing multiple location servers in the system together with a coordinator. Individual blind nodes are assigned to location servers by the coordinator to ensure load leveling among the location servers. Each anchor is responsible for forwarding the RSS data received from a blind node to its assigned location server. This allows the system to scale as the number of blind nodes increase.

The distributed approach reverses the communication: the beacon is sent by the anchor and received by the blind node. This increases the power requirements and complexity of the blind node, requiring it to be operational continuously to listen for beacons and process the updates to calculate the current position. However, multilateration algorithm can be reduced to an approximate linear least squares estimate problem for this computation on low footprint devices. This approach minimizes the complexity and power requirements of the anchors, which only transmit a simple beacon periodically. The simplicity of the anchors may also cause the anchor beacons to collide and not be received by the blind node without some additional level of synchronization.

Multilateration is more suitable for the distributed location, where the anchor position and environment-specific information (α, β) must be available at the blind node. The location of each anchor can be broadcast as part of the beacon message from the anchor. Knowing the signal strength and location of each anchor, the blind node can calculate its position and display it locally or relay the information to the appropriate user interface. Fingerprinting is best suited to the centralized approach to minimize the duplication of the fingerprint database at each blind device.

The selection of central vs. distributed operation is a critical design choice of the location system within the wireless network. The application domain frequently has a significant impact on this choice, which is often dictated by the cost and power requirements of the anchor and blind node. Some applications require mobile devices to determine their own position for purposes of privacy or security. For example, soldier tracking requires stealth operation, where no signals such as the location beacon can be transmitted by the soldier. Others require the mobile tag to be optimized for power resulting in minimal tag functionality. Asset and people locating in a hospital is an example requiring minimal tag functionality.

7.5.5 Anchor Separation and Sources of Location Error

In essence, the accuracy of both multilateration and fingerprinting approaches are dependent on the density of the anchors that are placed in the site. Both algorithms are dependent on a consistent RF channel. Increasing the distance between the anchor and the blind node will result in increased signal distortion. Consequently, increasing the number of anchors, especially anchors in different orientations, will improve overall location accuracy over increasing the number of sample points collected in the space.

The simplest method of estimating anchor density is by dividing the size of the space to be monitored by the number of anchors. This assumes a uniform placement of anchors, which is frequently not possible. In many environments, anchors cannot be placed uniformly due to physical constraints such as mounting access or availability of power. When anchors are not placed uniformly, the expected average performance of the entire system is affected by low performance in areas where the anchor placement is the sparsest. A method to estimate this expected difference in performance across the entire facility is possible by dividing the area into the Voronoi partitions. Each Voronoi cell consists of all the points closest to a particular anchor. The size of the largest Voronoi cell provides an estimate of the worst case expected performance.

We have developed a normalized accuracy measure that has worked well principally in 2D for a given setting based on factors of the size of the area of localization and the number of anchors. Our metric establishes the average localization error to be a function of a characteristic separation between the anchors. In fairly uniform anchor spacings, this reduces to the average distance between any two anchors and is calculated by $\sqrt{A/n}$, where A is the area of the convex hull formed by the anchors and $n$is the number of anchors. In the various tests that we have done, we have found that the mean accuracy of RSS-based multilateration estimates range from around 15% in highly uniform environments with uniform anchor spacing, to 35%, where the environment is more challenging.

7.6 RSS-Based Approaches: Experimental Results

We have developed three location test beds to validate our understanding of the underlying issues of signal strength location. In each case, the performance of the location system was validated by randomly generating test points within the network and collecting the estimated location position together with the actual position. Statistical analysis of each data set produced performance metrics allowing a comparison of the computed result with the ground truth evaluating the "goodness" of the approach. This provided the ability to compare results across different environments and different wireless networks.

Our work has been focused principally on understanding the impacts of anchor spacing within different wireless networks and estimating the performance of the weighted multilateration algorithm in different physical environments. The first environment is an outdoor 802.11b/g network in an industrial refinery. The environment allows testing outdoors but contained numerous metal structures. A second test bed also used 802.11b/g access points, which formed the corporate IT wireless network in an office environment. Access points used as anchors are pre-existing and had been placed to provide connectivity throughout the facility rather than for location. This created some challenges in obtaining beacons from at least three anchors at all the test points. However, the performance is impressive considering no investment in the infrastructure was required. The third test bed consists of an array of low power 900 MHz radios equally spaced on a 10 m grid on two floors within an office building.

Each of the test beds was constructed similarly with three principal components: (1) network of stationary RF beaconing anchors at known locations, (2) one or more blind nodes collecting signal strength measurements from the anchors, and (3) a location engine that calculates the position from the signal strength measurements received by the blind nodes.

The analysis of the test beds was conducted similarly. A number of sample points were defined within the test bed. At each sample point, a number of received signal strength indicator (RSSI) measurements provided by the radio were collected from each anchor and the values averaged along with a truth value of the actual location. The averaged RSSI measurements from each anchor were used by the location engine to derive an estimated position that is compared with the actual position. A statistical analysis of the errors produced was used to characterize each test bed. Each test bed will be discussed in the following sections.

7.6.1 Industrial Refinery Wi-Fi Network

Location of people and assets within industrial facilities is becoming more accessible with the availability of pervasive IEEE 802.11b/g networks that can be used for location and tracking. We conducted a number of tests in an active industrial

refinery consisting of a large outdoor facility with large metal structures throughout. Although open and outdoors, the large metal structures represented significant obstacles to uniform RF propagation. In addition, one of the requirements of this test facility was the ability to track people and assets in continuous 3D space; however, we will present only the 2D results to allow comparison with the other test beds. A total of 69 test points were defined within the facility where signal strength measurements were collected from each of the 11 IEEE 802.11b/g access points. The access points were from two different manufacturers and had different performance characteristics. Consequently, different signal strength-distance mappings were created for each type. Three tags were used for testing, consisting of two small 802.11 transceivers – one embedded as a belt clip and the other attached to a helmet, and a third tag that was a handheld device with a built-in 802.11b/g receiver. Each tag collected at least 10 beacon messages from each access point, which were averaged together to produce the location estimate.

The entire facility was approximately 17,000 m^2 in size. The 11 access points were used also as the communication infrastructure to send the RSS information to a centralized location node connected to the network. The location estimate was calculated using weighted multilateration at the central node. A set of RSSI readings taken at various distances was used to calculate the values of α and β and account for extrinsic factors in the environment.

RSSI-based location was estimated at 34 distinct points in the industrial setting. The test points and access points are shown in Fig. 7.3.

The results of the testing showed that the average error was 12 m in 2D for all points within the convex hull formed by the access points. The difference between the three tags was not significant as similar results were observed at each location whether the tag was on the waist or on the helmet or handheld. The average inter-anchor point density was calculated to be $\sqrt{17{,}000\,m^2/11} = 39.3$ m giving a normalized error of $12/39.3 = 30.6\%$. The results are shown in Fig. 7.4.

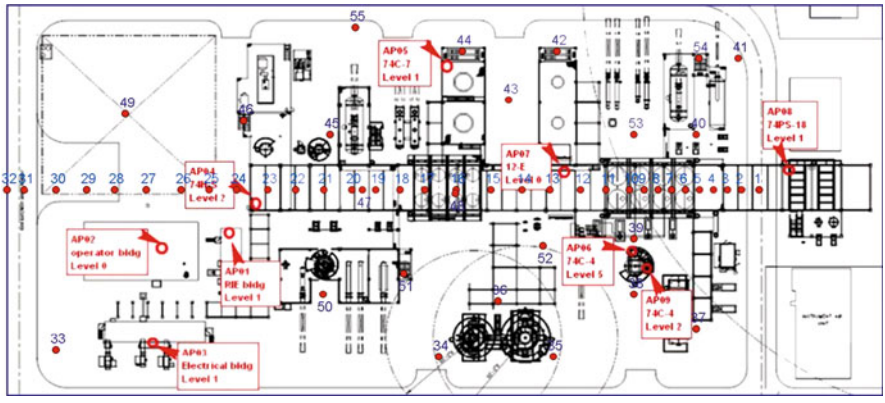


Fig. 7.3 Anchor points and test points in the industrial test bed

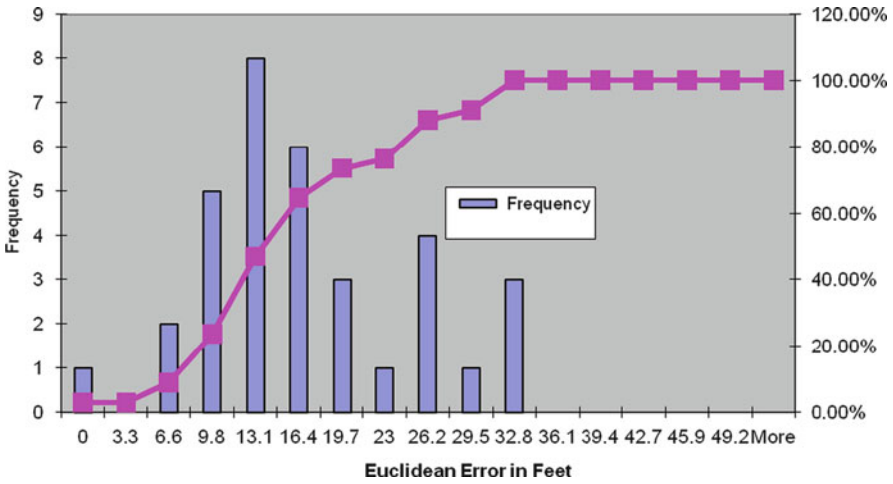


Fig. 7.4 Error distribution of test points

Table 7.1 Location test results for office environment

Test point	Error (meters)
1	0.85344
2	4.23672
3	6.266688
4	2.977896
5	6.501384
6	2.14884
7	4.13004
8	3.197352
9	2.52984
10	2.325624
11	25.2984
12	0.908304
14	6.15696
15	2.551176
16	2.176272
17	5.596128
Average error	4.87

7.6.2 WiFi Building Radio Network

A second test bed was configured in a large office building using the 802.11b/g wireless IT infrastructure. The environment consisted of 14 anchors within an area of 5,794 m^2 as shown in Fig. 7.5.

Each anchor point was a commercial off the shelf 802.11b/g access point and the blind node was a Wi-Fi device with an Intel 802.11b/g wireless interface. Within the space, 17 sample points were selected where the signal strength from each of

Fig. 7.5 Office environment test bed with anchors

the anchors could be captured. Although the access points were installed by the local IT department for Wi-Fi coverage, the same infrastructure was used as the location anchor points. As with the industrial test bed, the Wi-Fi device collected the RSS information from all anchors and sent the information to a centralized location engine. The distance vs. signal strength relation was established empirically. The weighted multilateration algorithm was used to provide the location estimate.

This test bed contained greater anchor point density compared to the industrial test bed and hence the absolute error was an average of 4.87 m. The average distance between the anchor points in this test bed was $\sqrt{5{,}979\,\mathrm{m}^2/14} = 20.6\,\mathrm{m}$. In addition, the requirement for this testbed was to determine the 2D location only. This test bed demonstrated a higher level of normalized accuracy of $4.87/20.6 = 23.6\%$ over the 30.7% found in the industrial refinery test bed discussed earlier. This difference is assumed to be principally a result of the greater uniformity of the walls and obstructions that were in the building environment. The test environment consisted of sheetrock interior walls with metal framing studs with limited denser supporting structures. As can be seen in Fig. 7.5, the Wi-Fi anchors were reasonably uniformly spaced within the test area. The results of the testing are shown in Fig. 7.6.

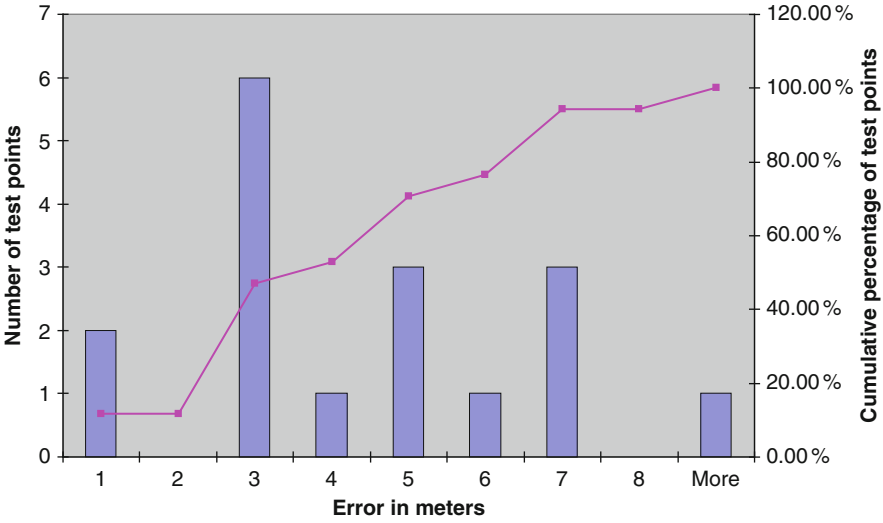


Fig. 7.6 Error distribution for indoor test

7.6.3 Building 900 MHz 802.15.4 Radio Network

In the third testbed, we attempted to validate our accuracy metric using consistent anchor spacing rather than a calculated average spacing. We assembled a wireless network of devices in an office area using regular spacing of the anchors on a 10 m grid as shown in Fig. 7.7. A total of 21 anchors were attached to the ceiling in the 2,044 m^2 office area. The area is open with cubicles at a height of 1.5 m. Each anchor consisted of a battery powered radio using a Texas Instruments Incorporated CC1101 radio transmitting a beacon message at 914 MHz. Each beacon contained the unique identifier for the anchor. Anchors were not synchronized, leading to some message collisions where anchor beacons were not received. However, this condition was not considered significant due to the very short beacon transmit time of a few milliseconds sent every 2 s. The blind node consisted of the same CC1101 radio board connected to a laptop collecting the RSSI data. A total of 36 sample points were collected from within the test area and used to measure the accuracy of the location system. The location engine used the same weighted multilateration algorithm used in the other test beds.

The testing produced an average accuracy of 1.8 m. With the fixed 10 m anchor separation, the nominalized accuracy is $1.8/10 = 18\%$. The high normalized accuracy achieved in this environment can be explained to be a result of the consistent anchor placement and the lack of obstructions. Since there are no large Voronoi cells, the performance throughout the area was found to be reasonably consistent as shown in Fig. 7.8. Also the lack of solid obstructions minimized distortion of the RF channel.

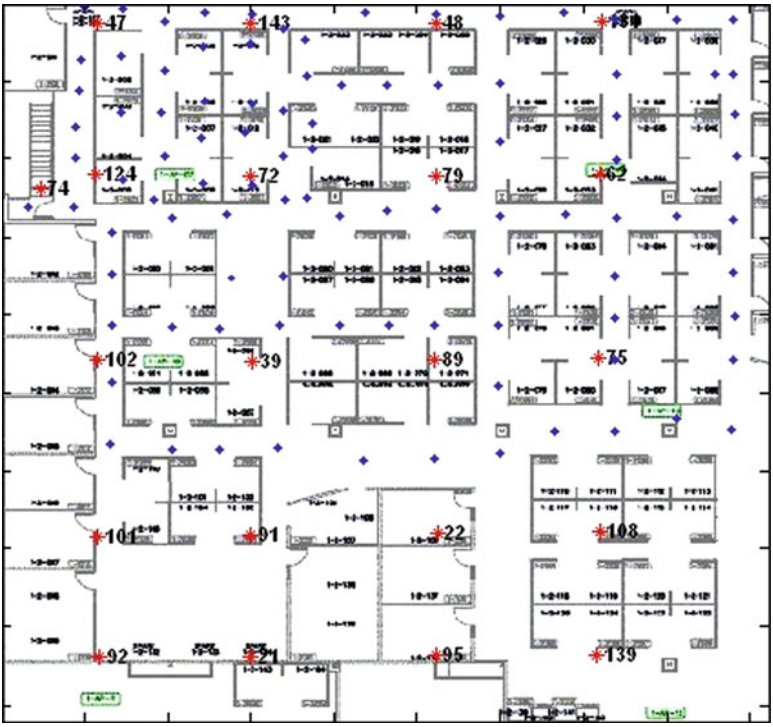


Fig. 7.7 Anchor and reference points

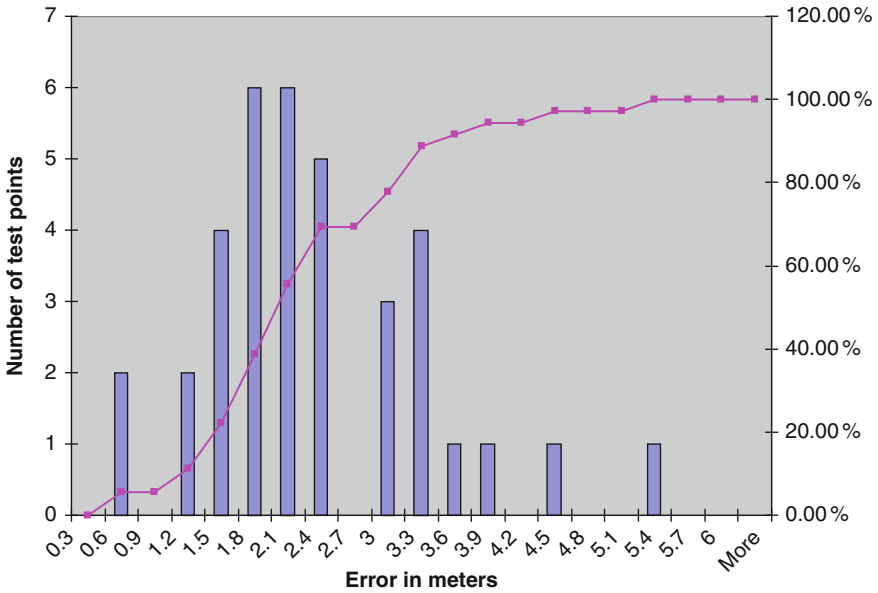


Fig. 7.8 Error distribution for indoor network at 10 m anchor separation

7.7 Conclusions

Advances in RF hardware design and cost reductions in location sensors will continue to improve the locatability of wireless sensors. Localization using measured signal strength provides a simple approach to estimating location with reasonable accuracy as required by most applications. Our experience has shown the accuracy of the signal strength-based approach can be estimated by characterizing the average separation distance between anchors and will range from 15% to 35%. This accuracy range is most directly affected by the uniformity of the anchor spacing as well as the consistency of the environment that results in a uniform channel model.

Our interest is in understanding the accuracy limits particularly with respect to estimating performance of the system in new environments. This metric we have suggested provides some basic guidance on the performance that can be expected. Approaches for enhancing the correct placement of anchors by the installer to ensure the required density and uniformity are key to broader use of these systems and are a principal area of future work.

Acknowledgments Wi-Fi is a trademark of the Wi-Fi Alliance. All other trademarks used herein are properties of their respective owners.

References

1. Ayday, E., Delgosha F., Fekri, F., Location-Aware Security Services for Wireless Sensor Networks Using Network Coding Ayday, 26th IEEE International Conference on Computer Communications, INFOCOM 2007, May 6–12, 2007, pp. 1226–1234
2. Bahl, P., Padmanabhan, V.N., RADAR: an In-Building RF-based User Location and Tracking System, INFOCOM 2000 vol. 2, Mar 26–30, 2000
3. Blazevic, L., et al., Self-Organization in Mobile Ad-Hoc networks: the Approach of Terminodes, IEEE Communications Magazine, vol. 39, no. 6, pp. 166–174, 2001
4. Capkun, S., Hamdi, M., Hubaux, J.P., GPS-free Positioning in Mobile ad hoc Networks, Conference on System Sciences, Jan 3–6, 2001
5. Duan, C., Orlik, P., Sahinoglu, Z., Molisch, A.F., A Non-Coherent 802.15.4a UWB Impulse Radio, in Proceedings of IEEE ICUWB 2007, Sept. 2007
6. Elnahrawy, E., Austen-Francisco, J., Martin, R., Adding Angle of Arrival Modality to Basic RSS Location Management Techniques, ISWPC 2007
7. Gwon, Y., Jain, R., Error Characteristics and Calibration-free Techniques for Wireless LAN-based Location Estimation, MobiWac 2004
8. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P., The Anatomy of a Context-Aware Application, MOBICOM 1999
9. Hightower, J., Borriello, G., Localization Systems for Ubiquitous Computer, IEEE Computer, pp. 57–66, August 2001
10. Kaemarungsi, K., Krishnamurthy, P., Modeling of Indoor Positioning Systems Based on Location Fingerprinting, in IEEE Infocom, March 2004
11. Karp, B., Kung, H., GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, in Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00), August 2000, pp. 243–254

12. Ko, Y.B., Vaidya, N.H., Location-Aided Routing (LAR) in Mobile ad hoc Networks, Wireless Networks, vol. 6, no. 4, pp. 307–321, July 2000
13. Lanzisera, S., Lin, D.T., Pister, K., RF Time of Flight Ranging for Wireless Sensor Network Localization, Fourth Workshop on Intelligent Solutions in Embedded Systems, 2006
14. Li, B., Dempster, A.G., Rizos, C., Barnes, J., Hybrid method for localization using WLAN, Spatial Sciences Conference, Melbourne, Australia, September 12–16, pp. 341–350, 2005c, CD-ROM procs
15. Li, B., et al., Method for Yielding a Database of Location Fingerprints in WLAN, IEE Proceedings, vol. 152, no. 5, October 2005
16. Li, B., Salter, J., Dempster, A.G., Rizos, C., Indoor Positioning Techniques Based on Wireless LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia, March 13–16, 2006, paper 113
17. Li, J., et al., A Scalable Location Service for Geographic Ad Hoc Routing, M.I.T. Laboratory for Computer Science, MobiCom, Boston, MA, pp. 120–130, August 6–11, 2000
18. Lin, T., et al., A Microscopic Examination of an RSSI-Signature-Based Indoor Localization System, HotEmNets Charlottesville, Va, June 2–3, 2008
19. Niculescu, D., Nath, B., Ad hoc positioning system (aps), in Proceedings of Globecom, San Antonio, TX, USA, 2001
20. Pathirana, P.N., Location Based Power Control for Energy Critical Sensors in a Disconnected Network. 2006 IEEE International Conference on Industrial Informatics, 16–18 Aug. 2006, pp. 653–658
21. Patwari, N., O'Dea, R., Wang, J.Y., Relative Location in Wireless Networks, in Proceedings of the Spring 2001 IEEE Vehicular Technology Conference (VTC), Rhodes, Greece, vol. 2, pp. 1149–1153, May 2001
22. Peng, R., Sichitiu, M., Angle of Arrival Localization for Wireless Sensor Networks. 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2006
23. Prasithsangaree, P., Krishnamurthy, P., Chrysanthis, P., On Indoor Position Location with Wireless LANs Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium, vol 2, pp. 720–724, September 15–18, 2002
24. Savvides, A., Garber, W., Moses, R., Srivastava, M., An Analysis of Error Inducing Parameters in Multihop Sensor Node Localization, in IEEE Transactions on Mobile Computing, vol. 79, no. 6, pp. 567–577, Nov–Dec 2005
25. Savvides, A., Garber, W., Adlakha, S., Moses, R., Srivastava, M., On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks, in Proceedings of 2nd IPSN, Palo Alto, CA, 2003
26. Seada, K., Helmy, A., Goyindan, R., On the Effect of Localization Errors on the Geographic Face Routing in Sensor networks. IPSN 2004
27. Seidel, S.Y., Rappaport, T.S., 914 Mhz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings, IEEE Transactions on Antennas and Propagations, February 1992
28. Seybold, J., Electromagnetics and RF Propagation, Introduction to RF Propagation. Hoboken, NJ: Wiley, 2005
29. Shrivastave, N., Mudumbai, R., Madhow, U., Suri S., Target Tracking with Binary Proximity Sensors Fourth ACS Conference on Embdded Networked Sensor Systems (SenSys 2006) 2006
30. Spagnolini, U., Bosisio, A., Indoor Localization by Attenuation Maps: Model-based Interpolation for Random Medium, ICEAA International Conference on Electromagnetics in Advanced Applications. 2005
31. Texas Instruments, Inc, System-on-Chip for 2.4 GHz ZigBee(TM) IEEE 802.15.4 with Location Engine (Rev. B), CC2431 Technical Document, June 15, 2007
32. Valenzuela, R.A., Ray tracing prediction of indoor radio propagation. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 1, pp. 140–144, 1994
33. Want, R., Hopper, A., Falcao, V., Gibbons, J., The Active Badge Location System, ACM Transactions on Information Systems, January 1992

Chapter 8
Counting and Rendezvous: Two Applications of Distributed Consensus in Robotics

Carlos H. Caicedo-Núñez and Miloš Žefran

Abstract Consensus protocols have been widely studied in recent years in the control community. We discuss two applications of consensus protocols in robotics: counting and rendezvous. For counting, the main issue is how each agent can estimate the total number of robots in a network by using limited communications with its neighbors. For rendezvous, the aim is to make the agents converge to a common meeting point, again by only allowing them to communicate with immediate neighbors. We present a formal analysis of the proposed algorithms and prove their convergence properties by relying on the theory of consensus protocols.

Keywords Consensus protocol · Distributed counting · Rendezvous · Robotics

8.1 Introduction

Consensus protocols were introduced by French in 1956 [21], and then revisited by DeGroot in 1974 [17]. In engineering, they were first applied by Tsitsiklis in 1984 [42]. They were then re-discovered independently by the control community with the work of Jadbabaie et al. [27]. Subsequent research inspired by this work led to the continuous time version, presented by Olfati-Saber and Murray [37] and its generalization by Moreau [35]. The reader interested in this type of algorithms is referred to the survey [38] and the references therein. In this chapter, the focus is on the discrete time consensus algorithm.

The problem of achieving agreement between independent agents has been widely studied in the computer science community [4, 24, 33], and it has been only recently that the control community turned its attention to this problem.

In the 1980s, John Tsitsiklis [42, 43] studied the agreement-problem in the context of parallel computing, where several processes try to agree on certain values of interest to ensure the correct operation ofthe network. In 1995, Vicsek et al. [44]

M. Žefran (✉)
Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA
e-mail: mzefran@uic.edu

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_8,

analyzed the problem but from a different perspective: they were interested on how to control a discrete set of *particles* that were traveling at the same speed, so they all share the same direction or *heading*. Arguably, it was not until 2003 that the control community realized the potential of this work. In 2003, Jadbabaie et al. [27] presented a formal analysis of Vicsek's work (which was only empirical). In 2004, Olfati-Saber and Murray [37] introduced a continuous time approach to the consensus problem and in 2005 Moreau [35] presented a generalization for the discrete time scenario. The family of algorithms introduced in the aforementioned works, all have the goal of driving a set of processors (or independent agents) in the network so that they agree on some variable. In [27, 44], the variable is the heading of the agents, and in the case of [35, 37, 42, 43] it is an auxiliary variable associated with each element of the formation. Such protocols are now known simply as *distributed consensus* or *agreement protocols*. Remarkably enough, the discrete time family of protocols introduced during 2003–2005 in the references falls under the problem studied by Tsitsiklis almost 20 years earlier.

One of the reasons why this family of protocols has become so popular is because of their wide range of applications: from motion coordination [27] to parallel computing [42] to the *gossiping algorithms* in communication systems [6] (also known as *aggregation protocols* [28]), all are examples of the uses that such results currently have.

In this chapter, we focus on two applications for distributed robotic networks: distributed counting of agents, and solving the *rendezvous* problem. In distributed robot networks, the way to allocate the resources depend on both the nature of the task and the number of robots in the formation [7, 8]. Henceforth, knowing how many agents are there becomes a crucial aspect for the performance of the system. A central node might perform this task, but if we want a distributed solution, an alternative has to be found. We describe how to use consensus protocols to achieve this goal, and present two formulations (a synchronous and an asynchronous algorithm) to perform the counting.

In robotic networks, rendezvous refers to the task of moving each of the agents in the network toward a common meeting point using only the observations of the neighboring agents. Several distributed algorithms for solving the rendezvous problem are currently available in the literature. Originally presented in [1], the problem has been extended to both synchronous [15] and asynchronous [20, 32] cases. The proposed algorithms are all distributed in the sense that each robot takes the decision based only on the observations of a certain subset of the agents in the formation.

Observe that if the agents move freely in $\mathbb{R}^n$, then for the robotic network to achieve rendezvous is equivalent for it to achieve consensus in $\mathbb{R}^n$. Such relationship between consensus and rendezvous has not been unnoticed to researchers [36, 39, 40]. Nonetheless, an explicit investigation on the nature of such relationship has not been presented, so the results in this chapter fill this gap. The rendezvous algorithm proposed here is shown to be equivalent to a consensus protocol, and hence its convergence properties are inherited from the latter. Furthermore, it is shown that in this way the analysis of the rendezvous algorithms is greatly simplified. Most existing studies on the rendezvous problem assume that

the evolution of the system is deterministic: there are no random influences on the measurements and the evolution of the state. This assumption is difficult to justify in real-life applications, where both measurements and the evolution of the system have some degree of uncertainty. Some authors [16, 22, 26] have analyzed the effect of noise for a particular version of the consensus protocol, and [34] considers the effect of uniformly distributed measurement noise for a particular class of rendezvous algorithms. The authors have explored the noisy rendezvous algorithm in [10,11], where the only assumption on the noise is that it is zero-mean and bounded. The focus of this chapter, however, is on the deterministic evolution of Consensus protocols and its applications. We mention that although the rendezvous algorithms presented here generalize most of the results in the literature, there are solutions to rendezvous that escape the scope of our work, where state estimation is not performed [48].

8.2 Consensus Protocols

We present a brief overview of consensus protocols. For more extensive reading, we refer the reader to [27, 35, 37, 42]. We only address the case when there are no communication delays between the agents, since for the applications of interest such delays are not relevant as long as they are uniformly bounded. In other words, for any two agents in the network the information exchange is guaranteed to terminate after a certain finite time T that is common to every pair in the formation.

8.2.1 Notation

Consider a network of robotic agents with n robots denoted by $\mathscr{R} = \{1, 2, \ldots, N\}$. Such agents are thought as independent robots with communication and sensing capabilities, rather than different processors in a parallel computing systems as in [42].

To each agent i, a corresponding measurable function $x_i : [0, \infty) \to \mathbb{R}^n$ is assigned. For simplicity, we assume that $n = 1$, although the discussion extends trivially to higher dimensions. Such a function x_i can represent either a physical quantity associated with the agent (its temperature, distance to a target, speed, heading, etc) or a variable required for network control. In either case, it is assumed that the variable has the same meaning for all the agents in the formation.

Definition 1. The agents in the network are said to reach consensus at time $\tau > 0$ if for every i, j with $1 \leq i, j \leq n$,

$$x_i(\tau) = x_j(\tau). \tag{8.1}$$

Instead, if $\left|x_i\left(\tau\right) - x_j\left(\tau\right)\right| < \eta$, then the agents in the network have reached η-consensus.

Although it is desired that the network reaches consensus for all $\tau > T$ for some finite $T \geq 0$, or at least in some interval $[T,\ T + B)$ for suitable B, this condition is usually hard to guarantee, and in most cases even impossible. Instead, it is usually easier to guarantee *asymptotical consensus* for the agents in the formation.

Definition 2. The network is said to reach asymptotical consensus if for every $\epsilon > 0$ there exists a T such that for all $\tau > T$ and for every $i,\ j$ with $1 \leq i,\ j \leq n$,

$$\|x_i(\tau) - x_j(\tau)\| < \epsilon. \tag{8.2}$$

Communication in a network is frequently modeled by a graph that describes whether two nodes can exchange messages. This is the model that is used in what follows. The presentation follows that in [47].

Definition 3. A graph $\mathcal{G}$ is a triple $(V,\ E,\ \sim)$ where the set V is called the set of vertices, $E \subset V \times V$ is the set of edges, and $\sim$ is the binary relation corresponding to $E:\ (v_i,\ v_j) \in E \Leftrightarrow v_i \sim v_j$. When $\sim$ is not symmetric, the graph is said to be directed, otherwise $\mathcal{G}$ is an undirected graph.

When modeling the communication structure in the network as a graph, there is a bijection between the agents in the formation and the nodes of $\mathcal{G}$, with an edge between the vertices associated with agents i and j if and only if i is capable of communicating with j. In case that the communication is guaranteed to be bidirectional for any two agents, then the underlying graph is undirected, otherwise it is directed. When the graph is undirected, and for any two agents $i,\ j$ there exists a path from i to j, the graph is said to be connected.

Definition 4. The undirected graph $\mathcal{G} = (V,\ E,\ \sim)$ is said to be *connected* if for every two vertices $v_i,\ v_j \in V$ there is a path between them. This is that there exist vertices $v_{i,1},\ v_{i,2},\ \ldots,\ v_{i,m}$ such that $v_i \sim v_{i,1},\ v_{i,1} \sim v_{i,2},\ \ldots,\ v_{i,l} \sim v_{i,l+1}$ for $1 \leq l \leq m-1$ and $v_{i,m} \sim v_{i,j}$.

In a network with mobile agents, it is natural that the set of robots with which an agent can communicate with changes as they move. Such changes are reflected in variations of the associated graph $\mathcal{G}$. When such situation arises, it is said that the communication topology of the network has changed. Note that different communication topologies can result in isomorphic graphs.

In graph theory terminology, two vertices $v_i,\ v_j \in V$ are said to be neighbors if $v_i \sim v_j$.

Definition 5. For each vertex $v_i \in V$, the set of its neighbors is given by

$$\mathcal{N}_i = \left\{v_j \in V \middle|\, v_i \sim v_j \text{ and } v_i \neq v_j\right\} \cup \{v_i\}. \tag{8.3}$$

In other words, it is the set that represents all the vertices that are connected with an edge to v_i, including itself.

8.2.2 Continuous and Discrete Time Consensus Algorithms

Consider a network of agents for which the continuous time dynamics associated with the corresponding vector of variables for each agent i is given by:

$$\dot{x}_i(t) = u_i(t) \tag{8.4}$$

and its discrete time version by:

$$x_i(k+1) = x_i(k) + \varepsilon u_i(k) \tag{8.5}$$

for some suitable *small* $|\varepsilon| > 0$. The role of this ε is described later.

The authors in [37] focus on the analysis of the continuous time system in (8.4) where the inputs u_i are given by

$$u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij} \left(x_j \left(t - \tau_{ij} \right) - x_i \left(t - \tau_{ij} \right) \right), \tag{8.6}$$

where $a_{ij} \geq 0$ for all i, j and τ_{ij} is the delay associated with the communication between the agents represented by vertices v_i and v_j. In the case that $\tau_{ij} = 0$ then no delay is expected in the transmission. As was mentioned before, as long as the delays are uniformly bounded, their effect is not critical for the performance of the network and hence we assume that $\tau_{ij} = 0$. We next present an overview of the results in [37] on consensus algorithms for fixed communication topology and no communication delay. This discussion also provides some of the background necessary for the analysis of the discrete time version.

The vector of the associated variables $x_1, x_2, \ldots, x_n$ at time t is denoted by $\mathbf{x}(t)$,

$$\mathbf{x}(t) = \left[x_1(t)\ x_2(t) \cdots x_n(t) \right]^T \tag{8.7}$$

From (8.4) and (8.6), the dynamics of the associated variables can be written as

$$\dot{\mathbf{x}}(t) = -\mathscr{L}\mathbf{x}(t), \tag{8.8}$$

where the matrix $\mathscr{L} = \left[l_{ij} \right]$ has entries given by

$$l_{ij} = \begin{cases} \sum_{k=1, k \neq i}^{n} a_{ik}, & j = i \\ -a_{ij} & j \neq i. \end{cases} \tag{8.9}$$

Here, the a_{ij} are taken as in (8.6). Hence if $v_s \notin \mathcal{N}_i$, then $a_{is} = 0$. The matrix defined in this way is known as the *graph Laplacian* [23] for the graph induced by the system in (8.4) and with inputs as in (8.6).

For the linear system given in (8.8) to be stable, it is required that the nonzero eigenvalues of the matrix $-\mathscr{L}$ have negative real parts, and that $-\mathscr{L}$ should not have more than one eigenvalue equal to zero [14].

Geršgorin's circle theorem [3,25] implies that all the eigenvalues of $\mathscr{L}$ are either zero or have negative real part. The proof of this claim can be found in [37]. Due to the construction of $\mathscr{L}$, the graph Laplacian has the vector $\mathbf{1}$ as an eigenvector associated with the eigenvalue 0 (recall that the sum of each row in the matrix is 0), and hence $\text{rank}(\mathscr{L}) \leq n-1$. It is a well-known result in algebraic graph theory that if the graph $\mathscr{G}$ with n vertices has r disjoint connected components, then $\text{rank}\,(\mathscr{L}) = n-r$ [23]. Therefore, if G is connected, $\mathscr{L}$ has one single eigenvalue at zero. The system in (8.8) is thus stable, its equilibrium position $\mathbf{x}^\star$ exists and is a scalar multiple of $\mathbf{1}$, which implies that the system reaches asymptotic consensus.

When the system is modeled as a discrete time process as in (8.5), it can be written as

$$\mathbf{x}(k+1) = (I - \varepsilon\mathscr{L})\,\mathbf{x}(k). \tag{8.10}$$

The matrix $I - \varepsilon\mathscr{L}$ is the one dictating the stability and convergence properties for this protocol. This matrix is known as the *Perron Matrix* induced by $\mathscr{G}$ and denoted by P_ε,

$$P_\varepsilon = (I - \varepsilon\mathscr{L})\,\mathbf{x}(k). \tag{8.11}$$

Under the assumption that

$$\varepsilon < \frac{1}{\max_i l_{ii}} \tag{8.12}$$

all the entries of P_ε are positive, and add up to 1. Because of this, Geršgorin's Circle Theorem guarantees that all the eigenvalues of P_ε either lie in the interior of the unit circle or are equal to 1. For the system to be stable, it is required that P_ε has at most one eigenvalue of norm 1[14]. If λ is an eigenvalue of $\mathscr{L}$, then $1 - \varepsilon\lambda$ is an eigenvalue of P_ε. Furthermore, by comparing the characteristic polynomials of $\mathscr{L}$ and P_ε it follows that the multiplicity of $1 - \varepsilon\lambda$ as an eigenvalue of P_ε is the same multiplicity of λ as an eigenvalue of $\mathscr{L}$. Therefore, if $\mathscr{G}$ is connected, and (8.12) holds, P_ε has a single eigenvalue at 1 and the system described by (8.10) is stable. This argument guarantees that as long as the underlying graph $\mathscr{G}$ remains constant, the discrete time system given in (8.10) converges toward consensus.

The algorithm described by (8.4) ((8.5)) solves the consensus problem but not necessarily the *average-consensus* problem[1]. This means that the equilibrium position $\mathbf{x}^\star$ for (8.8) (or (8.10)) does not necessarily coincide with $\mu_{\mathbf{x}(0)} = \frac{1}{n}\mathbf{1}^T\mathbf{x}(0)$. To achieve the average-consensus, the following additional condition needs to hold:

$$\frac{\mathrm{d}\left(\mathbf{1}^T\mathbf{x}(t)\right)}{\mathrm{d}t} = \sum_{i=1}^{n} \dot{x}_i(t) = 0 \tag{8.13}$$

[1] This is the problem when the equilibrium position $\mathbf{x}^\star$ has the same sum as the initial vector $\mathbf{x}(0)$.

which is equivalent to

$$\mathbf{1}^T \mathscr{L} = \mathscr{L}\mathbf{1} = 0. \tag{8.14}$$

In discrete time, this becomes

$$\mathbf{1}^T \mathbf{x}[k+1] = \mathbf{1}^T \mathbf{x}[k], \tag{8.15}$$

which is equivalent to

$$\mathbf{1}^T P_\varepsilon = \mathbf{1}^T. \tag{8.16}$$

This condition is automatically true (both in continuous and in discrete time) in an undirected graph since $\mathscr{L}^T = \mathscr{L}$.

This particular form of the Laplacian matrix was studied by Jadbabaie et al. [27] and Olfati-Saber and Murray [37]. Instead, we consider a more general family of matrices, as discussed in [35, 42]. Let $\mathbf{x}_0 \in \mathbb{R}^n$ be a vector, and let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a square matrix with the following properties:

1. $\mathbf{A}$ is primitive: there is a positive integer k such that $\mathbf{A}^k$ has all its entries positive.
2. $\mathbf{A}$ is stochastic: all its entries are nonnegative, and the sum of the entries in each row is equal to 1.

The matrix P_ε discussed earlier falls into this category, but has the additional property that $\mathbf{1}^T P_\varepsilon = \mathbf{1}^T$. Matrices with this property are called *doubly stochastic* [41]. Requiring a matrix to be doubly stochastic is quite restrictive, as we discuss in Sect. 8.3. We briefly review the convergence analysis for matrices that satisfy conditions (1) and (2) above.

It follows from Geršgorin's circle theorem and the Perron–Frobenius Theorem for primitive matrices [41] that $\mathbf{A}$ has all but one of its eigenvalues in the interior of the complex unit circle, that the remaining eigenvalue is equal to 1 and has $\mathbf{1}$, the vector in $\mathbb{R}^n$ which has all its entries equal to 1, as its associated eigenvector. From here, it follows that the discrete time linear system given by

$$\mathbf{x}_m = \mathbf{A}^m \mathbf{x}_0 \tag{8.17}$$

is stable, and converges to an equilibrium point that is a scalar multiple of $\mathbf{1}$, the eigenvector associated with the eigenvalue 1[14]. Such a matrix $\mathbf{A}$ is called a *consensus matrix*.

It is shown, among others in [35, 42], that if $\{\mathbf{A}_i\}_{i \in \mathbb{N}} \subset \mathbb{R}^{n \times n}$ are all consensus matrices, and their positive entries are uniformly bounded below by a certain $\alpha > 0$, then

$$\lim_{m \to \infty} \prod_{i=1}^{m} \mathbf{A}_i = \mathbf{v}^T \otimes \mathbf{1}, \tag{8.18}$$

where $\otimes$ denotes the Kronecker product and $\mathbf{v}$ is a vector such that it has nonnegative entries that add up to 1. This is the first important difference with the previous scenario: even if the communication graph associated with the network changes, as long as the communication structure is described by a stochastic matrix $\mathbf{A}_i$, asymptotic consensus is attained.

The following two lemmas describe some useful properties of the consensus matrices.

Lemma 1 ([41]). *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ *be two nonnegative matrices. If both* $\mathbf{A}$ *and* $\mathbf{B}$ *have their zero and positive entries in the same positions (we will say that they have the same* profile*), then either both matrices are primitive, or none of them is. In other words, for nonnegative matrices the condition of being primitive only depends on the profile of the matrix.*

Lemma 2. *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$. *If* $\mathbf{A}$ *is a consensus matrix and* $\mathbf{B}$ *is a stochastic matrix, then* $\beta\mathbf{A} + (1-\beta)\mathbf{B}$ *is a consensus matrix for all* $\beta \in (0,\ 1)$.

Proof. Let $\mathbf{C} = \beta\mathbf{A} + (1-\beta)\mathbf{B}$, $\beta \in (0,\ 1)$. Since $\mathbf{C}$ has nonnegative entries, and $\mathbf{C1} = (\beta\mathbf{A} + (1-\beta)\mathbf{B})\,\mathbf{1} = \mathbf{1}$, it follows that $\mathbf{C}$ is stochastic. To show that it is also primitive, observe that since $\mathbf{A}$ is primitive, there is a positive integer k such that $\mathbf{A}^k$ only has positive entries. Since $\mathbf{C}^k = (\mathbf{A}+\mathbf{B})^k = \mathbf{A}^k + \cdots$ and the entries of both $\mathbf{A}$ and $\mathbf{B}$ are nonnegative, the entries of each summand are also nonnegative. Therefore, since all the entries of $\mathbf{A}^k$ are positive, so are those of $\mathbf{C}^k$, implying that it is primitive. □

8.2.3 Proximity Graphs

The concept of *proximity graph* is useful to introduce the binary relation $\sim$ which defines the communication graph. In this section, we follow the presentation in [15].

Let $\mathbb{F}(\mathbb{R}^n)$ be the set of finite point sets in $\mathbb{R}^n$. Let $\mathscr{P} = \{p_1, \ldots, p_m\} \subset \mathbb{R}^n$ be a typical element of $\mathbb{F}(\mathbb{R}^n)$, where $p_1, \ldots, p_m$ are distinct points. Let $\mathbb{G}(\mathbb{R}^n)$ be the set of undirected graphs, whose vertex set belongs to $\mathbb{F}(\mathbb{R}^n)$.

A *proximity graph function* $\mathscr{G} : \mathbb{F}(\mathbb{R}^n) \to \mathbb{G}(\mathbb{R}^n)$ is a map that associates with each element $\mathscr{P} \in \mathbb{F}(\mathbb{R}^n)$ a graph with vertices given by the elements of $\mathscr{P}$, and with the set of edges $\mathscr{E}$ defined by the function $\mathscr{E}_\mathscr{G} : \mathbb{F}(\mathbb{R}^n) \to \mathbb{F}(\mathbb{R}^n \times \mathbb{R}^n)$ contained in $\mathscr{P} \times \mathscr{P}$.

The matrix $\mathbf{A}_\mathscr{G} \in \mathbb{R}^{n\times n}$ is induced by the proximity graph $\mathscr{G}(\mathscr{P})$ if its entries are nonnegative, if it has nonzero diagonal terms, and the entry $a_{ij} \neq 0$ if and only if $(p_i,\ p_j) \in \mathscr{E}$. In case that the proximity graph is undirected, then $(p_i,\ p_j) \in \mathscr{E} \Rightarrow (p_j,\ p_i) \in \mathscr{E}$. For simplicity, the edges of the graph $\mathscr{G}$ are denoted either by $\mathscr{E}$ or by the function $\mathscr{E}_\mathscr{G}$ that defines the set.

Lemma 3. *If* $\mathscr{G}(\mathscr{P})$ *is connected, then* $\mathbf{A}_\mathscr{G}$ *is primitive.*

Proof. By Lemma 1, it is enough to show the result when the positive entries of the matrix $\mathbf{A}_\mathscr{G}$ are equal to 1. Observe that the $(i,\ j)$ entry in the product $\mathbf{A}_\mathscr{G}^k$ is given by

$$\sum_{(l_1, l_2, \ldots l_{k-1}) \subseteq \{1,\ldots,n\}^{k-1}} a_{il_1} a_{l_1 l_2} \ldots a_{l_{k-1} j},$$

which is positive if and only if it is possible to arrive from position p_i to position p_j, passing through at most k different vertices (in fact, the entry in the position (i, j) is the number of ways of doing so). Since $\mathcal{G}$ ($\mathcal{P}$) is connected, the result follows. □

8.3 Counting

We present an application to distributed counting based on consensus protocols. We distinguish between both synchronous and asynchronous implementations.

Although distributed counting is a well-known problem in parallel computing [45, 46], in that context it refers to a variable, which provides information about a shared process in the network. In our case, *counting* refers to a local estimation of the total number of agents in the network. In other words, we are interested in an algorithm that allows each agent *to count on its own the total number of agents in the network*. Such algorithm is presented in Algorithm 1. Before deployment, the variable **count** is assigned to 0 to all the agents, with the exception of one of them, which has it set to 1. The agents then converge to average-consensus on the variable **count**, which leads them to the common value $1/n$.

Observe that the characteristics of Algorithm 1 are given by the particular average-consensus algorithm that is implemented in Line 2.

Suppose that the average-consensus protocol implemented for the agents is that presented in (8.10). Note that since the agents can move, the matrix P_ε, defined in (8.11), might change with time, and hence the stability argument needs to be modified. Although Kingston and Beard [30] show that it is possible to achieve average-consensus under switching topology using the protocol in (8.10), we present an alternative proof, which is more suitable for our needs. The proof in [30] relies on the results of [35] to show that

$$\lim_{N\to\infty} A[k+N]A[k+1]\cdots A[k] = \frac{1}{n}\mathbf{1}\mathbf{1}^T, \tag{8.19}$$

where the matrices $\{\mathbf{A}[i]\}_{i\geq k}$ are doubly stochastic. Note the similarity between (8.19) and (8.18). The special form in (8.19) is due to the doubly stochastic nature

Algorithm 1 Distributed counting

Require: Each agent has its own variable **count**. The sum of all the variables **count** is equal to 1.
Require: Agent i at time m.
1: Identify the set of neighbors $\mathcal{N}_i$.
2: Run an iteration of the average-consensus algorithm for **count** in $\mathcal{N}_i$.
3: Repeat.

of the consensus matrices, which is imposed since we are solving the average-consensus problem for the variable **count**.

To make explicit the changes in the communication topology, the Perron matrix $P_\varepsilon = I - \varepsilon\mathscr{L}$ is now denoted as $P_{\varepsilon,\mathscr{L}}$. The system in (8.10) can then be written as

$$\mathbf{x}[k+1] = P_{\varepsilon,\mathscr{L}}\mathbf{x}[k]. \tag{8.20}$$

Although it is possible to have double stochastic matrices that are not associated with an undirected graph, such implementations require for the agents to have additional knowledge about the network. Since this compromises the robustness of the approach, we thus assume that the graph is undirected.

Claim. The discrete time linear system given by (8.20) satisfies

$$\mathbf{x}[k+1]^T\mathbf{x}[k+1] \leq \mathbf{x}[k]^T\mathbf{x}[k] \tag{8.21}$$

with equality if and only if $\mathbf{x}[k]$ is a multiple of $\mathbf{1}$.

Proof. Equation (8.21) is equivalent to:

$$\mathbf{x}[k]^T P_{\varepsilon,\mathscr{L}}^T P_{\varepsilon,\mathscr{L}}\mathbf{x}[k] \leq \mathbf{x}[k]^T\mathbf{x}[k],$$

which holds if and only if

$$\mathbf{x}^T[k]\left(I - P_{\varepsilon,\mathscr{L}}^T P_{\varepsilon,\mathscr{L}}\right)\mathbf{x}[k] \geq 0. \tag{8.22}$$

Since the underlying graph is undirected, $\mathscr{L}$ is real and symmetric, hence so is $P_{\varepsilon,\mathscr{L}}$ and there exist matrices U and Λ such that:

$$P_{\varepsilon,\mathscr{L}} = U^T\Lambda U, \tag{8.23}$$

where Λ is a diagonal matrix and $U^T = U^{-1}$ is orthonormal [25]. Since the elements in the diagonal of Λ are the eigenvalues of $P_{\varepsilon,\mathscr{L}}$, all its entries are real, bounded in absolute value by 1, and the matrix $I - \Lambda^m$ is then positive semidefinite for any natural number m.

Therefore,

$$\begin{aligned}\mathbf{x}^T[k]\left(I - P_{\varepsilon,\mathscr{L}}^T P_{\varepsilon,\mathscr{L}}\right)\mathbf{x}[k] &= \mathbf{x}^T[k]U^T\left(I - \Lambda^2\right)U\mathbf{x}[k]\\ &= \mathbf{y}[k]^T\left(I - \Lambda^2\right)\mathbf{y}[k]\\ &\geq 0,\end{aligned}$$

which coincides with (8.22). Equality holds if and only if the vector $\mathbf{y}[k] = U\mathbf{x}[k]$ lies in the null space of $\left(I - \Lambda^2\right)$, which is equivalent to say that the vector $\mathbf{x}[k]$ is an eigenvector associated with the eigenvalue 1 of the matrix $P_{\varepsilon,\mathscr{L}}^T P_{\varepsilon,\mathscr{L}}$. Since

this matrix can be written as $U^T \Lambda^2 U$, and the eigenvalue 1 has multiplicity 1 in $P_{\varepsilon,\mathscr{L}} = U^T \Lambda U$ with $\mathbf{1}$ being an eigenvector associated with it, then $\mathbf{x}[k]$ must be a multiple of $\mathbf{1}$. □

Corollary 1. *The system given by* (8.20) *converges to average-consensus.*

Proof. Since $\|\mathbf{x}[k]\|_2^2$ is decreasing unless the system is in equilibrium, by LaSalle's invariance principle the system converges to the largest invariant set for (8.20), which is exactly the one-dimensional space given by span $\{\mathbf{1}\}$. Since the sum of the elements of $\mathbf{x}$ remains constant, the system converges to average-consensus, despite possible changes in the communication topology. □

It follows that the algorithm given by (8.20) leads to average-consensus in the network. When this is the consensus algorithm implemented in Algorithm 1, the agents can thus estimate the number of robots.

One of the key elements in the previous argument was the connectivity assumption of the underlying graph, which implies that the communication process obeys some synchronism. The connectivity assumption can be relaxed, and hence the synchronism requirements can be weakened. Similar to the discussion in Sect. 8.4.3, instead of requiring the underlying graph to be connected at every time, t, it is enough to guarantee that the formation is *jointly connected* (see Definition 8).

The aggregation algorithm, which is discussed next, can be seen as the extreme case when the previous connectivity assumption is relaxed, and we only require for the agents to communicate in pairs. In this way, at each iteration, the graph we are considering is disconnected, and its components are either isolated vertices (agents that are not communicating with other agents) or connected subgraphs with two vertices and an edge (that represents the two agents that are communicating). By imposing that the formation is jointly connected (which implies that every agent communicates at least with one of its neighbors every T iterations), average-consensus can be reached without imposing synchronism in the network. This hypothesis allows the algorithm to be fully asynchronous, something that cannot be guaranteed by the system in (8.20).

8.3.1 The Aggregation Algorithm

In aggregation, the agents only update their information in pairs (that can change in time), and the information exchange between agents i and j is given by

$$x_i(t+1) = \frac{1}{2}x_i(t) + \frac{1}{2}x_j(t) = x_j(t+1). \tag{8.24}$$

We assume that when agent i communicates with j, then j also communicates with i, so both agents update simultaneously their values to the mean of their current associated values. Note that if $x_i = x_j$ then the associated valued does not

change. We assume then that this protocol is used only when $x_i \neq x_j$. The algorithm described by (8.24) is not exclusive variation to the control community. For instance, in communications and computer networks it is known as the *gossiping algorithm* [5,6].

Claim. After one update, the following inequality holds:

$$\mathbf{x}[k+1]^T\mathbf{x}[k+1] \leq \mathbf{x}[k]^T\mathbf{x}[k], \tag{8.25}$$

with equality if and only if the update in the associated values occurred between two agents that were already in consensus.

Proof. Suppose that agents x_i and x_j updated their values, then

$$\begin{aligned}
\mathbf{x}[k+1]^T\mathbf{x}[k+1] &= \sum_{r=1}^{n} x_r[k+1]^2 \\
&= \sum_{r=1, r\neq i,j}^{n} x_r[k]^2 + 2\left(\frac{x_i[k]+x_j[k]}{2}\right)^2 \\
&= \sum_{r=1}^{n} x_r[k]^2 + 2\left(\frac{x_i[k]+x_j[k]}{2}\right)^2 - x_i[k]^2 - x_j[k]^2 \\
&= \mathbf{x}[k]^T\mathbf{x}[k] - \frac{1}{2}\left(x_i[k]-x_j[k]\right)^2,
\end{aligned}$$

which proves the claim. □

Corollary 2. *If the network is jointly connected, then the variables x_i converge to average-consensus.*

Proof. From Claim 8.3.1, $\|\mathbf{x}[k]\|_2^2$ is strictly decreasing unless every agent is in consensus with all the other agents. Since the network is jointly connected, if not all the x_i are equal, then there is a time when there is a communication between two agents with different associated variables. From LaSalle's principle, the x_i thus converge to the largest invariable set where they are defined, which is exactly the set on which they are all equal to each other. □

8.3.2 Distributed Counting of Agents

As we discussed earlier, by using any of the two previously discussed average-consensus algorithms as the average consensus in Line 2 of Algorithm 1, the variable **count** reaches consensus at the value $1/n$, providing the information on the number of agents. The initialization of the variables can be done when the network is

initially deployed, so it is reasonable to expect the right distribution for the variable **count**.

An interesting question is how the counting algorithm performs when the limited numerical accuracy of the computations is considered. With m bits computation, the estimated value $1/\hat{n}$ is within $1/2^m$ of $1/n$. The agent approximates $1/\hat{n}$ to the closest number of the form $1/n$ where n is an integer. The estimate is thus be correct if

$$\frac{1}{2^{m-1}} < \frac{1}{n-1} - \frac{1}{n} = \frac{1}{n(n-1)} < \frac{1}{(n-1)^2}. \tag{8.26}$$

In other words, if

$$n < \sqrt{2^{m-1}} + 1, \tag{8.27}$$

then the agents in the formation can obtain a correct estimate of n. With 32-bit computation, the agents can estimate the size of networks with about 32,000 agents. If 64-bit computation is available, then networks with less than 1 billion agents can reach the correct estimate on the number of robots. This brief discussion is related to the problem of *quantized consensus*, which is an active field of research nowadays, and we refer the reader to [2, 12, 13, 29] and the references therein.

In [19] it was shown that the asymptotic consensus cannot be achieved even if only one of the agents is malfunctioning. Aggregation is not an exception, given it is just a special case of consensus, so the network should be capable of dealing with such failures. Of course, if there are many agents failing or if some of them are malicious and attempt to prevent the correct operation of the network, the system will fail if no control is implemented to deal with such a situation [18]. We thus assume that the agents are not malicious, that at most one agent can fail at any time t, and that the agents will not be failing frequently (the last two assumptions are not relevant but simplify the presentation). We assume that the agents have a way to identify which of the neighbors, if any, are malfunctioning and exclude them from any operation inside the network. We next describe how the counting algorithm can handle the situations when agents are added to the network, or some agents need to be removed.

8.3.3 Adding and Removing Agents from the Formation

There might be situations where the number of agents originally deployed in the network is insufficient for the task at hand, and more robots need to be added. The agents that are already deployed in the formation do not know beforehand how many *new* agents will be added to the system in case they are required, so they need to implement a way to count them. Also, agents might start malfunctioning and the neighboring agents need to exclude them from the formation while notifying the rest of the network. In this section, we present a modification to Algorithm 1 that allows us to deal with these two scenarios.

8.3.3.1 Adding Agents to the Formation

If r agents are added to the system, the consensus value changes from $1/n$ to $1/(n+r)$. Introducing the new agents with their variable **count** initialized at zero allows the system to converge to the new required consensus value. Observe that the only requirement in Algorithm 1 is that the sum of the **count** variables is equal to one, which remains valid since the new agents are not modifying such summation. Henceforth, if the network keeps running Algorithm 1, the agents reach consensus for **count** at the value $1/(n+r)$.

Remark 1. Adding agents to the formation implies an increase in the dominant diagonal term for the Laplacian matrix, $\max_i l_{ii}$, which might lead to a change in the parameter ε in (8.20). In this case, the agents need to update ε (perhaps through a request from a central node) or the agents fail to reach the right consensus. Note that this problem is not presented when the aggregation protocol is used.

8.3.3.2 Removing Faulty Agents from the Formation

In a wireless sensor network, there is always a possibility that an agent either runs out of battery or fails. We assume that any agent is capable of recognizing which of its neighbors (if any) are malfunctioning. Once a faulty agent has been detected, the agents need to update their agent count estimates. The following discussion assumes that the agents are working asynchronously by using the aggregation protocol.

Definition 6. A *session* is the set of events that occur between the last time an agent malfunctioning was detected and the new detection of a faulty robot.

Observe that *session* is a global concept. The system starts with session 0 and stays in session 0 until an agent detects a problem with at least one of its neighbors.

Algorithm 2 presents how an agent reacts when one of its neighbors fails. When a faulty agent is detected, the agent chooses a random number $r > 0$, adds it to the previous session number, and updates its list of malfunctioning robots **malfuncid**. Then, it resets its variable **count** to one, and propagates the message containing its new **count** value, its new list of malfunctioning agents **malfuncid**, and the new session number for the network.

Algorithm 2 Session update

Require: Agent i detects that neighbor j is malfunctioning.
1: Reset **count** to one.
2: Choose random $r > 0$.
3: $Session_i \leftarrow Session_i + r$.
4: Append id j to **malfuncid**.

Algorithm 3 Robust Counting Algorithm

Require: Agent i communicates with agent j.
1: **if** Agent j malfunctioning **then**
2: **do** Session
3: **else**
4: **if** $Session_i < Session_j$ **then**
5: $Session_i \leftarrow Session_j$.
6: Takes **malfuncid** from j.
7: Resets **count** to zero.
8: **end if**
9: **count**←Average(**count** of i, **count** of j).
10: **end if**

When two agents communicate, they proceed as in Algorithm 3. If they have decided that the other agent is not malfunctioning, then each agent compares the session number of the other robot with its own. If the new session number is higher, then it updates it session number, drops its previous **count** value, resets it to zero, takes the list of failing agents from the robot that had the higher session number, and performs the averaging operation of the aggregation algorithm with the **count** variables.

At each time, each agent chooses among its neighbors either the one, if any, that is still in the previous session, or the one in the new session for which the difference between the two values is maximal. If more than one agent is in some of these situations, it chooses which one to pick at random. This guarantees that, with probability 1, each robot is chosen infinitely often. In the improbable case that two agents decide to reset their values and initiate new sessions at the same time, the session with the highest code prevails.

Proposition 1. *By using the robust counting algorithm in Algorithm 3, every agent will be eventually included in the new session.*

Proof. Suppose there is at least one agent that has not been included in the new session. Since the communication graph for the system is jointly connected, there is a path between any agent that is not in the new session and the agent that started it. Therefore, there is at least one agent in the path that is not in the new session while one of its neighbors is. The number of agents in the old session is thus decreasing and it follows that after a finite time, all of the agents are in the updated session. □

Remark 2. Observe that changes in the communication topology do not affect the above discussion because the underlying communication graph is still jointly connected.

8.4 Rendezvous

We now study another application of the consensus algorithms. We focus on how to use this family of protocols to solve the rendezvous task.

8.4.1 Model

Assume a robot formation consists of N agents and let $\mathscr{R} = \{q_i\}_{i=1}^{N} \in \mathbb{F}(\mathbb{R}^n)$ be their locations described in a fixed coordinate frame $\mathscr{Q}$. It is assumed that the agents have no knowledge about $\mathscr{Q}$. The goal of the network is to achieve *rendezvous*: all the agents should meet at a single (unspecified) point.

We assume that each robot is capable of identifying those agents in the formation that satisfy a certain relation $\sim \subseteq \mathbb{R}^n \times \mathbb{R}^n$. This relation defines the edges of the proximity graph, $\sim = \mathscr{G}_{\mathscr{E}}(\mathbb{R}^n)$. In other words, $(p_i, p_j) \in \mathscr{G}_{\mathscr{E}}(\mathscr{R})$ if and only if $p_i \sim p_j$. For each agent i, we denote the set of all the agents j to which it is related by $\mathscr{N}_i$.

Remark 3. If rather than observing and estimating the position of the agents (which can be done, for instance, using cameras or ultrasonic sensors), agents are allowed to communicate their location, then there needs to be a universal frame $\mathscr{Q}$ known to all the robots.

The motion of each agent i will be assumed to be governed by a simple first-order discrete-time dynamics:

$$q_i[m+1] = q_i[m] + u_i[m], \tag{8.28}$$

where q_i is the position of the agent with respect to the coordinate frame $\mathscr{Q}$[2], and u_i is the control input that determines how the agent moves. We next describe the algorithm for computing the control law u_i that achieves rendezvous.

8.4.2 Consensus-Based Rendezvous

As noted earlier, achieving rendezvous is equivalent to the agents achieving consensus on their position in $\mathbb{R}^n$. This motivates formulating the control law for computing the control input u_i in (8.28) using a distributed consensus algorithm. It will be shown that not only does such a formulation inherit all the convergence properties of consensus algorithms, but that it also generalizes most existing rendezvous algorithms.

Definition 7. Given a finite set $\mathscr{X} = \{x_0, x_1, \ldots, x_n\} \subset \mathbb{R}^n$ and $\epsilon > 0$, an *ϵ-convex combination in $\mathscr{X}$* is a convex combination $\sum_{j=1}^{n} \beta_j x_j$ of the elements of $\mathscr{X}$ (that is, $\beta_i \geq 0$ and $\sum_{j=1}^{n} \beta_j = 1$), where each coefficient β_i is either 0, or bounded below by ϵ.

Given $x_i \in \mathscr{X}$, a convex combination $\sum_{j=1}^{n} \beta_j x_j$ is centered at x_i if β_i is positive.

[2] The reference to a global coordinate frame Q will be relaxed in Sect. 8.4.2.

Algorithm 4 Consensus-based rendezvous

Require: Constants ϵ and ξ, such that $1 \gg \epsilon > 0$ and $0 < \xi < 1$.
1: $m \leftarrow 1$
2: **loop**
3: **for** every agent i **do**
4: Identify the set of neighbors $\mathcal{N}_i = \{i, i_1, i_2, \ldots, i_{r_i(m)}\}$.
5: Determine the position $p^i_{i_j}$, $1 \leq j \leq r_i(m)$ of each neighbor, and its own position p^i_i with respect to an arbitrary coordinate frame $\mathcal{Q}_i$.
6: Let $q^i_i = \sum_{j=0}^{r_i(m)} \beta^j_i(m)\, p^i_{i_j}$ be an *arbitrary* ϵ-convex combination in $\mathcal{N}_i$, centered at p_i.
7: Set $u_i[m] = \xi\left(q^i_i - p^i_i\right)$.
8: **end for**
9: $m \leftarrow m + 1$
10: **end loop**

Algorithm 4 describes *Consensus-Based Rendezvous* (CBR). In the algorithm, lines 4–7 compute the control input u_i for each agent i. The constant ξ used in Line 7 is assumed to be known to all the agents. However, it will be shown later that the algorithm also converges if each agent chooses its own ξ_i independently from the other robots.

The following lemma shows that the control input u_i, and therefore, the next position $q_i[m+1]$ does not depend on the coordinate frame $\mathcal{Q}_i$.

Lemma 4. *The evolution of each agent is independent of its local coordinate frame* $\mathcal{Q}_i$.

Proof. Let $p_{i_1}, p_{i_2}, \ldots, p_{i_{r_i(m)}}$ be the positions of the neighbors of i with respect to the fixed coordinate frame $\mathcal{Q}$. Let $p^i_{i_1}, p^i_{i_2}, \ldots, p^i_{i_{r_i(m)}}$ be these positions expressed with respect to the coordinate frame $\mathcal{Q}_i$. Recall that the change of the coordinate system between $\mathcal{Q}_i$ and $\mathcal{Q}$ is given by a translation d_i and a rotation R_i so that $p_{i_j} = d_i + R_i\, p^i_{i_j}$. Therefore, with respect to the coordinate frame $\mathcal{Q}$ the convex combination q^i_i in Line 6 of Algorithm 4 becomes

$$q_i = \beta^0_i\left(d_i + R_i\, p^i_i\right) + \cdots + \beta^{r_i(m)}_i\left(d_i + R_i\, p^i_{i_{r_i(m)}}\right) \tag{8.29}$$

$$= \left(\beta^0_i + \cdots + \beta^{r_i(m)}_i\right) d_i + R_i\left(\beta^0_i\, p^i_i + \cdots + \beta^{r_i(m)}_i\, p^i_{i_{r_i(m)}}\right) \tag{8.30}$$

$$= d_i + R_i q^i_i, \tag{8.31}$$

which only depends on the coefficients β^j_i and not on the choice of the coordinate frame $\mathcal{Q}_i$. □

Remark 4. The orthonormality of the matrix R_i was not used in the proof above. The result is thus valid for any transformation induced by an invertible matrix R'_i.

If we write each p_i as a row vector, due to Lemma 4, we can stack together the equations for each agent, and write the evolution of the system in the matrix form as

$$\mathbf{q}[m+1] = \mathbf{I}\,\mathbf{q}[m] + \mathbf{U}[m],$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix, $\mathbf{q} \in \mathbb{R}^{N\times n}$ is the state vector, where each row represents the position of one of the agents, and $\mathbf{U} \in \mathbb{R}^{N\times n}$. Under the assumption that the proximity graph $\mathscr{G}(\mathscr{R})$ is connected, the induced matrix $\mathbf{A}_{\mathscr{G}}$ with $a_{i,j} = \beta_i^j$ is a consensus matrix. This makes $\mathbf{U}[m] = \xi\,(\mathbf{A}_{\mathscr{G}} - \mathbf{I})\,\mathbf{q}[m]$, $\xi \in (0,\,1)$ so each iteration of Algorithm 4 can be described as

$$\mathbf{q}[m+1] = [(1-\xi)\mathbf{I} + \xi\mathbf{A}_{\mathscr{G}}]\,\mathbf{q}[m]. \tag{8.32}$$

Applying Lemma 2, under the assumption that $\mathscr{G}(\mathscr{R})$ is connected and considering the previous remark, (8.32) can be rewritten as

$$\mathbf{q}[m+1] = \mathbf{C}[m]\mathbf{q}[m], \tag{8.33}$$

where $\mathbf{C} = [(1-\xi)\mathbf{I} + \xi\mathbf{A}_{\mathscr{G}}]$ is a consensus matrix.

As was mentioned before, each agent can actually choose the value of $\xi_i > 0$ that it prefers as long as there is a uniform constant α so that $\xi_i > \alpha > 0$, and the formulation of the problem would be equivalent.

Lemma 5. *If each agent i chooses its own ξ_i, the evolution of the network can be written as in (8.33), where the resulting matrix $\mathbf{C}$ is a consensus matrix.*

Proof. Recall that agent i updates its position as in (8.28). By replacing u_i with the expression from Line 7 in the CBR algorithm, it follows that

$$q_i[m+1] = q_i[m] + \xi_i \left[\left(\beta_i^0 q_i[m] + \sum_{j\in\mathscr{N}_i} \beta_i^j q_j[m] \right) - q_i[m] \right], \tag{8.34}$$

which can be rewritten as

$$q_i[m+1] = (1-\xi_i)\,q_i[m] + \xi_i \left(\beta_i^0 q_i[m] + \sum_{j\in\mathscr{N}_i, j\neq i} \beta_i^j q_j[m] \right) \tag{8.35}$$

As before, by representing each agent's position p_i as a row vector and stacking them together, the evolution of the system can be written in the matrix form as in (8.33). The resulting matrix $\mathbf{C}$ is stochastic, and since $\mathscr{G}(\mathscr{R})$ is connected, Lemma 3 ensures it is also primitive. Hence, $\mathbf{C}$ is a consensus matrix. □

Now, the main result of the section is presented.

Theorem 1 (Convergence of the CBR algorithm). *A discrete time system evolving according to (8.33) reaches consensus.*

Proof. Since the matrix $\mathbf{C}[m]$ is a consensus matrix, the theory of consensus algorithms guarantees the result. □

Observe that what the Theorem 1 implies is that all the robots in $\mathscr{R}$ converge toward a single location, meaning that they achieve rendezvous.

Remark 5. The constant ϵ in Line 6 is needed to guarantee the convergence of the algorithm. For practical implementations, this condition is always satisfied, since ϵ would be given by the arithmetic precision of the processors used by the agents.

Remark 6. As discussed in previous sections, to achieve *average consensus* the network needs to operate synchronously. Consensus protocols that *do not solve* the averaging problem (which is typically the case with the rendezvous) can be implemented asynchronously.

8.4.3 Robustness of the Algorithm

So far, the results assume that the proximity graph $\mathscr{G}(\mathscr{R})$, induced by the formation at time m, $m \geq 0$, is always connected and that each agent is always capable of correctly detecting and localizing each of its neighbors. In this section, we explore the case when some agents miss one (or more) of their neighbors as they execute Algorithm 4.

Following the notation of the previous sections, the state of the formation $\mathscr{R}$ at time m is denoted by $\mathscr{R}[m]$. The proximity graph induced at time m is denoted by $\mathscr{G}(\mathscr{R})[m]$, and its set of edges by $\mathscr{G}_{\mathscr{E}}(\mathscr{R})[m]$.

The *real proximity graph* of the formation $\mathscr{R}$ at time m is thus defined as the graph whose vertices are the agents in the formation, and whose edge set $\mathscr{G}'_{\mathscr{E}}(\mathscr{R})[m] \subseteq \mathscr{G}_{\mathscr{E}}(\mathscr{R})[m]$ contains the pair (i, j) if and only if the agent i *does detect* the agent j as one of its neighbors. Since the communication is not allowed between the agents, the fact that the agent i misses j does not imply that j also misses i even when $\sim$ is symmetric. Observe that when there are no errors, the *real proximity graph* coincides with $\mathscr{G}(\mathscr{R})[m]$.

Definition 8 (Jointly Connected Formation [35]). A robotic network $\mathscr{R}$ is said to be *jointly connected* if there exists a positive integer T such that, for every $m > 0$, the graph with vertices induced by $\mathscr{R}$, and the edge set defined by the union $\mathscr{E} = \bigcup_{i=0}^{T-1} \mathscr{G}'_{\mathscr{E}}(\mathscr{R})[m+i]$, is connected.

Note that the matrix $\mathbf{C}[m]$ having its entry $c_{i,j} = \beta_i^j$, the weight assigned by agent i to the position of agent j, for every pair (i, j) is always stochastic. Under the assumption that the underlying graph for $\mathscr{R}$ (the real proximity graph) is jointly connected, the following theorem (which follows directly from the standard theory of consensus algorithms) can be stated. Its proof case can be found, among others, in [27, 35, 42].

Theorem 2. *Under the assumption that the real proximity graph induced by the formation is jointly connected, the CBR algorithm is robust under failures by the agents in detecting some of the neighbors; the network $\mathscr{R}$ thus converges to rendezvous.*

Remark 7. The condition on jointly connectivity is a reasonable one to impose. Without it, rendezvous can only be considered within disconnected components.

8.4.4 Geometric Rendezvous

In the previous section, we presented an algorithm where the agents achieve rendezvous by implementing a consensus algorithm on their locations. In this section, we consider instead the *geometric rendezvous*, where rendezvous is achieved by the agents moving toward a *convex combination* of their locations. The algorithm is given in Algorithm 5.

Most of the current algorithms that are proven to solve the rendezvous problem [1, 15, 32, 34] let each agent move toward a point in the convex hull and are therefore a special case of the geometric rendezvous algorithm (GRA). In the rest of the section, we will show that any GRA can be approximated arbitrarily well with a CBR algorithm so its behavior can be studied within the framework of consensus protocols.

The following lemmas show that points in the interior of convex polytopes can be described as centered convex combinations. Lemma 9 can be easily deduced from the Krein–Millman theorem [31], and we presented a proof in [9].

Lemma 6 (Carathéodory's Theorem). *Let $\mathscr{F}$ be a convex polytope in $\mathbb{R}^n$, and let $p \in \mathscr{F}$ be any interior point. Then p can be written as a convex combination of at most $n+1$ of the vertices of $\mathscr{F}$.*

Lemma 7. *Let $\mathscr{P} = \{p_1, \ldots, p_m\} \in \mathbb{F}(\mathbb{R}^n)$, and let $Co(\mathscr{P})$ be the convex hull of $\mathscr{P}$. Let $p \in Co(\mathscr{P})$, and let q be an interior point of $Co(\mathscr{P})$. Then, q can be written as a convex combination of $\{p\} \cup \mathscr{P}$ centered at p.*

Algorithm 5 Geometric Rendezvous Algorithm

Require: Constant ξ such that $0 < \xi < 1$.
1: $m \leftarrow 1$
2: **loop**
3: **for** every agent i **do**
4: Identify the set of neighbors $\mathscr{N}_i[m] = \{i, i_1, i_2, \ldots, i_{r_i(m)}\}$.
5: Evaluate $Co(\mathscr{N}_i[m])$.
6: Set q_i^i to be any arbitrary point in $Co(\mathscr{N}_i[m])$.
7: Set $u_i[m] = \xi_i \left(q_i^i - p_i^i\right)$, where p_i^i is the position of i with respect to the same coordinate frame the agent used to define q_i^i.
8: **end for**
9: $m \leftarrow m + 1$
10: **end loop**

The previous two lemmas imply that it is possible to write any point q in a convex set as a convex combination involving a particular point p from the set. The following lemma shows an even stronger result: any point q in a convex set can be approximated arbitrarily well with an ϵ-convex combination centered at a some other chosen point p in the set.

Lemma 8. *Let $\mathscr{P} \in \mathbb{F}(\mathbb{R}^n)$. For any given $\delta > 0$, there exists $\epsilon > 0$ such that for every $p_0 \in \mathscr{P}$ and $q \in Co(\mathscr{P})$, there exists $q^\star$ which is an ϵ-convex combination in $\mathscr{P}$ centered at p_0 such that $\|q^\star - q\| < \delta$.*

Proof. Since $\mathscr{P}$ is finite, it is bounded. Therefore, there is a positive real number k such that if $x \in \mathscr{P}$, then $\|x\| < k$. If q is on the boundary of $Co(\mathscr{P})$ there is a point q' in the interior of $Co(\mathscr{P})$ such that $\|q' - q\| < \frac{\delta}{2}$. If q is in the interior of $Co(\mathscr{P})$ then let $q = q'$. According to Lemma 7, q' can be written as a convex combination of $\mathscr{P}$ centered at p_0. Let $\mathscr{P}' = \{p_0, p_1, \ldots, p_{r-1}\} \subseteq \mathscr{P}$ be the subset of $\mathscr{P}$ that denotes the points with positive coefficients in this convex combination. Since $\mathscr{P} \subset \mathbb{R}^n$, $r \leq n+1$ according to Lemma 6. Let $\delta^\star = \min\{\delta/4, k/2\}$. We will show that any $\epsilon < \delta^\star/((n+1)k)$ guarantees the existence of an ϵ-convex combination that satisfies the requirements of the lemma. So choose $\epsilon < \delta^\star/((n+1)k)$. If $p_0 = q'$, then $\mathscr{P}' = p_0$, $\beta_0 = 1$ and the lemma holds. Suppose then that $p_0 \neq q'$. Write q' as a convex combination in $\mathscr{P}'$ centered at p_0 as in Lemma 7:

$$q' = \sum_{i=0}^{r} \beta_i p_i. \tag{8.36}$$

Because of our construction of $\mathscr{P}'$, each $\beta_i > 0$. Let $\Lambda_0 = \{j \,|\, \beta_j < \epsilon\}$ and let $\Lambda_1 = \{i \,|\, \beta_i \geq \epsilon\}$. Since $(n+1)\epsilon < 2(n+1)\epsilon < 2\delta^\star/k \leq 1$, Λ_1 is nonempty. Rearrange Equation (8.36) as

$$q' = \sum_{j \in \Lambda_0} \beta_j p_j + \sum_{i \in \Lambda_1} \beta_i p_i. \tag{8.37}$$

Let $\sum_{j \in \Lambda_0} \beta_j = \beta^0 = m\epsilon + l$ where $m \in \mathbb{Z}$ and $0 \leq l < \epsilon$. Suppose $\beta^0 \geq \epsilon$. This implies that $|\Lambda_0| > m \geq 1$. Choose a set with m elements $\mathscr{J} = \{j_0, j_1, \ldots, j_{m-1}\} \subset \Lambda_0$ such that if $0 \in \Lambda_0$ then $0 \in \mathscr{J}$. For $j \in \mathscr{J}$ let $\beta_j^\star = \epsilon$. For $k \in \Lambda_0 \setminus \mathscr{J}$, make $\beta_k^\star = 0$. Choose $i \in \Lambda_1$ and set $\beta_i^\star = \beta_i + l$; for $k \in \Lambda_1 \setminus \{i\}$ set $\beta_k^\star = \beta_k$. Observe that $\sum_{i=1}^{r} \beta_i^\star = 1$. Let $q^\star = \sum_{i=0}^{r} \beta_i^\star p_i$. Note that

$$\begin{aligned} q^\star - q' &= \left(\sum_{j \in \mathscr{J}} \beta_j^\star p_j + \beta^{\star i} p_i + \sum_{i' \in \Lambda_1 \setminus \{i\}} \beta_{i'}^\star p_{i'} \right) - \left(\sum_{j \in \Lambda_0} \beta_j p_j + \sum_{i \in \Lambda_1} \beta_i p_i \right) \\ &= \sum_{j \in \mathscr{J}} \epsilon p_j - \sum_{j \in \Lambda_0} \beta_j p_j + l p_i, \end{aligned}$$

which can be bounded as follows

$$\|q^\star - q'\| \leq \epsilon \sum_{j \in \mathscr{J}} \|p_j\| + \epsilon \sum_{j \in \Lambda_0} \|p_j\| + \epsilon \|p_i\|$$
$$\leq (2(r-1)+1)\, k\epsilon < 2nk\epsilon < 2\delta^\star \leq \delta,$$

This implies $\|q - q^\star\| \leq \|q - q'\| + \|q' - q^\star\| < \delta$. The proof for the case when $\beta < \epsilon$ is analogous. □

We now state the main result of this section that relates GRAs and the CBR algorithm. In particular, we show that any GRA can be essentially implemented as a CBR algorithm.

Theorem 3. *Let $\delta > 0$. Suppose that, under a Geometric Rendezvous Algorithm, and the proximity graph $\mathscr{G}_1$, a formation converges to a point p. Then, there exists a proximity graph $\mathscr{G}_2$ so that there is a realization of the CBR algorithm that converges to a point $p^\star$, where $\|p - p^\star\| < \delta$. That is, any Geometric Rendezvous Algorithm can be approximated arbitrarily well by a Consensus-Based Rendezvous Algorithm.*

Proof. Since the formation converges to rendezvous under the Geometric Rendezvous Algorithm (GRA), there exists an integer M so that all of the agents are within $\delta/2$ of the rendezvous location p after M iterations.

Let the proximity graph $\mathscr{G}_2$ be an extension of $\mathscr{G}_1$; this means that the edge set for $\mathscr{G}_1$ satisfies $\mathscr{G}_{1\mathscr{E}} \subseteq \mathscr{G}_{2\mathscr{E}}$. For instance, if the proximity graph $\mathscr{G}_1$ is induced by the distance between the agents so that $(i,\ j) \in \mathscr{G}_{1\mathscr{E}}$ if and only if the distance between agents i and j is less than a certain threshold b_1, an extension of $\mathscr{G}_1$ can be obtained by setting the threshold b_2 for $\mathscr{G}_2$ to be $b_2 = b_1 + \delta$, for some suitable $\delta > 0$. We are doing this extension to guarantee that for any agent i, its neighboring set $\mathscr{N}_i$ under $\mathscr{G}_1$ is contained in the neighboring set $\mathscr{N}_i'$ under $\mathscr{G}_2$.

Let $p_i[k]$ be the location of agent i under the realization of the GRA; and let $p_i^\star[k]$ be its location under the approximation given by the CBR approach at time-step k. We will show by induction that at the kth iteration, the distance $\left\| p_i[k] - p_i^\star[k] \right\| < \frac{\delta}{2}\left(1 - 2^{-k}\right)$.

When the network is initially deployed, the locations under the GRA and the CBR coincide, so we have the initial step. Suppose now that the result holds at step k.

Let us look at agent i. Under the GRA, the agent is moving to a point $p_i[k+1]$ inside the convex hull given by the rest of the agents. From Lemma 6, this point can be represented as a convex combination of some of its neighbors under the GRA formation. Hence,

$$p_i[k+1] = \sum_{j \in \mathscr{N}_i} \lambda_j\, p_j[k]. \tag{8.38}$$

By induction hypothesis, we can write $p_j[k] = p_j^\star[k] + d_j[k]$, where $\|d_j[k]\| < \frac{\delta}{2}\left(1 - 2^{-k}\right)$. Substituting into (8.38), we obtain

$$p_i[k+1] = \sum_{j \in \mathcal{N}_i} \lambda_j p_j^\star[k] + \sum_{j \in \mathcal{N}_i} \lambda_j d_j[k]. \tag{8.39}$$

Let $p_i^{\star\prime}[k+1] = \sum_{j \in \mathcal{N}_i} \lambda_j p_j^\star[k]$. Observe that

$$\left\| p_i[k+1] - p_i^{\star\prime}[k+1] \right\| = \left\| \sum_{j \in \mathcal{N}_i} \lambda_j d_j[k] \right\| \leq \frac{\delta}{2}\left(1 - 2^{-k}\right). \tag{8.40}$$

Since $\mathcal{G}_2$ is an extension of $\mathcal{G}_1$, if we construct the point $p_i^\star[k+1]$ as in Lemma 8, with $\left\| p_i^\star[k+1] - p_i^{\star\prime}[k+1] \right\| < \delta/2^{k+2}$ then we guarantee that this point can be constructed by the agents under the CBR algorithm. Henceforth,

$$\begin{aligned} \left\| p_i[k+1] - p_i^\star[k+1] \right\| &= \left\| p_i[k+1] - p_i^{\star\prime}[k+1] + p_i^{\star\prime}[k+1] - p_i^\star[k+1] \right\| \\ &\leq \left\| p_i[k+1] - p_i^{\star\prime}[k+1] \right\| + \left\| p_i^{\star\prime}[k+1] - p_i^\star[k+1] \right\| \\ &\leq \frac{\delta}{2}\left(1 - 2^{-k}\right) + \delta/2^{k+2} = \frac{\delta}{2}\left(1 - 2^{-(k+1)}\right). \end{aligned} \tag{8.41}$$

Hence, after M iterations, since the agents under the GRA are within $\delta/2$ of the rendezvous location p, the CBR agents would be, at most, δ away from such point p. This implies that the convex hull of the CBR agents lies within a circle of radius δ centered at p. Since the agents evolving under CBR converge to a rendezvous location interior to its convex hull, they will converge to a point $p^\star$ with $\|p - p^\star\| < \delta$. This concludes the proof. □

Remark 8. The Theorem establishes that it is possible to approximate a GRA algorithm that converges to rendezvous, with the CBR algorithm. The proximity graph $\mathcal{G}_2$ introduced at the beginning of the proof is needed to ensure that the point $p_i^\star[k+1]$ for CBR can be constructed: in our proof, we need that for each agent i, its set of neighbors $\mathcal{N}_i$ under the GRA algorithm (defined by the proximity graph $\mathcal{G}_1$), is a subset of its set of neighbors under the CBR algorithm (defined by the proximity graph $\mathcal{G}_2$). If this were not the case, we would not be able to invoke Lemma 8 to construct the CBR realization. The extension $\mathcal{G}_2$ is used simply for purposes of the proof, and it is not necessary for a physical implementation of a GRA. The analysis for a finite M is necessary to guarantee that there is a uniform ϵ to implement the ϵ-centered convex combinations required for the CBR algorithm. Note also that since the convex hull of the original location of the formation describes a compact set in $\mathbb{R}^n$, ϵ can be chosen independently from the point p.

8.4.5 Simulations

The CBR algorithm was implemented by uniformly deploying 30 agents in a square region of side 10. At each time-step, each agent chooses randomly a convex combination of its neighbors. Each pair of neighbors has a probability $p_{i,j} < 1$ of failure. By adjusting the radius in which other agents can be detected, it is possible to

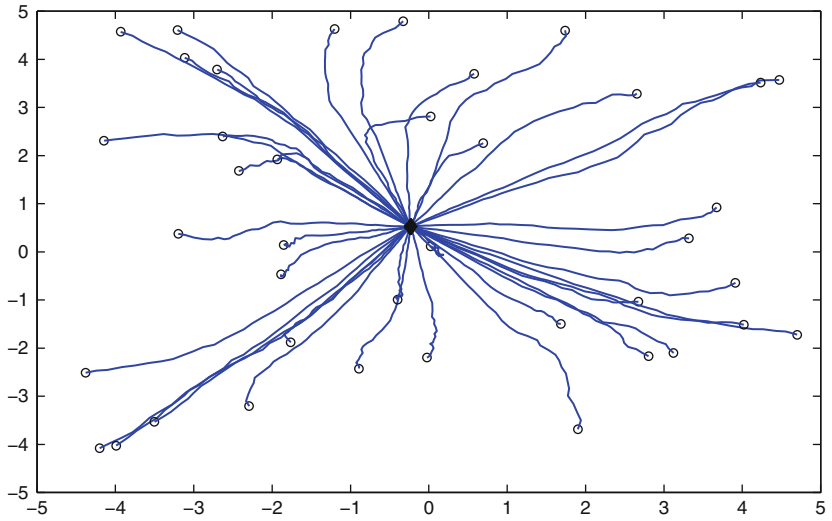


Fig. 8.1 Robot trajectories when the underlying graph is connected

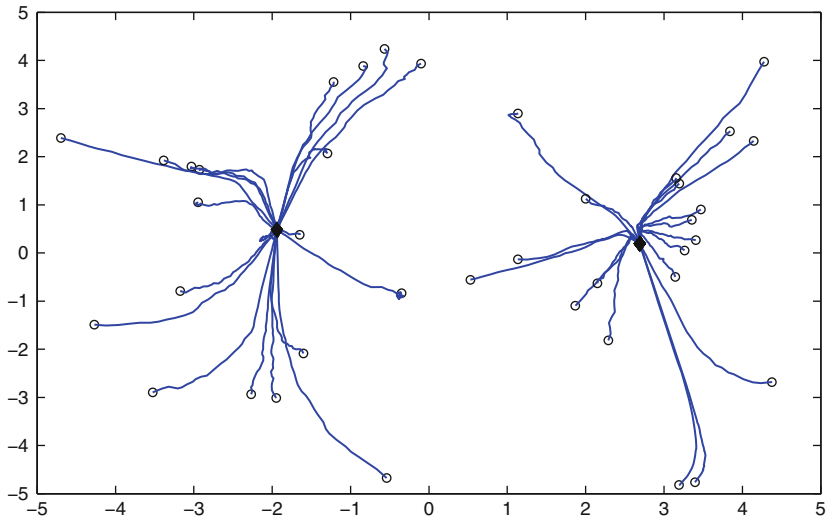


Fig. 8.2 Robot trajectories when the underlying graph is disconnected. The agents in each connected component achieve rendezvous

simulate the case where the real proximity graph is the complete graph with N vertices, all the way to the case in which it is a collection of disconnected graphs.

In confirmation to our theoretical results, when $\mathscr{G}(\mathscr{R})$ is jointly connected the formation achieves rendezvous, as shown in Fig. 8.1. When $\mathscr{G}(\mathscr{R})$ is not connected, the results confirm the theoretical prediction that rendezvous is achieved in each connected component, as shown in Fig. 8.2.

8.5 Conclusions

Two applications of consensus protocols were presented. First, we used them to present a robust and distributed counting strategy that allows each of the agents to know how many robots are in the network. We proved this algorithm to be robust to changes in both the communication topology and the size of the network.

Then, we presented a general family of algorithms for rendezvous, by exploiting the relationship between rendezvous and the consensus problems. The convergence of the proposed family of algorithms thus parallels convergence properties of the consensus protocols. The proposed algorithms are fully distributed (hence scalable), stable, robust under failures and require no communication between the agents as long as they are capable of evaluating the relative positions of their neighbors. We also show that most existing rendezvous algorithms can be seen as a special case of the CBR algorithm.

For both of our applications, we discussed the differences and constrains between synchronous and asynchronous implementations.

References

1. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5), 1999.
2. T.C. Aysal, M. Coates, and M. Rabbat. Rates of convergence of distributed average consensus using probabilistic quantization. In *Allerton Conference on Communication, Control and Computing*, 2007.
3. H.E. Bell. Gerschgorin's theorem and the zeros of polynomials. *American Mathematical Monthly*, 72, 1965.
4. D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, NJ, 1989.
5. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005*, 2005.
6. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6), 2006.
7. C.H. Caicedo-N. and M. Žefran. Balancing sensing and coverage in mobile sensor networks: A min-max based approach. In *Proceedings of the 46th IEEE Conferece on Decision and Control*, 2007.

8. C.H. Caicedo-N. and M. Žefran. Balancing sensing and coverage in mobile sensor networks: Aggregation based approach. In *ICRA Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*, 2007.
9. C.H. Caicedo-N. and M. Žefran. Consensus-based rendezvous. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 1031–1036, 2008.
10. C.H. Caicedo-N. and M. Žefran. Rendezvous under noisy measurements. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.
11. C.H. Caicedo-N. and M. Zefran. Probabilistic guarantees for rendezvous under noisy measurements. In *Proceedings of the 2009 American Control Conference*, pages 5180–5185, 2009.
12. R. Carli, F. Bullo, and S. Zampieri. Quantized average consensus via dynamic coding/decoding schemes. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4916–4921, 2008.
13. R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2008.
14. Chi-Tsong Chen. *Linear System, Theory and Design*. The Oxford Series in Electrical and Computer Engineering. Oxford University Press, New York, 1999.
15. J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8), 2006.
16. F. Cucker and E. Mordecki. Flocking in noisy environments. *Journal de Mathmatiques Pures et Appliques*, 89(3):278–296, 2008.
17. M.H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
18. A. Fagiolini, A. Bicchi, G. Dini, and I.M. Savino. Tolerating malicious monitors in detecting misbehaving robots. In *IEEE International Workshop on Safety, Security, and Rescue Robotics*, Tohoku University Aobayama Campus, Sendai, Japan, 2008.
19. M.J. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2), 1985.
20. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1–3):147–168, 2005.
21. J.R.P. French Jr. A formal theory of social power. *Psychological Review*, 63(3):181–194, 1956.
22. A. Garulli and A. Giannitrapani. A set-membership approach to consensus problems with bounded measurement errors. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.
23. C. Godsil and G. Royle. *Algebraic Graph Theory*. Number 207 in Graduate Texts on Mathematics. Springer, New York, 2001.
24. A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*, second edition. Addison-Wesley, Boston, 2003.
25. R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990.
26. M. Huang and J.H. Manton. Stochastic approximation for consensus seeking: Mean square and almost sure convergence. In *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.
27. A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transaction on Automatic Control*, 48(6), 2003.
28. M. Jelasity and A. Montresor. Epidemic-style proactive aggregation in large overlay networks. In *Proceedings of The 24th International Conference on Distributed Computing Systems (ICDCS 2004)*, pages 102–109, 2004.
29. A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.
30. Derek B. Kingston and Randal W. Beard. Discrete-time average-consensus under switching network topologies. In *American Control Conference*, 2006.
31. M. Krein and D. Milman. On extreme points of regular convex sets. *Studia Mathematica*, 9:133–138, 1940.

32. J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem – the asynchronous case. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 1926–1931, 2004.
33. N.A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, San Mateo, 1996.
34. S. Martínez. Practical rendezvous through modified circumcenter algorithms. In *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.
35. L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2), 2005.
36. R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 2007.
37. R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions On Automatic Control*, 49:1520–1533, s2004.
38. W. Ren, R.W. Beard, and E.M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.
39. L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.
40. L. Schenato and S. Zampieri. On the performance of randomized communication topologies for rendezvous control of multiple vehicles. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, 2007.
41. E. Seneta. *Non-negative Matrices and Markov Chains*, second edition. Springer Series in Statistics. Springer, New York, 1981.
42. J.N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1984.
43. J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), 1986.
44. Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, Aug 1995.
45. Roger Wattenhofer and Peter Widmayer. An inherent bottleneck in distributed counting. *Journal of Parallel and Distributed Computing*, 49(1), 1998.
46. Roger Wattenhofer and Peter Widmayer. The counting pyramid: an adaptive distributed counting scheme. *Journal of Parallel and Distributed Computing*, 64(4), 2004.
47. Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, second edition, 2001.
48. J. Yu, S. LaValle, and D. Liberzon. Rendezvous without coordinates. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.

Chapter 9
Design Principles of Wireless Sensor Networks Protocols for Control Applications

Carlo Fischione, Pangun Park, Piergiuseppe Di Marco, and Karl Henrik Johansson

Abstract Control applications over wireless sensor networks (WSNs) require timely, reliable, and energy efficient communications. This is challenging because reliability and latency of delivered packets and energy are at odds, and resource constrained nodes support only simple algorithms. In this chapter, a new system-level design approach for protocols supporting control applications over WSNs is proposed. The approach suggests a joint optimization, or co-design, of the control specifications, networking layer, the medium access control layer, and physical layer. The protocol parameters are adapted by an optimization problem whose objective function is the network energy consumption, and the constraints are the reliability and latency of the packets as requested by the control application. The design method aims at the definition of simple algorithms that are easily implemented on resource constrained sensor nodes. These algorithms allow the network to meet the reliability and latency required by the control application while minimizing for energy consumption. The design method is illustrated by two protocols: Breath and TREnD, which are implemented on a test-bed and compared to some existing solutions. Experimental results show good performance of the protocols based on this design methodology in terms of reliability, latency, low duty cycle, and load balancing for both static and time-varying scenarios. It is concluded that a system-level design is the essential paradigm to exploit the complex interaction among the layers of the protocol stack and reach a maximum WSN efficiency.

Keywords Wireless sensor networks · Networked Control · IEEE 802.15.4 · Optimization

C. Fischione (✉)
ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden
e-mail: carlofi@ee.kth.se

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_9,

9.1 Introduction

Wireless sensor networks (WSNs) are networks of tiny sensing devices for wireless communication, actuation, control, and monitoring. Given the potential benefits offered by these networks, e.g., simple deployment, low installation cost, lack of cabling, and mobility, they are specially appealing for control applications in home and industrial automation [1–3]. The variety of application domains and theoretical challenges for WSNs has attracted research efforts for more than one decade. Nevertheless, a lively research and standardization activity is ongoing [2–5] and there is not yet a widely accepted protocol stack for WSNs for control applications.

The lack of efficient protocol solutions is due to that the protocols for control applications face complex control and communication requirements. Traditional control applications are usually designed by a top-down approach from a protocol stack point of view, whereby most of the essential aspects of the network and sensing infrastructure that has to be deployed to support control applications are ignored. Here, packet losses and delays introduced by the communication network are considered as nonidealities and uncertainties and the controllers are tuned to cope with them without having any influence on them. The top-down approach is limited for two reasons: (1) it misses the essential aspect of the energy efficiency that is usually required to WSNs [2], and (2) it can be quite conservative and therefore inefficient, because the controllers are built by presuming worst case wireless channel conditions that may be rarely experienced in the reality. On the other side, protocols for WSNs are traditionally designed to maximize the reliability and minimize the delay. This is a bottom-up approach, where controller specifications are not explicitly considered even though the protocols are used for control. This approach is energy inefficient because high reliability and low latency may demand significant energy consumption [2, 6]. Therefore, it follows that there is the essential need of a new design approach.

Traditional WSNs applications (e.g., monitoring) need a high probability of success in the packet delivery (reliability). In addition to reliability, control applications ask also for timely packet delivery (latency). If reliability and latency constraints are not met, the correct execution of control decisions may be severely compromised, thus creating unstable control loops [7]. High reliability and low latency may demand significant energy expenditure, thus reducing the WSN lifetime. Controllers can usually tolerate a certain degree of packet losses and delay [8]: For example, the stability of a closed-loop control system may be ensured by high reliable communications and large delays, or by low delays when the packet loss is high. In contrast to monitoring applications, for control applications there is no need to maximize the reliability. A tradeoff between latency, packet losses, and stability requirements can be exploited for the benefit of the energy consumption, as proposed by the system-level design approach [9]. Therefore, we claim that the protocol design for control needs a system-level approach whereby the need of a parsimonious use of energy and the typical requirements of the control applications are jointly taken into account and control and WSNs protocols are co-designed.

Exploiting such a tradeoff poses extra challenges when designing WSNs protocols for control applications when compared to more traditional communication networks, namely:

- Reliability: Sensor information must be sent to the sink of the network with a given probability of success, because missing these data could prevent the correct execution of control actions or decisions concerning the phenomena sensed. However, maximizing the reliability may increase substantially the network energy consumption [2]. Hence, the network designers need to consider the tradeoff between reliability and energy consumption.
- Latency: Sensor information must reach the sink within some deadline [6]. A probabilistic delay requirement must be considered instead of using average packet delay since the delay jitter can be too difficult to compensate for, especially if the delay variability is large [10]. Retransmission of old data to maximize the reliability may increase the delay and is generally not useful for control application [7].
- Energy efficiency: The lack of battery replacement, which is essential for affordable WSN deployment, requires energy-efficient operations. Since high reliability and low delay may demand a significant energy consumption of the network, thus reducing the WSN lifetime, the reliability and delay must be flexible design parameters that need to be adequate for the requirements. Note that controllers can usually tolerate a certain degree of packet losses and delay [8–12]. Hence, the maximization of the reliability and minimization of the delay are not the optimal design strategies for the control applications we are concerned within this chapter.
- Adaptation: The network operation should adapt to application requirement changes, varying wireless channel and network topology. For instance, the set of control application requirements may change dynamically and the communication protocol must adapt its design parameters according to the specific requests of the control actions. To support these changing requirements, it is essential to have an analytical model describing the relation between the protocol parameters and performance indicators (reliability, delay, and energy consumption).
- Scalability: Since the processing resources are limited, the protocol procedures must be computationally light. These operations should be performed within the network to avoid the burden of too much communication with a central coordinator. This is particularly important for large networks. The protocol should also be able to adapt to size variation of the network for example, caused by moving obstacles, or addition of new nodes.

In this chapter, we propose a design methodology for WSNs protocols for control application that embraces the issues mentioned above. The remainder of the chapter is organized as follows: in Sect. 9.2, we discuss the related literature. In Sect. 9.3, we describe the design method. Such a method is then applied in Sect. 9.4, where the Breath protocol is described and experimentally evaluated, and in Sect. 9.5, where the TREnD protocol is described and experimentally tested. In Sect. 9.6, we conclude the chapter.

9.2 Background

Although the standardization process for WSNs is ongoing, there is not any widely accepted complete protocol stack for WSNs for control [2]. The IEEE 802.15.4 protocol [4], which specifies physical layer and medium access control (MAC), is the base of recent solutions in industrial environments as WirelessHART [13], ISA100 [2], and ROLL [5]. Hence, we consider IEEE 802.15.4 as the reference standard in our investigation.

There have been many contributions to the problem of protocol design for WSNs, both in academia (e.g., [2, 14]) and in industry (e.g., [15–17]). New protocols have been built around standardized low-power protocols, such as IEEE 802.15.4 [4], Zigbee [18], and WirelessHART. WirelessHART is a promising solution for the replacement of the wired HART protocol in industrial contexts. However, the power consumption is not a main concern in WirelessHART, whereas the data link layer is based on Time Division Multiple Access (TDMA), which requires time synchronization and pre-scheduled fixed length time-slots by a centralized network manager. Such a manager should update the schedule frequently to consider reliability and delay requirements and dynamic changes of the network, which demands complex hardware equipments. WirelessHART is thus in contrast with the necessity of simple protocols able to work with limited energy and computing resources. In Table 9.1, we summarize the characteristics of the protocols that are relevant to the category of applications we are concerned with in this chapter. In the table, we have evidenced whether indications as energy **E**, reliability **R**, and delay **D** have been included in the protocol design and validation, and whether a cross-layer approach has been adopted. We discuss these protocols in the following.

Table 9.1 Protocol comparison. The letters **E**, **R**, and **D** denote energy, reliability, and communication delay

Protocol	*E*	*R*	*D*	Layer
GAF [19]	◯	·	·	bridge
SPAN [20]	◯	·	·	bridge
XMAC [21]	⊕	·	·	MAC
Flush [23]		⊕		MAC
Fetch [25]	·	⊕	·	phy, MAC, routing
GERAF [27]	◯		◯	MAC, routing
Dozer [26]	⊕	⊕		MAC, routing
MMSPEED [28]		◯	◯	routing
Breath	⊕	⊕	⊕	phy, MAC, routing
TREnD	⊕	⊕	⊕	phy, MAC, routing

The circle denotes that a protocol is designed by considering the indication of the column, but it has not been validated experimentally. The circle with plus denotes that the protocol is designed by considering the indication and experimentally validated. The dot denotes that the protocol design does not include indication and hence cannot control it, but simulation or experiment results include it. The term "bridge" means that the protocol is designed by bridging MAC and routing layers

GAF [19], SPAN [20], and X-MAC [21] consider the energy efficiency as a performance indicator, which is attained by algorithms under the routing layer and above the MAC layer so-called bridge layer. Simulation results of reliability and delay are reported in [19,20]. These protocols have not been designed out of an analytical modeling of reliability and delay, so there is not systematic control of them. One of the first protocol for WSNs designed to offer a high reliability is RMST [22], but energy consumption of the network or delay have not been accounted for in this protocol. The same lack of energy efficiency and delay requirements can be found in the reliable solutions presented in [23–25]. Dozer [26] comprises the MAC and routing layer to minimize the energy consumption while maximizing the reliability of the network, but an analytical approach has not been followed. Specially, Fetch [25] and Dozer [26] are designed for monitoring application, which mainly deals with lower traffic load than control applications. The latency of Fetch [25] is significantly dependent on the depth of the routing tree and is around some hundred seconds. In addition, experimental results of Dozer [26] show good energy efficiency and reliability under very low traffic intensity (with data sampling interval of 120 s) but the delay in the packet delivery is not considered, which is essential for control applications [7, 8]. Energy efficiency with delay requirement for MAC and randomized routing is considered in GERAF [27], without simulation or experimental validation.

The focuses of the protocols mentioned above [19–27] are the maximization of the energy efficiency or reliability, or just minimization of the delay, without considering simultaneously application requirements in terms of reliability and delay in the packet delivery. In other words, these protocols are mostly designed for monitoring applications and do not support typical control application requirements. Control and industrial applications are able to cope with a certain degree of packet losses and delay [8, 10, 11], which implies that the approaches followed in the protocols mentioned above are not the ideal solution for these applications. The maximization of the energy efficiency and reliability may give a long delay, which are bad for the stability of the closed-loop control system. Analogously, the maximization of the reliability may be energy demanding and may give long delay, all of which are not tolerable for control applications. In addition, the protocols mentioned above do not support an adaptation to the changes of the reliability and delay, which may be required by the controllers.

The protocols MMSPEED [28] and SERAN [9] are appealing for control and industrial applications. However, MMSPEED is not energy efficient because it considers a routing technique with an optimization of reliability and delay without energy constraints. The protocol satisfies a high reliability requirement by using duplicated packets over multi-path routing. Duplicated packets increase the traffic load with negative effect on the stability and energy efficiency of the network. In SERAN, a system-level design methodology has been presented for industrial applications, but even though SERAN allows the network to operate with low energy consumption subject to delay requirements, it considers neither tunable reliability requirements nor duty-cycling policies, which are essential to

reduce energy consumption. Furthermore, SERAN focuses on low-traffic networks. These characteristics limit the performance of SERAN in terms of both energy and reliability in our application setup.

Given the availability of numerous techniques to reduce energy consumption and ensure reliability and low delays, a cross-layer optimization is a natural approach for WSNs protocols. Some cross-layer design challenges of the physical, MAC, and network layers to minimize the energy consumption of WSNs have been surveyed in [29–31]. However, many of the cross-layer solutions proposed in the literature are hardly useful for the application domain we are targeting, because they require sophisticated processing resources, or instantaneous global network knowledge, which are out of the capabilities of real nodes. Moreover, the requirements of the control applications are not taken into account.

In the following sections, we present a general approach to the design of WSNs protocols for control. We illustrate this approach by the protocols Breath and TREnD, which are presented in this chapter. These protocols embrace simultaneously the physical layer, the MAC layer, the networking layer, and the control application layer, see Table 9.1.

9.3 Protocol Design for Control Applications

In this chapter, we are concerned with the design of protocols for WSNs used to close the loop between plants and controllers, see Fig. 9.1. The nodes connected to the plants take state information and transmit it to the sink via a multi-hop WSN. The controllers are attached to the sink of the network. The sink must receive packets from the nodes of the plants with a desired probability of success and within a latency constraint demanded by the controllers so that the control decision can be correctly taken. We assume that the communication network must be energy efficient to guarantee a long network lifetime.

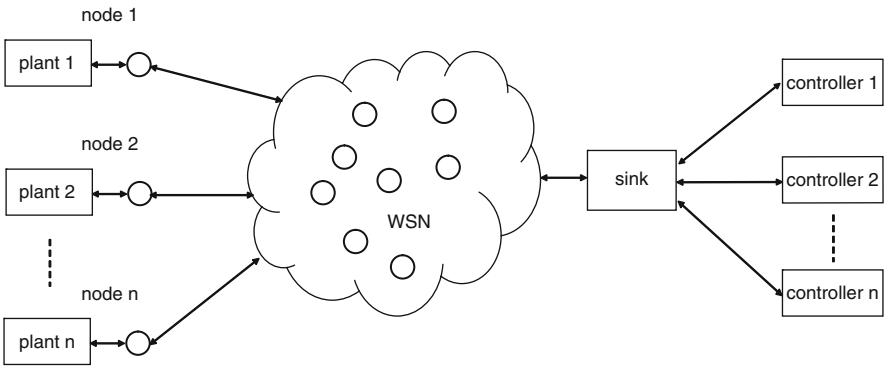


Fig. 9.1 Control over a wireless sensor network. There are n plants and n controllers. A network closes the loop from sensors attached to plants to the controllers

Give the system model reported in Fig. 9.1, we formulate the new WSNs protocol design approach for control applications by the following optimization problem, where the parameters of the physical layer, the MAC layer, the networking layer, and the control application layer are co-designed by a joint optimization:

$$\min_{\mathrm{x}} \quad E_{\mathrm{tot}}(\mathrm{x}) \tag{9.1a}$$

$$\text{s.t.} \quad R_i(\mathrm{x}) \geq \Omega_i, \qquad i = 1, \ldots, n, \tag{9.1b}$$

$$\Pr[D_i(\mathrm{x}) \leq \tau_i] \geq \Delta_i \qquad i = 1, \ldots, n, \tag{9.1c}$$

Each decision variable is a protocol parameter. These decision variables are denoted by the vector x, which can include the radio power used by the nodes, the MAC parameters (e.g., the access probability or the TDMA slot duration, or the random backoff of IEEE 802.15.4 MAC, etc.), and the routing parameters. $E_{\mathrm{tot}}(\mathrm{x})$ is the total energy consumption of the network, which is obviously function of the protocol parameters. $R_i(\mathrm{x})$ is the probability of successful packet delivery (reliability) from node i to the sink, and Ω_i is the minimum desired probability. $D_i(\mathrm{x})$ is a random variable describing the delay to transmit a packet from node i to the sink. τ_i is the desired maximum delay, and Δ_i is the minimum probability with which such a maximum delay should be achieved. Notice that $E_{\mathrm{tot}}(\mathrm{x})$, $R_i(\mathrm{x})$, $D_i(\mathrm{x})$, $i = 1, \ldots, n$, are implicit functions of the wireless channel, network topology, and traffic load generated by the nodes.

We remark that τ_i, Δ_i, and Ω_i are the *control requirements*, and x collects the *protocol parameters* that must be adapted to the wireless channel conditions, network topology, and the control requirements for an efficient network operation. Since controllers may need some reliability and latency during certain time intervals, and a different pair of requirements during some other time intervals, the reliability and latency requirements may change dynamically depending on the state of the plant and the history of the control actions. This means that problem (9.1) must be solved periodically and nodes need to know the optimal solution to adapt their protocol operation so that a global optimum can be achieved. It follows that the proposed design method allows us to perform a systematic co-design between the control requirements of the application and the network energy consumption.

Note that problem (9.1) can be a mixed integer-real optimization problem, because some of the protocol parameters can take on integer values. The solution to this problem can be obtained at the sink node, or by distributed algorithms, where each node performs local computation to achieve the optimal global solution. However, as it was noted in [32], the complex interdependence of the decision variables (sleep disciplines, clustering, MAC, routing, power control, etc.) lead to difficult optimization problems even in simple network topologies, where the analytical relations describing reliability, latency, and energy consumption may be highly nonlinear expressions. Such a difficulty is further exacerbated when considering non-TDMA scheme [33]. We propose a design approach that offers a computationally attractive solution by simplifications of adequate accuracy. We show how to model the relations of problem (9.1) and compute the solution to the problem in

the following two sections. Specifically, in Sect. 9.4 we apply this design method to a system where there is only a plant that needs to be controlled over a multi-hop network, whereas in Sect. 9.5, the design method is applied to a system where there are multiple plants.

9.4 Breath

The Breath protocol is designed for the scenario depicted in Fig. 9.2, where a plant is remotely controlled over a WSN [7, 11]. Outputs of the plant are sampled at periodic intervals by the sensors with total packet generation rate of λ pkt/s. We assume that packets associated with the state of the plant are transmitted to a sink, which is connected to the controller, over a multi-hop network of uniformly and randomly distributed relaying nodes. No direct communication is possible between the pant and the sink. Relay nodes forward incoming packets. When the controller receives the measurements, they are used in a control algorithm to compensate the control output. The control law induces constraints on the communication delay and the packet loss probability. Packets must reach the sink within some minimum reliability and maximum delay. The application requirements are chosen by the control algorithm designers. Since they can change from one control algorithm to another, or a control algorithm can ask to change the application requirements from time to time, we allow them to vary. We assume that nodes of the network cannot be recharged, so the operations must conserve energy. The system scenario is quite general, because it applies to any interconnection of a plant by a multi-hop WSN to a controller tolerating a certain degree of data loss and delay [8–12].

A typical example of the scenario described above is an industrial control application. In particular, a WSN with nodes uniformly distributed in the environment (e.g., in the ceiling) can be deployed as the network infrastructure that support the

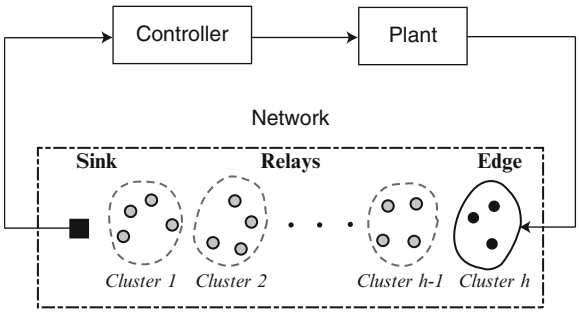


Fig. 9.2 Wireless control loop. An wireless network closes the loop from sensors to controller. The network includes nodes (*black dots*) attached to the plant, $h - 1$ relay clusters (*grey dots*), and a sink (*black rectangular*) attached to the controller. Note that the lines between the sink and the controller, and between the controller and the plant are not communication links, but control links

control of the state of the robots in a manufacturing cell [13, 34]. Typically, a cell is a stage of an automation line. Its physical dimensions range around 10 or 20 m on each side. Several robots cooperate in the cell to manipulate and transform the same production piece. Each node senses the state and has to report the data to the controller within a maximum delay. The decision making algorithm runs on the controller, which is usually a processor placed outside the cell. Multi-hop communication is needed to overcome the deep attenuations of the wireless channel due to moving metal objects and save energy consumption.

The Breath protocol groups all N nodes between the cluster of nodes attached to the plan and the sink with $h - 1$ relay clusters. Data packets can be transmitted only from a cluster to the next cluster closer to the sink. Clustered network topology is supported in networks that require energy efficiency, since transmitting data through relays consumes less energy than routing directly to the sink [35]. In [36], a dynamic clustering method adapts the network parameters. In [35] and [37], a cluster header is selected based on the residual energy levels for clustered environments. However, the periodic selection of clustering may not be energyefficient, and does not ensure the flexibility of the network to a time-varying wireless channel environment. A simpler geographic clustering is instead used in Breath. Nodes in the forwarding region send short beacon messages when they are available to receive data packets. Beacon messages are exploited to carry information related to the control parameters of the protocol.

In the following sections, we will describe the protocol stack and state machine of Breath in Sects. 9.4.1 and 9.4.2, respectively.

9.4.1 The Breath Protocol Stack

Breath uses a randomized routing, a hybrid TDMA at the MAC, radio power control at the physical layer, and sleeping disciplines. We give details in the following.

In many industrial environments, the wireless conditions vary heavily because of moving metal obstacles and other radio disturbances. In such situations, routing schemes that use fixed routing tables are not able to provide the flexibility over mobile equipments, physical design limitations, and reconfiguration typical of an industrial control application. Fixed routing is inefficient in WSNs due to the cost of building and maintaining routing tables. To overcome this limitation, routing through a random sequence of hops has been introduced in [27]. The Breath protocol is built on an optimized random routing, where next hop route is efficiently selected at random. Randomized routing allows us to reduce overhead because no node coordination or routing state needs to be maintained by the network. Robustness to node failures is also considerably increased by randomized routing. Therefore, nodes route data packets to next-hop nodes randomly selected in a forwarding region.

Each node, either transmitter or receiver, does not stay in an active state all time, but goes to sleep for a random amount of time, which depends on the traffic and channel conditions. Since traffic, wireless channel, and network topology may be

timevarying, the Breath protocol uses a randomized duty-cycling algorithm. Sleep disciplines turn off a node whenever its presence is not required for the correct operation of the network. GAF [19], SPAN [20], and S-MAC [38] focus on controlling the effective network topology by selecting a connected set of nodes to be active and turning off the rest of the nodes. These approaches require extra communication, since nodes maintain partial knowledge of the state of their individual neighbors. In Breath, each node goes to sleep for an amount of time that is a random variable dependent on traffic and network conditions. Let μ_c be the cumulative wake-up rate of each cluster, i.e., the sum of the wake-up rates that a node sees from all nodes of the next cluster. The cumulative wake-up rate of each cluster must be the same for each cluster to avoid congestions and bottlenecks.

The MAC of Breath is based on a CSMA/CA mechanism similar to IEEE 802.15.4. Both data packets and beacon packets are transmitted using the same MAC. Specifically, the CSMA/CA checks the channel activity by performing clear channel assessment (CCA) before the transmission can commence. Each node maintains a variable NB for each transmission attempt, which is initialized to 0 and counts the number of additional backoffs the algorithm does while attempting the current transmission of a packet. Each backoff unit has duration T_{ca} ms. Before performing CCAs a node takes a backoff of random$(0, W - 1)$ backoff units i.e., a random number of backoffs with uniformly distributed over $0, 1, \ldots, W - 1$. If the CCA fails, i.e., the channel is busy, NB is increased by one and the transmission is delayed of random$(0, W - 1)$ backoff periods. This operation is repeated at most M_{ca} times, after which a packet is discarded.

The Breath protocol assumes that each node has a rough knowledge of its location. This information, which is commonly required for the applications we are targeting [2], can be obtained running a coarse positioning algorithm, or using the Received Signal Strength Indicator (RSSI), which is typically provided by off-the-shelf sensor nodes [39]. Some radio chip already provide a location engine based on RSSI [40]. Location information is needed for tuning the transmit radio power and to change the number of hops, as we will see later. The energy spent for radio transmission plays an important role in the energy budget and for the interference in the network. Breath, therefore, includes an effective radio power control algorithm.

9.4.2 State Machine Description

Breath distinguishes between three node classes: edge nodes, relays, and the sink.

The edge nodes wake-up as soon as they sense packets generated by the plant to be controlled. Before sending packets, an edge node waits for a beacon message from the cluster of nodes closer to the edge. Upon the reception of a beacon, the node sends the packet.

Consider a relay node k. Its detailed behavior is illustrated by the state machine of Fig. 9.3, as we describe in the following:

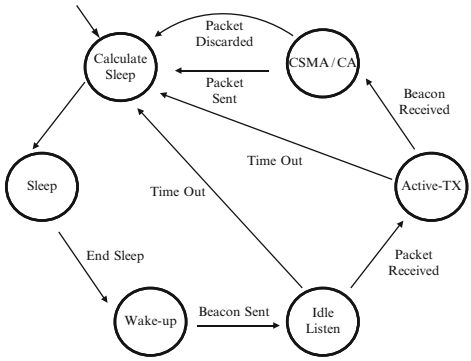


Fig. 9.3 State machine description of a relay node executing the Breath protocol

- *Calculate Sleep State*: the node calculates the parameter μ_k for the next sleeping time and generates an exponentially distributed random variable having average $1/\mu_k$. After this, the node goes back to the Sleep State. μ_k is computed such that the cumulative wake-up rate of the cluster μ_c is ensured.
- *Sleep State*: the node turns off its radio and starts a timer whose duration is an exponentially distributed random variable with average $1/\mu_k$. When the timer expires, the node goes to the Wake-up State.
- *Wake-up State*: the node turns its beacon channel on, and broadcasts a beacon indicating its location. Then, it switches to listen to the data channel, and it goes to the Idle Listen State.
- *Idle Listen State*: the node starts a timer of a fixed duration that must be long enough to receive a packet. If a data packet is received, the timer is discarded, the node goes to the Active-TX State, and its radio is switched from the data channel to the beacon channel. If the timer expires before any data packet is received, the node goes to the Calculate Sleep State.
- *Active-TX State*: the node starts a waiting timer of a fixed duration. If the node receives the first beacon coming from a node in the forwarding region within the waiting time, it retrieves the node ID and goes to the CSMA/CA State. Otherwise if the waiting timer is expired before receiving a beacon, the node goes to the Calculate Sleep State.
- *CSMA/CA State*: the node switches its radio to hear the data channel, and it tries to send a data packet to a node in the next cluster by the CSMA/CA MAC. If the channel is not clean within the maximum number of tries, the node discards the data packet and goes to the Calculate Sleep State. If the channel is clear within the maximum number of attempts, the node transmits the data packet using an appropriate level of radio power and goes to the Calculate Sleep State.

The sink node sends periodically beacon messages to the last cluster of the network to receive data packets. Such a node estimates periodically the traffic rate and the wireless channel conditions. By using this information, the sink runs an algorithm to optimize the protocol parameters, as we describe in the following section. Once the results of the optimization are achieved, they are communicated to the nodes by beacons.

According to the protocol given above, the packet delivery depends on the traffic rate, the channel conditions, number of forwarding regions, and the cumulative wake-up time. In the next sections, we show how to model and optimize online these parameters.

9.4.3 Protocol Optimization

The protocol is optimized dynamically by the constrained optimization problem (9.1). The objective function, denoted by $E_{\text{tot}}(h, \mu_{\text{c}})$, is the total energy consumption for transmitting and receiving packets from the edge cluster to the sink. The constraint are given by the end-to-end packet reception probability and end-to-end delay probability. In the Breath specific case, problem (9.1) is written in the following form:

$$\min_{h,\mu_{\text{c}}} \quad E_{\text{tot}}(h, \mu_{\text{c}}) \tag{9.2a}$$

$$\text{s.t.} \quad R(h, \mu_{\text{c}}) \geq \Omega, \tag{9.2b}$$

$$\Pr[D(h, \mu_{\text{c}}) \leq \tau] \geq \Delta, \tag{9.2c}$$

$$h \geq 2, \tag{9.2d}$$

$$\mu_{\min} \leq \mu_{\text{c}} \leq \mu_{\max}. \tag{9.2e}$$

The decision variables are the cumulative wake-up rate μ_{c} of each cluster and the number of relay clusters, $h - 1$. $R(h, \mu_{\text{c}})$ is the probability of successful packet delivery (reliability) from the edge cluster to the sink, and Ω is the minimum desired probability. $D(h, \mu_{\text{c}})$ is a random variable describing the delay to transmit a packet from the edge cluster to the sink. τ is the desired maximum delay, and Δ is the minimum probability with which such a maximum delay should be achieved. Constraint (9.2d) is due to that there is at least two hops from the edge cluster to the sink. Constraint (9.2e) is due to that the wake-up rate cannot be less than a minimum value $\mu_{\min}$, and larger than a maximum value $\mu_{\max}$ due to hardware reasons. Note that Problem (9.2) is a mixed integer-real optimization problem, because μ_{c} is real and h is integer. We need to have Δ and Ω close to one. We let $\Delta \geq 0.95$ and $\Omega \geq 0.9$, namely we assume that the delay τ must be achieved at least with a probability of 95%, and the reliability must be larger than 90%. We remark that τ, Δ, and Ω are application requirements, and h, μ_{c} and nodes' radio transmit power are *protocol parameters* that must be adapted to the traffic rate λ, the wireless channel conditions, and the application requirements for an efficient network operation.

In the following, we shall propose an approach to model the quantities of Problem (9.2), along with a strategy to achieve the optimal solution, namely the values of h^* and μ_{c}^* that minimize the cost function and satisfy the application requirements. As we will see later, the system complexity prevents us to derive the exact expressions for the analytical relations of the optimization problem. An approximation of the requirements and an upper bound of the energy consumption will be used.

9.4.3.1 Reliability Constraint

In this subsection, we provide an analytical expression for the reliability constraint (9.2b) in Problem (9.2). We have the following result

Claim. The reliability constraint (9.2b) is

$$R(h, \mu_c) = \prod_{i=1}^{h} p_i \sum_{n=1}^{\infty} \psi_{sb}(n)\psi_{sc}(n), \tag{9.3}$$

where p_i denotes the probability of successful packet reception during a single-hop transmission from cluster i to cluster $i-1$, n is the number of nodes in the cluster, $1/\lambda$ is the data packet transmission period,

$$\psi_{sc}(n) = \frac{\zeta(1-\zeta)^{n-1}}{1-(1-\zeta)^n},$$

where

$$\zeta = \sum_{i=0}^{m} b_{i,0} = \frac{2}{W+3}.$$

Proof. A proof is provided in [41]. □

Since the components of the sum in (9.3) with $n \geq 2$ give a small contribution, we set $n = 2$. In [41], we see that (9.3) provides a good approximation of the experimental results because it is always around 5% of the experiments for reliability values of practical interest (larger than 0.7). The same behavior is found for h up to 4.

We can rewrite the reliability constraint $R(h, \mu_c) \geq \Omega$ by using (9.3) with $n = 2$, thus obtaining

$$\begin{aligned} \mu_c \geq f_r(h, \Omega) &\triangleq \lambda \ln(2C_r) \\ &- \lambda \ln\left[C_r - 1 + \sqrt{(C_r-1)^2 - 4C_r\left(\Omega^{1/h}/p_{\min} - 1\right)}\right], \end{aligned} \tag{9.4}$$

where $C_r = \zeta(1-\zeta)/(1-(1-\zeta)^2)$, and $p_{\min} = \min(p_1, \ldots, p_h)$. Note that we used the worst channel condition of the network $p_{\min}$, which is acceptable for optimization purpose because in doing so we consider the minimum of (9.3). Since the argument of the square root in (9.4) must be positive, an additional constraint is introduced:

$$h \leq h_r \triangleq \frac{\ln(\Omega)}{\ln(p_{\min})}. \tag{9.5}$$

We will use (9.4) and (9.5) in Sect. 9.4.4 to find the solution of Problem (9.2). Now, we turn our attention to the delay constraint.

9.4.3.2 Delay Constraint

In this section, we give the expression for the delay constraint in (9.2c). The delay distribution is approximated by a Gaussian random variable. We have the following result:

Claim. The delay constraint given by (9.2c) is approximal by

$$\Pr[D \leq \tau] \approx 1 - Q\left(\frac{\tau - \mu_D}{\sigma_D}\right) \geq \Delta, \tag{9.6}$$

where

$$\mu_D = \frac{h}{\mu_c} + \frac{h(M_{ca}+1)(W-1)T_{ca}}{2}, \tag{9.7}$$

$$\sigma_D^2 = \frac{h}{\mu_c^2} + \frac{h(M_{ca}+1)(W^2-1)T_{ca}^2}{12}. \tag{9.8}$$

and $Q(x) = 1/\sqrt{2\pi}\int_x^\infty \mathrm{e}^{-t^2/2}\mathrm{d}t$.

Proof. A proof is provided in [41]. □

We are now in the position to express the delay constraint in Problem (9.2) by using (9.7) and (9.8) that we just derived. After some manipulations, it follows that (9.6) can be rewritten as

$$\mu_c \geq \frac{12\,C_{d1}\,h + 2\sqrt{3\,C_{d3}\,h\left[12\,C_{d1}^2 + C_{d2}\,(h - C_{d3})\right]}}{12\,C_{d1}^2 - C_{d2}\,C_{d3}},$$

where

$$C_{d1} = \tau - \frac{h(M_{ca}+1)(W-1)T_{ca}}{2},$$
$$C_{d2} = h(M_{ca}+1)(W^2-1)T_{ca}^2,$$
$$C_{d3} = \left(Q^{-1}(1-\Delta)\right)^2.$$

Since $T_{ca}^2 = 0.1024 \times 10^{-6}$ [15], and h, M_{ca}, W are positive integers, it follows that $T_{ca}^2 \ll h(M_{ca}+1)(W^2-1)$. Then $C_{d2} \ll C_{d1}$ and (9.6) is approximated by

$$\mu_c \geq f_d(h,\tau,\Delta) \triangleq \frac{2\left[h + Q^{-1}(1-\Delta)\sqrt{h}\right]}{2\tau - h(M_{ca}+1)(W-1)T_{ca}}. \tag{9.9}$$

Inequality (9.9) has been derived under the additional constraint

$$h \leq h_d \triangleq \frac{2\tau}{(M_{\text{ca}}+1)(W-1)T_{\text{ca}}}. \tag{9.10}$$

We will use (9.9) and (9.10) in Sect. 9.4.4 to find the solution of the optimization problem (9.2). Now, we investigate the total energy consumption.

9.4.3.3 Energy Consumption

In this subsection, we give an approximation of the energy consumption. Such an energy is the sum of the energy for data packet communication, plus the energy for control signaling. First, we need some definitions. Each time a node wakes up, it spends an energy given by the power needed to wake-up A_{w} during the wake-up time T_{w}, plus the energy to listen for the reception of a data packet within a maximum time T_{ac}. After a node wakes up, it transmits a beacon to the next cluster. Let the wireless channel loss probability be $1-p_i$ of i cluster, then nodes of $i-1$ cluster have to wake-up on average $1/p_i$ times to create the effect of a single wake-up so that a transmitter node successfully receives a beacon. Recalling that there are h hops and a cumulative wake-up rate per cluster μ_{c}, the total cost in a time T. Let E_r be the fixed cost of the RF circuit for the reception of a data packet. Let $E_{\text{ca}}(\mu_{\text{c}})$ be the energy spent during the CSMA/CA state.

We have the following result:

Claim. The total energy consumption is

$$\begin{aligned} E_{\text{tot}}(h,\mu_{\text{c}}) = T\lambda \Bigg[& Q_m\left(\frac{S}{h-1}\right) + Q_m\left(\frac{S}{h-1}\right)(h-1) \\ & \times u(h-1) + h\left(\frac{A_{\text{rx}}}{\mu_{\text{c}}} + E_{\text{ca}}(\mu_{\text{c}}) + E_r\right)\Bigg] \\ & + \frac{T\mu_{\text{c}}}{p_{\min}}\Bigg[2Q_b\left(\frac{S}{h-1}\right) + Q_b\left(\frac{2S}{h-1}\right)(h-2) \\ & \times u(h-2) + h\left(A_{\text{w}}T_{\text{w}} + A_{\text{rx}}(T_{\text{ac}}-T_{\text{w}})\right)\Bigg], \end{aligned} \tag{9.11}$$

where $p_{\min}$ is the worst reception probability, $Q_b(d_i)$ and $Q_m(d_i)$ are the expected energy consumption to transmit a beacon message and the energy consumption for radio transmission, respectively, where d_i is the transmission distance to which a data packet has to be sent.

Proof. A proof is provided in [41]. □

9.4.4 Optimal Protocol Parameters

In this section, we give the optimal protocol parameters used by Breath. Consider the reliability and delay constraints, and the total energy consumption as investigated in

Sects. 9.4.3.1, 9.4.3.2, and 9.4.3.3. The optimization problem (9.2) becomes

$$\begin{aligned} \min_{h,\mu_c} \quad & E_{\text{tot}}(h,\mu_c) \\ \text{s.t.} \quad & \mu_c \geq \max(f_r(h,\Omega),\, f_d(h,\tau,\Delta)), \\ & 2 \leq h \leq \min(h_r, h_d), \\ & \mu_{\min} \leq \mu_c \leq \mu_{\max}, \end{aligned} \tag{9.12}$$

where the first constraint comes from (9.4) and (9.5), and the second from (9.9) and (9.10). We assume that this problem is feasible. Infeasibility means that for any $h = 2, \ldots, \min(h_r, h_d)$, then $\mu_c \geq \max(f_r(h,\Omega),\, f_d(h,\tau,\Delta)) > \mu_{\max}$, namely it is not possible to guarantee the satisfaction of the reliability and delay constraint given the application requirements. This means that the application requirements must be relaxed, so that feasibility is ensured and the problem can be solved. The solution of this optimization problem, h^* and μ_c^*, is derived in the following.

By using the numerical values given for the Tmote sensors [15] for all the constants in the optimization problem, we see that the cost function of Problem (9.12) is increasing in h and convex in μ_c. This allows us to derive the optimal solution in two steps: for each value of $h = 2, \ldots, \min(h_r, h_d)$, the cost function is minimized for μ_c, achieving $\mu_c^*(h)$. Then, the optimal solution is found in the pair $h, \mu_c^*(h)$ that gives the minimum energy consumption. We describe this procedure next.

Let h be fixed. From the properties the cost function of Problem (9.12), the optimal solution $\mu_c^*(h)$ is attained either at the minimum of the cost function or at the boundaries of the feasibility region given by the requirements on μ_c. The minimum of the cost function can be achieved by taking its derivative with respect to μ_c. To obtain this derivative in an explicit form, we assume that CSMA/CA energy consumption can be approximated by a constant value since the numerical value is smaller than other factors. Under this assumption, the minimization by the derivative is approximated by

$$\mu_e(h) = (p_{\min}\lambda\ A_{\text{tx}})^{\frac{1}{2}} \left[\frac{h-2}{h} Q_b\left(\frac{2S}{h-1}\right) u(h-2) + \frac{2}{h} Q_b\left(\frac{S}{h-1}\right) + A_{\text{w}} T_{\text{w}} + A_{\text{rx}}(T_{\text{ac}} - T_{\text{w}}) \right]^{-\frac{1}{2}}. \tag{9.13}$$

The approximation was validated in [41], where it was shown that the error is less than 2%.

By using (9.13), we see that an optimal solution $\mu_c^*(h)$ provided h is given by $\mu_e(h)$ if $\mu_{\min} \leq \mu_e(h) \leq \mu_{\max}$ and $\mu_e(h) \geq \max(f_r(h,\Omega),\, f_d(h,\tau,\Delta))$, otherwise an optimal solution is given by the value between $\mu_{\max}$ and $\max(f_r(h,\Omega),\, f_d(h,\tau,\Delta))$ that minimizes $E_{\text{tot}}(h,\mu_c)$. Therefore, for any $h = 2, \ldots, \min(h_r, h_d)$, we compute $\mu_c^*(h)$. Then, the optimal solution h^* and μ_c^* is given by the pair $\mu_c^*(h), h$ that minimizes the cost function.

9.4.5 Adaptation Mechanisms

In the previous sections, we showed how to determine the optimal number of clusters and cumulative wake-up rate by solving an optimization problem. Here, we present in detail some adaptation algorithms that the sink must run to determine correctly h^* and μ_c^* as the traffic rate and channel condition changes. These algorithms allow us to adapt the protocol parameters to the traffic rate and channel condition without high message overhead.

9.4.5.1 Traffic Rate and Channel Estimation

The sink node estimates the traffic rate λ and the worst channel probability $p_{\min}$ of the network. To estimate the global minimum of the worst channel condition, each p_i should be estimated at a local node and sent to the sink for each link of the path $i = 1, \ldots, h$. This might increase considerably the packet size. To avoid this, we propose the following strategy. Consider a relay node of the i th cluster. It estimates p_i by the signal of the beacon packet. Then the nodes compares p_i with the channel condition information carried by the received data packet and selects the minimum. This minimum is then encoded in the data packet and sent with it to the next-hop node. After the sink node retrieves the channel condition of the route by receiving a data packet, it computes an average of the worst channel conditions among the last received data packets. Using this estimate, the sink solves the optimization problem as described in Sect. 9.4.4. Afterward, the return value of the algorithm, h^* and μ_c^*, can be piggybacked on beacons that the sink sends toward the relays closer to the sink. Then, these protocol parameters are forwarded when the nodes wake-up and send beacons to the next cluster toward the edge nodes. During the initial state, nodes set $h = 2$ before receiving a beacon.

9.4.5.2 Wake-Up Rate and Radio Power Adaptation

Once a cluster received μ_c^*, each node in the cluster must adapt its wake-up rate so that the cluster generates such a cumulative wake-up rate. We consider the natural solution of distributing μ_c^* equally between all nodes of the cluster. Let μ_k be the wake-up rate of node k, and suppose that there are l nodes in a cluster. The fair solution is $\mu_k = \mu_c^*/l$ for any node. However, a node does not know and cannot estimate efficiently the number of nodes in its cluster.

To overcome this problem, we follow the same approach proposed in [31], where an Additive Increase and Multiplicative Decrease (AIMD) algorithm leads to a fair distribution of the wake-up duties within a single cluster. Specifically, each node that is waiting to forward a data packet observes the time before the first wake-up in the forwarding region. Starting from this observation, it estimates the cumulative wake-up rate $\tilde{\mu}_c$ of the forwarding region and it compares it with the optimal value of the wake-up rate μ_c^* when a node receives a beacon. Note that the node retrieves

information on h^*, μ_c^* and location information of the beacon node. If $\tilde{\mu}_c < \mu_c^*$ the node sends by the data packet an Additive Increase (AI) command for the wake-up rate of next-hop cluster, else it sends a Multiplicative Decrease (MD) command. Furthermore, the node updates the probability of successful transmission p_i based on the channel information using the RSSI and distance information d_k between its own location and beacon node. After the node updates the channel condition estimation, it sets the data packet transmission power to $P_t(d_k)$, and encodes the channel estimation in the packet as described in Sect. 9.4.5.2.

If a data packet is received, the node retrieves information on wake-up rate update: if AI then $\mu_k = \mu_k + \theta$, else $\mu_k = \mu_k/\phi$, where θ and ϕ are control parameters. From experimental results, we obtained that $\theta = 3$ and $\phi = 1.05$ achieve good performance. Furthermore, the node runs the reset mechanism for load balancing of wake-up rate as discussed in Sect. 9.4.5.2. The command on the wake-up rate variation is piggybacked on data packets and does not require any additional message.

However, this approach may generate a load balancing problem because of different wake-up rates among relays within a short period. Load balancing is a critical issue, since some nodes may wake-up at higher rate than desired rate of other nodes, thus wasting energy. To overcome this situation, each relay node runs a simple reset mechanism. We assign an upper and lower bound to the wake-up rate for each node. If the wake-up rate of a node is larger than the upper bound $(1 + \xi)\mu_c^*(h^* - 1)/N$ or is smaller than the lower bound $(1 - \xi)\mu_c^*(h^* - 1)/N$, then a node resets its wake-up rate to $\mu_c^*(h^* - 1)/N$, where ξ assumes a small value and $(h^* - 1)/N$ is an estimation of number of nodes per cluster.

9.4.6 Experimental Implementation

In this section, we provide an extensive set of experiments to validate the Breath protocol. The experiments enable us to assess Breath in terms of reliability and delay in the packet transmission, and energy consumption of the network both in stationary and in transitionary condition. The protocol was implemented on a test bed of Tmote sensors [15], and was compared with a standard implementation of IEEE 802.15.4 [4], as we discuss next.

We consider a typical indoor environment with concrete walls. The experiments were performed in a static propagation (AWGN) and time-varying fading environment (Rayleigh), respectively:

- AWGN environment: nodes and surrounding objects were static, with minimal time-varying changes in the wireless channel. In this case, the wireless channel is well described by an Additive White Gaussian Noise (AWGN) model.
- Rayleigh fading environment: obstacles were moved within the network, along a line of 20 m. Furthermore, a metal object was put in front of the edge node, so the edge node and the relays were not in line-of-sight. The edge node was moved on a distance of few tens of centimeters.

A node acted as edge node and generated packets periodically at different rates ($\lambda = 5$, 10, and 15 pkt/s). Fifteen relays were placed to mimic the topology in Fig. 9.2. The edge node was at a distance of 20 m far from the sink. The sink node collected packets and then computed the optimal solution as described in Sects. 9.4.4 and 9.4.5. The delay requirement was set to $\tau = 1$ s and the reliability to $\Omega = 0.9$ and 0.95. In other words, we imposed that packet must reach destination within 1s with a probability of Ω. These requirements were chosen as representative for control applications.

We compared Breath against an implementation of the unslotted IEEE 802.15.4 [4] standard, which is similar to the randomized MAC that we use in this chapter. In such an IEEE 802.15.4 implementation, we set nodes to a fixed sleep schedule, defined by $C\,T_{\text{ac}}$ were C is integer number (recall that T_{ac} is the maximum node listening time in Breath). We defined the case $\mathcal{L}$ (low sleep), where the IEEE 802.15.4 implementation is set with $C = 1$, whereas we defined the case $\mathcal{H}$ (high sleep) by setting $C = 4$. The case $\mathcal{H}$ represents a fair comparison between Breath and IEEE 802.15.4, while in the case $\mathcal{L}$ nodes are let to listen much longer time than nodes in Breath. The power level in the IEEE 802.15.4 implementation where set to -5 dBm. We set the IEEE 802.15.4 protocol parameters to default values *macMinBE* $= 3$, *aMaxBE* $= 5$, *macMaxCSMABackoffs* $= 4$. Details follows in the sequel.

9.4.7 Protocol Performance

In this subsection, we investigate the performance of Breath about the reliability, average delay, and energy consumption that can be achieved in a stationary configuration of the requirements, i.e., during the experiment there was no change of application requirements. Data was collected out of 10 experiments, each lasting 1 h. For performance of the protocol when the requirements are time varying, see [41].

9.4.7.1 Reliability

Figure 9.4 indicates that the network converges by Breath to a stable error rate lower than $1 - \Omega$ and hence satisfies the required reliability with traffic rate $\lambda = 10$ pkt/s, the delay requirement $\tau = 1$s, and $\Omega = 0.9$, 0.95. IEEE 802.15.4 $\mathcal{H}$ in AWGN channel provides the worse performance than the other protocols because of lower wake-up rate. Observe that $\Omega = 0.9$ in Rayleigh fading environment gives the better reliability than $\Omega = 0.95$ in AWGN channel due to higher wake-up rate to compensate the fading channel condition. Notice that the higher fluctuation of reliability between the number of received packets $2,500$ and $2,800$ for Rayleigh fading environment with $\Omega = 0.95$ is due to deep attenuations in the wireless channel.

Figure 9.5 shows the reliability of Breath and IEEE 802.15.4 $\mathcal{L}$, $\mathcal{H}$ as a function of the reliability requirement $\Omega = 0.9, 0.95$ and traffic rate $\lambda = 5, 10, 15$ pkt/s

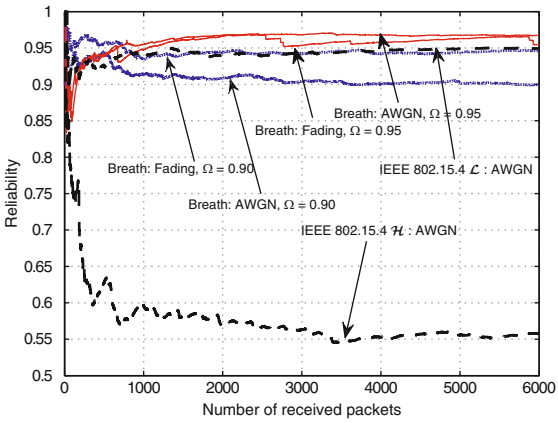


Fig. 9.4 Convergence over time of the reliability for IEEE 802.15.4 $\mathcal{L}$, $\mathcal{H}$ in AWGN, and Breath with reliability requirements $\Omega = 0.9, 0.95$ and traffic rates $\lambda = 10$ pkt/s in AWGN and Rayleigh fading environments

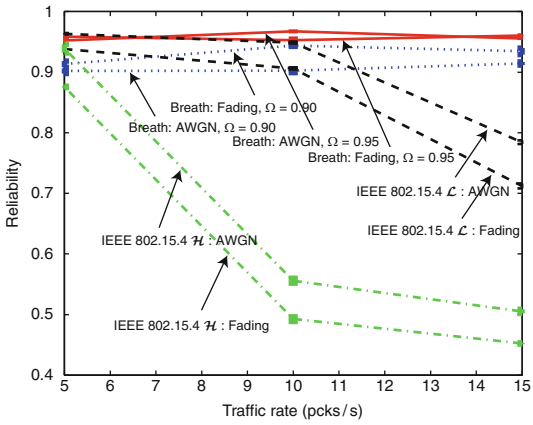


Fig. 9.5 Reliability in IEEE 802.15.4 $\mathcal{L}$, $\mathcal{H}$, and Breath with requirement $\Omega = 0.9, 0.95$ for traffic rates $\lambda = 5, 10, 15$ pkt/s in AWGN and Rayleigh fading environments

in AWGN and Rayleigh fading environments, with the vertical bars indicating the standard deviation as obtained out of 10 experimental runs of 1 h each. Observe that the reliability is stable around the required value for Breath, and this holds for different traffic rates and environments. However, IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$ do not ensure the reliability satisfaction for large traffic rates. Specifically, IEEE 802.15.4 $\mathcal{H}$ shows poor reliability in any case, and performance worsen as the environment moves from the AWGN to the Rayleigh fading. Furthermore, even though IEEE 802.15.4 $\mathcal{L}$ imposes that nodes wakes up more often, it does not guarantee a good reliability in higher traffic rates. The reason is found in the sleep schedule of the IEEE 802.15.4 case, which is independent of traffic rate and wireless channel conditions. The result is that the fixed sleep schedule is not feasible to support high traffic and time-varying wireless channels. Moreover, the fixed sleep schedule does not guarantee a uniform distribution of cumulative wake-up rate within certain time in a cluster, which means that there may be congestions. On the contrary, Breath presents an excellent behavior in any situation of traffic load and channel condition.

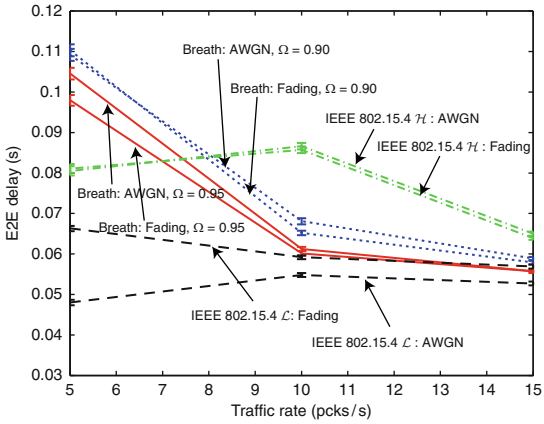


Fig. 9.6 Temporal average of the delay of Breath and IEEE 802.15.4 $\mathcal{L}, \mathcal{H}$ with reliability requirement $\Omega = 0.9, 0.95$ and delay requirement $\tau = 1$ s over traffic rates $\lambda = 5, 10, 15$ pkt/s in AWGN and Rayleigh fading environment

9.4.7.2 Delay

In Fig. 9.6, the sample average of the delay for packet delivery of Breath, IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$ are plotted as a function of the reliability requirement Ω and traffic rate λ in AWGN and Rayleigh fading environments, with the vertical bars indicating the standard deviation of the samples around the average. The sample variance of the delay exhibits similar behavior as the average. The delay meets quite well the constrains. Observe that delay decreases as the traffic rate rises. This is due to that Breath increases linearly the wake-up rate of nodes when the traffic rate increases (see (9.4)). The delay is larger for worse reliability requirements. Note that (9.4) increases as the reliability requirement Ω increases. IEEE 802.15.4 $\mathcal{L}$ has lower delay than IEEE 802.15.4 $\mathcal{H}$ because nodes have higher wake-up time. Breath has an intermediate behavior with respect to IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$ after $\lambda = 7$. From these experimental results, we conclude that both Breath and IEEE 802.15.4 meet the delay requirement. However, notice that the delay for IEEE 802.15.4 is related to only packet successfully received, which may be quite few.

9.4.7.3 Duty Cycle

In this section, we study the energy consumption of the nodes.

As energy performance indicator, we measured the node's duty cycle, which is the ratio of the active time of the node to the total experimental time. Obviously, the lower is the duty cycle, the better is the performance of the protocol on energy consumption.

Figure 9.7 shows the sample average of duty cycle of Breath, IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$ with respect to the traffic rates $\lambda = 5, 10, 15$ pkt/s and $\Omega = 0.9, 0.95$, both in AWGN and in Rayleigh fading environments, with the vertical bars indicating the standard deviation of the samples. Note that IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$ do not exhibit a clear relationship with respect to traffic rate and have almost flat duty cycle around

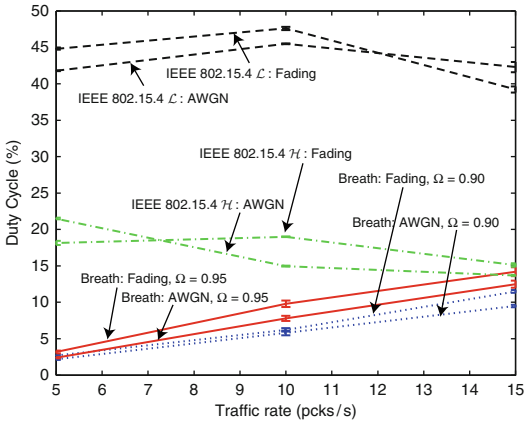


Fig. 9.7 Sample average of the node's duty cycle in IEEE 802.15.4 $\mathcal{L}$, $\mathcal{H}$ and Breath with reliability requirement $\Omega = 0.9, 0.95$ for traffic rates $\lambda = 5, 10, 15$ pkt/s in AWGN and Rayleigh fading environments

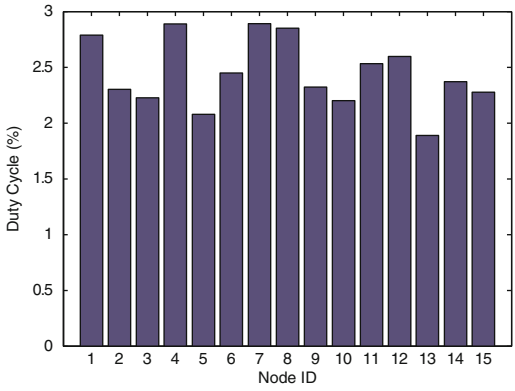


Fig. 9.8 Distribution of the duty cycle in each node with $N = 15$ relays. The reliability requirement is $\Omega = 0.95$ and traffic rate is $\lambda = 5$ pkt/s

42 and 18%, respectively, because of fixed sleep time. Considering Breath, observe that the duty cycle increases linearly with the traffic rate and reliability requirement. As for the delay, this is explained by (9.4). Since Breath minimizes the total energy consumption on the base of a tradeoff between wake-up rate and waiting time of beacon messages (recall the analysis in Sect. 9.4.3.3), lower wake-up rates do not guarantee lower duty cycle. Observe that choosing a lower active time for the nodes of the IEEE 802.15.4 implementation would obviously obtain energy savings comparable with Breath; however, the reliability of the IEEE 802.15.4 implementation would be heavily affected (recall Fig. 9.5). In other words, ensuring a duty cycle for the IEEE 802.15.4 implementation comparable with Breath would be very detrimental with respect to the reliability.

Figure 9.8 shows the experimental results for the duty cycle of each relay node for $\lambda = 5$ pkt/s and $\Omega = 0.95$. A fair uniform distribution of the duty cycles among all nodes of the network is achieved. This is an important result, because the small variance of the wake-up rate among nodes signifies that duty cycle and load are uniformly distributed, with obvious advantages for the network lifetime.

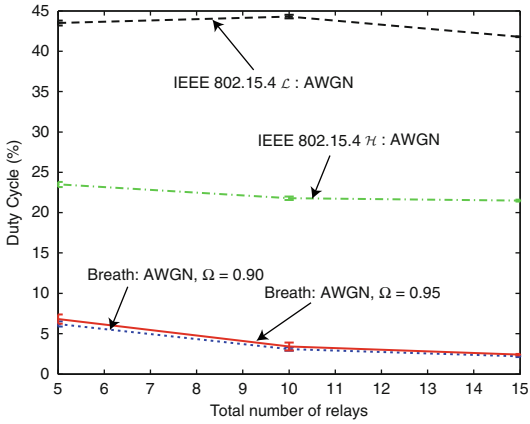


Fig. 9.9 Sample average of the duty cycle in IEEE 802.15.4 $\mathcal{L}$, $\mathcal{H}$ and Breath with reliability requirement $\Omega = 0.9, 0.95$ and traffic rates $\lambda = 5, 10, 15$ pkt/s in AWGN environment for different networks, each with a different number of relaying nodes

Figure 9.9 reports the case of several networks, where each network corresponds to a number of relays between the edge and the sink in an AWGN environment. From the figure it is possible to evaluate how much Breath extends the network lifetime compared to IEEE 802.15.4 $\mathcal{L}$ and $\mathcal{H}$. Observe that the duty cycle is proportional to the density of nodes. Hence, the network lifetime is extended fairly by adding more nodes without creating load balancing problems.

Finally, Breath uses a radio power control, so that further energy savings are actually obtained with respect to the IEEE 802.15.4 implementation.

9.5 TREnD

In this section, we present the TREnD protocol. The acronym remarks the significant characteristics simultaneously embraced by the protocol as opposed to other solutions available from the literature: timeliness, reliability, energy efficiency, and dynamic adaptation. TREnD is an energy-efficient protocol for control applications over WSNs organized into clusters.

9.5.1 System Model

TREnD considers a general scenario for an industrial control application: the state of a plant must be monitored at locations where electrical cabling is not available or cannot be extended, so that wireless sensor nodes are an appealing technology.

Information taken by nodes, which are uniformly distributed in clusters, is sent to the sink node by multi-hop communication. The clustered topology is motivated by the energy efficiency, since transmitting data directly to the sink may consume more than routing through relays. The cluster formation problem has been thoroughly investigated in the literature and is out of the targets of this chapter (see, e.g., [35, 37]).

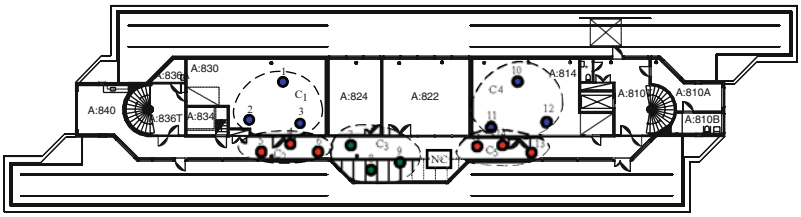


Fig. 9.10 Experimental setup

In Fig. 9.10, the system model is reported. Nodes are deployed in an indoor environment with rooms. Each dotted curve defines a cluster of nodes. Nodes of a cluster can send packets only to the nodes of next cluster toward the controller, which takes appropriate actions upon the timely and reliable reception of source information. Hence, nodes not only send their sensed data, but also forward packets coming from clusters further away from the sink. The network controller is the sink node, which, being a node of the networks, is equipped with light computing resources.

We assume that the controller knows cluster locations and the average number of nodes in each cluster, and nodes know to which cluster they belong. The controller can estimate the amount of data generated by each cluster, which is used to adapt the protocol to the traffic regime. These assumptions are reasonable in industrial environments [2].

9.5.2 TREnD Protocol Stack

In this section, we introduce the protocol stack of TREnD.

Similarly to SERAN [9], the routing algorithm of TREnD is hierarchically subdivided into two parts: a static route at interclusters level and a dynamical routing algorithm at node level. This is supported at the MAC layer by an hybrid TDMA and carrier sensing multiple access (TDMA/CSMA) solution.

The static schedule establishes which one is the next cluster to which nodes of a given cluster must send packets by calculating the shortest path from every cluster to the controller. The network controller runs a simple combinatorial optimization problem of latency-constrained minimum spanning tree generation [42]. Alternatively, if the number of clusters is large, the static routing schedule is precomputed offline for a set of cluster topologies and stored in the sink node in a look-up array. No intra cluster packet transmission is allowed.

The static routing algorithm is supported at MAC level by a weighted TDMA scheme that regulates channel access among clusters. Nodes are awake to transmit and receive only during the TDMA-slot associated with the cluster for transmission and reception, respectively, thus achieving consistent energy savings. The organization of the TDMA-cycle must consider the different traffic regimes

depending on the cluster location. Since clusters closer to the sink may experience higher traffic intensity, more than one transmitting TDMA-slot can be assigned to them. It is natural to first forward packets of clusters close to the controller, since this minimizes the storage requirement in the network. To minimize the global forwarding time, the evacuation of packets of a cluster is scheduled path-by-path. By following these rules, the controller is able to generate an appropriate TDMA scheduling table.

The dynamic routing is implemented by forwarding the packets to a node within the next-hop cluster in the path chosen at random, as proposed in [27] and [35]. In such an operation, no cluster-head node is needed within clusters, and nodes need to be aware only of the next-hop cluster connectivity. The procedure for random selection of next-hop node is performed by considering a duty cycling in the receiving cluster combined with beacon transmissions.

The communication stage between nodes during a TDMA-slot is managed at MAC layer by a p-persistent CSMA/CA scheme to offer flexibility for the introduction of new nodes, robustness to node failures, and support for the random selection of next-hop node. As we will see in Sect. 9.5.5, in hybrid TDMA/CSMA solutions the p-persistent MAC gives better performance than the binary exponential backoff mechanism used by IEEE 802.15.4.

MAC operations of nodes are described in the following. Each node in the transmitter cluster having a packet to be sent wakes up in CSMA-slots with probability α and enters in listening state. At the receiver cluster, each node wakes up with probability β and multicasts a small length of beacon message to the nodes in the transmitter cluster. An awake node that correctly receives the beacon at the transmitter side, senses the channel and, if clean, tries to unicast its packet to the beacon sender. An acknowledgement (ACK) may conclude the communication if a retransmission mechanism is implemented. If no beacon is sent or there is a collision, the awake nodes in the transmitter cluster keep on listening in the next CSMA-slot with probability α or go to sleep with probability $1 - \alpha$.

If we compare TREnD to SERAN [9], we see that SERAN has the drawback that nodes in the receiver cluster have to be listening for the overall TDMA-slot duration, due to a contention-based transmission of the ACKs. In TREnD, the selection of the forwarding nodes follows a random policy regulated by β. The main advantage of this novel solution is the absence of delays between packets exchange during a CSMA-slot. This allows TREnD to work with a much higher traffic regime when compared to SERAN.

TREnD offers the option of data aggregation to fairly distribute the traffic load and energy consumption among clusters. The aggregation has the advantage of reducing the number of TDMA-slots per cluster and of the traffic for clusters closer to the sink. However, packet aggregation gives significant advantages only when the traffic is sufficiently high, as we will see in Sect. 9.5.5, because nodes have to idle-listen longer to catch more than one packet per time and perform the aggregation, and idle-listening is energy inefficient.

9.5.3 Protocol Optimization

In this section, optimization problem (9.1) to select the TREnD protocol parameters assumes the following form:

$$\min_{S,\alpha,\beta} \quad E_{\text{tot}}(S,\alpha,\beta) \tag{9.14a}$$

$$\text{s.t} \quad R(S,\alpha,\beta) \geq \Omega \tag{9.14b}$$

$$\Pr[D(S,\alpha,\beta) \leq D_{\max}] \geq \Pi. \tag{9.14c}$$

In this problem, $E_{\text{tot}}(S,\alpha,\beta)$ is the total energy consumption of the network and $R(S,\alpha,\beta)$ is the reliability constraint and Ω is the minimum desired reliability imposed by the control application. We denote by $D(S,\alpha,\beta)$ the random variable describing the distribution of the latency, by $D_{\max}$ the maximum latency desired by the control application, and by Π the minimum probability with which such a maximum latency should be achieved. The parameters Ω, $D_{\max}$, and Π are the requirements of the control application. The decision variables of the optimization problem are the TREnD parameters, namely the TDMA-slot duration S, the access probability α and the wake-up probability in reception β. In the following, we develop the expressions needed in the optimization problem, and derive the solution.

9.5.3.1 Energy Consumption

The total energy consumed by the network over a period of time is given by the combination of two components: listening and transmitting cost.[1]

Listening for a time t gives an energy consumption that is the sum of a fixed wake-up cost E_{w} and a time-dependent cost $E_l t$. The energy consumption in transmission is given by four components: beacon sending E_{bc}, CCA E_{cca}, packet sending E_{pkt} and ACK sending E_{ack}.

Consider a general topology with N nodes per cluster and suppose that there are G paths in the static routing scheduling (recall Sect. 9.5.2). Let h_i be the number of clusters (hops) per path, we define $h_{\max} = \max_{i=1,\ldots,G} h_i$. We define also W, as the number of listening TDMA-slots in a TDMA-cycle.

Recalling that the TDMA-cycle is $T_{\text{cyc}} = S M_s$, where M_s is the number of TDMA-slots in a TDMA-cycle, we have the following result:

Claim. Given traffic rate λ, the total energy consumed in a period T_{tot} is

$$E_{\text{tot}}(S,\alpha,\beta) = \frac{T_{\text{tot}}}{\gamma\, S} \sum_{j=1}^{\lambda T_{\text{cyc}}} j\alpha\beta\, E_{\text{cca}} + T_{\text{tot}} M_s \lambda \left(E_{\text{pkt}} + E_{\text{ack}}\right) + \beta \frac{N W\, T_{\text{tot}}}{M_s} \left(\frac{E_{\text{bc}}}{\delta} + \frac{E_{\text{w}}}{S} + E_l\right). \tag{9.15}$$

[1] Note that the costs for the initialization of the network are negligible in the energy balance.

Proof. A proof is provided in [43]. □

9.5.3.2 Reliability Constraint

Considering the p-persistent slotted CSMA MAC and the duty cycling in reception, we have the following result:

Claim. The probability of successful transmission in a CSMA-slot while there are k packets waiting to be forwarded in the cluster is

$$p_k = \gamma\, p_{\text{bc}}\,(1-(1-\alpha)^k)\,(1-p_{\text{cl}})^{\alpha(k-1)}, \tag{9.16}$$

where $p_{\text{bc}} = \gamma N\beta\,(1-\beta)^{N-1}$ is the successful beacon probability and p_{cl} is the probability of an erroneous sensing of a node, when it competes with another node.

Proof. A proof is provided in [43]. □

In TREnD, a radio power control is implemented, so that the attenuation of the wireless channel is compensated by the radio power, which ensures a desired packet loss probability, as proposed in [44] and [45]. As a consequence of the power control, the channel can be abstracted by a random variable with good channel probability γ. Such a modeling has been adopted also in other related works (e.g., [9, 22]). Considering the collision probability p_{cl}, we observe that for optimization purposes an upper bound suffices. Experimental results show that a good upper bound is $p_{\text{cl}} = 0.2$.

By using Claim 9.5.3.2, we can derive the following result:

Claim. Let $V(n) = \{1-p_n, 1-p_{n+1}, ..., 1-p_k\}$, where p_n is the generic term given in (9.16) and $A(n) = [a_{i,j}]_{M_c}^{S-k+n}$ be a matrix containing all the M_c combinations with repetition of the elements in $V(n)$, taken in groups of $S-k+n$ elements. Let $h_{\max}$ be the maximum number of hops in the network. Then, the reliability of TREnD is

$$R(S,\alpha,\beta) = \left[\sum_{n=0}^{k}\frac{k-n}{k}\prod_{l=n+1}^{k}p_l\left(\sum_{i=1}^{M_c}\prod_{j=1}^{S-k+n}a_{i,j}\right)\right]^{h_{\max}}. \tag{9.17}$$

Proof. A proof is provided in [43]. □

With packet aggregation enabled, the following result holds:

Claim. Let h_i be the number of hops in the path i. Let R_z be the reliability in a single hop when z packets are aggregated. The reliability of a packet that experiences j hops to the controller is

$$R_j^{\text{ag}}(S,\alpha,\beta) = R_{j-1}^{\text{ag}}\, r_{B_i-j+1}, \tag{9.18}$$

where $r_j = \sum_{i=1}^{j}(1 - r_{i-1}) \prod_{z=1}^{j-i+1} R_z$, with $r_0 = 0$.

Proof. A proof is provided in [43]. □

If the data aggregation is disabled or the size of aggregated packets does not change significantly, then we can simplify (9.18) and obtain the relation in (9.17). The previous claims are illustrated and verified by experiments in [43].

9.5.3.3 Latency Constraint

The furthest cluster from the controller is the one experiencing the highest latency. Therefore, the latency of packets coming from such a cluster must be less than or equal to a given value $D_{\max}$ with a probability Π.

Recalling that the maximum number of hops in the network is $h_{\max}$, an upper bound on the TDMA-slot duration S is $S_{\max} = D_{\max}/h_{\max}$. Then, we can provide the following result:

Claim. The latency constraint in (9.14c) is well approximated by

$$\Pr[D(S, \alpha, \beta) \leq D_{\max}] \approx 1 - \frac{1}{2}\operatorname{erfc}\left(\frac{A - \mu}{\sigma}\right), \tag{9.19}$$

$$\text{where} \quad \mathrm{A} = \begin{cases} S & \text{if} \quad S \leq \frac{D_{\max}}{h_{\max}} \\ D_{\max} - (h_{\max} - 1)S & \text{if} \quad S > \frac{D_{\max}}{h_{\max}} \end{cases}$$

$\mu = \sum_{j=1}^{k} 1/p_j$, and $\sigma^2 = \sum_{j=1}^{k}(1 - p_j)/p_j^2$.

Proof. A proof is provided in [43]. □

9.5.3.4 Protocol Optimization

In the previous subsection, we have established the expressions of the energy consumption in (9.15), the reliability in (9.18), and the latency constraint in (9.19). We observe that all these expressions are highly nonlinear in the decision variables. Sensor nodes are not equipped with a high processing capacity to use these equations, therefore, we provide a computationally affordable suboptimal solution to the optimization problem. In the following, we show that such a strategy still gives satisfactorily results.

First, we provide an empirical result on the access probability α and wake-up probability β, for a given TDMA-slot duration S.

Claim. Let N be the number of nodes is a cluster. Let λ be the traffic rate, the access probability α^*, and the wake-up probability β^*, which optimize the reliability in (9.17), are

$$\alpha^* = \frac{c_1}{\lambda\, S\, M_s + c_2} \qquad \beta^* = \frac{1}{N}. \tag{9.20}$$

with coefficients $c_1 = 2.17$, $c_2 = 1.81$.

Proof. A proof is provided in [43]. □

We note here that such choices are suboptimal because are limited to strategies with constant access and wake-up probabilities per each node.

By using (9.20) for the access probability and wake-up probability, and by assuming S as a real-valued variable, we can show that E_{tot}, given in (9.15), is a convex and monotonically decreasing function of S. It follows that a simple solution for the TDMA-slot duration, S^*, is given by the maximum integer value of S that satisfies one out of the two constraints in the problem (9.14a), which are given explicitly by (9.18) and (9.19), respectively. The search of the optimal S can be done by a simple additive increasing multiplicative decreasing algorithm, which we initialize by observing that $S^* \leq S_{\max}$. Indeed, as shown in Sect. 9.5.3.3, the maximum latency requirement $D_{\max}$ provides an upper bound for S, given by $S_{\max} = D_{\max}/h_{\max}$.

9.5.4 Protocol Operation

Suppose that the network user deploys a WSN of nodes implementing the TREnD protocol, and sets the desired control application requirements Ω, $D_{\max}$, and Π. During an initial phase of operation the sink node retrieves the traffic and the cluster topology by the received packets. After computing or reading from a look-up array the static routing schedule and TDMA-cycle, the sink computes the optimal parameters as described in Sect. 9.5.3.4. Then, the sink communicates these values to the nodes of the network by tokens. Such a token passing procedure ensures synchronization among nodes and allows for initializing and self-configuring of the nodes to the optimal working point of the protocol. The token is then forwarded by the nodes closer to the sink to other nodes of the clusters far away by using the ACK mechanism described in [9]. Such tokens need also to be updated so that our protocol adapts dynamically to new nodes added in the clusters, variations in the source traffic, control application requirements, and time drift of the clocks. We experienced that a 20 TDMA-cycles period for the refreshing procedure gives satisfactory performance to maintain an optimal network operation with negligible extra energy consumption.

9.5.5 Experimental Implementation and Validation

In this section, we present a complete implementation of TREnD by using TinyOS 2.x [46] and Tmote Sky nodes[15]. To benchmark our protocol, we implemented

also SERAN [9] and IEEE 802.15.4 [4]. Recall that IEEE 802.15.4 is the base for WirelessHart and other protocols for industrial automation, and that there is no other protocol available from the literature that is energy efficient, and ensures reliable and timely packet transmission, as we summarized in Table 9.1. We used the default MAC parameters of IEEE 802.15.4 so that the protocol fits in the higher level TDMA structure and routing algorithm of SERAN and TREnD.

We reproduced the reference test-bed topology reported in Fig. 9.10, where clusters are placed in an indoor environment. Each cluster is composed by 3 sensors, deployed at random within a circle with one meter radius. We analyze different scenarios with different sets of traffic rate λ and control application requirements (Ω, $D_{\max}$, and Π), which we report in Table 9.2. For each scenario, Table 9.2 shows also the optimal TDMA-slot duration, access and wake-up probabilities as obtained by the optimization in Sect. 9.5.3.4. We measured the duty cycle of nodes as indicator of the energy efficiency.

9.5.5.1 Performance Comparison

In the first set of experiments, we show the performance improvements in TREnD, when compared to SERAN. In Fig. 9.11, the reliability is reported as function of the traffic rate λ, by fixing $\Omega = \Pi = 95\%$, and $D_{\max} = 3, 9$ s. TREnD has high reliability for all traffic rate conditions and SERAN is significantly outperformed. In particular with $D_{\max} = 3$ s, as traffic rate increases over $\lambda = 0.3$ pkt/s, the reliability of SERAN significantly decreases.

Table 9.2 Application requirements and experimental results

Scenario	λ	$D_{\max}$	Π	Ω	S^*	α^*	β^*
$\mathcal{LR}$	0.1 pkt/s	9 s	95%	95%	3.3 s	0.41	0.33
$\mathcal{HR}$	0.3 pkt/s	3 s	95%	95%	1.2 s	0.43	0.33

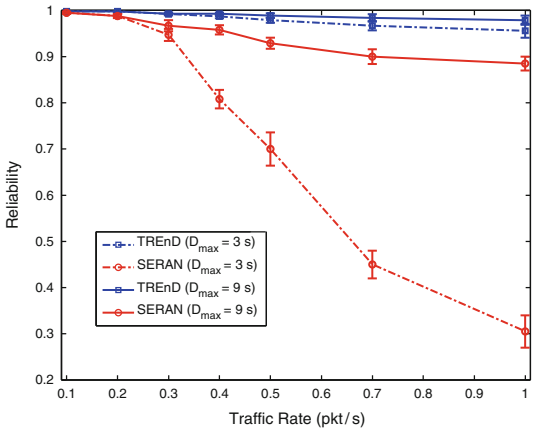


Fig. 9.11 TREnD and SERAN: reliability vs. traffic rate λ, for $\Omega = \Pi = 95\%$

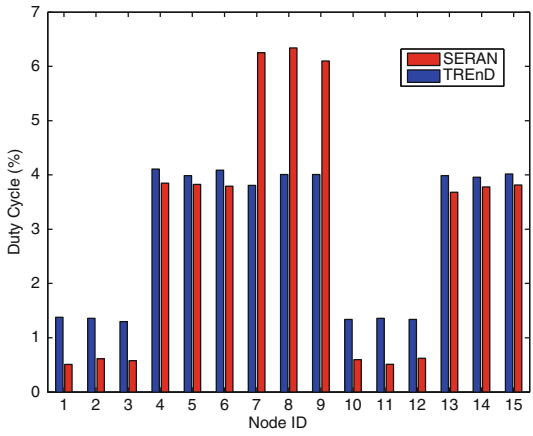


Fig. 9.12 TREnD and SERAN: duty cycle distribution among nodes for $\lambda = 0.3$ pkt/s, $\Omega = \Pi = 95\%$

In Fig. 9.12, we compare the energy consumption of the two protocols, showing the average duty cycle of each node, for fixed $\Omega = \Pi = 95\%$, $D_{\max} = 3$ s and $\lambda = 0.3$ pkt/s. As discussed above, in this operative condition both SERAN and TREnD meet the reliability and latency constraints. By implementing TREnD with data aggregation, we observe a more balanced duty cycle among clusters, particularly for the last hop clusters. However, the price to pay for having a better load balancing is a slight increasing of the average duty cycle. In fact, TREnD presents a slightly higher duty cycle for most of the nodes, but it reduces to about 30% of the energy consumption for nodes 7, 8, and 9 (cluster C_3), which are critical for the network operation since they also forward information from clusters C_1 and C_2. This suggests that packet aggregation is a viable choice only for the clusters supporting high traffic, as those next to the sink. Hence, it is recommended to implement packet aggregation only for those clusters, while for the others no aggregation is needed. In conclusion, TREnD ensures higher reliability, load balancing and a longer network lifetime than SERAN, without any significant difference in the complexity of the scheme.

Given these results, in the following performance evaluation of TREnD we disregard SERAN and consider IEEE 802.15.4. We present two sets of experimental results, evaluated for scenarios $\mathcal{LR}$ and $\mathcal{HR}$ as specified in Table 9.2. Figure 9.13 reports the average values of reliability, latency, and duty cycle as achieved by the experiments for TREnD and IEEE 802.15.4. Data of clusters belonging to the same paths are joined by lines. We see that TREnD always ensures the satisfaction of the reliability and latency constraints specified in Table 9.2. TREnD guarantees much better reliability, in particular for cluster C_1 (3 hops). In fact in C_1, IEEE 802.15.4 does not fulfill the requirement. The average latency of IEEE 802.15.4 is slightly lower than TREnD, but observe that the latency of IEEE 802.15.4 is computed only for packets arriving successfully at the sink. We observe similar behavior also for other scenarios.

Finally, we present some results about the duty cycle. According to the traffic load supported by the clusters and their allotted TDMA time slots, we observe that

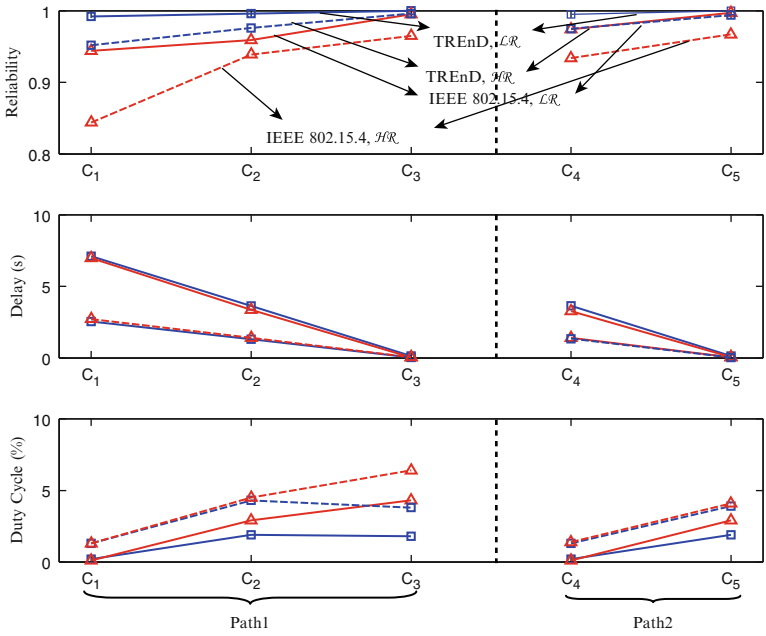


Fig. 9.13 TREnD and IEEE 802.15.4: reliability, latency, and duty cycle for scenarios $\mathcal{LR}$ and $\mathcal{HR}$

the duty cycle depends on the number of times a cluster wakes up for the forwarding procedure. The duty cycle is the same for the clusters far away from the sink (C_1 and C_4, see Fig. 9.10), but for all other clusters TREnD gives a consistent reduction of the duty cycle with respect to IEEE 802.15.4.

We remark here that the duty cycle strongly depends on the traffic load in the network. In Dozer [26], an average duty cycle 0.2% is achieved for a network of 40 nodes with a packet generation period of 120 s each (total traffic load $\simeq$0.3 pkt/s). TREnD gives an average duty cycle 2.5%, but the total traffic load is much higher ($\simeq$5 pkt/s) than Dozer.

9.5.5.2 Dynamic Adaptation

In the previous section, we used a static network topology where each node is placed at fixed position and the application requirements do not change with time. In this section, we show the dynamical behavior of the protocol. As we discussed before, no protocol in literature allows for a dynamical adaptation of the parameters to the application requirements.

We present the experimental results of dynamic changes between two scenarios ($\mathcal{LR}$ and $\mathcal{HR}$ in Table 9.2) in static and time-varying channel conditions. A Rayleigh fading channel is obtained by moving the nodes around their initial

position and also by placing metal obstacles in front of the source nodes so that the line-of-sight with the sink is lost. The network starts with scenario $\mathcal{LR}$ and static channel, then after 20 TDMA-cycles we introduce a Rayleigh fading channel which persists until the TDMA-cycle 60. At TDMA-cycle 40, the application requirements change to scenario $\mathcal{HR}$.

Figures 9.14 and 9.15 report the resulting snapshot of the experiment in terms of reliability and latency. The reliability is measured at the sink node as average on each TDMA-cycle, while the latency is measured for each successfully received packet. In Fig. 9.14, we observe that TREnD guarantees the reliability requirement for both static and Rayleigh fading conditions, continuously adapting to the severe fading. The protocol is also robust to the change of scenario at TDMA-cycle 40. In Fig. 9.15, a snapshot of the latency is reported for clusters at different hops to the controller. We observe that the peaks of delay are limited due to the TDMA structure, the average and dynamics of the delay are slightly increasing in the time-varying stage but the latency constraint is fulfilled. Moreover, the protocol adapts well to the change of scenario at TDMA-cycle 40.

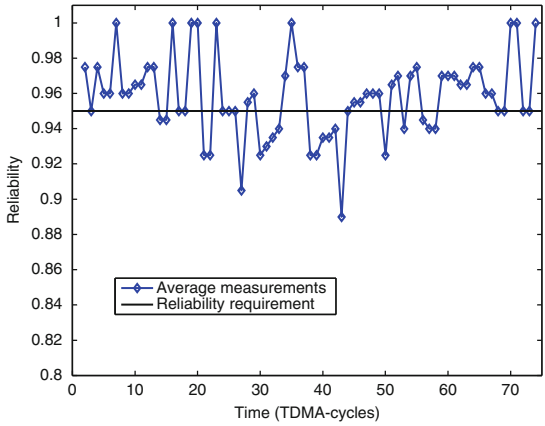


Fig. 9.14 TREnD: Reliability trace given by the experiments

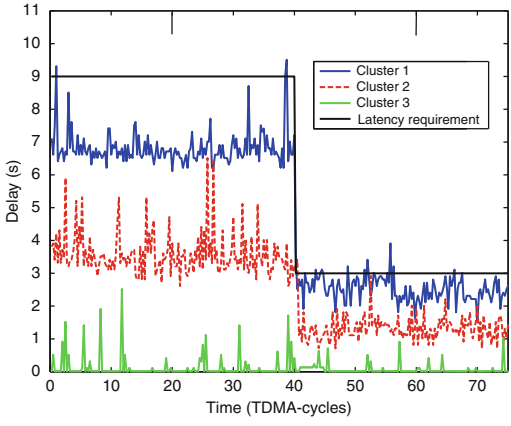


Fig. 9.15 TREnD: Latency trace given by the experiments

9.6 Conclusions

We proposed a novel approach to the design of protocols for control applications over WSNs. The approach guarantees the respect of control requirements on reliability and latency while minimizing energy consumption. Duty cycle, routing, medium access control, and physical layers were considered all together to maximize the network lifetime by taking into account the tradeoff between energy consumption and application requirements for control applications. The design approach was based on optimization problems to select the protocol parameters by simple algorithms that can run on resource constrained nodes.

The design methodology was illustrated by the proposal of two protocols: Breath and TREnD. The Breath protocol was designed for scenarios, where a plant must be controlled over a multi-hop network. The TREnD protocols were designed for environments where multiple plants have to be controlled by a multi-hop network. For these protocols, we developed the analytical expressions of the total energy consumption of the network, as well as reliability and delay for the packet delivery. These relations allowed us to pose constrained optimization problems to select optimally the number of hops in the multi-hop routing, the wake-up rates of the nodes, and the transmit radio power as a function of the routing, MAC, physical layer, traffic generated by the plants, and hardware platform.

We provided a complete test-bed implementation of the protocols that we designed on the base of the method proposed in this chapter. We built a WSN with TinyOS and Tmote sensors. An experimental campaign was conducted to test the validity of Breath and TREnD in an indoor environment with both AWGN and Rayleigh fading channels. Experimental results showed that the protocols achieve the reliability and delay requirements, while minimizing the energy consumption. They outperformed a standard IEEE 802.15.4 implementation in terms of both energy efficiency and reliability. In addition, the protocols showed good load balancing performance, and is scalable with the number of nodes. Given these good performance, Breath and TREnD are good candidates for many control and industrial applications, since these applications ask for both reliability and delay requirements in the packet delivery.

Acknowledgements This work was supported by the EU project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation, and the Swedish Governmental Agency for Innovation Systems. Some parts of this book chapter have been presented to IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (IEEE SECON 08), San Francisco, CA, USA, June 2008, and to IEEE International Conference on Communications 2010 (IEEE ICC 10), Cape Town, South Africa, May 2010.

References

1. A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, 2005.
2. A. Willig, "Recent and emerging topics in wireless industrial communication," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008.

3. V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.
4. *IEEE Std 802.15.4-2996: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2006. [Online]. Available: http://www.ieee802.org/15
5. *Routing Over Low power and Lossy networks (ROLL)*. [Online]. Available: http://www.ietf.org/dyn/wg/charter/roll-charter.html
6. T. Abdelzaher, T. He, and J. Stankovic, "Feedback control of data aggregation in sensor networks," in *IEEE CDC*, December 2004.
7. J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, 2007.
8. W. Zhang, M. S. Braniky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, 2001.
9. A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System Level Design for Clustered Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, pp. 202–214, August 2007.
10. J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, 2007.
11. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, 2007.
12. E. Witrant, P. Park, M. Johansson, C. Fischione, and K. H. Johansson, "Predictive control over wireless multi-hop networks," in *IEEE MSC*, 2007.
13. *Wirelesshart*, 2007. [Online]. Available: http://www.hartcomm.org/
14. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *ACM MobiCOM*, 2000.
15. *Tmote Sky Data Sheet*, Moteiv, San Francisco, CA, 2006. [Online]. Available: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf
16. P. Buonadonna, D. Gay, J. Hellerstein, W. Hong, and S. Madden, "TASK: Sensor network in a box," Intel Research Lab Report, Tech. Rep., 2007.
17. R. Steigman, and J. Endresen, "Introduction to WISA and WPS, WISA-wireless interface for sensors and actuators and WPS-wireless proximity switches," *White paper*, 2004. [Online]. Available: http://www.eit.uni-kl.de/litz/WISA.pdf
18. *The ZigBee Alliance.* [Online]. Available: http://www.zigbee.org
19. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *ACM MobiCom*, vol. pp. 70–84, 2001.
20. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *ACM MobiCom*, 2001.
21. M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *ACM SenSys*, 2006.
22. F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," in *IEEE SNPA*, 2003.
23. S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica, "Flush: a reliable bulk transport protocol for multihop wireless networks," in *ACM SenSys*, 2007.
24. O. B. Akan and F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, 2005.
25. G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *USENIX OSDI*, 2006.
26. N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," in *IEEE/ACM IPSN*, 2007.
27. M. Zorzi and R. R. Rao, "Energy and latency performance of geographic random forwarding for ad hoc and sensor networks," in *IEEE WCNC*, 2003.
28. E. Felemban, C. G. Lee, and E. Eylem, "MMSPEED: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

29. J. Misic, S. Shafi, and V. Misic, "Cross-layer activity management in an 802.15.4 sensor network," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 131–136, 2006.
30. L. van Hoesel, T. Nieberg, J. Wu, and P. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 78–86, 2004.
31. J. Van Greuen, D. Petrovic, A. Bonivento, J. Rabaey, K. Ramchandran, and A. Sangiovanni-Vincentelli, "Adaptive sleep discipline for energy conservation and robustness in dense sensor networks," in *IEEE ICC*, 2004.
32. R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, 2006.
33. P. Chen and S. Sastry, "Latency and connectivity analysis tools for wireless mesh networks," in *IEEE/ACM ROBOCOMM*, 2007.
34. *Dust Networks Applications*, Dust Networks, 2009. [Online]. Available: http://www.dustnetworks.com/applications.
35. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
36. Y. Ma and J. H. Aylor, "System lifetime optimization for heterogeneous sensor networks with a hub-spoke topology," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 286–294, 2004.
37. O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
38. W. Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, 2004.
39. *CC2420 Data Sheet*, Chipcon, Oslo, Norway, 2005. [Online]. Available: http://www.chipcon.com/files/CC2420-Data-Sheet-1-3.pdf.
40. *CC2431 Data Sheet*, Texas Instruments. [Online]. Available: http://focus.tij.co.jp/jp/lit/ds/symlink/cc2431.pdf.
41. P. Park, "Protocol design for control applications using wireless sensor networks," Royal Institute of Technology (KTH), Tech. Rep. TRITA-EE 2009:041, Oct. 2009, licentiate Thesis.
42. C. Oliveira and P. Pardalos, "A Survey of Combinatorial Optimization Problems in Multicast Routing," *Computers and Operations Research*, Aug. 2005.
43. P. Di Marco, P. Park, C. Fischione, and K. H. Johansson, "A Cross-Layer Protocol for Wireless Sensor Networks in Control and Automation," Royal Institute of Technology (KTH), Tech. Rep., August 2009.
44. P. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-adapting protocol for wireless sensor networks in control and automation," in *IEEE SECON*, 2008.
45. M. Zuniga, B. Krishnamachari, "Analyzing the transitional region in low power wireless links," *IEEE SECON*, 2004.
46. D. Gay, P. Levis, and D. Culler, "Software Design Patterns for TinyOS," *ACM LCTES*, 2005.

Chapter 10
Multi-Robot Redeployment Control for Enhancing Wireless Networking Quality

Feng-Li Lian, Yi-Chun Lin, and Ko-Hsin Tsai

Abstract Mobile multi-robot systems are designed to perform tasks cooperatively. Hence, an effective coordination algorithm of communication, sensing and mobility is an important research issue. For example, when one of these robots is performing a monitoring task and every robot transmited the sensing data to the base station at the same rate, the communication network might easily have a congestion problem at the robot who is routing a large amount of data. Furthermore, if one special event occurs, the usage of whole network capacity will be largely decreased. This chapter discusses a distributed moving algorithm for redeploying these mobile robots with sensing capability. The functional goal of this algorithm is to adaptively allocate the channel capacity based on the amount of the sensing rate at each robot, as well as to dynamically spread these robots for increasing the coverage area and related utility. A simulation platform using MATLAB is also presented, and related statistical results including mean, standard deviation, coverage and convergence are used to illustrate the performance of the proposed algorithm.

Keywords Mobile wireless sensor network · Artificial force · Repdeployment algorithm

10.1 Introduction

Research on wireless sensor network (WSN) has received a great attention in recent years. A WSN has demonstrated its useful applications in many aspects such as environment monitoring, pollution detection, military reconnaissance, and building security. Conventionally speaking, a WSN consists of a set of stationary devices used to monitor an operation area. However, these nodes are only designed to perform some simple tasks passively. To make a WSN be more active and multi-functional, the mobility of these sensor nodes is added and hence, a mobile wireless

F.-L. Lian (✉)
Department of Electrical Engineering, National Taiwan University, Taipei, 10617, Taiwan
e-mail: fengli@ntu.edu.tw

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_10,

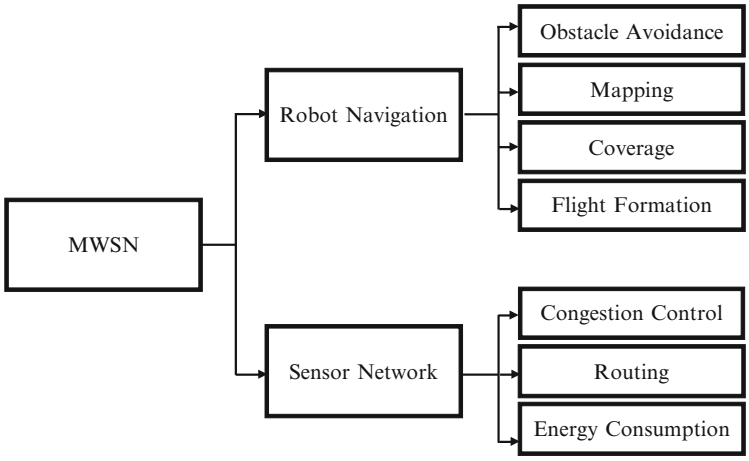


Fig. 10.1 MWSN needs to be given consideration to two issues

sensor network (MWSN) is then formed. Generally speaking, an MWSN is a WSN where each sensor is located on top of a vehicle, which can drive the sensor node to any desired position, for example, for redeployment. The mobility of a sensor node introduces the flexibility of reforming the network topology because each node can dynamically change its geographical location.

In an MWSN, in addition to the research issues studied in a WSN, related research problems for vehicle navigation should also be considered [1]. Related research issues in WSN include congestion control, message routing, and energy consumption; while those in vehicle/robot navigation includes obstacle avoidance, mapping/localization, coverage, and formation. Figure 10.1 shows a classification tree addressing these issues. Both issues are important for a wireless mobile network. For example, in daily life, when a user receives a phone call through a mobile phone, and if the communication quality is poor, the user should move to another new position to obtain a better communication quality (or higher transmission rate). In an MWSN, the communication quality may be affected by many factors including interference from other wireless communicating nodes, environment noise, distance between two nodes, assigned bandwidth, and these factors should be considered when designing the mobility of these nodes [2].

In this chapter, a new moving algorithm for redeploying sensor nodes is proposed and operated at each node in an MWSN. The functional goal of the algorithm is to effectively relocate the position of a node and reallocate the capacities between any pair of communicating nodes. The allocation is based on specified communication constraints, such as preassigned bandwidth. Therefore, the redeployment algorithm is designed for enhancing the allocation of the channel capacity over the entire WSN. The proposed algorithm is based on the one presented in [1] and includes the concept of spreading capability studied in [3] to enhance the area coverage and related utility in an MWSN. Furthermore, extensive simulation studies are performed and the simulation results are analyzed in detail. The proposed redeployment

algorithm demonstrates excellent performance in terms of communication cost and relative moving distance. Furthermore, the communication capacity of all the nodes achieves their individual desired values. In addition to the combination of previously proposed concept in the literature, the following three concepts are proposed and simulated in this study:

1. Fixed topology: In the proposed algorithm of [1], the routing table changes every fixed period. This probably reduces the effect of relocating the nodes, since the algorithm relocates the node position according to the routing paths. As long as the routing table changes, the prerelocation may be useless or need to relocate. Therefore, in the proposed algorithm, the maintenance of topology is significant and emphasized.
2. Design of artificial forces: To find allocating criteria of capacities, several new indicators including *desired channel capacity* and *minimum channel capacity* are defined and discussed.
3. Performance and the accuracy: Based on the simulation result using MATLAB platform, the numerical performance indices are proposed and discussed.

The proposed moving algorithm can be mainly applied in the monitoring area, which needs to continuously transmit a large amount of data such as video or audio, or whose sensing data of an event is much larger than the normal ones [1]. This chapter has six sections including the Introduction section. Section 10.2 discusses literature survey of related research. Section 10.3 presents the problem formulation studied in the chapter. Section 10.4 describes the proposed redeployment algorithm in detail. The algorithm is divided into four parts including the initial preparation, the artificial force design for inner nodes and leaf nodes, and the stopping criteria. Also, the condition for achieving maximal capacity is discussed. Section 10.5 illustrates different cases of simulation study and presents related statistical analysis of these results. Finally, Sect. 10.6 concludes the chapter and outlines the future work.

10.2 Literature Survey of Related Researches

This section discusses the literature survey of related topics in MWSN including hardware, protocols, and algorithms.

10.2.1 Moving Algorithms for Mobile Wireless Sensor Network

For the applications in an MWSN, many moving algorithms were proposed in recent years. Heo and Varshney [3] proposed a spreading algorithm and related performance matrices such as coverage, uniformity, time, and distance. In [1], Popa et al. proposed a robot deployment algorithm for emphasizing the combination of classical robotic team concepts and traditional sensor network concepts. In [4], an indoor experiment is proposed. The experiment uses a robot deployment algorithm [5] and a discrete event controller.

10.2.2 Localization and Distance Measurement

In a WSN or an MWSN, the location of one sensor is important information for analyzing the sensing data. The location can be easily obtained if the WSN is a stationary network, and the sensor deployment is predefined. However, in an MWSN or an airdrop WSN, the location information is difficult to obtain unless the sensor node is equipped with GPS or other localization device. Hence, the mapping and the localization problems should be studied. In [6], Savvides et al. proposed a localization algorithm for WSN. The algorithm utilizes some anchor nodes distributed in the network to calculate and approach the correct location of other nodes. The location of the anchor nodes is known in advance. In [7], two distance measurement methods by using wireless communication variables, received signal strength (RSS) and time-of-arrival (TOA), are introduced. The measurement for TOA is more accurate than RSS.

10.2.3 Topology Control

When establishing a WSN, how to determine its connectivity line is a very important issue. If each node uses a maximum power for its data transmission, the interference in other node's communication task will be critical. If each node only uses a minimum power for its data transmission, the connectivity of this network will be frail and undependable. In general, these issues are considered as topology control. In [8] the authors used an optimal theorem to solve the topology control problem with transmission power adjustment. In the algorithm, the constraints are connectivity and biconnectivity. The objective is to optimize the power consumption. The topology control algorithm has been studied by many researchers in recent years. To analyze the global performance of a localized topology control, four topology control algorithms are proposed in [9]. The major differences in these four algorithms are the value of the constrained cone-angle and the minimum coverage radius. In addition, a distributed algorithm, called spanning tree, is proposed in [10]. In the algorithm, each node selects an appropriate transmission range and achieves the destination for saving energy consumption. In general, a topology control includes two kinds of algorithms. One is purely centralized algorithm; the other is purely decentralized algorithm. The cluster-based topology control framework is a combination of centralized and decentralized algorithms. This algorithm achieves both scalability and strong connectivity [11].

Up to now, most real applications of sensor networks are not adopted to the mobile nodes. In addition to the energy problem, some of the important difficulties are its cost and complexity. But an MWSN still has many advantages to be worthy to study. The aim of the proposed algorithm is to use node relocation approach to allocate the capacity of each communication link. The relocation has two superiorities relative to the static sensor networks. One is the number of nodes and the other is the effect of congestion control. For example, for the monitoring area, fewer mobile

nodes can achieve almost the same utility as more static nodes do. Furthermore, in a static network, there are likely many capacity bottlenecks due to the failure between two nodes. Node relocation can overcome this problem as long as shortening their distance. It can also be collocated to the flow control or congestion control protocols such as node collaboration and data aggregation. The network communication can be more effective and more robust.

10.3 Problem Formulation

In this section, related assumptions for the redeployment algorithm are discussed, and an MWSN model and related parameters are introduced.

10.3.1 Problem Description

The MWSN is assumed to be in a two-dimensional plane. N identical robots, including one stationary base station, are operated in the plane. Fundamental functions and capabilities of a node include sensing, wireless communication with the nodes within its communication range, motion, and detecting relative location of other nodes.

Initially, these networked nodes establish a routing table based on a given routing protocol. Then, every node transmits the sensed data to the base station through the path from the routing table. If an event occurs, the nodes which can sense the event need to transmit a large amount of data. To send and/or forward these data, the transmitting rate to the base station should be increased. Hence, the goal of the redeployment design is to relocate some of these nodes to other locations where a good transmission rate can be obtained. One scenario is shown in Fig. 10.2. Two aspects should be considered in the redeployment design. From sensing aspect, when the density of nodes is enough, the team of nodes can monitor the area effectively.

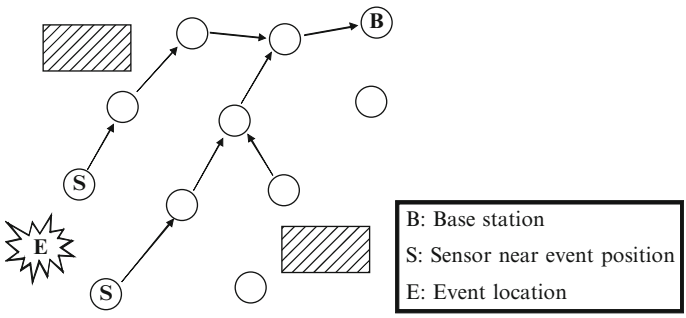


Fig. 10.2 Illustration for an event occurring in an MWSN

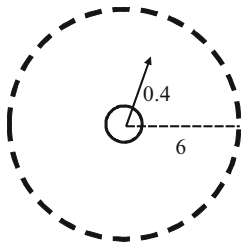


Fig. 10.3 Communication range and moving distance of one node

On the other hand, from the communication aspect, the nodes located nearby the event location, and the nodes between the event location and the base station should be able to support the required data transmission rate.

10.3.2 MWSN Model

The nodes can arbitrarily move to any direction in the two-dimensional plane. The maximum movement distance during a timestep is set as D_{lim}. The communication range of every robot is set as CR. For simplicity, the size of a robot is assumed to be zero. An example of $D_{\text{lim}} = 0.4$ and $\text{CR} = 6$ is shown in Fig. 10.3. For establishing routing table, the maximum number of connectable nodes is set as 5. In the redeployment design, every node calculates the resultant of artificial forces. This resultant force is used as the next movement distance unless the value is larger than D_{lim}. In that case, the node only moves D_{lim} unit.

Figure 10.4 illustrates the MWSN model used in this chapter. The inputs to every node are the initial position of communicable nodes, the distance between connected nodes, the transmitting rate, and the number of communicable nodes. The redeployment algorithm then outputs the movement for every node at next timestep. The routing table is constructed based on a distributed routing protocol which is a modified and simplified version of the ZRP routing protocol discussed in [12]. Shannon's channel capacity formula is used to construct the wireless channel feature [1]. The transmitting rate of each node is the same as the sensing rate, and the sensing rate may be changed according to the characteristics of different events.

10.3.3 Performance Index

Stability, mean, and standard deviation are the main performance indices used in this study. To the nodes in an MWSN, how to achieve the destination by using less moving distance is very important (i.e., energy consumption). In addition, the efficiency of their movement is also important (i.e., accuracy and time consumption). Hence, the stability performance is adopted to estimate the energy consumption, and mean and standard deviation are adopted to estimate the performances of accuracy and time consumption.

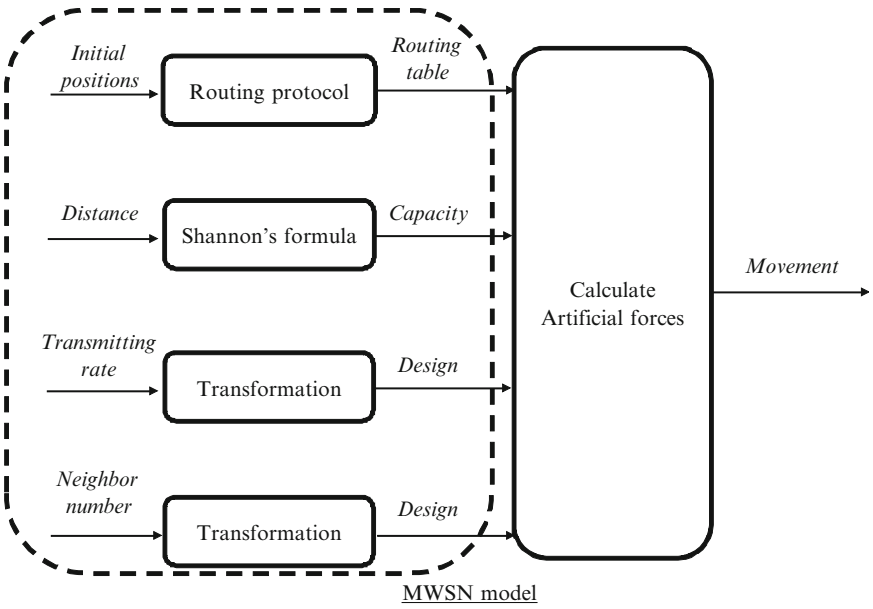


Fig. 10.4 Illustration of the MWSN model with redeployment algorithm

The stability denotes that nodes are stable if they move more and more shortly as time increasing. If they are not in stable condition, the node moving with large distance almost all the time, the power consumption would be very large. On the contrary, if they are in a stable condition, the new relocation is quickly found. The power consumption can decrease effectively. Mean and standard deviation are the observers for the algorithm accuracy. For analyzing the simulation results, the sensing rates of all the nodes are set as the same. To judge whether the network capacity assignment fulfills the goal, the accuracy of fitness can be easily observed from the mean and the standard deviation.

10.4 Redeployment Algorithm

In this section, the proposed redeployment design algorithm is discussed in detail. The algorithm is a distributed algorithm for assigning the capacity of every communication link in an MWSN. In a conventional centralized algorithm, the movement of one node needs a centralized command and every node transmits the state information back to the central control station. This kind of algorithms may cause a lot of communication delay and cost. In the proposed algorithm, each node only needs the relative location and ID of other nodes within its communication range. By analyzing the information, the algorithm can then drive the node to a proper location.

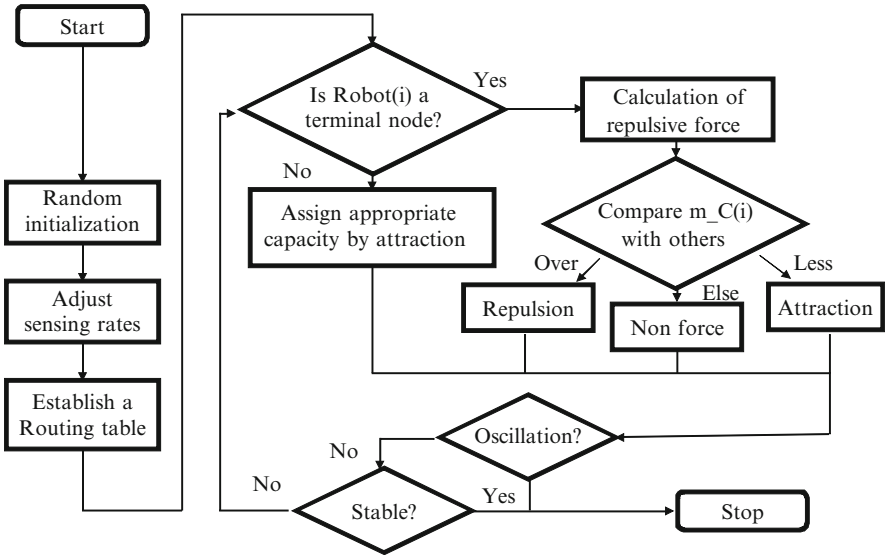


Fig. 10.5 Flow chart of redeployment algorithm

The ultimate goal of the algorithm is to enhance the channel capacity between any pair of communicating nodes. The channel capacity mainly depends on their relative distance. Therefore, the mechanism for assigning the channel capacity is to let each node find a suitable location and move to it. In an MWSN, the nodes need to transmit their sensed data to the base station, and the data is transmitted by hopping based on a preestablished routing table. If all the nodes transmit data at the same rate and the channel capacity assignment is inappropriate, many bottlenecks may be distributed in the wireless networking and then degrade the communication performance. Furthermore, if some nodes detect an interesting event and need to transmit large amount of sensed data back to the base station, the original channel capacity assignment may not be enough. At this time, if there are other idle nodes which can support extra data transmission, the channel utilization can then be reorganized. To effectively overcome the situation, the algorithm adapts the channel capacity according to the sensing rate on each node. Figure 10.5 shows the fundamental flow chart of the redeployment design algorithm. The algorithm is divided into four parts, including initial preparation, artificial forces for inner nodes, artificial forces for leaf nodes, and stopping criteria. These parts are discussed in detail in the following.

10.4.1 Initial Preparation and Fundamental Definitions

The first part of the redeployment design algorithm is the initial preparation, including the information of the initial location of nodes, the initial transmission rate,

and the established routing table. The key mechanisms are "Random initialization", "Adjust sensing rates", and "Establishing the routing table" as shown in Fig. 10.5. The initial sensing rate of every node is set as 1 unit. That is, each node transmits the sensed data to the base station at the same rate. It is assumed that the transmission rate is the same as the sensing rate. The routing table is established based on the ZRP with some modification [12].

The following fundamental definitions are used to calculate the artificial force for the movement in the redeployment design.

- $\boldsymbol{A\_C_{ij}(k)}$ is the actual channel capacity between robot i and robot j at the kth time-step. The relationship between the capacity and the distance is established based on the Shannon's channel formula, introduced in [2]. This formula is discussed and simplified as a simple model in [1] (Fig. 10.6). That is, the channel capacity and the relative distance between two communicating robots are inversely related. Furthermore, if the relative distance is less than a lower limit, $r_{\min}$, the relative capacity is maintained as a saturated value. On the other hand, if the distance is larger than an upper limit, r_{zone}, the channel capacity is close to zero. For the relative distance is between $r_{\min}$ and r_{zone}, the channel capacity decays exponentially with the increased relative distance.
- $\boldsymbol{D\_C_{ij}(k)}$ is the desired capacity between robot i and robot j at the kth time-step. The desired capacity is calculated based on the current routing table and the current sensing rate. Figure 10.7 shows the relationship of a simple example. Because there is no event detected and each robot senses and

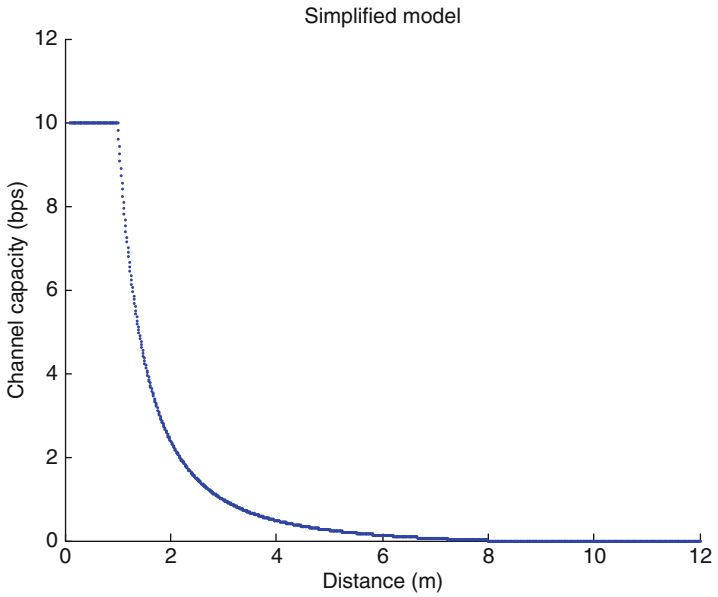


Fig. 10.6 Channel capacity as a function of its distance with $r_{\min} = 1$ and $r_{\text{zone}} = 8$. [1]

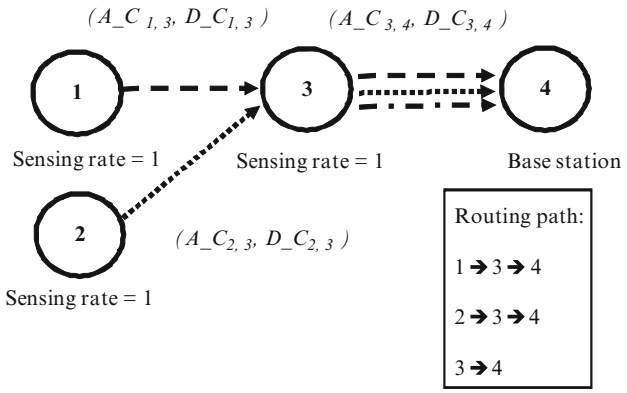


Fig. 10.7 Illustration of the definitions with $D\_C_{1,3} = 1$, $D\_C_{2,3} = 1$, $D\_C_{3,4} = 3$

transmits data at the same rate, it is assumed that the sensing rate of each robot is 1 unit. The routing paths are $1 \to 3 \to 4$, $2 \to 3 \to 4$, and $3 \to 4$, where node 4 is the base station. Accordingly, the D_C of each link can be calculated. Because $\text{Link}_{1,3}$, $\text{Link}_{2,3}$, $\text{Link}_{3,4}$ have 1-, 2-, and 3-unit data, $D\_C_{1,3} = 1$, $D\_C_{2,3} = 1$, and $D\_C_{3,4} = 3$, respectively.

- ***m_C_x(k)*** is the minimum channel capacity of the routing path for robot x at the kth time-step. It can be used to indicate the bottleneck in the routing path. In some shared channels (i.e., $D\_C \neq 1$), the capacity is only used to transmit data to the base station. The channel capacity shared is based on the function shown as follows.

$$m\_C_x = \min_{i,j \in RP_x} \left(A\_C_{ij} \times \frac{\text{SR}_x}{D\_C_{ij}} \right), \tag{10.1}$$

where SR_x is the sensing rate of each node x, and RP_x is the routing path to the base station for node x. Equation (10.1) shows that m_C_x is the minimal assigned capacity of the routing path with respect to node x. In other words, m_C_x denotes the bottleneck capacity of a whole routing path, and node x transmits data to the base station limited by this value.

10.4.2 Artificial Force for Inner Nodes

In this section, the artificial force for inner nodes is discussed. The artificial force for leaf nodes is discussed in the next section.

After the initial preparation, each node starts to calculate the direction and the movement distance at the next time-step. In this design, the team of networked nodes is divided into two groups, namely, leaf nodes and inner nodes. Leaf nodes are the

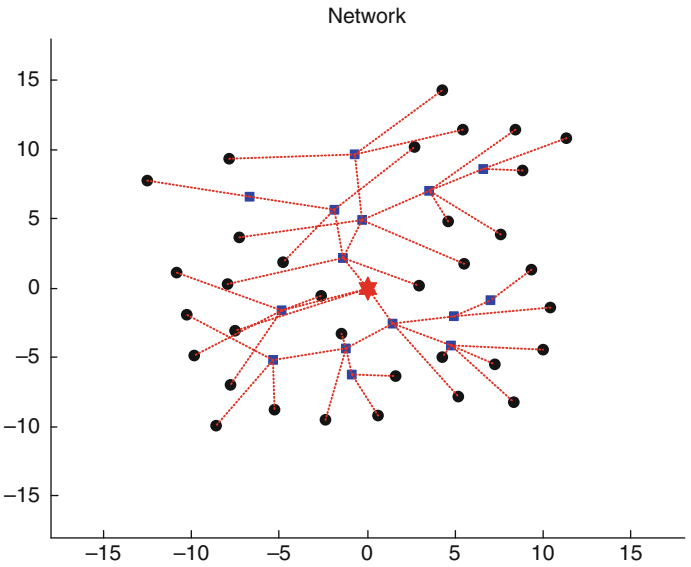


Fig. 10.8 Illustration of leaf nodes (●) and inner nodes (■). Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

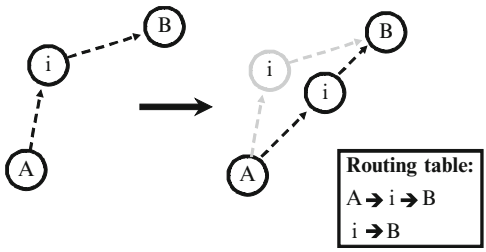


Fig. 10.9 Movement illustration of inner nodes

ones that do not need to forward the sensed data of other nodes to the base station. On the contrary, inner nodes are the ones that may need to forward the sensed data of other nodes to the base station. Figure 10.8 shows an example of leaf nodes and inner nodes.

The design is distributed and independently performed at each node. The mission of the design at node i is to assign the channel capacity between node i and communicating nodes based on the desired channel capacity, D_C, and to maximize all the channel capacities. An illustrated example is shown in Fig. 10.9, where node B is the base station, node A is a leaf node, and node i is any inner node (i.e., it is not a leaf node in a routing path). For ease of illustration, nodes A and B are assumed to be stationary. In this scenario, node i is affected by two attractive forces from nodes A and B, respectively. The resultant of these two forces then drives node i to a new

location where the channel capacity is maximum. The forces from nodes A and B are the same and shown as follows.

$$F_{ij} = \left(m\_C_i - A\_C_{ij} \times \frac{\mathrm{SR}_x}{D\_C_{ij}} + \mathrm{Shift} \right) \times \frac{1}{A\_C_{ij}/D\_C_{ij}} \times \frac{p_j - p_i}{|p_j - p_i|}, \quad (10.2)$$

where p_x is the position of node x The first part in (10.2) works as a constant. The second part is a ratio. If the ratio is large, the attractive force becomes smaller to increase the relative distance; if the ratio is small, the attractive force becomes large to decrease the relative distance. The third part works as a unit vector from p_i to p_j.

10.4.3 Artificial Force for Leaf Nodes

In this section, the artificial force for leaf nodes is discussed. The position of leaf nodes can determine the approximate sum of path capacities. One of the functional goals of the proposed moving algorithm is to allocate the capacities to each communication link. For this goal, this artificial force is designed to adjust the position of leaf nodes to get the desired capacity. In addition, a repulsive artificial force is designed to let the node density become smaller. The combination of previous two forces is for the leaf nodes.

If node i is a leaf node, it first calculates the total channel capacity in the first link of the routing path. If the total channel capacity needs to be adjusted, node i change its location accordingly. An illustrated example is shown in Fig. 10.10, where node i is a leaf node, node A is an inner node, and node B is the base station. When the related distance between node i and node B is too large, the sum of channel capacities $Link_{A,i}$ and $Link_{A,B}$ could be too small. On the other hand, when the related distance is too small, the sum of the two channel capacities could be too large. In this case, node A needs to provide a force to adjust the sum. The artificial force executed at node i is shown as follows.

$$F_{ij} = (m\_C_i - \mathrm{average}(m\_C)_i) \times \frac{p_j - p_i}{|p_j - p_i|}, \quad (10.3)$$

where average $(m\_C)_i$ is the average of m_C_x, and x denotes the set of nodes within the communication range of node i. Node j is the node first receiving the

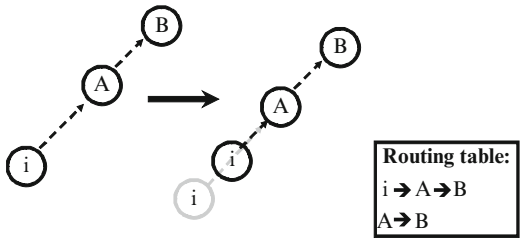


Fig. 10.10 Illustration for leaf nodes

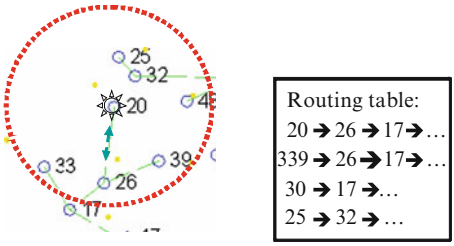


Fig. 10.11 Illustration of a leaf node (node 20) and its neighbor nodes

message from node i according to the routing table (in Fig. 10.10, node j is node A). While $m\_C_i > \text{average}(m\_C)_i$, the force is positive (i.e., attractive), and, when $m\_C_i < \text{average}(m\_C)_I$, the force is negative (i.e., repulsive).

In addition to the artificial force, node i is also affected by the repulsive forces from the other nodes within its communication range. The goal of these forces is to average the density (the number of nodes within the communication range) in an MWSN. When the density is large, this force could be strong. On the contrary, if the density is small, this force could be weak. The formula of this force is given by [3]:

$$F_{ij} = D_j \times (\text{CR} - |p_i - p_j|) \times \frac{p_j - p_i}{|p_j - p_i|}, \tag{10.4}$$

where CR is the communication range and D_j is the density within node j. This force increases the uniformity of a team of networked nodes. Figure 10.11 illustrates the artificial forces to a leaf node. Considering node *20*, the dotted circular line is its communication range. In this case, node *26* provides the force of (10.2) to node *20*. Nodes *25*, *32*, *33*, *39*, and *48* provide the force of (10.3) to node *20*.

10.4.4 Artificial Force Toward Events

In addition to previous two kinds of artificial forces, another special artificial force is produced when an event occurs. The force is applied to any mobile node in an MWSN when the node can detect the event. If the force is not added, the node located inside and near the boundary of the sensible range may be in an oscillating status. During the event occurring, the node should increase its sensing rate. Hence, it needs to move closed to its neighboring nodes to increase the channel capacity. This may make the node leave the sensible range.

This artificial force is designed and the fundamental concept of this force is introduced in [1], and shown as follows:

$$\frac{1}{|p_{\text{event}} - p_i|} \times \frac{p_{\text{event}} - p_i}{|p_{\text{event}} - p_i|}, \tag{10.5}$$

where p_{event} is the location of occurring event, p_i is the location of nodes which sense the event. Some weighting gain can be used to adjust the influence of the artificial force to the total force.

10.4.5 Relationships Between the Artificial Forces

To adaptively combine all the artificial forces, every force needs times a gain to adjust the ratio of them. These gains represent the weighting of each force. In addition, a force limitation is designed to avoid the situation that any force is too strong to move unreasonably.

Figure 10.12 illustrates the gains and the limitation for node i. If node i is an inner node, the block "Receive" is the attractive force from a node which first receives the messages from node i, and another block "Transmit" is the attractive force from a node which transmits messages to node i. The two forces are calculated based on (10.2). The gain *G2* denotes a reasonable ratio of the two forces. If node i is a leaf node, the block "Spreading" is the force to spread the networked nodes (10.3), and another block "Capacity" can be a repulsive or an attractive force to adjust the sum of channel capacities in the routing path (10.4). The gain *G4* denotes a reasonable ratio of these two forces. In addition to these four kinds of artificial forces, the block "Event" is calculated based on (10.5) and the gain *G5* denotes the weighting of this artificial force. Every node would calculate this force as it can detect an event.

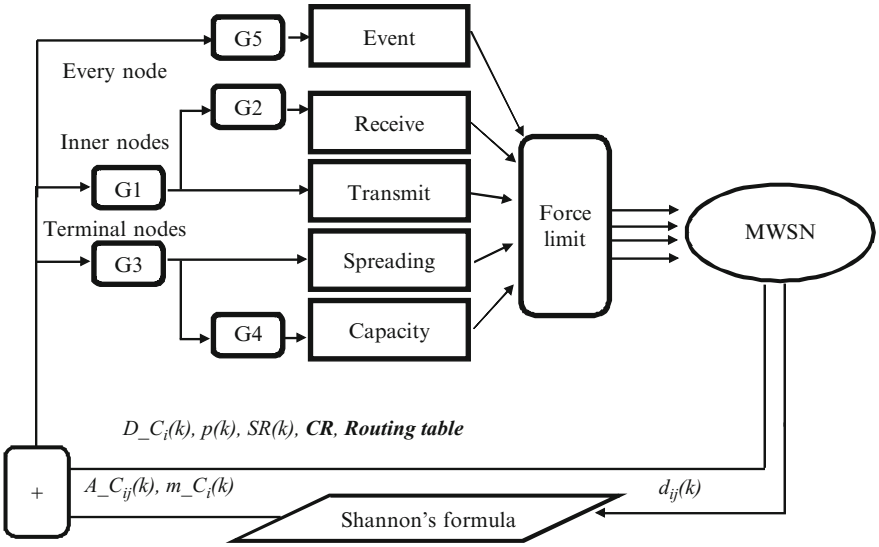


Fig. 10.12 Closed-loop structure of redeployment algorithm

"Receive" and "Transmit" blocks denote the artificial forces bore on inner nodes and leaf nodes, respectively. Before those gains and forces, two gains *G1* and *G2* are used to control the strengths of the resultant for inner nodes and the resultant for leaf nodes, respectively. At last, a filter for force limitation bounds the strength of each force. The reason for using the filter is to avoid the link broken and an unreasonable movement. Then, each node calculates the forces acting on it and its movement. After the movement, each node can update new parameters from the nodes within its communication range, and recalculate a new movement for next time-step. The system continually operates in a closed-loop fashion until the stopping criterion is achieved.

10.4.6 Stopping Criteria

How to stop the movement is an important issue for MWSN. In [3], two stopping criteria were introduced. "If a node moves less than e for the time duration S_{lim}, this node can be considered to be in the stable status and that node stops its movement." In addition to the introduced part, a speed-decreased buffer is designed and added in this algorithm. If a node is considered to be in the stable status, the node will not stop its movement immediately. The speed buffer decreases the distance interval of the node step by step until a zero value is reached. This can increase the accuracy of the capacity assignment. But during a monitoring task, an MWSN needs to process a varied environment at any time. A stable state maintaining forever is not existent. For this reason, an additional condition for the first stopping criterion is proposed. If a bore force over a force limit, $\text{force}_{\text{lim}}$, is calculated, the node starts its movement again.

Another stopping criterion is: "If a node moves back and forth between almost the same locations many times, this node is regarded in the oscillation status." How to judge this oscillation status is designed in the following process. First, each node will calculate the distance between the current location and the location at last "five" time-step. If this distance is less than a pre-designed parameter $\text{distance}_{\text{lim}}$, this node continues calculating the distance between the current location and the location at last "ten" time-step. The loop lasts several times to judge if the node is in an oscillation state. If it is the case, the node will start the speed-decreased buffer as first criterion did.

10.4.7 Maximal Achievable Capacity

The maximal capacity that every node can transmit the data to the base station is called the maximal achievable capacity. The maximal achievable capacity to all MWSN nodes is determined based on its initial routing table. In this section, a calculated method is proposed. To demonstrate this algorithm, Fig. 10.13 is used for

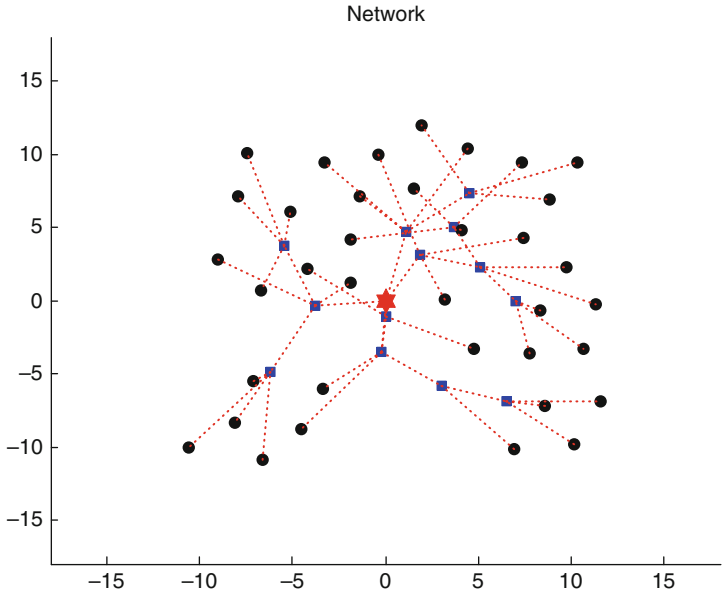


Fig. 10.13 Spreading MWSN. Circles (●) are leaf nodes and squares (■) are inner nodes. Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

the illustration. Four nodes are directly connected to the base station. These nodes are called shared nodes. The number of nodes transmitting data through the shared nodes to the base station is defined as shared number (includes shared node itself). The function of maximal achievable capacity is shown as follows:

$$\max(\text{achievable\_capacity}) = \frac{\text{Saturated\_capacity}}{\max(\text{Shared\_number})} \tag{10.6}$$

For this example, the maximal shared number is 16 and its shared node is node *12*. According to the channel capacity model discussed in Sect. 10.4.1, the saturated channel capacity is 10 bps. Therefore, if the shared channel of node *12* is in the saturated state, the channel capacity that every node shared in this branch can be dealt out is $10/16 = 0.625$ (bps). If any node in this branch has a shared capacity over the maximal achievable capacity, there is at least one node in this branch that cannot achieve this capacity.

The achievable capacity of a branch which has maximal shared number nodes is the critical achievable capacity of the whole MWSN. As long as the nodes in this branch can achieve this capacity, all other nodes in the MWSN can achieve too. Hence, this capacity is the maximal achievable capacity of the MWSN.

10.5 Simulation Study

In the section, the artificial forces for driving the movable nodes are briefly discussed in Sects. 10.5.1 and 10.5.2, respectively. Then, a fundamental simulation study for an MWSN and related analysis is discussed in Sect. 10.5.3. Figure 10.14 shows the flow chart of this fundamental simulation study. In this simulation, the MWSN has 50 identical nodes and the assumed communication range is 6 units. The simulation scenario is further divided into three modes for different capacity convergence. The analysis of these different modes is presented in detail in this section. Finally, an advanced simulation study and related analysis are discussed in Sect. 10.5.4. Related simulation studies are performed at the MATLAB platform. Related analysis and performance index including mean and standard deviation are discussed in detail.

10.5.1 Artificial Force for Inner Nodes

In the fundamental simulation for inner nodes, it is assumed that only the inner node is moveable. Figure 10.15 shows the moving trajectory of inner nodes driven by the artificial force. In Fig. 10.15, node 4 is a base station, and every node has to transmit data to node 4. It is assumed that the sensing rate of nodes 1, 2, and 3 is 1 unit/time. The artificial force algorithm assigns the capacities between communicable nodes according to the parameter D_C of each communication link.

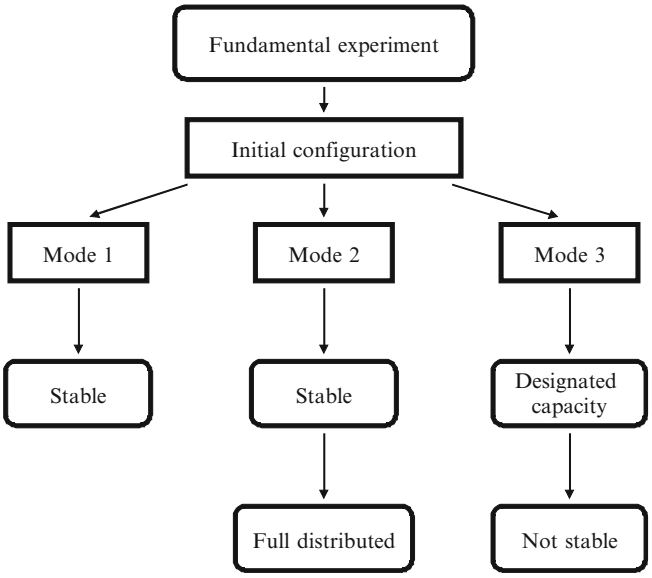


Fig. 10.14 Structure of fundamental experiment

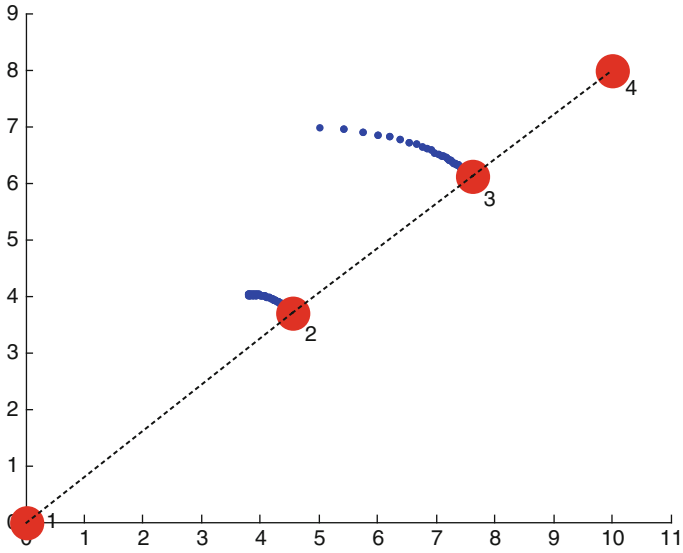


Fig. 10.15 Trajectory of two inner nodes, nodes 2 and 3. Node 4 is the base station and node 1 is a leaf node. Dotted lines indicate the routing paths

In this case, the ratio of each link capacities ($A\_C_{12} : A\_C_{23} : A\_C_{34}$) is 1:2:3. After the operation of algorithm, the distances of each link are $\text{Distance}_{1,2} = 5.8638$, $\text{Distance}_{2,3} = 3.9042$, and $\text{Distance}_{3,4} = 3.0384$. The capacities of each link are $A\_C_{1,2} = 2.5278$, $A\_C_{2,3} = 5.0539$, and $A\_C_{3,4} = 7.579$. Hence, this simulation result satisfies the requirement.

In another case, as shown in Fig. 10.16, node 4 is also a base station. The sensing rate of every node is 1 unit/time. Nodes 1 and 2 transmit data to the base station through node 3. Node 3 also transmits its sensed data to base station. Then, D_C_{34} is 3, and both D_C_{13} and D_C_{23} are 1. Then, the inner node, node 3, needs to move a proper location to fulfill the requirement of D_C at each link. In this case, the distances of each link are $\text{Distance}_{1,3} = 4.3925$, $\text{Distance}_{2,3} = 4.3507$, and $\text{Distance}_{3,4} = 2.8205$. The capacities of each link are $A\_C_{1,3} = 4.1577$, $A\_C_{2,3} = 4.2246$, and $A\_C_{3,4} = 8.5305$. However, this situation generates a bottleneck in $Link_{3,4}$, because A_C is not enough to support the desired transmitting rate. To resolve this problem, a weighting term K is added. The function of this term is to weaken the force strength to decrease the waste of channel capacity. Figure 10.17 shows the simulation result. The capacities of each link are $A\_C_{1,3} = 3.8443$, $A\_C_{2,3} = 3.8454$, and $A\_C_{3,4} = 10.251$.

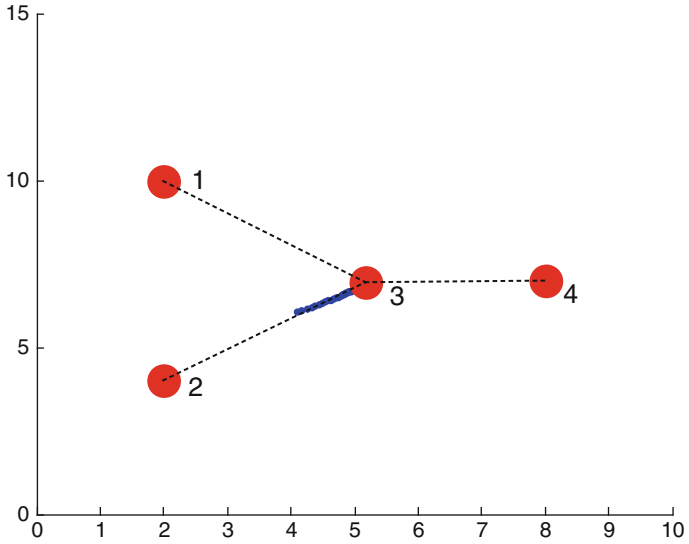


Fig. 10.16 Trajectory of an inner node with three leaf nodes. Node 3 is an inner node. Node 4 is the base station. Nodes 1 and 2 are leaf nodes. Dotted lines indicate the routing paths

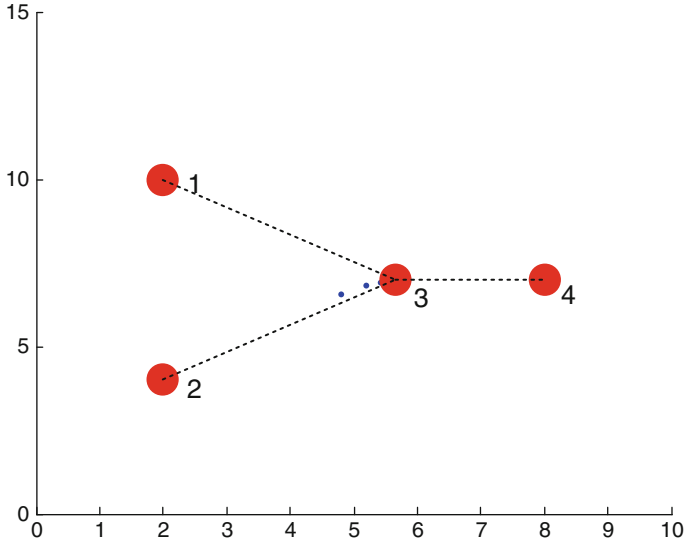


Fig. 10.17 Trajectory of an inner node with added K into the artificial force. Node 3 is an inner node. Node 4 is the base station. Nodes 1 and 2 are leaf nodes. Dotted lines indicate the routing paths

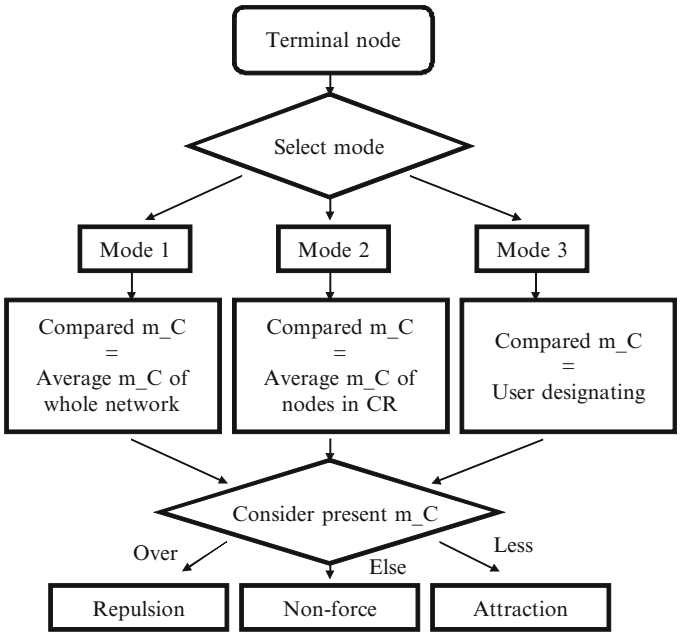


Fig. 10.18 Three convergence modes for the capacity of leaf nodes

10.5.2 Three Convergence Modes for Leaf Nodes

In the redeployment algorithm, the main mission of leaf nodes is to determine the maximal sum of link capacities in a routing path. Figure 10.18 shows three modes to select the compared m_C:

- Mode 1: Each leaf node compares its m_C with the average m_C of the whole network.
- Mode 2: Each leaf node compares its m_C with the average m_C of the nodes within its communication range.
- Mode 3: Each leaf node compares its m_C with a pre-designated destination.

The leaf nodes of an MWSN should compare its m_C with another value to determine whether the artificial force is repulsion or attraction or not as discussed in Sect. 10.4.3. In the following, the performance and stability of these three modes will be compared and discussed.

10.5.3 Fundamental Simulation with $N = 50$

In this section, a static simulation will be analyzed. The simulation focuses on the stability analysis of the designed MWSN system. Figure 10.19 illustrates random

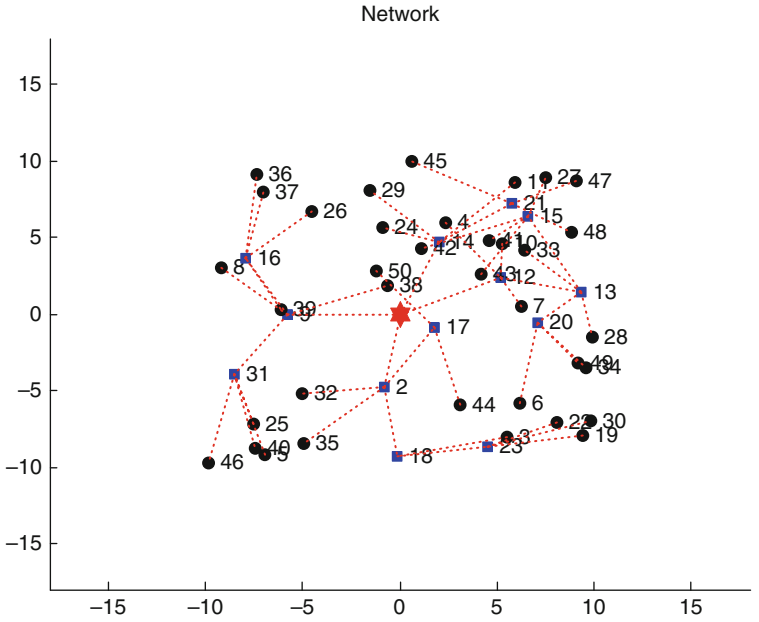


Fig. 10.19 Initial position of $N = 50$ nodes with and CR $= 6$. Circles (●) are leaf nodes and squares (■) are inner nodes. Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

position of 50 nodes and its routing table. These nodes are distributed in a 20×20 unit plane, and the sensing rate of each node is 1 unit/time. Figure 10.20 shows the initial m_C (bps) of each node, where the x-coordinate is the ID of each node and the y-coordinate is m_C (bps). Since the sensing rates are 1 unit/time, the goal of the algorithm is to equally assign the capacities to each node. Other parameters used in this simulation are: $G1 = 0.006$, $G2 = 1$, $G3 = 0.000006$, $G4 = 1{,}000$, Force limit $= 0.4$, and Converged range $= 0.9$. For the same initial configuration, three modes are discussed in Sects. 10.5.3.1, 10.5.3.2, and 10.5.3.3, respectively.

10.5.3.1 Mode 1

The final distributed configuration of simulation result is shown in Fig. 10.21. It is more dispersive than the original distributed configuration as shown in Fig. 10.19. Figure 10.22 shows the m_C of each node after the artificial force at the 1401st time-step. The mean of m_C is *0.1690* and the standard deviation is *0.0097*. Figure 10.23 shows that the algorithm has an excellent convergence performance on the channel capacity assignment. The convergence includes both capacity and distance interval of each time-step.

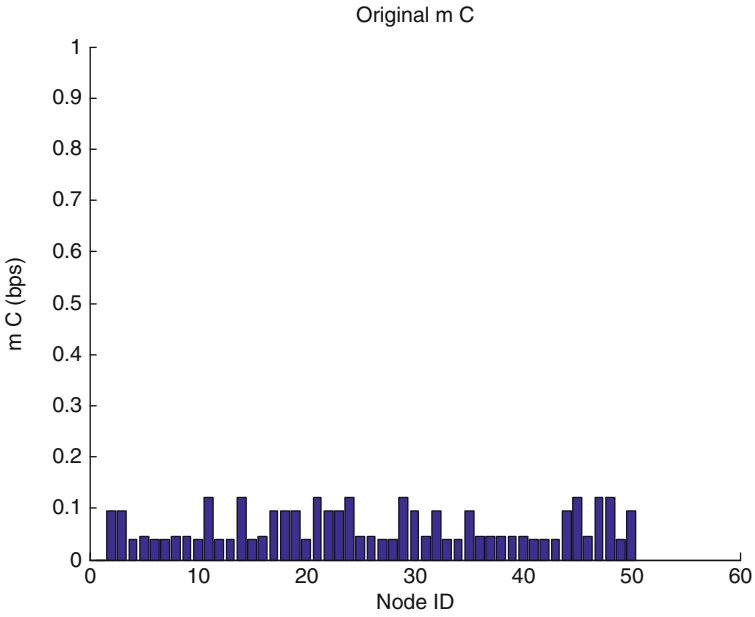


Fig. 10.20 Initial m_C of each node before processing redeployment algorithm

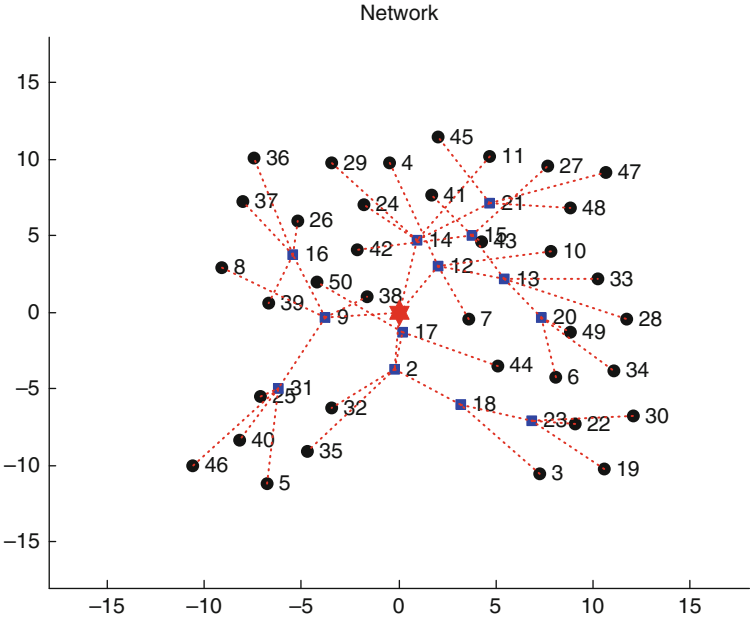


Fig. 10.21 Simulation result with $N = 50$, time-step $= 1401$, and mode $= 1$. Circles (●) are leaf nodes and squares (■) are inner nodes. Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

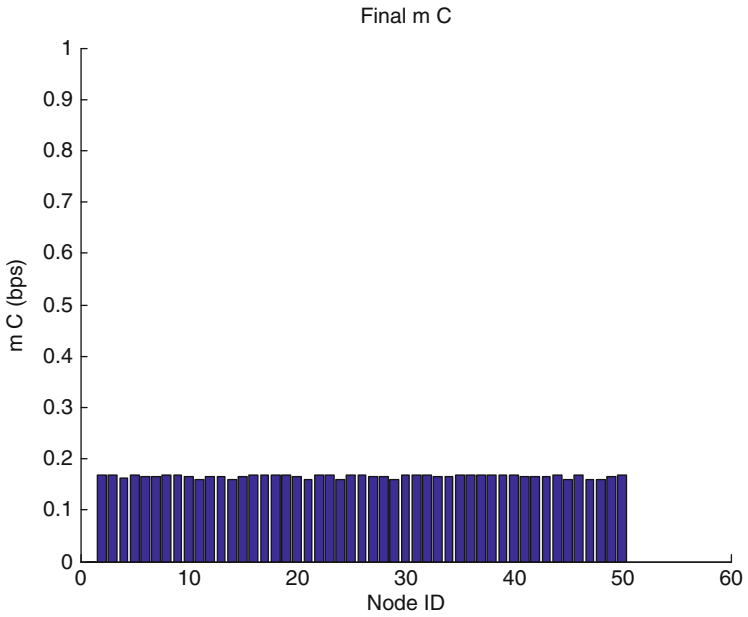


Fig. 10.22 Bar chart for m_C of each node after processing the algorithm

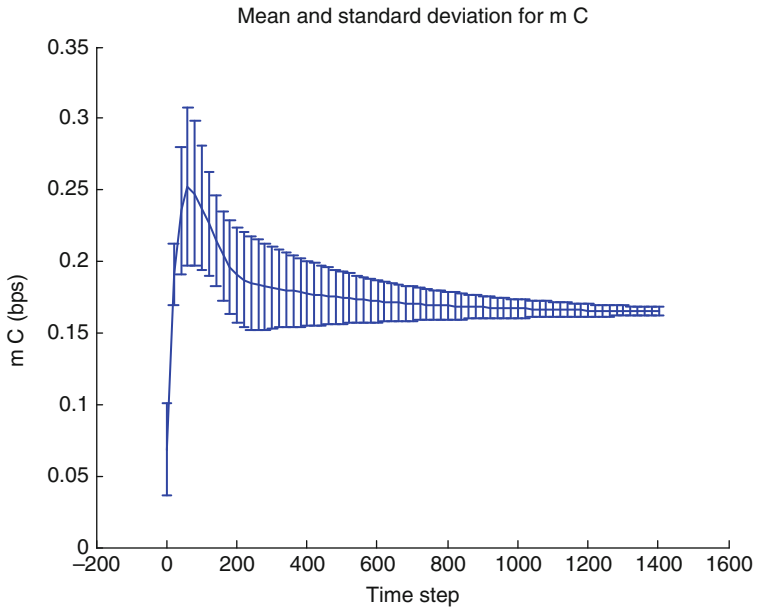


Fig. 10.23 Error bar chart for m_C form time = 1 to time = 1,401

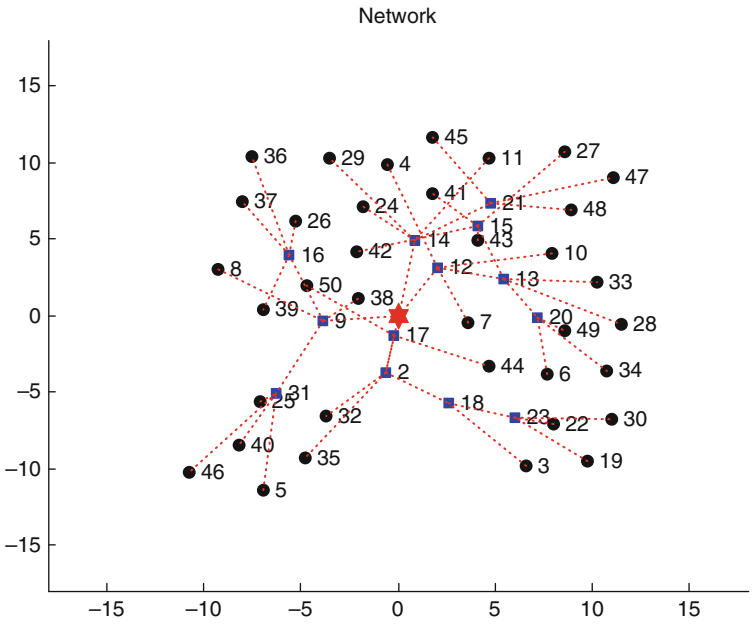


Fig. 10.24 Simulation result with $N = 50$, time-step $= 1401$, and mode $= 2$. Circles (●) are leaf nodes and squares (■) are inner nodes. Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

10.5.3.2 Mode 2

In mode 1, the compared *m_C* is the mean capacity of whole network. To make the algorithm to be more distributed, in mode 2, the compared *m_C* is modified to the capacity mean of the nodes within the communication range of the considered leaf node. Figure 10.24 shows the network configuration of the simulation result.

In Fig. 10.25, the mean and the standard deviation of *m_C* are *0.1630* and *0.0186*, respectively at the 1401st time-step. This result is slightly worse than the result of mode 1. In mode 1, the mean and the standard deviation of *m_C* are *0.1690* and *0.0097*, respectively at the 1401st time-step. The reason is that the performance of a distributed algorithm is, in general, worse than a centralized one. In addition to this disadvantage, the result of mode 2 is close to the result of mode 1. But mode 2 is distributed.

10.5.3.3 Mode 3

In this section, a predesignated goal of *m_C* for the maximal achievable capacity is studied. Figure 10.26 shows the simulation result with a goal value at 0.17. At the 1401st time-step, the mean is *0.1648* and the standard deviation is *0.0089*. This result indicates that the network and the capacities can achieve the designated goal.

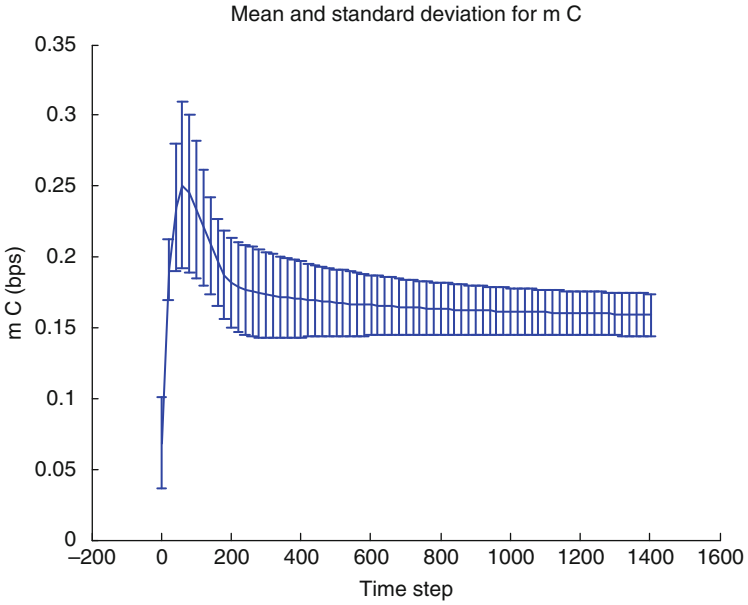


Fig. 10.25 Error bar chart for m_C from the 1st to the 801st time-step with mode 2

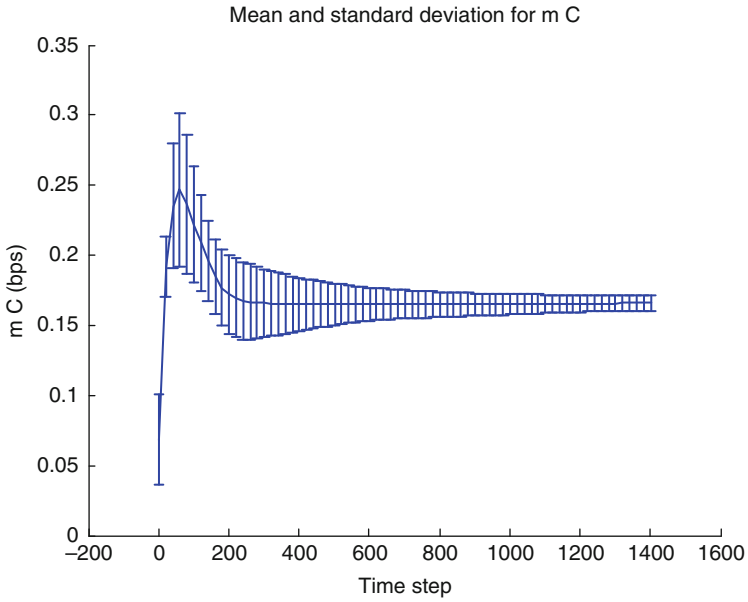


Fig. 10.26 Error bar chart for m_C from the 1st to the 1401st time-step with mode 3

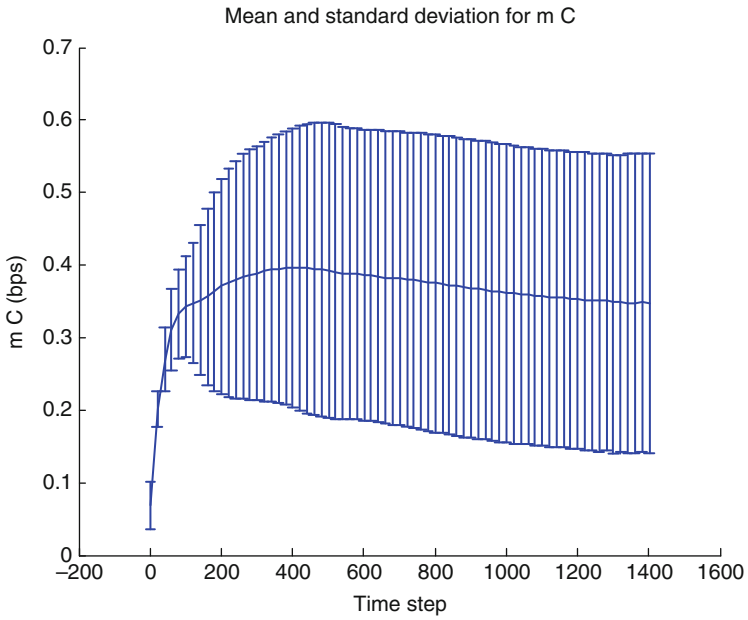


Fig. 10.27 Error bar chart for *m_C* with designated capacity = 0.625

To test the upper bound of capacity, the goal value is set to the maximal achievable capacity. In this case, the maximal achievable capacity is 0.625. Figure 10.27 shows the simulation result that the goal capacity is 0.625. In this simulation result, the mean is *0.3747* and the standard deviation is *0.2051* at the 1401st time-step. It is clear that the final capacity mean did not achieve the goal value. The standard deviation and the movement are not stable either. The reason is that the design of the artificial forces has some limitations.

To overcome these limitations, some gains in the algorithm must be modified. In the case, $G2 = 2$ and $G4 = 1{,}000{,}000$. Then, the network configuration of the simulation result is shown in Fig. 10.28. In Fig. 10.29, the error bar chart shows that the mean is *0.5661* and the standard deviation is *0.0097* at the 1401st time-step. The mean is very close to the designated capacity and the error is also very low.

10.5.4 Simulation with Time Variation

In this section, two simulations with environment variation are studied. In the first simulation, the goal values of capacity are reset at two different time steps. In the beginning, the initial goal capacity is 0.2. At the 301st time-step, the goal capacity is changed to 0.4. Finally, the goal capacity is changed to 0.1 at the 601 time-step. Figure 10.30 shows the error bar chart of *m_C* for the simulation result. In this

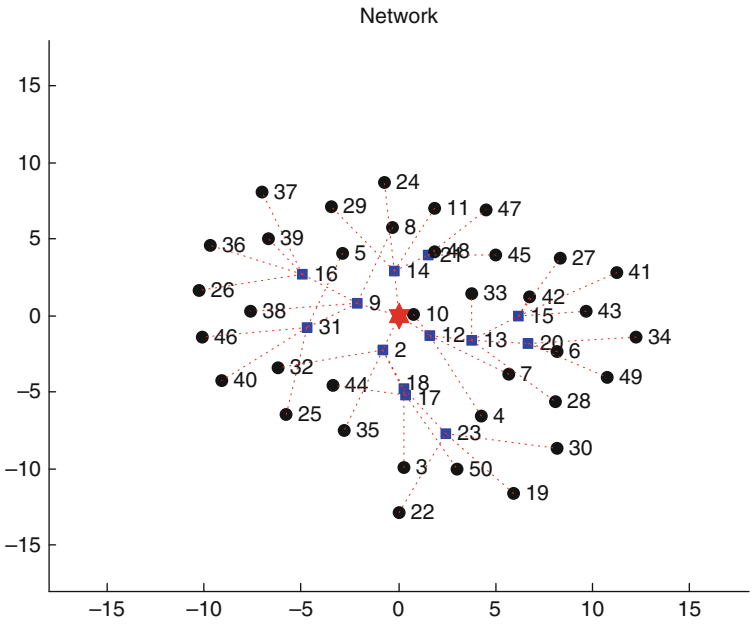


Fig. 10.28 Final configuration with $G2 = 2$ and $G4 = 1{,}000{,}000$. Circles (●) are leaf nodes and squares (■) are inner nodes. Star (★) indicates the base station. Dotted lines (- - -) indicate the routing paths

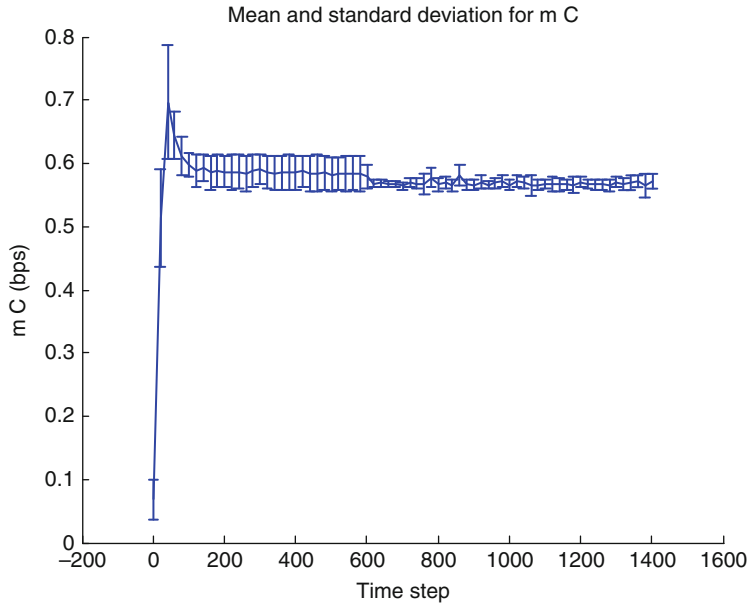


Fig. 10.29 Error bar chart for m_C with $G2 = 2$ and $G4 = 1{,}000{,}000$

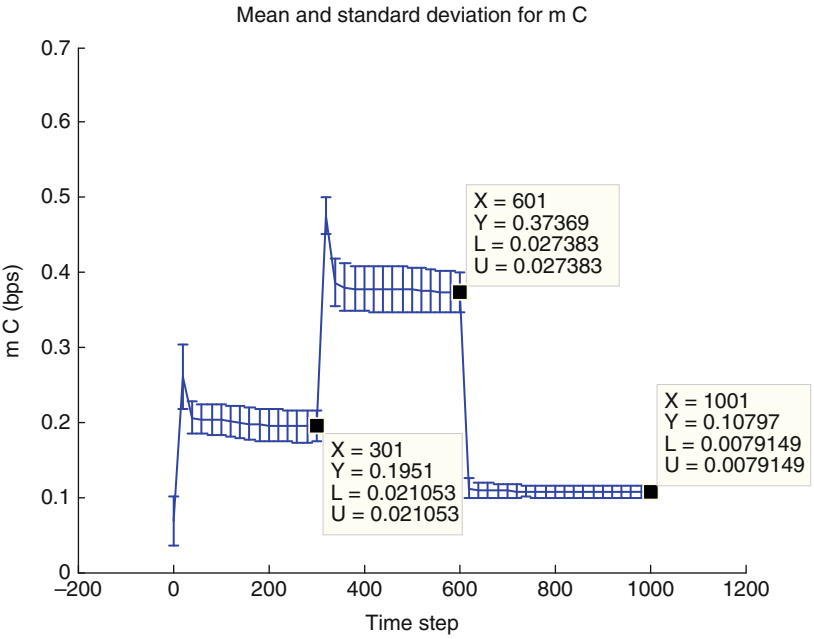


Fig. 10.30 Error bar chart for *m_C* with time variation of designation

figure, the values of m_C are close to the goal value and the standard deviations are very low. At the 301st time-step, the mean is *0.1951* (the designated value is 0.2) and the standard deviation is *0.021053*. At the 601st time-step, the mean is *0.037369* (the designated value is 0.4) and the standard deviation is *0.027383*. At the 1001st time-step, the mean is *0.10797* (the designated value is 0.1) and the standard deviation is *0.0079149*. In these results, the ratios of standard deviation to mean are all smaller than 10.8%.

In the second simulation, the sensing rate of each node is changed twice to simulate a monitoring task with the MWSN. A set of nodes in the MWSN will increase their sensing rates to adapt to a new event detected in their sensing range. At the beginning of the simulation, the sensing rate of every node is 1 unit/time and the goal capacity is 0.2 bps. At the 301st time-step, nodes 1–20 increase their sensing rate to 2 unit/time. At the 601st time-step, nodes 21–50 also increase their sensing rate to 1.5 unit/time. Figures 10.31 and 10.32 show the simulation result. In Fig. 10.31, the standard deviations from the 301st to the 1001st time-step seem very large. The reason is that there are two different sets of sensing rates. The mean and the standard deviations of these two different sets are illustrated in Fig. 10.32. Figure 10.32a–c show the *m_C* of each node at the 301st, the 601st, and the 1001st time-steps, respectively. The capacities of each node fulfill the requirement of the goal capacity and the sensing rates.

In Fig. 10.32b, the mean of nodes 2–20 is *0.4054* (the designated value is 0.2×2 (sensing rate) $= 0.4$) and its standard deviation is *0.0249*. The mean of nodes

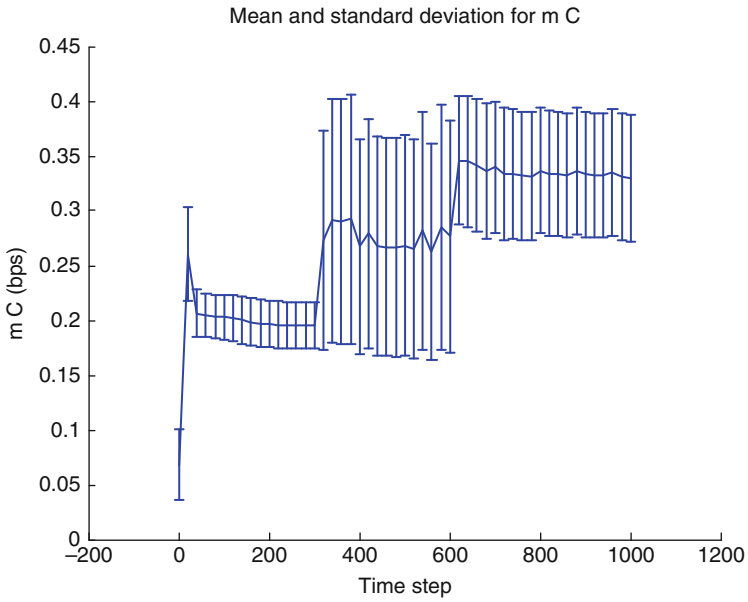


Fig. 10.31 Error bar chart for m_C with sensing rate variation

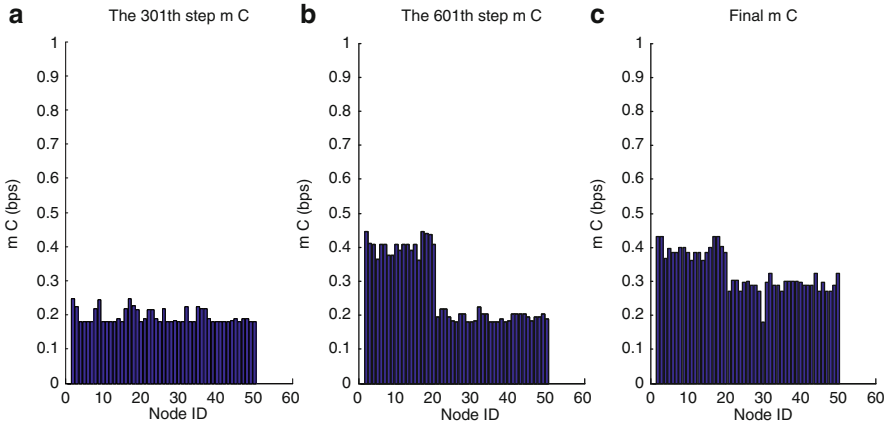


Fig. 10.32 **(a)** *m_C* of each node at the 301st time-step **(b)** *m_C* of each node at the 601st time-step **(c)** *m_C* of each node at the 1001st time-step

21–50 is *0.1946* (the designated value is 0.2×1 (sensing rate) = 0.2) and its standard deviation is *0.0131*. In Fig. 10.32c, the mean of nodes 2–20 is *0.3952* (the designated value is 0.2 × 2 (sensing rate) = 0.4) and its standard deviation is *0.0228*. The mean of nodes 21–50 is *0.2886* (the designated value is 0.2 × 1.5 (sensing rate) = 0.3) and its standard deviation is *0.0255*.

10.6 Conclusion

In this chapter, a distributed moving algorithm for MWSN is proposed. The ultimate goal of the redeployment algorithm is to reallocate the channel capacity of each link between two communicating nodes. The proposed algorithm can arbitrarily change the channel capacity to any node in the MWSN as long as the goal capacity is less than the maximal achievable capacity. By using the algorithm, the network can perform the monitoring task more flexibly and actively.

In addition to the initial configuration such as initial sensing rate and designated capacity, the algorithm could be implemented at each node by itself and without any centralized command from any node or the base station. The advantage of the distributed feature is that the task can be performed normally even if some nodes are broken or failed, and the delay or the consumption according to the centralized command transmission can be avoided. Extensive simulation studies at the MATLAB platform are discussed, and the performance of this algorithm is demonstrated.

Currently, the redeployment algorithm works only for the case when the routing paths are complete. If any inner node is breakdown or the link of two communicating nodes is broken, the algorithm cannot perform successfully. The process of reestablishing the routing path is an important issue for this algorithm. This issue should be further studied in the future. In addition, the power consumption is also an important issue. For example, finding a way to utilize the remaining power of a mobile node with respect to the requirement of communication capacity is also an important research topic.

Acknowledgments This work was supported in part by the National Science Council, Taiwan, ROC, under the grants: NSC98-2221-E-002-160-MY3, NSC99-2623-E-002-007-D, NSC98-2218-E-002-008, NSC95-2221-E-002-303-MY3, and Ministry of Economic Affairs and Industrial Technology Research Institute, and Automotive Research & Testing Center, Taiwan, ROC.

References

1. D.O. Popa, C. Helm, H.E. Stephanou, A.C. Sanderson, Robotic Deployment of Sensor Networks Using Potential Fields, Proceedings of IEEE International Conference on Robotics & Automation, New Orleans LA, pp. 642–647, April 2004
2. T.S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall PTR, 1996
3. N. Heo, P.K. Varshney, A distributed self spreading algorithm for mobile wireless sensor networks, Proceedings of IEEE International Conference on Wireless Communications and Networking, 3, 1597–1602, March 2003
4. M.F. Mysorewala, D.O. Popa, V. Giordano, F.L. Lewis, Deployment Algorithms and Indoor Experimental Vehicles for Studying Mobile Wireless Sensor Network, in Proceedings of the 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Las Vegas, Nevada, USA, pp. 290–298, Jun. 19–20, 2006
5. D.O. Popa, K. Sreenath, F.L. Lewis, Robotic Deployment for Environmental Sampling Applications, Proceedings of the IEEE International Conference on Control & Automation, Budapest, Hungary, pp. 197–202, June 2005

6. A. Savvides, H. Park, M.B. Srivastava, The Bits and Flops of the N-hop Multilateration Primitive For node Localization Problems, in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 112–121, Sep. 28, 2002
7. N. Patwari, A. Hero, M. Perkins, N. Correal, R. O'Dea, Relative Location Estimation in Wireless Sensor Network, IEEE Transaction on Signal Processing, 51(8), 2137–2148, 2003
8. R Ramanathan, R Rosales-Hain, Topology Control of Multihop Wireless Networks using Transmit Power Adjustment IEEE INFOCOM, 2, 404–412, 2000
9. A Jiang, J Bruck, Monotone Percolation and The Topology Control of Wireless Networks INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 327–338, 1, March, 2005
10. P Kulsirmongkol, C Saivichit, Topology control based on energy-efficient concept of mobile ad hoc networks for multimedia communication Intelligent signal processing and communication system 17–20 2005
11. C-C Shen, Z Huang, CLTC: A Cluster-Based Topology Control Framework for Ad Hoc Networks IEEE Transactions on Mobile Computing, 3(1), 2004
12. Z.J. Haas, A new routing protocol for the reconfigurable wireless networks, Proceedings of IEEE International Conference on Universal Personal Computing, 2, 562–566, Oct. 1997

Chapter 11
Adaptive IEEE 802.15.4 Medium Access Control Protocol for Control and Monitoring Applications

Pangun Park, Carlo Fischione, and Karl Henrik Johansson

Abstract The IEEE 802.15.4 standard for wireless sensor networks (WSNs) can support energy efficient, reliable, and timely packet transmission by tuning the medium access control (MAC) parameters *macMinBE*, *macMaxCSMABackoffs*, and *macMaxFrameRetries*. Such a tuning is difficult, because simple and accurate models of the influence of these parameters on the probability of successful packet transmission, packet delay, and energy consumption are not available. Moreover, it is not clear how to adapt the parameters to the changes of the network and traffic regimes by algorithms that can run on resource-constrained nodes. In this chapter, a generalized Markov chain is proposed to model these relations by simple expressions without giving up the accuracy. In contrast to previous work, the presence of limited number of retransmissions, acknowledgments, unsaturated traffic, and packet size is accounted for. The model is then used to derive an adaptive algorithm for minimizing the power consumption while guaranteeing reliability and delay constraints in the packet transmission. The algorithm does not require any modification of the IEEE 802.15.4 standard and can be easily implemented on network nodes. Numerical results show that the analysis is accurate and that the proposed algorithm satisfies reliability and delay constraints, and ensures a longer lifetime of the network under both stationary and transient network conditions.

Keywords IEEE 802.15.4 · Wireless sensor network · Markov chain model · Optimization

11.1 Introduction

The IEEE 802.15.4 standard has received considerable attention as a low data rate and low power protocol for wireless sensor network (WSN) applications in industry, control, home automation, health care, and smart grids, e.g., [1–4]. Many of these

P. Park (✉)
ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden
e-mail: pgpark@ee.kth.se.

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_11,

applications require that packets[1] are received with a given probability of success. In addition to such a reliability constraint, other applications ask for timely packet delivery [5]. It is known that IEEE 802.15.4 may have poor performance in terms of power consumption, reliability and delay [6], unless the MAC parameters are properly selected. It follows that (a) it is essential to characterize the protocol performance limitations, and (b) to develop methods to tune the IEEE 802.15.4 MAC parameters to enhance the network lifetime and improve the quality of the service experienced by the applications running on top of the network.

This book chapter focuses on the modelling and optimization of the performance metrics (reliability, delay, power consumption) for IEEE 802.15.4 WSNs. This problem is specially appealing for many control and industrial applications [2, 7]. We show that existing analytical studies of IEEE 802.15.4 MAC are not adequate to capture the real-world protocol behavior, when there are retry limits to send packets, acknowledgements (ACKs), and unsaturated traffic. We derive and use a new model to pose an optimization problem, where the objective function is the power consumption of the nodes, and the constraints are the reliability and delay of the packet delivery. Our aim is the design of distributed and adaptive algorithms that are simple to implement on sensor nodes, but still flexible, scalable, and able to provide high quality of service for WSN applications.

The remainder of this chapter is as follows. Section 11.2 presents the overview of IEEE 802.15.4 standard. In Sect. 11.3, we summarize existing work on analytical modelling and adaptive tuning of the IEEE 802.15.4 MAC protocol. Section 11.4 presents the problem studied in the chapter. In Sect. 11.5, we propose a generalized Markov chain model of the carrier sense multiple access with collision avoidance (CSMA/CA) with retry limits and unsaturated traffic regime. In Sect. 11.6, the optimization problem to adapt the MAC parameters is investigated. In addition, practical issues on how to implement the algorithm on sensors are also discussed. Numerical results achieved during stationary and transitionary conditions are reported in Sect. 11.7. Finally, Sect. 11.8 concludes the chapter. An extended version of this chapter with further details is available as [8].

11.2 Overview of the IEEE 802.15.4

In this section, we give an overview of the key points of IEEE 802.15.4. The IEEE 802.15.4 standard specifies the physical layer and the MAC sublayer for low-rate Wireless Personal Area Networks (WPANs). The standard defines two channel access modalities: the beacon-enabled modality, which uses a slotted CSMA/CA and the optional Guaranteed Time Slot (GTS) allocation mechanism, and a simpler unslotted CSMA/CA without beacons. In the following, we focus on the

[1] Throughout this paper, we refer to packets as medium access control (MAC) protocol data units, or MAC frames.

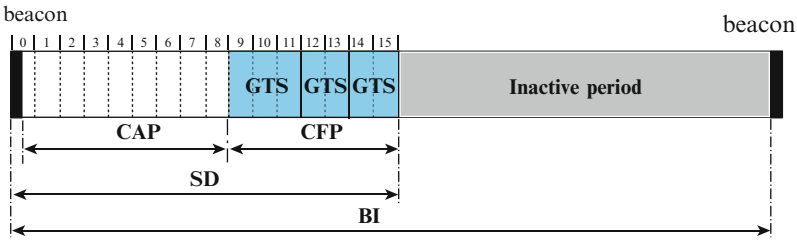


Fig. 11.1 Superframe structure of IEEE 802.15.4

beacon-enabled modality. Figure 11.1 shows a superframe structure of the beacon-enabled mode. The coordinator periodically sends the beacon frames in every beacon interval (BI) to identify its personal area networks and to synchronize devices that communicate with it. The coordinator and devices can communicate during active period, called the superframe duration (SD), and enter the low-power mode during the inactive period. The structure of the superframe is defined by two parameters, the beacon order (BO) and the superframe order (SO), which determine the length of the superframe and its active period, respectively. The length of the superframe and the length of its active period are then defined as

$$\mathrm{BI} = aBaseSuperframeDuration \times 2^{\mathrm{BO}}, \tag{11.1}$$

$$\mathrm{SD} = aBaseSuperframeDuration \times 2^{\mathrm{SO}}, \tag{11.2}$$

where $0 \leq \mathrm{SO} \leq \mathrm{BO} \leq 14$ and *aBaseSuperframeDuration* is the number of symbols forming a superframe when SO is equal to 0. In addition, the superframe is divided into 16 equally sized superframe slots of length *aBaseSlotDuration*. Each active period can be further divided into a Contention Access Period (CAP) and an optional Contention Free Period (CFP), composed of GTSs. A slotted CSMA/CA mechanism is used to access the channel of nontime critical data frames and GTS requests during the CAP. In the CFP, the dedicated bandwidth is used for time critical data frames. In the following section, we describe the data transmission mechanism for both CAP and CFP.

11.2.1 CSMA/CA Mechanism of CAP

Consider a node trying to transmit a data packet during CAP. In slotted CSMA/CA of IEEE 802.15.4, first the MAC sublayer initializes four variables, i.e., the number of backoffs ($\mathrm{NB} = 0$), contention window ($\mathrm{CW} = 2$), backoff exponent ($\mathrm{BE} = macMinBE$), and retransmission times ($\mathrm{RT} = 0$). Then the MAC sublayer delays for a random number of complete backoff periods in the range $[0, 2^{\mathrm{BE}} - 1]$ units. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer shall pause the backoff countdown

at the end of the CAP and resume it at the start of the CAP in the next superframe. Otherwise, the MAC sublayer counts its backoff timer. When the backoff period is zero, the node needs to perform the first clear channel assessment (CCA). The MAC sublayer proceeds if the remaining CSMA/CA algorithm steps (i.e., two CCAs), the frame transmission, and any acknowledgment can be completed before the end of the CAP. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and apply a further random backoff delay in the range $[0, 2^{\mathrm{BE}}-1]$ units before evaluating whether it can proceed again. Otherwise the MAC sublayer proceeds the CCA in the current superframe. If two consecutive CCAs are idle, then the node commences the packet transmission. If either of the CCA fails due to busy channel, the MAC sublayer increases the value of both NB and BE by one, up to a maximum value *macMaxCSMABackoffs* and *macMaxBE*, respectively. Hence, the values of NB and BE depend on the number of CCA failures of a packet. Once BE reaches *macMaxBE*, it remains at the value *macMaxBE* until it is reset. If NB exceeds *macMaxCSMABackoffs*, then the packet is discarded due to channel access failure. Otherwise, the CSMA/CA algorithm generates a random number of complete backoff periods and repeats the process. Here, the variable *macMaxCSMABackoffs* represents the maximum number of times the CSMA/CA algorithm is required to backoff. If channel access is successful, the node starts transmitting packets and waits for ACK. The reception of the corresponding ACK is interpreted as successful packet transmission. If the node fails to receive ACK due to collision or ACK timeout, the variable RT is increased by one up to *macMaxFrameRetries*. If RT is less than *macMaxFrameRetries*, the MAC sublayer initializes two variables CW=0, BE=*macMinBE* and follows the CSMA/CA mechanism to re-access the channel. Otherwise, the packet is discarded due to the retry limit. Note that the default MAC parameters are $macMinBE = 3$, $macMaxBE = 5$, $macMaxCSMABackoffs = 4$, $macMaxFrameRetries = 3$. See [1] for further details.

11.2.2 GTS Allocation of CFP

The coordinator is responsible for the GTS allocation and determines the length of the CFP in a superframe. To request the allocation of a new GTS, the device sends the GTS request command to the coordinator. The coordinator confirms its receipt by sending an acknowledgment frame within CAP. Upon receiving a GTS allocation request, the coordinator checks whether there are sufficient resources and, if possible, allocates the requested GTS. The GTS capacity in a superframe satisfies the following requirements:

1. The maximum number of GTSs to be allocated to devices is seven, provided there is sufficient capacity in the superframe.
2. The minimum length of a CAP is *aMinCAPLength*.

Therefore, the CFP length depends on the GTS requests and the currently available capacity in the superframe.

If there is sufficient bandwidth in the next superframe, the coordinator determines a device list for GTS allocation, based on a first-come-first-served (FCFS) policy. Then the coordinator includes the GTS descriptor, which is the device list that obtains GTSs in the following beacon to announce the allocation information. The coordinator makes this decision within *aGTSDescPersistenceTime* superframes. Note that on receipt of the acknowledgment to the GTS request command, the device continues to track beacons and wait for at most *aGTSDescPersistenceTime* superframes. A device uses the dedicated bandwidth to transmit the packet within the CFP. In addition, a transmitting device ensures that its transaction is complete one interframe spacing (IFS) period before the end of its GTS. See [1] for further details.

11.3 Related Work

We first discuss the literature concerning the analysis of IEEE 802.15.4, then we review previous work about adaptive MAC mechanisms for these protocols.

11.3.1 Analytical Model of MAC

In this section, we first discuss the literature concerning the analysis of CAP in IEEE 802.15.4, then we review previous work about the GTS allocation of CFP.

11.3.1.1 Analytical model of CAP

The modelling of CAP in IEEE 802.15.4 is related to IEEE 802.11 [9]. Both IEEE 802.11 and 802.15.4 are based on a MAC that uses a binary exponential backoff scheme. Bianchi's model describes the basic functionalities of the IEEE 802.11 through a Markov chain under saturated traffic and ideal channel conditions [10]. Extensions of this model have been used to analyze the packet reception rate [11], the delay [12], the MAC layer service time [13], and throughput [14] of IEEE 802.11.

The analysis of the packet delay, throughput, and power consumption of IEEE 802.15.4 WSNs has been the focus of several simulation-based studies, e.g., [15,16], and some more recent analytical works, e.g., [6]–[21]. Inspired by Bianchi's work, a Markov model for IEEE 802.15.4 and an extension with ACK mechanism have been proposed in [6] and [17]. A modified Markov model including retransmissions with finite retry limits has been studied in [18] as an attempt to model the CSMA/CA mechanism. However, the analysis gives inaccurate results because the power consumption and throughput expressions under unsaturated traffic with finite retry limits show a weak matching with simulation results. In [22], the authors consider finite retry limits with saturated traffic condition which is not a realistic scenario for WSN applications.

While the aforementioned works do not consider the active and inactive periods of the IEEE 802.15.4 nodes, this aspect is investigated in [19]–[21]. In [19], the authors assume unsaturated traffic and source nodes with a finite buffer, but only uplink communications. In [20], instead, downlink communications are also taken into account, but nodes are assumed to be equipped with infinite size buffers. Furthermore, in [20], the power consumption, reliability, and delay performance are not investigated. In [21], a throughput analysis has been performed by an extension of the Markov chain model proposed in [20]. The superframe structure, ACK, and retransmissions are considered. It is shown that the models in [19, 20], although very detailed, fail to couple with realistic simulation results. However, the proposed Markov chain does not model the length of data and ACK packets, which is crucial to analyze the performance metrics for IEEE 802.15.4 networks with low data rate. In [23], the authors approximate the CAP as the simple nonpersistent CSMA, which is a similar approach to [24]. However, the authors assume that the entire superframe duration is active, that is, SO = BO without considering the inactive period.

The multihop network scenario is considered in [25]–[27]. In [25], the authors extended the framework proposed in [23] to 2-hop network scenarios, where sensors communicate with the coordinator through an intermediate relay node, which simply forwards the data packets received from the sources. In [26, 27], the authors propose the use of a relay for interconnecting two IEEE 802.15.4 clusters and analyze the performance using a queuing theory.

11.3.1.2 Analytical Model of CFP

Most of the literature does not consider satisfactorily the CFP, where the GTS mechanism operates. Simulation studies in [28]–[30] consider the CAP and CFP. An interesting theoretical performance evaluation of the GTS allocation has been proposed by Koubaa et al. [31]–[33] by using network calculus. These papers focus on the impact of the IEEE 802.15.4 standard parameters (SO, BO), the delay, throughput and energy consumption of GTS allocation. In [34], a round-robin scheduler is proposed to improve the bandwidth utilization based on a network calculus approach. Network calculus, however, assumes a continuous flow model (whereas communication happens through low data rate packets in reality) and it analyzes the worst-case of traffic flows (which leads to severe under-utilization of time slots in actual environments). Consequently, the difference between the network flow model of the network calculus approach and the actual behavior may be quite large. In [35], the authors analyze the stability, delay, and throughput of the GTS allocation mechanism based on the Markov chain model.

Some interesting algorithms have been proposed to improve the performance of GTS allocation mechanism. To maximize the bandwidth utilization, a smaller slot size and an offline message scheduling algorithm are proposed in [36] and [37], respectively. In [38], the delay constraint and bandwidth utilization are considered to design a GTS scheduling algorithm. Huang [39] proposes an adaptive GTS allocation scheme by considering low delay and fairness.

11.3.2 Adaptive Tuning of MAC

Several algorithms to tune the MAC of IEEE 802.11 and IEEE 802.15.4 protocols have been proposed. The algorithms can be grouped in those based on the use of physical layer measurements, and those based on the use of link-layer information.

An adaptive tuning based on physical layer measurements has been investigated in [24, 40, 41], where a p-persistent IEEE 802.11 protocol has been considered to optimize the average backoff window size. The channel access probability p that maximizes the throughput or minimize the power consumption is derived. This algorithm and its scalability to the network size have been studied also for IEEE 802.15.4 [40]. However, that study was less successful, because the channel sensing mechanism, the optional ACK, and retransmission mechanisms are hard to be approximated by a p-persistent MAC. Furthermore, in [40] and [41] a saturated traffic regime is assumed, which is a scenario of little interest for typical WSN applications.

Link-based optimizations for IEEE 802.11 and 802.15.4 have been investigated in [42]–[46], where simple window adjustment mechanisms that are based on ACK transmissions have been considered. In these papers, the algorithms adapt the contention window size depending on the successful packet transmission, packet collision and channel sensing state, but the algorithms are not grounded on an analytical study. In [42], different backoff algorithms are presented to improve the channel throughput and the fairness of channel usage for IEEE 802.11. A fair backoff algorithm is studied also in [43] and [44]. A link-based algorithm of the IEEE 802.15.4 random backoff mechanism to maximize the throughput has been presented in [45]. In [46], a dynamic tuning algorithm of the contention window size is evaluated on goodput, reliability, and average delay.

An IEEE 802.15.4 enhancement based on the use of link-layer information has some drawback. First, it requires a modification of the standard. Then, although link-based mechanisms are simple to implement, the ACK mechanism may be costly since it introduces large overhead for small packets. For instance, alarm messages in an industrial application are a single byte, whereas the ACK has a size of 11 bytes. In addition, the ACK mechanism requires extra waiting time. Moreover, link-based algorithms adapt the MAC parameters for each received ACK, which mean a slow and inefficient adaptation to network, traffic, and channel variations.

11.4 Problem Formulation

We consider a star network with a coordinator, and N nodes transmitting to the coordinator. These nodes use the beacon-enabled slotted CSMA/CA and ACK. The important parameters of the CSMA/CA algorithm for our study are the minimum value of the backoff exponent *macMinBE*, the maximum number of backoffs *macMaxCSMABackoffs* and the maximum number of retries *macMaxFrameRetries* that each node can select. See details of IEEE 802.15.4 in [1] and [47].

In this book chapter, we propose a novel modelling and adaptive tuning of IEEE 802.15.4 for reliable and timely communication while minimizing the energy consumption. The protocol is adjusted dynamically by a constrained optimization problem that each node of the network solves. The objective function, denoted by $\widetilde{E}_{\text{tot}}$, is the total energy consumption for transmitting and receiving packets of a node. The constraints are given by the probability of successful packet delivery (reliability) and average delay. The constrained optimization problem for a generic transmitting node in the network is

$$\min_{\mathbf{V}} \quad \widetilde{E}_{\text{tot}}(\mathbf{V}) \tag{11.3a}$$

$$\text{s.t.} \quad \widetilde{R}(\mathbf{V}) \geq R_{\min}\,, \tag{11.3b}$$

$$\widetilde{D}(\mathbf{V}) \leq D_{\max}\,, \tag{11.3c}$$

$$\mathbf{V}_0 \leq \mathbf{V} \leq \mathbf{V}_m\,. \tag{11.3d}$$

The decision variables of the node $\mathbf{V} = (m_0, m, n)$ are

$$\begin{aligned} m_0 &\triangleq \mathit{macMinBE}\,,\\ m &\triangleq \mathit{macMaxCSMABackoffs}\,,\\ n &\triangleq \mathit{macMaxFrameRetries}\,. \end{aligned}$$

$\widetilde{R}(\mathbf{V})$ is the reliability, and $R_{\min}$ is the minimum desired probability for successful packet delivery. $\widetilde{D}(\mathbf{V})$ is the average delay for a successfully received packet, and $D_{\max}$ is the desired maximum average delay. The constraint $\mathbf{V}_0 \leq \mathbf{V} \leq \mathbf{V}_m$ captures the limited range of the MAC parameters. In the problem, we used the symbol $\widetilde{\ }$ to evidence that the energy, reliability, and delay expression are approximations. The corresponding variables without $\widetilde{\ }$ denote the true values. We will show later that we use approximations of high accuracy and reduced computational complexity so that nodes can solve the problem.

Main contributions of this chapter are the following: (a) the modelling of the relation between the MAC parameters of IEEE 802.15.4 and the selected performance metrics, (b) the derivation of simple relations to characterize the operations of the MAC by computationally affordable algorithms, (c) formulation and solution of a novel optimization problem for the MAC parameters, (d) discussion on a practical implementation of the optimization by an adaptive algorithm and (e) performance evaluations of the algorithm by simulation of both stationary and transient network conditions.

Unlike previous work, we propose a generalized Markov model of the exponential backoff process including retry limits, ACKs, and unsaturated traffic regime. However, the numerical evaluation of these performance metrics asks, in general, for heavy computations. This is a drawback when using them to optimize the IEEE 802.15.4 MAC parameters by in-network processing [48] because a complex computation is out of reach for resource-limited sensing devices. Therefore, we devise a simplified and effective method that reduces drastically the computational complexity while ensuring a satisfactory accuracy.

Based on our novel modelling, we propose an adaptive tuning of MAC parameters that uses the physical layer measurement of channel sensing. This adaptive IEEE 802.15.4 MAC protocol is furnished with two distinctive features: it does not require any modification of the existing standard, and it makes an optimization of all the MAC parameters of IEEE 802.15.4. Specifically, in contrast to link-based adaptation [42]–[46], our algorithm does not require ACK mechanism or request to send/clear to send (RTS/CTS) handshakes (or related standard modifications). In contrast to [24]–[41], we do not use the (inaccurate) p-persistent approximation and the modification of the standard therein proposed, and we do not require any hardware modification to make an estimate of the signal-to-noise ratio. Our adaptive tuning optimizes the considered MAC parameters, all at once, and not only some of them, as proposed in [24]–[46].

The proposed adaptive IEEE 802.15.4 MAC improves the power efficiency substantially while guaranteeing reliability and delay constraints. The adaptation is achieved by distributed asynchronous iterations that only require channel condition information, the number of nodes of the network, and the traffic load. We show that the convergence is fast and robust to errors in the estimation of the channel condition, number of nodes, and traffic load. A good fairness is also achieved.

11.5 Analytical Modelling of IEEE 802.15.4

In a star network, all N nodes contend to send data to the PAN coordinator, which is the data sink. We assume no hidden node. Throughout this paper, we consider applications where nodes asynchronously generate packets with probability $1-q$, when a node sends a packet successfully, discards a packet, or the sampling interval expires. Otherwise, a node stays for $L_0 S_b$ s without generating packets with probability q, where L_0 is an integer and S_b is the time unit *aUnitBackoffPeriod* (corresponding to 20 symbols). The data packet transmission is successful if an ACK packet is received. In addition, we assume that the channel is sensed busy or idle without errors.

In such a scenario, we propose an effective analytical model of the slotted CSMA/CA by a Markov chain. The chain enables us to derive the objective function, energy (11.3a), and constraints on reliability (11.3b) and delay (11.3c) of the optimization problem.

11.5.1 Markov Chain Model

In this section, we develop a generalized Markov chain model of the slotted CSMA/CA mechanism of beacon-enabled IEEE 802.15.4. Compared to previous results, e.g., [6, 17]–[21], the novelty of this chain consists in the modelling of the retry limits for each packet transmission, ACK, the inclusion of unsaturated traffic regimes, and packet size.

Let $s(t), c(t)$ and $r(t)$ be the stochastic processes representing the backoff stage, the state of the backoff counter and the state of retransmission counter at time $t = 0, 1, \ldots \infty$ experienced by a node to transmit a packet. By assuming independent probability that nodes start sensing, the stationary probability τ that a node attempts a first carrier sensing in a randomly chosen slot time is constant and independent of other nodes. The triple $(s(t), c(t), r(t))$ defines the three-dimensional Markov chain in Fig. 11.2, where we use (i, k, j) to denote its state. We denote the MAC parameters by $\mathbf{V} = (m_0, m, n), m_b \triangleq macMaxBE, W_0 \triangleq 2^{m_0}, W_m \triangleq 2^{\min(m_0+m, m_b)}$.

The Markov chain consists of four main parts corresponding to the idle-queue states, backoff states, CCA states, and packet transmission states. The states

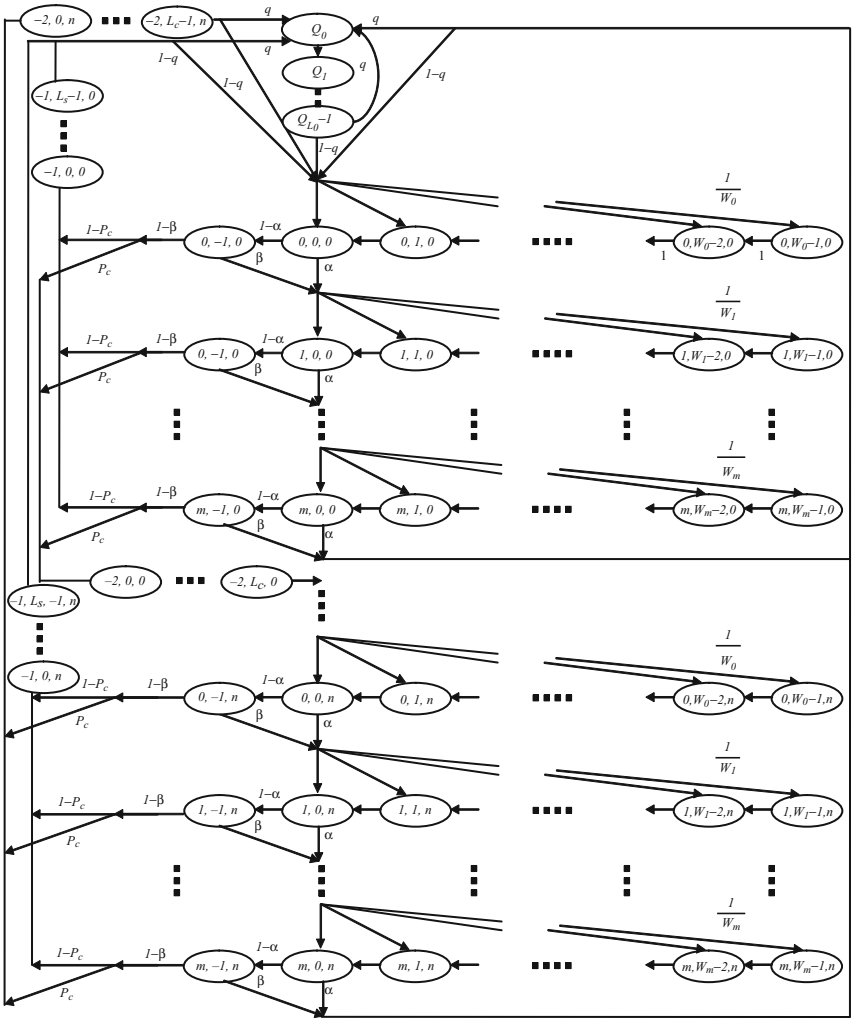


Fig. 11.2 Markov chain model for CSMA/CA of IEEE 802.15.4

$(Q_0, \ldots, Q_{L_0-1})$ in the top of Fig. 11.2 correspond to the idle-queue states when the packet queue is empty and the node is waiting for the next packet generation time. Note that these states take into account the sampling interval. The states $(i, W_m - 1, j), \ldots, (i, W_0 - 1, j)$ represent the backoff states. The states $(i, 0, j)$ and $(i, -1, j)$ represent first CCA (CCA_1) and second CCA (CCA_2), respectively. Let α be the probability that CCA_1 is busy, and β the probability that CCA_2 is busy. The states $(-1, k, j)$ and $(-2, k, j)$ correspond to successful transmission and packet collision, respectively. By knowing the duration of an ACK frame, ACK timeout, IFS, data packet length, and header duration, we define the time duration of a successful packet transmission L_s and the corresponding time for packet collision L_c as

$$\begin{aligned} L_s &= L + t_{\text{ack}} + L_{\text{ack}} + \text{IFS}\,, \\ L_c &= L + t_{\text{m,ack}}, \end{aligned} \tag{11.4}$$

where L is the total length of a packet including overhead and payload, t_{ack} is ACK waiting time, L_{ack} is the length of the ACK frame, and $t_{\text{m,ack}}$ is the timeout length of the ACK, see details in [1].

We have the following result on how to compute the stationary probability of the Markov chain:

Lemma 1. *Let the stationary probability of the Markov chain in Fig. 11.2 be denoted*

$$b_{i,k,j} = \lim_{t\to\infty} P(s(t) = i, c(t) = k, r(t) = j),$$

where $i \in (-2, m), k \in (-1, \max(W_i - 1, L_s - 1, L_c - 1))$, $j \in (0, n)$. *Then, for* $0 \le i \le m$

$$b_{i,k,j} = \frac{W_i - k}{W_i} b_{i,0,j}\,, \quad 0 \le k \le W_i - 1\,, \tag{11.5}$$

where

$$W_i = \begin{cases} 2^i W_0, & 0 \le i \le m_b - m_0\,, \\ 2^{m_b}, & \textit{otherwise}\,, \end{cases}$$

and

$$b_{i,0,j} = \left[(1-\alpha)(1-\beta) P_c \sum_{i=0}^{m} (\alpha + (1-\alpha)\beta)^i \right]^j (\alpha + (1-\alpha)\beta)^i b_{0,0,0}\,, \tag{11.6}$$

with

$$b_{0,0,0} = \begin{cases} \Big[\frac{1}{2}\Big(\frac{1-(2x)^{m+1}}{1-2x}W_0 + \frac{1-x^{m+1}}{1-x}\Big)\frac{1-y^{n+1}}{1-y} + (1-\alpha)\frac{1-x^{m+1}}{1-x}\frac{1-y^{n+1}}{1-y} \\ +(L_{\mathrm{s}}(1-P_{\mathrm{c}}) + L_{\mathrm{c}}P_{\mathrm{c}})(1-x^{m+1})\frac{1-y^{n+1}}{1-y} + L_0\frac{q}{1-q}\Big(\frac{x^{m+1}(1-y^{n+1})}{1-y} \\ +P_{\mathrm{c}}(1-x^{m+1})y^n + (1-P_{\mathrm{c}})\frac{(1-x^{m+1})(1-y^{n+1})}{1-y}\Big)\Big]^{-1}, \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if} \quad m \le m_b - m_0\,, \\ \Big[\frac{1}{2}\Big(\frac{1-(2x)^{m_b-m_0+1}}{1-2x}W_0 + \frac{1-x^{m_b-m_0+1}}{1-x} + (2^{m_b}+1)x^{m_b-m_0+1}\Big) \\ \times\frac{(1-x^{m-m_b+m_0})(1-y^{n+1})}{(1-x)(1-y)} + (1-\alpha)\frac{(1-x^{m+1})(1-y^{n+1})}{(1-x)(1-y)} + (L_{\mathrm{s}}(1-P_{\mathrm{c}}) \\ +L_{\mathrm{c}}P_{\mathrm{c}})(1-x^{m+1})\frac{1-y^{n+1}}{1-y} + L_0\frac{q}{1-q}\Big(\frac{x^{m+1}(1-y^{n+1})}{1-y} + P_{\mathrm{c}}(1-x^{m+1}) \\ y^n + (1-P_{\mathrm{c}})\frac{(1-x^{m+1})(1-y^{n+1})}{1-y}\Big)\Big]^{-1}, \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise}\,, \end{cases} \tag{11.7}$$

where $x = \alpha + (1-\alpha)\beta$, $y = P_{\mathrm{c}}(1-x^{m+1})$, *and* P_c *is the collision probability. Moreover,*

$$b_{-1,k,j} = (1-P_{\mathrm{c}})(1-x)\sum_{i=1}^{m} b_{i,0,j}\,, \quad 0 \le k \le L_{\mathrm{s}}-1\,, \tag{11.8}$$

and

$$b_{-2,k,j} = P_{\mathrm{c}}(1-x)\sum_{i=1}^{m} b_{i,0,j}\,, \quad 0 \le k \le L_{\mathrm{c}}-1\,. \tag{11.9}$$

Proof. A proof is given in [8]. □

We remark here that the stationary probability $b_{0,0,0}$, which plays a key role in the analysis below, is different from the corresponding term in [6, 17]–[21] due to our more detailed modelling of the retransmissions, ACK, unsaturated traffic, and packet size. In the following section, we demonstrate the validity of the Markov chain model by Monte Carlo simulations.

Now, starting from Lemma 1, we derive the channel sensing probability τ and the busy channel probabilities α and β. The probability τ that a node attempts CCA_1 in a randomly chosen time slot is

$$\tau = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,0,j} = \frac{1-x^{m+1}}{1-x}\frac{1-y^{n+1}}{1-y}b_{0,0,0}. \tag{11.10}$$

This probability depends on the probability P_{c} that a transmitted packet encounters a collision, and the probabilities α and β. These probabilities are derived in the following.

The term P_c is the probability that at least one of the $N-1$ other nodes transmits in the same time slot. If all nodes transmit with probability τ, we have

$$P_c = 1 - (1-\tau)^{N-1}.$$

Similar to [6], we derive the busy channel probabilities α and β as follows. We have

$$\alpha = \alpha_1 + \alpha_2, \tag{11.11}$$

where α_1 is the probability of finding channel busy during CCA_1 due to data transmission by one of the other $N-1$ nodes, namely,

$$\alpha_1 = L(1-(1-\tau)^{N-1})(1-\alpha)(1-\beta),$$

and α_2 is the probability of finding the channel busy during CCA_1 due to ACK transmission:

$$\alpha_2 = L_{\text{ack}} \frac{N\tau(1-\tau)^{N-1}}{1-(1-\tau)^N}(1-(1-\tau)^{N-1})(1-\alpha)(1-\beta),$$

where L_{ack} is the length of the ACK. In a similar way, the probability of finding the channel busy during CCA_2 is

$$\beta = \frac{1-(1-\tau)^{N-1} + N\tau(1-\tau)^{N-1}}{2-(1-\tau)^N + N\tau(1-\tau)^{N-1}}. \tag{11.12}$$

Now, we are in the position to derive the carrier sensing probability τ and the busy channel probabilities α and β by solving the system of nonlinear equations (11.10)–(11.12), see details in [49]. From these probabilities, one can derive the expressions for the reliability, delay for successful packet delivery, and power consumption in (11.3). Unfortunately, there is no closed-form expression for these probabilities, but the system of (11.10)–(11.12) must be solved by numerical methods. This may be computationally demanding and therefore inadequate for use in embedded sensor devices. In the following, we therefore instead present a simplified analytical model of the reliability, delay for successful packet delivery, and power consumption. The key idea is that sensor nodes can estimate the busy channel probabilities α and β and the channel sensing probability τ. Therefore, nodes can exploit local measurements to evaluate the performance metrics, rather than solving complex nonlinear equations. Details follow in the sequel, where we derive these approximate expressions for (11.3a)–(11.3c).

11.5.2 Reliability

The main contributions of this section are a precise and approximated expression of the reliability (11.3b) of the optimization problem (11.3), where we recall the reliability is the probability of successful packet reception.

Proposition 1. *The reliability is*

$$R(\boldsymbol{V}) = 1 - \frac{x^{m+1}(1-y^{n+1})}{1-y} - y^{n+1}\,, \tag{11.13}$$

where x and y are given in Lemma 1.

Proof. A proof is given in [8].

Approximation 1. *An approximation of the reliability is*

$$\widetilde{R}(\boldsymbol{V}) = 1 - x^{m+1}(1+\widetilde{y}) - \widetilde{y}^{n+1} \tag{11.14}$$

where

$$\begin{aligned}\widetilde{y} =&(1-(1-(1+x)(1+\hat{y})\widetilde{b}_{0,0,0})^{N-1})(1-x^2)\,,\\ \widetilde{b}_{0,0,0} =&2[W_0(1+2x)(1+\hat{y}) + 2L_s(1-x^2)(1+\hat{y})\\ &+ L_0 q/(1-q)(1+\hat{y}^2+\hat{y}^{n+1})]^{-1}\,,\end{aligned}$$

and $\hat{y} = (1-(1-\tau)^{N-1})(1-x^2)$.

Proof. A proof is given in [8]. □

We remark that $\widetilde{R}(\mathbf{V})$ is a function of the measurable busy channel probabilities α and β, the channel access probability τ and the MAC parameters m_0, m_b, m, n. The approximation is based on estimated values of x and τ. Monte Carlo simulations in [8] validate the approximated model of the reliability.

11.5.3 Delay

In this section, we derive the constraint of the average delay (11.3c). The average delay for a successfully received packet is defined as the time interval from the instant the packet is at the head of its MAC queue and ready to be transmitted, until the transmission is successful and the ACK is received. In this section, we develop an approximation for such an average delay, which is given by Approximation 2. To this aim, we need some intermediate technical steps. In particular, we characterize (a) the expression of the delay for a successful transmission at time $j+1$ after jth events of unsuccessful transmission due to collision and (b) the expected value of the approximated backoff delay due to busy channel.

Let D_j be the random time associated with the successful transmission of a packet at the jth backoff stage. Denote with $\mathcal{A}_j$ the event of a successful transmission at time $j+1$ after jth events of unsuccessful transmission. Let $\mathcal{A}^{\text{tot}}$ be the event of successful transmission within the total attempts n. Then, the delay for a successful transmission after j th unsuccessful attempts is

$$D = \sum_{j=0}^{n} \mathbb{1}_{\mathcal{A}_j|\mathcal{A}^{\text{tot}}} D_j \,,$$

where $\mathbb{1}_{\mathcal{A}_j|\mathcal{A}^{\text{tot}}}$ is 1 if $\mathcal{A}_j|\mathcal{A}^{\text{tot}}$ holds, and 0 otherwise and $D_j = L_{\text{s}} + j\, L_{\text{c}} + \sum_{h=0}^{j} T_h$, with T_h being the backoff stage delay, L_{s} is the packet successful transmission time, and L_{c} is the packet collision transmission time as defined in (11.4).

Lemma 2. *The probability of successful transmission at time $j+1$ after jth events of unsuccessful transmission due to collision is*

$$\Pr(\mathcal{A}_j|\mathcal{A}^{\text{tot}}) = \frac{(1-y)\, y^j}{1-y^{n+1}} \,. \tag{11.15}$$

Proof. A proof is given in [8]. □

In the following, we derive the total backoff delay T_h. Let $T_{h,i}$ be the random time needed to obtain two successful CCAs from the selected backoff counter value at backoff level i. Recall that a node transmits the packet when the backoff counter is 0 and two successful CCAs are detected [1]. Denote with $\mathcal{B}_i$ the event occurring when the channel is busy for i times, and then idle at time $i+1$. Let $\mathcal{B}^{\text{tot}}$ be the event of having a successful sensing within the total number of m sensing attempts. If the node accesses an idle channel after its ith busy CCA, then

$$T_h = \sum_{i=0}^{m} \mathbb{1}_{\mathcal{B}_i|\mathcal{B}^{\text{tot}}} T_{h,i} \,,$$

where

$$T_{h,i} = 2\, T_{sc} + \sum_{k=1}^{i} T_{h,k}^{sc} + \sum_{k=0}^{i} T_{h,k}^{b} \,, \tag{11.16}$$

and $2T_{sc}$ is the successful sensing time, $\sum_{k=1}^{i} T_{h,k}^{sc}$ is the unsuccessful sensing time due to busy channel during CCA, and $\sum_{k=0}^{i} T_{h,k}^{b}$ is the backoff time. The expected value of the approximated backoff delay is

$$\mathbb{E}[\widetilde{T}_h] = 2S_b \left(1 + \frac{1}{4}\left(\frac{1-\gamma}{1-\gamma^{m+1}}\left(2W_0 \frac{1-(2\gamma)^{m+1}}{1-2\gamma} - \frac{3(m+1)\gamma^{m+1}}{1-\gamma}\right) + \frac{3\gamma}{1-\gamma} - (W_0+1)\right)\right) \,, \tag{11.17}$$

where $\gamma = \max(\alpha, (1-\alpha)\beta)$. A proof is given in [8].

Now, we are in the position to derive an approximation of the average delay for successfully received packets.

Approximation 2. *The expected value of the approximated delay is*

$$\widetilde{D}(\boldsymbol{V}) = T_s + \mathbb{E}[\widetilde{T}_h] + \left(\frac{y}{1-y} - \frac{(n+1)\,y^{n+1}}{1-y^{n+1}}\right)(T_c + \mathbb{E}[\widetilde{T}_h])\,. \tag{11.18}$$

Proof. A proof is given in [8]. □

Validation of the approximated model of the average packet delay is given in [8].

11.5.4 Power Consumption

Here, we derive the objective function, power consumption of the node (11.3a) of the optimization problem (11.3). We propose two models for the average power consumption, depending on the radio state during the backoff mechanism specified by the IEEE 802.15.4 standard. Let us denote by *I-mode* and *S-mode* the situation when the radio is set in idle mode or in sleep mode during backoff period, respectively.

Approximation 3. *The approximated energy consumption of the I-mode* $\widetilde{E}_{\text{tot,i}}(\boldsymbol{V})$ *is*

$$\begin{aligned}\widetilde{E}_{\text{tot,i}}(\boldsymbol{V}) =& \frac{P_{\text{i}}\tau}{2}\left[\frac{(1-x)(1-(2x)^{m+1})}{(1-2x)(1-x^{m+1})}W_0 - 1\right] + P_{\text{sc}}(2-\alpha)\tau + (1-\alpha)\\ &\times (1-\beta)\tau\,(P_{\text{t}}L + P_{\text{i}} + L_{\text{ack}}\,(P_{\text{r}}(1-P_{\text{c}}) + P_{\text{i}}P_{\text{c}})) + P_{\text{w}}q\\ &\times \left(x^{m+1}(1+y) + P_{\text{c}}(1-x^2)y^n + (1-P_{\text{c}})(1-x^2)(1+y)\right)\widetilde{b}_{0,0,0}\end{aligned} \tag{11.19}$$

and of the S-mode is

$$\begin{aligned}\widetilde{E}_{\text{tot,s}}(\boldsymbol{V}) =& P_{\text{sc}}(2-\alpha)\tau + (1-\alpha)(1-\beta)\tau\,(P_{\text{t}}L + P_{\text{i}} + L_{\text{ack}}\,(P_{\text{r}}(1-P_{\text{c}})\\ &+ P_{\text{i}}P_{\text{c}})) + P_{\text{w}}\left(\tau - \frac{\widetilde{b}_{0,0,0}(1-(0.5x)^{m+1})(1-y^{n+1})}{W_0(1-0.5x)(1-y)}\right)\end{aligned} \tag{11.20}$$

where

$$\widetilde{b}_{0,0,0} \approx \frac{2}{W_0 r_1 + 2r_2} \tag{11.21}$$

with

$$\begin{aligned} r_1 &= (1+2x)(1+\hat{y})\,,\\ r_2 &= L_{\text{s}}(1-x^2)(1+\hat{y}) + K_0(1+\hat{y}^2+\hat{y}^{n+1})\,,\\ \hat{y} &= (1-(1-\tau)^{N-1})(1-x^2)\,. \end{aligned}$$

P_i, P_{sc}, P_{sp}, P_w, P_t, and P_r are the average power consumption in idle-listen, channel sensing, sleep, wake-up, transmit, and receive states, respectively.

Proof. A proof is given in [8]. □

Monte Carlo simulations validate also this approximated model in [8].

11.6 IEEE 802.15.4 Optimization

In the previous sections, we developed the expressions of the performance metrics. Here, we present a novel approach where each node locally solves an optimization problem. Consider the reliability, delay, and power consumption as investigated in Sect. 11.5. The optimization problem (11.3) can be defined by using (11.14) of Approximation 1 for the reliability constraint, (11.18) of Approximation 2 for the delay constraint and (11.19) or (11.20) of Approximation 3 for the power consumption. Note that the power consumption is given by (11.19) if the *I-mode* is selected, and it is given by (11.20), if the *S-mode* is selected. The solution of the optimization problem gives the optimal MAC parameters (m_0, m, n) that each node uses to minimize its energy expenditure, subject to reliability and delay constraints. Notice that the problem is combinatorial because the decision variables take on discrete values.

A vector of decision variables $\mathbf{V}$ is feasible if the reliability and delay constraints are satisfied. The optimal solution may be obtained by checking every combination of the elements of $\mathbf{V}$ that gives feasibility, and then checking the combination that gives the minimum objective function. Clearly, this approach may have a high computational complexity, since there are $6 \times 4 \times 8 = 192$ combinations of MAC parameters to check [1]. Therefore, in the following we propose an algorithm that gives the optimal solution by checking a reduced number of combinations.

We remark here that the reliability and power consumption of both *I-mode* and *S-mode* are increasing functions of n, see details in [8]. This property is quite useful to solve (11.3) by a simple algorithm with reduced computational complexity, as we see next.

The search of optimal MAC parameters uses an iterative procedure according to the component-based method [50]. In particular, the probabilities α, β, and τ are estimated periodically by each node. If a node detects a change of these probabilities, then the node solves the local optimization problem (11.3) using these new estimated values. The solution is achieved by finding the value of n that minimizes the energy consumption given a pair of values for m_0 and m. Since the power consumption is increasing with n, it follows that the minimum is attained at the lowest value of n that satisfies the constraints. Given that the reliability is increasing with n, simple algebraic passages give that such a value is $n = f(m_0, m)$, with

$$f(m_0, m) = \left\lceil \frac{\ln(1 - x^{m+1}(1 + \widetilde{y}) - R_{\min})}{\ln(\widetilde{y})} - 1 \right\rceil, \tag{11.22}$$

where $\widetilde{y} = (1 - (1 - \widetilde{\tau})^{N-1})(1 - x^2)$ and

$$\widetilde{\tau} = \frac{2r_3}{2^{m_0} r_1 + 2r_2},$$

with

$$\begin{aligned} r_1 &= (1 + 2x)(1 + \hat{y}), \\ r_2 &= L_s(1 - x^2)(1 + \hat{y}) + \frac{L_0 q(1 + \hat{y}^2 + \hat{y}^{n+1})}{1 - q}, \\ r_3 &= (1 + x)(1 + \hat{y}), \end{aligned}$$

and $\hat{y} = (1-(1-\tau)^{N-1})(1-x^2)$. Equation (11.22) returns the optimal retry limits given a pair m_0, m. Notice that x and $\hat{y}$ are measurable since the node estimates α, β, and τ. By using this simple algorithm, a node checks just $6 \times 4 = 24$ combinations of the MAC parameters m_0, m instead of $6 \times 4 \times 8 = 192$ combinations, which would be required by an exhaustive search.

We have seen by the Approximations 1, 2 and 3 that the performance metrics are a function of the busy channel probabilities α and β and the channel access probability τ. Once these probabilities are known at a node, the optimal MAC parameters of that node can be readily computed by the proposed algorithm. In the algorithm, the number of nodes and packet generation rates are assumed to be known, whereas the busy channel probability and channel access probability are periodically estimated during the sensing states of the MAC layer, and they do not require an ACK mechanism. The robustness of the algorithm to possible errors in the estimation of the number of nodes and traffic load is investigated in Sect. 11.7.3.

The average busy channel probabilities α and β are estimated at each node while sending a data packet to the coordinator. These probabilities are initialized at the beginning of the node's operation. The estimations of the busy channel probabilities and the channel access probability use a first-order filter. When the node senses the channel at CCA_1 or CCA_2, these probabilities are updated using the following recursions

$$\begin{aligned} \alpha_{k+1} &= \delta_b \alpha_k + (1 - \delta_b)\hat{\alpha}_k, \\ \beta_{k+1} &= \delta_b \beta_k + (1 - \delta_b)\hat{\beta}_k, \end{aligned}$$

where k denotes the update step for some $\delta_b \in (0, 1)$, respectively. Note that $\hat{\alpha}$ and $\hat{\beta}$ are the busy channel probability measurements of CCA_1 and CCA_2, respectively. Each node only counts the number of busy channel events during CCA_1 and CCA_2 state to obtain $\hat{\alpha}$ and $\hat{\beta}$, respectively. Therefore, a node does not require any extra communication and sensing state to estimate these probabilities compared to the IEEE 802.15.4 standard. By contrast, the estimation algorithms for IEEE 802.11 proposed in [24] and [51] are not energy efficient since a node needs to sense the channel state during the backoff stage. This allows one to estimate the average

length of the idle period. Hence, these schemes are implementable only in *I-mode*. By contrast, our scheme is applied in both *I-mode* and *S-mode* and does not require any computation load during the backoff stage. An analysis of the impact of parameter estimation errors is investigated in Sect. 11.7.3.

During an initialization phase of the algorithm, each node uses the initial MAC parameters $m_0 = 3, m_b = 8, m = 4, n = 3$. The busy channel probabilities α and β and the channel access probability τ are estimated during the channel sensing state of IEEE 802.15.4 without any extra states. The application requirements are communicated by the coordinator to the node if there are changes. It is also possible that each node makes a decision of application requirements depending on the data type, such as strict delay requirements for alarm messages.

11.7 Numerical Results

In the following, we present Monte Carlo simulations to analyze the performance of our adaptive tuning algorithm of the MAC parameters, under both stationary and transient conditions. The analytical modelling that we have proposed in Sect. 11.5 is based on a Markov chain that has been validated experimentally in [52]. The Monte Carlo simulations that we use here are representative of the real-world behavior of the network.

In the stationary conditions, the application requirements and network scenario are constant, whereas in transient condition there are variations. The simulations are based on the specifications of the IEEE 802.15.4 and the practical implementation aspects described in Sect. 11.6. In the simulations, the network considers the *I-mode* and *S-mode* of the node to compare the performance on the reliability, average packet delay, and power consumption. Furthermore, we investigate the fairness of resource allocation, robustness to network changes, and sensitivity to inaccurate parameter estimations. Note that it is hard to compare our algorithm to other algorithms from the literature as the link-based ones [42–46], because they modify the IEEE 802.15.4 standard and are focused on different performance metrics (e.g., throughput). However, it is possible to show that our algorithm outperform significantly the results in [42–46], due to that these results use the ACK feedback, which has a low update frequency with respect to the channel and network variations, whereas our algorithm reacts much faster. Details follow in the sequel.

11.7.1 Protocol Behavior in Stationary Conditions

In this subsection, we are interested in the improvement of performance metrics of the proposed scheme at stationary conditions of the network, namely without changing application requirements and network scenarios. We also present a fairness analysis of the adaptive protocol. Simulation data was collected out of 5 runs, each lasting 2×10^5 time slots.

Figure 11.3 compares the reliability, average delay, and power gain values of the protocol as obtained by our algorithm and with default MAC parameters. Both the *I-mode* and *S-mode* for various traffic configurations and requirements are considered. The requirements for both the *I-mode* and *S-mode* are $R_{\min} = 0.9, 0.95$, $D_{\max} = 50$, and $R_{\min} = 0.95$, $D_{\max} = 20, 100$ ms, respectively. Figure 11.3a shows that both *I-mode* and *S-mode* satisfy the reliability constraint for different traffic regimes. We observe strong dependence of the reliability of the default MAC protocol with different traffic regime due to the fixed MAC parameters. At the high traffic regime $q = 0.2$, the reliability of default MAC is 0.86. In Fig. 11.3b, the delay constraint is fulfilled in both *I-mode* and *S-mode*. Observe that the average delay in the *I-mode* decreases when traffic regime is low $q \geq 0.5$. This is due to that the optimal MAC parameters at higher traffic regime increase more than the ones at lower traffic regime to satisfy the reliability constraint.

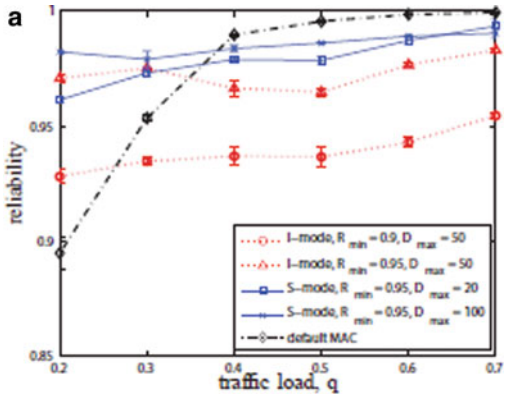
Recall that the target of our proposed adaptive algorithm is to use the tradeoff between application constraints and energy consumption instead of just maximization of reliability or minimization of delay. Therefore, to characterize quantitatively the power consumption, we define the power gain as

$$\rho = \frac{E_{\text{def}} - E_{\text{tot}}(\mathbf{V})}{E_{\text{def}}}$$

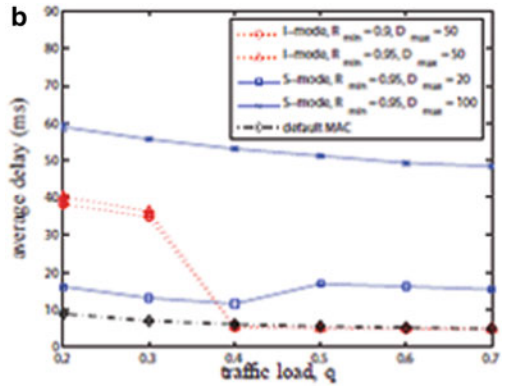
where E_{def} and $E_{\text{tot}}(\mathbf{V})$ are the average power consumption of *I-mode* or *S-mode* for default MAC and proposed scheme, respectively. The closer ρ is to 1, the better power efficiency. Figure 11.3c shows that the power gain increases as traffic load increases. This improvement is higher for *S-mode* than *I-mode*, e.g., power gain $\rho \approx 0.49$ for *S-mode* with $R_{\min} = 0.95$, $D_{\max} = 100$. Although there is a strong dependence of the power gain on the traffic regime, our proposed algorithm gives a better energy efficiency than the default MAC. Therefore, the numerical results show clearly the effectiveness of our adaptive IEEE 802.15.4 MAC protocol while guaranteeing the constraints.

Next, we observe the tradeoff between the power consumption, reliability, and delay constraints. Figure 11.4 shows the dependence of the power consumption in *S-mode* with reliability and delay constraints for a given traffic load, packet length, and number of nodes. Observe that as the delay constraint becomes strict the power consumption increases. In other words, the reliability constraint of *S-mode* is less critical than delay constraint, see more results in [8].

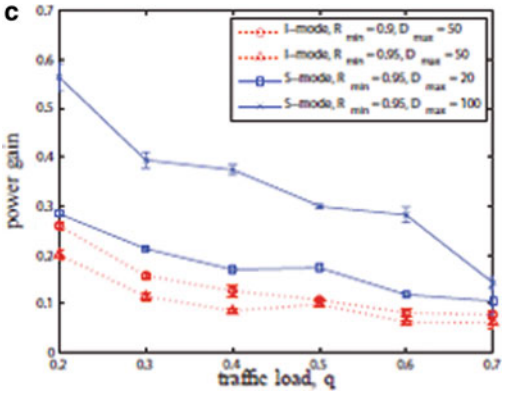
The fairness of resource management is one of the most important concerns when implementing the tuning algorithm of the MAC parameters. We use Jain's fairness index [53] to show the fairness of our proposed scheme for both *I-mode* and *S-mode*. We compute the fairness index of 10 nodes in a stable network. The closer fairness index to 1, the better the achieved fairness. Figure 11.5 shows the fairness index of the reliability for the different requirements and traffic configurations with a given length of the packet and number of nodes. Figure 11.5 reports a very high fairness achievement on reliability greater than 0.999. A similar behavior is found for delay



Reliability



Average delay



Power gain

Fig. 11.3 Stationary condition: reliability, average delay, and power gain of the *I-mode*, *S-mode* of proposed scheme and IEEE 802.15.4 with default parameter ($macMinBE = 3$, $macMaxBE = 5$, $macMaxCSMABackoffs = 4$, $macMaxFrameRetries = 3$) as a function of the traffic load $q = 0.2, \ldots, 0.7$, the reliability requirement $R_{\min} = 0.9, 0.95$ and delay requirement $D_{\max} = 20, 50, 100$ ms for the length of the packet $L = 7$ and $N = 10$ nodes. Note that "default MAC" refers to IEEE 802.15.4 with default MAC parameters

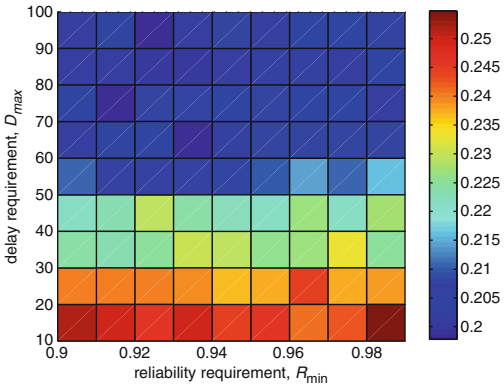


Fig. 11.4 Stationary network condition: power consumption of *S-mode* as a function of reliability constraint $R_{\min} = 0.9, \ldots, 0.99$ and delay requirement $D_{\max} = 10, \ldots, 100$ ms for the traffic load $q = 0.5$, the length of packet $L = 3$ and $N = 10$ nodes

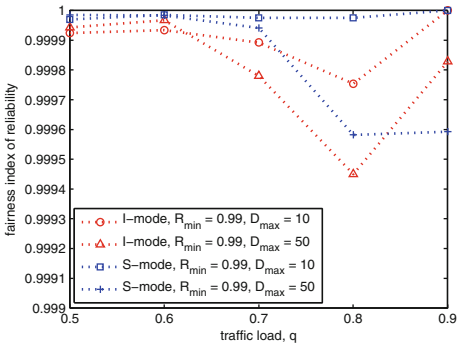


Fig. 11.5 Fairness index of the reliability as a function of the traffic load $q = 0.5, \ldots, 0.9$, reliability requirement $R_{\min} = 0.99$, and delay requirement $D_{\max} = 10, 50$ ms for the length of the packet $L = 3$ and $N = 10$ nodes

and power consumption. In other words, the MAC parameters of each node converge to the optimal MAC parameter values. Therefore, we conclude that most of the nodes can equally share the common medium.

11.7.2 Protocol Behavior in Transient Conditions

The adaptive IEEE 802.15.4 MAC protocol is based on the estimation of the busy channel probabilities α and β and the channel access probability τ. In this section, we investigate the convergence time of the optimal MAC parameters obtained by our adaptive algorithm when the delay constraint changes.

Figure 11.6 shows the behavior of channel state, MAC parameters, reliability, and packet delay when the delay requirement changes for both *I-mode* and *S-mode* with a given traffic load, length of packets, and number of nodes. Figure 11.6a reports the busy channel probabilities α and β and channel access probability τ over time. In Sect. 11.6, we noticed that the update frequency of α, β, τ is different. τ is updated in each *aUnitBackoffPer- iod* and α and β are updated when a node stays in CCA_1 and CCA_2. Hence, the update order of α, β, and τ is τ first, then α, and finally β. We remark here that the update frequency of link-based adaptation is lower than the update frequency of β of our algorithm since link-based adaptation requires an ACK transmission [42]–[46]. The update frequency of channel estimation is a critical issue where the traffic load is low, such as in monitoring applications.

Figure 11.6b shows the adaptation of the MAC parameters. The optimal (m_0, m, n) of *I-mode* and *S-mode* adapts to $(3, 2, 0)$ and $(8, 5, 0)$, respectively, before the requirement changes. Observe that the algorithm returns different parameters for *I-mode* and *S-mode* due to the different power consumption model, see details in Sect. 11.5. After the requirement changes at time 26 s, the MAC parameters (m_0, m, n) of *S-mode* adapt from $(8, 5, 0)$ to $(5, 2, 0)$. We observe that the convergence of the MAC parameters is very fast since our algorithm is based on an analytical model instead of heuristic considerations as in link-based adaptation, where the algorithms adapt the contention window size by the ACK transmission [42]–[46]. In addition, recall that our adaptive IEEE 802.15.4 MAC is based on the physical sensing information before transmitting packets.

Figures 11.6c shows the cumulative packet reception rate of *I-mode* and *S-mode*. Note that the oscillation of reliability is due to packet loss. In Fig. 11.6c, the reliability of *S-mode* is larger than *I-mode* since the MAC parameters m_0 and m are larger than the ones of *I-mode* before the requirement changes. By the same argument, we observe that the packet delay of *S-mode* is about six times the one measured of *I-mode* in Fig. 11.6d. In addition, the packet delay is much more variable in *S-mode* than the one in *I-mode*. Specifically, with *I-mode*, we have a reduction in the average MAC delay and a shorter tail for the MAC delay distribution with respect to the *S-mode*. After the requirement changes, the packet delay converges to around 10 ms. In addition, the reliability decreases due to the decreasing of the parameters m_0 and m in Fig. 11.6c.

11.7.3 Robustness and Sensitivity Analysis

The performance analysis carried out so far assumed that the number of nodes and traffic configuration are fixed. This assumption has allowed us to verify the effectiveness of our adaptive algorithm for IEEE 802.15.4 MAC in steady-state conditions. However, one of the critical issues in the design of wireless networks is that of time-varying conditions. Therefore, in the following analysis, we will investigate how our algorithm reacts to changes in the number of nodes and traffic load when each node has an erroneous estimation of these parameters.

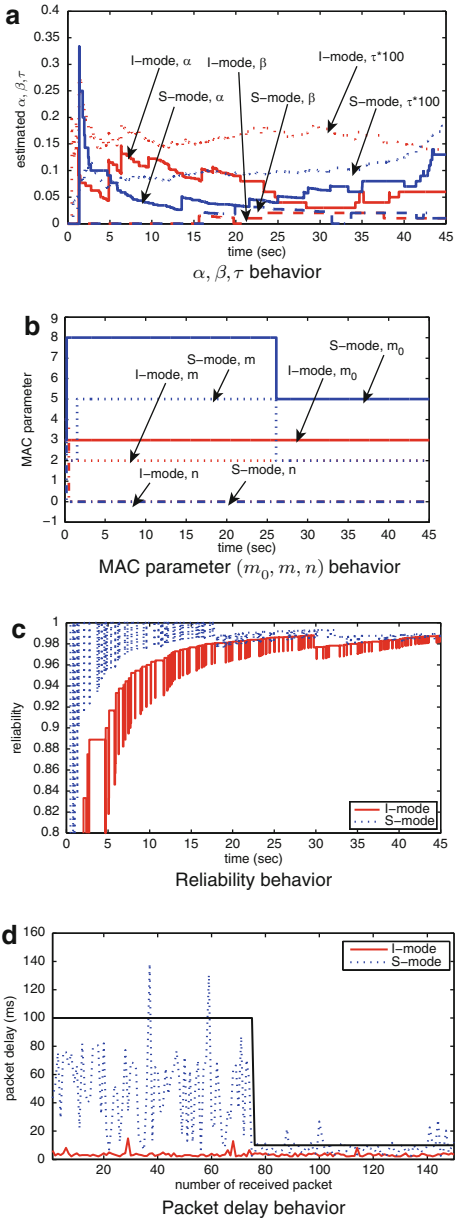


Fig. 11.6 Transient condition: busy channel probabilities, channel access probability, MAC parameters, reliability and delay of *I-mode* and *S-mode* for the traffic load $q = 0.6$, length of the packet $L = 3$ and $N = 10$ nodes when the delay requirement changes from $D_{\max} = 100$ ms to $D_{\max} = 10$ ms at 26 s

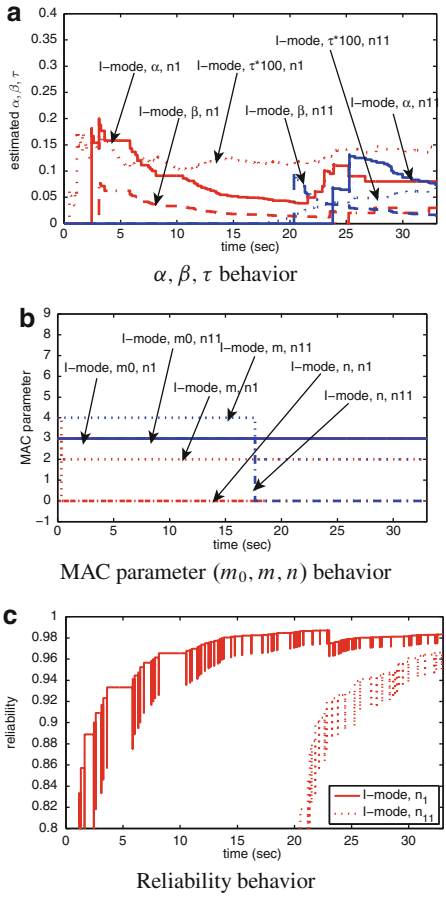


Fig. 11.7 Robustness when the number of nodes changes: busy channel probabilities, channel access probability, MAC parameters, and reliability behavior of *I-mode* when the number of nodes changes sharply from $N = 10$ to $N = 20$ at time 17.6 s. Note that n_1 and n_{11} represent the behavior of one of $N = 10$ nodes plus new nodes after time 17.6 s. Traffic load is $q = 0.6$, length of the packet is $L = 3$, and the reliability and delay constraint are $R_{\min} = 0.95$ and $D_{\max} = 100$ ms, respectively

Figure 11.7 shows the dynamical behavior of nodes using the *I-mode* when the number of nodes changes from $N = 10$ to $N = 20$ with an erroneous estimation of the number of nodes. At time 17.6 s, the number of nodes sharply increases to 20, when it was estimated to be 10. We assume that the wrong estimation happens due to some errors in the estimation phase or a biasing induced by the hidden-node phenomenon. This causes a significant increase of the contention level. Note that n_1 is one of the existing nodes before the network change and n_{11} is one of the new nodes that enters the network at time 17.6 s using its initial MAC parameters. In Fig. 11.7a, we observe that the busy channel and channel access probabilities of

node n_{11} become stable after the network changes by updating the MAC parameters. Fig. 11.7b shows that the MAC parameters (m_0, m, n) converge to $(3, 2, 0)$ for nodes n_1 and n_{11}. The figures indicate that the system reacts correctly to the erroneous estimation of the number of nodes after a few seconds. In Fig. 11.7c, the reliability fulfills the requirement $R_{\min} = 0.95$ for both the existing and new nodes. Similar behaviors are observed for *S-mode*, see further details in [8].

Figure 11.8 presents the behavior of the node when the traffic load changes sharply from $q = 0.8$ to $q = 0.5$ at time 25.6 s. Nodes use a wrong estimation of the traffic load, which is estimated to be $q = 0.8$, after the traffic load changes. The results indicate that our algorithm is quite effective for the traffic configuration

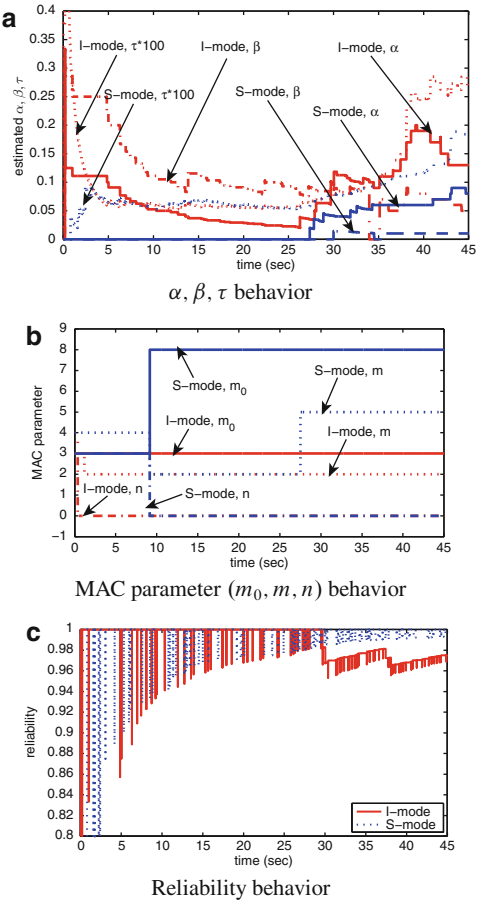


Fig. 11.8 Robustness when the traffic load changes: busy channel probabilities, channel access probability, MAC parameters, reliability and delay behavior of *I-mode* and *S-mode* when the traffic load changes sharply from $q = 0.8$ to $q = 0.5$ at time 25.6 s. The length of the packet is $L = 3$ and the reliability and delay constraint are $R_{\min} = 0.95$ and $D_{\max} = 100$ ms, respectively

change. In Fig. 11.8a, the busy channel and channel access probability increase as a result of higher traffic regime $q = 0.5$ for both *I-mode* and *S-mode*. Figure 11.8b shows that the parameter m of *S-mode* updates from 2 to 5 due to the increasing busy channel probability after the traffic load changes at time 28 s. The figure indicates that the system reacts correctly to the erroneous estimation of traffic configuration and, in a few seconds, the estimation of α, β, and τ makes it possible to reach the optimal MAC parameters. In Fig. 11.8c, the reliability requirement $R_{\min} = 0.95$ is fulfilled for both *I-mode* and *S-mode*. The reliability of *I-mode* is greater than 0.95 with some fluctuations after the traffic load increases.

In Sect. 11.5, we assume that the ideal channel sensing capability of hardware without hidden node terminals. However, this assumption may not be practical due to the hardware failure and time-varying wireless condition. Hence, it is important analyze the sensitivity of our adaptive IEEE 802.15.4 MAC to the estimation errors. Figure 11.9 illustrates the sensitivity of the proposed scheme with respect to the estimation errors to the busy channel probabilities α and β and the channel access probability τ. The normalized root mean squared deviation (NRMSD) between the optimal MAC parameters with exact estimation and the ones with erroneous estimation is used as the indicator of sensitivity. The normalization is taken over the range of MAC parameters (m_0, m, n). The NRMSD is approximately below 10% if the percentage of error is smaller than 20% for α, β, and τ. It is interesting to observe that m_0 of *I-mode* is very robust to errors. This is due to the power consumption model, i.e., to the dominant factor m_0 of power consumption in *I-mode*. The robustness of MAC parameter is $m_0 > n > m$ and $n > m > m_0$ for *I-mode* and *S-mode*, respectively. We can show that errors below 20% in the estimation of α, β, τ give a performance degradation below 3% in terms of reliability, packet delay and energy gain for low traffic load.

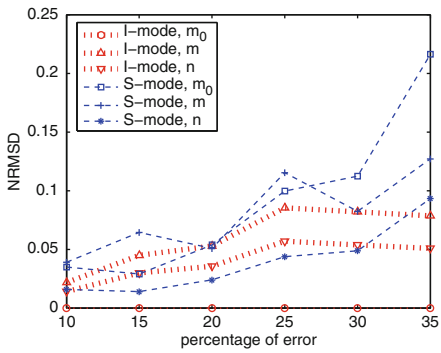


Fig. 11.9 Sensitivity: NRMSD of *I-mode* and *S-mode* when the traffic load $q = 0.6$, length of the packet $L = 3$, reliability requirement $R_{\min} = 0.95$ and delay requirement $D_{\max} = 100$ ms, and $N = 10$ nodes with different percentage error in busy channel probabilities α and β and channel access probability τ

11.8 Conclusions

In this chapter, we developed an analysis based on a generalized Markov chain model of IEEE 802.15.4, including retry limits, ACKs, and unsaturated traffic regime. Then, we presented an adaptive MAC algorithm for minimizing the power consumption while guaranteeing reliability and delay constraints of the IEEE 802.15.4 protocol. The algorithm does not require any modifications of the standard. The adaptive algorithm is grounded on an optimization problem, where the objective function is the total power consumption, subject to constraints of reliability and delay of the packet delivery, and the decision variables are the MAC parameters (*macMinBE*, *macMaxCSMABackoffs*, *macMaxFrameRetries*) of the standard. The proposed adaptive MAC algorithm is easily implementable on sensor nodes by estimating the busy channel and channel access probability.

We investigated the performance of our algorithm under both stationary and transient conditions. Numerical results showed that the proposed scheme is efficient and ensures a longer lifetime of the network. In addition, we showed that, even if the number of active nodes, traffic configuration, and application constraints change sharply, our algorithm allows the system to recover quickly and operate at its optimal parameter, by estimating just the busy channel and channel access probabilities. We also studied the robustness of the protocol to possible errors during the estimation process on number of nodes and traffic load. Results indicated that the protocol reacts promptly to erroneous estimations.

Acknowledgements This work was supported by the EU project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation, and the Swedish Governmental Agency for Innovation Systems. A preliminary version of this book chapter has been presented at ACM/IEEE International Conference on Information Processing in Sensor Networks, Stockholm, Sweden, 2010, and has been submitted to a journal.

References

1. *IEEE Std 802.15.4-2996, September, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2006. [Online]. Available: http://www.ieee802.org/15.
2. A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, pp. 1130–1151, June 2005.
3. A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, pp. 609–621, May 2008.
4. C. Fischione, A. Speranzon, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Peer-to-peer estimation over wireless sensor networks via Lipschitz optimization," in *ACM/IEEE IPSN*, 2009.
5. T. Abdelzaher, T. He, and J. Stankovic, "Feedback control of data aggregation in sensor networks," in *IEEE CDC*, 2004.
6. S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, F. Catthoor, A. Bahai, and P. Varaiya, "Performance analysis of slotted carrier sense IEEE 802.15.4 acknowledged uplink transmissions," in *IEEE WCNC*, 2008.

7. J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, pp. 29 – 47, Jan. 2007.
8. P. Park, P. D. Marco, C. Fischione, and K. H. Johansson, "Adaptive IEEE 802.15.4 protocol for reliable and timely communications," KTH, Tech. Rep., 2009, submitted for journal publication.
9. *IEEE Std 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 1999. [Online]. Available: http://www.ieee802.org/11.
10. G. Bianchi, "Performance analysis of the IEEE 802.11 distributed cordination function," *IEEE Journal on Selected Areas in Communications*, pp. 535 – 547, March 2000.
11. P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "IEEE 802.11 packet delay - A finite retry limit analysis," in *IEEE GLOBECOM*, 2003.
12. Z. Hadzi-Velkov and B. Spasenovski, "Saturation throughput-delay analysis of IEEE 802.11 in fading channel," in *IEEE ICC*, 2003.
13. O. Tickoo and B. Sikdar, "Queueing analysis and delay mitigation in IEEE 802.11 random access MAC based wireless networks," in *IEEE INFOCOM*, 2004.
14. H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," in *IEEE INFOCOM*, 2002.
15. J. Zheng and M. L. Lee, "A comprehensive performance study of IEEE 802.15.4," in *IEEE Press Book*, 2004.
16. A. Koubaa, M. Alves, and E. Tovar, "A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks," in *IEEE IWFCS*, 2006.
17. S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. V. D. Perre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," in *IEEE GLOBECOM*, 2006.
18. P. K. Sahoo and J. P. Sheu, "Modeling IEEE 802.15.4 based wireless sensor network with packet retry limits," in *ACM PE-WASUN*, 2008.
19. J. Mišić and V. Mišić, "Access delay for nodes with finite buffers in ieee 802.15.4 beacon enabled pan with uplink transmissions," *Computer Communications*, pp. 1152–1166, April 2005.
20. J. Mišić, S. Shaf, and V. Mišić, "Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic," *IEEE Transactions Parallel and Distributed Systems*, pp. 361–376, April 2006.
21. C. Y. Jung, H. Y. Hwang, D. K. Sung, and G. U. Hwang, "Enhanced markov chain model and throughput analysis of the slotted CSMA/CA for IEEE 802.15.4 under unsaturated traffic conditions," *IEEE Transactions on Vehicular Technology*, pp. 473–478, Jan. 2009.
22. Z. Tao, S. Panwar, G. Daqing, and J. Zhang, "Performance analysis and a proposed improvement for the ieee 802.15.4 contention access period," in *IEEE WCNC*, 2006.
23. I. Ramachandran, A. K. Das, and S. Roy, "Analysis of the contention access period of ieee 802.15.4 mac," *ACM Transactions on Sensor Networks*, pp. 641–651, 2007.
24. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," *IEEE Journal on Selected Areas in Communications*, pp. 1774 – 1786, Sept. 2000.
25. M. Martalo, S. Busanelli, and G. Ferrari, "Markov chain-based performance analysis of multihop ieee 802.15.4 wireless networks," *Performance Evaluation*, pp. 722–741, 2009.
26. J. Mišić, J. Fung, and V. Mišić, "Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis," in *IEEE ISPAN*, 2005.
27. J. Mišić and R. Udayshankar, "Slave-slave bridging in 802.15.4 beacon enabled networks," in *IEEE WCNC*, 2007.
28. G. Lu, B. Krishnamachari, and C. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *IEEE IPCCC*, 2004.
29. N. Timmons and W. Scanlon, "Analysis of the performance of IEEE 802.15.4 for medical sensor body area networking," in *IEEE SECON*, 2004.
30. L. Changle, L. Huan-Bang, and R. Kohno, "Performance evaluation of ieee 802.15.4 for wireless body area network (wban)," in *IEEE ICC Workshops*, 2009.

31. A. Koubaa and Y. Q. Song, "Evaluation and improvement of response time bounds for real-time applications under non-preemptive fixed priority scheduling," *International Journal of Production and Research (IJPR)*, pp. 2899–2913, July 2004.
32. A. Koubaa, M. Alves, and E. Tovar, "GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks," in *IEEE IPDPS*, 2006, pp. 25–29.
33. ——, "Energy and delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor networks," *International Journal of Communication Systems*, 2006.
34. A. Koubaa, M. Alves, and E. Tovar", "i-GAME: An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks," in *Proc. 18th Euromicro Conf. Real-Time Systems (ECRTS 06)*, 2006.
35. P. Park, C. Fischione, and K. H. Johansson, "Performance analysis of GTS allocation in beacon enabled IEEE 802.15.4," in *IEEE SECON*, 2009.
36. L. Cheng, X. Zhang, and A. G. Bourgeois, "GTS allocation scheme revisited," *Electronics Letters*, pp. 1005–1006, 2007.
37. S. E. Yoo, D. Y. Kim, M. L. Pham, Y. M. Doh, E. C. Choi, and J. D. Huh, "Scheduling support for guaranteed time services in IEEE 802.15.4 low rate WPAN," in *IEEE RTCSA*, 2005.
38. C. Na, Y. Yang, and A. Mishra, "An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks," *Computer Networks*, pp. 2543–2557, Sept 2008.
39. Y. Huang, A. Pang, and H. Hung, "An adaptive GTS allocation scheme for IEEE 802.15.4," *IEEE Transactions Parallel Distrib. Syst.*, pp. 641–651, 2008.
40. K. Yedavalli and B. Krishnamachari, "Enhancement of the IEEE 802.15.4 MAC protocol for scalable data collection in dense sensor networks," in *ICST WiOPT*, 2008.
41. R. Bruno, M. Conti, and E. Gregori, "Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs," *IEEE Transactions on Mobile Computing*, pp. 10 – 31, Jan. 2002.
42. Q. Pang, S. C. Liew, J. Y. B. Lee, and V. C. M. Leung, "Performance evaluation of an adaptive backoff scheme for WLAN: Research articles," *Wirel. Commun. Mob. Comput.*, pp. 867–879, Dec. 2004.
43. V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," *ACM SIGCOMM*, 1994.
44. B. Bensaou, Y. Wang, and C. C. Ko, "Fair medium access in 802.11 based wireless ad-hoc networks," in *ACM MobiHoc*, 2000.
45. J. G. Ko, Y. H. Cho, and H. Kim, "Performance evaluation of IEEE 802.15.4 MAC with different backoff ranges in wireless sensor networks," in *IEEE ICCS*, 2006.
46. A. C. Pang and H. W. Tseng, "Dynamic backoff for wireless personal networks," in *IEEE GLOBECOM*, 2004.
47. S. C. Ergen, "IEEE 802.15.4 summary," Advanced Technology Lab of National Semiconductor, Tech. Rep., 2004.
48. A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communication Magazine*, pp. 97–107, April 2006.
49. P. Park, P. D. Marco, P. Soldati, C. Fischione, and K. H. Johansson, "A generalized markov chain model for effective analysis of slotted IEEE 802.15.4," in *IEEE MASS*, 2009.
50. D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
51. F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, pp. 785 – 799, Dec. 2006.
52. S. C. Ergen, P. D. Marco, and C. Fischione, "MAC protocol engine for sensor networks," in *IEEE Globecom*, 2009.
53. R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, Tech. Rep., 1984.

Chapter 12
Co-design of IEEE 802.11 Based Control Systems

George W. Irwin, Adrian McKernan, and Jian Chen

Abstract This chapter is concerned with wireless control using IEEE 802.11 or WiFi technology. A cart-mounted inverted pendulum was used to compare wireless control, using a packet-based DMC predictive controller against the conventional hard-wired performance. Importantly, the random delay from the 802.11 channel is modelled by an Inverse Gaussian probability distribution derived experimentally from practical measurements in a wireless reverberation chamber. The same application is also employed to expose the advantages of variable sampling for wireless control, specifically using state differential sampling. In both cases, the round trip delay constitutes a QoS measure to support the overall co-design approach to control, enabling the controller to respond to changing channel conditions to maintain the designed performance. While the chapter has a strong application focus, the results are presented in the general context of an up-to-date review of the relevant research in wireless network control. This includes alternative approaches to theoretical closed-loop stability and proposals for packet-based predictive control and sample rate adaptation.

Keywords Feedback control systems · Dynamic matrix control · Packet-based control · Co-design · IEEE 802.11 · Qos-based sampling · Event-based sampling · State-differential sampling · Inverted pendulum · Delay distribution

12.1 Introduction

Wireless networked control systems (WNCs), whose controller and field devices are connected through a common wireless network, are attracting significant interest from both academic researchers and industry as demonstrated by the growing literature [17, 21]. The distributed architecture of a WNCS provides an opportunity to realise cost-effective, flexible and reliable systems [13, 52]. Future oil and gas

G.W. Irwin (✉)
Intelligent Systems and Control, School of Electronics,
Electrical Engineering and Computer Science, Queens University Belfast, UK
e-mail: g.irwin@qub.ac.uk

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_12,

Table 12.1 Comparison of IEEE 802.15.1(Bluetooth), IEEE 802.15.4 (ZigBee) and IEEE 802.11(a/b/g) technologies

	IEEE 802.15.1 Bluetooth	IEEE 802.15.4 ZigBee	IEEE 802.11 (a/b/g) WiFi
Range	10 m	10 m	30-100 m
Throughput	1 Mb/s	20, 40, 250 kb/s	11Mb/s, 54 Mb/s
Power consumption	Low	Very low	Medium

production [35, 39] and process automation [37], being two examples where large numbers of sensors and actuators are placed at different locations with limited installation space.

The commercial benefits of standardisation in the computing and IT sectors have led to a number of wireless technologies such as Bluetooth [2, 19], IEEE 802.15.4 [1] and IEEE 802.11 (a/b/g) [24, 38, 41], which are now being seriously considered for industrial control. For industrial applications, wireless networks must offer real-time operation, reliability, redundancy and determinism in contrast to existing office/commercial systems. Three of the most ubiquitous wireless networks are IEEE 802.11, Bluetooth and Zigbee (Table 12.1), each designed for use in different scenarios. ZigBee is suitable when communication is infrequent, with small packet sizes and where power consumption must be minimised. Bluetooth is able to transmit at higher data rates than Zigbee and also has a low power consumption. The high bandwidth and low latency of IEEE 802.11 communications networks make them the most suitable for a broad spectrum of feedback control including slow hydraulic and chemical processes and fast-acting electromechanical applications.

The challenge is to understand how to utilise this wireless technology in the time-critical scenarios typical of factory automation systems, for example. Since wireless channels are prone to possible transmission errors from channel outages (when the received signal strength drops below a critical threshold) and/or interference, reliability and performance are more likely to be jeopardised than in the case of a wired network. In considering the drawbacks of wireless networks, it becomes evident that WNCs constitute a much more challenging feedback design control arena than wired NCS.

12.2 Problem Formulation

Introducing a control system onto regular data networks poses significant difficulties, as real-time control demands that time-critical control actions are executed strictly before a hard deadline. Some technical challenges arising are now explained.

12.2.1 Time Delays

In a distributed control system, controller and field devices (sensor/actuator etc.) are individually wired with negligible propagation delay and without any contention for the use of the communication medium. Contention for bandwidth on the channel of a communication network will introduce delays in both sensor and actuator signals. The inevitable network-induced time delays caused by sharing a common network can significantly degrade the control performance and even lead to closed-loop instability [31].

Communication delays can be subdivided into two parts: the access delay (τ_{access}), the time a frame stays in the queue waiting to transmit; and propagation delay, the time taken to transmit the frame across the medium (τ_{frame}). The delays between each node are labelled τ_{sc} for the delay between the sensors and the controller and τ_{ca} for that between the controller and plant actuators. Each of these delays encapsulates access and propagation time (12.1).

$$\tau_{\text{sc}} = \tau_{\text{access}} + \tau_{\text{frame}} \qquad \tau_{\text{ca}} = \tau_{\text{access}} + \tau_{\text{frame}} \tag{12.1}$$

Like all sampled-data control systems, there is also an underlying computational delay τ_{c} representing the time that controller requires to provide the desired input signals. However, this delay can be lumped with either τ_{sc} or τ_{ca} without loss of generality as it is often negligible when compared to the contention time for gaining access to the network. The structure of time delays in a WNCS is illustrated in Fig. 12.1, and the round-trip delay is then given as $\tau_{\text{rtd}} = \tau_{\text{sc}} + \tau_{\text{ca}}$. The time it takes for a frame to propagate over a network once it has won contention is often very short. For example, a UDP frame containing one measurement, stored as a double precision floating point number, would take 146.9 µs for all headers and 5.8 μs for the data on an IEEE 802.11b network at full rate. Also, for IEEE802.11b, there are 4 transmission speeds, 11, 5.5, 2 and 1 Mbps and τ_{frame} changes only if the transmission speed of the network changes which does not happen with wired networks. Thus, the fundamental component that influences the nature of the delay in an NCS is the access delay, as this can be orders of magnitude larger than the propagation time. Its characteristics (constant or time-varying, random or deterministic, bounded or unbounded, known or unknown) are dependent on various factors: network topology, communication protocol, transmission rate and size of

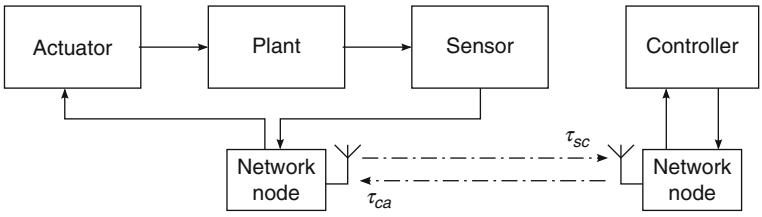


Fig. 12.1 Illustration of time delays in a wireless NCS

message packet [53]. For a wireless network, the time delay depends heavily on protocol at the Medium Access Control (MAC) sub-layer, which determines when a station transmits. This delay is usually timevarying and unknown. Packet loss, another factor which will affect the control system stability, is considered as a packet having an 'infinite' delay. Data packet loss is often therefore not treated as a separate issue. Control design for unknown time-varying delay systems is difficult and remains an open area of research. The problem lies in characterising the channel and compensating for the time-dependent network induced delay.

12.2.2 Communication Channel Errors

The use of the network introduces errors due to station failures and message collisions. This problem is exacerbated in wireless networks, where the channels are inherently prone to transmission errors caused by either channel outages (which occur when the received signal strength drops below a critical threshold) and/or interference, which can be in the form of multipath reflections or packet collisions from other stations. Such errors will lead to packet loss if not retransmitted. Most protocols retransmit erroneous frames (e.g., all unicast frames in IEEE 802.11) but retransmissions cause a growth in random communication delays [10], and control data loses its value if not delivered within its time bound. In the majority of systems, an improvement in performance can in fact be obtained if outdated control signals are dropped. If this strategy is employed, several aspects are of interest such as the optimum data transmission rate that achieves the desired performance and how the missing data is to be modelled by the controller or actuator.

12.2.3 Standardisation

The limitations of fieldbus technology in the automation industry reflects the effect of a lack of a single unified international standard. The challenge nowadays is to determine suitable open standards for WNCs by using readily available standardised hardware or equipment. Several proprietary industrial solutions already exist [54]. However, these are not likely to become the dominant standard since these networks require protocol-specific hardware according to the different implementations and applications. This has led to considerable interest in the use of Commercial-Off-The-Shelf (COTS) technology (Table 12.1) for the WNCS domain.

As highlighted above, the use of wireless channels in feedback control presents some interesting challenges for both control design and implementation. The research literature on networked control (see the taxonomy in Fig. 12.2) generally has dealt with stability analysis, packet-based control, scheduling of NCS/WNCs, as well as combined control and communication co-design solutions in the presence of network delays and packet dropout. The following sections will outline recent

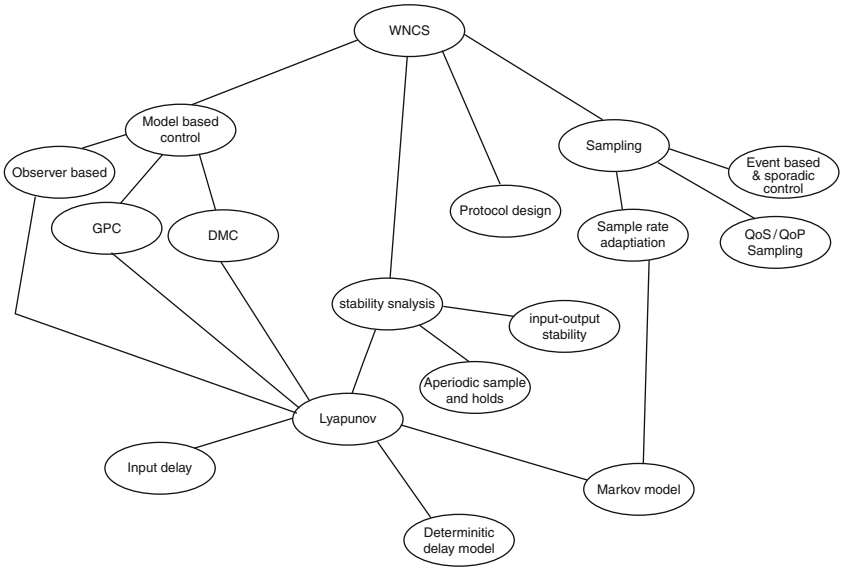


Fig. 12.2 A taxonomy of active research topics in WNCS

relevant research aspects of wired and wireless NCS with an emphasis on the more important/seminal papers. The aim is to go some way to establish the state-of-the-art in WNCs, and to contrast the relative paucity of significant results compared to the better established NCS field.

12.3 Network Induced Delays

Much control theoretical work to date on WNCs, and, indeed, NCS in general, has assumed that network-induced delays are constant or randomly varying, despite the suggestion that the roundtrip delay (τ_{rtd}) distribution in a WLAN is, in fact, unimodal and asymmetric, with a long tail on the right-hand side [18]. In the communications field, the research focus has been on (open-loop) throughput [6] and also on the problem of contending stations [49]. However, recent work on IEEE 802.11 WNCs [11] has shown that induced delays increase exponentially with worsening channel conditions but only grow linearly as more stations are added.

In [25], an experimental setup was used to record the delays that occur in an IEEE 802.11 wireless network with multipath wireless links, where the traffic consists of periodic small packets. This type of traffic was used in contrast to the bursty traffic models common in the communication literature [49] as it is more representative of control applications. Two computers communicated over an IEEE 802.11 wireless channel inside a reverberation chamber (Fig. 12.3). The reverberation chamber [5] consisted of a sealed metallic box and two computer con-

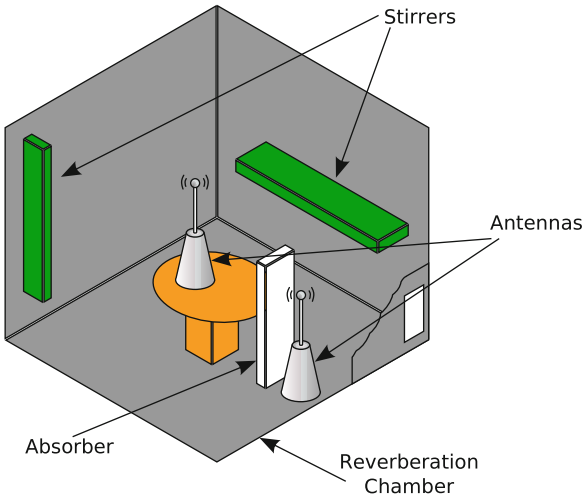


Fig. 12.3 Experimental reverberation chamber setup

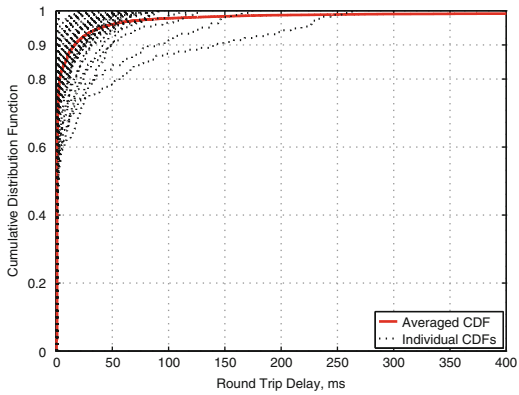


Fig. 12.4 Cumulative density functions of multiple runs of measured τ_{rtd} delay

trolled moving metallic paddles. The aim here was to record and analyse the actual delay characteristics between two nodes in a WNCS under non-ideal wireless link conditions.

This was done by having a 'plant' computer send out a stream of time-stamped measurement packets at intervals of 10 ms, akin to a time-driven plant with constant sampling. These packets arrived at a second 'controller' computer which then replied with another packet which included the original time stamp. When the packet arrived at the original 'plant' computer the time stamp was used to calculate the τ_{rtd}, producing a measurement for a WNCS with time-driven sampling and event-driven control and actuation.

The resulting experimental recorded τ_{rtd} was found to have a unimodal characteristic. Figure 12.4 shows a total of 40 recorded τ_{rtd} distributions as the multipath conditions varied in the reverberation chamber, with the solid line showing the

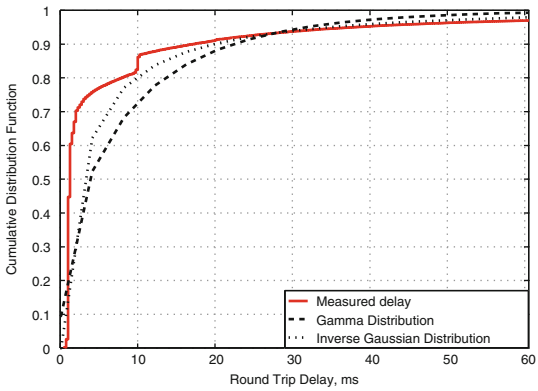


Fig. 12.5 Cumulative density functions of measured τ_{rtd} delay and statistical models

average. Figure 12.5 indicates that this averaged distribution has a minimum value of 0.5 ms, with 90% of the measured delays being less than 17.25 ms and a maximum value of almost 400 ms. Such a skewed distribution can be explained in terms of the use of the back-off algorithm to resolve contention by the IEEE 802.11 MAC layer. In an IEEE 802.11, wireless network packets are positively acknowledged at this layer. When a packet is not successfully acknowledged, it is retransmitted up to a certain limit (nominally six times for IEEE 802.11b). The contention window is doubled for each non-positively acknowledged transmission before being reset by a successful one. This increases the delays as stations spend longertimes contending for transmission, and if the inter-arrival time of new packets (i.e. samples) remains the same, this will lead to the data queue building up exponentially.

An Inverse Gaussian was found to be a better fit than a Gamma distribution [14] to the experimentally recorded τ_{rtd}, as shown in Fig. 12.5. The resultant mathematical model for the cumulative density function of τ_{rtd} is then given by:

$$P(\tau) = \Phi\left(\sqrt{\frac{\lambda}{\tau}}\left(\frac{\tau}{\mu} - 1\right)\right) + \exp\left(\frac{2\lambda}{\mu}\right)\Phi\left(-\sqrt{\frac{\lambda}{\tau}}\left(\frac{\tau}{\mu} + 1\right)\right) \tag{12.2}$$

Here, $\Phi(\cdot)$ denotes a Gaussian cumulative density function. The shape parameter $\lambda = 1.80$, the mean $\mu = 8.24$ and the variance can be calculated as $\frac{\mu^3}{\lambda} = 310.90$. This model will be used in an application study later in this chapter.

12.4 Stability Analysis

There have been several studies to derive criteria for both NCS and WNCs to guarantee stability under certain conditions. The two main issues dealt with in most of the literature are time-varying delays and, especially for WNCS, packet dropout. Theoretical results from the time-delay literature have been applied to the networked

control area and used to define a maximum allowable delay for which a network will be stable [50]. This section outlines recent important results, which may be applicable to the stability of networked control, as well as papers, that are specific to NCS/WNCs. The main areas covered are input delay systems; input-output stability; switched and hybrid systems; stability of predictive control approaches; stability for deterministic and stochastic delays; and systems with time-varying sampling periods.

The input–output stability of an NCS was dealt with in [46], where the $\mathscr{L}_p$ stability of an NCS scheduled with a Persistently Exited (PE) scheduling algorithm was proven. In this case, each node is accessed at least once within a fixed period of time. Time-varying sampling was the subject of [43], where a Linear Matrix Inequality (LMI) gridding method was used to prove stability for different sampling periods. Controllers were synthesised for multiple sampling periods within a given range and applied to a DC motor example to show that a real-time implementation is feasible.

Robust H_∞ control for an uncertain NCS was investigated in [57], with both time-varying network-induced delays and data packet dropouts considered simultaneously. A new method for H_∞ performance analysis and H_∞ control synthesis was derived by introducing some free-weighting matrices and employing information about the nonzero lower bound for the network delay. The H_∞ control design followed from a set of LMIs. This was further studied in [26] where a new Lyapunov-Krasovskii functional, using both lower and upper bounds for the time-varying delay, produced a less conservative delay-dependent H_∞ stabilisation criterion. A sufficient condition for the existence of a robust H_∞ controller took the form of a convex matrix inequality, from which a feasible solution followed from solving a minimisation problem in terms of LMIs. A different stability analysis of a closed-loop control system with a constant network delay in the feedback path using stability regions was presented in [59]. For the delay-only case eigenvalue analysis was used, while with packet drop-out included, switched system theory was explored to prove Lyapunov stability.

Stability of an NCS was explored in [50] with a novel Try-Once-Discard protocol. The error at each node was assigned a weighting and the one with the largest weight won the right to transmit. Although practical determination of the node with the largest weight in a distributed system was not discussed, Lyapunov stability was shown.

In [51], a NCS was modelled as a discrete-time, parameter-uncertain system with a delayed input. Sufficient conditions in LMI form were derived for asymptotic stabilisation using modified robust control theory in the time domain. Since stability for the maximum allowable delay bound from a numerical example was guaranteed, the authors claimed that the probability distribution for the delays need not to be known.

In [58], the random delays in the forward and feedback paths were modelled separately as two Markov chains, with the resulting closed-loop systems being jump linear systems with two modes characterised by these two Markov chains. Necessary and sufficient stability conditions were obtained in terms of a set of LMIs with matrix inversion constraints, from which the state-feedback gain could be

calculated. The network was also modelled as a Markov chain in [28], with H_∞ and H_2 norms being used to compare the performance of estimated control actions from a zero-order hold (ZOH) when control packets are delayed by more than 1 sample period, with estimation giving better performance under poor network conditions.

The stability of a networked predictive control system in [30] included two scenarios: constant delays in both channels and a random time delay with a known upper bound in the forward path and a fixed delay in the feedback one. For constant delays, closed-loop stability was easy to achieve if all the roots of the closed-loop characteristic polynomial lay within the unit circle. In the second case, the NCS was converted to a switched delay system, a sufficient condition for stability being that the eigenvalues of the closed-loop system matrix must be within the unit circle.

A different NCS stability analysis method, for an event-time-driven actuator and controller was introduced in [45]. Here, the controller and actuator would update once new data had arrived but the values were held constant only for a certain time interval after which the data automatically reset to zero if no new packet arrived. A switched delay system including an unstable subsystem was obtained under this mode. A Lyapunov Functional Exponential Estimation method was adopted to arrive at sufficient conditions to guarantee exponential stability.

12.5 Packet-Based Control

A very important advantage of NCS is that the communication network can transmit several measurements as a single packet of data. In IEEE 802.11 for example, the data-packet size in each frame is up to 2312 bytes [24]. The size of any additional control/measurement data will thus have a negligible effect on the transmission times and therefore on the loop time delays. This provides a physical condition for constructing control schemes which operate at a network level, with packets of data being sent through the network rather than using individual values as in conventional digital control. Grouping measurements together in one packet has the added benefit of reducing overheads from both contention and the ratio of header-to-payload. Based on this feature, so-called packet-based control [16] involved sending multiple control actions in a single packet over one transmission period. This new protocol required all the sample values acquired since the last transmission being grouped with the most recent ones for transmission within one packet. The control signals over one transmission period were partitioned to match the A/D period and were sent in a single packet from controller to the actuator. To solve this multipoint-packet control problem, an H_2-optimal controller with a general causality constraint on its feedthrough matrix was derived.

A Generalised Predictive Controller (GPC) to predict a finite control sequence for transmission as a single packet to compensate for time delays was proposed in [47]. This incorporated a minimum-effort polynomial-type estimator [29], which employed a multivariable controlled-auto-regressive-integrated-moving-average (CARIMA) [9] model to estimate dropped or delayed sensor data and a

variable-horizon adaptive GPC controller. Action buffers were placed before the actuators to sequence the future control signals. The variable prediction strategy, with the future control packet size as small as possible to reduce network traffic, was determined from the most recent known controller-to-actuator delay τ_{ca}. The multivariable GPC strategy was finally tested rigorously by controlling a hydraulic servo positioning system on an industrial fish-processing machine over an Ethernet network. A corresponding stability theorem for this strategy was developed in [48] along with an estimate of the worst-case time delay that could be tolerated.

A similar packet-based predictive controller for SISO systems in polynomial form was proposed in [23]. Here, the future control sequences were calculated from τ_{rtd} rather than the individual forward and backward values due to the practical difficulties in determining these individually. The system described consisted of a networked control predictor and a conventional one. The compensator on the plant side selected the control input from the latest arrived packet or from the previous one if there was a packet dropout of τ_{rtd} exceeded a given limit. The control prediction generator, designed by employing GPC, generated the control sequences according to the previous control input and system output. These were buffered at the plant side and transmitted across the network along with time-stamps. This method was also applied to a GPRS wireless network [7] with theoretical and experimental results showing its effectiveness.

Another packet-based scheme [32] relied on multiple observers, corresponding to network-induced delays that were multiples of the sample interval. These supplied the state estimates needed for LQR control. Two sets of observers were utilised, Lost Sample Observers (LSO) to handle packet dropouts and State Prediction Observers (SPO) to compensate for time-varying delays. Lyapunov theory and Linear Matrix Inequalities (LMIs) were used to prove closed-loop stability.

12.6 Example 1: Packet-Based Dynamic Matrix Control

This section presents an extended case study using packet-based Dynamic Matrix Control (PB-DMC). A packet-based state-space formulation of dynamic matrix control was first applied to WNCs in [8]. The control included a set of possible future control actions in every packet transmitted to the actuator, which was then able to select the one closest to its arrival time. Here, a time stamp was attached to each measurement packet before it was sent out from the sensor. On arrival at the actuator this time stamp was compared to the local time to calculate the τ_{rtd}. The delay model (12.2) was applied to provide realistic worst-case wireless conditions to support a simulation study.

To compensate for random delays the buffer at the actuator node only took the most recent generated control sequence in the event that more than one such packet arrives within a given sample interval. It then chose the control signal to use according to the time stamp. In (12.3) below, $p(k_{min[\tau_1,\tau_2,\ldots\tau_k]})$ is the latest predicted control sequence, and $\hat{u}(k) = u(k|k - min\{\tau_i\})$ is the optimal predicted control

value for sample time k. If the buffer is not refreshed at the next sampling time, the next predicted control signal in the last packet is used by the actuator. Thus,

$$p(k_{\tau_i}) = \{u(k - \tau_i + j|k - \tau_i) \quad for \ \ i \in 1, k; j \in 0, \ldots, N-1\} \tag{12.3}$$

12.6.1 Application

The well-known cart-mounted inverted pendulum [36] (Fig. 12.6) is used here to illustrate the performance and stability of the control scheme proposed in [8]. It comprises an open-loop unstable system and therefore provides a challenging real-time WNCS application. The WNCS has the same configuration as Fig. 12.1, with a controller remote from co-located sensing and actuation. The measurements of the cart displacement x and the pendulum angle θ were transmitted to the controller over a wireless IEEE 802.11b channel. The controller then computed the control signal u and transmitted these to the actuator, which applied the appropriate force to the cart. The desired performance criteria for the closed-loop system and are listed in Table 12.2.

The continuous-time model of the inverted pendulum is a 4th order, single-input, two-output plant, given in state-space form as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-(I+ml^2)b}{p} & \dfrac{m^2gl^2}{p} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{-mlb}{p} & \dfrac{mgl(M+m)}{p} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{(I+ml^2)b}{p} \\ 0 \\ \dfrac{ml}{p} \end{bmatrix} u. \tag{12.4}$$

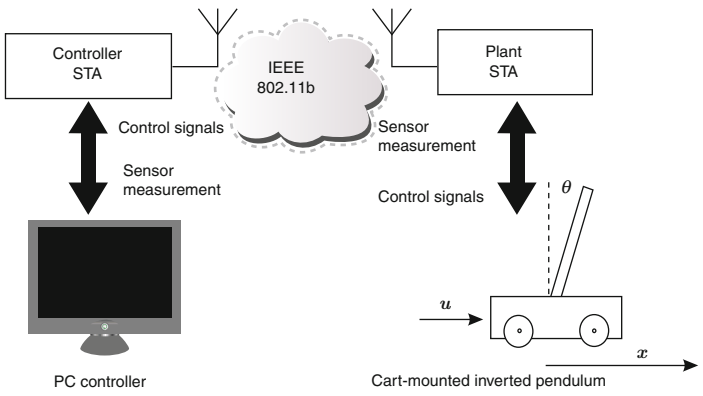


Fig. 12.6 Configuration of the WNCS with a cart-mounted inverted pendulum

Table 12.2 Design criteria for cart-mounted inverted pendulum

Symbol	Criterion	Value
x	Settling Time	< 5 s
x	Rise Time	< 1 s
θ	Stable overshoot	$< 12°$
x and θ	Accuracy	2%

Table 12.3 Parameters and values

Parameter	Description	Value
M	Mass of cart	0.5 kg
m	Mass of pendulum	0.2 kg
b	Friction of cart	0.1 Nms^{-1}
l	Length of pendulum	
	Centre of mass	0.3 m
I	Inertia of pendulum	0.006 kg m^2
g	Gravitational force	9.8 ms^{-2}

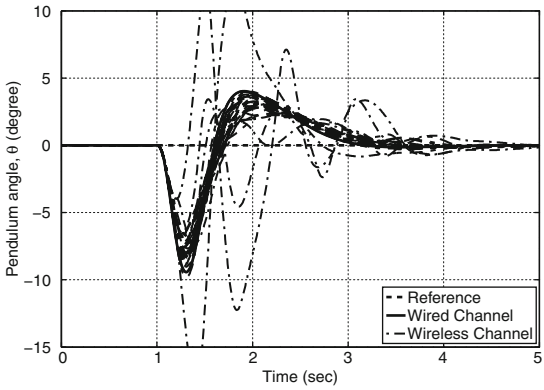


Fig. 12.7 Variation in pendulum angle step-responses for 30 Monte-Carlo simulations for a wireless channel incorporating an Inverse Gaussian model of τ_{rtd} with PB-DMC

The physical parameters $(b, m, l, g, I$ and $M)$ along with their values in (12.4) are defined in Table 12.3, while the parameter p follows from:

$$p = I(M + m) + Mml^2$$

The closed-loop pendulum angle step response shown in Fig. 12.7 were derived from 30 Monte-Carlo simulation trials using random τ_{rtd} values generated from (12.2). The corresponding hard-wired response has been included for comparison purposes. Most of the controlled step-responses are close to that of the hard-wired controller and meet the system specification. By contrast a few others reflect the effect of large delays in the wireless channel, and fall well outside the design specifications.

Figure 12.8 shows the *average* step-response in pendulum angle derived from the Monte-Carlo trials. This clearly demonstrates how well the regulated pendulum can meet the design criteria shown in Table 12.2. It is obvious that the system under

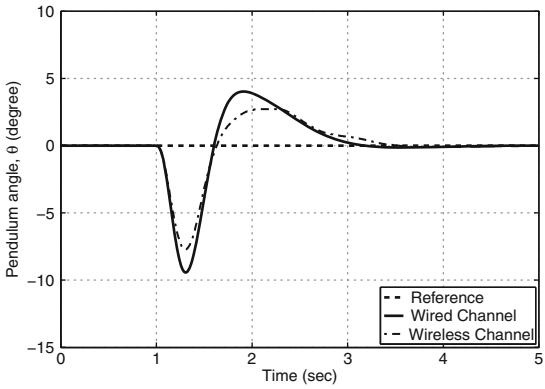


Fig. 12.8 Variation in *averaged* pendulum angle step-responses from 30 Monte-Carlo simulations for a wireless channel incorporating an Inverse Gaussian model of τ_{rtd} with PB-DMC

wireless control takes longer to reach steady-state, although the peak overshoot of the step-response is pleasingly smaller than the hard-wired case.

12.7 Scheduling of WNCS

One area that has seen a growth in WNCs research is the scheduling of samples. This can take several forms including communications protocol design, event-based sampling or sample rate adaption. This section discusses some techniques and results from this field.

From a communications point of view, WNCS design is similar to that of any application involving time-dependent data, such as VoIP or streaming video, and can be dealt with by the protocol prioritising the time sensitive data. An example of this is seen in [55] where a new protocol, based upon IEEE 802.11b, uses the rarely implemented Point Coordination Function (PCF), which allows for a contention free polling period, to create three priority levels of data. Alarm signals have the highest priority then control packets which have no queue, leading to a reduction in time-varying delays, and finally general data communications have the lowest priority. Experimental results showed the effect of using a wireless channel for closed-loop control mixed with standard multimedia data.

12.7.1 QoS Based Sampling

Instead of dealing with the data at a network level to reduce delays, a reduction in the sampling rate, or rate of transmission of those samples, will reduce the network induced delay. If the sampling rate is reduced, then less stations are contending to transmit at the same time, leading to short contention periods and small queues. In a wired network, it is simple to physically restrict access, and the available

bandwidth of the physical medium does not vary with time. However for a wireless channel even if access to the channel can be limited to only equi-sampled control loops, meaning the requested bandwidth is constant, the available bandwidth can be reduced. This happens when the link quality is reduced, causing more packets to arrive in error as the probability of a symbol being corrupted is increased. To combat this, in IEEE 802.11 the transmission rate of each node on the network is reduced (for IEEE 802.11b it will go from 11Mbps to 5.5MBps then 2Mbps).

It stands to reason that varying the sample rate dependent upon some measure of the link quality can stop the control system from saturating the network bandwidth, as increasing the sampling period will reduce the bandwidth used by the control loop. In turn this will stop queue buildup and the consequential increase in network-induced delays. An example of this is focused in [40], where a PI control law was used to adjust the sampling rate to maintain a 5% data loss. This method employed time-based sensing and actuation with event-based control, meaning a remote controller does not need to be clock synchronised to the plant. As the sample rate adapted, the controller gains remained constant and the sample time was upper bounded. This scheme was tested using simulations of a plant and controller communicating over a physical IEEE802.11b wireless network. The plant simulation was later replaced with a physical plant in the form of a rotating inverted pendulum. [27] continued this work and modelled the network as a two-state Markov chain with open-loop and closed-loop states. The frame loss ratio was tracked over a sliding window and, if it was above a system specifically bound, the sample period was increased by a predefined amount. Similarly, if it fell below a predefined value, the same period was reduced. To negate any effects from switch transients, the sample period could only be changed slowly within defined upper and lower limits.

Another paper suggested changing the sample period according to the measured τ_{rtd} of the previous packet [11]. Here, the transitions between sampling times were based on an underlying Markov Chain and stability was shown in a mean-square sense. The approach was also verified using Simulnet, a custom Matlab implementation of the 802.11b Medium Access Control (MAC) layer created in Simulink [12]. [34] changed the delay between successive packets in a batch to reduce the control system bandwidth according to QoS measurements of the previous batch. The number of retransmissions of each packet were likewise adjusted, depending upon previous batch QoS metrics.

12.7.2 Event-Based or Quality of Performance Based Sampling

In addition to basing the sample rate on the current network conditions, a measure of the closed-loop control performance can be employed to decide when to sample. This allows bandwidth to be allocated on the basis of which controlled plant needs it most. The decision on when a plant should send a sample or receive a control update can be done either in a distributed way, with each node deciding or alternatively a centralised decision by some network scheduler can be made.

As mentioned previously, the bandwidth of a wireless channel varies so it make sense to utilise it prudently. Event-based systems sample once a threshold is reached and have been used in many applications but are not as widespread as those with time-driven sampling due to a lack of mature theory. Such systems employ a form of distributed decision making as it is simple for each sensor node to detect when a threshold has been reached and to then decide when to sample. In [3, 4], periodic sampling and event-based sampling, where the system was sampled continuously after the threshold was reached, were studied. Event-based sampling was shown to give a three-fold reduction in error variance compared to periodic control. The discrete case (sporadic control) was dealt with in [20]. The system states were observed at regular intervals and, once they became larger than a set threshold, a sample was taken and the sampler then entered an inactive period before another one could be taken. This strategy allows for the use of hardware designed for discrete measurements. Overall sporadic control can give a reduced error variance compared to the periodic alternative. For WNCs an extra cost can be added to that of sampling, to reflect the cost of communication. A distributed system where each sensor node estimated the states of all nodes with periodic actuation and event-based sampling was presented in [56]. Once a sensor's estimate of the locally measured state varied by a predefined threshold from the measured value, it was broadcast to all the nodes, reducing the overall bandwidth used but increasing the computational load at each node.

In addition to each node deciding if its measurement is sufficiently significant to be transmitted, when several control loops are closed over a common network, a central decision can be made about which plant would benefit most from access to the network. A suitable predefined off-line scheduling algorithm for this case was presented in [42]. Separate LQ controllers, each with a time-varying sample rate, were designed with the requisite time-varying gains computed off-line along with an optimised sampling scheme. In [22], an interrupt-based scheduling policy for Quality of Performance (QoP) networked control was presented. Several control loops were closed over a common idealised communications network with no delay but limited bandwidth. The plant with states furthest from the origin was allowed to close its loop and communicate until the states returned to some predefined distance from the origin. Stability was proven in a Lyapunov sense, although there was no explanation as to how, in a distributed system, the plant with the largest state could be determined.

12.8 Example 2: State Differential Sampling with QoS

This section presents a second case study from the work of the chapter authors. State differential sampling with Quality of Service (QoS) incorporates both a measurement of link quality and one of plant performance. Similar to sporadic control, the states are observed locally at a constant rate until an event is detected, at which point a sample is taken and transmitted to the controller. The observations then enter

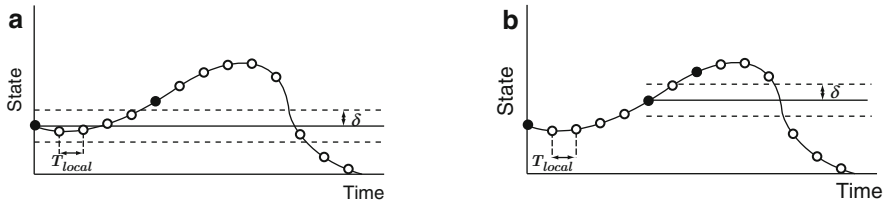


Fig. 12.9 Illustration of state differential sampling, transmitted samples shown as *solid circles*

an inactive period. Events are defined here as the observed state differing by more than certain threshold from the previous sample, with this threshold varying with the quality of the wireless link. This leads to plants with a high state variance being given a faster sampling rate while the overall bandwidth is reduced when the link quality is reduced.

States are first observed at a regular interval T_{local}. If there has been a change in the states greater than some threshold δ (Fig. 12.9) since the last time the sampled states were transmitted, then a sample is taken and transmitted to the controller. The observations then enter an inactive period T_{wait}, where samples cannot be taken. This means that while locally the states are observed at a constant rate T_{local}, they are only sampled and transmitted to the controller at varying multiples of T_{local}. Hence, the input is also only updated at multiples of T_{local}.

As the network can only maintain a maximum throughput, under even ideal wireless conditions, equivalent to a single system being sampled at T_{min}, then the minimum sampling period T must be greater than T_{min}. To detect events quickly, T_{local} is kept small and T_{wait} defined in (12.5) can be increased until $T \geq T_{\text{min}}$.

$$T_{\text{wait}} = T - T_{\text{local}} \tag{12.5}$$

An important factor in state differential sampling is that each node independently decides when to transmit a sample. This eliminates communications overheads from a central coordinator used to delegate sampling rates or slot times. This distribution of bandwidth control is in keeping with the philosophy of IEEE 802.11 networks. Here, each station will increase its contention window if a positive acknowledgement is not received and it may also reduce its transmission rate through Automatic-Rate-Fallback [15].

In [33], the cart-mounted inverted pendulum in Example 1 was used to compare periodic and state differential sampling, over a wireless IEEE 802.11b channel. Simulnet ([12]) a set of Matlab S-functions was used to simulate the behaviour of the IEEE 802.11b wireless network.

In both the state differential and the periodic control cases, a continuous-time Linear Quadratic Regulator (LQR) was designed with $Q = diag([1000\ 0\ 100\ 0])$ and $R = 1$. This produced the constant feedback gain matrix F in (12.6)

$$F = [-31.6 \quad -21.8 \quad 77.1 \quad 20.4]. \tag{12.6}$$

The state measurements were subjected to Gaussian measurement noise $w(t)$, with a mean of zero and variance equal to 10^{-4}.

The minimum sampling period $T_{\min}$ can be derived from the maximum channel bandwidth available for the data after all overheads have been removed. For an IEEE802.11b network, the minimum sampling period is $T = 2\,\text{ms}$ [11]. If $T_{\text{local}} = 1\,\text{ms}$, then $T_{\text{wait}} \geq 1\,\text{ms}$.

The QoS metric used to vary the event threshold was the τ_{rtd} and the state used for event detection was $\dot{\theta}$ with $\delta_{\min} = 10^{-3}\,\text{rad}\,\text{s}^{-1}$. The update policy used for δ was defined by

$$\delta = \begin{cases} 0.001 : \tau_{\text{rtd}}(t_{k+1}) < 10\,\text{ms} \\ u \times 0.005 : u \times 10\,\text{ms} \leq \tau_{\text{rtd}}(t_{k+1}) < (u+1) \times 10\,\text{ms} \\ 0.05 : \tau_{\text{rtd}}(t_{k+1}) \geq 90\,\text{ms} \end{cases} \tag{12.7}$$

For both the state differential and the periodically sampled systems, this is event-based control in that the remote controller produces a control packet whenever a measurement packet arrives. The actuation is also event driven and incorporates a ZOH, which only updates upon arrival of a new control packet.

Two experiments were used to compare the sampling schemes; a single control loop closed over a wireless channel and two control loops closed over a shared channel. Each sampling scheme was simulated five times for the two situations and then repeated with worsening wireless channel conditions, as quantified by a nominal frame error rate (FER), the value that would be experienced by a constant packet source sending one packet every 1 ms.

Table 12.4 lists the results for both constant and state differential sampling with an increasing FER. The Integral-of-Time multiplied by Absolute Error (ITAE) in (12.8) was used to compare performance over the multiple runs as it highlights slow transients [44].

$$ITAE = \int_0^\infty t|e(t)|\text{d}t \tag{12.8}$$

Table 12.4 Performance of varying state differential sampling under different channel conditions (average of 5 Monte-Carlo step responses) for a single control loop

Sampling policy	Frame error rate (%)	Average ITAE	Average sample period, ms	Average τ_{rtd}, ms
State differential	0	0.0868	2.1	1.92
	6	0.0868	2.1	3.15
	14	0.0919	2.5	6.16
	18	0.0941	2.8	9.23
	20	0.0975	3.0	11.1
Constant sampling	0	0.0874	2.0	1.86
	6	0.0877	2.0	5.56
	14	24.6309	2.0	46.2
	18	2.3385×10^{10}	2.0	189
	20	6.4412×10^{9}	2.0	360

Table 12.5 Performance of varying state differential sampling under different channel conditions (average of 5 Monte-Carlo step-responses) for two control loops

Sampling policy	Frame error rate (%)	Average ITAE	Average sample period, ms	Average τ_{rtd}, ms
State differential	0	0.0877	4.1	2.26
	6	0.0883	4.2	3.28
	14	0.0979	4.8	6.45
	18	0.1086	5.0	9.26
	20	0.0975	5.3	10.6
Constant sampling	0	0.0883	4.0	2.65
	6	0.0885	4.0	3.82
	14	0.1041	4.0	19.3
	18	0.8400	4.0	34.2
	20	3.9731×10^7	4.0	101

For an FER of 0%, the average sample rate of the state differentially sampled plant is 0.0021 s, and this increases to 0.003 s for an FER of 20%, with the average delay growing linearly from 1.92 to 11.1 ms. However, for the constant sampled plant, the delay grows exponentially from 1.86 ms for an FER of 0% to 360 ms for an FER of 20%. This causes the constant sampled systems to become unstable as shown by the increase of ITAE to 24.6 for a FER 14%.

Table 12.5 shows the same performance metrics for both equi-sampled and state differential sampled plants, where two control loops are closed over the same network. As before, the state differential sampled plants remain stable and maintain a good performance, achieving an ITAE of less than 0.1, except for an FER of 18% when the ITAE is 0.1086. This is due to the single step response in Fig. 12.11, which had a maximum of 0.3216 and a minimum of −0.2922. Although this constituted a minor degradation in performance compared to the other state differential sampled step responses, it still remained stable. However, it can clearly be seen from Fig. 12.10 that all the equi-sampled plants become unstable between 4 and 5 s.

12.9 Concluding Remarks

This chapter has dealt with wireless control using IEEE 802.11 or WiFi technology. A cart-mounted inverted pendulum was used to compare wireless control, using a packet-based DMC predictive controller with conventional hard-wired performance. Importantly, the random delay from the 802.11 channel was produced from an Inverse Gaussian probability distribution derived experimentally from practical measurements in a wireless reverberation chamber. The same application was also employed to expose the advantages of variable sampling for wireless control,

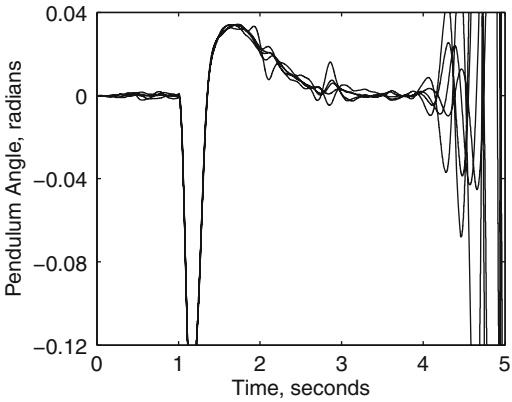


Fig. 12.10 Angular response for one of two remotely controlled plants with a nominal frame error rate of 18%, equi-sampled at 4ms

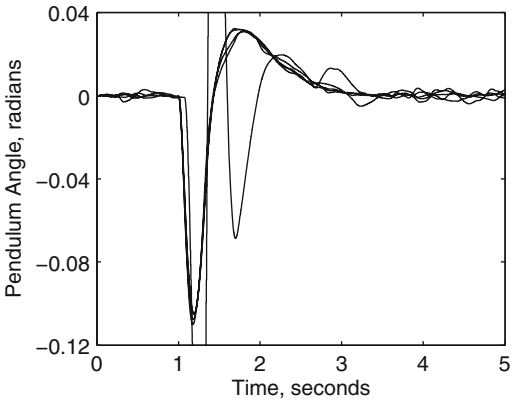


Fig. 12.11 Angular response for one of two remotely controlled plants with a nominal frame error rate of 18%, state differential sampled plant

specifically using state differential sampling. In both cases, the round trip delay was used as a QoS measure to support the co-design approach to control, enabling the controller to respond to changing channel conditions to maintain the designed performance.

While the chapter has a strong application focus, the results are presented in the general context of an up-to-date review of the relevant research in wireless networked control. This includes alternative approaches theoretical closed-loop stability, proposals for packet-based predictive control and sample rate adaptation.

Acknowledgement Jian Chen wishes to acknowledge the financial support of the Virtual Engineering Centre, Queen's University of Belfast, for his doctoral research studies. Likewise, Adrian McKernan thanks the Northern Ireland Department for Education and Learning. All the authors have benefitted from technical input from Prof. William Scanlon on wireless aspects.

References

1. LAN/MAN Standards Committee of the IEEE Computer Society, Information technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (2003)
2. Bluetooth Special Interest Group, Specification of the Bluetooth System, Version 1.1, (December 1999.)
3. Åström, K.J.: Event based control. In: Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori. Springer (2007)
4. Åström, K.J., Bernhardsson, B.: Comparison of Riemann and Lebesque sampling for first order stochastic systems. In: Proceedings of the 41st IEEE Conference on Decision and Control (2002)
5. Bäckström, M., Lundén, O., Kildal, P.S.: Reverberation chambers for EMC susceptibility and emission analyses. Review of Radio Science 1999–2002 pp. 429–452 (2002)
6. Bianchi, G.: Performance analysis of the ieee 802.11 distributed coordination function. Selected Areas in Communications, IEEE Journal on **18**(3), 535–547 (2000)
7. Chai, S., Liu, G., Rees, D.: Predictive control strategy for a wireless networked system. In: Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea (2008)
8. Chen, J., Irwin, G.W., Scanlon, W.G., McKernan, A.: A model predictive approach to wireless networked control. In: Proceeding of the UKACC International Control Conference, Manchester, September 2008 (2008)
9. Clarke, D.W., Mohtadi, C., Tuffs, P.S.: Generalized predictive control - part i. the basic algorithm. Automatica **23**(2), 137–148 (1987)
10. Colandairaj, J.: An approach to wireless networked control. Ph.D. thesis, Queen's University of Belfast (2006)
11. Colandairaj, J., Irwin, G.W., Scanlon, W.: Wireless networked control systems with QoS-based sampling. IET Control Theory and Applications **1**(1), 430–438 (2007). http://dx.doi.org/10.1049/iet-cta:20060519
12. Colandairaj, J., Scanlon, W., Irwin, G.W.: Understanding wireless networked control systems through simulation. Computing & Control Engineering Journal **16**(2), 26–31 (2005)
13. Flammini, A., Ferrari, P., Marioli, D., Sisinni, E., A.Taroni: Wired and wireless sensor networks for industrial applications. Microelectronics Journal **40** (August 2008)
14. Gamez, R., Marti, P., Velasco, M., Fuertes, J.: Wireless network delay estimation for time-sensitive applications. Research report ESAII RR-06-12 (2006)
15. Gast, M.S.: 802.11 Wireless Networks, 2nd edn. O'Reilly (2005)
16. Georgiev, D., Tilbury, D.: Packet-based control: The H2-optimal solution. Automatica **42**(1), 137–144 (2006)
17. Goodman, D., Pollini: Network control for wireless communications. Communications Magazine, IEEE **30, Issue: 12**, 116–124 (1992)
18. Gunawardena, D., Massoulie, L.: Network characteristics: modelling, measurements and admission control. In: International workshop on quality of service, Berkeley CA,, vol. 2707, pp. 3–20 (June, 2003)
19. Haartsen, J.C.: The Bluetooth radio system. IEEE Personal Communications **1.7**, 28–36 (Feb 2000)
20. Henningsson, T., Johannesson, E., Cervin, A.: Sporadic event-based control of first-order linear stochastic systems. Automatica **44**(11), 2890–2895 (2008)
21. Hespanha, J., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proceedings of the IEEE **95**(1), 138–162 (2007)
22. Hristu-Varsakelis, D., Kumar, P.: Interrupt-Based Feedback Control over a Shared Communication Medium (I). In: IEEE Conference on Decision and Control, vol. 3, pp. 3223–3228. IEEE; 1998 (2002)

23. Hu, W., Liu, G.P., Rees, D.: Design and implementation of networked predictive control systems based on round trip time delay measurement. Proceedings of the American Control Conference (2006)
24. IEEE: IEEE Standard 802.11b: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: (1999)
25. Irwin, G.W., Chen, J., McKernan, A., Scanlon, W.G.: Co-design of predictive controllers for wireless network control. IET Control Theory & Applications. (Feb. 2010)
26. Jiang, X., Han, Q.L., Liu, S., Xue, A.: A new H_∞stabilization criterion for networked control systems. IEEE Transactions on Automatic Control **53**(4), 1025–1032 (2008)
27. Kawka, P., Alleyne, A.: Stability and performance of packet-based feedback control over a markov channel. In: American Control Conference, 2006, p. 6. (2006). 10.1109/ACC.2006.1656649
28. Kawka, P., Alleyne, A.: An analysis framework for evaluating dropout compensation strategies in wireless servo systems. Journal of Dynamic Systems, Measurement, and Control **130**, 034, 506 (2008)
29. Kaynak, O., Hashimoto, H., Lewis, P.: Minimum effort control of a servo system. pp. 755–759. Kobe, Japan (1991)
30. Liu, G.P., Mu, J., Rees, D., Chai, S.: Design and stability analysis of networked control systems with random communication time delay using the modified mpc. International Journal of Control **79**(4), 288–297 (2006)
31. Luo, R., Chen, T.M.: Development of a multibehaviour-based mobile robot for remote supervisory control through the internet. IEEE Transactions on Mechatronics **5(4)**, 376–385 (May, 2000)
32. McKernan, A., Ariño, C., Irwin, G.W., Scanlon, W.G., Chen, J.: A multiple observer approach to stability in wireless network control systems. Proceeding of the UKACC International Control Conference, Manchester (2008)
33. McKernan, A., Irwin, G.W.: Event-based sampling for wireless network contol systems with QoS. Submitted to American Control Conference 2010.
34. Nikolakopoulos, G., Panousopoulou, A., Tzes, A.: Experimental controller tuning and qos optimization of a wireless transmission scheme for real-time remote control applications. Control Engineering Practice **16**(3), 333–346(2008). http://dx.doi. org/10.1016/j.conengprac.2007.04.015
35. Nixon, M., Shepard, R., Bennett, B., Chen, D., Mok, A.K.: A framework to transmit process control data over commercial wireless networks. pp. 855–866. Houston, TX, United states (2004)
36. Ogata, K.: Moden Control Engineering. Prentice-Hall, fourth edition (2001)
37. Paavola, M.: Wireless technologies in process automation - a review and an application example. Tech. rep., University of Oulu, Control Engineering Laboratory (Dec. 2007)
38. Panousopoulou, A., Nikolakopoulos, G., Tzes, A., Lygeros, J.: Recent trends on QoS for wireless networked controlled systems. Tech. rep., University of Patras, Electrical and Computer Engineering Dpartment, Greece
39. Petersen, S., Myhre, B., Doyle, P., Mikkelsen, E., Carlsen, S., Sjong, D., Skavhaug, A., Van Der Linden, J.H., Sansom, M.: A survey of wireless technology for the oil and gas industry. Society of Petroleum Engineers - Intelligent Energy Conference and Exhibition: Intelligent Energy 2008 **2**, 826–835 (2008)
40. Ploplys, N., Alleyne, A.: UDP network communications for distributed wireless controls. In: American Control Conference, 2003, vol. 4, pp. 3335–3340 (2003). 10. 1109 /ACC.2003.1244046
41. Ploplys, N.J., Kawka, P.A., Alleyne, A.G.: Closed-loop control over wireless networks. Control Systems Magazine, IEEE **24**(3), 58–71 (2004)
42. Rehbinder, H., Sanfridson, M.: Scheduling of a limited communication channel for optimal control. Automatica **40**(3), 491–500 (2004)
43. Sala, A.: Computer control under time-varying sampling period: An lmi gridding approach. Automatica **41**(12), 2077–2082 (2005). http://dx.doi.org/10.1016/j.automatica.2005.05.017

44. Shinners, S.M.: Modern Control System Theory and Design, 2nd edn. Wiley (1998)
45. Sun, X.M., Liu, G.P., Rees, D., Wang, W.: A novel method of stability analysis for networked control systems. IFAC Proceedings Volumes (IFAC-PapersOnline) **17**(1) (2008)
46. Tabbara, M., Nesic, D., Teel, A.R.: Stability of wireless and wireline networked control systems. IEEE Transactions on Automatic Control **52**(9), 1615–1630 (2007). http://dx.doi.org/10.1109/TAC.2007.904473
47. Tang, P., De Silva, C.: Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. IEEE Transactions on Control Systems Technology **14**(4), 707–718 (2006)
48. Tang, P., de Silva, C.: Stability validation of a constrained model predictive networked control system with future input buffering. International Journal of Control **80**(12), 1954–1970 (2007)
49. Tickoo, O., Sikdar, B.: Modeling Queueing and Channel Access Delay in Unsaturated IEEE 802.11 Random Access MAC Based Wireless Networks. Networking, IEEE/ACM Transactions on **16**(4), 878–891 (2008)
50. Walsh, G., Ye, H., Bushnell, L., et al.: Stability analysis of networked control systems. IEEE Transactions on Control Systems Technology **10**(3), 438–446 (2002)
51. Xie, G., Wang, L.: Stabilization of networked control systems with time-varying network-induced delay. pp. 3551 – 3556. Nassau, Bahamas (2004)
52. Yang, S.H., Cao, Y.: Networked control systems and wireless sensor networks: Theories and applications. International Journal of Systems Science **39**(11), 1041–1044 (2008)
53. Yang, T.: Networked control system: a brief survey. IEE Proceedings-Control Theory and Applications **153**(4), 403–412 (2006)
54. Ye, H., Walsh, G., Bushnell, L.: Wireless local area networks in the manufacturing industry. In: American Control Conference, 2000. Proceedings of the 2000, **4**, 2363–2367 (2000)
55. Ye, H., Walsh, G., Bushnell, L.: Real-time mixed-traffic wireless networks. IEEE Transactions on Industrial Electronics **48**(5), 883–890 (2001)
56. Yook, J., Tilbury, D., Soparkar, N.: Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. Control Systems Technology, IEEE Transactions on **10**(4), 503–518 (2002). 10.1109/TCST.2002.1014671
57. Yue, D., Han, Q., Lam, J.: Network-based robust H control of systems with uncertainty. Automatica **41**(6), 999–1007 (2005)
58. Zhang, L., Shi, Y., Chen, T., Huang, B.: A new method for stabilization of networked control systems with random delays. IEEE Transactions on Automatic Control **50**(8), 1177–1181 (August 2005)
59. Zhang, W., Branicky, M., Phillips, S.: Stability of networked control systems. IEEE Control Systems Magazine **21**(1), 84–99 (2001)

Chapter 13
Coordinated Control of Robotic Fish Using an Underwater Wireless Network

Daniel J. Klein, Vijay Gupta, and Kristi A. Morgansen

Abstract We consider an application of control over a wireless network to coordinate members of a small school of free-swimming underwater vehicles. While these vehicles are capable of limited speed modulation, we restrict them to swimming at constant forward speed. The control task in consideration requires the school to track a moving target whose speed could be considerably less than that of the pursuit vehicles. The coordination task is achieved using an ad-hoc communication network of ultra-low frequency radios, which provide short-range communication links. We report on underwater communication technologies, the design of the coordinating controller, and the application of the controller to a network of robotic fish.

Keywords Distributed control · Networked control systems · Robotic fish · Control under water · Control across wireless networks

13.1 Introduction

Networks of underwater vehicles can be used for a number of practical applications including inspecting piers and docks, securing waterways, and tracking the movements of enemy submarines and marine animals. Tracking marine animals is needed not only to study migration and feeding habits for basic science, but also to aid in conservation efforts. One such conservation effort stems from a recent conflict between high-power active sonar testing by the Navy and marine animals that use sonar for communication, hunting, and navigation [14]. Tracking the animals would cause testing to be delayed to take place only when necessary. Incidentally, high-power sonar testing is used to hone the ability to locate and track an enemy submarine.

We focus on the problem of tracking a mobile target "vehicle" with a school of free-swimming underwater pursuit vehicles (robotic fish). The motions of

D.J. Klein (✉)
Department of Electrical and Computer Engineering,
University of California, Santa Barbara, CA, USA
e-mail: djklein@ece.ucsb.edu

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1_13,

the individual pursuit vehicles will be coordinated to surround the mobile target while maintaining a constant forward speed. This coordinated control objective is achieved over an ad-hoc underwater communication network of low-frequency radio modems. Underwater communication networks are limited in a number of ways, making coordinated control a challenging task.

Communication to and from many underwater vehicles is achieved using a tether. Tethers provide a simple and effective means of communication; however, they come with a number of disadvantages. In particular, they can become snagged or twisted, and they can restrict the mobility of the vehicle considerably.

At the water's surface, satellite communication links can be established for control and coordination purposes. One underwater vehicular system that uses surface-only communication is a glider [4]. A glider is an underwater robot that can sample the ocean for long periods of time by repeatedly diving and surfacing. Coordination of multiple gliders using surface-only communication can be found in [5] and [12]. Note that satellite communication and GPS units cannot be used underwater because their Gigahertz-band radio waves propagate less than a few inches through water.

Manned submarines have long received ultra-low frequency (ULF) radio communications. Low-frequency signals are much better able to propagate through water than high-frequency signals of the same power. However, ULF radio is not appropriate for small underwater vehicles, such as the ones studied in this chapter, because transmitting such low-frequency signals requires very large and specialized equipment.

Marine animals are able to achieve levels of underwater coordination far superior to state-of-the-art underwater robotics. These animals use not only advanced sonar communication, but also vision and lateral-line sensors to coordinate their motions for predator evasion and hunting purposes. Our inspiration for designing underwater robots in a fish-like configuration comes from the efficiency and agile maneuvering exhibited by fish. Many others have designed free-swimming underwater robots including [1, 3], and [16], but few, if any, have multiple vehicles capable of direct vehicle-to-vehicle underwater communication for coordinated control purposes.

The work in this chapter builds upon previous work by the authors. The initial design of the target tracking controller was presented in [10]. Extensions of a portion of the control to a discrete-time setting were made in [15] and [9]. Experiments with this portion of the controller were discussed in [8]. Here, we further the extension to discrete time with the development of oscillation and biasing terms that act to keep the pursuit vehicles near the target.

This chapter is organized as follows. In the next section, we present the underwater coordinated target tracking problem in detail. Then in Sect. 13.3, we review principles of underwater communication and discuss underwater wireless networks. In Sect. 13.4, we present a coordinated heading controller for target tracking and discuss how this controller can be implemented using an ad-hoc underwater network. Results from simulation and from a demonstration with three robotic fish as pursuit vehicles are presented in Sect. 13.5. Concluding remarks are given in Sect. 13.6.

13.2 Problem Formulation

We consider coordination of several fin-actuated underwater vehicles (see Fig. 13.1) that use messages passed through an underwater wireless communication network. The focus is on the design of a feedback controller capable of producing pursuit vehicle trajectories that would enable the generation high-fidelity state estimates of a target vehicle from a gimbaled directional sensor, like a camera. While the pursuit vehicles could move independently, coordinating their motions allows the target to be viewed from diverse vantage points, resulting in more informative target measurements.

A high-fidelity model of the individual fin-actuated vehicle dynamics is available in [13]. However, this model is overly complicated for coordination purposes, and inner loops are readily available to stabilize the main components of the dynamics. Thus, we will concentrate on coordinated control design for a simple abstraction of the full dynamics known as the planar unicycle,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k = \begin{bmatrix} s\cos(\theta) \\ s\sin(\theta) \\ su \end{bmatrix}_k, \quad k = 1, \ldots, N. \tag{13.1}$$

Here, $[x, y, \theta]^T \in SE(2)$ is the state of a single vehicle at a fixed depth, N is the number of vehicles, and s is the forward speed. Although the robotic fish are capable of varying their forward speed, we will take s to be constant. This assumption reduces the control design problem to a single input per vehicle and allows the resulting controller to be applied to a broader class of systems, such as unmanned aerial vehicles. Also, the inner loops running on vehicles such as the robotic fish and unmanned aircraft need to be tuned around a trim speed, and operators prefer to keep the vehicle speed near the trim speed to avoid unpredictable behavior. Note that the speed can be taken as unity through a scaling of time, so we will drop s in the remainder of this text. The target vehicle need not have the same dynamics as

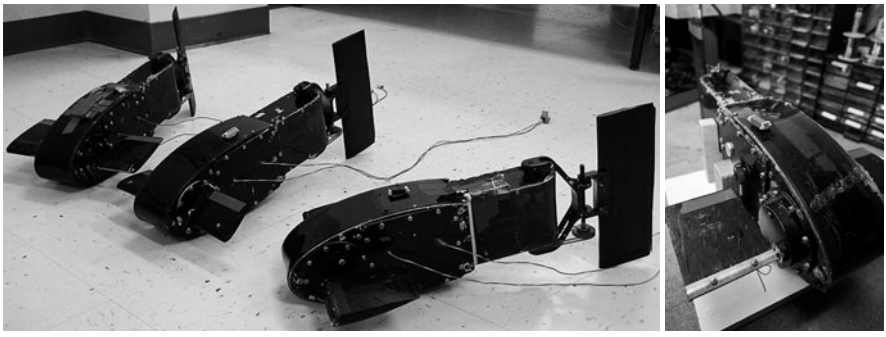


Fig. 13.1 The three fin-actuated underwater vehicles on which the coordinated controller will be implemented

the individual pursuit vehicles, and thus it could be moving at a speed $0 < s_t < 1$ that is significantly slower than the constant pursuit vehicle speed. We will refer to the ratio of the target speed to the pursuit vehicle speed as the *speed ratio*. Further, the speed of the target may be nonconstant. Knowledge of the closed-loop target vehicle dynamics can be used to improve the coordinated target tracking performance, but the information is not necessary for the methods here.

The pursuit vehicles are able to pass information using a wireless underwater communication network. The underwater environment is a hostile place for communication, resulting in severe limitations in bandwidth, topology, and reliability. In our system, the network is achieved at the physical layer using low-frequency radio modems. These modems allow for short-range communication links in fresh water. In salt water or for longer distances in freshwater, acoustic modems can be used. An acoustic modem network is able to support reliable communication at the rate of 80 bits/sec total, whereas the radio modems support a slightly higher bit rate. The radios we have developed use a single carrier frequency, and thus time-division multiplexing must be used in combination with broadcast transmissions. The controller must ultimately achieve coordination while respecting these severe communication limitations.

13.3 Underwater Communication

Wireless underwater communication is traditionally achieved using acoustic modems. These devices have an antenna that converts electrical signals into longitudinal compression waves. These waves can propagate long distances in open water, but they experience reflections and other problems in small or cluttered environments. The demonstration presented later in this chapter was conducted in a 36,000 liter above-ground fresh water pool in which reflections caused by hard walls create multiple signal paths between the transmitter and receiver, rendering acoustic communication extremely challenging. Thus, we have developed low-frequency radio modules that are able to communicate across the 25 m diagonal of the pool. While the range of these radios is limited, coordinated control applications often require the vehicles to work at close range.

To better understand the challenges of underwater communication, we begin by reviewing the fundamentals of electromagnetic wave propagation in water. Then, we briefly introduce the radios we have designed for the purpose of underwater coordinated control before describing the medium access protocol used to support the coordinated controller.

13.3.1 Electromagnetic Propagation in Water

Underwater radio communication presents a number of design challenges. These challenges include range and throughput limitations in addition to multi-path, fading, and signal-loss. At the root of these challenges is conductivity, a measure

of the free charges present in a medium. To gain insight into the challenges of underwater radio communication, we begin with the physics underlying plane wave propagation in a lossy medium.

The equations for a plane wave propagating in an arbitrary direction z can be found from Maxwell's equations. The resulting electric and magnetic fields lie entirely in the x-y plane, and are mutually orthogonal. For an x-polarized electromagnetic wave, the electric field, denoted E, is aligned with the x-axis and the magnetic field, denoted B, is aligned with the y-axis. An electromagnetic wave with this polarization angle and initial electric field strength of E_0 has amplitudes given by

$$E_x(t,z) = E_0 \mathrm{e}^{-\alpha z} \cos(\omega t - \beta z) \tag{13.2}$$

$$H_y(t,z) = \frac{E_0}{|\eta|} \mathrm{e}^{-\alpha z} \cos(\omega t - \beta z - \angle\eta), \tag{13.3}$$

where η is the complex *intrinsic impedance*, defined as the ratio of the electric field to the magnetic field. The parameter α is known as the *attenuation constant* and describes the rate at which the amplitude of the field decreases with increasing distance from the source. The *phase constant*, β, characterizes the rate at which the the phase changes with distance.

The propagation parameters, α, β and η, can be determined by three physical properties of the medium: permittivity ε (measured in Farads per meter), permeability μ (measured in Henrys per meter), and conductivity σ (measured in Siemens per meter). Values of these parameters are shown for a vacuum, air, fresh water, and sea water in Table 13.1.

These physical properties ultimately determine the performance and effectiveness of radio communication. The following equations relate the physical properties of the medium to the propagation parameters of electromagnetic waves:

$$\alpha = \frac{\omega\sqrt{\mu\varepsilon}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} - 1 \right]^{\frac{1}{2}} \quad \text{Np/m} \tag{13.4}$$

$$\beta = \frac{\omega\sqrt{\mu\varepsilon}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} + 1 \right]^{\frac{1}{2}} \quad \text{rad/m} \tag{13.5}$$

Table 13.1 Physical properties of several media. For underwater communication, the most significant of these parameters is conductivity, which is much higher for sea water due to dissolved salts

Medium	Permittivity (F/m)	Permeability (H/m)	Conductivity (S/m)
Vacuum	$\varepsilon_0 = 8.854 \times 10^{-12}$	$\mu_0 = -1.256 \times 10^{-6}$	$\sigma_0 \approx 0$
Air	$\varepsilon_\mathrm{a} = 1.006\varepsilon_0$	$\mu_\mathrm{a} \approx \mu_0$	$\sigma_\mathrm{a} \approx 0$
Fresh water	$\varepsilon_\mathrm{f} = 81.0\varepsilon_0$	$\mu_\mathrm{f} \approx \mu_0$	$\sigma_\mathrm{f} = 10^{-3}$
Sea water	$\varepsilon_\mathrm{s} = 70.0\varepsilon_0$	$\mu_\mathrm{s} \approx \mu_0$	$\sigma_\mathrm{s} = 4.0$

$$\eta = \sqrt{\frac{j\omega\mu}{\sigma + j\omega\varepsilon}} \quad \Omega \ . \tag{13.6}$$

The attenuation of the field amplitude decays exponentially at a rate described by α. Air is essentially a lossless medium in that $\alpha \approx 0\,\text{Np/m}$. In fresh and salty water, $\alpha = 1.1 \times 10^{-3}\,\text{Np/m}$ and $\alpha = 7.1 \times 10^{-2}\,\text{Np/m}$, respectively, for a wave with a frequency of 315 Hz. Low-frequency waves propagate further in lossy media because the attenuation constant increases with the square-root of the wave frequency.

The drastic amplitude decay with penetration distance in water is due primarily to conductivity. The incident electromagnetic wave creates an alternating current due to free electrons oscillating at the wave frequency. Conductivity is a measure of the free electrons, and thus factors into the attenuation. The rate of attenuation is characterized by a distance known as the *skin depth*,

$$\delta = \frac{1}{\alpha}. \tag{13.7}$$

The skin depth is the distance over which the electromagnetic wave is attenuated by a factor of e^{-1}. A radio wave can only travel a few skin depths before a receiver will be unable to detect it, unless the source is very powerful. The range of underwater communication can be increased by reducing the carrier frequency or by increasing the transmission power.

13.3.2 Radios for Underwater Coordinated Control

We have developed radios for underwater communication in support of control applications, see Fig. 13.2. The design of these radios was governed by a number of principles and constraints. Based on the physics of the underwater communication described above, we selected TXM-315-LR and RXM-315-LR transmitter and receiver modules from Linx Technologies. These modules operate at a frequency of

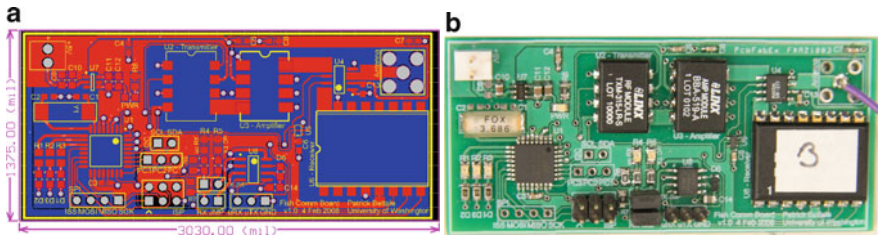


Fig. 13.2 The radio design (**a**) and the resulting printed circuit board (**b**). The antenna attaches in the upper right corner and is extended into the water to prevent loss at the air-water interface

315 MHz, giving a skin depth of nearly 900 m in fresh water, but only 14 m in sea water. Full details of the design and implementation of these radios can be found in [2].

13.3.3 Medium Access to Support Coordinated Control

Because all radios on the network have the same carrier frequency, each transmission will be received by all other vehicles, unless the packet is lost. Simultaneous transmissions will collide, resulting in certain data corruption. To prevent these collisions, we employ a time-division multiplexing for medium access. In these schemes, each vehicle is typically allotted a particular time slot during which it may transmit. One of the drawbacks of this access protocol is the requirement of synchronized clocks.

The communication network need only carry a single packet of data per time slot to support coordinated control as developed in the next section. This structure allows the usual clock synchronization requirement to be relaxed through the use of a round-robin access protocol in which each vehicle broadcasts soon after receiving a message from its sequential predecessor. To maintain the cycle in the presence of dropped packets, the base station (or one of the vehicles) transmits at a regular interval of length T seconds, see Fig. 13.3.

13.4 Coordinated Controller Design

In this section, we develop a controller that coordinates multiple planar unicycles (13.1) for the purpose of tracking a target vehicle. The initial design of the controller

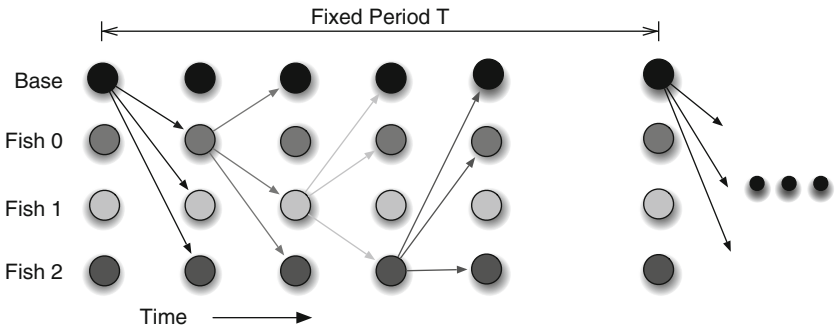


Fig. 13.3 Each vehicle accesses the channel after receiving a message from its sequential predecessor to prevent collisions in the absence of clock synchronization. The base station transmits every T seconds to prevent the cycle from being broken by a dropped packet

is done while ignoring the physical positions of the individual vehicles, focusing instead on only their headings. Thus, the *heading-only* system we initially focus on is of the form

$$\dot{\theta}_k = u_k, \quad k = 1, \ldots, N, \tag{13.8}$$

and we later return to studying the full system.

The motivation for studying this reduced system comes from the fact that the motion of the group as a whole, as described by the centroid velocity, depends only on the headings of the individuals,

$$\bar{\mathbf{r}} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{r}_k, \quad \mathbf{r}_k = \begin{bmatrix} x & y \end{bmatrix}^T \tag{13.9}$$

$$\bar{\mathbf{x}} = \dot{\bar{\mathbf{r}}} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_k, \quad \mathbf{x}_k = \begin{bmatrix} \cos(\theta_k) & \sin(\theta_k) \end{bmatrix}^T. \tag{13.10}$$

The centroid is denoted $\bar{\mathbf{r}}$, and its velocity, denoted $\bar{\mathbf{x}}$, is commonly referred to as the *phasor centroid* in the coupled oscillator literature.

The phasor centroid allows phase-coupled oscillator models to be written in a mean field coupling form. One such coupled oscillator system is the Kuramoto model,

$$u_k = \omega_k + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_k), \tag{13.11}$$

which for homogeneous natural frequencies ($\omega_k = \omega, \ \forall k$) can be written in mean field coupling form,

$$u_k = \omega + K \bar{\rho} \sin(\bar{\theta} - \theta_k). \tag{13.12}$$

Here, $\bar{\rho}$ and $\bar{\theta}$ are the magnitude and orientation of the phasor centroid.

The interesting feature of the homogeneous Kuramoto model is that it exhibits desirable coordination phenomena. By choosing the coupling strength K to be a positive number, the system (13.11) will arrive at a state in which all headings are aligned, corresponding to a maximal group velocity. Choosing K to be a negative number results in anti-aligned headings, corresponding to zero net group motion. To achieve group motions between aligned and anti-aligned, in the next section we introduce a reference vector and modify the homogeneous Kuramoto model to stabilize the phasor centroid (i.e., centroid velocity) to the reference vector.

13.4.1 Controlling the Velocity of the Group

For a constant reference velocity, $\mathbf{x}_{\text{ref}}$, the following theorem adapted from [10] guarantees stability of the centroid velocity to the reference velocity.

Theorem 1 (Reference Matching). *Consider a constant reference velocity vector,* $\mathbf{x}_{\text{ref}} = \rho_{\text{ref}} \angle \theta_{\text{ref}}$*, within the unit circle. Let the error between the group centroid and this reference vector be*

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} - \mathbf{x}_{\text{ref}} = \tilde{\rho} \angle \tilde{\theta}. \tag{13.13}$$

Then, the velocity of the group centroid is asymptotically stabilized to the reference with control input

$$u_k = K \tilde{\rho} \sin(\tilde{\theta} - \theta_k), \ K < 0. \tag{13.14}$$

provided the initial condition *satisfies either: (a) the matrix*

$$P = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix} - \mathbb{1}_N \otimes \mathbf{x}_{\text{ref}} \tag{13.15}$$

has full row rank or (b) $\bar{\mathbf{x}} = \mathbf{x}_{\text{ref}}$*. Here,* $\otimes$ *denotes the Kronecker product.*

Proof. We will not provide a formal proof here, but the main idea is that the controller can be seen as a gradient controller of the *phasor centroid error potential*,

$$\tilde{U} = \frac{N}{2} \|\tilde{\mathbf{x}}\|^2 . \tag{13.16}$$

This potential is everywhere positive except at points where the phasor centroid matches the reference vector, where the potential is zero. Using this potential as a Lyapunov function, a proof by LaSalle's Invariance Principle follows directly. □

If the reference vector is not constant due to a maneuvering target vehicle, the controller can be modified to include an additional term which enables improved tracking. This additional term comes from a feedforward calculation based on knowledge of the motion of the target. It is at this point that a model of the target motion can be incorporated into the controller to yield improved tracking. More details can be found in [10].

For implementation in an underwater wireless network, the controller (13.14) would require each vehicle to know the heading of each other vehicle at every time instant. This requirement is clearly incompatible with physics of underwater communication. Instead, we consider a ΔT time-discretized version of the controller,

$$\theta_i(h+1) = \theta_i(h) - K \Delta T \tilde{\rho}(h) \sin(\tilde{\theta}(h) - \theta_i(h)), \tag{13.17}$$

where $\tilde{\rho}(h)$ and $\tilde{\theta}(h)$ are the magnitude and phase of the centroid error at time-step h. This controller is guaranteed to stabilize the group velocity to a constant reference velocity provided that $-2/(2 + \rho_{\text{ref}}) < K \Delta T < 0$ and that several other technical assumptions are satisfied, see [15] for details.

13.4.2 Controlling the Average Velocity of Each Vehicle

The coordination controller designed in the previous section enables the velocity of the group center to be manipulated through the reference vector without specifically considering the positions of the individual vehicles. In fact, for a target moving at a constant velocity, the distance between the vehicles and the target will grow without bound because the vehicles will not turn once equilibrium has been reached.

To keep the vehicles near the target without breaking the asymptotic stability of the reference velocity matching input, we consider an additive control term v_k,

$$\dot{\theta}_k = u_k + v_k, \tag{13.18}$$

where u_k is either the continuous (13.14) or discrete (13.17) control law designed above. The purpose of the v_k term is to cause oscillation, either circling[1] or back-and-forth, in the headings such that the average velocity of *each vehicle* matches the reference velocity. For this reason, we will henceforth refer to this term as the *oscillatory input*.

Constraining the vector of oscillatory inputs, $\mathbf{v} = [v_1, \ldots, v_N]^T$, to preserve the current centroid velocity,

$$\mathbf{v} \in \text{kernel}\left(\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_N\end{bmatrix}\right), \tag{13.19}$$

ensures compatibility with the reference velocity matching inputs. Further, we seek a homogeneous controller in the sense that the oscillatory inputs of two vehicles i and j are exchanged ($v_i \leftrightarrow v_j$) if their headings are exchanged ($\theta_i \leftrightarrow \theta_j$).

For a group of size $N=3$, an oscillatory input meeting all of these requirements is

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \kappa \begin{bmatrix} \sin(\theta_3 - \theta_2) \\ \sin(\theta_1 - \theta_3) \\ \sin(\theta_2 - \theta_1) \end{bmatrix}, \tag{13.20}$$

where κ is a control gain that determines the oscillation frequency. The key element behind this input is a degree of freedom that exists in the space of headings for which the centroid velocity matches the reference velocity. The heading of the first vehicle can be placed (almost) arbitrarily, after which the headings of the other two vehicles are determined by the velocity matching constraint ($\bar{\mathbf{x}} = \mathbf{x}_{\text{ref}}$). The oscillatory input (13.20) walks the headings through this one-dimensional space in a symmetric manner, guaranteeing that the average velocity of each vehicle matches the reference velocity. A similar controller can be developed for $N > 3$, but such a discussion is beyond the scope of this chapter.

[1] The circling oscillatory mode does not produce circular trajectories relative to the target. Such circular orbits are incompatible with velocity matching. Instead, by circling we mean that the heading winds around once per period.

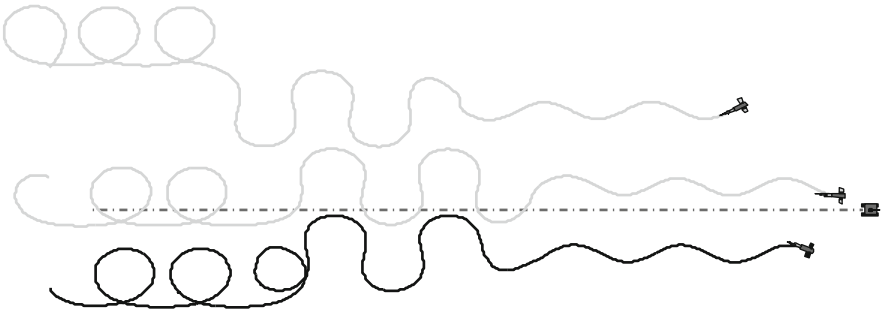


Fig. 13.4 The target speed ratio was increased from 30 to 50% at 60 s, and then further increased to 95% at 100 s. Without changing the controller, the oscillations switch from a circling mode to a back-and-forth mode

While previous studies have considered heuristic circling or back-and-forth motions to manipulate the long-time average pursuit vehicle velocity [6, 11], the oscillatory input (13.20) guarantees the average velocity of each vehicle will match that of the target. A direct consequence of this motion is an automatic transition from circling to back-and-forth motion as the target speed is increased beyond 33% of the pursuit vehicle speed, as shown in Fig. 13.4. This transition is highly desirable as a circling motion is preferable for slow-moving targets, but breaks down for relatively fast-moving targets. The break down is caused by the fact that the vehicles have a physical turn rate limit, which is certainly violated for a circular trajectory about a fast-moving target. A back-and-forth motion can be straightened to a line to allow tracking of fast-moving targets, even as the speed ratio approaches one.

As with the velocity matching controller, implementing the oscillatory feedback law would require each vehicle to know the precise heading of the other vehicles at every time instant. This data is unavailable as communication operates only at discrete instants of time. Fortunately, there is a simple way to approximate the oscillatory input with a signal that depends only on locally available information. We will use the oscillatory input for the first vehicle as an example, but the technique works for all vehicles by the intentional symmetry of the control design. To begin, geometric arguments reveal that v_1 can be equivalently written as

$$v_1 = \kappa q c_{23} \|\bar{\mathbf{x}}_{23}\|, \tag{13.21}$$

where $q = \text{sign}(v_1) \in \{-1, 1\}$ is the sign of the input, $c_{23} = \|\mathbf{x}_3 - \mathbf{x}_2\|$, and $\bar{\mathbf{x}}_{23} = (\mathbf{x}_2 + \mathbf{x}_3)/2$ is the phasor centroid of $\mathbf{x}_2$ and $\mathbf{x}_3$. The reference velocity matching portion of the controller drives the phasor centroid to the reference vector,

$$\bar{\mathbf{x}} \to \mathbf{x}_{\text{ref}} = \frac{1}{3}(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3), \tag{13.22}$$

so we can assume that these two vectors are equal. The centroid of $\mathbf{x}_2$ and $\mathbf{x}_3$ can then be written in terms of locally available information,

$$\bar{\mathbf{x}}_{23} = \frac{3\mathbf{x}_{\text{ref}} - \mathbf{x}_1}{2}, \tag{13.23}$$

and the distance separating $\mathbf{x}_2$ and $\mathbf{x}_3$ can be found geometrically to be

$$c_{23} = 2\sqrt{1 - \|\bar{\mathbf{x}}_{23}\|^2}. \tag{13.24}$$

The only remaining issue lies in determining the sign in (13.21). For this term, we compute $q = \sin(\theta_3 - \theta_2)$ using the latest information available from communication.

The resulting local controller only approximates its continuous-time counterpart for two main reasons. First is the assumption that the phasor centroid matches the reference vector perfectly. In practice, small errors exist between these vectors during transients caused by initialization and target acceleration. The second source of approximation comes from computing the sign using old data.

Nonetheless, frequency separation using $K \gg \kappa$ and small time increments allow formal verification of the local oscillatory control. The simulation in Fig. 13.4 was computed using the local oscillatory controller with communication updates occurring every $\Delta T = 0.25\,\text{s}$ and gains of $K = 5, \kappa = 0.75$.

13.4.3 Regulating Spatial Errors While Circling

The problem remains that the spatial position of the individual vehicles relative to the target is uncontrolled, as can be seen in Fig. 13.4. For the circling oscillatory mode, spatial errors between the individual vehicles and the target, whose position will be denoted $\mathbf{x}_{\text{tgt}}$, can be regulated by biasing the motion of each vehicle in the direction of an error vector, $\mathbf{e}_k$. This error vector (and the resulting corrective input) should go to zero as the trajectory becomes centered over the target, and thus it cannot come from the instantaneous position difference between the kth vehicle and the target.

Instead, we choose to calculate the error vector e_k at specific state-driven instants. To center the oscillatory trajectory over the target, the error vector should be computed at a point on the centerline of the oscillation. However, this point is difficult to detect for the circling oscillatory mode; so we compute the error when the vehicle is aligned with the target (i.e., when $|v_k|$ is minimal). Thus, the error vector is computed as

$$\mathrm{e}_k = \mathbf{x}_{\text{tgt}} - \mathbf{x}_k, \text{ when } |v_k| \text{ is minimal.} \tag{13.25}$$

This error vector can be driven to zero by proper biasing of the steering input. One such bias can be generated by weighting the input by the dot product of the vehicle velocity with the error vector,

$$\dot{\theta}_k = w_k(u_k + v_k) \tag{13.26}$$

$$w_k = 1 + \min(K_w\|\mathrm{e}_k\|, c)\frac{\mathbf{x}_k \cdot \mathrm{e}_k}{\|\mathrm{e}_k\|}, \quad K_w > 0,\ 0 < c \ll 1. \tag{13.27}$$

Constants K_w and c determine the magnitude of the bias. This weighting causes the vehicle to turn slightly slower when aligned with the error (i.e., when pointed toward the target) and turn faster when anti-aligned with the error.

13.5 Demonstration

To demonstrate the use of the coordinated controller and underwater wireless network, we have implemented the controller on each of the three fin-actuated underwater vehicles. These vehicles are free to swim within a large above-ground pool that has a width and depth of 2.5 m and a length of 6 m. Each vehicle is equipped with a custom-built radio modem operating at 315 MHz.

The vehicles have inner-loop PID controllers for stabilization of heading, depth, and speed to desired values. We use the coordinated controller to set the reference heading for each vehicle and allow the inner loops to reject errors between the current heading, sensed by a magnetic compass, and the reference heading. The vehicles were commanded to swim at different depths to avoid possible collisions.

Because the vehicles are unable to sense their own position, the pool has been instrumented with an underwater computer vision system, see Fig. 13.5. Using parallel particle filters, the vision system uses measurements from four calibrated underwater video cameras to track the position, orientation, and velocity of all three fish robots and the target vehicle. The x-y positions of all vehicles are broadcast at once every $T = 0.35$ s, leaving enough time between broadcasts for the other vehicles to sequentially access the channel. Details on the vision system can be found in [7].

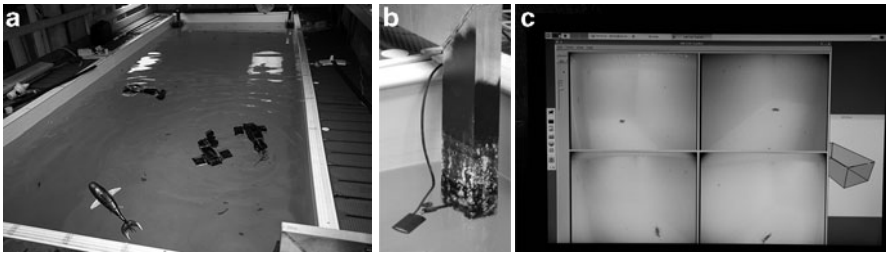


Fig. 13.5 The vision system for tracking four or more simultaneous targets (**a**). Images are captured by four underwater cameras (**b**), and targets are tracked using particle filters (**c**)

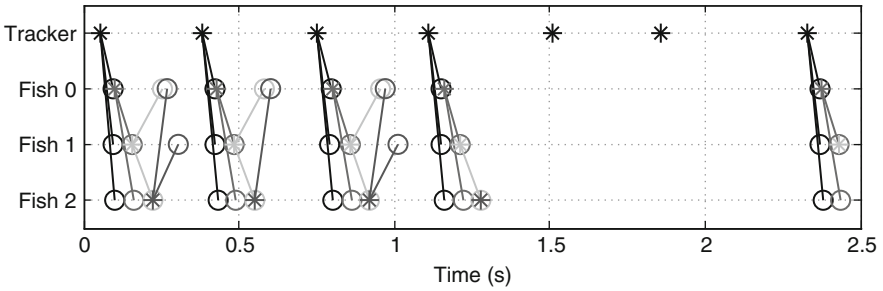


Fig. 13.6 A typical communication sequency from an experiment showing frequent packet loss. Transmissions are marked with asterisks and receptions are marked with circles. The tracker broadcasts every $T = 0.35$ s and is not configured to receive

13.5.1 Packet Loss and Controller Modification

The underwater radios were able to deliver messages in the desired manner; however, many packets failed a checksum test or were lost entirely. The success rate of packet delivery is difficult to quantify as it varied with the relative positions and orientations of the vehicles. A typical communication sequence from an underwater experiment is depicted in Fig. 13.6.

The theoretical guarantees, including Theorem 1 and the discrete time extensions, do not extend to lossy communication, and in fact the controller performed erratically without modification. To add robustness with respect to packet loss, the controller on each vehicle was modified to maintain an estimate of the state of the other two vehicles. These estimates were corrected each time data was received via communication, and the latest estimate was used in determining the steering command.

Another implementation deviation from the controller designed above stems from the broadcast communication topology. In the discrete time system model, all vehicles receive information from all other vehicles before updating their headings. Instead of waiting to receive this information in its entirety, the modified controller updates each time new data arrives. The modified controller exhibits improved stability and robustness in experiment and simulation and will be the subject of future investigation.

13.5.2 Tracking a Constant Reference Vector

To verify the basic operation of the controller with the fish robots, we specified a reference velocity with speed 30% of the pursuit vehicle speed, aligned with the length of the pool. For this experiment, centering of the oscillations over the target was disabled by setting c and K_w to zero (13.27). The control gains were selected as

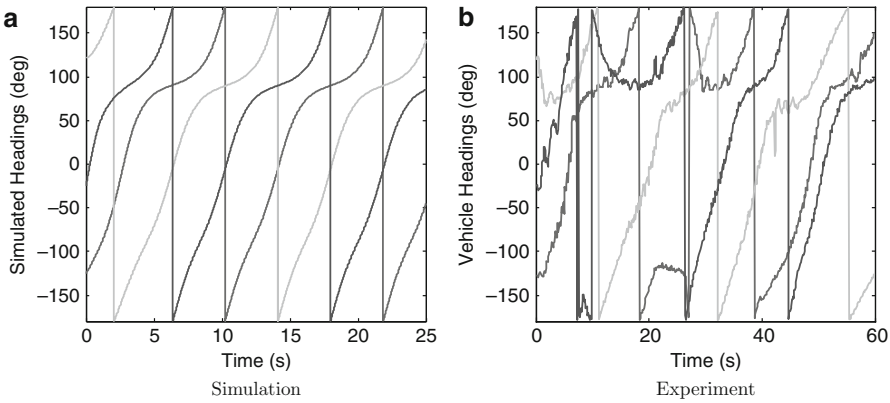


Fig. 13.7 These plots show the commanded headings for the three vehicles as a function of time for a constant reference vector $\mathbf{x}_{\text{ref}} = 0.3\angle 90°$. The simulation (**a**) closely matches data from the robots (**b**) despite numerous lost packets and control modifications

$K = 5, \kappa = 1$, and each complete round of communication took $\Delta T = 0.35$ s (although the control data was updated each time new information arrived). Figure 13.7 shows that the measured vehicle headings closely match data from a simulation in which packet loss was not included.

13.5.3 Target Tracking Simulation

The pool was too small to demonstrate target tracking without the walls significantly affecting the vehicle trajectories, so we resort to simulation. Results from a simulation trial are shown in Fig. 13.8. These results show the effectiveness of the complete discrete-time controller, including the biasing term to bring each individual vehicle to the target. Gains of $K = 0.5$, $\kappa = 0.5$, and $c = 0.2$ were used to generate this data.

13.6 Conclusion

The work in this chapter has discussed the limitations of underwater communication and developed a coordinated target tracking controller compatible with these limitations. Results indicate that cooperative target tracking will be possible despite the loss of a significant number of packets.

A number of the challenges encountered in the coordinated target tracking problem appear in virtually any application of control over a wireless network. One such challenge is that a possibly nonlinear continuous time plant must be controlled using data that arrives at discrete time instants from the communication network. Further, messages are frequently dropped, and the communication topology is

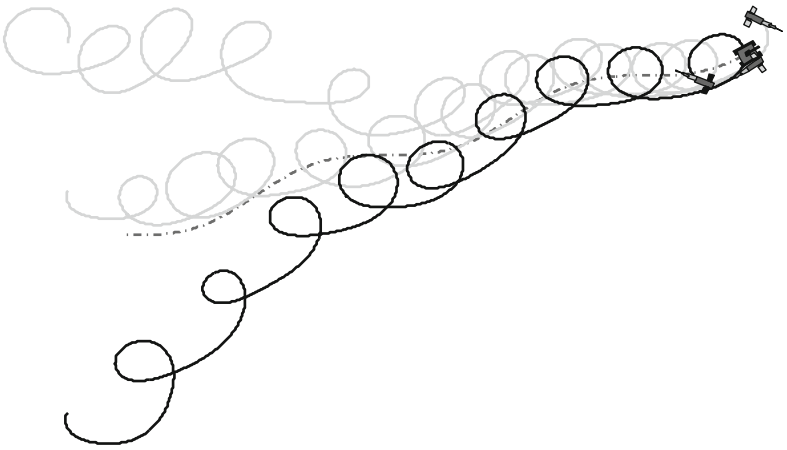


Fig. 13.8 A simulation of three robotic fish robots tracking a target. The key feature here is that the spatial error between each vehicle and the target is driven to zero by the biasing term (13.26)

limited by interference and medium access protocols. With these challenges in mind, applications requiring coordination of low-numbers of vehicles over a wireless network will benefit from having each vehicle estimate the state of the others. While this estimation requires additional computational resources, the benefit of robustness to communication uncertainties is worthwhile.

In future work, we will first study the stability of the coordinated controller with packet drops and updates after each communication broadcast. Next, we will study how the performance of the system is related to the control gains, to select the gains optimally. Finally, we will experimentally demonstrate the complete target tracking system in an appropriately large body of water.

References

1. J. M. Anderson and P. A. Kerrebrock. The vorticity control unmanned undersea vehicle (VCUUV)-An autonomous vehicle employing fish swimming propulsion and maneuvering. In *International Symposium on Unmanned Untethered Submersible Technology*, pages 189–195, 1997.
2. P. K. Bettale. Design of a reliable embedded radio transceiver module with applications to autonomous underwater vehicle systems. Master's thesis, University of Washington, 2008.
3. X. Deng and S. Avadhanula. Biomimetic micro underwater vehicle with oscillating fin propulsion: system design and force measurement. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3312–3317, 2005.
4. C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi. Seaglider: A long-range autonomous underwater vehicle for oceanographic research. *IEEE Journal of Oceanic Engineering*, 26(4):424–436, 2001.
5. E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4):935, 2006.

6. D. Kingston and R. Beard. UAV Splay State Configuration for Moving Targets in Wind. *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, 369:109, 2007.
7. D. J. Klein. *Coordinated Control and Estimation for Multi-agent Systems: Theory and Practice*. PhD thesis, University of Washington, 2008.
8. D. J. Klein, P. K. Bettale, B. I. Triplett, and K. A. Morgansen. Autonomous underwater multivehicle control with limited communication: Theory and experiment. In *Proceedings of the Second IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Killaloe, Ireland, April 2008.
9. D. J. Klein, P. Lee, K. A. Morgansen, and T. Javidi. Integration of communication and control using discrete time Kuramoto models for multivehicle coordination over broadcast networks. In *IEEE Conference on Decision and Control*, pages 13–19, December 2007.
10. D. J. Klein and K. A. Morgansen. Controlled collective motion for trajectory tracking. In *Proc. Amer. Contr. Conf*, 2006.
11. E. Lalish, K. A. Morgansen, and T. Tsukamaki. Oscillatory control for constant-speed unicycle-type vehicles. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5246–5251, 2007.
12. N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. Collective motion, sensor networks, and ocean sampling. *PROCEEDINGS-IEEE*, 95(1):48, 2007.
13. K. A. Morgansen, B. I. Triplett, and D. J. Klein. Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Transactions on Robotics*, 23(6):1184–1199, 2007.
14. U. S. Supreme Court Ruling. Winter v. Natural Resources Defense Council, October 2008.
15. B. I. Triplett, D. J. Klein, and K. A. Morgansen. Discrete time Kuramoto models with delay. *Lecture Notes in Control and Information Sciences*, 331:9, 2006.
16. C. Zhou, Z. Cao, S. Wang, and M. Tan. The posture control and 3-D locomotion implementation of biomimetic robot fish. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5406–5411, 2006.

Index

S.K. Mazumder (ed.), *Wireless Networking Based Control*,
DOI 10.1007/978-1-4419-7393-1,