Summer 2014

Efficient Routing and Scheduling in Wireless Networks

Anand Seetharam

EFFICIENT ROUTING AND SCHEDULING IN WIRELESS NETWORKS

A Dissertation Presented

by

ANAND SEETHARAM

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2014

School of Computer Science

EFFICIENT ROUTING AND SCHEDULING IN WIRELESS NETWORKS

A Dissertation Presented

by

ANAND SEETHARAM

Approved as to style and content by:

Jim Kurose, Chair

Don Towsley, Member

Arun Venkataramani, Member

Dennis Goeckel, Member

Lori A. Clarke, Chair
School of Computer Science

To
Arti, amma, appa and akka

ACKNOWLEDGMENT

I express my deepest gratitude to Prof. Jim Kurose who has been a great advisor as well as a true mentor to me. In my opinion, he is definitely one of the best advisors one can get. His encouragement and constructive feedback have molded my research style; his intellectual curiosity and fundamental insights into complex problems have been a continual inspiration. He has also taught me the importance of clearly articulating one's thoughts and asking the 'right research questions'.

I thank my committee members, Prof. Don Towsley and Prof. Dennis Goeckel for their guidance. They have always been available to discuss research ideas; their contribution has been critical in preparing this document. They have also taught me the importance of being mathematically precise. I thank Prof. Arun Venkataramani for his inputs on this thesis. It has been a pleasure working with him on other research projects as well.

I thank Partha Dutta and Vijay Arya with whom I have had the good fortune of working on one of the problems in this thesis. It was an enjoyable experience working with them during my internship at IBM India. I am deeply indebted to my undergraduate advisor, Prof. Mrinal Naskar from Jadavpur University and my high school teacher Sujit Bose. Without their able guidance and support, I would have never made it this far.

I thank my labmates at CNRG and colleagues at UMASS for their camaraderie and wonderful discussions, particularly Yung-Chih Chen, Bo Jiang, Dan Gyllstrom, Mostafa Dehghan, Misha Badov, Yeonsup Lim, Sookhyun Yang, Chang Liu, Kyungsoo Lee, Simon Heimlicher, Elisha Rosensweig, Xiaozheng Tie, Abhigyan Sharma, Sandeep Kalra, Veena Udayabhanu, Venkatesh Murthy, Harshal Pandya, Vikas Ku-

mar, Sai Keshavan Balchand and Monojit Bag. Thanks is also due to Arijit Biswas, Aritra Banerjee, Ayan Acharya and Rini Chowdhury for their encouragement and support.

Finally, I thank my family - my amma and appa for providing me great education and material comfort, in spite of their hardships; my amma for motivating me, instilling in me the courage to pursue my dreams, always having belief in my abilities and my akka for being a constant source of support and helping me in all my endeavors. My brother-in-law Sougata deserves thanks for valuable advice he has given me at different points in my career.

Most of all, I thank my wife, Arti for supporting me and having more confidence on me than myself. I also thank her for bearing with me during my paper deadlines and reassuring me during my job search. It was also a great experience taking the Algorithms course with her at UMASS. Finally, I want to thank her for making my life at UMASS so much fun.

ABSTRACT

EFFICIENT ROUTING AND SCHEDULING IN WIRELESS NETWORKS

SEPTEMBER 2014

ANAND SEETHARAM

B.E., JADAVPUR UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Jim Kurose

The temporal and spatial variation in wireless channel conditions, node mobility make it challenging to design protocols for wireless networks. In this thesis, we design efficient routing and scheduling algorithms which adapt to changing network conditions caused by varying link quality or node mobility to improve user-level performance.

We design and analyze routing protocols for static, mobile and heterogeneous wireless networks. We analyze the performance of opportunistic and cooperative forwarding in static mesh networks showing that opportunism outperforms cooperation; we identify interference as the main cause for mitigating the potential gains achievable with cooperative forwarding. For mobile networks, we quantitatively analyze the tradeoff between state information collection (sampling frequency and number of bits per sample) and power consumption for a fixed source-to-destination goodput constraint. For heterogeneous networks comprising of both static and mobile nodes, we

propose a greedy algorithm (*adaptive-flood*) which dynamically classifies individual nodes as routers/flooders depending on network conditions and demonstrate that it achieves performance equivalent to, and in some cases significantly better than, that of network-wide routing or flooding alone.

Last, we consider an application-level wireless streaming scenario where multiple clients are streaming different videos from a cellular base station. We design a greedy algorithm for efficiently scheduling multiple video streams from a base station to mobile clients so as to minimize the total number of application-playout stalls. We develop models for coarse timescale wireless channel variation to aid network and application-layer protocol design.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The temporal and spatial variation in wireless channel conditions makes it challenging to design protocols for wireless networks. In order to develop efficient and robust protocols, it is essential to understand the inherent characteristics of wireless networks such as connectivity, coverage and varying channel conditions. Factors such as multipath fading, shadowing and path loss cause wireless channel variability at different timescales (in the milliseconds, seconds and tens of seconds timescale respectively). Node mobility also plays an important role in determining wireless channel variability. Changes in network connectivity and topology caused by node mobility and fluctuating channel conditions mean that protocols have to make design decisions based on partial or outdated network state information. These variations, however, present opportunities to leverage the dynamic (varying) nature of these networks to improve application-level performance. *The goal of this thesis is to design efficient routing and scheduling algorithms that adapt to changing network conditions caused by varying link quality or node mobility to improve user-level performance.*

The last decade has witnessed the growth and deployment of diverse networks such as mesh, ad-hoc, WiFi and 4G (LTE/WiMAX) for various commercial and military purposes. One major question still remains unanswered: *How can one adapt and leverage wireless channel variability to improve the application-level performance of clients using these networks?* We seek to answer this question by developing efficient protocols that take advantage of wireless channel properties (such as multipath fading and shadowing) to make important decisions related to routing, resource scheduling and network state information collection for a wide range of wireless networks, both

static and mobile. We also develop models for coarse timescale wireless channel variation to aid network and application-layer protocol design. In this context, we study the following problems.

1.1 How to Route in Wireless Networks?

In the first part of this thesis, we develop and analyze routing protocols for static, mobile and heterogeneous networks. We study static multi-hop wireless mesh networks, modeling and analyzing two classes of routing protocols - opportunism and cooperation, under varying channel conditions and in the presence of interfering transmitters. The broadcast nature of wireless networks allows a much richer set of approaches to be taken when forwarding packets between source and destination than traditional hop-by-hop forwarding along pre-specified paths. These strategies fall into two broad categories - opportunistic forwarding, which exploits relay diversity by opportunistically selecting an overhearing relay as a forwarder, and cooperative forwarding, which relies on the synchronized transmissions of relays to reinforce received signal strengths. Our objective is to understand which among these two approaches provides higher performance (throughput) in presence of multiple competing and interfering network flows. We observe that opportunism outperforms cooperation and identify interference resulting from the larger number of transmissions under cooperative forwarding as a cause for mitigating the potential gains achievable with cooperative forwarding.

Mobility in wireless networks introduces additional sources of channel variation and makes routing even more challenging. Frequent changes in network topology require additional control overhead for gathering link state information needed for determining routes. We therefore analyze the tradeoff between the amount of signaling overhead incurred in path selection in a MANET with time-varying wireless channels and the application-level goodput and end-to-end power expended on the selected path. In dynamic network scenarios, increased overhead increases the accuracy

of link state estimates used in path selection but decreases the amount of bandwidth available for application use. We develop an information-theoretic, bounding approach to quantify the signaling overhead. Specifically, we investigate *(i)* the time granularity at which link state is sampled and communicated, and *(ii)* the minimum number of bits needed to encode this link state information, such that the expected power consumption within a sampling interval is minimized subject to a fixed source-destination goodput constraint. We formulate an optimization problem that provides a numerically computable solution to these questions, and quantitatively demonstrate that short sampling intervals incur significant overhead while long intervals fail to take advantage of the temporal correlation in link state. Additionally, we find that using small number of bits per sample does not provide sufficient information about the network, while using too many bits provide little additional information at the expense of increased overhead.

In practice, mobility and connectivity characteristics observed in real-world measurements are often *heterogeneous*: while some nodes may have few or highly dynamic links, there are also well-connected nodes forming sizable connected components [41,89]. In heterogeneous networks comprised of both stable and highly dynamic components, it is likely that neither routing nor flooding alone may perform particularly well. Stateful protocols such as OLSR are suitable for networks connected by stable paths, but are outperformed by stateless flooding in sparse and rapidly changing networks. Therefore, our goal is to design a protocol that adapts seamlessly and dynamically to changing network conditions and provides superior performance over the full range of network operating conditions. Rather than design a new protocol for routing in heterogeneous mobile networks from scratch, we utilize an approach that leverages prior work by operating nodes individually as routers or flooders and switching mode in response to changing network conditions. We present a greedy algorithm (*adaptive-flood*) that dynamically and individually classifies nodes as routers or flooders. Nodes classified as routers forward data according to the forwarding

table computed by the native routing protocol, and those classified as flooders broadcast their traffic to all neighbors. Our simulations show that by effectively adapting the individual operations of nodes as routers/flooders, *adaptive-flood* achieves performance equivalent to, and in some cases significantly better than, that of network-wide routing or flooding alone.

1.2 How to Schedule Video Streams in Cellular Networks?

In the second part of this thesis, we explore a video streaming application for cellular networks and develop a scheduling algorithm to enhance the users' viewing experience. To aid network and application-layer protocol design, we also develop a Markov chain model for shadowing to capture its effect on received power. Shadowing is the variation in signal strength at the seconds timescale caused by large objects (e.g., buildings, trees) between the transmitter and receiver.

We investigate scheduling algorithms for transmitting multiple video streams from a base station to mobile clients with the objective of minimizing the number of application-playout stalls. We present an epoch-by-epoch framework to fairly allocate wireless transmission slots to streaming videos. First, we show that the problem of allocating slots fairly is NP-complete, even for a constant number of videos and then present a fast lead-aware greedy algorithm for the problem. Our greedy algorithm is optimal when the channel quality of a user remains unchanged within an epoch. Our experimental results, based on public MPEG-4 video traces and wireless channel traces that we collected from a WiMAX test-bed, show that the lead-aware greedy approach results in a fair distribution of stalls across clients when compared to other algorithms, and result in similar or fewer average number of stalls per client.

Efficient application-layer protocol design relies heavily on models which effectively capture variations in wireless channel conditions. We develop and study the effectiveness of a finite-state Markov chain model that captures variations due to shadowing, which occur at coarser time scales. Our work is in contrast to prior work,

which has focused primarily on channel modeling on a short, per-packet timescale (millisecond). The Markovian model is constructed by partitioning the entire range of shadowing into a finite number of intervals. We determine the Markov chain transition matrix in two ways: *(i)* via an abstract modeling approach in which shadowing effects are modeled as a log-normally distributed random process affecting the received power, and the transition probabilities are derived as functions of the variance and autocorrelation function of shadowing; *(ii)* via an empirical approach, in which the transition matrix is calculated by directly measuring the changes in signal strengths (collected over a WiMAX and a multi-hop mesh network). We validate the abstract model by comparing its steady state and transient performance predictions with those computed using the empirically derived transition matrix and those observed in the actual traces themselves.

1.3 Thesis Contributions

Having provided an overview of this thesis, we enumerate the main contributions of our research.

1. We construct Markovian models to analyze the performance of opportunistic and cooperative forwarding. We show that opportunism outperforms cooperation and identify interference as the main cause for mitigating the potential gains achievable with cooperative forwarding.

2. We quantitatively analyze the tradeoff between state information collection (sampling frequency and number of bits per sample) and power consumption for a fixed source-to-destination goodput constraint. We demonstrate that small number of bits per sample carry very little information about the network while large number of bits per sample carry marginal additional information. Similarly, we find that short sampling intervals incur significant overhead while long intervals fail to take advantage of the temporal correlation in link state.

3. For heterogeneous ad-hoc networks comprising of both static and mobile nodes, we propose a greedy algorithm (*adaptive-flood*) that dynamically classifies individual nodes as routers/flooders depending on network conditions and demonstrate that it achieves performance equivalent to, and in some cases significantly better than, that of network-wide routing or flooding alone.

4. We design a greedy algorithm for scheduling multiple video streams from a base station to mobile clients and show that our approach is fair and is successful in minimizing the number of application-playout stalls.

5. To aid application-layer protocol design, we design a Markovian model to capture the effect of shadowing on the received power. We develop analytical and empirical approaches to compute its transition matrix and show via experiments that the steady state and transient state performance of the Markovian model is close to that observed from real traces.

The rest of this thesis is structured as follows. In the first part of this thesis, we investigate routing protocols for static, mobile and heterogeneous networks. We compare opportunistic and cooperative forwarding for static mesh networks in Chapter 2. We investigate the tradeoff between network state information collection and power consumption in mobile networks in Chapter 3. Heterogeneous networks are studied in Chapter 4, where we develop a greedy algorithm for classifying individual nodes as routers/flooders and show its superior performance. In the second part of this thesis, we study the problem of scheduling multiple videos (simultaneously being streamed from a base station to different mobile clients) so as to minimize the total number of application-playout stalls in Chapter 5 and propose a greedy algorithm to address this issue. In Chapter 6, we present a Markovian model to model power variations in wireless networks. Finally, in Chapter 7, we summarize the contributions of this thesis and discuss future research directions. Throughout this thesis, we discuss re-

lated work in each of the individual chapters, in context of the research challenges addressed in that chapter.

CHAPTER 2

OPPORTUNISM VERSUS COOPERATION

2.1 Introduction

Unlike wireline networks, the broadcast nature of wireless communication allows a much richer variety of approaches for forwarding packets between a source and destination than traditional hop-by-hop forwarding along pre-specified paths. In particular, multiple nodes (in addition to the intended next-hop recipient) can overhear transmissions in a wireless network and serve as ad hoc relays to assist forwarding. Recently, two approaches have emerged that seek to exploit wireless channel characteristics when forwarding packets between source and destination in a multi-hop wireless setting:

(1) *Opportunistic Forwarding*: Because of the broadcast nature of the wireless medium, several neighboring nodes may overhear transmissions, even if none of them is the intended next-hop or final destination. A suitable relay can often be selected opportunistically among these overhearing nodes to forward the packet downstream, until it reaches its final destination [9, 13, 15, 18, 58, 59, 72, 106].

(2) *Cooperative Forwarding*: In properly synchronized and coded networks, the signal strengths of multiple simultaneous transmissions of the same packet can be additive. Thus, if multiple nodes have received the same packet and can synchronize their forwarding transmissions of that packet, the signal strengths at downstream receivers will be increased, thus improving the reception probability at these downstream nodes [28, 51, 77].

Although opportunistic forwarding and cooperative forwarding are well-known in the literature, their analysis and comparison in a network setting is rather limited. One

of the challenges is to find a simple and analyzable model that realistically captures important characteristics of the wireless medium, such as signal interference strength and random fading. Most extant work either focuses on link-level analysis in one-hop networks using a complex channel fading model (e.g., [51,77]), or multi-hop network-level analysis using a very simple channel fading model (e.g., [13,17]). Moreover, it is also important that multiple competing flows and their interaction/interference with each other be considered and understood.

In this chapter, we compare the performances of idealized and representative opportunistic and cooperative forwarding strategies under common (and realistic) assumptions. We note that the opportunistic and cooperative forwarding strategies studied in this chapter are simple and the performance of both schemes can be enhanced by careful design decisions. We stress that our goal here is *not* to propose new protocols or investigate a specific opportunistic or cooperative transmission protocol in detail. Instead, our more fundamental goal is to characterize and understand the differences between these two approaches to forwarding in various multi-hop wireless scenarios with multiple competing flows. Our contributions are as follows:

(1) We derive closed-form formulas for the packet reception probability in the presence of cooperative transmitters, interfering transmitters, and random fading. These results are subsequently used to study the performance (throughput) of opportunistic and cooperating forwarding strategies in multi-hop wireless networks with random fading.

(2) We then analyze a simple n-hop linear network supporting a single flow (e.g., a wireless network along a road) under opportunistic and cooperative forwarding. We observe that in the single flow case, cooperation outperforms opportunism. This result is intuitive; in the single flow case there is no interference among packets and as there are larger number of transmitters in cooperative forwarding, the downstream packet probability reception is greater than opportunistic

forwarding. Studying a single flow case in this special setting provides useful insights and helps us appreciate the results for multiple competing flows.

(3) We develop a Markovian model to determine the throughput achievable by opportunistic and cooperative forwarding for a general network with multiple competing flows. We analyze this model for a simple topology and show that opportunistic forwarding can achieve higher throughput than cooperative forwarding. We study larger-scale networks via simulation and observe that opportunism outperforms cooperation on average. The worse performance of cooperative forwarding can be largely attributed to the higher interference due to multiple competing flows. Lastly, we develop a fixed-point model for efficiently, but approximately computing the throughput of the Markov model, allowing performance comparisons in larger-scale networks.

Together, our results indicate that the relatively simple (and lower control overhead) opportunistic forwarding strategies are preferable to more complex cooperative counterparts in large networks with multiple competing flows. Our results also highlight the importance of considering multiple flows, since insights gained from single flow scenarios do not always carry over to the more complex multi-flow scenario, where interference among flows becomes important.

The rest of this chapter is organized as follows. We describe the forwarding strategies in detail in Section 2.3 and the wireless communication model in Section 2.4. We analyze a linear network supporting a single flow in Section 2.5. For multiple flows and general topologies, we present a Markovian model in Section 2.6, which we use to study a simple diamond topology, together with simulations on random topologies. In Section 2.7, we provide the fixed-point iteration for solving the Markov model. We discuss the effect of correlated channels on the performance of opportunistic and cooperative forwarding in Section 2.8 and conclude the chapter in Section 2.9.

2.2 Related Work

The first work proposing opportunistic forwarding is [13]. Since then, several strategies have been proposed to improve the performance of opportunistic forwarding [18, 59, 72, 106]. Research efforts have also theoretically analyzed the benefits of opportunism, including [58], where the authors performed a Markovian analysis to determine the expected number of network-wide link-layer transmissions needed to transfer a packet from source to destination in a wireless mesh network. [58] mostly assumes that link success probabilities are provided *a priori* and does not consider random fading in a SINR model, an important component of our models. Also, [17] provides a recursive relation for estimating the minimum number of required opportunistic transmissions. Similarly, [15] proposes an analytical model to study the performance (expected transmission count) of opportunistic routing protocols. [9] quantifies the average end-to-end delay obtained by using opportunistic schemes and demonstrates that it is about half that obtained using typical shortest path routing. None of these works consider a realistic SINR model with random fading.

A considerable amount of research has also considered cooperation in wireless networks [51, 64, 77]. [77] and [51] summarize much of this prior work in cooperative diversity and demonstrate how cooperation improves network performance. [68] is one of the few papers that describes an implementation of cooperative forwarding. It demonstrates that by properly synchronizing sender transmissions to symbol boundaries, it is possible to outperform opportunistic routing in the absence of interference for a simple topology. Most past research on cooperation has been in the context of the physical layer, with only a few efforts exploring how cooperation interacts with higher network layers [77] and in the presence of multiple interfering flows. In [28] the authors discuss how to effectively schedule cooperative transmissions for multiple access scenarios by helping sources with poor channels to the destination use relays that have better channel quality. We note that our work differs from prior work in that we address primarily the network-layer concern (with multihop forwarding), with

the goal of comparing opportunistic forwarding and cooperative forwarding – using a simple model of SINR with random fading, and in a multihop setting.

2.3 Forwarding Strategies

This section describes the two forwarding strategies compared in this chapter – opportunistic forwarding and cooperative forwarding. We focus on generic and representative instantiations of these strategies.

(**a**) *Opportunistic Forwarding*: If the packet cannot reach the destination in one hop, it is relayed by the overhearing node closest to the destination[1]. This proceeds in multiple steps, until the packet reaches the destination. In the literature, there are proposals [13,55] to address implementation details, such as how to select the appropriate relay when multiple nodes overhear the transmission[2]. We abstract away these details, and focus on analyzing this idealized implementation in order to shed insight into the main advantage offered by opportunism - the ability to opportunistically select a relay that is closest to the destination.

(**b**) *Cooperative Forwarding*: To exploit the additive property of wireless signals, multiple overhearing relays can transmit the packets towards the destination, when proper synchronization (e.g., by GPS) among multiple transmitters is feasible. This is the key innovation introduced in a cooperative strategy. We assume that a flow maintains a list of relays associated with it. When a node belonging to a list of relays of a particular flow overhears the transmission from the flow, it will be assigned as a relay. In the case of multiple network flows, we do not assume that nodes are allowed to coordinate their transmissions

[1]In more sophisticated settings, it can be relayed by the node that has the best estimate channel condition (in some metrics [72]) to the destination. We focus on the simplest setting for our analysis.

[2]One solution is to use a very low-data rate, reliable out-of-band control channel to transmit the ACKs among overhearing relays [13], while the relays can be selected in a way to ensure that they can overhear ACKs among themselves [72].

with other nodes that receive packets from other flows, as this would involve prohibitively high overhead. In this case, competing flows are essentially treated as interference.

A more sophisticated variant of cooperative forwarding is:

(**c**) *Selective Cooperative Forwarding*: Although cooperation can reinforce signal strength, it can also increase the interference level to other simultaneous flows. A more refined strategy is not to assign all nodes as relays, but to instead select only a small subset of nodes that are closest to the destination or have advantageous wireless channel conditions when transmitting towards the destination. This is essentially a hybrid strategy of both opportunistic and cooperative forwarding. For convenience of analysis, we focus on a simple selective cooperative forwarding strategy that only assigns two nodes as relays that are closest to the destination among the overhearing nodes in the list of potential relays.

2.4 Wireless Communication Model

In order to compare the performance of different forwarding strategies, we use the following channel model to account for SINR and random fading. We proceed in multiple steps. Let us assume that there are C nodes in the network.

(**a**) *Single Transmission*: Let us first consider the simplest case with a single transmission between node i and node j ($\forall\ i, j$ =1 to C, $i \neq j$). Denote by $\mathsf{S}_{i,j}$ the signal-to-noise ratio from transmitter i to receiver j:

$$\mathsf{S}_{i,j} \triangleq \frac{|x_{i,j}|^2 \mathsf{P} d_{i,j}^{-\alpha}}{\mathsf{N}_0} \tag{2.1}$$

where N_0 is a constant background noise, $|x_{i,j}|^2$ is the Rayleigh fading coefficient (the flat fading channel is modeled as a Gaussian random process $x_{i,j}$ [107]), $d_{i,j}$ is the distance between i and j, α is the path loss exponent, and P is the transmission power at i. Note that $d_{i,j} \geq 1$ and $\alpha \geq 2$.

We assume that parameters $\mathsf{N_0}, \alpha, \mathsf{P}, d_{i,j}$ are constants – either known or measured a priori. On the other hand, $|x_{i,j}|^2$ is a random variable, assumed to be exponentially distributed[3] with normalized mean 1. We assume that $|x_{i,j}|^2$ are i.i.d. between different node pairs. We also assume that there is no temporal correlation i.e., $|x_{i,j}|^2$ is i.i.d. in different time slots between nodes i and j. We discuss how to relax this assumption in Section 2.8. The probability that $\mathsf{S}_{i,j} \geq s$ (where $s > 0$) is

$$\mathbb{P}\{\mathsf{S}_{i,j} \geq s\} = \exp\left(\frac{-s\mathsf{N_0}}{\mathsf{P}d_{i,j}^{-\alpha}}\right) \tag{2.2}$$

We model the physical layer coding scheme by assuming that a received packet can be decoded successfully when $\mathsf{S}_{i,j} \geq \beta$ for a certain threshold β. An important quantity is the *packet reception probability* that j can successfully receive the packet from i, denoted by:

$$P_{i,j} \triangleq \mathbb{P}\{\mathsf{S}_{i,j} \geq \beta\} \tag{2.3}$$

(**b**) *Cooperative Synchronized Transmissions*: We next consider a set of cooperative transmitters $T = \{i_1, ..., i_m\}$ that can synchronize their transmissions such that the signal-to-noise ratio at receiver j is the sum of the individual signal-to-noise ratios from the transmitters (see Figure 2.1 (a)). Note that j cannot belong to T. The total signal-to-noise ratio $\mathsf{S}_{T,j}$ from transmitters T to j is:

$$\mathsf{S}_{T,j} \triangleq \frac{\sum_{r \in T} |x_{r,j}|^2 \mathsf{P} d_{r,j}^{-\alpha}}{\mathsf{N_0}} \tag{2.4}$$

Since the individual signal-to-noise ratio is an exponential random variable, the total signal-to-noise ratio is the sum of exponential random variables. Let $f_{T,j}(s)$ be the probability density function of S_T, which is a convolution of functions $f_{i_1,j}(s), ..., f_{i_m,j}(s)$, defined by:

[3]In narrowband Rayleigh fading channel, the power of a signal with envelope as Rayleigh distribution is an exponential random variable [96].

$$f_{T,j}(s) \triangleq f_{i_1,j}(s) \circledast \cdots \circledast f_{i_m,j}(s) \tag{2.5}$$

where $f_{i_1,j}(s) \triangleq \frac{\mathsf{N}_0}{\mathsf{P}d_{i,j}^{-\alpha}} \exp\left(\frac{-s\mathsf{N}_0}{\mathsf{P}d_{i,j}^{-\alpha}}\right)$ is the probability density function of individual signal-to-noise ratio $\mathsf{S}_{i,j}$.

The probability that j can successfully receive the packet from a set of transmitters T is given by $P_{T,j} \triangleq \mathbb{P}\{\mathsf{S}_{T,j} \geq \beta\}$. Deriving a general formula for $P_{T,j}$ for an arbitrary set of transmitters T is challenging. Hence, we assume that $d_{i',j} \neq d_{i,j}$ for every pair of transmitters i, i' and any node receiver j. This significantly simplifies the proofs on the convolution of exponential distribution functions (see Lemma 1). This mild assumption, likely satisfied by real topologies, is useful to simplify the expression of $P_{T,j}$.

Lemma 1. *Denote $f_{\circledast m}(s)$ as the probability density function of the sum of m independent exponential random variables with distinct means $(\lambda_1, ..., \lambda_m)$.*

$$f_{\circledast m}(s) = \Big(\prod_{r=1}^{m} \lambda_r\Big) \sum_{r=1}^{m} \frac{\exp(-s\lambda_r)}{\prod_{r'=1:r'\neq r}^{m}(\lambda_{r'} - \lambda_r)} \tag{2.6}$$

Proof. See [11]. □

We remark that the general case with non-distinct values λ_r are called hypoexponential random variables [1]. There are formulas in [5,32] for hypoexponential random variables, which appear too complicated for the analysis of network-level performance. Hence, we will rely on Lemma 1 under the assumption of distinct values of λ_r.

Lemma 2. *The probability that j can successfully receive the packet from a set of transmitters T is:*

$$P_{T,j} = \sum_{r\in T} \frac{\exp\left(\frac{-\beta\mathsf{N}_0}{\mathsf{P}d_{r,j}^{-\alpha}}\right)}{\prod_{r'\in T\setminus\{r\}}\left(1 - \left(\frac{d_{r,j}}{d_{r',j}}\right)^{\alpha}\right)} \tag{2.7}$$

Proof. Based on Lemma 1, see Appendix A.1. □

(**c**) *Competing Interfering Transmissions*: Lemma 2 only considers the case of cooperative synchronized transmitters. To address the case of competing interfering

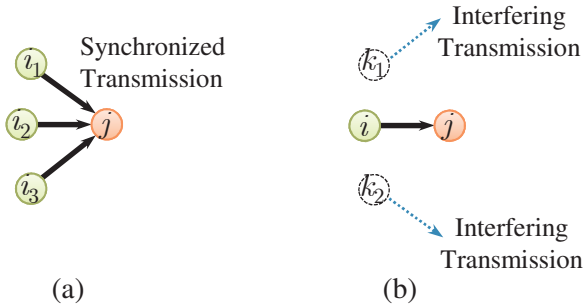


Figure 2.1. (a) Cooperative synchronized transmission (b) Competing interfering transmission

transmissions (e.g., Figure 2.1 (b)), let I be the set of simultaneously competing transmitters. The signal-to-interference-and-noise ratio $\mathsf{S}_{i,j}^{I}$ from transmitter i to receiver j in the presence of a set of interfering transmitters I is defined as:

$$\mathsf{S}_{i,j}^{I} \triangleq \frac{|x_{i,j}|^2 \mathsf{P} d_{i,j}^{-\alpha}}{\mathsf{N}_0 + \sum_{k \in I} |x_{k,j}|^2 \mathsf{P} d_{k,j}^{-\alpha}} \tag{2.8}$$

It is clear that $i \neq j$ and that i and j cannot belong to I. The probability that j can successfully receive the packet from transmitter i is given by $P_{i,j}^{I} \triangleq \mathbb{P}\{\mathsf{S}_{i,j}^{I} \geq \beta\}$, which can be obtained from the following lemma.

Lemma 3.

$$P_{i,j}^{I} = \sum_{k \in I} \frac{\exp\left(\frac{-\beta \mathsf{N}_0}{\mathsf{P} d_{i,j}^{-\alpha}}\right)}{\left(1 + \beta\left(\frac{d_{i,j}}{d_{k,j}}\right)^{\alpha}\right) \prod_{k' \in I \setminus \{k\}} \left(1 - \left(\frac{d_{k,j}}{d_{k',j}}\right)^{\alpha}\right)} \tag{2.9}$$

Proof. See Appendix A.2 for proof. □

(**d**) *Mixed Cooperative & Interfering Transmissions*: Last, we consider the general case with a set of cooperative transmitters T and a set of simultaneously competing transmitters I. The signal-to-interference-and-noise ratio $\mathsf{S}_{T,j}^{I}$ from a set of cooperative transmitters T to j in the presence of a set of interfering transmitters I is:

$$\mathsf{S}_{T,j}^{I} \triangleq \frac{\sum_{r \in T} |x_{r,j}|^2 \mathsf{P} d_{i,j}^{-\alpha}}{\mathsf{N}_0 + \sum_{k \in I} |x_{k,j}|^2 \mathsf{P} d_{k,j}^{-\alpha}} \tag{2.10}$$

Note that $T \cap I = \varnothing$ and j cannot belong to T or I. The probability that j can successfully receive the packet from a set of cooperative transmitters T in spite of

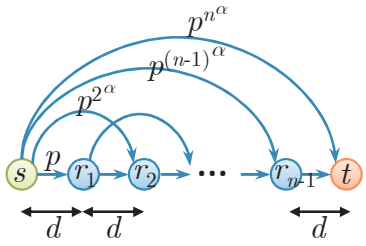


Figure 2.2. An n-hop linear network.

interfering transmitters I is given by $P^I_{T,j} \triangleq \mathbb{P}\{\mathsf{S}^I_{T,j} \geq \beta\}$, which can be obtained by the following lemma, derived using Lemmas 2-3.

Lemma 4.

$$P^I_{T,j} = \sum_{r\in T}\sum_{k\in I} \frac{\exp\left(\frac{-\beta \mathsf{N}_0}{\mathsf{P} d_{r,j}^{-\alpha}}\right)}{\left(1+\beta(\frac{d_{r,j}}{d_{k,j}})^\alpha\right) \prod\limits_{r'\in T\setminus\{r\}} \left(1-(\frac{d_{r,j}}{d_{r',j}})^\alpha\right) \prod\limits_{k'\in I\setminus\{k\}} \left(1-(\frac{d_{k,j}}{d_{k',j}})^\alpha\right)}$$

2.5 Networks with Single Flow

Section 2.4 provides single-transmission/reception models for wireless networks with random fading. We now use this model to construct simple recurrence relations for source-destination paths and compare the performance of opportunistic to cooperative forwarding strategies in a linear network. We observe that in the single flow case (where there is no network interference), the throughput provided by cooperative forwarding is greater than that provided by opportunistic forwarding. In the following, we consider the single packet case[4] - the source sends no new packet until the packet reaches the destination.

2.5.1 Opportunistic Forwarding

In Figure 2.2, we consider only one flow in a n-hop linear network, where s is the source, t is the destination, and $s = r_0, r_1, ..., r_{n-1}, r_n = t$ are the relays. Assume that

[4]We discuss the multiple packet case in Chapter 7

the distance between r_{i-1} and r_i ($\forall\, i, j$= 1 to n) in the linear network is d, and denote by $p \triangleq \exp\left(\frac{-\beta \mathsf{N_0}}{\mathsf{P}d^{-\alpha}}\right)$ the packet reception probability for a transmission over one hop. Hence, the probability that i can successfully transmit packets to j when they are n hops apart is given by:

$$P_{i,j} = \exp\left(\frac{-\beta \mathsf{N_0}}{\mathsf{P}(nd)^{-\alpha}}\right) = p^{n^\alpha} \tag{2.11}$$

For convenience of analysis, we assume α is an integer.

There are two quantities of interest. One quantity is the throughput of the linear network. Denote by $N_{\rm op}[n]$ the expected number of transmissions required by opportunistic forwarding to reach the destination from the source in a n-hop linear network. We obtain:

$$\begin{aligned} N_{\rm op}[1] &= \tfrac{1}{p} \\ N_{\rm op}[n] &= p^{n^\alpha} + \sum_{i=1}^{n-1} \prod_{j=i+1}^{n} (1-p^{j^\alpha}) p^{i^\alpha} (1 + N_{\rm op}[n-i]) \\ &\quad + \prod_{j=1}^{n} (1-p^{j^\alpha})(1 + N_{\rm op}[n]) \end{aligned} \tag{2.12}$$

To write a recursive equation for $N_{\rm op}[n]$ (2.12), we have to consider three cases: 1) With probability p^{n^α}, the source can reach the destination in one transmission. 2) With probability $\prod_{j=i+1}^{n}(1-p^{j^\alpha})p^{i^\alpha}$, the source can reach the node that is $(n-i)$ hops away from the destination in one transmission, from which the expected number of transmissions to reach the destination is $N_{\rm op}[n-i]$. 3) Otherwise, with probability $\prod_{j=1}^{n}(1-p^{j^\alpha})$, the source cannot reach any other nodes.

Because there is only a single flow, the throughput is:

$$\mathrm{T}_{\rm op}[n] = \frac{1}{N_{\rm op}[n]} \tag{2.13}$$

Another quantity of interest denoted by $H_{\rm op}[n]$ is the average number of hops traversed in one transmission, given that the destination is n hops away. We obtain the recurrence equation:

$$
\begin{aligned}
H_{\mathrm{op}}[1] &= p \\
H_{\mathrm{op}}[n] &= np^{n^\alpha} + (1 - p^{n^\alpha}) H_{\mathrm{op}}[n-1]
\end{aligned}
\tag{2.14}
$$

We can solve $H_{\mathrm{op}}[n]$ in closed form.

Lemma 5.

$$
H_{\mathrm{op}}[n] = \sum_{j=1}^{n} jp^{j^\alpha} \Big(\prod_{\ell=j+1}^{n} 1 - p^{\ell^\alpha} \Big) \tag{2.15}
$$

When $n \geq 2$,

$$
H_{\mathrm{op}}[n] = p + 2p^{2^\alpha} - p^{1+2^\alpha} + 3p^{3^\alpha} + \mathrm{O}(p^{1+3^\alpha}) \tag{2.16}
$$

Proof. See Appendix A.3. □

Theorem 1. *The throughput is upper bounded by:*

$$
\mathrm{T}_{\mathrm{op}}[n] \leq \frac{1}{n} H_{\mathrm{op}}[n] \tag{2.17}
$$

Proof. See Appendix A.4. □

In general, we observe that the upper bound is tight (see Figure 2.3).

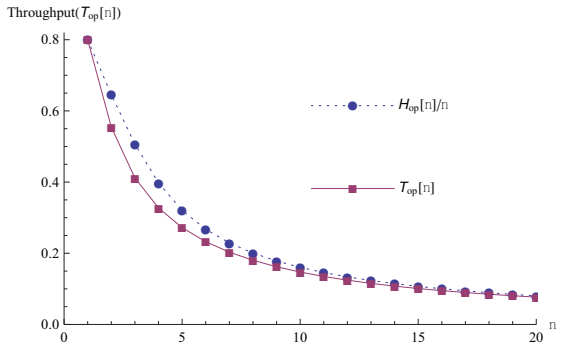


Figure 2.3. ($p = 0.8$) The throughout $\mathrm{T}_{\mathrm{op}}[n]$ is tightly bounded by $\frac{1}{n} H_{\mathrm{op}}[n]$.

2.5.2 Cooperative Forwarding

Next, we consider cooperative forwarding using all overhearing relays to transmit the packets to the destination.

We consider the idealized case of *perfect* cooperative forwarding, where we can use all the relays between the source and farthest overhearing relay in the linear network for cooperative forwarding. Thus, in Figure 2.2, if r_k overhears the packet, then we assume all relays $r_1, ..., r_{k-1}$ also overhear the packets. We aim at bounding the gap between opportunistic forwarding and cooperative forwarding. Hence, it suffices to consider perfect cooperative forwarding in order to establish an upper bound on this gap.

Assuming perfect cooperation, let $H_{\text{co}}[n]$ be the average number of hops reached in one transmission by cooperative forwarding, given that the destination is n hops away.

Theorem 2. *The expected number of hops a packet can reach in one time slot by cooperative forwarding is related to that of opportunistic forwarding by:*

$$H_{\text{co}}[n] = \text{O}(\sqrt{n}) \cdot H_{\text{op}}[n] \tag{2.18}$$

Proof. See Appendix A.5. □

In this section, we have seen that cooperative forwarding provides higher performance (at most sub-linear, i.e., $\sqrt{n}$ improvement) than opportunistic forwarding in the single-flow, linear-network case. As we will see in subsequent sections, where we consider multiple competing flows within the network, transmission interference among flows (which is *not* present in the single flow case) becomes a critical factor. This will mitigate the advantages of cooperative forwarding found in this section, suggesting that the relative advantages of opportunistic and collaborative forwarding depend strongly on network topology and assumptions about traffic flows.

2.6 Networks with Multiple Flows

Having studied the single flow case for a linear network in the previous section, we next consider a general setting with an arbitrary network topology and multiple

flows. The major difference between the single flow and the multiple flow case is the increased interference due to competing flows. We formulate Markovian models for analyzing the different forwarding strategies in the multiple flow scenario, using the packet reception probabilities from Section 2.4. Using these models, we study a simple topology, and show that opportunistic forwarding can outperform cooperative forwarding in the absence of inter-flow cooperation. For more general networks we use simulation and observe that opportunism outperforms cooperation on average. Thus, we conclude that interference mitigates the potential gains of cooperative forwarding.

2.6.1 Opportunistic Forwarding

First, we present the Markov model for opportunistic forwarding in a general network topology and multiple flows; we then evaluate this model for a simple diamond network. We denote a set of flows by $\mathscr{F}$. Each flow $f \in \mathscr{F}$ has a list of participating nodes denoted by $\mathcal{P}_f = (v_{\mathrm{s}(f)}, v_1, ..., v_{\mathrm{d}(f)})$, where $v_{\mathrm{d}(f)}$ is the destination and $v_{\mathrm{s}(f)}$ is the source. Each succeeding node in $\mathcal{P}_f$ (e.g., v_i) has a higher priority than its preceding nodes (i.e., v_j for all $j < i$) for forwarding the packet, until the packet reaches $v_{\mathrm{d}(f)}$. Formally, we denote "$v_i \succ_f v_j$" to represent that v_i has a higher priority than v_j in $\mathcal{P}_f$.

We denote the state of the network as $r \triangleq (r_f \in \mathcal{P}_f : f \in \mathscr{F})$, where r_f is the active relay for flow f for the next forwarding operation. Recall that $P^I_{T,j}$ is the packet reception probability at j from a set of cooperative transmitters T in the presence of interfering transmitters I. We denote by $\mathbf{P}_{r,r'}$ the state transition probability from state r to state r', where ($r'_f \succ_f r_f$ or $r'_f = r_f$) and $r'_f \neq v_{\mathrm{d}(f)}$, for at least one flow $f \in \mathscr{F}$. Let $r_{\neg f} \triangleq \{r_{f'} : f' \in \mathscr{F} \backslash \{f\}\}$. We obtain:

$$\mathbf{P}_{r,r'} \triangleq \prod_{f \in \mathscr{F}} P^{r_{\neg f}}_{r_f, r'_f} \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r'_f} (1 - P^{r_{\neg f}}_{r_f, v}) \tag{2.19}$$

Namely, $\mathbf{P}_{r,r'}$ is the packet reception probability that every flow f can receive a packet from r_f to r'_f, subject to the condition that the set of succeeding nodes $\{v \in \mathcal{P}_f : v \succ_f r'_f\}$ that cannot receive the packet.

Recall that we assume that the flow's source transmits a new packet after the successful delivery of a packet to the the flow's destination. Therefore when a flow reaches state $r_f = v_{\mathrm{d}(f)}$ (i.e., the packet reaches the destination), the state transition in the next time step will correspond to the states reachable from the source with their respective probabilities.

We remark that Eqn. (2.19) contains two prohibited cases: 1) two packets cannot be received by the same receiver at the same time; and 2) a node cannot be receiving and transmitting at the same time. Because we assume $\beta > 1$, Eqn. (2.19) will give zero transition probability for the above two cases.

The stochastic behavior of the network is characterized by the Markov chain defined by state transition probability $\mathbf{P}_{r,r'}$ for every pair of states (r, r'). We then can evaluate the stationary distribution $\pi(r)$ for each state of the network r, which satisfies the following balance equation:

$$\sum_{r''} \pi(r) \cdot \mathbf{P}_{r'',r} = \sum_{r'} \pi(r') \cdot \mathbf{P}_{r,r'} \tag{2.20}$$

subject to $\sum_r \pi(r) = 1$.

The throughput of each flow f, $\mathrm{T}_{\mathrm{op}}(f)$, is given by:

$$\mathrm{T}_{\mathrm{op}}(f) = \sum_{r: r_f = v_{\mathrm{d}(f)}} \pi(r) \tag{2.21}$$

2.6.2 Cooperative Forwarding

For cooperative forwarding, we denote the state of the network as $R \triangleq (R_f \subseteq \mathcal{P}_f : f \in \mathscr{F})$, where R_f is a set of cooperative transmitters of flow f. Let $R_{\neg f} \triangleq \bigcup_{f' \in \mathscr{F} \setminus \{f\}} R_{f'}$. The state transition probability $\mathbf{P}_{R,R'}$, where $R_f \subseteq R'_f$ for at least one $f \in \mathscr{F}$ and $v_{\mathrm{d}(f)} \notin R_f$, is given by:

$$\mathbf{P}_{R,R'} \triangleq \prod_{f\in\mathscr{F}} \prod_{v\in R'_f\backslash R_f} P_{R_f,v}^{R_{\neg f}} \cdot \prod_{v'\in\mathcal{P}_f\backslash R'_f} \left(1 - P_{R_f,v'}^{R_{\neg f}}\right) \tag{2.22}$$

Similarly, Eqn. (2.22) also contains the prohibited cases, to ensure that (1) two packets cannot be received by the same receiver at the same time; and (2) a node cannot be receiving and transmitting at the same time.

The stationary distribution is denoted by $\pi(R)$, and the throughput of flow f, $\mathrm{T}_{\mathrm{co}}(f)$, is given by:

$$\mathrm{T}_{\mathrm{co}}(f) = \sum_{R,R':v_{\mathrm{d}(f)}\in R'_f} \pi(R)\cdot\mathbf{P}_{R,R'} \tag{2.23}$$

2.6.3 Selective Cooperative Forwarding

Selective cooperative forwarding only assigns the two closest nodes to the destination that currently have a copy of the packet as relays . We again denote the state of the network as R, where $R_f \subseteq \mathcal{P}_f$ is a set of potential transmitters of flow f that have received the packet. The two nodes $r_1, r_2 \in R_f$ are selected, such that $r_1 \succ_f r_2 \succ_f r$ for all $r \in R_f\backslash\{r_1, r_2\}$. Hence, we denote the two selected relays by a set $r(R_f)$.

Let $I_f(r(R)) \triangleq \bigcup_{f'\in\mathscr{F}\backslash\{f\}} r(R_{f'})$. The state transition probability $\mathbf{P}_{R,R'}$, where $R_f \subseteq R'_f$ for at least one $f \in \mathscr{F}$ and $v_{\mathrm{d}(f)} \notin R_f$, is given by:

$$\mathbf{P}_{R,R'} \triangleq \prod_{f\in\mathscr{F}} \prod_{v\in R'_f\backslash R_f} P_{r(R_f),v}^{I_f(r(R))} \cdot \prod_{v'\in\mathcal{P}_f\backslash R'_f} \left(1 - P_{r(R_f),v'}^{I_f(r(R))}\right)$$

2.6.4 Analysis of a Simple Diamond Network

Using the Markov models defined in Secs. 2.6.1 and 2.6.2, we compare the performance of opportunistic forwarding and cooperative forwarding with two flows in the diamond network depicted in Figure 2.4. There are two flows: $s_1 \to t_1$ and $s_2 \to t_2$. Relay r can contribute to either flow, depending on if it can overhear the flow. We assume $\beta > 1$, and by Eqn. (2.10) a node can receive a packet from only one flow at a time.

The forwarding operations for opportunistic forwarding and cooperative forwarding respectively generate the Markov chains in Figure 2.5. Each state corresponds to

a subset of transmitters that can forward the packet. In opportunistic forwarding, relay r can forward the packet for a flow during a time slot, provided that it received a packet previously. In cooperative forwarding, both source and relay will participate in forwarding. Hence, both Markov chains have the same structure, but different state transition probabilities.

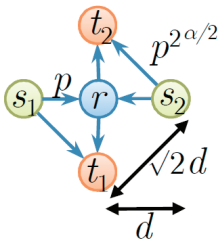
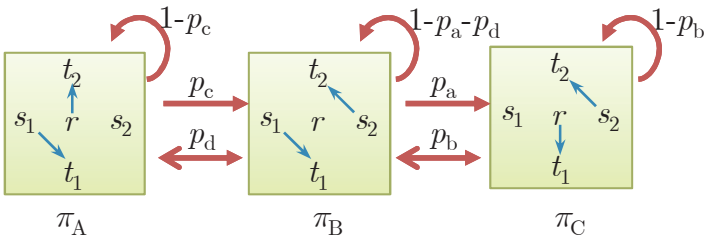
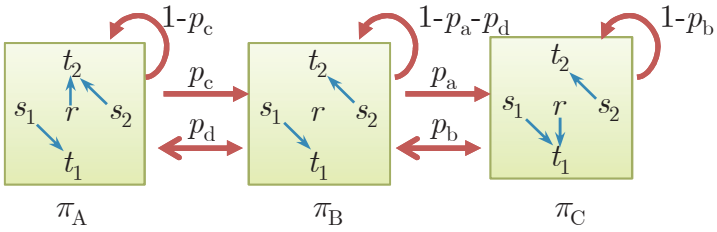


Figure 2.4. Simple diamond network



(a) Opportunistic



(b) Cooperative

Figure 2.5. Markov chains for the forwarding operations in a diamond network.

(**a**) *Opportunistic Forwarding*: The stationary distribution is:

$$\pi_A = \frac{p_b p_d}{p_a p_c + p_b p_c + p_b p_d}, \quad \pi_B = \frac{p_b p_c}{p_a p_c + p_b p_c + p_b p_d} \\ \pi_C = \frac{p_a p_c}{p_a p_c + p_b p_c + p_b p_d} \tag{2.24}$$

where the state transition probabilities can be expressed in terms of the packet reception probabilities:

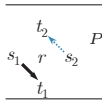
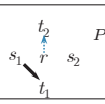
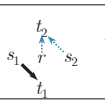
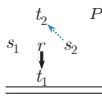
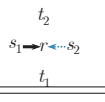
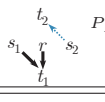
 $P^{s_2}_{s_1,t_1} = \frac{p^{2^{\frac{\alpha}{2}}}}{(1+\beta)}$	 $P^{r}_{s_1,t_1} = \frac{p^{2^{\frac{\alpha}{2}}}}{(1+\beta\cdot 2^{\frac{\alpha}{2}})}$	 $P^{\{s_2,r\}}_{s_1,t_1} = \frac{p^{2^{\frac{\alpha}{2}}}}{(1+\beta)(1+2^{\frac{\alpha}{2}})}$
 $P^{s_2}_{r,t_1} = \frac{p}{(1+\beta\cdot 2^{-\frac{\alpha}{2}})}$	 $P^{s_2}_{s_1,r} = \frac{p}{(1+\beta)}$	 $P^{s_2}_{\{s_1,r\},t_1} = \frac{p^{2^{\frac{\alpha}{2}}}}{(1+\beta)(1-2^{\frac{\alpha}{2}})} + \frac{p}{(1+\beta\cdot 2^{-\frac{\alpha}{2}})(1-2^{-\frac{\alpha}{2}})}$

Table 2.1. Packet reception probabilities for transmissions (without interference) in bold arrows, for transmissions (with interference) in dashed arrows.

$$\begin{aligned} p_{\mathrm{a}} &= (1 - P^{s_2}_{s_1,t_1})P^{s_2}_{s_1,r} \qquad & p_{\mathrm{d}} &= p_{\mathrm{a}} \\ p_{\mathrm{b}} &= P^{s_2}_{r,t_1} \qquad & p_{\mathrm{c}} &= p_{\mathrm{b}} \end{aligned} \tag{2.25}$$

In Table 2.1, we compute all the packet reception probabilities using Lemmas 2-4.

Consider $\alpha = 2$. The throughput of flow $s_1 \to t_1$ using opportunistic forwarding is given by:

$$\begin{aligned} \mathrm{T}^{s_1\to t_1}_{\mathrm{op}} &\triangleq \pi_{\mathrm{A}} P^{r}_{s_1,t_1} + \pi_{\mathrm{B}} P^{s_2}_{s_1,t_1} + \pi_{\mathrm{C}} P^{s_2}_{r,t_1} \\ &= \frac{p^2\big(p\big(2+3\beta+\beta^2\big)+4\big(1+3\beta+2\beta^2\big)-p^3(2+\beta)-p^2(2+4\beta)\big)}{2(1+2\beta)((1+\beta)^2-p^3(2+\beta)+p(2+3\beta+\beta^2))} \end{aligned} \tag{2.26}$$

(**b**) *Cooperative Forwarding*: The state transition probabilities can be expressed in terms of the packet reception probabilities by Lemmas 2-4 (see Table 2.1):

$$\begin{aligned} p_{\mathrm{a}} &= (1 - P^{s_2}_{s_1,t_1})P^{s_2}_{s_1,r} \qquad & p_{\mathrm{d}} &= p_{\mathrm{a}} \\ p_{\mathrm{b}} &= P^{s_2}_{\{s_1,r\},t_1} \qquad & p_{\mathrm{c}} &= p_{\mathrm{b}} \end{aligned} \tag{2.27}$$

The throughput of flow $s_1 \to t_1$ using cooperative forwarding is given by:

$$\begin{aligned} \mathrm{T}^{s_1\to t_1}_{\mathrm{co}} &\triangleq \pi_{\mathrm{A}} P^{\{r,s_2\}}_{s_1,t_1} + \pi_{\mathrm{B}} P^{s_2}_{s_1,t_1} + \pi_{\mathrm{C}} P^{s_2}_{\{s_1,r\},t_1} \\ &= \frac{p^2\left(-3+2p^2-4\beta\right)}{(1+2\beta)\left(2p^2-3(1+\beta)\right)} \end{aligned} \tag{2.28}$$

Theorem 3. *The throughput of opportunistic forwarding is* superior *to that of cooperative forwarding:*

$$\mathrm{T}^{s_1\to t_1}_{\mathrm{op}} \geq \mathrm{T}^{s_1\to t_1}_{\mathrm{co}} \tag{2.29}$$

Proof. See the Appendix A.6. □

We also plot the throughput of a flow using opportunistic forwarding and cooperative forwarding in the case of two competing flows in Figure 2.6. This corroborates our intuition that cooperation can degrade performance due to the increased amount of interference generated by the larger number of simultaneous transmitters.

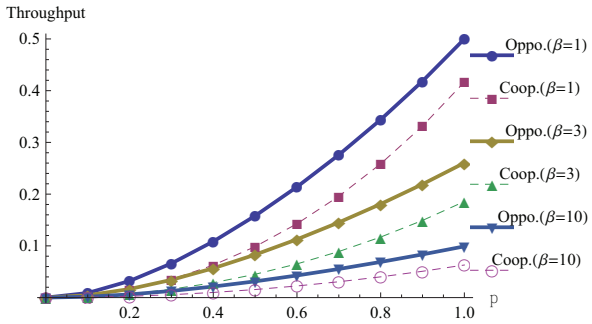


Figure 2.6. Throughput of opportunistic forwarding and cooperative forwarding in a diamond network.

2.6.5 Simulations for Random Networks

Having studied a simple diamond network via a Markovian analysis, we next use simulation to study larger network settings. As we will see, the insights gained in the small scale setting generally apply in this more general setting. We consider 50 nodes uniformly distributed in a 100×100 area. We select 4 distinct source-destination pairs (referred to as a 'configuration') at random from the 50 nodes. We simulate the link quality between different pairs of nodes for every time slot using the Rayleigh fading channel model. The simulation begins by all 4 sources transmitting packets. A node is able to receive a packet if the SINR between the transmitter and itself is above a threshold. The opportunistic and cooperative routing protocols govern the nodes that transmit packets in the next time slot. When a packet reaches the corresponding destination, the source starts transmitting a new packet. We conduct this simulation for 5000 time slots and keep track of the number of packets received at the destination to calculate the throughput. We refer to the simulation of a given 'configuration' as a 'run'.

Recall that our earlier results revealed that interference among different flows played an important factor in determining performance. Thus, when presenting throughput comparisons in this section, we would like to quantify such interference among packets flowing from source to destination along "paths" between given sets of source-destination pairs. Note, however, that there is no well-defined notion of a deterministic "path" along which packets flow for either opportunistic or cooperative forwarding. Thus we characterize the level of interference among flows by taking 10 points equidistantly-spaced between a flow's source and destination for each flow. Let $\mathcal{L}(f)$ denote the set of these 10 points for flow f. We consider the distance between all pairs of such points i, j in the following manner:

$$\mathsf{Intf\text{-}Metric} \triangleq \sum_{f \in \mathcal{F}} \sum_{f' \in \mathcal{F} \setminus \{f\}} \sum_{i \in \mathcal{L}(f)} \sum_{j \in \mathcal{L}(f')} d_{i,j}^{-\alpha} \tag{2.30}$$

Intf-Metric provides a coarse measure of the interference (as caused by the nearness of potentially interfering nodes for different flows). Higher value of Intf-Metric indicates a greater amount of interference in the network. In Figure 2.7 we plot the difference in throughput between the opportunistic and cooperative strategies versus the Intf-Metric for a β of 4. The figure is obtained by conducting 100 'runs', each time with a different 'configuration'. Points above the line drawn at Throughput Difference=0 indicate the cases where opportunism performs better than cooperation while the points below the line depict the opposite. The results in Figure 2.7 indicate that on average the performance of opportunism is better than that of cooperation and this is true for a wide range of Intf-Metric values.

2.7 Fixed-point Model

Section 2.6 provided a Markovian model of multiple flows in a general network setting. However, the number of states increases exponentially with the number of flows in this model. Hence, evaluating the stationary distribution of the model quickly becomes intractable. Thus, in this section we introduce a fixed-point model

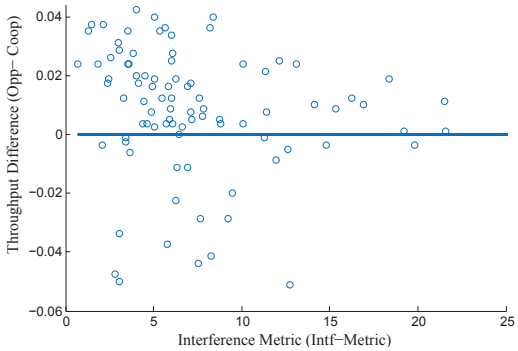


Figure 2.7. Throughput difference between opportunism and cooperation in random networks.

to simplify the evaluation. We will show that the throughput obtained from the fixed-point model is a lower bound to the actual throughput of the Markov model. In order to establish this model as a lower bound, we assume that the sets of participating nodes among distinct flows are disjoint (i.e., $\mathcal{P}_f \cap \mathcal{P}_{f'} = \varnothing$).

Our approach is to model each flow independently and capture their dependence by accounting for interference between flows within each flow model. This gives rise to a set of fixed-point equations for the stationary distributions, which can be obtained via efficient iterative methods.

2.7.1 Opportunistic Forwarding

The state of a flow $f \in \mathscr{F}$ is specified by the active relay $r \in \mathcal{P}_f$ that overhears the transmission and has the highest priority among the overhearing participating nodes. We next describe a set of fixed-point equations for individual flows.

First, suppose that the stationary distribution of a node $j \in \mathcal{P}_f$ to become an active relay is given by $\hat{\pi}_f(j)$. Then, the expected interference to j from all other flows w.r.t. stationary distributions $\hat{\pi}_{-f} \triangleq \{\hat{\pi}_{f'} : f' \in \mathscr{F}\backslash\{f\}\}$ becomes:

$$\begin{aligned} \hat{I}_j^f(\hat{\pi}_{-f}) &= \sum_{f' \in \mathscr{F}\backslash\{f\}} \sum_{i \in \mathcal{P}_{f'}} \hat{\pi}_{f'}(j) \cdot \mathbb{E}[|x_{i,j}|^2] \cdot \mathsf{P} d_{i,j}^{-\alpha} \\ &= \sum_{f' \in \mathscr{F}\backslash\{f\}} \sum_{i \in \mathcal{P}_{f'}} \hat{\pi}_{f'}(j) \cdot \mathsf{P} d_{i,j}^{-\alpha} \end{aligned} \tag{2.31}$$

where the fading coefficient $|x_{i,j}|^2$ is an i.i.d. exponential random variable with normalized mean 1. Note that $\hat{I}_j^f(\hat{\pi})$ does not depend on $\hat{\pi}_f$, but only on $\{\hat{\pi}_{f'} : f' \in \mathscr{F}\backslash\{f\}\}$.

Suppose that the interference from other flows remains stationary and has distribution $\hat{\pi}_{\neg f}$. Then the packet reception probability that j can successfully receive the packet from i in flow f w.r.t. $\hat{\pi}_{\neg f}$ is given by:

$$\hat{P}_{i,j}^f(\hat{\pi}_{\neg f}) \triangleq \mathbb{P}\Big\{\frac{|x_{i,j}|^2 \mathsf{P} d_{i,j}^{-\alpha}}{\mathsf{N}_0 + \hat{I}_j^f(\hat{\pi}_{\neg f})} \geq \beta\Big\} = \exp\Big(\frac{-\beta\big(\mathsf{N}_0 + \hat{I}_j^f(\hat{\pi}_{\neg f})\big)}{\mathsf{P} d_{i,j}^{-\alpha}}\Big) \tag{2.32}$$

Next, we focus on the Markov model of an individual flow. In such a model, we denote by $\hat{\mathbf{P}}_{r,r'}^f(\hat{\pi}_{\neg f})$ the state transition probability from an active relay $r \in \mathcal{P}_f$ to another active relay $r' \in \mathcal{P}_f$ such that ($r' \succ_f r$ or $r' = r$) and $r \neq v_{\mathrm{d}(f)}$, w.r.t. $\hat{\pi}_{\neg f}$:

$$\hat{\mathbf{P}}_{r,r'}^f(\hat{\pi}_{\neg f}) \triangleq \hat{P}_{r,r'}^f(\hat{\pi}_{\neg f}) \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r'} \Big(1 - \hat{P}_{r,v}^f(\hat{\pi}_{\neg f})\Big) \tag{2.33}$$

Denote the stationary distribution of this model as $\hat{\pi}_f$. It satisfies the following balance equations for all $r \in \mathcal{P}_f$:

$$\sum_{r'' \in \mathcal{P}_f} \hat{\pi}_f(r) \cdot \hat{\mathbf{P}}_{r'',r}^f(\hat{\pi}_{\neg f}) = \sum_{r' \in \mathcal{P}_f} \hat{\pi}_f(r') \cdot \hat{\mathbf{P}}_{r,r'}^f(\hat{\pi}_{\neg f}) \tag{2.34}$$

subject to $\sum_{v \in \mathcal{P}_f} \hat{\pi}_f(v) = 1$.

Eqns. (2.31)-(2.34) form a set of fixed-point equations for $(\hat{\pi}_f : f \in \mathscr{F})$. Solving the fixed-point $(\hat{\pi}_f : f \in \mathscr{F})$ can be achieved by an iterative method. We first assume a certain distribution $(\hat{\pi}_f^0 : f \in \mathscr{F})$. Then we obtain $\hat{\pi}_f^1$ from Eqns. (2.31)-(2.34) w.r.t. $\hat{\pi}_{\neg f}^0$, for all $f \in \mathscr{F}$. We repeat the process for t steps, until $\hat{\pi}_f^t$ has a small deviation from $\hat{\pi}_f^{t-1}$.

The throughput of the fixed-point model is defined by:

$$\hat{\mathrm{T}}_{\mathrm{op}}(f) = \hat{\pi}_f(v_{\mathrm{d}(f)}) \tag{2.35}$$

Theorem 4. *The throughput obtained from the fixed-point model is a lower bound to the actual throughput of the Markov model:*

$$\mathrm{T}_{\mathrm{op}}(f) \geq \hat{\mathrm{T}}_{\mathrm{op}}(f) \tag{2.36}$$

Proof. See the Appendix A.7. □

2.7.2 Cooperative Forwarding

The fixed-point model for cooperative forwarding is similar to that of opportunistic forwarding. But the state of a flow is specified by the set of cooperative transmitters $R \subseteq \mathcal{P}_f$. By Lemma 2, the packet reception probability that j can successfully receive the packet from the set of cooperative transmitters R in a flow f is given by:

$$\hat{P}^f_{R,j}(\hat{\pi}_{\neg f}) = \sum_{r \in R} \frac{\exp\left(\frac{-\beta(\mathsf{N}_0 + \hat{I}^f_j(\hat{\pi}_{\neg f}))}{\mathsf{P} d^{-\alpha}_{r,j}}\right)}{\prod_{r' \in R \setminus \{r\}} \left(1 - \left(\frac{d_{r,j}}{d_{r',j}}\right)^{\alpha}\right)} \tag{2.37}$$

Note that the state of flow is R, a subset of cooperative transmitters. Then the stationary distribution of a node $j \in \mathcal{P}_f$ is given by:

$$\hat{\pi}_f(j) = \sum_{R \subseteq \mathcal{P}_f : j \in R} \hat{\pi}_f(R) \tag{2.38}$$

In the Markov model of an individual flow f, the state transition probability $\mathbf{P}_{R,R'}$ from $R \subseteq \mathcal{P}_f$ to $R' \subseteq \mathcal{P}_f$ such that $R \subseteq R'$ and $v_{\mathrm{d}(f)} \notin R$, is defined by:

$$\hat{\mathbf{P}}^f_{R,R'}(\hat{\pi}_{\neg f}) \triangleq \prod_{v \in R' \setminus R} \hat{P}^f_{R,v}(\hat{\pi}_{\neg f}) \cdot \prod_{v' \in \mathcal{P}_f \setminus R'} \left(1 - \hat{P}^f_{R,v'}(\hat{\pi}_{\neg f})\right) \tag{2.39}$$

To solve the fixed-point $(\hat{\pi}_f : f \in \mathscr{F})$, we rely on a similar iterative approach as for the case of opportunistic forwarding. The throughput obtained from the fixed-point model can be shown as a lower bound to the actual throughput of the Markov model, using the same argument as in Theorem 4.

2.7.3 Selective Cooperative Forwarding

The case of selective cooperative forwarding is similar to basic cooperative forwarding, except with a modification to consider the two best relays instead all relays. Specifically, let the two best relays be $r_1, r_2 \in R$ for flow f, such that $r_1 \succ_f r_2 \succ_f r$ for all $r \in R \backslash \{r_1, r_2\}$. We denote the two selected relays by a set $r_f(R)$. The packet reception probability that j can successfully receive the packet from the set of cooperative transmitters R in a flow f is given by $P^{\hat{\pi}}_{r_f(R),j}$. And the stationary distribution of a node $j \in \mathcal{P}_f$ is given by:

$$\hat{\pi}_f(j) = \sum_{R \subseteq \mathcal{P}_f : j \in r_f(R)} \hat{\pi}_f(R) \tag{2.40}$$

The state transition probability $\mathbf{P}_{R,R'}$ from $R \subseteq \mathcal{P}_f$ to $R' \subseteq \mathcal{P}_f$ such that $R \subseteq R'$ and $v_{\mathrm{d}(f)} \notin R$, is defined by:

$$\hat{\mathbf{P}}^f_{R,R'}(\hat{\pi}_{\neg f}) \triangleq \prod_{v \in R' \backslash R} \hat{P}^f_{r_f(R),v}(\hat{\pi}_{\neg f}) \cdot \prod_{v' \in \mathcal{P}_f \backslash R'} \left(1 - \hat{P}^f_{r_f(R),v'}(\hat{\pi}_{\neg f})\right) \tag{2.41}$$

To solve the fixed-point $(\hat{\pi}_f : f \in \mathscr{F})$, we rely on a similar iterative approach as for the case of opportunistic forwarding. The throughput obtained from the fixed-point model can be shown as a lower bound to the actual throughput of the Markov model, using the same argument as in Theorem 4.

2.7.4 Comparison of Fixed Point and Simulation

We compare the performance of our models with the simulation results. For these simulations we consider a 5×5 grid topology and consider opportunistic forwarding and selective cooperative forwarding. The simulation procedure is similar to the one outlined in Section 2.6.5. For the model we iteratively solve the fixed point equations in Section 2.7.1 and 2.7.3 for the opportunistic and selective cooperative forwarding strategies. Results in Figure 2.8 are obtained considering 5 parallel flows, each moving vertically downwards in the grid. Flows are given ids ranging from 1

to 5 starting from one end of the grid to the other. As expected, we find that for both schemes Flows 1 and 5 have maximum and comparable throughput as they experience the minimum amount of interference from other flows. Flow 3 has the minimum throughput because it is situated in the middle and experiences maximum interference. Moreover the throughput of opportunism is greater than cooperation. This is primarily due to increased interference in the cooperative case because of greater number of transmitters. We note that although the throughput obtained by our fixed-point model is lower than that obtained by simulation the relative ordering between opportunistic and cooperative forwarding is preserved.

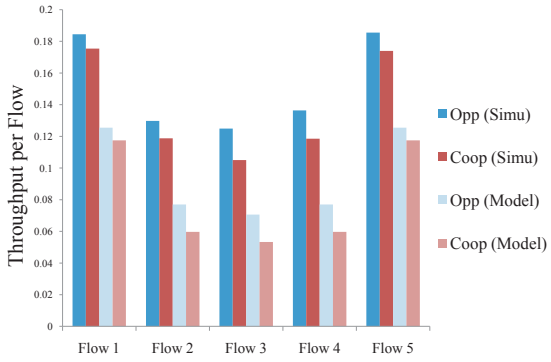


Figure 2.8. Comparison between simulation and fixed-point model

2.8 Discussion

In this chapter so far, we assumed that fading is i.i.d. (uncorrelated) in different time slots. This assumption will hold true only for fast fading where the coherence time is smaller than the duration of a time slot. In the case of slow fading, where the i.i.d. assumption will not hold, cooperation is likely to have additional benefits over opportunism because of multiple transmitters.

References [108, 109] take into account fading correlation and show how a Markov model for a block error process is a good approximation. The authors assume a flat fading channel modeled as a complex Gaussian process (x_{ij}) ($|x_{ij}|^2$ is the Rayleigh fading coefficient). The fading correlation is modeled in a standard fashion as a modified Bessel function of the first kind and zeroth order. They consider the packet

reception model similar to ours (i.e., $SNR > \beta$). They then show using information theory that a Markov model for a success/failure process in case of packet transmission is a good approximation. We denote the parameters of the Markov chain as $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. The authors calculate expressions for $a,\ b,\ c,\ d$ [108, 109]. Here c denotes the $P[success|failure]$ (Equation 47 in [109]) and this is calculated by taking fading correlation into account. Let c_1, c_2 and c_3 denote the probability of successfully receiving a packet by a node located 1-hop , 2-hop and 3-hop away from the transmitter respectively (considering correlated fading), given that the previous transmission to that node was a failure.

We next analytically show how to model greedy opportunistic forwarding considering correlated fading for a simple linear network (the approach can be easily generalized to N-hop linear networks). Modeling cooperative forwarding (as opposed to opportunistic forwarding) considering correlated fading as a Markov chain may not be possible (it is just not sufficient to know whether the previous transmission was successful or not; it is necessary to know the signal strength values of the previous transmissions). Additionally, one has to consider the summation of two complex random variables and it is not apparently clear what that sum should be. We leave the problem of analyzing cooperative forwarding for correlated fading as future work.

2.8.1 Greedy Opportunistic Forwarding Under Correlated Fading

Let us consider a simple linear network of four nodes similar to Figure 2.2. When we assume that fading is i.i.d. among the different time slots, we can model opportunistic forwarding using a Markov chain. There are three states in the Markov chain namely $A = \{1, 0, 0, 0\}$, $B = \{0, 1, 0, 0\}$ and $C = \{0, 0, 1, 0\}$ where 1 denotes that a node has a packet and 0 denotes that the node does not have a packet. We do not need a fourth state denoting that the packet is received at destination t because in the next time slot, s will transmit a new packet and hence one can assume a transition

to state A if t receives the packet. Therefore, if there are n nodes in the network the state space will be $n-1$.

When we relax the i.i.d. assumption, the state space will increase to $2(n-1)$. Each state will now be split into two states. Let us consider state A. We split this one state into two states $A_1 = \{1,0,0,0\}$ and $A_2 = \{1^*,0,0,0\}$. Each node can effectively be in three states:

- 0 denotes that the node does not have a packet.
- 1 denotes that a node is going to transmit a packet for the first time, i.e., it will receive an independent fade to all nodes downstream.
- 1^* denotes that a node transmitted in the previous time slot and none of the nodes downstream received the packet, i.e., it has a bad fade to all nodes downstream.

Note that this simple classification is sufficient. For example, if node s transmits in a time slot and one of the nodes downstream (say node r_2) receives the packet, then node s *will not* transmit in the next time slot (greedy opportunism). The states of the Markov Chain are $A_1 = \{1,0,0,0\}$, $A_2 = \{1^*,0,0,0\}$, $B_1 = \{0,1,0,0\}$, $B_2 = \{0,1^*,0,0\}$, $C_1 = \{0,0,1,0\}$, $C_2 = \{0,0,1^*,0\}$. We can now write the transition matrix P_M as

$$\begin{pmatrix} p_3 & q_1q_2q_3 & p_1q_2q_3 & 0 & p_2q_3 & 0 \\ c_3 & (1-c_1)(1-c_2)(1-c_3) & c_1(1-c_2)(1-c_3) & 0 & c_2(1-c_3) & 0 \\ p_2 & 0 & 0 & q_1q_2 & p_1q_2 & 0 \\ c_2 & 0 & 0 & (1-c_1)(1-c_2) & c_1(1-c_2) & 0 \\ p_1 & 0 & 0 & 0 & 0 & q_1 \\ c_1 & 0 & 0 & 0 & 0 & 1-c_1 \end{pmatrix} \tag{2.42}$$

where p_1, p_2 and p_3 denote the 1-hop , 2-hop and 3-hop packet reception probabilities respectively, considering i.i.d. fading while q_1, q_2 and q_3 are corresponding probabilities of the packet not being received. Let π denote the steady state distribution of

the Markov chain. The throughput T is calculated as $T = p_3\pi_{A_1} + c_3\pi_{A_2} + p_2\pi_{B_1} + c_2\pi_{B_2} + p_1\pi_{C_1} + c_1\pi_{C_2}$.

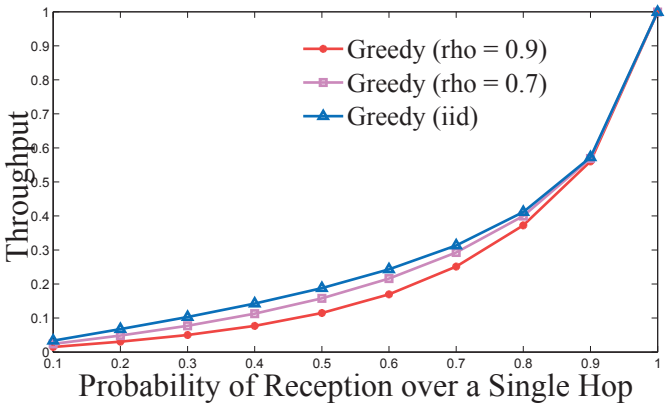


Figure 2.9. Comparing greedy opportunistic routing for correlated and i.i.d. fading

Figure 2.9 shows the comparison of greedy opportunistic routing for correlated and uncorrelated fading. In the figure, *rho* denotes the correlation coefficient between successive fading samples. We observe from the figure that the throughput of opportunistic routing for correlated fading is lower than that for uncorrelated fading. This result corresponds with our intuition - in greedy opportunistic routing, it is always the case that among the nodes which have a copy of a packet, the one closest to the destination transmits the packet. If a transmitting node (say s) has a bad channel to nodes closer to the destination than itself, then the transmission will be unsuccessful and s will have to retransmit the packet. But in case of correlated fading, unlike the uncorrelated channel case, the channel is likely to remain bad in the future as well, with the effect that the future transmissions will also be unsuccessful. When s has a good channel to the nodes closer to the destination than itself, the transmission is likely to be successful. But in the next time slot, s will not get the chance to transmit again as some other node closer to the destination will transmit the packet. As greedy opportunistic routing fails to take advantage of good channel conditions, the its performance is poorer for correlated fading in comparison to uncorrelated fading.

2.9 Conclusion

In this chapter, we used modeling and analysis to investigate the performance benefits of using opportunism and cooperation in wireless networks assuming i.i.d. fading in successive time slots. Our goal here was to compare the performance of idealized and representative opportunistic and cooperative forwarding strategies using generic models and under common realistic assumptions. We began with a single flow linear network, and observed that cooperation outperforms opportunism. We then considered the case of more general network topologies with multiple flows and observed that unlike the linear network case, opportunism outperformed cooperation on average (assuming uncorrelated fading channels). We identified the interference resulting from the larger number of transmissions under cooperative forwarding as a cause for mitigating the potential gains achievable with cooperative forwarding.

CHAPTER 3

OPTIMIZING CONTROL OVERHEAD FOR POWER-AWARE ROUTING

3.1 Introduction

The overhead of gathering state/control information (e.g., link states, node locations, queue lengths) can be significant in a mobile ad-hoc wireless network (MANET) when bandwidth is limited and network structure and state may change frequently. In such dynamic scenarios, it may still be advantageous to collect state information, provided that this information leads to better decisions that more than compensate for the additional overhead incurred. For example, the decrease in available path bandwidth as a result of state gathering overhead may be more than compensated for by the choice of better paths for routing data packets. Efficient bandwidth use is not the only metric of concern in ad hoc networks; since nodes are typically battery powered, minimizing power consumption is also important.

Understanding the tradeoff between the cost incurred in state information collection in a network and the resulting performance is a fundamental, yet largely unexplored problem. In this chapter, we analyze this tradeoff between the amount of state information collected (at what precision?, how often?) and overhead incurred, and the resulting performance in wireless networks while providing goodput guarantees. We develop an information-theoretic, bounding approach to analyze the tradeoff between the amount of signaling overhead incurred in path selection in a MANET with time-varying wireless channels and the application-level goodput and end-to-end power expended on the selected path.

We consider a network of n nodes with multiple source-destination pairs. We assume each source has m disjoint paths to the destination with k links on each path and that time is divided into intervals. At the beginning of every interval, each source collects 'noisy' estimates about the links in the network. By 'noise' we refer to the quantization error arising from finite precision representation of link states. The link state estimates in our model characterize the (time-varying) effect of shadowing on the received power.

We use the information-theoretic rate-distortion approach to quantify the noise in the link measurements - as we use more bits to encode time-varying link state, the fidelity of the estimates increase, but the control overhead also increases. Moreover, we assume each source also desires to achieve a fixed amount of goodput, which is defined as the total throughput (including control and data) minus the control overhead. The source selects a path i among the m paths such that the expected power consumed in that interval is minimized. The problem we address can be then stated in the following manner.

At what time granularity should links be sampled and at what rate (bits) should link values be encoded such that the expected power consumed in any interval is minimized subject to a fixed source-to-destination goodput constraint? We formulate an optimization problem that provides a numerically computable solution to these questions. The optimization problem takes as input the desired goodput, and leverages the distribution and autocorrelation of the shadowing process to determine the optimum value of the sampling interval and the number of bits per sample such that minimum power (for data and control) is consumed. Our optimization problem is solved off-line and provides network operators a tool for determining optimal operating points (state update frequency, number of bits per sample).

As expected, our evaluation quantitatively demonstrates that short sampling intervals incur significant overhead while long intervals fail to take advantage of the temporal correlation in link state. We also observe that using a small number of

bits per sample does not provide sufficient information about the network while using too many bits provides little additional information at the expense of increased overhead. Additionally, we simulate a network with varying link states and compare the performance of the numerical and simulation results.

The rest of the chapter is organized as follows. We discuss related work in Section 3.2 and provide a brief overview of rate-distortion in Section 3.3. In Sections 3.4 and 3.5 we describe our network model and the optimization problem respectively. We then provide a solution for the optimization problem in Sections 3.6 and 3.7. We present the numerical and simulation results in Sections 3.8 and 3.9 respectively and finally conclude the chapter in Section 3.10.

3.2 Related Work

Theoretical studies characterizing the overhead of routing protocols in MANETs has been done by Abouzeid *et. al* [93, 97, 104, 105]. Zhou and Abouzeid [105] mathematically analyze the overhead of reactive routing protocols and estimate the overhead associated with route discovery and route failure. They validate their numerical results via simulations of regular and random topologies. Information-theoretic techniques have been used to obtain lower bounds on memory requirements and routing overhead for hierarchical proactive routing in mobile ad hoc networks in [104]. The tradeoff between network properties such as connectivity, unpredictability and resource contention and state (control or data or both) information collection has been studied by Manfredi *et. al* [60].

Our work is closest to [93] where the authors use rate-distortion techniques for analyzing the protocol overhead of link state MANET routing. They derive lower bounds on the minimum bit rate at which a node must receive link state information in order to route data packets with a guaranteed delivery ratio. We differ from the above mentioned works because we consider the path selection problem and analyze the tradeoff between the signaling overhead (state update frequency and the number

of bits per sample) and power consumption in time-varying channels while providing goodput guarantees.

Most prior work has adopted simulation-based techniques to study the overhead of routing protocols in mobile wireless networks [21, 40]. Simulation has been used to study the performance of AODV and OLSR protocols in both VANETs [40] and MANETs [21]. Viennot *et. al* [22] perform a simple analysis of the control traffic for reactive and proactive protocols in MANETs considering parameters such as the average degree per node, the average number of routes created/sec and then compare analytical and simulation results for AODV, DSR and OLSR.

Power consumption in wireless networks is also a well explored field [12,56]. In [12] the authors consider the problem of joint routing, scheduling and power control in wireless networks and provide an approximate algorithm with performance guarantees to address it. Liu *et.al* [56] study the optimal power allocation scheme which maximizes the throughput with delay and average power consumption constraints. The primary difference between existing literature on power optimization and our work is that we model state gathering overhead/costs and are interested in determining the optimal sampling frequency and number of bits for encoding samples so as to minimize the power dissipation while maintaining a fixed goodput.

3.3 Background

We begin with a brief overview of information-theoretic rate-distortion theory. A thorough description of this approach is available in [24]. Our goal here is to introduce the reader to this technique and describe its application to our problem.

Rate distortion theory describes the minimum rate (bits) required to achieve a particular distortion, where distortion is defined as the expected distance between a random variable and its reconstruction from its representation in bits (i.e., quantization). The theory also tells us that given a sequence of n i.i.d. random variables it is

possible to achieve a lower rate at a given distortion if we represent the sequence of the n variables jointly instead of considering them individually.

Let X be the (source/encoded) alphabet and $\hat{X}$ be the (receiver/decoded) alphabet. Similarly, let X^n and $\hat{X}^n$ denote the encoded and decoded sequences and denote f and g as the encoding and decoding functions respectively. The distortion $d(x, \hat{x})$ is a measure of the cost of representing the symbol x by the symbol $\hat{x}$. The distortion between two sequences x^n and $\hat{x}^n$ is denoted by $d(x^n, \hat{x}^n)$ is defined as $d(x^n, \hat{x}^n) = \frac{1}{n}\sum_{i=1}^{n} d(x_i, \hat{x}_i)$. For a given encoding and decoding scheme, $D = \mathbb{E}[d(X^n, g(f(X^n))]$ where the expectation is calculated over X.

The rate distortion function $R(D)$ for an i.i.d. source X with distribution $p(x)$ and bounded distortion function $d(x, \hat{x})$ is equal to the associated information rate distortion function $R^{(I)}(D)$ and is defined by equation (3.1).

$$R(D) = R^{(I)}(D) = \min_{p(\hat{x}|x):\mathbb{E}[d(X,\hat{X})\leq D]} I(X; \hat{X}) \tag{3.1}$$

where $I(X; \hat{X})$ is the mutual information between X and $\hat{X}$ and the minimization is taken over all possible distributions $p(\hat{x}|x)$. $R(D)$ thus represents the minimum number of bits required to encode each symbol X, given that the entire sequence X^n is encoded. The rate distortion function thus tells us that there exists some f and g such that the expected distortion is bounded by D if rate $R(D)$ is employed.

3.4 Network Model

In this section, we describe our network model and assumptions. We consider a network of n nodes with multiple source-destination pairs where each source has m disjoint paths to the destination with k links on each path. We assume time is divided into intervals of duration T_s and at the beginning of every interval, each source receives 'noisy' estimates about the links in the network.

In our model these link state estimates characterize the (time-varying) effect of shadowing on the received power. Shadowing is assumed to be a lognormally distributed random process (in dB it is normally distributed) [70]. Consider any sampling interval and let t be a time of interest in that interval, $0 \leq t < T_s$. Let us consider the i^{th} path and the j^{th} link along this path at some time t.

Let $L_{ij}(t)$ be the lognormal shadowing process and $X'_{ij}(t) = 10 \log_{10} L_{ij}(t)$ be its value in dB. $X'_{ij}(t)$ is assumed to be a stationary Gaussian random process with mean $\mu = 0$ and autocorrelation function $R_{X'}(\tau) = \sigma'^2 e^{-\lambda\tau}$ [38]. The autocorrelation coefficient function ($\rho'(\tau)$) for any stationary random process $X'(t)$ may be defined as $\rho'(\tau) = \frac{R_{X'}(\tau)-\mu^2}{R_{X'}(0)-\mu^2}$. Thus for the shadowing process, the autocorrelation coefficient function is given by : $\rho'(\tau) = e^{-\lambda\tau}$.

For ease of analysis we express $\ln L_{ij}(t) = \frac{\ln 10}{10} X'_{ij}(t) = X_{ij}(t)$ replacing the logarithm to base 10 with the natural logarithm. Hence, $X_{ij}(t)$ is also Gaussian random process with mean 0 and autocorrelation function $R_X(\tau) = \sigma^2 e^{-\lambda\tau}$ where $\sigma^2 = (\frac{\ln 10}{10})^2 \sigma'^2$. Therefore, the autocorrelation coefficient function($\rho(\tau)$) of $X(t)$ is given by $\rho(\tau) = \rho'(\tau)$. The correlation of $X_{ij}(t)$ indicates how the link state varies during the sampling interval, given its value at the beginning of the sampling interval. Knowledge of the correlation is essential for computing the expected power expended in an interval.

At the beginning of the sampling interval, the source receives $\hat{X}_{ij}(0)$, which are 'noisy' estimates of $X_{ij}(0)$. As $X_{ij}(0)$ are drawn from a continuous distribution, encoding them exactly will require an infinite number of bits. The 'noise' therefore corresponds to the quantization error and thus $\hat{X}_{ij}(0)$ are finite precision representation of $X_{ij}(0)$. The number of bits used to encode the values of $X_{ij}(0)$ determines the closeness of $\hat{X}_{ij}(0)$ to $X_{ij}(0)$; thus, the inaccuracy in $\hat{X}_{ij}(0)$ decrease as more bits are used for encoding. If ϵ is the noise or quantization error then, $\hat{X}_{ij}(0) = X_{ij}(0) + \epsilon$.

We model ϵ as Gaussian noise with mean 0 and variance σ_e^2 [61]. We consider that all the link state values are encoded together and sent to the source. We use rate-

distortion techniques [24] to upper bound σ_e^2. In particular, define the distortion as the squared-error distortion, $d(x, \hat{x}) = (x - \hat{x})^2$. Then $\sigma_e^2 = E[(\hat{X}_{ij}(0) - X_{ij}(0))^2] \leq D$. The rate distortion function $R(D)$ for any $N(0, \sigma^2)$ source with squared-error distortion is given in [24]:

$$R(D) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma^2}{D} & 0 \leq D \leq \sigma^2 \\ 0 & D > \sigma^2 \end{cases} \tag{3.2}$$

Equation (3.2) thus represents the minimum number of bits required to encode each shadowing sample. It is also clear that $\hat{X}_{ij}(0)$ is a Gaussian random variable with mean 0 and variance σ_D^2 given by $\sigma_D^2 = \sigma_e^2 + \sigma^2$.

We assume that the path loss and thus the distance between any two pairs of nodes in the network is the same. Later in Section 3.7 we discuss how to relax this assumption.

3.5 Minimum Power Problem

In this section, we describe the *Minimum Power Problem*. Each source desires a goodput G. Let C_b and C_t be the control overhead and the overall throughput (combined control and data) respectively. Therefore we have $C_t = G + C_b$. At the beginning of each sampling interval, the source collects noisy link state estimates. The source desires to minimize the expected power spent in any interval to achieve goodput G. Based on the noisy link state estimates collected, the source calculates the expected power consumed (for data and control) along each of the M paths to the destination in that sampling interval. It then selects the path i for which the expected power consumed is least. Note that we do not consider the power expended in the sampling process itself.

The goal of the *Minimum Power Problem* is to determine T_s (the sampling duration) and D (the measure of distortion) such that over all possible instantiations of

link estimates the expected power consumed (for transmitting both control and data) in any sampling interval to achieve a goodput requirement G is minimized.

Let Q_i be the expected power dissipated along the i^{th} path in a sampling interval, given the sampling interval T_s, the distortion D and the link state estimates $\hat{X}_{ij}(0)$ at the beginning of the interval. The source selects the path that dissipates the minimum expected power in the sampling interval and thus the *Minimum Power Problem* can be formally stated as,

Objective: $\min_{T_s,D} \mathbb{E}[\min_i Q_i]$

subject to the constraint:

$$C_t - C_b = G$$

3.6 Power Consumption and Control Overhead

In this section, we begin by modeling the transmit power expended along each path needed to achieve a fixed throughput during the sampling interval. We then model the control overhead as a function of the total number of links in the network and the rate distortion function. These models for power, control overhead and shadowing are then used to obtain an approximate solution to the *Minimum Power Problem* in Section 3.7.

3.6.1 Power Consumption

The transmitted power $P_i(t)$ along the i^{th} path at time t to achieve a total throughput C_t (data and control) is obtained by summing the per-link power of each hop. Let $P_{ij}^W(t)$ be the transmit power on the j^{th} link along the i^{th} path at time t when W and B are the transmission rate at any node and the available channel bandwidth in Hz respectively. We assume a homogeneous network with equal path loss between any two nodes; let d denote the distance between any two nodes in the network. Further let us consider a reference distance d_0 and let $P_t(d_0)$ and $P_r(d_0)$ be the transmit and

received power between two nodes separated by d_0. Shannon's formula [24] in Eqn. (3.3) relates the transmission rate, the shadowing, the AWGN and the power.

$$W = B \log_2(1 + \frac{P_{ij}^W(t) L_{ij}(t)}{F N_0}) \tag{3.3}$$

where N_0 is the noise, $F = \frac{P_t(d_0)}{P_r(d_0)}(\frac{d}{d_0})^\alpha$ and α is the path loss exponent. Hence we can transform the above equation in the following manner:

$$P_{ij}^W(t) = \frac{2^{W/B} - 1}{L_{ij}(t)} F N_0 \tag{3.4}$$

There is a subtle point to be noted here. Although the transmission rate is W, the source can only achieve a lower throughput C_t, as the wireless medium is a shared resource - if multiple nodes transmit together, interference and packet loss can occur. We assume that there is a scheduling algorithm that determines the time periods during the sampling interval when each source gets the opportunity to transmit. Each source transmits for only a fixed fraction of time during a sampling interval, e.g., it is allocated a fixed number of transmission slots in an interval. Let T_1 be the amount of time a source transmits in an interval of duration T_s.

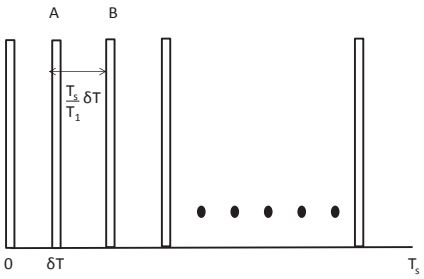


Figure 3.1. Power transmitted in a sampling interval

We abstract away the scheduling details and define the scheduling factor as $S = \frac{T_1}{T_s}$. S depends on the scheduling algorithm and the number of nodes and is a parameter in our model. Further, we consider a MANET with fast moving nodes (e.g., a military

MANET) such that C_t is much smaller than W. We also note that any arbitrary value of C_t is not achievable, e.g., the achievable C_t is bounded by results such as the Gupta-Kumar result [39].

Each source transmits for a duration T_1 in a sampling interval. Abstracting away the scheduling details, we assume that the source transmits at rate W uniformly for small durations (δT) throughout the sampling interval and the time between two consecutive transmissions is $\frac{T_s}{T_1}\delta T$. This is shown in Figure 3.1. The bars in the figure depict time periods when transmissions take place. This abstract modeling approach is necessary as one cannot assume that the source transmits continuously at rate W for a duration T_1 in the sampling interval. This would lead to an incorrect estimate of the expected power expended during the sampling interval, because the effects of the correlation of the shadowing process would be incorrectly accounted for if an interval T_1 is considered instead of T_s.

Our objective is to derive an expression for $P_{ij}(t)$, the transmit power on the j^{th} link required to achieve a constant throughput (C_t) for the entire sampling interval T_s similar to Eqn. (3.4). We model $P_{ij}(t)$ by:

$$P_{ij}(t) = \frac{a}{L_{ij}(t)} F N_0 \tag{3.5}$$

The value of a should be such that the total energy consumed and the total number of bits transmitted in the sampling interval are the same when transmitting at W for time T_1 and at C_t for time T_s. Ensuring that the total number of bits transmitted in both cases are the same, leads to Eqn. (3.6).

$$W = C_t \frac{T_s}{T_1} = \frac{C_t}{S} \tag{3.6}$$

We must also ensure that the total energy consumed is equal. Consider any two consecutive transmission time periods, i.e., points A and B in Figure 3.1. We assume

that as δT is very small, the shadowing value remains constant during the time interval $\frac{T_s}{T_1}\delta T$. Therefore we have,

$$a = (2^{W/B} - 1)\frac{T_1}{T_s} = (2^{C_t/SB} - 1)S \tag{3.7}$$

Substituting Eqn. (3.7) in Eqn. (3.5) we obtain the expression for $P_{ij}(t)$. The total power $P_i(t)$ expended along the i^{th} path is the sum of the per-link power of each hop and thus:

$$P_i(t) = \sum_{j=1}^{k} P_{ij}(t) = \sum_{j=1}^{k} \frac{2^{C_t/SB} - 1}{L_{ij}(t)} SFN_0 \tag{3.8}$$

3.6.2 Control Overhead

Following [93], we model the minimum overhead for gathering link state information as,

$$C_b = \frac{n(n-1)}{2}\frac{R(D)}{T_s} \tag{3.9}$$

The rationale behind this abstract model is that the total number of links must be less than $\frac{n(n-1)}{2}$ (the total number of links is $O(n^2)$), and that a source must know the state of all network links to compute its best path to the destination.

3.7 Solving the Optimization Problem

In this section, we approximately solve the *Minimum Power Problem*. We begin by expressing $P_i(t)$ (Eqn 3.8) as:

$$P_i(t) = \sum_{j=1}^{k} CY_{ij}(t) \tag{3.10}$$

where $C = (2^{C_t/SB} - 1)SFN_0$ and $Y_{ij}(t) = \frac{1}{L_{ij}(t)}$. Therefore, $Y_{ij}(t)$ is also a lognormal random process and we have $\ln Y_{ij}(t) = -X_{ij}(t)$.

Recall that Q_i is the expected power consumed along the i^{th} path in a sampling interval, given the sampling duration T_s, the distortion D and the link state estimates $\hat{X}_{ij}(0)$. Note that the centralized solution to the optimization problem only has estimated, finite-precision $\hat{X}_{ij}(0)'s$ available to it and not the true values $X_{ij}(0)$. Q_i can be formally expressed as,

$$Q_i = \frac{1}{T_s} \int_0^{T_s} \mathbb{E}[P_i(t)|\hat{X}_{i1}(0)\hat{X}_{i2}(0).....\hat{X}_{ik}(0); T_s, D]dt \tag{3.11}$$

Note that T_s and D are model parameters and are not random variables: we thus omit them while expressing conditional expectations. The expression for Q_i can be rewritten as,

$$Q_i = \frac{1}{T_s} \int_0^{T_s} \mathbb{E}\big[\mathbb{E}[P_i(t)|X_{i1}(0), ...X_{ik}(0)]|\hat{X}_{i1}(0), ...\hat{X}_{ik}(0)\big]dt \tag{3.12}$$

The above simplification can be done because given $X_{ij}(0)$, $P_i(t)$ is independent of $\hat{X}_{ij}(0)$, i.e., the underlying process itself does not depend on the observation $\hat{X}_{ij}(0)$. We first determine $H_i = \mathbb{E}[P_i(t)|X_{i1}(0), ...X_{ik}(0)]$ which can be done in the following way (Eqn. (3.13)). At any given time t, $X_{ij}(t)|X_{ij}(0)$ is a Gaussian random variable with mean $\mu_x(t) = \rho(t)X_{ij}(0)$ and variance $\sigma_x^2(t) = \sigma^2\left(1 - \rho^2(t)\right)$ [26]. Hence at any given time t, $Y_{ij}(t)|X_{ij}(t)$ is a lognormal random variable with mean $e^{-\mu_x(t)+\frac{\sigma_x^2(t)}{2}}$ [2].

$$H_i = C\sum_{j=1}^{k} \mathbb{E}[Y_{ij}(t)|X_{i1}(0), ...X_{ik}(0)]dt = C\sum_{j=1}^{k} A(t)e^{-\rho(t)X_{ij}(0)}dt \tag{3.13}$$

where $A(t) = e^{\frac{\sigma^2}{2}(1-\rho^2(t))}$. Substituting Eqn. (3.13) in Eqn. (3.12) we have,

$$Q_i = \frac{C}{T_s}\sum_{j=1}^{k} \int_0^{T_s} \mathbb{E}[A(t)e^{-\rho(t)X_{ij}(0)}|\hat{X}_{ij}(0)]dt = \frac{C}{T_s} \int_0^{T_s} A(t)e^{\frac{\rho^2(t)\sigma_e^2}{2}} \sum_{j=1}^{k} e^{-\rho(t)\hat{X}_{ij}(0)}dt \tag{3.14}$$

Equation (3.14) uses the fact that the quantization error ϵ is independent of $\hat{X}_{ij}(0)$. Moreover, at any given time t, $\rho(t)\epsilon$ is a Gaussian random variable with mean 0 and variance $\rho^2(t)\sigma_e^2$. Therefore, at any given time t, $e^{\rho(t)\epsilon}$ is a lognormal random variable with mean $e^{\frac{\rho^2(t)\sigma_e^2}{2}}$ [2].

We can still further simplify the expression for Q_i. We approximate the sum of lognormal random variables by a lognormal random variable [33]. In Eqn. (3.14), at any given time t, $Y'_{ij}(t) = e^{-\rho(t)\hat{X}_{ij}(0)}$ is a lognormal random variable with mean $\mu_{y'}(t) = e^{\frac{\rho^2(t)\sigma_D^2}{2}}$ and variance $\sigma^2_{y'}(t) = (e^{\rho^2(t)\sigma_D^2} - 1)e^{\rho^2(t)\sigma_D^2}$. Therefore, $Y'_i(t) = \sum_{j=1}^{k} Y'_{ij}(t)$ is approximated by a lognormal random variable with mean $\mu_1(t) = k\mu_{y'}$ and variance $\sigma_1^2(t) = k\sigma^2_{y'}$. Let $Z_i(t)$ be the Gaussian variable corresponding to $Y'_i(t)$. We can express its variance $\sigma_z^2(t) = \ln\left[\frac{e^{\rho^2(t)\sigma_D^2}-1}{k} + 1\right]$ and mean $\mu_z(t) = \ln k + \frac{\rho^2(t)\sigma_D^2}{2} - \frac{\sigma_z^2(t)}{2}$ [33]. Further, let $A_1(t) = A(t)e^{\frac{\rho^2(t)\sigma_e^2}{2}} = e^{\frac{\rho^2(t)\sigma_e^2+\sigma^2(1-\rho^2(t))}{2}}$. We then express Eqn. (3.14) as,

$$Q_i = \frac{C}{T_s}\int_0^{T_s} A_1(t)\sum_{j=1}^{k} Y'_{ij}(t)dt \approx \frac{C}{T_s}\int_0^{T_s} A_1(t)e^{Z_i(t)}dt \tag{3.15}$$

We define $H' = \min_i Q_i$. H' can be expressed as,

$$H' = \min_i \frac{C}{T_s}\int_0^{T_s} A_1(t)e^{Z_i(t)}dt \geq \frac{C}{T_s}\int_0^{T_s} A_1(t)e^{-\max\{-Z_i(t)\}}dt \tag{3.16}$$

The inequality is due to the fact that minimum of a summation is greater than the summation of the minimum. For solving the *Minimum Power Problem* we then need to determine $\mathbb{E}[H']$ which can be written as,

$$\mathbb{E}[H'] > \frac{C}{T_s}\int_0^{T_s} A_1(t)\mathbb{E}[e^{-\max\{-Z_i(t)\}}]dt \tag{3.17}$$

The next step is to determine the distribution of $U = \max\{-Z_i(t)\}$. It is clear that $\{-Z_i(t)\}$ are i.i.d. Gaussian random variables with mean $-\mu_z(t)$ and variance $\sigma_z^2(t)$.

The maximum of i.i.d. Gaussian random variables follows a Gumbel distribution asymptotically, as m the number of paths goes to ∞ with scaling factor $a_m = \frac{\sigma_z(t)}{\sqrt{2\ln m}}$ and location factor $b_m = \sigma_z(t)(\sqrt{2\ln m} - \frac{\ln\ln m + \ln(4\pi)}{2\sqrt{2\ln m}}) - \mu_z(t)$ respectively [14]. Let us consider the random variable V such that $\ln V = U$. V follows a log-Gumbel distribution with the same parameters as U [42]. Therefore as $Z_i(t)$ are Gaussian, the mean of the log-Gumbel distribution exists and it follows a gamma function multiplied by an exponential.

But, we are interested in $U' = -U$, which follows a negative Gumbel distribution. Define $\ln V' = U'$. It can be easily shown that $E[V'] = e^{-b_m}\Gamma(1 + a_m)$.

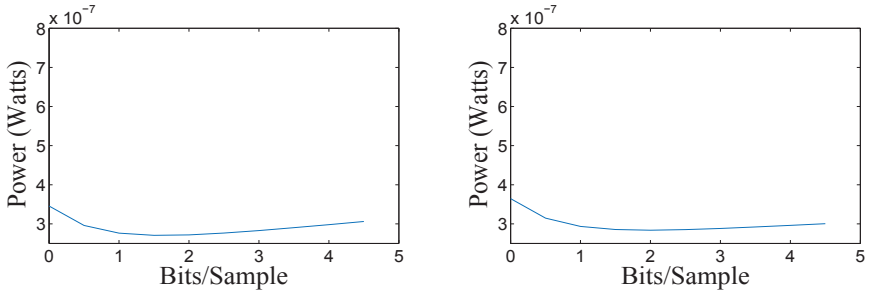
$$\mathbb{E}[H'] \approx \frac{C}{T_s}\int_0^{T_s} A_1(t)\mathbb{E}[e^{U'}]dt = \frac{C}{T_s}\int_0^{T_s} A_1(t)e^{-b_m}\Gamma(1 + a_m)dt \tag{3.18}$$

$\mathbb{E}[H']$ computed from Eqn. (3.18) will be an approximation to $\mathbb{E}[\min_i Q_i]$. The optimization problem thus reduces to $\min_{T_s,D} \frac{C}{T_s}\int_0^{T_s} A_1(t)e^{-b_m}\Gamma(1+a_m)dt$, which can be easily computed numerically.

Equation (3.18) holds for the equal path loss scenario. But if this assumption is relaxed, the above analysis holds with minor modification until Eqn. (3.17) - we only need to model the Gaussian variable $Z_i(t)$ to take into account the different values of C for the different links resulting from the unequal path loss assumption. If the *Minimum Power Problem* is to be solved in an unequal path loss scenario, one can obtain the distribution of $\max\{-Z_i(t)\}$ numerically (which is easy as $Z_i(t)$ are Gaussian) and then determine $\mathbb{E}[H']$. However, note that such a procedure will be computationally expensive.

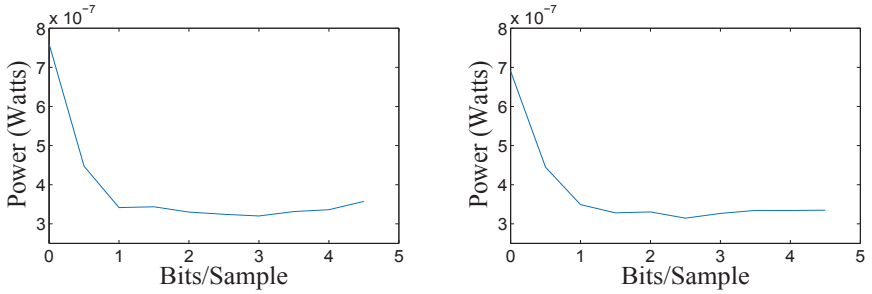
3.8 Evaluation

In this section, we present numerical results obtained by solving the optimization problem using Eqn. (3.18). We first study the tradeoff between the sampling interval



(a) Sampling interval =1 sec (b) Sampling interval =2 sec

Figure 3.2. Numerical: number of bits per sample versus sampling interval tradeoff



(a) Sampling interval =1 sec (b) Sampling interval =2 sec

Figure 3.3. Simulation: number of bits per sample versus sampling interval tradeoff

and the number of bits per sample for a specific set of parameters and then proceed to investigate the impact of the various parameters on this tradeoff. We consider a network of 100 nodes with $G = 75Kbps$, $B = 10MHz$, $S = 0.05$ and $\lambda = \frac{1}{5}sec$. The variance of shadowing is 25 dB. Further, we assume $m = 5$ and $k = 5$, i.e., the source has 5 disjoint paths with 5 links each. The results are obtained by increasing the number of bits per sample at a granularity of 0.5. In order to facilitate the comparison between G and C_b, we note that when $R(D) = 2$ and $T_s = 1sec$, $C_b \approx 10Kbps$ (Eqn. (3.9)). We also use this same configuration when we study the effect of the different parameters on the sampling interval and number of bits per sample (except for the parameter under investigation).

Figure 3.2 shows the variation of the transmit power with the number of bits per sample for different values of sampling interval. We observe that with a small number of bits per sample (very little information about network link state), the expected power consumed is high irrespective of the length of the sampling interval.

In particular, when the number of bits per sample is 0 (equivalent to choosing a path at random), the power consumed is high. Conversely, when the number of bits per sample is high, the additional information is of marginal use in determining the minimum power path, but the overhead expended in transmitting these control bits is high.

We are interested in obtaining the global minima of the power consumed considering the entire range of the sampling interval and number of bits per sample. We observe that for the parameter values considered, the optimal value of the sampling interval is 1 second and the number of bits per sample is 1.5. Although the results in Figure 3.2 are obtained for $S = 0.05$, similar figures were obtained for other values of S. In the throughput range of interest (when C_t is small), the factor $(2^{C_t/SB} - 1)$ in (Eqn. (3.8)) linearizes, making the power almost independent of S and vary linearly with C_t.

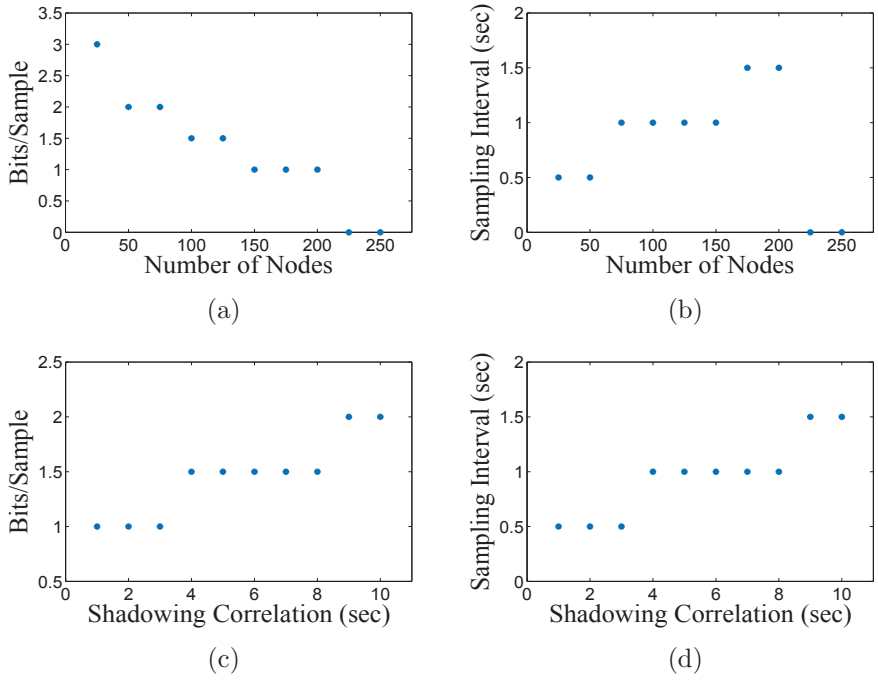


Figure 3.4. Variation of number of bits per sample and sampling interval with number of nodes and shadowing correlation $(\frac{1}{\lambda})$

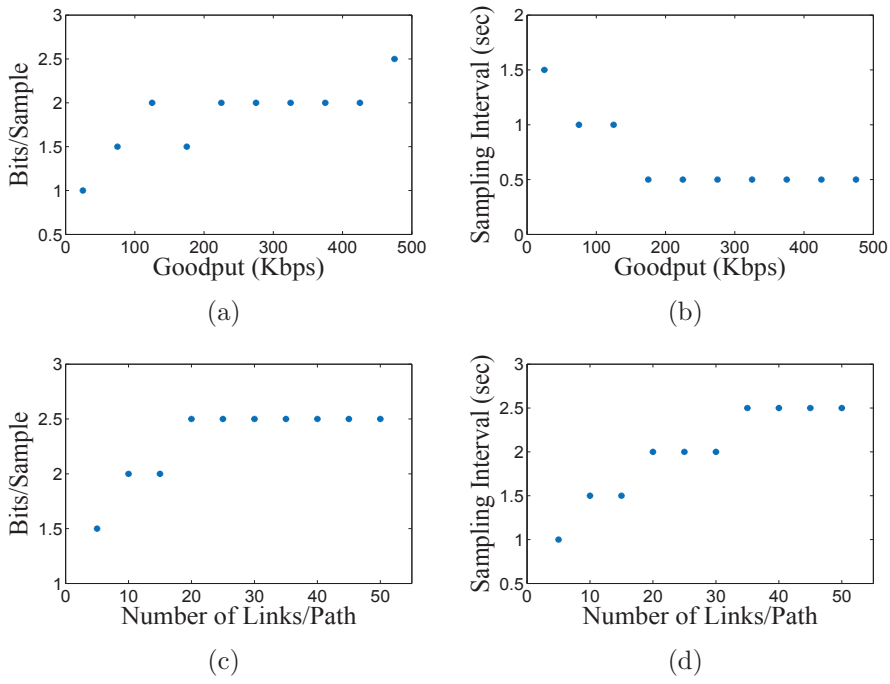


Figure 3.5. Variation of number of bits per sample and sampling interval with goodput and number of links per path

We next study the impact of the various parameters (number of nodes, shadowing correlation ($\frac{1}{\lambda}$), goodput, number of links in a path, number of paths) on the tradeoff between the number of bits per sample and the sampling interval. As these results are obtained by increasing the sampling interval and the number of bits per sample at a granularity of 0.5, the graphs are discontinuous.

Figures 3.4(a) and 3.4(b) show the variation of the number of bits per sample and the sampling interval with the number of nodes. We observe that as the number of nodes increases, the number of bits per sample decrease and the sampling interval increases. This is intuitive since as the number of nodes in the network increases, the control overhead also increases (roughly as $O(n^2)$). Therefore, when the number of nodes is low the optimum decision is to have a small sampling interval (i.e., to sample the network frequently) and encode the samples using a greater number of bits. On the other hand when the number of nodes is large, increased overhead results in the optimum sampling interval being high and number of bits per sample being low. Note

that when the number of nodes is very high, the optimal strategy is to select a path at random - this corresponds to the case when the number of bits per sample is equal to zero in Figure 3.4(a).

We study the variation of the number of bits per sample and the sampling interval with the correlation of the shadowing process ($\frac{1}{\lambda}$) in Figures 3.4(c) and 3.4(d) respectively. Figures 3.4(c) and 3.4(d) show that both the number of bits per sample and the sampling interval increase with the shadowing correlation. This is because as shadowing correlation increases, the optimal configuration takes advantage of this by sampling at a lower frequency (longer sampling interval). Simultaneously, the number of bits per sample also increases, since the decrease in overhead due to a longer sampling interval provides the network an opportunity to gather high fidelity samples.

Figures 3.5(a) shows that with increasing goodput, the number of bits per sample increases. This is because as the goodput is much larger than the overhead, additional bits can be used to encode link state values. At the same time, Figure 3.5(b) shows that as the goodput increases the sampling interval decreases, which can also be attributed to the fact that the overhead is smaller in comparison to the goodput.

In Figures 3.5(c) and 3.5(d), we observe that the number of bits per sample and the sampling interval increases with the number of links in a path. As the number of links in a path increases, the probability that at least one of these links is in a bad state (i.e., requiring high power to meet the goodput requirement) increases. Because of the exponential dependence of power on link quality, the power consumed along the entire path will be dominated by the bad links. Further, as the error in estimating the expected power over a path in an interval increases with the number of links in it, it is advantageous to use more bits for encoding the samples, so that the correct decision is taken i.e., that path with the minimum power is chosen. The increased overhead resulting from the larger number of bits used can then be compensated for by choosing a larger sampling interval.

We also studied the variation of transmit power with the number of paths and found that the number of bits per sample decreases and then becomes zero. As the number of paths increases, the chance of selecting a good path goes up and thus the number of bits per sample decreases. When there are many available paths, selecting a path at random suffices and there is no need to collect state information.

3.9 Simulation

In this section, we report on our use of simulations using Eqn. (3.14) to drive the simulation, to validate our numerical results. Specifically, we study the impact of the inequality in Eqn. (3.16) and the two main assumptions of the model - *(i)* approximating the sum of lognormals by a lognormal and *(ii)* approximating the maximum of i.i.d. Gaussian random variables by a Gumbel distribution - on the accuracy of our numerical results.

We consider the same set of parameters used in the numerical evaluation. For a particular value of sampling interval and number of bits per sample, we generate shadowing measurements (from a Gaussian distribution) for all links to emulate the link state values collected at the beginning of the sampling interval. We determine the expected power consumed for the entire interval along each of the m paths and then select the path for which the expected power consumed is minimum. For each pair of values of sampling interval and number of bits per sample, we repeat this process 500 times to obtain the mean power consumed.

Simulation results depicting the tradeoff between the number of bits per sample and sampling interval with the transmit power are shown in Figure 3.3 and should be comparable to the numerical results in Figure 3.2. As in the case of our numerical evaluation, the simulation results also show that the expected power decays rapidly with an increasing number of bits per sample and then begins increasing again.

We note that the power consumption is higher in case of simulation, particularly so for a small number of bits per sample (approaching 0). This is because our nu-

merical analysis is an approximation that becomes better as the number of bits per sample increases. A careful examination of Figures 3.2 and 3.3 reveals that when the number of bits per sample is 0, the expected power consumed increases for numerical evaluation and decreases for simulation with increasing sampling interval. The intuitive explanation as to why the expected power decreases with an increase in the sampling interval in case of a real system (i.e., in our simulation) is the following.

Let us consider for the sake of simplicity that paths are of two types - good and bad; paths are classified as good when the power consumed at the beginning of the sampling interval is low and bad when it is high. The expected power consumed in any sampling interval is thus the additive sum of the conditional expected power consumed given a path of a specific quality (good or bad), multiplied by the probability that the selected path is of the specified quality. The above fact holds true irrespective of the duration of the sampling interval.

Let us next consider the probability of selecting a good or bad path. As shadowing is Gaussian distributed, the probability of a path being good or bad is the same and is independent of the sampling interval. As the number of bits per sample is zero (equivalent to selecting a path at random), the chance of selecting good and bad paths is the same. Further, because of the exponential dependence of power on path quality, expected power expended during a sampling interval is higher when the selected path is bad in comparison to when it is good.

So far we have only considered the effect of path quality on expected power consumption. We will now reason about the impact of the sampling interval on expected power consumption. When the selected path is bad, expected power expended during a sampling interval will be higher for a shorter sampling interval than for a longer sampling interval since shadowing correlation decays exponentially. Similarly, when a good path is selected, expected power expended during the sampling interval will be lower for a shorter sampling interval.

But, the positive difference in the expected power expended between small and large sampling interval when the selected path is bad, is not compensated for by the negative difference in expected power expended between them when the selected path is good. Thus, when the number of bits per sample is zero, expected power consumed when the sampling interval is small is higher than when the sampling interval is long.

Note that although there is a mismatch between the numerical and simulation results when the number of bits per sample is small, our goal is not to study any specific scenario, but rather to determine the optimal sampling interval and the number of bits per sample. From our simulation, we find that the minimum expected power is consumed for bits per sample=2.5 and sampling interval=2 seconds, which is comparable to the numerical results (bits per sample=1.5; sampling interval=1 second). Hence we conclude that the approximations in Section 3.7 help in modeling the system accurately. We have also studied the tradeoff between the number of bits per sample and sampling interval for a network with unequal path loss via simulation and observed that a tradeoff similar to the equal path loss case.

3.10 Conclusion

In this chapter, we formulated an optimization problem to determine the frequency at which a source should gather link state estimates and the number of bits used to encode these estimates such that the expected power consumed over a sampling interval is minimized subject to goodput constraints. We observed that long sampling intervals fail to take advantage of the temporal correlation of link state estimates while short sampling intervals incur significant overhead. Similarly, small number of bits per sample provide very little information about the network state while large number of bits provide marginal additional information. Our work can be viewed as a first step for providing network designers a tool for determining optimal operating points (state update frequency, number of bits per sample).

CHAPTER 4

ROUTING WITH ADAPTIVE FLOODING IN HETEROGENEOUS MOBILE NETWORKS

4.1 Introduction

Uncertainty and change in network connectivity are fundamental characteristics of mobile ad hoc networks (MANETs). A variety of forwarding strategies have been proposed for such scenarios, ranging from stateful routing protocols [65] to flooding [91]. Recently, [60] showed that in mobile networks with *homogeneous* node mobility and link characteristics, stateful routing protocols such as OLSR [46] perform well in dense and stable networks, whereas flooding is preferable in sparse and rapidly changing networks. However, mobility and connectivity characteristics observed in real-world measurements are often *heterogeneous*: while some nodes may have few or highly dynamic links, there are also well-connected nodes forming sizable connected components [41, 89]. In such networks with both stable and dynamic components, it is likely (and we will see) that neither routing nor flooding alone may perform particularly well in a given scenario.

We propose a simple approach towards forwarding in heterogeneous mobile networks: based on local link characteristics as well as network-wide considerations, determine which nodes should forward according to the forwarding table computed by the native routing protocol, and which nodes should locally broadcast their traffic to all neighbors. Our work is driven by the intuition that nodes with particularly reliable and stable links should be well-suited to operate as routers, since the next-hop towards a given destination determined by the native routing algorithm would continue to work well in the future. Conversely, a node with highly dynamic or unreliable links

might better operate as a flooder, exploiting the broadcast nature of omni-directional antennas to forward a packet to all neighbors in a single transmission; packet copies can then be forwarded from one or more of those neighbors (either via routing or flooding by that neighbor) towards the destination. Our simple approach leverages the vast amount of past research on both routing and flooding in mobile networks with minimal changes to existing protocols. From a performance standpoint, we will show that our approach not only matches the performance of network-wide routing or flooding in stable or dynamic settings, respectively, but also performs better than either of them in heterogeneous scenarios.

In our approach, we have a single decision to make for each node — should it operate as a router or flooder — in such a way as to maximize global network goodput. Despite its apparent simplicity, this is a challenging problem. First, available link state information may be stale due to mobility and variability inherent in wireless links. Second, while it is tempting to think that classifying a node as a router or flooder only requires local information, flooding at one node increases network traffic at downstream nodes (due to increased number of packet copies that are created and thus will be routed, or broadcast forwarded, towards the destination) and may ultimately reduce overall goodput due to congestion. In addition, one node being a flooder may affect the usefulness of turning another node into a flooder, implying subtle dependencies in the decision process.

We present a simple greedy algorithm (*adaptive-flood*) to determine which nodes should operate as routers and which nodes should operate as flooders. The algorithm assumes that an underlying native routing protocol is available and then iteratively selects those nodes as candidate flooders that maximize the *overall expected network goodput*. The algorithm selects nodes as flooders in decreasing order in which they contribute to maximizing expected network-wide goodput; it stops when converting any of the remaining routers into a flooder would result in a decrease in expected goodput. Practically, this means that each node needs to determine only one piece

of information, namely whether to unicast packets to the next-hop neighbor specified in its forwarding table, or to locally flood each packet to all neighbors.

We make the following contributions in this chapter. First, we present our *adaptive-flood* algorithm to determine which nodes should operate as routers and which nodes should operate as flooders (Section 4.4). Second, we show via simulation that *adaptive-flood* outperforms network-wide routing or flooding. In particular, at low network loads flooding outperforms routing while at high network loads, performance is reversed. In contrast, *adaptive-flood* matches or outperforms both baseline approaches over most or all of the range of loads in varied settings (Section 4.5). From these results, we conclude that routing combined with adaptive flooding is a promising solution to solve challenges inherent in mobile networking.

The rest of this chapter is organized as follows. We discuss related work in Section 4.2. We formalize the problem and the underlying network model in Section 4.3 and describe the *adaptive-flood* algorithm for router/flooder classification in Section 4.4. The performance of these algorithms is evaluated via simulation presented in Section 4.5. We finally conclude this chapter in Section 4.6.

4.2 Related Work

Several past research efforts [7, 60, 69] have addressed the challenge of classifying MANETs based on connectivity and predictability – concerns that are of central importance to us in this chapter. [60] proposes a framework for organizing the decision space of communication strategies (i.e., determining whether the network as a whole should operate by flooding, routing, or store-carry-and-forward) in a *homogeneous* MANET based on connectivity and unpredictability so as to maximize goodput. Similar approaches for classifying networks (as connected, intermittently connected or disconnected) based on connectivity (i.e., presence of paths) and mobility (i.e., contact time, meeting) have been investigated [7, 69]. In contrast to prior

work where classification has been done for the network as a whole, we develop per-node classification strategies (route or flood) in order to maximize goodput.

A number of past efforts have sought to exploit characteristics such as connectivity, predictability and mobility of wireless networks to design forwarding protocols that enhance performance. Epidemic routing [91, 100] and multicopy routing [85] are designed for sparsely-connected networks and use a store-carray-and-forward mechanism and packet replication to battle poor connectivity. [10, 25] make assumptions on the mobility pattern and network topology to design forwarding protocols for intermittently connected networks. A survey of different forwarding strategies designed for MANETs and DTNs is available in [3, 102]. None of this past research, however, investigate the question of which nodes should flood/route in a MANET with time-varying connectivity.

Our work is closest to [89], which proposes a routing protocol, R3, that provides robust performance in diverse and varying connectivity regimes. They identify packet replication as the key factor governing performance for networks at opposite ends of the connectivity spectrum (meshes and DTN). R3 replicates packets along two paths for each flow, pruning one of the paths in the event of network congestion. They also propose the SWITCH protocol, in which nodes make decisions locally and flood packets only when the designated next hop for that packet is unavailable. SWITCH performs close to R3 in their evaluation. Our work differs from R3 in that we determine which nodes should flood/route in a network-wide context, taking multiple flows into account when making a routing/flooding decision and and may perform packet broadcast (rather than 2-replication [89]) at any node within the network, rather than only at the source, as in R3. Our algorithm also differs from SWITCH in that we determine which nodes should flood over an epoch of time, not just flood when a next-hop neighbor towards a destination in is unavailable.

A significant amount of past research has been devoted to demonstrating the capacity scaling of both flat and hybrid MANETs [39, 54, 99]. Similarly several for-

Notation	Definition
N	set of all nodes
E	set of all edges
$\mathscr{F}$	set of flows
F	list of flooders
N_i	neighbors of node i
f_s, f_d	source and destination node of flow f
$\mathbb{N}$	is the list of N_i, $\forall i \in N$
H_{if}	next hop forwarder for node i for flow f
$\mathbb{H}$	next hop forwarder matrix
p_{ij}	link success probability between nodes i and j
$\mathbb{P}$	link success probability matrix
Φ_{ij}	traffic originating from i and destined for j
Φ	traffic matrix

Table 4.1. Notation

warding strategies (routing, flooding and hybrid) aimed at improving goodput have been proposed and extensively studied for both MANETs and DTN [10, 21, 65]. Our work differs from existing literature in that we are not proposing a new forwarding protocol from scratch for a particular setting; rather we study the problem of enhancing network goodput by selectively classifying a subset of nodes as flooders and the remaining as routers.

4.3 Network Model

Let us begin by defining our network scenario and the router/flooder classification problem. We consider a network with $|N|$ nodes. Let $\mathscr{F}$ be the set of flows in the network. The source and destination for any flow f are denoted by f_s and f_d respectively. A summary of our notation is available in Table 4.1.

Time Periods. Time is slotted and we introduce two intervals beyond the minimal interval defined as one time *slot*. Specifically, packet transmissions occur at each time slot, node mobility takes place at the beginning of each *interval,* which is a period of several time slots, and finally, routing tables are updated at the beginning of every *epoch,* which is a period of multiple intervals.

Link State Information. We assume that during each epoch, state information characterizing the connectivity between nodes is collected. Let us consider any two nodes i and j and informally consider p_{ij} as the probability of successfully transmitting a packet directly from node i to node j (we will substantially sharpen this definition of p_{ij} in Section 4.5, where we use simulation to assess the performance of our greedy algorithms). The link quality can vary both due to the wireless channel and mobility of the nodes during the epoch, but we abstract away these details via the p_{ij} link characterization. $\mathbb{P}$ is the matrix of p_{ij}'s and is referred to as the link success probability matrix.

We next represent the network as a graph $G(N, E)$ where E denotes the set of all edges in the graph; an edge exists if $p_{ij} > 0$.

Routing and flooding. We consider a simple case where $\mathbb{P}$ is the only state information available at the beginning of an epoch and is used for determining routes. At the beginning of each epoch, we assume that routes are calculated using Dijkstra's shortest path algorithm where an edge between node i and j has link weight $1/p_{ij}$. Note that any other routing algorithm could be used; we use Dijkstra's algorithm for simplicity. H_{if} denotes the next hop neighbor for node i for flow f, as obtained by the routing algorithm.

Each node in the network can either act as a router or a flooder, but cannot perform both actions preferentially based on the destination of the packet. If node i operates as a router, it forwards packets according to H_{if}; otherwise it floods all packets. Let N_i be a list denoting the neighbors of i (node j is said to be a neighbor of i if $p_{ij} > 0$). If i operates as a flooder, it sends the same packet to every node in N_i. To prevent packets from circulating in the network in loops, nodes perform duplicate packet transmission suppression.

Traffic and Capacity. We assume that node i transmits data at a rate of C_i packets per time slot. Therefore all outgoing links from i can carry data at the maximum rate of C_i. C_i and p_{ij} together capture the capacity constraint for the link

ij. Φ_{ij} is the amount of traffic originating at node i and destined for node j; Φ is the corresponding the traffic matrix.

Overall network goodput. We define a flow's *goodput* as the number of unique packets received at the destination for the flow per time slot. The overall network goodput is the sum of goodputs for the different flows.

Based on this model, we define the problem to be solved by the classification algorithm *at the beginning of every epoch* as follows.

Given the above network model and the forwarding tables from the routing algorithm, classify certain routers as flooders so as to maximize overall network goodput. We note that our algorithm for solving this problem pre-supposes the presence of a native routing algorithm that executes periodically, following the execution of the native routing algorithm; our work thus falls squarely in the network control plane. We discuss alternate approaches for maximizing overall goodput in Chapter 7.

4.4 Router/Flooder Classification Algorithm

In this section, we propose a simple greedy algorithm (*adaptive-flood*) for router/flooder classification. Our algorithm, which operates in the MANET control plane (e.g., would execute following the periodic execution of the network's native stateful routing algorithm), classifies each network node as a router or as a flooder. Nodes classified as routers will unicast-route packets according to forwarding tables computed by the MANET's native stateful routing algorithm; nodes classified as flooders will *locally* flood a packet to all one-hop neighbors, who will then in turn unicast-route or flood (depending on their own classification) that packet. We note that similar to stateful routing protocols such as OLSR, our router/flooder classification algorithms can be run *locally* given broadcast link state updates, as discussed shortly. We then compare the performance of *adaptive-flood* with two baseline approaches: *routing* (where all network nodes operate as routers and forward packets to next hop neighbors based on the MANET's native routing algorithm, which we assume is based on Dijkstra's

algorithm) and *flooding* (where all network nodes operate as flooders and forward every packet to all their neighbors). Flooders perform duplicate suppression so as to not forward the same packet twice.

Adaptive-flood is an iterative greedy algorithm that turns one router into a flooder at each iteration, selecting that router whose change to a flooder would result in the maximum increase in expected network goodput. The algorithm terminates when turning any remaining router into a flooder would result in a decrease in expected total goodput. The details of the algorithm are given in Algorithm 1.

Adaptive-flood begins with all network nodes initially classified as routers. It takes as input the graph $G(N, E)$ whose edges are all initially unweighted. It then computes the expected total (over all source/destination pairs) goodput for the $\mathscr{F}$ flows by calling the *total-goodput()* function. In Algorithm 1, $\mathscr{G}_F$ is the expected total network goodput when there are F flooders. During each iteration in *adaptive-flood* (lines 4–14), the expected total goodput is calculated if router s were to be turned into the flooder, given the current list of routers and flooders. The algorithm then selects that particular router (F') that gives the maximum increase in expected total goodput if it were to be turned into a flooder (lines 6–10). This router is then added to the list of flooders F. The algorithm terminates when either all routers have been classified as flooders or if converting any of the remaining routers into a flooder (individually) results in a *decrease* in expected total goodput. When a router is added to F, the usefulness (in terms of goodput) of converting some other router into a flooder can change, since converting a node into a flooder can change the incoming traffic rates at other network nodes.

Calculating the effect of a router-to-flooder change on total network goodput. The *total-goodput()* function computes the overall (over all flows) goodput, given a list of routers and a list of flooders and the next-hop forwarding matrix $\mathbb{H}$ computed via Dijkstra's algorithm. Doing so is challenging for two reasons. First, link capacities are finite and buffer overflows will occur when the incoming traffic rate

exceeds a node's capacity to send that traffic on its going link(s). Second, given the presence of flooding nodes in the network, multiple copies of the same packet may be received at a node, and via duplicate suppression, only a single copy of that packet will be forwarded. Thus, traffic input rates to nodes need not equal their output rate, even in the absence of congestion losses due to limited link capacities.

Modeling the effects of finite buffer overflow. We model buffer overflow by adopting a fluid model in which nodes probabilistically drop packets if the expected incoming traffic rate exceeds that node's outgoing transmission capacity, C_i. Let a_i denote the probability that a packet is successfully received and forwarded through node i, assuming no losses due to transmission errors. We refer to a_i as the *packet-passage probability* at node i. Let R_i be the incoming traffic rate at node i.

$$a_i = \min\{1, \frac{C_i}{\mathbb{E}[R_i]}\} \tag{4.1}$$

Thus, when the expected incoming traffic rate is less than link capacity, all arriving packets are successfully forwarded by that node, ($a_i = 1$). When the expected incoming traffic rate exceeds the outgoing rate, arriving packets are successfully forwarded by that node with probability $\frac{C_i}{\mathbb{E}[R_i]}$. We note that this simple model of congestion is used *only* for calculating goodput in our control plane algorithm, *adaptive-flood.* The MANET's data plane itself performs packet dropping due to buffer overflow according to its native policy; our model calculations of goodput only affect the control-plane router/flooder classification.

Modeling multiple copies of a packet, duplicate suppression. When one or more network nodes are classified as flooders, multiple copies of the same packet in flow f may arrive at a node due to upstream flooders. Duplicate copies would be suppressed, resulting in only a single copy being forwarded to the node's output interface. Therefore the set of nodes and links traversed by a given flow's packets

will form a directed acyclic graph (rather than a path) between the flow's source and destination nodes.

The *calculate-DAG* algorithm (Line 34 of Algorithm 1) maintains two lists: the observed list - O and the explore list - X. For every flow f, DAG construction begins from the source f_s. X and O initially contain only f_s and f_d respectively (line 36). The while loop in line 37 then iterates until the explore list is empty. At every iteration of the while loop, the node (m) at the head of X (line 38) is considered (in the first iteration the node is f_s). Recall that when the *calculate-DAG* function is called, there are F flooders in the network. Therefore m can be either a router or a flooder. In either case (lines 41–44 for a flooder; lines 46–48 for a router), we update the weights of the links from m to its one or more neighbors and add each neighbor to the explore list if it is not already on the explore or observed list. In lines (50-52) we construct the graph D_f by determining the set of edges in it. Note that our construction of D_f allows for the existence of isolated nodes in it.

Total-goodput() function. The *total-goodput()* function (Line 16 of Algorithm 1) returns the total goodput for the $\mathscr{F}$ flows in the network, given the list of flooders (F). The total network goodput is calculated by summing the goodput for the individual flows in the network (Lines 30–32). To calculate the goodput of individual flows, it is necessary to determine a flow's DAG and also the packet-passage probabilities at all network nodes.

Even though packets for a given flow traverse a DAG, when there are multiple flows in the network, it is possible that some node j will receive traffic from node i and vice versa. In this case, since a_i depends on R_i which includes traffic arriving from j, and a_j depends on R_j which includes traffic arriving from i, we'll need to compute the packet-passage probabilities via a set of simultaneous equations. Lines 21–29 in Algorithm 1 are a fixed point iteration for calculating the packet-passage probabilities, $\vec{a}$. The algorithm begins with an initial feasible packet-passage probability ($\vec{a}^0$) (in our case 0). For each flow f, the fixed-point iteration then uses $\vec{a}^{l-1}$ at iteration l to

calculate the incoming rate at every node (line 23 via the *calculate-incoming-rate()* function). $\vec{I_f}$ is a vector of the incoming rates at different nodes for flow f, while I_{fs} denotes the incoming rate for flow f at node s. The fixed point iteration then uses the incoming rates to compute the packet-passage probabilities to be used in iteration $l+1$ (line 26), $\vec{a}^l$. The fixed-point iteration converges when the maximum absolute difference between the packet-passage probabilities in two successive iterations are all below a threshold τ (line 27).

total-goodput() then computes the total goodput and returns this value to the *adaptive-flood* algorithm (Lines 30–32). Note that the goodput for flow f is simply the incoming traffic for flow f at node f_d; the *calculate-goodput()* function is thus very similar to the *calculate-incoming-rate()* function and returns I_{ff_d}.

Calculate-incoming-rate() function. All that remains to be discussed is how $\vec{I_f}$, the incoming traffic rate at node i, is determined in line 23 of *total-goodput()*. The primary complication here is that multiple copies of the same packet in flow f may arrive at a node due to upstream flooders in flow f's DAG. We adopt an approximate approach for computing $\vec{I_f}$, the incoming traffic rate at node i, as follows. To compute $\vec{I_f}$ the algorithm first determines the DAG (D_f) traversed by that flow's packets (lines 19–20 in Algorithm 1). The algorithm then performs a topological sort V_f for D_f. For a directed acyclic graph, the topological ordering provides a linear ordering of its vertices such that for every directed edge from vertex u to vertex v, u comes before v in the ordering. The source f_s and destination f_d are the first and last nodes in this ordering (V_f). We then pass D_f and V_f as parameters to the *calculate-incoming-rate()* function (line 23 in Algorithm 1). Note that D_f and V_f are calculated only once in the *total-goodput()* function because the list of flooders does not change between iterations of the while loop in line 21 in Algorithm 1.

We next evaluate the probability of a packet reaching the nodes in V_f in order in the *calculate-incoming-rate()* function. To determine the probability of a flow f's packet (which has been flooded one or more times upstream in flow f's DAG) reaching

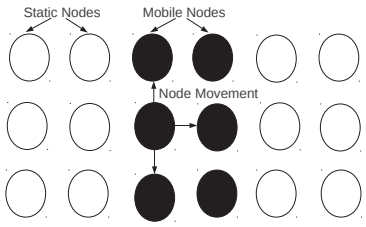


Figure 4.1. Topology: 18 node network

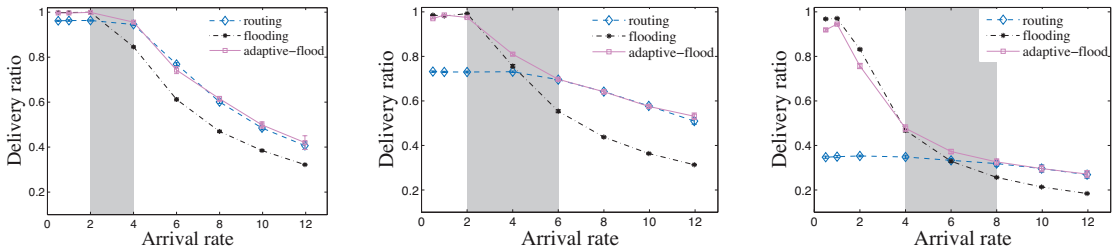
node i, we assume that the probabilities of i receiving that packet on its incoming DAG links are independent of one another. This is clearly an approximation, since two input links at i may share common upstream nodes in the DAG. Let α_{if} be the probability that a packet reaches node i for flow f. Let us consider any node j in V_f and let U_j denote the list of the nodes in V_f appearing before j in this ordering. We approximate the probability of a packet reaching j by:

$$\alpha_{jf} = 1 - \prod_{i \in U_j} (1 - \alpha_{if} p_{ij} a_i) \tag{4.2}$$

Equation (4.2) takes into account the fact that the packet can be received along multiple incoming links. It is also takes the successful link transmission probabilities and the packet-passage probabilities into account. Traversing V_f in order ensures that when the algorithm calculates α_{jf} for node i, α_{jf} of all nodes j in U_j has already been computed. $\Phi_{f_s f_d} \alpha_{jf}$ thus denotes the incoming rate (I_{fj}) at node j and the goodput (g_f) for flow f is given by $\Phi_{f_s f_d} \alpha_{f_d f}$.

4.5 Simulation Results

In this section, we report on simulations comparing the performance of the *adaptive-flood* algorithm with pure network-wide routing and flooding. We find that *adaptive-flood* captures the best of both approaches (routing and flooding), achieving performance equivalent to (and sometimes better than) that of network-wide routing or flooding alone.



(a) Delivery ratio (scenario 1) (b) Delivery ratio (scenario 2) (c) Delivery ratio (scenario 3)

Figure 4.2. Delivery ratio with different sets of flows for an 18 node network

4.5.1 Simulation Scenarios

Our simulations are conducted on a grid topology with r rows and c columns. An example topology with 18 nodes is shown in Figure 4.1; this specific topology was used to generate the results in Figure 4.2. In Figure 4.1, the nodes colored white are stationary while the ones colored black are mobile. Thus there are two regions with stationary nodes separated by an intervening mobile region. We chose this topology in order to stress-test and study our algorithms, ensuring that we could create controlled scenarios in which source-destination flows pass through both static and mobile regions. We also present results for a larger 48-node topology later in this section. At the beginning of the simulation, there is one node per grid location.

Recall that we have assumed a time-slotted system and consider three time periods of different granularity - slots, intervals and epochs. A *slot* is the time required for a packet transmission to occur. Any node can transmit directly only to nodes in the four adjacent grid positions. When a node operates as a router and has a packet to forward, the packet will be delivered to the next-hop node if that designated next-hop node is located in any one of the adjacent positions; otherwise it will be dropped. When a node operates as a flooder, its transmitted packet can be received by neighbors located in any of the four adjacent grid positions.

An *interval* consists of multiple slots. A mobile node moves equi-probably to any of the adjacent positions on the grid (up, down, left or right) at the beginning of an interval. However, mobile node movement is confined to the mobile region (the black

region in Figure 4.1). Multiple nodes may be in the same grid location during some intervals. If a mobile node is at the border of the mobile region and its movement would take it out of mobile region, it reflects and moves in the opposite direction.

Recall that the time period of the longest duration is an *epoch,* consisting of multiple intervals. Unicast routes are calculated using Dijkstra's algorithm at the beginning of each epoch. The *adaptive-flood* algorithm also executes at this time granularity, classifying nodes as routers and flooders. In Dijktsra's algorithm, link weight values are equal to $1/p_{ij}$, where p_{ij} is the fraction of intervals in the previous epoch that node i and j were in adjacent or the same grid positions; the value of p_{ij} between two adjacent stationary nodes i and j is thus always 1. The link success probability matrix $\mathbb{P}$ is populated at the beginning of the epoch, before Dijkstra's algorithm is executed. In order to focus on mobility, we assume all variation in link quality (i.e., the ability of one node to send to another) is only due to mobility.

Since our goal is to investigate the performance gains of the *adaptive-flood* algorithm (containing both routing and flooding nodes) and will be compared against pure routing and pure flooding, we make several simplifying assumptions in the simulator. We do not model the effect of interference in the network, assuming that a node can receive multiple packets in the same time slot (one along each of its incoming links); this would be possible when a node has multiple interfaces operating on different channels, when a node has multiple directional antennae, and in some CDMA settings. A node can, however, send only one packet in one time slot. If a node is a router, the packet will be received and processed only by the designated next hop; if it is a flooder, its transmitted packet can be received by all its neighbors present in adjacent grid positions. In our simulation, we model data plane forwarding only; since *adaptive-flood* and routing all take advantage of common link state control information, we do not explicitly simulate link state transfer.

All nodes have a single finite buffer of size 300 packets, and packets arriving to a full buffer are dropped. Each data point in our simulation is obtained for the same

number of total exogenous packet arrivals (15000 packets). The number of intervals is 30 and the number of slots per interval is 10. The number of epochs is adjusted so that the expected number of arrivals is 15000 exogenous packets.

We first report results for an 18-node network in Figure 4.1 for different sets of network flows. In each case, we vary the exogenous arrival rate and study two different performance metrics: overall network goodput and delivery ratio. As discussed earlier, a flow's goodput is the average number of unique packets delivered to the destination per time slot; the delivery ratio is the ratio of the total number of unique packets delivered to the total number of exogenous packet arrivals for the entire duration of the simulation. The arrival rate is the expected total number of exogenous packet arrivals per time slot at a source node. For each flow, each source node has the same probability of generating an exogenous packet arrival at the beginning of a time slot. We multiply this probability by the total number of flows to obtain the exogenous packet arrival rate. We increase the arrival rate by increasing the probability of an exogenous packet arrival. We report results as mean values obtained after multiple runs; the length of the error bars denotes twice the standard deviation of the delivery ratio.

We study three scenarios (with different sets of flows) for the 18-node network.

- **Scenario 1:** We consider mostly short (2-hop) flows. Every node in the static region has a single 2-hop flow destined to a randomly chosen other node in the same static region (12 flows in all). There are also 3 single-hop flows in the mobile region.

- **Scenario 2:** We have 12 flows in the static regions, as before. There are also 12 short flows originating from, and destined to, the mobile region and 3 flows originating from one static region and destined to the other static region.

- **Scenario 3:** In contrast to the other two cases, we have 25 flows in all, (some destined from one static region to other, some within the mobile region and some between mobile and static regions).

In the first scenario, most flows are confined to the static region; in the second there is a mix of flows in the static and mobile region; while in the third scenario, flows either cross, or are destined to, the mobile region. Hence, the main difference among the three scenarios is that the overall reliability of routes decreases, progressing from the first scenario to the third. Consequently, one would intuitively expect routing to generally outperform flooding in the first scenario, while the opposite would occur in the third scenario.

4.5.2 Simulation Results: Comparing Routing, Flooding and *Adaptive-flood*

Comparison of pure routing and flooding. Figure 4.2 shows the delivery ratio of the different algorithms for the above three scenarios. For scenario 1 (Figure 4.2(a)), we observe that pure (i.e., network-wide) routing performs comparable to pure flooding in the low arrival rate regime but then outperforms pure flooding as the arrival rate increases. The difference in delivery ratio at low arrival rate is due to the fact that in case of pure routing, packets are dropped in the mobile region; in the case of pure flooding, packet duplication via flooding ensures that at least one copy of most packets get delivered to the destination. The reason for the relatively poorer performance of flooding at higher arrival rates is that as the network becomes congested, duplicate packets cause other packets to be discarded at intermediate routers, resulting in decreased goodput.

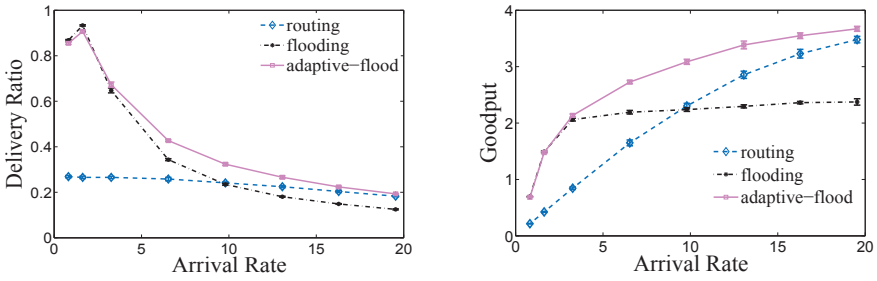
In scenario 2 (Figure 4.2(b)), we observe that pure flooding outperforms pure routing at low arrival rates, while the relative performance ordering is reversed at higher arrival rates. Since approximately half of the flows are in the mobile region (and these flows have less reliable paths), pure routing has a low delivery ratio at low

arrival rates. Once again, increased congestion results in poor performance of flooding at higher arrival rates. In scenario 3 (Figure 4.2(c)), as end-end path reliability is low, pure routing performs poorly, marginally overtaking pure flooding as the arrival rate increases. The three scenarios thus demonstrate situations when pure routing outperforms pure flooding, and vice versa.

***Adaptive-flood* outperforms both pure flooding and pure routing.** Next, we turn our attention to the performance of our *adaptive-flood* algorithm[1]. The shaded regions in Figure 4.2 indicate the arrival rate regime where the *adaptive-flood* outperforms both routing and flooding. It is evident from the figure that while flooding and routing perform well at low and high arrival rates respectively, the *adaptive-flood* algorithm achieves performance equivalent to (and better in the shaded regions) than that of pure routing or flooding alone. For example in Figure 4.2(b), the performance of *adaptive-flood* exceeds that of both routing and flooding in the shaded region, for arrival rates between 2 and 6 arrivals per time slot. The superior performance of the *adaptive-flood* algorithm can be attributed to the fact that it dynamically adapts the number of flooders selected based on the arrival rate. For example in Figure 4.2(b), the algorithm selects around 10 nodes as flooders when the arrival rate is 2 and selects 2.11 nodes on average as flooders when the arrival rate is 12. We noted that *adaptive-flood* also often selects stationary nodes as flooders. Turning stationary nodes into flooders can present multiple entry points into the mobile region. Also if a given stationary node is congested because of a large number of flows through it, turning other stationary nodes into flooders can help find additional paths for these flows, thus increasing goodput.

Performance results for 48-node network with large number of flows. We also conducted experiments on a larger 48-node network, with nodes arranged in a

[1]In some cases, packet-passage probabilities in the fixed point iteration in the *calculate-goodput* function of *adaptive-flood* algorithm do not converge to a fixed point, often oscillating between two sets of values. In such cases we select one of the sets of values and use it to determine the total goodput.



(a) Delivery ratio (b) Goodput

Figure 4.3. Delivery ratio and goodput for a 48 node network

grid with 4 rows and 12 columns. In this network each node originates flows destined to every other node. The static and mobile areas thus consist of 4*4 grids. Figures 4.3(a) and 4.3(b) shows the delivery ratio and goodput for this network with all nodes sending packets to all other nodes. We observe that the *adaptive-flood* algorithm outperforms pure routing and flooding both in terms of delivery ratio and goodput. Interestingly, although the delivery ratio decreases with increasing exogenous arrival rate, the goodput increases since the absolute number of packets delivered increases with higher arrival rate. We also conducted experiments for the 18-node network scenario, where each node originates flows destined to every other node and observed similar results.

4.6 Conclusion

We have studied the problem of forwarding in heterogeneous mobile networks that comprise both stable as well as highly dynamic components and in which uniform routing or flooding at all network nodes does not perform well. Instead of designing a new protocol, we leveraged past efforts and proposed a simple greedy algorithm (*adaptive-flood*) that individually determines for each node whether it should operate as a router or a flooder based on considerations such as link quality, the amount of traffic traversing it, and the effect of turning a router into a flooder on overall goodput. We demonstrated via simulation that our *adaptive-flood* algorithm yields

performance equivalent to, and often significantly better than, that of baseline routing or flooding alone. In future, we plan to investigate the performance gains achievable by preferentially routing or flooding packets based on their destination.

Algorithm 1 *adaptive-flood* router/flooder classification

function F = *adaptive-flood*$(G, \mathbb{P}, \Phi, \mathbb{N}, \mathbb{H}, \mathscr{F}, \tau)$
  $F = [\,]$
  $\mathscr{G}_F$ = *total-goodput*$(F, G, \mathbb{P}, \Phi, \mathbb{N}, \mathbb{H}, \mathscr{F}, \tau)$
  while $|F| \neq |N|$ **do**
    $F' = [\,]$
    for all $s \notin F$ **do**
      $T = F + [s]$
      $\mathscr{G}_T$ = *total-goodput*$(T, G, \mathbb{P}, \Phi, \mathbb{N}, \mathbb{H}, \mathscr{F}, \tau)$
      if $\mathscr{G}_T > \mathscr{G}_F$ **then**
        $F' = s$, $\mathscr{G}_F = \mathscr{G}_T$
    if $F' == [\,]$ **then**
      return F
    else
      $F = F + [F']$
  return F

function $\mathscr{G}_F$ = *total-goodput*$(F, G, \mathbb{P}, \Phi, \mathbb{N}, \mathbb{H}, \mathscr{F}, \tau)$
  $\vec{a}^0 = 0$, $l = 1$
  for all $f \in \mathscr{F}$ **do**
    D_f = *calculate-DAG*$(f, F, G, \mathbb{N}, \mathbb{H})$
    V_f = *topological-sort*(D_f)
  while (true) **do**
    for all $f \in \mathscr{F}$ **do**
      $\vec{I}_f$ = *calculate-incoming-rate*$(D_f, V_f, \Phi, \mathbb{P}, f, \vec{a}^{l-1})$
    for all $s \in N$ **do**
      $R_s = \sum_{f \in \mathscr{F}} I_{fs}$
      $a_s^l = \min\{1, \frac{C_s}{R_s}\}$
    if $\max|\vec{a}^l - \vec{a}^{l-1}| < \tau$ **then**
      return false
    $l = l + 1$
  for all $f \in \mathscr{F}$ **do**
    g_f = *calculate-goodput*$(D_f, V_f, \Phi, \mathbb{P}, f, \vec{a}^l)$
  $\mathscr{G}_F = \sum_{f \in \mathscr{F}} g_f$
  return $\mathscr{G}_F$

function D_f = *calculate-DAG*$(f, F, G, \mathbb{N}, \mathbb{H})$
  $w_{ij} = 0, \forall i, j \in N$, $E' = \emptyset$
  $X = [f_s]$, $O = [f_d]$
  while $X \neq [\,]$ **do**
    $[m] = head(X)$
    $O = O + [m]$
    if $m \in F$ **then**
      for all $i \in N_m$ **do**
        if $i \notin O$ **and** $i \notin X$ **then**
          $X = X + [i]$
          $w_{mi} = 1$
    else
      if $H_{mf} \notin O$ **and** $H_{mf} \notin X$ **then**
        $X = X + [H_{mf}]$
        $w_{mi} = 1$
    $X = X - [m]$
  for all $w_{ij} = 1$ **do**
    add edge $\{i, j\}$ *to* E'
  $D_f = (N, E')$
  return D_f

CHAPTER 5

QUALITY OF EXPERIENCE MANAGEMENT OF MULTIPLE VIDEO STREAMS

5.1 Introduction

With the deployment of broadband wireless networks, the popularity of multimedia content on mobile devices is expected to significantly increase. A large portion of multimedia traffic is forecasted to be recorded videos such as movies, YouTube videos, and TV shows [20]. The inherent variability of both the wireless channel and the bit rate of compressed videos makes streaming videos on wireless networks a challenging task. This work investigates how multiple Variable Bit Rate (VBR) videos can be scheduled over a time-varying wireless channel while still maintaining a good QoE at the mobile clients.

A wireless video streaming system consists of a video server connected to a base station over a high bandwidth wired backbone link and clients at Mobile Stations (MS) that communicate with the Base Station (BS) using a wireless channel (Figure 5.1). The server stores pre-encoded videos, and upon receiving requests, streams videos to the requesting clients. A video stream is composed of a sequence of frames that the client buffers and plays according to their playout times. If a frame is not received by its playout time, the client degrades the quality of the displayed video and/or may *stall* the video to wait for more frames to arrive. Here we consider systems that stall in response to delayed frames. Namely, we consider the general case of VBR videos being streamed with the rate available to each wireless client varying over time.

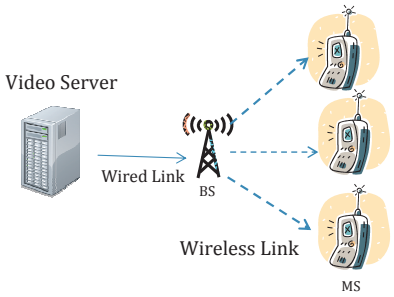


Figure 5.1. A video streaming system

In this chapter, we consider a wireless video streaming system where multiple mobile clients are streaming different VBR videos from a base station. Our goal is to develop a fair packet scheduling algorithm at the base station, for packet transmission over the wireless channel that minimizes playout stalls across all mobile clients. We assume that time is divided into slots and scheduling decisions are taken at beginning of an epoch (which consists of multiple slots). Prior work [29, 49] that studies the impact of video quality on user behavior demonstrate quantitatively (based on real world datasets) that frequent stalling can result in users abandoning their video streams. The number of stalls per client thus appears to be a good metric to capture the quality of user experience and minimizing it can lead to reduced user abandonment.

We formulate this problem as an optimization problem that takes into account the varying rate of the video streams and wireless channel at the clients and allocates slots so as to maximize the minimum playout lead among all videos in an epoch. Our contributions are as follows. (a) We show that the optimization problem of maximizing the minimum lead is NP-complete even for two videos. (b) We develop a fast application-playout lead aware greedy scheduling algorithm that is sub-optimal for wireless channels, but show that this algorithm is optimal when the channel quality of a user does not vary within an epoch, even with different users possibly having different channel quality. (c) Finally, we conduct trace-driven simulations with publicly available MPEG-4 video traces, and wireless channel quality

traces that we collected from a WiMAX test-bed. Our simulations demonstrate that the greedy algorithm achieves a fair distribution of stalls across clients while maintaining a low average number of stalls per client. In particular, when the wireless network is average-provisioned as compared to the total average bit rate of the videos (a case that is interesting in practice), the greedy algorithm reduces the number of stalls by a factor of 3, when compared to other algorithms in our simulations. We also study the sensitivity of the greedy algorithm against changes in epoch duration, client's stall-recovery scheme, different video traces and poor channel conditions.

The rest of this chapter is organized as follows. We provide an overview of related work in Section 5.2. Sections 5.3 and 5.4 describe the video streaming system characteristics and scheduling problem formulation respectively. Hardness results are stated in Section 5.5, followed by the greedy algorithm in Section 5.6. The evaluation framework and experimental results are described in Section 5.7 and Section 5.8 respectively. We discuss the adaptability and scalability of the greedy algorithm in Section 5.9 and conclude the chapter in Section 5.10.

5.2 Related Work

Although compression techniques reduce the mean bit rate of video streams, it introduces considerable rate variability over several time scales [35, 71]. Resource allocation for VBR video streaming has been studied extensively for wired networks. Smoothing video transmission is one of the primary techniques used for reducing the effect of bit rate variability. By pre-fetching some of the initial video frames before their display times, smoothing techniques can minimize the effect of bit rate variability under various resource constraints, such as peak bit rate, client buffer size, and initial playout delay [50, 63, 80, 83].

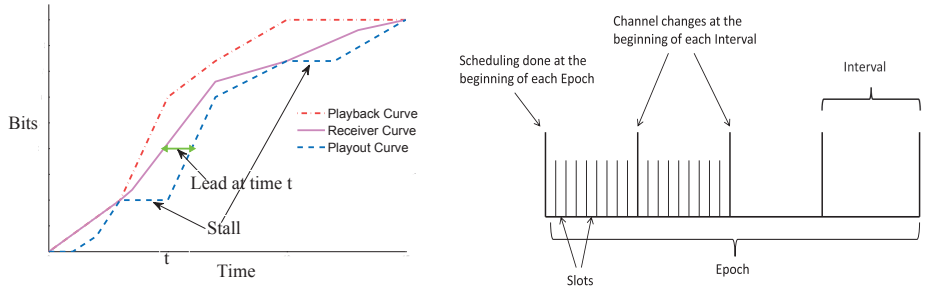
Rate allocation for multiple video streams is a well studied problem [37, 57, 75, 86, 103]. [75] investigates minimizing rate variability when transmitting multiple video streams given the client buffer size in a high-speed wired network. In the RCBR

service introduced in [37], the rate of each video is renegotiated at the end of each interval to provide statistical QoS guarantees. [103] presents a call-admission scheme at a statistical multiplexer and bounds the aggregate loss probability. A linear programming model is proposed in [86] to compute a globally optimized smoothing scheme to stream multiple videos. [57] derives bounds on the dropped frames, delay and buffer requirement that can be obtained by statistically multiplexing VBR streams at the video server by using a two-tiered bandwidth allocation. Although our algorithm performs periodic rate allocation among multiple video streams, our work differs from the above papers in two crucial aspects: our primary objective of fairly managing playout stalls across the videos, and our focus on time-varying wireless channel.

Scheduling algorithms for improving user QoE in cellular networks have also been designed ([82, 88] and the references therein). Our work is closely related to [82, 88], where the authors have proposed greedy algorithms for optimizing Mean Opinion Score (MOS) for resource allocation in wireless networks (3G and LTE). The main difference between our work and the above mentioned papers is the user QoE metric - we specifically consider video stalling whereas they mainly consider MOS. Another aspect that we consider in this work which is not explored in [82, 88] is that we demonstrate the hardness of our scheduling problem. In [44], the authors consider the problem of transmitting multiple VBR videos to mobile clients, but the work focusses on maximizing bandwidth utilization while reducing energy consumption and does not address the issue of video playout stalling.

Our work is closest to the work presented in [52, 53] for managing stalls. Given the initial playout delay and the receiver buffer size, [53] determines upper and lower bounds on the probability of stall-free display of a video. [52] develops an analytical framework to find the stall distribution while streaming a VBR video over a wireless channel. However, unlike our work, both papers consider a single video stream.

Gracefully degrading the quality of the displayed video when network conditions deteriorate is an active area of research. Scalable video coding for [67, 78, 87] and



(a) (b)

Figure 5.2. (a) Playback, receiver and playout curves of a video stream (b) Epochs, intervals, slots

prioritization of packets [45] are two such methods used for video streaming. Recently, there have been measurement studies on the quality of videos streamed over deployed WiMAX networks [84]. The authors in [43] compare video streaming over a WiMAX network and a wired broadband network (with equal reserved rates), and demonstrate that with fine-tuning of network parameters, performance over WiMAX is comparable to the wired networks in terms of the network QoS metrics. Recently, in [92], authors have investigated the impact of WiMAX network parameters on the end-user's QoE in video streaming. However, none of these papers consider mechanisms to multiplex video streams.

5.3 Network Model

In this section, we describe the video streaming system and our wireless channel model.

5.3.1 Streaming System Characteristics

We consider a video streaming system similar to [52], as shown in Figure 5.1. We assume that the server simultaneously and separately streams n videos $v_1, \ldots, v_n$ to n clients $1, \ldots, n$ via the base station. A video object is composed of a sequence of frames that are displayed at a constant frame rate by the client. However, since the

size of each frame varies significantly, the required transmission rate of a video varies with time.

For a video v_i, its *playback curve* $p_i(t)$ specifies the cumulative data needed in the first t time units of the video playout, in order to play the video without interruptions. Thus, $p_i(t)$ is the sum of the sizes of the first tF frames of the video, where F denotes the frame rate. The playback curve is a characteristic of a video and is independent of the underlying channel.

For a client i, its *receiver curve* $G_i(t)$ specifies the cumulative amount of data it has received by time t. The cumulative amount of data played out by time t is given by its *playout curve* $O_i(t)$. Note that $G_i(t)$ and $O_i(t)$ depend on the channel conditions and transmission scheme at the base station, and $O_i(t)$ additionally depends on the buffering scheme of the client. In particular, unlike the playback curve, the playout curve may vary between different streaming instances of the same video. Figure 5.2(a) shows an example playback, receiver, and playout curve for a client. The notation used in this chapter is summarized in Table 5.1.

We assume that clients have sufficient buffer space to buffer frames that have been received but not yet displayed. If the next frame to be displayed is not received within its playout time, the client stalls playout for a certain duration during which it continues to buffer data received from the server. It resumes playout based on its *stall-recovery buffering scheme*. Common buffering schemes include: (i) waiting for a fixed amount of time, (ii) waiting for a fixed amount of future playout data, and (iii) waiting for a fixed number of future playout frames.

5.3.2 Timing Consideration

We assume a broadband wireless system (such as WiMAX/LTE) in which scheduling decisions are taken at the time granularity of *epochs*. Epochs are divided into *intervals* (Figure 5.2(b)). The duration of an interval is small enough so that the channel state does not change significantly within an interval. Intervals are divided

Notation	Definition
n	number of clients
p_i, G_i, O_i	playback, receiver and playout curves (respectively)
R, A	channel rate vector and transition matrix (respectively)
$N_{ep}^{in}, N_{in}^{sl}, N_{ep}^{sl}$	#intervals/epoch, #slots/interval and #slots/epoch (respectively)
I_i	initial probability distribution of channel state
F	frames played out per second
Y_i, V_i	#bits and #complete frames (respectively) transmitted in epoch
L_i	lead at the end of the epoch
Φ_i	inverse playback curve
r_{ij}	#bits that can be transmitted to client i in slot j

Table 5.1. Notation (note: subscript i refers to client i and # denotes 'number of')

into a fixed number of (transmission) *slots* that are allocated to clients. The base station can transmit to at most one client in a slot. Depending on channel conditions, each client receives a certain bit rate in the allocated slots. Following [52], we assume that the wireless channel is error-free due to an error control mechanism such as ARQ.

5.3.3 Channel Model

We model the wireless channel between each client and the base station (i.e., bit rate received at the client), as a discrete-time Markov chain. We assume that the Markov chain changes state at the beginning of an interval. The possible channel states are identified by the transmission rates $R = (r_1, r_2, \ldots, r_K)$ (R is also called the rate vector). Here r_i denotes the number of bits that can be transmitted in a time slot when the channel is in state i [52]. As the Markov chain changes state at the beginning of each interval, the bit rate for a client remains the same in all slots within an interval. Let A denote the transition matrix of the Markov chain. We assume A is available at the server, with each client's channel modeled as an independent Markov chain.

5.4 Modeling the Scheduling Problem

Our goal in this chapter is to design a scheduling algorithm that executes periodically (at the beginning of each epoch) at the base station. Informally, the goal of the scheduling algorithm is to transmit video data to clients (some clients being allocated more transmission slots in an interval than others) in order to minimize playout stalls among all clients; we will precisely formulate this optimization problem shortly.

Minimizing the number of stalls within an epoch directly is difficult as it can incur high complexity. To determine whether a client will stall or not during an epoch, it is necessary to determine the probability of the client receiving a specific number of bits during that epoch. Computing this probability is hard as one has to deal with summation of dependent random variables (the number of bits received in an interval for a client follows a Markov chain). We discuss this issue further, later in this section.

To motivate our strategy for allocating base station transmission slots to clients, we note that a client's current buffer size (in bits) indicates its vulnerability to stalling; the smaller the buffer, the more likely is the occurrence of a stall. However, for VBR videos, a client's current buffer size may be an inaccurate indicator of this vulnerability, since it does not consider the amount of data needed to play the next few frames. On the other hand, the *playout lead* of the video, i.e., the duration of additional time a client can play the video using only its currently buffered data, takes into account the VBR nature of the video.

Therefore, in our scheme the server attempts to prevent stalls by fairly maximizing the playout lead among all receivers. To ensure that stalls are evenly distributed across all videos, slots are allocated such that the minimum lead among all clients is maximized. In contrast, if the scheduler goal were to maximize the minimum current buffer size (in bits), it would refrain from allocating bits to a client with large buffer, with the effect that this client could stall multiple times in succession if that large number of bits corresponded to short amount of played-out video. Indeed, we

will see later that using current buffer size as the optimization metric can result in non-uniform allocation of stalls.

To implement this scheduling algorithm, we assume that at the beginning of each epoch, clients communicate their channel state to the server, as already done in numerous wireless standards. Clients also communicate their playout lead to the server. The initial state of the client's channel and the transition matrix of the Markov chain is used to determine the expected rate available to clients in different intervals during an epoch.

We do not consider client channel state during previous epochs while scheduling slots for the current epoch. As the server obtains fresh client channel state at the beginning of each epoch along with client playout lead, considering client channel information from previous epochs does not provide any additional value given a Markovian channel model. We also do not consider subsequent epochs because wireless channel prediction for longer than an epoch may not be accurate.

Preliminaries. Let N_{ep}^{in} and N_{in}^{sl} denote the number of intervals in an epoch and the number of slots in an interval respectively. Thus the total number slots in an epoch is $N_{ep}^{sl} = N_{ep}^{in} N_{in}^{sl}$. Each video is played at the constant rate of F frames per second.

Consider the i^{th} client in a particular epoch. Let I_i be the state vector denoting the probability distribution of channel states at the i^{th} client at the beginning of the epoch. Then, given the Markov channel model, the probability distribution of the channel state at the client at the beginning of the k^{th} interval in the epoch is $I_i A^k$.

Let X_{ik} be the random variable denoting the number of bits that can be transmitted to client i in any slot of the k^{th} interval. Then, its expectation $E[X_{ik}]$ is the dot product of $I_i A^k$ and the channel transmission rate vector R. Suppose that the server assigns s_{ik} slots to client i in the k^{th} interval. Then the random variable Y_i for the number of bits transmitted to client i in this epoch can be expressed as $\sum_{k=1}^{N_{ep}^{in}} s_{ik} X_{ik}$. From linearity of expectation, $E[Y_i] = \sum_{k=1}^{N_{ep}^{in}} s_{ik} E[X_{ik}] = \sum_{k=1}^{N_{ep}^{in}} s_{ik} I_i A^k R$.

Before proceeding further, we discuss briefly why determining the probability of client i stalling (p_i) in an epoch is computationally expensive. Let c_i be the amount of data that client i has to receive by the end of the epoch to avoid stalling. Then $p_i = 1 - P[Y_i > c_i] = 1 - P[\sum_{k=1}^{N_{ep}^{in}} s_{ik}X_{ik} > c_i]$. It is difficult to determine the above probability because X_{ik} are dependent random variables. To determine the probability of a client stalling, it is necessary to determine the joint distribution of $(X_{i1}, X_{i2}, ...X_{iN_{ep}^{in}})$ which is computationally expensive.

Playout Lead. The playout lead of a client at any given time is the additional duration of time that its video can be played out using the data available in its buffer; it is equal to the number of complete frames in the client buffer divided by the frame rate F. Let l_i denote the playout lead of client i at the beginning of an epoch. Let o_i denote the total amount of *time* for which the video has been played out at the client i (calculated from the playout curve). Let g_i be the time for which the data received at the client can be played out (calculated from the playout curve). Thus $l_i = g_i - o_i$, is a known constant value at the beginning of the epoch. Note that o_i and g_i account for the data consumption at the client and the amount of data received during the previous epoch, respectively. In Figure 5.2(a), the green bar denotes the playout lead for the video at time t.

Let L_i be a *random variable* denoting the playout lead of the video at the end of an epoch (assuming that the video stalls during the epoch), and V_i be the *random variable* denoting the number of additional frames that can be *completely* received by the end of the current epoch. Then, $L_i = l_i + (V_i/F)$.

Inverse Playback Curve. For an epoch, we now define a deterministic function that maps the number of bits received to the number of *complete* frames received. The *inverse playback curve* Φ_i for each video i is defined as follows: if b bits are transmitted to video i in this epoch, then the number of complete frames that are received increases by $\Phi_i(b)$ at the end of the epoch. Thus, $V_i = \Phi_i(Y_i)$. (Note that partially transmitting a frame does not increase the lead of the video.)

Estimating expected playout lead. We know that $L_i = l_i + (V_i/F)$. As l_i is a known constant at the beginning of an epoch, $E[L_i] = l_i + E[V_i]/F$. Now $E[V_i] = E[\Phi_i(Y_i)]$. Unfortunately, since the video frame sizes can vary, the mapping Φ_i from bits to frames is non-linear, and hence, we approximate $E[V_i] \approx \Phi_i(E[Y_i])$. The main benefit of this approximation is that computation of $\Phi_i(E[Y_i])$ is simple, making the execution of our greedy algorithm in Section 5.6 fast. Thus the expected lead is estimated as $E[L_i] \approx l_i + (1/F)\Phi_i(E[Y_i]) = l_i + (1/F)\Phi_i(\sum_{k=1}^{N_{ep}^{in}} s_{ik} I_i A^k R)$.

The Video Scheduling Problem: Our aim, at the beginning of an epoch, is to assign slots with the goal of maximizing the minimum expected lead at the end of the epoch. This problem can be expressed as follows:

Objective: $\max \min\{E[L_1], \ldots, E[L_n]\}$

subject to the constraints:

1. $\sum_{i=1}^{n} s_{ik} = N_{in}^{sl}, \forall k \leq N_{ep}^{in}$
2. $s_{ik} \geq 0$, $\forall i \leq n, \forall k \leq N_{ep}^{in}$ (5.1)

5.5 Hardness Result

We now investigate the multiplexing problem described in the previous section. We formulate it as a combinatorial problem and call it *Lead-based Multiple Video Transmission (LMVT)* problem. (We assume that all slots from all intervals of an epoch are numbered sequentially from 1 to N_{ep}^{sl}.)

Inputs. At the beginning of an epoch, the i^{th} client has an initial lead of l_i seconds i.e., its buffer contains data corresponding to the $F * l_i$ frames received after the last played frame. Let r_{ij} be the expected number of bits that can be transmitted to client i in slot j. Thus if slot j belongs to interval k, then $r_{ij} = I_i A^k R$. For ease of presentation, we also call r_{ij} the rate of client i in slot j.

The LMVT Problem. Given the above inputs, we need to find a slot allocation that maximizes the minimum lead among all clients at the end of the epoch. Here, 'lead' refers to the expected playout lead in Eqn. (5.1). A slot allocation for an epoch essentially specifies for each slot, the client to which that slot is allocated.

We now show that the following decision version of LMVT is NP-complete: given a constant L, does there exist a slot allocation such that all videos have a lead of at least L seconds at the end of the epoch?

Lemma 6. *The decision version of the LMVT problem is NP-complete.*

Proof. Clearly the decision version of LMVT is in NP. We show that the problem is NP-complete by reducing subset-sum [23] to LMVT. The decision version of subset-sum is as follows: given a set S of positive integers $\{x_1, \ldots, x_P\}$, and a positive integer B, does there exist a subset $S' \subseteq S$ such that the sum of elements in S' is exactly B [23]. Let Π denote the index set $\{1, \ldots, P\}$ and let $Y = \sum_{j \in \Pi} x_j$. It is assumed $B < Y$, otherwise the subset-sum instance is trivial to solve.

For an instance of subset-sum, we construct an instance of LMVT as follows. Let Π be the set of slots in the epoch with one slot per interval. Let there be two videos v_1 and v_2. Let the set S map to the rates available in each slot as follows. Let x_j be the rate available to both the videos in slot j i.e., $x_j = r_{1j} = r_{2j}$. Let the initial lead for both the videos be zero and both play at the rate of 1 frame/second. Let the inverse playback curve of v_1, $\Phi_1(b)$, be a function which is 0 for $b < B$, and 1 for $b \geq B$. An example of such a video is one that contains a single frame of size B bits. Similarly, let $\Phi_2(b)$ be a function which is 0 for $b < Y - B$, and 1 for $b \geq Y - B$. Let the required minimum lead L for each video be 1.

We now show that the above instance of subset-sum has a solution if and only if the constructed instance of LMVT has a solution.

Subset-sum to LMVT: Suppose the subset-sum problem instance has a solution given by a subset S' of S. We construct a solution for the instance of LMVT as follows: for each $j \in \Pi$, if $x_j \in S'$ then we allocate the slot j to video v_1, else we allocate the slot to video v_2. In either case, x_j bits are transmitted in slot j for the allocated video. Since, the sum of all elements in S' is B, this allocation results in transmission of B

bits and $Y - B$ bits for v_1 and v_2, respectively. Thus, both videos have a lead 1 at the end of the epoch.

LMVT to Subset-sum: For the reverse direction, assume that we have a solution of LMVT in which both the video have a lead of 1. Thus, v_1 and v_2 are transmitted at least B bits and $Y - B$ bits, respectively. In the solution, suppose that $\Pi_1 \subseteq \Pi$ be the set of slots that are allocated to v_1, and the remaining slots are allocated to v_2.

Note that, for each $j \in \Pi_1$, the number of bits transmitted to v_1 is $r_{1j} = x_j$. Since at least B bits are transmitted to v_1, $B \leq \sum_{j \in \Pi_1} r_{1j} = \sum_{j \in \Pi_1} x_j$. Similarly, for video v_2, $Y - B \leq \sum_{j \in \Pi \setminus \Pi_1} r_{2j} = \sum_{j \in \Pi \setminus \Pi_1} x_j$. However by construction, $\sum_{j \in \Pi} x_j = Y$, so $\sum_{j \in \Pi_1} x_j = B$ and $\sum_{j \in \Pi \setminus \Pi_1} x_j = Y - B$. Thus, the subset $\{x_j \ : \ j \in \Pi_1\}$ of S is a solution of the subset-sum instance. □

For a constant number of videos, we have designed a pseudo-polynomial time algorithm to optimally solve LMVT using dynamic programming. The time complexity of the dynamic programming algorithm is high; it is exponential in the number of videos.

Lemma 7. *For a constant number of videos, there is a pseudo-polynomial time algorithm to optimally solve LMVT.*

Let us now present an optimal dynamic programming algorithm for LMVT. We present a brief description of the algorithm here while a detailed proof is presented in Appendix B.1.

We begin by introducing a simple definition. A transmission vector (or *Tx-vector*) is an n-tuple $< a_1, \ldots, a_n >$, where the i^{th} element indicates the number of bits to be transmitted to video i. For a Tx-vector T, we denote by $T[i]$ the i^{th} element of T. For a given number of total slots, say z, and a Tx-vector T, we say that T is z-feasible if there is a slot allocation such that, for each $1 \leq i \leq n$, video v_i receives a total of $T[i]$ bits in the allocation.

Our dynamic programming algorithm iterates over the number of slots m that varies from 1 to N_{ep}^{sl} and determines the feasible Tx-vectors. In each iteration (say for slot m), the algorithm does the following. 1) It computes the m-feasibility of the Tx-vectors based on the $(m-1)$-feasibility of a subset of Tx-vectors (computed in the previous step). 2) For each feasible Tx-vector for slot m, then computes the minimum lead considering all videos. 3) For each feasible Tx-vector T, stores an *allocation* pointer to the Tx-vector from the previous step from which its m-feasibility was computed.

Finally, in the iteration when $m = N_{ep}^{sl}$, we maintain a pointer to determine the N_{ep}^{sl}-feasible vector with the maximum value of its min-lead among all the N_{ep}^{sl}-feasible vectors. Thus, at the end of algorithm, we obtain a pointer to a N_{ep}^{sl}-feasible Tx-vector T' with the maximum value of min-lead, and we follow the N_{ep}^{sl} *allocation* pointers from T' to $< 0, \ldots, 0 >$ to obtain an optimal slot allocation.

5.6 A Lead-Aware Greedy Algorithm

We now present a fast lead-aware greedy algorithm for the LMVT problem. The algorithm is optimal for LMVT when channel conditions remain constant within an epoch, but different users may have different channel quality (as shown in Lemma 8 below). Later in our simulations, we numerically evaluate the algorithm for the general case when the channel conditions of users may vary.

Lead-Aware Greedy Algorithm: Starting with the initial playout leads of the videos and all the slots in the epoch to be allocated, the greedy algorithm allocates slots one by one (Algorithm 2) as follows. In each iteration, the algorithm selects a video i with the minimum expected lead, such that video i has the lowest id among the videos with the minimum lead. Then the algorithm allocates client i a slot j in which client i has the highest rate r among all available (yet to be scheduled) slots. Before moving to the next iteration, slot j is marked unavailable for all videos, and the expected lead of client i is increased corresponding to the transmission of r bits to video i using the

inverse playback curve Φ_i (line 12 of Algorithm 2). The algorithm iterates until there are no available slots in the epoch. Note that, the client with the minimum lead that is selected by the algorithm may change between any two slot allocations. Hence, the algorithm allocates slots one by one even though each client's channel condition does not change within an interval.

Algorithm 2 A greedy algorithm (executed at the beginning of each epoch)

1: **function** initialization
2: $AvailableSlots \leftarrow \{1, \ldots, N_{ep}^{sl}\}$; $j \leftarrow 1$
3: $\forall$ client i: $lead_i \leftarrow$ initial lead of i; $I_i \leftarrow$ initial state distribution; $rcvbits_i \leftarrow 0$
4: $\forall$ client i: compute the inverse playback curve Φ_i for this epoch
5: $\forall$ client i: **for** $1 \leq k \leq N_{ep}^{in}$ **do** {*for all intervals in epoch*}
6: **while** $j < kN_{in}^{sl}$ **do** {*for all slots in interval*}
7: $r_{ij} \leftarrow I_i A^k R$; $j \leftarrow j + 1$
8: **function** greedy algorithm: **while** $AvailableSlots \neq 0$ **do**
9: select a client with the lowest id i s.t. $(\forall q \leq n, lead_i \leq lead_q)$
10: select a slot j s.t. $(j \in AvailableSlots)$ and $(\forall x \in AvailableSlots, r_{ij} \geq r_{ix})$
11: allocate slot j to client i; $rcvbits_i \leftarrow rcvbits_i + r_{ij}$
12: $lead_i \leftarrow$ initial lead of video $i + \frac{\Phi_i(rcvbits_i)}{F}$
13: remove j from $AvailableSlots$

Complexity analysis. The total number of slots considering all epochs and intervals is given by N_{ep}^{sl}. We now evaluate the runtime of the greedy algorithm in Algorithm 2.

Time Complexity: Initialization

Lines 5 -7 : $O(\max(nN_{ep}^{sl}, nN_{in}^{sl}K^2))$. This is because the matrix multiplication $(I_i A^k R)$ will require $O(K^2)$ time.

Time Complexity: Greedy algorithm

Line 9: $O(n)$

Line 10 $O(N_{ep}^{sl})$

Lines 11-13 $O(1)$ (assuming constant computation time for $\Phi(.)$)

Lines 9-13 $O(N_{ep}^{sl}) + O(n) = O(\max(N_{ep}^{sl}, n)) = O(N_{ep}^{sl})$ (as $N_{ep}^{sl} > n$ usually)

Lines 8-13 are executed (N_{ep}^{sl}) times and thus the greedy algorithm takes $O({N_{ep}^{sl}}^2)$. Total Time Complexity : $O(\max({N_{ep}^{sl}}^2, nN_{in}^{sl}K^2))$. We provide further details in Appendix B.2.

To motivate our choice of the above greedy algorithm, we now show that the algorithm is optimal for LMVT when each client's channel condition does not change within an epoch (but different clients may have different rates).

Lemma 8. *If the rate of each client does not change within an epoch, the greedy algorithm yields an optimal solution for LMVT.*

The sketch of the proof is as follows. As the rate of a client i does not change within an epoch, each slot allocated to the client i provides a constant number of bits, say r_i. The greedy algorithm simply chooses the client i that has the lowest id among the clients with the minimum lead, and selects the next available slot and allocates it to i. The proof of optimality is by induction on the number of allocated slots and is shown in Appendix B.3.

As a special case of the above lemma, when the transmission channel is of Constant Bit Rate (CBR), i.e., the rate of slots do not change within an epoch or across the users, e.g., in a wired link, the greedy algorithm is optimal.

Corollary 1. *For a CBR channel, the greedy algorithm yields an optimal solution for LMVT.*

5.7 Experimental Setup

5.7.1 Scheduling Algorithm: Parameters

To evaluate our epoch-by-epoch scheduling strategy based on playout lead we need to specify the epoch duration, interval size and the number of slots in an interval. Recall that in our scheduling strategy, epochs are divided into intervals, which are subdivided into slots (Figure 5.2(b)).

For ensuring a smooth viewing experience, it is undesirable to have small or large epochs as the former will result in frequent glitches while the latter will significantly delay playout. Hence in our experiments we consider epochs to be in the seconds timescale. We perform our experiments considering an epoch duration of 10 seconds (except Figure 5.6 where we vary the epoch duration). We choose an interval duration to be 1 second in our experiments because we want to capture channel variation due to path loss and shadowing effects. The fast fading behavior of the channel will average out for video frames (as their transmission time is typically large with respect to the fast fading timescale). In our experiments, we vary the number of slots in an interval. By varying the number of slots in an interval we can vary the total resource (in terms of bandwidth) available at the base station because the rates in our Markov model correspond to the number of bits received in a slot.

The main objective of our experiments is to demonstrate that the proposed greedy algorithm is able to achieve its goal of minimizing the number of stalls across a broad range of epoch durations, interval sizes and number of slots per interval. Determining the optimal epoch duration, the interval size or the number of slots in an interval so as to maximize viewer satisfaction is beyond the scope of this work.

We assume the following buffering scheme at the client - if the client does not have enough data to playout for the whole duration of the epoch, it stalls for the entire epoch. We also assume that the clients have infinite large buffers to store all received packets.

5.7.2 Trace-Driven Experiments

To demonstrate the efficacy of the greedy algorithm, we perform trace-driven experiments. Our evaluation uses two types of traces:
(i) VBR Video Traces that provide the variation in the frame sizes of videos for emulating video playouts.

(ii) User-Level Wireless Channel Traces that provide the rates achieved by different users in every interval of each epoch.

5.7.2.1 VBR Video Traces

We use the publicly available MPEG-4 *VBR Video Traces* [8, 79] in our experiments. The videos play out at a constant frame rate of 30 frames per second. We perform experiments with video traces encoded in Common Intermediate Format (CIF) and Quarter CIF (QCIF). All evaluation is performed in a scenario where 8 different videos are being simultaneously streamed to 8 different users over the shared wireless infrastructure. Unless mentioned otherwise, all results are reported for CIF videos. A brief description of the 8 CIF video traces used, is given in Table 5.2. The duration of the videos used in our experiments is approximately 27 minutes. Detailed information about the CIF and QCIF traces is available in [79].

5.7.2.2 User-Level Wireless Channel Traces

Signal Strength Measurement. The wireless channel traces we use were obtained from signal strength measurements over a (802.16e) WiMAX network deployed in WINLAB at Rutgers University. The WiMAX base station is installed in WINLAB. During our trace collection, the base station continuously transmitted data packets, and signal strength (RSSI) was recorded at the receiver (a laptop) under vehicular and pedestrian mobility. As our interval duration is 1 second, we obtain signal strength quality one second apart from each another. To eliminate any fast fading effects, we consider the average signal strength at the beginning of each second. A brief description of the parameters of the WiMAX network used in our trace collection is given in Table 5.3. The vehicular mobility traces were collected by driving a car around the campus multiple times while the pedestrian mobility experiments were performed by walking around the same campus. We conducted 4 vehicular and 4 pedestrian mobility experiments, each of duration approximately 10 minutes. As

the base station only has a range of 500m, the entire range was effectively covered by these experiments.

RSSI-Rate Mapping. To obtain a mapping between the RSSI values and the rates achieved, we use the mapping between the modulation and coding schemes (MCS) and the SINR values for a WiMAX network provided in [6]. A common approach is to divide the SINR regime into a number of ranges and for each range there exists an MCS that maximizes throughput. The MCS indicate the rates achievable in practice. Six different rates are achievable in practice and they have the following ratio $[1, 1.5, 2, 3, 4, 4.5]$ [6]. As mentioned earlier, our base station reports RSSI values, which is similar to the SINR values reported in [6]. The minimum and maximum values of RSSI measured in our experiments are -85 dBm and -37 dBm and we map them to the corresponding SINR values in [6]. We use linear extrapolation to determine the mapping between RSSI ranges and the rates achieved. We use the RSSI-rate mapping to generate the *rate traces* (i.e., traces indicating the rates achieved over time) for the vehicular and pedestrian mobility experiments. We then generate 8 different *User-Level Wireless Channel Traces* (each 27 minutes long) emulating the real channel conditions (separately for vehicular and pedestrian mobility) from the *rate traces*.

Markov Chain Model. Our Markov channel model has 6 different states corresponding to the rates achieved. The states of our Markov model correspond to the number of bits successfully transmitted in a slot. The vector of transmission rates is taken to be $R = [1, 1.5, 2, 3, 4, 4.5] * 50000$ bits for the CIF videos. SNR based Markov chain models describing the wireless channel have been well studied in literature. [27] provides a detailed description of the various models available in literature. Similarly the use of SNR to bit rate mapping is also common [28], [29]. We determine the transition matrix of the Markov chain empirically (from the *rate traces*) by counting the number of transitions from one state (say i) to other states and then normalizing them by the total number of transitions from state i.

Name of video	Mean bit rate (Mbps)	Mean frame size (Kb)	Standard deviation (Kb)
Star Wars IV	0.42	14	17.6
Lord of the Rings I	0.65	21.6	22.7
Tokyo Olypmics	1.06	35.4	39.4
Matrix I	0.41	13.4	17.1
Matrix II	0.61	20.2	25.5
Matrix III	0.52	17.1	20.5
NBC News	1.33	44	34
Silence of the Lambs	0.44	14.7	22.2

Table 5.2. CIF video trace statistics

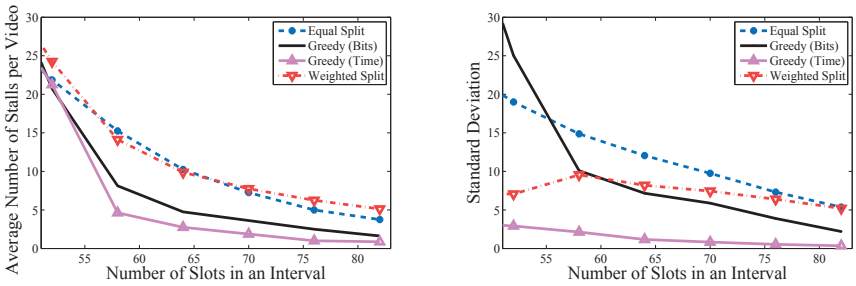
Parameter	Value
PHY	OFDMA
Carrier Frequency	2.59 GHz
Channel Bandwidth	10 MHz
Frame duration	5 ms
Transmission power	30 dbm
Antenna model	Sector
Fragmentation/Packing	ON
ARQ	OFF

Table 5.3. WiMAX system parameters

We note here that after about 40 steps (i.e., 40 seconds), the probability distribution obtained from any starting state using the transition matrix reaches very close (5%) to the steady state distribution for both vehicular and pedestrian mobility scenarios. Therefore, the transition matrix does not reach steady state during the duration of an epoch (which is 10 seconds) and is thus useful as a prediction mechanism for making scheduling decisions.

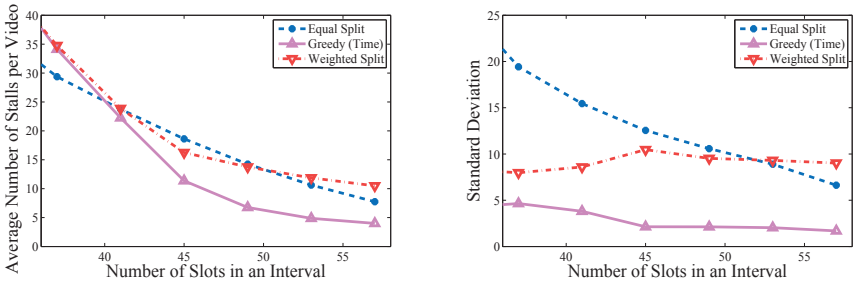
5.8 Results

In this section, we present and discuss results for the various experiments conducted. We compare the performance of the greedy algorithm against two baseline approaches: the equal-split and the weighted-split algorithms. In the equal-split approach, we divide the number of slots available in every interval equally among all



(a) Average number of stalls (b) Fairness

Figure 5.3. Vehicular mobility: distribution of stalls with variation of wireless channel resource (slots) for CIF videos

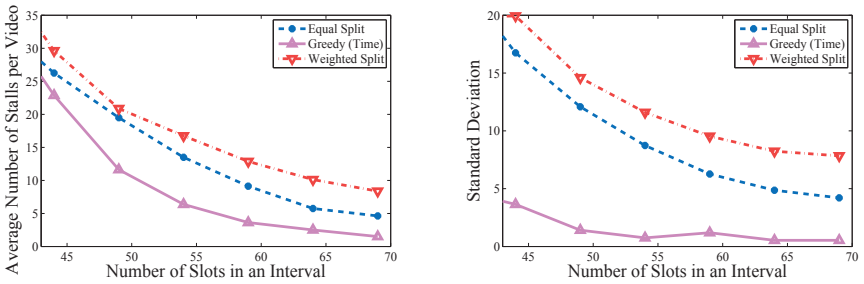


(a) Average number of stalls (b) Fairness

Figure 5.4. Pedestrian mobility:distribution of stalls with variation of wireless channel resource (slots) for CIF videos

the users. In the weighted-split the total number of slots in any interval is divided in proportion to the mean bit rate of the individual video streams. While allocating the slots, these two algorithms neither consider the playout lead nor the wireless channel variability, and hence, we expect them to be unfair, and have lower overall performance compared to our greedy strategy.

To emphasize the importance of making scheduling decisions based on playout lead, we also consider a variant of our greedy algorithm from Section 5.6 (we denote our algorithm from Section 5.6 by greedy-time). We consider a greedy-bit algorithm which is similar to our greedy-time algorithm except for one crucial aspect: it allocates the next slot to the video with the minimum lead in terms of playout bits (buffer size) instead of playout time. To avoid cluttering the plots with many lines, we show



(a) Average number of stalls (b) Fairness

Figure 5.5. Mixture of vehicular and pedestrian mobility: distribution of stalls with variation of wireless channel resource (slots) for CIF videos

only a few results for the greedy-bit algorithm. The greedy-bit approach ignores the variability in the frame sizes (i.e., burstiness) of a video with the result that it allocates fewer resources to a video experiencing a burst, thereby unfairly making it stall for longer durations.

5.8.1 Distribution of Stalls

In this subsection we study stall distribution as a function of the number of slots in an interval (keeping the interval duration constant). Using the steady state probabilities of the Markov model, one can compute the expected number of bits received per slot.

5.8.1.1 Vehicular Mobility

Figure 5.3 shows the variation of the average number of stalls for four scheduling algorithms: equal-split, weighted-split, greedy-bit and greedy-time. Table 5.4 provides the expected bit rate in the steady state for different values of the number of slots per interval. In our experiment, the mean bit rate of the 8 CIF videos is approximately 5.4 Mbps. Thus, from Table 5.4, we note that 34, 58 and 82 slots per interval correspond to the wireless channel being under-provisioned, average-provisioned and over-provisioned, respectively for the vehicular mobility scenario.

In terms of the average number of stalls per video, both the greedy algorithms perform better than the equal-split and the weighted-split approaches for the average and over-provisioned scenarios. With respect to fairness, the standard deviation of the number of stalls shows that in terms of evenly distributing the stalls among the videos, our greedy-time algorithm performs significantly better than other algorithms. We observe that the greedy-bit algorithm is unfair in distributing stalls (Figure 5.3), and so we will not consider this algorithm further.

To highlight the performance of the greedy-time algorithm, we present results for the average number of stalls experienced for the mildly over-provisioned case (64 and 70 slots) in Table 5.5. The mildly over-provisioned case is the scenario of interest in practice and we observe that the greedy-time algorithm reduces the number of stalls by a factor of 3 to 4 when compared to equal-split and weighted-split. Overall, we observe that the greedy-time multiplexing algorithm gives the best performance both in terms of reducing the average number of stalls per video and evenly distributing the stalls among the videos.

5.8.1.2 Pedestrian Mobility

We also conducted experiments under pedestrian mobility and the results are shown in Figure 5.4. We observe that the greedy-time algorithm again outperforms the equal and weighted split algorithms in terms of both average number of stalls and fairness.

5.8.1.3 Mix of Vehicular and Pedestrian Mobility

In practical situations, we will usually have a mix of pedestrian and vehicular users, streaming different videos from the base station. Figure 5.5 shows the simulation results considering 4 vehicular and 4 pedestrian users. We observe that the greedy-time algorithm outperforms the other two schemes. Interestingly, in Figure 5.5(b), the weighted split algorithm has higher standard deviation when compared to the vehicular (Figures 5.3(b) and 5.4(b)). This is because unlike the vehicular

Table 5.4. Expected steady state bit rate for a given number of slots

Number of Slots	Expected Bit Rate (Mbps)
34	3.23
58	5.7
82	8.0

Table 5.5. Average number of stalls per video for an average-provisioned network

Scheme	Number of Stalls (Slots 64)	Number of Stalls (Slots 70)
Equal Split	10.25	7.25
Weighted Split	9.875	7.75
Greedy-time	2.75	1.875

and pedestrian mobility cases, where all users have similar channel quality, in Figure 5.5(b) we have both pedestrian and vehicular users and the weighted split approach (which divides the number of available slots proportional to the mean bit rate of the videos without taking the channel conditions into account) results in unfair distribution of stalls. In contrast to this, the greedy-time heuristic continues to distribute the stalls fairly. We note that similar to Figure 5.3(b), the greedy-bit algorithm is unfair in distributing the stalls for the experiments conducted in sections 5.8.1.2 and 5.8.1.3 as well. In the remaining sections we only present the results for the vehicular mobility case.

5.8.2 Sensitivity to Epoch Duration

In the experiments presented thus far, the epoch duration was fixed at 10 seconds. In Figure 5.6, we present the variation in the average number of stalls per video as a function of the epoch duration. The number of slots in an interval is 64. We observe that the average number of stalls for the greedy-time algorithm decreases slightly as the epoch duration increases. As the epoch duration increases, the number of stalls for the other schemes decreases faster in comparison to the greedy scheme. This is because as the greedy scheme starts with a significantly lower number of stalls, increasing epoch duration does not benefit it much. We note, however, the total stall duration averaged over all videos increases with increasing epoch duration.

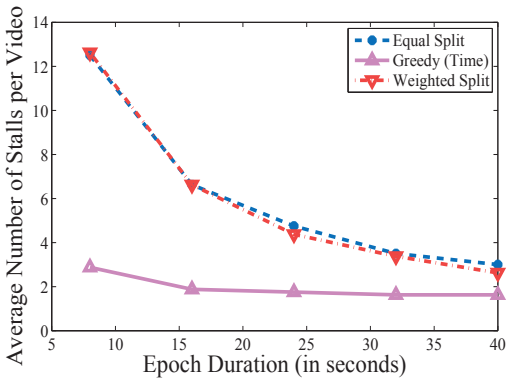
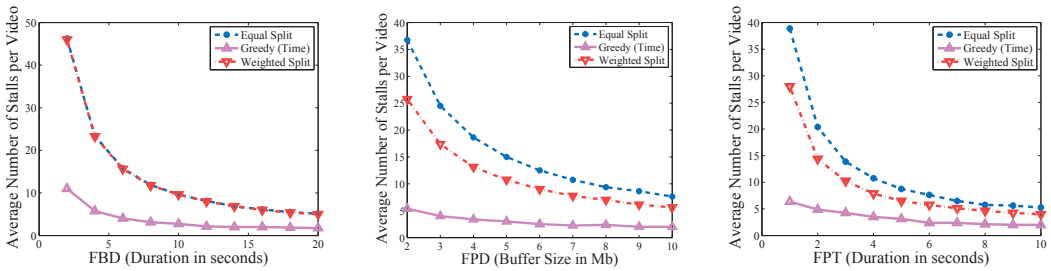


Figure 5.6. Sensitivity to epoch duration



(a) Fixed buffering delay strategy(FBD)

(b) Fixed buffered playout data (FPD)

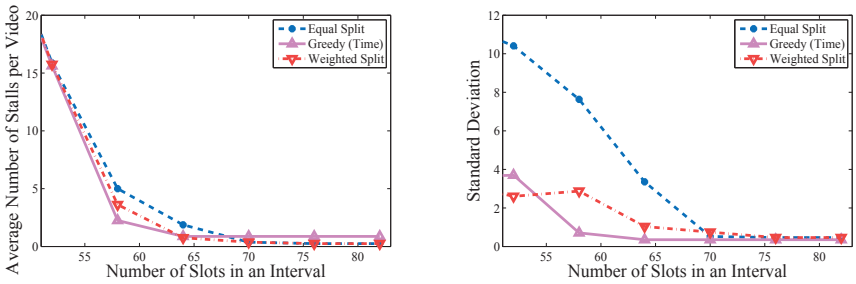
(c) Fixed buffered playout time (FPT)

Figure 5.7. Sensitivity to different buffering schemes

5.8.3 Sensitivity to Buffering schemes

Recall that in the results presented above, we have assumed a client stall-recovery buffering scheme in which the client stalls for the entire epoch when there is not enough buffered data available for playout for the whole epoch. However, the media players at the clients may have a different buffering scheme. Following [52], we now consider the three common buffering schemes:

- Fixed Buffering Delay (FBD): Once a stall occurs, resume playout only after a fixed duration of time.
- Fixed Buffered Playout Data (FPD): Once a stall occurs, resume playout only after a fixed amount of data is received.



(a) QCIF set 1: average stalls (b) QCIF set 1: fairness

Figure 5.8. Vehicular mobility: distribution of stalls with variation of wireless channel resource (slots) for QCIF videos

- Fixed Buffered Playout Time (FPT): Once a stall occurs, resume playout only after the receiver has accumulated enough data corresponding to a fixed playout duration.

We performed experiments to determine whether our algorithm's performance is sensitive to different client buffering schemes. Figures 5.7(a), 5.7(b), and 5.7(c) show the variation of the average number of stalls for the FBD, FPD and FPT buffering schemes, respectively. In these simulations we again considered 64 slots in each interval. In terms of playout stalls, the greedy-time algorithm still outperforms the other schemes irrespective of the buffering scheme adopted by the player at the client. We also observed that the greedy-time algorithm performs better in terms of evenly distributing the stalls across the videos.

5.8.4 Sensitivity to Different Video Traces

We also conducted experiments with two sets of 8 QCIF video traces, available from [8, 79]. We show results for one set of QCIF videos here. The results, plotting the average number of stalls and the standard deviation of stalls versus the number of slots in an interval, are shown in Figure 5.8. Given the low mean bit rate requirement of the QCIF videos, all the rates in the Markov channel model, i.e., the number of bits received in a slot, were scaled down by 10. This scaling down is done to investigate

algorithm performance near the average provisioned and mildly over provisioned cases, which are the scenarios that are interesting in practice. For QCIF videos, we observe that the greedy-time algorithm outperforms the other approaches in terms of fairness, but its performance is similar to the weighted split algorithm in terms of average number of stalls. Note that when the number of slots in an interval is larger than 65 (this corresponds to the highly overprovisioned case), the other algorithms slightly outperform the greedy algorithm. The total number of stalls experienced by any video in this case is only 0 or 1; the difference between the algorithms is that some videos experience a stall in case of the greedy algorithm while no stalls occur for the other algorithms.

5.8.5 Sensitivity to Poor Channel Condition

A potential drawback of maximizing the minimum playout lead is the case where some clients have poor channel condition for a protracted period of time. Maximizing the minimum playout lead in this situation can degrade entire system performance. One way to tackle this issue is to restrict the maximum number of slots that can be allocated to any user.

For simulations we consider a vehicular mobility scenario where two out of eight clients have poor channel quality: these clients transition only between the lowest two rates of the Markov model with probability 0.5. Since we do not have real world traces mimicking this kind of channel behavior we create synthetic traces for these two users. For generating the synthetic traces we assume that in any interval, each of two users can be in one of the two lowest rates with probability 0.5. Figure 5.9 shows the result for this simulation. The plot x-Thd in the figure signifies our greedy-time algorithm with the modification that the maximum number of slots allowed for any client is $x\frac{TotalSlots}{n}$, where n is the number of videos.

We observe that if there is no restriction on the maximum number of slots allocated for a client (i.e., our original greedy-time algorithm), the algorithm performs worse

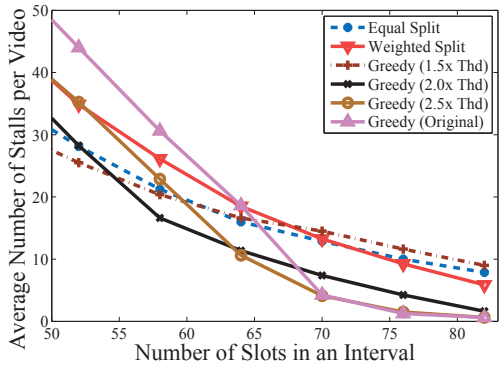


Figure 5.9. Effect of poor channel quality

than the baseline approaches with respect to the average number of stalls when the number of slots is small and has superior performance for the overprovisioned case. We can observe from Figure 5.9 that clearly there is a tradeoff in the performance of the greedy-time algorithm between the number of slots in an interval and the threshold imposed. The greedy-time algorithm with a low threshold performs best when the number of slots is small, while the opposite is true when the number of slots is large. As expected, as the threshold is increased, the performance of the greedy-time algorithm tends to the original algorithm with no threshold. In terms of standard deviation, as expected the Greedy (Original) algorithm performs best with the standard deviation increasing as we impose a lower threshold. Overall we observe that the $2.0x$-Thd greedy algorithm performs the best for the scenario chosen in this experiment.

5.9 Discussion

In this section, we discuss issues related to the adaptability and scalability of the greedy algorithm. In this chapter, we have only considered video streaming applications, but our algorithm can also be adapted for the case when there is other concurrent traffic through the base station. The other applications will consume a fraction of the base station resources (time slots in this case); the QoE requirement

of these applications being different from video streaming, our greedy algorithm can execute on the remaining timeslots (after timeslots required by other applications have been allocated). Since our approach operates using the slots available to video, it would find application in any scheme (even dynamic) in which a provider coarsely partitioned slots among applications.

Our greedy algorithm can also be adapted to work with two dimensional (time-slot and sub-carrier) allocation of data - as such the model can be enriched by having a separate Markov chain for the wireless channel on each subcarrier. The expected rate received in the various time slots for the different subcarriers can be determined using the Markov chains. Note that our greedy algorithm assumes that channel quality remains unchanged within an interval. So long this assumption holds, the greedy algorithm can be applied (it does not matter whether slots within an interval are divided in time domain, frequency domain or both).

We have also not considered the scenario where users can join/depart in the middle of an epoch. Our algorithm can easily be adapted to this situation. Users departing from the system will cause resources (slots) allocated to them for that epoch to be unused. This issue can be dealt with by randomly allocating the freed slots in the epoch among the different clients. If a new user joins in the middle of an epoch, this user will not have data sent to it during that epoch because all slots have already been allocated to other users a priori. This will cause an additional delay (with maximum duration of one epoch) to the new user. However in the beginning of the next epoch, this user will be given preference by the greedy algorithm (and thereby more slots allocated to it) as it will have playout lead equal to zero.

In this chapter, we addressed the problem of streaming stored video to various clients. The stored video might be considered as videos cached at devices at the edge of the telecommunication network. The video playback curve is just the set of frame sizes. This information regarding frame sizes can be made easily available at the base station. For example, Netflix manifest files already contain this information on a

per-chunk basis, where a chunk is approximately 4 seconds of data [4]. As the frame rate is only 30 frames/sec, the amount of information that is to be stored per video is not quite small (in the order of a few Mbits). Hence the memory required for storing video playout information is small. The runtime of the algorithm $O(N_{ep}^{sl\,2})$ and thus the greedy algorithm is easily scalable. Nowadays computational power is available at the base station [19] and thus base stations should be able to periodically execute the low complexity greedy algorithm.

Another issue might be the communication overhead for the greedy algorithm. Though overhead is not explicitly modeled here, the information required to be communicated by each client at the beginning of an epoch is only the playout lead and the current channel state (which is only a few bytes of information per client).

5.10 Conclusion

In this chapter, we investigated scheduling schemes for transmitting multiple video streams from a base station to mobile clients. We showed that the problem of allocating slots fairly is NP-complete even for a constant number of videos. We then presented a greedy algorithm based on a criterion of maximizing the minimum playout lead to manage stalls for multiple video streams transmitted over a time-varying bandwidth-constrained wireless channel. We demonstrated that the greedy algorithm is fair and is also capable of minimizing the average number of playout stalls.

CHAPTER 6

A MARKOVIAN MODEL FOR COARSE TIMESCALE CHANNEL VARIATION

6.1 Introduction

A large number of finite-state Markov chain models have been proposed to study the wireless channel quality and the received signal strength, beginning with the early Gilbert and Elliot two-state Markov channel [31, 36]. Variation in received signal strength over a wireless channel is caused by three main factors: multipath fading, path loss and shadowing. Among these three effects, fading is caused by constructive or destructive effects of multipath waves and changes in the order of milliseconds depending on the speed of the receiver and the frequency of transmission. Conversely, shadowing and path loss cause fluctuations in the signal level in the order of seconds and tens of seconds respectively. Path loss is the deterministic distance-dependent component of the received power. Superimposed on path loss is shadowing - a random process that captures variations in the received signal caused by changes in the environment (buildings, foliage and motion in the surroundings). Informally, shadowing is the variation in signal strength at a coarse timescale (few seconds) that is independent of the distance between the transmitter and receiver.

We focus on shadowing in this chapter and develop and validate a Markov chain to model the effects of shadowing on the received signal strength, that occur on the order of seconds. This shadowing model can be used in analyzing performance of wireless network protocols (e.g., for route adaptation, or for video transmission) that adapt their behavior in response to link-level changes at the timescale of seconds. We discuss applications of coarse-timescale channel modeling in detail in Section 6.2. The un-

derlying physical channel model assumes that the variation in received signal strength due to shadowing is a lognormally-distributed random variable with zero mean [70] and has an exponential autocorrelation function [38]. An exponential autocorrelation function in turn implies that shadowing follows a First Order Autoregressive AR(1) process [95]. The AR(1) process is a Markov process [95] because the current value of the process at time t depends only the value at $t-1$. These assumptions together enable the construction of a Markov chain model that captures the impact of shadowing on received power. We divide the entire range of shadowing into a finite number of intervals with each interval corresponding to a state in the Markov chain. We then determine the transition matrix of the Markov chain, investigating two methods for determining this transition matrix:

- *Model-based transition matrix.* In this method we derive mathematical expressions for the transition probabilities of the Markov chain using the properties of shadowing (log-normal distribution and exponential autocorrelation). This approach is parsimonious in nature as the transition probabilities depend only on the variance (σ^2) and the exponent (ρ) of the exponential autocorrelation function of shadowing. We refer to the transition matrix derived using this approach as the analytical one.

- *Empirical transition matrix.* The transition matrix can also be determined by conducting real world experiments, collecting received signal strength measurements, extracting the shadowing values and then determining the transitions from one state to the other. We refer to the transition matrix derived using this approach as the empirical one.

We test the assumptions and the performance of our model using signal strength measurements collected over an 802.16e (WiMAX) network and multi-hop wireless mesh network (TFA network at Rice University). We use hypothesis testing to assess the stationarity (via Augmented Dickey-Fuller (ADF) Test, Philip-Pheron (PP) Test)

of the signal strength measurements and the log-normal assumption (via Kolmogorov-Smirov (KS) goodness of fit test) of shadowing.

We observe that the signal strength measurement traces collected over both WiMAX and TFA network are stationary. We find that that the log-normal assumption of shadowing holds true for the WiMAX experiments. Interestingly, we observe that though the shadowing samples obtained from the TFA network appear to follow a normal distribution visually, they fail the KS test. The main reason for the traces collected over the TFA network to fail the KS test is the large number of samples collected; this results in the critical value for the KS test to reject the null hypothesis to be small. Our experiments show that the exponential autocorrelation assumption is not validated for the WIMAX network traces, while it approximately holds true for the TFA network traces.

We then determine the values of the analytical and empirical transition matrices using the measurements collected. Finally we compare the results (steady state occupancies and the transient behavior of the Markov chain) obtained by the two approaches with the observed shadowing-state distributions and find that they are quite close to one another even though some of the assumptions are not corroborated by empirical measurements.

The rest of this chapter is organized as follows. In Section 6.2, we discuss related work. We describe our Markov chain model in Section 6.3 and describe two approaches for deriving its transition matrix in Section 6.4. The test the validity of the assumptions of the model in Section 6.5 while a comparison of the experimental and analytical results are presented in Section 6.6. We finally conclude the chapter in Section 6.7.

6.2 Related Work and Applications

There is a great deal of research on developing Markov chain models for wireless channels, with the earliest work in this area being the simple, two-state model pro-

posed by Gilbert and Elliot [31, 36]. We discuss several of these previous works here, focusing on those that are most closely related to our own work and highlighting our contributions. In [94] a range of signal-to-noise ratio (SNR)values represents a state in the Markov chain. Based on this assumption the authors provide analytical expressions for the state transition probabilities and error probabilities in each state. In [108] the authors investigate the accuracy of a first-order Markov model for the success/failure of data blocks. A detailed survey of various channel models along with a description of their evolution over time is available in [73]. Our work differs from these existing Markov chain models for wireless channels in the sense that we concentrate on modeling channel variations at a much coarser time granularity, typically in the order of a few seconds and use shadowing to construct our model. We also validate our assumptions and results obtained from the model using data collected via real world experiments in a variety of different settings.

We next survey literature specifically focused on characterizing the properties of shadowing. A thorough description of the different random processes causing variation in the received signal strength over the wireless channel is available in [70, 90]. The log-normal nature of shadowing has been reported in [70, 101] and other prior work. [38] is the seminal paper modeling shadowing autocorrelation as an exponential function. Recent research has proposed refined versions of the autocorrelation depending on the environment. In [101] the authors propose a new autocorrelation model for shadowing in urban environments based on data collected in a Chinese city. The correlation properties of shadowing for an indoor channel have been studied in [47, 81]. The authors in [47] observed that shadowing is very environment specific and that correlation can be found in well-separated links if their environment is similar. Oesteges et. al perform an empirical characterization of the received power over a wireless channel in [62] for the outdoor-to-outdoor and indoor-to-indoor environment. They introduce several new aspects specific to multi-user distributed channels and also suggest that shadowing be divided into two components: a static

and a dynamic one. Most previous work on shadowing has focused primarily on the underlying process and on studying and characterizing the different properties (distribution, autocorrelation, cross-correlation) of shadowing itself. Only few prior work leverage these properties and use it for modeling or prediction purposes. In [48] the authors exploit the exponential autocorrelation assumption of shadowing to model it as a linear system and then design a Kalman filter to predict the variation of shadowing. Our work is unique in that we construct a Markov chain model assuming the log-normal distribution and exponential autocorrelation of shadowing and then validate our model with real data collected over different types of wireless networks.

Before describing our Markov chain model in detail we first discuss several applications where coarse-timescale channel prediction is potentially valuable; this will help motivate the application of the results of this work. The first application is the scheduling of multiple video streams over a LTE/WiMAX network with the objective of minimizing the number of playout jitters. Let us assume a simple time slotted scheme in which a video stalls if there is not enough data to play out in a timeslot. Such a model would require channel estimation from one timeslot to the other. Further to facilitate a smooth viewing experience the timeslots should be in the order of seconds instead of milliseconds to avoid experiencing large number of small glitches.

Bulk transfer of data in energy constrained mobile sensor nodes would be facilitated by coarse timescale prediction as it would provide ample time to the nodes to boot up from sleep when the channel is good and then transmit their data. Disconnection prediction and topology management in mobile ad-hoc networks would also be aided by channel quality prediction at a coarse time granularity. Rate control on a block of data is gaining popularity and a successful implementation of a block based scheme would require a coarse timescale channel model to predict channel variations from one block to the next (a block can take 1-2 seconds to be transmitted) coupled with a fine grained tracking of signal strength fluctuations within a block.

6.3 A Shadowing-based Channel Model

In this section, we describe the Markov chain model for shadowing and discuss its applicability in mobile wireless systems. Previous theoretical and practical studies indicate that the average received power varies logarithmically with the distance between the transmitter and receiver; this is the deterministic path loss component of the received power. Superimposed on the path loss is log-normally distributed random shadowing, which takes into account the fact that the received signal strength at the same transmitter-to-receiver separation can vary due to changes in the environmental surroundings.

Let d, α, d_0 be the transmitter-to-receiver separation, the path loss coefficient and the close-in reference distance respectively. The received power $P_r(d)$ in [dBm] considering log-normal shadowing [70] is given by

$$P_r(d)[dBm] = \bar{P}_r(d_0) - 10\alpha log\frac{d}{d_0} + X \tag{6.1}$$

where $\bar{P}_r(d_0)$ is the average received power at the reference distance d_0, the second term reflects the logarithmic dependence of received power on distance, and X is the shadowing - a zero-mean Gaussian random variable with variance σ^2 in [dB]. Therefore, Eqn. (6.1) demonstrates the effect of shadowing on received power.

Shadowing (in dB) [70] is assumed to be $N(0, \sigma^2)$ while both its spatial and temporal autocorrelation functions are assumed to be exponential [38, 98, 101]. Let X_i and X_{i+n} be the shadowing samples at time i and $i + n$ respectively. There are n samples between i and $i + n$ and let the time difference between two consecutive samples be δt. The temporal autocorrelation between X_i and X_{i+n} is given by,

$$\rho_n = \frac{E[X_i X_{i+n}]}{\sigma^2} = e^{-\frac{n\delta t}{\tau}} \tag{6.2}$$

If the autocorrelation between two successive samples is denoted by $\rho = e^{-\frac{\delta t}{\tau}}$, we have $\rho_n = \rho^n$. We denote ρ as the autocorrelation coefficient.

An exponential autocorrelation function implies that the random process is a first-order autoregressive AR(1) process [95]. Therefore the shadowing samples form an AR(1) process [98], and we can write the following equation

$$X_i = \rho X_{i-1} + (1 - \rho)e_i \tag{6.3}$$

where e_i is white noise and is $\sim N(0, \sigma_e^2)$. Furthermore e_i and X_{i-1} are independent of each other. X_i being an AR(1) process also implies that shadowing is a Markovian process [95]. This is evident from Eqn. (6.3) as well, since X_i depends only on X_{i-1}.

Our Markov chain model for shadowing is constructed as follows. The entire range of shadowing is partitioned into a finite number of intervals (N), where each state of the Markov chain corresponds to one such interval. Let us assume that the shadowing range is divided in the following way; $(A_0, A_1....A_N)$ where A_0 and A_N correspond to $-\infty$ and ∞ respectively, as shadowing is Gaussian distributed. Let Y_i denote that the X value is between A_{i-1} and A_i. Therefore, the set $\{Y_i\}$ denotes the states of the Markov chain. The goal is to derive the state transition matrix of the Markov chain, i.e., the transition probabilities P_{ij} from range Y_i to range Y_j, $\forall i, j \in N$. We describe approaches for numerically computing the transition matrix in Section 6.4.

In this section, we constructed a Markov chain model that captures the effects of shadowing on the received power. We note that the overall variation in received power can only be captured by modeling both the variation in the distance and in shadowing, as evident in Eqn. (6.1). However, if we assume that the distance remains constant during the time interval of interest, changes in the signal strength can be represented by modeling the effects of shadowing alone. The distance/average signal strength may well change more slowly, and can be updated at the sender based on feedback from the receiver at a coarser timescale. This may be a valid assumption for most applications, especially for those with lower mobility operating over ad-hoc and cellular networks.

6.4 Determining the Transition Matrix

In this section, we describe the analytical and empirical approaches for determining the transition matrix of the Markov model for shadowing.

6.4.1 Analytical Approach

From the previous section, we know that shadowing (X) is normally distributed and that it is a Markov process (Eqn. (6.3)). We now determine the state transition probabilities ($P'_{ij}s$,) and begin by stating the following lemma.

Lemma 9. *Two consecutive shadowing samples are jointly Gaussian*

Proof. From Eqn.(6.3), e_i and X_{i-1} are independent and both are themselves Gaussian. Hence e_i and X_{i-1} are jointly Gaussian. From the Cramer-Wold Device it is known that X_i and X_{i-1} will be jointly Gaussian if any linear combination of them is Gaussian. Any linear combination of X_i and X_{i-1} can be represented as

$$Z = \alpha X_i + \beta X_{i-1} = (\alpha\rho + \beta)X_{i-1} + \alpha(1-\rho)e_i \tag{6.4}$$

e_i and X_{i-1} are jointly Gaussian which means that Z is Gaussian. Hence using the Cramer-Wold Device we have that X_i and X_{i-1} are jointly Gaussian. □

To calculate the transition probability P_{ij}, we must determine the probability of transitioning from range Y_i to range Y_j at any time step k. X_k and X_{k-1} being jointly Gaussian implies that $X_{k+1}|X_k \sim N(\rho x_k, \sigma^2(1-\rho^2))$. Moreover we have that $X_k \sim N(0, \sigma^2)$. Therefore, we have

$$\begin{aligned} P_{ij} &= P(X_{k+1} \in Y_j | X_k \in Y_i) \\ &= \frac{P(\{X_{k+1} \in Y_j\} \cap \{X_k \in Y_i\})}{P(\{X_k \in Y_i\})} \\ &= \frac{\int_{Y_i}(\int_{Y_j} f_{X_{k+1}|X_k}(x_2|x_1)dx_2) f_{X_k}(x_1)dx_1}{\int_{Y_i} f_{X_k}(x_1)dx_1} \end{aligned} \tag{6.5}$$

As the distributions of $X_{k+1}|X_k$ and X_k are Gaussian, $\int_{Y_j} f_{X_{k+1}|X_k}(x_2|x_1)dx_2$ and $\int_{Y_i} f_{X_k}(x_1)dx_1$ can easily be calculated using error functions. The absolute value of P_{ij} can then be numerically calculated, as the integral in the numerator can be easily solved using a mathematical package like MATLAB, once the values of ρ and σ have been determined.

6.4.2 Empirical Approach

The transition matrix can also be determined by performing signal strength measurements at the receiver for experiments conducted over any desired network. The first task is to extract the shadowing values by eliminating the deterministic distance dependent path loss. We then determine the states of the Markov chain to which each of the shadowing values correspond to. Therefore, we have the sequence of states through which the Markov chain has progressed. The subsequent step is to determine the number of transitions from each state to the others by observing the sequence of states. For example, suppose there are 6 states in all and that the sequence of states is $\{.....2, 4, 6, 2, 4.....\}$. The subsequence $\{2, 4\}$ means thats we increment the number of transitions from state 2 to state 4 by one. The next transitions are from states 4 to 6, 6 to 2 followed by another transition from 2 to 4. Once all the transitions have been considered, we use the relative values of the numbers of transitions from state i to state j for all states j to determine the empirical transition probabilities from state i to all states j, P_{ij}.

We determine the parameters (σ, ρ) needed for the analytically-determined transition matrix and the directly observed transition probabilities $P_{i,j}$ in the empirical transition matrix from experiments conducted over a WiMAX network and a large multi-hop wireless network (TFA network). From the received power measurements we first extract the shadowing values. The variance and autocorrelation coefficient of the shadowing values are then determined, which are used to obtain the transition

matrix analytically. The shadowing values are also used to obtain the empirical transition matrix by observing the transitions between successive shadowing samples.

6.5 Validating the Model

In the preceding sections, we developed a Markov chain model for channel prediction based on changes in the received signal strength due to shadowing. Our goal in this section and the next is to conduct real world experiments and extract the shadowing samples to *(i)* corroborate the underlying model assumptions - that shadowing follows a normal distribution and that the autocorrelation of shadowing has an exponential decay, *(ii)* determine the transition matrix of the Markov chain analytically and empirically, from the data collected, and *(iii)* compare the analytically and empirically obtained transition matrices, and the channel performance predictions made via the Markov chain models using these transition matrices to assess the ultimate usefulness of our model.

6.5.1 Experimental Setup

We test the validity of the model assumptions and the performance of our model with data collected over different types of networks (WiMAX and TFA network).

6.5.1.1 WiMAX Experiments

We collected data under varying levels of user mobility (pedestrian and vehicular) for experiments carried out over a 802.16e (WiMAX) network as described in Chapter 5. We obtained signal strength quality one second apart from each another by eliminating the fading effects (described in Chapter 5). While collecting the signal strength measurements, the distance variation from the outdoor base station was captured using a GPS device attached to the laptop. The GPS device provides latitude and longitude information, which was then converted to 2D-Cartesian coordinates. The height of the base station from the ground was also measured and the transmitter-

to-receiver distances were calculated from this information. The shadowing samples were extracted by observing the deviation of the received power samples from the log distance relation. In all, three vehicular and two pedestrian traces were collected, each having a duration of approximately 8 minutes.

6.5.1.2 TFA Network Experiments

We also evaluate the performance of our model using data collected over the TFA network [16, 34]. The TFA network is a large multi-hop wireless network deployed in Southeast Houston by Rice University. The nodes are equipped with 802.11b wireless cards. The TFA network is a multi-tier architecture consisting of the *access tier* and the *backhaul tier*. The clients connect to the access points (APs) in the *access tier* while the *backhaul tier* wirelessly interconnects the APs to forward client traffic to and from the wired gateways. The TFA network consists of 17 nodes which span a 3 sq. km area. The coordinates (latitude and longitude) of the different APs are also provided [34].

The researchers at Rice University conducted a large number of experiments and have made the traces publicly available through the website [34]. We use the data trace collected by them via the following experiment [34]. In this experiment, the authors employ wardriving (which is the act of searching WiFi networks in a moving vehicle using a portable computer), where a vehicle collects beacons from APs while passing the streets of the neighborhood. The vehicle follows arbitrary paths and this process is repeated 15 times on different days. Their data trace has the following information (latitude, longitude, signal strength and unix time) where latitude and longitude correspond to the GPS coordinates of the receiver in the vehicle. From this data trace we obtain signal strength measurements from every AP on a per second timescale. The distance between the receiver and the different APs every second can also be determined using the GPS measurements. We then subtract the path loss

from the signal strength measurements to extract the shadowing values. We obtain this information for each of the 17 APs that we found in this data trace.

As mentioned above, the receiver collects beacons transmitted by the various APs as the authors drive the vehicle in the neighborhood of the APs. The APs transmit beacons using WiFi (802.11b) and therefore the vehicle frequently moves out their limited transmission range. As a result of poor connectivity between the vehicle and any AP, the data collected for any AP is not a continuous time series and there are frequently missing data points.

6.5.2 Stationarity Testing

We tested the stationarity of the WiMAX and TFA network traces using hypothesis testing. The most popular methods used for wide sense stationarity testing are the ADF Test (Augmented Dickey-Fuller Test) [27, 74], the PP Test (Philips-Perron Test) [66] and the Variance Ratio Test.

The ADF Test and PP Test can be used to assess the null hypothesis of a unit root in a univariate time series against the alternate hypothesis that the data is from an AR process (i.e., wide sense stationary). Here by root, we mean the roots of the characteristic equation of the AR(p) process. The main difference between the ADF Test and the PP Test is that the PP Test is non-parametric, i.e., it corrects for any serial correlation and heteroskedasticity in the errors non-parametrically. In an AR(p) process, p is referred to as the 'lag'. The Variance Ratio Test assesses the null hypothesis that the data is from a random walk. For all the three tests if the null hypothesis is rejected, then one can conclude that the data is stationary.

The ADF Test and the PP Test are most suited for our study as they can be directly used to determine whether the data is stationary or not, while the Variance Ratio Test tests whether the data has a particular type of 'non-stationary' behavior. We used the 'adftest', 'pptest', 'vratiotest' functions available in the econometrics toolbox in MATLAB to perform these tests.

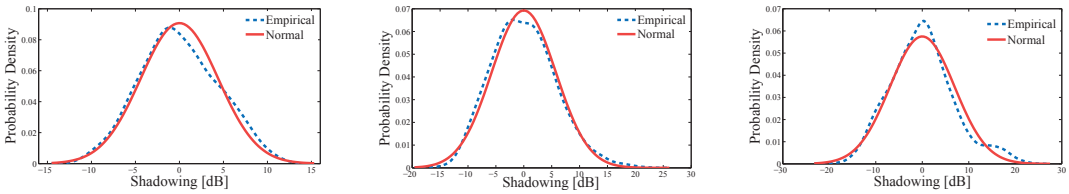
WiMAX Measurements. We have 3 vehicular and 2 pedestrian mobility traces, each consisting of approximately 500 samples. For the 'adftest' and the 'pptest' we tested with large range of lag values and observed that the null hypothesis is rejected for all values of lag for vehicular mobility traces at 5% and 1% levels of significance. For pedestrian mobility traces the null hypothesis is rejected for values of lag less than or equal to 10 and is accepted for lag values greater than 10 at 5% level of significance. But very large values of lag are not of concern to us, as we model the shadowing process as an AR(1) process. At 1% level of significance we observed that the null hypothesis is accepted at lag values greater than 4. The Variance Ratio Test rejects the null hypothesis that the data is from a random walk at 5% and 1% levels of significance. *The hypothesis testing results suggest that the WiMAX measurement traces (vehicular and pedestrian mobility) are stationary.*

TFA Network Measurements. We observed that for the ADF Test, all 17 APs reject the null hypothesis for lag values below 40 at both 5% and 1% levels of significance. For lag values greater than 40 and less than 100, all APs except 2 reject the null hypothesis at 5% level of significance. Similarly, for lag values greater than 40 and less than 100, all APs except 3 reject the null hypothesis at 1% level of significance. For the PP test we observed that for lag values less than 100, all APs reject the null hypothesis at 5% and 1% level of significance. For the Variance Ratio Test, we observed that all APs except 1 reject the null hypothesis of a random walk. *The hypothesis testing results suggest that the TFA Network measurement traces are stationary.*

6.5.3 Normality Testing

We use the *Kolmogorov-Smirnov* goodness of fit (KS test) to determine the normality of shadowing for the traces collected. Let σ^2_{sam} denote the variance of the collected samples. The null hypothesis is the following: The samples are drawn from a normal distribution having mean 0 and variance σ^2_{sam}.

WiMAX Measurements. Our tests failed to reject the null hypothesis at any acceptable level of significance for both the vehicular and pedestrian mobility traces. The smoothed probability distribution obtained for one of the vehicular traces using the kernel density estimation method and the corresponding normal distribution are shown in Figure 6.1(a). The standard deviation for this trace is 4.4 dB.



(a) WiMAX: vehicular mobility (b) TFA Network: Quince (c) TFA Network: Woodridge

Figure 6.1. Distribution of shadowing

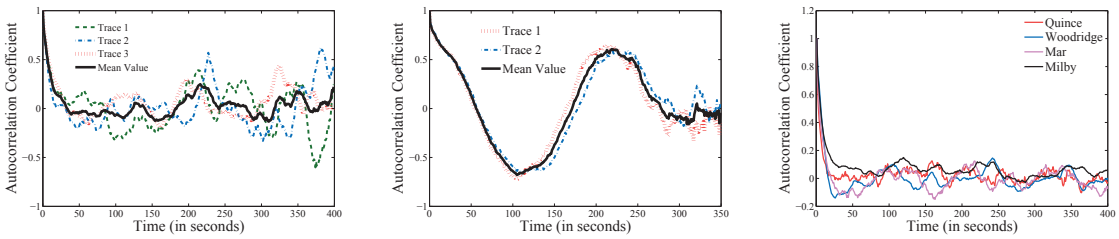
TFA Network Measurements. We observe that though shadowing for most of the APs looks Gaussian visually, they fail to pass the KS test at acceptable levels of significance. Let us consider Figures 6.1(b) and 6.1(c) to appreciate this observation. These figures show the distribution of shadowing for two different APs (Quince and Woodridge) along with the normal distribution obtained with mean 0 and variance σ^2_{sam}. To avoid cluttering the chapter with similar kinds of graphs in the remaining sections we will show results only for these two APs. We note that other APs also report similar results. While studying the transient behavior (Section 6.6.2.2) we found that Woodridge reported the highest total variation between the analytical and true occupancy (0.178) for the 2-step distribution.

The main reason for the APs to fail the KS test is the large number of samples collected; this results in the critical value for the KS test to reject the null hypothesis to be small. Moreover, in the experiments conducted, the authors drove their vehicle a large number of times in the same region and collected a lot of samples. Therefore, it is possible that there are more samples from a certain location as opposed to another, thereby biasing the data collected. Some areas around the access point may

not be accessible by road and hence would not be covered by the this data trace. Interestingly, a study done by researchers at Rice [16] reports results for a smaller dataset that the authors collect over this network. In this paper they claim that shadowing is Gaussian (but they do it visually and do not use the KS test).

6.5.4 Exponential Autocorrelation Testing

WiMAX Measurements. The temporal autocorrelation function of shadowing for the three different vehicular traces along with the mean of these three traces is shown in Figure 6.2(a). We observe that the autocorrelation function does *not* follow an exponential decay when the traces are considered individually. The data for pedestrian mobility in Figure 6.2(b) similarly shows that the autocorrelation function for these traces is not exponential. Moreover, while we observed that the average autocorrelation is roughly exponentially for the case of vehicular mobility, this is not the case for pedestrian mobility.



(a) Vehicular mobility (b) Pedestrian mobility (c) TFA network

Figure 6.2. Autocorrelation of shadowing

TFA Network Measurements. The measurements taken from the TFA network are not continuous time measurements, as mentioned earlier. We ignore the missing data points and consider the available data as a time series. Our goal here is to determine whether the autocorrelation decays sharply and is approximately exponential; for smaller values of time lag the autocorrelation is more accurate than larger values of time lag. We show the autocorrelation function for 4 APs in Figure

6.2(c). We observe visually that the autocorrelation is roughly exponential for the TFA Network measurements unlike the WiMAX measurements.

6.6 Results

In this section, we construct the Markov chain by dividing shadowing into the following intervals $\{-\infty, -\sigma_{sam}, -\frac{\sigma_{sam}}{2}, 0, \frac{\sigma_{sam}}{2}, \sigma_{sam}, \infty\}$. The size of the intervals is chosen in this manner so that there are sufficient data points in each interval. As shadowing follows a normal distribution, the probability of receiving shadow samples becomes very small as we move away from the mean and so we consider the interval beyond σ_{sam} or $-\sigma_{sam}$ as an open interval.

We then determine the transition matrix analytically and empirically for the WiMAX and TFA network traces using the approaches outlined in Section 6.4. Having determined the transition matrices, the ensuing step is to examine the closeness of the system state behavior (e.g., steady state and transient behavior), as calculated via one of the Markov chain models, and as observed empirically.

To have a better understanding of the closeness of the different distributions we quantify them in terms of the total variation [76]. The total variation between a probability distribution P and a probability distribution Q with n outcomes is given by,

$$Total\ Variation = \frac{1}{2}\sum_{i=1}^{n} |p_i - q_i| \tag{6.6}$$

We first discuss the results obtained for the WiMAX network and then the TFA network.

6.6.1 WiMAX Network

We compute the standard deviation and autocorrelation coefficient needed to determine the analytic transition matrix. Table 6.1 lists these values for the vehicular and pedestrian traces. As expected, the values for the vehicular trace are close to one another while the same is true for the pedestrian traces. Tables 6.2 and 6.3 show the

		Standard Deviation	Autocorrelation Coefficient
Vehicular	Trace 1	4.4	0.84
	Trace2	4.3	0.83
	Trace 3	4.6	0.86
Pedestrian	Trace 1	3.6	0.84
	Trace 2	3.6	0.87

Table 6.1. Standard deviation and autocorrelation coefficient

	State 1	State 2	State 3	State 4	State 5	State 6
State 1	0.6433	0.2334	0.0983	0.0211	0.0023	0.0001
State 2	0.2471	0.3290	0.2811	0.1169	0.0235	0.0024
State 3	0.0815	0.2202	0.3374	0.2519	0.0915	0.0175
State 4	0.0175	0.0915	0.2519	0.3374	0.2202	0.0815
State 5	0.0024	0.0235	0.1169	0.2811	0.3290	0.2471
State 6	0.0001	0.0023	0.0211	0.0983	0.2334	0.6433

Table 6.2. Vehicular mobility: analytical transition matrix

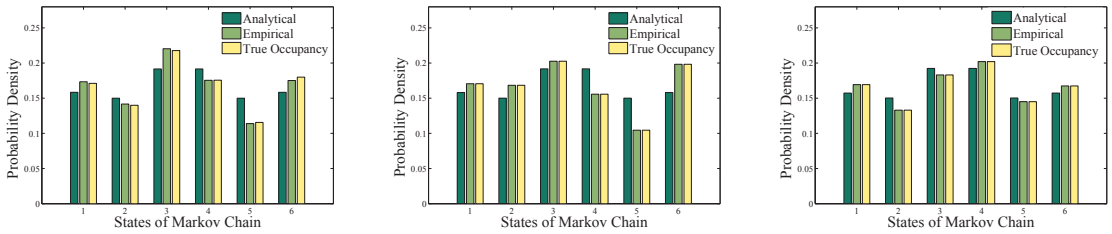
analytical and empirical transition matrix respectively for the vehicular trace whose distribution is characterized in Figure 6.1(a).

6.6.1.1 Steady State Behavior

In this subsection, we obtain the steady state distributions using the analytical and empirical transition matrices. We then compare them with the empirically observed shadowing-state occupancies (True Occupancy). The True Occupancy is calculated by counting the number of shadowing samples in each interval and then normalizing

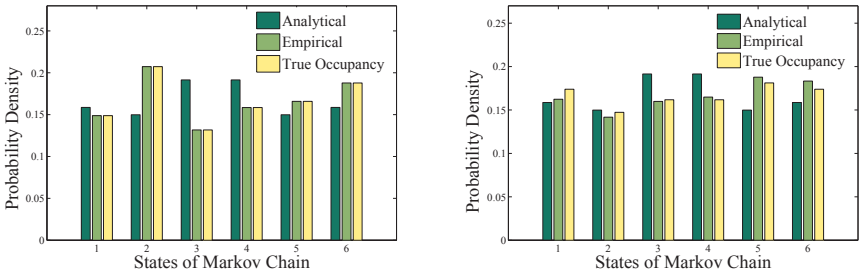
	State 1	State 2	State 3	State 4	State 5	State 6
State 1	0.6883	0.2078	0.0909	0	0	0.0130
State 2	0.2381	0.3651	0.3016	0.0794	0	0.0159
State 3	0.0619	0.1649	0.4948	0.1856	0.0515	0.0412
State 4	0.0380	0.0633	0.2152	0.4304	0.1519	0.1013
State 5	0	0.0192	0.0577	0.2885	0.3846	0.2500
State 6	0	0.0244	0.0488	0.0854	0.1829	0.6585

Table 6.3. Vehicular mobility: empirical transition matrix



(a) Trace 1 (b) Trace 2 (c) Trace 3

Figure 6.3. WiMAX-vehicular mobility: comparison of analytical and empirical steady state distribution of the Markov chain with the observed occupancy



(a) Trace 1 (b) Trace 2

Figure 6.4. WiMAX-pedestrian mobility: comparison of analytical and empirical steady state distribution of the Markov chain with the observed occupancy

them by the total number of samples. Figures 5.3 and 5.4 show the steady state behavior for the three vehicular and two pedestrian traces respectively. We observe that in terms of the steady state distribution, the parsimonious analytical approach and the empirical method match the true occupancies very closely; figures 5.3 and 5.4 show good agreement in the model-predicted and observed steady state shadowing values.

6.6.1.2 Transient Behavior

Having studied and validated the steady state behavior in the previous subsection, we focus on the transient state analysis here. We begin by determining the empirically observed distribution of transitioning to the different shadowing states as a function of the number of time steps (starting from any state). For example, from the traces

		Mean (2 step)	Variance (2 step)	Mean (5 step)	Variance (5 step)
Trace 1	Anal	3.19	1.99	3.35	2.64
	Emp	3.21	1.88	3.34	2.67
	True	3.08	1.57	3.2	2.28
Trace 2	Anal	3.19	1.99	3.35	2.63
	Emp	3.17	1.95	3.30	2.77
	True	3.29	1.89	3.5	2.88
Trace 2	Anal	3.17	1.8	3.3	2.54
	Emp	3.29	1.94	3.42	2.66
	True	3.16	1.19	3.47	2.47

Table 6.4. Vehicular mobility: transient state behavior

		Mean (2 step)	Variance (2 step)	Mean (5 step)	Variance (5 step)
Trace 1	Anal	3.18	1.93	3.33	2.6
	Emp	2.86	2.0	3.1	2.75
	True	2.78	1.06	2.86	1.57
Trace 2	Anal	3.15	1.74	3.30	2.52
	Emp	3.12	1.79	3.33	2.78
	True	3.06	0.92	3.07	1.05

Table 6.5. Pedestrian mobility: transient state behavior

collected, we calculate the probability of transitioning to the other states after 2 time steps starting from say, state 3. We once again refer to this as the True Occupancy. The transition probability distribution as a function of the number of time steps is also obtained from the analytical and empirical transition matrices. As the number of time steps increase the transient behavior will approach steady state.

We study the transient behavior of the Markov chain by comparing the first and second moments (the mean and variance) of the distributions obtained by the various approaches. We assign numerical values 1 through 6 for the different states of the Markov Chain. The states of the Markov chain being abstract, the absolute values of the mean and variance do not have any physical interpretation. The goal of this analysis is to compare the moments obtained by the different methods to determine

	Anal-True (2 step)	Emp-True (2 step)	Anal-True (5 step)	Emp-True (5 step)
Trace1	0.15	0.10	0.06	0.08
Trace2	0.22	0.11	0.11	0.06
Trace3	0.13	0.14	0.07	0.05

Table 6.6. Vehicular mobility: total variation

the closeness of the distributions. For sake of conciseness we represent the 2 and 5 time step transitions from state 3 in Tables 6.4 and 6.5 for the vehicular and pedestrian traces respectively. From Table 6.4 we observe that the mean and variance obtained by the analytical and empirical methods are close to true occupancies for the vehicular mobility scenario. For the pedestrian mobility scenario, the performance of the analytical and empirical methods are comparable to one another. But unlike the vehicular mobility case, their performance is not that close to the True Occupancy. We also studied the transition probability distribution from the other states graphically for the vehicular and pedestrian traces and similarly observed that performance of the empirical and analytical approaches were comparable but were sometimes not that close to the True Occupancy.

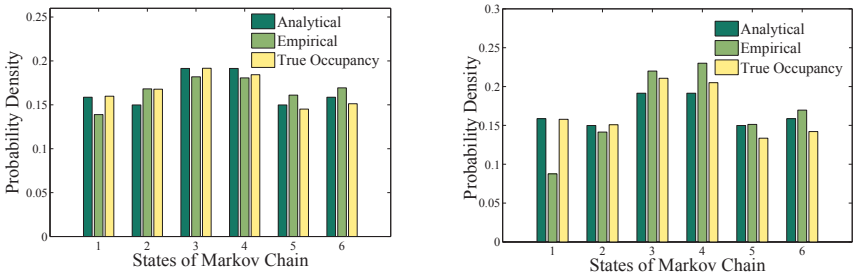
Tables 6.6 and 6.7 present the 2 and 5 time step total variation between the analytical and true occupancies as well as the empirical and true occupancies for the vehicular and pedestrian traces respectively. We observe from these tables that the total variation is small, which implies that the distributions are close to each other. Once again, we observe that the vehicular mobility results are better than the pedestrian mobility case. We would like to note here that the after about 25 steps, the probability distribution obtained by the analytical and empirical transition matrices from any state reaches very close (5%) to the steady state distribution for all the vehicular and pedestrian mobility cases.

	Anal-True (2 step)	Emp-True (2 step)	Anal-True (5 step)	Emp-True (5 step)
Trace1	0.21	0.20	0.20	0.16
Trace2	0.22	0.21	0.31	0.33

Table 6.7. Pedestrian mobility: total variation

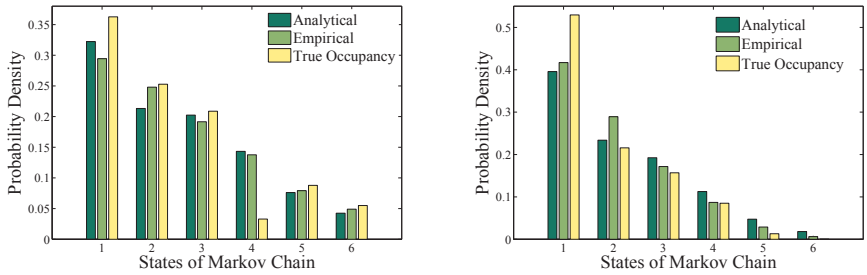
6.6.2 TFA Network

We study the steady state and transient state performance of our Markov Chain model with data collected over the TFA network. From Section 6.4, we know that in order to calculate the analytical transition matrix we first need to determine the variance and autocorrelation coefficient of shadowing. By using these two parameters we obtain the analytical transition matrix for the different APs. While determining the empirical transition matrix we ensure that inaccuracies are not introduced due to the temporal discontinuity in the traces. For example, if we have shadowing samples ranging from $t = 1$ to $t = 10$ and then again from $t = 15$ we neglect the state transition from $t = 10$ to $t = 11$ due to unavailability of this data. We then continue parsing the trace from time $t = 15$.



(a) Quince (b) Woodridge

Figure 6.5. TFA network: comparison of analytical and empirical steady state distribution of the Markov chain with the observed occupancy



(a) Quince (b) Woodridge

Figure 6.6. TFA network: comparison of analytical and empirical transient state (2-step) distribution of the Markov chain with the observed occupancy

6.6.2.1 Steady State Behavior

We compare the steady state performance of the analytical transition matrix, the empirical transition matrix and the True Occupancy for all the APs, but we show the results for only Quince and Woodridge in Figures 6.5(a) and 6.5(b) respectively. It is evident from Figure 6.5 that the heights of the bars for the three approaches are similar to each other which indicates that their steady state distributions are comparable to one another. We once again quantify the closeness of the distributions by the total variation. We observe that the mean total variation between the analytical and True Occupancy is 0.055 between the empirical and True Occupancy is 0.073 considering the traces from the 17 different APs.

6.6.2.2 Transient State Behavior

For the transient state analysis we once again look at the 2-step and 5-step distributions for the three methods, starting from any initial state for all the traces. Figures 6.6(a) and 6.6(b) depict the 2 -step transient distribution for Quince and Woodridge starting from state 1. The purpose of these graphs is just to let the readers visually appreciate how the transient state performance of the analytical, empirical and True Occupancy compare against one another. It can be observed from Figure 6.6 that after 2 steps the probability of transitioning to nearby states is higher than that to states further away.

We also study the mean and variance of the distributions of the three approaches as a function of the number of time steps. Our observations for the TFA network traces are similar to the WiMAX traces; we find that the mean and variance of the analytical and empirical approaches match closely. The means of the analytical and empirical methods are similar to the True Occupancy, but sometimes their variances differ a bit from the True Occupancy. Quantifying the closeness, we observe that considering all APs the difference between the means for the 2 step and 5 step distributions for the (analytical/empirical) methods and the True Occupancy is approximately 7% and 12% respectively. The difference between their variances at 2 step and 5 steps is however around 27% and 33% respectively. We observed that the mean total variation between the analytical and True Occupancy and that between the empirical and True Occupancy considering all APs is 0.13 and 0.09 respectively for the 2 step distribution. The same values for the 5-step distribution are 0.2 and 0.17 respectively.

Our analysis of the steady state and transient behavior for WiMAX and TFA network traces shows that the Markov chain model has good agreement between the model-predicted and true distributions, though the assumption of shadowing having an exponential autocorrelation function is violated, in particular for the WiMAX traces.

6.7 Conclusion

In conclusion, we can say that we developed and validated a finite-state Markov chain channel model to capture wireless channel variations due to shadowing. We obtained the Markov chain transition matrix in two ways: (i) via a parsimonious modeling approach in which shadowing effects are modeled as a log normally distributed random variable affecting the received power, and the transition probabilities are derived as functions of the variance and autocorrelation function of shadowing; (ii) via an empirical approach, in which the Markov chain transition matrix is calculated by directly measuring the changes in signal strengths collected in WiMAX network

and a multi-hop wireless network. Our experimental evaluation shows that the log-normal assumption of shadowing and the exponential autocorrelation assumption do not always hold true. Nonetheless, the Markov chain model showed good agreement between the model-predicted and observed values of shadowing for both the steady state and transient behavior.

CHAPTER 7

CONCLUSION

7.1 Thesis Summary

This thesis examined efficient routing and scheduling algorithms which exploit wireless channel variability to improve user-level performance.

We used modeling and analysis in Chapter 2 to investigate the performance benefits of opportunistic and cooperating forwarding in presence of multiple interfering transmissions. Rather than proposing new protocols or investigating the performance of specific opportunistic or cooperative transmission protocols, our goal was to compare the performance of idealized and representative opportunistic and cooperative forwarding strategies under common realistic assumptions. We began with a single flow linear network, and observed that cooperation outperforms opportunism. We then considered the case of more general network topologies with multiple flows and observed that unlike the linear network case, opportunism outperforms cooperation on average. We identified the interference resulting from the larger number of transmissions under cooperative forwarding as a cause for mitigating the potential gains achievable with cooperative forwarding.

Next, in Chapter 3 we investigated the tradeoff between state information collection (sampling frequency and number of bits per sample) and power consumption for a fixed source-to-destination goodput constraint. We formulated this problem as an optimization problem and observed that long sampling intervals fail to take advantage of the temporal correlation of link state estimates while short sampling intervals incur significant overhead. Similarly, using small number of bits per sample provides

very little information about the network state while large number of bits provides marginal additional information.

In Chapter 4, we studied the problem of data forwarding in heterogeneous networks that comprise of both stable as well as highly dynamic components and in which uniform routing or flooding at all network nodes does not perform well. We proposed a greedy algorithm (*adaptive-flood*) that dynamically classifies individual nodes as routers/flooders depending on network conditions with the objective of increasing the overall network goodput. We demonstrated via simulation that *adaptive-flood* achieves performance equivalent to, and in some cases significantly better than, that of network-wide routing or flooding alone.

A video streaming application was studied in Chapter 5, where we investigated the problem of scheduling different users streaming different video streams from a base station. We demonstrated that the problem is hard and proposed a lead-aware greedy algorithm for allocating channel resources (time slots) to different users. Real VBR video and wireless network traces were used to evaluate the performance of the greedy algorithm; we observed that the greedy algorithm has lower average number of application playout stalls and lower standard deviation when compared to other algorithms.

To aid application and network layer protocol design, we designed a Markovian model to capture the effect of shadowing on the received power in Chapter 6. We developed analytical and empirical approaches to compute the transition matrix of the Markov chain. We used signal strength measurements collected over a WiMAX network and a multi-hop wireless network and showed via experiments that the steady state and transient state performance of the Markovian model is close to that observed from real traces.

7.2 Future Work

This thesis outlined techniques for enhancing wireless network performance by designing efficient routing and scheduling algorithms for varied networks. The work done here can be extended in different ways and provides new avenues for future research. We outline some possible research directions here.

Consider the work on opportunism versus cooperation in Chapter 2. In this work, we have assumed that for any link i.i.d. fading samples are obtained at the beginning of each time slot. In realistic environments, fading will be correlated from one time slot to the next [108]. We demonstrated how to model opportunistic forwarding for correlated channels for a small linear network. A direction of future research is to develop models for analyzing opportunistic and cooperative forwarding for large general networks in presence of interference assuming correlated multipath fading. Another challenging topic for future research is to account for the intra-flow and inter-flow effect when there are multiple concurrent packets within a flow. This could potentially be done by considering a minimum spatial separation ('guard zone') between concurrently transmitting nodes such that their transmissions cause minimum self-interference for that flow. A potential complexity here will be to determine the size of the guard zone. Too large a guard zone will decrease the pipelining efficiency, whereas too small a guard zone would result in high interference. Another challenge is to compare opportunistic and cooperative forwarding in the presence of competing flows, with optimized (centrally or distributed) scheduling. The practical, but important, question of the overhead needed to achieve this coordination in practice, and whether this additional complexity is warranted by the increase in performance is also a question for future research.

In Chapter 3, we made several assumptions such as equal link-level path loss between different pairs of nodes, Gaussian quantization noise and exponential autocorrelation of shadowing. An immediate extension of our work is to relax these assumptions and study their impact on the sampling interval versus number of bits

per sample tradeoff. We are also interested in understanding the functional relationship between the sampling interval and the number of bits per sample. In the present formulation, we assumed that link values are encoded anew at the beginning of each sampling interval. We are interested in exploring the sampling interval versus number of bits per sample tradeoff by relaxing this assumption by leveraging the correlation of the underlying shadowing process and differentially encoding the link samples from one interval to the other. In our current work we considered a single source having multiple disjoint paths to a destination. A possible extension is to consider overlapping paths and multiple interfering sources. Some techniques and ideas related to modeling interference from Chapter 2 may be borrowed to address this question.

Let us next consider the data forwarding problem in heterogeneous networks in Chapter 4. As an extension of this work, we plan to evaluate the performance of *adaptive-flood* on general network topologies and on real network traces. While we studied the problem of classifying individual nodes as flooders/routers, we are also interested in exploring the case of preferentially routing/flooding data packets based on destinations. We assumed the presence of an underlying native routing algorithm and our router/flooder classification algorithm executed after routes had been determined. A different perspective for approaching this problem would be to take into account the mobility and link quality variations in the past, use it to predict future connectivity and then base the data forwarding strategy at individual nodes on these predictions. The plethora of prior work in mobility modeling [73] can be leveraged to predict future node mobility and connectivity. Goodput is not the only metric of concern in wireless networks - an equally important metric is delay and designing data forwarding strategies for heterogeneous networks with the objective of minimizing overall delay is an avenue for future research. Finally, considering different classes of data traffic and determining operation mode for individual nodes (flooder/router) so as to optimize performance is another challenging problem to address.

For the problem of scheduling multiple videos streams simultaneously from a base station in Chapter 5, a future research direction is to implement and test the performance and scalability of the greedy algorithm on a real testbed. Some of the challenges that can arise in such an implementation is accounting for delays in collecting state information from the different users, tuning the greedy algorithm to work in real time, especially when there is limited computational capability at the base station. An alternate approach for maximizing user QoE with respect to stalls is to schedule users so as to minimize the probability of stalling within an epoch instead of maximizing the minimum playout lead. The optimization problem can also be enriched by adding constraints related to the channel quality of the users. We have performed some preliminary investigation demonstrating the effect of bad channel quality on user experience. In this work, we assumed that the videos being streamed are available at the base station. Closely coupled with this problem is the problem of determining what videos to store, how to update stored video content over time and how to serve users whose requested video streams are currently not available at the base station. Designing scheduling algorithms for real-time videos instead of stored ones is another future research direction.

The Markov model described in Chapter 6 does not capture the effect of path loss on the received power and hence can be used for received power prediction, only in scenarios where the path loss is assumed to be constant in the time period of interest. Therefore as an extension of this work, we plan to explore the coarse timescale power prediction problem. Filtering techniques such as Kalman Filter [48] or Particle Filter [30] could be used to capture the effect of both path loss and shadowing on the received power. To construct such a filter one might model the state variables of the filter to be the received power, the distance between sender and receiver and the shadowing. Constructing a hidden Markov model to capture the effect of path loss and shadowing on the received power is also a direction of future

research. Testing the application-level performance improvement (e.g., in the video streaming application) by using these models is also a direction of future research.

APPENDIX A

CHAPTER 2

A.1 Proof of Lemma 2

Proof. By Lemma 1, substitute $\lambda_r = \frac{\mathsf{N}_0}{\mathsf{P}d_{r,j}^{-\alpha}}$,

$$\begin{aligned} P_{T,j} &= \int_{\beta}^{\infty} f_{\circledast m}(s)\mathrm{d}s \\ &= \Big(\prod_{r\in T}\lambda_r\Big)\sum_{r\in T}\frac{\exp(-\beta\lambda_r)}{\lambda_r\prod_{r'\in T\setminus\{r\}}(\lambda_{r'}-\lambda_r)} \\ &= \Big(\prod_{r\in T} d_{r,j}^{\alpha}\Big)\sum_{r\in T}\frac{\exp\left(\frac{-\beta\mathsf{N}_0}{\mathsf{P}d_{r,j}^{-\alpha}}\right)}{d_{r,j}^{\alpha}\prod_{r'\in T\setminus\{r\}}(d_{r',j}^{\alpha}-d_{r,j}^{\alpha})} \end{aligned} \tag{A.1}$$

□

A.2 Proof of Lemma 3

Proof. First, we write $\mathsf{S}_{i,j}^{I} = \frac{\hat{\mathsf{S}}_{i,j}}{\mathsf{N}_0+\hat{\mathsf{S}}_{I,j}}$ where $\hat{\mathsf{S}}_{i,j}$ and $\hat{\mathsf{S}}_{I,j}$ are $\mathsf{S}_{i,j}$ and $\mathsf{S}_{I,j}$ respectively, when $\mathsf{N}_0 = 1$. Hence, $\mathsf{S}_{i,j}^{I} \geq \beta$, if and only if $\hat{\mathsf{S}}_{i,j} \geq \beta\mathsf{N}_0$ and $\frac{\hat{\mathsf{S}}_{i,j}}{\beta} - \mathsf{N}_0 \geq \hat{\mathsf{S}}_{I,j}$.

Denote $\hat{f}_{i,j}(s)$ and $\hat{f}_{I,j}(s)$ as the probability density functions of $\hat{\mathsf{S}}_{i,j}$, and $\hat{\mathsf{S}}_{I,j}$ respectively.

By Lemma 1, we obtain:

$$\begin{aligned} P_{i,j}^{I} &= \int_{\beta\mathsf{N}_0}^{\infty}\hat{f}_{i,j}(b)\int_{0}^{\frac{b}{\beta}-\mathsf{N}_0}\hat{f}_{I,j}(s)\mathrm{d}s\mathrm{d}b \\ &= \int_{\beta\mathsf{N}_0}^{\infty}\lambda_i e^{-b\lambda_i}\int_{0}^{\frac{b}{\beta}-\mathsf{N}_0}\Big(\prod_{k\in I}\lambda_k\Big)\sum_{k\in I}\frac{e^{-s\lambda_k}}{\prod\limits_{k\in I\setminus\{k\}}(\lambda_{k'}-\lambda_k)}\mathrm{d}s\mathrm{d}b \end{aligned}$$

where $\lambda_i = \frac{1}{\mathsf{P}d_{i,j}^{-\alpha}}$ and $\lambda_k = \frac{1}{\mathsf{P}d_{k,j}^{-\alpha}}$.

Further calculation shows that

$$\int_{\beta \mathsf{N}_0}^{\infty} e^{-b\lambda_i} \int_0^{\frac{b}{\beta}-\mathsf{N}_0} e^{-s\lambda_k} \mathrm{d}s\mathrm{d}b = \frac{e^{-\beta \mathsf{N}_0 \lambda_i}}{\lambda_i \lambda_k + \beta \lambda_i^2} \tag{A.2}$$

Therefore,

$$P_{i,j}^I = \Big(\prod_{k\in I} \lambda_k\Big) \sum_{k\in I} \frac{e^{-\beta \mathsf{N}_0 \lambda_i}}{(\lambda_k + \beta\lambda_i) \prod\limits_{k'\in I\setminus\{k\}} (\lambda_{k'} - \lambda_k)} \tag{A.3}$$

□

A.3 Proof of Lemma 5

Proof. Eqn. (2.15) follows from Lemma 10, by substituting $p[i] = ip^{i^\alpha}$ and $q[i] = 1 - p^{i^\alpha}$. When $n \geq 2$, we expand the terms in Eqn. (2.15) for the first few terms. Then, we obtain:

$$\begin{aligned} \mathsf{H}_{\mathrm{op}}[n] &= \sum_{j=1}^{3} jp^{j^\alpha} \Big(\prod_{\ell=j+1}^{2} 1 - p^{\ell^\alpha} \Big) + \mathrm{O}(p^{4^\alpha}) \\ &= p + 2p^{2^\alpha} - p^{1+2^\alpha} + 3p^{3^\alpha} + \mathrm{O}(p^{1+3^\alpha}) \end{aligned} \tag{A.4}$$

□

Lemma 10. *Consider a more general recurrence eqn.:*

$$F_{p,q}[1] = p[1], \quad F_{p,q}[n] = p[n] + q[n]F_{p,q}[n-1] \tag{A.5}$$

where $p[n]$ and $q[n]$ are general functions of n. Then, we solve $F_{p,q}[n]$ by:

$$F_{p,q}[n] = \sum_{j=1}^{n} p[j] \Big(\prod_{\ell=j+1}^{n} q[\ell] \Big) \tag{A.6}$$

Proof. First, it is easy to see $F_{p,q}[1] = p[1]$ from Eqn. (A.6). Next, we substitute Eqn. (A.6) into Eqn. (A.5).

$$\begin{aligned} & p[n] + q[n]F_{p,q}[n-1] \\ &= p[n] + q[n]\sum_{j=1}^{n-1} p[j]\Big(\prod_{\ell=j+1}^{n-1} q[\ell]\Big) \\ &= p[n] + \sum_{j=1}^{n-1} p[j]\Big(\prod_{\ell=j+1}^{n} q[\ell]\Big) \\ &= \sum_{j=1}^{n} p[j]\Big(\prod_{\ell=j+1}^{n} q[\ell]\Big) = F_{p,q}[n] \end{aligned} \tag{A.7}$$

Hence, Eqn. (A.6) is a solution to Eqn. (A.5). □

A.4 Proof of Theorem 1

Proof. Recall that $H_{\text{op}}[n]$ to be the expected number of hops that one transmission can reach, when the destination is n hops away. Then, $H_{\text{op}}[n]$ satisfies Eqn. (2.14), because with probability p^{n^α}, the source can reach the destination in one transmission, otherwise it reaches some node before the destination as if the destination is $(n-1)$ hops away, from which the expected number of hops is $H_{\text{op}}[n-1]$.

To see $n/H_{\text{op}}[n] \leq N_{\text{op}}[n]$, we note that

$$H_{\text{op}}[n] > H_{\text{op}}[n-1] > ... > H_{\text{op}}[1] \tag{A.8}$$

There are a decreasing expected number of hops that opportunistic forwarding can reach by sequential transmissions. Hence, $n/H_{\text{op}}[n]$ is always smaller than the actual expected number of transmissions $N_{\text{op}}[n]$. □

A.5 Proof of Theorem 2

We first define some notations. Let $T_m = \{s, r_1, ..., r_{m-1}\}$ be a group of cooperative transmitters. Using Eqn. (2.7) and substitute $d_{r,t} = rd$, then we obtain the probability

that destination t can successfully receive the packet from a set of transmitters T_m is:

$$P_{T_m,t}[n] = \sum_{r=0}^{m-1} \frac{p^{(n+r)^\alpha}}{\prod_{r'=0:r'\neq r}^{m-1} \left(1 - \left(\frac{r+n}{s+n}\right)^\alpha\right)} \tag{A.9}$$

Denote by $P_{\text{co}}^{\text{suc}}[n]$ the probability that the packet successfully reaches the destination in one time slot via cooperative forwarding. Since $m \leq n$,

$$P_{\text{co}}^{\text{suc}}[n] \leq P_{T_n,t}[n] \tag{A.10}$$

Proof. First, by Lemma 11, $P_{\text{co}}^{\text{suc}}[n] = \mathrm{O}\big(np^{3n}\big)$. There always exist some constants C and c, such that $P_{\text{co}}^{\text{suc}}[n] \leq Cp^{cn}$, because $n^{-1} \gg p^{\epsilon n}$ for any small ϵ.

Next, consider $F_{p,q}[n]$, the general form of the expected number of hops a packet can reach in one time slot, as defined in Eqn. (A.5). Note that $F_{p,q}[n] \geq F_{p',q'}[n]$ if $p[j] \geq p'[j]$ and $p[j] + q[j] = 1 = p'[j] + q'[j]$ for all j. This can be shown by mathematical induction on Eqn. (A.5), and under the assumption that $0 < F_{p,q}[n] \leq n$.

We consider $\tilde{H}_{\text{co}}[n]$, which is the solution to the following recurrence equation:

$$\begin{aligned} \tilde{H}_{\text{co}}[n] &= nP_{T_n,t}[n] + (1 - P_{T_n,t}[n])\tilde{H}_{\text{co}}[n-1], \\ \tilde{H}_{\text{co}}[1] &= P_{T_1,t}[1] \end{aligned} \tag{A.11}$$

We note that $0 < \tilde{H}_{\text{co}}[n] \leq n$.

By Lemma 11, $H_{\text{co}}[n] \leq \tilde{H}_{\text{co}}[n]$. Hence, we obtain:

$$\begin{aligned} H_{\text{co}}[n] &\leq \sum_{j=1}^{n} Cjp^{cj}\Big(\prod_{\ell=j+1}^{n} 1 - Cp^{c\ell}\Big) \\ &= \sum_{j=1}^{n} C\sqrt{j}\cdot\sqrt{j}p^{cj}\Big(\prod_{\ell=j+1}^{n} 1 - Cp^{c\ell}\Big) \\ &\leq \frac{C}{\sqrt{c}}\sqrt{n}\sum_{j=1}^{n} \sqrt{cj}p^{(\sqrt{cj})^2}\Big(\prod_{\ell=j+1}^{n} 1 - p^{(\sqrt{c\ell})^2}\Big) \\ &= \mathrm{O}(\sqrt{n})\cdot H_{\text{op}}[n] \end{aligned} \tag{A.12}$$

□

Lemma 11. *Consider $\alpha = 2$. When $p < \frac{1}{4}$,*

$$P_{\text{co}}^{\text{suc}}[n] \leq P_{T_n,t}[n] = \mathrm{O}\big(np^{3n}\big) \tag{A.13}$$

Proof. Define $\mathfrak{A}_r \triangleq p^{(n+r)^2} / \prod_{r'=0:r'\neq r}^{m-1} \left(1 - \left(\frac{r+n}{s+n}\right)^2\right)$. We assume that m is even, for convenience of analysis.

$$P_{T_m,t}[n] = \sum_{r=0}^{m-1} \mathfrak{A}_r = \sum_{r'=0}^{m/2} \mathfrak{A}_{2r'}(1 + \frac{\mathfrak{A}_{2r'+1}}{\mathfrak{A}_{2r'}}) \tag{A.14}$$

Note that

$$\begin{aligned}
\frac{\mathfrak{A}_{r+1}}{\mathfrak{A}_r} &= \frac{p^{(n+r+1)^2} \prod_{r'=0:r'\neq r}^{m} \left(1 - \left(\frac{r+n}{r'+n}\right)^2\right)}{p^{(n+r)^2} \prod_{r'=0:r'\neq r+1}^{m} \left(1 - \left(\frac{r+1+n}{r'+n}\right)^2\right)} \\
&= \frac{p^{(2n+2r+1)} \left(1 - \left(\frac{r+n}{r+1+n}\right)^2\right)}{\left(1 - \left(\frac{r+1+n}{r+n}\right)^2\right)} \prod_{r'=0:r'\neq r,r+1}^{m} \frac{1 - \left(\frac{r+n}{r'+n}\right)^2}{1 - \left(\frac{r+1+n}{r'+n}\right)^2} \\
&= \frac{-p^{(2n+2r+1)}(n+r)^2}{(1+n+r)^2} \prod_{r'=0:r'\neq r,r+1}^{m} \frac{(r-r')(2n+r+r')}{(1+r-r')(1+2n+r+r')} \\
&= \frac{-p^{(2n+2r+1)}(n+r)(2n+r)(m-r)}{(1+n+r)(1+r)(1+m+2n+r)}
\end{aligned} \tag{A.15}$$

Note that $\frac{(n+r)(2n+r)(m-r)}{(1+n+r)(1+r)(1+m+2n+r)} \leq \frac{2nm}{m+2n}$. Since $m \leq n$, $p^{(2n+2r+1)} \frac{2nm}{m+2n} < 1$. Hence, we can bound $P_{T_m,t}$ by:

$$\begin{aligned}
P_{T_m,t}[n] &\leq \sum_{r=0}^{m/2} \Big(1 - \frac{p^{(2n+2r+1)}(n+r)(2n+r)(m-r)}{(1+n+r)(1+r)(1+m+2n+r)}\Big) \\
&\qquad \cdot \prod_{r'=0}^{2r} \big(p^{(2n+2r'+1)} \frac{2nm}{m+2n}\big) \\
&\leq \sum_{r=0}^{m/2} \big(p^{(2n+4r+1)} \frac{2nm}{m+2n}\big) = \mathrm{O}\Big(\frac{mnp^{2n}}{m+2n}\Big)
\end{aligned} \tag{A.16}$$

Also, further calculation shows that

$$\mathfrak{A}_0 = \frac{p^{n^2}}{\prod_{s=1}^{m-1} \left(1 - \left(\frac{n}{s+n}\right)^2\right)} = \mathrm{O}(4^n p^{n^2}) \tag{A.17}$$

Hence,

$$P_{T_m,t}[n] = \mathrm{O}\Big(\frac{mnp^{2n}4^n p^{n^2}}{m+2n}\Big) \tag{A.18}$$

Consider $p < \frac{1}{4}$, and $m \leq n$, when p is small:

$$P_{T_n,t}[n] = \mathrm{O}\big(np^{3n}\big) \tag{A.19}$$

□

A.6 Proof of Theorem 3

Proof. By substitution, we obtain:

$$= \frac{\frac{\mathrm{T}_{\mathrm{op}}^{s_1\to t_1}}{\mathrm{T}_{\mathrm{co}}^{s_1\to t_1}}}{\frac{\big(2p^2-3(1+\beta)\big)\big(p\big(2+3\beta+\beta^2\big)+4\big(1+3\beta+2\beta^2\big)-p^3(2+\beta)-p^2(2+4\beta)\big)}{2(-3+2p^2-4\beta)((1+\beta)^2-p^3(2+\beta)+p(2+3\beta+\beta^2))}}$$

Because $0 \leq p \leq 1$,

$$\begin{aligned}
&\big(p\,(2+3\beta+\beta^2)+4\,(1+3\beta+2\beta^2)-p^3(2+\beta)\\
&\quad -p^2(2+4\beta)\big)-\big((1+\beta)^2-p^3(2+\beta)+p\,(2+3\beta+\beta^2)\,\big)\\
&=3+10\beta+7\beta^2-2p^2(1+2\beta)\geq 1+6\beta+7\beta^2\\
&\quad (2p^2-3(1+\beta))-2\,(-3+2p^2-4\beta)\\
&=3-2p^2+5\beta\geq 1+5\beta
\end{aligned}$$

Therefore, we obtain $\frac{\mathrm{T}_{\mathrm{op}}^{s_1\to t_1}}{\mathrm{T}_{\mathrm{co}}^{s_1\to t_1}} \geq 1$ □

A.7 Proof of Theorem 4

Proof. First, we consider the actual Markov chain of multiple flows defined in Section 2.6.1. We recall that the stationary distribution of the actual Markov chain of multiple flows is π over the set $\{r : r_f \in \mathcal{P}_f\}$. Note that π is the fixed-point to

Eqns. (2.19)-(2.20). Let $\mathcal{R}$ be the random set of active relays that are transmitting for the flows. Let the random total interference level to node j be:

$$I_j(\mathcal{R}) \triangleq \sum_{r \in \mathcal{R} \setminus \{j\}} |x_{r,j}|^2 \cdot \mathsf{P} d_{r,j}^{-\alpha} \tag{A.20}$$

Let $P_{i,j}^f$ be the packet reception probability from node i to j, where j belongs to flow f. Since the set of active relays is random, $P_{i,j}^f$ is a random variable, with expected value

$$\begin{aligned} \mathbb{E}[P_{i,j}^f] = & \quad \mathbb{E}\Big[\mathbb{P}\Big\{ \tfrac{|x_{i,j}|^2 \mathsf{P} d_{i,j}^{-\alpha}}{\mathsf{N}_0 + I_j(\mathcal{R})} \geq \beta \Big\}\Big] \\ = & \quad \mathbb{E}\Big[\exp\Big(\tfrac{-\beta\left(\mathsf{N}_0 + I_j(\mathcal{R})\right)}{\mathsf{P} d_{i,j}^{-\alpha}} \Big)\Big] \end{aligned} \tag{A.21}$$

The throughput of a flow can be obtained by averaging over time. By the ergodicity of the Markov model, it is equivalent to averaging over the stationary distribution. Let the stationary distribution of each state j of flow f be $\pi_f(j) \triangleq \sum_{r: r_f = j} \pi(r)$, and $\bullet\,(r \to v_{\mathrm{d}(f)})$ be the indicator function that there is a state transition from r to $v_{\mathrm{d}(f)}$ for flow f at a timeslot.

$$\begin{aligned} \mathrm{T}_{\mathrm{op}}(f) = & \sum_{r \in \mathcal{P}_f \setminus \{v_{\mathrm{d}(f)}\}} \pi_f(r) \cdot \mathbb{E}_\pi\Big[\bullet\,(r \to v_{\mathrm{d}(f)})\Big] \\ = & \sum_{r \in \mathcal{P}_f \setminus \{v_{\mathrm{d}(f)}\}} \pi_f(r) \cdot \mathbb{E}_\pi\Big[P_{r,v_{\mathrm{d}(f)}}^f \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r} \Big(1 - P_{r,v}^f\Big)\Big] \\ = & \sum_{r \in \mathcal{P}_f \setminus \{v_{\mathrm{d}(f)}\}} \pi_f(r) \cdot \mathbb{E}_\pi[P_{r,v_{\mathrm{d}(f)}}^f] \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r} \Big(1 - \mathbb{E}_\pi[P_{r,v}^f]\Big) \end{aligned}$$

The last equality is due to the assumption of independent random fading among pairs of nodes (i.e., fading coefficient $|x_{i,j}|^2$ is an i.i.d. random variable for any pair of nodes i, j).

Second, recall that $(\hat{\pi}_f : f \in \mathscr{F})$ is the fixed-point to Eqns. (2.31)-(2.34), which is also a fixed-point to Eqn. (2.34) and the following equation:

$$\hat{\mathbf{P}}_{r,r'}(\hat{\pi}) \triangleq \prod_{f \in \mathscr{F}} \hat{P}^f_{r_f,r'_f}(\hat{\pi}_{\neg f}) \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r'_f} \Big(1 - \hat{P}^f_{r_f,v}(\hat{\pi}_{\neg f})\Big) \tag{A.22}$$

Note that Eqns. (A.22) & (2.34) are comparable to Eqns. (2.19)-(2.20) for stationary distribution π.

The throughput can be given by:

$$\begin{aligned}
\hat{\mathrm{T}}_{\mathrm{op}}(f) &= \sum_{r \in \mathcal{P}_f \setminus \{v_{\mathrm{d}(f)}\}} \hat{\pi}_f(r) \cdot \mathbf{P}^f_{r,v_{\mathrm{d}(f)}}(\hat{\pi}_{\neg f}) \\
&= \sum_{r \in \mathcal{P}_f \setminus \{v_{\mathrm{d}(f)}\}} \hat{\pi}_f(r) \cdot \hat{P}^f_{r,v_{\mathrm{d}(f)}}(\hat{\pi}_{\neg f}) \cdot \prod_{v \in \mathcal{P}_f : v \succ_f r} \Big(1 - \hat{P}^f_{r,v}(\hat{\pi}_{\neg f})\Big)
\end{aligned}$$

Next, we compare $\mathbb{E}_{\tilde{\pi}}[P^f_{r,r'}]$ and $\hat{P}^f_{r,r'}(\tilde{\pi}_{\neg f})$ under a certain distribution $\tilde{\pi}$ over r. Since $\exp(-x)$ is a convex function, by Jensen's inequality,

$$\begin{aligned}
\mathbb{E}_{\tilde{\pi}}[P^f_{i,j}] = &\ \mathbb{E}_{\tilde{\pi}}\Big[\exp\Big(\tfrac{-\beta\left(\mathsf{N}_0 + I_j(\mathcal{R})\right)}{\mathsf{P} d_{i,j}^{-\alpha}}\Big)\Big] \\
= &\ \geq \exp\Big(\tfrac{-\beta\left(\mathsf{N}_0 + \mathbb{E}_{\tilde{\pi}}[I_j(\mathcal{R})]\right)}{\mathsf{P} d_{i,j}^{-\alpha}}\Big)
\end{aligned} \tag{A.23}$$

Then, by the definition of $P^f_{i,j}(\tilde{\pi}_{\neg f})$, we have $\mathbb{E}_{\tilde{\pi}}[I_j(\mathcal{R})] = \hat{I}^f_j(\tilde{\pi}_{\neg f})$. This implies

$$\mathbb{E}_{\tilde{\pi}}[P^f_{i,j}] \geq P^f_{i,j}(\tilde{\pi}_{\neg f}) \tag{A.24}$$

If $\mathbb{E}_{\tilde{\pi}}[\hat{P}^f_{i,j}] \geq \hat{P}^f_{i,j}(\tilde{\pi}_{\neg f})$ for any pair of nodes i, j and any distribution $\tilde{\pi}$, then every forwarding operation carries a lower packet reception probability in the latter case. Hence, the latter case must have decreased throughput under the respective fixed-point:

$$\mathrm{T}_{\mathrm{op}}(f) \geq \hat{\mathrm{T}}_{\mathrm{op}}(f) \tag{A.25}$$

□

APPENDIX B

CHAPTER 5

B.1 Algorithm for proof of Lemma 7

In this section, we present an optimal dynamic programming based algorithm for LMVT. If the number of videos is a constant, the algorithm runs in time that is pseudo-polynomial in terms of the input. We first introduce an assumption and some notations for the algorithm.

Preliminaries. In this algorithm, we assume that the server is allowed to transmit any $b_{ij} \leq r_{ij}$ number of bits to the video i in slot j. (Recall that, r_{ij} is the rate of video i in slot j.) We can modify a solution in which the server transmits $b_{ij} < r_{ij}$ bits to video i in slot j, to another solution in which the server transmits r_{ij} bits in slot j to video i, without decreasing the objective value. Thus, the optimal value of a given problem instance remains the same with and without this assumption.

A transmission vector (or *Tx-vector*) is a an n-tuple $< a_1, \ldots, a_n >$, where the i^{th} element indicates the number of bits to be transmitted to video i. For a Tx-vector T, we denote by $T[i]$ the i^{th} element of T. For a given number of total slots, say z, and a Tx-vector T, we say that T is z-feasible if there is a slot allocation such that, for each $1 \leq i \leq n$, video v_i receives a total of $T[i]$ bits in the allocation. Let $F(z, T)$ denote the predicate whether Tx-vector T is z-feasible. Define $F(0, T)$ to be true if $T =< 0, \ldots, 0 >$, and false, otherwise. For any pair of Tx-vector $T1$ and $T2$, we define the relation $T1 \preceq T2$ to be true if and only if $T1[i] \leq T2[i]$ for every $1 \leq i \leq n$. Note that, the maximum number of bits that can be transmitted to a video v_i in the epoch is $Q(\sum_{j=1}^{N_{ep}^{sl}} r_{ij})$, and we denote this value by b_i^{max}. This maximum value is achieved when all slots are allocated to v_i. (Recall that N_{ep}^{sl} denotes the total number of slots

in an epoch, and let the slots in an epoch be numbered from 1 to N_{ep}^{sl}.) Finally, for any Tx-vector T, and for each $1 \leq i \leq n$, let W_i^T denote a Tx-vector that is identical to T except that $W_i^T[i]$ is maximum$(T[i] - r_{im}, 0)$.

Algorithm. In our algorithm, using dynamic programming we find a Tx-vector T (and its corresponding slot allocation) that is N_{ep}^{sl}-feasible, and which has the highest objective value. We first note that for any N_{ep}^{sl}-feasible Tx-vector T, $T \preceq < b_1^{max}, \ldots, b_n^{max} >$, since b_n^{max} is the maximum number of bits that can be transmitted to a video v_i in the epoch.

We present our algorithm in two steps. First, we present an algorithm that finds the z-feasibility of all Tx-vectors $< 0, \ldots, 0 > \preceq T \preceq < b_1^{max}, \ldots, b_n^{max} >$, for all $1 \leq z \leq N_{ep}^{sl}$. Then we extend the algorithm to compute the min-lead values for the Tx-vectors, and select a vector that has the maximum value of min-lead.

We first state the following straightforward lemma.

Lemma 12. *For any pair of Tx-vectors $T1$ and $T2$ such that $T1 \preceq T2$, and for any $z \geq 0$, if $T2$ is z-feasible then $T1$ is also z-feasible.*

The above lemma holds because a slot allocation corresponding to z-feasibility of $T2$ can be easily modified to a slot allocation corresponding to z-feasibility of $T1$ by appropriately reducing the number of transmitted bits in the former allocation. We omit this straightforward proof. Next we present a lemma that immediately gives our dynamic programming algorithm.

Lemma 13. *For $m \geq 1$, $F(m, T)$ is true if and only if at least one of the n predicates $F(m-1, W_i^T)$, $1 \leq i \leq n$ is true.*

Proof. Suppose $F(m-1, W_i^T)$ is true for some $1 \leq i \leq n$. Then, there is a slot allocation using $m-1$ slots such that for every $1 \leq j \leq n$, video v_j is transmitted $W_i^T[j]$ bits. Consider a slot allocation for m slots that is identical to that for W_i^T until slot $m-1$, and the m^{th} slot is allocated to v_i with minimum$(r_{im}, T[i])$ transmission

bits. Then, in this new slot allocation, every video is transmitted the number of bits specified in Tx-vector T. Thus, $F(m, T)$ is true.

For the converse, suppose $F(m, T)$ is true for some T and $m \geq 1$, and hence, there is a slot allocation for T using m slots. Consider the video, say v_i, to which slot m is allocated. Let b denote the number of bits that are allocated in slot m. Then, by removing the allocation for slot m in T, we obtain an allocation using $m-1$ slots for a Tx-vector T' that is identical to T except that $T'[i] = T[i] - b$. Thus, $F(m-1, T')$ is true. Since, $b \leq r_{im}$, $W_i^T \preceq T'$. Then, from Lemma 12, $F(m-1, W_i^T)$ is also true. □

Finding feasible Tx-vectors. Our dynamic programming algorithm iterates over the number of slots m that varies from 1 to N_{ep}^{sl}. In a step of the iteration, the algorithm computes $F(m, T)$ from $F(m-1, *)$ using Lemma 13, for all $T \preceq < b_1^{max}, \ldots, b_n^{max} >$. Also, within a step of the iteration, $F(m, *)$s are computed in arbitrary but fixed order of the Tx-vectors, since their values are dependent only on $F(m-1, *)$s that are computed in the previous step of the iteration.

Finding feasible Tx-vectors with maximum value of its min-lead. Recall that, we approximated the (expected) lead of a video $E[L_i]$ by $g_i - o_i + \Phi_i(E[Y_i])$, where Y_i is the number of bits transmitted to client i in the epoch, and Φ_i is the inverse playback curve of the video for client i. Before starting our algorithm, we pre-compute the inverse playback curve for each video. Then, upon computing each entry $F(*, T)$ in the above dynamic programming algorithm, if $F(*, T)$ is true then (1) we compute the min-lead of T by computing the lead of each video i using $g_i - o_i + \Phi_i(T[i])$, and (2) we store an *allocation* pointer to a Tx-vector W_i^T that is true for T (in Lemma 13). Finally, in the iteration when $m = N_{ep}^{sl}$, we maintain a pointer to a N_{ep}^{sl}-feasible vector with the maximum value of its min-lead among all the N_{ep}^{sl}-feasible vectors seen so far (and the pointer is updated after each $F(N_{ep}^{sl}, *)$ computation). Thus, at the end of algorithm, we obtain a pointer to a N_{ep}^{sl}-feasible Tx-vector T' with the maximum

value of min-lead, and we follow the N_{ep}^{sl} *allocation* pointers from T' to $< 0, \ldots, 0 >$ to obtain an optimal slot allocation.

Time and space requirements. For each of the N_{ep}^{sl} slots, there are at most $\prod_{i=1}^{n} b_i^{max}$ Tx-vectors. Thus, there are $D = N_{ep}^{sl} \prod_{i=1}^{n} b_i^{max}$ entries in the dynamic programming table. To compute each entry $F(m, T)$, we need to lookup n entries in the table, and to compute the min-lead for each entry, we compute n lead values. Thus, the time-complexity of the above algorithm is $O(nD)$. We store the boolean feasibility value, min-lead value and the allocation pointer for each entry in the dynamic programming table. The size of the allocation pointer is at most D, and we can safely assume that the maximum value of lead is less than D. Thus the space-complexity of the algorithm is $O(Dlog(D))$.

B.2 Time Complexity of Greedy Algorithm

We elaborate here on the time complexity of the initialization of the greedy algorithm. At the beginning of each epoch when the greedy algorithm is executed, it is necessary to determine the expected rate received by clients in the different slots. For the k^{th} interval, the algorithm multiplies matrices $I_i A^k$ with R, and therefore it incurs a time complexity of $O(K^2)$. Note that to compute $I_i A^k$ the algorithm has to only multiply $I_i A^{k-1}$ with A. The matrix multiplication has to be performed for each interval and for each client incurring an overall complexity of $O(nN_{in}^{sl}K^2)$. Further, the algorithm has to assign r_{ij} for every client in each time slot which incurs a time complexity of $O(nN_{ep}^{sl})$. Hence the total complexity of the initialization step is $O(\max(nN_{ep}^{sl}, nN_{in}^{sl}K^2))$.

We note that the total time complexity of the implementation (both initialization and greedy function) can be further improved by keeping only a single variable for all r_{ij} in an interval for a user and using better data structures such as heaps.

B.3 Proof of Lemma 8

Proof. As the rate of a client i does not change within an epoch, each slot that is allocated to the client i provides a constant number of bits, say r_i. In this setting, the greedy algorithm simply chooses the client i that has the lowest id among the clients with the minimum lead, and selects the next available slot and allocates it to i. The proof of optimality is by induction on the number of allocated slots.

For the induction, we first introduce some notation and observations. At any point in the execution of the LMVT algorithm, the lead of a client can only change on receiving sufficient slots for the client's next video frame, and therefore, the client's lead can change only by a multiple of $1/F$. For any LMVT solution (slot allocation to clients) X, let l_i^X denote the lead of client i in solution X, and let $l_{min}^X = min_i\{l_i^X\}$ be the minimum lead in X. Let $sl(X, j)$ denote the number of slots allocated to client j in solution X. Note that for a solution Y and client k, if $l_j^X > l_k^Y$ then $sl(X, j) > sl(Y, k)$, on the other hand, if $sl(X, j) \geq sl(Y, k)$ then $l_j^X \geq l_k^Y$.

Base Case: If only 1 slot is available, the greedy algorithm allocates it to a client with the minimum lead and therefore the minimum lead is maximized.

Induction Step: Let us assume that the greedy algorithm yields an optimal solution G for every $d \leq c$ slots. Let $G(c+1)$ be the solution given by the greedy algorithm for $c+1$ slots. We must prove that $G(c+1)$ is optimal. To show by contradiction, let us assume that there exists an alternate solution $S(c+1) \neq G(c+1)$ that is optimal for $c+1$ slots, and $S(c+1)$ has a higher minimum lead than $G(c+1)$. Thus, $l_{min}^{S(c+1)} > l_{min}^{G(c+1)}$ (i.e., $l_{min}^{S(c+1)} \geq l_{min}^{G(c+1)} + 1/F$) [Observation **A0**]. Let client i have the lowest id among the clients with the minimum lead in $G(c)$. After the $(c+1)$th slot is allocated to i by the greedy algorithm, we have one of the following two cases:

Case 1: Minimum lead changes, i.e., $l_{min}^{G(c+1)} > l_{min}^{G(c)}$.

Let j be a client with the minimum lead in $G(c+1)$, i.e., $l_{min}^{G(c+1)} = l_j^{G(c+1)}$ (j need not be different from i). Then $l_j^{S(c+1)} \geq l_{min}^{S(c+1)} > l_{min}^{G(c+1)} = l_j^{G(c+1)}$ [Observation **A**]. Thus, j is allocated at least one more slot in $S(c+1)$ than in $G(c+1)$. Let us remove a slot from j in $S(c+1)$ to obtain a solution $S(c)$ for c slots. Since we have only removed one slot from j in $S(c+1)$ to obtain $S(c)$, $l_j^{S(c)} \geq l_j^{S(c+1)} - 1/F \geq l_j^{G(c+1)}$ [Observation **B**], and $l_{min}^{S(c)} = min\{l_j^{S(c)}, l_{min}^{S(c+1)}\} \geq l_j^{G(c+1)}$ (where the last inequality follows from inequalities **A** and **B**). Thus, we have $l_{min}^{S(c)} \geq l_j^{G(c+1)} = l_{min}^{G(c+1)} > l_{min}^{G(c)}$ which is a contradiction since $G(c)$ is optimal for c slots.

Case 2: Minimum lead remains unchanged at some value z, i.e., $l_{min}^{G(c+1)} = l_{min}^{G(c)} = z$.

Observe that this can happen either when (a) i has not received data constituting an entire frame and therefore its lead has not advanced (b) i received data constituting one or more frames and its lead advanced but there is another client j such that $l_j^{G(c)} = l_i^{G(c)} = z$.

We first consider the case when $z = 0$. As $l_{min}^{S(c+1)} \geq z + 1/F > 0$ (from **A0**), in $S(c+1)$ every client is allocated enough slots for at least its first frame, Thus, for each client j, the minimum number of slots needed for the first frame, say sl'_j, is less or equal to than $sl(S(c+1), j)$, and therefore, $\sum_j sl'_j \leq c+1$. Now consider the execution of the greedy algorithm until the minimum lead (over all videos) becomes greater than 0. The algorithm selects a client j, in the increasing order of their client id, and allocates client j enough slots for its first frame, i.e., sl'_j, and then moves to the next frame. Therefore, given $c + 1 \geq \sum_j sl'_j$ slots, the greedy algorithm will allocate sufficient slots to each client for its first frame, and hence, the allocation will have a minimum lead of at least $1/F$. Thus, $l_{min}^{G(c+1)} \geq 1/F$, a contradiction.

We now consider the case when $z > 0$. Let us look back in time to the point in the greedy algorithm's execution when the minimum lead in G has last changed. Let us assume that this occurred δ slots back, i.e., $l_{min}^{G(c-\delta)} = z - 1/F$ and $l_{min}^{G(c-\delta+1)} = \ldots =$

$l_{min}^{G(c+1)} = z$ [Observation **C**]. Thus, in the solution $G(c+1-\delta)$, there must have been a set of clients P each with lead z.

Consider the period of execution of the greedy algorithm while going from $G(c+1-\delta)$ to $G(c+1)$. In this period, the algorithm must have assigned slots only to clients in P. Also, no client in P would have received slots more than what is required for its next one frame (because on receiving slots required for one frame, the client's lead increases, and it does not remain a client with the minimum lead) [Observation **C1**]. Let $P1$ be the set of clients in P that have received sufficient slots for their next frame in this period, and $P2$ be the remaining set of clients in P (that have not received enough slots for their next frame in this period). We note that $P2$ cannot be an empty set, otherwise, the lead of $G(c+1)$ would be higher than $G(c+1-\delta)$.

Let q be any client in $P2$. Then $l_q^{G(c+1)} = z$. Since, from our initial assumptions, $l_{min}^{S(c+1)} > l_{min}^{G(c+1)} = z$, $l_q^{S(c+1)} \geq l_{min}^{S(c+1)} > z = l_q^{G(c+1)}$ [Observation **D**]. Also, for any client j in $P1$, $l_j^{G(c+1)} = z + 1/F$ (since it has received slots for the next frame) [Observation **D1**]. As, $l_j^{S(c+1)} \geq l_{min}^{S(c+1)} > l_{min}^{G(c+1)} = z$, we have, $l_j^{S(c+1)} \geq z + 1/F = l_j^{G(c+1)}$ [Observation **E**].

To show a contradiction, let us modify the solution $S(c+1)$ by removing $\delta + 1$ slots to obtain a solution $S(c-\delta)$ for $c-\delta$ slots as follows. For every client j in P, we remove any $sl(G(c+1), j) - sl(G(c+1-\delta), j)$ slots from its slot allocation, and in addition, we remove one more slot from one (arbitrarily chosen) client, say w, in $P2$. (The removed slots add up to $\delta + 1$ because δ slots were allocated by the greedy algorithm to obtain $G(c+1)$ from $G(c+1-\delta)$.) We now show that the minimum lead in $S(c-\delta)$ is higher than the minimum lead in $G(c-\delta)$, thus resulting in a contradiction (because $G(c-\delta)$ is optimal for $c-\delta$ slots). Let q be the client with the minimum lead $S(c-\delta)$. We consider four possible cases.

(1) q is not in P. In this case, no slots were removed from q to obtain $S(c-\delta)$ from $S(c+1)$, and so q had the minimum lead in $S(c+1)$ as well. Therefore, $l_q^{S(c-\delta)} = l_{min}^{S(c+1)} > l_{min}^{G(c+1)} = z > l_{min}^{G(c-\delta)}$ (from **A0** and **C**).

(2) q belongs to $P1$. Note that, since a process in $P1$ receives the minimum number of slots that is required for its lead to be $z + 1/F$ in $G(c+1)$ (from **C1** and **D1**), and $l_q^{S(c+1)} \geq l_{min}^{S(c+1)} \geq l_{min}^{G(c+1)} + 1/F = z + 1/F$ (from **A0**), q receives equal or more slots in $S(c+1)$ than in $G(c+1)$. Then, $sl(S(c-\delta), q) = sl(S(c+1), q) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) \geq sl(G(c+1), q) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) = sl(G(c+1-\delta), q)$. Therefore, $l_q^{S(c-\delta)} \geq l_q^{G(c+1-\delta)} = z > l_{min}^{G(c-\delta)} = z - 1/F$ (where the last inequality follows from **C**).

(3) q belongs to $P2$ but is distinct from w. Since $q \in P2$, $l_q^{S(c+1)} > l_q^{G(c+1)}$ (from **D**), and therefore $sl(S(c+1), q) > sl(G(c+1), q)$. Now, $sl(S(c-\delta), q) = sl(S(c+1), q) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) > sl(G(c+1), q)) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) = sl(G(c+1-\delta), q)$. Therefore, $l_q^{S(c-\delta)} \geq l_q^{G(c+1-\delta)} = z > l_{min}^{G(c-\delta)} = z - 1/F$ (where the last inequality follows from **C**).

(4) $q = w$. Since $q \in P2$, $l_q^{S(c+1)} > l_q^{G(c+1)}$ (from **D**), and therefore $sl(S(c+1), q) > sl(G(c+1), q)$. Now, $sl(S(c-\delta), q) = sl(S(c+1), q) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) - 1 > sl(G(c+1), q)) - (sl(G(c+1), q) - sl(G(c+1-\delta), q)) - 1 \geq sl(G(c+1-\delta), q)$. Therefore, $l_q^{S(c-\delta)} \geq l_q^{G(c+1-\delta)} = z > l_{min}^{G(c-\delta)} = z - 1/F$ (where the last inequality follows from **C**). □

BIBLIOGRAPHY

[1] Hypoexponential distribution. http://en.wikipedia.org/wiki/Hypoexponential_distribution.

[2] Relationships between mean and variance of normal and lognormal distributions. In *Lecture Notes, Dept. Of Civil Engg, MIT*. http://ocw.mit.edu/courses/civil-and-environmental-engineering/1-010-uncertainty-in-engineering-fall-2008/lecture-notes/notes_08.pdf.

[3] Abolhasan, Mehran, Wysocki, Tadeusz, and Dutkiewicz, Eryk. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks* (2004).

[4] Adhikari, Vijay, Guo, Yang, Hao, Fang, Varvello, Matteo, Hilt, Volker, Steiner, Moritz, and Zhang, Zhi-li. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *IEEE INFOCOM* (2012).

[5] Amari, Suprasad, and Misra, Ravindra. Closed-form expression for distribution of the sum of independent exponential random variables. *IEEE Transactions on Reliability 46*, 4 (1997), 519–522.

[6] Andrews, Jeffrey, Ghosh, Arunabha, and Muhamed, Rias. *Fundamentals of WiMAX: Understanding Broandband Wireless Networking*. Prentice Hall Communications Engineering and Emerging Technologies Series, 2007.

[7] Antonellis, Dimitrois, Mansy, Ahmed, Psounis, Konstantinos, and Ammar, Mostafa. Towards distributed classication for mobile ad hoc networks. In *In Proceedings of the 4th Annual International Conference on Wireless Internet* (2008).

[8] Auwera, Geert, David, Prasanth, and Reisslein, Martin. Traffic and quality characterization of single-layer video streams encoded with H.264/AVC advanced video coding standard and scalable video coding extension. *IEEE Transactions on Broadcasting 54*, 3 (2008).

[9] Baccelli, Francois, Blaszczyszyn, Barlomiej, and Muhlethaler, Paul. On the performance of time-space opportunistic routing in multihop mobile ad hoc networks. In *WiOPT* (2008).

[10] Balasubramanian, Aruna, Levine, Brian, and Venkataramani, Arun. Dtn routing as a resource allocation problem. In *ACM SIGCOMM* (2007).

[11] Balázs, Márton. Sum of independent exponential random variables with different parameters. http://www.math.bme.hu/~balazs/sumexp.html.

[12] Bhatia, Randeep, and Kodialam, Murali. On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control. In *IEEE INFOCOM* (2004).

[13] Biswas, Sanjit, and Morris, Robert. EXOR: Opportunistic routing for wireless networks. In *ACM SIGCOMM* (2005).

[14] Bovier, Anoton. Extreme values of random processes. In *Lecture Notes, Institute for Applied Mathematics, Bonn Germany.* http://wt.iam.uni-bonn.de/fileadmin/WT/Inhalt/people/Anton_Bovier/lecture-notes/extreme.pdf.

[15] Cacciapuoti, Angela, Caleffi, Marcello, and Paura, Luigi. Design, implementation and evaluation of an effecient opportunistic retransmission protocol. In *ICUMT* (2009).

[16] Camp, Joseph, Robinson, Joshua, Steger, Christopher, and Knightly, Edward. Measurement driven deployment of a two-tier urban mesh access network. In *MobiSys* (2006).

[17] Chachulski, Szymon. Trading structure for randomness in wireless opportunistic routing. Master's thesis, Massachusetts Institute of Technology, 2005.

[18] Chachulski, Szymon, Jennings, Michael, Katti, Sachin, and Katabi, Dina. Trading structure for randomness in wireless opportunistic routing. In *ACM SIGCOMM* (2007).

[19] Chetlur, Malolan, Dutta, Partha, Devi, Umamaheswari, Gupta, Parul, Chen, L, Zhu, Zhenbo, Kalyanaraman, Shivkumar, and Lin, Yonghua. A software wimax medium access control layer using massively multithreaded processors. *IBM Journal of Research and Development 54* (2010).

[20] Cisco. Visual Networking Index: Global mobile data traffic forecast update, 2009-2014.

[21] Clausen, Thomas, Jacquet, Philippe, and Viennot, Laurent. Comparative study of routing protocols for mobile ad-hoc networks. In *MedHoc* (2002).

[22] Clausen, Thomas, Jacquet, Philippe, and Viennot, Laurent. Analyzing control traffic overhead versus mobility and data traffic activity in mobile ad-hoc network protocols. *Wireless Networks 10*, 4 (2004).

[23] Cormen, Thomas, Leiserson, Charles, Rivest, Ronald, and Stein, Clifford. *Introduction to Algorithms, Second Edition.* The MIT Press and McGraw-Hill Book Company, 2001.

[24] Cover, Thomas, and Thomas, Joy. *Elements of Information Theory.* A John Wiley and Sons, INC., Publication, 2006.

[25] Demmer, Micheal, and Fall, Kevin. Dtslr: Delay tolerant routing for developing regions. In *ACM NSDR* (2007).

[26] Derin, Haluk. Probability and random processes for engineers. In *Lecture Notes, Dept. Of ECE, Univ. of Massachusetts, Amherst.*

[27] Dickey, David, and Fuller, Wayne. Distribution of the estimators for autoregressive time series with a unit root. *Journal of American Statistical Association 74*, 366 (1979).

[28] Ding, Zhiguo, Leung, Kin, Goeckel, Dennis, and Towsley, Don. A relay assisted cooperative transmission protocol for wireless multiple access systems. *IEEE Transactions on. Communications* (2008).

[29] Dobrian, Florin, Awan, Asad, Joseph, Dilip, Ganjam, Aditya, Zhan, Jibin, Sekar, Vyas, Stoica, Ion, and Zhang, Hui. Understanding the impact of video quality on user engagement. In *ACM SIGCOMM* (2011).

[30] Doucet, Arnaud, and Johansen, Adam. A tutorial on particle filtering and smoothing: Fifteen years later. In *Department of Statistics, University of British Columbia.*

[31] Elliot, E.O. Estimates of error rates for codes on burst-noise channels. *Bell System Technical Journal 42* (1963).

[32] Favaro, Stefano, and Walker, Stephen. On the distribution of sums of independent exponential random variables via Wilk's integral representation. *Acta Applicandae Mathematicae 109* (2010), 1035–1042.

[33] Fenton, Lawrence. The sum of log-normal probability distributions in scatter transmission systems. *IEEE Transactions on Communications Systems 8*, 1 (1960).

[34] Fiore, Marco hand Giannoulis, Anastasios, and Knightly, Edward. Characterization of coverage and signal propagation in the tfa network through wardriving. In *http://www.tfa.rice.edu/measurements/* (2006).

[35] Garrett, Mark, and Willinger, Walter. Analysis, modeling and generation of self-similar VBR video traffic. In *ACM SIGCOMM* (1994).

[36] Gilbert, E.N. Capacity of a burst-noise channel. *Bell System Technical Journal 39* (1960).

[37] Grossglauser, Matthias, Keshav, Srinivasan, and Tse, David. RCBR: a simple and efficient service for multiple time-scale traffic. *IEEE/ACM Transactions on Networking 5*, 6 (1997).

[38] Gudmundson, Mikael. Correlation model for shadow fading in mobile radio systems. *Electronic Letters 27*, 23 (1991).

[39] Gupta, Piyush, and Kumar, P. R. The capacity of wireless networks. *IEEE Transactions on Information Theory 26*, 02 (2000).

[40] Haerri, Jerome, Felali, Fethi, and Bonnet, Christian. Performance comparison of aodv and olsr in vanets urban environments under realistic mobility patterns. In *Med-Hoc-Net* (2006).

[41] Heimlicher, Simon, and Salamatian, Kavé. Globs in the Primordial Soup: The Emergence of Connected Crowds in Mobile Wireless Networks. In *ACM MobiHoc* (2010).

[42] Heo, J, and Salas, J. Estimation of quantiles and confidence intervals for the log-gumbel distribution. *Stochastic Hydrology and Hydraulics 10*, 3 (1996).

[43] Hrudey, Will, and Trajković, Ljiljana. Streaming video content over IEEE 802.16/WiMAX broadband access. In *OPNETWORK* (2008).

[44] Hsu, Cheng-Hsin, and Hefeeda, Mohamed. On statistical multiplexing of variable bit rate video streams in mobile systems. In *ACM Multimedia* (2009).

[45] Huang, Jie, Krasic, Charles, Walpole, Jonathan, and Feng, Wu-chi. Adaptive live video streaming by priority drop. In *AVSS* (2003).

[46] Jacquet, Philippe, Muhlethaler, Pual, Clausen, Thomas, Laouiti, Anis, Qayyum, Amir, and Viennot, Laurent. Optimized link state routing protocol for ad hoc networks. In *Hipercom Project, INRIA*.

[47] Jalden, Niklas, Zetterberg, Per, Ottersten, Bjorn, Hong, Aihua, and Thoma, Reiner. Correlation properties of large scale fading based on indoor measurements. In *IEEE PIMRC* (2007).

[48] Jiang, Tao, Sidiropoulos, Nicholas, and Giannakis, Georgios. Kalman filtering for power estimation in mobile communications. *IEEE Transactions on Wireless Communications 2*, 1 (2003).

[49] Krishnan, S. Shunmuga, and Sitaraman, Ramesh. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *IMC* (2012).

[50] Lam, Simon, Chow, Simon, and Yau, David. An algorithm for lossless smoothing of MPEG video. *ACM SIGCOMM Computer Communication Review 24*, 4 (1994).

[51] Laneman, J. Nicholas. Cooperative diversity: Models, algorithms and architectures. In *Cooperation in Wireless Networks: Principles and Applications*. Springer, 2006.

[52] Liang, Guanfeng, and Liang, Ben. Balancing interruption frequency and buffering penalties in VBR video streaming. In *IEEE INFOCOM* (2007).

[53] Liang, Guanfeng, and Liang, Ben. Effect of delay and buffering on jitter-free streaming over random VBR channels. *IEEE Transactions on Multimedia 10*, 6 (2008).

[54] Liu, Benyuan, Liu, Zhen, and Towsley, Don. On the capacity of hybrid wireless networks. In *IEEE INFOCOM* (2003).

[55] Liu, Haitao, Zhang, Baoxian, Mouftah, Hussein, Shen, Xiaojun, and Ma, Jian. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *IEEE Communications Magazine* (December 2009), 103–109.

[56] Liu, Xiangheng, and Goldsmith, Andrea. Optimal power allocation over fading channels with stringent delay constraints. In *ICC* (2002).

[57] Londono, Jorge, and Bestavros, Azer. A two-tiered on-line server- side bandwidth reservation framework for the real-time delivery of multiple video streams. *BUCS-TR-2008-012, Boston University* (2008).

[58] Lu, Mei-Hsuan, Steenkiste, Peter, and Chen, Tsuhan. Video transmission over wireless multihop networks using opportunistic routing. In *Packet Video Workshop* (2007).

[59] Lu, Mei-Hsuan, Steenkiste, Peter, and Chen, Tsuhan. A theoretical model for opportunistic routing in ad hoc networks. In *ACM MobiCom* (2009).

[60] Manfredi, Victoria, Crovella, Mark, and Kurose, Jim. Understanding stateful vs stateless communication strategies for ad hoc networks. In *ACM MobiCom* (2011).

[61] Mazo, James. Quantization noise and data transmission. *The Bell Labs Technical Journal* (1968).

[62] Oestges, Claude, Czink, Nicolai, Bandemer, Bernd, and Castigoline, Paolo. Experimental characterization and modeling of outdoor-to-outdoor and indoor-to-indoor distributed channels. *IEEE Transactions On Vehicular Technology 59* (2010).

[63] Ott, Teunis, Lakshman, T. V., and Tabatabai, Ali. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *IEEE INFOCOM* (1992).

[64] Ozgur, Ayfer, Leveque, Oliver, and Tse, David. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Transactions on Information Theory 53*, 10 (Oct. 2007), 3549–3572.

[65] Perkins, Charles, and Royer, Elizabeth. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Moble Computing Systems and Applications* (1999).

[66] Philips, Peter, and Perron, Pierre. Testing for unit root in time series regression. *Biometrika 75*, 2 (1988), 335–346.

[67] Radha, Hayder, van der Schaar, Mihaela, and Chen, Yingwei. The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Transactions on Multimedia 3*, 1 (2001).

[68] Rahul, Hariharan, Hassanieh, Haitham, and Katabi, Dina. Sourcesync: A distributed wireless architecture for exploiting sender diversity. In *ACM SIGCOMM* (2010).

[69] Ramanathan, Ram, Basu, Prithwish, and Krishnan, Rajesh. Towards a formalism for routing in challenged networks. In *ACM CHANTS* (2007).

[70] Rappaport, Theodore. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[71] Reibman, Amy, and Berger, Arthur. Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking 3*, 3 (1995).

[72] Rozner, Eric, Sheshadri, Jayesh, Mehta, Yogita, and Qiu, Lili. SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing 8*, 12 (December 2009), 1622–1635.

[73] Sadeghi, Parastoo, Kennedy, Rodney A., Rapajic, Predrag, and Shams, Ramtin. Finite-state markov modeling for fading channels- a survey of principles and applications. *IEEE Signal Processing Magazine 25*, 5 (2008).

[74] Said, Said, and Dickey, David. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika 71*, 3 (1984), 599–607.

[75] Salehi, James, Zhang, Zhi-Li, Kurose, Jim, and Towsley, Don. Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking 6*, 4 (1998).

[76] Sason, Igal. Entropy bounds for discrete random variables via coupling. `http://webee.technion.ac.il/people/sason/Presentation_entropy_bounds_isit2013.pdf`.

[77] Scaglione, Anna, Goeckel, Dennis, and Laneman, Nicholas. Cooperative communications in mobile ad-hoc networks: Rethinking the link abstraction. *IEEE Signal Processing Magazine 23*, 5 (Sept. 2006), 18–29.

[78] Schwarz, Heiko, Marpe, Detlev, and Wiegand, Thomas. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology 17*, 9 (2007).

[79] Seeling, Patrick, Reisslein, Martin, and Kulapala, Beshan. Network performance evaluation with frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys and Tutorials 6*, 3 (2004). CIF Video traces availalble at http://trace.eas.asu.edu/mpeg4/index.html and QCIF Video traces available at http://trace.eas.asu.edu/cgi-bin/main.cgi.

[80] Sen, Subhabrata, Dey, Jayanta, Kurose, Jim, Stankovic, John, and Towsley, Don. Streaming CBR transmission of VBR stored video. In *SPIE Symposium on Voice Video and Data Communications* (1997).

[81] Sharma, Rajesh, and Wallace, Jon. Indoor shadowing correlation measurements for cognitive radio systems. In *IEEE APSURSI* (2009).

[82] Shehada, Mohammed, Thakolsri, Srisakul, Despotovic, Zoran, and Kellerer, Wolfgang. Qoe-based cross-layer optimization for video delivery in long term evolution mobile networks. In *WPMC* (2011).

[83] Shroff, Ness, and Schwartz, Mischa. Video modeling within networks using deterministic smoothing at the source. In *IEEE INFOCOM* (1994).

[84] Sousa, Bruno, Pentikousis, Kostas, and Curado, Marilia. Experimental evaluation of multimedia services in WiMAX. In *MobiMedia* (2008).

[85] Spyropoulos, Thrasyvoulos, Psounis, Konstantinos, and Raghavendra, Cauligi. Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Transactions on Networking* (2008).

[86] Stern, Helman, and Hadar, Ofer. Optimal video stream multiplexing through linear programming. In *IEEE International Symposium on Information Technology* (2002).

[87] Su, Guan-Ming, and Wu, Min. Efficient bandwidth resource allocation for low-delay multiuser video streaming. *IEEE Transactions on Circuits and Systems for Video Technology 15*, 9 (2005).

[88] Thakolsri, Srisakul, Khan, Shoaib, Steinbach, Eckehard, and Kellerer, Wolfgang. Qoe-driven cross-layer optimization for high speed downlink packet access. *Journal of Communications 4*, 9 (2009).

[89] Tie, Xiaozheng, Venkataramani, Arun, and Balasubramanian, Aruna. R3: Robust replication routing in wireless networks with diverse connectivity characteristics. In *ACM MobiCom* (2011).

[90] Tse, David, and Vishwanath, Pramod. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[91] Vahdat, Amin, and Becker, David. Epidemic routing for partially connected ad hoc networks. *Technical Report, Duke University* (2000).

[92] Vishwanath, Arun, Dutta, Partha, Chetlur, Malolan, Gupta, Parul, Kalyanaraman, Shivkumar, and Ghosh, Amitabha. Perspectives on quality of experience for video streaming over WiMAX. *Mobile Computing and Communications Review 13*, 4 (2009).

[93] Wang, Di, and Abouzeid, Alhussein. Link state routing overhead in mobile ad hoc networks: A rate-distortion. In *IEEE INFOCOM* (2008).

[94] Wang, Hong Shen, and Moayeri, Nadar. Finite-state markov channel- a useful model for radio communication channels. *IEEE Transactions On Vehicular Technology 44*, 1 (1995).

[95] Wei, William. *Time Series Analysis: Univariate and Multivariate Methods.* Pearson Addison Wesley, 2006.

[96] Wong, Daniel, and Cox, Donald. Estimating local mean signal power level in a rayleigh fading environment. *IEEE Transactions on Vehicular Technology 48*, 3 (May 1999), 956–959.

[97] Wu, Huaming, and Abouzeid, Alhussein. Cluster-based routing overhead in networks with unreliable nodes. In *IEEE WCNC* (2004).

[98] Wui, Jianming, Affes, Sofiene, and Mermelstein, Paul. Forward-link soft-handoff in cdma with multiple-antenna selection and fast joint power control. *IEEE Transactions on Wireless Communications 2*, 3 (2003).

[99] Zemlianov, Alexander, and De Venciana, Gustavo. Capacity of an ad hoc wireless network with infrastructure support. *IEEE Journal on Selected Areas in Communications 3*, 25 (2005).

[100] Zhang, Ellen, Neglia, Giovanni, Kurose, Jim, and Towsley, Don. Performance modeling of epidemic routing. *Computer Networks* (2007).

[101] Zhang, Yu, Zhang, Jianhua, Dong, Di, Nie, Xin, Liu, Guanngyi, and Zhang, Ping. A novel spatial autocorrelation model of shadow fading in urban macro environments. In *IEEE GLOBECOM* (2008).

[102] Zhang, Zhensheng. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials* (2006).

[103] Zhang, Zhi-Li, Kurose, Jim, Salehi, James, and Towsley, Don. Smoothing, statistical multiplexing, and call admission control for stored video. *IEEE Journal on Selected Areas in Communications 15*, 6 (1997).

[104] Zhou, Nianjun, and Abouzeid, Alhussein. Routing in ad hoc networks: a theoretical framework with practical implications. In *IEEE INFOCOM* (2005).

[105] Zhou, Nianjun, Wu, Huaming, and Abouzeid, Alhussein. The impact of traffic patterns on the overhead of reactive routing protocols. *IEEE Transactions on Selected Areas of Communications 23*, 3 (2005).

[106] Zorzi, Michele, and Rao, Ramesh. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing 2*, 4 (October 2003), 349–365.

[107] Zorzi, Michele, Rao, Ramesh, and Milstein, Laurence. On the accuracy of a first-order markov model for data block transmission on fading channels. In *IEEE ICUPC* (1995).

[108] Zorzi, Michele, Rao, Ramesh, and Milstein, Laurence. A markov model for block errors on fading channels. In *PIMRC* (1996).

[109] Zorzi, Michele, Rao, Ramesh, and Milstein, Laurence. Arq error control for fading mobile radio channels. *IEEE Transactions on Vehicular Technology* (1997).