Effects of Diversity Routing on TCP Performance in Networks with Stochastic Channels

by

Yinuo (Mia) Qian

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering

at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2010
[June 2010]

Author .....
Department of Electrical Engineering and Computer Science
May 21, 2010

Certified by .....
Vincent W.S. Chan
Joan and Irwin M. Jacobs Professor of Electrical Engineering and Computer Science

Certified by .....
John M. Chapin
Visiting Scientist at Communication and Network Group, RLE

Accepted by .....
Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Effects of Diversity Routing on TCP Performance in Networks with Stochastic Channels

by

Yinuo (Mia) Qian

Submitted to the
Department of Electrical Engineering and Computer Science
May 21, 2010
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering

Abstract

Transmission Control Protocol (TCP) is a widely used transport protocol in today's multimedia applications. TCP was originally designed for wired networks, and its performance is highly degraded in networks with stochastically variable links, such as satellite and wireless networks. One significant class of degradation is caused by link dropouts, which may occur due to multipath fading effects in wireless networks, or due to signal attenuation by changing weather conditions and turbulence in satellite channels.

In this thesis, we study the effects of a diversity routing strategy on TCP performance in networks with stochastic links. This strategy allows each data sender to send copies of data packets along multiple paths in order to reduce the probability of packet losses. We define TCP efficiency to be the ratio of a data sender's throughput to the network capacity used by this sender. We explore the optimization of TCP efficiency by varying the number of paths along which data packets are sent,denoted as n. We find the optimal number of diversity routing paths n^* that maximizes the TCP efficiency, and we observe that the value of n^* depends on the variability of the stochastic links.

Thesis Supervisor: Vincent W.S. Chan
Title: Joan and Irwin M. Jacobs Professor of Electrical Engineering and Computer Science

Thesis Supervisor: John M. Chapin
Title: Visiting Scientist at Communication and Network Group, RLE

Acknowledgments

It is a great pleasure to thank the many people who made this thesis possible.

First and foremost, I would like to thank my thesis advisors Prof. Vincent Chan and Dr. John Chapin for their guidance and support. It is difficult to overstate my gratitude to Prof. Chan. With his inspiration, his great efforts to explain things clearly and simply, his patience to lead me through my research work, he helped to make the 2009-2010 school year the most fruitful year for me. Throughout my thesis-writing period, he provided encouragement, countless advice, good company and invaluable suggestions. I would have been lost without him. As my co-advisor, Dr. Chapin also gave me great support. In particular, he taught me how to present technical contents clearly and efficiently, put in many hours to help me edit my thesis, and enriched my knowledge of communication networks.

I wish to thank the many friends who have helped me understand the materials and accompany me throughout the thesis-writing period. In particular, Kang Gao and Lei Zhang provided significant insights to the models developed in this thesis. My friends Linjiang Cai and Manjae Kwon, gave me great support in the revision period of my thesis.

I wish to thank all the members in the Claude E. Shannon Communication and Network Group, Etty Lee, Andrew Puryear, Rui Li and Matthew Carey. They made my life as a graduate student so much more enjoyable. I would like to give special thanks to our secretary, Donna Beaudry, for coordinating all the meeting between me and my thesis advisor, and for giving me support when I was sick.

Last but not least, I want to thank my family for their love and support throughout my life. In particular,my mom, Yi Shen, bore me, raised me, supported me, taught me and loved me. My dear late grandfather, Jie Shen, has supported me and loved me since I was a child. To them I dedicate this thesis.

Contents

List of Figures

Chapter 1

Introduction

As communication network architectures evolve, the next generation of networks will most likely be heterogeneous with the inclusion of wired, wireless and satellite technologies [11]. In Figure 1-1, we give an example of various interconnected communication networks. In this example, the communication paths between typical network applications (e.g. personal computers and cellular phones) may include different types of network links.

The inclusion of different types of network channels introduces new challenges to future network architectures. One of the largest challenges is making the traditional network protocols adapt well to the different types of sub-networks in a heterogeneous network. Many properties of heterogeneous networks make it difficult to assure that one set of protocols work well across all of the interconnected sub-networks. As an important example, Transmission Control Protocol (TCP) experiences performance degradation in heterogeneous networks.

TCP is one of the core protocols in the IP Protocol Suite for traditional wired networks. It is believed to be the most widely used transport protocol in today's multimedia applications [11]. TCP has been designed for wired networks to efficiently manage the network flow. However, TCP is known to suffer from performance degradation in wireless network environments. Simulation results in [9] and [11] demonstrate this performance degradation in wireless networks. Similar degradation exists in satellite network environments. For example, [10] experimentally verified

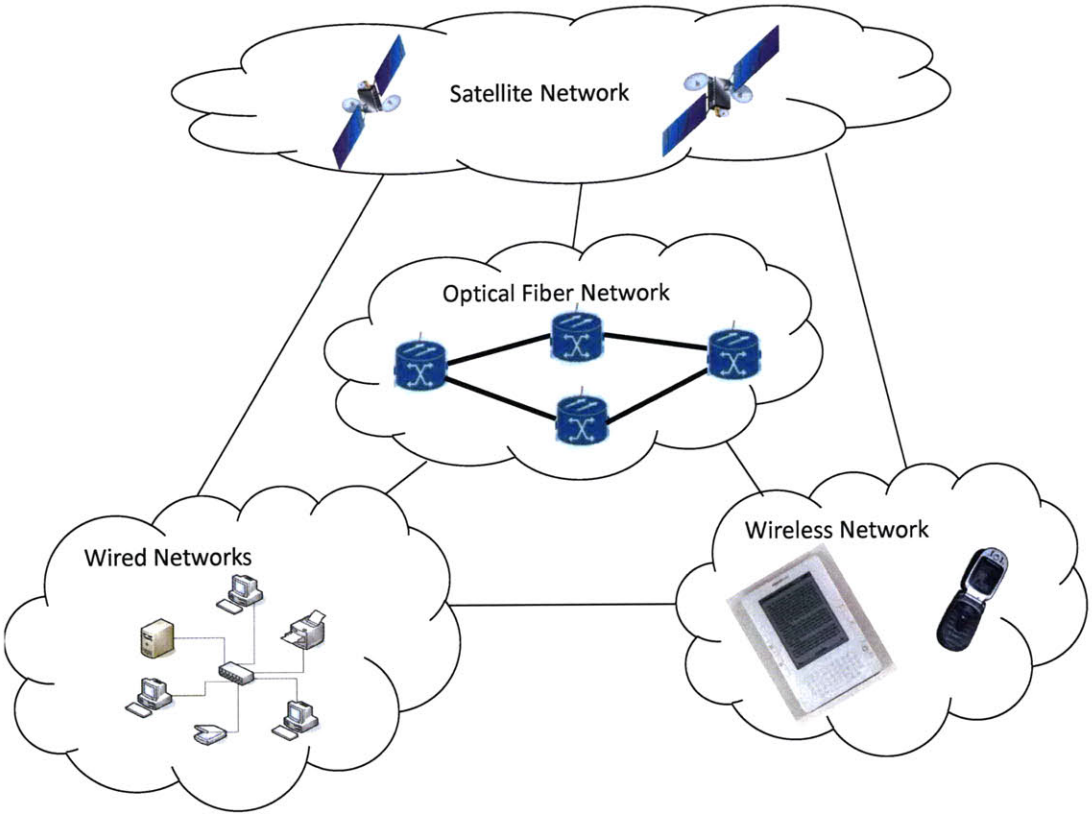

Figure 1-1: Schematic illustration of different components in a heterogeneous network. This is a combination of wired network, wireless network, optical network and satellite network. Typical applications of wired and wireless networks are shown in the figure. We see that the applications of the heterogeneous networks have the option to communicate through more than one type of networks when properly set up. For example, an application in the wireless network, such as a cellular phone, may have the option to receive data transmitted by the satellite network.

that TCP throughput is sharply degraded in GEO satellite networks. More specifically, an experiment conducted over a 1 Mbps actual GEO satellite link verifies that TCP is only able to utilize small portion of the available bandwidth [10].

The reasons for TCP throughput degradation in wireless and satellite networks are well understood. For satellite networks, the degradation is mainly due to the effects of long propagation delays and link dropouts [1]. For wireless networks, the degradation is due to packet dropouts caused by channel fading effects. We refer to these two types of links as *stochastic links* in the rest of the paper, in contrast to copper wired and fiber optical links which have very stable link capacity over time.

To improve TCP performance in networks with stochastic links, one category of solutions suggested by previous research is modifying TCP to accommodate the needs of different heterogeneous sub-networks [1, 11]. For example, in satellite networks, it takes TCP a fairly long time to reach a high data sending rate if the traditional version of TCP is deployed. To overcome this problem, a modified version of TCP, TCP-Peach, incorporates two new mechanisms: sudden start and rapid recovery. However, this category of solutions sometimes requires modification not only to TCP itself, but also to applications [11]. In addition, because these TCP modifications are specific to one type of network, they cannot be easily generalized.

Another category of research proposes more proactive solutions, in the sense that the data sender collects information about the network dynamics, and adjusts the data sending rate based on this information [2, 4, 7, 11]. As an example of this, TCP-Westwood is one variant of TCP in which the sender dynamically estimates the available network bandwidth to adjust the sending rate [7]. However, in networks with long RTT and high data rate, such as GEO satellite networks, TCP-Westwood would not perform as well as it does in networks with shorter RTT and lower data rate, such as LEO satellite networks. Another variant of TCP, TCP-Veno, monitors the network congestion level and adjusts its data sending rate less drastically if it makes an educated guess that a packet loss is not due to congestion [4]. TCP-Veno works poorly if consecutive packet losses occur.

In this thesis, we propose a diversity routing strategy in which the sender sends

copies of data packets along multiple paths. By increasing the number of paths along which the packets are sent, we are able to reduce packet losses caused by link dropouts and increase TCP throughput. We define TCP efficiency to be the ratio of a data sender's throughput to the network capacity used by this sender, and we explore the optimization of TCP efficiency. Moreover, the model we develop to calculate TCP throughput over stochastic links is much simpler than the models previously used. This simple model enables us to analytically study TCP performance over complicated network configurations.

The rest of the thesis is structured as follows. In Chapter 2, we present a two-state Markov model that describes the stochastic properties of satellite and wireless links. In Chapter 3, we study TCP throughput and efficiency in a simple network model where the sender has the option to use diversity routing over multiple single-link paths. We start our analysis with a slightly complicated model and later on reduce it to a much simpler, but fairly accurate model. In Chapter 4, we study TCP performance over a more complicated network configuration, where there are multiple paths each of which consists of multiple stochastic channels. In Chapter 5, we draw conclusions based on our study, and discuss further work to be done.

Chapter 2

Preliminaries

In this chapter, we introduce the physical properties and architectural design of networks with stochastic links, i.e. wireless and satellite networks. The goal is to choose some basic models that can be extended to study more complicated network configurations. We use Markov models to represent stochastic links. We also give a brief description of TCP's congestion control algorithm, and present the upper and lower bounds of TCP performance. We will see that our models can be extended to represent complicated network configurations, and that they capture the important properties of satellite and wireless channels. In the rest of the text, we use the words *channel* and *link* interchangeably.

2.1 Channel model for wireless networks

Wireless channels have varying capacity, ranging from kilobits per second to gigabits per second. Because wireless channels use the open air as the transmission medium, they are subject to many factors affecting link quality, such as weather conditions or mobility of wireless [11]. As a result, wireless channels have much higher bit error rates than traditional wired links.

Multipath interference causes signal attenuation in wireless channels, which is called fading. Fading can lead to temporary link dropouts that result in packet

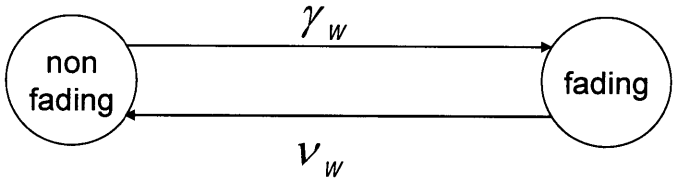

Figure 2-1: On-off Markov model for fading wireless channel.

losses. In addition, dynamically changing routes due to wireless devices' mobility can cause packet loss and a high latency variation over time. It turns out that fading is the major factor that causes packet losses in wireless links [7]. Thus in our research we will focus on wireless channels' fading effects.

Markov processes have been used to model fading channels in previous research. In [14], Milstein et al. showed that using a first-order Markov model to approximate fading wireless channels gives satisfactory results. In [13], Zhang and Kassam developed a methodology to partition the received signal-to-noise ratio (SNR) of fading channels into a finite number of Markovian states. The SNR partition depends on the fading speed of the channel. In [7], Gerla et al used a two-state Markov model to represent the behavior of lossy wireless links, and study TCP-Westwood's throughput over such links.

In our study, we model wireless channels by a Markov process with two states: fading and non-fading, as shown in Figure 2-1. This simple channel model helps us qualitatively understand the effects of fading in a tractable fashion.

The lengths of time spent in the two states are denoted by random variables $X_{w_{nf}}$ and X_{w_f} with the following distribution functions

$$\begin{aligned} F_{X_{w_{nf}}}(x) &= 1 - \exp(-\gamma_w x) \\ F_{X_{w_f}}(x) &= 1 - \exp(-\nu_w x) \end{aligned} \tag{2.1}$$

The determination of the values of γ_w and ν_w should be done by measurements in a wide range of channel environments. However, a rough estimation of γ_w and ν_w will provide significant insights on the quality of the wireless links. More specifically,

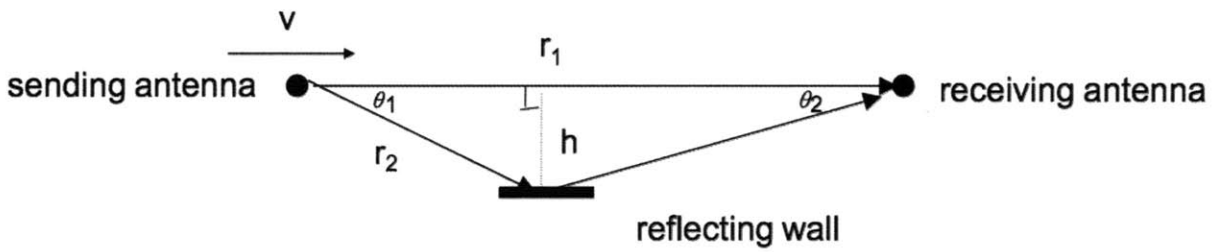

Figure 2-2: Illustration of a direct path and a reflective path for moving wireless signal sending antenna. The receiving antenna receives a combine waveform of the two paths, which leads to a multi-path fading effect.

the order of magnitude of γ_w and ν_w shows how frequently wireless links transition between the two states, and how long each transition lasts.

In Figure 2-2, we illustrate a scenario under which a moving wireless device experiences multi-path fading effects. In this scenario, the sending antenna is moving towards the receiving antenna at a constant velocity v, and is sending a signal at frequency f_0. Moreover, there is a perfect reflecting wall parallel to the line segment connecting the sender and the receiver, at a distance h from the line. Thus, there are two paths for the signal to arrive at the receiver: one direct path from the sender to the receiver, and one reflected path from the sender reflected by the wall to the receiver.

We denote the length of the direct path by r_1, and the length of the reflected path by r_2. From the geometry of this scenario,

$$r_1 = \frac{h}{\tan\theta_1} + \frac{h}{\tan\theta_2} - vt \tag{2.2}$$

and

$$r_2 = \frac{h}{\sin\theta_1} + \frac{h}{\sin\theta_2} - vt\cos\theta_1 \tag{2.3}$$

Because there are two paths for the signal to arrive at the receiver, the signals are combined and depending on the relative phase, signal cancellation may occur, which leads to fading effects. In Appendix A, we see that the combined waveform is dominated by a cosine term that depends on $\frac{vf_0}{c}$. The combined wave has a period of $T(f_0) = \frac{c}{f_0 v \cos^2 \frac{\theta_1}{2}}$. If we pick f_0 as 2 GHz, $\theta_1 = 10°$, $c = 3 \times 10^8$ mps and a fading signal threshold of $\frac{1}{2}$ (see Appendix A for details), we find the order of magnitude of ν_w should be in thousands of $\sec^{-1}$, and that of γ_w should be in several $\sec^{-1}$. The ratio of the two parameters should be around 10^2 to 10^3.

2.2 Channel model for satellite networks

Long propagation delays increase the latency of satellite channels. This results in long round trip times (RTT) for satellite channels. For example, in a GEO satellite, the round trip time is at least 480 milliseconds (ms) [10]. We will see the effects of longer RTT on TCP throughput in Chapter 3.

Satellite communication (Satcom) channels have variable capacity. Satcom channel capacity changes due to fading effects caused by turbulence and weather conditions. These time-varying fading effects are characteristic of satellite channels at high frequencies above 10 GHz. Satellite channels' fading effects are mainly caused by rain; while on clear days, scintillation effects are the major causes of fading [3].

In [3], the spectrum of the log amplitude of satellite channels' signal strength was approximated by a 1-pole autoregressive model. We will approximate the signal attenuation of satellite channels by a Markov process. A similar method was used in [7] to model a LEO satellite channel.

In our research, for simplicity and consistency with the wireless channel model, we still use a two-state Markov model for satellite channels, illustrated in Figure 2-3. The lengths of time spent in the two states are denoted by random variables X_{s_u} and X_{s_d} with the following distribution functions,

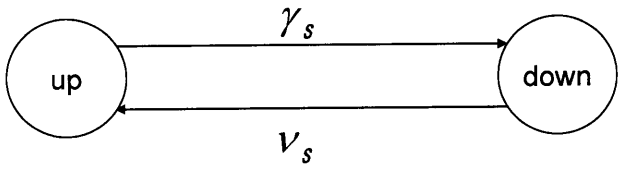

Figure 2-3: On-off Markov model for satellite channel

$$F_{X_{s_u}}(x) = 1 - \exp(-\gamma_s x)$$
$$F_{X_{s_d}}(x) = 1 - \exp(-\nu_s x) \tag{2.4}$$

[3] plots the sampled power spectral density (PSD) and its corresponding autoregressive model for satellite channels. In this plot, the 3 dB corner frequency is at around 0.11 Hz. Thus the coherence time for the Satcom link is around $\frac{1}{0.11\ \text{Hz}} = 9$ sec. This coherence time can be used as the average length that the Markov process remain in the *up* state on average. From the exponential distribution, expected *up* time is 1 over the rate γ_s,

$$E[\text{up time}] = \frac{1}{\gamma_s} = 9 \text{ sec} \tag{2.5}$$

We get $\gamma_s = 0.11$ sec^{-1}. The ratio between E[up time] and E[down time] can vary. We will make different assumptions on this ratio in later chapters.

2.3 Brief review on the Transmission Control Protocol (TCP)

In this section, we give a brief review of TCP's congestion control algorithm, and propose a model that captures its dynamics. Each TCP session is a peer process between a sender node and a receiver node. The two nodes collaborate to control the number of packets sent into the network. The sender starts by sending one packet per round trip time (RTT). Each packet sent has a packet sequence number (SN).

For every packet received in sequence, the receiver sends an acknowledgement packet (ACK) to the sender. For example, if the receiver receives packet $1, 2, 3, \ldots, n$,

it will send the ACKs for all of these packets. Upon receiving the ACKs, the sender learns that packets 1 to n were correctly delivered successfully. If the next packet received is not packet $n+1$, but some packet with a larger packet number, it must be the case that packet $n+1$ is lost or delayed in the network. Instead of sending an ACK with SN greater than $n+1$, the receiver will continue sending an ACK with SN equal to n, known as duplicate ACKs.

The sender maintains a value called the window size. It dynamically changes its window size in response to congested network situations to achieve a fairly good throughput. In addition, we define "maximum window size", W_m, to be the maximum number of packets that the sender has sent but has not received an ACK for. W_m is usually 128 packets or 256 packets in most deployed versions of TCP. There is also a standard modification to TCP called TCP window scaling [8] that allows the TCP window size to be increased to larger values. In our thesis, we assume that the maximum window size is set to be the maximum possible number of packets in flight, denoted by M.

The traditional version of TCP has two phases for the sender to increase its window size. There is a Slow Start phase, during which the sender increases its window size exponentially until the window size reaches a threshold (*ssthresh*). After this point, the sender operates in the Congestion Avoidance phase, in which the sender increases its window size linearly until it reaches the maximum window size. The sender detects a packet loss in two different ways and reduces its window size accordingly. First, if the sender receives 3 duplicate ACKs for a packet, it assumes that the packet is lost due to network congestion, and reduces its window size by half. In addition, the sender keeps a timer for each packet, and the timer has an expiration value called Retransmission Time Out (RTO). If RTO expires, the window size is reduced to one packet, and enters the Slow Start phase again.

For now, we assume that all packet losses are due to the variability of the stochastic links, not due to congestion. We also assume that TCP always closes its window and goes back to the Slow Start phase when it determines a packet loss. Moreover, as shown in Figure 2-4, if the sender uses only exponential window increasing, the

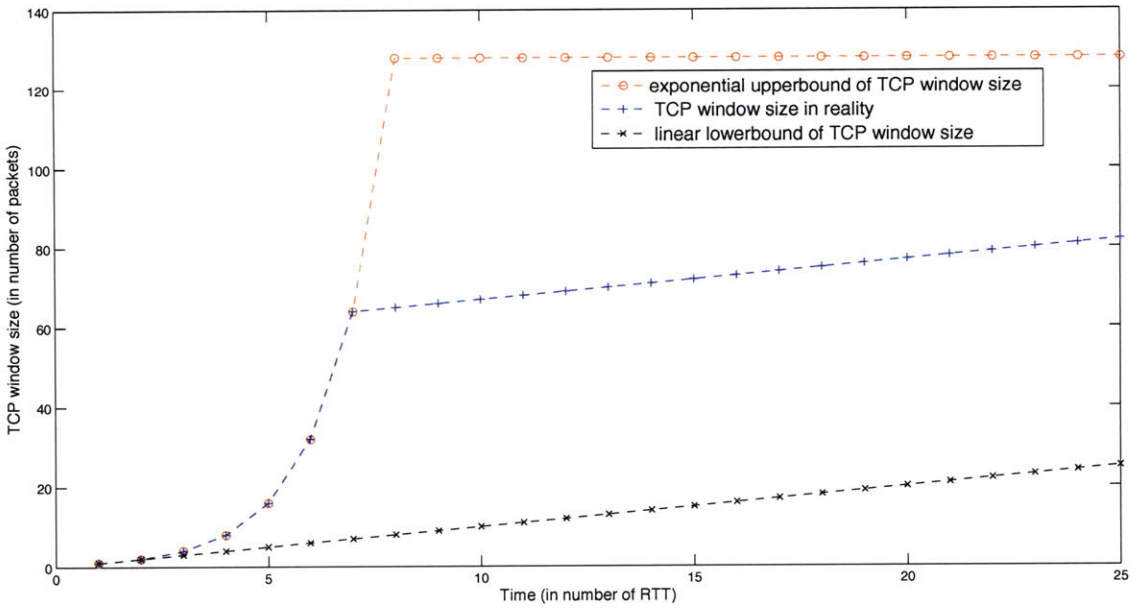

Figure 2-4: Illustration of different phases of TCP window increase. W_m=128 packets. We see that TCP window size increasing is upper-bounded by the exponential increasing pattern, and lower-bounded by the linear increasing pattern.

number of packets sent will set an upper bound on the number of packets sent in the two-phase pattern; if the sender uses only linear window increasing, the number of packets sent will set a lower bound on the two-phase pattern. In our research, we assume that TCP always operates in the Congestion Avoidance phase.

In the next two chapters, we will combine the channel models and the TCP model described in this chapter to analyze the performance of TCP in different network configurations.

Chapter 3

TCP performance analysis over a network model with parallel single-link paths

In this chapter, we study the performance of TCP over simple networks with stochastic links, using the models developed in Chapter 2. In this chapter's analysis, we quantify the level of degradation of TCP performance in networks with stochastic links, such as satellite networks and wireless networks. We also evaluate the improvement of TCP performance by applying the diversity routing strategy.

Before we dive into the quantitative analysis, we want to understand why the performance of TCP is highly degraded in satellite and wireless networks. In traditional wired networks, the sender manages the TCP window size to achieve congestion control for the network. A data sender reduces its window size when there is network congestion, i.e., when there are more packets released into the network than the routers can handle at a time. TCP was designed for traditional wired networks, and works well in managing the network flow.

The sender assumes every packet loss is due to congestion, which is a fair assumption in wired networks. However, in Satcom and wireless networks, the variability of the channels will also cause packet losses. TCP is unable to distinguish the causes of the packet losses, which is the major reason for its performance degradation. In this

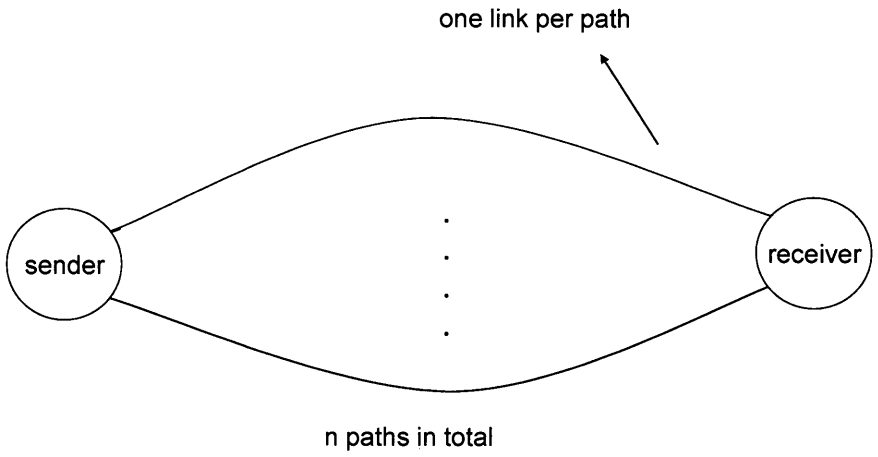

Figure 3-1: n identical single-link paths between sender and receiver

chapter, we want to solve the degradation problem by applying a diversity routing strategy, which instead of modifying TCP, allows the sender to send copies of data along multiple links to reduce the probability of packet losses.

In this chapter, we focus our analysis on a simple case, where there are several identical parallel paths between the sender and the receiver, and each path consists of only one link, as shown in Figure 3-1. All the parallel paths are independent from each other. Later on we will extend our analysis to more complicated network models.

The rest of the chapter is organized as follows. Section 3.1 gives a detailed description of the diversity routing strategy; Section 3.2 derives an exact expression for TCP throughput based our assumptions; Section 3.3 presents an approximation to the expression we derived in Section 3.3, and the approximation is much easier to calculate and use; Sections 3.4 and 3.5 analyze the performance of TCP and the effects of the diversity routing strategy in several network scenarios, some of which

correspond to real-life network systems.

3.1 Diversity routing strategy

Diversity routing is a routing strategy that allows the sender to send identical copies of data packets via multiple paths to its downstream network. As long as the receiver receives one of the copies of a certain packet, that is, whichever copy that first arrives at the receiver, it will send an acknowledgement with the packet number. The receiver will send copies of ACKs on the later copies of the same packets as well. As long as one of the copies of ACKs arrive at the sender, the sender knows that the data packet was correctly received.

To see why we need diversity routing in networks with stochastic links, let us first see how the data packets are routed in traditional wired networks. In wired networks, the routers each keep a routing table that reflects the network status. Usually, there are multiple paths that a packet can be routed. The routers will compute an optimal path using the routing table and a shortest path and/or minimum cost algorithm.

In networks with stochastic links, if we still follow this routing strategy, there exists a non-negligible possibility that as the packet is routed along the optimal path, one or more stochastic links on that optimal path are temporarily dropped out and the packet is lost. Thus, computing the optimal path based on a shortest path algorithm does not suffice to provide delivery with high reliability, leading to TCP window closing and severe reduction in throughput. If we use diversity routing, we may reduce the possibility of packet loss caused by link dropouts, thus make the connection between the sender and the receiver more reliable.

Now that we have introduced the rationale behind using diversity routing in networks with stochastic links, we want to understand how efficient this strategy is. If we define the TCP efficiency to be the ratio of TCP throughput to the network capacity used by sender. A question that follows immediately is how many paths should the sender use to send data. Are more paths always more efficient? Or is there an optimal

number of paths that is the most efficient? We see that in diversity routing, the more paths along which the sender routes the data packets, the less likely there will be a disconnection; on the other hand, the more paths the sender uses, the more network capacity is consumed to send the same data packet. If we use too many paths in diversity routing, it might be inefficient because we allocate too much network resources to send the same data. Thus, the answer should certainly not be the more paths, the better TCP efficiency achieved. There exists some optimal number of diversity routing paths that maximizes the TCP efficiency. In the sections to follow we will verify this, and determine what parameters this optimal number of paths depends on.

3.2 Steady-state TCP throughput by diversity routing

3.2.1 Channel models revisited and assumptions

In the network model illustrated in Figure 3-1, there are N identical parallel paths between the sender and the receiver, and each path consists of only one link. From Chapter 2, we see that each of the links can be modeled by a two-state Markov process as shown in Figure 3-2. The N links are all independent from each other. The links can be those of a wireless network, or those of a satellite network. Note that we are only looking at the case that all N paths are identical for now.

The sender sends n ($n \leq N$) copies of a certain data packet, and waits for the first ACK of this data packet. When $n = 1$, it means that the sender does not implement any diversity routing strategy.

In our analysis, time is measured in one of two units: round trip time (RTT) or seconds, under different contexts. When we focus on the sender's TCP window size control, we choose RTT as our time unit because in reality, the basic unit of time for the sender to take reaction in adjusting window size is one round trip time. On the other hand, when we measure the throughput of the sender, we use bits per second

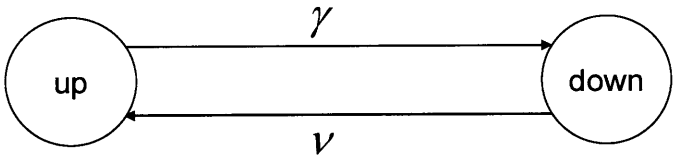

Figure 3-2: Two-state Markov process model for a stochastic channel

(bps) as the measurement unit, so that we can compare networks with different RTT lengths.

Each of the N links is modeled by a two-state Markov process. We denote the two states of the link as *up* and *down*, *up* when the link is nominal and *down* when it has dropouts. We use a set of identically and independently distributed (IID) random variables $\{X_u^i; i = 1, 2, ...N\}$ to denote the length of time that the ith link spends in its *up* state. Similarly, we use a set of IID random variables $\{X_d^i; i = 1, 2, ...N\}$ to denote the length of time that the ith link spends in its *down* state. Since $\{X_u^i\}$ are IID, we simply use X_u to denote each of them. Similarly, we use X_d to denote each of the $\{X_d^i\}$. Both X_u and X_d are exponentially distributed, with rates γ and ν, respectively. Figure 3-2 gives an illustration of the channel model.

For each of the N links, we have the following distribution functions,

$$\begin{aligned} F_{X_u}(x) &= 1 - \exp(-\gamma x) \\ F_{X_d}(x) &= 1 - \exp(-\nu x) \end{aligned} \tag{3.1}$$

The expectations of the length of time that a single link will spend in its *up* and *down* states are

$$\begin{aligned} E[X_u] &= \frac{1}{\gamma} \\ E[X_d] &= \frac{1}{\nu} \end{aligned} \tag{3.2}$$

Let π_u and π_d represent the steady state probabilities for a single link to be in its *up* and *down* states, respectively. The steady state probabilities can be expressed as

follows:

$$\pi_u = \frac{E[X_u]}{E[X_u] + E[X_d]} = \frac{\nu}{\gamma + \nu}$$
$$\pi_d = \frac{E[X_d]}{E[X_u] + E[X_d]} = \frac{\gamma}{\gamma + \nu} \tag{3.3}$$

In this thesis, we assume that all packet losses are caused by the variability of the stochastic links. We further assume that the packets sent while the sender and the receiver are connected can all be received correctly, except for the last window before the connection is broken. A portion of or the entire last TCP window before the connection is broken may be lost. If TCP deploys a selective-repeat strategy, the lost packets will be re-transmitted; if TCP deploys a go-back-N strategy, the entire last window will be re-transmitted.

We also assume that as soon as the sender senses a packet loss, it immediately reduces the window size to 1 packet per RTT starting from the next round trip time, until the next time that the sender and the receiver are connected. This is realistic since link dropouts usually results in multiple packet losses, and causes TCP window closing. For there to be a packet loss, the sender and the receiver must be disconnected, implying the n links must all be in the *down* state at the same time.

As we have mentioned in Section 2.3, there exists a maximum TCP window size W_m, which is the maximum number of packets that a sender can send during any round trip time. In our model, we assume that W_m is the maximum number of packets in flight in any round trip time, denoted by M.

In our model, each link has a capacity $R_{\max}$, the maximum rate at which the link can transmit data, in bits per second (bps). We assume that all the N links always operate at their maximum data rates as long as they are in the *up* state. We also take the round trip time, RTT, as a given parameter in our analysis for now. Moreover, the size of a single packet is denoted by K, in bits. Thus the maximum number of packets in flight, M, during any RTT, is given by

$$\begin{aligned} M &= \frac{RTT}{\text{time to transmit one packet}} \\ &= \frac{RTT \cdot R_{\max}}{K} \end{aligned} \tag{3.4}$$

3.2.2 TCP's window size control over stochastic links: a renewal process perspective

As long as one of the n links is still *up*, the packets can be delivered to the receiver. When a link goes from the *up* state to the *down* state, or vice versa, we say that there is a "link state change." We assume that in a time interval $(t,t+\delta)$ for a very small δ, the probability of one link state change is some rate η times the length of the interval δ, $\eta\delta$; in addition, the probability of two or more link state changes happening in the small interval δ is $o(\delta)$, where the term $o(\delta)$ is used to describe a function of δ that goes to 0 faster than δ as $\delta \to 0$, i.e., $\lim_{\delta\to 0} \frac{o(\delta)}{\delta} = 0$. Based on this assumption, if at time t there are k out of the n paths that are *up* $(k = 1, 2, ...n)$, then at time $t+\delta$ we can have at most $k+1$ links that are in the *up* state, and at most $n-k+1$ links that are in the *down* state.

Since only one link state change can happen during any small time interval δ, we know that at any time t and $t+\delta$, the number of links that are in the *up* state $N_{up}(t)$ and $N_{up}(t+\delta)$ can be different by at most 1, i.e., $|N_{up}(t) - N_{up}(t+\delta)| = 0$ or 1. Thus, it is natural to use a $(n+1)$-state Markov process to model the behavior of the status of the n paths between the sender and the receiver, as shown in Figure 3-3. We call the model in Figure 3-3 Markov Process 1 (MP1).

In MP1, State k $(k = 0, 1, 2, ...n)$ represents that there are k paths(links) in the *up* state; State 0 represents that all of the n paths are in the *down* state; in other words, the sender and the receiver are disconnected. The transition rates between the states are shown as in Figure 3-3.

When Markov process 1 is in State 0, there is no path between the sender and the destination that is *up*. We assume the sender still sends 1 packet per RTT, until the time epoch that Markov Process 1 transits to State 1 from State 0. The reason why the sender still sends 1 packet per RTT while all paths are *down* is that the sender needs to constantly check the status of the n paths. While all paths are *down*, this packet will be lost and the sender will not receive an ACK. Whenever the sender receives an ACK for this packet, it knows that at least one path becomes *up* and that

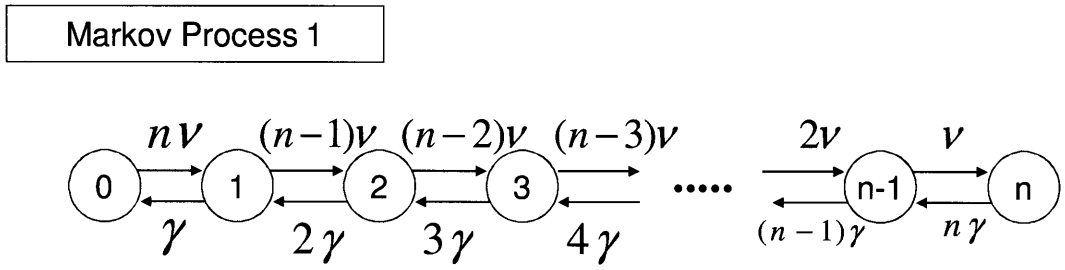

Figure 3-3: Markov Process 1: $n+1$ state Markov process for n identical paths in parallel. State i represents that i out of n links are in the *up* state at the same time.

it is ready to increase the window size again.

When MP1 is in States 1 to n, the sender and the receiver are connected, and the sender keeps on increasing the window size; when MP1 is in State 0, the sender and the receiver are disconnected. Thus, the connection between the sender and the receiver alternately goes on and off as MP transitions out of and into State 0. The alternating behavior has a repeating pattern, and we can view it by defining a renewal process.

We define the following event as a renewal: every time that MP1 is right about to make a transition out of State 0, the process renews itself. In Figure 3-4 we give an illustration of the TCP window size during each renewal of the process, assuming that TCP always increases window size linearly. The packets colored in red represent the packets that are not received.

We define a new random variable, T_u, to denote the time elapsed from the epoch when a renewal begins, until the epoch that all paths just go back to the *down* state. Alternatively, T_u is the time from the time epoch when MP1 enters its State 1, which is the same epoch as when MP1 is right about to transition out of State 0, until the time epoch when it goes back to State 0 for the first time. We define another random variable, T_d, to denote the time elapsed from the epoch that all n links just go to the *down* state until the time epoch that one of the n link is right about to go back to the *up* state. Alternatively, T_d is the time from the epoch that MP1 just enters its State 0 until the epoch that MP1 is about to go back to State 1 from State 0 for the first time. We see that $T = T_u + T_d$ is the length of the inter-renewal interval.

From the beginning of each renewal, TCP starts increasing its window size in every round trip time (RTT): for the first RTT after a renewal, the window size is 1 packet; for the second RTT after renewal, as long as Markov Process 1 is still in a non-zero state, TCP increases its window size to 2 packets. TCP keeps on increasing its window size until the next epoch that all n paths are *down*, i.e., until the next time that Markov Process 1 goes into State 0. This manner of TCP window size increasing is depicted in Figure 3-4. In the terminology of renewal processes, we may say that the number of successfully received TCP packets is a reward function for the

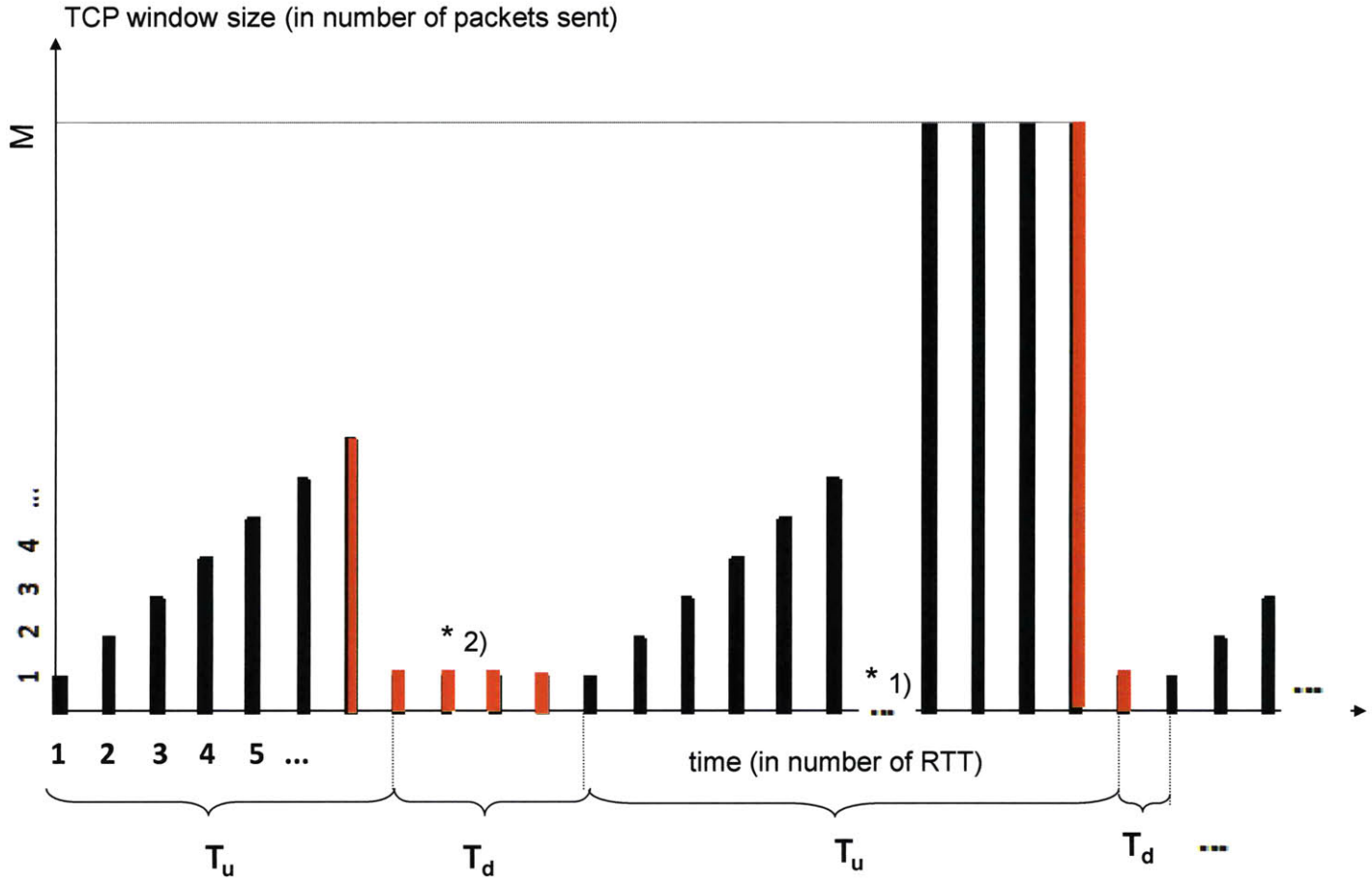

Figure 3-4: Linear TCP window size increasing modeled as renewal reward process, with each renewal defined as when one of the n *down* paths is right about to go *up*. From the beginning of each renewal, TCP starts linearly increasing its window size in every RTT. The last window before all links go *down* is assumed to be lost. This is a lower bound of the number of packets sent under a selective-repeat strategy.

renewal process defined in the previous paragraph. The renewal reward function is denoted by $R(t)$. We see that the packets colored in red in Figure 3-4 are not part of the reward function because they are never successfully delivered to the receiver.

Up until now, we have prepared enough to derive the TCP throughput. Throughput means the average rate of successful information delivery over a communication channel or network. Thus, we define the TCP throughput (in number of packets) to be

$$\lim_{t\to\infty} \frac{\int_{\tau=0}^{t} R(\tau)\, d\tau}{t} \text{ (in number of packets)} \tag{3.5}$$

Note that by saying $t \to \infty$ we mean that the process has been going on for a while, and it has reached steady state.

By the Strong Law of Large Numbers for Renewal Reward Process in [5, Chapter 3], we have

$$\lim_{t\to\infty} \frac{\int_{\tau=0}^{t} R(\tau)\, d\tau}{t} = \frac{E[R_i]}{E[T]}, \text{ with probability 1} \tag{3.6}$$

where $E[T]$ is the expected length of the inter-renewal interval, and $E[R_i]$ is the expected cumulative reward within one inter-renewal interval, which is the same for any positive integer i.

From Equation (3.6), we see that to find the steady-state throughput of TCP, we just need to find $E[R_i]$ and $E[T]$. In the context of our problem, $E[T]$ is the expected length of each inter-renewal interval. $E[R_i]$ is the expected number of successfully received TCP packets within one inter-renewal interval. Since $T = T_u + T_d$, we have

$$E[T] = E[T_u] + E[T_d] \tag{3.7}$$

In order to derive $E[R_i]$, we need to derive the cumulative distribution function of T_u, $F_{T_u}(t)$. We have

$$F_{T_u}(t) = \Pr(T_u \leq t) \tag{3.8}$$

This derivation will be done in Section 3.2.3.

We also need to derive the distribution of T_d. T_d is the length of time that Markov Process 1 spends in State 0; as long as one of the n parallel links goes back to *up*, T_d

will end. Thus, the process that one of the n links goes back to *up* is just a merged process of n independent processes, each of which follows an exponential distribution of rate ν. Thus, $F_{T_d}(t)$ has an exponential distribution with rate $n\nu$.

$$F_{T_d}(t) = 1 - \exp(-n\nu t) \tag{3.9}$$

and

$$E[T_d] = \frac{1}{n\nu} \tag{3.10}$$

Because the n independent links each has a *down* state steady state probability of π_d, the steady state probability for MP1 to be in State 0, π_0, is just the product of each link's steady state probability of being in the *down* state. Thus,

$$\begin{aligned} \pi_0 &= \pi_d^n \\ &= \frac{\gamma^n}{(\gamma+\nu)^n} \end{aligned} \tag{3.11}$$

We also know that the ratio of the expected length of T_u and T_d is the same as the ratio of the steady state probabilities that MP1 is in a non-zero state versus that MP1 is in State 0. Hence,

$$\begin{aligned} \frac{E[T_u]}{E[T_d]} &= \frac{\sum_{k=1}^{n} \pi_k}{\pi_0} \\ &= \frac{1-\pi_0}{\pi_0} \\ &= (\frac{\gamma+\nu}{\gamma})^n - 1 \end{aligned} \tag{3.12}$$

and $E[T_u]$ is

$$E[T_u] = \frac{1}{n\nu} \cdot [(\frac{\gamma+\nu}{\gamma})^n - 1] \tag{3.13}$$

As we can see, the values of $E[T_u]$ and $E[T_d]$ depend on the value of γ, ν and n. Keeping n fixed, $E[T_d] \propto \frac{1}{\nu}$. Keeping n and ν fixed, $E[T_u]$ increases as the value of γ increases. This makes sense because a small value of ν means that the links are fairly variable: the transition rate out of the *down* state is small, holding Markov

Process 1 in State 0 for a longer time. Similarly, if the ratio $\frac{\nu}{\gamma}$ is high, it means that the stochastic link is less variable, and we should expect that for the majority of the time, the sender and the receiver are connected, holding Markov Process 1 in one of its non-zero states.

In addition, the values of $E[T_u]$ and $E[T_d]$ also have a dependency on n. When n is large, keeping γ and ν fixed, we would expect $E[T_u]$ to be much larger than $E[T_d]$. This makes sense because as the number of paths along which the sender sends copies of data increases, we should expect the connection between the sender and receiver to be less likely to drop out, which leads to a much larger $E[T_u]$ than $E[T_d]$.

3.2.3 Derivation of the probability distribution function of T_u

As a reminder, we derive the probability distribution function of T_u because we want to find the expression for the expected number of packets successfully sent during T_u. Readers who are not interested in the derivation may skip this section and only read Equation (3.15) and its related text.

Recall that T_u is the time from the epoch that MP1 enters its State 1 until the epoch that it goes back to State 0 for the first time. In MP1, it is hard to capture when it is the first time that State 1 transits back to State 0. Thus, we make State 0 of Markov Process 1 into a trapping state, i.e., the process will never transit out once State 0 has been entered. Figure 3-5 illustrates this modification, and we call the modified process Markov Process 2 (MP2). All the other transition rates of Markov Process 2 are exactly the same as those of Markov Process 1.

For MP2, we have the following Kolmogrov backward differential equation [5, Chapter 3]:

$$\frac{d[P(t)]}{dt} = [Q][P(t)], \text{ for } t \geq 0 \tag{3.14}$$

$[P(t)]$ is a $(n+1) \times (n+1)$ matrix, where $P_{ij}(t)$ is the probability that the Markov process is in state j given that it was in state i at time 0; $[Q]$ is an $(n+1) \times (n+1)$ matrix with Q_{ij} as the rate of transition to state j from state i if $i \neq j$, and the rate

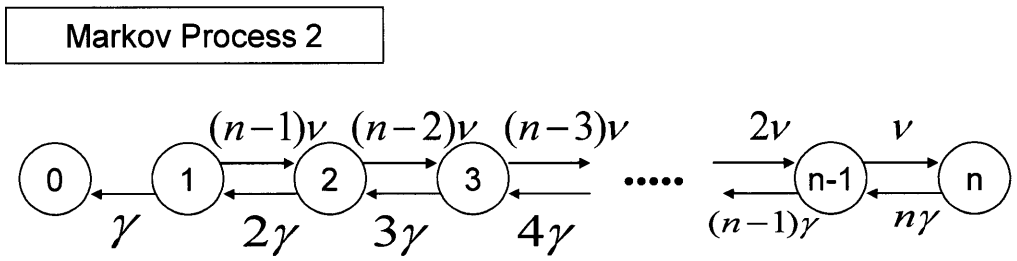

Figure 3-5: Modification to Markov Process 1 by making State 0 a trapping state (denoted as Markov Process 2)

of exiting state i if $i = j$.

We see that $P_{10}(t)$, the probability that Markov Process 2 is in State 0 at time t given that it was in State 1 at time 0, is the time that it takes for State 1 in MP1 to transit back to State 0 for the first time. Thus $P_{10}(t)$ and $F_{T_u}(t)$ are the same distribution function.

The matrix $[Q]$ will always have an eigenvalue 0, and a corresponding left eigenvector $\mathbf{1}$. We denote the other n eigenvalues of $[Q]$ as $\lambda_1, \lambda_2, ...\lambda_n$, with corresponding eigenvectors $v_1, v_2, ...v_n$. Note that $\lambda_1, \lambda_2, ...\lambda_n$ are all strictly negative. The initial condition of $[P(t)]$, denoted $[P(0)]$, is a diagonal matrix with diagonal entries all equal to 1 [5, Chapter 5].

The solution of $P_{10}(t)$ can be written as a linear combination of exponential terms with rates that are the eigenvalues of $[Q]$, i.e.

$$\begin{aligned} F_{T_u}(t) &= P_{10}(t) \\ &= 1 + \Sigma_{k=1}^{n} C_k \exp(\lambda_k t) \end{aligned} \tag{3.15}$$

where the coefficients C_k can be calculated from the eigenvalues, the eigenvectors and the initial conditions. In Appendix B, we solve the case where $n = 2$ analytically. The probability density function of $F_{T_u}(t)$, denoted $f_{T_u}(t)$, is

$$f_{T_u}(t) = \Sigma_{k=1}^{n} C_k \lambda_k \exp(\lambda_k t) \tag{3.16}$$

3.2.4 Expression of lower bound on TCP throughput

Now we have the distribution function $F_{T_u}(t)$, and we can find the expression for the sender's steady state TCP throughput. First, we find $E[R_i]$. We view time as a concatenation of round trip times. We assume the entire window sent in the last RTT before the sender and the receiver are disconnected is lost, which sets a lower bound on the linear lower bound of TCP throughput.

If $T_u = t \leq W_m \cdot RTT$, the sender will not reach the maximum window size, and during T_u it will send a total number of $\sum_{j=1}^{\frac{t}{RTT}-1} j = \frac{\frac{t}{RTT}(\frac{t}{RTT}-1)}{2}$ packets. If $T_u = t > W_m \cdot RTT$, during T_u the sender will send a total number of $\sum_{j=1}^{W_m} j + (\frac{t}{RTT} - W_m - 1) \cdot W_m = \frac{(W_m+1)W_m}{2} + (\frac{t}{RTT} - W_m - 1) \cdot W_m$ packets. To find the expression of R_i, we just need to take the integrals for both cases. Thus,

$$\begin{aligned} E[R_i] &= E[\text{number of succesfully sent packets within one inter-renewal interval}] \\ &= \int_0^{W_m \cdot RTT} \frac{\frac{t}{RTT}(\frac{t}{RTT}-1)}{2} f_{T_u}(t)\, dt \\ &\quad + \int_{W_m \cdot RTT}^{\infty} \{\frac{(W_m+1)W_m}{2} + (\frac{t}{RTT} - W_m - 1)W_m\} f_{T_u}(t)\, dt \end{aligned} \tag{3.17}$$

Carrying out the integration gives us the following result:

$$E[R_i] = \Sigma_{k=1}^{n} H_k \tag{3.18}$$

where

$$\begin{aligned} H_k &= \frac{C_k \{-2 - \lambda_k RTT + \exp(\lambda_k W_m RTT) \cdot [1 + W_m(-2 + RTT(-1 + W_m))]\}}{2\lambda_k^2 RTT^2} \\ &\quad - \frac{C_k \exp(\lambda_k W_m RTT) W_m(-2 + RTT(-1 + W_m)\lambda_k)}{2\lambda_k RTT} \end{aligned} \tag{3.19}$$

As stated before, we relax the maximum window size W_m to M in our analysis. We plug $E[R_i]$ into Equation (3.6) and we can get the TCP throughput in packets per second. Because the packet size is K bits, we can convert the TCP throughput in Equation (3.6) to bits per second.

$$\text{TCP throughput} = \frac{E[R_i]}{E[T_u] + E[T_d]} \cdot K \text{ (in bps)} \tag{3.20}$$

This expression for the lower bound of TCP throughput is accurate based on our models and assumptions, but it does not provide much intuition on what factors drive TCP throughput, because it contains many complicated terms. For example, from this TCP expression, we cannot tell easily how the throughput will change as the values of γ and ν change. We will plot the throughput expressed in this form in later sections, and those plots will help us understand what factors drive TCP throughput.

In addition, the derivation of this expression is complicated. The eigenvalues and eigenvectors of the $[Q]$ matrix become hard to solve as n becomes large. This drawback will become obvious when we want to study complicated network models. Thus, we will give an approximated model to Markov Process 1 in the next section, and we will see that we can derive the TCP throughput much more easily using this approximated model. Moreover, we will show that the approximation is fairly accurate when $\frac{\nu}{\gamma}$ becomes large.

3.3 TCP performance analysis by state aggregation

In this section, we give a simpler model to calculate the TCP throughput that gives similar results as the model we developed in Section 3.2.

3.3.1 A simplified Markov model by state aggregation

We noticed that in Markov Process 1, although we use $n+1$ states to represent the number of links that are *up* at any given time, the connection between the sender and the receiver only has two states: *connected* and *disconnected*. We saw that State 0 of MP1 represents the disconnection of the sender and the receiver, while the union of all the non-zero states of MP1 represents that the sender and the receiver are connected.

What if we aggregate States 1 to n of MP1 into one State called *connected*, and keep State 0 to form another state on its own, called *disconnected*? If we can make this aggregation to MP1, no matter how large n is, we can always model the connection between the sender and the receiver by two states, instead of $n+1$ states.

We make an audacious simplification to MP1 by claiming that the transitions between the sender and the receiver being *connected* to *disconnected* can be modeled by a two-state Markov process, denoted by Markov Process 3 as shown in Figure 3-6. If we use MP3 to analyze TCP throughput and if it gives us similar results to that of MP1, we may claim that MP3 is a good approximation to MP1 in terms of throughput calculation. In the next subsection, we will plot the two results and test how accurate the state aggregation is.

Now we need to find the state holding time distributions of Markov Process 3. Again we start from the steady state probability of the two states, π_c and π_{dc}. There are n independent links in parallel, each with a steady state probability $\pi_d = \frac{\gamma}{\gamma+\nu}$ of being in the *down* state, but the sender and the receiver will only be disconnected if all n links are in the *down* state at the same time. Hence, we have the following

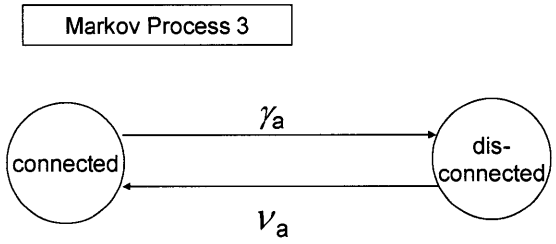

Figure 3-6: Aggregated Markov process to model n parallel paths. In this aggregated model, we combine all non-zero states of Markov Process 1 into one state, *connected*, and we leave State 0 of Markov Process 1 to be another state, *disconnected*. If this state aggregation gives us similar calculated TCP throughput, we may use this model because it greatly reduces the difficulty of calculation.

steady state probabilities of Markov Process 3:

$$\begin{aligned} \pi_{dc} &= \pi_d^n \\ &= (\frac{\gamma}{\gamma+\nu})^n \\ \pi_c &= 1-\pi_{dc} \end{aligned} \tag{3.21}$$

For the same reason as we discussed in Section 3.2, the disconnected time, T_{dc}, has the same distribution with T_d in Markov Process 1,

$$F_{T_{dc}}(t) = 1 - \exp(-n\nu t) \tag{3.22}$$

and

$$E[T_{dc}] = \frac{1}{n\nu}. \tag{3.23}$$

Under the Markov assumption, the ratio of expected time spent in each of the two states is equal to the ratio of the steady state probabilities of the two states,

$$\frac{E[T_c]}{E[T_{dc}]} = \frac{1-\pi_{dc}}{\pi_{dc}} \tag{3.24}$$

We get the expression for $E[T_c]$,

$$E[T_c] = \frac{1}{n\nu} \cdot \frac{1-\pi_{dc}}{\pi_{dc}} \tag{3.25}$$

In MP3, the connected time T_c is exponentially distributed with rate γ_a. Thus,

$$\begin{aligned} \gamma_a &= \frac{1}{E[T_c]} \\ &= \frac{n\nu\gamma^n}{(\gamma+\nu)^n - \gamma^n} \end{aligned} \tag{3.26}$$

Therefore, we write the distribution functions of the state holding time T_c and T_{dc} as:

$$F_{T_c}(t) = 1 - \exp(-\gamma_a t) \tag{3.27}$$

$$F_{T_{dc}}(t) = 1 - \exp(-n\nu t) \tag{3.28}$$

Now we want to find the TCP throughput over this simplified network model. This follows exactly the same procedure as we used in Section 3.2.4,

$$\begin{aligned} E[R_i] &= \int_0^{W_m \cdot RTT} \frac{\frac{t}{RTT}(\frac{t}{RTT}-1)}{2} f_{T_c}(t)\,dt \\ &\quad + \int_{W_m \cdot RTT}^{\infty} \{\frac{(W_m+1)W_m}{2} + (\frac{t}{RTT} - W_m - 1)W_m\} f_{T_c}(t)\,dt \end{aligned} \tag{3.29}$$

with $W_m = M$. Converting to bits per second, we get

$$\text{TCP throughput} = \frac{E[R_i]}{E[T_c] + E[T_{dc}]} \cdot K(\text{bps}) \tag{3.30}$$

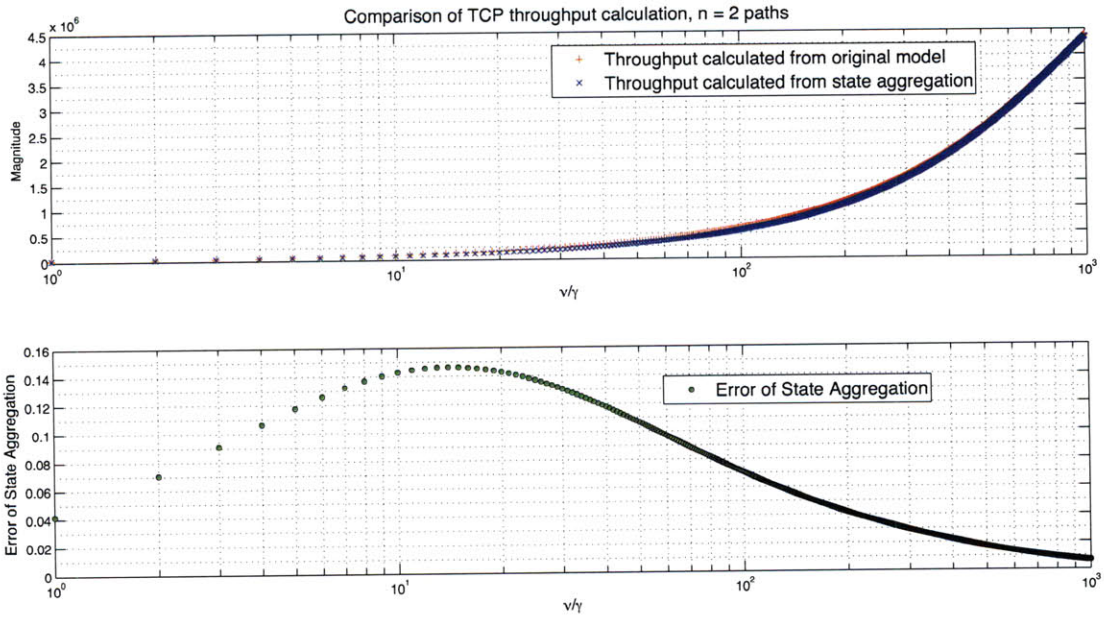

Figure 3-7: Comparison of TCP throughput calculated from Markov Process 1 and Markov Process 3 over a wireless channel. 2 parallel paths. $R_{\max} = 10$ Mb; packet size= 10 kb; $\frac{1}{\gamma} = 0.1$ sec, which is the expected length of connection between sender and receiver; $RTT = 0.1$ sec; $W_m = M = 100$ packets/RTT.

3.3.2 Numerical evidence of the accuracy of the simplified Markov model

Now we want to compare the lower bounds of TCP throughput calculated by MP1 and MP3, and see what factors drive the TCP throughput. To see how well MP3 can approximate MP1 measured in terms of TCP throughput, we plot the TCP throughput calculated from both equations on the same plot, and see how close they are.

First, we use the same value of n for both equations, and see how the TCP calculations differ. We still start from the case $n = 2$ and $W_m = M$. We realize that the ratio of ν to γ indicates the variability of the stochastic channels. If we fix γ, the smaller this ratio is, the more variable the channel is, or the more likely it is for the channel to go into the *down* state. We may expect that TCP achieves a better throughput as we increase the value of $\frac{\nu}{\gamma}$.

In Figure 3-7 A, we show the change of TCP throughput with respect to the

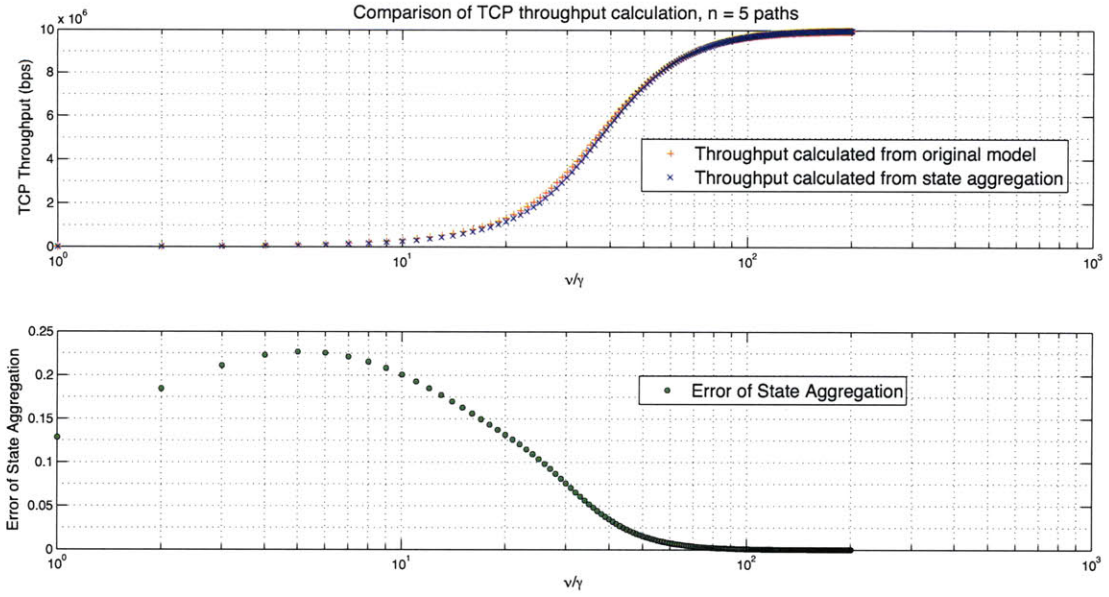

Figure 3-8: Comparison of TCP throughput calculated from Markov Process 1 and Markov Process 3 over a wireless channel. 5 parallel paths. $R_{\max} = 10$ Mb; packet size= 5 kb; $\frac{1}{\gamma} = 0.1$ sec, which is the expected length of connection between sender and receiver; $RTT = 0.1$ sec; $W_m = M = 200$ packets/RTT.

change of $\frac{\nu}{\gamma}$ in a wireless channel. Note that the horizontal axes are in log scale. The value of γ is fixed at 1 sec^{-1} since from Appendix A we know that the γ value for wireless channels should be on the order of one second. We constantly increase the value of $\frac{\nu}{\gamma}$ along the horizontal axis. The red curve is the throughput calculated from Markov Process 1, and the blue curve is the throughput calculated from the approximated Markov Process 3. We see that as the ratio $\frac{\nu}{\gamma}$ grows higher and higher, it is less likely for the paths to go *down*, and the TCP throughput increases.

In Figure 3-7 B, we plot the error of the approximated Markov process by state aggregation. We denote the throughput calculated from Markov Process 1 by TP^1, and the throughput calculated from Markov Process 3 by TP^3, and we define the error of throughput calculation to be the following:

$$\text{Error of Caculated TCP Throughput} = \left| \frac{TP^3 - TP^1}{TP^1} \right| \tag{3.31}$$

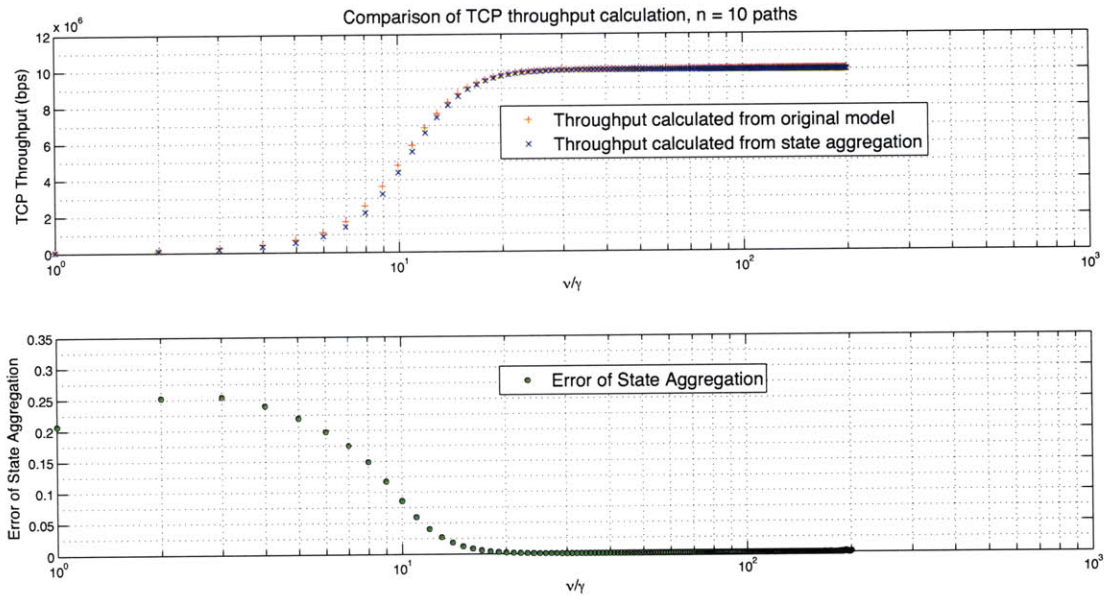

Figure 3-9: Comparison of TCP throughput calculated from Markov Process 1 and Markov Process 3 over a wireless channel. 10 parallel paths. $R_{\max} = 10$ Mb; packet size= 5 kb; $\frac{1}{\gamma} = 0.1$ sec, which is the expected length of connection between sender and receiver; $RTT = 0.1$ sec; $W_m = M = 200$ packets/RTT.

The curve in Figure 3-7 B shows that in our case, the worst error is about 15%, when $\frac{\nu}{\gamma} = 2$. When $\frac{\nu}{\gamma} > 2$, the error decreases as $\frac{\nu}{\gamma}$ increases, and the error converges to 0. We expect this pattern of error to occur when n becomes larger, which is plotted in Figure 3-8 for the case of $n = 5$, and in Figure 3-9 for the case of $n = 10$. Again, the error converges to 0 as $\frac{\nu}{\gamma}$ becomes higher, and the two curves TP^1 and TP^3 move closer to each other.

From the above three plots, we conclude that numerical analysis verifies that as $\frac{\nu}{\gamma}$ becomes large, the throughput calculation by state aggregation converges to the calculation by Markov Process 1.

3.3.3 Analysis of the accuracy of the simplified Markov model

To understand why the approximation becomes accurate when $\frac{\nu}{\gamma}$ becomes large, we compare the distributions of T_u and T_c. When n becomes large, it is hard to find the eigenvalues and eigenvectors of the matrix $[Q]$, but we can study the distribution

functions $F_{T_u}(t)$ and $F_{T_c}(t)$ analytically when $n = 2$.

From Equation (3.15), (3.27) and Appendix B, we see that when $n = 2$,

$$\begin{aligned} F_{T_u}(t) &= 1 + c_1 a \exp(\lambda_1 t) + c_2 b \exp(\lambda_2 t) \\ F_{T_c}(t) &= 1 - \exp(\frac{2\gamma^2}{2\gamma + \nu} t) \end{aligned} \tag{3.32}$$

If we can prove that the above two distribution functions become close when $\frac{\nu}{\gamma}$ become large, we may say that the state aggregation is reasonable. Looking at the terms of $F_{T_u}(t)$, and defining $\rho = \frac{\gamma}{\nu} \ll 1$, we know that

$$\begin{aligned} a &= \frac{1}{4\gamma}(\gamma - \nu + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \\ &= \frac{1}{4\gamma}\{\gamma - \nu + \nu[1 + \frac{1}{2}(6\rho + \rho^2) + o(\rho^2)]\} \\ &= \frac{1}{4\gamma}[\gamma + 3\gamma + o(\rho^2)] \\ &= 1 + o(\rho^2)) \end{aligned} \tag{3.33}$$

From the first equality to the second equality, we used the Taylor series of the square root

$$\sqrt{1+\eta} = 1 + \frac{1}{2}\eta - \frac{1}{8}\eta^2 + ..., \ |\eta| < 1 \tag{3.34}$$

Following the same logic, we see that

$$\begin{aligned} b &= \frac{1}{4\gamma}(\gamma - \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \\ &= \frac{1}{4\gamma}\{\gamma - \nu - \nu[1 + \frac{1}{2}(6\rho + \rho^2) + o(\rho^2)]\} \\ &= \frac{1}{4\gamma}[\gamma - 2\nu - 3\gamma + o(\rho^2)] \\ &= -\frac{1}{2} - \frac{1}{2\rho} + o(\rho^2) \end{aligned} \tag{3.35}$$

Thus,

$$\begin{aligned} c_1 &= \frac{b-1}{a-b} \\ &= \frac{b-1}{1-b+o(\rho^2)} \\ &= -1+o(\rho^2) \end{aligned} \tag{3.36}$$

and

$$\begin{aligned} c_2 &= \frac{1-a}{a-b} \\ &= \frac{o(\rho^2)}{1-b+o(\rho^2)} \\ &= o(\rho^2) \end{aligned} \tag{3.37}$$

Now we turn to look at the 2 eigenvalues of the $[Q]$ matrix.

$$\begin{aligned} \lambda_1 &= -\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2+6\gamma\nu+\nu^2} \\ &= -\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{\nu}{2}[1+\frac{1}{2}(6\rho+\rho^2)+\frac{1}{8}36\rho^2+o(\rho^2)] \\ &= -\frac{2\gamma^2}{\nu}+o(\rho^2) \end{aligned} \tag{3.38}$$

$$\begin{aligned} \lambda_2 &= -\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2+6\gamma\nu+\nu^2} \\ &= -\frac{3}{2}\gamma - \frac{1}{2}\nu - \frac{\nu}{2}[1+\frac{1}{2}(6\rho+\rho^2)+\frac{1}{8}36\rho^2+o(\rho^2)] \\ &= -3\gamma-\nu+o(\rho^2) \end{aligned} \tag{3.39}$$

If we plug all these parameters into the expression for F_{T_u}, we see that

$$\begin{aligned} F_{T_u}(t) &= 1+[-1+o(\rho^2)]\cdot[1+o(\rho^2))]\exp\{[-\frac{2\gamma^2}{\nu}+o(\rho^2)]t\} \\ &+ o(\rho^2)\cdot[-\frac{1}{2}-\frac{1}{2\rho}+o(\rho^2)]\exp\{[-3\gamma-\nu+o(\rho^2)]t\} \end{aligned} \tag{3.40}$$

We see that when ρ becomes large, the third term is very small compared to the second term in the above equation. Thus, we can ignore the third term,

$$F_{T_u}(t) = 1 + [-1 + o(\rho^2)] \cdot [1 + o(\rho^2))] \exp\{[-\frac{2\gamma^2}{\nu} + o(\rho^2)]t\} \tag{3.41}$$

Comparing $F_{T_u}(t)$ with $F_{T_c}(t)$ as in Equation (3.27), we see that when ρ becomes large, $T_u \to T_c$ in distribution. Since the distributions of the first passage time to State 0 are asymptotically equivalent when $n = 2$, the calculated TCP throughput should be the same. Intuitively, when n is larger the 2, the state aggregation still achieves satisfactory accuracy because the error made by aggregation will not blow up as n becomes large.

3.4 TCP efficiency

Remember in diversity routing strategy, we copy the data n times and send each copy along one of n independent paths. Thus, the network capacity used by the sender is $nR_{\max}$, and at most $\frac{1}{n}$ of this capacity is efficiently used. As the value of n increases, the TCP throughput increases monotonically; however, the portion of the network capacity that is used to send non-repetitive data becomes less and less. Thus, it may become less efficient if the sender sends too many copies of the same data packet. To measure how efficient the diversity routing strategy is, we define the TCP efficiency for a single network user as the following:

$$\begin{aligned} \text{TCP Efficiency} &= \frac{\text{TCP Throughput by Sender}}{\text{Network Capacity Used by Sender}} \\ &= \frac{\text{TCP Throughput}}{nR_{\max}} \end{aligned} \tag{3.42}$$

TCP efficiency is essentially the efficient network capacity utilization. It represents how much of the network capacity is used to successfully send non-repetitive data. Sending copies of data along multiple links is more costly to the users than

the traditional shortest path/minimum cost algorithm. We are interested in looking for an optimal number of links, n^*, that can maximize the TCP efficiency. If we can confirm the existence of this n^* and study its dependency on the values of γ, ν and RTT, we may further understand how to optimize the TCP throughput for users of networks with stochastic links.

To find the value of n^*, we just need to take partial derivative of the TCP efficiency's expression with respect to n. Expressing the TCP efficiency, taking the partial derivative and setting it to 0, we have

$$\begin{aligned}
&\frac{\partial}{\partial n}\{\frac{1}{nW_m(\frac{1}{n\nu}+\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)}{n\nu})}\cdot \\
&\{\frac{\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(2(\gamma+\nu)^n+\gamma^n(-2+nRTT(-1+W_m)\gamma))}{2nRTT\nu} \\
&-\frac{\gamma^{-2n}}{2n^2RTT^2\nu^2}[-2(\gamma+\nu)^{2n}-\gamma^{2n}(2+nRTT\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu) \\
&+e^{\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n}}[2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^2(-4+nRTT(-1+2W_m)\nu) \\
&+\gamma^{2n}(2+nRTT\gamma(1-2W_m+nRTT(-1+W_m)W_m\nu))]]\}\} \\
=\ &0 \qquad (3.43)
\end{aligned}$$

In Appendix D, we carry out the partial derivative in Equation (3.43) and display the transcendental equation that can be used to solve for n^*. Note that we assume $M = W_m$. Due to the complexity of this equation, instead of solving it, we use a numerical analysis approach to study this optimization problem.

In the rest of this section, we are going to study TCP efficiency in several different scenarios. These scenarios are simplified representations of real life network systems. For each scenario, we are going to generate plots using the models we developed in the previous sections, look for the n^* in each scenario, and discuss any insights we can get from these plots.

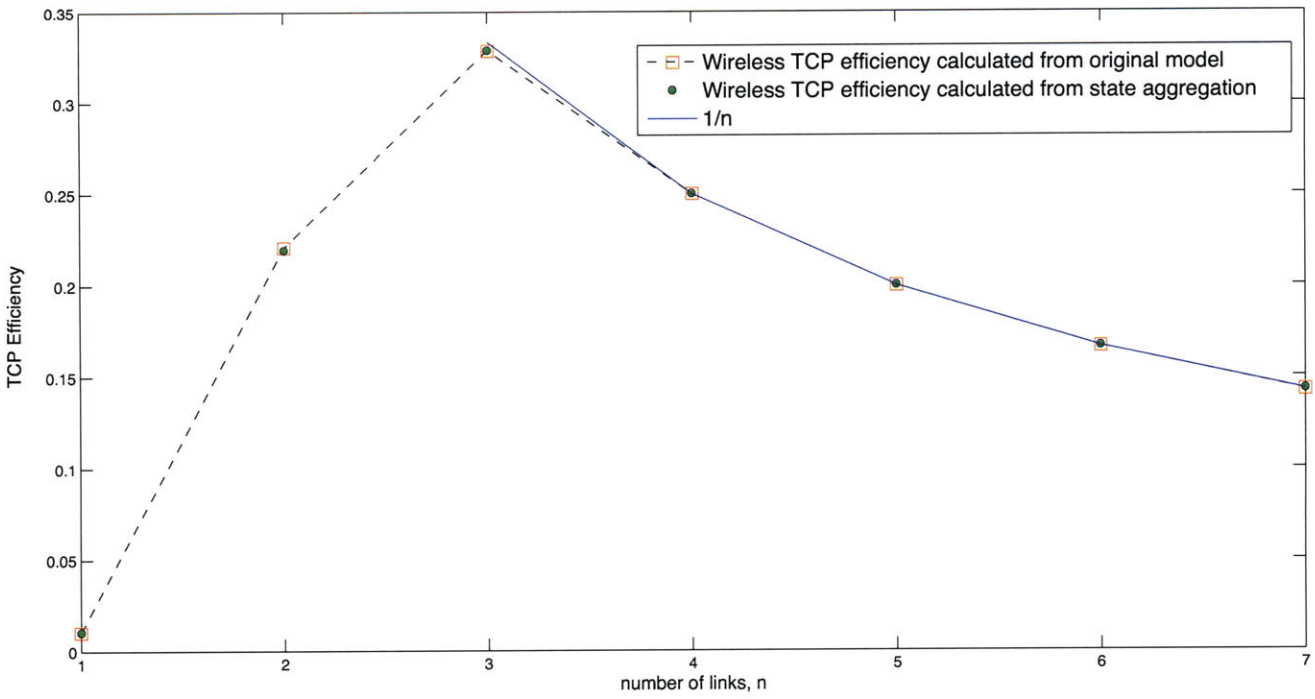

Figure 3-10: TCP efficiency in a wireless network

3.4.1 Networks consisting of wireless links

In wireless networks, we pick the maximum data rate $R_{\max}$ to be 0.1 Gbps. We choose the round trip time to be 0.1 sec, $\gamma = 1s^{-1}$, $\nu = 100s^{-1}$. In Figure 3-10, we plot the TCP efficiency of the sender for different values of n; for comparison we plot the curve $\frac{1}{n}$ as the maximum portion of network capacity that can be used to send non-repetitive data packets when n copies of data packet are sent.

As we see from Figure 3-10, in the wireless network case, the value of n^* that maximizes the TCP efficiency is 3, with the maximum efficiency 33%. We see that when $n > n^*$ the achieved TCP efficiency decreases and approaches $\frac{1}{n}$; for $n < n^*$, the achieved TCP efficiency was far below the maximum efficiency. We also see that the calculated TCP efficiency from both Markov Process 1 and Markov Process 3 are very close for all n, and the value of n^* is fairly small, due to the fact that for a single wireless channel, it is already fairly unlikely for it to go *down*.

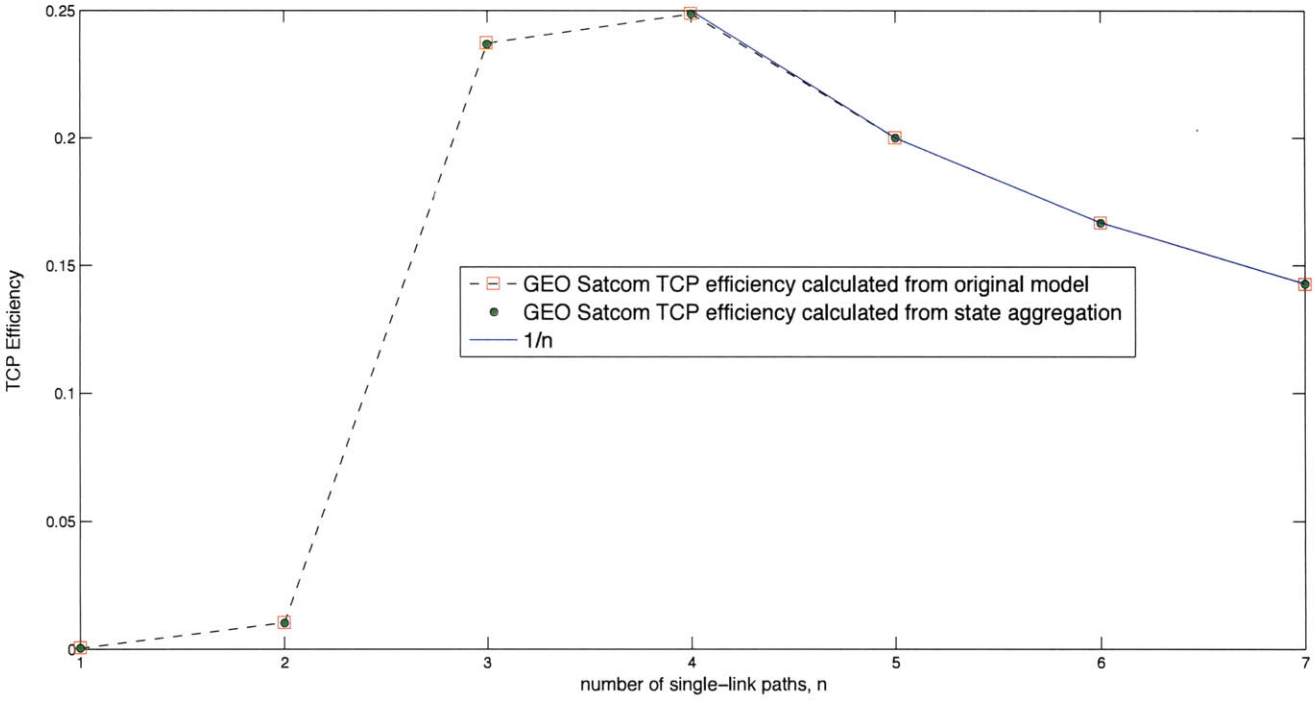

Figure 3-11: TCP efficiency in GEO satellite network versus the number of diversity routing paths. $R_{\max} = 1$ Gb; packet size= 10 kb ; $\frac{1}{\gamma} = 10$ sec, which is the expected length of connection between sender and receiver;$\frac{1}{\nu} = 0.01$ sec, which is the expected length of disconnection between sender and receiver; $RTT = 0.5$ sec; $W_m = M =$ 50000 packets/RTT. $n^* = 4$

3.4.2 Networks consisting of GEO satellite links

In GEO satellite networks, we pick the maximum data rate $R_{\max}$ to be 1 Gb, or 10^9 bps. We choose the RTT to be 0.5 sec due to the long propagation delays in satellite networks. $\gamma = 0.1s^{-1}$, $\nu = 100s^{-1}$. In Figure 3-11 we plot the TCP efficiency of the sender for different values of n; as a comparison, we also plot the curve $\frac{1}{n}$ as the maximum portion of the network capacity that can be used by a sender to send non-repetitive data packets.

As we see from Figure 3-10, the value of n^* that maximizes the TCP efficiency is 4, with a maximum efficiency of 24%. For $n > n^*$, the achieved TCP efficiency decreases

and approaches $\frac{1}{n}$. On the other hand, for $n < n^*$, the achieved TCP efficiency is far below the maximum efficiency. Moreover, we see that the TCP efficiency calculations from Markov Process 1 and Markov Process 3 are very close to each other for all values n, which demonstrates again that the state aggregation provides satisfactory results as an approximation to MP1.

3.4.3 Parameters affecting TCP Efficiency

In this section, we vary the value of $\frac{\nu}{\gamma}$, as well as the value of RTT, to further understand what factors drive the TCP efficiency and how the value of n^* is related to these factors.

First, we want to see how changes in $\frac{\nu}{\gamma}$ will make a difference in the value of n^*. In Figure 3-12, we have set the ratio $\frac{\nu}{\gamma}$ to be $1:1$, with both γ and ν equal to 1 sec^{-1}. We keep all the other parameters the same as those in Figure 3-10. Because a 1:1 ratio of γ and ν indicates that the links are much more variable in Figure 3-12 than in Figure 3-10, we now need $n^* = 12$ for the sender to achieve the optimum efficiency; the maximum efficiency is only 7.2%, which is far below the maximum TCP efficiency achieved when the links are less variable. We also noticed that the TCP calculations from Markov Process 1 and 3 have the the same trends, though due to the low $\frac{\nu}{\gamma}$ value, the two calculations are a little off when n is small.

In Figure 3-13, we plot the value of n^* versus the ratio $\frac{\nu}{\gamma}$, keeping γ fixed. On the x-axis, we vary the ratio $\frac{\nu}{\gamma}$ and plot it on the log scale. On the y-axis is the corresponding value of n^* that maximizes the TCP efficiency in each case, plotted in both linear scale and log scale. We see that as $\frac{\nu}{\gamma}$ becomes larger, the channels become less variable; accordingly, the number of links needed to maximize TCP efficiency decreases, and the achieved maximum throughput increases.

In Figure 3-14, we plot exactly the value of n^* versus the ratio $\frac{\nu}{\gamma}$ in networks that consist of satellite channels. We observe the same trend of n^* versus $\frac{\nu}{\gamma}$ in this case. As $\frac{\nu}{\gamma}$ becomes larger, the satellite links become less variable, and we need fewer links to maximize the TCP efficiency. We see that n^* is a non-increasing function of $\frac{\nu}{\gamma}$.

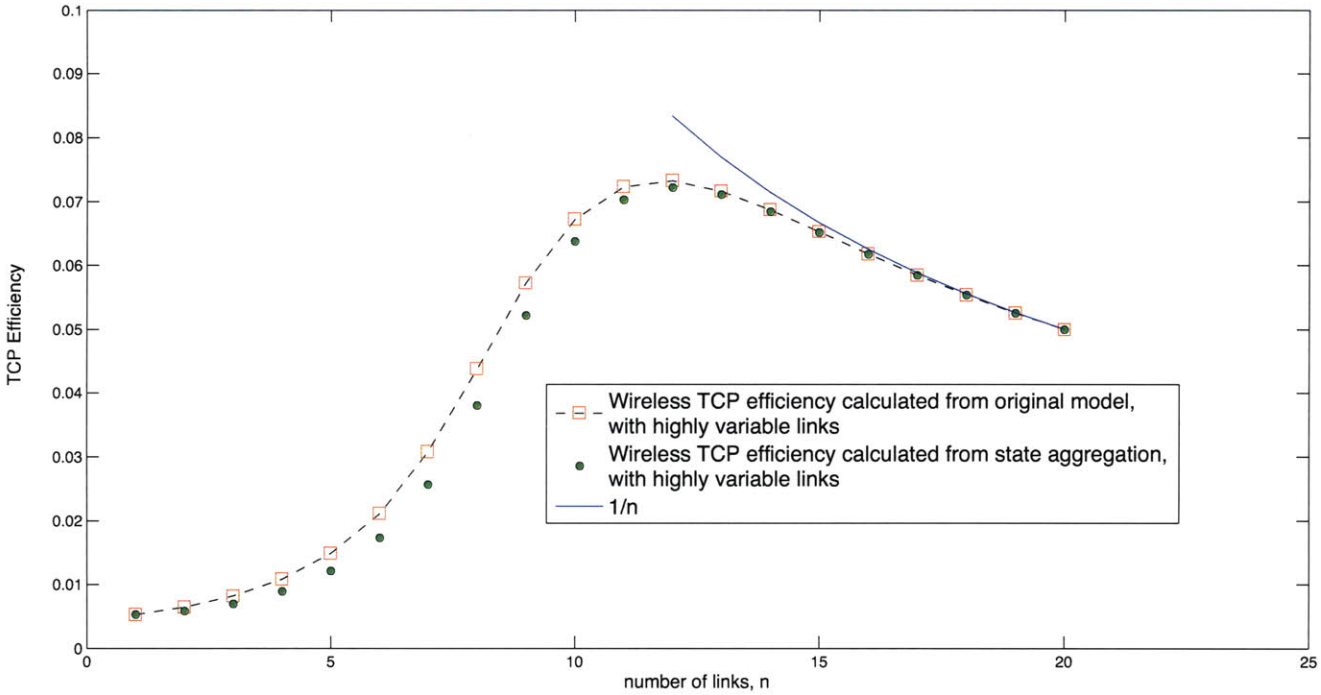

Figure 3-12: TCP efficiency in wireless network versus the number of diversity routing paths, when the links are highly variable:$\frac{1}{\gamma} = 1$ sec,which is the expected length of connection between sender and receiver, $\frac{1}{\nu} = 1$ sec,which is the expected length of disconnection between sender and receiver. $R_{\max} = 1$ Gb; packet size= 10 kb; $RTT = 0.1$ sec; $W_m = M = 1000$ packets/RTT. $n^* = 12$

We also noticed that the value of RTT may affect the value of n^*. The effect of RTT becomes more obvious when the value of $\frac{\nu}{\gamma}$ is relatively small. In Figure 3-15 we vary the round trip time of a stochastic channel to be 0.01 sec, 0.1 sec and 0.5 sec. We see that when the RTT is longer, we need fewer links to optimize the TCP efficiency under *any* ratio of $\frac{\nu}{\gamma}$. Intuitively, this phenomenon makes sense because the longer the RTT is, the larger the maximum window size $W_m = M$ is, and the longer it takes the sender to re-initiate sending packets to achieve M every time after it closes its window. To optimize TCP efficiency, we need to allow the sender to have enough expected time to achieve M; the way to give the sender more time to send data packets is to increase the number of links. That is why when RTT is longer, the value of n^* increases accordingly.

In conclusion, we see that for a network that consists of multiple identical stochastic links, such as satellite links and wireless links, we can model the sender's throughput under diversity routing strategy by setting up a $(n+1)$-state Markov process, or by an approximated Markov process that only has two states. When the value $\frac{"}{\nu}\gamma$ becomes large, the approximation gives satisfactory results. Moreover, there is an optimal number of diversity routing paths, n^*, that can maximize the expected efficiency of TCP. The value of n^* is a non-increasing function with respect to the value of $\frac{\nu}{\gamma}$, which is a direct indicator of the variability of the link: the link is more variable when the value $\frac{\nu}{\gamma}$ is smaller. Given that everything else is the same, the more variable the links are, the larger n^* is. Finally, we see that the value of n^* is a non-decreasing function of the round trip time, under any ratio of $\frac{\nu}{\gamma}$. This is because under a longer RTT, it takes the sender more time to achieve its maximum window size. As a result, we need to send the data packets along more paths to increase the expected time during which the sender and the destination are *connected*, which gives the sender longer time to ramp up its window size.

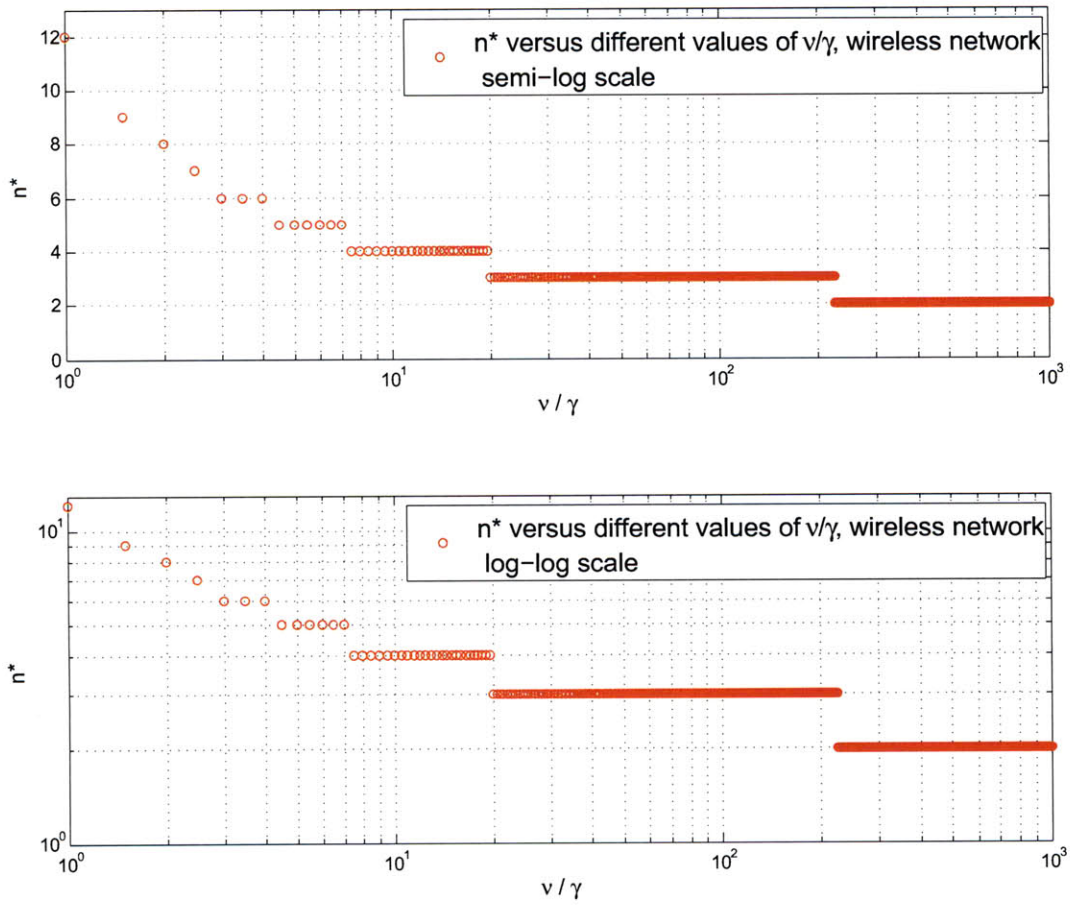

Figure 3-13: Number of links that maximizes TCP efficiency in a wireless channel versus different $\frac{\nu}{\gamma}$, denoted by n^*, plotted in semi-log and log-log scale. $\frac{1}{\gamma} = 1$ sec, which is the expected length of connection between sender and receiver, $\frac{1}{\nu}$ varies. $R_{\max} = 0.1$ Gb; packet size= 10 kb; $RTT = 0.1$ sec; $W_m = M = 1000$ packets/RTT. We see that when $\frac{\nu}{\gamma} \leq 10$, the slope of the log-log scale plot is approximately $-\frac{1}{2}$, i.e., $n^* \propto (\frac{\nu}{\gamma})^{-\frac{1}{2}}$.

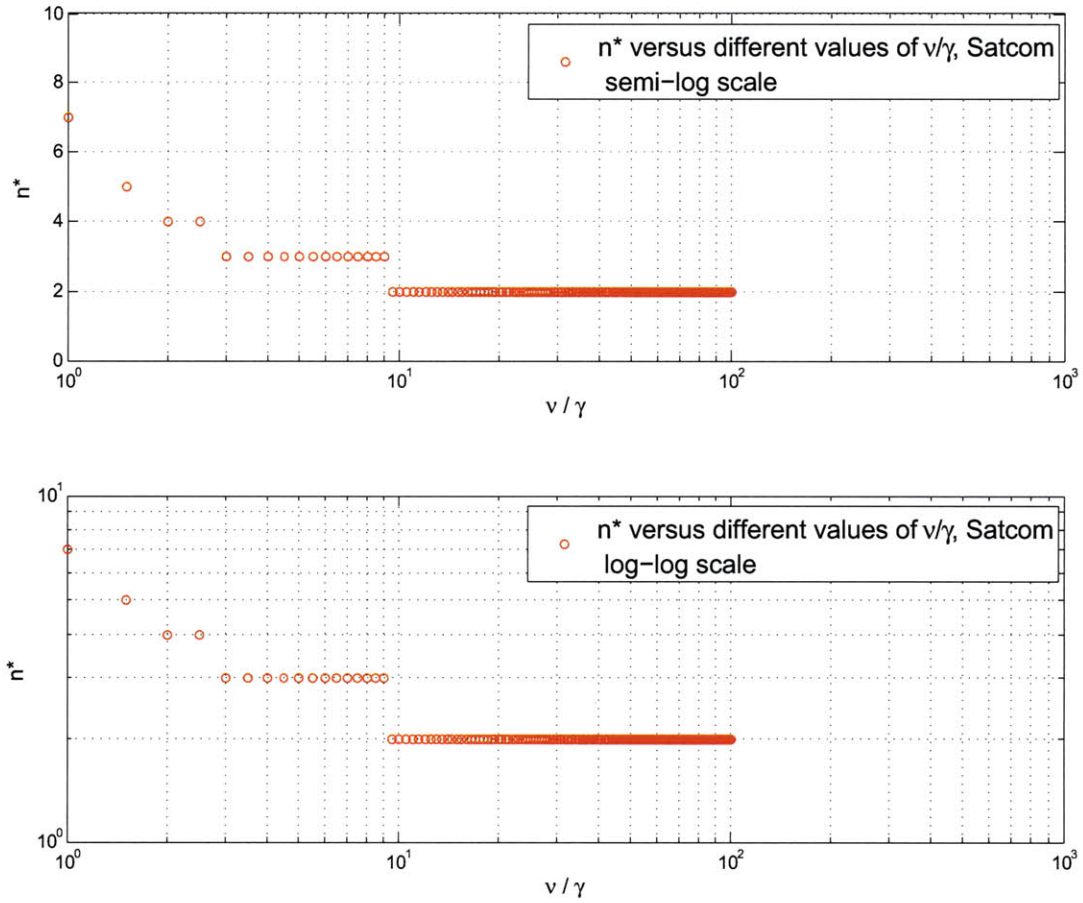

Figure 3-14: n^* in a GEO satellite channel versus different $\frac{\nu}{\gamma}$, denoted by n^*, plotted in semi-log and log-log scale. $\frac{1}{\gamma} = 0.1$ sec, which is the expected length of connection between sender and receiver, $\frac{1}{\nu}$ varies. $R_{\max} = 1$ Gb; packet size = 10 kb; $RTT = 0.5$ sec; $W_m = M = 50000$ packets/RTT.

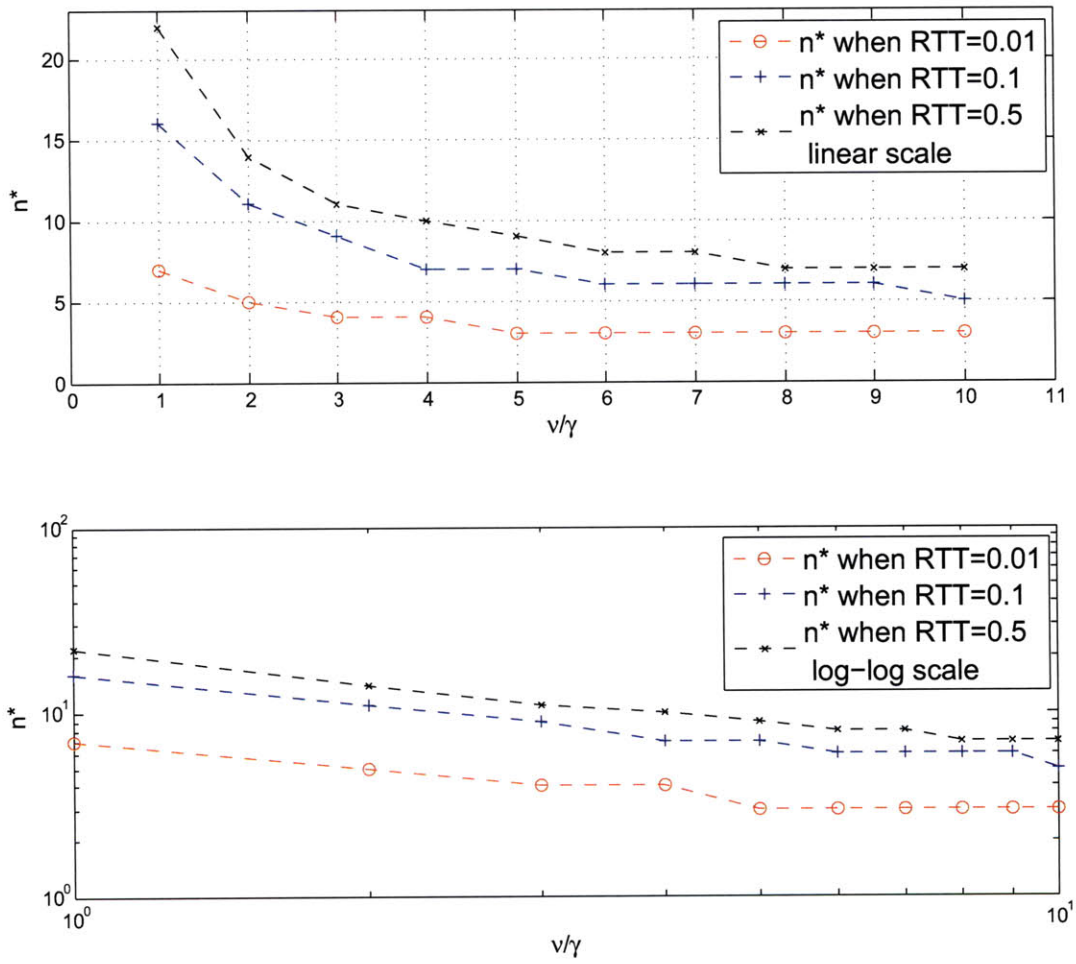

Figure 3-15: n^* versus different $\frac{\nu}{\gamma}$ in a wireless network with 3 different round trip times, plotted in both linear scale and log-log scale. $\frac{1}{\gamma} = 1$ sec, which is the expected length of connection between sender and receiver. $\frac{1}{\nu}$ varies. $R_{\max} = 1$ Gb and packet size= 10 kb. We see that as the value of RTT increases, n^* increases. We also see that $\log(n^*)$ decreases almost linearly as $\frac{\nu}{\gamma}$ increases.

Chapter 4

TCP Performance analysis over a network with parallel multi-link paths

In this chapter, we study the performance of TCP over a more complicated network model with stochastic links, where there are n paths in parallel between the sender and the receiver, and each of the n paths consists of m identical, independent links. The procedure and models that we use to analyze the TCP performance of these network models are extended from those developed in Chapter 3. In the end of this chapter, we will discuss how our study can be used to improve the routing strategy for the sender to achieve better efficiency and/or throughput.

4.1 Network model with n parallel paths and m identical links on each path

In Figure 4-1, we give an illustration of the first extended model that we solve, when there are n identical parallel paths between the sender and the receiver, and each path consists of m identical links. Each of these links again can be modeled by a

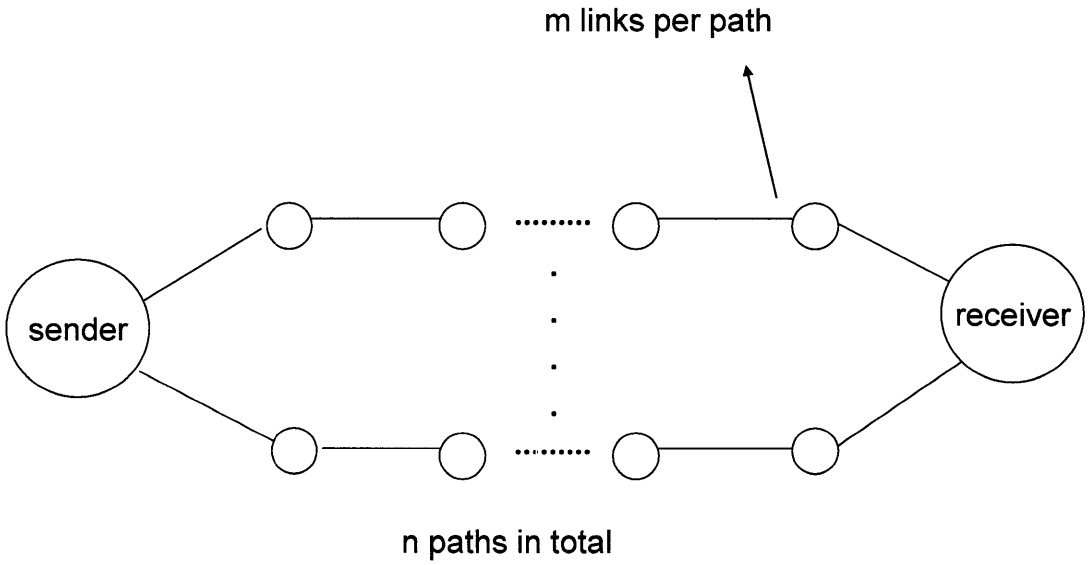

Figure 4-1: n parallel paths with m identical links on each path

two-state Markov process, with rates γ and ν.

We are going to analyze the TCP throughput of this network model. We first consider the m links on each path. Because the m links are in concatenation, as long as one of them goes *down*, the path is disconnected. Assume at a given time all m links on a path are all *up*. It is obvious to see that the time it takes one of the m links to go *down* is exponentially distributed with rate $m\gamma$. Use T_{pc} to denote the time that an m link path will remain connected before it goes disconnected, we have

$$F_{T_{pc}} = 1 - \exp(-m\gamma t) \tag{4.1}$$

and

$$E[T_{pc}] = \frac{1}{m\gamma} \tag{4.2}$$

We also see that the steady state probabilities for an m-link path to be connected or disconnected, π_{pc} and π_{pd} are

$$\begin{aligned} \pi_{pc} &= \pi_u^m \\ &= (\frac{\nu}{\gamma+\nu})^m \\ \pi_{pd} &= 1 - \pi_{pc} \end{aligned} \tag{4.3}$$

Again, the ratio of expected time the path spends connected (denoted T_{pc}) and disconnected (denoted T_{pd}) is equal to the ratio of the steady state probabilities,

$$\frac{E[T_{pc}]}{E[T_{pd}]} = \frac{\pi_{pc}}{\pi_{pd}}$$

$$E[T_{pd}] = \frac{1}{m\gamma} \cdot \frac{1-\pi_{pc}}{\pi_{pc}} \tag{4.4}$$

Consider a network where there is only one path between the sender and the receiver, and there are m links on this path (Figure 4-2). We calculate the TCP

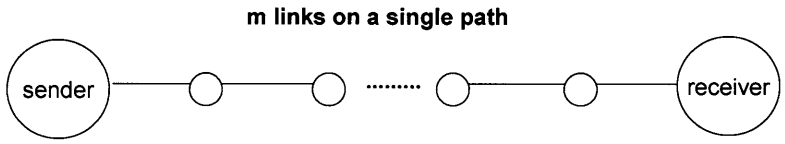

Figure 4-2: A single path consisting of m identical links

throughput of this sender, still assuming the maximum window size W_m equal to M

$$M = \frac{RTT \cdot R_{\max}}{K} \tag{4.5}$$

The TCP throughput will have the same format as in that derived in Equation (3.29), except that we are integrating with respect to $f_{T_{pc}}(t)$ now, and the expected inter-renewal interval length (i.e. the denominator of Equation (3.30)) becomes $E[T_{pc}] + E[T_{pd}]$.

In Figure 4-3 we plot the calculated TCP throughput for different values of m in a satellite network. Notice that the case when $m = 1$ should give the same throughput as the case when $n = 1$ in the Markov Process 1 Model developed in Chapter 3, both representing that there is a single stochastic link between the sender and the receiver.

We discover from Figure 4-3 that as the number of links in concatenation gets larger, the TCP throughput decreases. This makes intuitive sense because as long as one of the m links goes *down*, the path between the sender and the receiver is discon-

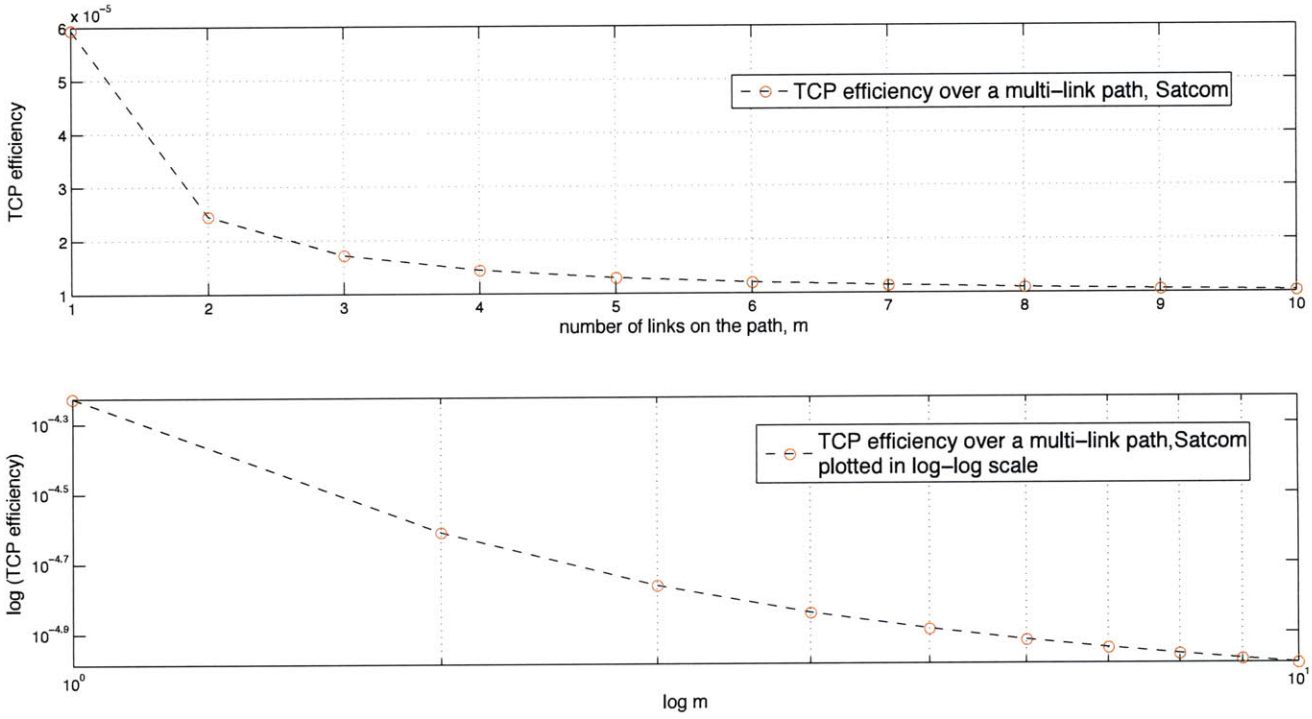

Figure 4-3: TCP efficiency over one path consisting of m identical, independent GEO satellite links, plotted in both linear scale and log-log scale. $\frac{1}{\gamma} = 1$ sec, which is the expected length of connection between sender and receiver, $\frac{1}{\nu} = 0.01$ sec. $R_{\max} = 1$ Gb; packet size = 10 kb; $RTT = 0.5$ sec; $W_m = M = 50000$. We see that TCP efficiency decreases almost linearly as m increases, in log-log scale.

nected, thus the sender needs to re-start its window size increasing from 1 packet per RTT. The more links on a single paths there are, the more likely it is at any time for one of them to break *down*, thus the less time we should expect the path between the sender and receiver to remain connected. That is why we observe that TCP throughput decreases as the number of links in concatenation on a path increases.

In Appendix C, we prove that we can use a similar state aggregation trick to approximate the distributions of T_{pc} and T_{pd}. we use T_{pac} and T_{pad} to denote the path connected and disconnected time in the state aggregation model. The approximated distributions, $F_{pac}(t)$ and $F_{pad}(t)$ are exponential with transition rates γ_{pa} and ν_{pa}, respectively. The two transition rates are

$$\begin{aligned} \gamma_{pa} &= -m\gamma \\ \nu_{pa} &= \frac{m\gamma(\frac{\nu}{\gamma+\nu})^m}{1-(\frac{\nu}{\gamma+\nu})^m} \end{aligned} \tag{4.6}$$

Now we go back to the network model in Figure 4-1 after we study the performance of the TCP over one of the n paths. Since each of the n paths can be modeled by an on-off Markov process, we can treat the n paths as n stochastic links in parallel, each one with transitions rates $m\gamma$ and ν_{pa}. This will lead us back to the problem discussed in Chapter 3, where there are n single-link stochastic links in parallel. We will not present plots due to the repetitive information they will contain.

4.2 A baby version of a diversity routing strategy for a data sender

So far we have developed a methodology for the sender to evaluate its throughput and efficiency by applying diversity routing. We want to develop a baby version of diversity routing algorithm for the sender in a network with N parallel paths. As the first step to implement this baby version of routing algorithm, we need to evaluate each possible path by using the two-state Markovian approximation we discussed in the previous subsection. Using the γ and ν values of each link on a path, we may

determine the expected TCP throughput along any path. Record the calculated TCP throughput on each single path i as thu_{pi}.

In this work we assume the network is not congested. The sender is going to send its data packets along k paths in order to achieve a high efficiency, at the same time meet its own minimum requirement of throughput. The value of k is to be determined.

To determine the value of k, think about the optimization problem we solved in Chapter 3 to find n^*. We know that n^* is small if the paths are not too variable. In our problem, the sender is facing N parallel paths, each has its own variability. Sort the paths according to their variability, i.e., the ratio γ/ν, and keep the sorted path number into a set, $\{P_1, P_2 ... P_N\}$ such that the variability of the paths in the set is in ascending order. The sender first calculate its throughput and efficiency assuming it sends data along P_1, and denote the calculated throughput as thr_1, the calculated efficiency as e_1. Then, the sender can calculate the throughput and efficiency of diversity routing along both P_1 and P_2, and denote the calculated throughput thr_2 and e_2. We already know that as more paths are included, $thr_{i+1} > thr_i$ for all integers $i \geq 1$; however there is an optimal number of paths k^*, at which $e_{k^*-1} \leq e_{k^*}$ and $e_{k^*+1} \leq e_{k^*}$. The efficiency by routing along fewer than k^* paths is non-decreasing as k increases, and the efficiency by routing along more than k^* paths is non-increasing. For $j \geq k^*$, choose k to be the minimum j that meet the requirement $thr_j \geq thr_{required}$, where $thr_{required}$ is the required TCP throughput by the sender. This number k is the number of paths that the sender should diversity route its data packets, and the paths $P_1, P_2, ... P_k$ should be the k paths along which the sender sends data.

We see that if the sender keeps on sending the links along the optimal k paths, it is going to cause congestion in the network. Thus, in a real network, the sender needs to evaluate the congestion status of the network, combined with the optimization of its expected throughput and efficiency, to determine its routing strategy. This work is left to be done in the future.

Chapter 5

Conclusion and future work

5.1 Methodologies

In this thesis, we developed a systematic way to understand how diversity routing can improve TCP performance in networks with stochastic links, such as satellite networks and wireless networks. We looked at two major types of network models: first a simple network model with independent, identical single-link paths, and second a more complicated network model with multi-link paths.

In the single-link paths case, when there are n paths for diversity routing, we first used a $n+1$ state Markov model to represent the number of paths that are *up* at a given time, and used renewal processes to analyze the TCP window size control algorithm. By viewing TCP data sending as a renewal reward function over time, we were able to derive an expression for the linear lower bound of TCP throughput over networks with single-link paths.

This $n+1$-state Markov model is correct but hard to extend to large values of n. For large n, we aggregated all non-zero states of the Markov model to form a new two-state Markov model, with the two states representing whether or not all the n paths for diversity routing are disconnected. We gave both numerical and analytical evidence to show that the state-aggregation preserves satisfactory accuracy in calculated TCP throughput.

Using state-aggregation, we extended our analysis to a more complicated network

model, where there are n independent, identical parallel paths each consisting of m identical, independent and concatenated links. This network configuration can still be modeled by a two-state Markov process because we can first model the m links on each path by a two-state Markov model, and thus treat the n paths as n single-link paths. Then, we reduce the problem to the previous single-link paths case.

5.2 Major results

Our numerical analysis has shown that diversity routing can significantly increase the throughput of TCP over networks with stochastic links. The more paths along which copies of data are sent, the higher throughput that can be achieved.

However, focusing on the absolute value of throughput is not the best way to evaluate TCP performance. Instead, we define TCP efficiency as the portion of the network capacity used by a sender that is used to send non-repetitive data. The efficiency when n copies are sent is always upper bounded by $\frac{1}{n}$. We found the optimal number of paths along which data should be sent to maximize the sender's TCP efficiency, and this number is denoted by n^*. We understood that this number varies with the variability of the stochastic links, as well as the round trip time. The more variable the links are, the the larger n^* is; the longer the RTT is, the larger n^* is.

Based on our results, we have introduced a possible improvement to the routing strategy for a user of a network with stochastic links. Different from the traditional way of calculating the shortest path between the sender and receiver, we suggest that the sender may evaluate the variability of each stochastic path and calculate the expected TCP efficiency. Thus the sender is able to find the optimal number of paths along which it can send copies of data packets.

5.3 Future work

Future work that follows this research may involve the following: we need to systematically investigate the routing algorithm in Chapter 4.2, in particular we need to study how it can be combined with the traditional TCP congestion control algorithm. We need to test its improvement on scenarios when congestion is present. Our research has also made some simplifying assumptions. For example, the maximum TPC window size was set to be the maximum number of packets in flight per round trip time. The link variability statistics, i.e., the γ and ν values, are not extracted from real-life data, and we only have conjectures about the order of magnitude of these parameters, not about the exact values of them. These assumptions are reasonable because they simplify our research. However, to bring the research to an implementation level, we need to assess the impact of these assumptions on our results.

We also conjecture that a network model with n parallel paths, each consisting of distinct, independent and concatenated stochastic links can be modeled by using state-aggregation. The simplest example of such a network is when there are two distinct single-link paths between the sender and the receiver, each of which can be model by a two-state Markov process with transition rates γ_i and ν_i, where $i = 1, 2$ respectively.

We draw a corresponding 4-state Markov Process, MP4 to represent the transitions among the links. State "11" represents that both links are *up*; State "10" represents that Link 1 is *up* and Link 2 is *down*; State "01" represents Link 1 is *down* and Link 2 is *up*; State "00" represents both links are *down*. The transitions among the states are illustrated in Figure 5-1.

In general, we need to use 2^n-state Markov model to represent a network with n single-link paths. We propose that the state-aggregation method can still be used in this case. Further more,

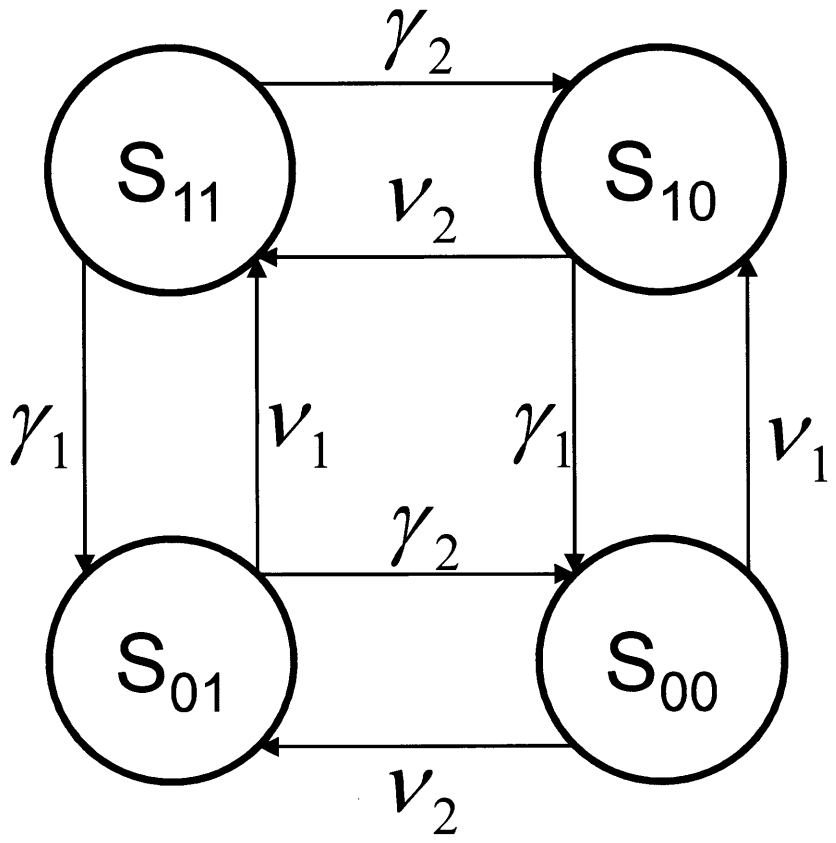

Figure 5-1: Markov Process 4: a four-state Markov process for 2 distinct single-link path network model.

Appendix A

Estimation of the Multipath Fading Time Parameters of Wireless Links

As described in the scenario of Figure 2-2, the sending antenna is moving towards the signal receiver at a constant velocity v, and is sending signal at frequency f_0. The sending antenna is fairly far away from the receiving antenna, thus $\theta_1 \ll \frac{\pi}{2}$ and $\theta_2 \ll \frac{\pi}{2}$.

We have

$$r_1 = \frac{h}{tan\theta_1} + \frac{h}{tan\theta_2} - vt \tag{A.1}$$

Let $R_1 = \frac{h}{tan\theta_1} + \frac{h}{tan\theta_2}$ for simplicity.

$$r_2 = \frac{h}{sin\theta_1} + \frac{h}{sin\theta_2} - vtcos\theta_1 \tag{A.2}$$

Let $R_2 = \frac{h}{sin\theta_1} + \frac{h}{sin\theta_2}$ for simplicity.

From Robert Gallager's *Principles of Digital Communication*[6], the signal field at the receiving antenna can be expressed as

$$E_r(f,t) = \frac{R\{\alpha \exp[2\pi i f_0(t - \frac{r_1}{c})]\}}{r_1 - vt} + \frac{R\{\alpha \exp[2\pi i f_0(t - \frac{r_2}{c})]\}}{r_2 - vt} \tag{A.3}$$

where c is the speed of light and $R\cdot$ represents the real part of the function inside the braces.

Since we assume $\theta_1 \ll \frac{\pi}{2}$ and $\theta_2 \ll \frac{\pi}{2}$, we can approximate the denominator of the second term to be $r_1 - vt$ as well. Thus,

$$\begin{aligned} E_r(f,t) &\propto R\{\alpha \exp[2\pi i f_0(t - \frac{r_1}{c})]\} + R\{\alpha \exp[2\pi i f_0(t - \frac{r_2}{c})]\} \\ &\propto R\{\alpha \exp[2\pi i f_0(t - R_1 + \frac{vt}{c})]\} + R\{\alpha \exp[2\pi i f_0(t - R_2 + \frac{vt\cos\theta}{c})]\} \\ &\propto \alpha \cos[2\pi f_0(t - R_1 + \frac{vt}{c})] + \alpha \cos[2\pi f_0(t - R_2 + \frac{vt\cos\theta}{c})] \end{aligned} \tag{A.4}$$

Since $\cos\theta_1 + \cos\theta_2 = 2\cos\frac{\theta_1+\theta_2}{2}\cos\frac{\theta_1-\theta_2}{2}$, we have

$$\begin{aligned} E_r(f,t) \propto 2\alpha \cos[\pi f_0(2t - R_1 - R_2 + \frac{vt}{c} + \frac{vt\cos\theta_1)}{c}] \\ \cdot \cos[\pi f_0(R_2 - R_1) + \pi f_0 \frac{v}{c}(1 - \cos\theta_1)t] \end{aligned} \tag{A.5}$$

From the above equation, we see that that the received signal field is proportional to the product of two cosine terms. Since the coefficient of t in the second term is smaller, the second cosine term is dominant. Thus, we know that the combined waveform will have a frequency f, and

$$\begin{aligned} f &= \frac{v}{2c} f_0(1 - \cos\theta_1) \\ &= \frac{v}{c} f_0 \cos^2\frac{\theta_1}{2} \end{aligned} \tag{A.6}$$

Thus the period of the combined wave form is

$$T(f_0) = \frac{c}{f_0 v \cos^2\frac{\theta_1}{2}} \tag{A.7}$$

We define the fading time $T_{\text{fading}}(f_0)$ as the period of time from when the received signal level falls below $\frac{1}{2}$ of the original signal sent, to the time when it goes up above $\frac{1}{2}$ of the original signal again. In a cosine waveform, $\arccos(\frac{1}{2}) = \frac{\pi}{6}$. This corresponds to $\frac{1}{12}$ of the period $T(f_0)$, and we need $2 \times \frac{1}{12} \times T(f_0) = \frac{T(f_0)}{6}$ for the signal level to

fade below the threshold and go back above the threshold. Thus,

$$T_{\text{fading}}(f_0) = \frac{c}{6 f_0 v \cos^2 \frac{\theta_1}{2}} \tag{A.8}$$

In one example wireless link, f_0 is 2GHz, $\theta_1 = 10°$, $c = 3 \times 10^8$ mps. The moving data sending (or receiving) antenna may be a car, with velocity around 30 miles/hr, or 13.3 mps. Plugging these numbers into the expression for $T_{\text{fading}}(f_0)$ gives $T_{\text{fading}}(f_0) = 0.0019$ sec.

Note that this is a rough estimation of the fading time for a single wireless link. We also know that in our two-state Markov process model,

$$\begin{aligned} f_{X_{w_{nf}}}(x) &= \gamma \exp(-\gamma_w x) \\ f_{X_{w_f}}(x) &= \nu \exp(-\nu_w x) \end{aligned} \tag{A.9}$$

Thus,

$$E[\text{fading time}] = \frac{1}{\nu_w} = 0.0019 \text{ sec} \tag{A.10}$$

We get that $\nu_w = 526$ sec^{-1}. We can assume that every 10 meters there is such a reflecting wall, thus

$$E[\text{non-fading time}] = \frac{1}{\gamma_w} = \frac{10\text{m}}{20\text{mps}} = 0.5 \text{ sec} \tag{A.11}$$

This suggests that the order of magnitude of ν_w is thousands of sec^{-1}, while that of γ_w is several sec^{-1}. The ratio of the two parameters is roughly 10^2 to 10^3.

Appendix B

Derivation of distribution of T_u over a network with 2 identical single-link paths

This appendix is a continuation of the derivation of the distribution of T_u in Chapter 3.2, but written out analytically for the case n=2.

For $n = 2$,we write out $[P(t)]$ and $[Q]$ analytically as

$$\begin{aligned} P &= \begin{bmatrix} P_{00}(t) & P_{01}(t) & P_{02}(t) \\ P_{10}(t) & P_{11}(t) & P_{12}(t) \\ P_{20}(t) & P_{21}(t) & P_{22}(t) \end{bmatrix} \\ Q &= \begin{bmatrix} 0 & 0 & 0 \\ \gamma & -\gamma-\nu & \nu \\ 0 & 2\gamma & -2\gamma \end{bmatrix} \end{aligned} \tag{B.1}$$

The eigenvalues of $[Q]$ are

$$\begin{aligned} \lambda_0 &= 0 \\ \lambda_1 &= -\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2} \\ \lambda_2 &= -\frac{3}{2}\gamma - \frac{1}{2}\nu - \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2} \end{aligned} \tag{B.2}$$

The corresponding eigenvectors of $[Q]$ are

$$\begin{aligned}
\mathbf{v_0} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \\
\mathbf{v_1} &= \begin{bmatrix} 0 & \frac{1}{4\gamma}(\gamma - \nu + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) & 1 \end{bmatrix}^T \\
\mathbf{v_2} &= \begin{bmatrix} 0 & \frac{1}{4\gamma}(\gamma - \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) & 1 \end{bmatrix}^T
\end{aligned} \tag{B.3}$$

The initial condition for the first column of $[P(t)]$ is

$$\begin{bmatrix} P_{00}(t) \\ P_{10}(t) \\ P_{20}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{B.4}$$

and for simplicity, we denote $\mathbf{v_1}(2)$ by a, and $\mathbf{v_2}(2)$ by b. Using this initial condition, we get the following solution for $P_{10}(t)$

$$P_{10}(t) = 1 + c_1 a \exp(\lambda_1 t) + c_2 b \exp(\lambda_2 t) \tag{B.5}$$

where

$$\begin{aligned}
c_1 = \frac{b-1}{a-b} &= \frac{-3\gamma - \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}} \\
c_2 = \frac{1-a}{a-b} &= \frac{3\gamma + \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}
\end{aligned} \tag{B.6}$$

Thus,

$$\begin{aligned}
F_{T_u}(t) &= P_{10}(t) \\
&= 1 + c_1 a \exp(\lambda_1 t) + c_2 b \exp(\lambda_2 t)
\end{aligned} \tag{B.7}$$

The expectation of T_u is

$$\begin{aligned}
E[T_u] &= \int_0^\infty (1 - F_{T_u}(t))\, dt \\
&= \frac{c_1 a}{\lambda_1} + \frac{c_2 b}{\lambda_2} \\
&= -\frac{3\gamma + \nu + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}} \frac{1}{4\gamma} (\gamma - \nu + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \frac{-\frac{3}{2}\gamma - \frac{1}{2}\nu - \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\gamma^2} \\
&\quad + \frac{3\gamma + \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}} \frac{1}{4\gamma} (\gamma - \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \frac{-\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}}{2\gamma^2} \\
&= \frac{1}{8\gamma^3\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}} \{(\gamma - \nu + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2})(\frac{3}{2}\gamma + \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2})^2 \\
&\quad - (\gamma - \nu - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2})(\frac{3}{2}\gamma + \frac{1}{2}\nu - \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2})^2\} \\
&= \frac{1}{8\gamma^3\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}} \{2\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}(\frac{9}{4}\gamma^2 + \frac{1}{4}\nu^2 + \frac{3}{2}\gamma\nu + \frac{1}{4}\gamma^2 + \frac{1}{4}6\gamma\nu + \frac{1}{4}\nu^2) \\
&\quad + 2(\gamma - \nu)(\frac{3}{2}\gamma + \frac{1}{2}\nu)\sqrt{\gamma^2 + 6\gamma\nu + \nu^2}\} \\
&= \frac{1}{8\gamma^3}(8\gamma^2 + 4\gamma\nu) \\
&= \frac{1}{\gamma} + \frac{\nu}{2\gamma^2}
\end{aligned} \tag{B.8}$$

We can verify $E[T_u]$ in another way. In Markov Process 1, when $n = 2$, we have the following steady state probability

$$\begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \end{bmatrix} = \begin{bmatrix} \frac{\gamma^2}{(\gamma+\nu)^2} \\ \frac{2\gamma\nu}{(\gamma+\nu)^2} \\ \frac{\nu^2}{(\gamma+\nu)^2} \end{bmatrix} \tag{B.9}$$

We also know that the length of time spent in State 0 follows an exponential distribution

$$f_{T_d}(t) = 2\nu \exp(-2\nu t) \tag{B.10}$$

and

$$E[T_d(t)] = \frac{1}{2\nu} \tag{B.11}$$

From the definition of the steady state probabilities,

$$\frac{E[T_u]}{E[T_d]} = \frac{\pi_1 + \pi_2}{\pi_0} \tag{B.12}$$

Thus,

$$\begin{aligned} E[T_u] &= \frac{\frac{2\gamma\nu}{(\gamma+\nu)^2} + \frac{\nu^2}{(\gamma+\nu)^2}}{\frac{\gamma^2}{(\gamma+\nu)^2}} \frac{1}{2\nu} \\ &= \frac{1}{\gamma} + \frac{\nu}{2\gamma^2} \end{aligned} \tag{B.13}$$

which agrees with Equation (B.8).

Appendix C

State aggregation Markov model for network with one multi-link path

In this appendix we show that the network model in which there is a single path between the sender and the receiver consisting of m identical and independent stochastic links can be approximated by a two-state Markov process. If we build an $(m+1)$-state Markov process with state i representing the number of links that are *down*, then State 0 represents the fact that the sender and the receiver is *connected*, and the aggregation of the rest m states represents the fact that the sender and the receiver is *disconnected*, as shown in Figure C-1. We denote this Markov process by MP-A1. We prove analytically that when $m = 2$, the state-aggregation Markov model is a good approximation to the $(m+1)$-state Markov model.

When $m = 2$, we know that T_{pc} is exponentially distributed as

$$F_{T_{pc}}(t) = 1 - \exp(-2\gamma t) \tag{C.1}$$

and we want to find the time distribution of T_{pd}.

We see that T_{pd} is the time epoch from the epoch that MP-A1 just enters State 1, until the time epoch that MP-A1 is about to go back to State 0 from State 1

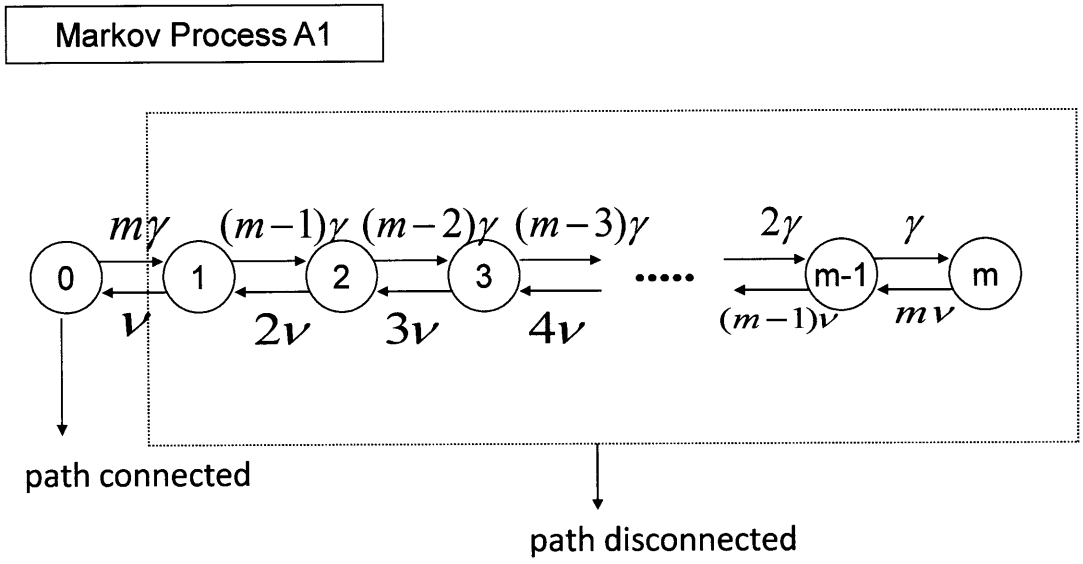

Figure C-1: Markov model for m identical links on one path

for the FIRST time. Again, we use the trick that makes State 0 a trapping state (see Markov Process A-2 in Figure C-2), and solve for $P_{10}(t)$ of the $[Q]$ matrix of of MP-A2, denoted by $P'_{10}(t)$.

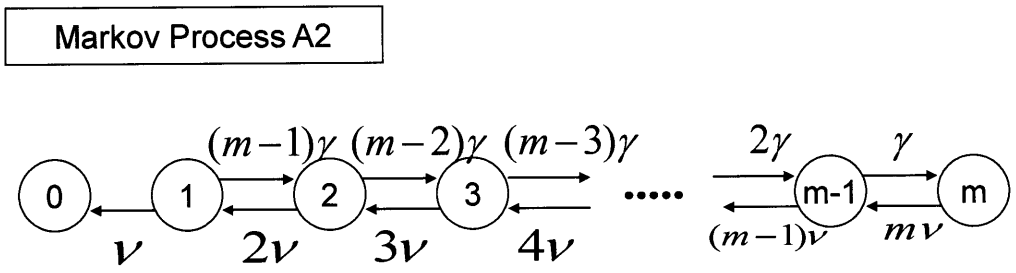

Figure C-2: $m+1$-state Markov model for a single path consisting of m independent links

For MP-A2, we still have the following Kolmogrov backward differential equation

$$\frac{d[P(t)]}{dt} = [Q][P(t)]; t \geq 0 \tag{C.2}$$

In MP-A2, $[P(t)]$ and $[Q(t)]$ are now defined as

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ \nu & -\gamma-\nu & \gamma \\ 0 & 2\nu & -2\nu \end{bmatrix} \tag{C.3}$$

The eigenvalues of the $[Q]$ matrix of MP-A2, λ'_0, λ'_1 and λ'_2 are

$$\begin{aligned} \lambda'_0 &= 0 \\ \lambda'_1 &= -\frac{3}{2}\nu - \frac{1}{2}\gamma + \frac{1}{2}\sqrt{\gamma^2+6\gamma\nu+\nu^2} \\ \lambda'_2 &= -\frac{3}{2}\nu - \frac{1}{2}\gamma - \frac{1}{2}\sqrt{\gamma^2+6\gamma\nu+\nu^2} \end{aligned} \tag{C.4}$$

The corresponding eigenvectors of $[Q]$ are

$$\begin{aligned} \mathbf{v'_0} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \\ \mathbf{v'_1} &= \begin{bmatrix} 0 & \frac{1}{4\nu}(\nu-\gamma+\sqrt{\gamma^2+6\gamma\nu+\nu^2}) & 1 \end{bmatrix}^T \\ \mathbf{v'_2} &= \begin{bmatrix} 0 & \frac{1}{4\nu}(\nu-\gamma-\sqrt{\gamma^2+6\gamma\nu+\nu^2}) & 1 \end{bmatrix}^T \end{aligned} \tag{C.5}$$

and for simplicity, we denote $\mathbf{v'_1}(2)$ by a', and $\mathbf{v'_2}(2)$ by b'. Plugging in the initial conditions for $[P(t)]$, we get the following solution for $P'_{10}(t)$

$$P'_{10}(t) = 1 + c'_1 a' \exp(\lambda'_1 t) + c'_2 b' \exp(\lambda'_2 t) \tag{C.6}$$

where

$$\begin{aligned} c'_1 &= \frac{b'-1}{a'-b'} \\ c'_2 &= \frac{1-a'}{a'-b'} \end{aligned} \tag{C.7}$$

On the other hand, we may aggregate States $1, 2, 3, \ldots, m$ into one state called *P-disconnected*, representing the fact that the path between sender and the receiver is disconnected; we also keep State 0 as one single state, *P-connected*, representing

the fact that the path between sender and the receiver is connected. We denote this two-state Markov process by MP-A3 with transition rates are γ_{pa} and ν_{pa}, as shown in Figure C-3. MP-A3 is an approximation to MP-A1.

Now we need to find the state holding time's distribution of MP-A3. We still

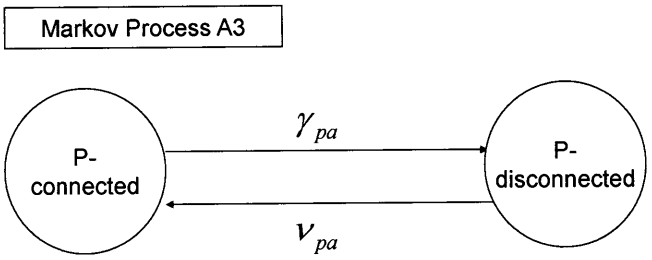

Figure C-3: Modification to Markov Process A1 by making State 0 a trapping state

start from the steady state probability of the two states, π_{pac} and π_{pad}:

$$\begin{aligned} \pi_{pac} &= \pi_u^2 \\ &= (\frac{\nu}{\gamma+\nu})^2 \\ \pi_{pad} &= 1 - \pi_{pac} \\ &= \frac{\gamma^2 + 2\gamma\nu}{(\gamma+\nu)^2} \end{aligned} \tag{C.8}$$

For the same reason as we discussed in Section 3.2, the state holding time of the *P-connected* state, T_{pac}, has the same exponential distribution as T_{pc} in Markov Process A1,

$$F_{T_{pac}}(t) = 1 - \exp(-2\gamma t) \tag{C.9}$$

and

$$E[T_{pac}] = \frac{1}{2\gamma}. \tag{C.10}$$

Because the ratio of expected time spent in each of the two states is equal to the ratio of the steady state probabilities of the two states,

$$\frac{E[T_{pac}]}{E[T_{pad}]} = \frac{\pi_{pac}}{\pi_{pad}} \tag{C.11}$$

hence we get the expression for $E[T_{pad}]$ as

$$E[T_{pad}] = \frac{1}{2\gamma} \cdot \frac{\pi_{pad}}{\pi_{pac}} \tag{C.12}$$

In MP3, the disconnected time T_{pad} is exponentially distributed with rate ν_{pa}, thus

$$\begin{aligned} \nu_{pa} &= \frac{1}{E[T_{pad}]} \\ &= \frac{2\nu^2}{\gamma + 2\nu} \\ &\to \nu \quad \text{as } \frac{\gamma}{\nu} = \rho \to 0 \end{aligned} \tag{C.13}$$

Thus we write the distribution functions of the state holding time T_{pac} and T_{pad} as:

$$F_{T_{pac}}(t) = 1 - \exp(-2\gamma t) \tag{C.14}$$

$$F_{T_{pad}}(t) \to 1 - \exp(-\nu t) \quad \text{as } \frac{\gamma}{\nu} = \rho \to 0 \tag{C.15}$$

If we can prove that the distribution of T_{pd} converges to the same form when $\rho \to 0$, we may say that the state aggregation is reasonable and will provide satisfactory accuracy when ρ is small, and the TCP throughput calculated from both models will be close in value as well.

Analyzing the terms of $F_{T_{pd}}(t)$ with $\rho = \frac{\gamma}{\nu} \ll 1$, we know that

$$\begin{aligned} a' &= \frac{1}{4\nu}(\nu - \gamma + \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \\ &= \frac{1}{4\nu}\{\nu - \gamma + \nu[1 + \frac{1}{2}(6\rho + \rho^2) + o(\rho^2)]\} \\ &= \frac{1}{4\nu}[\nu - \gamma + \nu + 3\gamma + o(\rho^2)] \\ &= \frac{1}{2} + \frac{\rho}{2} + o(\rho^2)) \end{aligned} \tag{C.16}$$

and

$$\begin{aligned} b' &= \frac{1}{4\nu}(\nu - \gamma - \sqrt{\gamma^2 + 6\gamma\nu + \nu^2}) \\ &= \frac{1}{4\nu}\{\nu - \gamma - \nu[1 + \frac{1}{2}(6\rho + \rho^2) + o(\rho^2)]\} \\ &= \frac{1}{4\nu}[\nu - \gamma - \nu - 3\gamma + o(\rho^2)] \\ &= -\rho + o(\rho^2) \end{aligned} \tag{C.17}$$

Thus,

$$\begin{aligned} c_1' &= \frac{b' - 1}{a' - b'} \\ &= \frac{-(1+\rho) + o(\rho^2)}{\frac{1}{2}(1 + 3\rho)} \\ &\to -2 \quad \text{as } \rho \to 0 \end{aligned} \tag{C.18}$$

and

$$\begin{aligned} c_2' &= \frac{1 - a'}{a' - b'} \\ &= \frac{1 - \rho}{1 + 3\rho} + o(\rho^2) \\ &\to 1 \quad \text{as } \rho \to 0 \end{aligned} \tag{C.19}$$

Now we turn to analyze the eigenvalues of the $[Q]$ matrix of MP-A2.

$$\begin{aligned} \lambda_1 &= -\frac{3}{2}\gamma - \frac{1}{2}\nu + \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2} \\ &= -\frac{3}{2}\nu - \frac{1}{2}\gamma + \frac{\nu}{2}[1 + \frac{1}{2}(6\rho + \rho^2) + \frac{1}{8}36\rho^2 + o(\rho^2)] \\ &\to -\nu \quad \text{as } \rho \to 0 \end{aligned} \tag{C.20}$$

$$\begin{aligned} \lambda_2 &= -\frac{3}{2}\nu - \frac{1}{2}\gamma + \frac{1}{2}\sqrt{\gamma^2 + 6\gamma\nu + \nu^2} \\ &= -\frac{3}{2}\nu - \frac{1}{2}\gamma - \frac{\nu}{2}[1 + \frac{1}{2}(6\rho + \rho^2) + \frac{1}{8}36\rho^2 + o(\rho^2)] \\ &\to -2\nu \quad \text{as } \rho \to 0 \end{aligned} \tag{C.21}$$

If we plug all these parameters into the expression for $F_{T_{pd}(t)}$, or $P'_{10}(t)$, we see that

$$F_{T_{pd}}(t) \to 1 - \exp(-\nu t) + \exp(-2\nu t) \quad \text{as } \rho \to 0 \tag{C.22}$$

We see that $F_{T_{pd}}(t)$ is dominated by the first exponential term.

We notice that the expression of $F_{T_{pac}}(t)$ is exactly the same as that of $F_{T_{pc}}(t)$, and the expressions of $F_{T_{pd}}(t)$ and $F_{T_{pad}}(t)$ both approach to $1 - \exp(-\nu t)$ as $\rho \to 0$, or equivalently, as the ratio $\frac{\nu}{\gamma}$ becomes large. In real life systems, the ratio $\frac{\nu}{\gamma}$ is usually at around 100 or higher, and we may conclude that state-aggregation is a good model to approximate the original $(m+1)$-state model. In Figure C-4, we plot $F_{T_{pad}}(t)$ and $F_{T_{pd}}(t)$ when $\gamma = 1s^{-1}$ and $\nu = 100s^{-1}$. We see that the two cumulative distribution functions are very close to each other. Thus, we can approximate the connected and the disconnected time distribution of a path that consists of m identical stochastic links by the state-holding times of a two-state Markov process.

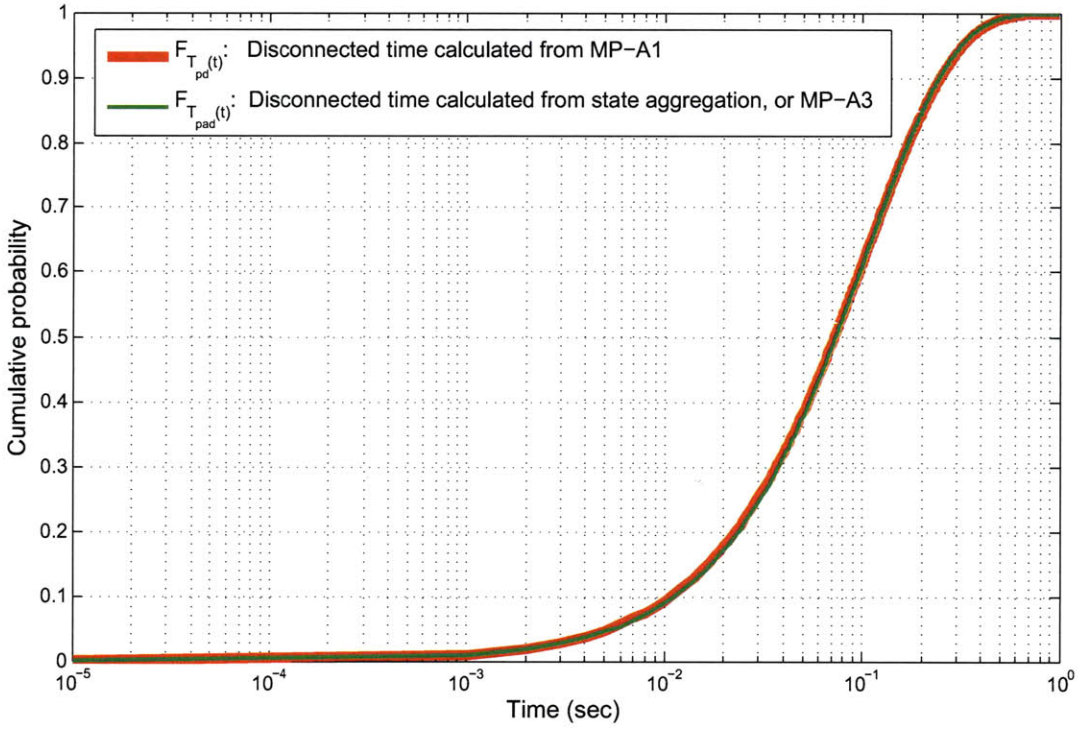

Figure C-4: Cumulative distribution functions for $F_{T_{pad}}(t)$ and $F_{T_{pd}}(t)$, plotted in semi-log scale. The conformity of the two distributions shows that the state aggregation is a good approximation to the original model.

Appendix D

Transcendental equation to solve for n^*

In this appendix, we carry out the partial derivative in Equation (3.44) and display the transcendental equation that can be used to solve for n^*. Note that we assume $M = W_m$.

$$
\begin{aligned}
&-\frac{1}{n^2 W_m(\frac{1}{n\nu}+\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)}{n\nu})}\cdot \\
&[\frac{\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(2(\gamma+\nu)^n+\gamma^n(-2+nRTT(-1+W_m)\nu))}{2nRTT\nu} \\
&-\frac{1}{2n^2RTT^2\nu^2}\cdot \\
&(\gamma^{-2n}(-2(\gamma+\nu)^{2n}-\gamma^{2n}(2+nRTT\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu) \\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})(2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu) \\
&+\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu)))))] \\
&\frac{1}{nW_m(\frac{1}{n\nu}+\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)}{n\nu})^2}\cdot \\
&\{\frac{\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(2(\gamma+\nu)^n+\gamma^n(-2+nRTT(-1+W_m)\nu))}{2nRTT\nu} \\
&-\frac{\gamma^{-2n}}{2n^2RTT^2\nu^2}[-2(\gamma+\nu)^{2n}-\gamma^{2n}(2+nRTT\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu) \\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})(2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)
\end{aligned}
$$

$$
\begin{aligned}
&+\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu)))]\}\cdot\\
&[-\frac{1}{n^2\nu}-\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)}{n^2\nu}-\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)\ln(\gamma)}{n\nu}\\
&+\frac{\gamma^{-n}(-\gamma^n\ln(\gamma)+(\gamma+\nu)^n\ln(\gamma+\nu))}{n\nu}]\\
&+\frac{1}{nW_m(\frac{1}{n\nu}+\frac{\gamma^{-n}(-\gamma^n+(\gamma+\nu)^n)}{n\nu})}\cdot\\
&\{-\frac{\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(2(\gamma+\nu)^n+\gamma^n(-2+nRTT(-1+W_m)\nu))}{2n^2RTT\nu}\\
&+\frac{\gamma^{-2n}}{n^3RTT^2\nu^2}[-2(\gamma+\nu)^{2n}-\gamma^{2n}(2+nRTT\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu)\\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})(2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)\\
&+\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu)))]\\
&-\frac{\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(2(\gamma+\nu)^n+\gamma^n(-2+nRTT(-1+W_m)\nu))\ln(\gamma)}{2nRTT\nu}\\
&+\frac{1}{n^2RTT^2\nu^2}\cdot\\
&\gamma^{-2n}[-2(\gamma+\nu)^2n-\gamma^{2n}(2+nRTT\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu)\\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})\cdot\\
&(2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)\\
&+\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu)))]\cdot\ln(\gamma)\\
&+\frac{1}{2nRTT\nu}\cdot\\
&[\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})W_m\gamma^{-n}(RTT(-1+W_m)\gamma^n\nu+\gamma^n(-2+nRTT(-1+W_m)\nu)\ln(\gamma)\\
&+2(\gamma+\nu)^n\ln(\gamma+\nu))]\\
&-\frac{1}{2n^2RRT^2\nu^2}\gamma^{-2n}(-RTT\gamma^{2n}\nu+RTT\gamma^n\nu(\gamma+\nu)^n-2\gamma^{2n}(2+nRTT\nu)\ln(\gamma)\\
&\gamma^n(\gamma+\nu)^n(4+nRTT\nu)\ln(\gamma)-4(\gamma+\nu)^{2n}\ln(\gamma+\nu)+\gamma^n(\gamma+\nu)^n(4+nRTT\nu)\ln(\gamma+\nu)\\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})[RTT(-1+2W_m)\gamma^n\nu(\gamma+\nu)^n+\gamma^{2n}(nRTT^2(-1+W_m)W_m\nu^2\\
&+RTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu))+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)\ln(\gamma)\\
&+2\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu))\ln(\gamma)+4(\gamma+\nu)^{2n}\ln(\gamma+\nu)\\
&+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)\ln(\gamma+\nu)]\\
&+\exp(\frac{nRTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n})(2(\gamma+\nu)^{2n}+\gamma^n(\gamma+\nu)^n(-4+nRTT(-1+2W_m)\nu)\\
&+\gamma^{2n}(2+nRTT\nu(1-2W_m+nRTT(-1+W_m)W_m\nu)))\cdot
\end{aligned}
$$

$$
\begin{aligned}
& (\frac{RTTW_m\gamma^n\nu}{\gamma^n-(\gamma+\nu)^n}+\frac{nRTTW_m\gamma^n\nu\ln(\gamma)}{\gamma^n-(\gamma+\nu)^n}+\frac{nRTTW_m\gamma^n\nu\ln(\gamma)}{\gamma^n-(\gamma+\nu)^n} \\
& -\frac{nRTTW_m\gamma^n\nu(\gamma^n\ln(\gamma)-(\gamma+\nu)^n\ln(\gamma+\nu))}{(\gamma^n-(\gamma+\nu)^n)^2}))\} \\
= \; & 0 \qquad \text{(D.1)}
\end{aligned}
$$

Bibliography

[1] I.F. Akyildiz, G. Morabito, and S. Palazzo. TCP-Peach: a new congestion control scheme for satellite IP networks. *Networking, IEEE/ACM Transactions on*, 9(3):307 –321, June 2001.

[2] L.S. Brakmo and L.L. Peterson. TCP Vegas: end to end congestion avoidance on a global Internet. *Selected Areas in Communications, IEEE Journal on*, 13(8):1465 –1480, October 1995.

[3] J.P. Choi and V.W.S. Chan. Adaptive communications over fading satellite channels. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 9, pages 2635 –2639 vol.9, June 2001.

[4] Cheng Peng Fu and S.C. Liew. TCP Veno: TCP enhancement for transmission over wireless access networks. *Selected Areas in Communications, IEEE Journal on*, 21(2):216 – 228, February 2003.

[5] R. G. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Norwell, Massachusetts, 1996.

[6] R. G. Gallager. *Principles of Digital Communication.* Cambridge University Press, New York, New York, 2008.

[7] M. Gerla, M.Y. Sanadidi, Ren Wang, A. Zanella, C. Casetti, and S. Mascolo. TCP Westwood: congestion window control using bandwidth estimation. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3, pages 1698 –1702 vol.3, November 2001.

[8] V. Jacobson, R.Braden, and D. Borman. TCP Extensions for High Performance. *Internet Engineering Task Force*, May 1992.

[9] F. Lefevre and G. Vivier. Understanding TCP's behavior over wireless links. In *Communications and Vehicular Technology, 2000. SCVT-200. Symposium on*, pages 123 –130, 2000.

[10] T.K. Saini, M. Kumar, and M.K. Dhaka. Experimental Validation of TCP Performance in GEO Satellite link Conditions. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 597 –602, June 2009.

[11] Ye Tian, Kai Xu, and N. Ansari. TCP in wireless environments: problems and solutions. *Communications Magazine, IEEE*, 43(3):S27 – S32, March 2005.

[12] Kai Xu, Ye Tian, and N. Ansari. TCP-Jersey for wireless IP communications. *Selected Areas in Communications, IEEE Journal on*, 22(4):747 – 756, May 2004.

[13] Qinqing Zhang and S.A. Kassam. Finite-state Markov model for Rayleigh fading channels. *Communications, IEEE Transactions on*, 47(11):1688 –1692, November 1999.

[14] M. Zorzi, R.R. Rao, and L.B. Milstein. On the accuracy of a first-order Markov model for data transmission on fading channels. In *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, pages 211 –215, June 1995.