# Cross-Layer Design and Analysis for Wireless Networks

by

LONG BAO LE
M.Eng., Asian Institute of Technology, 2002

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

Prof. E. Hossain, Supervisor, Dept. of Electrical & Computer Engineering

Prof. A. S. Alfa, Member, Dept. of Electrical & Computer Engineering

Prof. E. Shwedyk, Member, Dept. of Electrical & Computer Engineering

Prof. J. Misic, Outside Member, Dept. of Computer Science

Prof. X. Shen, External Examiner

University of Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION

Cross-Layer Design and Analysis for Wireless Networks

BY

LONG BAO LE

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

DOCTOR OF PHILOSOPHY

**Supervisor:** Professor Ekram Hossain

## ABSTRACT

We develop novel analytical models for radio link level protocol analysis and design for point-to-point transmissions in multi-rate wireless networks. Specifically, error control and scheduling, which are two primary components of any radio link level protocol, are analyzed. Multi-rate transmission is assumed to be achieved by the implementation of adaptive modulation and coding (AMC) in the physical layer. The delay statistics for Go-Back-N and Selective Repeat automatic repeat request (ARQ)-based error control protocols with non-zero feedback delay are derived analytically. We develop queueing models to calculate delay statistics and throughput performances for two classes of scheduling policies, namely, weighted round-robin (WRR) scheduling and opportunistic scheduling schemes. Here, multiple users are assumed to share one channel in a time multiplexing manner. The analytical model for the channel-quality-based scheduling can be applied to any scheduling schemes as long as the evolution of joint service/vacation and channel processes can be determined. As an example, we analyze the max-rate (MR) scheduling scheme. Applications of the analytical models for cross-layer design and packet-level admission control under statistical delay constraints are illustrated. The proposed analytical models provide frameworks to fairly compare different scheduling policies considering different traffic, system and channel parameters.

Besides link level protocol design and analysis in single hop wireless networks, end-to-end protocol design issues for multi-hop wireless networks pose significant research challenges. For the end-to-end transmission scenario, both single-path QoS routing and optimal multi-path routing protocols are developed. In particular, we propose both exact and approximate decomposition approaches to solve a tandem queueing problem considering the implementation of AMC in the physical layer and ARQ-error recovery in the link layer. The decomposition approach is then employed to develop a single-path QoS routing protocol incorporating all important QoS measures, namely, end-to-end bandwidth, average delay, loss rate and statistical delay requirements. For the optimal multi-path routing problem, we employ the dual decomposition approach from convex optimization to develop cross-layer optimization frameworks for multi-hop wireless networks using decode-and-forward cooperative diversity. Specifically, we propose two distributed algorithms capturing functionalities in different layers of the protocols stack, namely the relay selection and power allocation in the physical layer, routing in the network layer and congestion control in the transport layer. The convergence of the proposed algorithms and the significant gains in terms of power consumption and transmission rates due to the cooperative diversity are illustrated through typical numerical results.

The proposed analytical models and protocols in this dissertation provide important frameworks for cross-layer design and optimization in wireless networks. These frameworks exploit different network degrees of freedom such as link adaptation, multiuser diversity, cooperative diversity, etc. They also solve research challenges due to the decentralized architecture of future wireless networks.

Examiners:

Prof. E. Hossain, Supervisor, Dept. of Electrical & Computer Engineering

Prof. A. S. Alfa, Member, Dept. of Electrical & Computer Engineering

Prof. E. Shwedyk, Member, Dept. of Electrical & Computer Engineering

Prof. J. Misic, Outside Member, Dept. of Computer Science

Prof. X. Shen, External Examiner

# Table of Contents

# List of Figures

# Lists of Abbreviations

| | |
|---|---|
| **ACK** | Acknowlegment |
| **AMC** | Adaptive Modulation and Coding |
| **AP** | Access Point |
| **ARQ** | Automatic Repeat reQuest |
| **ASTA** | Arrival See Time Average |
| **BG** | Broadcast Group |
| **BMAP** | Batch Markovian Arrival Process |
| **BS** | Base Station |
| **BW** | Bandwidth |
| **CDMA** | Code Division Multiple Access |
| **DF** | Decode-and-Forward |
| **FDMA** | Frequency Division Multiple Access |
| **FSMC** | Finite State Markov Channel |
| **GBN-ARQ** | Go-Back-N ARQ |
| **GPS** | Generalized Processor Sharing |
| **CSI** | Channel State Information |
| **HOL** | Head Of Line |
| **MAC** | Medium Access Control |
| **MC** | Markov Chain |
| **MGM** | Matrix Geometric Method |
| **MR** | Max Rate |
| **NACK** | Negative Acknowledgment |
| **OFDM** | Orthogonal Frequency Division Multiplexing |
| **PER** | Packet Error Rate |
| **QAM** | Quadrature Amplitude Modulation |
| **QBD** | Quasi-Birth and Death |

| | |
|---|---|
| **QoS** | Quality of Service |
| **RRP** | Route Reply |
| **RRQ** | Route ReQuest |
| **SNR** | Signal to Noise Ratio |
| **SR-ARQ** | Selective Repeat ARQ |
| **TDM** | Time Division Multiplexing |
| **TDMA** | Time Division Multiple Access |
| **WLAN** | Wireless Local Area Network |
| **WRR** | Weighted Round Robin |

# *Acknowledgement*

I would like to express my profound gratitude and appreciation to Professor Ekram Hossain for his continuous encouragement, support, understanding. His endless enthusiasm has created a lot of motivation for me during the graduate studies. I really appreciate his prompt response and wonderful suggestions on every single work of my research. I hope that I could continue receiving his support and advice in my future career. I would like to thank Professor Attahiru Sule Alfa for his good course, nice comments and suggestions on many research issues in the thesis.

I am grateful to Professor Edward Shwedyk for serving in my examination committee and for his interesting lectures. I really love Professor Shwedyk's teaching style and the way he visualizes and interprets complicated concepts by simple examples. Thanks to Professor Jelena Misic and Professor Xuemin (Sherman) Shen for their time and effort to serve in my thesis committee.

I thank Professor Tho Le-Ngoc for his great mentoring and support. I am indebted to Professor Michele Zorzi for his excellent comments and suggestions on ARQ modeling issues. My joint work with him has led to results in chapter 3 of the thesis.

I would like to thank all my friends at University of Manitoba for their cooperation, friendship and sharing which make the period of my graduate studies interesting and memorable.

Special thanks to my dear parents and family members for their loves, support and encouragement. Last but not least, I would like to thank my wife, Trang Cao, for her love, sacrifice and understanding which have been the tremendous source of motivation for me to finish my graduate works.

# Chapter 1

# Introduction

Wireless technologies have seen unprecedented advancements over the past decade. Wireless cellular and wireless local area networks (WLAN) have been deployed rapidly all over the world [1]-[2]. There are urgent demands for ubiquitous wireless services with higher data rates and more diverse quality of service (QoS) requirements. These requirements have resulted in several design paradigm shifts for future wireless networks. First, different wireless systems need to inter-operate with each other to enhance transmission rates, coverage, and QoS performances. Here, how to share the frequency spectrum efficiently while maintaining desired QoS requirements for users of different wireless systems is a challenging research problem. Second, implementation of decentralized architectures will result in more flexibility in designing future wireless networks and can potentially increase the network capacity. Mobile nodes may help their partners to forward data packets to the desired destinations (e.g., base station, internet). In fact, the *multi-hop cellular concept* has been proposed recently in the literature [3], [4], [5]. Wireless mesh networks are emerging as the promising technologies for future wireless networks [6], [7]. The mesh architecture can be implemented for cellular and WLAN networks where each base station (BS) or access point (AP) can serve a router to forward data to the destinations. Third, discovering and exploiting different network degrees of freedom are the keys to enhance data rates and QoS performances in wireless networks.

## 1.1 Scope of This Dissertation

This dissertation evolves around solving research problems due to the above mentioned trends and challenges. We investigate different network degrees of freedom in different layers of the protocol stack and develop several cross-layer analytical models or protocols as follows.

- **Physical layer:** A physical layer employing link adaptation technique is considered in this dissertation. Link adaptation can be achieved by using a finite number of transmission modes each of which corresponds to one particular modulation and coding scheme. The cooperative diversity concept in the physical layer is also investigated which can significantly decrease the total power consumption and/or increase the transmission rates of wireless nodes.
- **Link layer:** Implementation of Automatic Repeat Request (ARQ) protocol to retransmit erroneous packets over a multi-rate wireless link is considered in different analytical models and protocols. For the multi-user case, different wireless scheduling schemes are assumed to allocate transmission opportunities for active users.
- **Network layer:** For a multi-hop transmission scenario, an efficient routing protocol is the key to ensure successful end-to-end packet delivery. A solution to the QoS routing problem has been provided in this dissertation.
- **Transport layer:** End-to-end rate control is important to avoid congestion in multi-hop wireless networks and guarantee desired end-to-end QoS performance. From this perspective, the joint congestion control, routing and resource allocation is crucial to optimize performance of wireless networks. In this dissertation, this problem has been tackled from a cross-layer optimization point of view.

Due to the existence of diverse techniques and numerous parameters in different layers, cross-layer design would play a key role in enhancing system performance [8]. In this dissertation, we consider protocol design and analysis at one particular BS/AP/router and research issues related to end-to-end transmission in decentralized architecture of wireless networks. In the sequel, we will refer to the considered BS/AP/router simply as a BS (i.e., the proposed analytical models/protocols are also applicable for an AP in WLAN systems or a router in mesh networks). All the proposed analytical models and protocols capture the link adaptation technique in the physical layer with either discrete or continuous rate adaptation.

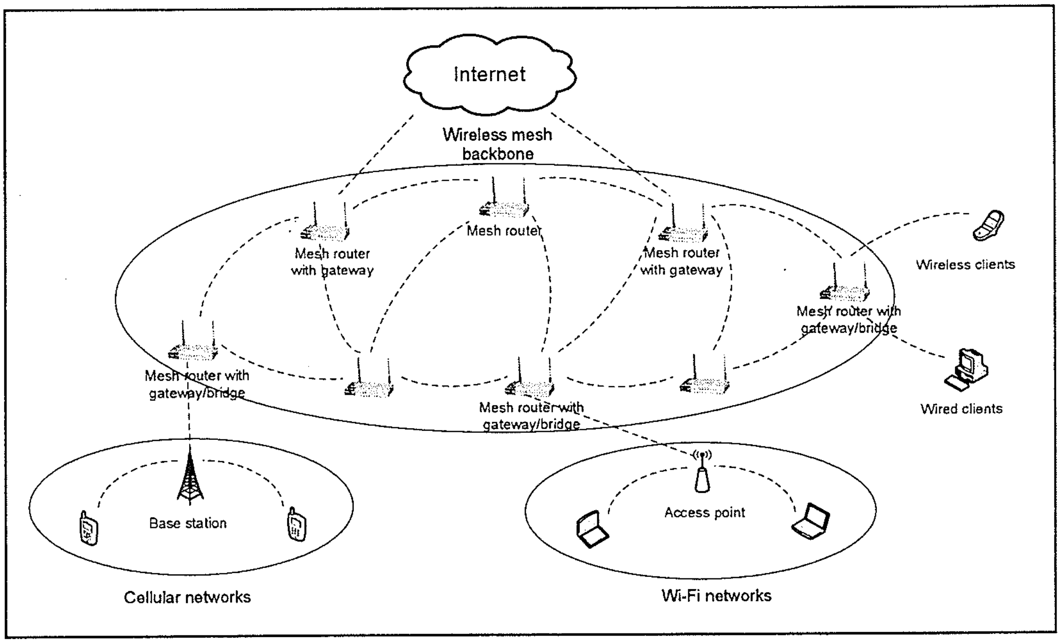


**Figure 1.1.** *Infrastructure/backbone wireless mesh network.*

The infrastructure/backbone wireless mesh network is illustrated in Fig. 1.1 where different wireless systems inter-operate with each other [6]. In this figure, point-to-point design issues correspond to communications between each BS/AP/router with its mobile users while multi-hop transmissions are involved in the end-to-end communications among BSs/APs/routers to deliver data to/from internet.

## 1.2 Literature Review

### 1.2.1 Link Adaptation and ARQ Protocol Modeling

Achieving high-speed transmission and provisioning of quality of service for emerging data-oriented wireless applications through intelligent and flexible radio resource management are the key challenges for future-generation wireless networks. In the physical layer, adaptive modulation and coding (AMC) has been adopted to increase the transmission rate in most of the 2.5/3G wireless systems [9]-[21]. Most of the emerging data applications are somewhat delay-tolerant which creates more opportunities and flexibility for radio access systems design. The implementation of automatic repeat request (ARQ) in the link layer is very efficient to eliminate the residual error and to alleviate the costly use of a strong error correction code in the physical layer [22]-[29].

The employment of link adaptation techniques through adaptive modulation and coding (AMC) to enhance the spectral efficiency is very common in all 2.5/3G wireless systems [9]-[14]. Depending on the channel quality, the transmission mode at the transmitter is adapted accordingly. For implementation, the receiver estimates the channel quality and transmits this channel state information (CSI) to the transmitter to choose the suitable transmission mode (i.e., a pair of modulation and coding scheme). In practice, the performance degrades due to feedback delay and error. The impacts of feedback error and delay on the bit error rate performance of AMC schemes were investigated in [15],[16]. In [20], queueing analysis of a wireless system using AMC with finite buffer was presented. The authors, however, considered only a single user case without ARQ in the link layer.

Performance evaluation of different ARQ protocols and the design of scheduling techniques have often been treated as two separate problems in the literature. Analysis of ARQ protocols for a two-state Markov channel and an independent channel were performed in [22]-[25] and [29], respectively. The throughput derived under the two-state Markov channel was observed to approximate well the actual throughput [24]. However, the reliability of this two-state channel model for analyzing the delay performance was not investigated.

The derivation of delay distribution was first given in [23] for both ideal (no feedback delay) and Go-Back-N ARQ protocols. In [22], the performance of ARQ protocols under different error control codes were investigated. All of these analysis were performed under the two-state Markov channel which cannot capture the multi-rate transmission being employed in most modern wireless systems. The spectral efficiency of a truncated ARQ protocol with AMC in the physical layer was derived in [21] but the delay analysis was not pursued. The analytical models for ARQ protocols with AMC in the physical layer under zero and non-zero feedback delay were conducted in [30], [31], respectively.

### 1.2.2 Wireless Scheduling

Implementation of a scheduler at the BS is important for flexible radio resource allocation among multiple users to satisfy their QoS requirements and at the same time improve the utilization of the system resources by exploiting radio channel specific features such as multiuser diversity [32]-[39]. The key differences in implementation of a scheduling policy in uplink and downlink directions results from the exchange of necessary information to make the scheduling decision. A wireless scheduler is usually located at the base station (BS) while the data buffers are at the BS for downlink direction and at the mobiles for uplink direction, respectively. The implementation of a wireless scheduler at one BS in the downlink direction is illustrated in Fig. 1.2.

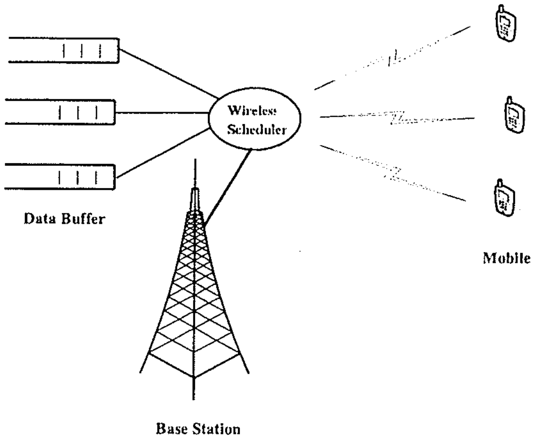


**Figure 1.2.** *Scheduling in downlink direction.*

Most of the wireless scheduling schemes developed in the literature aim at maximizing system throughput while meeting different performance objectives such as fair allocation of system throughput [34]-[39], access time [40] or guaranteeing packet delivery delay and packet dropping probability [41]. A good review of the different scheduling schemes for wireless networks can be found in [42], [44]. Some of the recently proposed 'opportunistic' scheduling policies take advantage of the multiuser diversity gain inherent in the wireless channel dynamics by exploiting the relatively independent channel fluctuations of different users to increase the system throughput.

All of the above works mainly focus on constructing a scheduling rule under certain predefined design objectives such as maximizing the throughput while providing the fairness between different traffic flows. Because of these design goals, it is generally assumed that the buffers of all the backlogged flows are saturated, and therefore, the buffer dynamics and delay behavior are not investigated.

In [45], the mean delay of a polling wireless system with ARQ under two-state Markov channel was analysed. The worst-case performance analysis for the well-known generalised processor sharing (GPS) was done in [46] where the arriving traffic is shaped by a leaky bucket. The statistical delay bound of the GPS scheduling was derived in [47] when the burtiness of the traffic source is bounded. Enhanced versions of the traditional weighted round robin scheduling were proposed in [48], where the implementation complexity was reduced compared to the previously proposed weighted fair scheduling schemes.

Some of the recent works on opportunistic scheduling are as follows. The scheduler proposed in [35] provides fairness among users by using a compensation model to tackle the lagging and leading effects due to bursty channel errors. In [36], the proportional fair scheduling was proposed, which can ensure asymptotically fair allocation of access time among the users and it exploits the multiuser diversity from the channel dynamics.

The use of multiple transmit antennas to induce the large and fast channel fluctuations to enhance the multiuser diversity gain along with the proportional fair scheduling was proposed in [37]. In [40], a GPS-like scheduling scheme was proposed for code division multiple access (CDMA) systems. A simple delay bound for this fair scheduling scheme was derived when the incoming traffic is shaped by a leaky bucket. Another asymptotically fair scheduling was suggested in [34] and the corresponding scheduling gains in single- and multi-cell environments were obtained.

Liu et al. proposed an opportunistic scheduling policy in [38] which ensures fairness in access time among users while maximizing the average system performance. A credit-based fair queueing scheme with a guaranteed statistical fairness bound was proposed in [39]. The tradeoff between throughput and fairness of this scheme can be varied to meet the desirable fairness level. In [49], an optimal scheduling policy was derived taking the burstiness in the traffic arrival process into account. However, the authors assumed a simple on-off channel model and considered single-rate transmission only. Also, the analysis for delay was not performed. A simple queueing analysis was given in [50] for a single-user scenario.

The queueing analysis for a general radio link level scheduling rule taking multi-rate transmission and ARQ-based error recovery into account is a very challenging problem. It is also crucial for fair comparison of different scheduling schemes, and after all, for radio link control design and engineering in wireless systems. In [51], [52], we have derived delay statistics and throughput performance for weighted round robin and opportunistic scheduling schemes considering AMC in the physical layer and ARQ in the link layer, respectively.

### 1.2.3 Single-path QoS Routing and Optimal Multi-path Routing for Multi-hop Wireless Networks

Routing protocol is an important component of multi-hop wireless networks whose responsibility is to find multi-hop routes from source nodes to destination nodes [53], [54]. Most of routing algorithms in the literature find the routes for each incoming connections based on the minimum hop count without any QoS guarantee. QoS routing, on the other hand, is an important subclass of routing algorithms where some specific end-to-end QoS requirements must be satisfied [54]-[57]. Routing algorithms can also be classified as being of either single-path [58]-[66] or multi-path type [67]-[69]. Although multi-path routing usually offers better load balancing, it incurs more overhead. Also, compared to single-path routing, it is more difficult to provide QoS assurance for multi-path routing.

One important component for any routing algorithm is the route discovery task, where good routes from the source node to the destination node are to be found for data delivery. For route discovery, each node maintains a routing table which contains the routes to all other nodes in the network. The route update is performed periodically to keep track of changes in the network topology and traffic load in the network [59]. To reduce control overhead and memory requirements, several hierarchical routing schemes were proposed in the literature [60]. Hierarchical routing schemes basically group wireless nodes into clusters where intra-cluster and inter-cluster routes are found separately in different hierarchical levels of the routing architecture. For on-demand routing algorithms, route discovery is only performed when there is a demand to establish a route for an incoming connection [62]-[66]. Since on-demand routing scales well to the network size, most of the routing algorithms adopted by the IETF's MANET working group belong to this routing category.

For on-demand single-path QoS routing algorithms, link and path quality metrics are incorporated into the route discovery phase to find good routes for an incoming connection. Two most popular QoS metrics for existing routing algorithms are bandwidth and delay [54]. In [55] and [57], the authors assumed that link delay can be estimated/measured with some uncertainty. A recent work in the literature incorporates the retransmission effect due to an ARQ protocol into the link metric [64]. This work, however, did not consider any end-to-end QoS guarantee for incoming connections. In [65], we developed tandem queue and single-path QoS routing frameworks for multihop wireless networks. The decomposition queueing approach was used to calculate all link QoS metrics accurately.

Optimal multi-path routing is another important class of routing protocols for multi-hop wireless networks. Optimal routing is applicable for splittable traffic where data from the source node are split into multiple flows which follow different routing paths to reach the destination [117]. Recently, the dual decomposition technique of nonlinear optimization has been shown to be a useful tool to construct distributed optimal routing algorithms [70]. This technique can also be used for cross-layer design where the master problem can be decomposed into several subproblems corresponding to different layers of the protocol stack [76], [77], [78]. Among these subproblems, link layer subproblem is usually the bottleneck of the whole problem [78].

### 1.2.4 Cooperative Diversity

Cooperative diversity has received significant attention recently as an efficient way to exploit diversity in a wireless network via a virtual distributed antenna array where each antenna belongs to a different node [79]-[81]. Cooperative diversity was initially intended/proposed for centralized wireless systems such as cellular systems. Relaying problem, however, is also a fundamental one in multi-hop wireless networks [82], [83] which is still an open research issue.

In fact, the decode-and-forward and amplify-and-forward cooperative protocols were first proposed in [79], [80] for cellular networks. Outage analysis for these two cooperative protocols was conducted in [81]. The ergodic capacity was obtained for several cooperative and relay strategies in [84]. In [85], coded cooperation protocols were proposed.

Recently, some initial efforts have been put on higher layer protocol aspects of wireless networks using cooperative diversity. Cooperative MAC protocols were provided in [86], [87]. The key idea behind cooperative MAC problem is to opportunistically exploit relays for transmission if relaying achieves higher throughput than that due to direct transmission. In [83], the authors proposed a MAC protocol for relay selection working along with a single-path routing protocol. In [90], we proposed an analytical model to quantify end-to-end performance for a general ARQ cooperative diversity scheme.

In [88], the cooperative routing problem was formulated as a dynamic programming problem. Cooperative diversity was used for multicasting in wireless ad hoc networks in [89]. These works considered transmission of a single packet to a destination node and these models are difficult to extend for distributed implementation. Also, the incorporation of congestion control seems to be difficult for these models, which is important in distributed networks. In [91], we developed cross-layer frameworks for multihop wireless networks using decode-and-forward cooperative diversity.

## 1.3 Motivations and Contributions of This Dissertation

As mentioned before, we consider both protocol analysis and design at one particular BS/AP/router (i.e., point-to-point transmission) and design issues for end-to-end transmission scenarios. For the design problems at one BS/AP/router, we consider two cases. The first case applies for a transmission scenario where each mobile user is allocated an orthogonal channel to communicate with its BS (e.g., one orthogonal channel corresponds to an orthogonal code in CDMA systems or simply a frequency channel in FDMA systems). We will refer to this case as a single-user scenario due to the inherent separation of different users. In the second case, multiple users share one single channel in time multiplexing manner as in CDMA-TDMA or FDMA-TDMA wireless systems. Here, we assume a wireless scheduler at one specific BS to allocate transmission time slots for different users. Two classes of scheduling schemes are investigated: weighted round robin and opportunistic scheduling schemes.

For the single-user scenario, we develop an analytical model to calculate delay statistics for two ARQ protocols, namely Go-Back-N (GBN) and Selective Repeat (SR) ARQ protocols, under non-zero feedback delay when adaptive modulation and coding (AMC) is implemented in the physical layer. The multi-rate transmission due to AMC over Nakagami-$m$ is modeled by the Finite State Markov Channel (FSMC) model. Existing analytical models for these ARQ protocols in the literature mostly assumed zero feedback delay and/or two state Markov channel model. In the two state channel model, there is one good state and one bad state and only one packet can depart from the queue if the channel is in the good state. This assumption cannot capture the multi-rate feature due to the implementation of AMC in the physical layer as in most 3G and beyond wireless networks [9]-[19]. In [28], the authors assumed an N-state Markov channel but only one packet can depart from the queue in one time slot; thus, it did not take the multi-rate feature into account.

For the multi-user case where a wireless scheduler is employed at the BS, existing works in the literature mostly assumed saturated buffers where all active users always have packets to transmit. In practice, data buffers are not always saturated if the traffic intensity is light. In fact, admission control should be performed at both connection and packet levels in such a way that QoS performances such as overflow probability and link level delay remain below target levels. Specifically, the admission control operation renders the assumption of saturated buffers invalid. Also, the actual system performances are different from those derived under the saturated buffer condition. Therefore, investigation of queueing performances for different scheduling schemes is very important to determine the achievable throughput and delay performances under different channel, system, traffic conditions. We develop queueing models for two classes of scheduling schemes, namely, weighted round robin and opportunistic scheduling schemes in this dissertation.

For the end-to-end transmission scenario, we develop both QoS routing and optimal routing algorithms. In particular, a tandem queueing and QoS routing framework is proposed where the route discovery is conducted by using decomposition tandem queueing model. Most QoS routing protocols in the literature assumed that link QoS metric such as delay can be estimated and stored at each wireless node and packet loss due to buffer overflow was usually ignored. The tandem queueing model provides an efficient tool to accurately calculate these QoS metrics including end-to-end delay statistics which were not considered before. In addition, we construct the joint optimal routing and cooperative diversity framework using the dual decomposition approach from nonlinear optimization. Then, we extend the framework to capture the congestion control function in the transport layer through maximizing the power and rate utility tradeoff. In fact, employment of cooperative diversity in the physical layer results in significant system gains in terms of power consumption and/or transmission rates.

The main contributions of this dissertation can be summarized as follows:

- Queueing models for GBN-ARQ and SR-ARQ protocols are developed assuming non-zero feedback delay and AMC implementation in the physical layer. Both queue length and delay distributions are derived analytically. The developed models provide a guideline to tune the system parameters such that good radio link level delay performance can be achieved. They also provide a tool to quantify the tradeoff between performance and implementation complexity of the two ARQ protocols.
- Exact distributions for queue length and packet delay are derived for two classes of scheduling policies: weighted round robin (WRR) scheduling and opportunistic scheduling schemes. The analytical model for the opportunistic scheduling schemes can be applied to any scheduling rule as long as the evolution of a joint service/vacation and channel processes can be captured by a corresponding probability transition matrix. These analytical models capture the implementations of an ARQ protocol in the link layer and AMC in the physical layer. Based on the analysis of both scheduling classes, we derive user throughput under both saturated and non-saturated (i.e., dynamic) buffer conditions. Application of the analytical framework to the max rate (MR) scheduling scheme is given as a specific example. The usefulness of the presented analysis is then illustrated through a cross-layer design example and packet-level admission control under statistical delay constraints for the MR scheduling policy.
- Tandem queueing and QoS routing framework is developed which is applicable

for different technologies in the physical layer. In particular, both exact and approximate decomposition approaches to solve the tandem queueing problem are proposed. The decomposition approach is then employed to construct QoS routing protocols. The tandem queueing model allows different end-to-end QoS requirements, namely, end-to-end bandwidth, delay, and loss, to be incorporated into the QoS routing algorithms. The approximate end-to-end delay distribution is also derived which can be used to discover routing paths with a statistical delay constraint.

- Optimal routing routing protocol is developed for multi-hop wireless networks using cooperative diversity. Specifically, traffic in each wireless link can be transmitted by either direct transmission or cooperative transmission with a relay node by using decode-and-forward cooperative diversity scheme. We construct the joint optimal routing and cooperative resource allocation algorithm which minimizes the total power consumption in the network. Then, the cross-layer framework is extended to capture congestion control in the transport layer. In particular, the joint congestion control, routing, and cooperative resource allocation is developed which optimizes the power and rate utility tradeoff. For both the algorithms, the master problems can be decomposed into multiple subproblems in different layers which are coupled through the corresponding prices.

## 1.4 Organization of This Dissertation

The remaining of this dissertation is organized as follows:

- Chapter 2 summarizes the mathematical background which will be used in the subsequent chapters. The background covers link adaptation technique, modeling of Nakagami-$m$ wireless channels, queueing and nonlinear optimization techniques.
- In Chapter 3, analytical models of GBN and SR-ARQ protocols are developed for multi-rate wireless networks. The models are used for cross-layer optimization and quantification of the tradeoff between delay performance and implementation complexity of the two ARQ protocols.

- Chapter 4 presents the analysis for WRR scheduling. The queue length and delay distributions are derived exactly under the discrete Batch Markovian Arrival Process (BMAP), which can capture burstiness in the arrival traffic.
- The analytical framework for opportunistic scheduling schemes is presented in Chapter 5. Again, the exact queue length and delay distributions are derived for the general scheduling scheme where the evolution of joint service/vacation and channel processes can be determined.
- In Chapter 6, we present a tandem queueing and QoS routing framework for multi-hop wireless networks. Both exact and approximate decomposition approaches to solve the tandem system of queues as well as the employment of the decomposition approach to construct QoS routing protocols are described.
- Chapter 7 presents two algorithms for cross-layer optimization in multi-hop wireless networks using cooperative diversity.
- In Chapter 8, the main results are summarized and a few directions for future research are outlined.

## 1.5 Notations

We will denote matrix and vector as boldface letters and scalars as regular letters. We denote $\mathbf{1}_m$ as a column vector of all ones with dimension $m$. When no subscript is used, the dimension of vector $\mathbf{1}$ depends on the corresponding context. We abuse the notation by using the same symbols to denote similar quantities in different chapters. For example, $\mathbf{A}_i$ and $\mathbf{x}_i$ are used to denote the transition probabilities and steady-state probabilities for Markov chains in different chapters. However, the notations should be unambiguous from the contexts that they are used in.

# Chapter 2

# Mathematical Background

In the chapter, some important mathematical fundamentals which are used in the subsequent chapters are briefly presented. They include link adaptation technique, channel modeling, traffic modeling and a quick review of the matrix-geometric queueing method and the subgradient algorithm of convex optimization.

## 2.1 Link Adaptation Technique

For analytical models and protocols presented in subsequent chapters, link adaptation technique, which adapts the transmission rate with the signal to noise ratio (SNR), is assumed at the radio link level. Either continuous or discrete rate adaptation is assumed. In particular, under the Nakagami-$m$ fading assumption, the probability density function (p.d.f) of the received SNR $x$ can be written as follows [19]:

$$p_X(x) = \frac{m^m x^{m-1}}{\overline{x}^m \Gamma(m)} \exp\left(-\frac{mx}{\overline{x}}\right) \tag{2.1}$$

where $\overline{x} = E\{x\}$ is the average SNR, $\Gamma(m) = \int_0^\infty t^{m-1} \exp(-t)dt$ is the Gamma function, and $m$ is the Nakagami fading parameter ($m \geq 1/2$). The Nakagami fading model is very general and it includes the Rayleigh fading model as a special case when $m = 1$ and the Ricean fading can also be approximated by the Nakagami model [113].

For continuous rate adaptation, the achievable rate in terms of b/s/Hz can be written as

$$r = \log_2(1 + \frac{x}{G}) \tag{2.2}$$

where $x$ is SNR, $G$ denotes the gap to capacity. For wireless systems using M-QAM without coding, $G \approx -\log(5\text{BER})/1.5$, where BER is the bit error rate and log denotes the natural logarithm [17].

For discrete rate adaptation, the received SNR $x$ is partitioned into finite number of intervals. Let $X_0(=0) < X_1 < X_2 < \cdots < X_{K+1}(=\infty)$ be the thresholds of the received SNR for different channel states. The channel is said to be in state $k$ if $X_k \leq x < X_{k+1}$ $(k = 0, 1, 2, \cdots, K)$. In the subsequent chapters, adaptive modulation (with or without coding) using QAM modulation is assumed. For example, in channel state $k$, modulation scheme $2^k$-QAM $(k = 1, 2, \cdots, K)$ is chosen. We assume that in state 0, no transmission is allowed to avoid high probability of transmission errors.

The receiver is assumed to decode the received data using the maximum likelihood decoding technique [111]-[112]. For coded systems, the closed-form derivation for the *packet error rate* (PER) is not easy, so we use the following approximation as in [21]:

$$\mathrm{PER}_k(x) \approx \begin{cases} 1, & \text{if} \quad 0 < x < X_{pk} \\ a_k \exp(-g_k x), & \text{if} \quad x \geq X_{pk} \end{cases} \tag{2.3}$$

where $a_k$, $g_k$ and $X_{pk}$ are obtained by fitting the actual PER curve [21]. In fact, similar approximations for bit error rate (BER) have been used in the literature [18]-[19]. The average PER for mode $k$ can be written as follows:

$$\begin{aligned} \overline{\mathrm{PER}}_k &= \int_{X_k}^{X_{k+1}} a_k \exp(-g_k x) p_X(x) dx \\ &= \frac{1}{\Pr(k)} \frac{a_k}{\Gamma(m)} \left(\frac{m}{\overline{x}}\right)^m \frac{\Gamma(m, b_k X_k) - \Gamma(m, b_k X_{k+1})}{(b_k)^m} \end{aligned} \tag{2.4}$$

where $k = 1, \cdots, K$, $b_k = m/\overline{x} + g_k$ and $\Pr(k)$ is the probability of channel state $k$ which can be calculated as [19]

$$\Pr(k) = \int_{X_k}^{X_{k+1}} p_X(x) dx = \frac{\Gamma(m, mX_k/\overline{x}) - \Gamma(m, mX_{k+1}/\overline{x})}{\Gamma(m)}. \tag{2.5}$$

where $\Gamma(m, x) = \int_x^\infty t^{m-1} \exp(-t) dt$ is the complementary incomplete Gamma function.

The packet error rate calculation presented above essentially models the physical layer of a multi-rate wireless network. The remaining work is to determine the SNR thresholds $X_k$. There exist several ways to do this in the literature. The authors in [92] and [93] calculate these thresholds such that all channel states have the same probability. In [94], the SNR thresholds are found to make the average time of all states equal. In [20], the authors proposed a searching algorithm to find the

SNR thresholds while constraining $\text{PER}_k = P_0$. The SNR thresholds chosen by this algorithm allows us to guarantee the PER in the physical layer and also allows us to perform cross-layer design. The algorithm is as follows:

**Algorithm 2.1: SNR thresholds search algorithm**

1. Set $k = K$ and $X_{K+1} = \infty$.
2. For each $k$, search for $X_k \in [0, X_{k+1}]$ such that $\text{PER}_k = P_0$.
3. If $k > 1$, go to 2, otherwise go to 4.
4. Set $X_0 = 0$.

Here, the design parameter is $P_0$ which will be obtained to achieve the desired system performance.

## 2.2 Channel Modeling

In this section, we describe channel modeling for multi-state wireless channels. Throughout the dissertation, we assume that the channel state remains static in one time slot and may change in consecutive time slots. One of two channel models will be assumed. The first one is an independent and identically-distributed (i.i.d.) channel. For the i.i.d. channel, the channel state in each time slot is one of the possible channel states with probability calculated as in (2.5) and it is independent of that in the previous time slots.

The second channel is the one which captures correlation among channel states in consecutive time slots and is called a Finite-State Markov Channel (FSMC) model. In fact, the Markov channel with one good and one bad channel state was first proposed by Gilbert and Elliott in [95]-[96] which is a special case of FSMC channel. Zorzi et al. [24] showed how to calculate the parameters and they also validated the model. The extension to a multiple state model was proposed in [92] and was validated in [97]. The calculation of FSMC model parameters for a slow Nakagami-$m$ fading channel was presented and validated in [98]. Let $f_d$ be the Doppler shift and $T_s$ be the time slot interval. The i.i.d channel model can be applied when the normalized fading rate $f_dT_s \approx 1$ while the FSMC channel model can be used for slow fading channels with $f_dT_s << 1$.

The FSMC channel model is described by a probability transition matrix for the channel states $\mathbf{T}$ whose elements $\mathbf{T}_{i,j}$ denote the transition probability from state $i$ to state $j$. These probabilities can be approximated as follows [92]:

$$\begin{aligned} \mathbf{T}_{k,k+1} &\approx N_{k+1}T_s/\Pr(k), \quad k = 0, 1, 2, \cdots, K-1 \qquad (2.6)\\ \mathbf{T}_{k,k-1} &\approx N_k T_s/\Pr(k), \quad k = 1, 2, 3, \cdots, K \qquad (2.7) \end{aligned}$$

where $N_k$ is the level crossing rate evaluated at $X_k$, $T_s$ is the time slot interval, and $\Pr(k)$ is the stationary probability of state $k$. The value of $N_k$ can be calculated as follows [20]:

$$N_k = \sqrt{2\pi \frac{mX_k}{\overline{x}}} \frac{f_d}{\Gamma(m)} \left(\frac{mX_k}{\overline{x}}\right)^{m-1} \exp\left(-\frac{mX_k}{\overline{x}}\right). \qquad (2.8)$$

The remaining probabilities are obtained as follows:

$$\begin{aligned} \mathbf{T}_{k,k} &= 1 - \mathbf{T}_{k,k+1} - \mathbf{T}_{k,k-1}, \quad 1 \leq k \leq K-1 \\ \mathbf{T}_{0,0} &= 1 - \mathbf{T}_{0,1} \\ \mathbf{T}_{K,K} &= 1 - \mathbf{T}_{K,K-1} \\ \mathbf{T}_{k,l} &= 0, |k-l| > 1 \end{aligned}$$

where the transitions are only allowed among neighboring states [92]-[98]. The channel transition probability matrix, therefore, can be written as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{0,0} & \mathbf{T}_{0,1} & \cdots & \cdots & 0 \\ \mathbf{T}_{1,0} & \mathbf{T}_{1,1} & \mathbf{T}_{1,2} & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \mathbf{T}_{K-1,K-2} & \mathbf{T}_{K-1,K-1} & \mathbf{T}_{K-1,K} \\ 0 & \cdots & \cdots & \mathbf{T}_{K,K-1} & \mathbf{T}_{K,K} \end{bmatrix} \qquad (2.9)$$

where only few elements in this transition matrix are non-zero.

## 2.3 Traffic Modeling

In the literature, the most commonly-used traffic models are Poisson and Bernoulli renewal processes. Other more elaborate traffic models which can take traffic burstiness into account were proposed in [99]-[101]. In the following, we will introduce the discrete batch Markovian arrival process (BMAP), which will be used in chapter 4 of this dissertation. BMAP can be used as an elegant traffic model because it is tractable and it can represent the traffic burstiness inherent in many types of practical traffic sources.

The discrete batch Markovian arrival process (BMAP) can be described by $M+1$ sub-stochastic matrices $\mathbf{U}_m$, $(m = 0, 1, ..., M)$ which have the same order. The elements $(\mathbf{U}_m)_{i,j}$ describe a transition from phase $i$ to phase $j$ with $m$ arrivals. Let $\mathbf{U} = \sum_{m=0}^{M} \mathbf{U}_m$, $\mathbf{w} = \mathbf{wU}$ and $\mathbf{w1} = 1$. Then the average arrival rate $\lambda$ can be calculated as follows [100]:

$$\lambda = \mathbf{w} \sum_{m=1}^{M} m\mathbf{U}_m\mathbf{1} \tag{2.10}$$

where $\mathbf{1}$ is a column vector of all ones with appropriate dimension.

The Markovian source used in [25] is actually a special case of the BMAP arrival process. The authors in [25] represented the Markovian source by an $(M+1)$-state transition probability matrix, where at each state $m$ $(m = 0, \cdots, M)$, $m$ packets are generated in one time interval. This Markovian source can be modeled as a BMAP if we consider the state of this Markovian source as the phase of the BMAP.

The batch Bernoulli arrival process in which $m$ packets $(m = 0, 1, ..., M)$ arrive in one time interval with probability $\lambda_m$ is also a special case of BMAP arrival where $\mathbf{U}_m = \lambda_m$. In this case, $\mathbf{U}_m$ degenerate into scalars which do not capture traffic correlation any more.

## 2.4 Matrix-Geometric Method

In this section, we describe the fundamentals of the matrix-geometric method (MGM) for solving a queueing problem. Specifically, the solution for a quasi-birth and death process (QBD) with finite and infinite buffers are presented. More details about these techniques can be found in [114].

### 2.4.1 Infinite Buffer Case

In this case, the probability transition matrix for the Markov chain describing the queueing system has the following form:

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}_{0,0} & \mathbf{D}_{0,1} & & & & \\ \mathbf{D}_{1,0} & \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & & & \\ & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_0 & & \\ & & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{2.11}$$

The above probability transition matrix has a special structure. Except for the matrix blocks in the boundary (the first and second rows of matrix blocks), the structure is repeated and it increases or decreases only by one level. The steady-state probability vector $\boldsymbol{\pi} = [\boldsymbol{\pi}_0\, \boldsymbol{\pi}_1\, \boldsymbol{\pi}_2 \cdots]$ of the above transition matrix satisfies $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ and $\sum_{i=0}^{\infty} \boldsymbol{\pi}_i \mathbf{1} = 1$. In fact, there is a stochastic matrix $\mathbf{R}$ such that $\boldsymbol{\pi}_{i+1} = \boldsymbol{\pi}_i \mathbf{R}$ $(i \geq 1)$, where $\mathbf{R}$ is the minimal non-negative solution to the following equation:

$$\mathbf{R} = \mathbf{D}_0 + \mathbf{R}\mathbf{D}_1 + \mathbf{R}^2\mathbf{D}_2. \tag{2.12}$$

Thus, we can find $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_1$, and $\boldsymbol{\pi}_2$ from the boundary and normalization conditions using the following relations:

$$[\boldsymbol{\pi}_0\, \boldsymbol{\pi}_1\, \boldsymbol{\pi}_2] = [\boldsymbol{\pi}_0\, \boldsymbol{\pi}_1\, \boldsymbol{\pi}_2] \begin{bmatrix} \mathbf{D}_{0,0} & \mathbf{D}_{0,1} & \mathbf{0} \\ \mathbf{D}_{1,0} & \mathbf{D}_{1,1} & \mathbf{D}_{1,2} \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{D}_1 + \mathbf{R}\mathbf{D}_2 \end{bmatrix} \tag{2.13}$$

$$\boldsymbol{\pi}_0 \mathbf{1} + \boldsymbol{\pi}_1 \mathbf{1} + \boldsymbol{\pi}_2 (\mathbf{I} - \mathbf{R})^{-1} \mathbf{1} = 1 \tag{2.14}$$

where $\mathbf{1}$ denotes the column vector of all ones with appropriate dimension. In fact, $\boldsymbol{\pi}_0$ may have a dimension different from that of $\boldsymbol{\pi}_1$.

**Stability condition:** The stability condition is $\boldsymbol{\varphi}\mathbf{D}_0\mathbf{1} < \boldsymbol{\varphi}\mathbf{D}_2\mathbf{1}$, where $\boldsymbol{\varphi}$ is obtained from $\boldsymbol{\varphi} = \boldsymbol{\varphi}\mathbf{D}$ and $\boldsymbol{\varphi}\mathbf{1} = 1$ for $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1 + \mathbf{D}_2$ [114].

### 2.4.2 Finite Buffer Case

Assume that the buffer size is $K_0$ packets. In this case, the transition probability matrix for the Markov chain describing the queueing system has the following form:

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}_{0,0} & \mathbf{D}_{0,1} & & & & \\ \mathbf{D}_{1,0} & \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & & & \\ & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_{0b} \\ & & & & \mathbf{D}_{2b} & \mathbf{D}_{1b} \end{bmatrix}. \tag{2.15}$$

The stationary probability $\boldsymbol{\pi} = [\boldsymbol{\pi}_0\ \boldsymbol{\pi}_1\ \boldsymbol{\pi}_2 \cdots \boldsymbol{\pi}_{K_0}]$ satisfies

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}, \quad \sum_{i=0}^{K_0} \boldsymbol{\pi}_i \mathbf{1} = 1 \tag{2.16}$$

where $K_0$ is the number of rows of blocks in (2.15). We can easily find the relationship between $\boldsymbol{\pi}_{i-1}$ and $\boldsymbol{\pi}_i$ as follows:

$$\boldsymbol{\pi}_{i-1} = \boldsymbol{\pi}_i \mathbf{R}_i \quad (i \geq 1) \tag{2.17}$$

where

$$\mathbf{R}_1 = \mathbf{D}_{1,0}(\mathbf{I} - \mathbf{D}_{0,0})^{-1}, \quad \mathbf{R}_2 = \mathbf{D}_2(\mathbf{I} - \mathbf{D}_{1,1} - \mathbf{R}_1\mathbf{D}_{0,1})^{-1}$$
$$\mathbf{R}_i = \mathbf{D}_2(\mathbf{I} - \mathbf{D}_1 - \mathbf{R}_{i-1}\mathbf{D}_0)^{-1} \quad (4 \leq i \leq K_0 - 1).$$
$$\mathbf{R}_3 = \mathbf{D}_2(\mathbf{I} - \mathbf{D}_1 - \mathbf{R}_2\mathbf{D}_{1,2})^{-1}, \quad \mathbf{R}_{K_0} = \mathbf{D}_{2b}(\mathbf{I} - \mathbf{D}_1 - \mathbf{R}_{K_0-1}\mathbf{D}_0)^{-1}.$$

We can find $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_{K_0}$ using the following relations:

$$\sum_{i=0}^{K_0} \boldsymbol{\pi}_i \mathbf{1} = 1. \tag{2.18}$$

$$\boldsymbol{\pi}_0\mathbf{D}_{0,0} + \boldsymbol{\pi}_1\mathbf{D}_{1,0} = \boldsymbol{\pi}_0 \tag{2.19}$$

$$\boldsymbol{\pi}_{K_0-1}\mathbf{D}_{0b} + \boldsymbol{\pi}_{K_0}\mathbf{D}_{1b} = \boldsymbol{\pi}_{K_0} \tag{2.20}$$

Using (2.17), these three equations can be rewritten in the same order as follows:

$$\boldsymbol{\pi}_0\mathbf{1} + \boldsymbol{\pi}_{K_0}\left(\sum_{i=2}^{K_0}\prod_{j=K_0}^{i}\mathbf{R}_j\right)\mathbf{1} + \boldsymbol{\pi}_{K_0}\mathbf{1} = 1 \tag{2.21}$$

$$\boldsymbol{\pi}_0\mathbf{D}_{0,0} + \boldsymbol{\pi}_{K_0}\left(\prod_{i=K_0}^{2}\mathbf{R}_i\right)\mathbf{D}_{1,0} = \boldsymbol{\pi}_0 \tag{2.22}$$

$$\boldsymbol{\pi}_{K_0}\mathbf{R}_{K_0}\mathbf{D}_{0b} + \boldsymbol{\pi}_{K_0}\mathbf{D}_{1b} = \boldsymbol{\pi}_{K_0} \tag{2.23}$$

where $\prod_{j=K_0}^{i}\mathbf{R}_j = \mathbf{R}_{K_0}\cdots\mathbf{R}_i$. Other probability vectors $\boldsymbol{\pi}_i$ $(1 \leq i < K_0)$ can be calculated by using (2.17).

We would like to emphasize that MGM is only one queueing approach developed recently. Other queueing approaches such as the one based on z-transform technique is also commonly used in performance evaluation of communications systems and protocols. However, the advantage of the MGM approach is that it is easier to derive the queue length and delay distributions, which is very important for system evaluation and design purposes. We recommend readers to consult [115] for more detailed treatment of this queueing approach.

## 2.5 Subgradient Algorithm for Convex Optimization Problems

Consider the following convex optimization problem

$$\begin{array}{ll} \text{minimize} & w_0(\boldsymbol{x}) \\ \text{subject to} & w_i(\boldsymbol{x}) \leq 0, \quad i = 1, \cdots, m \\ & v_i(\boldsymbol{x}) = 0, \quad i = 1, \cdots, p \\ & \boldsymbol{x} \in C \end{array} \tag{2.24}$$

where $w_i$ $(i = 0, \cdots, m)$ are convex functions and $v_i$ $(i = 1, \cdots, p)$ are affine functions over variable $\boldsymbol{x} \in \mathbb{R}^n$, and $C$ is the polyhedral and bounded. By introducing the Lagrange multipliers for the equality and inequality constraints of this optimization problem, we can write the corresponding Lagrangian as follows:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = w_0(x) + \sum_{i=1}^{m}\lambda_i w_i(\boldsymbol{x}) + \sum_{i=1}^{p}\nu_i v_i(\boldsymbol{x}) \tag{2.25}$$

where $\lambda_i$ and $\nu_i$ are elements of the Lagrange multiplier vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$, respectively. The dual function can be defined from the Lagrangian as follows:

$$D(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x}\in C} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{2.26}$$

And the dual problem can be defined as

$$\begin{array}{ll} \text{maximize} & D(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} & \boldsymbol{\lambda} \succeq \mathbf{0}. \end{array} \tag{2.27}$$

For convex optimization problems defined in (2.24) where the strong duality holds, the duality gap is zero [116] and solutions of the original optimization problem in (2.24) can be recovered via the dual problem defined in (2.27). The optimization problem in (2.24) is called the primal problem. Solving the dual problem usually results in a distributed algorithm which is very desirable for multi-hop wireless networks. This is because the collection and distribution of network information in multi-hop wireless networks results in significant communication overhead, and therefore, consumes too much network resources.

Because the dual function may not be differentiable, we solve the dual problem by the subgradient method. Similar to the standard gradient method, the subgradient method updates dual variables by using the subgradient. Definition of the subgradient for a convex function is as follows.

**Definition 2.1:** Given a convex function $f : \mathbb{R}^n \longrightarrow R$, vector $d$ is the subgradient of $f$ at $\boldsymbol{x}$ if $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + d^T(\boldsymbol{y} - \boldsymbol{x})$, $\forall \boldsymbol{y} \in \mathbb{R}^n$ where $^T$ denotes the transposition.

Subgradients of dual function $-D(\boldsymbol{\lambda}, \boldsymbol{\nu})$ at $\lambda_i$ and $\nu_i$ can be easily shown to be

$$f_i(\lambda_i) = w_i(\boldsymbol{x}^*) \quad \text{and} \quad g_i(\nu_i) = v_i(\boldsymbol{x}^*) \tag{2.28}$$

where $\boldsymbol{x}^*$ is the solution of the optimization problem (2.26). The subgradient algorithm updates the dual variables $\lambda_i$ and $\nu_i$ as follows

$$\begin{aligned} \lambda_i(t+1) &= [\lambda_i(t) - \beta(t) f_i(\lambda_i(t))]^+ \\ \nu_i(t+1) &= \nu_i(t) - \beta(t) g_i(\nu_i(t)) \end{aligned}$$

We have the following results for the convergence of the subgradient algorithm [119].

**Property:** Given that the sequence for the stepsize $\beta(t)$ is chosen to be *nonsummable diminishing* which satisfies

$$\lim_{t \longrightarrow \infty} \beta(t) = 0, \quad \sum_{t=1}^{\infty} \beta(t) = \infty \tag{2.29}$$

The subgradient algorithm converges to the globally optimal solution.

# Chapter 3

# Analysis for Go-Back-N and Selective Repeat ARQ Protocols

Design of strong and reliable error correction codes plays a key role in error protection for applications with strict delay requirements (e.g., voice). Deployment of delay-tolerant data services in wireless networks, however, makes automatic repeat request (ARQ)-based error protection very attractive to counteract the residual errors without using costly error correction codes in the physical layer.

Among the three main ARQ protocols (namely, stop-and-wait, go-back-N (GBN-ARQ) and selective repeat (SR-ARQ)), SR-ARQ is the most efficient. GBN-ARQ is less efficient than SR-ARQ but its implementation is simpler than SR-ARQ because packets are always received in order at the receiving buffer [31]. The key weakness of analytical models for ARQ protocols in the literature is the use of a two-state Markov channel model, which could not capture the multi-rate transmission feature of most current wireless networks.

Delay statistics for GBN-ARQ with non-instantaneous feedback delay was derived in [23]. In [25], the approximated average delay for SR-ARQ was obtained for a two-state Markov channel under heavy traffic condition. The exact delay statistics for SR-ARQ over a two-state Markov channel was obtained in [26]. The analytical model in [26] was extended in [28] for channels with $N$ states. However, the transmission rate was assumed to be constant (i.e., for all channel states one packet is transmitted in one time slot); therefore, the model did not truly take the multi-rate transmission into account. Developing an analytical framework for evaluating radio link level queueing performances which can capture multi-rate transmission and non-instantaneous feedback delay is very desirable but challenging.

Note that the availability of radio link-level delay statistics allows wireless network design and engineering under statistical delay constraints of the form Pr $\{\text{delay} > D_{\max}\}$ $< P_t$ instead of those based on the average delay or delay bounds. Also, it would be very useful in predicting the higher layer protocol (e.g., TCP (Transmission Control Protocol)) performance (e.g., to estimate the round trip time of a TCP flow). So far, investigations of higher layer protocol performance have been done mostly by simulations. The link model presented in this chapter is therefore an important step towards investigating the interaction of TCP with lower layers protocols [106], [107].

A finite-state Markov channel (FSMC) model was proposed for Rayleigh and the more general Nakagami-$m$ fading channels in the literature [92], [94], [97], [98]. This channel model was validated in [97] and extensively used to analyze system performance at the packet level [20], [93]. In [20], the authors developed a queueing model that takes into account AMC in the physical layer. However, ARQ was not considered in the link layer and the delay statistics cannot be derived from their model. The difficulty of using FSMC model to analyze the performance of multi-rate wireless networks comes from the batch transmission effect where the transmission batch size varies according to the chosen transmission modes in the physical layer.

In this chapter, we propose radio link layer queueing models for GBN and SR ARQ protocols in multi-rate wireless networks by using the matrix geometric method (MGM) [114]. The proposed model applies to the scenario where each user in the network is allocated one separate channel to communicate with its BS (i.e., single-user scenario). The analytical model enables us to quantify the impacts of physical/radio link and channel parameters on the system performance. Also, it provides interesting insights into the system design. For example, SNR thresholds for different transmission modes can be calculated to achieve good link level delay performance. Note that, delay is the primary quality of service (QoS) metric for many data applications and the SNR thresholds which maximize the link level throughput may not be, in general, delay-optimal. Comparison of delay performance of the two ARQ protocols is also highlighted for different values of feedback delay which is necessary to quantify the tradeoff between delay performance and implementation complexity for link layer protocol design.

## 3.1 System Model and Assumptions

### 3.1.1 System Description

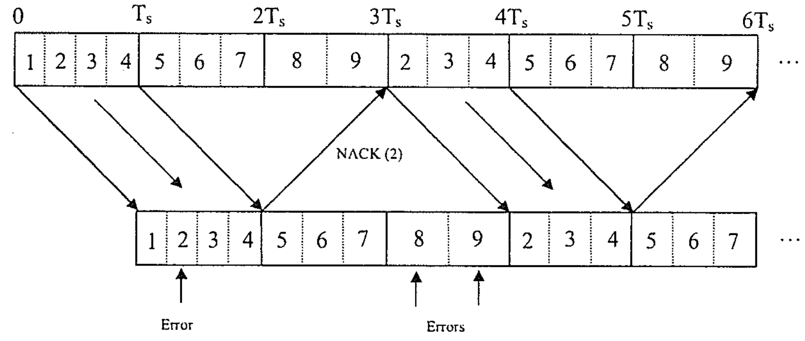
We consider a transmitter node using adaptive modulation and coding in the physical layer and ARQ-based error recovery in the link layer to communicate with a receiver node over a wireless channel. Transmissions occur in fixed-size time slots where the number of packets transmitted during each time slot depends on the chosen transmission mode. The receiver decodes the received packets and sends a feedback packet (i.e., containing acknowledgment (ACK) or negative acknowledgment (NACK) information) to the transmitter. In case of transmission failure of one or more packets transmitted during a time slot, an error recovery based on either GBN-ARQ or SR-ARQ protocol is initiated.

For both ARQ protocols, the transmitter continuously transmits packets from the buffer in sequence until it detects a transmission error through NACK in the feedback packet. In case of transmission failure(s), the SR-ARQ protocol only retransmits the erroneous packet(s) while the GBN-ARQ protocol retransmits all the packets starting from the first erroneous one.

We assume that the feedback packet (i.e., the ACK/NACK information) arrives at the transmitter node $n$ slots after the beginning of the corresponding transmission slot. In this chapter, an error-free feedback channel is assumed[1]. In addition to the ACK/NACK information, the feedback channel also carries the channel state information (CSI) or the selected transmission mode to be used for dynamic link adaptation. We assume that CSI is available at the transmitter without delay. This assumption is reasonable in slow fading channels where the channel conditions are static over several transmission intervals (or time slots). The maximum number of retransmissions allowed for a packet is assumed to be unbounded. Therefore, the delay obtained in this paper can be considered as an upper bound for the case where finite number of retransmissions are allowed in the link layer.

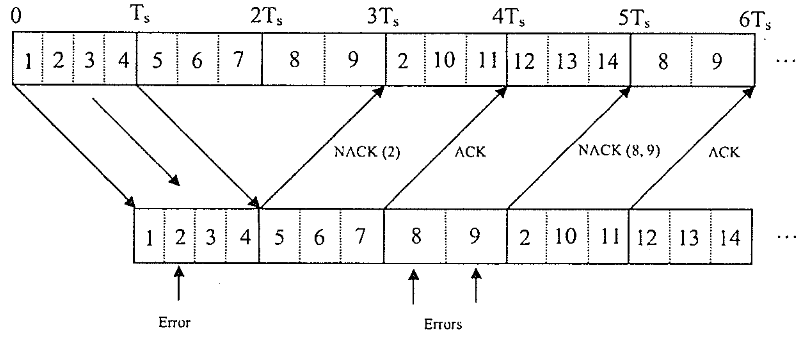
**Examples**: The operations of GBN-ARQ and SR-ARQ protocols are illustrated in Fig. 3.1 and Fig. 3.2, respectively, for $n = 3$, where the transmission batch size is

[1] This is a very standard assumption because in practice a strong error correction code can be used in the feedback channel.



Rate: 4, 3, 2, 3, 3, 2, ... (packets/slot)

**Figure 3.1.** *GBN-ARQ timing diagram.*



Rate: 4, 3, 2, 3, 3, 2, ... (packets/slot)

**Figure 3.2.** *SR-ARQ timing diagram.*

denoted by the rate defined in terms of packets/slot. In these two figures, packet 2 in time slot 1 and packets 8, 9 in time slot 3 are assumed to be in error. For the GBN-ARQ protocol, the NACK for packet 2 arrives at the transmitter side at the end of time slot 3 and retransmission of all packets starting from packet 2 begins at time slot 4. Note that, packets $3, 4, \cdots, 7$ are retransmitted even though they were correctly received before. For the SR-ARQ protocol, packet 2 and packets 8, 9 are "selectively" retransmitted in time slots 4 and 6, respectively, together with the new packets when the channel state allows.

The channel is modeled as an FSMC with $K+1$ states $(0, 1, \cdots, K)$ as described in *Chapter 2*. When the channel is in state $k$ $(1, 2, \cdots, K)$, $h_k$ packets are transmitted in one time slot. In fact, each channel state corresponds to one transmission mode of the AMC technique. We further assume that the transmitter does not transmit in channel state 0 to avoid high probability of transmission errors.

In the physical layer of the considered system, each transmission mode corresponds to a unique modulation and coding scheme. The number of packets transmitted in mode $k$ (equal to $h_k$) is therefore proportional to the spectral efficiency of mode $k$. For example, if the spectral efficiencies of five transmission modes in a particular system are 0.5, 1, 1.5, 2.5 and 3.5 (bits/s/Hz) and 1 packet can be transmitted in mode one (with spectral efficiency of 0.5), the number of packets transmitted in one time slot using the other modes is 2, 3, 5, 7, respectively. The maximum number of packets that can be transmitted in one time slot (equal to $h_K$) is denoted by $N$.

The radio link level queueing for both ARQ protocols is modeled in discrete time with one time interval equal to one time slot and the system states are observed at the beginning of each time slot. The buffer size is assumed to be infinite. Packet arrivals follow a Bernoulli process with arrival probability $\lambda$. We assume that a packet arriving during time interval $t-1$ cannot be transmitted until time interval $t$ at the earliest.

## 3.2 Analysis of Go-Back-N ARQ Protocol

### 3.2.1 Queueing Model

Since the result of the decoding process for each packet only reaches the transmitter $n$ slots after the beginning of the transmission slot, if a transmitted packet is in error in time slot $t$, all transmissions from time slot $t+1$ to time slot $t+n-1$ will be discarded. Therefore, we need to keep track not only of the channel state, which determines how many packets can be transmitted in one time slot, but also the *useful* time slot, which is defined as the slot where the transmitted packets, if successfully decoded, will be accepted by the receiver.

Let $q(t) \geq 0$ represent the number of packets in the queue, including packets which will be retransmitted but whose NACKs have not yet been received, $0 \leq s(t) \leq n-1$ track the useful time slot, and $0 \leq c(t) \leq K$ represent the channel state. We assign the value for $s(t)$ as follows. If a transmission failure occurs in a useful time slot, $s(t)$ will be equal to $n-1$ in the next time slot. Then, it will be decreased during the subsequent slots until $s(t)=0$, where a useful transmission period starts. This is illustrated in Fig. 3.3 for $n=4$, where the evolution of $s(t)$ is shown. As is evident, the number of *useless* slots following transmission error(s) in a *useful* slot is $n-1$. It can be shown that the random process $\mathbf{X}(t)=\{q(t), s(t), c(t)\}$ forms a discrete-time Markov chain (MC). For brevity, we will omit time index $t$ in the related variables if it does not cause confusion.

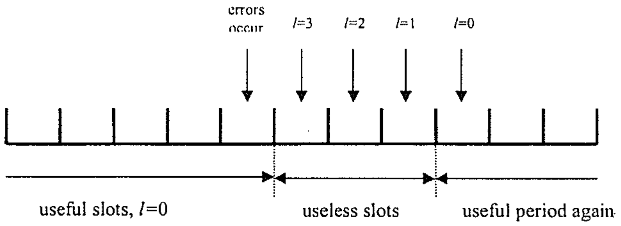


**Figure 3.3.** *Modeling of GBN-ARQ for* $n=4$.

In order to calculate the steady state probability for the underlying MC, it is important to put its transition probability matrix in a nice form where its specific transition structure can be exploited. Now, let $(i,j,k)$ be the generic system state (i.e., $q=i$, $s=j$, and $c=k$) and $(i,j,k)\to(i',j',k')$ denote the system transition from state $(i,j,k)$ to state $(i',j',k')$. For fixed $i$, the probabilities corresponding to system state transitions $(i,*,*)\to(i+1-l,*,*)$ can be written in a matrix block $\mathbf{D}_{i,l}$. We further put the probabilities of state transitions $(i,j,*)\to(i+1-l,j',*)$ into a sub-matrix $\mathbf{D}_{i,l}(j,j')$ of $\mathbf{D}_{i,l}$. Also, the probability of transition $(i,j,k)\to(i+1-l,j',k')$ is denoted by $\mathbf{D}_{i,l}(j,j')(k,k')$, which is an element of $\mathbf{D}_{i,l}(j,j')$. In fact, the probabilities corresponding to transitions from state $(i,j,k)$ to any other state will be in the $(i(K+1)n+j(K+1)+k)$-th row of the probability transition matrix and they are elements of $\mathbf{D}_{i,l}$ for some value of $l$ depending on the destination state. The following example clarifies further how the system state transition probabilities are ordered in the matrix form.

**Example:** For ease of exposition, we consider a very simple case where there are 2 channel states (states 0, 1) and feedback delay is 2 slots (i.e., $n=2$). The matrix block $\mathbf{D}_{i,l}$ can be expanded as in (3.1). In this equation, element $\mathbf{D}_{i,l}(1,0)(1,0)$, for example, represents the probability of transition $(i,1,1) \rightarrow (i+1-l,0,0)$.

$$\mathbf{D}_{i,l} = \left[ \begin{array}{cc|cc} \mathbf{D}_{i,l}(0,0)(0,0) & \mathbf{D}_{i,l}(0,0)(0,1) & \mathbf{D}_{i,l}(0,1)(0,0) & \mathbf{D}_{i,l}(0,1)(0,1) \\ \mathbf{D}_{i,l}(0,0)(1,0) & \mathbf{D}_{i,l}(0,0)(1,1) & \mathbf{D}_{i,l}(0,1)(1,0) & \mathbf{D}_{i,l}(0,1)(1,1) \\ \hline \mathbf{D}_{i,l}(1,0)(0,0) & \mathbf{D}_{i,l}(1,0)(0,1) & \mathbf{D}_{i,l}(1,1)(0,0) & \mathbf{D}_{i,l}(1,1)(0,1) \\ \mathbf{D}_{i,l}(1,0)(1,0) & \mathbf{D}_{i,l}(1,0)(1,1) & \mathbf{D}_{i,l}(1,1)(1,0) & \mathbf{D}_{i,l}(1,1)(1,1) \end{array} \right] \qquad (3.1)$$

The resulting transition matrix for $X(t)$ is written in (3.2) for $N = 3$. Recall that $N$ is the maximum number of packets which can be transmitted in one time slot (i.e., equal to $h_K$). In this probability transition matrix, $\mathbf{D}_{i,l}$ contains the probabilities of system transitions where $q = i$ before the transitions. All transition probabilities captured by $\mathbf{D}_{i,l}$ for different $l$ will be called transitions in level $i$ of the transition matrix in the sequel. Note that, in the generic system state $(i, j, k)$, $j$ can have $n$ possibilities and $k$ can have $K + 1$ possibilities (i.e., $K + 1$ channel states). Thus, the order of $\mathbf{D}_{i,l}$ is $n(K + 1) \times n(K + 1)$. The derivations of matrix blocks $\mathbf{D}_{i,l}$ and $\mathbf{D}_l$ are detailed in **Appendix A**. As can be seen in Appendix A, for $i \geq N$, $\mathbf{D}_{i,l}$ is independent of the level index $i$; therefore, for brevity we denote $\mathbf{D}_{i,l}$ by $\mathbf{D}_l$ in (3.2).

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}_{0,1} & \mathbf{D}_{0,0} & & & & & \\ \mathbf{D}_{1,2} & \mathbf{D}_{1,2} & \mathbf{D}_{1,0} & & & & \\ \mathbf{D}_{2,3} & \mathbf{D}_{2,2} & \mathbf{D}_{2,1} & \mathbf{D}_{2,0} & & & \\ \mathbf{D}_4 & \mathbf{D}_3 & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_0 & & \\ & \mathbf{D}_4 & \mathbf{D}_3 & \mathbf{D}_2 & \mathbf{D}_1 & \mathbf{D}_0 & \\ & & & & \ddots & \ddots & \ddots \end{bmatrix}. \qquad (3.2)$$

In (3.2), there is at most one arriving packet and at most $N = 3$ packets successfully transmitted in one time slot. Therefore, for level $i \geq 3$, the transitions can go up at most one level (represented by $\mathbf{D}_0$) and go down at most three levels (represented by $\mathbf{D}_4$). The transition matrix in (3.2) describes a GI/M/1 Markov chain, where the solution can be found by the well-established method proposed by Neuts [114] which is presented in *Chapter 2*. In fact, the steady-state probability $\mathbf{x} = [\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ \cdots]$ satisfies

$$\mathbf{xP} = \mathbf{x}, \qquad \sum_{i=0}^{\infty} \mathbf{x}_i \mathbf{1} = 1 \qquad (3.3)$$

where $\mathbf{1}$ is a column vector of all ones with the same dimension as $\mathbf{x}_i$ which is $n(K+1)$. We can find $\mathbf{x}_0$, $\mathbf{x}_1$, $\cdots$, $\mathbf{x}_N$ using the boundary and the normalization conditions.

Other values of $\mathbf{x}_i$ ($i > N$) can be calculated from $\mathbf{x}_N$ by using a non-negative matrix $\mathbf{R}$ as follows: $\mathbf{x}_i = \mathbf{x}_N \mathbf{R}^{i-N}$ [114]. Here, the order of matrix $\mathbf{R}$ is $n(K+1) \times n(K+1)$.

### 3.2.2 Delay Analysis

In this section, we derive the delay distribution of a packet arriving at the queue for the GBN ARQ protocol. The delay is the time it takes all packets ahead of the target packet (if any) and itself to successfully leave the queue. In the following calculation, the delay is considered at the transmitter side. Let the arrival slot be numbered as slot zero. It is not included in the delay calculation.

Now let $\mathbf{\Phi}(p,d)$ be matrices with order $n(K+1) \times n(K+1)$ whose element $(\mathbf{\Phi}(p,d))\ (j,j')(k,k')$ $(0 \le j,j' \le n-1,\ 0 \le k,k' \le K)$ is the probability of state transition $(p,j,k) \rightarrow (0,j',k')$ in $d$ time slots. In short, $\mathbf{\Phi}(p,d)$ contains system transition probabilities such that $p$ packets are successfully transmitted in $d$ slots. From the definition of $\mathbf{\Phi}(p,d)$, $(\mathbf{\Phi}(p,d))(j,j')$ contains the channel state transition probabilities such that $s$ (in the system state $(q,s,c)$) evolves from $j$ to $j'$. Thus, the order of $(\mathbf{\Phi}(p,d))(j,j')$ is $(K+1) \times (K+1)$.

We also define $\mathbf{C}_{h,p}$ to be matrices of order $n(K+1) \times n(K+1)$ with the same structure as $\mathbf{\Phi}(p,d)$ whose elements are the system transition probabilities such that $h$ packets are successfully transmitted in one particular time slot given that there are $p$ packets in the queue at the beginning of the time slot. The derivation of $\mathbf{C}_{h,p}$ is given in **Appendix A**. We have the following recursive relation:

$$\mathbf{\Phi}(p,d) = \sum_{h=0}^{N} \mathbf{C}_{h,p} \mathbf{\Phi}(p-h, d-1), \quad \text{where} \quad \mathbf{\Phi}(0,0) = \mathbf{I}_{n(K+1)}. \tag{3.4}$$

Equation (3.4) can be interpreted as follows. If there are $p$ packets which must be delivered in $d$ time slots (captured by $\mathbf{\Phi}(p,d)$) and $h$ packets are successfully transmitted in the first time slot (captured by $\mathbf{C}_{h,p}$), there are remaining $p-h$ packets to be delivered in $d-1$ slots (captured by $\mathbf{\Phi}(p-h,d-1)$). Here, $\mathbf{\Phi}(0,0)$ simply captures the end point where the target packet leaves the queue.

To calculate the delay statistics, we need to obtain the steady-state vector seen by a packet arriving to the queue. Note that we have assumed a Bernoulli arrival process so that the ASTA (arrivals see time averages) property holds here. Also, packets arriving to the queue after the tagged packet do not affect the delay experienced by the tagged packet so they are ignored in the following derivation. Let $\mathbf{y}_i$ be a vector of dimension $n(K+1)$ which represents the system state probabilities where an arriving packet sees $i$ head-of-line (HOL) packets at the end of its arrival time slot. We have

$$\mathbf{y}_i = \sum_{h=0}^{N} \mathbf{x}_{i+h} \mathbf{C}_{h,i+h}. \tag{3.5}$$

Equation (3.5) can be interpreted as follows. If there are $i+h$ packets in the queue at the beginning of the arrival time slot (captured by $\mathbf{x}_{i+h}$) and $h$ packets successfully leave the queue in this time slot (captured by $\mathbf{C}_{h,i+h}$), the arriving packet will see exactly $i$ HOL packets (captured in $\mathbf{y}_i$) at the end of this time slot. The probability that the delay is $D$ slots (not including the arrival slot) can therefore be written as follows:

$$P_d(D) = \sum_{h=0}^{DN-1} \mathbf{y}_h \mathbf{\Phi}(h+1, D) \mathbf{1}_{n(K+1)} \tag{3.6}$$

where the sum in (3.6) is limited to $DN-1$ since at most $N$ packets can be successfully transmitted in one time slot.

## 3.3 Analysis of Selective Repeat ARQ Protocol

### 3.3.1 Queueing Model

Since the outcome of the decoding process for each packet only reaches the transmitter $n$ slots after the beginning of the corresponding transmission slot and only erroneous packets are "selectively" retransmitted, we need to keep track of the number of erroneous packets in a window of $n$ slots.

Let $\mathbf{b}(t) = [b_1(t), b_2(t), \cdots, b_n(t)]$ be an $n$-dimensional vector whose elements $b_i(t)$, $(i = 1, \cdots, n)$ represent the number of erroneous packets among those transmitted in the slot which is $i$ slots before the current slot $(0 \leq b_i(t) \leq N)$. Note that, we do not need to differentiate two different cases which can lead to $b_i(t) = 0$: no packet is transmitted because the channel is in state zero and all of the transmitted packets are successfully decoded at the receiver. This is due to the operation of the protocol where only erroneous packets are retransmitted if any.

We can observe that $\mathbf{b}(t+1) = [\beta, b_1(t), b_2(t), \cdots, b_{n-1}(t)]$, where $\beta$ is the number of packets in error among those transmitted in time slot $t$. To facilitate the analysis, we represent vector $\mathbf{b}(t)$ by a number $y(t) = \sum_{i=1}^{n} b_i \times (N+1)^{(i-1)}$. Since $\mathbf{b}(t+1) = (\beta, b_1(t), b_2(t), \cdots, b_{n-1}(t))$, for a given $y(t) = k$ there are at most $N+1$ transitions to $y(t+1) = l$ corresponding to different values of $\beta$. Also note that there is a unique mapping between $y(t)$ and $\mathbf{b}(t)$; therefore, the use of $y(t)$ instead of $\mathbf{b}(t)$ makes the analysis easier without affecting the semantics of the problem.

**Example**: Consider the SR-ARQ protocol with feedback delay $n = 4$ (time slots) and $N = 4$. Suppose that the current vector $\mathbf{b}$ is $\mathbf{b}(t) = (b_1(t), b_2(t), b_3(t), b_4(t)) = (1, 0, 3, 4)$ and two packets among those transmitted in the current time slot are in error. The vector $\mathbf{b}$ in the next time slot will be $\mathbf{b}(t+1) = (2, 1, 0, 3)$. The corresponding transition is $\mathbf{b}(t) \rightarrow \mathbf{b}(t+1) \equiv \{y(t) = 576\} \rightarrow \{y(t+1) = 382\}$.

Let $q(t) \geq 0$ represent the number of packets in the queue excluding the packets which were transmitted and the transmitter is waiting for their ACKs/NACKs, $y(t)$ corresponding to vector $\mathbf{b}(t)$ capture the transmission outcomes in the past $n$ time slots, and $0 \leq c(t) \leq K$ represent the channel state. Then, the random process $\mathbf{Y}(t) = \{q(t), y(t), c(t)\}$ forms a discrete-time Markov chain.

In any time slot $t$, the number of packets available for transmission is $q(t) + b_n(t)$, where $b_n(t)$ is the number of erroneous packets which were transmitted $n$ slots before the current slot and are being retransmitted. The number of packets transmitted at time $t$ is given by $\min\left\{h_{c(t)}, q(t) + b_n(t)\right\}$, which is the minimum of the number of available packets in the queue (equal to $q(t) + b_n(t)$) and the transmission capability of channel state $c(t)$ (equal to $h_{c(t)}$). If $a(t)$ denotes the number of arriving packets in time slot $t$, we have $q(t+1) = q(t) + a(t) + b_n(t) - \min\left\{h_{c(t)}, q(t) + b_n(t)\right\}$. The protocol modeling is illustrated in Fig. 3.4. We will omit time index $t$ in the related variables if it does not cause confusion in the sequel. Note that, element $b_i$ of vector $\mathbf{b}$ stores the number of packets transmitted erroneously in the past regardless of how many packets were really transmitted in the corresponding time slots.

Similar to the model for GBN-ARQ protocol, we put the transition probability matrix of $\mathbf{Y}(t)$ in a matrix form. Now, let $(i, j, k)$ be the generic system state and $(i, j, k) \rightarrow (i', j', k')$ denote the system transition from state $(i, j, k)$ to state $(i', j', k')$.

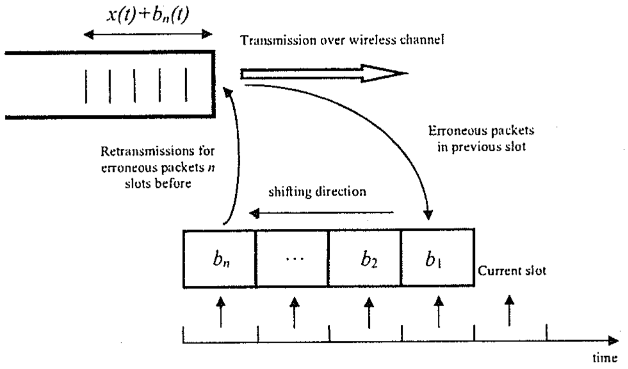

$x(t)+b_n(t)$
Transmission over wireless channel
Erroneous packets in previous slot
Retransmissions for erroneous packets $n$ slots before
shifting direction
$b_n$
$\cdots$
$b_2$
$b_1$
Current slot
time

**Figure 3.4.** *The SR-ARQ model.*

For fixed $i$, we write the probabilities of system transitions $(i,*,*) \to (i+N+1-l,*,*)$ in a matrix block $\mathbf{A}_{i,l}$. In the generic system state, $j$ has $(N+1)^n$ possibilities (because each element $b_i$ of vector $\mathbf{b}$ has $N+1$ possibilities) and $k$ has $K+1$ possibilities (i.e., $K+1$ channel states); therefore, the order of $\mathbf{A}_{i,l}$ is $(K+1)(N+1)^n \times (K+1)(N+1)^n$.

The resulting transition matrix for $\mathbf{Y}(t)$ is written in (3.7) for $N=3$. As before, each row of matrix blocks in (3.7) captures the transitions in one level of the transition matrix. Also, the elements of $\mathbf{A}_{i,l}$ are $(\mathbf{A}_{i,l})(j,j')(k,k')$, which is the probability of system transition $(i,j,k) \to (i+N+1-l,j',k')$. The derivations of matrix blocks $\mathbf{A}_{i,l}$ and $\mathbf{A}_l$ are given in **Appendix B**. Note that, for $i \geq N$, we denote $\mathbf{A}_{i,l}$ by $\mathbf{A}_l$ for brevity because these matrix blocks are independent of the level index. Since we have $q(t+1)-q(t) = a(t)+b_n(t)-\min\left\{h_{c(t)}, q(t)+b_n(t)\right\}$, each level in the transition matrix (3.7) can go up at most $N+1$ levels (when $a(t)=1$, $b_n(t)=N$, and $\min\left\{h_{c(t)}, q(t)+b_n(t)\right\}=0$) and go down at most $N$ levels (when $a(t)=0$, $b_n(t)=0$, and $\min\left\{h_{c(t)}, q(t)+b_n(t)\right\}=N$).

$$\mathbf{P} = \left[\begin{array}{c|cccc|cccc|ccccc}
\mathbf{A}_{0,4} & \mathbf{A}_{0,3} & \mathbf{A}_{0,2} & \mathbf{A}_{0,1} & \mathbf{A}_{0,0} & & & & & & & & & \\
\hline
\mathbf{A}_{1,5} & \mathbf{A}_{1,4} & \mathbf{A}_{1,3} & \mathbf{A}_{1,2} & \mathbf{A}_{1,1} & \mathbf{A}_{1,0} & & & & & & & & \\
\mathbf{A}_{2,6} & \mathbf{A}_{2,5} & \mathbf{A}_{2,4} & \mathbf{A}_{2,3} & \mathbf{A}_{2,2} & \mathbf{A}_{2,1} & \mathbf{A}_{2,0} & & & & & & & \\
\mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & & & & & & \\
0 & \mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & & & & & \\
\hline
 & & \mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & & & & \\
 & & & \mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & & & \\
 & & & & \mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & & \\
 & & & & & \mathbf{A}_{7} & \mathbf{A}_{6} & \mathbf{A}_{5} & \mathbf{A}_{4} & \mathbf{A}_{3} & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & \\
\hline
 & & & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right]. \tag{3.7}$$

### 3.3.2 Steady-State Solution

We re-block the transition matrix as indicated in (3.7) to obtain a quasi-birth and death (QBD) process, and the re-blocked transition matrix can be written as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_{0,1} & \mathbf{B}_{0,0} & & & & \\ \mathbf{B}_{1,2} & \mathbf{B}_{1,1} & \mathbf{B}_{1,0} & & & \\ & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 & & \\ & & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{3.8}$$

The solution for the QBD process can be found by the well-established method proposed by Neuts [114]. Let $\boldsymbol{\pi} = [\boldsymbol{\pi}_0,\ \boldsymbol{\pi}_1,\ \boldsymbol{\pi}_2, \cdots]$ be the steady state probability vector of (3.8) and $\mathbf{z} = [\mathbf{z}_0,\ \mathbf{z}_1,\ \mathbf{z}_2, \cdots]$ be the steady state probability vector of the original transition matrix (3.7) where $\mathbf{z}_i$ corresponds to level $i$ of the transition matrix (3.7). Recall that we have combined $N+1$ blocks in each dimension of the transition matrix (3.7) to obtain (3.8). Thus, letting $N_1 = (K+1)(N+1)^n$, the dimension of $\mathbf{z}_i$ $(i \geq 0)$ is $N_1$ and the dimension of $\boldsymbol{\pi}_i$ $(i \geq 1)$ is $N_1(N+1)$. For $i \geq 1$, $\boldsymbol{\pi}_i$ can be written as $\boldsymbol{\pi}_i = [\boldsymbol{\pi}_{i,1},\ \boldsymbol{\pi}_{i,2}, \cdots,\ \boldsymbol{\pi}_{i,N+1}]$. Hence, we have

$$\mathbf{z}_0 = \boldsymbol{\pi}_0, \quad \mathbf{z}_{(i-1)(N+1)+j} = \boldsymbol{\pi}_{i,j} \tag{3.9}$$

for $i \geq 1$ and $1 \leq j \leq N+1$.

### 3.3.3 Delay Analysis

In this section, we derive the delay distribution of a packet arriving at the queue for the SR-ARQ protocol. Again, the delay is considered at the transmitter side. Let the arrival slot be numbered as slot zero and it is not included in the delay calculation.

Let $N_2 = (N+1)^n - 1$. We define the following matrices:

- $\boldsymbol{\Omega}(p,d)$ are matrices of order $N_1 \times N_1$ in which element $(\boldsymbol{\Omega}(p,d))\ (j,j')(k,k')$, $(0 \leq j,j' \leq N_2,\ 0 \leq k,k' \leq K)$ is the probability of system transition $(p,j,k) \rightarrow (0,j',k')$ in $d$ time slots. In short, $\boldsymbol{\Omega}(p,d)$ contains system transition probabilities such that in addition to the $p$ packets captured in $q(t)$ all erroneous packets in the past $n$ time slots are successfully transmitted in $d$ time slots.

- $\Theta_{(p,h)}$ are matrices of order $N_1 \times N_1$ in which element $\left(\Theta_{(p,h)}\right)(j,j')(k,k')$, $(0 \leq j, j' \leq N_2, 0 \leq k, k' \leq K)$ is the probability of system transition $(p, j, k) \rightarrow (p-h, j', k')$ in one time slot. The derivations of $\Theta_{(p,h)}$ are given in **Appendix C**.

We have the following recursive relation:

$$\boldsymbol{\Omega}(p, d) = \sum_{h=-N}^{N} \Theta_{(p,h)} \boldsymbol{\Omega}(p-h, d-1) \tag{3.10}$$

where

$$\boldsymbol{\Omega}(0,0) = \begin{bmatrix} \mathbf{I}_{K+1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

In (3.10), $\boldsymbol{\Omega}(p, d)$ captures the system evolution from time slot $d$ to time slot $d-1$ counting back from the last time slot. Also $\boldsymbol{\Omega}(0,0)$ simply captures the ending point where all HOL packets (if any), erroneous packets in a window of $n$ slots and the target packet have successfully left the queue. Note that, due to the special structure of $\boldsymbol{\Omega}(0,0)$, only transitions for which all remaining packets successfully leave the system in the last time slot are allowed to occur. This is a bit different from the corresponding transitions captured in (3.4) for the GBN-ARQ protocol.

To calculate the delay statistics, we need to obtain the steady-state vector seen by an arriving packet to the queue. Let $\mathbf{w}_i$ be a vector of dimension $N_1$ which represents the system state probability at the end of the arrival slot where an arriving packet sees $i$ HOL packets in the queue. Then, we have

$$\mathbf{w}_i = \sum_{h=-N}^{N} \mathbf{z}_{i+h} \Theta_{(i+h,h)} \tag{3.11}$$

where $\mathbf{z}_{i+h} = \mathbf{0}$, and $\Theta_{(i+h,h)} = \mathbf{0}$ if $i + h < 0$.

The interpretations of (3.10) and (3.11) are similar to those of (3.4) and (3.5). The probability that the delay is $D$ time slots (not including the arrival slot) can, therefore, be written as follows:

$$P_d(D) = \sum_{h=0}^{DN-1} \mathbf{w}_h \boldsymbol{\Omega}(h+1, D) \mathbf{1}_{N_1} \tag{3.12}$$

where the summation above contains $DN$ terms, since at most $N$ packets can be successfully transmitted in one time slot.

## 3.4 Model Validation and Numerical Results

We use the PER fitting values of $a_k$ and $g_k$ in Table I of [20] to obtain the SNR thresholds of the FSMC model such that $\overline{\text{PER}}_k = P_0$ for all the transmission modes, where $P_0$ is a certain target packet error rate. A wireless system using adaptive modulation is considered where $h_k = k$ (i.e., the transmitter transmits $k$ packets/slot in channel state $k$). To save simulation time in validating the analytical model, we consider three transmission modes (i.e., $K = 3$) in Fig. 3.5, while for the other results we assume five transmission modes (i.e., $K = 5$). We assume that the time slot interval $T_s = 1$ ms. The complementary delay distributions (i.e., $\Pr(\text{delay} > d) = 1 - \sum_{k=1}^{d} P_d(k)$) obtained from both the simulation and the analytical model for both GBN-ARQ and SR-ARQ are shown in Fig. 3.5.

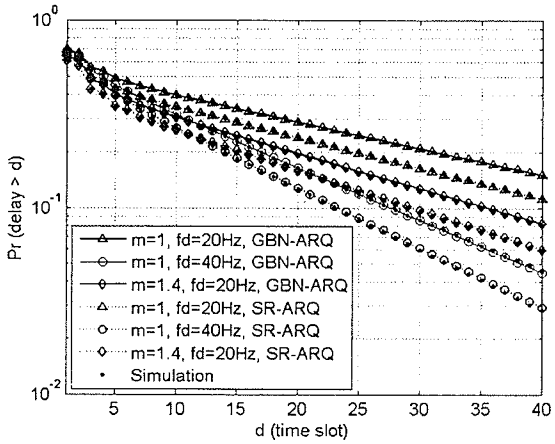


**Figure 3.5.** *Complementary cumulative delay distributions (for arrival rate $\lambda = 0.2$, feedback delay $n = 2$, average SNR $= 10$ dB, $P_0 = 0.3$, $m = 1, 1.4$, and Doppler shift $f_d = 20, 40$ Hz).*

The simulations are done by using a discrete-event simulator. For the given channel and system parameters, we calculate the channel transition probability matrix $\mathbf{T}$.

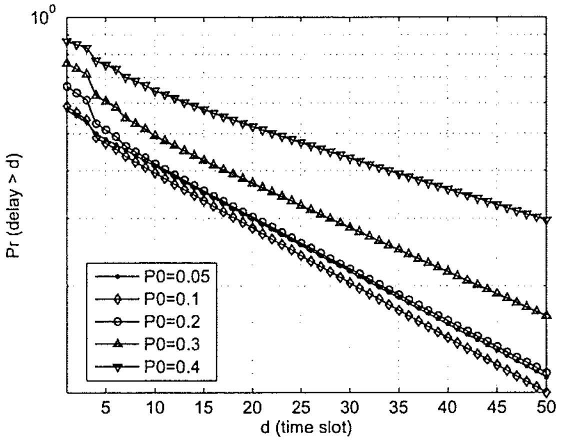


**Figure 3.6.** *Complementary cumulative delay distribution under different target packet error rate $P_0$ (for arrival rate $\lambda$ = 0.2, feedback delay $n$ = 3, average SNR = 10 dB, $m$ = 1, and Doppler shift $f_d$ = 20 Hz).*

In each time slot, the channel state is randomly generated according to the transition matrix $\mathbf{T}$, which determines the maximum number of packets that can be transmitted. For a given transmission error probability $P_0$, the number of packets reaching the receiver is determined by the corresponding probability and the ARQ protocol rule without introducing any approximation. As can be seen, the analytical results follow the simulation results very closely. We emphasize that the delay statistics for GBN-ARQ and SR-ARQ obtained here is for a very general system model which takes the dynamic radio link adaptation into account, and therefore, is more generally applicable compared to those obtained for a two-state Markov channel.

We observe that the higher the value of the Nakagami parameter $m$ and/or the Doppler shift $f_d$, the smaller the delay. The analytical model thus enables us to analyze the impact of channel parameters on the delay performance. In fact, most data applications have certain delay limits such that packets arriving after the delay limit would be useless.

An important issue in designing dynamic link adaptation mechanisms is the selection of the mode switching SNR thresholds for the different transmission modes (i.e., the SNR thresholds $X_k$, $k = 1, 2, \cdots, K$, presented in *Chapter 2*). Specifically, we have obtained the SNR thresholds such that the average packet error rate for all

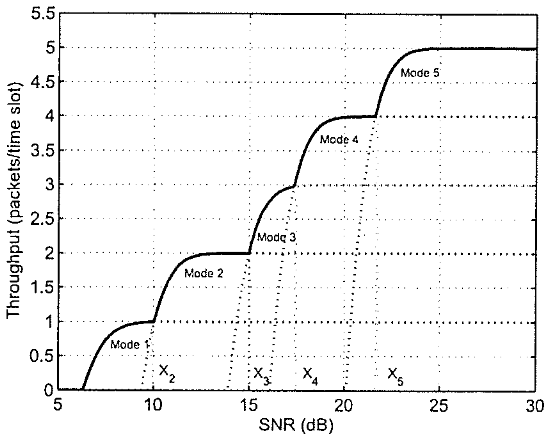


**Figure 3.7.** *Throughput for different transmission modes under varying SNR.*

modes is equal to some target packet error rate $P_0$ (i.e., $\overline{\text{PER}}_k = P_0$). Different values of $P_0$ result in different sets of SNR thresholds for the AMC in the physical layer. Variations in the complementary cumulative delay distributions for GBN-ARQ with different values of $P_0$ are illustrated in Fig. 3.6. The lowest delay is obtained for $P_0 = 0.1$ in this case. Basically, for higher values of $P_0$, the average transmission rate increases but the wireless link becomes less reliable. In other words, we are more likely to use high transmission modes by choosing large $P_0$. However, the high probability of transmission errors may require many retransmissions, which may increase the link level delay. Thus, there exists a value of $P_0$ for which the link level delay is minimized. Similar trends can also be observed for SR-ARQ under different channel and system parameters. We have observed that the best delay distribution can be obtained when $P_0$ is in the range of 0.1-0.2, and is quite insensitive to the other system and traffic parameters (e. g., $m$, $f_d$ and $\lambda$).

For dynamic link adaptation, the SNR thresholds for the different transmission modes are usually chosen based on the achieved physical layer throughput taking the packet error rate for each transmission mode into account [14]. In particular, the transmitter transmits $h_k$ packets in one time slot when the channel is in state $k$. Then the throughput (in packets/slot) is $h_k(1 - \text{PER}_k(x))$ when the received SNR is $x$ and the transmission mode is $k$. In Fig. 3.7, we plot the variations in throughput with received SNR for five transmission modes using adaptive modulation without coding.

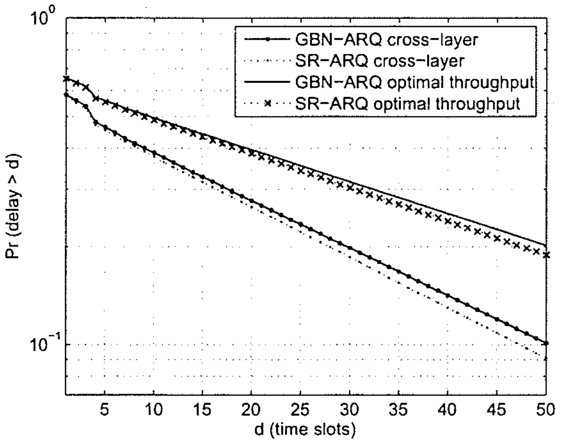


**Figure 3.8.** *Complementary cumulative delay distribution under GBN-ARQ and SR-ARQ for two mode switching threshold designs (for arrival rate $\lambda = 0.2$, feedback delay $n = 3$, average SNR = 10 dB, $P_0 = 0.1$, $m = 1$, and Doppler shift $f_d =$ 20 Hz).*

For this traditional SNR threshold design, the transmission mode which achieves the highest throughput will be chosen for each SNR value. For throughput-based link adaptation, the SNR thresholds for the different transmission modes are, therefore, determined by the intersection of these throughput curves as shown in this figure.

We now compare the delay performance obtained by the traditional SNR threshold calculation (as shown in Fig. 3.7) and our SNR threshold calculation, which is done such that the average packet error rate is equal to a certain target PER $P_0$ for all transmission modes. We denote the results obtained from the traditional SNR threshold calculation by "optimal throughput" and denote the results obtained from our SNR threshold calculation by "cross-layer" in Fig. 3.8 for both GBN-ARQ and SR-ARQ. Evidently, the delay obtained from "cross-layer" calculation is significantly lower than that obtained from "optimal throughput" calculation for both the ARQ protocols. This gain in delay performance is achieved simply by choosing the proper SNR thresholds without increasing the system complexity.

Typical variations in the complementary cumulative delay distributions for both GBN-ARQ and SR-ARQ are shown in Fig. 3.9 for different values of the feedback delay $n$. With the mode switching thresholds chosen such that the average packet

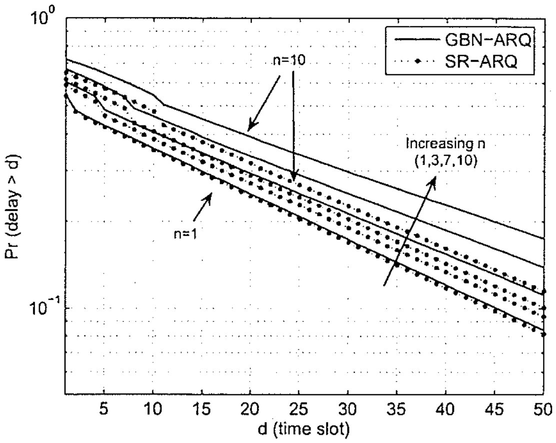


**Figure 3.9.** *Complementary cumulative delay distribution under different feedback delay $n$ (for arrival rate $\lambda = 0.2$, average SNR = 10 dB, $P_0 = 0.1$, $m = 1$, and Doppler shift $f_d = 20$ Hz).*

error rate is equal to 0.1 for all transmission modes, the delay of GBN-ARQ is only significantly larger than that of SR-ARQ when $n$ is large enough (e.g., $n \geq 10$), which justifies the use of SR-ARQ over GBN-ARQ in this region. Of course, under different choices of the SNR switching thresholds for the AMC, the average PER may be larger than 0.1 and the delay gap between these two protocols may be larger. However, such design is not optimal from the delay point of view as is evident in Fig. 3.6. Also, the delay-optimal design of the system is important because delay is the ultimate QoS perceived by some of the data applications.

For a particular wireless system, we can quantify the feedback delay, which is the sum of the propagation delay and the processing delay. Therefore, the performance gap between GBN-ARQ and SR-ARQ can be determined exactly under certain system and channel parameter settings. This would enable us to decide which ARQ protocol to implement in the link layer considering the tradeoff between delay performance and system complexity. Also, the obtained delay statistics can be used to perform packet level admission control under statistical delay constraints. For example, for a given maximum delay $D_{\max}$ and delay outage probability $P_t$, via a simple search, we can find the maximum arrival probability $\lambda_{\max}$ such that the condition $\Pr\{\text{delay} > D_{\max}\} < P_t$ is satisfied.

Before ending this section, we briefly discuss the computational complexity of the presented analytical models. We would like to emphasize that our target systems are cellular and/or wireless LAN systems where the feedback delay may not be very large. In fact, the feedback delay is the sum of the propagation and packet processing delays at the receiver. The propagation delay is generally very small compared to the time slot interval. We would expect that the processing delay is no more than a few time slots for most practical systems. Thus, the computational complexity for both models would be acceptable.

## 3.5 Chapter Summary

For a multi-rate wireless network, queueing models for GBN-ARQ and SR-ARQ have been developed under non-instantaneous feedback. The presented model removes the drawback of some of the existing works on analysis of ARQ protocols which do not capture the multi-rate feature in the physical layer. The radio link layer delay statistics thus obtained for a very general system setup enable radio link protocol design under statistical delay constraints. Also, it is useful in designing dynamic radio link adaptation thresholds based on delay performance (in contrast to the traditional methods based on throughput performance). The obtained delay statistics would be very useful to quantify the tradeoff between delay performance and implementation complexity between these two ARQ protocols.

# Chapter 4

# Analysis of Weighted Round-Robin Scheduling Scheme

When multiple users share one channel in the time multiplexing manner, a wireless scheduler is usually employed to allocate transmission time slots for the users [42], [44]. Here, a fair scheduling policy is important to guarantee that users access the channel in a fair manner. In fact, several fair scheduling schemes were proposed in the literature [35], [48]. In fact, designing a wireless scheduler which can provide differentiated services with guaranteed quality of service (QoS) for different users has been an active research topic for several years.

The generalized processor sharing (GPS) scheduling discipline [46] (also known as the weighted fair queueing) has been widely studied as an efficient way to implement differentiated services in a multi-user environment. It can guarantee the throughput share proportional to the assigned weight of each user. Unfortunately, exact delay statistics for this scheduling scheme cannot be found and only deterministic or statistical delay bounds can be derived [46], [47]. These delay bounds may not be very tight, which may lead to low resource utilization. It was shown in [46] that compared with the GPS scheme, the weighted round-robin (WRR) scheduling is simpler to implement, however, the performances of these two scheduling schemes are very close to each other.

In this chapter, we present a cross-layer analytical framework for wireless networks using WRR considering realistic physical and link layer design. In essence, the proposed framework applies to a multi-user scenario with fair radio resource sharing. In the physical layer, AMC is employed where the number of transmitted packets in one time slot varies depending on the channel condition. An ARQ protocol is employed in the link layer to counteract the residual error of an error correction code in the physical layer. The exact queue length and delay distributions are derived analytically. The impacts of different channel parameters on the system performance are investigated. We also highlight the usefulness of the analytical model by illustrating a cross-layer design and admission control example.

## 4.1 System Model and Assumptions

Suppose that there are $\mu$ separate radio link level queues at the base station (BS) which correspond to $\mu$ different mobile users. One common channel for downlink transmission is shared by all users in a time-division multiplexing (TDM) fashion. The transmission time is slotted and a WRR scheduler is used to schedule transmissions corresponding to different users. For the sake of simplicity, we assume that there are only two classes of users: high priority (class one) and low priority (class two). High priority users and low priority users receive two and one service slots in one cycle, respectively. One cycle is defined to be the smallest interval with time slot assignments for all $\mu$ users and it repeats periodically.

The receiver decodes the received packets and sends negative acknowledgments (NACKs) to the transmitter asking for retransmission of the erroneous packets (if any). In this chapter, an error-free and instantaneous feedback channel is assumed so that the transmitter knows exactly if there is any transmission error at the end of each service time slot. This assumption holds in many cases because the propagation delay and the processing time for the error detection code can be very small in comparison with the time slot interval. In fact, the effect of feedback errors can be easily included in the channel model as in [23]. The delay obtained in this chapter, therefore, can be regarded as the lower bound of the delay obtained with these effects.

As in the previous chapter, AMC is employed in the physical layer with $K$ transmission modes corresponding to different channel states. We assume that, when the channel is in state $k$, the transmitter transmits $h_k$ packets. We further assume that $h_0 = 0$ (i.e., the transmitter does not transmit in channel state 0 to avoid the high probability of transmission errors) and $h_K = N$. The channel state information (CSI) is fed back to the transmitter to choose the suitable transmission mode. The feedback channel, therefore, carries both CSI for AMC and ACK/NACK for the ARQ protocol. To capture the variations of the multi-state Nakagami fading channel, we employ the FSMC model and the average packet error rate for each mode $k$ ($\overline{\mathrm{PER}}_k$) can be calculated as being presented in *Chapter 2*.

## 4.2 Formulation of the Queueing Model

### 4.2.1 Queueing Model and Analysis

The queueing analysis for a target queue can be performed by using a vacation queueing model. When a particular target user is served in his assigned slots, the queue is assumed to be in service; otherwise, it is said to be on vacation. Since the queueing performances for different users of each class are statistically the same, we focus on one user from each class only. In the following, we analyze the queueing performance for each user class separately. For convenience, let the service period start from slot one of each cycle for each user class. Assuming that there are $\mu_1$ high priority users and $\mu_2 = \mu - \mu_1$ low priority users, one cycle consists of $L = 2\mu_1 + \mu_2$ time slots.

The queueing problem for both classes of users can be modeled in discrete time with one time interval equal to one time slot. Packet arrival is described by a batch Markovian arrival process (BMAP), which is represented by $M + 1$ sub-stochastic matrices $\mathbf{U}_m$ $(m = 0, 1, 2, \cdots, M)$ each of which has order $M_1 \times M_1$ as being described in *Chapter 2*. The buffer is assumed to be finite with size of $Q$ packets. We assume that packets arriving during time slot $t - 1$ cannot be served until time slot $t$ at the earliest. Furthermore, packet transmissions in a time slot are assumed to finish before arriving packets enter the queue. Any arriving packet which sees the full buffer will be lost.

The discrete-time Markov chain (MC) describing the system has state space $\{(q(t), a(t), u(t), c(t)),\ 0 \leq q(t) \leq Q,\ \ 1 \leq a(t) \leq M_1, 1 \leq u(t) \leq L, 0 \leq c(t) \leq K\ \}$, where $q(t)$ is the number of packets in the queue, $a(t)$ is the arrival phase, $u(t)$ is the slot in a particular cycle, and $c(t)$ is the channel state in time slot $t$. The number of packets transmitted in the service time slot $t$ is $\min\{q(t), h_{c(t)}\}$.

Let $\mathbf{P}$ and $(i, j, h, k)$ denote the transition matrix and a generic system state for this MC, respectively, and let $(i, j, h, k) \rightarrow (i', j', h', k')$ denote the transition of this

MC from state $(i, j, h, k)$ to state $(i', j', h', k')$. For fixed $i$ and $i'$, the probabilities corresponding to these state transitions can be written in matrix blocks $\mathbf{A}_{i,k}$, which correspond to transitions in level $i$ of the transition matrix. Thus, level $i$ of the transition matrix represents the system state transitions where there are $i$ packets in the queue before the transitions.

The transition matrix describing the MC is written in (4.1) for $M = 3$ and $N = 4$. Derivations of matrix blocks in (4.1) are given in **Appendix D**. In (4.1), the system state transitions $(i, *, *, *) \longrightarrow (i - k + M, *, *, *)$ are represented by $\mathbf{A}_{i,k}$ for $i < N$ and by $\mathbf{A}_k$ for $i \geq N$. These matrix blocks capture the transitions among the arrival phases, the slot number in a cycle, and the channel states for the target queue. Note that there are at most $M$ arriving packets, and at most $N$ packets can be successfully transmitted in one time slot. Therefore, the transitions can go up by at most $M$ levels and go down by at most $N$ levels.

$$\mathbf{P} = \left[\begin{array}{c|cccc|cccc|cccc}
\mathbf{A}_{0,3} & \mathbf{A}_{0,2} & \mathbf{A}_{0,1} & \mathbf{A}_{0,0} & & & & & & & & & \\
\hline
\mathbf{A}_{1,4} & \mathbf{A}_{1,3} & \mathbf{A}_{1,2} & \mathbf{A}_{1,1} & \mathbf{A}_{1,0} & & & & & & & & \\
\mathbf{A}_{2,5} & \mathbf{A}_{2,4} & \mathbf{A}_{2,3} & \mathbf{A}_{2,2} & \mathbf{A}_{2,1} & \mathbf{A}_{2,0} & & & & & & & \\
\mathbf{A}_{3,6} & \mathbf{A}_{3,5} & \mathbf{A}_{3,4} & \mathbf{A}_{3,3} & \mathbf{A}_{3,2} & \mathbf{A}_{3,1} & \mathbf{A}_{3,0} & & & & & & \\
\mathbf{A}_7 & \mathbf{A}_6 & \mathbf{A}_5 & \mathbf{A}_4 & \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & & & & & \\
\hline
 & \mathbf{A}_7 & \mathbf{A}_6 & \mathbf{A}_5 & \mathbf{A}_4 & \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & & & & \\
 & & \mathbf{A}_7 & \mathbf{A}_6 & \mathbf{A}_5 & \mathbf{A}_4 & \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & & & \\
 & & & \mathbf{A}_7 & \mathbf{A}_6 & \mathbf{A}_5 & \mathbf{A}_4 & \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & & \\
 & & & & \mathbf{A}_7 & \mathbf{A}_6 & \mathbf{A}_5 & \mathbf{A}_4 & \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \\
\hline
 & & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right]. \quad (4.1)$$

As will be seen later, for level $i \geq N$ the state transition probabilities are independent of level index $i$. Therefore, for brevity, we omit the level index in the matrix blocks. Since there are $M_1$ arrival phases, $K + 1$ channel states, and a cycle consists of $L$ slots, the order of the matrix blocks $\mathbf{A}_k$ and $\mathbf{A}_{i,k}$ is $N_1 \times N_1$, where $N_1 = M_1 L(K + 1)$. The steady state probability vector $\mathbf{x} = [\mathbf{x}_0\, \mathbf{x}_1\, \mathbf{x}_2 \cdots \mathbf{x}_Q]$, where $\mathbf{x}_i$ corresponds to level $i$ of the transition matrix can be found by blocking the transition matrix to obtain a quasi-birth and death (QBD) process which can be solved by the technique presented in *Chapter 2*.

### 4.2.2 Delay Distribution for a Low Priority User

In this section, we derive the *delay* distribution for an arriving packet to the queue of a low priority user. The delay is the time required for all packets ahead of the target packet (if any) and itself to successfully leave the queue. Because a low priority user is assumed to receive service in slot one of a cycle, counting from the end of the arrival slot, the target queue may have to be idle for a while before it is served for the first time. Let the arrival slot be numbered as slot zero. It is not included in the delay calculation. To avoid confusion, we use "slot $u$ of a cycle" to indicate the $u$th slot of a particular cycle and "slot $v$" to indicate slot $v$ from the arrival slot. Now, if slot one right after the arrival coincides with slot $u$ of a cycle, the target user is in service for the first time in slot $v$ which satisfies

$$v = \begin{cases} L - u + 2 \text{ modulo } L, & L - u + 2 \text{ indivisible by } L \\ L, & \text{otherwise.} \end{cases} \tag{4.2}$$

To calculate the delay for the target packet, we need to keep track of the channel evolution from its arriving slot to the ending slot, where it leaves the queue. The probabilities representing channel transitions and transmission outcomes can be put in the matrix form to facilitate the delay analysis. To this end, let us define the following matrices:

- $\mathbf{\Psi}_u(k,n)$ are matrices of order $(K+1)\times(K+1)$ whose elements $(\mathbf{\Psi}_u(k,n))\,(i,j)$ describe the probability that an arriving packet spends $n$ slots in the queue given that it sees $k$ packets waiting in the queue, slot one from the arrival slot coincides with slot $u$ of a cycle, the channel state is $i$ at the beginning of slot one and is $j$ at the end of slot $n$.
- $\mathbf{\Omega}(k,n)$ are matrices of order $(K+1)\times(K+1)$ whose elements $(\mathbf{\Omega}(k,n))\,(i,j)$ represent the probability that $k$ packets are successfully transmitted in $n$ slots counting from the end of the first service slot (slot $v$), starting in channel state $i$ and ending in channel state $j$.
- $\mathbf{\Gamma}^{(v)}_{k,l}$ are matrices of order $(K+1)\times(K+1)$ whose elements $(\mathbf{\Gamma}^{v}_{k,l})\,(i,j)$ represent the probability that $l$ packets are successfully transmitted in slot $v$ given that there were $k$ packets in slot one (there is no transmission from slot one to slot $v-1$), the channel state is $i$ at the beginning of slot one and is $j$ at the end of slot $v$.

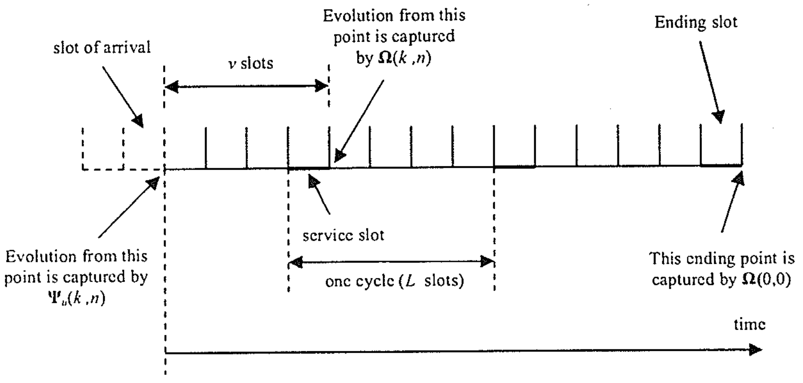

Evolution from this point is captured by $\Omega(k, n)$
Ending slot
slot of arrival
$v$ slots
service slot
Evolution from this point is captured by $\Psi_u(k, n)$
one cycle ($L$ slots)
This ending point is captured by $\Omega(0,0)$
time

**Figure 4.1.** *Delay modeling for a low priority user.*

The modeling of delay for a packet arriving to the queue of a low priority user is illustrated in Fig. 4.1. We have the following recursive relations:

$$\mathbf{\Psi}_u(k, n+v) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(v)} \mathbf{\Omega}(k-l+1, n) \tag{4.3}$$

$$\mathbf{\Omega}(k, n) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(L)} \mathbf{\Omega}(k-l, n-L) \tag{4.4}$$

$$\mathbf{\Omega}(0,0) = \mathbf{I}_{K+1}. \tag{4.5}$$

Equation (4.3) captures the case where a packet that has arrived sees $k$ packets waiting in the queue and the first transmission occurs in slot $v$ with $l$ successfully transmitted packets. Thus, there are $k-l+1$ packets in the queue including the target packet at the end of slot $v$ if we turn off the arrival source after the target packet enters the queue. These packets will successfully leave the queue in $n$ slots. Equation (4.4) describes the transmission from the end of slot $v$, where transmissions occur once in each cycle of $L$ slots. We also have $\mathbf{\Gamma}_{k,l}^{(v)} = \mathbf{T}^{v-1}\mathbf{\Lambda}_{k,l}$ where $\mathbf{\Lambda}_{k,l}$ is defined in **Appendix D**.

Suppose that the delay for the target packet is $D$ slots (not including the arrival slot). Recall that in $D$ slots, the first transmission takes place in slot $v$ ($\leq L$) from the arrival and other $J$ transmissions occur periodically, once every $L$ slots from slot $v$. We can calculate $J$ as follows:

$$J = \begin{cases} D/L - 1, & D \text{ is divisible by } L \\ \lfloor D/L \rfloor, & \text{otherwise.} \end{cases} \tag{4.6}$$

Slot $v$, when the first transmission from the arrival takes place, can be calculated as $v = D - JL$. We can calculate the corresponding $u$ from (4.2). Let $\mathbf{z}_{i,u}$ be a $(K+1)$-dimensional row vector, whose element $\mathbf{z}_{i,u}(k)$ is the probability that an arbitrary arriving packet sees $i$ packets in the queue, slot one from the arrival coincides with slot $u$ of a cycle, and the channel state is $k$ at the beginning of slot one. If a batch of $m$ $(m = 1, 2, \cdots, M)$ packets enters the queue, the target packet can be at any position in the arriving batch with probability $1/m$. In fact, if the target packet is at the $j$th position in the arriving batch, there are $j-1$ packets ahead of it in the arriving batch.

Let $\mathbf{y}_i$ be a $N_1$-dimensional row vector whose entry $\mathbf{y}_{i,a,u,k}$ is the probability that the target packet sees $i$ packets ahead of it, the arrival phase is $a$, slot one coincides with slot $u$ of a cycle, and the channel state is $k$ at the beginning of slot one. Let $\beta$ denote the probability that there is at least one packet arriving to the queue. Then, $\mathbf{y}_i$ can be calculated as follows:

$$\mathbf{y}_0 = \frac{1}{\beta} \sum_{m=1}^{M} \sum_{l=0}^{N} \frac{1}{m} \mathbf{x}_l \mathbf{U}_m \otimes \mathbf{H}_{l,l}^{(j)} \tag{4.7}$$

$$\mathbf{y}_i = \frac{1}{\beta} \sum_{m=1}^{M} \sum_{h=1}^{m} \sum_{l=0}^{N} \frac{1}{m} \mathbf{x}_{i+l-h+1} \mathbf{U}_m \otimes \mathbf{H}_{i+l-h+1,l}^{(j)} \tag{4.8}$$

where $1 \leq i \leq Q + M - 1$ and $j = 1, 2$ (each value of $j$ results in the corresponding vector $\mathbf{y}_i$ for a class-$j$ user), $\mathbf{H}_{i,k}^{(j)}$ are defined in Appendix D. For delay analysis, we are interested only in the arriving packets which are admitted into the queue. The probability that an arriving packet sees the full buffer is $P_f = \sum_{i=Q}^{Q+M-1} \mathbf{y}_i \mathbf{1}_{N_1}$. Under the admitting condition of the target packet to the queue, let $\mathbf{y}_i^{'} = \mathbf{y}_i/(1 - P_f)$, $(0 \leq i \leq Q-1)$ be the vector corresponding to $\mathbf{y}_i$.

It is easy to observe that $\mathbf{z}_{i,u}$ are actually the partitions of $\mathbf{y}_i^{'}$ where all arrival phases are lumped together. To obtain $\mathbf{z}_{i,u}$ from $\mathbf{y}_i^{'}$, let $\varphi_u$ $(u = 1, 2, \cdots, L)$ be a matrix of size $N_1 \times (K+1)$ given as

$$\varphi_u = [\underbrace{\mathbf{0} \ \cdots \ \mathbf{I}_{K+1} \ \cdots \ \mathbf{0}}_{L \text{ blocks}} \ \cdots \ \underbrace{\mathbf{0} \ \cdots \ \mathbf{I}_{K+1} \ \cdots \ \mathbf{0}}_{L \text{ blocks}}]^T$$

where there are $M_1$ groups, each consisting of $L$ blocks as indicated above, and $\mathbf{I}_{K+1}$ is the identity matrix at the $u$th position in each group. Then, we have $\mathbf{z}_{i,u} = \mathbf{y}_i^{'} \varphi_u$.

The probability that the delay for the target packet is $D$ slots (not including the arrival slot) can be written as

$$P_d^{(2)}(D) = \sum_{i=0}^{W_1} \mathbf{z}_{i,u} \mathbf{\Psi}_u(i, D) \mathbf{1}_{N_1} \tag{4.9}$$

where $W_1 = JN + N - 1$. In (4.9), the summation is limited by $W_1$ since at most $N$ packets can be successfully transmitted in one time slot.

### 4.2.3 Delay Distribution for a High Priority User

In this section, we derive the delay distribution for an arriving packet to the queue of a high priority user. For a high priority user, there are two consecutive service slots (assumed to be the first slot and the second slot in each cycle of $L$ slots). If slot one (the first slot right after the arrival slot) coincides with slot $u$ of a cycle, the target user is in service for the first time in slot $v$ which can be calculated as

$$v = \begin{cases} L - u + 2 \text{ modulo } L, & u \neq 2 \\ 1, & u = 2. \end{cases} \tag{4.10}$$

Because the target packet and its head-of-line packets can only leave the queue in either slot one or slot two of a cycle, we need to keep track the transmission outcomes in these two service slots. Also, we have to keep track the channel evolution during the vacation periods. Again, the channel transition probabilities and transmission outcomes can be captured in matrix forms to ease the analysis. Now, let us define the following matrices:

- $\mathbf{\Omega}^{(1)}(k, n)$ are matrices of order $(K+1) \times (K+1)$ whose elements $(\mathbf{\Omega}^{(1)}(k, n))\,(i, j)$ represent the probability that $k$ packets are successfully transmitted in $n$ slots counting from the end of the first slot of a cycle, starting in channel state $i$ in slot one and ending in channel state $j$ in slot $n$.
- $\mathbf{\Omega}^{(2)}(k, n)$ are matrices of order $(K+1) \times (K+1)$ whose elements $(\mathbf{\Omega}^{(2)}(k, n))\,(i, j)$ represent the probability that $k$ packets are successfully transmitted in $n$ slots counting from the end of the second slot of a cycle, starting in channel state $i$ in slot one and ending in channel state $j$ in slot $n$.

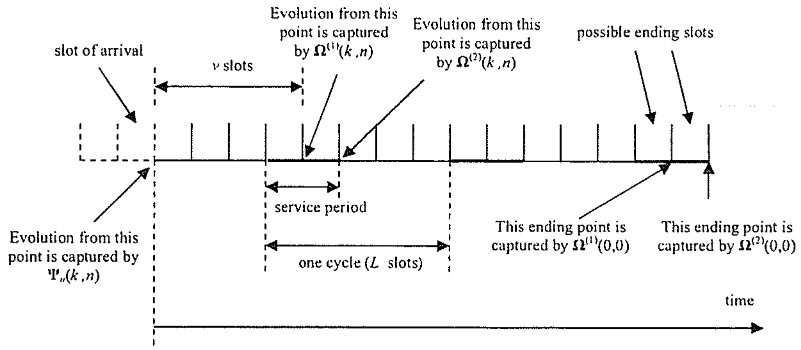

Evolution from this point is captured by $\Omega^{(1)}(k,n)$
Evolution from this point is captured by $\Omega^{(2)}(k,n)$
possible ending slots
slot of arrival
$\nu$ slots
service period
Evolution from this point is captured by $\Psi_u(k,n)$
This ending point is captured by $\Omega^{(1)}(0,0)$
This ending point is captured by $\Omega^{(2)}(0,0)$
one cycle ($L$ slots)
time

**Figure 4.2.** *Delay modeling for a high priority user.*

The modeling of delay for an arriving packet to the queue of a high priority user is illustrated in Fig. 4.2. We have the following recursive relations for these matrices:

$$\mathbf{\Psi}_u(k, n+v) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(v)} \mathbf{\Omega}^{(1)}(k-l+1, n), \text{ if } u \neq 2 \tag{4.11}$$

$$\mathbf{\Psi}_u(k, n+1) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(1)} \mathbf{\Omega}^{(2)}(k-l+1, n), \text{ if } u = 2 \tag{4.12}$$

$$\mathbf{\Omega}^{(2)}(k, n) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(L-1)} \mathbf{\Omega}^{(1)}(k-l, n-L+1) \tag{4.13}$$

$$\mathbf{\Omega}^{(1)}(k, n) = \sum_{l=0}^{N} \mathbf{\Gamma}_{k,l}^{(1)} \mathbf{\Omega}^{(2)}(k-l, n-1) \tag{4.14}$$

$$\mathbf{\Omega}^{(1)}(0,0) = \mathbf{I}_{K+1}, \quad \mathbf{\Omega}^{(2)}(0,0) = \mathbf{I}_{K+1}. \tag{4.15}$$

We can explain the above recursions as follows. Equation (4.11) describes the case where the first service after the arrival slot occurs in slot $v$, which coincides with slot one of a cycle. Equation (4.12) represents the case where the first slot after the arrival slot is slot two of a cycle; therefore, the queue is served in this slot. In both the cases, if there are $k$ packets ahead of the target packet and we turn off the arrival source after the target packet enters the queue, and $l$ packets are successfully transmitted in slot $v$, there will be $k-l+1$ remaining packets which must be transmitted successfully in $n$ slots. Equation (4.13) captures the fact that counting from the end of the second slot of a cycle, the next service occurs $L-1$ slots after that. Equation (4.14) describes the fact that from the end of the first slot of a cycle, the queue is still in service in the second slot of that cycle (because a high priority user receives service in slot one and slot two of a cycle).

We know that the target packet may leave the queue in slot one or slot two of a cycle. Suppose that the first slot after the arrival slot coincides with either slot $u_1$ or $u_2$ of a cycle for these two cases, respectively such that the delay is $D$ time slots. The probability that the delay is $D$ slots (not including the arrival slot) can be written as follows:

$$P_d^{(1)}(D) = \sum_{i=0}^{W_2} \mathbf{z}_{i,u_1} \mathbf{\Psi}_{u_1}(i, D)\mathbf{1}_{N_1} + \sum_{i=0}^{W_3} \mathbf{z}_{i,u_2} \mathbf{\Psi}_{u_2}(i, D)\mathbf{1}_{N_1}. \tag{4.16}$$

Again, the two sum-terms in (4.16) are limited by $W_2$ and $W_3$ since at most $N$ packets can be successfully transmitted in one service slot. The values of $W_2$ and $W_3$ depend on $D$ but we have $W_2$, $W_3 \leq 2N \lceil D/L \rceil$.

### 4.2.4 Extension for the General Priority Case

In this section, we extend the previous model by considering a more general scenario with more than two service classes. Suppose there are $\eta$ user classes and a class-$j$ user receives $d_j$ service slots in one cycle. If there are $\mu_j$ class-$j$ users in the system, a cycle consists of $L = \sum_{j=1}^{\eta} \mu_j d_j$ time slots. To analyze the queue corresponding to a class-$j$ user, we can assume that it receives service from slot one to slot $d_j$ of a cycle. The state space of the queue corresponding to the class-$j$ user is still $\{(q(t), a(t), u(t), c(t)),\ 0 \leq q(t) \leq Q,\ 1 \leq a(t) \leq M_1, 1 \leq u(t) \leq L, 0 \leq c(t) \leq K\ \}$. The state transition probabilities can be put in the matrix blocks for each level and the transition matrix of the MC can still be written as in (4.1).

The derivations of matrix blocks of the transition matrix for a class-$j$ user can be done in the same way as before where the matrices $\mathbf{H}_v^{(j)}$ and $\mathbf{R}_{i,k}^{(j)}$ can be written as follows:

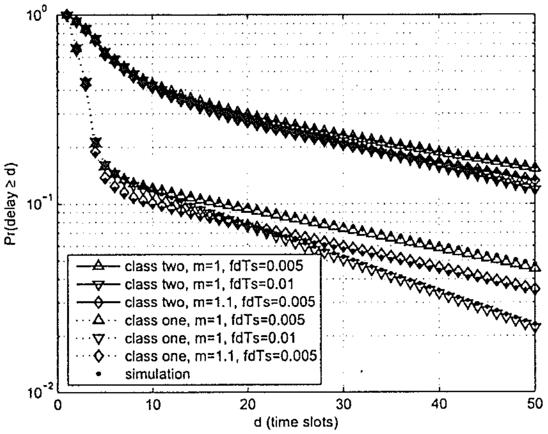


**Figure 4.3.** *Complementary cumulative delay distribution (for buffer size $Q = 70$, $L = 4$, average SNR = 12 dB, $P_0 = 0.1$, Nakagami parameter $m = 1,\ 1.1$, $f_dT_s = 0.005, 0.01$).*

$$\mathbf{H}_v^{(j)} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & \mathbf{T} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & \mathbf{T} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & \mathbf{T} \\ \mathbf{T} & 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \tag{4.17}$$

$$\mathbf{R}_{i,k}^{(j)} = \begin{bmatrix} 0 & \mathbf{\Lambda}_{i,k} & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{\Lambda}_{i,k} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \tag{4.18}$$

where there are $L$ blocks of rows in these matrices, each of them consists of $K+1$ rows, which captures the channel evolution in the corresponding time slot of a cycle. In

$\mathbf{R}^{(j)}_{i,k}$, there are $d_j$ non-zero blocks of rows, which correspond to $d_j$ service slots where $k$ packets successfully leave the queue. In $\mathbf{H}^{(j)}_v$, there are $L - d_j$ non-zero blocks of rows, which capture the channel evolution in the vacation slots. The matrix blocks $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$ can still be calculated as in (D.9)-(E.1) in **Appendix D**. To calculate the delay distribution for a class-$j$ user, we have to define $\mathbf{\Omega}^{(h)}(k, n)$, $(h = 1, \cdots, d_j)$ which captures the system evolution from the end of slot $h$ of a cycle where $k$ packets are successfully transmitted in $n$ slots. Similar recursive relations as in (4.11)-(4.15) can be developed, and delay distribution can be calculated where the target packet leaves the queue in one of the $d_j$ service slots of a cycle.

## 4.3 Validation and Applications of the Queueing Model

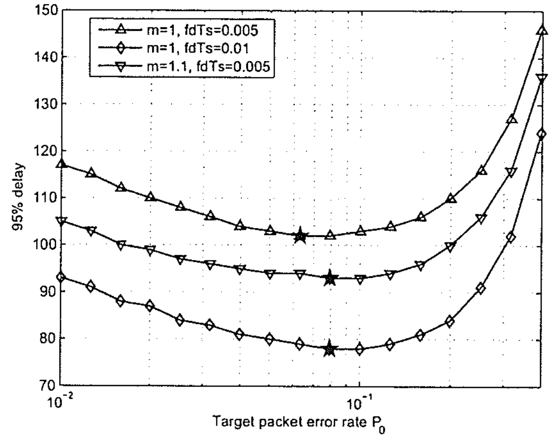


**Figure 4.4.** *The 95-th percentile delay for a class-one user versus target packet error rate $P_0$ (for buffer size $Q = 70$, $L = 4$, average SNR = 12 dB, Nakagami parameter $m = 1, 1.1$, $f_d T_s = 0.005, 0.01$).*

We assume that the SNR thresholds for the FSMC model are chosen such that $\overline{\text{PER}}_k = P_0$ and these thresholds are obtained as being presented in *Chapter 2*. We assume that $h_k = bS_k$, $b = 2$, where $S_k = 0.5, 1.0, 1.5, 3.0, 4.5$ are the spectral efficiencies of five transmission modes (see Table II in [20]). Let $f_d$ and $T_s$ denote the

Doppler shift and the time slot interval, respectively; then $f_d T_s$ represents the normalized fading rate of the wireless channel. A two-state Markovian traffic source, which is a special case of BMAP, with the following arrival state transition matrix is used to obtain the numerical results

$$\mathbf{U} = \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{bmatrix}.$$

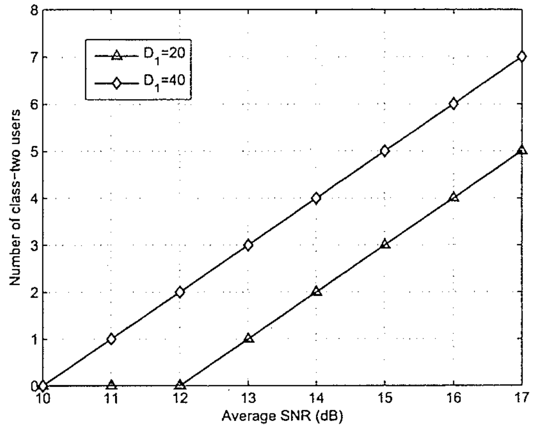


**Figure 4.5.** *Maximum admissible number of class-two users versus average SNR (for $D_1$ = 20, 40, $Q = 100$, $P_0$ = 0.1, Nakagami parameter $m = 1$, $f_d T_s = 0.005$).*

The complementary cumulative delay distributions for users of both classes obtained by simulation and from the analytical model are shown in Fig. 4.3. Note that, $\Pr\{\text{delay} \geq D\} = 1 - \sum_{i=1}^{D-1} P_d^{(j)}(i)$ for a class-$j$ user. As is evident, the simulation results follow the analytical results very closely. It can also be observed that the higher the Doppler shift $f_d$ and/or Nakagami parameter $m$, the lower the delay. The complete delay statistics obtained for both user classes enables us to design and engineer the system under statistical delay constraints. Suppose that we are interested in the 95-th percentile delay, which refers to the smallest value of $D$ such that $\Pr\{\text{delay} < D\} > 0.95$. One important design problem is how to choose the SNR thresholds for different transmission modes such that the delay at the radio link layer is minimized. In Fig. 4.4, we plot the 95-th percentile delay values versus the target packet error rate $P_0$ for different channel parameters. The optimal point is indicated

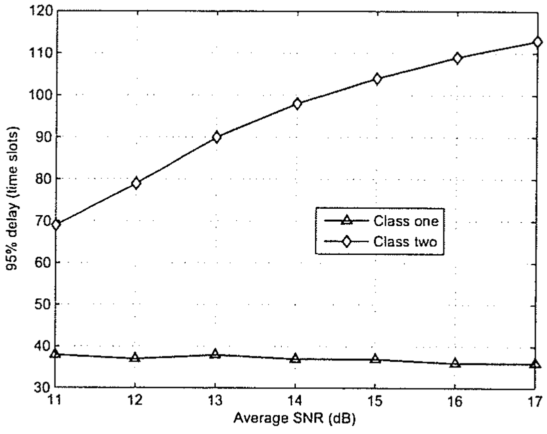


**Figure 4.6.** *The 95-th percentile delay versus average SNR with maximum admissible number of class-two users (for $D_1 = 40$, $Q = 100$, $P_0 = 0.1$, Nakagami parameter $m = 1$, $f_d T_s = 0.005$).*

by a "*big star*". Evidently, using the delay statistics, the mode switching thresholds for AMC can be selected to improve the delay performance.

The obtained delay statistics can also be used for admission control under statistical delay constraints. Typical numerical results on admission control are shown in Fig. 4.5, where one class-one user has already been admitted into the system. Under the constraint $\Pr\{\text{delay} \geq D_1\} \leq 5\%$ for a class-one user, the maximum admissible number of class-two users is determined here. With a very strict delay requirement of $D_1 = 20$, no class-two user can be admitted if the average SNR is less than 12 dB while for $D_1 = 40$, if the average SNR is greater than 11 dB, class-two users can be admitted into the system. In Fig. 4.6, we show the variations in the 95-th percentile delay for both user classes with the maximum admissible number of class-two users and one class-one user in the system. The achieved 95-th percentile delay for a class-one user is observed to be always smaller than the desired constraint ($D_1 = 40$). Since the delay for a class-two user is not constrained, it is quite large for high average SNR.

## 4.4 Chapter Summary

We have developed an analytical framework for radio link level performance evaluation of a multi-rate wireless network with weighted round-robin scheduling and ARQ-based error control. We have captured the key physical layer and the radio link layer features in the analytical model. Traffic arrival has been modeled as a batch Markovian arrival process (BMAP), which allows correlation in the arrival traffic. The probability distributions for queue length and delay have been obtained analytically, and therefore, the impacts of different channel and system parameters on the system performance can be quantified. Based on the obtained results, we have proposed an admission control policy under the statistical delay requirement. Also, a cross-layer design example has been shown to highlight the usefulness of the presented model. In summary, the presented analytical framework would be very useful for cross-layer analysis, design, and optimization of multi-rate wireless systems.

# Chapter 5

# Analysis of Opportunistic Scheduling Schemes

A new form of diversity called multi-user diversity can enhance the wireless system throughput significantly [36]-[40]. Here, multi-user diversity is achieved by an opportunistic scheduler which chooses a user with favorable channel condition to serve in each time slot. Some popular scheduling schemes can be summarized as follows. Proportional fair scheduling scheme was proposed to exploit the channel dynamics while ensuring fair access of the wireless channel for different users [36]. An optimization-based scheduling scheme was developed in [38] which ensures fairness in access time among users while maximizing the average system performance. A credit-based fair queueing scheme with a guaranteed statistical fairness bound was proposed in [39]. Compared to "perfectly fair" scheduling schemes such as WRR considered in the previous chapter, opportunistic scheduling trades fairness for throughput by exploiting wireless channel dynamics among different users.

Existing works in the literature on scheduling issues mainly focused on constructing scheduling rules under certain predefined design objectives such as maximizing throughput while providing fairness for different users. Because of these design goals, it is generally assumed that the buffers of all the backlogged users are saturated, and therefore, the buffer dynamics and the packet-level delay behavior were not investigated. In [40], a simple delay bound for a fair scheduling scheme was derived when the incoming traffic is shaped by a leaky bucket. In [49], an optimal scheduling policy was derived taking the burstiness in the traffic arrival process into account. However, the authors assumed a simple on-off channel model and considered single-rate transmission only. Also, the analysis for delay was not performed.

The queueing analysis for a general radio link level scheduling rule taking multi-rate transmission and ARQ-based error recovery into account is a very challenging problem. However, it is crucial for fair comparison among different scheduling schemes, and after all, for radio link control design and engineering in wireless systems. In this chapter, we propose an analytical model for opportunistic scheduling schemes. The analysis presented can be applied to any scheduling policy which has finite memory and the evolution of the joint service/vacation and channel processes can be determined. Here, the joint service/vacation and channel processes capture the joint transition of channel state and service state of a particular target user (i.e., the target user is served or not in the considered time slot). The exact queue length and delay distributions are derived. Based on the analysis, we derive the user throughput under saturated buffer and dynamic buffer scenarios.

The application of the analytical framework to a max rate scheduling scheme is given as a specific example. Using these results, we validate our analysis by simulation and highlight some interesting results. The usefulness of the presented model is then illustrated in a cross-layer design example and admission control of arrival traffic for wireless systems using the MR scheduling policy.

## 5.1 System Model and Assumptions

Suppose that there are $L$ separate radio link level buffers which correspond to $L$ different mobile users. These buffers can be located either at the base station (BS) in case of downlink transmission or at the mobiles in case of uplink transmission. A wireless scheduler is deployed at the BS to schedule the transmissions corresponding to the different users in a time-division multiplexing (TDM) fashion. Transmissions occur within the fixed-sized time slots and during each time slot, the scheduler grants transmission for only one user. This type of scheduling offers higher throughput performance than the one which allows simultaneous transmissions [32], [34].

Adaptive modulation and coding is employed in the physical layer with $K$ transmission modes corresponding to different channel states of a finite-state Markov channel. The channel is assumed to have $K+1$ states $(0, 1, \cdots, K)$. Each transmission mode corresponds to a pair of a modulation scheme and an error control code. Now, we assume that when the channel is in state $k$, the transmitter transmits $h_k$ packets in one time slot as in previous chapters. We further assume that $h_0 = 0$ (i.e., the transmitter does not transmit in channel state 0 to avoid high probability of transmission error), and $h_K = N$.

A block diagram of the assumed system model is shown in Fig. 5.1. The receiver estimates the signal-to-noise ratio (SNR) of the received signal and chooses the suitable transmission mode. The selected mode, which represents the channel state information (CSI), is fed back to the transmitter and may also be provided to the scheduler to make the scheduling decision.

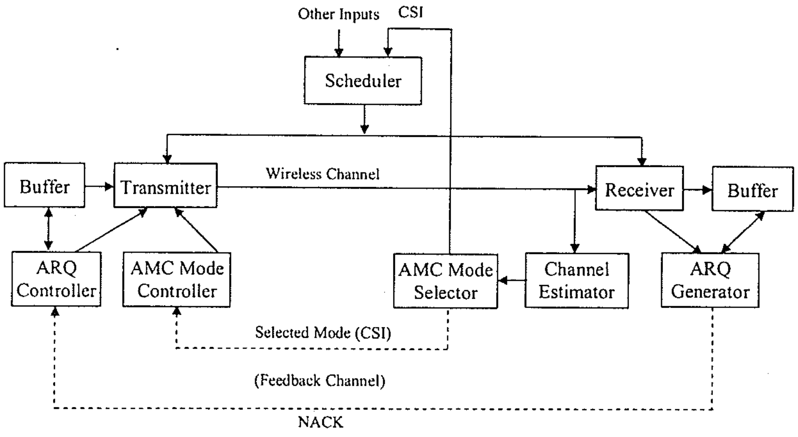


**Figure 5.1.** *System block diagram.*

The receiver decodes the received packets and it transmits negative acknowledgments (NACKs) to the transmitter asking for retransmission of the erroneous packets. An infinite persistent selective repeat ARQ protocol is assumed where the maximum number of retransmissions allowed for a packet is unbounded. The feedback channel carries both the CSI (i.e., selected transmission mode) and the NACKs of for the ARQ protocol. An error-free and instantaneous feedback channel is assumed in this chapter.

The channel is represented by the finite state Markov channel (FSMC) model. In the physical layer of the considered system, adaptive modulation (with or without coding) using QAM modulation is employed. To calculate the *packet error rate* (PER) for the AMC technique, we use the exponential approximation of the actual PER.

The average PER for mode $k$ ($\overline{\mathrm{PER}}_k$) can be calculated as described in *Chapter 2.*

## 5.2 Formulation of Queueing Model

### 5.2.1 Joint Service/Vacation and Channel Processes

The queueing analysis for a target queue can be performed by using a vacation queueing model. While a particular target user is served in a particular time slot, the queue is assumed to be in service; otherwise, it is said to be on vacation. The queueing analysis can be performed if the evolution of joint service/vacation and channel processes for the target queue can be determined. This is because packets can be transmitted from the target queue only in the service state and the channel state determines how many packets are transmitted during a particular time slot. In addition, for a channel-quality-based scheduling, the service/vacation process depends on the channel process.

We first present the analysis for the case where the service/vacation process has one-step memory while the generalization to the case with any finite memory will be outlined later in the chapter. Now, let $s(t) = 0$ represent the service state, $s_n = 1$ represent the vacation state, and $c(t)$ $(0, 1, \cdots, K)$ be the channel state in time slot $t$. The queueing framework developed later in this chapter can be applied as long as the evolution of a two-dimensional variable $(s(t), c(t))$ can be determined. Thus, the queueing analysis for any scheduling policy of interest reduces to the determination of the conditional probability $\Pr\{s(t+1), c(t+1)|s(t), c(t)\}$. To facilitate the queueing analysis, we represent this joint transition probability in the following matrix form:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{0,0} & \mathbf{S}_{0,1} \\ \mathbf{S}_{1,0} & \mathbf{S}_{1,1} \end{bmatrix} \tag{5.1}$$

where $\mathbf{S}_{i,j}$ is a $(K+1) \times (K+1)$ matrix whose elements $\mathbf{S}_{i,j}(k, l)$ are defined as follows:

$$\mathbf{S}_{i,j}(k, l) = \Pr\{s(t+1) = j, c(t+1) = l|s(t) = i, c(t) = k\}. \tag{5.2}$$

### 5.2.2 Queueing Model and Analysis

The queueing problem is modeled in discrete time with one time interval equal to one time slot. The buffer size is assumed to be infinite. The packet arrival process is assumed to be Bernoulli with arrival probability $\lambda$. We assume that packets arriving during time interval $t-1$ cannot be transmitted until time interval $t$ at the earliest. Let $q(t)$ be the number of packets in the target queue, $s(t)$ be the service/vacation state, and $c(t)$ be the channel state at the beginning of time slot $t$. The number of packets transmitted during time slot $t$ is the minimum of the number of packets in the queue $(q(t))$ and the transmission capacity in that time slot $(h_{c(t)})$ (i.e., min $\{q(t), h_{c(t)}\}$). The system state can be described by the process

$$\mathbf{X}(t) = \{(q(t), s(t), c(t)),\ q(t) \geq 0;\ s(t) = 0, 1;\ 0 \leq c(t) \leq K\},\ t \geq 0.$$

It can be shown that $\mathbf{X}(t)$ is a Markov chain (MC). Let $\mathbf{P}$ and $(i, j, h)$ denote the transition matrix and a generic system state for this MC, respectively, and $(i, j, h) \rightarrow (i', j', h')$ denote the transition of this MC from state $(i, j, h)$ to state $(i', j', h')$. For fixed $i$ and $i'$, the probabilities corresponding to these state transitions can be written in matrix blocks $\mathbf{A}_{i,k}$, which correspond to transitions in level $i$ of the transition matrix. Thus, level $i$ of the transition matrix represents the system state transitions where there are $i$ packets in the queue before the transition.

The transition matrix describing the MC is written in (5.3) where the derivation of the matrix blocks is given in **Appendix E**. Inside each level, we capture the service/vacation states and the channel states for the target queue, which are represented by $j$ and $h$ in the generic system state, respectively. In (5.3), the system state transitions $(i, *, *) \rightarrow (i-k+1, *, *)$ are represented by $\mathbf{A}_{i,k}$ for $i < N$ and by $\mathbf{A}_k$ for $i \geq N$. As can be seen from **Appendix E**, for $i \geq N$ the state transition probabilities are independent of the level index $i$, therefore, for brevity we omit the level index in the matrix blocks. Since there are two service/vacation states and $K+1$ channel states, the order of the matrix blocks $\mathbf{A}_k$ and $\mathbf{A}_{i,k}$ is $2(K+1) \times 2(K+1)$.

$$\mathbf{P} = \begin{bmatrix} \mathbf{A}_{0,1} & \mathbf{A}_{0,0} & & & & & & & \\ \mathbf{A}_{1,2} & \mathbf{A}_{1,1} & \mathbf{A}_{1,0} & & & & & & \\ \mathbf{A}_{2,3} & \mathbf{A}_{2,2} & \mathbf{A}_{2,1} & \mathbf{A}_{2,0} & & & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & & & \\ \mathbf{A}_{N-1,N} & \mathbf{A}_{N-1,N-1} & \mathbf{A}_{N-1,N-2} & \cdots & \cdots & \mathbf{A}_{N-1,0} & & & \\ \mathbf{A}_{N+1} & \mathbf{A}_{N} & \mathbf{A}_{N-1} & \cdots & \cdots & \mathbf{A}_{1} & \mathbf{A}_{0} & & \\ & \mathbf{A}_{N+1} & \mathbf{A}_{N} & \cdots & \cdots & \mathbf{A}_{2} & \mathbf{A}_{1} & \mathbf{A}_{0} & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \qquad (5.3)$$

The transition matrix in (5.3) shows that this MC is a special case of the GI/M/1 type, where the solution can be found by a well-established method [114]. Let $\mathbf{x} = [\mathbf{x}_0 \ \mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ ]$ be the steady-state probability vector corresponding to the transition matrix $\mathbf{P}$, where $\mathbf{x}_i$ corresponds to level $i$ of the transition matrix $\mathbf{P}$, and has dimension $2(K+1)$. There exists a matrix $\mathbf{R}$ which is the minimal non-negative solution to the following matrix equation: $\mathbf{R} = \sum_{k=0}^{N+1} \mathbf{R}^k \mathbf{A}_k$ such that $\mathbf{x}_i = \mathbf{x}_{i-1}\mathbf{R}$ for $i > N$. The marginal probability representing all combinations such that there are $i$ packets in the queue is just the sum of all the elements in $\mathbf{x}_i$, which is equal to $\mathbf{x}_i \mathbf{1}_{2(K+1)}$. The average queue length can be calculated as

$$\begin{aligned} L_q &= \sum_{i=1}^{\infty} i\mathbf{x}_i \mathbf{1}_{2(K+1)} = \sum_{i=1}^{N-1} i\mathbf{x}_i \mathbf{1}_{2(K+1)} + \sum_{i=0}^{\infty} (i+N)\mathbf{x}_{i+N} \mathbf{1}_{2(K+1)} \\ &= \sum_{i=1}^{N-1} i\mathbf{x}_i \mathbf{1}_{2(K+1)} + N\mathbf{x}_N \sum_{i=0}^{\infty} \mathbf{R}^i \mathbf{1}_{2(K+1)} + \mathbf{x}_N \mathbf{R} \sum_{i=0}^{\infty} (i+1)\mathbf{R}^i \mathbf{1}_{2(K+1)} \\ &= \sum_{i=1}^{N-1} i\mathbf{x}_i \mathbf{1}_{2(K+1)} + N\mathbf{x}_N (\mathbf{I}-\mathbf{R})^{-1} \mathbf{1}_{2(K+1)} + \mathbf{x}_N \mathbf{R} (\mathbf{I}-\mathbf{R})^{-2} \mathbf{1}_{2(K+1)}. \qquad (5.4) \end{aligned}$$

Using Little's law, the average delay can be written as follows:

$$D_l = \frac{L_q}{\lambda}. \qquad (5.5)$$

### 5.2.3 Delay Distribution

In this section, we derive the distribution of the total *delay* for an arriving packet to the target queue. Let the arrival slot be numbered as slot zero. This slot is not included in the delay calculation. If an arriving packet sees $i$ head-of-line packets in the queue, the delay for this target packet is the time required for $i + 1$ packets ($i$ head-of-line packets and the target packet itself) to successfully leave the queue.

Now, to calculate the delay distribution, we need to determine the steady-state distribution seen by an arriving packet and the system evolution from the beginning of slot one to the ending point where the target packet successfully leaves the queue. Since a system state is represented by the service/vacation states and the channel states, the probabilities describing the system state evolution in each time slot during the entire period in which the target packet is in the queue can be put in a matrix form to facilitate the analysis. To this end, let us define the following matrices:

- $\mathbf{\Omega}(i, D) = \begin{bmatrix} (\mathbf{\Omega}(i,D))(0,0) & (\mathbf{\Omega}(i,D))(0,1) \\ (\mathbf{\Omega}(i,D))(1,0) & (\mathbf{\Omega}(i,D))(1,1) \end{bmatrix}$ is a $2(K+1) \times 2(K+1)$ matrix whose elements $(\mathbf{\Omega}(i, D))\,(i_1, i_2), (j_1, j_2)\ (i_1, i_2 = 0, 1;\ 0 \leq j_1, j_2 \leq K)$ represent the probability that $i$ packets are successfully transmitted in $D$ slots, service/vacation starts in state $i_1$ and finishes in state $i_2$, the transmission starts in channel state $j_1$ and finishes in channel state $j_2$ at the end of slot $D$.
- $\mathbf{H}_{i,k} = \begin{bmatrix} (\mathbf{H}_{i,k})(0,0) & (\mathbf{H}i,k)(0,1) \\ (\mathbf{H}_{i,k})(1,0) & (\mathbf{H}_{i,k})(1,1) \end{bmatrix}$ is a $2(K+1) \times 2(K+1)$ matrix whose elements $(\mathbf{H}_{i,k})(i_1, i_2), (j_1, j_2)\ (i_1, i_2 = 0, 1;\ 0 \leq j_1, j_2 \leq K)$ represent the probability that $k$ packets are successfully transmitted in one particular slot given that there are $i$ packets in the queue at the beginning of the slot, service/vacation state changes from state $i_1$ to state $i_2$, the transmission starts in channel state $j_1$ and finishes in channel state $j_2$.

Note that, $(\mathbf{H}_{i,k})(i_1, i_2)$ and $(\mathbf{\Omega}(i, D))(i_1, i_2)$ have order $(K + 1) \times (K + 1)$, whose elements capture the channel state transition. We have the following recursive relations:

$$\mathbf{\Omega}(i, D) = \sum_{k=0}^{N} \mathbf{H}_{i,k} \mathbf{\Omega}(i - k, D - 1) \tag{5.6}$$

$$\mathbf{\Omega}(0, 0) = \mathbf{I}_{2(K+1)} \tag{5.7}$$

where $\mathbf{H}_{i,k}$ is calculated in the **Appendix E**. We can explain the above recursive relations as follows. If there are $i$ packets which need to be transmitted in $D$ slots, and $k$ packets are successfully transmitted in the current slot, there remains $i - k$ packets to be transmitted in $D - 1$ slots. $\mathbf{\Omega}(0, 0)$ simply captures the ending point where the target packet leaves the queue.

Now, let the steady-state probability vector seen by an arriving packet be denoted by $\mathbf{y} = [\mathbf{y}_0\, \mathbf{y}_1\, \mathbf{y}_2\, \cdots]$. Then, we have

$$\mathbf{y}_i = \sum_{k=0}^{N} \mathbf{x}_{i+k}\mathbf{H}_{i+k,k}. \tag{5.8}$$

Equation (5.8) can be interpreted as follows. If there are $i + k$ packets ahead of the target packet in the queue at the beginning of the arrival slot and $k$ packets are successfully transmitted in the arrival slot, then the target packet sees exactly $i$ head-of-line packets at the beginning of the next time slot. The probability that the delay is $D$ slots (not including the arrival slot) can be written as follows:

$$P_d(D) = \sum_{i=0}^{DN-1} \mathbf{y}_i \Omega(i+1, D)\mathbf{1}_{2(K+1)}. \tag{5.9}$$

The above summation is limited to $DN-1$ since at most $N$ packets can be successfully transmitted in one time slot.

### 5.2.4 Throughput Calculation

In this section, we calculate the throughput for the target user in terms of number of successfully transmitted packets per time slot under saturated buffer and dynamic buffer scenarios. The latter refers to the situation where the queueing dynamics are taken into account. For this case, the buffer occupancy (queue length) distribution has been derived in Section 5.2.2. In contrast, under the saturated buffer scenario the queue is assumed to be highly loaded at all times and the service operation is determined by the service/vacation and channel states, where the number of packets available in the queue is always greater than the transmission capacity of the channel (i.e., $N$ packets). Assuming that packet errors are independent, the average packet error rate, which is the ratio between the average number of packets in error and the average number of transmitted packets, is given by [21]

$$p = \frac{\sum_{k=1}^{K} h_k \Pr(k)\overline{\text{PER}}_k}{\sum_{k=1}^{K} h_k \Pr(k)}. \tag{5.10}$$

The average number of transmissions for each packet can be written as

$$\overline{N} = \sum_{k=0}^{\infty} p^k = \frac{1}{1-p}. \tag{5.11}$$

Note that, ARQ is employed to retransmit erroneous packets and $p$ represents the packet retransmission probability. Under the saturated buffer scenario, the throughput is determined by the joint service/vacation and the channel processes. Let us define vector $\mathbf{z}$ which satisfies $\mathbf{zS} = \mathbf{z}$ and $\mathbf{z1}_{2(K+1)} = 1$. We can partition $\mathbf{z}$ as follows: $\mathbf{z} = [\mathbf{z}_0,\ \mathbf{z}_1] = [\mathbf{z}_0(0),\ \cdots,\mathbf{z}_0(K),\ \mathbf{z}_1(0),\ \cdots,\mathbf{z}_1(K)]$. In fact, $\mathbf{z}$ is a row vector of dimension $2(K+1)$; $\mathbf{z}_0(k)$ and $\mathbf{z}_1(k)$ represent the probability that the target queue is in service and vacation, respectively, where the channel is in state $k$. The throughput for the target user under the saturated buffer case can be calculated as

$$\mathrm{TP}_s = \frac{\sum_{k=1}^{K} h_k \mathbf{z}_0(k)}{\overline{N}} = (1-p)\sum_{k=1}^{K} h_k \mathbf{z}_0(k). \tag{5.12}$$

Similarly, the steady-state probability vector $\mathbf{x}_i$ of (5.3) can be partitioned as $\mathbf{x}_i = [\mathbf{x}_{i,0},\ \mathbf{x}_{i,1}] = [\mathbf{x}_{i,0}(0),\ \cdots,\ \mathbf{x}_{i,0}(K),\ \mathbf{x}_{i,1}(0),\ \cdots,\mathbf{x}_{i,1}(K)\ ]$. Taking the buffer dynamics into account, the throughput of the target user is given by

$$\mathrm{TP}_b \ = \ \frac{\sum_{i=1}^{\infty}\sum_{k=1}^{K} \min\{i, h_k\}\mathbf{x}_{i,0}(k)}{\overline{N}} = (1-p)\sum_{i=1}^{\infty}\sum_{k=1}^{K} \min\{i, h_k\}\mathbf{x}_{i,0}(k). \tag{5.13}$$

### 5.2.5 Service/Vacation Process with Finite Memory

In the previous sections, the queueing analysis was performed for service/vacation process with one-step memory. In this section, we extend the above analysis for a more general case where the service/vacation process has more than one-step memory. This may be the case for scheduling schemes which aim at providing fairness among the different flows. To achieve the fairness goal, the service/vacation processes for all the backlogged flows are retained in the memory over a window of time slots so that the lagging flows can be prioritized over the leading flows to ensure fairness among users in terms of throughput or channel access time.

We consider the case where the service/vacation process has a general finite $M$-step memory. That is, the evolution of the joint service/vacation and channel processes is captured in the probability $\Pr\{s(t+1),\ c(t+1)|s(t), s(t-1), \cdots, s(t-M+1), c(t)\ \}$. Defining $\mathbf{v}(t) = [s(t-M+1), \cdots, s(t-1), s(t)]$, where $s(t-k) \in \{0,1\}$ ($k \in \{0, 1, \cdots, M-1\}$), the discrete-time MC describing the system has state space $\{(q(t), \mathbf{v}(t), c(t));\ q(t) \geq 0,\ \ 0 \leq c(t) \leq K\ \}$. Now, each level in the corresponding transition matrix of this MC captures all combinations of $\mathbf{v}(t)$ and $c(t)$. Since $\mathbf{v}(t)$ has $2^M$ combinations and $c(t)$ has $K+1$ combinations (i.e., different channel states), the order of matrix blocks $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$ in the transition matrix is $2^M(K+1)\times 2^M(K+1)$. Although the state space increases, the joint service/vacation and channel transition probabilities $\Pr\{s(t+1), c(t+1)|s(t), s(t-1), \cdots, s(t-M+1), c(t)\}$ can be put into the matrix form and the above analysis can still be applied.

## 5.3 Analysis for Max-Rate Scheduling Scheme

The max-rate scheduling scheme works as follows. At any time slot, the channel states (each state is one of the $K+1$ states of the FSMC) of all active users are assumed to be available at the scheduler without delay. Although this assumption may not be strictly valid due to feedback delay, the performance degradation is very small if the channel is quite static over a short period of time. We further assume that the channel processes for all the users are independent. This assumption often holds in practice because of the location-dependent characteristics of the wireless channel. The max-rate scheduler grants the transmission to the user with the highest rate. If there are more than one user with the highest channel state, the scheduler chooses one of them randomly.

As has been mentioned before, the application of the presented model to any scheduling rule of interest reduces to determination of the joint transition matrix $\mathbf{S}$. The order of matrix $\mathbf{S}$ depends on the memory length $(M)$ of the joint service/vacation and channel processes. For max-rate scheduling scheme, $M=1$, and therefore, we only need to find the joint conditional probabilities $\Pr\{s(t+1), c(t+1)|s(t), c(t)\}$. Let $s^{(i)}(t)$ and $c^{(i)}(t)$ $(i=1,\cdots,L)$ denote the service/vacation state and the channel state for user $i$ in time slot $t$. We consider queue one, where data packets corresponding to user one are buffered. For notational simplicity, let $s^{(1)}(t)=s(t)$ and $c^{(1)}(t)=c(t)$. We have

$$\Pr\{s(t+1), c(t+1)|s(t), c(t)\} = \Pr\{s(t+1)|s(t), c(t+1), c(t)\} \times \Pr\{c(t+1)|c(t)\} \tag{5.14}$$

where we have used the fact that $\Pr\{c(t+1)|s(t), c(t)\} = \Pr\{c(t+1)|c(t)\}$ since the channel process is independent of the service/vacation process. Here, $\Pr\{c(t+1)=l|$

$c(t) = k\} = T_{kl}$ is the channel state transition probability, which is available from the FSMC model. We also have

$$\begin{aligned}\Pr\{s(t+1)|s(t), c(t+1), c(t)\} &= \frac{\Pr\{s(t+1), s(t)|c(t+1), c(t)\}}{\Pr\{s(t)|c(t+1), c(t)\}} \\ &= \frac{\Pr\{s(t+1), s(t)|c(t+1), c(t)\}}{\Pr\{s(t)|c(t)\}} \qquad (5.15)\end{aligned}$$

where $\Pr\{s(t)|c(t+1), c(t)\} = \Pr\{s(t)|c(t)\}$ since the service state in time slot $t$ depends only on the channel state in time slot $t$. Since $s(t+1)$ can be either 0 or 1, we have

$$\begin{aligned}&\Pr\{s(t+1) = 1-j|s(t) = i, c(t+1) = l, c(t) = k\} \\ &\qquad = 1 - \Pr\{s(t+1) = j|s(t) = i, c(t+1) = l, c(t) = k\}. \quad (5.16)\end{aligned}$$

Therefore, we need to consider only the case when $s(t+1) = 0$. Let us calculate the denominator and the numerator of (5.15) in the following sections.

### 5.3.1 Calculation of $\Pr\{s(t)|c(t)\}$

We can write the denominator of (5.15) with $s(t) = 0$ as follows:

$$\Pr\{s(t) = 0|c(t) = k\} = \sum_{c^{(2)}(t)=0}^{K} \sum_{c^{(3)}(t)=0}^{K} \cdots \sum_{c^{(L)}(t)=0}^{K} \Pr\left\{s(t) = 0, c^{(2)}(t), \cdots, c^{(L)}(t)|c(t) = k\right\}. \qquad (5.17)$$

Here, $\Pr\left\{s(t) = 0, c^{(2)}(t), \cdots, c^{(L)}(t)|c(t) = k\right\}$ can be calculated as

$$\Pr\left\{s(t) = 0, c^{(2)}(t), ..., c^{(L)}(t)|c(t) = k\right\} = \begin{cases} 0, & \text{if } \exists i, (2 \leq i \leq L) \text{ s.t. } c^{(i)}(t) > k \\ \frac{1}{a}\prod_{i=2}^{L} \Pr\left\{c^{(i)}(t)\right\}, & \text{otherwise} \end{cases} \qquad (5.18)$$

where $a$ is the number of users such that $c^{(i)}(t) = k$, $\Pr\left\{c^{(i)}(t)\right\}$ is the channel state probability for user $i$, which is given by the FSMC model.

Equation (5.18) can be interpreted as follows. In time slot $t$, user one is in service, given that its channel is in state $k$ when all other $L-1$ users have channel state lower than or same as that for user one (i.e., state $k$). If there are $a$ users with the same channel state $k$, user one is chosen for transmission with probability $1/a$. Note that, this equation holds because we assume that channel processes of all users are independent. For $s(t) = 1$, $\Pr\{s(t) = 1|c(t) = k\}$ can be written as follows:

$$\Pr\{s(t) = 1|c(t) = k\} = 1 - \Pr\{s(t) = 0|c(t) = k\}. \tag{5.19}$$

### 5.3.2 Calculation of $\Pr\{s(t+1), s(t)|c(t+1), c(t)\}$

The numerator of (5.15) can be written as

$$\Pr\{s(t+1) = s, s(t) = v|c(t+1) = l, c(t) = k\} = \sum_{c^{(2)}(t+1)=0}^{K} \cdots \sum_{c^{(L)}(t+1)=0}^{K} \sum_{c^{(2)}(t)=0}^{K} \cdots \sum_{c^{(L)}(t)=0}^{K}$$
$$\Pr\left\{s(t+1) = s, s(t) = v, c^{(2)}(t+1), ..., c^{(L)}(t+1), c^{(2)}(t), ..., c^{(L)}(t)|c(t+1) = l, c(t) = k\right\} \tag{5.20}$$

where the following two cases are considered for the terms inside the summations.

**Case I:** $s(t+1) = 0, s(t) = 0$

For this case, the corresponding term inside the summations in (5.20) can be written as

$$\Pr\left\{s(t+1) = 0, s(t) = 0, c^{(2)}(t+1), ..., c^{(L)}(t+1), c^{(2)}(t), ..., c^{(L)}(t)|c(t+1) = l, c(t) = k\right\}$$
$$= \begin{cases} 0, & \text{if } \exists i, (2 \le i \le L) \text{ s.t. } c^{(i)}(t+1) > l \text{ or } c^{(i)}(t) > k \\ \frac{1}{ab} \prod_{i=2}^{L} \Pr\left\{c^{(i)}(t+1), c^{(i)}(t)\right\}, & \text{otherwise} \end{cases} \tag{5.21}$$

where $a$ and $b$ are the number of users (including the target user) having channel states satisfying $c^{(i)}(t) = k$ and $c^{(i)}(t+1) = l$, respectively.

Equation (5.21) can be interpreted as follows. User one is in service in two consecutive time slots $t$ and $t+1$ given that the channel state for user one is $k$ and $l$ in time slots $t$ and $t+1$, respectively, when the channel states for all other $L-1$ users are smaller than or equal to $k$ and $l$ in time slots $t$ and $t+1$, respectively. If there are $a$ and $b$ users (including the target user) with channel states such that $c^{(i)}(t) = k$ and $c^{(i)}(t+1) = l$, respectively, user one is granted transmission in both time slots with probability $1/ab$.

**Case II:** $s(t+1) = 0, s(t) = 1$

For this case, the corresponding term inside the summations in (5.20) can be written as

$$\Pr\left\{s(t+1)=0, s(t)=1, c^{(2)}(t+1), ..., c^{(L)}(t+1), c^{(2)}(t), ..., c^{(L)}(t) | c(t+1)=l, c(t)=k\right\}$$
$$= \begin{cases} 0, \quad \text{if } \left\{\exists i, (2 \le i \le L) \text{ s.t. } c^{(i)}(t+1) > l\right\} \text{ or } \left\{c^{(i)}(t) < k, \ \forall i, (2 \le i \le L)\right\} \\ \frac{1}{b}\prod_{i=2}^{L} \Pr\left\{c^{(i)}(t+1), c^{(i)}(t)\right\}, \quad \text{if } \exists i, (2 \le i \le L) \text{ s.t. } c^{(i)}(t) > k \\ \frac{a-1}{ab}\prod_{i=2}^{L} \Pr\left\{c^{(i)}(t+1), c^{(i)}(t)\right\}, \quad \text{otherwise} \end{cases} \tag{5.22}$$

where $a$ and $b$ are the number of users including the target user with channel states satisfying $c^{(i)}(t) = k$ and $c^{(i)}(t+1) = l$, respectively.

To explain (5.22), we consider the following cases. First, if there are users with channel state higher than that for user one in time slot $t+1$ or all other $L-1$ users have channel state lower than that for user one in time slot $t$, user one will not be in service in slot $t+1$ and will not be on vacation in slot $t$. Second, if there are users with higher channel state than that for user one in time slot $t$ and there are $b$ users having the same channel state same as that of user one (other users have lower channel states), user one is on vacation in time slot $t$ and in service in time slot $t+1$ with probability $1/b$. Third, if there are $a$ users with channel state same as that for user one in slot $t$ (other users have lower channel states) and there are $b$ users with channel state same as that for user one in slot $t+1$ (other users have lower channel states), the probability that user one is on vacation in slot $t$ and in service in slot $t+1$ is $(a-1)/ab$. In (5.21) and (5.22), $\Pr\left\{c^{(i)}(t+1) = l, c^{(i)}(t) = k\right\} = \Pr\left\{c^{(i)}(t+1) = l | c^{(i)}(t) = k\right\} \times \Pr\left\{c^{(i)}(t) = k\right\}$, which can be calculated using the transition probability matrix for the FSMC model.

# 5.4 Numerical and Simulation Results: Model Validation and Useful Implications

## 5.4.1 System Parameters and Assumptions

In this section, we present typical numerical results considering an uncoded wireless system with five transmission modes. The fitting parameters of PER are taken from

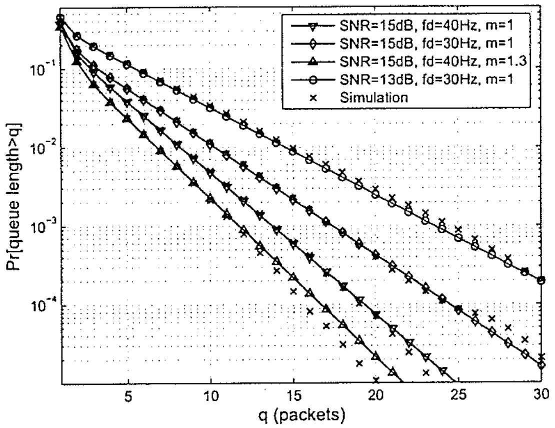


**Figure 5.2.** *Complementary cumulative distribution of queue length (for packet arrival probability* $\lambda = 0.1$*,* $L = 3$*, average SNR = 13, 15 dB,* $P_0$ *= 0.1,* $m = 1, 1.3$*, and* $f_d = 30, 40$ *Hz).*

[20] for the uncoded wireless system. We assume that $h_k = k$, time slot interval $T_s = 0.5$ $ms$ and the SNR thresholds of the FSMC model are found such that the average packet error rate $\overline{\text{PER}}_k = P_0$ $(k = 1, \cdots, 5)$ as in [20]. To save the simulation time in validating the analytical results, only two transmission modes are used, i.e., $K = 2$ (in Figs. 5.2-5.3). All other results are obtained with five transmission modes.

### 5.4.2 Simulation Methodology

The simulation results are obtained for the tagged user (i.e., user one) for the max-rate opportunistic scheduling scheme as follows. Given the system and channel parameters, the channel transition matrix $\mathbf{T}$ is calculated. The simulation run time is chosen to be $5 \times 10^6$ time slots, where in each time slot the channel states of all users are generated based on their channel states in the previous time slot and the corresponding channel state transition probabilities. User one transmits if the channel state for user one is higher than that for each of the other users (ties are broken randomly). The number of packets transmitted during any service time slot is determined by the channel state. The number of packets successfully leaving the queue is determined based on the packet error probability $P_0$. The queue length is updated at every

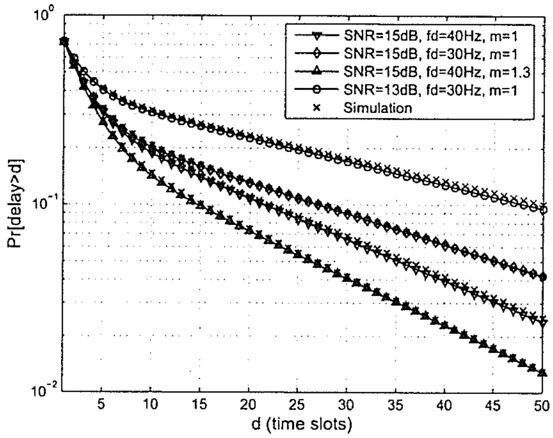


**Figure 5.3.** *Complementary cumulative distribution of delay (for packet arrival probability* $\lambda = 0.1$*,* $L = 3$*, average SNR = 13, 15 dB,* $P_0 = 0.1$*,* $m = 1, 1.3$*, and* $f_d = 30, 40$ *Hz).*

time slot by considering a packet arrival (which follows a Bernoulli process) and the number of successfully transmitted packets.

### 5.4.3 Queue Length and Delay Distributions

The complementary cumulative distributions for queue length and delay obtained from the analytical model and simulation are shown in Fig. 5.2 and Fig. 5.3, respectively. As is evident from these two figures, the simulation results match the analytical results very well. The effects of Doppler shift $f_d$ and Nakagami parameter $m$ on the delay distribution are also shown. As can be observed, with higher Doppler shift $f_d$ or Nakagami parameter $m$ the delay decreases. This implies that higher channel correlation adversely affects the delay performance. As expected, lower average SNR results in higher delay since the average transmission rate of the target user decreases.

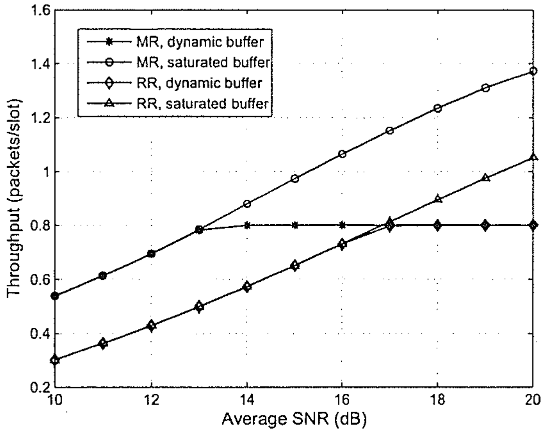


**Figure 5.4.** *Throughput of RR and MR scheduling schemes versus average SNR (for packet arrival probability* $\lambda = 0.8$*,* $L = 3$*,* $P_0 = 0.1$*,* $m = 1$*, and* $f_d = 40$ *Hz).*

### 5.4.4 Impact of Scheduling on Delay and Throughput Performance

We compare the throughput and the delay performances of the max-rate (MR) and the round-robin (RR) scheduling schemes. The analytical model for the RR scheduling is given in *Chapter 4*. Typical variations in throughput performance with average SNR for the MR and the RR scheduling schemes are presented in Fig. 5.4 under saturated buffer and dynamic buffer scenarios. As expected, the MR scheduling always provides higher throughput than the RR scheduling. This is due to the fact that MR scheduling exploits the channel fluctuations to enhance the throughput. However, it does not provide fairness among different users. In contrast, the RR scheduling ensures perfect fairness among the users, however, it does not take advantage of the multiuser diversity to achieve throughput gain.

Note that, under perfect power control the average channel conditions for all users are similar, and therefore, the MR scheduling provides long-term fairness while optimally increasing the system throughput via exploitation of small-scale fading. In general, these two scheduling rules are the two extremes from the viewpoint of the trade-off between throughput and fairness. The throughput performance under dynamic buffer case is always bounded by that for the saturated buffer case.

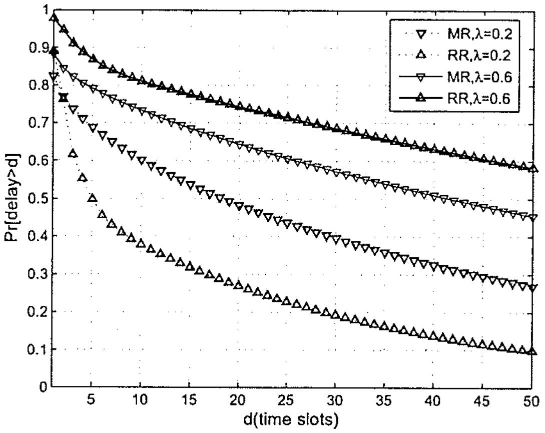


**Figure 5.5.** *Complementary cumulative distribution of delay for RR and MR scheduling schemes (for packet arrival probability* $\lambda = 0.2, 0.6$*,* $L = 3$*, average SNR = 15 dB,* $P_0$ *= 0.1,* $m = 1$*, and* $f_d = 40$ *Hz).*

The complementary cumulative delay distributions for MR and RR schemes are shown in Fig. 5.5 for packet arrival probability $\lambda = 0.2$ and $\lambda = 0.6$. Interestingly, under light traffic load conditions (e.g., $\lambda = 0.2$), RR scheduling offers much smaller delay than the MR scheduling, while under heavy traffic load conditions, the MR scheme offers better delay performance.

Impact of channel correlation on the delay performance for the MR and the RR scheduling schemes are shown in Fig. 5.6. We plot both average delay and the 95-th percentile delay which is denoted as "95% delay" in this figure. We observe that the impact of channel correlation on the performance of the MR scheme is more significant than that for the RR scheme. In fact, for the MR scheme, if the channel correlation is high, a particular user may gain or lose service for a long period of time which adversely affects the delay performance.

The above results have notable implications on the design of a scheduling policy. For example, in a correlated fading channel, a 'greedy' scheduling scheme such as MR scheduling does not ensure short-term fairness and it may result in undesirable delay performance under light traffic load conditions. The design of the scheduling policy, therefore, should be based on the QoS performance objectives considering different traffic load and channel conditions. The analytical framework presented in

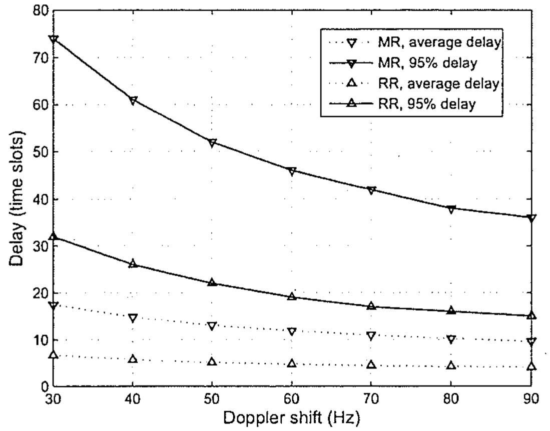


**Figure 5.6.** *Average and 95% delay of RR and MR scheduling schemes versus Doppler shift* $f_d$ *(for packet arrival probability* $\lambda = 0.2$*,* $L = 3$*, average SNR = 15 dB,* $P_0$ *= 0.1,* $m = 1$*).*

this chapter would be very useful for complete evaluation of a scheduling scheme under different traffic, channel and system conditions.

### 5.4.5 Example Applications

#### 5.4.5.1 Cross-Layer Design

Since the exact queueing performance considering the radio link layer and the physical layer aspects has been obtained, cross-layer design can be performed to optimally choose the radio link control parameters. To illustrate, we plot the average delay versus the target packet error rate $P_0$ in Fig. 5.7. The delay can be minimized when $P_0$ is in the range 0.05-0.1. The average transmission rate increases with increasing $P_0$, however, the probability of transmission failure also increases at the same time. Therefore, there exists a value of $P_0$ for which the delay performance is optimal.

#### 5.4.5.2 Packet-Level Admission Control

For some data services, a delay requirement needs to be satisfied since there is a delay limit over which a packet will be useless even if it reaches the destination correctly.

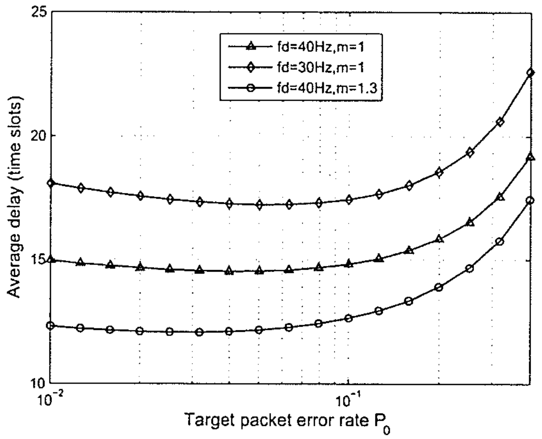


**Figure 5.7.** *Average delay versus $P_0$ (for packet arrival probability $\lambda = 0.1$, $L = 3$, average SNR = 15 dB, $m = 1, 1.3$, and $f_d = 30, 40$ Hz).*

The complete delay statistics obtained from the analytical model above enables us to perform admission control under statistical delay constraint of the following form: $\Pr(\text{delay} > D_{\max}) = 1 - \sum_{k=1}^{D_{\max}} P_d(k) \leq P_t$. For a certain setting of channel and system parameters, the admission control parameter $\gamma$, which can be, for example, the admissible number of users or the packet arrival rate, can be found by a simple search such that the delay constraint is satisfied.

As an example, in Fig. 5.8, we show typical variations in the maximum packet arrival probability with average SNR. As is evident, higher Doppler shift and/or larger average SNR reduce delay thus allowing packets to be admitted into the queue at a higher rate.

## 5.5 Chapter Summary

We have presented a queueing analytical framework for analyzing radio link level buffer dynamics under a general scheduling rule in a multi-rate wireless network employing ARQ-based error control under correlated fading channel. We have derived the exact distributions for queue length and delay and also obtained the user throughput under both saturated buffer and dynamic buffer cases. The specific case of the

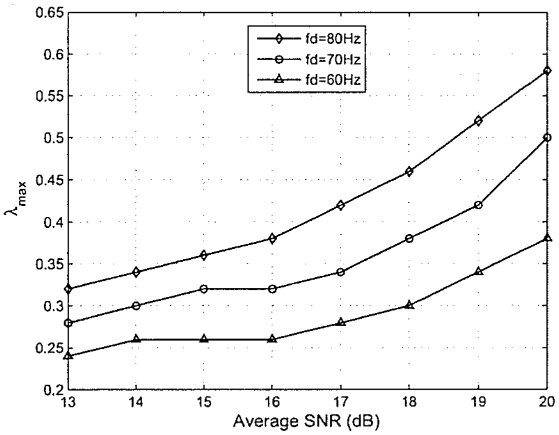


**Figure 5.8.** *Variations in maximum packet arrival probability with average SNR under statistical delay constraint (for $D_{\max} = 70$, $P_0 = 0.1$, $L = 3$, $m = 1$, and $f_d = 60, 70, 80$ Hz).*

max-rate scheduling has been analyzed by using the general analytical framework. To highlight the usefulness of the presented model, we have shown examples of cross-layer design and admission control under statistical delay constraints, based on the results obtained from the analytical model. This analytical framework would establish the base for fair comparison among different scheduling schemes and facilitate performance prediction in the higher layers of the protocol stack.

# Chapter 6

# Tandem Queue and QoS Routing Framework

In this chapter, we tackle a single-path QoS routing problem for multi-hop wireless networks. For QoS routing, the performance metrics should be measured or calculated in a timely manner so that the routing algorithm can adapt to the system dynamics. In fact, QoS routing has been an active research topic over the past several years. Typical routing metrics adopted in the literature are bandwidth and delay [54], [55], [57]. While bandwidth can usually be quantified, the existing works in the literature usually assume that routing algorithms have the capability to estimate the link delay. Also, the end-to-end loss rate is usually ignored.

Since on-demand routing scales well to the network size, most of the routing algorithms adopted by the IETF's MANET working group belong to this routing category. For on-demand QoS routing algorithms, link and path quality metrics are incorporated into the route discovery phase to find good routes for an incoming connection. In [55] and [57], the authors assumed that link delay can be estimated/measured with certain uncertainty. In practice, QoS metrics such as delay and loss rate can be calculated under realistic physical and link layer design considering the queueing dynamics at each node along the route to the destination. This leads to a tandem queueing problem which is the focus of this chapter.

A tandem queueing model is also useful for evaluating end-to-end performance for multi-hop wireless networks [71]. There are some tandem queueing models proposed in the literature. Tandem systems of two queues were modeled in discrete time in [72]. The end-to-end delay for time-division multiple access (TDMA) and ALOHA multiple access schemes was approximately derived in [73] for multi-hop networks assuming constant bit rate traffic. In [74], a decomposition approach for tandem queueing systems with blocking was proposed. The network calculus approach for statistical QoS provisioning of communication networks was proposed in [75]. Mainly proposed for wired networks, this approach, however, cannot be directly applied to wireless networks with sophisticated physical and link layer design.

Since there are diverse techniques employed in the physical layer of different wireless standards, the notion of bandwidth depends on the underlying wireless technology. A physical channel can be provided by a spreading code in code-division multiple access (CDMA) systems [102], a sub-carrier in orthogonal frequency-division multiplexing (OFDM) systems [103], [104] or simply a frequency band in frequency-division multiple access (FDMA) systems [105]. Each of these physical channels may be divided into equal-size time slots and different time slots may be used for transmissions on different links in a common neighborhood (i.e., in CDMA-TDMA, OFDM-TDMA and FDMA-TDMA systems).

In this chapter, we present queueing models to analyze a tandem of queues where batch traffic arrival process, multi-rate transmission in the physical layer, and automatic repeat request (ARQ)-based error recovery in the link layer are taken into account. Note that, multi-rate transmission in the physical layer using adaptive modulation and coding (AMC) and ARQ-based error recovery in the link layer are widely used techniques for most of the current wireless standards [2], [9]-[21]. We assume per-flow queueing along a routing path for which a separate queue is maintained for each flow at each node. Since the computational complexity of the exact model is very high, we propose a decomposition approach for the tandem queue. Using the decomposition approach, we can calculate the performance measures such as end-to-end loss rate, average delay, and delay distribution with much lower computational complexity. We then show how the decomposition approach can be incorporated into a QoS routing protocol so that the end-to-end QoS requirements are satisfied. We also extend the per-flow queueing-based QoS routing framework to a class-based queueing and QoS routing framework which supports a finite number of service classes with differentiated QoS requirements. The queueing and QoS routing frameworks presented in this chapter provides a unified solution for the problem of QoS provisioning in multi-hop wireless networks, which has not been thoroughly treated in the literature.

## 6.1 System Model

### 6.1.1 Network Model

We consider a wireless multi-hop network with multiple ongoing connections each of which spans several hops. Data traffic arriving at the source node is transmitted hop by hop to the destination node. We assume that each node in the network maintains a separate queue for each traffic flow traversing the link emanating from the node (i.e., per-flow queueing). A multi-hop network model with two ongoing connections is shown in Fig. 6.1 where for convenience, we show only one queue at each node.

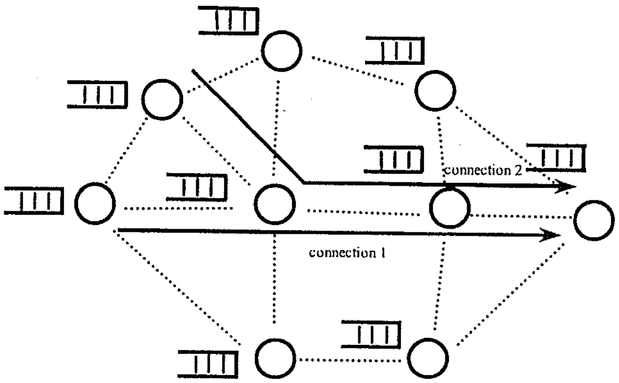


**Figure 6.1.** *A multi-hop wireless network with multiple ongoing connections.*

A particular amount of bandwidth is allocated for each hop along the routing path of the connection so that its end-to-end QoS requirements are satisfied. For a particular connection, the tandem system of queues along its routing path is illustrated in Fig. 6.2. This tandem queue has multiple concatenated queues where traffic coming out of each queue is fed into the next queue in the chain. The sequence of nodes that the traffic flow traverses is decided by a routing algorithm. The physical and link layer model for any hop along the routing path is described in the next subsection. Our objective is to find all end-to-end performance measures for a general tandem system of queues with an arbitrary number of hops. For notational convenience, we will occasionally construct a vector from the corresponding entries in the sequel. For example, vector **d** with elements $d_i$ $(i = 0, 1, \cdots, M)$ will be denoted as

$\mathbf{d} = [d_0, d_1, \cdots, d_M]$.

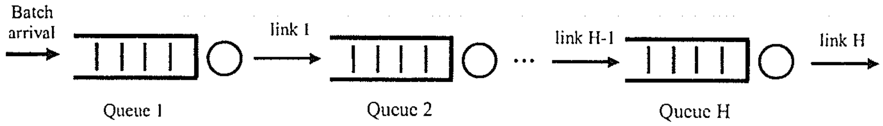


**Figure 6.2.** *A tandem queue.*

### 6.1.2 Physical and Link Layer Model

We model the physical layer in a general way such that the tandem queueing model can be applied to many different physical layer technologies. Assume that there is a finite number of physical orthogonal channels separated in spreading code (for CDMA systems [102]) or in frequency domain (for OFDM [103], [104] or FDMA [105] systems ). Transmission time on each orthogonal channel is divided into fixed-size time slots which are occupied by only one link or shared by different links in a common neighborhood as in [55]. We will refer to the former case as non-time-sharing systems and latter case as time-sharing systems.

For time-sharing systems, time slots are grouped into fixed-length time frames where the time slots in each time frame are periodically allocated for some transmission links in a common neighborhood. Each link may be allocated time slots from different orthogonal channels in each time frame. For non-time-sharing systems, each orthogonal channel is allocated to only one link. Therefore, a non-time-sharing system is a special case of a time-sharing system where one time frame is equal to one time slot. In the queueing model, we observe the system states at the beginning of each time frame without explicitly stating the detailed resource allocation mechanism (i.e., either time-sharing or non-time-sharing systems).

We assume that the packet length is fixed. The physical layer employs AMC technique where there are a finite number of transmission modes each of which corresponds a unique modulation and coding scheme. The transmission rate of each transmission mode is proportional to its spectral efficiency. We assume that if the channel is in channel state $k$, $h_k$ packets can be transmitted in one time slot. We also assume that $h_0 = 0$, (i.e. no packet is transmitted in channel state zero to avoid the high transmission error probability). The calculations of average PER and channel state probability are presented in *Chapter 2*.

We will choose the SNR thresholds for different transmission modes in each allocated channel such that the average PER for all transmission modes in allocated time slots of hop $l$ equal to a particular value denoted by $\beta^{(l)}$. The algorithm to find such SNR thresholds is presented in *Chapter 2*. In each hop, an infinite-persistent ARQ protocol is employed in the link layer where an erroneous packet is retransmitted until it is received correctly at the receiving end of each hop. This is justifiable due to the fact that a large number of transmission attempts is usually recommended in the link layer to shield wireless errors from the higher layers [106], [107]. Depending on the transmission outcome in each time frame, an acknowledgment (ACK) or negative acknowledgment (NACK) is fed back from the receiver to the transmitter of each hop for each transmitted packet. We assume that the ACK/NACK packets are available at the end of the transmission time frame and the feedback channel is error-free. Erroneous packets in one particular time frame will be retransmitted in the next time frame.

The channel state is assumed to be stationary in each time frame but changes independently in consecutive time frames (i.e., block fading channel). We assume that the transmission link in hop $l$ of the tandem system is allocated $\omega_l$ time slots from $\theta_l$ different orthogonal physical channels in each time frame. Note that channel states in different time slots of one time frame on any allocated orthogonal channel is the same. Let $\varphi_h^{(l)}(k)$ be the probability that the allocated channel $h$ of hop $l$ is in state $k$, which can be calculated as in *Chapter 2* by using the channel parameters on the corresponding allocated channel. Assuming that the channel states of different allocated channels are independent, we can calculate the probability that $i$ packets are transmitted during one time frame in hop $l$ as

$$p_i^{(l)} = \sum_{\Sigma_i} \prod_{k,h=1}^{h=\theta_l} \varphi_h^{(l)}(k) \tag{6.1}$$

where $0 \leq i \leq h_K \omega_l$ and $\Sigma_i$ is the combination of all possible channel states on $\theta_l$ allocated channels of hop $l$ such that the total number of packets transmitted in all allocated time slots is equal to $i$. Note that $h_K$ is the maximum number of transmitted packets in one allocated time slot. For the tandem system, we will use the terms "link

$l$" and "hop $l$" interchangeably in the sequel.

**Example:** A particular link $l$ of the tandem system is allocated four time slots on two different orthogonal channels: time slots one and two on channel one, time slots three and four on channel two. The channel states in time slots one and two (also in time slots three and four) in any time frame are the same because they belong to the same channel. If three and four packets can be transmitted in each time slot of channel one and two, respectively, the total number of packets which can be transmitted in all allocated time slots for this link is $3 \times 2 + 4 \times 2 = 14$ packets.

## 6.2 An Exact Tandem Queue Model

In this section, we present an exact model for the tandem queue. We model the tandem queue in discrete time with a time unit equal to a time frame. We observe the system state at the beginning of each time frame. We assume that traffic arrives at the source node buffer according to a batch Bernoulli arrival process where $i$ packets arrive in one time frame with probability $\mathbf{a}_i^{(1)}$ ($i = 0, 1, 2, \cdots, M$ where $M$ can go to $\infty$). We assume that packets arriving in time frame $t-1$ can only be transmitted during time frame $t$ at the earliest. The queues of the tandem system are numbered using an increasing sequence of integers where the source node maintains queue one and queue $i$ has the buffer size of $Q_i$ packets. Packets arriving at each buffer that could not find space will be dropped.

Packets successfully received at the receiving side of each link are buffered for either delivery to the application layer if it is the last link of the connection or otherwise transmission to the next hop. The transmission rate on each link in each time frame depends on the channel states in the allocated time slots. We assume that all wireless links employ AMC with the same number of $K$ modes. The probability that $i$ packets are transmitted on all allocated time slots of hop $l$ is $p_i^{(l)}$, which can be calculated using (6.1).

### 6.2.1 Two Queue Case

We first consider a simple tandem system with two queues. The more general case with $H$ queues ($H > 2$) will be considered in the next subsection. Let $q_i(t)$ be the number of packets in queue $i$ in time frame $t$. The random process $\mathbf{X}(t) = \{q_1(t), q_2(t)\}$, $(0 \leq q_1(t) \leq Q_1, 0 \leq q_2(t) \leq Q_2)$ forms a discrete-time Markov chain (MC). For notational convenience, we omit the time index $t$ in the related variables when it does not create confusion. Let $(x, y)$ be the generic system state (i.e., $q_1 = x$, $q_2 = y$) and $(x_1, y_1) \to (x_2, y_2)$ be the system transition from state $(x_1, y_1)$ to state $(x_2, y_2)$. The transition probabilities $\Pr\{(x_1, y_1) \to (x_2, y_2)\}$ for the underlying MC are derived in **Appendix F.1**.

Note that the number of packets transmitted on each link in any time frame is the minimum of the number of packets in the corresponding queue and the transmission capability in all allocated time slots. The number of packets in queue one can reduce by at most by $N = h_K\omega_1$, where $\omega_1$ is the total number of allocated time slots for link one in one time frame. Because there are at most $M$ packets arriving at queue one (from the source node) and at most $N$ packets enter queue two (due to successful transmissions from queue one) in one time frame, the number of packets can increase at most by $M$ for queue one and by $N$ for queue two.

Hence, if we write the transition probabilities $(x_1, *) \to (x_2, *)$ in a matrix block $\mathbf{A}_{x_1,x_2}$, the probability transition matrix of the MC $\mathbf{X}(t)$ can be written as in (6.2). The order of matrix block $\mathbf{A}_{x_1,x_2}$ is $(Q_2 + 1) \times (Q_2 + 1)$ and its $(y_1, y_2)$-th element is $\mathbf{A}_{x_1,x_2}(y_1, y_2) = \Pr\{(x_1, y_1) \to (x_2, y_2)\}$.

$$\mathbf{P} = \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,M} & & & & \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,M+1} & & & \\ \mathbf{A}_{N,0} & \mathbf{A}_{N,1} & \mathbf{A}_{N,2} & \mathbf{A}_{N,3} & \cdots & \mathbf{A}_{N,N+M} & & \\ & & & \ddots & \ddots & \ddots & \ddots & \\ & & \mathbf{A}_{Q_1-1,Q_1-N-1} & \mathbf{A}_{Q_1-1,Q_1-N} & \mathbf{A}_{Q_1-1,Q_1-N+1} & \mathbf{A}_{Q_1-1,Q_1-N+2} & \cdots & \mathbf{A}_{Q_1-1,Q_1} \\ & & & \mathbf{A}_{Q_1,Q_1-N} & \mathbf{A}_{Q_1,Q_1-N+1} & \mathbf{A}_{Q_1,Q_1-N+2} & \cdots & \mathbf{A}_{Q_1,Q_1} \end{bmatrix}. \tag{6.2}$$

Now, we are ready to derive the steady state probabilities for MC $\mathbf{X}(t)$. Let $\pi$ be the steady state probability vector for $\mathbf{X}(t)$. We have

$$\pi\mathbf{P} = \pi, \quad \pi\mathbf{1} = 1 \tag{6.3}$$

where $\mathbf{1}$ is a column vector of all ones with the same dimension as $\boldsymbol{\pi}$, which is $(Q_1+1)(Q_2+1)$. We can expand $\boldsymbol{\pi}$ as follows:

$$\boldsymbol{\pi} = \left[\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_{Q_1}\right]$$

where $\boldsymbol{\pi}_i$ is a row vector of dimension $Q_2+1$, which can be further expanded as $\boldsymbol{\pi}_i = \left[\pi_{i,0}, \pi_{i,1}, \pi_{i,2}, \cdots, \pi_{i,Q_2}\right]$, where $\pi_{i,j}$ is the probability that the queueing system is in state $(i,j)$. Given the steady state probability vector $\boldsymbol{\pi}$ which is calculated using (6.3), we can derive the following end-to-end QoS measures.

#### 6.2.1.1 End-to-End Loss Rate

Packets can be lost due to buffer overflow at one of the queues in the tandem. The buffer overflow probability for queue $k$ can be calculated as a ratio between the average number of dropped packets due to overflow at queue $k$ (denoted as $\overline{O}_k$) and the average number of packets arriving at queue $k$ in one time frame (denoted as $\overline{A}_k$). Hence, the buffer overflow probability for queue $k$ can be written as $P_l^{(k)} = \frac{\overline{O}_k}{\overline{A}_k}$.

Note that the average number of packets arriving at queue one in one time frame is $\overline{A}_1 = \sum_{i=1}^{M} i\mathbf{a}_i^{(1)}$. To calculate the average number of dropped packets due to overflow at queue one, let us define $\mathbf{z}_i$ as the marginal probability that there are $i$ packets in queue one. We have $\mathbf{z}_i = \boldsymbol{\pi}_i \mathbf{1}_{Q_2+1}$. The average number of dropped packets due to overflow at queue one can be calculated as

$$\overline{O}_1 = \sum_{i=1}^{M} \sum_{j=Q_1-M}^{Q_1} \mathbf{a}_i^{(1)} \mathbf{z}_j \times \max\{0, i+j-Q_1\}$$

where $\max\{0, i+j-Q_1\}$ is the number of dropped packets (if any) given that there are $j$ packets in queue one and $i$ arriving packets. Now, we calculate the buffer overflow probability at queue two. We first determine the arrival probability for packets entering queue two due to successful transmissions from queue one. In fact, the number of packets arriving at queue two are those successfully transmitted over link one. The probability that $i$ packets arrive at queue two can be approximated as

$$\mathbf{a}_i^{(2)} \approx \sum_{k=0}^{Q_1} \sum_{l=0}^{N} \mathbf{z}_k p_l^{(1)} \times \gamma^{(1)}(\min\{k,l\}, i).$$

where $\gamma^{(l)}(n,m)$ is the probability that $m$ packets are correctly received given $n$ packets were transmitted over link $l$ of the tandem system, which is given in **Appendix F.1**.

Average arrival rate to queue two can be calculated as $\overline{A}_2 = \sum_{i=1}^{N} i\mathbf{a}_i^{(2)}$. To calculate the average number of dropped packets due to overflow at queue two, let us define $\mathbf{w}_i$ as the marginal probability that there are $i$ packets in queue two, which can be calculated as $\mathbf{w}_i = \sum_{j=0}^{Q_1} \pi_{j,i}$. Similar to queue one, the average number of dropped packets due to overflow at queue two can be calculated as

$$\overline{O}_2 = \sum_{i=1}^{N} \sum_{j=Q_2-N}^{Q_2} \mathbf{a}_i^{(2)} \mathbf{w}_j \times \max\{0, i+j-Q_2\}.$$

Finally, the end-to-end loss rate can be approximated as

$$P_l \approx 1 - (1 - P_l^{(1)})(1 - P_l^{(2)}) \tag{6.4}$$

where the loss due to overflow at both buffers are taken into account. This approximation is tight when the loss rates at different queues are weakly dependent as being confirmed in section 6.5.2.

#### 6.2.1.2 End-to-End Average Delay

The end-to-end delay is the sum of delays that any packet experiences in all queues and links along its routing path. Assuming that the propagation delay over the wireless channel is negligible (i.e., only queueing delay is considered in the calculation), using Little's law, the end-to-end average delay can be written as

$$D = \frac{\sum_{i=1}^{Q_1} i\mathbf{z}_i}{\overline{A}_1(1 - P_l^{(1)})} + \frac{\sum_{i=1}^{Q_2} i\mathbf{w}_i}{\overline{A}_2(1 - P_l^{(2)})} \tag{6.5}$$

where the numerator of each term is the average length of each queue and the denominator is the average arrival rate considering packet loss due to overflow.

### 6.2.2 General Case ($H > 2$)

We consider a general tandem system with more than two queues as shown in Fig. 6.2. Now, the tandem system has $H$ queues ($H > 2$) which are concatenated to each other as a chain. The buffer size of queue $i$ is assumed to be $Q_i$ packets. Similar to the previous subsection, let $q_i(t)$ be the number of packets in queue $i$ in time frame $t$ $(i = 1, 2, \cdots, H)$. The random process $\mathbf{Y}(t) = \{q_1(t), q_2(t), \cdots, q_H(t)\}$, $(0 \leq q_1(t) \leq Q_1$, $0 \leq q_2(t) \leq Q_2$, $\cdots$, $0 \leq q_H(t) \leq Q_H)$ forms a discrete-time Markov chain (MC).

An approach similar to that in section 6.2.1 can be pursued to obtain the transition probabilities for this MC. The number of state transitions for this MC, however, grows exponentially with the number of queues in the tandem system. In fact, the order of the transition probability matrix $\mathbf{P}$ is $\prod_{i=1}^{H}(Q_i + 1) \times \prod_{i=1}^{H}(Q_i + 1)$. Therefore, the computational complexity is very high for the large number of queues and large buffer sizes. Theoretically we can follow the procedure similar to that in Subsection 6.2.1 to derive the transition probabilities, obtain the steady state probability vector, and subsequently, the end-to-end performance measures.

## 6.3 Solution of Tandem Queueing Model: Decomposition Approach

We present a novel decomposition approach to solve the general tandem queue where the computational complexity grows only linearly with the number of queues in the system. For ease of reference, buffers (queues) along the routing path are numbered in an increasing sequence of integers where the buffer at the source node is denoted as buffer (queue) one.

### 6.3.1 Markov Chain and Steady State Probability

We consider the tandem system with $H$ queues as in Fig. 6.2. For notational convenience, we assume that $i$ packets arrive at queue $k$ with probability $\mathbf{a}_i^{(k)}$. Note that the maximum batch size (maximum number of packets arriving in one time frame) captured in $\mathbf{a}_i^{(k+1)}$ for $k \geq 1$ is $N_k = h_K \omega_k$ while the maximum batch size captured in $\mathbf{a}_i^{(1)}$ for the first queue is $M$. The buffer sizes for all queues and queueing rules are as in Section 6.2.

We observe that the behavior of queue $i+1$ does not impact queue $i$ in the chain.

This is because the outcomes (i.e., successfully transmitted packets) from queue $i$ are fed into queue $i+1$. Thus, instead of forming the Markov chain which captures the queue length dynamics of all queues we could find the queue length dynamics for one queue at a time where its input is the output of the previous queue in the chain (except for queue one).

Specifically, we pursue the following steps. First, form the MC for queue one and calculate the corresponding steady state probability vector. Based on the steady state probabilities, we calculate the packet arrival probabilities to the next queue. These arrival probabilities are used to derive the arrival probabilities for the next queue in the chain. This procedure is repeated until we obtain the solutions for the last queue of the tandem system.

Obviously, by using this decomposition approach the joint steady state probability vector could not be found as in Section 6.2. However, the steady state probability vector for each queue in the chain is what we need to calculate the desired queueing performance measures. Essentially, the presented procedure requires us to solve $H$ separate queues each of which accepts batch arrival traffic and serves packets also in batches. Let us consider a particular queue $k$ of the chain and form the MC $X_k(t) = \{q_k(t)\}$, $(0 \leq q_k(t) \leq Q_k)$ where $q_k(t)$ denotes the number of packets in queue $k$ with arrival process described by $\mathbf{a}_i^{(k)}$. The transition probabilities for this MC are derived in **Appendix F.2**.

Given the transition probabilities, we can easily calculate the steady state probability vector of this MC which is denoted as $\boldsymbol{\pi}^{(k)} = \left[\pi_0^{(k)}, \pi_1^{(k)}, \cdots, \pi_{Q_k}^{(k)}\right]$ where $\pi_i^{(k)}$ denotes the probability that there are $i$ packets in queue $k$.

### 6.3.2 End-to-End Loss Rate and Average Delay

As in Section 6.2, the buffer overflow probability at queue $k$ can be calculated as

$$P_l^{(k)} = \frac{\overline{O}_k}{\overline{A}_k}. \tag{6.6}$$

The average arrival rate to queue $k$ can be written as $\overline{A}_k = \sum_{i=1}^{B^{(k)}} i\mathbf{a}_i^{(k)}$, where $B^{(k)}$ is the maximum batch size of the arrival process to queue $k$. The probability that $i$ packets are successfully transmitted from queue $k$ and arrive at queue $k+1$

can be approximated as

$$\mathbf{a}_i^{(k+1)} \approx \sum_{j=0}^{Q_k} \sum_{l=0}^{N_k} \pi_j^{(k)} p_l^{(k)} \times \gamma^{(k)}(\min\{j, l\}, i). \tag{6.7}$$

These arrival probabilities are used to derive the queueing solution for queue $k+1$ as mentioned before. And the average number of dropped packets due to overflow at queue $k$ can be calculated as

$$\overline{O}_k = \sum_{i=1}^{B^{(k)}} \sum_{j=Q_k-B^{(k)}}^{Q_k} \mathbf{a}_i^{(k)} \pi_j^{(k)} \times \max\{0, i+j-Q_k\}.$$

The end-to-end loss rate can be approximated as

$$P_l \approx 1 - \prod_{k=1}^{H} (1 - P_l^{(k)}) \tag{6.8}$$

and the end-to-end average delay can be written as

$$D = \sum_{k=1}^{H} D_k \tag{6.9}$$

where $D_k$ is the average queueing delay at queue $k$ which is given by

$$D_k = \frac{\sum_{i=1}^{Q_k} i\pi_i^{(k)}}{\overline{A}_k(1 - P_l^{(k)})}. \tag{6.10}$$

### 6.3.3 End-to-End Delay Distribution

The proposed decomposition approach for tandem queues enables us to derive the end-to-end delay distribution with reasonable accuracy, which is necessary for statistical delay provisioning in wireless multi-hop networks. Let $\Omega_{i,l}^{(k)}$ denote the probability that $i$ packets are successfully transmitted from queue $k$ in $l$ time frames. Because the tagged packet can see at most $Q_k - 1$ head-of-line (HOL) packets, the probability that the tagged packet sees $i$ HOL packets is $\chi_i^{(k)} = \pi_i^{(k)}/(1 - \pi_{Q_k}^{(k)})$. The probability that the tagged packet experiences a delay of $l$ time frames in queue $k$ of the tandem system can be calculated as

$$P_d^{(k)}(l) = \sum_{i=0}^{\Delta_l} \chi_i^{(k)} \Omega_{i+1,l}^{(k)} \tag{6.11}$$

where $\Delta_l = \min\{lN_k - 1, Q_k - 1\}$ because at most $N_k$ packets can be transmitted in one time frame and the tagged packet sees at most $Q_k - 1$ HOL packets. Let us put the delay distribution at each queue $k$ of the tandem system into a vector $\mathbf{P}_d^{(k)}$, and put the end-to-end delay distribution vector into vector $\mathbf{P}_d$. Then, we have

$$\mathbf{P}_d \approx \otimes_{k=1}^{H} \mathbf{P}_d^{(k)} \tag{6.12}$$

which is obtained by performing convolutions of $H$ vectors $\mathbf{P}_d^{(k)}$ $(k = 1, \cdots, H)$. Note that the first element of vector $\mathbf{P}_d$ represents the probability that the end-to-end delay is $H$ time frames, which is the minimum end-to-end delay. The remaining task is to derive $\Omega_{i,l}^{(k)}$, which can be done by using the following recursive relations

$$\Omega_{i,l}^{(k)} = \sum_{j=0}^{N_k} \Lambda_{i,j}^{(k)} \Omega_{i-j,l-1}^{(k)}, \quad \Omega_{0,0}^{(k)} = 1 \tag{6.13}$$

where $\Lambda_{i,j}^{(k)}$ is the probability that $j$ packets are successfully transmitted from queue $k$ given that there were $i$ packets in this queue before transmissions. Equation (6.13) can be interpreted as follows. If there are $i$ packets in queue $k$ which must be transmitted in $l$ time frames and $j$ packets are transmitted in the first time frame, there are remaining $i - j$ packets to be transmitted in $l - 1$ time frames. Now, $\Lambda_{i,j}^{(k)}$ can be calculated as

$$\Lambda_{i,j}^{(k)} = \sum_{v} p_v^{(k)} \times \gamma^{(k)}(\min\{i, v\}, j) \tag{6.14}$$

where the sum includes only $v$ such that $\min\{i, v\} \geq j$.

## 6.4 Application of Tandem Queueing Model for QoS Routing

We show how to incorporate the proposed tandem queueing model into a QoS routing algorithm. The tandem queueing models proposed in Sections 6.2-6.3 were solved for a particular connection given that its routing path is known. Now, we want to tackle the inverse problem where the tandem queueing model is used to discover a route for a connection from a source node to its desired destination node such that the QoS requirements for the connection are satisfied. One possible approach for QoS routing is to find all possible routes from the source to the destination. The source node upon gathering all possible routes will check the routes one by one to find the best feasible route using the presented tandem queueing model. This approach, however, results in a very large amount of signaling/communication overhead in the route discovery phase and a large computational burden for the source node.

We observe that the decomposition approach for the tandem queue has a nice distributed nature where the link QoS metrics (i.e., link average delay $D_k$ and loss rate $P_l^{(k)}$ for link $k$) can be calculated if the routing path up to a particular hop is known. The decomposition approach can be used as an efficient tool to search for feasible routes such that the QoS requirements of the connection are satisfied. Also, the route discovery can be done in a hop by hop basis and only potentially feasible routes are explored further. This reduces the route searching overhead significantly, and therefore, avoids huge computation effort at the source node.

The unique feature of our queueing and routing framework is that the three most important QoS metrics, namely, end-to-end bandwidth, delay, and loss rate can be integrated into the QoS routing algorithm compared with only delay and/or bandwidth as was done by most QoS routing algorithms available in the literature [54], [55]-[57]. In addition, most of existing works assumed that link delay can be measured and/or estimated in a timely manner [54], [57]. The dynamics of the traffic arrival process, wireless channel fading and the complex physical and link layer design of wireless systems, however, would render this delay measurement/estimation a time-consuming task. Our queueing model provides an accurate and efficient tool for link metric calculation.

In the remainder of this section, we will describe an on-demand unicast QoS routing algorithm using the tandem queue model based on the decomposition approach presented in Section 6.3. Like other QoS routing algorithms in the literature, two main components of our QoS algorithm are route discovery with bandwidth reservation and route maintenance. The QoS constraints for an incoming connection are end-to-end bandwidth, average delay, and loss rate. An additional statistical delay requirement can be also imposed by an incoming connection.

### 6.4.1 Route Discovery and Resource Reservation

To establish a connection, the source node broadcasts the route request packet (RRQ) into the network to search for good routes to the destination node which satisfy the QoS requirements of the connection. The incoming connection submits the traffic profile (i.e., packet arrival probabilities to the source node buffer $a_i^{(1)}$) as well as its QoS requirements to the source node. The RRQ packet contains the addresses of the source node and the destination node, the request ID, and the end-to-end QoS requirements. Let the target QoS requirements for an incoming connection $c$ be end-to-end bandwidth $B(c)$, end-to-end average delay $D(c)$, end-to-end loss rate $P_l(c)$, and an optional end-to-end statistical delay requirement of the form $\Pr\{\text{end-to-end delay} > D_t(c)\} \leq P_t(c)$.

The required amount of bandwidth $B(c)$ needs to be reserved for each link along the routing path. The bandwidth here refers to the time slots for transmissions using different channels. On any orthogonal channel, a particular time slot can be allocated to only one link in a common neighborhood. Resource allocation in multi-hop wireless networks is an active research topic by itself which is not the focus of this chapter. For ease of presentation, we assume that a static resource allocation scheme is adopted where each link in the network is preallocated a certain number of time slots for transmission using some orthogonal channels from the set of available channels (the allocation is assumed to be repeated in each frame time if time-sharing is implemented). An incoming connection may take some of these preallocated time slots of the link (i.e., the bandwidth taken by the connection is equal to its bandwidth requirement) if its routing path traverses the corresponding link. This static predetermined allocation should be done such that two different links using the same channel in a common neighborhood are not allocated the same time slot. Each node is assumed to have enough radios to communicate with its neighbors on the allocated channels as in CDMA or multi-channel networks [55], [105].

When a node receives the RRQ packet, it checks the available bandwidth on the outgoing links and only outgoing links having enough bandwidth to accommodate the new connection are considered further. The outgoing links with enough bandwidth will be called the BW-feasible links. A more flexible resource allocation scheme would allow a link to borrow bandwidth from its neighboring links if this mechanism can potentially enhance the system performance. This scheme, however, requires local negotiation and reservation which is more complicated. We assume that the transmitter of each link knows the channel parameters of the link (i.e., average SNR at the receiving end, Nakagami parameter $m$) by using some estimation technique.

Now, we describe how each node calculates the link QoS metrics together with the resource reservation mechanism mentioned above. Initially, the source node of the incoming connection calculates the average link delay and loss rate for each of the outgoing links which has enough bandwidth to accommodate the incoming connection. This is done by using the queueing model presented in Section 6.3. based on the submitted traffic profile and link channel parameters at the transmitting node. For outgoing links which satisfy the connection QoS requirements, the source node calculates the arrival probabilities to the next node along the corresponding outgoing link as in (6.7), records these arrival probabilities, average link delay, and loss rate into the RRQ packet header and forwards the RRQ packet to the receiving node of the corresponding link. The receiving nodes of these feasible links join a set of nodes called broadcast group (BG). Each node upon receiving the RRQ packet calculates the link delay, loss rate using the arrival probabilities retrieved from the RRQ packet header, and channel parameters for its BW-feasible outgoing links. The node then accumulates the QoS metrics and checks the feasibility of the QoS requirements for the connection. For each feasible outgoing link, the node records the arrival probabilities for the next node, route metrics into the RRQ packet header, and forwards the RRQ packet to the corresponding node.

In the above procedure, only routing paths along the links that have enough bandwidth required by the incoming connection with feasible path metrics are explored further. Now, we describe how to calculate the path metric and choose the best routing path. Let the average delay and loss rate over link $(i, j)$ be $D(i, j)$ and $P_l(i, j)$, respectively. The end-to-end average delay and loss rate for routing path $R = i \rightarrow j \rightarrow \ldots k \rightarrow l$ can be adapted from (6.8), (6.9) as

$$\begin{aligned} \text{delay}(R) &= D(i,j) + \cdots + D(k,l) && (6.15)\\ \text{loss}(R) &= 1-(1-P_l(i,j))\ldots(1-P_l(k,l)) \approx P_l(i,j)+\cdots+P_l(k,l) && (6.16) \end{aligned}$$

where the approximation is tight for small loss rate. Also, $P_l(i, j)$ and $D(i, j)$ can be calculated by using (6.6), (6.10), respectively, for the corresponding link (hop). Since there are multiple QoS requirements, the definition of the best routing path is not unique. To resolve this issue, we define the weighted average QoS metric as follows:

$$\text{metric}(R) = \alpha \frac{\text{delay}(R)}{D(c)} + (1-\alpha)\frac{\text{loss}(R)}{P_l(c)} \qquad (6.17)$$

where $\alpha$ determines the importance of the delay requirement in comparison with the loss rate requirement.

A node may receive the RRQ packet with same request ID more than once. If the QoS metric (i.e., metric($R$)) retrieved from the RRQ packet header for the current reception is smaller than that due to the previous reception, it will rebroadcast the RRQ packet with the new QoS metrics. Finally, if a node finds out that it is the destination of the connection, it sends the route reply packet (RRP) back to the source node. If the destination node finds a route to the source node in its route cache, it can send the RRP packet along this route. If this reverse route is not available but each link along the newly-discovered route works well in both directions, the RRP packet follows the reverse route to reach the source node. If the reverse route does not work well, the RRP packet can be piggybacked as proposed in [62]. In addition, we may use one of the following two ways to record the feasible routes. The first way is to record the route into the RRQ and RRP packet headers. The second way is to record the route at intermediate nodes in a hop-by-hop basis.

If the optional end-to-end statistical delay requirement is requested by the incoming connection, the channel and traffic parameters of the links along the discovered routes should be recorded into the RRP packet header and fed back to the source node. The source node upon receiving these parameters will check whether the end-to-end statistical delay requirement is satisfied or not. Then, the best feasible route can be used for data transmission. Before transmitting data on the newly-discovered route, each link along this route updates its available bandwidth for possible future connection. The available bandwidth on these links will also be updated when the connection is released. The other important thing is to limit the overhead caused by the route discovery process. The first way to reduce the overhead is to use ticket-based route discovery as proposed in [57]. Another way is to record the number of hops the RRQ packet has traversed and limit the number of hops the RRQ can be broadcast. The third way is to use timeout to invalid the RRQ packet. The route discovery algorithm is summarized in Fig. 6.3.

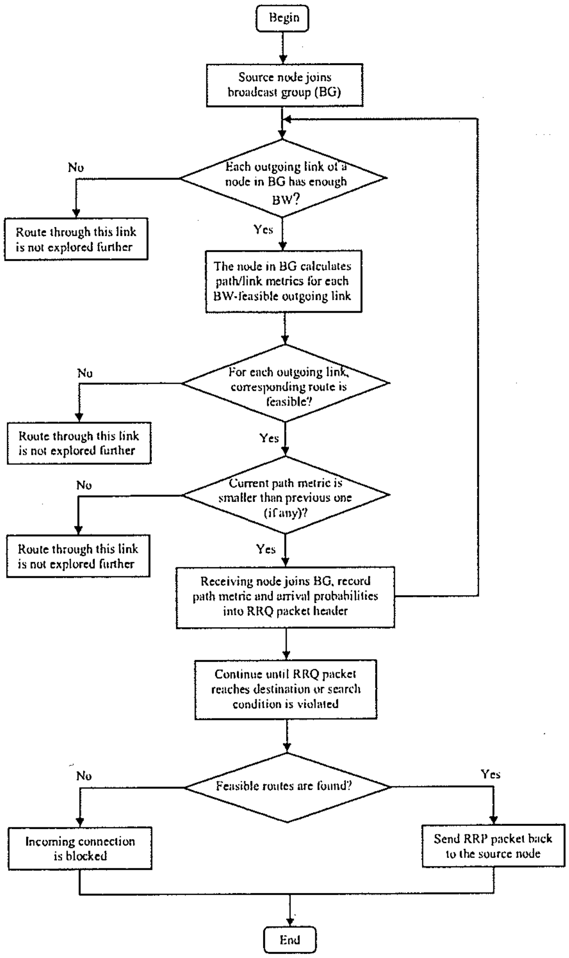

Begin
Source node joins broadcast group (BG)
Each outgoing link of a node in BG has enough BW?
No
Route through this link is not explored further
Yes
The node in BG calculates path/link metrics for each BW-feasible outgoing link
For each outgoing link, corresponding route is feasible?
No
Route through this link is not explored further
Yes
Current path metric is smaller than previous one (if any)?
No
Route through this link is not explored further
Yes
Receiving node joins BG, record path metric and arrival probabilities into RRQ packet header
Continue until RRQ packet reaches destination or search condition is violated
Feasible routes are found?
No
Yes
Incoming connection is blocked
Send RRP packet back to the source node
End

**Figure 6.3.** *Route discovery algorithm for QoS routing.*

## 6.4.2 Route Maintenance

Another important task for any routing algorithm is route maintenance to make sure that the route works well during the lifetime of the connection. Some links along the route may be broken due to factors such as node mobility, degradation of wireless links, etc. Because we consider wireless systems which employs the link level ARQ error recovery protocol, a broken link can be discovered if the transmitting node of a link does not receive ACK/NACK packets within a predetermined timeout period. The node which detects a link break sends a route error packet to the source node by using the same technique used for sending the RRP packet. The receiving node of the broken link can also detect the link break if it does not receive any data packet within a predetermined timeout period. This receiving node upon detecting the link break will also send the route error packet to release the route in the forward direction.

When the source node receives the route error packet, it may initiate a route discovery to find a new feasible route to the destination. Since the connection setup time may be very long, we may maintain multiple feasible routes. This can be done if the destination node sends the RRP packet containing several feasible routes to the source node. The route with the smallest QoS metric will be used for data transmission. If this best route is broken, the source node may try to use the next best route it has received through the RRP packet. In addition, the QoS of the chosen route may degrade during the lifetime of the connection. Hence, the source node may periodically send the route maintenance packet along the route to check the QoS feasibility of the route. If the QoS feasibility condition of the current route is violated, the next best route may be used or a route discovery may be initiated to find an alternative route to the destination.

## 6.5 Validation of Decomposition Approach and Typical Numerical Results

### 6.5.1 Parameter Setting

We consider wireless networks employing adaptive M-ary quadrature amplitude modulation (M-QAM) without coding using five transmission modes for all transmission links. We assume that $h_k = k$ ($k$ packets are transmitted in one time slot in channel state $k$), Nakagami parameter $m = 1$ (i.e., Rayleigh fading channel), time frame interval equal to 2 ms. The SNR switching thresholds for the transmission modes are chosen such that the average PER satisfies $\overline{\mathrm{PER}}_k = 0.1$ for all transmission modes in all hops (i.e., $\beta^{(l)} = 0.1$ for $l = 1, 2, \cdots, H$). The fitting parameters (i.e., $a_k$, $g_k$ for mode $k$) are available from Table I in [20]. The arrival probability vector to the source node queue is chosen to be $\mathbf{a}^{(1)} = [1 - 25A/48, A/4, A/8, A/12, A/16]$ where the average arrival rate is $A$. For all the results presented here, the buffer size of each queue in the tandem system is equal to 20 packets.

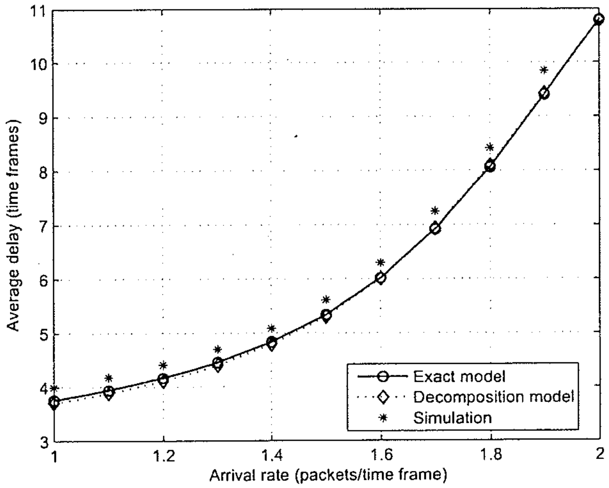


**Figure 6.4.** *End-to-end average delay versus packet arrival rate for a tandem system of two queues (for $H = 2$, Nakagami parameter $m = 1$, average SNR = 15 dB for both hops).*

The numerical results for QoS routing are obtained for networks with ten nodes which are randomly generated in an area of $800m \times 800m$. Node mobility is not considered in deriving the results. We consider a non-time-sharing system with static resource allocation where separate sets of orthogonal channels are allocated for different links in the network. Each node uses a fixed transmit power level on each allocated channel and the average SNR at the receiving node of link $(i, j)$ is modeled as $\text{SNR}(i, j) = K_0.d(i, j)^{-3}$ where $K_0 = 10^9$ captures transmission power, antenna gain and other factors; $d(i, j)$ is the distance from node $i$ to node $j$; the path-loss exponent is assumed to be three. Note that these assumed values are for presenting the illustrative results only while the queueing and QoS routing framework can be applied to many other network settings.

### 6.5.2 Numerical Results and Validation of Queueing Models

We validate the decomposition approach for the tandem queue and present some typical numerical results. Each link of the tandem system is allocated one time slot in each time frame. The average SNR for all links of the tandem system of queues is chosen to be 15 dB. Typical variations in end-to-end average delay and end-to-end loss rate with packet arrival rate are shown in Figs. 6.4-6.5, respectively, for a tandem system of two queues. In these two figures, we show results obtained from the exact queueing model (presented in Section 6.2), the decomposition approach (presented in Section 6.3), and the simulations. As is evident, the decomposition approach provides accurate measures for average delay and loss rate. The analytical results also follow the simulation very closely, which confirms the correctness of the proposed queueing models.

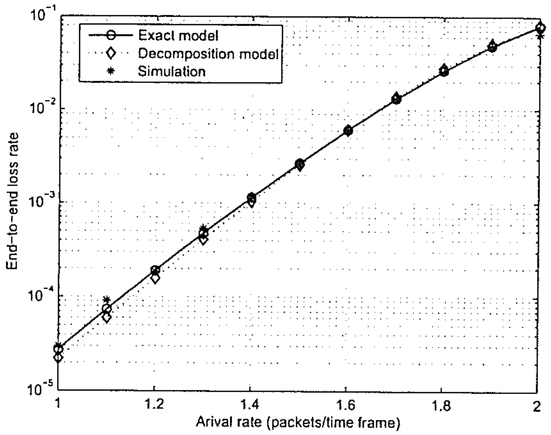


**Figure 6.5.** *End-to-end loss rate versus packet arrival rate for a tandem of two queues (for $H = 2$, Nakagami parameter $m = 1$, average SNR = 15 dB for both hops).*

We illustrate the complementary cumulative delay distributions (obtained in Section 6.3) in Fig. 6.6 for tandem systems with different number of queues ($H = 2, 4, 6$). The results obtained from simulations are also presented. Note that, the complementary cumulative delay distribution is represented by the probabilities $\Pr(\text{delay} > d) = 1 - \sum_{k=1}^{d} P_d(k)$, which can be calculated by using $\mathbf{P}_d$ in (6.12).

We observe that the analytical model slightly over-estimates the end-to-end delay in the statistical sense (i.e., $\Pr(\text{delay} > D)$ obtained from the analytical model is greater than that due to simulation for a given value of $D$). In fact, end-to-end delay of a target packet is the time it spends in all queues of the tandem system. To

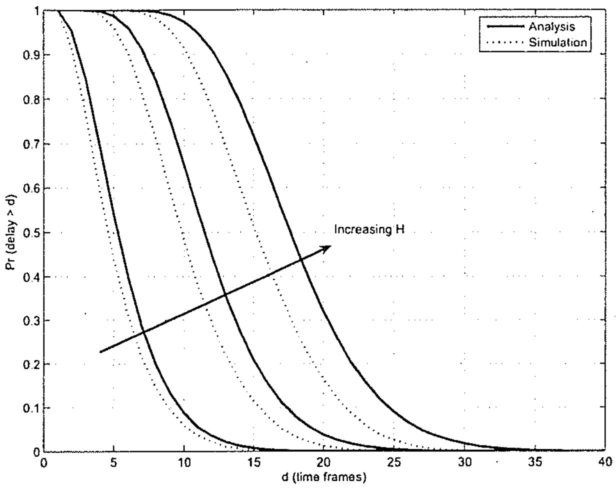


**Figure 6.6.** *End-to-end complementary cumulative delay distribution for tandem systems with different number of queues (for Nakagami parameter $m = 1$, average SNR = 15 dB for all hops, number of queues $H = 2, 4, 6$, packet arrival rate = 1.5 packets/time frame).*

calculate this delay, we can turn off the arrival traffic to the tandem system after the target packet enters the tandem system. This is because arriving packets following the target packet do not impact the delay experienced by the target packet.

Due to turning the arrival traffic off and the batch transmission effect (because of multi-rate transmission on the wireless channel), in a statistical sense, the target packet would see a smaller number of HOL packets in queue two onward compared to the marginal distribution derived in (10). Therefore, by calculating the delay distribution at each queue using the queue length distribution in (10), we over-estimate of traffic arrival probabilities to the queues in the chain except for queue one. A more accurate model can be developed by tracking the number of HOL packets in all queues until the target packet leaves the last queue of the chain. This procedure, however, has a very high computational complexity. The approximated method presented in Section 6.3.3 results in reasonably accurate results with low complexity.

Typical variations in end-to-end average delay and loss rate with packet arrival rate are presented in Figs. 6.7-6.8 for tandem systems consisting of different number of queues. As expected, the end-to-end delay increases almost linearly with the number

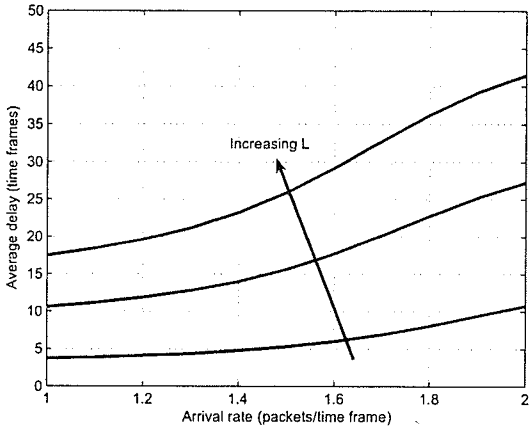


**Figure 6.7.** *End-to-end average delay versus packet arrival rate for tandem systems consisting of different number of queues (for $H = 2, 6, 10$, Nakagami parameter $m = 1$, average SNR = 15 dB).*

of queues in the tandem. In fact, the packet arrival rate to each queue in the chain is roughly the same because the loss rate at each of the queues is quite small. Thus, the average queueing delay at each queue is roughly the same given the same service rate (i.e., all transmission links are assumed to be the same). The variations of end-to-end loss rate with the number of hops can be interpreted in a similar manner considering the approximation obtained in (6.16) for small loss rates.

### 6.5.3 Numerical Results for Proposed QoS Routing Algorithm

We present some illustrative numerical results for the proposed QoS routing algorithm which is based on the decomposition approach for the tandem queue. We consider a network with ten nodes randomly located in an area of $800m \times 800m$ as shown in Figs. 6.9-6.10. A link exists between any two nodes if the distance between them is less than $400m$. For all the results presented in this subsection, each connection requires one channel for each link along its routing path.

To illustrate how the QoS algorithm works, we show the routing paths found by the the presented QoS routing algorithm. Each connection transmits data to its

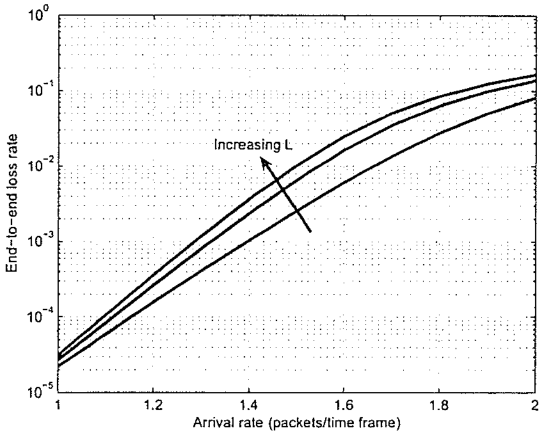


**Figure 6.8.** *End-to-end loss rate versus packet arrival rate for tandem systems with different number of queues (for $H$ = 2, 6, 10, Nakagami parameter $m = 1$, average SNR = 15 dB).*

destination where the packet arrival rate to the source node is equal to 2 packets/time frame. Each connection requires that the average end-to-end delay and the loss rate are less than $D(c) = 10$ time frames, and $P_l(c) = 0.05$, respectively.

In Figs. 6.9-6.10, we show the discovered routes for these two connections for two cases where each link in the network is statically allocated one and two channels, respectively. The source node for these two connections and the metrics for the discovered routes (denoted as $m_i$ for connection $i$) are shown in these two figures. In Fig. 6.9, because each link has only one channel, the routing paths for the two connections are non-overlapping. In Fig. 6.10, each link in the network has two channels and the minimum metric paths for these connections are partially overlapping. The path metric of the discovered route for connection two shown in Fig. 6.9 is larger than that shown in Fig. 6.10. The proposed algorithm, therefore, succeeds in finding the minimum metric path which satisfies the bandwidth requirement of the connection. Also, the optimum path may not be the minimum hop path as can be observed for connection two in both figures.

We now investigate the performance of the presented QoS algorithm for network topologies shown in Figs. 6.9-6.10. Connection requests arrive at each node in each time frame with connection arrival probability $\lambda_c$. Each incoming connection submits

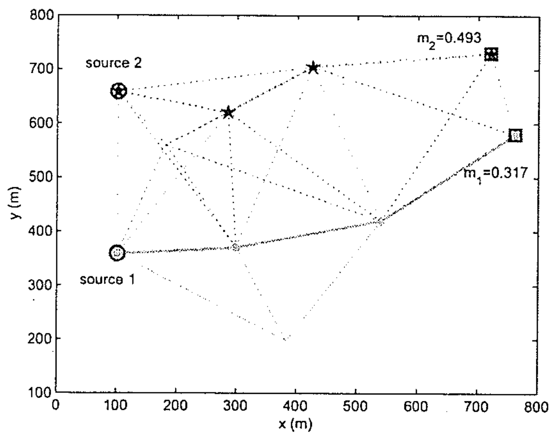


**Figure 6.9.** *Route discovery for two connections where each link was allocated one channel (for $\alpha$ = 0.5, each connection has packet arrival rate = 2 packets/time frame, QoS requirements are $B(c)$ = 1 channel, $D(c)$ = 10 time frames, $P_l(c)$ = 0.05).*

the traffic profile to the source node which initiates the route discovery process to find a routing path to its desired destination. The destination node for each incoming connection is chosen randomly among the remaining nodes. If the QoS routing algorithm succeeds in finding a feasible routing path, the connection remains in the network for an interval which is exponentially distributed with mean value equal to $\mu_c = 500$ time frames.

We show typical variations in connection blocking probability with network load where each link in the network is statically allocated different number of channels. The network load is calculated as $\rho =$ number of nodes $\times \mu_c \times \lambda_c$. We vary network load by changing the connection arrival probability $\lambda_c$. When each link in the network is allocated more channels, the network capacity increases; therefore, the connection blocking probability decreases. Also, the connection blocking probability increases with network load. Fig. 6.11 shows that the connection blocking probability decreases significantly when the number of channels allocated to each link increases from one to two. The improvement decreases when more channels are allocated to each link in the network.

Fig. 6.12 shows variations in connection blocking probability with packet arrival rate for different sets of QoS requirements. As expected, the more stringent the QoS

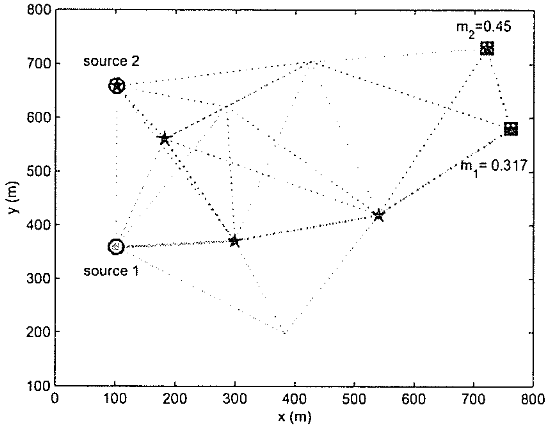


**Figure 6.10.** *Route discovery for two connections where each link was allocated two channels (for $\alpha$ = 0.5, each connection has packet arrival rate = 2 packets/time frame, QoS requirements are $B(c)$ = 1 channel, $D(c)$ = 10 time frames, $P_l(c)$ = 0.05).*

requirements are the less probable it is that the routing algorithm can find a feasible route to the destination. However, Fig. 6.12 shows that the performance degradation in terms of connection blocking probability may be moderate even when more stringent QoS requirements are imposed by the arriving connections. This implies that the proposed queueing and QoS routing framework performs load-balancing well by finding the low-load routes (if any).

## 6.6 Extension for Multi-hop Wireless Networks with Class-Based Queueing

In the previous sections, we have presented the tandem queueing models and its application for QoS routing assuming per-flow queueing at each of the nodes along a routing path. However, if the bandwidths of wireless links are large, a large number of flows may traverse each link. Therefore, per-flow queueing may not be scalable. In contrast, a class-based queueing would provide a more scalable solution where each node maintains a finite number of queues corresponding to a finite number of service classes with differentiated QoS requirements.

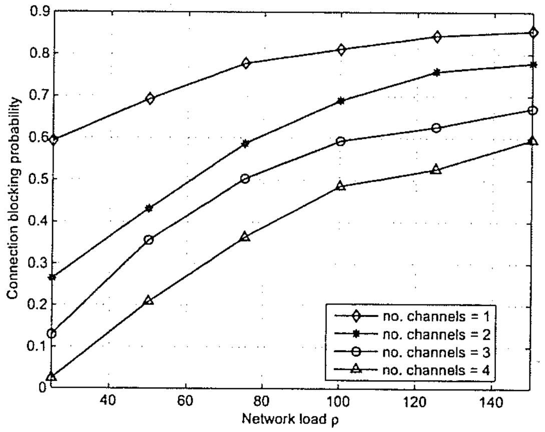


**Figure 6.11.** *Connection blocking probability versus network load for different number of channels (for $\alpha$ = 0.5, average connection holding time = 500 time frames, each connection has packet arrival rate = 2 packets/time frame, QoS requirements are $B(c)$ = 1 channel, $D(c)$ = 10 time frames, $P_l(c)$ = 0.05).*

In this section we extend the per-flow queueing-based QoS routing framework to a class-based QoS routing framework. With class-based queueing, the transmitting node of each link maintains a finite number of queues for each link which corresponds to different service classes. Traffic from connections of the same service class traversing a particular link is buffered in the same queue. The bandwidth allocated to each queue depends on the bandwidth requirement of each connection and the number of connections being served by the queue.

### 6.6.1 Tandem Queueing Model

We show how to extend the per-flow queueing model to this class-based queueing implementation. Specifically, we consider the tandem system of queues along a routing path for a connection of a particular service class as shown in Fig. 6.13. Data traffic entering each queue may come from different connections. As shown in this figure,

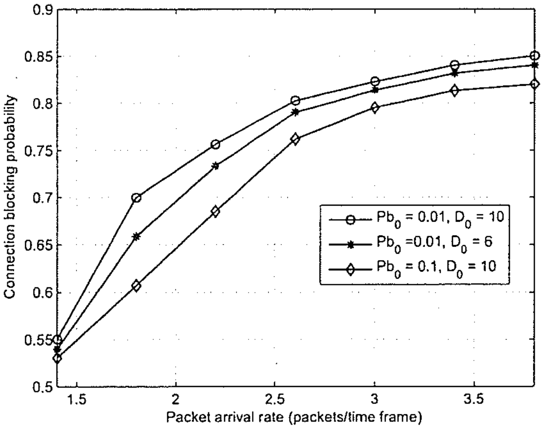


**Figure 6.12.** *Connection blocking probability versus packet arrival rate for different QoS requirements (for $\alpha$ = 0.5, number of channels per link = 2, connection arrival probability = 0.02, average connection holding time = 500 time frames).*

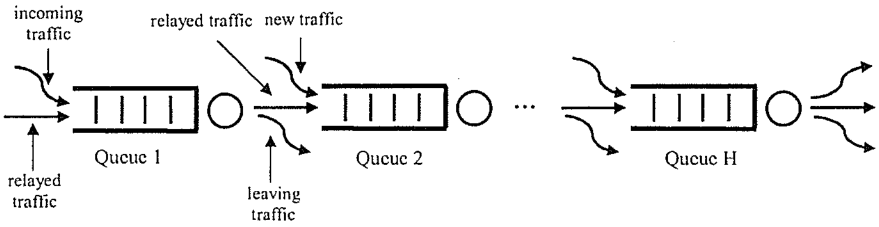


**Figure 6.13.** *A tandem queue for one connection of a particular service class.*

traffic from connections other than the considered connection may come and leave the tandem system at any queue.

Note that traffic of all connections entering a particular queue of the tandem system has the same queueing performance. Now, using the decomposition approach similar to that presented in Section 6.3, we need to solve $H$ single queues where the arrival traffic to each queue is from the previous queue of the tandem system and from other connections traversing the corresponding link (except for queue one). Given the allocated bandwidth for each link along the tandem system, we can determine the service rate probabilities from (6.1). Thus, to calculate the queueing performance measures (i.e., overflow probability, delay) for each queue of the tandem, we need to determine the arrival probabilities for the aggregate traffic to the considered queue.

For each queue of the tandem system (except queue one), we consider new, relayed, and leaving traffic as shown in Fig. 6.13. Note that these traffic sources are aggregated from different connections. Since traffic flows from different connections may be transmitted over several links of the tandem system, we need to keep track of the connections which constitute the traffic on each link. Let the sets of connections whose traffic constitutes the relayed and leaving traffic flows from queue $k$ of the tandem be $\Phi_{\text{rel}}^{(k)}$, $\Phi_{\text{lev}}^{(k)}$, respectively, and the set of connections whose traffic constitutes the new traffic flow to queue $k$ be $\Phi_{\text{new}}^{(k)}$.

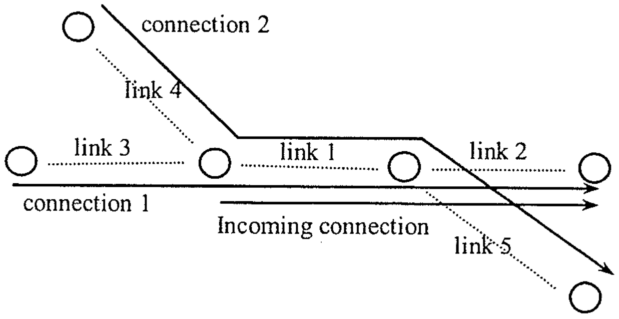


**Figure 6.14.** *An example for QoS routing with class-based queueing implementation*

Now, denote the arrival probability vector to queue $k$ due to connection $c$ as $\mathbf{a}^{(k,c)}$ and its average arrival rate as $\overline{A}^{(k,c)}$. Also, let $\mathbf{a}_{\text{lev}}^{(k)}$ denote the aggregate arrival probability vector of leaving traffic from queue $k$ and $\mathbf{a}_{\text{new}}^{(k)}$, $\mathbf{a}_{\text{rel}}^{(k)}$ denote the aggregate arrival probability vectors of new and relayed traffic to queue $k$, respectively. We have

$$\mathbf{a}_{\text{new}}^{(k)} = \otimes_{c \in \Phi_{\text{new}}^{(k)}} \mathbf{a}^{(k,c)} \tag{6.18}$$

where $\mathbf{a}_{\text{new}}^{(k)}$ is obtained by taking the convolutions of the arrival probability vectors $\mathbf{a}^{(k,c)}$. As can be seen in Fig. 6.13, data packets successfully transmitted from queue $k$ may enter queue $k+1$ (i.e., relayed traffic) or leave the considered tandem system (i.e., leaving traffic). These data packets which constitute the relayed traffic and leaving traffic flows belong to connections in the sets $\Phi_{\text{rel}}^{(k)}$ and $\Phi_{\text{lev}}^{(k)}$, respectively. Given the allocated bandwidth for link $l$ and the arrival probabilities to queue $k$, we can calculate the probabilities for data packets successfully transmitted from queue $k$ as in (6.7). Let $\mathbf{a}_{\text{suc}}^{(k)}$ be the probability vector for data packets successfully transmitted from queue $k$ where its element $\mathbf{a}_{\text{suc},i}^{(k)}$ can be approximated by adapting equation (6.7) as follows:

$$\mathbf{a}_{\text{suc},i}^{(k)} \approx \sum_{j=0}^{Q_k} \sum_{l=0}^{K} \pi_j^{(k)} p_l^{(k)} \times \gamma^{(k)}(\min\{j,l\},i) \tag{6.19}$$

where $\pi_j^{(k)}$, $p_l^{(k)}$, and $\gamma^{(k)}(\min\{j,l\},i)$ are defined as in Section 6.2 and $\mathbf{a}_{\text{suc},i}^{(k)}$ is the probability that $i$ packets are successfully transmitted from queue $k$ in one time frame. Denoting the aggregate traffic arrival rate to queue $k$ as $\overline{A}^{(k)}$, we can approximately calculate the probability vector for data packets of connection $c$ which is successfully transmitted from queue $k$ by scaling $\mathbf{a}_{\text{suc}}^{(k)}$ with a number representing the ratio between traffic arrival rate of connection $c$ to queue $k$ and the aggregate traffic arrival rate to queue $k$ as follows:

$$\begin{aligned} \mathbf{a}_i^{(k+1,c)} &\approx \frac{\overline{A}^{(k,c)}}{\overline{A}^{(k)}} \times \mathbf{a}_{\text{suc},i}^{(k)}, \quad 1 \le i \le N_k \\ \mathbf{a}_0^{(k+1,c)} &\approx 1 - \sum_{i=1}^{N_k} \mathbf{a}_i^{(k+1,c)} \end{aligned} \tag{6.20}$$

where $N_k$ is the maximum number of packets transmitted in one time frame from queue $k$. Similarly, the arrival probabilities for the aggregate relayed traffic to queue $k+1$ can be approximated as

$$\begin{aligned} \mathbf{a}_{\text{rel},i}^{(k+1)} &\approx \frac{\sum_{c\in\Phi_{\text{rel}}^{(k)}} \overline{A}^{(k,c)}}{\overline{A}^{(k)}} \times \mathbf{a}_{\text{suc},i}^{(k)}, \quad 1 \le i \le N_k \\ \mathbf{a}_{\text{rel},0}^{(k+1)} &\approx 1 - \sum_{i=1}^{N_k} \mathbf{a}_{\text{rel},i}^{(k+1)}. \end{aligned} \tag{6.21}$$

The arrival probabilities for the aggregate relayed traffic leaving queue $k$ can be approximated as

$$\begin{aligned} \mathbf{a}_{\text{lev},i}^{(k)} &\approx \frac{\sum_{c\in\Phi_{\text{lev}}^{(k)}} \overline{A}^{(k,c)}}{\overline{A}^{(k)}} \times \mathbf{a}_{\text{suc},i}^{(k)}, \quad 1 \le i \le N_k \\ \mathbf{a}_{\text{lev},0}^{(k)} &\approx 1 - \sum_{i=1}^{N_k} \mathbf{a}_{\text{lev},i}^{(k)}. \end{aligned} \tag{6.22}$$

The aggregate traffic arrival probability vector to queue $k+1$, therefore, can be calculated as

$$\mathbf{a}^{(k+1)} = \mathbf{a}_{\text{new}}^{(k+1)} \otimes \mathbf{a}_{\text{rel}}^{(k+1)} \tag{6.23}$$

where the arrival probability vectors for the new and relayed traffic flows are calculated by (6.18) and (6.21), respectively. Let us assume that traffic arriving to queue one is from the incoming connection with an arrival probability vector denoted by $\mathbf{a}_{\text{in}}$ and from the relayed traffic with an arrival probability vector denoted by $\mathbf{a}_{\text{rel}}^{(1)}$. Therefore, the aggregate arrival probability vector to queue one is $\mathbf{a}^{(1)} = \mathbf{a}_{\text{in}} \otimes \mathbf{a}_{\text{rel}}^{(1)}$.

In summary, the arrival probability vector to queue one can be calculated as $\mathbf{a}^{(1)} = \mathbf{a}_{\text{in}} \otimes \mathbf{a}_{\text{rel}}^{(1)}$ and the steady state probability vector for queue one $\boldsymbol{\pi}^{(1)}$ can be calculated as in Section 6.3. With the steady state probability vector $\boldsymbol{\pi}^{(1)}$, we can calculate the aggregate arrival probability to queue two by using (6.23). This procedure is repeated until the solution for the last queue of the tandem system is found.

### 6.6.2 QoS Routing Algorithm

With class-based queueing, we now show how to integrate the tandem queueing model presented in the previous subsection into the QoS routing algorithm. We only discuss the route discovery phase. As before, the incoming connection submits its traffic profile and service class with QoS requirements to the source node. For ongoing connections, we require each node along the routing path to record the aggregate arrival probability vectors to all queues of different service classes in all outgoing links and the current link QoS metrics on these links (i.e., link delay and loss) in its route cache.

The source node initiates the route discovery to find feasible routes to its desired destination as follows. First, the source node checks each outgoing link to see whether it has enough bandwidth to accommodate the incoming connection. The outgoing link with enough bandwidth will be called the BW-feasible link as before. Then, the required amount of bandwidth is allocated to the BW-feasible link which increases the average service rate of the queue corresponding to the service class of the incoming connection. Note that this queue may be buffering data flows of other ongoing connections. For each BW-feasible link, the source node calculates the updated aggregate arrival probability vector to the corresponding queue as follows:

$$\mathbf{a}^{(1)} = \mathbf{a}_{\text{in}} \otimes \mathbf{a}_{\text{rel}}^{(1)} \tag{6.24}$$

where $\mathbf{a}_{\text{in}}$ and $\mathbf{a}_{\text{rel}}^{(1)}$ denote the arrival probability vector of the incoming connection and the aggregate arrival probability vector of ongoing connections being relayed on the considered link, respectively. If there is no ongoing connection on the explored link, we can simply set $\mathbf{a}_{\text{rel}}^{(1)} = 1$.

The source node calculates link QoS metrics for each BW-feasible link using the updated arrival probability vector and allocated bandwidth. Based on the calculated QoS metrics, the RRQ packet is forwarded to receiving nodes via links which satisfy the QoS requirements of the incoming connection and do not degrade the QoS performances of the ongoing connections traversing these links. This guarantees that the QoS requirements of ongoing connections are not violated after admitting the incoming connection into the network. In this case, the source node calculates the aggregate relayed arrival probabilities to the next queue by using (6.21) and records this arrival probability vector into the RRQ packet header.

Now, each node upon receiving the RRQ packet will check the available bandwidth on its outgoing links. For each BW-feasible outgoing link, the node calculates the aggregate arrival probability vector from both the incoming connection and the ongoing connections traversing the explored link using (6.23). If the link QoS metrics for the incoming connection are satisfied and those for the ongoing connections traversing the link are not degraded due to the admission of the incoming connection, the RRQ packet will be forwarded to the receiving node of that link after the path QoS metrics and the aggregate relayed arrival probabilities have been recorded into the RRQ packet header. This procedure is repeated until either a feasible route to the destination is found or the incoming connection is blocked. The destination upon receiving RRQ packet with satisfied path QoS metrics will send a RRP packet to the source node as in Section 6.4. Finally, we need to update the arrival probability vectors to the queues along routing paths of affected ongoing connections and the QoS metrics of the affected links.

**Example:** Let us consider the network topology shown in Fig. 6.14 where there are two ongoing connections denoted as connection one and two when the incoming connection (connection three) requests to establish its communication section. All these connections belong to the same service class. Assume that each connection requires two channels for each link along its routing path. Assume that links one and two constitute a routing path for the incoming connection; other links are also numbered for the ease of reference. For this network setting, link one requires six channels, link two requires four channels, and each of the other links requires two channels. Assume that the best routing path for connection three is through link one and two as shown in the figure. The allocated bandwidth on link one and two should be sufficient to transmit the aggregate traffic under required QoS due to three connections on link one and due to connections one and three on link two. The aggregate arrival probability vector to link one queue is $\mathbf{a}^{(1)} = \mathbf{a}_{\text{in}} \otimes \mathbf{a}_{\text{rel}}^{(1)}$, where $\mathbf{a}_{\text{in}}$ and $\mathbf{a}_{\text{rel}}^{(1)}$ are the arrival probability vectors due to connection three and due to connections one and two, respectively. Also, the aggregate traffic transmitted over link two is only due to connections one and three. After this routing path is found, the arrival probability vector and link QoS metrics over link five should be updated and stored in the transmitter node for this link.

## 6.7 Tandem Queue with Blocking

In this subsection, we discuss the implementation and solution issues for tandem queues with blocking. In particular, queue $k+1$ ($k > 0$) in the tandem system may block transmissions from queue $k$ if its buffer is full. In addition, queue $k$ should know how many packets queue $k+1$ can accommodate in each time interval. To avoid buffer overflow, the number of transmitted packets from queue $k$ should be kept smaller than the room available in queue $k+1$.

Although it is possible to implement this blocking option, the communication overhead involved may not be desirable for most wireless applications. Also, the blocking implementation will result in higher overflow probability in queue one which may not be able to block arrivals from the underlying applications. Blocking may also increase the e2e delay and even e2e loss rate (due to high buffer overflow in queue one). Therefore, in the context of QoS routing where only routing paths with satisfied QoS requirements are chosen for e2e data transmission, it may be more desirable to avoid the blocking implementation.

For tandem queue with blocking, we can use a decomposition method which is similar in spirit to the method in [74]. The decomposition method is iterative and it works as follows. In each iteration, we solve all pairs of consecutive queues in the tandem and find steady state probability vectors (i.e., queue one and two, queue two and three and so on). Also, the steady state probability vectors of queues $k-1$ and $k+2$ in the iteration $t$ are used to solve a pair of queues $k$ and $k+1$ in iteration $t+1$ with blocking being taken into account. The calculation is repeated until a predefined convergence criterion is met.

## 6.8 Chapter Summary

We have proposed both exact and approximated decomposition approaches to solve a general tandem queue system. The proposed tandem queueing models capture realistic physical and link layer designs where the multi-rate transmission feature due to adaptive modulation and coding in the physical layer and the ARQ-based error recovery in the link layer have been incorporated into the queueing models. The proposed decomposition approach achieves very accurate queueing performance measures with much lower computational complexity compared to the exact approach. Using the decomposition queueing approach, we have constructed a unified queueing and QoS routing framework which is able to satisfy the QoS requirements in multi-hop wireless networks. The numerical results have shown that the framework works efficiently to find feasible routing paths in the network if they exist. The extension of the framework to wireless networks with class-based queueing implementation has also been presented.

# Chapter 7

# Cross-Layer Frameworks for Cooperative Wireless Networks

In this chapter, we deal with the multi-path routing problem for multi-hop wireless networks [68]-[70]. The decode-and-forward cooperative diversity is considered in the physical layer to enhance the network performance [81]. In general, multi-path routing applies to splittable traffic where traffic from each source node can be split into multiple flows which follow different routes to reach the destination. Here, the routing problem reduces to finding flow rate on each wireless link considering flow conservation and other radio resource constraints.

There are some initial works on higher layer protocols with cooperative diversity in the literature: MAC layer protocol design [86], [87], [90], cooperative routing [88], cooperative multicasting [89]. These works, however, deal with protocol design issues in one single layer or investigate simple inter-layer interactions, which still lack the system-wide insight. A systematic approach to cross-layer design of multi-hop wireless networks using cooperative diversity is important to achieve the maximum gain of cooperative diversity in the physical layer and harmonize the interaction with higher layers in such a way that the system performance is optimized. We develop such cross-layer optimization frameworks in this chapter.

Over the past recent years, nonlinear optimization has been proved to be an important tool to design and engineer wireless protocols and to construct distributed algorithms. In fact, the dual decomposition technique in convex optimization has been used for reverse-engineering of popular protocols such as TCP [110] and for optimal resource allocation and cross-layer design [70], [76]-[78]. In [76], an excellent survey on cross-layer design using convex optimization was provided where several design problems were succinctly presented.

These works, however, did not consider cooperative diversity which can potentially enhance the network performance considerably. In this chapter, we will apply the nonlinear optimization techniques to develop optimization frameworks for multihop wireless networks using cooperative diversity. Inspired by the work in [108], where it has been shown that the single "best" relay can achieve the whole diversity-multiplexing tradeoff, we allow only the best relay (if any) to be involved in the relaying process. The proposed frameworks incorporate congestion control, routing, relay selection, and power allocation tasks in different layers of the protocol stack. Based on these frameworks two algorithms are proposed. The first one is the joint routing and cooperative resource allocation algorithm which minimizes the total power consumption. The second one incorporates congestion control though a utility function to strike a balance between maximizing rate utility and minimizing power consumption.

The proposed algorithms are fully distributed where each node in the network iteratively updates the relevant system variables using other variables which are available through local message exchange. The potential gain from cooperative diversity in the physical layer is exploited to optimize the system-wide performance in a unified way. The proposed frameworks, therefore, solve the cross-layer design and optimization problems for multi-hop wireless networks using cooperative diversity from the system perspective.

## 7.1 System Models and Assumptions

Consider a multi-hop wireless network as a directed graph $G = (V, E)$, where $V$ is the set of nodes and $E$ is the set of directed links. There is a link between a pair of nodes if the communication link can be established between them. We denote the link from node $i$ to node $j$ as $(i, j)$ and its channel gain as $g(i, j)$. We define $O(i)$ as the set of links going out of node $i$ and $I(i)$ as the set of links going into node $i$. For the ease of referencing and understanding, we abuse the notation a bit by using the same notations for Lagrangian, dual function and others in different problem formulations. However, the notations should be unambiguous from the context.

### 7.1.1 Transmission Rate with Cooperative Diversity

We assume that transmissions on different links in a common neighborhood use orthogonal channels. To forward data from node $i$ to node $j$ of link $(i, j)$, either direct transmission is used or a particular node $k$ helps node $i$ to forward data to node $j$ using decode-and-forward (DF) cooperative diversity. We assume that an available channel pool has been assigned for different links such that simultaneous transmissions are possible where transmission on one link causes very weak interference to others. This assumption on orthogonal transmissions can be achieved, for example, by using different codes for different links in a code-division multiple access (CDMA) network. The code allocation algorithms for such a scenario are available in the literature [102]. Another possible scenario justifying this assumption is a multi-channel multi-radio wireless network where orthogonal channels are allocated for simultaneous transmissions on different links in each neighborhood using separate radios in each node [105].

We will develop distributed algorithms for optimizing different design objectives as will be mentioned in the next subsection. These algorithms will be run when the traffic load in the network changes due to the arrival or leaving of a particular section (i.e., a particular node starts or stops transmitting data to its desired destination) or the network topology changes. In essence, the frequent activation of these algorithms offsets the network design parameters (e.g., link flows, power) to compensate for the network changes. We assume that total interference and noise power at the receiving end of each link remains static during the running time of the algorithm. In addition, this interference and noise power is assumed to be estimated by the receiving node and fed back to the transmitting node periodically. In fact, the similar assumption was made in [64], [109] where the packet error rate (PER) for each link in the network was assumed to be static during the running time of their proposed on-demand routing algorithms. Since the PER depends directly on the signal to interference and noise ratio, the assumption in these papers is, therefore, equivalent to the assumption we make here in this chapter.

Let the transmission power for the direct transmission be $P_d(i, j)$, and the total interference and AWGN noise at the receiving side of this link be $N_0(i, j)$. Then, the achievable rate (b/s/Hz) in case of direct transmission is

$$r_d(i,j) = \log_2 \left(1 + \frac{g(i,j)P_d(i,j)}{GN_0(i,j)}\right) \tag{7.1}$$

where $G$ denotes the gap to capacity. For notational convenience, we will absorb $GN_0(i,j)$ into $g(i,j)$. Thus we can write

$$r_d(i,j) = \log_2 \left(1 + g(i,j)P_d(i,j)\right). \tag{7.2}$$

For the assumed DF cooperative diversity scheme, time is slotted and node $i$ transmits the data packets in the first time slot which are received by node $j$ and the relay node $k$. Relay node $k$ decodes the packets and forwards them to node $j$ in the second time slot. Let $P_{r,(i,j)}(k,j)$ be the power used at relay $k$ to forward packets to node $j$ for the direct link $(i,j)$. At node $j$, for decoding, the received signal in the second time slot will be combined with that due to the direct transmission. The achieved data rate at relay node $k$ in the first time slot and at node $j$ in the second time slot after using maximum ratio combining are given by

$$r_r(i,k) = \log_2 \left(1 + g(i,k)P_d(i,j)\right), \tag{7.3}$$

$$r_c(i,j) = \log_2 \left(1 + g(i,j)P_d(i,j) + g(k,j)P_{r,(i,j)}(k,j)\right), \tag{7.4}$$

respectively. Note that, for relaying to be useful, the achieved rate at relay node $k$ must be higher than that due to direct transmission. This condition can be easily found to be $g(i,j) < g(i,k)$ which will be used to limit the search for the best relay node for each link. To illustrate the useful cooperative region we assume that channel gains are simply due to large-scale path loss. Only nodes inside the circle illustrated in Fig. 7.1 are useful relays for the considered wireless link. It is also intuitive that wireless nodes in the forward direction toward the receiving node are potentially good relays for the link. For energy-efficient communications, we would allocate the right amount of power to relay node $k$ to forward data to destination node $j$ such that $r_c(i,j) = r_r(i,k)$. This condition is equivalent to

$$g(i,k)P_d(i,j) = g(i,j)P_d(i,j) + g(k,j)P_{r,(i,j)}(k,j). \tag{7.5}$$

Let the maximum achievable data rate on link $(i,j)$ be $r(i,j)$. This achievable rate on each link depends on the transmission powers, link gains, the transmission strategy (i.e., direct or cooperative transmission through a relay node). Thus, physical layer design goals considered in this chapter are to choose the optimal transmission strategy for each link and to allocate the optimal transmission power level for the chosen strategy such that the system performance captured through appropriate objective functions is optimized. From the above analysis, the achievable transmission rate on link $(i,j)$ for different transmission strategies can be written as

$$r(i,j) = \begin{cases} r_d(i,j), & \text{for direct transmission} \\ \frac{r_c(i,j)}{2} = \frac{r_r(i,j)}{2}, & \text{for cooperative transmission.} \end{cases} \tag{7.6}$$

Note that the actual data rate achieved by cooperative diversity is $r_c(i,j)/2$ because two time slots are needed to transmit the data. For the same reason, the average total power used for cooperative transmission can be written as

$$P_t(i,j) = (P_d(i,j) + P_{r,(i,j)}(k,j))/2. \tag{7.7}$$

Note that transmission power for the case of direct transmission is simply $P_t(i,j) = P_d(i,j)$. Now, we define the network flow concept and utility function to construct the objective functions in the following subsection.

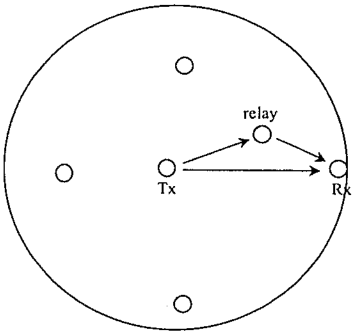


**Figure 7.1.** *Cooperative diversity and useful cooperative region.*

### 7.1.2 Network Flow and Utility Function

We use the network flow model for routing data to a single node in the network such as an access point in a wireless LAN or a data sink in a wireless sensor network. In this model, each node $i \neq d$ generates data with an average rate of $S_i$ to destination $d$. The total data rate received at the destination node $d$ is, therefore, $S_d = -\sum_{i \neq d} S_i$. We assume that a multipath routing protocol is used in the network layer where traffic from each source node is split into several flows which follow different multi-hop paths to reach the desired destination. Define link flow $x(i,j)$ to be the average aggregate transmission rate on link $(i,j)$. The aggregate data transmitted on each link may come from different source nodes under the multipath routing assumption. For flow conservation, the total flow going into a node is the same as total flow going out of that node. Hence,

$$\sum_{j \in O(i)} x(i,j) - \sum_{j \in I(i)} x(j,i) = S_i, \quad i \in V. \tag{7.8}$$

The flow on any link $(i,j)$ should be smaller than its transmission rate

$$x(i,j) \leq r(i,j), \quad (i,j) \in E. \tag{7.9}$$

Different objective functions can be optimized depending on the application context. In wireless networks, power minimization is usually one of the biggest concerns because the mobile devices (e.g., sensors) are energy-limited and also the transmitted energy by a user causes interference to other users. Let $P(i,j) = P_d(i,j) + \sum_k P_{r,(k,j)}(i,j)$ be the total power consumption on link $(i,j)$ both for its own transmission and for relaying packets from other links. When power consumption is the major concern, the objective function can be min $\sum_{(i,j)\in E} P(i,j)$.

Another objective can be to maximize the sum utility of the source rates. In this case, the objective function can be written as max $\sum_{i \in V} U_i(S_i)$, where the different utility functions $U_i(S_i)$ as in [110] for congestion avoidance and fairness control of different data sessions can be used. Another possible objective can be to strike a balance between maximizing sum utility of source rates and minimizing power consumption. For convenience, we will denote the link and node quantities such as link power, link flow, source rate for all links (node) in the network into the corresponding vectors. For example, the vector of link flows will be denoted as $\mathbf{x}$ with elements $x(i,j)$ being the link flow for each link $(i,j)$.

## 7.2 Joint Routing and Cooperative Resource Allocation

In this section, we develop a distributed algorithm for the joint routing and cooperative resource allocation problem. For this problem, the source rates $S_i$ are assumed to be fixed and the optimization problem, in essence, is the joint routing, relay selection, and power allocation in a multi-hop wireless network. The goals are to find the optimal transmission strategy for each link (i.e., either direct or cooperative transmission), the optimal power allocation for the chosen strategy, and the link flow to route data generated by the source nodes to the corresponding destination node. In essence, this is a cross-layer design problem for both physical layer (i.e., relay selection, power allocation) and network layer (i.e., routing of traffic flows). The problem can be stated as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in E} P(i,j) \\ \text{subject to} & \mathbf{x} \succeq \mathbf{0}, \quad \mathbf{P}_{\min} \preceq \mathbf{P} \preceq \mathbf{P}_{\max}, \\ & \text{and constraints in} \quad (7.8), (7.9). \end{array} \tag{7.10}$$

where $\preceq$, $\succeq$ denote the component-wise inequalities, $\mathbf{P}_{\min}$ ($\mathbf{P}_{\max}$) denotes the lower (upper) limit for the power vector with element $P_{\min}(i,j)$ ($P_{\max}(i,j)$) being the minimum (maximum) allowed power to transmit or relay packets on link $(i,j)$.

We will form the dual problem by introducing the Lagrange multipliers for constraints in (7.8) and (7.9). The Lagrangian can be written as

$$\begin{aligned} L(\mathbf{x},\mathbf{P},\boldsymbol{\lambda},\boldsymbol{\mu}) &= \sum_{(i,j)\in E} P(i,j) + \sum_{(i,j)\in E} \lambda_{(i,j)} \left[x(i,j) - r(i,j)\right] \\ &\quad + \sum_{i\in V} \mu_i \left[\sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i) - S_i\right] \end{aligned} \tag{7.11}$$

where $\mathbf{x}$ is the vector of link flows, $\mathbf{P}$ is the vector of allocated powers, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the vectors of Lagrange multipliers. The elements of these vectors for each link $(i,j)$ and node $i$ (i.e., $x(i,j)$, $P(i,j)$, $\lambda_{(i,j)}$, and $\mu_i$) are maintained at node $i$. From this Lagrangian, we can define the dual function $D(\lambda,\mu)$ as follows:

$$D(\boldsymbol{\lambda},\boldsymbol{\mu}) = \inf_{\{\mathbf{x},\mathbf{P}\}} L(\mathbf{x},\mathbf{P},\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{7.12}$$

The optimization problem defined in (7.10) is a convex one where the strong duality holds, and therefore, the duality gap is zero. In fact, for the inequality constraint in (7.9), we can choose the allocated power large enough such that (7.9) is satisfied with strict inequality so that the Slater's condition for strong duality holds [116]. This basically holds with $\mathbf{P}_{\max}$ being large enough. The original optimization problem in (7.10) is called the primal problem and its solution can be can be recovered via what is called the dual problem which can be written as

$$\begin{array}{ll} \text{maximize} & D(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{subject to} & \boldsymbol{\lambda} \succeq 0 \end{array} \tag{7.13}$$

where the Lagrange multipliers for the inequality constraints in (7.9) are constrained to be non-negative. The decision variables for the primal and dual problems are called primal variables (i.e., $\mathbf{x}$ and $\mathbf{P}$) and dual variables ($\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$), respectively.

Thus, the underlying optimization problem can be solved directly (i.e., by solving the primal problem (7.10)) or its solution can be obtained through solving the dual problem (7.13). Solving the primal problem usually results in a centralized algorithm where all network information such as link gains, interference and noise powers at the receiving end of all links should be sent to a particular node to calculate the link flow and allocated power solutions and these solutions should be distributed to the corresponding nodes in the network. The centralized algorithm, therefore, incurs huge communication overhead and lack of resilience to network changes. Tackling the dual problem using the dual decomposition method leads to distributed algorithms which are more useful for wireless networks without infrastructure such as ad hoc networks. In these distributed algorithms each node performs iterative exchanges of variables with its immediate neighbors. The dual decomposition method will be used to construct the distributed routing algorithm in the following subsections.

### 7.2.1 Dual Decomposition and Subgradient Method

Since the objective of the primal problem (7.10) is not strictly convex, the primal variables (i.e., flow and power vectors $\mathbf{x}$, $\mathbf{P}$) may not be immediately available from the dual problem solutions. To handle this difficulty, we use the approaches in [118] by adding a small regularization term $\epsilon \sum_{(i,j)\in E} x(i,j)^2$ into the objective function in

(7.10). By letting $\epsilon \to 0$, the optimal solution to the regularized problem tends to that in (7.10). The Lagrangian of the regularized primal problem can be written as

$$\begin{aligned} L(\mathbf{x}, \mathbf{P}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \sum_{(i,j)\in E} P(i,j) + \epsilon \sum_{(i,j)\in E} x(i,j)^2 + \sum_{(i,j)\in E} \lambda_{(i,j)} \left[x(i,j) - r(i,j)\right] \\ &\quad + \sum_{i\in V} \mu_i \left[\sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i) - S_i\right] \\ &= \left\{\epsilon \sum_{(i,j)\in E} x(i,j)^2 + \sum_{(i,j)\in E} \lambda_{(i,j)} x(i,j) \right. \\ &\quad \left. + \sum_{i\in V} \mu_i \left[\sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i)\right]\right\} \\ &\quad + \left\{\sum_{(i,j)\in E} P(i,j) - \sum_{(i,j)\in E} \lambda_{(i,j)} r(i,j)\right\}. \end{aligned} \tag{7.14}$$

The first term in the above equation depends only on the primal flow variables $x(i,j)$ and the second term depends only on the primal power variables $P(i,j)$ (because the achievable rate $r(i,j)$ for each link $(i,j)$ depends on $P(i,j)$ as modeled in Section 7.1.1). Thus, the dual function can be calculated by decomposing the optimization problem in (7.12) into the following subproblems:

$$D_{\text{phy}}(\boldsymbol{\lambda}) = \min_{\mathbf{P}} \left\{\sum_{(i,j)\in E} P(i,j) - \sum_{(i,j)\in E} \lambda_{(i,j)} r(i,j)\right\} \tag{7.15}$$

$$\begin{aligned} D_{\text{net}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \min_{\mathbf{x}\succeq 0} \left\{\epsilon \sum_{(i,j)\in E} x(i,j)^2 + \sum_{(i,j)\in E} \lambda_{(i,j)} x(i,j) \right. \\ &\quad \left. + \sum_{i\in V} \mu_i \left[\sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i)\right]\right\} \end{aligned} \tag{7.16}$$

and the dual function can be written as $D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = D_{\text{net}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) + D_{\text{phy}}(\boldsymbol{\lambda})$.

Thus, given the dual variables (i.e., $\lambda_{(i,j)}$, $\mu_i$), the dual function can be calculated by solving the routing subproblem in the network layer (i.e., in (7.16)) and the relay selection and power allocation subproblem in the physical layer (i.e., in (7.15)). Since the dual function may not be differentiable, we solve the dual problem using the subgradient projection method [119], [120]. The definition of sugradient was given in *Chapter 2* and the subgradients of the dual function are stated in the following Lemma.

**Lemma 7.1:** Subgradient of $-D(\lambda, \mu)$ at $\lambda_{(i,j)}$ and $\mu(i)$ are

$$f_1(\lambda_{(i,j)}) = r(i,j) - x(i,j) \tag{7.17}$$

$$g_1(\mu_i) = \sum_{j \in I(i)} x(j,i) - \sum_{j \in O(i)} x(i,j) + S_i. \tag{7.18}$$

The subgradient projection method is similar to the gradient projection method but the subgradient instead of the gradient of the objective function is used in each iteration. Given the locally optimal solutions $\mathbf{x}^*(t)$ and $\mathbf{P}^*(t)$ of the networking and the physical layer subproblems, the subgradient projection algorithm updates $\lambda_{(i,j)}$ and $\mu(i)$ as follows:

$$\lambda_{(i,j)}(t+1) = \left[\lambda_{(i,j)}(t) - \beta(t) f_1(\lambda_{(i,j)})\right]^+ \tag{7.19}$$

$$\mu_i(t+1) = \mu_i(t) - \beta(t) g_1(\mu_i(t)) \tag{7.20}$$

where $[x]^+ = \max(0, x)$ and $\beta(t)$ is the stepsize in iteration $t$.

In fact, the proposed algorithm iteratively updates the dual variables (i.e., using (7.19) and (7.20)) and primal variables (i.e., by solving (7.15) and (7.16) ) until the globally optimal solutions are obtained. Specifically, the subgradients of the dual function (i.e., $f_1(\lambda_{(i,j)})$ and $g_1(\mu_i)$) reflect the degree by which the constraints in (7.8), (7.9) are violated. Updating the dual variables based on the subgradient algorithm also has an interesting economic interpretation where the dual variables represent the shadow prices which strike a balance between the supply (transmission power) and demand (link flow) in such a way that globally optimal solutions can be achieved. We will solve these two subproblems in the next subsection. We will refer to $\lambda_{(i,j)}$ as the link price and $\mu_i$ as the node price in the sequel.

### 7.2.2 Networking and Physical Subproblem Solutions

We will solve the netwoking (i.e., routing) subproblem (7.16) and the physical layer subproblem (7.15) in this subsection. The routing subproblem in (7.16) can rewritten as follows:

$$D_{\text{net}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{(i,j)\in E} D_{\text{net}}(i,j) \tag{7.21}$$

where

$$D_{\text{net}}(i,j) = \min_{x(i,j)\geq 0} \left\{ \epsilon x(i,j)^2 + \lambda_{(i,j)} x(i,j) + \mu_i x(i,j) - \mu_j x(i,j) \right\}. \tag{7.22}$$

The routing subproblem, therefore, can be decomposed into multiple link subproblems. Given the link and node prices $\lambda_{(i,j)}$, $\mu_i$, $\mu_j$ for each link $(i,j)$ (these variables are maintained/available at node $i$), node $i$ calculates the locally optimal link flow as

$$x^*(i,j) = \left[ \frac{\mu_j - \mu_i - \lambda_{(i,j)}}{2\epsilon} \right]^+. \tag{7.23}$$

The physical layer subproblem in (7.15) is more difficult to solve because it involves both relay selection and power allocation. The physical layer subproblem (7.15) can also be decomposed into multiple link subproblems as

$$D_{\text{phy}}(\boldsymbol{\lambda}) = \sum_{(i,j)\in E} D_{\text{phy}}(i,j) \quad \text{where} \quad D_{\text{phy}}(i,j) = \min_{\mathbf{P}} \left\{ P_t(i,j) - \lambda_{(i,j)} r(i,j) \right\}. \tag{7.24}$$

For each link $(i,j)$, either direct transmission from node $i$ to node $j$ or cooperative transmission with the help a particular relay node $k$ can be pursued. Here, given the link price $\lambda_{(i,j)}$ for each link $(i,j)$, we need to find the best transmission strategy and the corresponding allocated power for it. From (7.24), if direct transmission is pursued for link $(i,j)$, we have

$$D_{\text{phy}}(i,j) = \min_{P_d(i,j)} \left\{ P_d(i,j) - \lambda_{(i,j)} r(i,j) \right\}. \tag{7.25}$$

Otherwise, if a neighboring node $k$ involves in the cooperative transmission, we have

$$D_{\text{phy}}(i,j) = \min_{P_d(i,j), P_{r,(i,j)}(k,j)} \left\{ (P_d(i,j) + P_{r,(i,j)}(k,j))/2 - \lambda_{(i,j)} r(i,j) \right\}. \tag{7.26}$$

Given link price $\lambda_{(i,j)}$, the best transmission strategy is the one which results in the smallest $D_{\text{phy}}(i,j)$. Since node $i$ has a finite number of neighbors which can serve as a relay for cooperative transmission on link $(i,j)$, the best transmission strategy can be easily searched for. Note that as discussed in Section 7.1, a possible relay node $k$ must satisfy $g(i,k) > g(i,j)$ for relaying to be useful. This condition limits the number of potential relay candidates. Now, we show how to calculate the locally optimal allocated power for direct transmission and for cooperative transmission. Based on these possible transmission strategies, the best transmission strategy can be easily found. For direct transmission, we have $r(i,j) = r_d(i,j) = \log_2(1 + g(i,j)P_d(i,j))$. Therefore, the optimal power can be easily found by setting the derivative of the objective function in (7.25) to zero as

$$P_d^*(i,j) = \left[\frac{\lambda_{(i,j)}}{\log 2} - \frac{1}{g(i,j)}\right]_{P_{\min}(i,j)}^{P_{\max}(i,j)} \tag{7.27}$$

where $[x]_a^b$ denotes the projection of $x$ on $[a,b]$. For cooperative transmission through relay node $k$, we have $r(i,j) = r_r(i,k)/2 = 1/2\log_2(1 + g(i,k)P_d(i,j))$. By setting the derivative of the objective function in (7.26) to zero and using condition (7.5), the locally optimal allocated powers can be obtained as

$$P_d^*(i,j) = \left[\frac{\lambda_{(i,j)}g(k,j)}{\log 2\,(g(i,k) + g(k,j) - g(i,j))} - \frac{1}{g(i,k)}\right]_{P_{\min}(i,j)}^{P_{\max}(i,j)} \tag{7.28}$$

$$P_{r,(i,j)}^*(k,j) = \left[\frac{g(i,k) - g(i,j)}{g(k,j)}P_d^*(i,j)\right]_{P_{\min}(k,j)}^{P_{\max}(k,j)}. \tag{7.29}$$

We now summarize the solution for the physical layer subproblem. The optimal allocated power for direct transmission is given in (7.27). For cooperative transmission through relay $k$, the optimal allocated powers for transmitter node $i$ and for relay node $k$ are given in (7.28) and (7.29), respectively. Given the optimal power solutions for these possible transmission strategies on link $(i,j)$, the corresponding $D_{\text{phy}}(i,j)$ can be calculated by using (7.25) for direct transmission and (7.26) for cooperative transmission. Then, the transmission strategy achieving the smallest $D_{\text{phy}}(i,j)$ is chosen and the corresponding transmission rate $r(i,j)$ is used to calculate the subgradient $f_1$ in the subgradient algorithm. The joint routing and cooperative resource allocation algorithm is summarized in **Algorithm 7.1**. The convergence of the algorithm is presented in chapter 2 where the proof can be found, for example, in [119].

**Algorithm 7.1: Joint Routing and Cooperative Resource Allocation**

1. Each node $i$ initializes its node price $\mu_i(0)$ and link price $\lambda_{(i,j)}(0)$, link flow $x(i,j)(0)$, transmission power $P_d(i,j)(0)$ for each outgoing links $(i,j)$.
2. Given $\lambda_{(i,j)}(t)$ and $\mu_i(t)$, each node $i$ solves the networking and physical subproblems for its outgoing link $(i,j)$ to obtain the locally optimal link flow $x^*(i,j)(t)$, transmission strategy, allocated powers for itself $P_d^*(i,j)(t)$ and for its relay partner $P^*_{r,(i,j)}(k,j)(t)$ if cooperative transmission through node $k$ is the best strategy. Node $i$ then transmits the link flow value $x^*(i,j)(t)$ to node $j$ and relay power $P^*_{r,(i,j)}(k,j)(t)$ to its relay partner $k$.
3. Given the locally optimal link flow, transmission strategy, and allocated powers, each node $i$ updates the link and node prices $\lambda_{(i,j)}(t+1)$, $\mu_i(t+1)$ using (7.19) and (7.20). Node $i$ transmits $\mu_i(t)$ to all neighboring nodes $j$.
4. Return to 2) until the algorithm converges.

We observe that **Algorithm 7.1** is fully distributed because each node $i$ in the network performs all its calculations using only variables which are available from its immediate neighbors. Specifically, node $i$ finds the optimal transmission strategy and optimal transmission powers using (7.27) (for direct transmission strategy) and (7.28), (7.29) (for cooperative transmission through node $k$). These calculations require $g(i,j)$, $g(i,k)$, $g(k,j)$ and the corresponding interference and noise powers (recall that we have absorbed these values into the corresponding channel gains) which can be made available at node $i$ though some estimation technique and local message exchanges. Similarly, the calculations of link flow $x_{(i,j)}(t)$, link and node prices $\lambda_{(i,j)}(t)$, $\mu_i(t)$ performed by node $i$ require variables which are available through message exchange operations in steps 2, 3 of the proposed algorithm.

## 7.3 Utility-Power Tradeoff With Cooperative Resource Allocation

In this section, we extend the presented algorithm to incorporate congestion control. In essence, the objective function under consideration strikes a balance between maximizing the sum utility and minimizing the total power consumption. The utility function $U_i(S_i)$ is assumed to be continuously differentiable, increasing, and strictly concave. In order to recover the optimal solution for the primal variables, we add a regularization term into the objective function as before. The optimization problem is given as

$$\begin{array}{ll} \text{maximize} & \gamma_1 \sum_{i \in V} U_i(S_i) - \gamma_2 \sum_{(i,j) \in E} P(i,j) - \epsilon \sum_{(i,j) \in E} x(i,j)^2 \\ \text{subject to} & \mathbf{x} \succeq \mathbf{0}, \quad \mathbf{S} \succeq \mathbf{0}, \quad \mathbf{P}_{\min} \preceq \mathbf{P} \preceq \mathbf{P}_{\max}, \\ & \quad \text{and constraints in} \quad (7.8), (7.9) \end{array} \tag{7.30}$$

where $\mathbf{S}$ is the vector of source rates with elements $S_i$ being the average data rate generated by node $i$ and $\gamma_1$, $\gamma_2$ are the parameters controlling the tradeoff.

### 7.3.1 Dual Decomposition and Subgradient Method

Proceeding in the same line as in the previous section, we form the following Lagrangian

$$\begin{aligned} L(\mathbf{x}, \mathbf{P}, \mathbf{S}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \gamma_1 \sum_{i \in V} U(S_i) - \gamma_2 \sum_{(i,j) \in E} P(i,j) - \epsilon \sum_{(i,j) \in E} x(i,j)^2 \\ &\quad - \sum_{(i,j) \in E} \lambda_{(i,j)} \left[ x(i,j) - r(i,j) \right] \\ &\qquad\qquad - \sum_{i \in V} \mu_i \left[ \sum_{j \in I(i)} x(j,i) - \sum_{j \in O(i)} x(i,j) + S_i \right] \\ &= \left\{ \sum_{i \in V} \gamma_1 U_i(S_i) - \mu_i S_i \right\} + \left\{ \sum_{(i,j) \in E} \lambda_{(i,j)} r(i,j) - \gamma_2 \sum_{(i,j) \in E} P(i,j) \right\} \\ &\quad + \left\{ -\epsilon \sum_{(i,j) \in E} x(i,j)^2 - \sum_{(i,j) \in E} \lambda_{(i,j)} x(i,j) \right. \\ &\qquad\qquad \left. + \sum_{i \in V} \mu_i \left[ \sum_{j \in O(i)} x(i,j) - \sum_{j \in I(i)} x(j,i) \right] \right\} \end{aligned} \tag{7.31}$$

where besides the flow rate vector $\mathbf{x}$ and the allocated power vector $\mathbf{P}$, the source rate vector $\mathbf{S}$ is introduced into the Lagrangian. The dual function is given by

$$D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sup_{\{\mathbf{x}, \mathbf{P}, \mathbf{S}\}} L(\mathbf{x}, \mathbf{P}, \mathbf{S}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{7.32}$$

This dual function can be calculated by decomposing the optimization problem in (7.32) into the following subproblems:

$$D_{\text{con}}(\boldsymbol{\mu}) = \max_{\mathbf{S}\succeq\mathbf{0}} \left\{ \sum_{i\in V} \gamma_1 U_i(S_i) - \mu_i S_i \right\} = \sum_{i\in V} \max_{S_i\geq 0} \{\gamma_1 U_i(S_i) - \mu_i S_i\} \tag{7.33}$$

$$\begin{aligned} D_{\text{net}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \max_{\mathbf{x}\succeq\mathbf{0}} \left\{ -\epsilon \sum_{(i,j)\in E} x(i,j)^2 - \sum_{(i,j)\in E} \lambda_{(i,j)} x(i,j) \right. \\ &\qquad \left. + \sum_{i\in V} \mu_i \left[ \sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i) \right] \right\} \\ &= \sum_{(i,j)\in E} \max_{x_{(i,j)}\geq 0} \left\{ -\epsilon x(i,j)^2 - \lambda_{(i,j)} x(i,j) + \mu_i x(i,j) - \mu_j x(i,j) \right\} \end{aligned} \tag{7.34}$$

$$\begin{aligned} D_{\text{phy}}(\boldsymbol{\lambda}) &= \max_{\mathbf{P}} \left\{ \sum_{(i,j)\in E} \lambda_{(i,j)} r(i,j) - \gamma_2 \sum_{(i,j)\in E} P(i,j) \right\} \\ &= \sum_{(i,j)\in E} \max_{\mathbf{P}\succeq\mathbf{0}} \left\{ \lambda_{(i,j)} r(i,j) - \gamma_2 P_t(i,j) \right\} \end{aligned} \tag{7.35}$$

and the dual function can be rewritten as $D(\boldsymbol{\lambda}, \boldsymbol{\mu}) = D_{\text{con}}(\boldsymbol{\mu}) + D_{\text{net}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) + D_{\text{phy}}(\boldsymbol{\lambda})$. Again, the optimization problem defined in (7.30) is a convex one where the strong duality holds, and therefore, the duality gap is zero. Thus, solutions of the primal problem in (7.30) can be recovered via its dual problem which can be written as $\min_{\boldsymbol{\lambda}\succeq\mathbf{0}} D(\boldsymbol{\lambda}, \boldsymbol{\mu})$. We solve the dual problem using the subgradient projection method. The subgradients of the dual function at $\lambda_{(i,j)}$ and $\mu_i$ are given in the following Lemma.
**Lemma 7.2:** The subgradient of $D(\lambda, \mu)$ can be shown to be

$$f_2(\lambda_{(i,j)}) = r(i,j) - x(i,j) \tag{7.36}$$

$$g_2(\mu_i) = \sum_{j\in O(i)} x(i,j) - \sum_{j\in I(i)} x(j,i) - S_i. \tag{7.37}$$

Given the locally optimal solutions $\mathbf{x}^*(t)$ and $\mathbf{P}^*(t)$ for the networking and the physical layer subproblems, the algorithm updates $\lambda_{(i,j)}$ and $\mu(i)$ as follows:

$$\lambda_{(i,j)}(t+1) = \left[\lambda_{(i,j)}(t) - \beta(t) f_2(\lambda_{(i,j)})\right]^+ \tag{7.38}$$

$$\mu_i(t+1) = \mu_i(t) - \beta(t) g_2(\mu_i). \tag{7.39}$$

Similar to the algorithm presented in Section 7.2, the algorithm for the optimization problem in (7.30) iteratively updates the dual variables (i.e., using (7.38) and (7.39)) and solves the congestion control subproblem in (7.33), the networking subproblem in (7.34), and the physical layer subproblem in (7.35) until the globally optimal solutions are achieved. We present how to solve these subproblems in the following subsection.

### 7.3.2 Solutions of Congestion Control, Networking, and Physical Subproblems

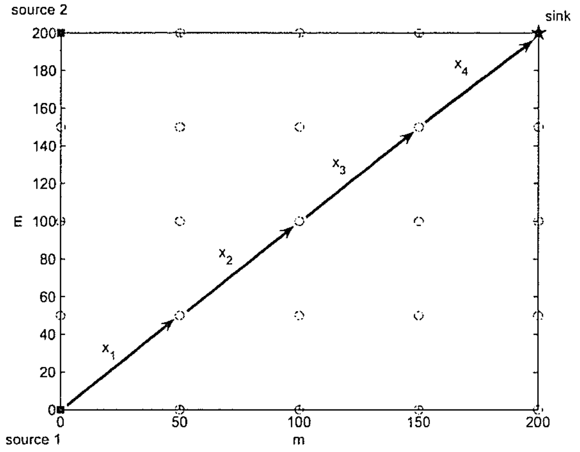


**Figure 7.2.** *Grid topology for 25 nodes with a routing path from a source node to the destination node.*

The congestion control subproblem in (7.33) and networking subproblem in (7.34) can be decomposed into multiple node and link subproblems, respectively. Given the node and link prices $\mu_i$, $\lambda_{(i,j)}$, the optimal solutions for these subproblems can be written as

$$S_i^* = \left[ U_i'^{-1}\left(\frac{\mu_i}{\gamma_1}\right)\right]^+ \quad \text{and} \quad x^*(i,j) = \left[\frac{\mu_i - \mu_j - \lambda_{(i,j)}}{2\epsilon}\right]^+ \tag{7.40}$$

where $U_i'^{-1}$ is the inverse function of the derivative of utility function $U_i$.

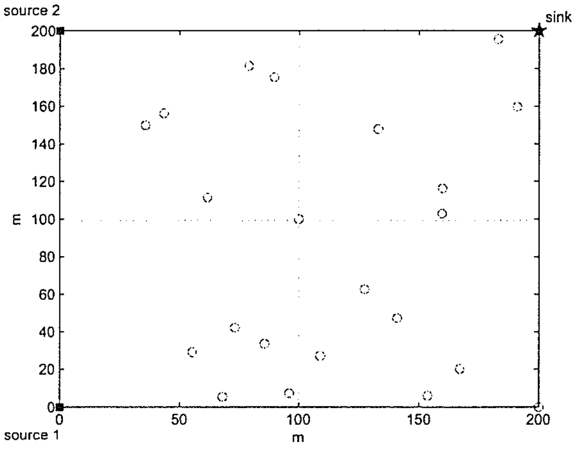


**Figure 7.3.** *Random topology for 25 nodes.*

As in the previous section, the physical layer subproblem involves both relay selection and power allocation. The optimization problem in (7.35) can be decomposed into multiple link subproblems, where each link subproblem searches for the best relay and the corresponding allocated power. The optimal amount of power allocated, if direct transmission is pursued, can be written as

$$P_d^*(i,j) = \left[\frac{\lambda_{(i,j)}}{\gamma_2 \log 2} - \frac{1}{g(i,j)}\right]_{P_{\min}(i,j)}^{P_{\max}(i,j)}. \tag{7.41}$$

For cooperative transmission through relay node $k$, the optimal allocated powers can be written as

$$P_d^*(i,j) = \left[\frac{\lambda_{(i,j)} g(k,j)}{\gamma_2 \left(g(i,k) + g(k,j) - g(i,j)\right) \log 2} - \frac{1}{g(i,k)}\right]_{P_{\min}(i,j)}^{P_{\max}(i,j)} \tag{7.42}$$

$$P_{r,(i,j)}^*(k,j) = \left[\frac{g(i,k) - g(i,j)}{g(k,j)} P_d^*(i,j)\right]_{P_{\min}(k,j)}^{P_{\max}(k,j)}. \tag{7.43}$$

Given the link price $\lambda_{(i,j)}$, channel gains, and interference plus noise powers, the strategy corresponding to the largest

$$D_{\text{phy}}(i,j) = \max_{\mathbf{P} \succeq \mathbf{0}} \left\{ \lambda_{(i,j)} r^*(i,j) - \gamma_2 \sum_{(i,j) \in L} P_t^*(i,j) \right\}$$

will be chosen as the best strategy and the optimal solutions for this strategy are used to update the subgradients in (7.38). The joint congestion control, routing, and cooperative resource allocation algorithm to solve (7.30) is summarized in **Algorithm 7.2**. As before, the convergence of this algorithm under the *non-summable diminishing* stepsize condition can be found in [119].

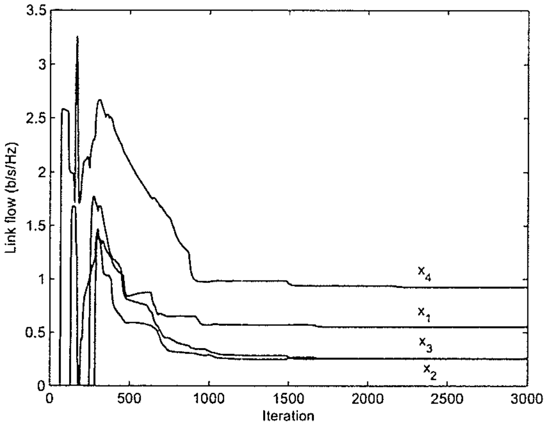


**Figure 7.4.** *Link flows for a routing path from source node one to the destination node with cooperative diversity for the grid topology (for $S_1 = S_2 = 4$ b/s/Hz).*

**Algorithm 7.2: Joint Congestion Control, Routing and Cooperative Resource Allocation for Utility-Power Tradeoff**

1. Each node $i$ initializes its node price $\mu_i(0)$ and link price $\lambda_{(i,j)}(0)$, link flow $x(i,j)(0)$, transmission power $P_d(i,j)(0)$ for each of the outgoing links $(i,j)$.
2. Given $\lambda_{(i,j)}(t)$ and $\mu_i(t)$, each node $i$ solves the congestion control problem to obtain the locally optimal source rate $S_i$, solves the networking (i.e., routing), and physical subproblems for its outgoing link $(i,j)$ to obtain the locally optimal link flow $x^*(i,j)(t)$, transmission strategy, allocated power for itself $P_d^*(i,j)(t)$ and power for its relay partner $P^*_{r,(i,j)}(k,j)(t)$ if cooperative transmission through node $k$ is the best strategy. Node $i$ then transmits the link flow value $x^*(i,j)(t)$ to node $j$ and relay power $P^*_{r,(i,j)}(k,j)(t)$ to its relay partner $k$.
3. Given the locally optimal source rate, link flow, transmission strategy, and allocated powers, each node $i$ updates the link and node prices $\lambda_{(i,j)}(t+1)$,

$\mu_i(t+1)$ as in (7.38) and (7.39). Node $i$ transmits $\mu_i(t)$ to all neighboring nodes $j$.

4. Return to 2) until the algorithm converges.

The presented algorithm for the utility-power tradeoff is also fully distributed. Besides routing and power allocation, which are performed by using local information only, the congestion control solution in (7.40) for each node $i$ requires only node price $\mu_i$ of itself which is immediately available.

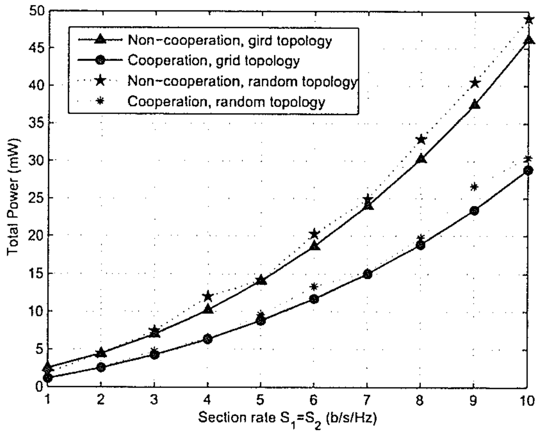


**Figure 7.5.** *Total power consumption versus source rate with and without cooperative diversity.*

## 7.4 Numerical Results

We consider a wireless network with 25 nodes distributed in an area of 200m × 200m. We investigate two topologies, namely, the grid topology (Fig. 7.2) and the random topology (Fig. 7.3). For the random topology, we fix 4 nodes at the four corners and one node at the center of the area; the other 20 nodes are positioned randomly with 5 nodes in each area of 100m × 100m as indicated in Fig. 7.3. There are two source nodes generating data to the single destination node as shown in Figs. 7.2-7.3. To show the convergence of the proposed algorithms, we show the link flows for a routing path as indicated in Fig. 7.2. To limit the number of links, we assume that a link

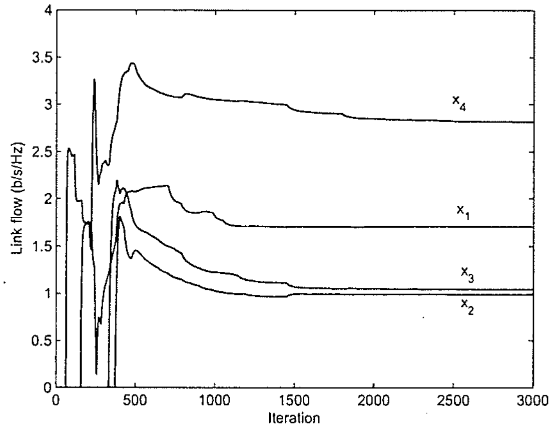


**Figure 7.6.** *Link flow with utility and power tradeoff optimization with cooperative diversity for $\gamma_2 = 1$ and $\gamma_1 = 100$.*

exists between a pair of nodes only if the distance between them is less than 150m. The power limits on each link $(i, j)$ are $P_{\min}(i, j) = 0$ mW, $P_{\max}(i, j) = 50$ mW for transmissions from both source nodes and relay nodes.

In order to obtain the numerical results, we run the proposed algorithms with stepsize chosen to be $\beta(t) = \max(0.2/\sqrt{t}, 0.001)$. The initial value of $\epsilon$ is chosen to be 0.1. Then it decreases exponentially until it reaches a certain value (e.g., $\epsilon_0 = 0.05$) after which it remains the same. There is a tradeoff between the convergence speed and the accuracy of the achieved solution: the higher the $\epsilon_0$, the faster the convergence but the less accurate the achieved results are. The channel gain for any link $(i, j)$ is modeled as $g(i, j) = K_0.d(i, j)^{-3}$, where $K_0 = 10^6$ and $d(i, j)$ is the distance between node $i$ and $j$ in meter. Note that we have absorbed $GN_0(i, j)$ into these channel gains. The utility function is chosen to be $U_i(S_i) = \log(S_i)$ for the two sources. This utility function provides the proportional fairness for the source rates. We compare the performance for networks with and without cooperative diversity implementation. When cooperative diversity is not employed, direct transmission on each link is always chosen without the cooperation of any relay nodes.

The link flows for the routing path indicated in Fig. 7.2 are shown in Fig. 7.4. This figure shows the convergence of **Algorithm 7.1**. The optimal flows show that data from both sources one and two are actually split into multiple paths to reach the

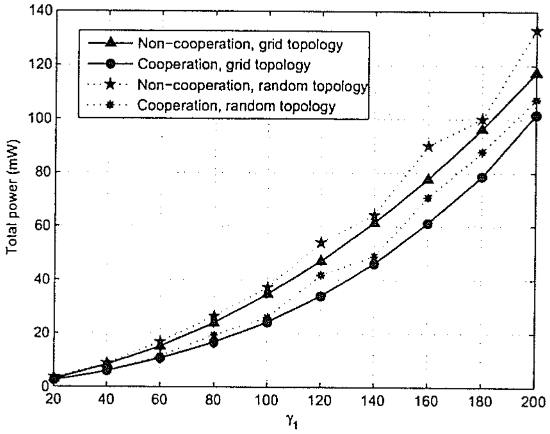


**Figure 7.7.** *Total power consumption versus $\gamma_1$ for utility-power tradeoff optimization.*

destination node (because the link flows along this routing path are smaller than the source rate which is 4 b/s/Hz). This implies that a cross-layer framework is necessary to obtain the optimal multipath routing solution. The proposed algorithm actually exploits both cooperative diversity gain from the physical layer and network topology to perform load balancing in such a way that the globally optimal solution can be achieved.

The total power consumed versus the source rate for both network topologies with and without cooperative diversity is shown in Fig. 7.5 (denoted by "cooperation" and "non-cooperation" in this figure, respectively). For the random topology, the total power is obtained by averaging over 20 simulation runs. This figure shows that the random topology actually requires higher power consumption than the grid topology for these particular source-sink pairs. The performance gain resulting from cooperative diversity is very significant which is about 40% for both the topologies. This performance gain is achieved without sacrificing the distributed nature of the proposed algorithm because only local information is needed to search for the best relay together with the optimal allocated power and routing solution.

To obtain the numerical results for **Algorithm 7.2**, we fix $\gamma_2 = 1$ and vary $\gamma_1$ to achieve the solutions for different utility-power tradeoff. The link flows for the routing path from source node one to the sink node are shown in Fig. 7.6. This

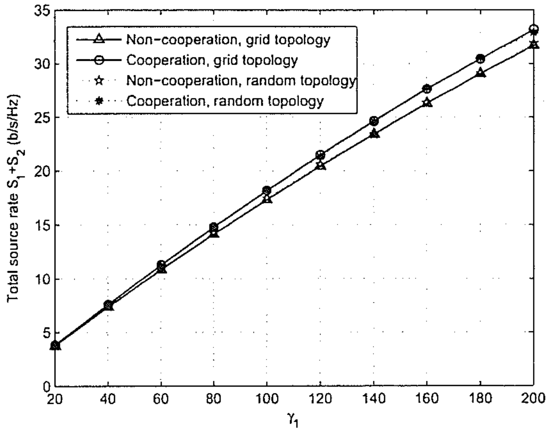


**Figure 7.8.** *Total source rate versus $\gamma_1$ for utility-power tradeoff optimization.*

figure conforms the convergence of **Algorithm 7.2**. The total power consumption for utility-power tradeoff optimization is shown in Fig. 7.7 for both the topologies with and without cooperative diversity implementation. Again cooperative diversity results in a performance gain although the power gain is smaller compared to that obtained in Fig. 7.5. This observation can be explained by noticing that the optimization of utility-power tradeoff results in higher total source rate for the cooperation case (Fig. 7.8). This implies that the performance gain from cooperative diversity is balanced between power and rate gains.

## 7.5 Chapter Summary

We have developed cross-layer design frameworks for power efficient communications in multi-hop wireless networks using cooperative diversity. The proposed distributed algorithms converge to the globally optimal solution where the best relay (if any) for each link and the allocated power in the physical layer and the optimal routing solution can be found in a distributed manner. Cross-layer optimization frameworks have been developed to either minimize the total power consumption or to maximize the utility-power tradeoff. For the latter case, optimal source rates are found to balance performance gain for both power consumption and achieved source rates.

The proposed algorithms are very appealing to achieve the cooperative diversity gain which has been shown to be very significant in this chapter.

# Chapter 8

# Summary and Directions for Future Research

## 8.1 Summary

In this dissertation, several analytical models and protocols have been developed for wireless networks considering realistic physical and link layer designs as being adopted by current wireless standards. In particular, link adaptation techniques in the physical layer and ARQ-based error recovery in the link layer have been taken in account in different analytical models. Both single-hop radio link level design problems and end-to-end research issues have been considered. This dissertation, therefore, tackles important design challenges for widespread-deployed centralized networks, namely cellular and WLAN networks as well as for future distributed networks such as multi-hop cellular and wireless mesh networks.

Analytical models to analyze delay statistics for Go-Back-N and Selective-Repeat ARQ protocols have been proposed. While the existing analytical models for ARQ protocols in the literature assumed either i.i.d. or Markov channels with only two states (i.e., good and bad states), we have assumed an FSMC channel model which captures the multi-rate transmission effects due to the implementation of AMC technique in the physical layer. The proposed models also assumed non-zero feedback delay as in wireless networks with large signal processing delay. Under these realistic assumptions, modeling the evolutions of the two ARQ protocols is a very challenging task. We have derived the exact queue length and delay distributions for these protocols considering variable arrival traffic. The models enable us to quantify impacts of different traffic, system, channel parameters on the link level delay performance.

They also provide useful frameworks to conduct cross-layer design such as to find SNR thresholds for different transmission modes such that good link level delay performance can be achieved.

For the case where multiple users share one channel in the time multiplexing manner, we consider two main classes of scheduling schemes to allocate the transmission opportunities to the users, namely weighted round robin and channel-quality-based opportunistic scheduling schemes. In essence, these two scheduling classes are the two extreme cases considering the tradeoff between throughput and fairness. Specifically, weighted round robin scheduling scheme can be used for QoS differentiation among different users with predetermined fairness. Opportunistic scheduling schemes, however, exploits the multi-user diversity to enhance the network throughput. Existing works on wireless scheduling issues mainly concentrate on developing the scheduling rules assuming saturated buffer conditions (i.e., all users always have packets to transmit). We, however, analyze the delay statistics and throughput performances for these two scheduling classes under random arrival traffic process with AMC implementation in the physical layer and ARQ in the link layer.

The model for the opportunistic scheduling can be applied to any scheduling policy as long as the joint evolution of service/vacation and channel processes can be determined. We have applied this model to analyze the max-rate scheduling scheme as an example. In general, the proposed analytical models for these two scheduling classes provide unified frameworks to compare the performances of different scheduling policies under different network settings. It is also very useful for performing connection and packet levels admission control under statistical delay constraints. In fact, it has been shown via numerical results that although the max-rate scheduling scheme results in higher throughput than the round-robin counterpart, the round robin scheme offers better delay performance than the max-rate scheme under light traffic load conditions.

For the multi-hop transmission scenario, we have proposed a tandem queue model and a QoS routing framework which take all important QoS requirements, namely, end-to-end bandwidth, average delay, loss rate and statistical delay constraints, into account. In fact, both exact and approximate tandem queueing models considering link adaptation and ARQ error protection have been proposed. Due to the "distributed" nature of the decomposition tandem queueing approach, it can be employed to construct a QoS routing algorithm. Using the framework, the proposed routing algorithm can efficiently remove the infeasible routing paths which would reduce communication overhead and connection setup time significantly. The framework also allows end-to-end loss rate and statistical delay constraints to be considered while these QoS constraints have usually been ignored by the existing protocols in the literature.

Finally, cross-layer optimization frameworks have been proposed for multi-hop wireless networks using cooperative diversity. Specifically, multipath routing protocols have been assumed where data traffic from source nodes is split into several flows following different multi-hop routes to reach the destinations. Cooperative transmission scenario has been considered where a relay node cooperates transmission on each wireless link using a decode-and-forward scheme. Two distributed algorithms have been developed by using the dual decomposition approach from convex optimization. The first algorithm, in essence, is the joint optimal routing, relay selection and power allocation which minimizes the total network power consumption. The second one strikes a balance between the optimal rate utility and transmission power with cooperative resource allocation. Numerical results have shown the convergence of the proposed algorithms and significant gains in terms of power consumption and/or transmission rates due to the implementation of cooperative diversity in the physical layer.

## 8.2 Future Research Directions

The distributed nature of the evolving wireless networks along with the opportunities for dynamic spectrum sharing give rise to several interesting research challenges. The cross-layer design paradigm will be useful to maximize the performance gain in the evolving wireless networks. My future research aims at answering the following questions:

- **How to achieve the provably optimal throughput/capacity for wireless networks with limited number of nodes and arbitrary topology?** This problem needs to be solved considering different resource and QoS constraints such as power and bandwidth constraints, loss and delay constraints, fairness constraint for different communication sections.

- **How to schedule the transmission of different wireless links considering the resource constraints and QoS requirements?** An interesting research question is whether graph-theoretic approach could be used for interference modeling of simultaneous transmissions on different links in the network. Also, the problem of finding the joint optimal congestion control, routing, scheduling, power control for wireless networks, where transmission rates on each wireless link is adapted based on its signal to noise plus interference ratio, is worth investigating.
- **How to find provably optimal cross-layer frameworks for the case where multiple orthogonal channels are available and each wireless node is equipped with multiple radios?** Although there are several solutions in the literature for this cross-layer design problem, they are mainly heuristics-based and their performances are still far from optimality. Developing a provably optimal cross-layer framework which maximizes throughput/capacity of wireless networks is an interesting research topic.
- **How to solve the emerging research challenges in cognitive radio wireless networks?** For cognitive radio networks (i.e., dynamic spectrum access networks), research challenges exist in different layers of the protocol stack. In the physical layer, developing efficient spectrum sensing techniques is important to detect spectrum holes and intelligent physical design using software-defined radio technologies is expected to play a more important role. Constructing an optimal spectrum allocation strategy which optimizes the system performance is the key to successful implementation of cognitive radio. Here, optimization and game theory are important tools. Also, because of distributed nature of future wireless networks, research issues related to routing and congestion control together with resource allocation and physical layer design issues continue to be important research challenges to tackle.

# Appendix A

# Derivations of Matrix Blocks in (3.2)

We derive the matrix blocks for the transition matrix (3.2) in this Appendix. The number of packets transmitted in time slot $t$ is the minimum of the number of packets available in the queue and the transmission capability (i.e., equal to $\min\left\{q(t), h_{c(t)}\right\}$). Let $a(t)$ be the number of arriving packets during slot $t$, and $d(t)$ be the number of packets which will not be retransmitted due to transmissions in slot $t$ (in fact, $d(t) \leq \min\left\{q(t), h_{c(t)}\right\}$), we have $q(t+1) = q(t) + a(t) - d(t)$.

To derive the matrix blocks $\mathbf{D}_{i,l}$ and $\mathbf{D}_l$ in (3.2), we consider the following cases which may occur in each time slot. First, the slot is not useful (i.e., $s(t) \neq 0$), and therefore, no packet can depart. We need to keep track of the channel state evolution only for this case. Second, the slot is useful (i.e., $s(t) = 0$) and all transmitted packets are received correctly. In this case, the number of packets in the queue changes according to the number of successfully transmitted packets and the number of arriving packets in that slot. Also, the next time slot will be a useful one (i.e., $s(t+1) = 0$). Third, the slot is useful (i.e., $s(t) = 0$) and there exists at least one packet among those transmitted in error. The number of packets in the queue at the end of the slot depends on the error pattern and the number of arriving packets in that slot. However, the next time slot will not be useful (in fact, $s(t+1) = n - 1$).

Now let us define the following matrices:

- $\mathbf{T}_k$ ($k = 0, 1, \cdots, K$) are constructed by keeping only the $(k+1)$-st row of the channel transition probability matrix $\mathbf{T}$ and setting all other rows to $\mathbf{0}$. These matrices capture the case the channel is in state $k$ at the beginning of a particular time slot.

- $\Psi_{i,j}^{(0)}$ are matrices of order $(K+1)\times(K+1)$ in which element $\Psi_{i,j}^{(0)}(k,k')$ is the probability that all $i$ transmitted packets are received correctly given that there were $j$ packets in the queue before transmission (i.e., $i = \min\{q(t), h_{c(t)}\} = \min\{j, h_{c(t)}\}$), the channel changes from state $k$ to state $k'$ in the transmission slot.
- $\Psi_{i,j}^{(1)}$ are matrices of order $(K+1)\times(K+1)$ whose element $\Psi_{i,j}^{(1)}(k,k')$ is the probability that $i$ transmitted packets are received correctly given that there were $j$ packets in the queue before transmission, there are at least one erroneous packet (i.e., $i < \min\{q(t), h_{c(t)}\} = \min\{j, h_{c(t)}\}$), and the channel changes from state $k$ to state $k'$ in the transmission slot.
- $\mathbf{C}_{i,j}^{(k)}$ $(k = 1,2,3)$ are matrices of order $n(K+1)\times n(K+1)$ representing the aforementioned three cases, respectively, whose element $\mathbf{C}_{i,j}^{(k)}(l,l')(h,h')$ $(0 \le l,l' \le n-1,\ 0 \le h,h' \le K)$ represents the probability of system transition $(j,l,h) \to (j-i,l',h')$ (i.e., $i$ packets are successfully transmitted given there were $j$ in the queue before transmissions).

From foregoing definitions, $\mathbf{C}_{i,j}^{(k)}$ can be written as follows:

$$\mathbf{C}_{i,j}^{(1)} = \mathbf{0} \quad \text{if } i \neq 0 \tag{A.1}$$

$$\mathbf{C}_{0,j}^{(1)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{T} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{T} & \mathbf{0} \end{bmatrix} \tag{A.2}$$

$$\mathbf{C}_{i,j}^{(2)} = \begin{bmatrix} \Psi_{i,j}^{(0)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.3}$$

$$\mathbf{C}_{i,j}^{(3)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Psi_{i,j}^{(1)} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{A.4}$$

Equation (A.2) simply captures the channel state transitions where $s(t) \neq 0$. Note that, $s(t)$ decreases by one in each time slot which explains the structure of $\mathbf{C}^{(1)}_{0,j}$ (i.e., $\mathbf{T}$ is in positions $(s, s-1)$). For $s(t) \neq 0$, no packet can depart; therefore, we have $\mathbf{C}^{(1)}_{i,j} = \mathbf{0}$ for $i \neq 0$. In (A.3), $\mathbf{C}^{(2)}_{i,j}$ contains the probabilities of transition between two useful slots (i.e., $s(t) = s(t+1) = 0$), and $\mathbf{C}^{(3)}_{i,j}$ contains the probabilities of transition from a useful time slot to a useless one (i.e., $s(t) = 0$ and $s(t+1) = n-1$), where at least one transmission error must have occurred. This explains the position of $\mathbf{\Psi}^{(0)}_{i,j}$ and $\mathbf{\Psi}^{(1)}_{i,j}$ in the matrices $\mathbf{C}^{(2)}_{i,j}$ and $\mathbf{C}^{(3)}_{i,j}$, respectively.

Before we derive the matrix blocks in (3.2), let $\mathbf{C}_{i,j} = \sum_{k=1}^{k=3} \mathbf{C}^{(k)}_{i,j}$, which contains the probabilities that $i$ packets are successfully transmitted given that there were $j$ packets in the queue before transmission without distinguishing the aforementioned three cases. As we discussed above, if $d(t)$ is the number of packets which will not be retransmitted due to transmissions in slot $t$, we have $q(t+1) = q(t) + a(t) - d(t)$. It can be easily seen that $\mathbf{D}_{i,l}$ and $\mathbf{D}_l$ contains the probabilities of system state transition where $q(t+1) = q(t) + 1 - l$. Thus, we have $a(t) - d(t) = 1 - l$. As a result, $d(t) = l - 1$ if $a(t) = 0$ (i.e., no arrival) and $d(t) = l$ if $a(t) = 1$ (i.e., one arrival). Therefore, $\mathbf{D}_{i,l}$ can be calculated as follows:

$$\mathbf{D}_{i,l} = (1-\lambda)\mathbf{C}_{l-1,i} + \lambda \mathbf{C}_{l,i}. \tag{A.5}$$

Similarly, $\mathbf{D}_l$ can be calculated as

$$\mathbf{D}_{l} = (1-\lambda)\mathbf{C}_{l-1,L} + \lambda \mathbf{C}_{l,N}. \tag{A.6}$$

It can be easily observed that $\mathbf{C}_{i,j} = \mathbf{C}_{i,N}$ for $j \geq N$.

The remaining task is to determine $\mathbf{\Psi}^{(0)}_{i,j}$ and $\mathbf{\Psi}^{(1)}_{i,j}$, which is pursued now. Let $\theta_k = \overline{\mathrm{PER}}_k$ be the probability of transmission error when the channel is in state $k$. Assuming that the transmission outcomes of different packets are independent and let us define

$$\eta^{(k)}_i = (1-\theta_k)^i. \tag{A.7}$$

Then, for $i, j > 0$ $\mathbf{\Psi}^{(0)}_{i,j}$ can be calculated as

$$\mathbf{\Psi}^{(0)}_{i,j} = \begin{cases} \eta^{(k)}_i \mathbf{T}_k, & i < j,\ i = h_k \\ \mathbf{0}, & i < j,\ \text{if} \nexists!\ k \text{ s.t. } i = h_k \\ \sum_{h=k}^{h=K} \eta^{(h)}_i \mathbf{T}_h, & i = j,\ k = \min\{c(t) : h_k \geq j\} \end{cases} \tag{A.8}$$

For $i = 0$, $\mathbf{\Psi}_{i,j}^{(0)}$ can be calculated as

$$\mathbf{\Psi}_{0,0}^{(0)} = \mathbf{T}, \quad \mathbf{\Psi}_{0,j}^{(0)} = \mathbf{T}_0, \quad j > 0. \tag{A.9}$$

Also, $\mathbf{\Psi}_{i,j}^{(1)}$ can be calculated as

$$\mathbf{\Psi}_{i,j}^{(1)} = \sum_{m=k}^{K} \eta_i^{(m)} \theta_m \mathbf{T}_m \tag{A.10}$$

where $k = \min\{c(t) : h_k > i\}$.

Equations (A.8)-(A.10) can be interpreted as follows. In (A.8), we must have $i = \min\{j, h_{c(t)}\}$; therefore, $h_{c(t)} = i$ if $j > i$ or $h_{c(t)} \geq i$ if $i = j$. For $\mathbf{\Psi}_{0,0}^{(0)}$, there is no transmission since there is no packet in the queue at the beginning of the slot. Therefore, the channel can be in any state without introducing any transmission error. For $\mathbf{\Psi}_{0,j}^{(0)}$, there are $j > 0$ packets in the queue in this case, and no transmission error occurs only if the channel is in state 0 (since no transmission is allowed in this channel state). The interpretation for (A.10) is similar but at least one packet among those transmitted must be in error (i.e., $i < \min\{j, h_{c(t)}\}$). Note that in this case the first $i$ packets are received correctly and the $(i+1)$-st packet must be in error.

# Appendix B

# Derivations of Matrix Blocks in (3.7)

Due to the structure of the probability transition matrix, we can write $\mathbf{A}_l$ and $\mathbf{A}_{i,l}$ as

$$\mathbf{A}_l = \begin{bmatrix} \mathbf{A}_l(0,0) & \mathbf{A}_l(0,1) & \cdots & \cdots & \mathbf{A}_l(0,N_2) \\ \mathbf{A}_l(1,0) & \mathbf{A}_l(1,1) & \cdots & \cdots & \mathbf{A}_l(1,N_2) \\ & \cdots & \cdots & \cdots & \\ \mathbf{A}_l(N_2,0) & \mathbf{A}_l(N_2,1) & \cdots & \cdots & \mathbf{A}_l(N_2,N_2) \end{bmatrix} \tag{B.1}$$

$$\mathbf{A}_{i,l} = \begin{bmatrix} \mathbf{A}_{i,l}(0,0) & \mathbf{A}_{i,l}(0,1) & \cdots & \cdots & \mathbf{A}_{i,l}(0,N_2) \\ \mathbf{A}_{i,l}(1,0) & \mathbf{A}_{i,l}(1,1) & \cdots & \cdots & \mathbf{A}_{i,l}(1,N_2) \\ & \cdots & \cdots & \cdots & \\ \mathbf{A}_{i,l}(N_2,0) & \mathbf{A}_{i,l}(N_2,1) & \cdots & \cdots & \mathbf{A}_{i,l}(N_2,N_2) \end{bmatrix} \tag{B.2}$$

where we have defined $N_2 = (N+1)^n - 1$, sub-matrices $\mathbf{A}_{i,l}(j,j')$ contains the probabilities of system transitions $(i,j,*) \to (i+N+1-l, j', *)$ and $\mathbf{A}_l(j,j')$ contains the probabilities of the same system transitions for $i \geq L$.

If $\theta_k = \overline{\mathrm{PER}}_k$ is the probability of packet transmission error when the channel is in state $k$, the probability that $i$ packets are received in error given that $j$ packets were transmitted when the channel is in state $k$ can be written as follows:

$$q_{i,j}^{(k)} = \begin{pmatrix} j \\ i \end{pmatrix} \theta_k^i \left(1-\theta_k\right)^{j-i}. \tag{B.3}$$

Now we show how to calculate $\mathbf{A}_l(j,j')$. It can be checked that $\mathbf{A}_l$ contains the probabilities of system transitions where $q(t+1) - q(t) = L+1-l$. Thus we have $q(t+1) - q(t) = N+1-l = a(t) + b_n(t) - \min\left\{h_{c(t)}, q(t) + b_n(t)\right\} = a(t) + b_n(t) - h_{c(t)}$ since $\min\left\{h_{c(t)}, q(t) + b_n(t)\right\} = h_{c(t)}$. Therefore, $h_{c(t)} = a(t) + b_n(t) + l - N - 1$. For given $l, j, j'$, we can find $c(t) = c_1$ for $a(t) = 0$ (no arrival) and $c(t) = c_2$ for $a(t) = 1$ (one arrival) from this relation if they exist. Hence, we have

$$\mathbf{A}_l(j,j') = (1-\lambda) q^{(c_1)}_{\beta,c_1} \mathbf{T}_{c_1} + \lambda q^{(c_2)}_{\beta,c_2} \mathbf{T}_{c_2} \tag{B.4}$$

where $\mathbf{T}_k$ was defined in **Appendix A**; $b_n(t)$ and $\beta$ can be found from the mapping $\mathbf{b}(t) \to \mathbf{b}(t+1) \equiv \{y(t) = j\} \to \{y(t+1) = j'\}$, with $\mathbf{b}(t) = [b_1(t), b_2(t), \cdots, b_n(t)]$ and $\mathbf{b}(t+1) = [\beta, b_1(t), b_2(t), \cdots,\ b_{n-1}(t)]$.

Similarly, to calculate $\mathbf{A}_{i,l}(j,j')$ we find $c(t)$ from the relation $q(t+1) - q(t) = N+1-l = a(t)+b_n(t) - \min\left\{h_{c(t)}, q(t)+b_n(t)\right\} = a(t)+b_n(t) - \min\left\{h_{c(t)}, i+b_n(t)\right\}$. In this case, we may find more than one $c(t)$ from this relation for the case $a(t) = 0$ (no arrival) and $a(t) = 1$ (one arrival), which are denoted as $c_3$ and $c_4$ in the following sums for these two cases, respectively. Hence, we have

$$\mathbf{A}_{i,l}(j,j') = (1-\lambda) \sum_{c_3} q^{(c_3)}_{\beta,d_1} \mathbf{T}_{c_3} + \lambda \sum_{c_4} q^{(c_4)}_{\beta,d_2} \mathbf{T}_{c_4} \tag{B.5}$$

where $d_1 = \min\{h_{c_3}, i + b_n(t)\}$, $d_2 = \min\{h_{c_4}, i + b_n(t)\}$. Again, $b_n(t)$ and $\beta$ can be found from the mapping $\mathbf{b}(t) \to \mathbf{b}(t+1) \equiv \{y(t) = j\} \to \{y(t+1) = j'\}$.

# Appendix C

# Derivations of $\Theta_{(p,h)}$

We can rewrite $\Theta_{(p,h)}$ as follows:

$$\Theta_{(p,h)} = \begin{bmatrix} \Theta_{(p,h)}(0,0) & \Theta_{(p,h)}(0,1) & \cdots & \Theta_{(p,h)}(0,N) \\ \Theta_{(p,h)}(1,0) & \Theta_{(p,h)}(1,1) & \cdots & \Theta_{(p,h)}(1,N) \\ & \cdots & \cdots & \\ \Theta_{(p,h)}(N,0) & \Theta_{(p,h)}(N,1) & \cdots & \Theta_{(p,h)}(N,N) \end{bmatrix} \tag{C.1}$$

where $\left(\Theta_{(p,h)}\right)(j,j')$ contains the probabilities of system transitions $(p,j,*) \rightarrow (p-h,j',*)$.

For transitions whose probabilities are captured in $\Theta_{(p,h)}$, we have $q(t+1)-q(t) = -h = b_n(t) - \min\left\{h_{c(t)}, q(t)+b_n(t)\right\}$. Note that, packets arriving at the queue after the target arriving packet do not affect the delay experienced by the target packet; therefore, we let $a(t)=0$ in this relation. We may find more than one $c(t)$ from this relation which are denoted by $c_5$ in the following sum. Hence, we have

$$\Theta_{(p,h)}(j,j') = \sum_{c_5} q_{\beta,d_3}^{(c_5)} \mathbf{T}_{c_5} \tag{C.2}$$

where $d_3 = \min\{h_{c_5}, p+b_n(t)\}$; $b_n(t)$ and $\beta$ can be calculated from the mapping $\mathbf{b}(t) \rightarrow \mathbf{b}(t+1) \equiv \{y(t)=j\} \rightarrow \{y(t+1)=j'\}$.

# Appendix D

# Derivations of Matrix Blocks in (4.1)

In this Apppendix, we derive the matrix blocks for the transition matrix in (4.1). Let $\theta_k = \overline{\mathrm{PER}}_k$ be the probability of transmission failure when the channel is in state $k$. Assuming that the transmission outcomes of consecutive packets are independent, the probability that $i$ packets are correctly received given that $j$ packets were transmitted when the channel state is $k$ can be written as

$$p_{i,j}^{(k)} = \begin{pmatrix} j \\ i \end{pmatrix} \theta_k^{j-i} (1 - \theta_k)^i .$$

Let us define the following matrices:

- $\mathbf{\Lambda}_{i,k}$ are matrices of order $(K+1) \times (K+1)$ whose elements $(\mathbf{\Lambda}_{i,k})(h_1, h_2)$ represent the probability that $k$ packets are successfully transmitted in a particular service slot given that there were $i$ packets in the queue before transmission and the channel changes from state $h_1$ to state $h_2$.
- $\mathbf{H}_{i,k}^{(j)}$ $(j = 1, 2)$ are matrices of order $(K+1)L \times (K+1)L$ whose elements $\left(\mathbf{H}_{i,k}^{(j)}\right)(l_1, l_2), (h_1, h_2)$ represent the probability that $k$ packets are successfully transmitted given that there were $i$ packets in the queue before transmission, the channel changes from state $h_1$ to state $h_2$ during the evolution from slot $l_1$ to slot $l_2$ of a cycle for a class-$j$ user .
- $\mathbf{H}_v^{(j)}$ $(j = 1, 2)$ are matrices of order $(K+1)L \times (K+1)L$ which have the same structure as $\mathbf{H}_{i,k}^{(j)}$ and capture the time slot and the channel evolution in the vacation slot(s) for a class-$j$ user.

We can calculate $\mathbf{\Lambda}_{i,k}$ as follows:

$$\mathbf{\Lambda}_{i,0} = \mathbf{T}_0 + \sum_{l=1}^{K} p_{0,\alpha_l}^{(l)} \mathbf{T}_l, \tag{D.1}$$

$$\mathbf{\Lambda}_{i,k} = \sum_{l=1}^{K} p_{k,\alpha_l}^{(l)} \mathbf{T}_l, \ k > 0, \tag{D.2}$$

where $\alpha_l = \min\{h_l, i\}$ and $p_{k,\alpha_l}^{(l)} = 0$ if $k > \alpha_l$. The evolution of the channel state and the time slots in a cycle is captured by matrix $\mathbf{C}$ as follows:

$$\mathbf{C} = \begin{bmatrix} 0 & \mathrm{T} & 0 & 0 & \dots & 0 \\ 0 & 0 & \mathrm{T} & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathrm{T} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathrm{T} \\ \mathrm{T} & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{D.3}$$

We can write $\mathbf{H}_v^{(j)}$ as follows:

$$\mathbf{H}_v^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathrm{T} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathrm{T} \\ \mathrm{T} & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{D.4}$$

$$\mathbf{H}_v^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \mathrm{T} & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathrm{T} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathrm{T} \\ \mathrm{T} & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{D.5}$$

Let $\mathbf{R}_{i,k}^{(j)}$ be defined as

$$\mathbf{R}_{i,k}^{(1)} = \begin{bmatrix} 0 & \mathbf{\Lambda}_{i,k} & 0 & 0 & \dots & 0 \\ 0 & 0 & \mathbf{\Lambda}_{i,k} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{D.6}$$

$$\mathbf{R}^{(2)}_{i,k} = \begin{bmatrix} 0 & \mathbf{\Lambda}_{i,k} & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{D.7}$$

Then, we have

$$\mathbf{H}^{(j)}_{i,k} = \begin{cases} \mathbf{C}\,, & i = k = 0 \\ \mathbf{R}^{(j)}_{i,k}\,, & i, k > 0 \\ \mathbf{H}^{(j)}_{v} + \mathbf{R}^{(j)}_{i,k}\,, & i > 0 \text{ and } k = 0, \end{cases} \tag{D.8}$$

where $j = 1, 2$. Each of these matrices has $L$ blocks of rows. Each block of rows captures the evolution of the channel states in a particular time slot of a cycle. Now, the matrix blocks of the transition matrix in (4.1) can be written as follows:

$$\mathbf{A}_{0,k} = \mathbf{U}_{M-k} \otimes \mathbf{C}, \quad 0 \leq k \leq M, \tag{D.9}$$

$$\mathbf{A}_{k} = \sum_{h=0}^{M} \mathbf{U}_{h} \otimes \mathbf{H}^{(j)}_{N,k-M+h}, \quad 0 \leq k \leq M + N, \tag{D.10}$$

$$\mathbf{A}_{i,k} = \sum_{h=0}^{M} \mathbf{U}_{h} \otimes \mathbf{H}^{(j)}_{i,k-M+h},\ 1 \leq i < N,\ 0 \leq k \leq i + M, \tag{D.11}$$

where $\otimes$ denotes Kronecker product.

# Appendix E

# Derivations of Matrix Blocks in (5.3)

We have defined $\mathbf{H}_{i,k} = \begin{bmatrix} (\mathbf{H}_{i,k})(0,0) & (\mathbf{H}_{i,k})(0,1) \\ (\mathbf{H}_{i,k})(1,0) & (\mathbf{H}_{i,k})(1,1) \end{bmatrix}$ in Section 5.2.3 which have the same structure as $\mathbf{A}_{i,k}$ and it captures the probabilities such that $k$ packets are successfully transmitted given that there were $i$ packets in the queue before transmission. As expected, $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$ can be calculated directly from $\mathbf{H}_{i,k}$ since the joint transition of service/vacation state and channel state is captured in the same way in these matrices. The only remaining factor which determines the transitions among the levels in the underlying MC is the arrival process. Since the arrival process and the channel evolution process are independent, we need to determine the combinations which result in the corresponding level transition for $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$. Now if $q(t+1)$ and $q(t)$ denote the number of packets in the queue in two consecutive time slots, then $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$ represent the level transition where $q(t+1) - q(t) = 1 - k$. If $a(t)$ and $e(t)$ denote the number of arriving packets and the number of packets successfully transmitted during time slot $t$, then $q(t+1) - q(t) = a(t) - e(t)$. Therefore, $e(t) = k$ if $a(t) = 1$ (i.e., one arrival) and $e(t) = k - 1$ if $a(t) = 0$ (i.e., no arrival). Therefore, $\mathbf{A}_{i,k}$ and $\mathbf{A}_k$ can be written as follows:

$$\mathbf{A}_{i,k} = (1-\lambda)\mathbf{H}_{i,k-1} + \lambda\mathbf{H}_{i,k} \tag{E.1}$$

where $1 \le i < N$, and $0 \le k \le i+1$; and

$$\mathbf{A}_k = (1-\lambda)\mathbf{H}_{N,k-1} + \lambda\mathbf{H}_{N,k} \tag{E.2}$$

where $0 \le k \le N+1$, and $\mathbf{H}_{i,k} = \mathbf{0}$ if $k < 0$ or $k > i$.

As will be seen from the following derivations, $\mathbf{H}_{i,k} = \mathbf{H}_{N,k}$ for $i \geq N$; therefore, the matrix blocks $\mathbf{A}_k$ are independent of the level index $i$. Finally, the two matrix blocks at level 0 can be calculated as follows:

$$\mathbf{A}_{0,1} = (1-\lambda)\mathbf{S}\,; \quad \mathbf{A}_{0,0} = \lambda\mathbf{S}. \tag{E.3}$$

The two matrix blocks in (E.3) above describe level transitions where no transmission occurs since there is no packet in the queue before the transition. To derive $\mathbf{H}_{i,k}$, we exploit the structure of matrix $\mathbf{S}$ written in (5.1). Since we observe the system state at the beginning of the time slot, $\mathbf{S}_{(1,0)}$ and $\mathbf{S}_{(1,1)}$ capture the state transitions in a vacation slot, where no packet transmission occurs. In contrast, $\mathbf{S}_{(0,0)}$ and $\mathbf{S}_{(0,1)}$ capture the state transitions in a service slot. Specifically, row $j$ ($j = 0, 1, \cdots, K$) of these two sub-matrices represents the fact that the channel is in state $j$ in the slot where $c_j$ packets are transmitted. Given the packet error probability and the number of transmitted packets, the probability that a particular number of packets are successfully received at the receiver can be determined. To this end, let us define the following:

- $\theta_k = \overline{\mathrm{PER}}_k$ denotes the probability of transmission error when the channel is in state $k$[1]. Assuming that packet errors are independent, the probability that $i$ packets are correctly received given that $j$ packets were transmitted when the channel was in state $k$ can be written as

$$p_{i,j}^{(k)} = \begin{pmatrix} j \\ i \end{pmatrix} \theta_k^{j-i} (1-\theta_k)^i .$$

- $\mathbf{W}_j = \begin{bmatrix} \mathbf{S}_{0,0}^{(j)} & \mathbf{S}_{0,1}^{(j)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ ($j = 0, \cdots, K$), where $\mathbf{S}_{h,l}^{(j)}$ is constructed by keeping the ($j+1$)-st row of $\mathbf{S}_{h,l}$ while setting all other rows to $\mathbf{0}$. This matrix represents the fact that the queue is in service and the channel is in state $j$ at the beginning of the transmission slot.
- $\mathbf{V} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{1,0} & \mathbf{S}_{1,1} \end{bmatrix}$. This matrix represents evolution of the joint service/vacation and channel state processes given that the queue is on vacation at the beginning of the transmission slot.

[1] Note that, $\theta_k = P_0$ is a special case. The following analysis is kept general for any value of $\theta_k$.

Now $\mathbf{H}_{i,k}$ can be calculated as follows:

$$\mathbf{H}_{i,k} = \sum_{j=1}^{K} p_{k,d_j}^{(j)} \mathbf{W}_j, \ k > 0 \tag{E.4}$$

$$\mathbf{H}_{i,0} = \mathbf{V} + \mathbf{W}_0 + \sum_{j=1}^{K} p_{0,d_j}^{(j)} \mathbf{W}_j \tag{E.5}$$

where $p_{k,d_j}^{(j)} = 0$ if $k > d_j$, and $d_j = \min\{i, c_j\}$; that is, the number of packets transmitted is the minimum of the number of packets in the queue (equal to $i$) and the transmission capacity of the channel (equal to $c_j$). Equations (E.4) and (E.5) can be interpreted as follows. The sum-term in (E.4) captures all combinations which occur in a service slot such that $k$ packets among those transmitted (equal to $d_j$) are correctly received given that there were $i$ packets in the queue before transmission. Equation (E.5) captures the fact that no packet can successfully leave the queue given that there are $i$ packets in the queue at the beginning of the slot. This can happen in three cases: the target user is on vacation during the slot, and therefore, no transmission occurs (captured by $\mathbf{V}$); the channel is in state 0 where no transmission is allowed (captured by $\mathbf{W}_0$); and all transmitted packets are in error (captured by the sum-term).

# Appendix F

# Derivations of Transition Probabilities for Markov Chains $\mathbf{X}(t)$ and $X_k(t)$

We derive the transition probabilities for MCs $\mathbf{X}(t)$ in Section 6.2 and $X_k(t)$ in Section 6.3 in this Appendix. Before deriving $\Pr\{(x_1, y_1) \to (x_2, y_2)\}$, let us define $\gamma^{(l)}(n,m)$ as the probability that $m$ packets are correctly received given that $n$ packets were transmitted over link $l$. Assuming that packet errors are independent, we can calculate $\gamma^{(l)}(n,m)$ as follows:

$$\gamma^{(l)}(n,m) = \begin{pmatrix} n \\ m \end{pmatrix} \left(\beta^{(l)}\right)^{n-m} \left(1-\beta^{(l)}\right)^{m} \tag{F.1}$$

where $\beta^{(l)}$ is the probability of transmission error on link $l$ as defined in Section 6.1.2.

## F.1 Derivations of Transition Probabilities for MC $\mathbf{X}(t)$

Let $s$ be the number of packets arriving at queue one in a particular time frame and the transmission capability on link one is $k$ packets. We need to find the conditions under which a general transition $(x_1, y_1) \to (x_2, y_2)$ occurs. The number of packets in queue one after accepting newly arriving packets is $\min(x_1+s, Q_1)$ and the number of packets transmitted on link one is $\min(x_1, k)$. Assuming that among these transmitted packets, $m$ packets are correctly received at the receiving end (i.e., these $m$ successfully transmitted packets will enter queue two), we have $x_2 = \min(x_1+s, Q_1)-m$. Note that due to the employment of the infinite-persistent ARQ protocol in the link layer, all the erroneous packets will stay in the buffer for retransmission until they are successfully transmitted. Similarly, assuming that the transmission capability of the second link is $l$ packets and $n$ packets among $\min(y_1, l)$ transmitted packets are correctly received at the receiving end of link two, we have $y_2 = \min(y_1 + m, Q_2) - n$. Hence, we can calculate $\Pr\{(x_1, y_1) \to (x_2, y_2)\}$ as

$$\begin{aligned}\Pr\{(x_1, y_1) \to (x_2, y_2)\} &= \sum_s \sum_k \sum_l \sum_m \sum_n \mathrm{a}_s^{(1)} p_k^{(1)} p_l^{(2)} \times \gamma^{(1)}(\min\{x_1, k\}, m) \\ &\quad \times \gamma^{(2)}(\min\{y_1, l\}, n) \qquad \text{(F.2)}\end{aligned}$$

where all possible cases such that $x_2 = \min(x_1 + s, Q_1) - m$ and $y_2 = \min(y_1 + m, Q_2) - n$ are included in the sum.

## F.2 Derivations of Transition Probabilities for MC $X_k(t)$

We derive the transition probabilities for MC $X_k(t)$ for a particular queue $k$ of the tandem system defined in Section 6.3. Let us consider a general transition probability $\Pr\{x_1 \to x_2\}$. Let $s$ be the number of packets arriving at queue $k$ and the transmission capacity of the wireless link is $l$ packets during the considered time frame and let us assume that $m$ packets among $\min\{x_1, l\}$ transmitted packets are correctly received. Then, we have $x_2 = \min\{x_1 + s, Q_k\} - m$. Thus, the transition probability $\Pr\{x_1 \to x_2\}$ can be found as

$$\Pr\{x_1 \to x_2\} = \sum_s \sum_l \sum_m \mathrm{a}_s^{(k)} p_l^{(k)} \times \gamma^{(k)}(\min\{x_1, l\}, m) \qquad \text{(F.3)}$$

where all combinations of $s$, $l$ and $m$ for which $x_2 = \min\{x_1 + s, Q_k\} - m$ are included in the sum.

# Bibliography

[1] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70-77, July 2000.

[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,* ANSI/IEEE Standard 802.11, 1999.

[3] R. Pabst et al., "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Commun. Mag.*, vol. 42, no. 9, pp. 80-89, Sept. 2004.

[4] L. B. Le and E. Hossain, "Multihop cellular networks: Potential gains, research challenges, and a resource allocation framework," submitted to *IEEE Commun. Mag.*

[5] H. Viswanathan and S. Mukherjee, "Performance of cellular networks with relays and centralized scheduling," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2318-2328, Sept. 2005.

[6] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, pp. 23-30, Sept. 2005.

[7] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. 123-131, March 2005.

[8] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74-80, Oct. 2003.

[9] *Physical Layer Standard for cdma2000 Spread Spectrum Systems (Release C)*, 3GPP2, C.S0002-C, Jan. 2002.

[10] *UTRA High Speed Downlink Packet Access; Overall Description (Release 5)*, 3GPP, 3G TR25.308 V5.0.0, Sept. 2001.

[11] *cdma2000 High Rate Packet Data Air Interface Specification*, 3GPP2, C.S0024, Version 4.0, Oct. 2002.

[12] *Physical Layer Aspects of UTRA High Speed Downlink Packet Access (Release 4)*, 3GPP, 3G TR25.848 V4.0.0, Mar. 2001.

[13] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson, "A comparison of the HIPERLAN/2 and IEEE 802.11a wireless LAN standards," *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 172-180, May 2002.

[14] S. Catreux, V. Erceg, D. Gesbert, and R. W. Heath Jr., "Adaptive modulation and MIMO coding for broadband wireless data networks," *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 108-115, June 2002.

[15] X. Tang, M. Alouini, and A. J. Goldsmith, "Effect of channel estimation error on MQAM BER performance in Rayleigh fading," *IEEE Trans. Commun.*, vol. 47 , no. 12, pp. 1856-1864, Dec. 1999.

[16] A. J. Goldsmith and S. Chua, "Variable-rate variable-power MQAM for fading channels ," *IEEE Trans. Commun.*, vol. 45 , no. 10, pp. 1218-1230, Oct. 1997.

[17] X. Qiu and K. Chawla, "On the performance of adaptive modulation in cellular systems," *IEEE Trans. Commun.*, vol. 47 , no. 6, pp. 884-895, June 1999.

[18] S. T. Chung and A. J. Goldsmith, "Degrees of freedom in adaptive modulation: A unified view ," *IEEE Trans. Commun.*, vol. 49 , no. 9, pp. 1561-1571, Sept. 2001.

[19] M. S. Alouini and A. J. Goldsmith, "Adaptive modulation over Nakagami fading channels," *Kluwer J. Wireless Commun.,* vol. 13, no. 1-2, pp. 119-143, May 2000.

[20] Q. Liu, S. Zhou, and G. B. Giannakis, "Queueing with adaptive modulation and coding over wireless link: Cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1142-1153, May 2005.

[21] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3 , no. 5 , pp. 1746-1755, Sept. 2004.

[22] M. Zorzi, "Some results on error control for burst-error channels under delay constraints" *IEEE Trans. Veh. Technol.*, vol. 50, no. 1, pp. 12-24, Jan. 2001.

[23] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state Markov channel" *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1537-1548, Oct. 1999.

[24] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels" *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 445-455, May 1997.

[25] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968-1981, Sept. 2000.

[26] L. Badia, M. Rossi, and M. Zorzi, "SR ARQ packet delay statistics in Markov channels in the presence of variable arrival rate," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1639-1644, July 2006.

[27] M. Rossi, L. Badia, and M. Zorzi, "Exact statistics of ARQ packet delivery delay over Markov channels with finite round-trip delay," in *Proc. GLOBECOM'03*, Dec. 2003.

[28] M. Rossi, L. Badia, and M. Zorzi, "SR-ARQ delay statistics on N-state Markov channels with finite round trip delay," in *Proc. IEEE GLOBECOM'04*, 29 Nov.-3 Dec. 2004.

[29] Z. Rosberg and M. Sidi, "Selective-repeat ARQ: The joint distribution of the transmitter and the receiver resequencing buffer occupancies," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1430-1438, Sept. 1990.

[30] L. B. Le, E. Hossain, and A. S. Alfa, "Radio link level performance evaluation in wireless networks using multi-rate transmission with ARQ-based error control," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 2647-2653, Oct. 2006.

[31] L. B. Le, E. Hossain, and M. Zorzi, "Queueing analysis for GBN and SR ARQ protocols under dynamic radio link adaptation with non-zero feedback delay," to appear in *IEEE Trans. Wireless Commun.*

[32] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. IEEE ICC'95*, 1995.

[33] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150-154, Feb. 2001.

[34] F. Berggren and R. Jantti, "Asymptotically fair transmission scheduling over fading channels," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 326-336, Jan. 2004.

[35] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Networkings*, vol. 7, no. 4, pp. 473-489, Aug. 1999.

[36] A. Jalali, R. Padovani,and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC'00 Spring*, May 2000.

[37] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas ," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277-1294, June 2002.

[38] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 2053-2064, Oct. 2001.

[39] Y. Liu, S. Gruhl, and E. W. Knightly, "WCFQ: An opportunistic wireless scheduler with statistical fairness bounds," *IEEE Trans. Wireless Commun.*, vol. 2, no. 5, pp. 1017-1028, Sept. 2003.

[40] L. Xu, X. Shen, and J. W. Mark, "Dynamic fair scheduling with QoS constraints in multimedia wideband CDMA cellular networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 60-73, Jan. 2004.

[41] P. Kong, K. Chua, and B. Bensaou, "A novel scheduling scheme to share dropping ratio while guaranteeing a delay bound in a multicode CDMA network," *IEEE/ACM Trans. Networkings*, vol. 11, no. 6, pp. 994-1006, Dec. 2003.

[42] Y. Cao and V. Li, "Scheduling algorithms in broadband wireless networks" *Proc. IEEE* , vol. 89, no. 1, pp. 76-87, Jan. 2001.

[43] W. S. Jeon, D. G. Jeong, and B. Kim, "Packet scheduler for mobile internet services using high speed downlink packet access," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1789-1801, Sept. 2004.

[44] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks" *Proc. IEEE* , vol. 83, no. 10, pp. 1374-1396, Oct. 1995.

[45] R. Fantacci and L. Zoppi "Performance evaluation of polling systems for wireless local communication networks" *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2148-2157, Nov. 2000.

[46] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 344-357, June 1993.

[47] Z. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of the generalized processor sharing scheduling discipline," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1071-1080, Aug. 1995.

[48] H. M. Chaskar and U. Madhow, "Fair scheduling with tunable latency: a round robin approach," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 592-601, Aug. 2003.

[49] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2566-2582, Nov. 2004.

[50] H. Wang and N. B. Mandayam, "A simple packet-transmission scheme for wireless data over fading channel," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1055-1059, July, 2004.

[51] L. B. Le, E. Hossain, and A. S. Alfa, "Service differentiation in multi-rate wireless networks with weighted round-robin scheduling and ARQ-based error control," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 208-215, Feb. 2006.

[52] L. B. Le, E. Hossain, and A. S. Alfa, "Delay statistics and throughput performance for multi-rate wireless networks under ARQ and multiuser diversity," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3234-3243, Nov. 2006.

[53] E. Royer and C. -K. Toh, " A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Pers. Commun.*, vol. 6, no. 2, pp. 46-55, Apr. 1999.

[54] B. Zhang and H. T. Mouftah, "QoS routing for wireless ad hoc networks: Problems, algorithms, and protocols," *IEEE Commun. Mag.*, vol. 43, no. 10, pp. 110-117, Oct. 2005.

[55] C. Richard and J. -S. Liu, "QoS routing in ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1426-1438, Aug. 1999.

[56] C. Zhu and M. Scott, "QoS routing for mobile ad hoc networks, " in *Proc. IEEE INFOCOM'01.*

[57] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks, " *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1488-1505, Aug. 1999.

[58] J. -H. Song, V. W. S. Wong, and V. C. M. Leung, "Efficient on-demand routing for mobile ad-hoc wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 7, pp. 1374-1383, Sept. 2004.

[59] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM'94.*

[60] A. Iwata et al., "Scalable routing strategies for ad hoc wireless networks, " *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1369-1379, Aug. 1999.

[61] D. Kim, C. -H. Min and S. Kim, "On-demand SIR and bandwidth-guaranteed routing with transmit power assignment in ad hoc mobile networks," *IEEE Trans. Veh. Technol.*, vol. 22, no. 7, pp. 1301-1321, Sept. 2004.

[62] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, E. Imielinski and H. Korth, eds., Kluwer Academic Publ.*, 1996.

[63] V. D. Park and M. S. Corson, "Temporally-ordered routing algorithms (TORA) version 1 functional specification, " IETF Internet Draft draft-ietf-manet-tora-spec-04.txt, July 2001.

[64] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing, " in *Proc. IEEE MOBICOM'03.*

[65] L. B. Le and E. Hossain, "Tandem queue models with applications to QoS routing in multihop wireless networks," submitted to *IEEE Trans. Mobile Computing.*

[66] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing, " IETF RFC 3561, July 2003.

[67] P. Pham and S. Perreau, "Performance analysis of reactive shortest path and multi-path routing mechanism with load balance, " in *Proc. IEEE INFOCOM'03.*

[68] A. Tsirigos and Z. J. Haas, "Analysis of multipath routing - Part I: The effect on the packet delivery ratio, " *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138-146, Jan. 2004.

[69] X. Lin and N. B. Shroff, "An optimization based approach for QoS routing in high bandwidth networks, " in *Proc. IEEE INFOCOM'04*.

[70] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136-1144, July 2004.

[71] C. -K. Toh, M. Delwar, and D. Allen, "Evaluating the communication performance of an ad hoc wireless network," *IEEE Trans. Wireless Commun.*, vol. 1, no. 3, pp. 402-414, July 2002.

[72] J. A. Morrison, "Two discrete-time queues in tandem," *IEEE Trans. Commun.*, vol. 27, no. 3, pp. 563-573, Mar. 1979.

[73] M. Xie and M. Haenggi, "Delay performance of different MAC schemes for multihop wireless networks," in *Proc. GLOBECOM'05.*

[74] A. Brandwajn and Y. L. L. Jow, "An approximation method for tandem queues with blocking," *Opns. Res.*, vol. 36, no. 1, pp. 73-83, Jan.-Feb. 1988.

[75] A. Burchard, J. Liebeherr, and S. D. Patek, "A min-plus calculus for end-to-end statistical service guarantees," *IEEE Trans. Inf. Theory,* vol. 52, no. 9, pp. 4105-4114, Sept. 2006.

[76] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, to appear.

[77] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadowing prices, proportional fairness, and stability," *J. Oper. Res. Soc.* , vol. 49, no. 3, pp. 237-252, Mar. 1998.

[78] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452-1463, Aug. 2006.

[79] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity - Part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927-1938, Nov. 2003.

[80] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity - Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939-1948, Nov. 2003.

[81] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, " Cooperative diversity in wireless network: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062-3080, Dec. 2004.

[82] M. Grossglauser and D. Tse, "Mobility increases the capacity of wireless adhoc networks," in *Proc. IEEE INFOCOM'01.*

[83] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. Krishnamurthy, and O. Ercetin, "A framework for distributed spatio-temporal communications in mobile ad hoc networks," in *Proc. IEEE INFOCOM'06.*

[84] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3037-3063, Sept. 2005.

[85] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, 2004.

[86] H. Zhu and G. Cao, "rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks," in *Proc. IEEE INFOCOM'05.*

[87] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative wireless communications: A cross-layer design," *IEEE Wireless Commun.*, vol. 13, no. 4, pp. 84-92, Aug. 2006.

[88] A. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in wireless networks," in *Allerton Conference on Communications, Control and Computing*, Oct. 2003.

[89] I. Maric and R. Yates, "Cooperative multicast for maximum network lifetime," *IEEE J. Sel. Areas Commun.*, vol. 23. no. 1, pp. 127-135, Jan. 2005.

[90] L. B. Le and E. Hossain, "An analytical model for ARQ cooperative diversity in multihop wireless networks," to appear in *IEEE Trans. Wireless Commun.*

[91] L. B. Le and E. Hossain, "Cross-layer optimization frameworks for multihop cooperative wireless networks," submitted to *IEEE Trans. Wireless Commun.*

[92] H. S. Wang and N. Moayeri, "Finite-state Markov channel - A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163-171, Feb. 1995.

[93] J. Razavilar, K. J. R. Liu, and S. I. Marcus, "Jointly optimized bit-rate/delay control policy for wireless packet networks with fading channels," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 484-494, Mar. 2002.

[94] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688-1692, Nov. 1999.

[95] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1266, Sept. 1960.

[96] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977-1997, Sept. 1963.

[97] H. S. Wang and P. -C. Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, no. 2, pp. 353-357, May 1996.

[98] Y. L. Guan and L. F. Turner, "Generalised FSMC model for radio channels with correlated fading," *IEE Proc. Commun.*, vol. 146, no. 2, pp. 133-137, Apr. 1999.

[99] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks, " *IEEE Commun. Mag.*, vol. 32, pp. 70-81, Mar. 1994.

[100] A. S. Alfa, "Algorithmic analysis of the BMAP/D/k system in discrete time ," *Advances in Applied Probability*, pp. 1131-1152, 2003.

[101] A. S. Alfa, "Discrete time analysis of MAP/PH/1 vacation queue with gated time-limited service," *Queueing Systems*, pp. 35-54, 1998.

[102] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 668-677, Dec. 1993.

[103] K. -D. Lee and V. C. M. Leung, "Fair allocation of subcarrier and power in an OFDMA wireless mesh network," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2051-2060, Nov. 2006.

[104] G. Kulkarni, S. Adlakha, and M. Srivastava, "Subcarrier allocation and bitloading algorithms for OFDMA-based wireless networks, *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 652-662, Nov./Dec. 2005.

[105] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multichannel multi-interface wireless networks," in *Proc. WCNC'05*, Mar. 2005.

[106] H. Ed. Inamura, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," RFC 3481, Feb. 2003.

[107] L. B. Le, E. Hossain, and T. Le-Ngoc, "Interaction between radio link level truncated ARQ and TCP in multi-rate wireless networks: A cross-layer performance analysis," accepted with minor revision in *IEE Proc. Commun.*

[108] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection, " *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 659-672, Mar. 2006.

[109] C. E. Koksal and H. Balakrishnan, "Quality-aware routing metrics for time-varying wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1984-1994, Nov. 2006.

[110] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 556-567, Oct. 2000.

[111] S. Lin and D. Costello, *Error Control Coding,* New York: Prentice-Hall, 1982.

[112] J. G. Proakis, *Digital Communications,* New York: McGraw-Hill, 3rd Edition, 1995.

[113] G. L. Stuber, *Principles of Mobile Communication,* Norwell, MA: Kluwer Academic, 2nd Edition, 2001.

[114] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models - An Algorithmic Approach*, John Hopkins Univ. Press, Baltimore, MD, 1981.

[115] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, John Wiley and Sons, 1985.

[116] S. Boyd and L. Vandenberge, *Convex Optimization*, Cambridge University Press, 2004.

[117] D. P. Bertsekas and R. G. Gallager, *Data Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[118] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[119] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, 1985.

[120] D. P. Bertsekas, *Nonlinear Programming*, 2nd Edition. Belmont, MA: Athena Scientific, 1999.

# VITA

| | | | |
|---|---|---|---|
| *Surname:* | Le | *Given Names:* | Long Bao |
| *Place of Birth:* | Vietnam | *Date of Birth:* | Feb. 05, 1976 |

### *Educational Institutions Attended*

| | |
|---|---|
| Ho Chi Minh City University of Technology (HCMUT) | 1994 to 1999 |
| Asian Institute of Technology (AIT) | 2001 to 2002 |

### *Degrees Awarded*

| | | |
|---|---|---|
| B.Eng. | HCMUT | 1999 |
| M.Eng. | AIT | 2002 |

### *Honors and Awards*

| | |
|---|---|
| University of Manitoba Graduate Fellowship (UMGF), UoM | 2005 - 2007 |
| Research Assistantship, UoM | 2003 - 2007 |
| Edward R. Toporeck Graduate Fellowship, two times, UoM | 2005-2007 |
| University of Manitoba Students' Union Scholarship, UoM | 2005 |
| Conference Travel Award, UoM | 2005 |
| IEEE Student Travel Award IEEE WCNC'2003, IEEE ICC'2005 | |
| Keikyu Fellowship, financial support for Master Program, AIT | 2001-2002 |
| Gold Medal, most outstanding student among graduates, HCMUT | 1999 |
| Third Prize, Vietnam National Mathematical Olympiad, high school time | 1994 |

### *Journal Publications*

1. L. B. Le, E. Hossain, and A. S. Alfa, "Service differentiation in multi-rate wireless networks with weighted round-robin scheduling and ARQ-based error control," *IEEE Transactions on Communications*, vol. 54, no. 2, pp. 208-215, Feb. 2006.
2. L. B. Le, E. Hossain, and A. S. Alfa, "Radio link level performance evaluation in wireless networks using multi-rate transmission with ARQ-based error control," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2647-2653, Oct. 2006.
3. L. B. Le, E. Hossain, and A. S. Alfa, "Delay statistics and throughput performance for multi-rate wireless networks under ARQ and multiuser diversity,"

*IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3234-3243, Nov. 2006.

4. L. B. Le, E. Hossain, and M. Zorzi, "Queueing analysis for GBN and SR ARQ protocols under dynamic radio link adaptation with non-zero feedback delay," to appear in *IEEE Transactions on Wireless Communications.*
5. L. B. Le and E. Hossain, "An analytical model for ARQ cooperative diversity in multihop wireless networks," to appear in *IEEE Transactions on Wireless Communications.*
6. L. B. Le, E. Hossain, and T. Le-Ngoc, "Interaction between radio link level truncated ARQ and TCP in multi-rate wireless networks: A cross-layer performance analysis," accepted with minor revision in *IEE Proceeding on Communications.*
7. L. B. Le and E. Hossain, "Tandem queue models with applications to QoS routing in multihop wireless networks," submitted to *IEEE Transactions on Mobile Computing.*
8. L. B. Le and E. Hossain, "Cross-layer optimization frameworks for cooperative diversity multihop wireless networks," submitted to *IEEE Transactions on Wireless Communications.*
9. L. B. Le and E. Hossain, "Multihop cellular networks: Potential gains, research challenges, and a resource allocation framework," submitted to *IEEE Communications Magazine.*

### *Conference Publications*

1. L. B. Le and E. Hossain, "Joint rate control and resource allocation in OFDMA wireless mesh networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'07)*, Hong Kong, Mar. 2007.
2. L. B. Le, A. T. Nguyen, and E. Hossain, "A tandem queue model for performance analysis in multihop wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'07)*, Hong Kong, Mar. 2007.
3. L. B. Le, E. Hossain, and T. Le-Ngoc, "Effects of link-level queueing and truncated ARQ on TCP throughput in multi-rate wireless networks," in *Proc. International Conference on Quality of Service in Heterogeneous Wired, Wireless Networks (QSHINE'06)*, Waterloo, Canada, Aug. 2006.
4. L. B. Le and E. Hossain, "Queueing analysis of go-back-N ARQ protocol in multi-rate wireless networks with feedback delay," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM'05)*, St. Louis, MO, USA, 28 Nov.-2 Dec. 2005.
5. L. B. Le and E. Hossain, "Delay statistics for selective repeat ARQ protocol in multi-rate wireless networks with non-instantaneous feedback," in *Proc.*

*IEEE Global Telecommunications Conference (GLOBECOM'05)*, St. Louis, MO, USA, 28 Nov.-2 Dec. 2005.

6. L. B. Le, E. Hossain, and A. S. Alfa, "Delay statistics in multi-rate wireless networks with ARQ and weighted round-robin scheduling," in *Proc. IEEE Vehicular Technology Conference (VTC'05 Fall)*, Dallas, TX, USA, 25-28 Sept. 2005.

7. L. B. Le, E. Hossain, and A. S. Alfa, "Queueing analysis and admission control for multi-rate wireless networks with opportunistic scheduling and ARQ-based error control," in *Proc. IEEE International Conference on Communications (ICC'05)*, Seoul, Korea, May 2005.

8. L. B. Le, E. Hossain, and A. S. Alfa, "Queueing analysis for radio link level scheduling in a multi-rate TDMA wireless network," in *Proc. IEEE Global Communication Conference (GLOBECOM'04)*, Dallas, Texas, USA, 29 Nov.-3 Dec. 2004.

9. L. B. Le and E. Hossain, "On the performance of spatial multiplexing MIMO cellular systems with adaptive modulation and scheduling," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'04)* , Atlanta, USA, 21-25 March 2004.

10. L. B. Le, K. Ahmed, and H. Tsuji, "Mobile location estimator with NLOS mitigation using Kalman filtering," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'03)*, New Orleans, Louisiana, USA, 16-20 March 2003.