PERFORMANCE EVALUATION OF CLASSICAL AND QUANTUM COMMUNICATION SYSTEMS

A Dissertation Presented

by

GAYANE VARDOYAN

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

DOCTOR OF PHILOSOPHY

August 2020

College of Information and Computer Sciences

PERFORMANCE EVALUATION OF CLASSICAL AND QUANTUM COMMUNICATION SYSTEMS

A Dissertation Presented

by

GAYANE VARDOYAN

Approved as to style and content by:

Don Towsley, Chair

Kris Hollot, Member

James Kurose, Member

Arya Mazumdar, Member

Saikat Guha, Member

James Allan, Chair
College of Information and Computer Sciences

DEDICATION

To my grandmother, Rima.

ACKNOWLEDGMENTS

Thanks to all those who have contributed to my incredible journey of graduate school, by providing support, friendship, academic advice, and various other forms of enrichment – without you, this experience would not have been nearly as gratifying.

I am deeply indebted to my advisor, Don Towsley – I could not have imagined a better person to guide me through graduate school and continuously present me with so many wonderful opportunities. I would be content to someday become just a fraction of the kind of advisor, mentor, lecturer, and researcher that he is.

I am very grateful to my thesis committee members Jim Kurose, Saikat Guha, Kris Hollot, and Arya Mazumdar. Thank you for the immense amount of support you have given me. It has been a great pleasure to collaborate with Kris and Saikat, and to revel in Jim and Arya's teaching and lecturing skills; thanks to all of you for investing your time in my education – both as a student and a researcher.

I would like to thank my labmates throughout the years for their support at every important milestone during grad school. Fabricio Murai, Kun Tu, James Atwood, Bo Jiang, Jian Li, Chang Liu, Nitish Panigrahy, Amir Reza Ramtin, Stefan Dernbach, Arman Mohseni-Kabir, Albert Williams, Thirupathaiah Vasantam, and Janice Chen – they came to all my practice talks, whether for a conference, job talk, or thesis-related, and truly helped me become a better speaker. I feel very lucky to have had this thoughtful team of people come to my aid when I needed it most.

I would like to thank all my research collaborators for their patience and time. Thanks to Nagi Rao for many hours spent collecting measurement data. Special thanks to Philippe Nain – I have learned, and continue to learn so much from him.

His mathematical and analytical skills are truly impressive, and just when I think I've seen it all, he continues to amaze me.

I thank my family for being patient with me as I spent many years away pursuing an academic career. Thanks to my parents Irina and Sargis who started all from scratch, twice, to give us better opportunities. Thanks especially to my mother and my sister, Anaid, for being an inspiration. Thanks to my grandmother Rima for her unconditional love. Her story is an inspiration to all of us, and I dedicate this thesis to her. Thanks to my partner Subhransu for being there in the toughest and happiest of times, and for nourishing me with amazing home-made meals when I had no time to worry about food. He is not only my best friend but also a role model. I thank the people who built the volleyball court next to Soda Hall in Berkeley where I met him. Thanks also to Subhransu's parents, Nilima and Proshanto, for cheering me on in my academic pursuits. Last but not least, thanks to Kitty for being the lovable, furry, affectionate, and comical animal that has brought us so much joy.

Finally, I thank the many others – colleagues and friends – who gave me the encouragement and confidence that I needed. Special thanks to Barna Saha, Erik Learned-Miller, Laura Haas, Joydeep Biswas, Eleanor Avrunin, Yuriy Brun, Alexandra Meliou, Akshay Krishnamurthy, Justin Domke, Adam O'Neill, Walter Krawec, Jay Taneja, Emily Kumpel, Colin Gleason, Binglei Gong, Phil Thomas, Sarah Bourbeau, Rui Wang, and Rajesh Bhatt for believing in me and in general for being the wonderful human beings that you are. Thanks also to the heroic staff members of CICS – Laurie Connors, Barbara Sutherland, Joyce Mazeski, and Vickie Rupp – for tirelessly supporting us all.

ABSTRACT

PERFORMANCE EVALUATION OF CLASSICAL AND QUANTUM COMMUNICATION SYSTEMS

AUGUST 2020

GAYANE VARDOYAN

B.Sc., UNIVERSITY OF CALIFORNIA, BERKELEY

M.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Don Towsley

The Transmission Control Protocol (TCP) is a robust and reliable method used to transport data across a network. Many variants of TCP exist, *e.g.*, Scalable TCP, CUBIC, and H-TCP. While some of them have been studied from empirical and theoretical perspectives, others have been less amenable to a thorough mathematical analysis. Moreover, some of the more popular variants had not been analyzed in the context of the high-speed environments for which they were designed. To address this issue, we develop a generalized modeling technique for TCP congestion control under the assumption of high bandwidth-delay product. In a separate contribution, we develop a versatile fluid model for congestion-window-based and rate-based congestion controllers that can be used to analyze a protocol's stability. We apply this model to CUBIC – the default implementation of TCP in Linux systems – and discover that under a certain loss probability model, CUBIC is locally asymptotically stable. The

contribution of this work is twofold: (i) the first formal stability analysis of CUBIC, and (ii) the fluid model can be easily adapted to other protocols whose window or rate functions are difficult to model. We demonstrate another application of this model by analyzing the stability of H-TCP, another popular variant used in data science networks.

On a different front, a wide range of quantum distributed applications, which either promise to improve on existing classical applications or offer functionality that is entirely unobtainable via classical means, are helping to fuel rapid technological advances in the area of quantum communication. In view of this, it is prudent to model and analyze quantum networks, whose applications range from quantum cryptography to quantum sensing. Several types of quantum distributed applications, such as the E91 protocol for quantum key distribution, make use of entanglement to meet their objectives. Thus, being able to distribute entanglement efficiently is one of the most important and fundamental tasks that must be performed in a quantum network – without this functionality, many quantum distributed applications would be rendered infeasible. Modeling such systems is vital in order to better conceptualize their operation, and more importantly, to discover and address the challenges involved in actualizing them. To this end, we explore the limits of star-topology entanglement switching networks and introduce methods to model the process of entanglement generation, a set of switching policies, memory constraints, link heterogeneity, and quantum state decoherence for a switch that can serve bipartite (and in a specific case, tripartite) entangled states. In one part of this work, we compare two modeling techniques: discrete time Markov chains (DTMCs) and continuous-time Markov chains (CTMCs). We find that while DTMCs are a more accurate way to model the operation of an entanglement distribution switch, they quickly become intractable when one introduces link heterogeneity or state decoherence into the model. In terms of accuracy, we show that not much is lost for the case of homogeneous links, infi-

nite buffer and no decoherence when CTMCs are employed. We then use CTMCs to model more complex systems. In another part of this work, we analyze a switch that can store one or two qubits per link and can serve both bipartite and tripartite entangled states. Through analysis, we discover that randomized policies allow the switch to achieve a better capacity than time-division multiplexing between bipartite and tripartite entangling measurements, but the advantage decreases as the number of links grows.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

INTRODUCTION

This thesis covers two main topics: the first concerns TCP and its performance evaluation, and the second concerns entanglement switching in a star-topology quantum network. Congestion control serves a vital function in the efficient operation of computer networks. Several congestion control variants have been developed, but detailed analytical models for some of the newer variants are scarce. A notable example is TCP CUBIC, which is currently the default variant used in Linux kernels. While there are empirical and comparative studies of several TCP variants, they are not enough to make careful statements about a protocol's performance and stability. To address this problem, we develop a novel generalized framework for TCP throughput prediction. Among other factors, our model takes into account buffer constraints, file size, multiple parallel flows, and maximum congestion window constraints imposed by the receiver. The latter is a novel and important component of the framework, as maximum window constraints come into play in high bandwidth-delay product environments and can make the difference between a good throughput prediction and an inaccurate one.

Stability is another important property of congestion control algorithms. Two of the most popular TCP variants, CUBIC and H-TCP, had not been formally analyzed from a stability perspective. This is partly due to the lack of a suitable modeling framework within which complex congestion controllers can be analyzed. For CUBIC, there exists an additional difficulty in that its fixed point lies at the saddle point of its congestion window function, rendering linearization of the model useless. To overcome these challenges, we develop a new model consisting of a system of two time-delayed differential equations, which we call the MWLI model. The novelty of

this model is that it describes the evolution of two essential components of any loss-based congestion controller: the time since last loss, and the size of the congestion window immediately before a loss occurs. This is in contrast to traditional methods, which model the evolution of the congestion window directly. For complex congestion controllers, the MWLI model succeeds where the traditional methods fail.

The area of quantum communication is relatively new, and certain aspects have not been subjected to mathematical analysis. A vital resource in a quantum network is end-to-end entanglement between sets of users, which can be used to support a number of applications, including quantum key distribution. We consider a single quantum entanglement distribution switch and model it as a Markov process to determine its achievable capacity and quantum memory occupancy for bipartite and tripartite entanglement distribution. One of the first questions posed in this work is model choice: *i.e.*, whether to use discrete- or continuous-time Markov chains to represent and analyze the device. We compare the two model types for a simple scenario and find that while the discrete-time model yields more accurate results, it cannot easily be extended to more complex settings. For instance, attempting to model a system with heterogeneous links or quantum state decoherence quickly proved to be intractable with the discrete model. On the other hand, such extensions are easy to incorporate into a continuous-time Markov chain, and in the simple scenario of identical links, infinite quantum memories, and no state decoherence, not much is lost in terms of accuracy when compared to the discrete model. Our continuous-time model takes into account link heterogeneity, memory constraints, measurement success probability, and decoherence. The results may be used to guide the design of quantum switches or similar devices (*e.g.*, in determining how much quantum memory is a reasonable amount) or they may serve as a useful benchmark for choosing efficient switching policies.

CHAPTER 1

PERFORMANCE EVALUATION OF TCP

1.1 Introduction

The congestion collapse in the ARPANET during the late 1980s prompted the adoption of what is now referred to as legacy Transmission Control Protocol (TCP). Later, the advent of high-BDP (Bandwidth-Delay Product) networks stimulated the development of several TCP variants to overcome the inefficiencies of legacy TCP. These include HSTCP (HighSpeed TCP) [18], FAST (FAST AQM Scalable TCP) [74], BIC (Binary Increase Congestion control) TCP [75], STCP (Scalable TCP) [41], CUBIC TCP [30], and H-TCP (Hamilton TCP) [47]. To an extent, all of these Congestion Avoidance (CA) algorithms are successful in improving bandwidth utilization, although some exhibit better fairness, friendliness, and convergence properties than others. Several empirical studies of these algorithms exist ([38, 48, 28, 49, 9], among others), and although CUBIC is currently the default in Linux kernels, as of yet, there is no definitive consensus on which CA algorithm is best.

Adding to this uncertainty, many other studies rely either on simulations or experiments on networks with relatively low bandwidths compared to current High-Speed Network (HSN) environments. Today, 10 Gbps Wide Area Network (WAN) links are not uncommon between large-scale datacenters and computing facilities. An example is the Extreme Science and Engineering Discovery Environment (XSEDE) [71], a collection of supercomputing resources spanning several universities, national labs, and other research institutions, all interconnected by 10 Gbps (and sometimes faster) links. XSEDE offers tools like Globus GridFTP [1] and *sftp* for data transfers (note

that FTP relies on TCP for reliable data movement), and such bulk data transfer mechanisms are becoming increasingly popular in HSNs.

Since BDPs have grown significantly and new bulk data transfer protocols that use parallel TCP streams have been introduced, it is important to reevaluate TCP both analytically and empirically. In this work, we present a framework within which we derive analytical models for TCP variants. Unlike prior work, where models are studied in isolation for each TCP variant, our framework is not variant-specific and can be used to model a range of loss-based TCP variants. We use measurements on an experimental testbed to motivate and validate these models.

In some environments, researchers share WAN links for data transfers. The drawback is that TCP streams belonging to different users may compete for bandwidth and other resources. An alternate method of transferring data is through the use of virtual circuits (VCs) dedicated to a single user or application. One example is the On-Demand Secure Circuits and Advance Reservation System (OSCARS) [27], which allows users to reserve high-bandwidth VCs for guaranteed performance. According to the Energy Sciences Network, more than 50 research networks deploy OSCARS, including the US LHC Network; and OSCARS VCs carry half of the Department of Energy's (DOE) science traffic. The increasing popularity of VC-based data transfer options serves as excellent motivation to study TCP behavior in a controlled setting. Moreover, it provides an incentive to examine scenarios with only a fixed, rather than a dynamically changing, number of TCP connections.

Most protocols that strive to achieve efficient bulk data transfers do so by providing features that allow users to tune them as they see fit. For example, GridFTP supports both parallelism and concurrency (parallel streams use one socket and concurrent streams use separate sockets; henceforth, we use these terms interchangeably). It has become clear that the reason for the emergence of tools like GridFTP is the fact that TCP is not able to keep up with the demands of today's high-BDP networks.

In this work, we develop a framework that enables us to study the performance of protocols like GridFTP in dedicated high-BDP networks. To do so, we concentrate on a possible underlying root cause of poor bandwidth utilization: the congestion control (CC) algorithms used by all TCP-based data transfer tools.

We take two approaches in an attempt to understand the dynamics of TCP in modern environments: (i) first-principled modeling and (ii) measurement-based. We create robust, detailed analytical models of variants of the protocol. At the same time, we evaluate these variants on a dedicated network link, in a controlled setting that allows us to emulate diverse experimental configurations while removing any interference (*e.g.*, I/O, background traffic) that could obscure TCP's intended behavior. We collect detailed measurements of memory-to-memory transfers on two different testbed configurations using *iperf* and *tcpprobe*, for three different TCP variants.

The main contributions of this work are:

– A general and comprehensive framework for modeling a diverse set of congestion control algorithms. The framework encompasses not only congestion avoidance, but also the slow-start mechanism. The latter takes into consideration the heuristic guidelines imposed by Hybrid slow-start (HyStart) [29], which is the implementation of slow-start used in current Linux kernels.

– A validation of these models using an extensive set of measurements.

– Last, we observe from our measurements that (i) CUBIC and H-TCP are comparable in terms of average throughput, while they both outperform STCP, and (ii) TCP performance benefits from the presence of a well-designed physical layer (*e.g.*, SONET).

We use only first principles to model the performance of each TCP variant in terms of its average throughput as a function of round trip time (RTT). The models also accept link capacity, buffer size, transfer size, number of parallel streams, and maxi-

mum congestion window (*cwnd*) size as parameters. Using measurements, we validate the models and compare the performance of the TCP variants. *To our knowledge, this work constitutes the first careful measurement-based modeling study of TCP congestion control algorithms in a high-BDP/HSN setting.*

The remainder of this chapter is organized as follows. We describe the three variants and related work in Section 1.2. In Section 1.3, we describe the testbed, measurement collection, and first-hand observations from the collected data. In Section 1.4, we delve into the analytical framework for slow-start, congestion avoidance, and each of the variants separately, presenting closed-form expressions for sending rate where possible. In Section 1.5, we present and validate our results. Finally, we conclude the chapter in Section 1.6.

1.2 Background

1.2.1 TCP Variants

We study CUBIC because it is the most commonly used variant in current HSN networks and the default CA algorithm in the Linux kernel. Unlike most TCP variants, CUBIC is not an acknowledgement (ACK)-based algorithm. Instead, CUBIC's *cwnd* is a cubic function of time since the last congestion event such that the inflection point is the maximum window size immediately before the most recent loss occurred.

We also study STCP because it was developed within the optimization-based framework proposed in [40]. STCP is a multiplicative increase, multiplicative decrease (MIMD) [1] algorithm with the following response functions:

$$cwnd \leftarrow cwnd + a$$

[1]Note that although the per-ACK update rule for STCP is additive, this CA algorithm is MIMD at the RTT level.

for every ACK received, where the increase factor a is usually set to 0.01. Upon loss detection,

$$cwnd \leftarrow b \times cwnd.$$

Usually, $b = 0.875$ for STCP.

Finally, we study H-TCP because of its favorable fairness and convergence properties [48]. H-TCP is an ACK-based generalized additive increase multiplicative decrease (AIMD) algorithm whose additive increase factor a is a function of the time t since the last congestion event. Specifically, a is defined as:

$$a \leftarrow 2(1-b)a(t)$$

where $a(t)$ is

$$a(t) = \begin{cases} 1 & t \leq \Delta^L \quad (1.1a) \\ 1 + 10(t - \Delta^L) + \left(\frac{t - \Delta^L}{2}\right)^2 & t > \Delta^L \quad (1.1b) \end{cases}$$

Δ^L is usually set to one second so that for small congestion epochs, H-TCP behaves like standard TCP. H-TCP's decrease factor b is defined as

$$b \leftarrow \frac{RTT_{min}}{RTT_{max}},\ b \in [0.5, 0.8]$$

where RTT_{min} and RTT_{max} are the minimum and maximum measured RTTs of a flow. Upon loss detection, *cwnd* is updated as follows:

$$cwnd \leftarrow b \times cwnd.$$

1.2.2 Related Work

There exist a number of analytical studies for modelling TCP. Kelly proposed an optimization-based framework for studying and designing CA algorithms in [40], where STCP was an output. In [69], Srikant presented a simple analysis of Jacobson's TCP CC algorithm. The derivation sets the maximum window constraint to the sum of the BDP and the size of the buffer. Our analysis is more refined in that it takes into consideration two different maximum window constraints, as discussed in Section 1.4. A model for slow-start is also presented in [69]. We extend this model by considering the latest version of slow-start currently in use by Linux kernels.

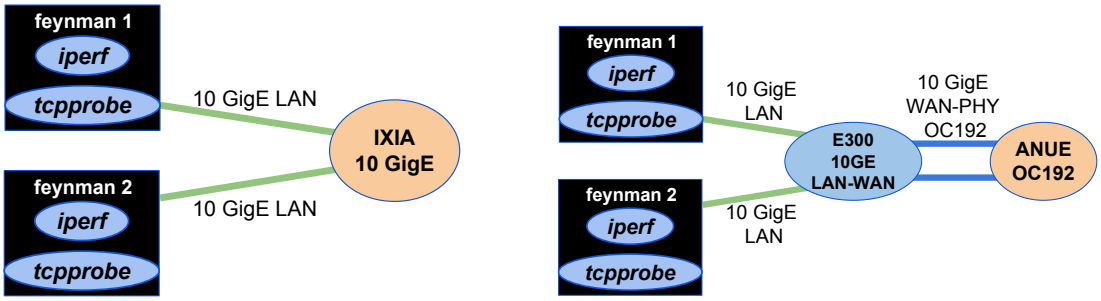
In [50], Misra *et al.* model TCP throughput using stochastic differential equations. El Khoury *et al.* [14] present a model for STCP that includes buffer size as a parameter, but only in the case of a very small buffer. In addition, they rely only on *ns-2* simulations for validation. Bao *et al.* propose Markov chain models for average CUBIC throughput, but for a wireless environment [2]. Moreover, they do not directly account for buffer constraints. Leith *et al.* present empirical evidence that H-TCP fares well in bandwidth utilization compared to other TCP variants [46], but the protocol's CC dynamics have not been analyzed in-depth.

There are some empirical studies that explore the behavior of TCP with multiple concurrent flows. Morris looks at a number of performance metrics using simulations and real packet traces, but does not explore different TCP variants [52]. Yu *et al.* compare the performance of three open-source big data transfer protocols in [76] using memory-to-memory transfers on a 10 Gbps international HSN. Bateman *et al.* compare different TCP variants for fairness at high speeds using *ns-2* and Linux [3]. As far as we know, no previous work attempts to model TCP with multiple flows using first principles.

1.3 Measurements

1.3.1 Emulation Testbed

Our testbed consists of two types of Linux hosts: 32-core and 48-core HP ProLiant servers, each with Broadcom 10 GigE NICs, running Linux 2.6 kernel (CentOS release 6.6). It also consists of ANUE OC192 and IXIA 10 GigE hardware connection emulators, and a 10 Gbps Force10 E300 WAN-LAN switch. Two separate configurations are utilized for 10 GigE and SONET measurements. These hardware connection emulators transport the physical packets between hosts, delaying them during transit by an amount specified at configuration. This process closely matches the effects of physical connections, particularly, the TCP dynamics of hosts connected to them, which in turn determine the throughput rates achieved. They more closely capture the real-time TCP dynamics compared to packet-level simulators (such as *ns-3* and *OPnet*) that are typically driven by discrete "packet delivery events".



(a) 10 GigE (b) OC192

Figure 1.1: Two testbed configurations for dedicated 10 Gbps connections.

We consider two configurations that use OC192 and 10 GigE physical layer modalities. In the first configuration, 10 GigE NICs of host systems are directly connected to two IXIA emulator ports as shown in Figure 1.1a. Long range 1250 nm optical transceivers are used on host NICs to communicate directly with the emulator. The SONET configuration shown in Figure 1.1b is more complicated compared to the 10 GigE case, since hosts are equipped with 10 GigE NICs and do not support SONET. In this case, the 10 GigE NICs are connected to a Force10 E300 switch

using multi-mode 850 nm optical transceivers in LAN-PHY mode. The E300 switch converts between 10 GigE LAN-PHY and WAN-PHY frames that are inter-operable with OC192 frames; the latter are sent through its WAN ports, which are directly connected to OC192 ANUE emulator ports as shown in Figure 1.1b. We note that the peak capacity of an OC192 connection is 9.6 Gbps. We utilize these emulators to collect TCP measurements for a suite of dedicated connections where we set the RTT to 11.8, 22.6. 45.6, 91.6, 183 and 366 ms. RTTs in the mid-range represent US cross-country connections, for example, ones between DOE sites provisioned using the OSCARS system. Higher RTTs represent transcontinental connections.

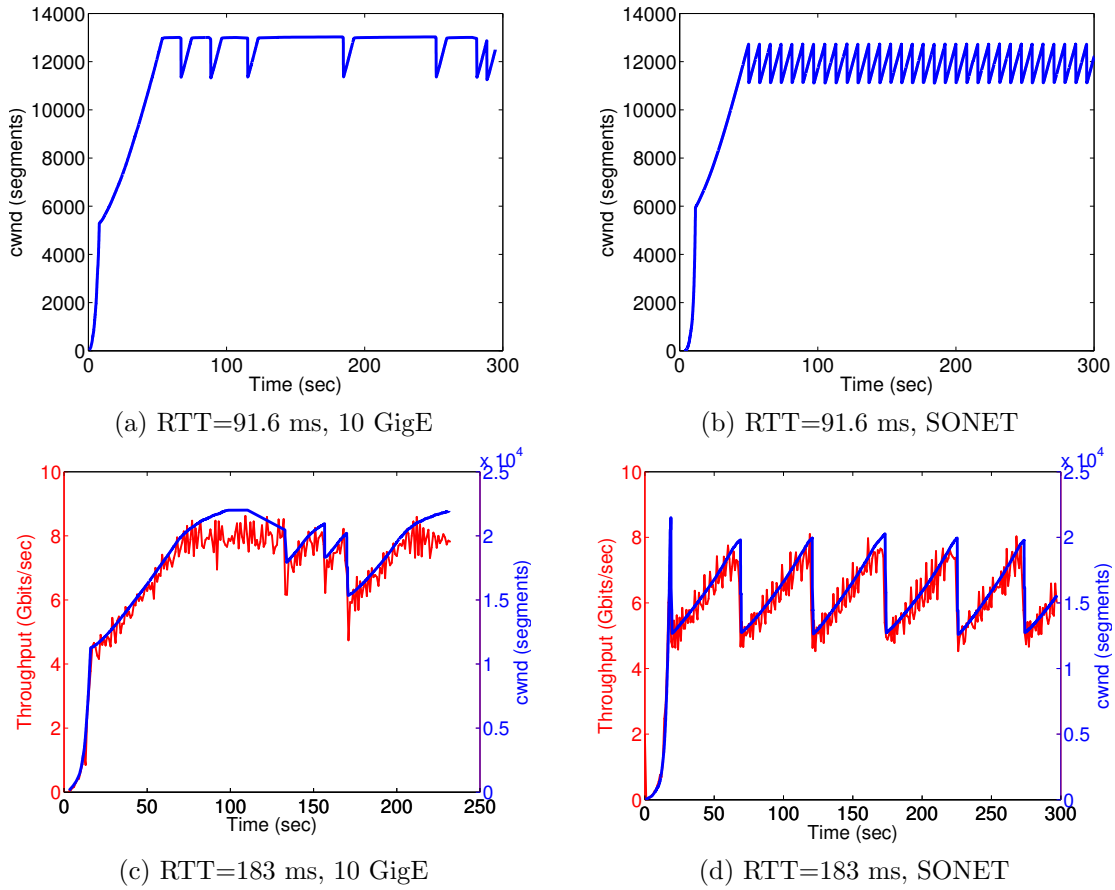
1.3.2 Measurement Collection

We collect TCP measurements for three TCP CC modules: CUBIC, STCP, and H-TCP (all available as loadable modules under the Linux 2.6 distribution). Two types of measurements are collected in each case.

(a) *Throughput measurements* for memory-to-memory transfers are collected using *iperf*. In addition, intermediate throughput values from *iperf* at one-second intervals are collected for a more detailed analysis.

(b) *Kernel traces:* Certain TCP variables including *cwnd*, are collected using the *tcpprobe* kernel module to support a more detailed analysis and parameter estimates for analytical models. In this case, *tcpprobe* is configured to collect *tcp_info* variables each time a TCP segment is processed.

The *iperf* and *tcpprobe* traces are collected concurrently to facilitate the correlation of TCP parameters with throughput. Each test regimen is executed using a *bash* script that coordinates the setup of parameters for emulators, invocation of *iperf* and *tcpprobe* codes and collection of their outputs.



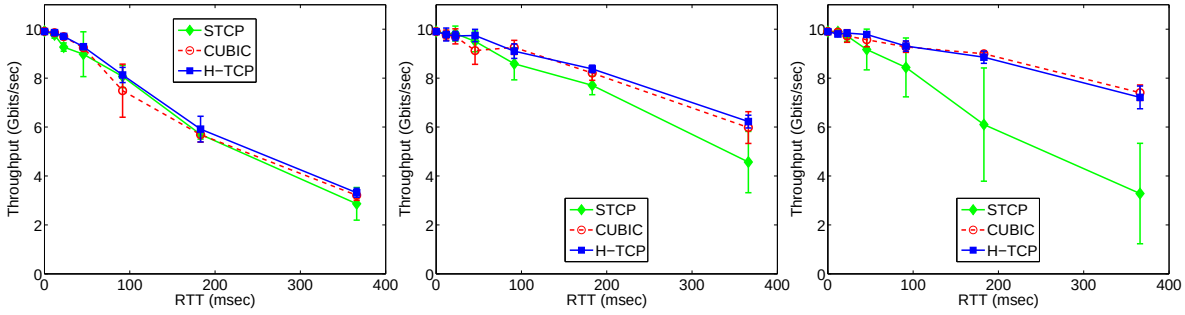
(a) RTT=91.6 ms, 10 GigE

(b) RTT=91.6 ms, SONET

(c) RTT=183 ms, 10 GigE

(d) RTT=183 ms, SONET

Figure 1.2: Comparisons between 10 GigE and SONET measurements. TCP variant: STCP. (a) and (b) show *tcpprobe*. (c) and (d) show both *iperf* and *tcpprobe*.



(a) three streams

(b) seven streams

(c) ten streams

Figure 1.3: Comparison of STCP, CUBIC, and H-TCP average throughputs on the 10 GigE testbed.

1.3.3 Empirical Observations

It is interesting to note the differences in measurements produced by the SONET and 10 GigE links. In general, we observe that the data obtained from SONET is well-behaved and more deterministic than that collected from 10 GigE. Figure 1.2 illustrates the stark contrast in TCP behavior between these two testbeds. The evolution of *cwnd* over time exhibits a consistent sawtooth pattern for SONET, whereas the 10 GigE transfer experiences non-uniformly-spaced losses and seemingly flat regions for smaller RTTs. The data shown is for STCP, for RTTs of 91.6 ms and 183 ms, although the results are consistent across different variants and other RTT values. These differences are important: they mean that for accurate predictions, the models must account for both the frequency of losses and the shapes of the *cwnd* curves. Section 1.4 goes into specifics on how this can be accomplished.

Figures 1.2c and 1.2d reveal another difference between these two modalities: 10 GigE has more aggregate buffer space than SONET. Visually, buffering occurs when *cwnd* grows as throughput remains relatively constant. The size of the buffer corresponds to the white space between the two curves, and since it is more prevalent in 10 GigE data, we know that this configuration has the larger buffer size. Figure 1.3 presents a comparison of the three variants in terms of their average throughput. The dataset was collected over the 10 GigE link. Each memory-to-memory flow was active until it transferred 10 GB of data. It is evident that for a small number of streams, the three variants perform almost equally well. However, as the number of flows grows, the differences in performance become more notable: CUBIC and H-TCP significantly outperform STCP with ten parallel flows.

1.4 Analysis

In this section, we first present a model for slow-start and discuss how it can accommodate an important component of HyStart. Then, we present single-stream

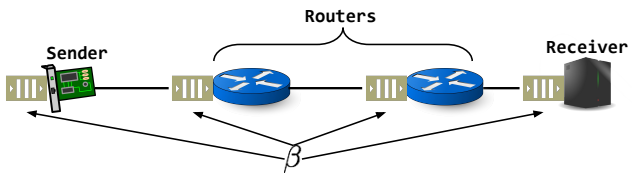


Figure 1.4: Buffer size β on a network. This includes buffers at sender and receiver network interfaces, and routers.

analytical models for two different types of loss-based congestion avoidance mechanisms: (i) those that are ACK-based and (ii) those that grow *cwnd* as a function of time since last loss (referred to as TSL-based variants). STCP and H-TCP are ACK-based, while BIC and CUBIC are examples of TSL-based algorithms. Finally, we demonstrate how single-stream models can be easily extended to incorporate multiple parallel TCP streams.

Table 2.1 contains definitions of variables used in this section. We assume that we know the link capacity C, aggregate buffer size β, round-trip time τ, size of the transfer F, and increase or decrease parameters. The minimum RTT, τ, is the delay measured before the buffer of size β begins to fill up. This can be measured by running a simple *ping* command between the source and receiver. β represents the aggregate size of all buffers present on the link (*e.g.*, on routers and NICs). An illustration is shown in Figure 1.4. W_{max} is a constraint on *cwnd* typically imposed by the receiver (receive-window).

1.4.1 Slow-Start

In traditional slow-start, *cwnd* approximately doubles every round trip time of length τ in which loss is not detected:

$$w(t) = 2^{t/\tau}.$$

Slow-start ends at time T_s, when $w(t)$ reaches *ssthresh* and CA begins.

Table 1.1: Variable definitions used in analysis.

Variable	Definition	Unit
C	link capacity	bits/sec
τ	minimum round trip time	sec
$w(t)$	*cwnd* as function of time	bits
W_{max}	maximum window constraint	bits
W_m	size of *cwnd* immediately before loss	bits
β	aggregate buffer size	bits
S	sending rate	bits/sec
Θ	throughput	bits/sec
$ssthresh$	slow-start threshold	bits
a	additive or multiplicative increase factor	N/A
b	multiplicative decrease factor	N/A
F	file size	bits
k	number of CA epochs for a transfer	N/A
n	number of parallel TCP flows	N/A

$$T_s = \tau \log_2(ssthresh). \tag{1.2}$$

The amount of data transferred during time T_s is N_s:

$$N_s = \frac{1}{\tau}\int_0^{T_s} 2^{t/\tau} dt = \frac{2^{t/\tau}}{\ln(2)}\Big|_0^{T_s} = \frac{ssthresh - 1}{\ln(2)}. \tag{1.3}$$

These expressions are accurate when $C\tau$ (the BDP) is greater than or equal to $ssthresh$. In HSN environments, this is a reasonable assumption that is validated by the measurements (see Figure 1.2 for examples).

At the time of writing, HyStart is the default slow-start algorithm used in Linux. This algorithm uses the same *cwnd* growth function as traditional slow-start. According to [29], HyStart sets an upper bound on *cwnd* equal to $C_a\tau/2 + \beta_a$, where C_a is the available bandwidth and β_a is the available buffer space. There is also a lower bound on *cwnd* equal to $(C_a\tau)/2$. In real scenarios (especially where packet loss is highly prevalent), it is likely that the value of $ssthresh$ will change several times during the lifetime of a TCP flow. However, for highly-reliable, dedicated con-

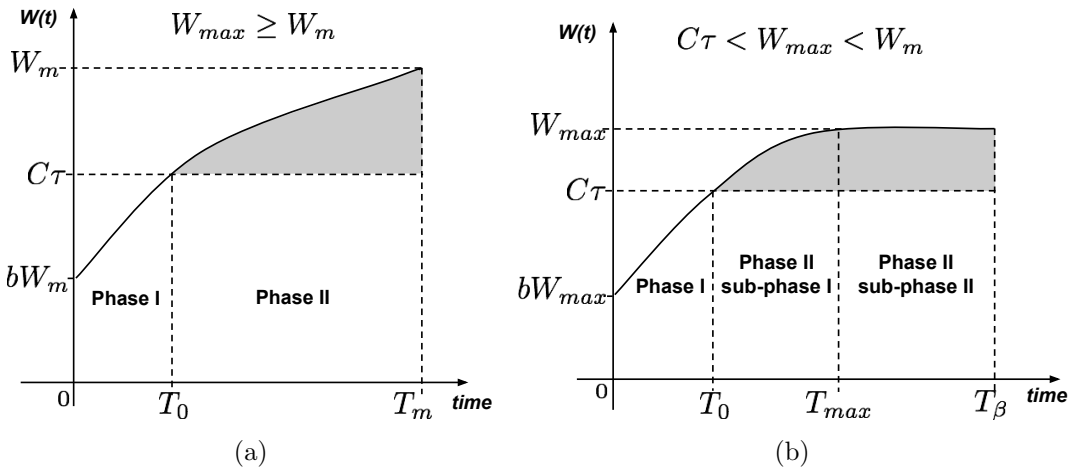


Figure 1.5: Example *cwnd* curves for one congestion avoidance epoch, with (a) and without (b) a W_{max} constraint. Shaded areas are the size of the buffer.

nections, such as ones encountered in HSN settings, it is safe to assume that a flow will enter slow-start only once (at the beginning of a transfer), and afterward will likely remain in CA state. This assumption is further validated by our *tcpprobe* measurements. In this case, only one estimate of *ssthresh* is required. At the beginning of a connection, most of the link capacity and buffer space are unused. Hence, we let $ssthresh = C\tau/2 + \beta$, which proves to work quite well in practice; in fact, we find it much better to model *ssthresh* as a function of τ, rather than leaving it as a constant.

1.4.2 Congestion Avoidance

For all TCP variants, CA consists of two phases. Figure 1.5 illustrates them using generic *cwnd* curves (not specific to any TCP variant). The goal is to derive expressions for N_a, the amount of data transferred during one such CA epoch, and T_a, the duration of one CA epoch. The number of CA epochs for a given transfer is approximately

$$k = \frac{F - N_s}{N_a}.$$

The sending rate S can be approximated as

$$S = \frac{N_s + kN_a}{T_s + kT_a}.$$

Then, throughput Θ can be estimated as

$$\Theta = S(1 - p)$$

where p is the packet loss probability for a TCP flow. For small values of p,

$$\Theta \approx S.$$

Next, we discuss how to obtain N_a and T_a for various cases.

Case 1: $W_{max} \leq C\tau$. In Phase I, *cwnd* grows until either the link capacity or W_{max} is reached (if $W_{max} \leq C\tau$). In the latter case, *cwnd* remains flat until the transfer ends, buffer overflow never occurs, and there is no second phase. The amount of data transferred in this case is $N_a = F - N_s$ (file size minus the amount of data transferred in slow-start), and since the sending rate is capped at W_{max}/τ, it takes $T_a = N_a\tau/W_{max}$ time until the transfer ends.

Case 2: $W_{max} > C\tau$. In this case, each congestion epoch will have two phases.

Phase I: The methodology for obtaining N_0 and T_0 for Phase I is the same for ACK-based and TSL-based variants. Given a congestion window function $w(t)$, we know that:

$$w(T_0) = C\tau.$$

Using this relation, we can solve for T_0: either directly (if $C\tau < W_{max} < W_m$ as in Figure 1.5b) or in terms of W_m (if $W_{max} \geq W_m$ as in Figure 1.5a). In order to distinguish between 1.5b and 1.5a, we must first solve for W_m (discussed below). The amount of data transferred in Phase I is:

$$N_0 = \frac{1}{\tau} \int_0^{T_0} w(t)dt.$$

Phase II: In Phase II, the sending rate is capped at C, and the buffer begins to fill up. We first solve for W_m with the hypothesis that 1.5a is the correct representation of our congestion epoch. Once a value for W_m is obtained, this hypothesis can be rejected or accepted by comparing W_m with W_{max}. However, how we solve for W_m depends on the type of CA mechanism.

1.4.2.1 ACK-Based CA

The following ordinary differential equation (ODE) describes the behavior for ACK-based TCP variants:

$$\frac{dw}{dt} = \frac{dw}{dA}\frac{dA}{dt} \tag{1.4}$$

where A represents an acknowledgement. Since the sending rate is C, this is also the rate at which ACKs are being received:

$$\frac{dA}{dt} = C.$$

For every ACK, *cwnd* increases by a, so

$$\frac{dw}{dA} = a.$$

Hence,

$$\frac{dw}{dt} = Ca. \tag{1.5}$$

The initial condition for this ODE is $w(0) = C\tau$. The ODE can now be solved to obtain a window function for Phase II, $w_2(t)$.

1.4.2.2 TSL-Based CA

In a TSL-Based variant, *cwnd* grows identically in Phase II as it does in Phase I. Hence, $w_2(t) = w(t)$.

Now that we have $w_2(t)$, we make the following useful observations:

$$w_2(T_m) = W_m, \tag{1.6}$$

$$N_m = \frac{1}{\tau}\int_0^{T_m} w_2(t)dt, \tag{1.7}$$

$$N_m - CT_m - \beta = 0. \tag{1.8}$$

Using (1.6), we can solve for the duration of Phase II, T_m, in terms of W_m. The amount of data transferred in Phase II, N_m, is given by (1.7), also in terms of W_m. Finally, we obtain a value for W_m by finding the roots of (1.8) and selecting the appropriate root (subject to the constraints that $W_m \in \mathbb{R}$ and $W_m \geq C\tau$).

Subcase 1: $W_m \leq W_{max}$. In this case, we simply substitute the value of W_m into the expressions for T_m, N_m, T_0 and N_0. Then,

$$N_a = N_0 + N_m,$$

$$T_a = T_0 + T_m.$$

Subcase 2: $W_m > W_{max}$. In this case, Figure 1.5b correctly depicts the shape of the congestion epoch. Since W_{max} is known, we can solve for T_{max}, the duration of sub-phase I of Phase II, directly:

$$w_2(T_{max}) = W_{max}.$$

The amount of data transferred during this sub-phase is

$$N_{max} = \frac{1}{\tau} \int_0^{T_{max}} w_2(t)dt.$$

The only unknown left is T_β, the duration of sub-phase II, and N_β, the amount of data transferred during that interval. However, we know that during sub-phase II, the sending rate is capped at W_{max}/τ, so

$$N_\beta = \frac{W_{max}}{\tau} T_\beta.$$

We can solve the following equation for T_β:

$$N_{max} + N_\beta - C(T_{max} + T_\beta) - \beta = 0.$$

Then,

$$N_a = N_0 + N_{max} + N_\beta,$$
$$T_a = T_0 + T_{max} + T_\beta.$$

1.4.3 Examples

We demonstrate the versatility of the generalized modelling framework described in previous sections through three examples: STCP and H-TCP (both ACK-based variants) and CUBIC (a TSL-based variant).

1.4.3.1 STCP

Since the additive increase factor for STCP is a constant, the solution to the ODE in (1.5) is simply

$$w_2(t) = C(at + \tau). \tag{1.9}$$

Using this equation, we obtain:

$$\begin{aligned} T_m &= \frac{W_m - C\tau}{Ca}, \\ N_m &= \frac{W_m^2 - (C\tau)^2}{2Ca\tau}, \\ W_m &= C\tau + \sqrt{2Ca\tau\beta}. \end{aligned}$$

If $W_m \leq W_{max}$, we have everything we need for Phase II. Otherwise, we use W_{max} instead of W_m in the expressions for T_m and N_m; this gives us T_{max} and N_{max}:

$$\begin{aligned} T_{max} &= \frac{W_{max} - C\tau}{Ca}, \\ N_{max} &= \frac{W_{max}^2 - (C\tau)^2}{2Ca\tau}. \end{aligned}$$

Further,

$$T_\beta = \frac{\beta\tau}{W_{max} - C\tau} - \frac{W_{max} - C\tau}{2Ca}.$$

This completes the analysis for Phase II. We now analyze Phase I by first constructing an ODE that describes *cwnd* growth before the BDP is reached:

$$\frac{dw}{dA} = a \quad \text{as in Phase II, but since } S \text{ is below } C, \quad \frac{dA}{dt} = \frac{w}{\tau}.$$

The final ODE and its solution are:

$$\begin{aligned} \frac{dw}{dt} &= \frac{w}{\tau}a, \\ w(t) &= ge^{at/\tau} \text{ where } g \text{ is a constant}, \\ w(0) &= bW_m = g, \\ w(t) &= bW_m e^{at/\tau}. \end{aligned} \tag{1.10}$$

Next, we can solve for T_0 and N_0:

$$\begin{aligned} T_0 &= \frac{\tau}{a} \ln \left(\frac{C\tau}{bW_s} \right), \\ N_0 &= \frac{C\tau - bW_s}{a} \end{aligned}$$

where

$$W_s = \min\left(W_m, W_{max}\right). \tag{1.11}$$

This completes the analysis for STCP.

1.4.3.2 CUBIC

Because CUBIC is not ACK-based, there is no need to derive two different window functions for its analysis: $w(t)$ grows as follows for both phases:

$$w(t) = c\left(t - \sqrt[3]{\frac{bW_s}{c}}\right)^3 + W_s \tag{1.12}$$

where W_s is defined in (1.11), t is the time since the last congestion event in unit of RTT, c is a scaling factor (usually equal to 0.4), and b is a multiplicative decrease factor usually equal to 0.2.

We present the closed-form solutions for the necessary variables. Since CUBIC is TSL-based, it helps to think of T_0 and T_m as points in time that delimit the phases, rather than durations of phases (let T_β remain as the duration of sub-phase II of Phase II).

$$\begin{aligned} T_m &= \sqrt[3]{\frac{bW_m}{c}}, \\ N_m &= \frac{W_m}{\tau}\sqrt[3]{\frac{bW_m}{c}}\left(1-\frac{b}{4}\right), \\ T_0 &= \sqrt[3]{\frac{C\tau - W_m}{c}} + \sqrt[3]{\frac{bW_m}{c}}. \end{aligned}$$

Above, N_m is the amount of data transferred in the interval $[0, T_m]$. Let N_1 be the amount of data transferred in the interval $[T_0, T_m]$.

$$\begin{aligned} N_1 &= \frac{1}{\tau}\int_{T_0}^{T_m} w(t)dt = -\frac{1}{\tau}\sqrt[3]{\frac{C\tau - W_m}{c}}\left(\frac{C\tau + 3W_m}{4}\right), \\ N_1 &- C(T_m - T_0) - \beta = 0. \end{aligned}$$

The last equation above can be solved for W_m:

$$W_m = C\tau - \sqrt[4]{c\left(\frac{4\beta\tau}{3}\right)^3}.$$

Since we know that $W_m > C\tau$, we must take the negative root of the fourth-root term above. If W_m is indeed less than or equal to W_{max}, then the sending rate is

$$S = \frac{N_s + kN_m}{T_s + kT_m}.$$

Otherwise, we use W_{max} in the expressions for T_m and N_m above to obtain T_{max} and N_{max}, respectively. Also,

$$T_\beta = \frac{\beta - N_{max} - C\sqrt[3]{\frac{C\tau - W_{max}}{c}}}{\frac{W_{max}}{\tau} - C}.$$

Finally, S is

$$S = \frac{N_s + k\,(N_{max} + N_\beta)}{T_s + k(T_{max} + T_\beta)}.$$

1.4.3.3 H-TCP

We do not present closed-form solutions for H-TCP, since they are too complex; the majority of the computations for this variant were performed using `Matlab`. Since Δ^L is usually set to one second, for simplicity we use it in the derivations below. We present the non-trivial case, in which the transfer lasts for T_s+1 seconds. For H-TCP, let T_0, T_m, T_{max}, and T_β be points in time, rather than phase durations. For the first second of CA, H-TCP uses the increase function shown in (1.1a). The *cwnd* function is then

$$w_1(t) = bW_m + \frac{2(1-b)t}{\tau}.$$

The amount of data transferred during this time is N_{Δ^L},

$$N_{\Delta^L} = \frac{1}{\tau}\int_0^{T_{\Delta^L}} w_1(t)dt$$

where

$$T_{\Delta^L} = \min\left(1s, \frac{\tau(C\tau - bW_m)}{2(1-b)}\right).$$

This constraint on T_{Δ^L} is required because it may take less than one second for $w_1(t)$ to reach $C\tau$.

After the first second of CA, if $w_1(1) < C\tau$, H-TCP's increase function changes as shown in (1.1b). The *cwnd* function changes to

$$w_2(t) = w_1(1) + \frac{2(1-b)t}{\tau}\left(1 + 10(t-1) + \left(\frac{t-1}{2}\right)^2\right)$$

until *cwnd* reaches $C\tau$ at time T_0. N_0, the amount of data transferred in the interval $[0, T_0]$, is:

$$N_0 = \begin{cases} N_{\Delta^L}, & \text{if } T_{\Delta^L} \leq 1s, \\ N_{\Delta^L} + \dfrac{1}{\tau}\displaystyle\int_1^{T_0} w_2(t)dt, & \text{otherwise.} \end{cases}$$

T_0 can be isolated from the relation $w_2(T_0) = C\tau$. After T_0 seconds have passed, the *cwnd* function changes again because the transfer transitions into Phase II. This new function is described by the ODE in (1.5), which uses the increase function shown in (1.1a) or (1.1b) depending on the value of T_{Δ^L}:

$$\begin{aligned} dw &= aCdt, \\ w_3(t) &= 2(1-b)C\int a(t)dt, \\ w_3(T_0) &= C\tau \text{ is the initial condition.} \end{aligned}$$

H-TCP then uses $w_3(t)$ for the rest of the congestion epoch, which ends at time T_m. It is possible to solve for T_m in terms of W_m using the relation $w_3(T_m) = W_m$. The amount of data transferred during Phase II is N_m:

$$N_m = \frac{1}{\tau}\int_{T_0}^{T_m} w_3(t)dt.$$

It is now possible to solve for W_m using the following equation:

$$N_m - C(T_m - T_0) = \beta.$$

If $W_{max} \geq W_m$, then S is:

$$S = \frac{N_s + k(N_0 + N_m)}{T_s + kT_m}.$$

Otherwise, T_β must be determined using a similar technique used for STCP and S changed to:

$$S = \frac{N_s + k\left(N_0 + \frac{1}{\tau}\int_{T_0}^{T_{max}} w_3(t)dt + \frac{W_{max}}{\tau}(T_\beta - T_{max})\right)}{T_s + kT_\beta}.$$

1.4.4 Parallel Flows

We now present a simple, yet effective modification to the single-stream modelling framework presented above to accommodate multiple flows. The main idea is to model n parallel flows as a single, more aggressive flow. The modifications are as follows:

1. For any single-flow TSL-based CA *cwnd* function $w(t)$, the multiple-flow *cwnd* function is $w(nt)$, where n is the number of flows. For any single-flow ACK-based CA *cwnd* function, the multiple-flow *cwnd* function is obtained by multiplying the increase factor (as in the case of STCP) or the increase function (as in the case of H-TCP) by n. The only exception to this rule are *cwnd* functions in Phase II of ACK-based CA variants, which must be left unchanged. For example, STCP's (1.10) becomes

$$w(t) = bW_m e^{ant/\tau}$$

while (1.9) remains the same. The reason for this is that once an ACK-based variant transitions into Phase II, ACKs continue to arrive at a constant rate, so that there

is no added advantage of using multiple flows past the point where $cwnd > BDP$. TSL-based variants, on the other hand, continue to grow the aggregate *cwnd* at a rate approximately n times faster than with a single flow, for as long as congestion is not detected.

2. W_{max} gets scaled up by a factor of n. However, there must be a hard constraint, W_{maxH}, on the aggregate *cwnd*, imposed by memory and buffer limitations. Therefore,

$$W_{max} = \min(nW_{max}, W_{maxH}).$$

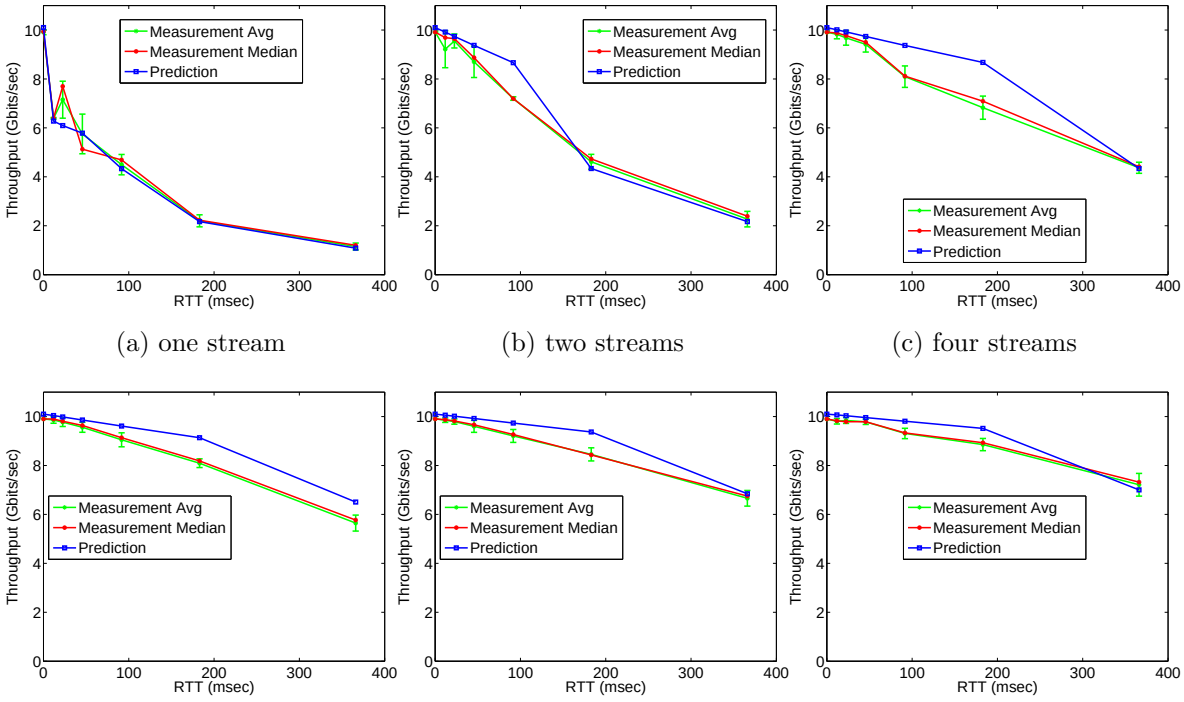
3. In slow-start, we keep the duration (T_s) the same as for single flows, but multiply N_s by n. In addition, instead of $ssthresh$, we use

$$thresh = \min(ssthresh, C\tau/n, W_{max}).$$

Interestingly, Crowcroft *et al.* have previously explored a related idea, but as a means of delegating some flows a higher fraction of the bandwidth on a network [11]. They propose a controller, MulTCP, that attempts to increase the throughput of a single flow by a factor of n by scaling the flow's additive increase parameter by the same amount . Simulations showed that MulTCP does indeed achieve a sending rate of approximately nS as long as n is not too large.

1.5 Model Performance

Figure 1.6 shows throughput predictions for one, two, four, six, eight, and ten H-TCP streams (we also did this for three, five, seven, and nine streams, not shown here). Measurement data obtained from the 10 GigE testbed is also presented in the figure and serves as validation for the models. The predictions were obtained using



(a) one stream (b) two streams (c) four streams

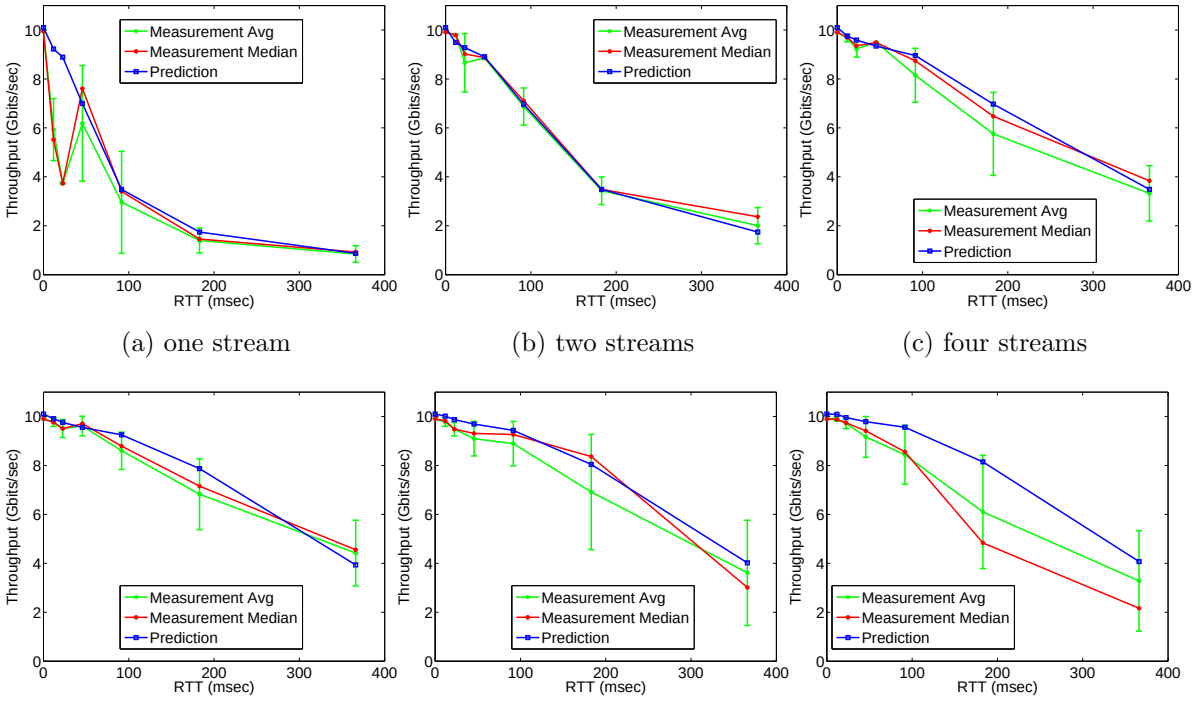
(d) six streams (e) eight streams (f) ten streams

Figure 1.6: Measurement averages and medians vs multiple-stream model predictions for H-TCP.

models presented in Section 1.4. Figure 1.7 shows the result of using the multiple-stream models to predict the throughput of one, two, four, six, eight, and ten parallel STCP flows (we also did this for three, five, seven, and nine streams, not shown here). Figure 1.8 shows the same for CUBIC.

Due to testbed configuration changes between the experiments, we use different β and W_{max} parameter values for prediction in each dataset (although β and W_{max} are kept constant within a dataset). A dataset refers to all data collected for a particular protocol, so there are a total of three. Note the odd 'dip' in throughput in Figure 1.7a for RTTs 11.8 ms and 22.6 ms: this is believed to be an anomaly. Define the average error as follows:

$$E := \frac{100}{|RTTs|} \sum_{r \in RTTs} \frac{|M_r - P_r|}{M_r}$$



(a) one stream (b) two streams (c) four streams

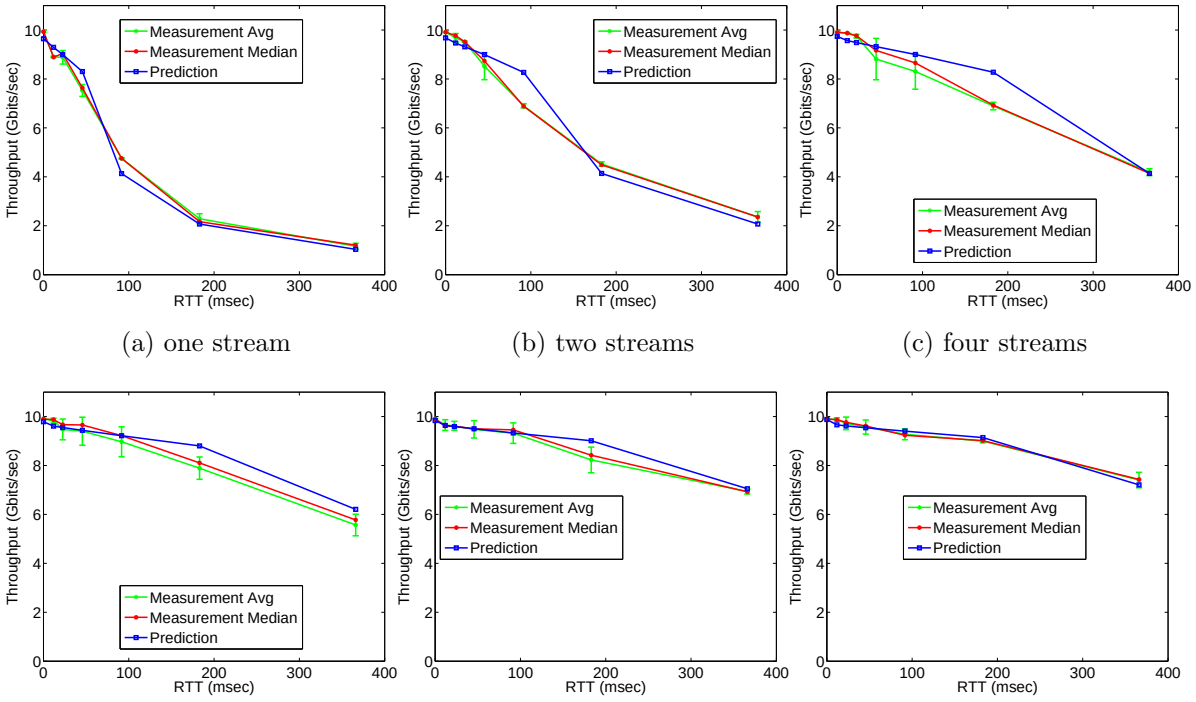
(d) six streams (e) eight streams (f) ten streams

Figure 1.7: Measurement averages and medians vs multiple-stream model predictions for STCP.

where M_r and P_r are the measured and predicted throughputs, respectively, for RTT $= r$. The mean errors (averaged across all RTTs and $n \in \{1, \ldots, 10\}$) for multiple-stream predictions are as follows: 5.2% for H-TCP, 9.3% for STCP, and 4.5% for CUBIC.

1.6 Conclusion

In this work, we have derived a unifying scheme for analyzing single-stream and multiple-stream memory-to-memory TCP transfers. We performed a detailed analysis for a diverse set of TCP variants: STCP – a MIMD algorithm; CUBIC – a non-ACK-based algorithm; and H-TCP – an adaptive AIMD algorithm. The models that emerged from this analysis were validated using an extensive set of measurements. The results show that our models can be used to achieve accurate and reliable



(a) one stream (b) two streams (c) four streams

(d) six streams (e) eight streams (f) ten streams

Figure 1.8: Measurement averages and medians vs multiple-stream model predictions for CUBIC.

throughput predictions. The measurements independently show that CUBIC and H-TCP consistently outperform STCP, and the difference in performance becomes more pronounced as the number of parallel flows grows. Interesting future directions are to expand the analysis to include (a) exogenous and time-variant loss, (b) I/O constraints, and (d) considerations of fairness and convergence.

CHAPTER 2

TOWARDS STABILITY ANALYSIS OF DATA TRANSPORT MECHANISMS: A FLUID MODEL AND ITS APPLICATIONS

2.1 Introduction

TCP carries most of the traffic on the Internet. One of its important functions is to perform end-to-end congestion control to alleviate congestion in the Internet and to provide fair bandwidth sharing among different flows. To date, many different congestion control algorithms (variants) have been developed, among which are Reno, Vegas, STCP [41], CUBIC [30], H-TCP [47], and BBR [10]. Stability is an imperative property for any dynamical system. The stability of several of these variants including Reno, Vegas, and STCP has been extensively and carefully studied, however, little is known about the stability properties of more recent variants such as CUBIC and H-TCP. These latter variants have typically been studied through simulation and experimentation, neither of which are adequate to make careful statements about stability. As we will observe, for some variants this deficiency is due to the lack of a modeling framework with which to develop appropriate models that are amenable to a formal stability analysis. The goals of this work are to point out deficiencies in the previous framework used to study variants such as Reno that make it unsuitable to study a variant such as CUBIC, and then to present a new framework and apply it to the analyses of CUBIC and H-TCP. Our choice of CUBIC is because it is a popular variant that is the default in the Linux distribution, and our choice of H-TCP is because it has been recommended by the Energy Sciences Network [16] and has been used within the Department of Energy's data transfer network [63].

The traditional approach for modeling a congestion control algorithm's behavior is to derive a differential equation (DE) for its congestion window (*cwnd*) or sending rate as a function of time. Such DEs typically account for the algorithm's increase and decrease rules, as well as loss probability functions, for example, to incorporate an active queue management (AQM) policy. This method is highly effective for modeling certain types of controllers, such as TCP Reno and STCP, whose *cwnd* update rules are very simple (*e.g.*, Reno's *cwnd* grows by one every round trip and decreases by half upon congestion detection). However, this approach reaches its limitations when presented with a controller whose *cwnd* update functions are complex, thereby making it difficult or impossible to directly write a DE for the *cwnd* or sending rate. For example, CUBIC's increase update rule is a function of *time since last loss* and of the *congestion window size immediately before loss*. Moreover, in the case of CUBIC, the steady-state value of *cwnd* lies at the saddle point of the window function, which obstructs the stability analysis of the protocol.

To overcome the impediments of the traditional approach, we develop a novel framework that exploits the fact that all *cwnd*- and rate-based controllers that utilize packet loss information[1] to make changes to the *cwnd* or rate have two variables in common: the value of *cwnd* (rate) immediately before loss and the time elapsed since last loss. As a consequence, one can derive a set of two DEs: the first describing the maximum *cwnd* (rate) as a function of time, and the second describing the duration of congestion epochs. This is a relatively easy task, compared to deriving a DE for *cwnd* (rate) of a complex algorithm directly. The advantage of such a model is that it offers tremendous versatility since it does not define *cwnd* or rate functions within the set of DEs, with the latter being identical for many controllers. Note that the

[1]Note that this includes not only ACK-based algorithms, but also packet marking schemes as in ECN (Explicit Congestion Notification). From this point forward, we refer to such schemes as "loss-based".

proposed model is applicable not only to TCP-based congestion controllers, but also to UDT [25] and QCN [58].

In this work, we use both event- and packet-based simulations to validate our analytical models. To validate the analytical stability results for CUBIC, we introduce a lightweight simulation framework that can easily be adapted to other congestion control variants. The simulation treats loss as a non-homogenous Poisson process and generates new loss events based on a user-defined loss model. We refer to this framework as the non-homogeneous Poisson loss (NHPL) simulation. The reason for its development is that it enables us to fully control every aspect of a model, such as changing link capacity throughout a flow's lifetime or using custom loss probability models. Perhaps the most important capability of this framework is the ease of specifying initial conditions for flows at the start of the congestion avoidance phase, since having control of these conditions is critical for testing the regions of stability for algorithms that are *locally* stable (as most congestion controllers in practice are). A description of the NHPL simulation framework is provided in Appendix A.1.

Further, at the time of writing, *ns-3* [56] – a popular discrete-event network simulator – did not yet natively support CUBIC, and existing implementations have scalability problems: as more flows are introduced, the simulation becomes quite slow. Hence, we use the NHPL framework to validate the DE model for CUBIC and observe that the average *cwnd*s predicted by both are in close agreement. As system parameters are varied, the simulation and CUBIC's DE model also agree on whether the system is stable. For TCP CUBIC, we observe that instability can be introduced by setting the initial conditions too far from their fixed-point values. While our analysis states that CUBIC is locally asymptotically stable, these simulations complement the theory by demonstrating that CUBIC is *not* globally stable.

We use *ns-3*, which unlike the NHPL framework is a packet-based simulation, to validate the DE model for H-TCP. We observe that *ns-3* is in close agreement

with the DE model in terms of average *cwnd* and average congestion epoch duration. We also note that when H-TCP operates in its high-speed regime, its *cwnd* tends to exhibit large oscillations that last indefinitely, as evidenced by both the DE model and numerical evaluation of H-TCP's stability condition. In contrast, CUBIC's *cwnd* exhibits convergence to its fixed point, and fewer oscillations.

A summary of the contributions of this work is as follows:

- a new modeling framework applicable to a diverse set of congestion control algorithms,
- applications of this model to CUBIC and H-TCP, and stability analyses of these algorithms,
- validation of this model with two different simulation frameworks.

We call the new modeling framework the *MWLI (Max Window Loss Interval) model*.

The rest of this chapter is organized as follows: we discuss related work in Section 2.2. We introduce the modeling framework in Section 2.3.1 and apply it to TCP Reno. In Section 2.3.2, we show that the new framework is equivalent to the one presented in [51]. In Section 2.4, we apply the MWLI model to TCP CUBIC, perform a careful stability analysis of the congestion controller, and present a convergence result. We also present a detailed analysis of CUBIC's fixed point and observe its limiting behavior in terms of link capacity and delay. In Section 2.5, we apply the model to H-TCP and present a sufficient condition for stability. Similar to the CUBIC analysis, we also perform a detailed analysis of H-TCP's fixed-point and derive its limiting behavior. In Section 2.6, we discuss the specific choice of a loss probability model used throughout our work. In Section 2.7, we validate the new model and the stability result for CUBIC and H-TCP using two different types of simulations and loss models. We draw conclusions in Section 2.8.

2.2 Background

There exist a number of analytical studies of TCP and its stability. In [51], Misra *et al.* derive a fluid model for a set of TCP Reno flows and show an application to a networked setting where RED (Random Early Detection) is the AQM policy. Kelly proposed an optimization-based framework for studying and designing congestion control algorithms in [40], where STCP was an output. In [69], Srikant presented a simple analysis of Jacobson's TCP congestion control algorithm. In [35], Hollot *et al.* analyze the stability of TCP with an AQM system implementing RED.

Huang *et al.* develop and analyze the stability of a general nonlinear model of TCP in [37], focusing on HighSpeed, Scalable, and Standard TCP for comparisons of relative stability. The authors rely on functions $f(w)$ and $g(w)$, which are additive and multiplicative parameters, respectively, and are both functions of the current congestion window size. Our model differs from these examples in that rather than modeling the congestion window directly, we instead model two interdependent variables (maximum *cwnd* and time between losses) that in turn determine the evolution of the window. This new method presents a window of opportunity for modeling complex, nonlinear transport algorithms for which it is not possible to write a DE for *cwnd* directly or whose $f(w)$ and $g(w)$ functions cannot be written in closed form.

Theoretical analyses of TCP CUBIC are rare, possibly due to the protocol's behavior around the fixed point, which significantly complicates the analysis of its stability. In one study, Bao *et al.* propose Markov chain models for average steady-state TCP CUBIC throughput, in a wireless environment [2]. In [61], Poojary *et al.* derive an expression for average *cwnd* of a single CUBIC flow under random losses. In contrast to [2] and [61], the model we present in this work for CUBIC provides insight into both the transient and steady-state behavior of the algorithm. Moreover, we utilize Lyapunov stability theory to prove that CUBIC is locally asymptotically stable in-

dependent of link delay and other system parameters (the parameters only affect the region of attraction). This result is one of the main contributions of this work.

Some studies attempt to gain insight into the stability of high-speed TCP variants using an empirical perspective. These usually involve characterizing stability (or instability) by the coefficient of variation (CoV) or stability index, as in [28] and [74], or even by simply using the standard deviation of the throughput as in [73]. While these observation-based and comparative studies are extremely valuable in assessing protocol behavior in deployment, they do not present a complete picture: to understand inherent protocol properties, modeling and performance analysis are required.

2.3 The Model

In this section, we present the new model, which is the focus of this work. As a proof of concept, we apply this model to TCP Reno and show that it is mathematically equivalent to the well-known DE model originally presented in [51]. We note that while the two models are equivalent, they make use of different types of information, which is essential for developing a fluid model for TCP CUBIC presented in Section 2.4 and for H-TCP presented in Section 2.5.

In the analysis that follows, we will use the notation $f \equiv f(t)$ to represent a function or variable that is not time-delayed. Similarly, we will use $f_T \equiv f(t-T)$ to represent a function or variable that is delayed by an amount of time T. We will also use $\dot{f} = df(t)/dt$ to represent the derivative of f with respect to time. The notation $\partial f(x)/\partial x$ denotes the partial derivative of f with respect to the variable x, and $\frac{\partial f(x)}{\partial x}|_{x=x^*}$ is the partial evaluated at $x = x^*$.

Table 2.1: Term definitions.

Term	Definition
C	per-flow capacity
τ	link delay in signaling a loss to the source
$W_{\max}(t)$	the size of the *cwnd* immediately before loss
$s(t)$	the time elapsed since loss
$W(t)$	the *cwnd* as a function of time
$p(t)$	a probability of loss function

2.3.1 The MWLI Model

Table 2.1 presents some useful definitions. The main idea behind the model is the following: instead of deriving a DE for the *cwnd* function $W(t)$ directly, which is specific to a data transport algorithm, we instead derive DEs for $W_{\max}(t)$ – the size of the *cwnd* immediately before the most recent loss, and $s(t)$ – the amount of time elapsed since last loss, which are variables common to all loss-based algorithms. Since $W(t)$ is a function of $W_{\max}(t)$ and $s(t)$, it is completely determined by their DEs. The result is the following model[2]:

$$\begin{aligned} \frac{dW_{\max}(t)}{dt} &= -(W_{\max}(t) - W(t))\frac{W(t-\tau)}{\tau}p(t-\tau) \\ \frac{ds(t)}{dt} &= 1 - s(t)\frac{W(t-\tau)}{\tau}p(t-\tau) \end{aligned} \tag{2.1}$$

Here, $p(t-\tau)$ is a loss probability function. The expression $W(t-\tau)p(t-\tau)/\tau$ describes the packet loss rate, delayed by τ. Here, τ represents the delay in signaling a loss to the source. The first DE describes the behavior of $W_{\max}$, which takes the value of $W(t)$ right before a loss. At the time of loss, if $W_{\max}(t) > W(t)$, then $W_{\max}$ decreases by the amount $W_{\max}(t) - W(t)$; otherwise, it increases by the same amount. The second DE describes the evolution of the time since last loss $s(t)$, which grows by one unit and is reset to zero upon loss. This system can be adapted to a rate-

[2]Note that $W(t)$ must be either derived explicitly, for example as in (2.2) for TCP Reno or given in the definition of the controller, as in (2.7) for CUBIC.

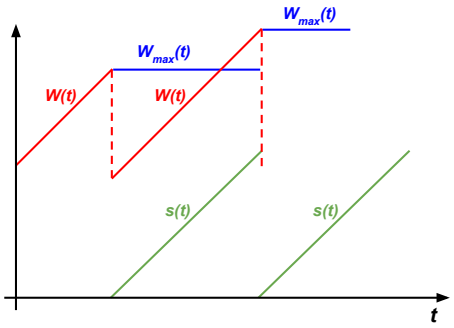


Figure 2.1: $W(t)$, $W_{\max}(t)$, and $s(t)$ for TCP Reno.

based scheme in terms of maximum rate and time since last rate decrease, simply by dividing each DE by τ. Since we will be describing applications of this model to TCP Reno, CUBIC, and H-TCP, which are all *cwnd*-based congestion controllers, we use (2.1) in the interest of this work.

Figure 2.1 illustrates $W_{\max}(t)$, $s(t)$, and $W(t)$ for TCP Reno. To adapt model (2.1) to TCP Reno, we define Reno's *cwnd* as a function of $W_{\max}(t)$ and $s(t)$. At the time of loss, $W(t) = W_{\max}(t)$ is halved. This becomes the initial value of $W(t)$ in the new congestion epoch. $W(t)$ then increases by one segment for every round-trip time, so the total increase is $s(t)/\tau$ after $s(t)$ time has elapsed since the last loss. Hence,

$$W(t) = \frac{W_{\max}(t)}{2} + \frac{s(t)}{\tau}. \tag{2.2}$$

Then the fluid model for Reno is (2.1) combined with (2.2).

The loss probability function can be customized according to the specific characteristics of a given system, such as queue size and AQM policy. For simplicity, when performing a formal stability analysis of a model, we use the following function:

$$p(t) = \max\left(1 - \frac{C\tau}{W(t)}, 0\right). \tag{2.3}$$

This function is presented in [69] as an approximation of the M/M/1/B drop probability when the buffer size $B \to \infty$. We will explore an alternate loss probability

model – one that incorporates more realistic queueing dynamics – when validating H-TCP using *ns-3*.

Note that model (2.1) does not specify $W(t)$, and therein lies the versatility of this scheme. For a given *cwnd*-based transport algorithm, the modeler need only substitute a function describing the evolution of *cwnd* over time, as we did for Reno. We demonstrate this technique again with CUBIC in Section 2.4. This property of the model is useful both for analyzing existing algorithms and examining the stability of new ones. Moreover, the MWLI model can be used to help design and fine tune new congestion control algorithms, as their behavior may be simulated efficiently and easily using (2.1) and any loss probability function.

2.3.2 Model Equivalence for TCP Reno

Consider the well-established model for TCP Reno's *cwnd* from [51] (equation (4) to be precise):

$$\frac{dW(t)}{dt} = \frac{1}{\tau} - \frac{W(t)}{2}\frac{W(t-\tau)}{\tau}p(t-\tau). \tag{2.4}$$

We assume τ to be constant for simplicity, even though the round-trip time in [51] varies in time as a function of both the propagation and queueing delays.

We now show that the two models (*i.e.*, the model represented by (2.1), (2.2) and the model represented by (2.4)) are mathematically equivalent. Differentiating (2.2) with respect to t, we have:

$$\dot{W} = \frac{\dot{W}_{\max}}{2} + \frac{\dot{s}}{\tau}. \tag{2.5}$$

Substituting (2.1) into (2.5) yields

$$\dot{W} = \frac{1}{\tau}\left(1 - s\frac{W_\tau}{\tau}p_\tau\right) + \frac{1}{2}\left((W - W_{\max})\frac{W_\tau}{\tau}p_\tau\right). \tag{2.6}$$

From (2.2), we know that $W_{\max} = 2(W - s/\tau)$. Substituting this expression for $W_{\max}$ into (2.6) and simplifying yields

$$\begin{aligned}\dot{W} &= \frac{1}{\tau}\left(1 - s\frac{W_\tau}{\tau}p_\tau\right) + \frac{1}{2}\left(\left(W - 2\left(W - \frac{s}{\tau}\right)\right)\frac{W_\tau}{\tau}p_\tau\right) \\ &= \frac{1}{\tau} - s\frac{W_\tau}{\tau^2}p_\tau - \frac{W}{2}\frac{W_\tau}{\tau}p_\tau + s\frac{W_\tau}{\tau^2}p_\tau \\ &= \frac{1}{\tau} - \frac{W}{2}\frac{W_\tau}{\tau}p_\tau.\end{aligned}$$

The last line corresponds to equation (2.4) and completes our proof of the equivalence of the models. When used with Reno, model (2.1) can be linearized and used to derive a transfer function. The latter can be analyzed to yield system parameter-dependent conditions for Reno's stability. This analysis is similar to the one presented in [69].

2.4 Analysis of TCP CUBIC

In this section, we perform a local stability analysis of TCP CUBIC. To do so, we first create a fluid model for this congestion control algorithm using the framework introduced in the previous section. Then, we show that the system has a unique fixed point and prove the existence and uniqueness of a solution. Next, we show that the linearization method yields inconclusive results when applied to the model, and are thus motivated to use Lyapunov's direct method to prove the stability of the system. First, we introduce a Lyapunov function candidate and since the system is time-delayed, use Razumikhin's Theorem to show that the candidate is suitable and that stability holds in a neighborhood of the fixed point of the system. A consequence of the failed linearization is that we will not prove exponential stability for CUBIC, but we can still show asymptotic and Lyapunov stability. In this section, we also analyze the limiting behavior of CUBIC's fixed point in terms of capacity and delay. Finally, we derive convergence results on the system's solution.

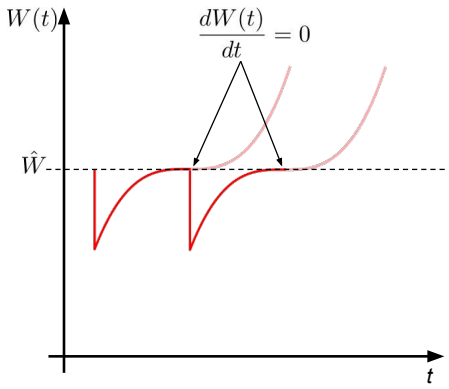


Figure 2.2: CUBIC's saddle point causes $dW(t)/dt$ to evaluate to zero at the fixed point of the system.

2.4.1 TCP CUBIC Fluid Model

TCP CUBIC's congestion window function is defined in terms of the time since last loss $s(t)$ and maximum value of *cwnd* immediately before the last loss $W_{\max}(t)$ [30]:

$$W(t) = c\left(s(t) - \sqrt[3]{\frac{W_{\max}(t)b}{c}}\right)^3 + W_{\max}(t) \tag{2.7}$$

where b is a multiplicative decrease factor and c is a scaling factor. Figure 2.2 illustrates the evolution of CUBIC's *cwnd* over time. The opaque red curves represent behavior in steady state: the window is concave until a loss occurs at CUBIC's fixed-point value of *cwnd*, $\hat{W}$. The light red curves describe *cwnd* behavior if a loss does not occur: the window becomes convex, also known as CUBIC's probing phase. The fluid model for CUBIC is then simply (2.1) coupled with (2.7), with (2.3) as the loss probability function. More specifically, (2.1), (2.3), and (2.7) represent the dynamical interaction of CUBIC with a congested network. Prior to the development of (2.1), we attempted to develop a fluid model by first computing the equilibrium point for CUBIC, but this exercise gave a value of s at (2.7)'s saddle point and consequently, a confounding linearization of $dW/dt = 0$. Further attempts at deriving dW/dt, taking

into account the time-dependencies $s(t)$ and $W_{\max}(t)$, resulted in a highly complex DE involving both $W_{\max}(t)$, $s(t)$, and their derivatives. Even obtaining the fixed points of this DE would be highly cumbersome, compared to obtaining the fixed point of (2.1).

2.4.2 Fixed Point Analysis

Let $\hat{W}_{\max}$, $\hat{s}$, $\hat{W}$, and $\hat{p}$ represent the fixed point values of $W_{\max}(t)$, $s(t)$, $W(t)$, and $p(t)$, respectively. Using the fact that in steady state, $W(t) = W(t-\tau) = \hat{W}$ and $p(t) = p(t-\tau) = \hat{p}$, system (2.1) becomes

$$-(\hat{W}_{\max} - \hat{W})\frac{\hat{W}}{\tau}\hat{p} = 0, \tag{2.8}$$

$$1 - \hat{s}\frac{\hat{W}}{\tau}\hat{p} = 0. \tag{2.9}$$

From (2.9), we see that

$$\hat{s} = \frac{\tau}{\hat{W}\hat{p}}. \tag{2.10}$$

It is clear that $\hat{W}$ and $\hat{p}$ do not equal zero in steady state. Using this information, along with (2.8), we conclude that $\hat{W}_{\max} = \hat{W}$. In steady state, (2.7) becomes

$$\hat{W} = c\left(\hat{s} - \sqrt[3]{\frac{\hat{W}b}{c}}\right)^3 + \hat{W}.$$

This equation yields

$$\hat{s} = \sqrt[3]{\frac{\hat{W}b}{c}}. \tag{2.11}$$

Combining (2.10) and (2.11), we have

$$\frac{\tau}{\hat{W}\hat{p}} = \sqrt[3]{\frac{\hat{W}b}{c}}$$

where $\hat{p} = 1 - C\tau/\hat{W}$ (since $\hat{p} > 0$ in steady state). Substitution yields

$$\hat{W}(\hat{W} - C\tau)^3 = \frac{\tau^3 c}{b},$$

which can be solved for $\hat{W}$ as a function of solely the system parameters c, b, C, and τ. This value can be used either with (2.10) or (2.11) to obtain a value for $\hat{s}$ solely as a function of the system parameters. This concludes the fixed point analysis. An interesting comparison is $\hat{W}$ as a function of $\hat{p}$ for Reno and CUBIC. Model (2.4) yields

$$\hat{W}_{Reno} = \sqrt{\frac{2}{\hat{p}}}, \quad \text{while} \quad \hat{W}_{CUBIC} = \sqrt[4]{\frac{\tau^3 c}{\hat{p}^3 b}}.$$

In other words, whereas throughput under Reno depends on loss probability as $\mathcal{O}(\hat{p}^{-1/2})$, CUBIC exhibits a $\hat{p}^{-3/4}$ dependence.

Next, we obtain the limiting behavior of $\hat{s}$.

Claim 2.4.1. $\hat{s} = \mathcal{O}((C\tau)^{1/3})$.

Proof. Above, we showed that

$$\hat{s} = \frac{\tau}{\hat{W}\hat{p}} = \frac{\tau}{\hat{W} - C\tau}. \tag{2.12}$$

Since $\hat{s} = (b\hat{W}/c)^{1/3}$, we also have that $\hat{W} = \hat{s}^3 c/b$. Substituting this expression for $\hat{W}$ into Eq. (2.12) and rearranging yields

$$\hat{s}^4 - \frac{b}{c} C\tau\hat{s} - \frac{b}{c}\tau = 0.$$

Given a quartic equation

$$c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 = 0,$$

the roots are

$$x_{1,2} = -\frac{c_3}{4c_4} - S \pm \frac{1}{2}\sqrt{-4S^2 - 2p + \frac{q}{S}},$$
$$x_{3,4} = -\frac{c_3}{4c_4} + S \pm \frac{1}{2}\sqrt{-4S^2 - 2p - \frac{q}{S}},$$

where,

$$p = \frac{8c_4c_2 - 3c_3^2}{8c_4^2}, \quad q = \frac{c_3^3 - 4c_4c_3c_2 + 8c_4^2c_1}{8c_4^3},$$
$$S = \frac{1}{2}\sqrt{-\frac{2}{3}p + \frac{1}{3c_4}\left(Q + \frac{\Delta_0}{Q}\right)}, \tag{2.13}$$
$$Q = \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}, \tag{2.14}$$
$$\Delta_0 = c_2^2 - 3c_3c_1 + 12c_4c_0, \text{ and}$$
$$\Delta_1 = 2c_2^3 - 9c_3c_2c_1 + 27c_3^2c_0 + 27c_4c_1^2 - 72c_4c_2c_0.$$

For CUBIC, $c_3 = 0$, $c_4 = 1$, $p = 0$, $q = -C\tau\frac{b}{c}$, $\Delta_0 = -12b\tau/c$, and $\Delta_1 = 27C^2(b\tau/c)^2$, so that

$$Q = \sqrt[3]{\frac{27}{2}\left(\frac{b}{c}C\tau\right)^2 + \frac{1}{2}\sqrt{27^2\left(\frac{b}{c}C\tau\right)^4 + 4\left(12\frac{b}{c}\tau\right)^3}} = \mathcal{O}((C\tau)^{2/3}),$$

and $\quad S = \frac{1}{2}\sqrt{\frac{1}{3}\left(Q - \frac{12b\tau}{c}\frac{1}{Q}\right)} = \mathcal{O}((C\tau)^{1/3})$.

Finally, to obtain $\hat{s}$, we choose the root $x_{3,4}$ with the plus sign and show later on that this root is indeed valid (*i.e.*, positive for any large $C\tau$). Thus,

$$\hat{s} = S + \frac{1}{2}\sqrt{-4S^2 - \frac{q}{S}} = \mathcal{O}((C\tau)^{1/3}). \tag{2.15}$$

□

In contrast, for TCP Reno it can be shown that $\hat{s} = \mathcal{O}(C\tau^2)$.

Claim 2.4.2. *for large $C\tau$, the quantity inside the square root in Eq. (2.15) is positive.*

Proof. Substituting the value for q above, we check if

$$\frac{C\tau b}{c}\frac{1}{S} - 4S^2 > 0. \tag{2.16}$$

When Eq. (2.16) holds, substituting for S and rearranging yield

$$\frac{2C\tau b}{c} > \left(\frac{1}{3}\left(Q - \frac{12b\tau}{c}\frac{1}{Q}\right)\right)^{3/2}.$$

Note that Q is always positive. Hence, it suffices to show

$$\frac{3^{3/2}2C\tau b}{c} > Q^{3/2}.$$

Substituting for Q and dividing by $\sqrt{27}$ yields

$$\frac{2C\tau b}{c} > \frac{1}{\sqrt{2}}\sqrt{\left(\frac{b}{c}C\tau\right)^2 + \sqrt{\left(\tau\frac{b}{c}\right)^3\left(C^4\tau\frac{b}{c} + \frac{256}{27}\right)}}$$

For large $C\tau$,

$$C^4\tau\frac{b}{c}+\frac{256}{27}<2C^4\tau\frac{b}{c},$$

so it suffices to show

$$\frac{2C\tau b}{c}>\frac{1}{\sqrt{2}}\sqrt{\frac{b^2}{c^2}C^2\tau^2(1+\sqrt{2})}.$$

Noting that $1+\sqrt{2}<4$, it suffices to show that

$$\frac{2C\tau b}{c}>\sqrt{2\frac{b^2}{c^2}C^2\tau^2}=\sqrt{2}\frac{b}{c}C\tau,$$
$$2>\sqrt{2}\ \checkmark$$

Hence, for large $C\tau$, we have selected the correct root for computing $\hat{s}$. □

2.4.3 Change of Variables

To simplify stability analysis, we perform a change of variables so that the fixed point of the system is located at the origin. To accomplish this, define $\mathbf{x}$ as follows:

$$\mathbf{x}(t)=\begin{bmatrix}x_1(t)\\ \\ x_2(t)\end{bmatrix}=\begin{bmatrix}W_{\max}(t)-\hat{W}_{\max}\\ \\ s(t)-\hat{s}\end{bmatrix}=\begin{bmatrix}W_{\max}(t)-\hat{W}\\ \\ s(t)-\hat{s}\end{bmatrix}$$

where the last equality follows because $\hat{W}_{\max}=\hat{W}$. Also, define $\Psi(t)$ and $\tilde{p}(t)$ as follows:

$$\Psi(t)=c\left(x_2(t)+\hat{s}-\sqrt[3]{\frac{b(x_1(t)+\hat{W})}{c}}\right)^3+x_1(t)+\hat{W},$$
$$\tilde{p}(t)=\max\left(1-\frac{C\tau}{\Psi(t)},0\right).$$

Then the new system is:

$$\begin{aligned} \dot{x}_1 &= \left(\Psi - x_1 - \hat{W}\right) \frac{\Psi_\tau}{\tau} \tilde{p}_\tau, \\ \dot{x}_2 &= 1 - (x_2 + \hat{s}) \frac{\Psi_\tau}{\tau} \tilde{p}_\tau. \end{aligned} \tag{2.17}$$

Note that Ψ_τ and $\tilde{p}_\tau$ are functions of $x_1(t-\tau) \equiv x_{1_\tau}$ and $x_2(t-\tau) \equiv x_{2_\tau}$. It is easy to verify that $\mathbf{x}^* = [x_1 \; x_2 \; x_{1_\tau} \; x_{2_\tau}]^T = \mathbf{0}$ is a fixed point of the new system.

Claim 2.4.3. $\mathbf{x}^* = \mathbf{0}$ *is a fixed point of the new system.*

Proof. At $\mathbf{x}^*$, we have:

$$\hat{\Psi} = c \left(\hat{s} - \sqrt[3]{\frac{b\hat{W}}{c}} \right)^3 + \hat{W}$$

$$\dot{x}_1 = \left(\Psi - \hat{W}\right) \frac{\Psi_\tau}{\tau} \tilde{p}_\tau \tag{2.18}$$

$$\dot{x}_2 = 1 - \hat{s} \frac{\Psi_\tau}{\tau} \tilde{p}_\tau \tag{2.19}$$

From the fixed point analysis of the original system, recall that $\hat{s} = \sqrt[3]{b\hat{W}/c}$, which yields $\hat{\Psi} = \hat{W}$. Plugging this into equation (2.18), we get $\dot{x}_1 = 0$. Similarly, plugging in $\hat{\Psi} = \hat{W}$ into (2.19), we have:

$$\dot{x}_2 = 1 - \hat{s} \frac{\hat{W}}{\tau} \hat{p}.$$

From the fixed point analysis of the original system, recall that $\hat{s} = \tau/(\hat{W}\hat{p})$. Therefore, $\dot{x_2} = 0$ as well, and the proof is complete. □

We can analyze the stability of the system (2.17) at the origin, which is equivalent to analyzing the stability of the original system (2.1) at the equilibrium values $\hat{W}_{\max}$ and $\hat{s}$. The CUBIC representation in (2.17) forms the basis for our subsequent analyses.

2.4.4 Existence and Uniqueness of Solution

We state the existence and uniqueness theorem as it appears in [24]:

Theorem 2.4.1 (Theorem 1.2 from *Stability of Time-Delay Systems*)**.** *Suppose that* Ω *is an open set in* $\mathbb{R} \times \mathcal{C}$ *(where* $\mathcal{C}$ *is the set of* $\mathbb{R}^n$*-valued continuous functions on* $[-\tau, 0]$*),* $f : \Omega \to \mathbb{R}^n$ *is continuous, and* $f(t, \phi)$ *is Lipschitzian in* ϕ *in each compact set in* Ω*, that is, for each given compact set* $\Omega_0 \subset \Omega$*, there exists a constant* L *such that*

$$||f(t, \phi_1) - f(t, \phi_2)|| \leq L||\phi_1 - \phi_2||$$

for any $(t, \phi_1) \in \Omega_0$ *and* $(t, \phi_2) \in \Omega_0$*. If* $(t_0, \phi) \in \Omega$*, then there exists a unique solution of* $\dot{x}(t) = f(t, x_t)$ *through* (t_0, ϕ)*.*

To prove existence and uniqueness for our system, it is sufficient to show that $\dot{x}_1$ and $\dot{x}_2$ are continuously differentiable functions in some neighborhood of the fixed point. We assume that this neighborhood is small enough so that $\tilde{p}_\tau > 0$. Then the system becomes:

$$\begin{aligned} \dot{x}_1 &= \left(\Psi - x_1 - \hat{W}\right) \frac{(\Psi_\tau - C\tau)}{\tau}, \\ \dot{x}_2 &= 1 - (x_2 + \hat{s}) \frac{(\Psi_\tau - C\tau)}{\tau}. \end{aligned}$$

Let

$$F \coloneqq \frac{b(x_1 + \hat{W})}{c} \quad \text{and} \quad \Phi \coloneqq x_2 + \hat{s} - \sqrt[3]{F}.$$

Following our usual convention,

$$\Phi \equiv \Phi(t), \quad \text{and} \quad \Phi_\tau \equiv \Phi(t - \tau).$$

Following are the partial derivatives of $\dot{x}_1$:

$$\begin{aligned}
\frac{\partial \dot{x}_1}{\partial x_1} &= -\frac{b}{\tau}\Phi^2 F^{-2/3}(\Psi_\tau - C\tau), \\
\frac{\partial \dot{x}_1}{\partial x_2} &= \frac{3c}{\tau}\Phi^2(\Psi_\tau - C\tau), \\
\frac{\partial \dot{x}_1}{\partial x_{1_\tau}} &= \frac{(\Psi - x_1 - \hat{W})}{\tau}\left(-b\Phi_\tau^2 F_\tau^{-2/3} + 1\right), \\
\frac{\partial \dot{x}_1}{\partial x_{2_\tau}} &= \frac{3c}{\tau}(\Psi - x_1 - \hat{W})\Phi_\tau^2.
\end{aligned}$$

These partials provide the first restriction to the region where stability is being analyzed. Specifically, the term $F^{-2/3}$ indicates that x_1 and x_{1_τ} should be restricted to an interval $[-\rho\hat{W}, \rho\hat{W}]$, $0 < \rho < 1$. Next, we look at the partial derivatives of $\dot{x}_2$:

$$\begin{aligned}
\frac{\partial \dot{x}_2}{\partial x_1} &= 0, \ \frac{\partial \dot{x}_2}{\partial x_2} = -\frac{(\Psi_\tau - C\tau)}{\tau}, \\
\frac{\partial \dot{x}_2}{\partial x_{1_\tau}} &= -\frac{(x_2 + \hat{s})}{\tau}\left(-b\Phi_\tau^2 F_\tau^{-2/3} + 1\right), \\
\frac{\partial \dot{x}_2}{\partial x_{2_\tau}} &= -\frac{3c}{\tau}(x_2 + \hat{s})\Phi_\tau^2.
\end{aligned}$$

Under the restriction stated above, these partials are also continuous, and hence, we have local Lipschitz continuity – the requirement for existence and uniqueness.

2.4.5 Stability Analysis

In general, the linearization of (2.17) and (2.3) about $\mathbf{x} = \mathbf{x}^* = \mathbf{0}$ is

$$\begin{bmatrix} \dot{x}_1 \\ \\ \dot{x}_2 \end{bmatrix} = \frac{1}{\hat{s}} A_0 \begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix} - \frac{\hat{s}}{\tau} A_1 \begin{bmatrix} x_{1_\tau} \\ \\ x_{2_\tau} \end{bmatrix},$$

where

$$A_0 = \begin{bmatrix} \frac{\partial \Psi}{\partial x_1} - 1 & \frac{\partial \Psi}{\partial x_2} \\ \\ 0 & -1 \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}^*} \quad \text{and} \quad A_1 = \begin{bmatrix} 0 & 0 \\ \\ \frac{\partial \Psi_\tau}{\partial x_{1_\tau}} & \frac{\partial \Psi_\tau}{\partial x_{2_\tau}} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}^*}.$$

For CUBIC, $\partial\Psi/\partial x_1|_{\mathbf{x}=\mathbf{x}^*} = 1$ and $\partial\Psi/\partial x_2|_{\mathbf{x}=\mathbf{x}^*} = 0$, so that $\dot{x}_1 = 0$. This means that the linearized system cannot be used to deduce the local stability of the nonlinear system. The key cause of this problem is the fact that the fixed point value of $x_2 = 0$, is the saddle point of the function Ψ (or equivalently, $\hat{s}$ is the saddle point of $W(t)$). Figure 2.2 illustrates this phenomenon. This causes all first-order partial derivatives of $\dot{x}_1$ to evaluate to zero at $\mathbf{x}^* = \mathbf{0}$. Hence, in order to incorporate a local contribution from $\dot{x}_1$ in the analysis, it is necessary to expand $\dot{x}_1$ further. Specifically, a third-order Taylor Series expansion is necessary, since all second-order terms also evaluate to zero at the origin. The expanded system looks as follows:

$$\begin{aligned} \dot{x}_1 &= -\alpha x_1^3 + \beta x_1^2 x_2 - \gamma x_1 x_2^2 + \delta x_2^3 + h_1 \\ \dot{x}_2 &= -\frac{1}{\hat{s}} x_2 - \frac{\hat{s}}{\tau} x_{1_\tau} + h_2 \end{aligned} \tag{2.20}$$

where $\alpha = \dfrac{b^3}{27c^2\hat{s}^7}$, $\beta = \dfrac{b^2}{3c\hat{s}^5}$, $\gamma = \dfrac{b}{\hat{s}^3}$, and $\delta = \dfrac{c}{\hat{s}}$

and h_1 and h_2 are higher-order terms of $\dot{x}_1$ and $\dot{x}_2$, respectively. To analyze the stability of (2.20), we use the Lyapunov-Razumikhin Theorem, the statement of which is given below as it appears in [24]. For the purpose of this theorem, we introduce some notation. Let $\mathcal{C} = \mathcal{C}([-\tau, 0], \mathbb{R}^n)$ be the set of continuous functions mapping the interval $[-\tau, 0]$ to $\mathbb{R}^n$, where τ is the maximum delay of a system. For any $A > 0$ and any continuous function of time $\psi \in \mathcal{C}([t_0 - \tau, t_0 + A], \mathbb{R}^n)$, and $t_0 \leq t \leq t_0 + A$, let $\psi_t \in \mathcal{C}$ be a segment of the function ψ defined as $\psi_t(\theta) = \psi(t + \theta)$, $-\tau \leq \theta \leq 0$. The general form of a retarded functional differential equation is

$$\dot{x}(t) = f(t, x_t). \tag{2.21}$$

Below, $\mathbb{R}_+$ is the set of positive real numbers, and $\bar{\mathbb{S}}$ is the closure of the set $\mathbb{S}$.

Theorem 2.4.2 (Lyapunov-Razumikhin Theorem)**.** *Suppose* $f : \mathbb{R} \times \mathcal{C} \to \mathbb{R}^n$ *takes* $\mathbb{R} \times$ *(bounded sets of* $\mathcal{C}$*) into bounded sets of* $\mathbb{R}^n$*, and* u*,* v*,* $w : \bar{\mathbb{R}}_+ \to \bar{\mathbb{R}}_+$ *are continuous nondecreasing functions,* $u(s)$ *and* $v(s)$ *are positive for* $s > 0$*, and* $u(0) = v(0) = 0$*,* v *strictly increasing. If there exists a continuously differentiable function* $V : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ *such that*

$$u(||x||) \leq V(t, x) \leq v(||x||), \quad \textit{for } t \in \mathbb{R} \textit{ and } x \in \mathbb{R}^n, \tag{2.22}$$

$w(s) > 0$ *for* $s > 0$*, and there exists a continuous nondecreasing function* $p(s) > s$ *for* $s > 0$ *such that*

$$\dot{V}(t, x(t)) \leq -w(||x(t)||) \tag{2.23}$$

$$\textit{if } V(t + \theta, x(t + \theta)) \leq p(V(t, x(t))) \tag{2.24}$$

for $\theta \in [-\tau, 0]$*, then the system (2.21) is uniformly asymptotically stable. If in addition* $\lim_{s\to\infty} u(s) = \infty$*, then the system (2.21) is globally uniformly asymptotically stable.*

The continuous nondecreasing function $p(s)$ defined in the theorem above is not to be confused with the loss probability function defined for the MWLI model earlier in the text. Note that in this work, we will only prove *local* stability for CUBIC. Therefore, our goal is to show that we can find a function V for which all conditions specified in the theorem are valid locally, *i.e.*, in a sufficiently small neighborhood around the fixed point.

A typical choice of Lyapunov candidate is the quadratic form, *i.e.*, a function of the form

$$Z(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}, \quad \text{where} \quad P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_4 \end{bmatrix} \tag{2.25}$$

is positive definite. Not surprisingly, the quadratic form Z, which is a sufficient form in working with linear dynamic systems, proves unsuitable. To understand why, consider the time derivative of (2.25) along solutions to (2.20):

$$\dot{Z} = 2\dot{x}_1(p_1x_1 + p_2x_2) + 2\dot{x}_2(p_2x_1 + p_4x_2). \tag{2.26}$$

The first term above is quartic in x_1 and x_2 (because $\dot{x}_1$ in (2.20) is cubic in x_1, x_2), but the second term is quadratic in x_1, x_2, and x_{1_τ}. In a small neighborhood of $\mathbf{x}^*$, the quadratic terms dominate; *i.e.*, (2.26) becomes

$$\dot{Z} = 2\left(-\frac{1}{\hat{s}}x_2 - \frac{\hat{s}}{\tau}x_{1_\tau}\right)(p_2x_1 + p_4x_2) + h.o.t.,$$

where *h.o.t.* denotes higher-order terms. We cannot guarantee negativity of these terms, even locally. The main problem with $\dot{Z}$ is that p_2 must be non-zero for $\dot{Z}$ to be negative definite, yet this is the same coefficient responsible for the cross term of x_1 and x_2 in (2.25), which prevents us from effectively bounding x_{1_τ} using condition (2.24) of Theorem 2.4.2. However, the failure of this quadratic Lyapunov function serves as an instructive example. Namely, we would like a Lyapunov candidate to have the following two properties: (i) it must prevent $\dot{x}_2$'s terms from dominating the Lyapunov derivative, and (ii) the cross terms of x_1 and x_2 in the Lyapunov function should be absent so that delayed terms (like x_{1_τ}) can be easily bounded with non-delayed versions (like x_1) using (2.24). With these motivations, consider the following Lyapunov-Razumikhin candidate:

$$V(\mathbf{x}) = \frac{d_1}{2}x_1^2 + \frac{d_4}{4}x_2^4 \tag{2.27}$$

where d_1 and d_4 are positive. In a subsequent discussion, we will specify the values of d_1 and d_4 in terms of system parameters. We will also show that V can be bounded by

functions $v(||\mathbf{x}||) = \epsilon_0||\mathbf{x}||_2^2$ and $u(||\mathbf{x}||) = \epsilon_1||\mathbf{x}||_2^4$, for appropriate choices of constants ϵ_0 and ϵ_1, and that these functions satisfy all conditions specified in Theorem 2.4.2. V satisfies (ii), as necessary, and allows us to choose a convenient function $p(V(\mathbf{x}(t)))$ for (2.24) (note: we can write $V(\mathbf{x}(t))$ instead of $V(t, \mathbf{x}(t))$ because V is autonomous, *i.e.*, it is not explicitly a function of time). Let $p > 1$ be a constant, which can be arbitrarily close to one. Then for (2.24), we can use $p(V(\mathbf{x}(t))) = pV(\mathbf{x}(t))$:

$$V(\mathbf{x}(t-\theta)) \leq pV(\mathbf{x}(t)), \text{ for } \theta \in [0, \tau].$$

Since there are no cross terms of x_{1_θ} and x_{2_θ} on the left-hand side of the above inequality, bounding the absolute values of these delayed variables individually is straightforward (and instrumental in the proofs that follow).

Now, consider the Lyapunov derivative:

$$\dot{V} = d_1 x_1 \dot{x}_1 + d_4 x_2^3 \dot{x}_2.$$

Note that both of the terms above are now *quartic* in either x_1, x_2, or both. However, $\dot{x}_2$ still contributes a term with x_{1_τ}, which poses a challenge in proving local stability. Indeed, at the core of the proof for $\dot{V}$'s negativity is managing the x_{1_τ} term, as well as proving that h_1 and h_2 in (2.20), which are higher-order in both the delayed and non-delayed variables, are also higher-order in *only* the non-delayed variables x_1 and x_2.

The focus of the following discussion is the term that contains x_{1_τ}. Substituting the expanded system (2.20) into $\dot{V}$ and rearranging terms, we have

$$
\begin{aligned}
\dot{V} &= d_1 x_1 \left(-\alpha x_1^3 + \beta x_1^2 x_2 - \gamma x_1 x_2^2 + \delta x_2^3\right) + d_4 x_2^3 \left(-\frac{1}{\hat{s}} x_2 - \frac{\hat{s}}{\tau} x_{1_\tau}\right) + d_1 x_1 h_1 + d_4 x_2^3 h_2 \\
&= d_1 \left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2 + \delta x_1 x_2^3\right) - \frac{d_4}{\hat{s}} x_2^4 - d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2 \\
&= d_1 \left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2\right) - \frac{d_4}{\hat{s}} x_2^4 + d_1 \delta x_1 x_2^3 - d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2 \\
&= \mathbf{y}^T Q \mathbf{y} + d_1 \delta x_1 x_2^3 - d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2
\end{aligned}
$$

$$
\text{where} \quad \mathbf{y} = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} -d_1\alpha & d_1\beta/2 & 0 \\ d_1\beta/2 & -d_1\gamma & 0 \\ 0 & 0 & -d_4/\hat{s} \end{bmatrix}.
$$

For the following, we will make use of the Mean Value Theorem (MVT), stated below.

Theorem 2.4.3 (Mean Value Theorem)**.** *Let $f : [a, b] \to \mathbb{R}$ be a continuous function on the closed interval $[a, b]$, and differentiable on the open interval (a, b), where $a < b$. Then there exists some c in (a, b) such that*

$$
f'(c) = \frac{f(b) - f(a)}{b - a}.
$$

Let I be the interval $[t - \tau, t]$. Then by the MVT, there exists some $\theta \in (0, \tau)$ such that

$$
\begin{aligned}
\dot{x}_1(t - \theta) &= \frac{x_1(t) - x_1(t - \tau)}{t - (t - \tau)} = \frac{x_1(t) - x_1(t - \tau)}{\tau}, \\
x_1(t - \tau) &= x_1(t) - \dot{x}_1(t - \theta)\tau, \ \theta \in (0, \tau), \\
&\text{or } x_{1_\tau} = x_1 - \dot{x}_{1_\theta}\tau,
\end{aligned}
$$

where $\dot{x}_{1_\theta} = \dot{x}_1(t - \theta)$. We would like to combine the terms $d_1 \delta x_1 x_2^3$ and $-d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau}$ in $\dot{V}$ using the MVT. To do so, let $d_1 = 1/\delta = \hat{s}/c$ and $d_4 = \tau/\hat{s}$.

$$\begin{aligned}\dot{V} &= \mathbf{y}^T Q\mathbf{y} + x_1 x_2^3 - x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2 \\ &= \mathbf{y}^T Q\mathbf{y} + x_2^3(x_1 - x_{1_\tau}) + d_1 x_1 h_1 + d_4 x_2^3 h_2 \\ &= \mathbf{y}^T Q\mathbf{y} + x_2^3 \dot{x}_{1_\theta}\tau + d_1 x_1 h_1 + d_4 x_2^3 h_2.\end{aligned}$$

Note that the last three terms above all have dependencies on x_{1_τ} and x_{2_τ}. Our goal is to show that these terms are of higher order than $\mathbf{y}^T Q\mathbf{y}$ in variables x_1 and x_2 alone. Consider $\dot{x}_{1_\theta}$:

$$\dot{x}_{1_\theta} = \Phi_\theta^3 \frac{\Psi_{\theta+\tau}}{\tau}\tilde{p}_{\theta+\tau},$$

where Φ_θ is the delayed version of Φ defined in Section 2.4.4. We would like to find an upper bound for $|\dot{x}_{1_\theta}|$ only in terms of x_{1_θ} and x_{2_θ}. Since $\theta \in (0, \tau)$, we can expand Φ_θ^3 about $\mathbf{v} = [x_{1_\theta}\ x_{2_\theta}] = [0\ 0]$ (in other words, the fixed point implicitly includes all $x_1(t-\xi)$, $x_2(t-\xi)$, $0 \le \xi \le \tau$, not just $x_1(t)$, $x_2(t)$, $x_1(t-\tau)$, and $x_2(t-\tau)$). Note also that by performing this expansion, we are applying a two-variable Taylor series expansion to a four-variable function. Specifically, we can use a second-order expansion and bound $|\dot{x}_{1_\theta}|$ using only the remainder, which consists of third-order partial derivatives.

In the expressions that follow, we use F and Φ as defined in Section 2.4.4. Also, let $\Gamma \coloneqq \Psi_{\theta+\tau}\tilde{p}_{\theta+\tau}/\tau$. The zero-, first-, and second-order terms in the expansion of $\dot{x}_{1_\theta}$ are zero when evaluated at $\mathbf{v} = \mathbf{0}$. The third-order partial derivatives are:

$$\begin{aligned}\frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{1_\theta}^3} &= \frac{-2b^3}{9c^2}\Gamma\left[F_\theta^{-2} + 6\Phi_\theta F_\theta^{-7/3} + 5\Phi_\theta^2 F_\theta^{-8/3}\right], \\ \frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{1_\theta}^2 x_{2_\theta}} &= \frac{2b^2}{3c}\Gamma\left[F_\theta^{-4/3} + 2\Phi_\theta F_\theta^{-5/3}\right], \\ \frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{1_\theta} x_{2_\theta}^2} &= -2b\Gamma F_\theta^{-2/3}, \quad \text{and} \\ \frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{2_\theta}^3} &= 6c\Gamma.\end{aligned}$$

We will bound the absolute values of these partial derivatives and use the following proposition [19].

Proposition 2.4.1. *If a function f is of class C^{k+1} on an open convex set S and $|\partial^\alpha f(\mathbf{x})| \leq M$ for $\mathbf{x} \in S$ and $|\alpha| = k+1$, then the absolute value of the remainder $R_{\mathbf{a},k}(\mathbf{h})$ of the kth-order Taylor series expansion of f about the point $\mathbf{a}$ can be bounded as follows:*

$$|R_{\mathbf{a},k}(\mathbf{h})| \leq \frac{M}{(k+1)!}\|\mathbf{h}\|^{k+1}, \text{ where}$$
$$\|\mathbf{h}\| = |h_1| + |h_2| + \cdots + |h_n|.$$

Above, ∂f^α is the generic $(k+1)$th-order partial derivative of f, and $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$. In our case, $\mathbf{a} = \mathbf{0}$, and $\mathbf{h} = [x_{1_\theta}\ x_{2_\theta}]$.

In three of these partial derivatives, there are terms of the form $\left(\frac{b(x_{1_\theta}+\hat{W})}{c}\right)^{-l}$, where l is a positive rational number. Hence, we must bound x_{1_θ} in a region $[-\rho\hat{W}, \rho\hat{W}]$, where $0 < \rho < 1$. Assuming that x_{1_θ}, x_{2_θ}, $x_{1_{\theta+\tau}}$, and $x_{2_{\theta+\tau}}$ are constrained in an appropriately-chosen local region $[-r, r]$ around $\mathbf{0}$, there exists a constant M such that $|\partial^3 f| \leq M$. Then, using the proposition,

$$|\dot{x}_{1_\theta}| \leq \frac{M}{3!}(|x_{1_\theta}| + |x_{2_\theta}|)^3. \tag{2.28}$$

By Razumikhin's Theorem, we require that $\dot{V}(\mathbf{x}) \leq -w(\|\mathbf{x}\|)$ whenever $pV(\mathbf{x}(t)) \geq V(\mathbf{x}(t-\theta))$, $\theta \in (0, \tau)$, for an $\epsilon > 0$ and some constant $p > 1$. When $pV(\mathbf{x}(t)) \geq V(\mathbf{x}(t-\theta))$,

$$p\left(\frac{d_1}{2}x_1^2+\frac{d_4}{4}x_2^4\right)\geq\frac{d_1}{2}x_{1_\theta}^2+\frac{d_4}{4}x_{2_\theta}^4,$$
$$p\left(d_1x_1^2+\frac{d_4}{2}x_2^4\right)\geq d_1x_{1_\theta}^2,$$
$$|x_{1_\theta}|\leq\sqrt{\frac{p}{d_1}}\sqrt{\left(d_1x_1^2+\frac{d_4}{2}x_2^4\right)}\leq\sqrt{\frac{p}{d_1}}\left(\sqrt{d_1x_1^2}+\sqrt{\frac{d_4}{2}x_2^4}\right)$$
$$\leq\sqrt{\frac{p}{d_1}}\left(\sqrt{d_1}|x_1|+\sqrt{\frac{d_4}{2}}x_2^2\right).$$

Similarly,

$$p\left(d_1x_1^2+\frac{d_4}{2}x_2^4\right)\geq\frac{d_4}{2}x_{2_\theta}^4,$$
$$x_{2_\theta}^4\leq\frac{2p}{d_4}\left(d_1x_1^2+\frac{d_4}{2}x_2^4\right),$$
$$|x_{2_\theta}|\leq\sqrt[4]{\frac{2p}{d_4}}\sqrt[4]{\left(d_1x_1^2+\frac{d_4}{2}x_2^4\right)}\leq\sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1x_1^2}+\sqrt[4]{\frac{d_4}{2}x_2^4}\right)$$
$$\leq\sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1}|x_1|^{1/2}+\sqrt[4]{\frac{d_4}{2}}|x_2|\right).$$

Substituting these results into (2.28), we have:

$$|\dot{x}_{1_\theta}|\leq\frac{M}{6}\left(\sqrt{\frac{p}{d_1}}\left(\sqrt{d_1}|x_1|+\sqrt{\frac{d_4}{2}}x_2^2\right)+\sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1}|x_1|^{1/2}+\sqrt[4]{\frac{d_4}{2}}|x_2|\right)\right)^3$$
$$=\frac{M}{6}\left(\sqrt{p}|x_1|+\sqrt{\frac{pd_4}{2d_1}}x_2^2+\sqrt[4]{\frac{2pd_1}{d_4}}|x_1|^{1/2}+\sqrt[4]{p}|x_2|\right)^3.$$

The lowest-order term in the equation above is $(|x_1|^{1/2})^3=|x_1|^{3/2}$. Hence, we see that the term $|x_2^3\dot{x}_{1_\theta}\tau|$ can be bounded by a function of order at least 4.5. We can use a similar procedure to bound the remainders of $\dot{x}_1$ and $\dot{x}_2$, as we will demonstrate.

First, consider the higher-order terms in $\dot{x}_1$. Since x_{2_θ} depends on $\sqrt{|x_1|}$, we cannot simply use a third-order expansion of $\dot{x}_1$ and bound the remainder of fourth-order partials, because a consequence of this is that the remainder will have a term

$\left(|x_1|^{1/2}\right)^4 = |x_1|^2$. Recall that in the Lyapunov derivative, $\dot{x}_1$ is being multiplied by x_1, so the resulting term will have an order of merely three. Using this logic, it is clear that we need an expansion of at least order six; this way, the lowest-order term in the remainder will be $(|x_1|^{1/2})^7 = |x_1|^{7/2}$, and $x_1|x_1|^{7/2}$ is order 4.5, which is sufficient. However, it is not enough to simply do a sixth-order expansion of $\dot{x}_1$: we must also ensure that any fourth-, fifth-, and sixth-order partial derivatives in the expansion in terms of x_1, x_2, x_{1_τ}, and x_{2_τ}, are of order 3.5 or more in terms of only x_1 and x_2 (so that when multiplied by x_1 in the Lyapunov derivative, we have terms of order at least 4.5).

Claim 2.4.4. *Except for the third-order terms, the sixth-order expansion of $\dot{x}_1$ in $[x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}]$ is of combined power at least 3.5 in $[x_1\ x_2]$.*

The proof for this claim is very similar to the above procedure for bounding $|\dot{x}_{1_\theta}|$. See Appendix A.2.1 for a full proof of Claim 2.4.4.

Next, we analyze the higher-order terms and remainder of $\dot{x}_2$. Recall that in the Lyapunov derivative, $\dot{x}_2$ is being multiplied by x_2^3. Hence, we require a second-order expansion of $\dot{x}_2$ about $[x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}] = \mathbf{0}$, and we must ensure that the second-order terms are of combined power greater than one.

Claim 2.4.5. *Except for the first-order terms, the second-order expansion of $\dot{x}_2$ in $[x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}]$ is of combined power at least 1.5 in $[x_1\ x_2]$.*

The proof for this claim is very similar to that of Claim 2.4.4. See Appendix A.2.2 for a full proof of Claim 2.4.5.

Finally, we show that the sum of the terms of order four is negative by proving that Q in $\mathbf{y}^T Q \mathbf{y}$ is negative definite.

$$\mathbf{y}^T Q \mathbf{y} = \begin{bmatrix} x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} -d_1\alpha & d_1\beta/2 & 0 \\ d_1\beta/2 & -d_1\gamma & 0 \\ 0 & 0 & -d_4/\hat{s} \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}.$$

This first leading principal minor of Q, $-d_1\alpha$, is always negative, as needed. The second leading principal minor should be positive:

$$\begin{aligned} d_1^2\alpha\gamma - d_1^2\frac{\beta^2}{4} &\stackrel{?}{>} 0 \\ \alpha\gamma - \frac{\beta^2}{4} &\stackrel{?}{>} 0 \\ \left(\frac{b^3}{27c^2\hat{s}^7}\right)\left(\frac{b}{\hat{s}^3}\right) - \frac{1}{4}\left(\frac{b^2}{3c\hat{s}^5}\right)^2 &\stackrel{?}{>} 0 \\ \frac{b^4}{27c^2\hat{s}^{10}} - \frac{1}{4}\left(\frac{b^4}{9c^2\hat{s}^{10}}\right) &\stackrel{?}{>} 0 \\ \frac{1}{27} - \frac{1}{36} &\stackrel{?}{>} 0 \ \checkmark \end{aligned}$$

The third leading principal minor should be negative:

$$\begin{aligned} -d_1\alpha\left(d_1 d_4\frac{\gamma}{\hat{s}}\right) - d_1\frac{\beta}{2}\left(d_1\frac{\beta}{2}\left(-\frac{d_4}{\hat{s}}\right)\right) &\stackrel{?}{<} 0 \\ -\alpha\left(\frac{\gamma}{\hat{s}}\right) - \frac{\beta}{2}\left(\frac{\beta}{2}\left(-\frac{1}{\hat{s}}\right)\right) &\stackrel{?}{<} 0 \\ -\alpha\gamma + \frac{\beta^2}{4} &\stackrel{?}{<} 0 \\ \alpha\gamma - \frac{\beta^2}{4} &\stackrel{?}{>} 0 \end{aligned}$$

We see that this condition is equivalent to the previous one (for the second leading principal minor), and hence it is satisfied.

So far, we have shown that for a sufficiently small neighborhood of $\mathbf{x}^* = \mathbf{0}$, the Lyapunov-Razumikhin candidate in (2.27) has a negative definite derivative, *i.e.*, $\dot{V}(\mathbf{x}) < 0$ for all $\mathbf{x} \neq \mathbf{0}$ in this neighborhood and $\dot{V}(\mathbf{x}) = 0$ if $\mathbf{x} = \mathbf{0}$. Finally, it remains

to bound V and $\dot{V}$ with functions u, v, and w that satisfy all conditions specified in Theorem 2.4.2. We note that the arguments above are valid for $|x_1|,\ |x_2| < r$, where $0 < r < 1$. We keep this in mind for the following bounds. Recall that previously, we let $d_1 = \hat{s}/c$ and $d_4 = \tau/\hat{s}$. Then the exact form of the Lyapunov-Razumikhin function is

$$V(\mathbf{x}) = \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4.$$

For all bounds, we can use any norm on $\mathbf{x}$, as long as we are being consistent. We choose the l_2-norm. Let

$$v(||\mathbf{x}||) = \epsilon_0||\mathbf{x}||_2^2 = \epsilon_0(x_1^2 + x_2^2),$$

where $\epsilon_0 = \max\left(\frac{\hat{s}}{2c}, \frac{\tau}{4\hat{s}}\right)$. Then $V(\mathbf{x}) \leq v(||\mathbf{x}||)$. Also, $v(||\mathbf{x}||)$ is strictly increasing, $v(||\mathbf{0}||) = 0$, and $v(||\mathbf{x}||)$ is positive for $||\mathbf{x}|| > 0$. Next, let

$$u(||\mathbf{x}||) = \epsilon_1||\mathbf{x}||_2^4 = \epsilon_1(x_1^2 + x_2^2)^2 = \epsilon_1(x_1^4 + 2x_1^2x_2^2 + x_2^4),$$

where ϵ_1 is a positive constant of our choice. We will show that $V(\mathbf{x}) \geq u(||\mathbf{x}||)$, or equivalently, $V(\mathbf{x}) - u(||\mathbf{x}||) \geq 0$ for some choice of ϵ_1.

$$\begin{aligned} V(\mathbf{x}) - u(||\mathbf{x}||) &= \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4 - \epsilon_1(x_1^4 + 2x_1^2x_2^2 + x_2^4) \\ &\geq \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4 - \epsilon_1(x_1^2 + 2x_1^2 + x_2^4) \\ &= \left(\frac{\hat{s}}{2c} - 3\epsilon_1\right)x_1^2 + \left(\frac{\tau}{4\hat{s}} - \epsilon_1\right)x_2^4 \\ &\geq 0 \text{ for } \epsilon_1 < \min\left(\frac{\hat{s}}{6c}, \frac{\tau}{4\hat{s}}\right). \end{aligned}$$

Also, $u(||\mathbf{x}||)$ is positive for $||\mathbf{x}|| > 0$ and $u(||\mathbf{0}||) = 0$. Hence, condition (2.22) is satisfied.

Next, we show that $\dot{V}$ is bounded by a suitable function w as specified in Theorem 2.4.2. Recall that we have shown that

$$\dot{V} = d_1\left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2\right) - \frac{d_4}{\hat{s}} x_2^4 + h.o.t.$$

We can express the lower-order terms in matrix form, as follows:

$$\dot{V} = -\begin{bmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} d_1\alpha & -\frac{d_1\beta}{2\sqrt{2}} & 0 \\ -\frac{d_1\beta}{2\sqrt{2}} & \frac{d_1\gamma}{2} & 0 \\ 0 & 0 & d_4/\hat{s} \end{bmatrix} \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} + h.o.t.$$

Let's call the matrix above $\tilde{Q}$. Since we have previously shown that the sum of the lower-order terms in $\dot{V}$ is a negative definite function, it follows that $\tilde{Q}$ is positive definite, so all of its eigenvalues are strictly positive. Since $\tilde{Q}$ is a real and symmetric matrix, its Rayleigh quotient is bounded below by $\lambda_{\min}[\tilde{Q}]$. Hence,

$$\begin{aligned} \dot{V} &\leq -\lambda_{\min}[\tilde{Q}] \left|\left| \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} \right|\right|_2^2 + h.o.t. \\ &= -\lambda_{\min}[\tilde{Q}](x_1^4 + 2x_1^2x_2^2 + x_2^4) + h.o.t. \\ &= -\lambda_{\min}[\tilde{Q}]||\mathbf{x}||_2^4 + h.o.t. \end{aligned}$$

Previously, we showed that the higher-order terms have orders of at least 4.5. Then for $|x_1|$, $|x_2|$ small enough, there exists a positive constant K such that

$$h.o.t. \leq K||\mathbf{x}||_2^4.$$

Claim 2.4.6.

$$\textit{Let} \quad f(x_1, x_2) = \frac{h.o.t.}{||\mathbf{x}||_2^4}. \quad \textit{Then} \quad \lim_{(x_1,x_2)\to(0,0)} \frac{h.o.t.}{||\mathbf{x}||_2^4} = 0.$$

A simple $\epsilon - \delta$ proof can be used for this claim. See Appendix A.2.3 for the proof of Claim 2.4.6.

By Claim 2.4.6, we can always find a K small enough by restricting x_1 and x_2 into a smaller neighborhood around $\mathbf{0}$. Hence, we can find a $K < \lambda_{\min}[\tilde{Q}]$, which would give us

$$\dot{V} \leq -(\lambda_{\min}[\tilde{Q}] - K)||\mathbf{x}||_2^4$$

where $\lambda_{\min}[\tilde{Q}] - K > 0$. By inspection, we have found a $w(||\mathbf{x}||)$ that satisfies condition (2.23) under (2.24). In addition, $w(||\mathbf{x}||)) > 0$ when $||\mathbf{x}|| > 0$, as necessary. Finally, $\lim_{||\mathbf{x}||\to\infty} u(||\mathbf{x}||) = \infty$. By Theorem 2.4.2, we have shown that the function that drives CUBIC's *cwnd*, (2.7), is locally uniformly asymptotically stable.

2.4.6 Convergence

Using the Lyapunov-Razumikhin function and its derivative, it is possible to explicitly demonstrate the convergence of trajectories to the fixed point. In the analysis below, we assume that $t_0 = 0$. Recall that

$$V(\mathbf{x}) \leq \epsilon_0 ||\mathbf{x}||_2^2$$
$$\to V^2(\mathbf{x}) \leq \epsilon_0^2 ||\mathbf{x}||_2^4.$$

This gives us

$$\dot{V} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} V^2$$
$$\rightarrow \frac{\dot{V}}{V^2} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}.$$

We note that

$$\frac{d}{dt}\left(-\frac{1}{V}\right) = \frac{\dot{V}}{V^2} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}, \quad \text{so}$$
$$\frac{d}{dt}\left(\frac{1}{V}\right) \geq \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}.$$

Then, solving the differential inequality,

$$\int_0^t \frac{d}{ds}\left(\frac{1}{V}\right) ds = \frac{1}{V(t)} - \frac{1}{V(0)} \geq \int_0^t \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} ds = \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t,$$
$$\frac{1}{V(t)} \geq \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{1}{V(0)},$$
$$V(t) \leq \frac{1}{\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{1}{V(0)}}.$$

Since $V(t) \geq \epsilon_1 ||\mathbf{x}||_2^4$,

$$||\mathbf{x}||_2^4 \leq \frac{1}{\frac{\epsilon_1(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{\epsilon_1}{V(0)}}. \tag{2.29}$$

We can simplify this bound using the definition of uniform stability:

Definition 2.4.1 (Definition 1.1 from Stability of Time-Delay Systems [24])**.** *For the system described by $\dot{x} = f(t, x_t)$, the trivial solution $x(t) = 0$ is said to be stable if for any $t_0 \in \mathbb{R}$ and any $\epsilon > 0$, there exists a $\delta(t_0, \epsilon) > 0$ such that $||x_{t_0}||_c < \delta$ implies $||x(t)|| < \epsilon$ for $t > t_0$. It is said to be uniformly stable if it is stable and $\delta(t_0, \epsilon)$ can be chosen independently of t_0. It is uniformly asymptotically stable if it is uniformly stable and there exists a $\delta_a > 0$ such that for any $\eta > 0$, there exists a $T(\delta_a, \eta)$, such that $||x_{t_0}||_c < \delta_a$ implies $||x(t)|| < \eta$ for $t \geq t_0 + T$ and $t_0 \in \mathbb{R}$.*

Above, $||\phi||_c = \max_{a \leq \xi \leq b} ||\phi(\xi)||$ for $\phi \in \mathcal{C}[a,b]$ and $x_{t_0} = \phi$ or $x(t_0 + \theta) = \phi(\theta)$, $-\tau \leq \theta \leq 0$. From (2.29), we have that

$$||\mathbf{x}||_2^4 \leq \frac{1}{\frac{\epsilon_1(\lambda_{\min}[\tilde{Q}]-K)}{\epsilon_0^2}t + \frac{\epsilon_1}{V(0)}} \leq \frac{1}{\frac{\epsilon_1}{V(0)}} = \frac{V(0)}{\epsilon_1} \leq \frac{\epsilon_0||\mathbf{x}(t_0)||_2^2}{\epsilon_1} < \frac{\epsilon_0\delta^2}{\epsilon_1}.$$

We would like

$$||\mathbf{x}||_2 < \epsilon, \text{ or}$$
$$||\mathbf{x}||_2^4 < \epsilon^4.$$

So let

$$\frac{\epsilon_0\delta^2}{\epsilon_1} < \epsilon^4$$
$$\rightarrow \delta < \epsilon^2\sqrt{\frac{\epsilon_1}{\epsilon_0}}.$$

This bound on δ provides a measure on the basin of attraction of the fixed point of a system using the system's parameters. *I.e.*, it indicates how close the initial conditions must be to the fixed point in order to guarantee stability. One possible way to apply this bound is in the implementation of TCP: one could specify the initial slow start threshold to be close to $\hat{W}$, so that when CUBIC's congestion avoidance phase begins, the systems is more likely to settle into its stable state.

2.4.7 Summary

For the system described by (2.17), the following properties hold:

(a) The system has a unique fixed point $\mathbf{x}^* = \mathbf{0}$.

(b) The system has a unique solution in a neighborhood of this fixed point.

(c) The fixed point is locally uniformly asymptotically stable for $\mathbf{x}$ small enough and in addition,

(i) x_1 and x_{1_τ} are constrained to $[-\rho\hat{W}, \rho\hat{W}]$, for $0 < \rho < 1$,

(ii) $|x_1|, |x_2| < 1$.

(d) The solution is bounded according to (2.29) for $|x_1|$ and $|x_2|$ small enough.

2.5 Analysis of H-TCP

In this section, we use the MWLI model to analyze the stability of H-TCP. As with CUBIC, we use (2.3) as the loss model. Unlike CUBIC, whose *cwnd* is given explicitly as a function of time, H-TCP's *cwnd* behavior is stated in terms of increase and decrease response functions, so we must derive its $W(t)$ in closed form. H-TCP behaves as follows [46]: on each acknowledgement, set

$$cwnd \leftarrow cwnd + \alpha(\Delta)/cwnd \tag{2.30}$$

and on each congestion event, set

$$cwnd \leftarrow \beta(t)cwnd, \tag{2.31}$$

where Δ is the time since the last congestion event and $\alpha(\Delta)$ and $\beta(t)$ are described in detail below. $\alpha(\Delta)$ is constructed so that after a congestion event, H-TCP's *cwnd* increases linearly for a period of Δ^L seconds; this is known as the low-speed regime of the protocol. After Δ^L seconds have passed, as long as there are no new congestion events, H-TCP transitions into its high-speed regime and evolves the *cwnd* according to a function of Δ. When a congestion event occurs, H-TCP uses a backoff factor $\beta(t) \in [0.5, 0.8]$, depending on current estimated maximum and minimum round-trip times. The full operation is as follows:

(a) On each acknowledgement, set

$$\overline{\alpha}(\Delta) = \begin{cases} 1, & \text{if } \Delta \leq \Delta^L, \\ 1 + 10(\Delta - \Delta^L) + \left(\frac{\Delta - \Delta^L}{2}\right)^2, & \text{if } \Delta > \Delta^L, \end{cases}$$

and then let $\alpha(\Delta) = \max\left(2(1-\beta(t))\overline{\alpha}(\Delta), 1\right)$. (Note: the original definitions, *e.g.*, in [47], [46], simply state $\alpha(\Delta) = 2(1-\beta(t))\overline{\alpha}(\Delta)$. However, if $\Delta \leq \Delta^L$, then $2(1-\beta(t))\overline{\alpha}(\Delta) = 2(1-\beta(t))$, which will be less than one for any value of $\beta(t) \in [0.5, 0.8]$ that is not 0.5. Since the increase factor must be at least one, we assume that $\alpha(\Delta) = 1$ in this case, hence the use of max above (also used in [45]).) Then, use Eq. (2.30) to adjust the *cwnd*.

(b) On each congestion event, set

$$\beta(t) = \frac{\tau_{\min}}{\tau_{\max}}, \quad \beta(t) \in [0.5, 0.8].$$

Then, use Eq. (2.31) to adjust the *cwnd*.

Above, $\tau_{\min}/\tau_{\max}$ is the ratio of minimum to maximum round-trip times experienced by the source. Δ^L is usually set to one second [46]. The authors in [46] also suggest scaling $\alpha(\Delta)$ with round-trip time to make the increase rate and convergence time invariant with RTT, as well as to potentially reduce unfairness. We first derive $W(t)$ without RTT scaling and later on introduce a scaling factor γ to discuss its effects on stability ($\gamma = \tau/\tau_{ref}$, where τ_{ref} is a reference RTT value as discussed in [45] and [44]). To simplify our analysis, we make the assumption that $\beta(t) = b$, for $b \in [0.5, 0.8]$. Note that this assumption is reasonable, since in steady state, $\tau_{\min} = \tau_{\max}$ so that the decrease factor is a constant.

Note that in the case where $\Delta > \Delta^L$, $2(1-b)\overline{\alpha}(\Delta)$ can also be less than one, so that $\alpha(\Delta) = 1$. This means that until some time t_L, H-TCP will operate in the

low-speed regime (*i.e.*, it will behave like TCP Reno even though $\Delta > \Delta^L$). We compute t_L to simplify future analysis. To do so, we solve the following equation for t_L:

$$2(1-b)\left(1+10(t_L-\Delta^L)+\left(\frac{t_L-\Delta^L}{2}\right)^2\right)=1.$$

The result is

$$t_L = \Delta^L - 20 + \sqrt{400-4+\frac{2}{(1-b)}}.$$

Note that for $b \in [0.5, 0.8]$, $t_L \geq \Delta^L$. We are now ready to write down the *cwnd* function for H-TCP:

$$W(t) = bW_{\max}(t) + \frac{1}{\tau}\min\left(t_L, s(t)\right) + \frac{\mathbb{1}\{s(t) > t_L\}}{\tau}\int_{t_L}^{s(t)} \alpha(\Delta)d\Delta. \tag{2.32}$$

Note that the indicator function above is required since the integral can evaluate to a positive value in some cases where $s(t) < t_L$.

2.5.1 Fixed Point Analysis

If $t_L \geq \hat{s}$, then the fixed point of the system and its stability analysis reduces to that of TCP Reno. From now on, we assume that $t_L < \hat{s}$; in other words, we analyze H-TCP in environments where the congestion epochs in steady state are sufficiently long for the *cwnd* to transition to the high-speed regime after the linear growth stage. This is common in high-BDP environments for which H-TCP was designed. Note that this causes the indicator function in (2.32) to evaluate to 1. In steady state, we know that $W(t) = W_{\max}(t) = \hat{W}$ and $s(t) = \hat{s}$. Applying this to (2.32), we have

$$\hat{W} = \frac{t_L}{(1-b)\tau} + \frac{2}{\tau}\left(\frac{(\hat{s}-\Delta^L)^3}{12} + 5(\hat{s}-\Delta^L)^2 + \hat{s} - t_L - \frac{(t_L-\Delta^L)^3}{12} - 5(t_L-\Delta^L)^2\right).$$

We now eliminate $\hat{W}$ from the equation above. From the second equation in (2.1), we obtain

$$\begin{aligned} \frac{\hat{s}\hat{W}\hat{p}}{\tau} &= 1, \\ \frac{\hat{s}\hat{W}(1 - C\tau/\hat{W})}{\tau} &= 1, \\ \hat{W} &= \frac{\tau}{\hat{s}} + C\tau. \end{aligned} \tag{2.33}$$

Substituting this into the fixed point equation above, we obtain

$$\frac{\tau}{\hat{s}} + C\tau = \frac{t_L}{(1-b)\tau} + \frac{2}{\tau}\left(\frac{(\hat{s} - \Delta^L)^3}{12} + 5(\hat{s} - \Delta^L)^2 + \hat{s} - t_L - \frac{(t_L - \Delta^L)^3}{12} - 5(t_L - \Delta^L)^2\right). \tag{2.34}$$

Note that (2.34) is quartic in $\hat{s}$. We can use it to analyze the limiting behavior of $\hat{s}$ in terms of C and τ (specifically, we assume large $C\tau$ and $C \gg \tau$). To simplify notation, let $x \equiv \hat{s}$. Recall that in steady state, $\tau_{\min} = \tau_{\max}$, which means that the decrease factor is $b = 0.8$. From now on, to simplify subsequent analysis, we assume $\Delta^L = 1$ s, which is the usual default value in H-TCP implementations. Using this, along with the steady-state value of b, we may now evaluate the steady-state value of t_L, which is given by $\sqrt{406} - 19$. Substituting this into (2.34), expanding and grouping by x yields

$$x^4 + 57x^3 - 105x^2 + \left(812\sqrt{406} - 16283 - 6C\tau^2\right)x - 6\tau^2 = 0.$$

Recall the form of the roots of a quartic equation from Section 2.4.2. For H-TCP, letting $\Gamma = 812\sqrt{406} - 16283$, we have

$$\Delta_0 = 105^2 - 171\left(\Gamma - 6C\tau^2\right) - 72\tau^2 = \mathcal{O}(C\tau^2),$$
$$\Delta_1 = -2(105^3) + 53865\left(\Gamma - 6C\tau^2\right) - 162(57^2)\tau^2 + 27\left(\Gamma - 6C\tau^2\right)^2 - 432(105)\tau^2$$
$$= \mathcal{O}(C^2\tau^4),$$
$$q = \frac{1}{8}\left(57^3 + 228(105)\right) + \left(\Gamma - 6C\tau^2\right) = \mathcal{O}(C\tau^2),$$

and p is a negative constant (no dependence on C or τ). From above and from Eq. (2.14), we obtain

$$\Delta_1^2 = \mathcal{O}(C^4\tau^8), \quad \Delta_0^3 = \mathcal{O}(C^3\tau^6), \quad \text{and} \quad Q = \mathcal{O}((C^2\tau^4)^{1/3}).$$

Further (see Appendix A.3 for derivation details),

$$\frac{\Delta_0}{Q} = \mathcal{O}((C^2\tau^4)^{1/3}), \quad Q + \frac{\Delta_0}{Q} = \mathcal{O}((C^2\tau^4)^{1/3}), \quad \text{and}$$
$$S = \frac{1}{2}\sqrt{\frac{2}{3}p' + \frac{1}{3}\left(Q + \frac{\Delta_0}{Q}\right)} = \mathcal{O}((C\tau^2)^{1/3}),$$

where $p' \equiv -p$ is positive. Finally, letting $q' \equiv -q$ and noting that for large $C\tau$, $q' > 0$, we use the plus-sign version of $x_{3,4}$ to obtain

$$x = -\frac{57}{4} + S + \frac{1}{2}\sqrt{2p' + \frac{q'}{S} - 4S^2} \tag{2.35}$$

and further noting that q'/S and S^2 are both $\mathcal{O}((C^2\tau^4)^{1/3})$, we obtain $\hat{s} = \mathcal{O}((C\tau^2)^{1/3})$. See Appendix A.3 for a proof of the limiting behavior of q'/S, as well as for a proof that the root in (2.35) is positive and real for large $C\tau$ and $C \gg \tau$.

2.5.2 Stability Analysis

We perform a change of variables similar to Section 2.4.3. Letting $\delta \equiv \hat{s} - \Delta^L$, $\xi \equiv t_L - \Delta^L$ and rewriting the *cwnd* function in terms of $x_1(t)$ and $x_2(t)$, yields

$$\Psi(t) = b(x_1(t) + \hat{W}) + \frac{t_L}{\tau} + \frac{2(1-b)}{\tau}\left(\frac{(x_2(t)+\delta)^3}{12} + 5(x_2(t)+\delta)^2 + x_2(t) + \hat{s} - t_L - \frac{\xi^3}{12} - 5\xi^2\right).$$

For the stability analysis, we need the partial derivatives of $\Psi(t)$ with respect to x_1 and x_2 evaluated at the fixed point. They are

$$\begin{aligned}
\frac{\partial \Psi}{\partial x_1} &= b, \\
\frac{\partial \Psi}{\partial x_2} &= \frac{2(1-b)}{\tau}\left(\frac{(x_2(t)+\delta)^2}{4} + 10(x_2(t)+\delta) + 1\right), \\
\left.\frac{\partial \Psi}{\partial x_2}\right|_{\mathbf{0}} &= \frac{2(1-b)}{\tau}\left(\frac{(\hat{s}-\Delta^L)^2}{4} + 10(\hat{s}-\Delta^L) + 1\right).
\end{aligned}$$

Let the quantity above (the partial with respect to x_2 evaluated at the fixed point) be denoted as F_{Δ^L}. Note that F_{Δ^L} is always positive: recall that $t_L \geq \Delta^L$ for all $b \in [0.5, 0.8]$ and by our assumption, $\hat{s} > t_L$, so it follows that $\hat{s} > \Delta^L$. The linearized system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{1}{\hat{s}} \begin{bmatrix} b-1 & F_{\Delta^L} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \frac{\hat{s}}{\tau} \begin{bmatrix} 0 & 0 \\ b & F_{\Delta^L} \end{bmatrix} \begin{bmatrix} x_{1_\tau} \\ x_{2_\tau} \end{bmatrix}.$$

Then, calling the first matrix (multiplied by $1/\hat{s}$) above A_0 and the second matrix (multiplied by $-\hat{s}/\tau$) A_1, note that the linearized system has the form

$$\dot{\mathbf{x}} = A_0\mathbf{x} + A_1\mathbf{x}_\tau.$$

Its Laplace transform is

$$s\mathbf{X}(s) - \mathbf{x}(0) = A_0\mathbf{X}(s) + A_1\mathbf{X}(s)e^{-s\tau}.$$

Solving for $\mathbf{X}(s)$ yields

$$\mathbf{X}(s) = (sI - A_0 - A_1 e^{-s\tau})\mathbf{x}(0).$$

We compute

$$sI - A_0 - A_1 e^{-s\tau} = \begin{bmatrix} s + \frac{1-b}{\hat{s}} & \frac{-F_{\Delta^L}}{\hat{s}} \\ \frac{b\hat{s}}{\tau} e^{-s\tau} & s + \frac{1}{\hat{s}} + \frac{\hat{s}}{\tau} F_{\Delta^L} e^{-s\tau} \end{bmatrix},$$

and the determinant of this matrix is

$$\Delta = \left(s + \frac{1-b}{\hat{s}}\right)\left(s + \frac{1}{\hat{s}}\right)\left(1 + \frac{\hat{s}}{\tau}\frac{F_{\Delta^L}}{\left(s + \frac{1-b}{\hat{s}}\right)} e^{-s\tau}\right).$$

Note that the first two components of Δ, $s + (1-b)/\hat{s}$ and $s + 1/\hat{s}$, are stable. For the third component, we can apply the Nyquist stability criterion. Let

$$H(s) \coloneqq \frac{\hat{s}}{\tau}\frac{F_{\Delta^L}}{\left(s + \frac{1-b}{\hat{s}}\right)} e^{-s\tau} = \frac{\hat{s} F_{\Delta^L}}{\left(s\tau + \frac{(1-b)\tau}{\hat{s}}\right)} e^{-s\tau}$$

and let $\rho \coloneqq (1-b)\tau/\hat{s}$ and $\theta \coloneqq \tan^{-1}(\omega\tau/\rho)$. Then

$$H(j\omega) = \frac{\hat{s} F_{\Delta^L}}{\sqrt{\omega^2\tau^2 + \rho^2}} e^{-j(\omega\tau + \theta)}.$$

Decomposing $H(j\omega)$ into real and imaginary components yields

$$\begin{aligned} \mathcal{I}(H(j\omega)) &= -\frac{\hat{s} F_{\Delta^L} \sin(\omega\tau + \theta)}{\sqrt{\omega^2\tau^2 + \rho^2}}, \\ \mathcal{R}(H(j\omega)) &= \frac{\hat{s} F_{\Delta^L} \cos(\omega\tau + \theta)}{\sqrt{\omega^2\tau^2 + \rho^2}}. \end{aligned}$$

The Nyquist stability criterion states the following: *The rational function* $1 + H(s)$ *has poles in the open left-half s-plane if and only if the Nyquist contour* Γ_H *in the*

$H(s)$-plane does not encircle the $(-1, 0)$ point when the number of poles of $H(s)$ in the right-hand s-plane is zero [12]. Clearly, $H(s)$ does not have any poles in the right-hand s-plane. It remains to check whether the point $(-1, 0)$ is encircled in the $H(s)$-plane or to derive conditions to ensure that it is not encircled. We can find the point where the contour intersects the real axis by setting $\mathcal{I}(H(j\omega)) = 0$, yielding

$$\sin(\omega\tau + \theta) = 0.$$

This equation is satisfied when $\omega\tau + \theta = \pm n\pi$, for $n = 0, 1, 2, \ldots$. Next, we analyze the real component. For $n = 0$ and n even, the numerator of the real part is equal to $\hat{s}F_{\Delta^L}$, which is always greater than zero. Hence, when $n = 0$ or even, $\mathcal{I}(H(j\omega)) > 0$. To analyze the case of n odd, we use the following:

$$\omega\tau = \pm n\pi - \theta,$$
$$|\omega\tau| = |\pm n\pi - \theta|.$$

Further, note that $|\theta| = |\arctan(\omega\tau/\rho)| < \pi/2$. Therefore, $|\omega\tau| > \pi/2$, so for n odd,

$$\mathcal{R}(H(j\omega)) = \frac{-\hat{s}F_{\Delta^L}}{(> \pi/2)} > \frac{-2\hat{s}F_{\Delta^L}}{\pi}.$$

For stability, we require that $\mathcal{R}(H(j\omega)) \geq -1$. This yields the following (sufficient) stability condition:

$$\hat{s}F_{\Delta^L} \leq \frac{\pi}{2},$$
$$\hat{s}\left(\frac{(\hat{s} - \Delta^L)^2}{4} + 10(\hat{s} - \Delta^L) + 1\right) \leq \frac{\pi\tau}{4(1-b)}.$$

Recall that $\hat{s}$ is a function of C, τ, and t_L, and the latter is a function of Δ^L. We have not been able to find values of C, τ, and Δ^L that satisfy the stability condition

above. In addition, we have not been able to find a set of parameters for which the simulation of DEs show a stable system. Recall that in the *cwnd* function definition (2.32), no RTT scaling is employed. RTT scaling can reduce unfairness, as well as enable congestion epoch duration and convergence time to be independent of RTT. When implemented in H-TCP, RTT scaling is only enabled in high-speed regimes (*i.e.*, when $\Delta > t_L$), and works by scaling $\alpha(\Delta)$ in this regime by τ/τ_{ref}, where the recommended value of $\tau_{ref} = 100$ ms [45]. With this change, the stability condition becomes

$$\frac{\hat{s}}{\tau_{ref}}\left(\frac{(\hat{s}-\Delta^L)^2}{4}+10(\hat{s}-\Delta^L)+1\right) \leq \frac{\pi}{4(1-b)},$$

where t_L is redefined as

$$t_L = \begin{cases} \Delta^L - 20 + \sqrt{400 - 4 + \frac{2\tau_{ref}}{(1-b)\tau}} & \text{if } \frac{2(1-b)\tau}{\tau_{ref}} \leq 1, \\ \Delta^L \text{ otherwise.} \end{cases}$$

Numerical analysis shows that in practical settings (*i.e.*, for realistic values of C, τ, and b), H-TCP with RTT scaling (with scaling parameter $\gamma \coloneqq \tau/\tau_{ref}$) is still unstable when it operates in the high-speed regime. In fact, we find that the only way to stabilize the systems is to use values of b and τ_{ref} that force the protocol to operate exclusively in the low-speed regime (*i.e.*, to behave as standard TCP). However, we note that certain values of γ tend to dampen the oscillation amplitudes of H-TCP's *cwnd*, albeit without affecting the convergence behavior of the protocol. One example of this phenomenon is illustrated in Figure 2.3. The *cwnd* curves are a result of the DE model with $C = 1$ Gbps, $\tau = 40$ ms, and $b = 0.8$. $\gamma = 1$ corresponds to no RTT scaling, while $\gamma = 0.4$ corresponds to using the recommended $\tau_{ref} = 100$ ms. Recall the fixed point expression for *cwnd* from Eq. (2.33): $\hat{W} = \tau/\hat{s} + C\tau$. Note from this equation that in high-BDP settings, $\hat{W}$ is dominated by the BDP,

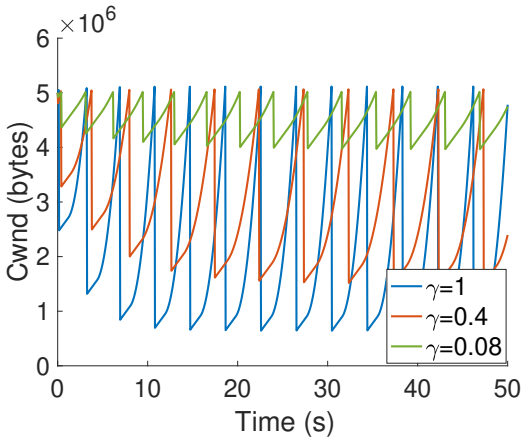


Figure 2.3: Effect of RTT scaling on H-TCP's *cwnd*. Curves produced by the DE model with $C = 1$ Gbps, $\tau = 40$ ms, $b = 0.8$, and scaling factor $\gamma = \tau/\tau_{ref}$. In all three cases, the protocol operates in the high-speed regime and is unstable. Decreasing γ reduces the magnitude of the oscillations.

$C\tau$, while varying γ only changes $\hat{s}$. Hence, $\hat{W}$ does not change much with γ. This is reflected in Figure 2.3, as all three flows reach approximately the same maximum value of *cwnd*, $\approx C\tau = 5 \times 10^6$ bytes. On the other hand, the flow that corresponds to $\gamma = 0.08$ deviates significantly less from the fixed point than do the other flows, with larger γ's. Interestingly, decreasing γ further precludes H-TCP from operating in the high-speed regime, and the protocol's behavior approaches that of standard TCP while not necessarily stabilizing it (*i.e.*, the *cwnd* retains its sawtooth profile and there is no convergence to the fixed point).

2.6 A Note on the Loss Model

We chose the probability of loss model given by Eq. (2.3) for its simplicity and dependence on few system parameters. However, it is worthwhile to note that this loss model is used often in fluid approximation models, especially under the assumption that the packet arrival process is a Poisson process [69]. Such an assumption is reasonable in certain scenarios, *e.g.*, with sufficiently long-lived data transfers and large buffer sizes at congestion points. However, in scenarios where traffic may be

bursty, the arrival process no longer behaves as a Poisson process, and using the loss model given by (2.3), or even its finite-version for $M/M/1/B$ queues with queue size B, may cause inaccuracies for throughput prediction [20].

Note that the MWLI model can accommodate different loss models, and as long as a model $p(t)$ can be written in closed form in steady state (*e.g.*, (2.3) simply becomes $p(t) = 1 - C\tau/\hat{W}$ in steady state), one can perform fixed point and stability analyses. A consequence of using a particular $p(t)$ is that the analysis is valid under the same assumptions for which $p(t)$ is a reasonable model. Hence, while H-TCP seems unstable under (2.3), it may behave very differently under a less harsh $p(t)$. In Section 2.7, we explore another loss model for H-TCP that incorporates queue size.

2.7 Simulations

We use simulation to validate model (2.17) and the stability analyses of TCP CUBIC and H-TCP. For H-TCP, we use *ns-3* and for CUBIC, we develop a simulation framework that treats loss as a non-homogenous Poisson process and generates new loss events based on a user-defined probability of loss model. Because of the loss generation method, we call this simulation framework the Non-Homogeneous Poisson Loss, or NHPL simulation. A detailed description of the framework is provided in Appendix A.1. An advantage of using this framework for validating the DEs over, for example, *ns-3*, is that we can observe the behavior of solely the congestion avoidance phase of an algorithm, which allows us to more easily verify the theoretical analysis of the controller's stability. Moreover, as we observe from simulations of the DEs, an algorithm's stability can be highly sensitive to the initial conditions specified at the beginning of the congestion avoidance phase (note that this observed sensitivity to initial conditions is consistent with our demonstration of local-only stability). The initial conditions are values of $W_{\max}(0)$ and $s(0)$ for all flows, and we can control them more easily with our simulation framework. This can be especially useful when testing

the region of stability for a given system. Another reason for using NHPL simulations for CUBIC is that at the time of writing, CUBIC is not natively supported in *ns-3*. As a result, certain experiments of CUBIC in *ns-3* take an exceedingly long time to complete (especially systems with high BDPs, which are of interest in this work).

Note that the NHPL simulation framework is event-based, rather than packet-based. To diversify our experiments, we use *ns-3* – a packet-based simulation framework – to validate the H-TCP model. For the experimental setup, we use a `PointToPoint` channel between two net devices. We choose the PI (Proportional Integral controller) AQM scheme [33] for H-TCP in *ns-3*. We choose PI because it is simple to implement within the MWLI model. Since *ns-3* does not natively support PI, we modify the similar AQM protocol PIE [57] implementation, which is available in *ns-3*.

In order to compare the DE model against *ns-3*, we alter the DEs' previous loss model (that of Eq. (2.3)) to incorporate some of the major components of PI, with the objective of implementing a simplified version of the AQM scheme that reasonably approximates PI. Specifically, every T_UPDATE seconds (we use 5 ms in all our experiments – similar to the value used in the experiments of [33]), we compute

$$\begin{aligned} &\Delta p = \hat{\alpha}(q - q_{ref}) + \hat{\beta}(q_{ref} - q_{old}) \quad \text{and set} \\ &p \leftarrow p + \Delta p \end{aligned} \tag{2.36}$$

as PI's new drop probability. Above, $\hat{\alpha}$ and $\hat{\beta}$ are parameters, q is the current length of the queue, q_{old} is the length of the queue in the previous iteration, and q_{ref} is the desired queue length. Note that this value of q_{ref} is a reference for an average flow within our DE model, as opposed to a reference for all flows in aggregate, and similarly for the values of q and q_{old}, so that the loss probability is computed for a single flow. In all our experiments, we set $\hat{\alpha} = 1.822 \times 10^{-5}$ and $\hat{\beta} = 1.816 \times 10^{-5}$, as in [33]. The value of p is then used to update s and $W_{\max}$ in the DE model.

For the DE model, we use the following queue model (based on a model from [34]):

$$\frac{dq(t)}{dt} = \frac{W(t)}{\tau} - C, \tag{2.37}$$

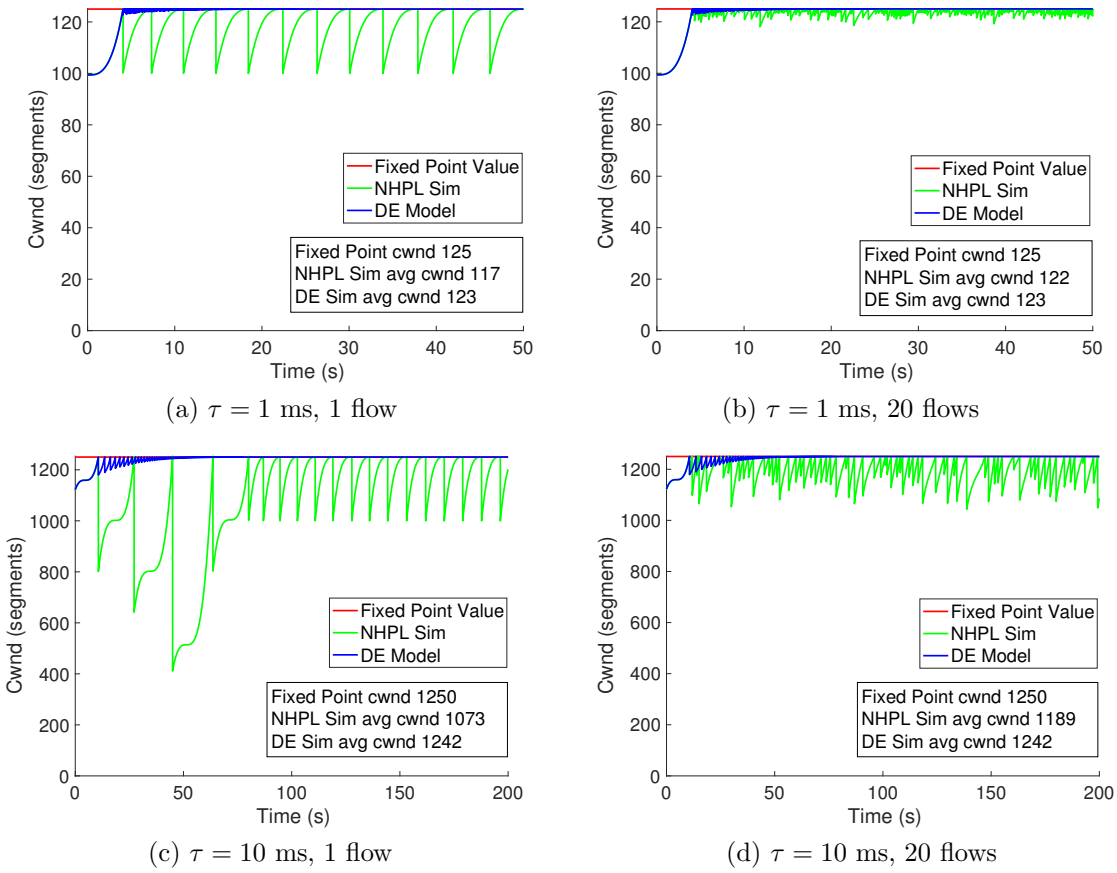
where $q(t)$ is the number of packets in the bottleneck queue. Then the queueing delay is given by $q_{del}(t) = q(t)/C$. Hence, the total delay is the sum of queueing and propagation delays, given by $q_{del}(t) + \tau$; this is the value we use instead of τ in (2.1). Note that for the experiment descriptions in this section, we only state the (constant) propagation delay τ.

It remains to choose a queue size for *ns-3* experiments: *i.e.*, the `MaxSize` variable for PI's queue, which we will call $q_{\max}$ from now on. Let N be the number of *ns-3* flows. For each *ns-3* experiment, we choose a value of N large enough so that bandwidth on the link is fully utilized. To choose a suitable value for $q_{\max}$, we multiply q_{ref} by N and choose a larger value, keeping in mind that choosing too large a $q_{\max}$ may cause instability.

For convenience, denote the aggregate (or link) capacity $C_A \coloneqq NC$. Note that the DE model takes C as a parameter, while *ns-3* sets the link bandwidth using C_A and divides it by N to set the sending rate of each flow. Note that in all calculations above, C may have to be scaled by a maximum segment size (MSS) and other constants: in all our simulations, we use the unit of segments/sec for C.

2.7.1 TCP CUBIC

Figure 2.4 compares the average *cwnd* generated by the NHPL simulations against the average value of *cwnd* generated by the DEs. The fixed-point value of *cwnd*, $\hat{W}$, is also shown (albeit sometimes entirely hidden by the DE curve because of fast convergence). All flows in this figure have a per-flow capacity of 1 Gbps, while the round-trip time is varied (these combinations of C and τ are sufficient to generate a



(a) $\tau = 1$ ms, 1 flow

(b) $\tau = 1$ ms, 20 flows

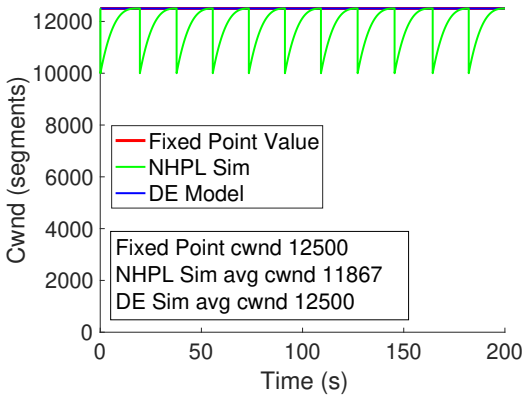
(c) $\tau = 10$ ms, 1 flow

(d) $\tau = 10$ ms, 20 flows

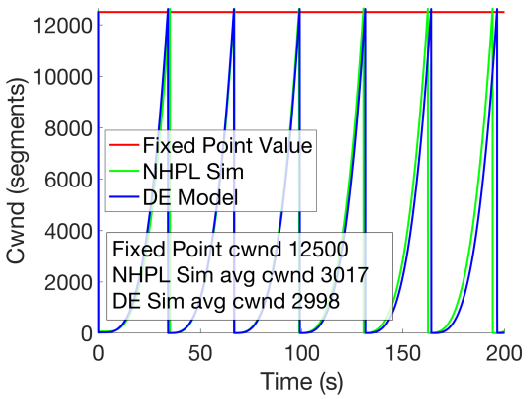
Figure 2.4: Comparison of average *cwnd* (computed post-transient phase) generated by NHPL simulations against steady-state *cwnd* generated by model (2.17) for TCP CUBIC. Also shown is the fixed-point value of *cwnd*. Per-flow capacity $C = 1$ Gbps.

diverse set of behaviors). All flows have $b = 0.2$ and $c = 0.4$ (the default values used in Linux implementations of CUBIC).

Figure 2.4a shows a single stable flow with $\tau = 1$ ms. The transient response of both simulations is clearly visible, and we observe that they reach steady-state within a similar period. Not shown in this panel is the value of $\hat{s} \approx 4$ seconds. By observing the time between losses in the NHPL simulation, we see that there is a close agreement. Figure 2.4b shows the same experiment, but with 20 flows. As expected, the average value of *cwnd* from the NHPL simulation approaches $\hat{W}$ as the number of flows increases. Figures 2.4c and 2.4d show one and 20 flows, respectively, for $\tau = 10$ ms. The initial conditions (values of $s(0)$ and $W_{\max}(0)$) are deliberately far enough



(a) $\tau = 100$ ms, stable



(b) $\tau = 100$ ms, unstable

Figure 2.5: The impact of initial conditions on stability. For both (a) and (b), $C = 1$ Gbps, $\tau = 100$ ms. In (a), there is one flow whose initial conditions $W(0)$ and $s(0)$ are very close to the fixed point values $\hat{W}$ and $\hat{s}$, respectively. Both the NHPL simulation and the model exhibit stability. In (b), there is one flow whose initial conditions are set too far from the fixed point value, destabilizing the flow in both the NHPL simulation and the DE system.

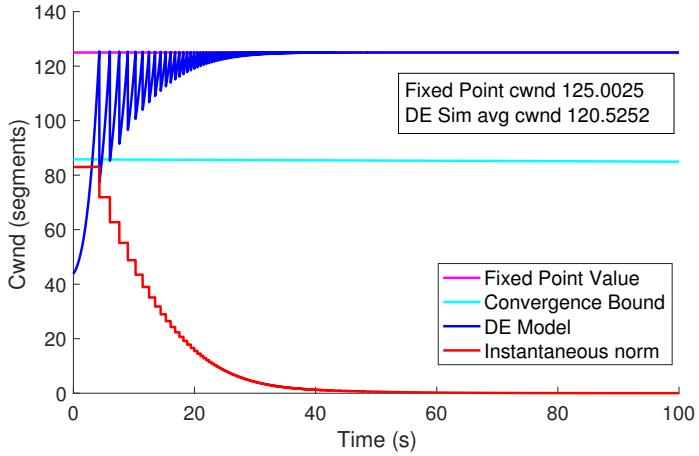


Figure 2.6: Convergence for CUBIC. At the top is the *cwnd* generated by DEs as it converges to the fixed point value of *cwnd*. Below these two curves is a comparison of the instantaneous norm $||\mathbf{x}||_2$ against the analytical bound in (2.29). Here, $C = 100$ Mbps, $\tau = 10$ ms.

from the fixed point to demonstrate a more dramatic transient response. Figure 2.5 shows two examples of 100 ms flows: in (a), there is a single flow that is stable, while the initial conditions in (b) cause instability for one flow in both the DEs and NHPL simulation.

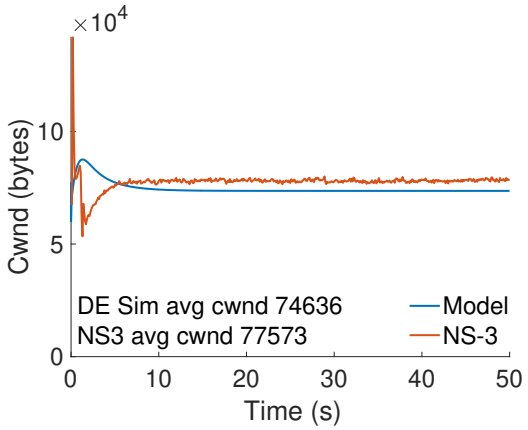
Figure 2.6 illustrates the transient and steady-state responses of a flow with $C =$ 100 Mbps and $\tau = 10$ ms, as well as $||\mathbf{x}||_2$ as it compares to the convergence bound (2.29). Observe that $||\mathbf{x}||_2$ always lies below the bound and approaches zero as the flow reaches steady state. The bound appears flat in this example because for this system, $V(t_0)$ dominates in the denominator. We observe this phenomenon for many systems; this implies that the initial conditions are crucial for a flow's stability.

2.7.2 H-TCP

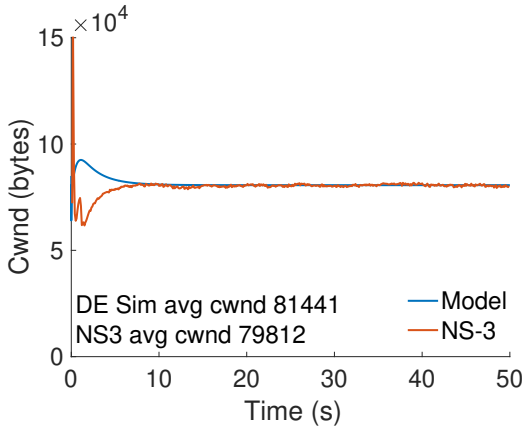
We present experiments with varying BDPs, ranging from low – wherein H-TCP operates exclusively in the linear regime, to high – where H-TCP's average congestion epoch is larger than Δ^L seconds. Since *ns-3* does not use RTT scaling, we set $\gamma = 1$ in all DE simulations. All *ns-3* curves are averages of five runs. Figure 2.7 presents comparisons of the MWLI DE model for H-TCP, using Eqs. (2.36) and (2.37) as loss probability function and queue model, respectively, against *ns-3* with PI as the AQM scheme. The experiment in Figure 2.7a uses a link capacity C_A of 500 Mbps with 15 flows in *ns-3*. This corresponds to $C = 33$ Mbps for the DE model. Packet size is set to 1000 bytes and MSS is 958 bytes. For this experiment, $\tau = 10$ ms, which causes the BDP to be low enough that on average, H-TCP operates in the low-speed regime. This is evidenced by the average time between losses, $\overline{s}$, defined as follows:

$$\overline{s} \coloneqq \frac{1}{R}\sum_{r=1}^{R}\frac{1}{N}\sum_{f=1}^{N}\frac{1}{L_{f,r}}\sum_{i=1}^{L_f} s_i^{f,r},$$

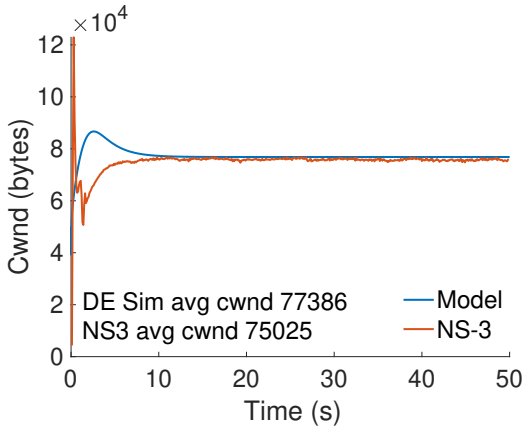
where R is the number of *ns-3* experiment runs, $L_{f,r}$ is the number of losses suffered by flow f during the duration of the experiment in the rth run, and $s_i^{f,r}$ is the time between the $(i-1)$th and ith loss of flow f in the rth run (for $i = 1$, we compute the time until the first loss). For Figure 2.7a, $\overline{s} = 0.29$ s, which is below the low-speed regime threshold $\Delta^L = 1$ s. The DE model outputs a value of 0.27 s for $\hat{s}$.



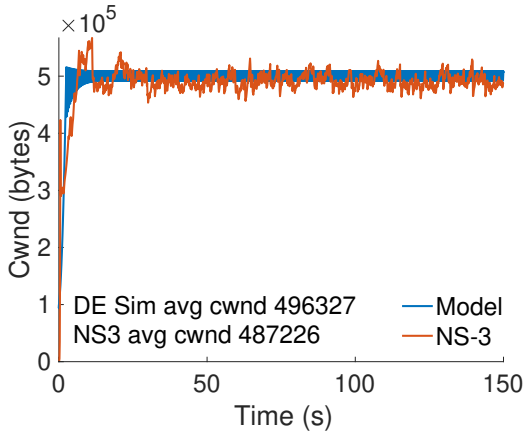
(a) $C_A = 500$ Mbps, $\tau = 10$ ms, $q_{ref} = 500$, $q_{\max} = 1000$, $N = 15$, MSS $= 958$



(b) $C_A = 1$ Gbps, $\tau = 10$ ms, $q_{ref} = 800$, $q_{\max} = 3500$, $N = 25$, MSS $= 958$



(c) $C_A = 1$ Gbps, $\tau = 20$ ms, $q_{ref} = 1000$, $q_{\max} = 4000$, $N = 45$, MSS=958



(d) $C_A = 2$ Gbps, $\tau = 40$ ms, $q_{ref} = 800$, $q_{\max} = 2500$, $N = 20$, MSS $= 958$

Figure 2.7: Validation of H-TCP's DE model using *ns-3*. Red curves represent average *cwnd* over all *ns-3* flows. *ns-3* uses PI as the AQM scheme and DE models use Eqs. (2.36) and (2.37). Aggregate capacity is given by $C_A := NC$. q_{ref} and $q_{\max}$ have units of packets, while MSS is in bytes. Each red curve is an average of five runs of the experiment.

For the experiment in Figure 2.7b, we increase the link capacity to one Gbps and the number of flows to 25. Here, *ns-3* yields $\overline{s} = 0.25$ s and the DE model yields $\hat{s} = 0.27$ s. For the experiment in Figure 2.7c, we use $C_A = 1$ Gbps and a propagation delay of 20 ms. The number of flows is set to 45 in *ns-3*. Here, $\overline{s} = 0.38$ s and $\hat{s} = 0.44$ s. In Figure 2.7d, $C_A = 2$ Gbps and $\tau = 40$ ms, which allows H-TCP to be able to spend time in the high-speed regime, with $\overline{s} = 2.19$ s and $\hat{s} = 2.2$ s. Finally,

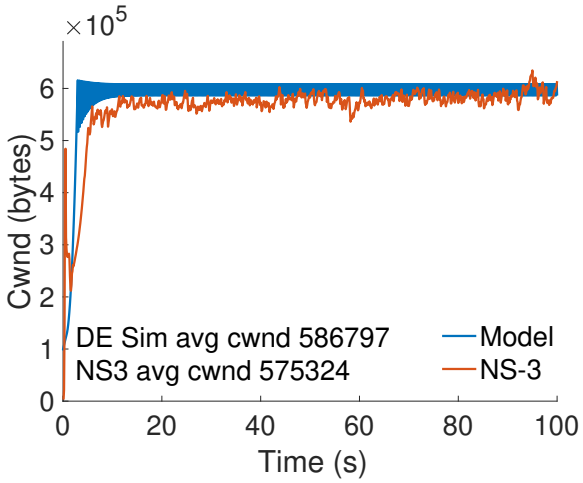


Figure 2.8: Validation of H-TCP's DE model using *ns-3*. Aggregate capacity $C_A = 4$ Gbps, $\tau = 30$ ms, $q_{ref} = 100$ packets, $q_{\max} = 1000$ packets, $N = 25$ flows, and MSS = 1158 bytes. The red curve is an average of five runs of *ns-3*.

Figure 2.8 presents an experiment with a relatively larger BDP compared to those of Figure 2.7: here, $C_A = 4$ Gbps and $\tau = 30$ ms. There are 25 *ns-3* flows and as with experiments in Figure 2.7, the average is plotted over five runs. In this plot, $\overline{s} = 1.7$ s and $\hat{s} = 1.8$ s. Note that for this larger BDP, the *ns-3* curve is slower to converge to the DE curve, compared to some of the experiments in Figure 2.7. Further, note that as the BDP increases (*e.g.*, Figure 2.7d and Figure 2.8), the DE model predicts a lack of convergence (which may be interpreted as a type of instability in the context of TCP) for H-TCP's *cwnd*. Similarly, the average *ns-3* flow experiences greater variation compared to *cwnd*s in lower-BDP settings. We observe this in general for larger BDPs, but predictably, the *cwnd* can be stabilized by introducing more flows, albeit at the cost of decreasing each flow's throughput and often forcing the protocol to operate in the low-speed regime. These observations are consistent with those of the numerical analysis discussion in Section 2.5.2. Note that in all our experiments, the average *cwnd*s match closely between *ns-3* and the DE model, and similarly, there is close agreement between $\overline{s}$ and $\hat{s}$. Hence, Figures 2.7 and 2.8 demonstrate the MWLI model's ability to effectively predict H-TCP's behavior.

2.8 Conclusion

The main contribution of this work is a novel and versatile fluid model, which we call the MWLI model, for *cwnd*- and rate-based data transport algorithms. The model is structured so that the differential equations do not depend on the specific window or rate function of a congestion controller. As a result, this framework offers opportunities to model and analyze the stability of a diverse set of controllers whose window or rate functions may not be linear and whose increase and decrease rules may not be given in explicit form. We applied this model to three different algorithms: TCP Reno, CUBIC and H-TCP. For the former, we prove that the new model is equivalent to a well-established model for Reno. For CUBIC, the new model succeeds where traditional methods of modeling *cwnd* are ineffective. We analyzed the fluid model for CUBIC and discovered that for a given probability of loss model, its window is locally uniformly asymptotically stable. We also derived a convergence bound on the solution of the system as a function of the system parameters. Further, we developed an event-based simulation framework to validate the model and related theoretical results for CUBIC.

For H-TCP, we performed a linear stability analysis of the congestion controller under certain assumptions. We find that H-TCP is unstable in most cases under a particular loss model, but that it can be stabilized under a more realistic loss model, *e.g.*, such as one whose operation is closer to that of the PI AQM scheme. We validated H-TCP's model using *ns-3* and find that the MWLI model is able to predict average *cwnd* and average time between losses well.

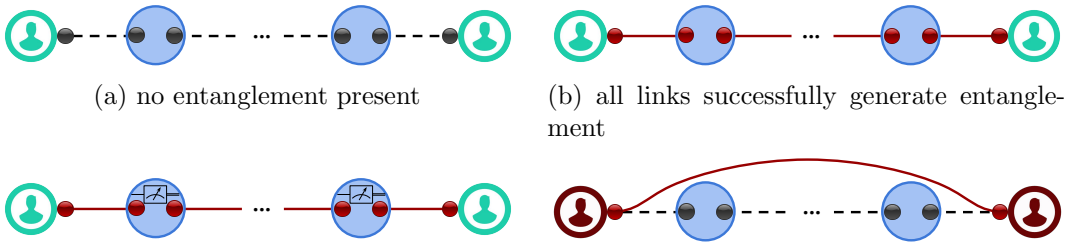
CHAPTER 3

ON THE STOCHASTIC ANALYSIS OF A QUANTUM ENTANGLEMENT DISTRIBUTION SWITCH

3.1 Introduction

Protocols that exploit quantum communication technology offer two main types of advantages: they can either extend or render feasible the capabilities of their classical counterparts, or they exhibit functionality that is entirely unachievable through classical means alone. For an example of the former, quantum key distribution (QKD) protocols such as E91 [13] and BBM92 [4] can in principle yield information-theoretic security by using entanglement to generate secure key bits. These raw secret key bits can then be distilled into a one-time pad to encode messages sent between two parties. For an example of the latter, distributed quantum sensing frameworks such as [15] and [77] employ entanglement to overcome the standard quantum limit [21].

While these applications hold a tremendous amount of potential for distributed quantum communication (and even computation, see, *e.g.*, [39]), a substantial challenge is reliable generation of entanglement – an essential component for many of these tasks – especially over a large distance. This is due to the fact that there is an exponential rate-versus-distance decay for quantum state propagation both through terrestrial free-space and optical fiber channels [60, 70]. Quantum repeaters positioned between communicating nodes can overcome this fundamental rate-versus-distance tradeoff [26, 53]. The process of quantum repeater-assisted entanglement generation is illustrated at a high level in Figure 3.1 and can be divided into two main steps. In step one, each segment connecting two adjacent nodes attempts to generate an entangled link. Qubits from a successfully-generated entanglement are stored in quantum



(a) no entanglement present

(b) all links successfully generate entanglement

(c) repeater nodes perform entangling measurements

(d) end nodes share an entangled state

Figure 3.1: Long-distance entanglement generation using quantum repeaters. The two nodes at the edges are communicating parties, and the nodes between them are quantum repeaters. Dashed lines represent lack of entangled links, while solid lines represent presence of entanglement. The gray and red circles are unoccupied and occupied quantum memories, respectively.

memories, one in each node (Figure 3.1b). Once entangled links are present on all segments, the quantum repeaters perform entanglement swapping [78] on their two locally-held qubits (Figure 3.1c). The result, if all swapping operations succeed, is an end-to-end entangled link between the communicating parties (Figure 3.1d).

In this work, we use the term "quantum switch" instead of "repeater" because in a more complex network than that of Figure 3.1, the device will likely be connected to several nodes or users, hence it is reasonable to assume that it will be equipped with entanglement switching logic. Quantum repeaters, switches, and similar devices (*e.g.*, trusted nodes) will serve as building blocks of large-scale quantum networks. It is natural, therefore, to ask questions about their fundamental limits from a mathematical perspective, in order to gain insight into what constitutes efficient operation for such a device, as well as to create a performance comparison basis for future protocols and algorithms that rely on these devices. To this end, we study a quantum switch that serves entangled states to pairs of users in a star topology, with the objectives of determining the capacity of the switch, as well as the quantum memory occupancy distribution and expected number of stored qubits in memory at the switch when it operates at capacity.

We accomplish these objectives by constructing a simple, yet descriptive model of a quantum switch: we determine a small number of important model parameters and abstract away the specifics of implementation. For instance, we do not focus on a specific method of entanglement generation on a link, and we do not analyze a specific quantum memory implementation; rather, we include the rate of entanglement generation and memory coherence times as configurable system parameters. This way, our model is agnostic to hardware architecture and protocol specifics, and is kept general. Subsequently, when we analyze the model, we obtain results that are often interpretable and intuitive.

In the most general case, the switch serves n-partite entangled states to sets of users according to incoming requests, where $n \leq k$. To achieve this, link-level entangled states are generated at a constant rate across each link, resulting in two-qubit maximally-entangled states (*i.e.*, Bell pairs or EPR states). These qubits are stored at local quantum memories: one from each Bell pair at the user and the other one at the switch. When enough of these entanglements are accrued (at least n of them), the switch performs multi-qubit measurements to provide end-to-end entanglement to user groups of size n. When $n = 2$, the switch uses Bell-state measurements (BSMs) and when $n \geq 3$, it uses n-qubit Greenberger-Horne-Zeilinger (GHZ) basis measurements [55]. In this work, we focus on the case of $n = 2$ – *i.e.*, the case of bipartite-only switching, although some prior work on $n \geq 3$, as well as n being allowed to switch between 2 *or* 3 will be discussed.

We consider systems in which links may generate entanglement at different rates and where the switch can store one or more qubits (each entangled with another qubit held by a user) per link. Throughout this chapter, we refer to these pairs of stored qubits as *stored entanglements*. Another factor that impacts performance is decoherence of quantum states; we model it and study its effect. The main metric of interest for this network is its capacity C, *i.e.*, the maximum possible number of end-

to-end entanglements served by the switch per time unit. Another metric of interest is the expected number of qubits Q in memory at the switch, $E[Q]$. Where possible, we also derive in closed-form the distribution of the number of stored qubits at the switch. Both C and $E[Q]$ depend on the values of k, n, entanglement generation and decoherence rates, number of quantum memories (often referred to as *buffer size* throughout this chapter), and the switching mechanism, including the scheduling policy used by the switch.

Contributions of this work are as follows: by modeling the switch as a continuous-time Markov chain (CTMC), we derive C and $E[Q]$ for $n = 2$ for a particular scheduling policy and study how they vary as functions of k, buffer size, and decoherence rate. From our analysis, we gain valuable insight into which factors influence capacity the most, and which ones are of lesser consequence. For instance, we find that when $n = 2$ and links are identical, the number of links and their entanglement generation rate are the most impactful, while decoherence and buffer size have little effect. However, the same is not true in the case of non-identical links, where the distribution of entanglement generation rates, combined with finite coherence time, can drastically affect both C and $E[Q]$.

The remainder of this chapter is organized as follows: in Section 3.2, we briefly cover relevant background for quantum communication. In Section 3.3, we discuss related work. In Section 3.4, we cover modeling techniques, assumptions, and objectives. In Section 3.5, we introduce our CTMC models for $n = 2$ and present their analyses. Numerical observations are discussed in Section 3.6. We conclude in Section 3.7.

3.2 Background

A qubit is the quantum analogue of a bit and can be described by a two-level quantum-mechanical system, *e.g.*, the polarization of a photon. Two qubits are said

to be entangled if the state of one cannot be described independently from the state of the other. One of the most essential resources for quantum communication is a maximally entangled two-qubit state known as a Bell state or Bell pair. The four Bell pairs are given by

$$|\Phi^{\pm}\rangle = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}} \quad \text{and} \quad |\Psi^{\pm}\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}$$

and can be used as an expendable resource in a number of distributed quantum tasks, such as teleportation [5], or to generate a raw secret key bit in entanglement-based QKD protocols. Another way to use these states is for long-distance entanglement generation as depicted in Figure 3.1, where each solid line is a Bell state. In the special case of long-distance Bell pair generation via connection of two shorter-distance Bell pairs, the switch performs the swapping operation via a Bell state measurement (BSM). In linear optics, this is a probabilistic but heralded operation, with the success probability dependent on the exact implementation of the BSM as well as gate operation efficiencies [64, 17, 23]. We address this phenomenon in our model by introducing a parameter that represents the BSM success probability.

In [32], Herbauts *et al.* implement an entanglement distribution network intended for quantum communication applications. The fidelities of entanglement generated in this network were 93% post-distribution, and fidelities of 99% were shown to be achievable. The demonstration entails distributing bipartite entanglements to any pair of users wishing to share entanglement in a multi-user network (there were eight users in the experimental setup). Delivering multiple bipartite entanglements was shown to be possible virtually simultaneously. The authors specifically cite a possible application of the network in a scenario where a single central switch dynamically allocates two-party entanglements to any pair of users in a static network. In this work, we study variants of this system, where we assume that the switch has the ability to store entangled qubits for future use, and that successfully-generated entanglement

– both at the link and end-to-end levels – has fidelity one. This latter assumption allows us to obtain an upper bound on the entanglement switching rate and is a good starting point for quantum switch analysis; the incorporation of non-unit state fidelity (and the possible need for some form of entanglement purification, *e.g.*, [6]) is left as an open question and a subject of future work.

In recent years, there have been other promising experimental demonstrations for realizing the fundamental components of quantum repeater architectures. For instance, in [8], Bhaskar *et al.* implement quantum-memory-enhanced quantum communication to overcome the fundamental limit of repeaterless communication [60]. At the same time, new architectures and protocols, which promise to yield higher-fidelity states and quicker end-to-end entanglement generation rates, have been proposed – *e.g.*, the quantum router proposal in [43] achieves both of these objectives. Such advances further emphasize the importance of analysis and theoretical studies to help guide hardware specifications and protocol design for quantum communication architectures.

3.3 Related Work

In [66], the authors use Markovian models to compute the expected waiting time in quantum repeaters with probabilistic entanglement swapping. Specifically, they consider entanglement distribution over a distance subdivided by repeater segments, and while they propose a method of computing the average waiting time for an arbitrary number of links, explicit expressions are provided for only up to four segments. In contrast, we consider a single quantum repeater-like device, but one that services an arbitrary number of links.

In Chapter 5, we analyze the capacity region of a quantum entanglement switch that serves users in a star topology and is constrained to store one or two qubits per link. The problem setup is quite similar to that of this work, with the exception that

the switch has the ability to serve bipartite and tripartite end-to-end entanglements, and all links are assumed to be identical. There, we examine a set of randomized switching policies and find policies that perform better than time-division multiplexing between bipartite and tripartite entanglement switching. Note that while in Chapter 5, we allow the switch to choose between two types of entanglement to serve at every time step, in this work we fix n and analyze it in more detail: for instance, in Chapter 5 all links are assumed to be identical, while in this work links may be heterogeneous and buffer sizes can be larger than one or two per link.

In [54], we study a quantum switch serving n-partite end-to-end entangled states to $k \geq n$ users and for $n \geq 2$. The setup is identical to that of this work, but limited to only the case of a homogeneous-link, infinite-buffer system with no quantum state decoherence. For the case of $n = 2$, the results are consistent with those of this work, and we build on them to explore more complex bipartite switching systems. Last, in Chapter 4, we model the $n = 2$ case for an identical-link, infinite buffer switch using a discrete-time Markov chain (DTMC). In Section 4.5, we compare the results of this logically more accurate model with those of this work and find that the differences in predictions of the performance metrics are small. On the other hand, as it will become evident in the following sections of this chapter, CTMCs are far easier to use in this context and therefore may be considered a more powerful tool for modeling quantum switches. As new quantum architectures and technologies emerge, we expect quantum networks to become more prevalent and suitable for practical use. With link-level and especially end-to-end entanglements being a valuable commodity in these networks, proper resource management will be imperative for reliable and efficient operation.

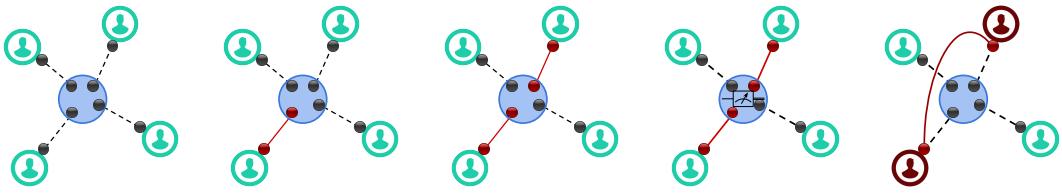


Figure 3.2: Example of quantum switch operation. No Bell pairs are present in **(a)**. When enough Bell pairs are successfully generated (solid lines in **(b)** and **(c)**), the switch performs a BSM **(d)**, entangling the two users' qubits **(e)**.

3.4 Model and Objectives

Consider first a fairly general setting of the proposed problem: k users are attached to a quantum entanglement distribution switch via k dedicated links. At any given time step, any set of n users (with $n \leq k$) may wish to share an end-to-end entangled state. The creation of an end-to-end entanglement involves two steps. First, users generate pairwise entanglements with the switch, which we call link-level entanglements. Each of these results in a two-qubit entangled (Bell) state, with one qubit stored at the switch and the other stored at a user. Once there are n link-level Bell pairs available to fulfill a request between n users, the process enters step two: the creation of an end-to-end entanglement. The switch chooses the set of n locally-held qubits (that are entangled with n qubits held by the n distinct users) corresponding to the request and performs an entangling measurement. If such a measurement is successful, the result is an n-qubit maximally-entangled state between the corresponding n users. If after this step more link-level entanglements are available and can be used to fulfill another request, the switch repeats the second step until either there are fewer than n local qubits left or until no more requests can be fulfilled. Figure 3.2 illustrates an example with $k = 4$ and $n = 2$.

In this chapter, our objective is to derive a tight upper bound on the entanglement switching rate when $n = 2$, *i.e.*, the maximum possible rate at which the switch may serve bipartite end-to-end entangled states – we call this quantity the bipartite

switching capacity of the system. Since this upper bound should hold for any workload, it is necessary for us to assume that *any* two users wish to share an entangled state; in fact, removing this assumption would necessarily decrease the rate at which the switch is allowed to serve end-to-end entanglement. With this request policy, the switch has no restrictions on which measurements to perform whenever two distinct link-level entanglements are available. Hence, in step two of entanglement distribution, the switch simply chooses a set of two qubits corresponding to Bell pairs on two distinct links, and uses them in the entangling measurement. Step two is repeated until there are no more two distinct links with available Bell pairs. The results of our analysis on the capacity of the switch can be used as a comparison basis for other types of scenarios, in which, for example, each pair of users may specify a desired rate of communication with each other through the switch. Another utility of this analysis is that by examining a switch that operates at or near maximum capacity, one may gain insight on the practical memory requirements of a switch.

Both link-level entanglement generation and entangling measurements can be modeled as probabilistic phenomena [26]. In this work, we model the former as a Poisson process: each link attempts entanglements at rate λ, and for link $l \in \{1, \ldots, k\}$, each attempt succeeds with probability $p_l \approx e^{-\beta L_l}$, where L_l is the length of the lth link (*e.g.*, optical fiber) and β its attenuation coefficient. Hence, link l generates successful entanglements with rate $\mu_l \coloneqq \lambda p_l$. We refer to the special case of $\mu_l = \mu_m,\ \forall l, m \in \{1, \ldots, k\}$ as a *homogeneous* system, and when they are not necessarily equal, as a *heterogeneous* system. We assume that whenever a link-level entanglement is successfully generated, it always has fidelity one, but in certain cases we will consider decoherence post-generation. In such cases, we model coherence time as an exponential random variable (r.v.) with mean $1/\alpha$. Under this assumption, an entanglement's fidelity goes from one directly to zero upon decoherence, *i.e.*, the fi-

delity does not degrade while the entanglement is in storage. We also assume that measurements performed by the switch succeed with probability q[1].

If at any time there are fewer than $n = 2$ link-level entanglements, the switch may choose to store the available entangled qubits and wait until there are enough new ones generated to create an end-to-end entanglement. We assume that the switch can store $B \geq 1$ qubits in its buffer, per link. If on the other hand, there are more than n link-level entanglements, the switch must decide which set(s) of them to use in measurement(s). Such decisions can be made according to a pre-specified scheduling policy: for example, a user or a set of users may be given higher priority for being involved in an end-to-end entanglement. Other scheduling policies may be adaptive, random, or any number of hybrid policies. In Chapter 4, we assume that the switch uses the *Oldest Link Entanglement First (OLEF)* rule, wherein the oldest link-level entanglements have priority to be used in entangling measurements. A practical reason for this rule is that quantum states are subject to decoherence, which is a function of time; hence, our goal is to make use of link-level entanglements as soon as possible.[2] Note that when it comes to choosing Bell pairs to be used in entangling measurements, the OLEF rule is only applicable if the system is operating in discrete time: if we model the system in continuous time, there is never ambiguity about which link-level Bell pair should be used first, since there are no "time slots" and the switch performs a BSM as soon as two pairs are successfully generated[3]. On the other hand, when we model systems with a finite number of quantum memories, then

---

[1]With a linear optical circuit, four unentangled ancilla single photons and photon number resolving detectors, with all the devices being lossless, $q = 25/32 = 0.78$ can be achieved for BSMs [17]. With other technologies q close to 1 can be achieved [23].

[2]If the system is operating in discrete time, there may arise instances in which two or more links are tied for having the oldest entanglements. In such cases, as long as the switch follows the OLEF rule, sets of link-level entanglements are chosen at random for measurements, provided that each set consists of n entanglements belonging to n distinct links.

[3]Another way to think about the OLEF rule is that it always applies, but its decisions are trivial in continuous time.

(even in continuous time) there may occur scenarios in which a link has used up all its available memories and must decide whether to discard an older Bell pair in order to store a newly-generated one. In such a case, the OLEF rule does apply, and we discard the qubit associated with the oldest stored entanglement to make space for the qubit from the newly-generated Bell pair.

The state space of this system can be represented by a vector $\mathbf{Q}(t) \in \{0, 1, \ldots, B\}^k$, where the lth element corresponds to the number of stored entanglements at link l at time t. One consequence of the assumption that any set of n users always wish to share an entangled state is that at most $n-1$ distinct users will store entanglements at any time. Hence, throughout this chapter, up to one link may have a stored Bell pair after step two of entanglement distribution. Our goal is to derive expressions for system capacity C (*i.e.*, the number of end-to-end entanglements produced per time unit) and the expected number of stored qubits $E[Q]$. Throughout the chapter, we use the result that if the balance equations of an irreducible CTMC have a unique and strictly positive solution, then this solution represents the stationary distribution of the chain.

3.5 Continuous Time Markov Chain for Bipartite Switching

In this section, we introduce and analyze a CTMC model of a bipartite entanglement switch serving k users. We first assume that memories do not decohere and obtain expressions for capacity and the expected number of qubits stored at the switch. We then modify the model to incorporate decoherence and analyze it. Last, we derive an upper bound for the capacity of the switch.

3.5.1 The Heterogeneous Case

Assume μ_l depends on l, *i.e.*, the links are heterogeneous. For subsequent analysis, it is useful to define

$$\gamma \coloneqq \sum_{l=1}^{k} \mu_l,$$

the aggregate entanglement generation rate over all links. Also, let $\mathbf{e}_l$ be a size k vector with all zeros except for the lth component, which is 1, and let $\mathbf{0}$ be a vector of size k with all entries equal to 0.

We are interested in the stationary distribution and stability conditions for a heterogeneous system with infinite and finite buffers. As discussed in Section 3.4, in bipartite entanglement switching, only one link stores entanglements at a time, but since links generate entanglements at different rates, we must keep track of which link is associated with the stored entanglement(s). Let $\mathbf{Q}(t) = (Q_1(t), \dots, Q_k(t)) \in \{0, 1, 2, \dots\}^k$ represent the state of the system at time t, where $Q_l(t)$ is the number of entanglements stored at link l, $l \in \{1, \dots, k\}$, at time t. As a consequence of the scheduling policy described in Section 3.4, if $Q_i(t) > 0$ for some i, then $Q_j(t) = 0$, $j \neq i$. In other words, $\mathbf{Q}(t)$ only takes on values $\mathbf{0}$ or $j\mathbf{e}_l$, $l \in \{1, \dots, k\}$, $j \in \{1, 2, \dots\}$. Here, $\mathbf{0}$ represents the state where no entanglements are stored, and $j\mathbf{e}_l$ represents the state where the lth link has j stored entanglements.

Define the following limits when they exist:

$$\pi_0 = \lim_{t\to\infty} P(\mathbf{Q}(t) = \mathbf{0}),$$
$$\pi_l^{(j)} = \lim_{t\to\infty} P(\mathbf{Q}(t) = j\mathbf{e}_l).$$

Once we obtain expressions for π_0 and $\pi_l^{(j)}$, we can derive expressions for capacity and the expected number of stored qubits $E[Q]$.

3.5.1.1 Infinite Buffer

Figure 3.3 presents the CTMC for a switch with an infinite buffer. Consider state $\mathbf{0}$ (no stored entanglements). From there, a transition along one of the k "arms" of the

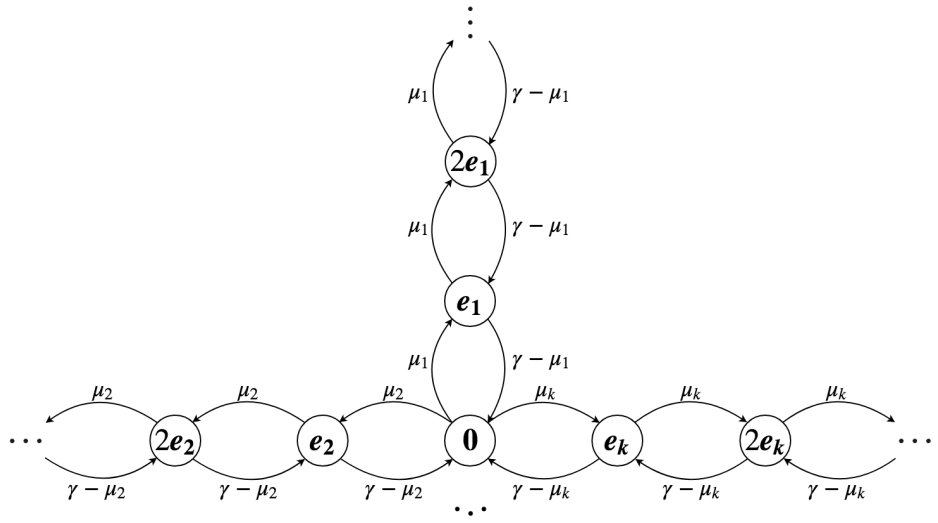


Figure 3.3: A CTMC for a k-user, infinite buffer, heterogeneous-link switch. μ_l is the entanglement generation rate of link l, while γ is the aggregate entanglement generation rate of all links. $\mathbf{e}_l$ is a vector of all zeros except for the lth position, which is equal to one.

CTMC occurs with rate μ_l, when the lth link successfully generates an entanglement. For a BSM to occur, any of the $k-1$ other links must successfully generate an entanglement: this occurs with rate $\gamma - \mu_l$. The balance equations are

$$\begin{aligned}
&\pi_0 \mu_l = \pi_l^{(1)}(\gamma - \mu_l),\ l \in \{1, \dots, k\},\\
&\pi_l^{(j-1)} \mu_l = \pi_l^{(j)}(\gamma - \mu_l),\ l \in \{1, \dots, k\},\ j \in \{2, 3, \dots\},\\
&\pi_0 + \sum_{l=1}^{k} \sum_{j=1}^{\infty} \pi_l^{(j)} = 1.
\end{aligned}$$

From above, we see that for $j = 1, 2, \dots,$

$$\pi_l^{(j)} = \rho_l^j \pi_0, \quad \text{where} \quad \rho_l \equiv \frac{\mu_l}{\gamma - \mu_l}, \forall\ l.$$

It remains to obtain π_0; we can use the normalizing condition:

$$\pi_0 + \pi_0 \sum_{l=1}^{k} \sum_{j=1}^{\infty} \rho_l^j = \pi_0 \left(1 + \sum_{l=1}^{k} \left(\sum_{j=0}^{\infty} \rho_l^j - 1 \right) \right) = 1.$$

Now, assume that for all $l \in \{1, \ldots, k\}$, $\rho_l < 1$. This implies that for all l, $\mu_l < \gamma/2$. This is the stability condition for this chain. Then,

$$\pi_0 = \left(1 + \sum_{l=1}^{k} \frac{\rho_l}{1-\rho_l}\right)^{-1}$$

and the capacity is

$$C = q \sum_{l=1}^{k} \sum_{j=1}^{\infty} \pi_l^{(j)} (\gamma - \mu_l) = \frac{q \sum_{l=1}^{k} \frac{\mu_l}{1-\rho_l}}{1 + \sum_{l=1}^{k} \frac{\rho_l}{1-\rho_l}} = \frac{q\gamma}{2}. \tag{3.1}$$

See Appendix B.1 for a proof of the last equality. The distribution of the number of stored entanglements is

$$P(Q = j) = \begin{cases} \pi_0, & \text{if } j = 0, \\ \sum_{l=1}^{k} \pi_l^{(j)} = \pi_0 \sum_{l=1}^{k} \rho_l^j, & \text{if } j > 0. \end{cases}$$

The expected number of stored entanglements is

$$E[Q] = \sum_{j=1}^{\infty} jP(Q = j) = \sum_{j=1}^{\infty} j\pi_0 \sum_{l=1}^{k} \rho_l^j = \frac{\sum_{l=1}^{k} \frac{\rho_l}{(1-\rho_l)^2}}{1 + \sum_{l=1}^{k} \frac{\rho_l}{1-\rho_l}}.$$

3.5.1.2 Finite Buffer

In the case of heterogeneous links and a finite buffer of size B, the CTMC has the same structure as in Figure 3.3, except that each "arm" of the chain terminates at $B\mathbf{e}_l$, $\forall l \in \{1, \ldots, k\}$. The balance equations are

$$\pi_0\mu_l = \pi_l^{(1)}(\gamma - \mu_l),\ l \in \{1,\dots,k\},$$
$$\pi_l^{(j-1)}\mu_l = \pi_l^{(j)}(\gamma - \mu_l),\ l \in \{1,\dots,k\},\ j \in \{2,\dots,B\},$$
$$\pi_0 + \sum_{l=1}^{k}\sum_{j=1}^{B}\pi_l^{(j)} = 1$$

and have solution

$$\pi_l^{(j)} = \rho_l^j \pi_0,\ l \in \{1,\dots,k\},\ j \in \{1,\dots,B\},$$

where ρ_l is defined as in the infinite buffer case. Then,

$$\pi_0\left(1 + \sum_{l=1}^{k}\sum_{j=1}^{B}\rho_l^j\right) = 1, \quad \text{hence} \quad \pi_0 = \left(1 + \sum_{l=1}^{k}\sum_{j=1}^{B}\rho_l^j\right)^{-1},$$

and the capacity is

$$C = q\sum_{l=1}^{k}\sum_{j=1}^{B}(\gamma - \mu_l)\pi_l^{(j)} = \frac{q\sum\limits_{l=1}^{k}\frac{\mu_l(1-\rho_l^B)}{1-\rho_l}}{1 + \sum\limits_{l=1}^{k}\frac{\rho_l(1-\rho_l^B)}{1-\rho_l}}. \tag{3.2}$$

The distribution of the number of stored qubits is given by

$$P(Q=j) = \begin{cases} \pi_0, & \text{if } j = 0, \\ \sum\limits_{l=1}^{k}\pi_l^{(j)} = \pi_0\sum\limits_{l=1}^{k}\rho_l^j, & \text{if } 0 < j \le B. \end{cases}$$

The expected number of stored qubits is

$$E[Q] = \sum_{j=1}^{B} jP(Q=j) = \frac{\sum\limits_{l=1}^{k}\frac{\rho_l\left(B\rho_l^{B+1} - (B+1)\rho_l^B + 1\right)}{(1-\rho_l)^2}}{1 + \sum\limits_{l=1}^{k}\frac{\rho_l(1-\rho_l^B)}{1-\rho_l}}.$$

The rate received by user l (connected to link l) is given by

$$C_l = q\left((\gamma-\mu_l)\sum_{j=1}^{B}\pi_l^{(j)} + \mu_l\sum_{\substack{m=1,\\ m\neq l}}^{k}\sum_{j=1}^{B}\pi_m^{(j)}\right), \tag{3.3}$$

where the first term represents the production of entanglements by link l (which get consumed by other links at rate $\gamma-\mu_l$) and the second term represents the consumption by link l of stored entanglements at other links. Note then, that if we were to sum all C_l, each end-to-end entanglement would be double-counted. Hence, $\sum C_l = 2C$. (Note: in the infinite-buffer case, $C_l = q\mu_l$, $l \in \{1,\ldots,k\}$; see Appendix B.1 for a proof. Then, $\sum C_l = q\gamma = 2C$, another proof of the last equality in Eq. (3.1).) The expected number of stored qubits at link l, $E[Q_l]$ can be obtained by taking the lth component of the sum in the numerator of the expression for $E[Q]$. In other words, when $B=\infty$,

$$E[Q_l] = \frac{\frac{\rho_l}{(1-\rho_l)^2}}{1+\sum_{l=1}^{k}\frac{\rho_l}{1-\rho_l}}.$$

For a homogeneous system, $E[Q_l] = E[Q]/k$.

3.5.2 The Homogeneous Case

Suppose all links (or users) have the same entanglement generation rates, *i.e.* $\mu_l = \mu$, $\forall\, l \in \{1,\ldots,k\}$. We can take advantage of this homogeneity as follows: since only one link can be associated with stored qubits at the switch at any given time, and all links have equal rates, it is only necessary to keep track of the *number* of stored entanglements, and not the *identity* of the link (or user). Hence, the state space of the CTMC can be represented by a single variable taking values in $\{0,1,\ldots,B\}$ where $B=\infty$ corresponds to the infinite buffer case, and $B<\infty$ the finite buffer case. We discuss each of these in detail next.

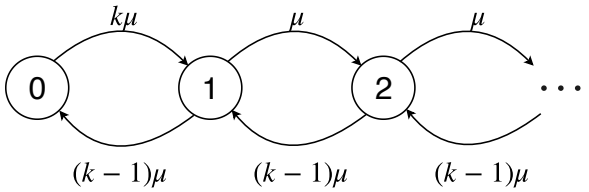


Figure 3.4: A CTMC model with k users, infinite buffer, and homogeneous links. μ is the entanglement generation rate.

3.5.2.1 Infinite Buffer

Figure 3.4 depicts the CTMC for k homogeneous links and $B = \infty$. When no entanglements are stored (system is in state 0), any of the k links can generate a new entanglement, so the transition to state 1 occurs with rate $k\mu$. Let S represent the link associated with one or more stored entanglements. From states 1 and above, transitioning "forward" (or gaining another entanglement in storage) occurs whenever link S generates a new entanglement. This event occurs with rate μ. Finally, moving "backward" through the chain (corresponding to using a stored entanglement, when the switch performs a BSM) occurs whenever any of the $k-1$ links other than S successfully generates an entanglement; this event occurs with rate $(k-1)\mu$. It is easy to show that when there are two links, the system is not stable (and a stationary distribution does not exist). Take, for instance, the stability condition for a heterogeneous system with infinite buffer from Section 3.5.1.1:

$$\mu_l < \frac{\gamma}{2} = \frac{\sum_{l=1}^{k} \mu_l}{2}.$$

Setting all μ_l's equal yields the stability condition $k > 2$ for the homogeneous system with infinite buffer. Henceforth, we only consider $k \geq 3$.

Note that the CTMC in Figure 3.4 is a birth-death process whose stationary distribution can be obtained using standard techniques found in literature (*e.g.* [42]).

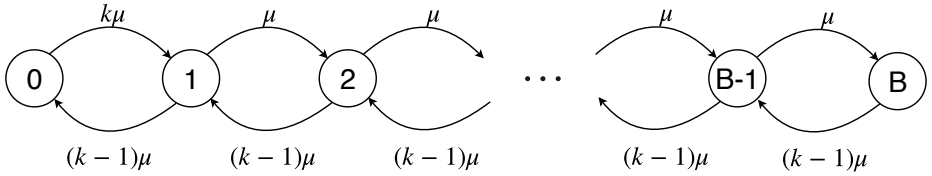


Figure 3.5: A CTMC model with k users, finite buffer of size B, and homogeneous links. μ is the entanglement generation rate.

The steady-state probability of being in state 0 is $\pi_0 = (k-2)/(2(k-1))$ and of being in state j is $\pi_j = k(k-2)/(2(k-1)^{j+1})$. The capacity is

$$C = q\sum_{i=1}^{\infty}\pi_i(k-1)\mu = q(k-1)\mu(1-\pi_0) = \frac{q\mu k}{2}.$$

The expected number of stored entangled pairs is given by

$$E[Q] = \sum_{i=0}^{\infty} i\pi_i = k\pi_0\sum_{i=1}^{\infty} i\left(\frac{1}{k-1}\right)^i = \frac{k}{2(k-2)}.$$

Note that the more general case of multipartite entanglement switching (*i.e.*, $n \geq 2$) for homogeneous-link systems with infinite buffer and no quantum state decoherence is covered in [54].

3.5.2.2 Finite Buffer

Figure 3.5 illustrates the CTMC for a system with k homogeneous links being served by a switch with finite buffer space B. When there are B stored entanglements and a new one is generated on link S, we assume that the switch drops the oldest stored entanglement, adhering to the OLEF policy. This CTMC is also a standard birth-death process whose solution can be found in literature (*e.g.* [42]) and has

$$\pi_0 = \frac{(k-2)(k-1)^B}{2(k-1)^{B+1} - k}.$$

Using the fact that $\sum_{i=1}^{B} \pi_i = 1 - \pi_0$, the capacity is

$$C = q \sum_{i=1}^{B} \mu(k-1)\pi_i = \frac{q\mu k \left(1 - \left(\frac{1}{k-1}\right)^B\right)}{2 - k\left(\frac{1}{k-1}\right)^{B+1}}.$$

Note that as $B \to \infty$, C for the finite buffer case approaches C for the infinite buffer case. The expected number of stored qubits is

$$E[Q] = \sum_{i=1}^{B} i\pi_i = \frac{k\left(B + (k-1)^{B+1} - (B+1)(k-1)\right)}{(2(k-1)^{B+1} - k)(k-2)}.$$

3.5.3 Decoherence

Assume now that quantum states in our system are subject to decoherence. Further, assume that all states decohere at the same rate α, even in the case of heterogeneous links, since coherence time is dependent on the quantum memories at the switch and not on the links themselves. Under the assumption that coherence time is exponentially distributed with rate $1/\alpha$, incorporating decoherence does not change the structure of the CTMC; it merely increases "backward" transition rates. Specifically, in the homogeneous case, the transition from any state $j \geq 1$ to state $j-1$ now has rate $(k-1)\mu + j\alpha$, where $j\alpha$ represents the aggregate decoherence rate of all j stored qubits. In the heterogeneous case, the transitions are modified in a similar manner for any state $j\mathbf{e}_l$, $l \in \{1, \ldots, k\}$, $j \geq 1$. The derivations of stationary distributions, capacities, and expected number of qubits stored are very similar to those for models without decoherence; we present the final relevant expressions here and leave details to Appendix B.2. All expressions below can be computed numerically.

Heterogeneous Links: For finite buffer size $B < \infty$,

$$\pi_0 = \left(1 + \sum_{l=1}^{k}\sum_{j=1}^{B}\prod_{i=1}^{j}\frac{\mu_l}{\gamma - \mu_l + i\alpha}\right)^{-1},$$
$$C = q\pi_0 \sum_{l=1}^{k}\sum_{j=1}^{B}(\gamma - \mu_l)\prod_{i=1}^{j}\frac{\mu_l}{\gamma - \mu_l + i\alpha},$$
$$E[Q] = \pi_0 \sum_{j=1}^{B} j \sum_{l=1}^{k}\prod_{i=1}^{j}\frac{\mu_l}{\gamma - \mu_l + i\alpha}.$$

For infinite-size buffer, let $B \to \infty$ in all expressions above.

Homogeneous Links: For finite buffer size $B < \infty$,

$$\pi_0 = \left(1 + k\sum_{i=1}^{B}\prod_{j=1}^{i}\frac{\mu}{((k-1)\mu + j\alpha)}\right)^{-1},$$
$$C = q(k-1)\mu(1-\pi_0),$$
$$E[Q] = \pi_0 k \sum_{i=1}^{B} i \prod_{j=1}^{i}\frac{\mu}{((k-1)\mu + j\alpha)}.$$

For infinite-size buffer, let $B \to \infty$ in all expressions above.

3.6 Numerical Observations

In this section, we investigate the capacity and buffer requirements of a bipartite entanglement switch based on our model. In particular, we are interested in how buffer capacity B and number of users k affect capacity and $E[Q]$. We then examine the effect of decoherence on homogeneous and heterogeneous switches with finite as well as infinite buffer capacities.

Throughout this section, we denote the distance of user l from the switch as L_l (measured in km). We assume that each user is connected to the switch with single mode optical fiber of loss coefficient $\beta = 0.2$ dB/km. We also assume that the switch is equipped with a photonic entanglement source with a raw (local) entanglement

generation rate of 1 Giga-ebits[4] per second. So, in every (1 ns long) time slot, one photon of a Bell state is loaded into a memory local to the switch, and the other photon is transmitted (over a lossy optical fiber) to a user, who loads the received photon into a memory (held by the user), which has a trigger which lets the user know the time slots in which their memory successfully loads a photon. Let us denote $\tau = 1$ ns as the time duration of one qubit of each entangled pair, and the entanglement generation rate between the switch and the user l, $\mu_l = c\eta_l/\tau$ ebits per second. Here, we take $c = 0.1$ to account for various losses other than the transmission loss in fiber, for example inefficiencies in loading the entangled photon pair in the two memories (at the switch and at the user), and any inefficiency in a detector in the memory at the user used for heralding the arrival of a photon (*e.g.*, by doing a Bell measurement over the received photon pulse and one photon of a locally-generated two-photon entangled state produced by the user). Here, η_l, the transmissivity of the optical fiber connecting user l and the switch is given by $\eta_l = 10^{-0.1\beta L_l}$. Channel loss to user l, measured in dB, is $10\log_{10}(1/\eta_l)$. Unless otherwise stated, all μ_l discussed in this section have units of Mega-ebits/sec.

3.6.1 Effect of Buffer Size: Homogeneous Links

In homogeneous-link systems, all users are equidistant from the switch (*i.e.* $L_l = L_m$, $\forall l, m \in \{1, \ldots, k\}$). In Figure 3.6, we compare models with infinite and finite buffer sizes as the number of links k is varied. Note that when links are homogeneous, $q\mu$ is simply a multiplicative factor in the expressions for C, and does not factor into formulas for $E[Q]$. Hence, we set $q\mu = 1$ for Figure 3.6 (left), and with $\mu = 1$, the links are 100 km long. For the finite buffer models, B is varied from one to five. Recall from Section 3.5.2.2 that as $B \to \infty$, the capacity of the finite-buffer model approaches

[4]An ebit is one unit of bipartite entanglement corresponding to the state of two maximally entangled qubits, the so-called Bell or EPR state.

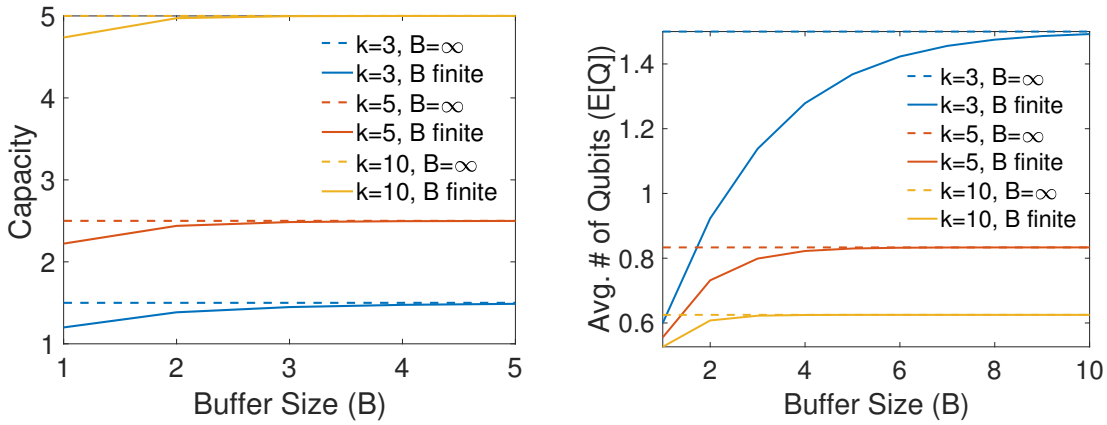


Figure 3.6: The effect of buffer size on capacity (left) and on the expected number of stored entanglements (right) in systems with homogeneous links. Capacity is in Mega-ebits/sec.

that of the infinite-buffer model, as expected, and note that the same is true when $k \to \infty$. Interestingly, this convergence occurs rapidly, even for the smallest value of k (3), and the maximum relative difference between the two capacities is 0.25 (even as μ increases). From this, we conclude that buffer does not play a major role in the capacity of a homogeneous system under the switching policy described in Section 3.4 and only a small quantum memory is required.

Figure 3.6 (right) shows the behavior of $E[Q]$ for infinite and finite buffer sizes and different values of k. As with capacity, the effect of buffer capacity on $E[Q]$ diminishes as k grows, and the largest relative difference occurs for $k = 3$ and $B = 1$, and equals 1.5 – less than two qubits. Note from the expressions for $E[Q]$ in Sections 3.5.2.1 and 3.5.2.2 that as $k \to \infty$, $E[Q] \to 1/2$. Numerically, we observe that convergence to this value occurs quickly: even for $k = 25$, $E[Q]$ is already 0.54 for both the infinite and finite models.

3.6.2 Effect of Buffer Size: Heterogeneous Links

Figure 3.7 illustrates how buffer size and number of users affect C and $E[Q]$ for a set of heterogeneous systems. We vary the number of links from three to nine. For

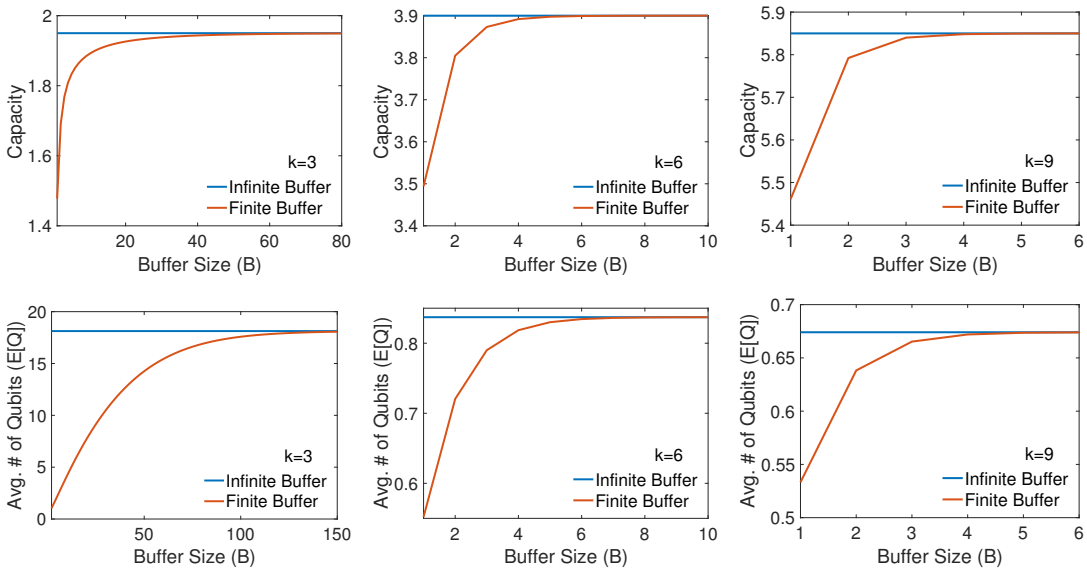


Figure 3.7: Capacity (Mega-ebits/sec) and expected number of qubits in memory $E[Q]$ for heterogeneous systems with varied number of links and buffer sizes. Links are divided into two classes: one class generates entanglements approximately twice as quickly as the other class.

each value of k, the links are split into two classes: links in the first class successfully generate entanglements at rate μ_1 and those in the second class at rate μ_2. We set $\mu_1 = 1.9\mu_2$ and $\mu_2 = 1$. This setting corresponds to links in class one having lengths 86 km and links in class two having lengths 100 km. Values of μ_1 and μ_2 are chosen in a manner that satisfies the stability condition for heterogeneous systems: recall from Section 3.5.1.1 that for all $l \in \{1, \ldots, k\}$, μ_l must be strictly less than half the aggregate entanglement generation rate. For all experiments, $q = 1$ since it only scales capacity.

For each value of k, the ratio of class 1 to class 2 links is 1:2 (so $k = 3, 6, 9$ have one, two, and three class 1 links, respectively). As with the homogeneous-link systems, we observe that the slowest convergence of the finite-buffer metrics C and $E[Q]$ to corresponding infinite-buffer metrics is for smaller values of k and the largest relative difference is for smaller values of B. However, the rate of convergence speeds up quickly as k increases from 3 to 6: with the latter, convergence is already observed

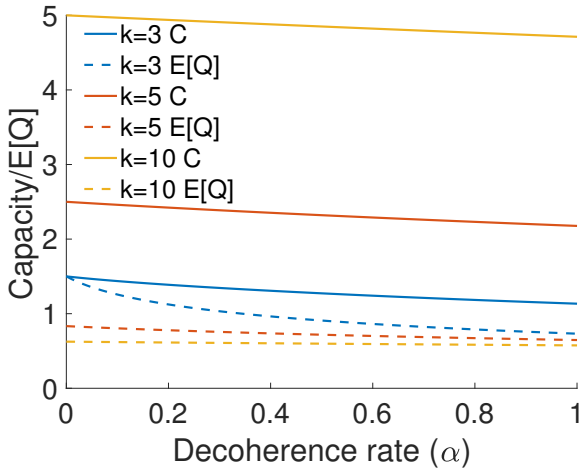


Figure 3.8: Effect of decoherence on capacity (Mega-ebits/sec) and expected number of stored qubits $E[Q]$, for varying number of users k. For all experiments, the entanglement generation rate is $\mu = 1$ for all links.

for $B < 10$. Meanwhile, when $k = 9$, there is little benefit in having storage for more than two qubits. Another interesting observation is that quantum memory usage is large when $k = 3$ but not for larger values of k. This is due to the system operating closer to the stability constraints for $k = 3$ than larger values of k. In the next section, we will see another example of a system that operates near the boundary of its stability region. In such cases, C and $E[Q]$ can be affected significantly as B is varied.

3.6.3 Effect of Decoherence

In this section, we study the effect of decoherence on capacity and expected number of stored qubits $E[Q]$. We set $q = 1$ for all experiments since it only scales capacity. Figure 3.8 presents C and $E[Q]$ for a homogeneous system with $\mu = 1$ (corresponding to 100 km long links), $B = \infty$ and different values of k, as decoherence rate α varies from 0 (the equivalent of previous models that did not incorporate decoherence) to $\mu = 1$. Note that in practice, α is expected to be much smaller than μ. We observe that even as α approaches μ decoherence does not cause major degra-

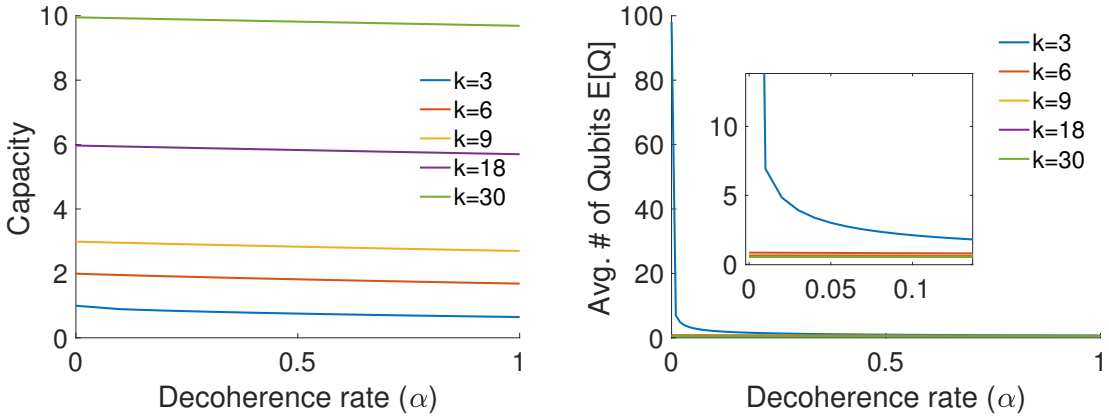
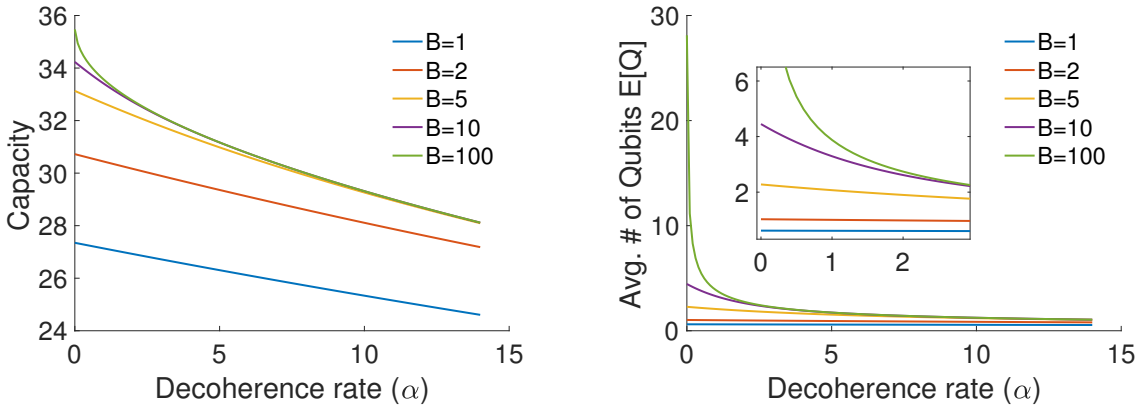


Figure 3.9: Effect of decoherence on capacity (Mega-ebits/sec) and expected number of stored qubits $E[Q]$, for varying number of users k. In all experiments, the links are heterogeneous.

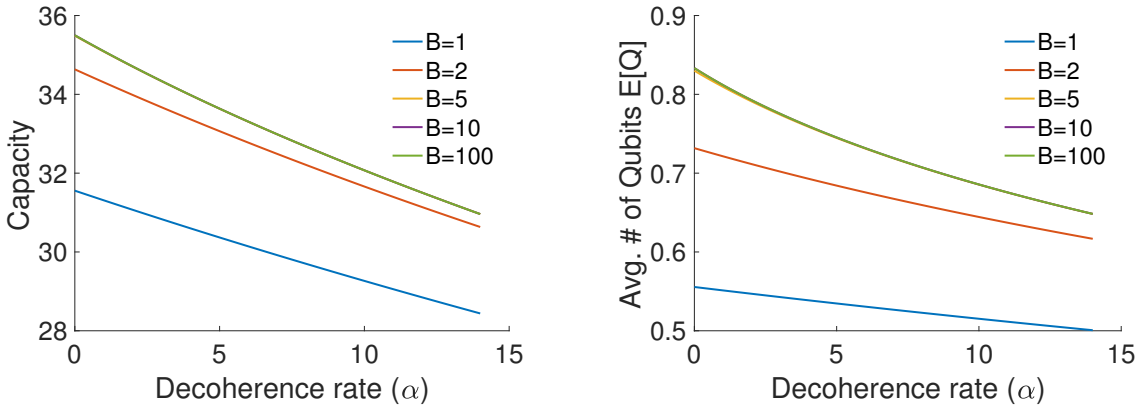
dation in capacity for homogeneous systems, and likewise does not introduce drastic variations in $E[Q]$.

Figure 3.9 presents the effect of α on the performance of a heterogeneous system with infinite-size buffer. In these experiments, entanglement generation rates are set in a similar manner to that of Section 3.6.2, with two classes of links configured so that the first class generates entanglements almost twice as fast as the second class (here, $\mu_1 = 0.99$ and $\mu_2 = 0.5$, corresponding to 100.2 km and 115 km long links for class one and two, respectively), and the number of links in class one to those in class two is 1:2. In these experiments, for each value of k, capacity behaves much as it would in a homogeneous system with μ set as the average of the μ_l from the heterogeneous system. Note that for $k = 3$, $E[Q]$ is very large when $\alpha = 0$; similar to the experiment in Figure 3.7 (see panel with $k = 3$) this is because the system is operating near the boundary of its stability region. In all other cases, $E[Q]$ is close to 0.

In Figure 3.10(a), we focus on a heterogeneous system that operates near the boundary of its stability region and observe the effects of both decoherence and buffer size on C and $E[Q]$. There are five links, with entanglement generation rates



(a) Heterogeneous-link system



(b) Homogeneous-link system

Figure 3.10: Effect of decoherence on capacity (Mega-ebits/sec) and expected number of stored qubits $E[Q]$ for $k = 5$ links and varying buffer sizes B. In (a), μ_l are (35 15 15 3 3), and in (b), μ is the average of μ_l, $l = 1, \ldots, 5$, *i.e.* 14.2. For all plots above, $B = 100$ curves behave equivalently to $B = \infty$.

(35 15 15 3 3) Mega-ebits/sec, corresponding to link lengths of 22.8, 41.2, 41.2, 76, and 76 km, respectively. For this system, $\gamma/2 = 35.5$, so the fastest link is just below the constraint when $\alpha = 0$. The average of the μ_l is 14.2, so α is varied from 0 to this value. B is varied from 1 to 100, with the latter being close enough to mimic infinite buffer behavior for C and $E[Q]$. Figure 3.10(b) presents the performance of a homogeneous system with $k = 5$ and $\mu = 14.2$ for a comparison. We observe that the homogeneous system achieves higher capacity for all values of B, even though the average entanglement generation rate is the same for both systems. Further, the

homogeneous system is more robust to changes in buffer size than the heterogeneous system: for the former, $B = 5, 10$ are equivalent to $B = 100$. Further, note that for $B = 100$ and $\alpha = 0$ the heterogeneous system performs almost as well as the homogeneous system in terms of capacity, but the memory usage is much higher for the former. Finally, for this buffer size, as α increases, the homogeneous system is more robust to the effects of decoherence: capacity degrades by 7.35 Mega-ebits/sec for the heterogeneous system between $\alpha = 0$ and $\alpha = 14$, while it degrades by 4.54 Mega-ebits/sec for the homogeneous system.

3.7 Conclusion

In this work, we examined variants of a system with k users who are being served bipartite entangled states by a quantum entanglement distribution switch in a star topology. Each user is connected to the switch via a dedicated link; we considered both the case of homogeneous and heterogeneous links. We also analyzed cases in which the switch has finite or infinite buffer space for storing entangled qubits. We obtained simple and intuitive expressions for switch capacity, as well as for the expected number of qubits in memory when the switch operates at or near capacity.

We made numerical comparisons of these two metrics while varying the number of users k and buffer sizes B. We observed that in most cases, little memory is required to achieve the performance of an infinite-memory system. We also made numerical observations for models that incorporate decoherence and concluded that the degradation of quantum states in homogeneous systems has little effect on performance metrics, while it can have more significant consequences in heterogeneous systems that operate near the boundaries of their stability regions.

CHAPTER 4

ON THE EXACT ANALYSIS OF AN IDEALIZED QUANTUM SWITCH

4.1 Introduction

In this chapter, we continue our study of the most basic and fundamental component of a quantum network – a single quantum switch. As in Chapter 3, we assume that the switch serves k users in a star topology and that each user has a dedicated link connected to the switch. Recall that in Chapter 3, the objective was to characterize the performance of such a device, for example by determining its capacity (defined as the maximum achievable rate of entanglement switching), and deriving expressions for the expected quantum memory occupancy under various assumptions – *e.g.*, while assuming a particular quantum memory coherence time or limitations on the available number of memories. In this chapter, we also construct a model that abstracts away various architecture and physical implementation details about the system, *e.g.*, the method used for entanglement generation or how quantum memories are realized, but in contrast to Chapter 3, here we use a discrete-time Markov chain (DTMC) instead of a continuous-time Markov chain (CTMC).

In Chapter 3, we demonstrated that certain physical phenomena, such as state decoherence, link heterogeneity, and finite quantum storage, are fairly easy to incorporate into a CTMC. An unfortunate property of our DTMC model is that it is difficult to extend to include the aforementioned system characteristics. Thus, we focus here on the simplest variant of the problem, wherein the links connecting the users to the switch are identical, there is no quantum state decoherence, and the

switch has infinite capacity to store qubits (as in Chapter 3, we often use the term *buffer* in reference to the quantum storage at the switch). Nevertheless, there is value in studying a quantum switch using a DTMC, as the system is inherently a discrete-time system. Hence, while CTMCs have been shown to be more expressive as a modeling technique, there will undoubtedly be some differences in the resulting performance metrics. To quantify these differences, and determine whether CTMCs are a reasonable approximation to the original system, we compare the performance metrics obtained from both models.

Following is a summary of the results:

– the capacity of the switch is given by

$$C = \frac{qkp}{2},$$

where k is the number of users or links, p is the probability of successfully generating entanglement at the link level, and q is the probability of a successful swapping operation;

– when the switch operates at capacity, the expected number of stored qubits is given by

$$E[Q] = \frac{1+\beta}{2(1-\beta)},$$

where Q is the number of qubits stored at the switch in steady state, across all links, and β is in the interval $(0, 1)$ and is the solution to the following equation:

$$(\beta p + \overline{p})^{k-1}(p + \beta\overline{p}) - \beta = 0;$$

– the DTMC is stable if and only if $k \geq 3$;

– the expression for the capacity of the switch obtained using the DTMC matches exactly that of the CTMC model from Chapter 3. On the other hand, the CTMC overestimates the expected number of qubits in memory in steady state, but since the discrepancy is not significant, we conclude that the CTMC model is a reasonable approximation to the behavior of the system considered in this work.

The rest of this chapter is organized as follows: in Section 4.2, we introduce the relevant background and discuss related work on quantum switch modeling. In Section 4.3, we formally introduce the DTMC model and state the objectives. The analysis is performed in Section 4.4. In Section 4.5, we compare the DTMC introduced in this chapter with the CTMC model from Chapter 3. We conclude in Section 4.6.

4.2 Background and Related Work

In general, all quantum states are subject to decoherence, which can be thought of as leakage of information from the quantum system into the environment. Fidelity, a number in $[0, 1]$, is a measure of closeness of a possibly mixed state to the desired pure state, with unit fidelity implying that the two states have equivalent representations. In this work, we assume that each successfully-generated quantum state has unit fidelity and that the quantum memories used for storing qubits are capable of noiseless storage and have infinite coherence times. While these assumptions create a highly idealized scenario, it is nevertheless valuable to study as the analysis will yield an upper bound on the entanglement switching rate of a quantum switch operating under more realistic conditions.

In Chapter 3, we introduce a CTMC to analyze a quantum switch that serves only bipartite end-to-end entangled states to pairs of users. While it is easier to extend this model to represent systems that are more complex than that of this work, an important question that arises is whether the CTMC model is a fair approximation

to a more realistic DTMC model. We answer this question in Section 4.5, from the perspective of the chains' stability condition and expressions for switch capacity and expected number of qubits in memory at the switch in steady state. In [54], we use a CTMC to analyze a multipartite entanglement distribution switch for a similarly idealized scenario as studied in our work: identical links, no quantum state decoherence, unit fidelities, and infinite quantum storage. While this switch serves n-partite Greenberger-Horne-Zeilinger (GHZ) states [55], note that setting $n = 2$ yields precisely the model presented in Chapter 3 (and thus, the analytical results are equivalent for the two CTMCs).

Some analyses focus on specific quantum repeater architectures or protocols; *e.g.*, in [26] the authors perform a rigorous and detailed analysis of the repeater architecture proposed in [67], accounting for various non-idealities at the channel, detectors, and quantum memories. In contrast, our take on analysis is from a rather opposite perspective in that we use mathematical tools to abstract away as many details of the physical platform as possible, while keeping only a few relevant and important parameters in order to complete a high-level analysis and gain a clear understanding of how they relate to the performance metrics of interest.

Note that the applications of the problem we have formulated in this work extend far beyond entanglement switching. In general, one may view the system as a stochastic assembly-like queue, or a "kitting" process, *e.g.*, as in [68, 62, 36], since in a sense, the switch "assembles" longer-distance entangled states using shorter-distance ones, whose "arrival" into the system is driven by a stochastic process. Interestingly, none of these similar problem formulations found in literature have a direct correspondence to our problem, as in our case, the number of users being serviced by the central node is allowed to be, in theory, infinite, and our goal is to derive exact results, as opposed to approximate ones, or bounds. Hence, the problem studied here is a novel one, and the results derived in this work are of independent interest to queueing theory.

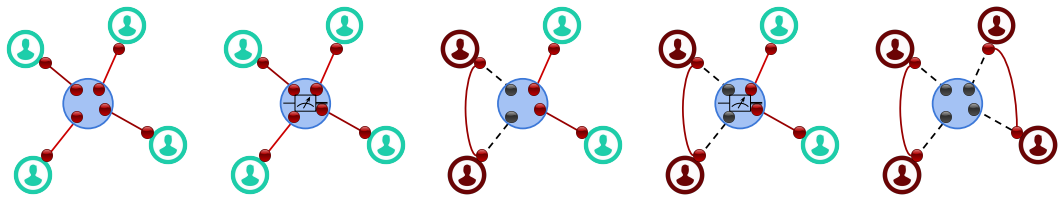


Figure 4.1: Example switch operation for a single time slot. At the beginning of the slot, **(a)**, all links have successfully generated Bell pairs. In **(b)**, the switch performs a BSM to entangle the two users on the left, see **(c)**. Next, still within the same time slot, the switch performs another BSM to entangle the two users on the right, shown in **(d)**, **(e)**.

4.3 Switch Description and Objectives

For completeness, we first restate the assumptions regarding switch operation from Chapter 3, but in the context of bipartite entanglement distribution performed in *discrete time*. We then introduce new modeling assumptions, as discrete-time switch operation causes some ambiguity with regard to which Bell pairs should be used in a measurement, and we must specify a switching policy that resolves this ambiguity.

Figure 3.2a illustrates the initial problem setup: $k \geq 2$ users are connected to the quantum switch via dedicated, identical links. Assume that time is slotted; the rest of Figure 3.2 presents an example of a sequence of events that may take place in subsequent time slots. The purpose of the switch is to facilitate end-to-end entanglement generation for pairs of users that request it. The creation of an end-to-end entanglement involves two steps. First, in each time slot users attempt to generate pairwise entanglements with the switch, which we call link-level entanglements. A successful link-level entanglement results in a two-qubit Bell state, with one qubit stored at the switch and the other stored at a user. In step two, the switch chooses two locally-held qubits, each entangled with a qubit held in a user's quantum memory, and such that the two users wish to share an entangled state, and performs a BSM.

If the measurement is successful, the result is a two-qubit maximally-entangled state between the corresponding pair of users. The switch continues to fulfill entanglement requests as long as there are available link-level entanglements for users who wish to communicate. If, at the end of the time slot, there are available link-level Bell pairs, but the switch cannot use them to fulfill requests based on current user demands, then the switch may choose to store the available entangled qubits in local quantum memories until these qubits can be used in entangling measurements. This two-step process is then repeated in the next time slot. Figure 4.1 illustrates a sequence of events within a single time slot.

One of our objectives is to derive the capacity of a quantum switch that operates as described above. This quantity serves as a useful benchmark against which to compare the performance of future entanglement switching protocols. In this work, we will also compute the expected number of qubits stored in memory at the switch, while the device operates at or near capacity. With this expression, we may obtain insight on the practical memory requirements of a switch. The capacity of the switch is defined as the maximum achievable entanglement switching rate of the device. This rate cannot be achieved with an arbitrary switching policy, or for an arbitrary set of user demands – if the switch is constrained to fulfill specific user requests, then the resulting rate would likely fall below the capacity. One way to ensure that the switch operates at capacity is to allow it to perform a BSM as soon as there are at least two Bell pairs available on two distinct links, during a given time slot. This amounts to the assumption that any pair of users wish to communicate within each time slot. The BSMs are assumed to take up a negligible amount of time, and the switch may perform as many of them as necessary in a single time slot, until there are no longer two distinct links with available Bell pairs.

Further, in this work we assume that the switch uses the *Oldest Link Entanglement First (OLEF)* rule when deciding which two users to pair up for an entangling

measurement. As the name suggests, when using this rule the switch prioritizes the oldest link-level Bell pairs for a BSM, as long as they belong to two different links. When there is more than one possible choice for such a pairing (*e.g.*, if there are three link-level Bell pairs of identical age and they are the oldest in the system), then the switch may choose any two at random. Note that the OLEF rule does not affect the switch capacity, but it does happen to minimize the number of stored Bell pairs at the end of each time slot and thus this rule affects the qubit occupancy distribution. Finally, to ensure that the end users being serviced by the switch do not limit switch performance, we allow end nodes to have infinite and noiseless quantum storage capability.

Recall from Section 4.2 that we study a somewhat idealized version of a quantum switch in this work in that the device has an infinite number of noiseless quantum memories, and quantum states that are successfully generated (either at the link or end-to-end level) have unit fidelities and are not subject to decoherence. Studying this simplified scenario is both valuable and prudent: the analysis performed here helps to lay out the foundation for – and possibly inspire – future work in quantum switch modeling, and our model serves as an easily-applicable comparison basis for alternate quantum switch models, such as that of Chapter 3. Finally, note that the capacity of this "idealized" quantum switch can also serve as an upper bound on the capacity of more "limited" systems, such as those with finite quantum memories, non-unit quantum state fidelities, and explicit user requests. Note also that one may obtain an upper bound on the capacity of a system with non-identical links, simply by converting that system into one with identical links, where each link behaves as the most efficient link – in terms of successful entanglement generation – of the original system.

4.4 Analysis

In this section, we describe the DTMC model and present its analysis. Our goal is to derive the switch capacity C (*i.e.*, the number of end-to-end entanglements produced per time unit), the expected number of stored qubits $E[Q]$ in steady state, and system stability conditions. A note on mathematical notation: in this chapter, we will use the convention that for any $y > x$, the term $\binom{x}{y} = 0$.

4.4.1 Model Description

We model a switch serving k users, each of whom has a separate, dedicated link to the switch, as a slotted system where each slot is of length τ seconds. Both link-level entanglement generation and entangling measurements can be modeled as probabilistic phenomena [26]. In this work, we model the former as follows: at each time step of length τ seconds, all k users attempt to generate link-level entanglements. In general, link l successfully generates an entanglement with probability $p_l \approx e^{-\alpha L}$, where L is the length of the link (*e.g.*, optical fiber) and α its attenuation coefficient. Since in this chapter, we assume that all links are identical, *i.e.*, they are of equal lengths and have the same attenuation coefficient, the success probabilities of all entanglement are equal. Hence, let p denote the probability that an entangled pair is successfully established on any link, and define $\overline{p} \equiv 1 - p$. Then the expected time to successfully create a link entanglement is given by τ/p (this will be useful in Section 4.5, when we make a comparison to a CTMC model). We assume that whenever a link-level entanglement or an end-to-end entanglement is successfully generated, it always has fidelity one to the corresponding ideal Bell state. We also assume that measurements performed by the switch succeed with probability q. As discussed in the previous section, we assume that any pair of users wishes to "communicate" (*i.e.*, share an entangled state) as long as link-level entanglements are available, and that the switch serves BSMs based on the OLEF policy described in detail in Section 4.3.

Table 4.1: Notation for the DTMC model.

Notation	Description
p	probability of a successful link entanglement
S	link with stored entanglements
P_f	probability of gaining an entanglement in memory
P_s	probability of remaining in current state
$P_{(j)}$	probability of using j of the stored entanglements
$P_{j,0}$	probability of going from state $j \in \{0, \ldots, k-1\}$ to 0
$P_{j,1}$	probability of going from state $j \in \{0, \ldots, k\}$ to 1

Note that because of our switching policy, only one link will have stored entanglements, since whenever a distinct pair of users has link-level entanglements, they are immediately paired up for a BSM. As a consequence of this and the identical-link assumption, it is not necessary to keep track of which link has stored entanglements: one need only keep track of *how many* are stored. Hence, the state space is given by $\Omega = \{0, 1, 2, \ldots\}$. Let S denote the link that has at least one stored entanglement. Figure 4.2 illustrates the possible transitions from a state $i \geq k+1$ (as we will see later, transitions for states $i \in \{0, 1, \ldots, k\}$ require special consideration). Table 4.1 provides a notation reference that is used in the analysis.

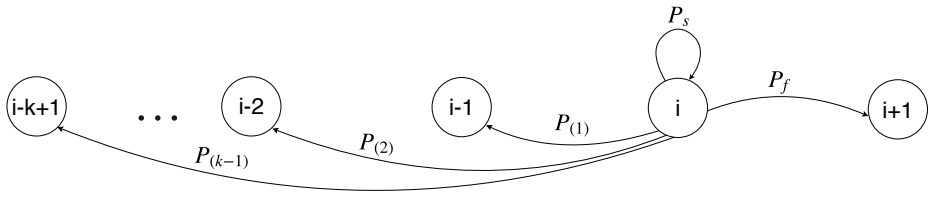


Figure 4.2: A DTMC model with k users, infinite buffer, and identical links. Here, $i \geq k+1$, P_f is the probability of advancing forward in the Markov chain, P_s is the probability of remaining in the current state, and $P_{(j)}$ is the probability of going back j states.

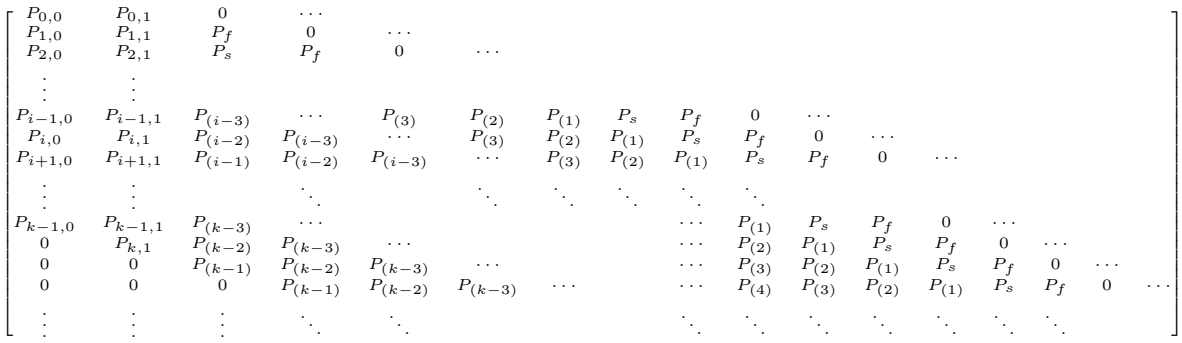


Figure 4.3: Transition probability matrix P for the DTMC model.

4.4.2 Analysis

First, we fully define the transition probabilities for this chain. We expect the stationary distribution to have a geometric form and show this to be true. However, a closed-form solution is not obtainable for large k, as it requires solving a polynomial of degree $k-1$ for an unknown factor, β. On the other hand, not having a closed-form solution for the stationary probability vector does not preclude us from deriving a simple expression for the capacity of the switch – it is $qkp/2$. We will also show that this system is stable if and only if $k \geq 3$. Finally, we also obtain a simple expression for the expected number of qubits in memory at the switch, but are constrained to compute it numerically due to its dependence on β.

4.4.2.1 Transition Probabilities

Figure 4.3 presents the transition probability matrix P for this DTMC. Note that repetition begins after the kth row of the matrix. We derive expressions for all non-zero transition probabilities. In the discussion that follows, we say that a link "succeeds" or "fails" for brevity, when referring to a link that successfully generates an entanglement or fails to do so, respectively. Throughout the following, we will also often refer to link S, which has at least one stored entanglement. First, consider any state $i > 1$. The transitions for this state are described as follows:

$i \to i+1$**:** the only way to advance forward in the chain is if S successfully generates a new entanglement, but all other links fail to do so. This probability is given by

$$P_f = p\overline{p}^{k-1}.$$

$i \to i$**:** there are two ways to remain in the current state: (a) all links fail or (b) S succeeds and only one of the $k-1$ other links succeeds. This occurs with probability

$$P_s = \overline{p}^k + (k-1)p^2\overline{p}^{k-2}.$$

$i \to i-j$, for $j \in \{1, \ldots, M\}$, where $M = k-1$ if $i \geq k+1$ and $M = i-2$ otherwise. Here, M signifies the maximum number of stored entanglements that can be used up when starting from state i. Note that even in the case all k links succeed and $i \geq k$, only $k-1$ of the stored entanglements get used: the entanglement that was generated by S cannot be paired with another entanglement from S. As stated above, we compute transition probabilities to states 0 or 1 separately, since they require special consideration. This is why $M = i-2$ for states $i < k+1$. Keeping these constraints in mind, the transition from i to $i-j$ occurs in two types of events:

(a) S fails and exactly j of the $k-1$ other links succeed,

(b) S succeeds and exactly $j+1$ of the $k-1$ other links succeed.

These events occur with probability

$$\begin{aligned} P_{(j)} &= \overline{p}\binom{k-1}{j}p^j\overline{p}^{k-1-j} + p\binom{k-1}{j+1}p^{j+1}\overline{p}^{k-1-(j+1)} \\ &= \binom{k-1}{j}p^j\overline{p}^{k-j} + \binom{k-1}{j+1}p^{j+2}\overline{p}^{k-j-2}. \end{aligned}$$

Next, we discuss transitions to states 0 and 1, which, unlike the probabilities above, depend on the value i of the state from which the transitions occur. To help with this task, we will first need to be able to compute two types of probabilities: the first is the probability that out of k link-level entanglement events, $j \geq i$ succeed, where j is either zero or an even number, and we call this probability $P_e(i,k)$; and the second is the probability that out of k events, $j \geq i$ succeed, where j is an odd number, and we call this $P_o(i,k)$. To compute these, we will use the following two indicator functions:

$$\mathbb{1}\{j \text{ is 0 or even}\} \coloneqq \frac{1+(-1)^j}{2}, \quad \mathbb{1}\{j \text{ is odd}\} \coloneqq \frac{1-(-1)^j}{2}.$$

Then,

$$P_e(i,k) = \sum_{j=i}^{k} \left(\frac{1+(-1)^j}{2}\right) \binom{k}{j} p^j \overline{p}^{k-j},$$
$$P_o(i,k) = \sum_{j=i}^{k} \left(\frac{1-(-1)^j}{2}\right) \binom{k}{j} p^j \overline{p}^{k-j}.$$

Now, for any state i, $1 \leq i \leq k$, the transition to state 1 occurs under the following conditions:

If i is even:

1. S fails and $j \geq i-1$ others succeed, j odd.
2. S succeeds and $j \geq i$ others succeed, j even.

$$P_{i,1} = \overline{p} P_o(i-1, k-1) + p P_e(i, k-1).$$

If i is odd:

1. S fails and $j \geq i-1$ others succeed, j even.

2. S succeeds and $j \geq i$ others succeed, j odd.

$$P_{i,1} = \overline{p}P_e(i-1, k-1) + pP_o(i, k-1).$$

Similarly, for any state $i \in \{1, \ldots, k-1\}$, transitioning to state 0 occurs under the following conditions:

If i is even:

1. S fails and $j \geq i$ others succeed, j even.
2. S succeeds and $j \geq i+1$ others succeed, j odd.

$$P_{i,0} = \overline{p}P_e(i, k-1) + pP_o(i+1, k-1).$$

If i is odd:

1. S fails and $j \geq i$ others succeed, j odd.
2. S succeeds and $j \geq i+1$ others succeed, j even.

$$P_{i,0} = \overline{p}P_o(i, k-1) + pP_e(i+1, k-1).$$

In the special case of $0 \to 0$, either all must fail or there must be an even number of entanglements. Hence, $P_{0,0} = P_e(0, k)$. Finally, in the special case of $0 \to 1$, there must be an odd number of entanglements, given by $P_{0,1} = P_o(1, k)$.

4.4.2.2 Stationary Distribution

We will show that a stationary distribution exists for $k \geq 3$. The balance equations for the DTMC are as follows:

$$\sum_{i=0}^{k-1} \pi_i P_{i,0} = \pi_0, \tag{4.1}$$

$$\sum_{i=0}^{k} \pi_i P_{i,1} = \pi_1. \tag{4.2}$$

For any state $i \geq 2$, the balance equations have the form:

$$\pi_{i-1} P_f + \pi_i P_s + \pi_{i+1} P_{(1)} + \cdots + \pi_{i+k-1} P_{(k-1)} = \pi_i, \tag{4.3}$$

and finally, the normalizing condition is

$$\sum_{i=0}^{\infty} \pi_i = 1. \tag{4.4}$$

We postulate that $\pi_i = \beta^{i-1}\pi_1$ for $i \geq 2$, with $\beta \in (0, 1)$. Introducing this value of π_1 in Eq. (4.3) yields $f(\beta) = 0$, with

$$f(\beta) \coloneqq (\beta p + \overline{p})^{k-1}(p + \beta\overline{p}) - \beta, \tag{4.5}$$

see Appendix C.1.1 for a proof. To show that $\pi_i = \beta^{i-1}\pi_1$ for $i \geq 2$ is indeed the solution to this system, we must prove that:

1. There exists $\beta \in (0, 1)$ satisfying Eq. (4.5), and that this β is unique.

2. Given the solution above, note that both Eqs (4.1) and (4.2) can be written in terms of only π_1 and π_0. Hence, for the proposed solution to be valid, one of these equations must be redundant, *i.e.*, we must show that Eq. (4.1) is equivalent to Eq. (4.2).

In Appendix C.1.2, we prove that the first statement above holds for $k \geq 3$ and that it does not hold for $k = 2$, implying that the DTMC is stable if and only if $k \geq 3$. The second statement is proven in Appendix C.1.3. We conclude that the proposed form for π_i, $i \geq 2$ is valid. Moreover, we can derive expressions for π_0 and π_1 in terms of β. From the normalizing condition (4.4), we have

$$\pi_0 = 1 - \frac{\pi_1}{1-\beta}. \tag{4.6}$$

In Appendix C.1.3, we rearranged (4.1) to look as follows:

$$\sum_{i=1}^{k-1} \beta^i P_{i,0} = \frac{\beta \pi_0}{\pi_1} P_{0,1} \tag{4.7}$$

and also showed that the left side of Eq. (4.7) equals

$$\begin{aligned}
&\frac{1}{2}\left[\frac{\beta}{1-\beta} - \frac{2\beta}{1-\beta^2}(p\beta+\overline{p})^{k-1}(p+\overline{p}\beta) - (\overline{p}-p)^k\frac{\beta}{1+\beta}\right] \\
&\quad = \frac{1}{2}\left[\frac{\beta}{1-\beta} - \frac{2\beta^2}{1-\beta^2} - (\overline{p}-p)^k\frac{\beta}{1+\beta}\right] \text{ by Eq. (4.5)}, \\
&\quad = \frac{1}{2}\left[\frac{\beta}{1+\beta} - (\overline{p}-p)^k\frac{\beta}{1+\beta}\right].
\end{aligned}$$

Therefore, Eq. (4.7) becomes

$$\begin{aligned}
\frac{\beta\pi_0}{\pi_1}P_{0,1} &= \frac{1}{2}\left[\frac{\beta}{1+\beta} - (\overline{p}-p)^k\frac{\beta}{1+\beta}\right], \text{ or} \\
\frac{\pi_0}{\pi_1}P_{0,1} &= \frac{1-(\overline{p}-p)^k}{2(1+\beta)}.
\end{aligned} \tag{4.8}$$

Next, we compute

$$\begin{aligned}
P_{0,1} = P_o(1,k) &= \sum_{i=1}^{k} \frac{1-(-1)^i}{2}\binom{k}{i}p^i\overline{p}^{k-i} \\
&= \sum_{i=0}^{k} \frac{1-(-1)^i}{2}\binom{k}{i}p^i\overline{p}^{k-i} = \frac{1}{2} - \frac{1}{2}(\overline{p}-p)^k.
\end{aligned}$$

Substituting this into Eq. (4.8),

$$\begin{aligned}
\frac{\pi_0}{\pi_1}\left(\frac{1-(\overline{p}-p)^k}{2}\right) &= \frac{1-(\overline{p}-p)^k}{2(1+\beta)}, \\
\frac{\pi_0}{\pi_1} &= \frac{1}{1+\beta}, \\
1-\frac{\pi_1}{1-\beta} &= \frac{\pi_1}{1+\beta} \text{ by Eq. (4.6)}, \\
\pi_1 &= \frac{1-\beta^2}{2}. \qquad (4.9)
\end{aligned}$$

Now, we can compute π_0 in terms of only β:

$$\pi_0 = 1-\frac{\pi_1}{1-\beta} = 1-\left(\frac{1}{1-\beta}\right)\frac{1-\beta^2}{2} = \frac{1-\beta}{2}.$$

4.4.2.3 Capacity and Qubits in Memory

Let Q represent the number of stored qubits at the switch. Let N denote the number of end-to-end entangled pairs generated in one time step of the DTMC. Then the capacity is defined as follows:

$$C = q\sum_{i=0}^{\infty}\pi_i E[N|Q=i].$$

To compute this expression, we consider two separate cases: case 1 is when $i \geq k-1$ and case 2 is when $i < k-1$. In case 1, there can be at most $k-1$ entanglements; the expected number is given by

$$E[N|Q=i\geq k-1] = \sum_{j=0}^{k-1} j\binom{k-1}{j}p^j\overline{p}^{k-1-j} = (k-1)p.$$

For case 2, we can have up to $i+m$ entanglements, where $m = \lfloor\left(\frac{k-i}{2}\right)^+\rfloor$. The expected number is then given by

$$E[N|Q=i<k-1] = \sum_{j=0}^{i+m} jP(N=j|Q=i\leq k-2).$$

For the sum above, consider first $j \in \{0, \ldots, i\}$. Here, we are looking for the probability that there are fewer new entanglements than the number stored, so the probability that we generate j pairs is given by

$$P(N = j | Q = i,\ i \leq k-2) = \binom{k-1}{j} p^j \overline{p}^{k-1-j}.$$

However, note that the case $j = i$ is a special one: another way we can generate i entanglements is if there are a total of $i+1$ successes from the $k-1$ links that have nothing stored, while S fails. Then, the extra entanglement has no pair, and the total number of pairs generated is still i. This is given by

$$i\binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1}.$$

Next, we focus on the case where $j \in \{i+1, \ldots, i+m\}$. After the first i successes, there need to be anywhere from 2 to at most $k-i$ "extra" successes to generate new pairs. Denote the number of these extra successes by the variable $l \in \{2, \ldots, k-i\}$, and the number of new pairs (or BSMs) generated from them is $\left\lfloor \frac{l}{2} \right\rfloor$. Then we can write the second sum as follows:

$$\sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l}.$$

Combining everything we have learned, we obtain

$$\begin{aligned} C = q \sum_{i=0}^{k-2} \pi_i \Bigg(& \sum_{j=0}^{i} j \binom{k-1}{j} p^j \overline{p}^{k-1-j} + i \binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1} \\ & + \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} \Bigg) + q(k-1)p \sum_{i=k-1}^{\infty} \pi_i. \end{aligned} \tag{4.10}$$

In Appendix C.2, we show that the above evaluates to

$$C = \frac{qkp}{2}. \tag{4.11}$$

Next, we derive the expected number of qubits stored at the switch, $E[Q]$. This is given by

$$E[Q] = \sum_{i=1}^{\infty} i\pi_i = \pi_1 \sum_{i=1}^{\infty} i\beta^{i-1} = \frac{\pi_1}{\beta} \sum_{i=1}^{\infty} i\beta^i$$
$$= \frac{\pi_1}{\beta} \frac{\beta}{(1-\beta)^2} = \frac{1-\beta^2}{2(1-\beta)^2} = \frac{1+\beta}{2(1-\beta)}. \tag{4.12}$$

4.5 Comparison of DTMC Model with a CTMC Model

In this section, we compare the DTMC model from this work to a CTMC model studied in Chapter 3 and validate the latter for the case of a system with identical links, no quantum state decoherence, and infinite quantum memories at the switch. Let us first review the CTMC model and the analytical results from Chapter 3. There, we model entanglement generation at the link level as a Poisson process with parameter μ representing the rate of successful Bell pair creation on any link. For the identical-link, bipartite-only switching case, the CTMC is a simple birth-death process, with each state representing the number of stored qubits corresponding to a single link (note that the assumption that any pair of users wish to communicate is also required for the capacity computation). The resulting capacity of the switch is

$$C_{\text{CTMC}} = \frac{qk\mu}{2}.$$

Recall that in the discrete model, the amount of time it takes to successfully generate a link entanglement is given by τ/p. In the continuous model, the rate of successful entanglement generation is μ, so the time to generate an entanglement is $1/\mu$. Hence, $\tau/p = 1/\mu$ or equivalently, $\mu = p/\tau$. Then, note that the DTMC capacity that we derived in Section 4.4.2.3 is the capacity per time slot of length τ seconds. Therefore, in order to make a comparison against the CTMC capacity, we

must perform a unit conversion: divide the discrete capacity by τ in order to obtain the number of entanglement pairs per *second*, as opposed to per *time slot*. This yields

$$C_{\text{DTMC}} = \frac{qkp}{2\tau} = \frac{qk\mu}{2} = C_{\text{CTMC}}.$$

We conclude that the capacities produced by the DTMC and CTMC models match exactly.

Next, we compare the expected number of qubits in memory in steady state at the switch, $E[Q]$ as predicted by the DTMC and the CTMC models. The CTMC analysis yields the following expression:

$$E[Q]_{\text{CTMC}} = \frac{k}{2(k-2)}.$$

Note that this expression has no dependence on μ, the link-level successful entanglement generation rate, implying that according to the CTMC model, the expected number of stored qubits in steady state does not depend on the probability p of successfully generating a Bell pair on a link. On the other hand, we can see from Eq. (4.12) that $E[Q]$ resulting from the DTMC analysis *does* depend on p, as it is a function of β. Hence, the DTMC is more accurately able to describe the buffer occupancy in steady state, at the cost of not being able to produce a closed-form expression for $E[Q]$.

Figure 4.4 compares numerically the discrete and continuous $E[Q]$'s as the number of users k and probability p vary. For each value of p and k, we use Eq. (4.5) to numerically solve for β. For each value of k, we report the maximum relative error, defined as

$$maxRelErr(k) = \max_{p\in(0,1)} \frac{|E[Q]_{\text{DTMC}}(k,p) - E[Q]_{\text{CTMC}}(k,p)|}{E[Q]_{\text{DTMC}}(k,p)},$$

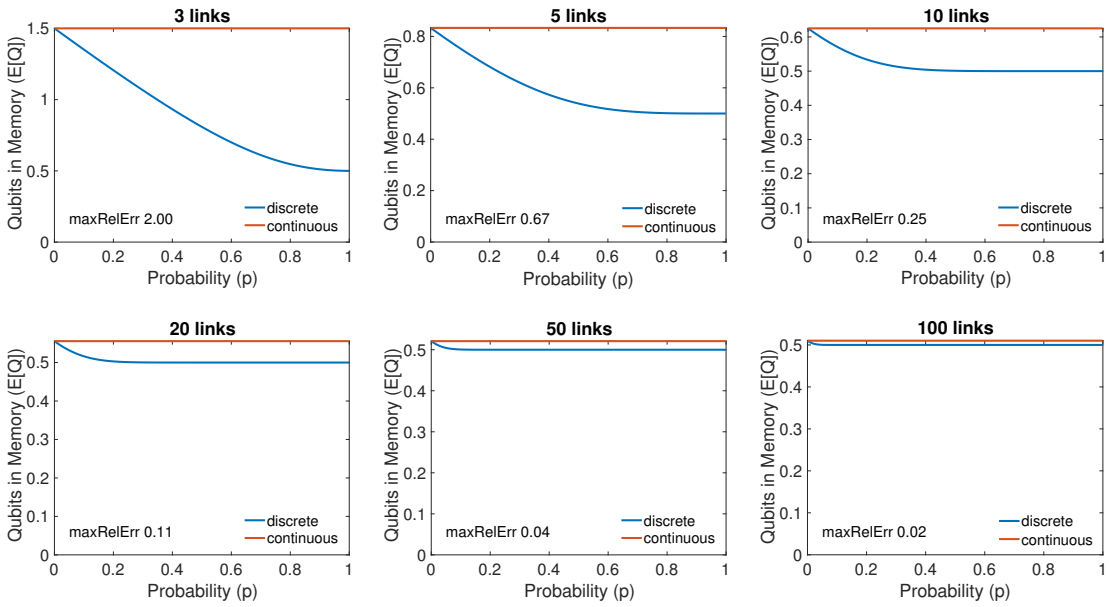


Figure 4.4: Comparison of the expected number of qubits in memory $E[Q]$ for the DTMC and CTMC models, as the number of links is varied $\in \{3, 5, 10, 20, 50, 100\}$ and for entanglement generation probabilities $p \in (0, 1)$. maxRelErr is the maximum relative error between the discrete and continuous expressions for $E[Q]$.

where $E[Q]_{\text{DTMC}}$ and $E[Q]_{\text{CTMC}}$ are the discrete and continuous functions for $E[Q]$, respectively. We observe that the error is largest when p is close to 1. Note that

$$\lim_{p \to 1} f(\beta) = \lim_{p \to 1} (\beta p + \overline{p})^{k-1}(p + \beta\overline{p}) - \beta = \beta^{k-1} - \beta.$$

Since $f(\beta) = 0$, we conclude that as $p \to 1$ and $k \to \infty$, $\beta \to 0$ (note: $\beta = 1$ is always a root of $f(\beta)$, but we always discard this root because it is not in $(0, 1)$). As $\beta \to 0$, $E[Q] \to 1/2$ according to Eq. (4.12), which is consistent with the numerical observations. Meanwhile, as $k \to \infty$, the continuous $E[Q]$ also approaches $1/2$. We conclude that as $k \to \infty$, $maxRelErr \to 0$, which can be observed in Figure 4.4. Also, the largest $maxRelErr$ occurs for the lowest value of $k = 3$, when $p \to 1$. But even in this (worst case), although the error is $maxRelErr(3) = 2$, it corresponds to discrete and continuous versions of $E[Q]$ differing by a prediction of only a single qubit.

Finally, both the continuous- and discrete-time Markov chains have stationary distributions if and only if $k \geq 3$. From these analytic and numerical observations, we conclude that the CTMC model is sufficiently accurate so as to be used to explore issues such as decoherence, link heterogeneity, and switch buffer constraints.

In Chapter 3, we introduce CTMCs for systems where the switch has buffer constraints, the links are not necessarily identical, and quantum memory coherence time is finite. The construction and analyses of these models is relatively simple compared to the DTMC model of this work. Even if one were to introduce a finite buffer into this model, several changes would be required to state transitions and balance equations, resulting in even more complex expressions for the stationary distribution (recall that even in the infinite-buffer case, we must solve for it numerically). Attempting to model decoherence in discrete time would require one to consider all possible combinatorial settings of stored qubit decoherence, further complicating the transition probabilities, but also increasing the *number* of possible transitions from each state. Consider, for instance, state i in Figure 4.2: each of the existing "backward" transitions $P_{(j)}$, $j \in \{1, \ldots, k-1\}$ would have to be modified based on the number of ways that l qubits can decohere and m new entanglements can be generated such that $l+m=j$, and in addition, extra transitions must be added from state i to states $\{0, 1, \ldots, i-k+2\}$ because any number of the stored qubits can decohere. This process can become highly cumbersome and prone to mistakes, while CTMCs seem to offer much more in modeling power, albeit incurring an accuracy cost that so far has only been quantified for the simplest variant of the entanglement switching problem.

4.6 Conclusion

We study an entanglement distribution switch that serves bipartite entangled states to pairs of users connected to the device via dedicated links. Using a DTMC,

we study a simple variant of the problem, wherein the links are identical, the switch has an infinite number of quantum memories, and quantum states do not decohere, although entanglement generation may fail both at the link level and at the end-to-end level. By studying this basic system, we learn that the DTMC model exhibits limitations such that introducing additional constraints to this model, such as finite buffers or quantum state decoherence, makes the resulting model exceedingly difficult to analyze, and therefore may not be the most attractive option for modeling more complex entanglement switching mechanisms.

We derive the capacity of the switch, the expected number of stored qubits at the switch in steady state, and the stability conditions for the system. We also derive the stationary distribution of the DTMC, albeit not in closed form. The results of our analysis are then compared to a different model, based on a CTMC. We conclude that while the CTMC model is more easily analyzed, it is less accurate than the DTMC model. For the expected number of stored qubits, the discrepancies between the two models are quantified, and we find that in the worst case, the predictions differ by less than one qubit. Hence, we conclude that the CTMC is a suitable model for this particular variant of the problem, but more work will be required in order to completely assess the accuracy of CTMC models for more complex switching scenarios. Our work is the first attempt at quantum switch analysis via a DTMC, and while the problem formulation is relatively simple, the analysis of the system is non-trivial. Moreover, the expression for switch capacity derived in this work can be used as an upper bound on the capacity of more complex systems, such as those with non-identical links and where quantum states may decohere. Finally, while our work was initially inspired by entanglement switching, the problem is independently interesting from a queueing-theoretic perspective, and the results can be applied to any stochastic assembly-like queueing system that services two customers/jobs at a time.

CHAPTER 5

ON THE CAPACITY REGION OF BIPARTITE AND TRIPARTITE ENTANGLEMENT SWITCHING

5.1 Introduction

Multi-qubit entangled states are fundamental building blocks for quantum computation, sensing, and security. Consequently there is a need for a quantum network that can generate such entanglement on demand between pairs and groups of users [59, 65, 72]. In this chapter, we study the performance of the simplest multi-user network, a star-topology quantum switch connecting k users, where each user is connected to the switch via a separate link. Bipartite, two-qubit maximally-entangled states, *i.e.*, Bell pairs (or EPR states) are generated at a constant rate across each link, with the qubits stored at local quantum memories at each end of the links. As these link entanglements start appearing, the switch uses two-qubit Bell-state measurement (BSM) between pairs of locally-held qubits and three-qubit Greenberger-Horne-Zeilinger (GHZ) basis measurements between triples of locally-held qubits to provide two-qubit and three-qubit entanglements to pairs and triples of users, respectively [55]. The capacity of such a switch to provide these two types of entanglements to the users depends on the switching mechanism, the number of quantum memories and their decoherence rates, and the number of links.

In this chapter, we study the capacity region when the switch can store either $B = 1$ or $B = 2$ qubits for each link at any given time. The number of quantum memories available to a link is referred to as its buffer size. We consider a simple time division multiplexing (TDM) policy between the two types of entanglements, along

with a class of randomized policies. When properly configured, the latter provide higher capacities than TDM. However the relative difference between the two policies goes to zero as $k \to \infty$. We also observe that increasing the number of memories from one to two increases capacity but that the increase diminishes as k increases. We also explore the effect that decoherence—the locally stored qubits at each end of the link being subject to a noise process that reduces the entanglement between the two qubits—has on capacity. In the cases of $B = 1$ with and without decoherence, we have simple closed form expressions for capacity whereas for the case of $B = 2$, our results are numerical.

The remainder of this chapter is organized as follows: in Section 5.2, we provide relevant background and related work. In Section 5.3, we formulate the problem and propose a method for solving it. In Section 5.4, we present the case where the system has a per-link buffer of size one, and provide analytical and numerical results. In Section 5.5, we present numerical results for the case where the system has a per-link buffer of size two and observe similar behavior to the buffer size one case. In Section 5.6, we introduce a simple technique for modeling quantum state decoherence and use it to examine the effect of decoherence on the bipartite-tripartite capacity region for systems with per-link buffer sizes one and two. For the former, we also have analytical results. We make concluding remarks in Section 5.7.

5.2 Background and Related Work

As pointed out in previous chapters, Bell states are an integral part of a diverse set of distributed quantum applications, including Quantum Key Distribution (QKD) [4, 13], superdense coding [7], teleportation [5], and distributed quantum computation [39]. Similarly, GHZ states can be used to implement a variety of quantum protocols, such as cryptographic conferencing [22], quantum sensing [15], and multipartite generalization of superdense coding [31]. The advantage of these applications is that they

offer functionality that cannot be achieved classically, *e.g.*, information-theoretic security. Future quantum networks will likely support a diverse set of applications, thus it is likely that quantum devices (such as switches and repeaters) may be equipped with the ability to perform different types of projective measurements.

In Chapter 3, we studied a switch that serves only BSMs, but state decoherence, link heterogeneity, and arbitrary buffer sizes (including infinite) are considered. In [54], we studied a multipartite entanglement distribution switch that serves n-partite GHZ states to users, for $n \geq 3$. In this work, links are assumed to be identical and the effects of state decoherence negligible. In contrast to this prior work, we no longer assume that the switch serves only one type of entangled state, *i.e.*, we allow n to be *either* two *or* three, and our goal is to design and evaluate a suitable switching policy. Another contrast to [54] is that there, the quantum switch is modeled to have an infinite number of quantum memories, while in this work we consider finite buffer sizes that scale with the number of links.

5.3 System Description and Assumptions

We consider a switch that connects k users over k separate links. The creation of an end-to-end entanglement requires two steps. First two-qubit Bell states are generated pairwise between a qubit stored locally at the switch and a qubit owned by a user. Once such link-level two-qubit entangled states have been created, the switch performs joint (entangling) measurements (over $j \geq 2$ locally-held qubits that are entangled with qubits held by j distinct users), which, if successful, produces a j-qubit maximally-entangled state between the corresponding j users. Link-level entanglement generation, as well as entangling measurements, when realized with practical systems, are inherently probabilistic [26]. We assume that only two-user (two-qubit) and three-user (three-qubit) entanglements are created, *i.e.*, BSMs and 3-qubit GHZ basis measurements are done at the switch. For simplicity, we will

assume that these $j = 2$ or 3 qubit measurements at the switch take negligible time and always succeed.

For completeness of this chapter, we restate some of the modeling assumptions from Chapters 4 and 3. Each link attempts two-qubit entanglements in each time slot of length τ seconds, and with probability p, establishes one entangled pair successfully. For simplicity, we model the time to successfully create a link entanglement as an exponential random variable with mean $1/\mu = \tau/p$. We assume that each link can store $B = 1, 2, \ldots$ qubits. We also assume that qubits at the switch can decohere and model decoherence time as an exponential r.v. with mean $1/\alpha$. We assume a step-function decoherence model where the two-qubit entanglement goes from a maximally-entangled qubit pair (one ebit) to zero entanglement. In this work, we only consider $B = 1, 2$. Last, when a qubit is stored at the switch, with its entangled pair stored at a user, we refer to this as a *stored link entanglement*.

We assume that all possible bipartite and tripartite user entanglements are of interest and consider two classes of probabilistic policies, one for $B = 1$ and the second for $B = 2$, that provide the flexibility to generate both types of entanglements with arbitrary rates. Policies in both classes incorporate the *oldest link entanglement first* (OLEF) rule whereby when a link entanglement is created it is always matched up with stored link entanglements when possible rather than be stored. This has the nice consequence, when coupled with the assumption that links are homogeneous but statistically independent, that the system can be modeled by a continuous time Markov chain (CTMC) where the state simply tracks the number of stored entanglements for two users. The next section describes the class of policies for $B = 1$ and Section 5.5 for the class of $B = 2$ policies.

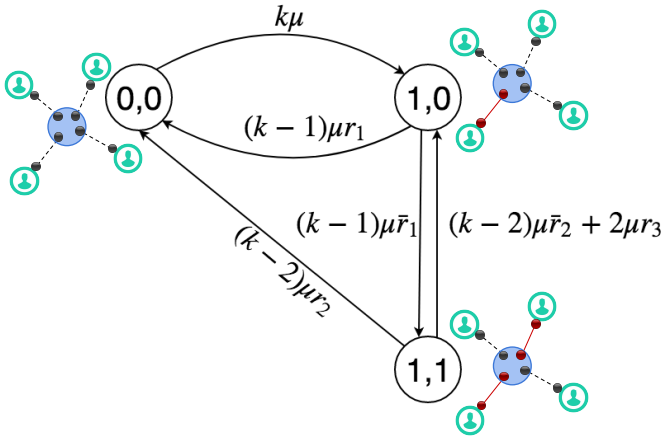


Figure 5.1: CTMC for a system with at least three links and buffer size one for each link. k is the number of links, μ is the rate of entanglement generation, and r_1, r_2, and r_3 are parameters that specify the scheduling policy.

5.4 System with Per-Link Buffer Size One

In this section, we assume that each link can store one qubit in the buffer, so that the per-link buffer size $B = 1$. We model this system using a CTMC, and by obtaining its stationary distribution, we are able to compute the capacity region of the switch. We discover that it is always possible to configure a randomized policy that outperforms TDM, although as the number of links grows, the advantage of using such a policy diminishes.

5.4.1 Description

In a system where the switch can make tripartite measurements, we must keep track of two variables for each state of the CTMC: each representing a link with a stored qubit. Hence, $(1, 1)$ represents the state where two of the k links have a qubit stored, one each. Note that we do not need to keep track of all links individually due to the OLEF rule and link homogeneity assumption. States $(1, 0)$ and $(0, 0)$ represent cases where only one link has a stored qubit or no link has a qubit, respectively.

The system is fully described in Figure 5.1. For a variable $x \in [0, 1]$, we use the notation $\bar{x} \equiv 1 - x$. When the system is in state $(0, 0)$, new entanglements are

generated with rate $k\mu$; this is the rate of transitioning from $(0,0)$ to $(1,0)$. When the system is in state $(1,0)$, any new entanglements generated on the link that already has one stored qubit causes the switch to drop one of the qubits. New entanglements on other links are generated with rate $(k-1)\mu$, and the switch must decide whether to immediately use the two qubits for a BSM or keep both and wait for a new link entanglement. To generalize the policy as much as possible, we add a policy parameter, $r_1 \in [0,1]$, that specifies the fraction of time the switch performs a BSM. Note that $r_1 = 1$ corresponds to the policy of always using qubits for BSMs. While this maximizes C_2, it also means that $C_3 = 0$.

Now, suppose that the system is in state $(1,1)$ and a third link generates an entanglement. This event occurs with rate $(k-2)\mu$. The switch has two choices: either use all three qubits for a tripartite measurement, or choose two of them for a BSM. We add another policy parameter, $r_2 \in [0,1]$, that specifies fractions of times the switch performs a BSM and tripartite measurements in the event of three qubits on three different links. Another event that can occur in the $(1,1)$ state is the generation of an entanglement on either of the two links that already have stored entanglements. This event occurs with rate 2μ. Since $B = 1$, the switch cannot store the new entanglement. A decision must be made: to either discard one of the link entanglements (and remain in state $(1,1)$) or perform a BSM on two of them and keep the third (and transition to state $(1,0)$). Since it is not clear which policy is most advantageous, we add another parameter, $r_3 \in [0,1]$, which specifies the fraction of time that the switch performs a BSM when it resides in this state.

5.4.2 Numerical Results

We plot the capacity region for the switch with $B = 1$ for all values of r_1, r_2, $r_3 \in [0,1]$ and compare it against TDM. The entanglement generation rate μ simply scales the capacities, so we set it equal to one. In Figure 5.2, the number of links is

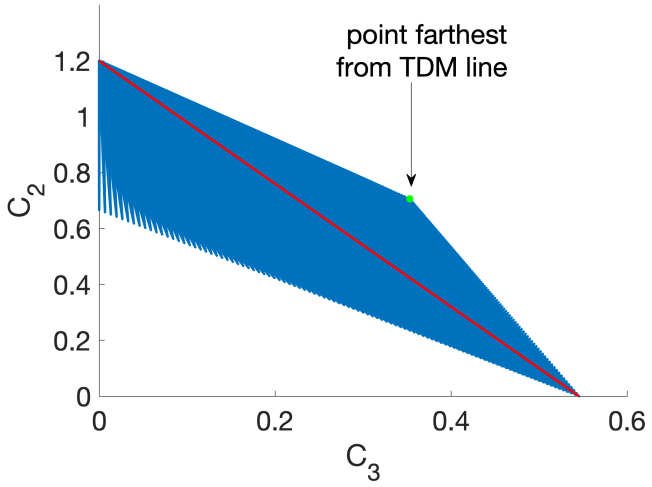
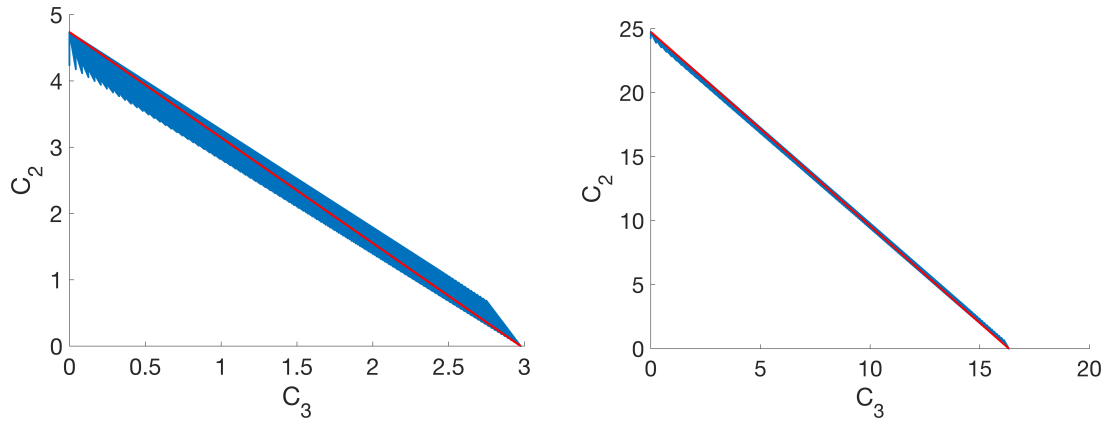


Figure 5.2: Capacity region for a system of buffer size one and three links. The red line represents the set of TDM policies.

three, and the TDM line is shown in red. Clearly, it is possible to design a policy that yields better performance than TDM: the triangular blue region above TDM represents the maximum capacities of the set of such policies.

Recall that TDM connects points $(0, C_2^*)$ and $(C_3^*, 0)$, where C_2^* and C_3^* are the maximum achievable capacities for bi- and tripartite measurements, respectively. The point farthest from and above TDM (the vertex of the triangular region above the line, shown in green in Figure 5.2) is achieved by setting $r_1 = 0$ and $r_2 = r_3 = 1$. In other words, the most "efficient" policy in terms of being the farthest from the TDM line is the following: (i) never perform BSMs in state $(1, 0)$; and (ii) when in state $(1, 1)$ and a third entanglement is generated on a *different* link, always use it in a tripartite measurement, but when a third entanglement is generated on one of the links that already has a stored qubit, *always* perform a BSM. Note that the latter rule directs the switch to not waste an entanglement whenever it is possible to use it in a measurement.

The capacity regions for $k = 10$ and 50 are shown in Figure 5.3. Note that as the number of links increases, the differences between TDM and the more efficient



(a) $k = 10$ (b) $k = 50$

Figure 5.3: Capacity region for a system of buffer size one and varying number of links. The red line represents the set of TDM policies.

random policies diminish. In the next section, we provide an analytical proof of this phenomenon.

5.4.3 Analysis

Let $\pi(0,0)$, $\pi(1,0)$, and $\pi(1,1)$ represent the stationary distribution of the CTMC in Figure 5.1. The balance equations (excluding μ, as it cancels out due to every transition rate being its multiple), are:

$$\pi(0,0)k = \pi(1,0)(k-1)r_1 + \pi(1,1)(k-2)r_2,$$
$$\pi(1,1)((k-2)r_2 + (k-2)\bar{r}_2 + 2r_3) = \pi(1,0)(k-1)\bar{r}_1,$$
$$\pi(0,0) + \pi(1,0) + \pi(1,1) = 1.$$

Solving these equations yields

$$\pi(1,1) = \frac{k(k-1)\bar{r}_1}{D},$$
$$\pi(1,0) = \frac{k(k-2+2r_3)}{D}, \text{ where}$$
$$D = (k-2+2r_3)((k-1)r_1 + k) + (k-1)\bar{r}_1((k-2)r_2 + k).$$

Then the bipartite and tripartite capacities for this system, $C_2 \equiv C_2(r_1, r_2, r_3)$ and $C_3 \equiv C_3(r_1, r_2, r_3)$, are

$$\begin{aligned}
C_2 &= \pi(1,0)(k-1)\mu r_1 + \pi(1,1)((k-2)\mu\bar{r}_2 + 2\mu r_3) \\
&= \frac{k(k-1)\mu(k-2+2r_3-(k-2)r_2\bar{r}_1)}{(k-2+2r_3)((k-1)r_1+k)+(k-1)\bar{r}_1((k-2)r_2+k)}, \\
C_3 &= \pi(1,1)(k-2)\mu r_2 \\
&= \frac{k(k-1)(k-2)\mu r_2\bar{r}_1}{(k-2+2r_3)((k-1)r_1+k)+(k-1)\bar{r}_1((k-2)r_2+k)}.
\end{aligned}$$

Claim 5.4.1. *The maximum value of C_2 is given by $C_2^* = C_2(r_1, 0, 1) = C_2(1, 0, r_3)$, where r_1 and r_3 are arbitrary values in $[0, 1]$. The maximum value of C_3 is given by $C_3^* = C_3(0, 1, 0)$.*

Proof. We start by proving this for C_2. First, note that to maximize C_2's numerator and minimize its denominator, r_2 must be set to 0. This yields

$$\begin{aligned}
C_2(r_1, 0, r_3) &= \frac{k(k-1)\mu(k-2+2r_3)}{(k-2+2r_3)((k-1)r_1+k)+k(k-1)\bar{r}_1} \\
&= \frac{k(k-1)\mu}{(k-1)r_1+k+\frac{k(k-1)\bar{r}_1}{k-2+2r_3}}.
\end{aligned}$$

Now, $r_3 = 1$ maximizes $C_2(r_1, 0, r_3)$, which yields

$$C_2(r_1, 0, 1) = \frac{k(k-1)\mu}{(k-1)r_1+k+(k-1)\bar{r}_1} = \frac{k(k-1)\mu}{2k-1}.$$

Note that $C_2(1, 0, r_3)$ yields the same expression as $C_2(r_1, 0, 1)$. Next, consider the expression for C_3. To minimize the denominator, we should set $r_3 = 0$. This yields

$$\begin{aligned}
C_3(r_1, r_2, 0) &= \frac{k(k-1)(k-2)\mu r_2\bar{r}_1}{(k-2)((k-1)r_1+k)+(k-1)\bar{r}_1((k-2)r_2+k)} \\
&= \frac{k(k-1)(k-2)\mu r_2\bar{r}_1}{(k-1)r_1((k-2)\bar{r}_2-k)+k(k-2)+(k-1)((k-2)r_2+k)}.
\end{aligned}$$

It is clear that r_1 must be 0, which yields

$$\begin{aligned} C_3(0, r_2, 0) &= \frac{k(k-1)(k-2)\mu r_2}{k(k-2) + (k-1)((k-2)r_2 + k)} \\ &= \frac{k(k-1)(k-2)\mu r_2}{k(2k-3) + (k-1)(k-2)r_2} \\ &= \frac{k(k-1)(k-2)\mu}{\frac{k(2k-3)}{r_2} + (k-1)(k-2)}. \end{aligned}$$

From above, we can see that r_2 must be 1, so the maximum is at $C_3^* = C_3(0, 1, 0)$. □

For brevity, let $(C_3(0,1,1), C_2(0,1,1)) \equiv (\hat{C}_3, \hat{C}_2)$; this is the point farthest above the TDM line within the achievable capacity region (*e.g.*, the green point in Figure 5.2). We prove this as part of the proof of the claim below.

Claim 5.4.2. *Any point (C_3, C_2) in the achievable capacity region satisfies the following constraints:*

$$C_2 \leq -\frac{3k-2}{2k-1} C_3 + \frac{\mu k(k-1)}{2k-1} \text{ and} \tag{5.1}$$

$$C_2 \leq -\frac{k(k-2) + 2(k-1)^2}{k(k-2)} C_3 + \mu(k-1), \tag{5.2}$$

$$C_2,\ C_3 \geq 0. \tag{5.3}$$

Moreover (5.1) and (5.2) define a tight upper bound on the achievable capacity region.

Proof. First, we must show that the point $(\hat{C}_3, \hat{C}_2)$ is indeed the farthest from the TDM line. To do so, let us find a point (C_3, C_2) on the plane such that the (negative) slope of the line that passes through it and $(0, C_2^*)$ is maximized. This is equivalent to minimizing the quantity

$$\frac{C_2^* - \hat{C}_2}{\hat{C}_3} = \frac{(3k-2)(k-2)r_2 + 2(k-1)(1-r_3)}{r_2(2k-1)(k-2)}.$$

To do so, we must set $r_3 = 1$. Next, note that the TDM line is given by the equation $f(x, y) = y - C_2^*(1 - x/C_3^*)$, and the distance between it and any point

(C_3, C_2) is given by $|f(C_3, C_2)|/\sqrt{1 + (C_2^*/C_3^*)^2}$. Hence, it is sufficient to maximize $|f(C_3(r_1, r_2, 1), C_2(r_1, r_2, 1))|$, given by

$$\frac{2\mu k(k-1)}{(2k-1)\left(k - 2 + \frac{2k^2-k}{(k-1)r_2(1-r_1)}\right)}.$$

It is clear that we must set $r_2 = 1$ and $r_1 = 0$, yielding $(\hat{C}_3, \hat{C}_2)$ as the point farthest from the TDM line, as expected.

Next, consider the line passing through $(0, C_2^*)$ and $(\hat{C}_3, \hat{C}_2)$:

$$y_1 = -\frac{3k-2}{2k-1}x_1 + \frac{\mu k(k-1)}{2k-1}, \tag{5.4}$$

and the line passing through $(\hat{C}_3, \hat{C}_2)$ and $(C_3^*, 0)$:

$$y_2 = -\frac{k(k-2) + 2(k-1)^2}{k(k-2)}x_2 + \mu(k-1). \tag{5.5}$$

It is not hard to show that for any point (C_3, C_2), (5.1) and (5.2) are satisfied. In other words, all points in the achievable capacity region fall on or below these two lines. To prove that this upper bound is tight, it remains to show that all points on lines (5.4) and (5.5) are achievable. To see this, let $r_1 = 0$ and $r_3 = 1$:

$$C_2(0, r_2, 1) = \frac{(k - (k-2)r_2)k(k-1)\mu}{k^2 + (k-1)(k + (k-2)r_2)},$$
$$C_3(0, r_2, 1) = \frac{(k-2)r_2 k(k-1)\mu}{k^2 + (k-1)(k + (k-2)r_2)}.$$

Note that any point $(C_3(0, r_2, 1), C_2(0, r_2, 1))$ is *on* line (5.4), and these two functions are continuous in $r_2 \in [0, 1]$. Similarly, letting $r_1 = 0$ and $r_2 = 1$, we have

$$C_2(0, 1, r_3) = \frac{2r_3 k(k-1)\mu}{k(k-2+2r_3) + 2(k-1)^2},$$
$$C_3(0, 1, r_3) = \frac{k(k-1)(k-2)\mu}{k(k-2+2r_3) + 2(k-1)^2}.$$

Any point $(C_3(0,1,r_3), C_2(0,1,r_3))$ is *on* line (5.5), and these two functions are continuous in $r_3 \in [0,1]$. Using these facts, we conclude that *all* points on (5.4) and (5.5) are achievable. □

Claim 5.4.3. *As $k \to \infty$, the benefit of using an alternate policy (one that lies above TDM) diminishes.*

Proof. We prove this by showing that as $k \to \infty$, the ratio of the achievable area above the TDM line, which we call $A_\triangle$ (because this area has the shape of a triangle) to the total area below the capacity region, which we call A_T, goes to zero. For $A_\triangle$, the length of the base of the triangle is simply the distance between the points $(0, C_2^*)$ and $(C_3^*, 0)$, or $\sqrt{(C_2^*)^2 + (C_3^*)^2}$. The height is given by $|f(\hat{C}_3, \hat{C}_2)|/\sqrt{1 + (C_2^*/C_3^*)^2}$. Then

$$A_\triangle = \frac{|f(\hat{C}_3, \hat{C}_2)| C_3^*}{2}.$$

Then, the area below the TDM line is given by

$$A_{TDM} = \frac{(C_2^*)^2 + (C_3^*)^2}{4}, \text{ so the total area is}$$
$$A_T = A_\triangle + A_{TDM} = \frac{2|f(\hat{C}_3, \hat{C}_2)| C_3^* + (C_2^*)^2 + (C_3^*)^2}{4}.$$

Then the ratio of the area above the TDM to the total area is

$$\frac{A_\triangle}{A_T} = \frac{2|f(\hat{C}_3, \hat{C}_2)| C_3^*}{2|f(\hat{C}_3, \hat{C}_2)| C_3^* + (C_2^*)^2 + (C_3^*)^2} = \frac{1}{1 + \frac{(C_2^*)^2 + (C_3^*)^2}{2|f(\hat{C}_3, \hat{C}_2)| C_3^*}}.$$

To prove that this ratio goes to zero with k, it suffices to show that the second term in the denominator goes to ∞. It can be shown that $((C_2^*)^2 + (C_3^*)^2)/2|f(\hat{C}_3, \hat{C}_2)|C_3^*$ simplifies to

$$\frac{39k^6 - 220k^5 + 493k^4 - 568k^3 + 362k^2 - 120k + 16}{4(6k^5 - 33k^4 + 67k^3 - 62k^2 + 26k - 4)} \xrightarrow{k \to \infty} \infty.$$

□

5.5 System with Per-Link Buffer Size Two

In a system with per-link buffer size two, there are three additional states, as shown in Figure 5.4. Our goal in this section is to show the existence of better policies than TDM, rather than to find the optimal policy. Hence, the design in Figure 5.4 does not encapsulate all possible policies: for instance, there is no r_1 parameter here, since our exhaustive search over the entire parameter space for the system with $B = 1$ revealed that r_1 is best set to zero. In addition, note that if the system is in state $(1, 1)$ and another entanglement is generated on one of the links that already has a stored qubit, the system is not allowed to use two of the qubits for a BSM. The reasoning is that since $B = 2$, there is enough space to keep the new qubit. Similarly, when the system is in state $(2, 1)$ a BSM is only allowed if (i) another entanglement is generated on one of the $k-2$ links that does not have a stored qubit, or (ii) another entanglement is generated on the link that already has two qubits stored. In the latter scenario, not performing a BSM would cause a qubit to be discarded. While this design does not grant the switch access to the full range of policies, it does enable us to find a class of policies that are more efficient than TDM.

Figure 5.5 shows capacity regions for $B = 2$ with number of links $k = 3$ and 10. We observe that policies more efficient than TDM can be found, but as the number of links grows, the advantage of such policies relative to TDM diminishes. This phenomenon mimics that of the $B = 1$ switch. Figure 5.6 shows a comparison of $B = 1$ and $B = 2$ switches for three and ten links. We observe that while there is a clear benefit to extra buffer space for a small number of users, the advantage becomes less apparent as the number of users grows. In addition, it appears that C_3 benefits more from the extra buffer space than C_2.

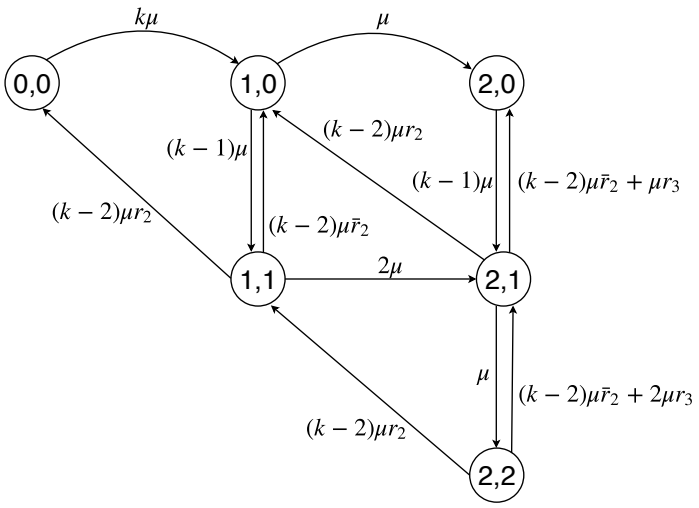
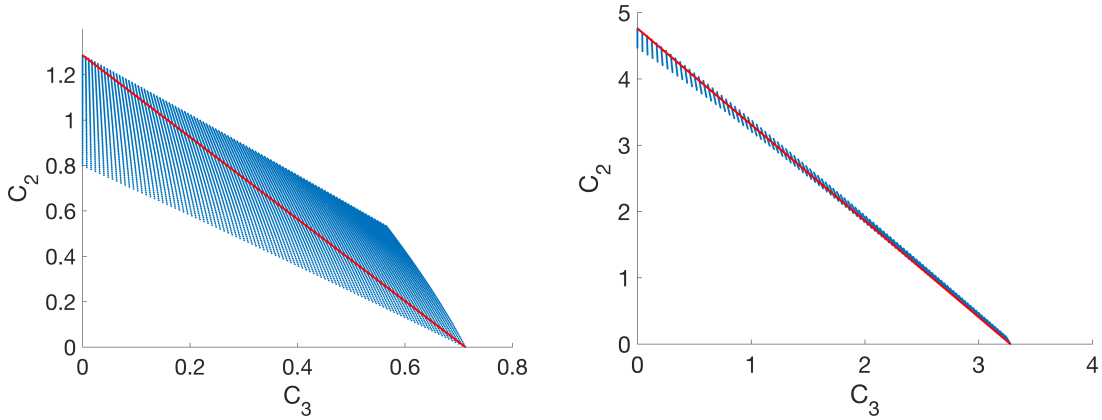
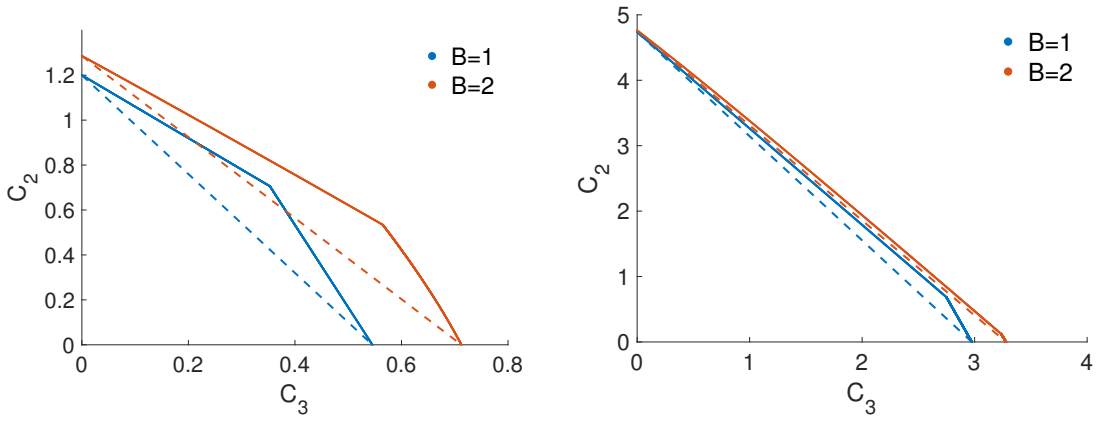


Figure 5.4: CTMC for a system with at least three links and buffer size two for each link. k is the number of links, μ is the rate of entanglement generation, and r_2 and r_3 are parameters that specify the scheduling policy.



(a) $k = 3$ (b) $k = 10$

Figure 5.5: Capacity region for per-link buffer size $B = 2$, for $k = 3$, 10 links. The red line represents the set of TDM policies.



(a) $k = 3$ (b) $k = 10$

Figure 5.6: Comparison of capacity regions for systems of buffer sizes one and two with varying number of links k, and entanglement generation rate $\mu = 1$.

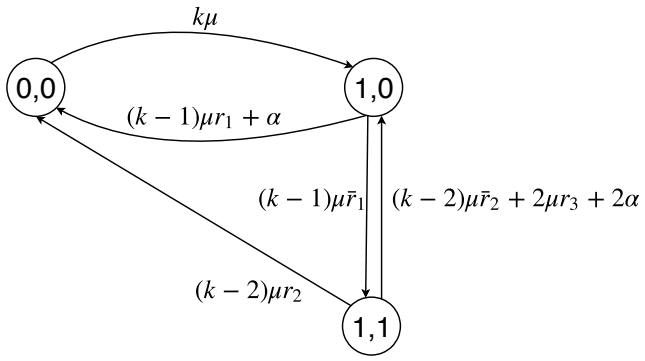


Figure 5.7: CTMC for a system with at least three links and buffer size one. k is the number of links, μ is the rate of entanglement generation, α is the decoherence rate, and r_1, r_2, r_3 are parameters that specify the scheduling policy.

5.6 Modeling Decoherence

In this section, we present a simple way to augment the model from Section 5.4 to account for the decoherence of quantum states. For switches with $B = 1$, we present both analytic and numerical results. We also augment the model from Section 5.5 to incorporate decoherence, but for switches with $B = 2$, we present only numerical results. Our decoherence model is described in Section 5.3. For $B = 1$, the resulting CTMC is illustrated in Figure 5.7.

The analysis of this model is almost identical to that of the $B = 1$ system without decoherence. As with the latter, the capacity region is bounded above by two lines:

$$y_1 = -\frac{\mu(3k-2)(\alpha+(k-2)\mu)+2\alpha^2}{\mu(k-2)((2k-1)\mu+\alpha)}x_1 + \frac{k(k-1)\mu^2}{(2k-1)\mu+\alpha},$$
$$y_2 = -\frac{2(k-1)^2\mu^2+(k\mu+\alpha)((k-2)\mu+2\alpha)}{\mu(k-2)(k\mu+\alpha)}x_2 + \frac{k(k-1)\mu^2}{k\mu+\alpha}.$$

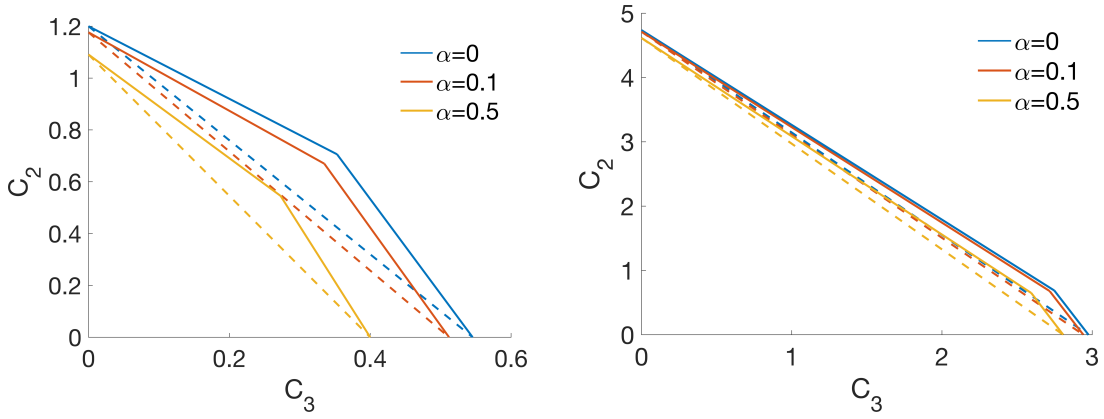
To avoid ambiguity, let C_2' and C_3' denote the bi- and tripartite capacities of a system with decoherence. As with the previous model, C_2' is maximized at $r_1 = 1$, $r_2 = r_3 = 0$; C_3' is maximized at $r_1 = r_3 = 0$, $r_2 = 1$, and the point farthest from TDM is obtained by setting $r_1 = 0$, $r_2 = r_3 = 1$. The first bounding line passes through the points $(0, C_2'(1,0,0))$ and $(C_3'(0,1,1), C_2'(0,1,1))$; and the second line passes through $(C_3'(0,1,1), C_2'(0,1,1))$ and $(C_3'(0,1,0), 0)$. Moreover, all points *on* the bounding lines are achievable, indicating that the bound is tight.

The capacities are given by

$$C_2' = (k(k-1)\mu^2\,(2(\alpha r_1+\mu r_3)+(k-2)\mu(1-r_2\bar{r}_1)))/D,$$
$$C_3' = (k\mu^3(k-1)(k-2)\bar{r}_1 r_2)/D, \text{ where}$$
$$D = (k-1)\mu\bar{r}_1((k-2)\mu r_2+k\mu)+(k\mu+(k-1)\mu r_1+\alpha)((k-2+2r_3)\mu+2\alpha).$$

Note that the denominator is quadratic in α. This causes both C_2' and C_3' to tend to zero as $\alpha \to \infty$.

Figure 5.8 shows a comparison of the capacity regions for systems with $B = 1$, for three and ten links and different decoherence rates. For all cases, μ is set to one: for qualitative results, we only need to concern ourselves with the value of α *relative* to μ. In real scenarios, we expect α to be at least one order of magnitude less than μ. From numerical results, we observe that the effect of decoherence on the capacity



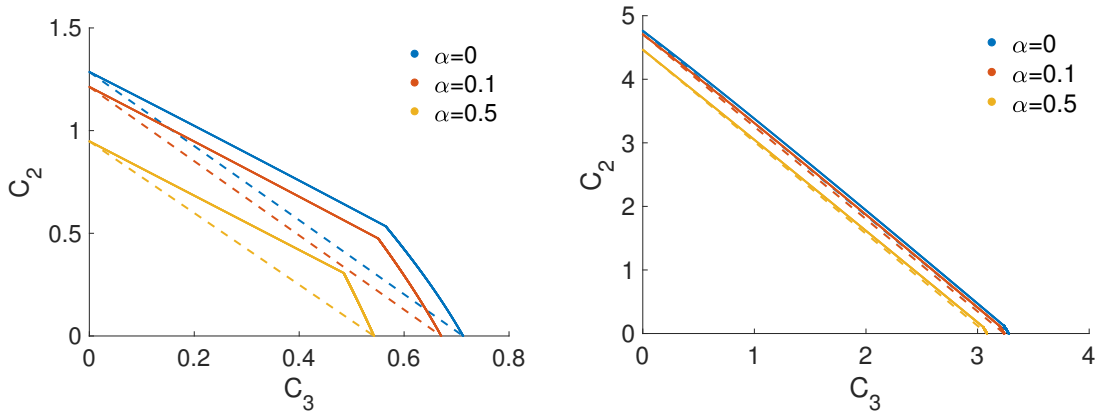
(a) $k = 3$ (b) $k = 10$

Figure 5.8: Capacity region for a system of buffer size one and varying number of links k, decoherence rates α, and entanglement generation rate $\mu = 1$. The solid lines are the upper boundaries of the capacity region, and the dashed are TDM lines.

region is not significant, especially as the number of links grows. Analysis supports this observation, since we can show that

$$\lim_{k\to\infty} \frac{C_2'}{C_2} = 1 \text{ and } \lim_{k\to\infty} \frac{C_3'}{C_3} = 1.$$

Figure 5.9 shows a comparison for systems with $B = 2$ and varying number of links



(a) $k = 3$ (b) $k = 10$

Figure 5.9: Capacity region for a system of buffer size two and varying number of links k, decoherence rates α, and entanglement generation rate $\mu = 1$. The solid lines are the upper boundaries of the capacity region, and the dashed are TDM lines.

and decoherence rates. Results are consistent with that of the case $B = 1$: the effects of decoherence on capacity are less apparent for larger k values.

5.7 Conclusion

In this work, we explored a set of policies for a quantum switch that can store up to two qubits per link and whose objective is to perform bipartite and tripartite projective measurements to distribute two and three qubit entanglement to pairs and triples of users. We presented analytical results for the case where the per-link buffer has size one. By comparing against TDM policies, we discovered that better policies in terms of achievable bipartite and tripartite capacities exist, but that as the number of links grows, the advantage of using such policies diminishes. We also compared the capacity regions for systems with different per-link buffer sizes and observed that systems with fewer links benefit more from the extra storage space than systems with a larger number of links. Finally, we modeled decoherence for both types of systems and presented analytical results for the case with per-link buffer size one. Observations and analysis showed that as the number of links increases, the effects of decoherence become less apparent on systems.

APPENDIX A

ON THE STABILITY ANALYSIS OF TCP

A.1 Non-Homogeneous Poisson Loss Simulation Framework

We introduce a method of simulating the evolution of a congestion window given $W(t)$ – *cwnd* as a function of time, and $\lambda(t)$ – loss rate as a function of time. We first describe the procedure for generating loss events given arbitrary $W(t)$ and $\lambda(t)$. We then consider a specific loss model and discuss the workarounds necessary when dealing with capacity constraints and time delays. The final result is an algorithm whose pseudocode we present in detail. Finally, we illustrate the operation of the algorithm using an example *cwnd* trajectory.

A.1.1 Generating Loss Events

We would like to generate inter-loss times given a loss rate function $\lambda(t)$. In order to do so, we apply the Inverse Transform Method on the Poisson distribution, described in the following proposition.

Proposition A.1.1. *Suppose a loss event occurs at time t_0. The time to the next loss is given by T where*

$$\int_{t_0}^{t_0+T} \lambda(t)dt = -\ln u,$$

where u is randomly generated from the uniform distribution $U(0,1)$.

Proof. Note that $\lambda(t)$ denotes a Non-Homogeneous Poisson Process, where the number of events between s and $t, N_s(t)$ has a Poisson distribution with parameter $m_s(t) = \int_s^t \lambda(\tau)d\tau$,

$$P(N_s(t) = k) = \frac{m_s(t)^k}{k!} e^{-m_s(t)}.$$

We can then write the CDF of the time from t_0 to the next loss as

$$\begin{aligned} F_{X_{t_0}}(T) &= 1 - P(N_{t_0}(T) = 0) = P(N_{t_0}(T) > 0) \\ &= 1 - \exp\Big(- \int_{t_0}^{t_0+T} \lambda(t)dt \Big). \end{aligned}$$

Note that a CDF can be seen as a random variable with uniform distribution $U(0, 1)$, and can be sampled by generating uniform random numbers (this is known as Inverse Transform Sampling). Therefore, inter-loss time samples can be generated as $T = F^{-1}_{X_{t_0}}(u)$. From the above equation we obtain

$$\int_{t_0}^{t_0+T} \lambda(t)dt = -\ln(1-u) \equiv -\ln u,$$

where the last equivalence follows from the fact that if u is uniformly distributed between 0 and 1, so is $1 - u$. □

A.1.2 Delays and Capacity Constraints

In TCP (and most other data transport protocols), the loss rate is a function of the sending rate $W(t)/\tau$ and of a probability of loss model $p(t)$:

$$\lambda(t) = \frac{W(t)p(t)}{\tau}. \tag{A.1}$$

Therefore, in order to obtain a sample of the time until next loss, the following equation can be solved for T:

$$\frac{1}{\tau} \int_{t_0}^{t_0+T} W(t)p(t)dt = -\ln(u). \tag{A.2}$$

Note that $W(t)$ and $p(t)$ are viewed from the perspective of the congestion point (*e.g.* a router) where the loss is being generated. Therefore, whenever a loss occurs, the subsequent reduction in the window size (multiplicative decrease) is not reflected in $W(t)$ until after a delay of approximately τ seconds. This is illustrated in Figure A.1, which shows an example trajectory of the *cwnd* function. Each time a loss i occurs at time l_i at a congestion point, a corresponding loss indication is reflected in $W(t)$ at time $T_i = l_i + \tau$. The caveat of using (A.2) to compute T is that $W(t)$ may have changed sometime in the time interval $[t_0, t_0 + T]$ (which can happen if a loss indication is scheduled in this interval; we call this a *pending loss indication* (PLI)). In such a case, the solution is to project the current $W(t)$ until the next loss indication, update $W(t)$ to a new function, and use this new function to generate a new loss event. Once a new loss event is generated, the process may need to repeat until we either produce a loss event that takes place before the next PLI or until we run out of PLIs.

Another complication may arise with certain probability of loss models. For example, in this work we consider the following model:

$$p(t) = \left(1 - \frac{C\tau}{W(t)}\right)^+.$$

As a consequence, $\lambda(t) = 0$ whenever $W(t) < C\tau$. This is depicted in Figure A.1, where losses only occur when $W(t) \geq C\tau$. In order to obtain an analytical solution for T during the ith loss event, we can first compute T_{BDP}, the time at which $W(t)$ reaches $C\tau$, or the bandwidth-delay product (BDP). Then, let $t_0 = \max(T_{BDP}, l_i) - T_{i-1}$, where T_{i-1} is the time of the most recent loss indication and l_i is the time of the most recent loss event at the congestion point.

Another feature of the simulation framework is the ability to generate multiple parallel flows. This feature is especially important for validating models that use a system of differential equations to characterize the behavior of congestion control

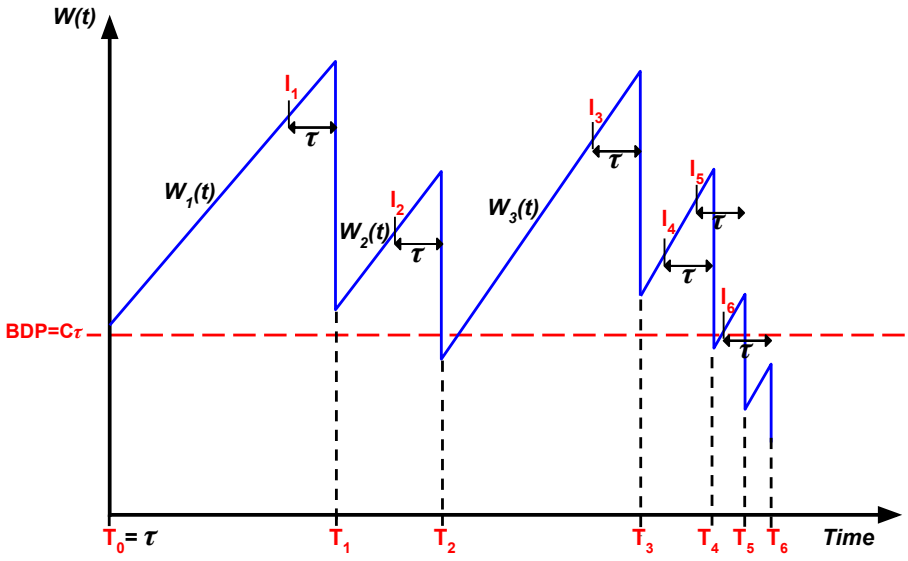


Figure A.1: Example trajectory of Reno's congestion window. l_i is the time when loss occurs at the congestion point (*e.g.* router). T_i is the time of the ith loss indication.

algorithms. The output of such models (*e.g.* *cwnd*) usually describes the behavior of the average flow in a large population of flows. Indeed, in Section 2.7, we note that the average *cwnd* size from simulation results matches closer to the steady-state value of the DE models as we increase the number of flows in the simulation.

When multiple flows are involved, T_{BDP} is the time at which the sum of their congestion windows reaches the BDP, and l_i is the time at which the most recent loss (across all flows) occurred. We must compute t_0 for each flow, which is given by

$$t_{0,f} = \max\left(T_{BDP}, l_i\right) - T_{i-1,f},$$

where $T_{i-1,f}$ is the most recent loss indication of flow f. T is then computed using the following equation:

$$\frac{1}{\tau}\sum_{f=1}^{N}\int_{t_{0,f}}^{t_{0,f}+T} W_f(t)p_f(t)dt = -\ln(u). \tag{A.3}$$

Any time a new loss event is generated, we must also choose a flow that will suffer the loss. The flow is picked based on its congestion window size at the time the loss is scheduled to occur (flows with larger windows are more susceptible to suffer a loss).

A.1.3 Pseudocode

Loss generation can be described by the pseudocode in `GeneratePoiLoss`. This function is called from the main procedure each time a loss is occurring at the congestion point in a given interval. (So, for the example in Figure A.1, `GeneratePoiLoss` would be called in the intervals containing the events l_i, $i \in \{1, \ldots, 6\}$.) The arguments of the function are as follows: $pendingLITs$ is a two-dimensional matrix whose first row is a list of pending loss indication times, and whose second row contains the corresponding flows that will suffer the losses. $LLIs$ is an array that keeps record of the last loss indication times of all flows. $GLLI$ is the most recent loss indication. T_l is the time of the most recent loss event. W_{loss} is an array containing the *cwnd* sizes of all flows immediately before their most recent loss events. $p(t)$ is a probability of loss function and τ is the round-trip time. For the example in Figure A.1, where there is only one flow, the procedure outlined in the pseudocode would do the following:

1. At time $t = 0$, a loss occurred at the congestion point (not shown in the figure), so a pending loss indication was scheduled for $T_0 = \tau$.

2. Also at the time of the loss (at $t = 0$), a new loss time was generated using `GeneratePoiLoss`. This loss time is l_1. Since l_1 occurs after the next pending loss indication (which is at T_0), the `while` loop in `GeneratePoiLoss` is triggered. We integrate the *cwnd* function from $t = 0$ to $T_0 = \tau$, compute a new W_{loss} (which is the size of the window right before T_0), and feed these values as parameters to `computeT`. The latter function computes the next loss arrival time; this is a new value of l_1. We compare this new l_1 to the next pending loss indication time (which in this case is ∞ since no other pending loss indications

function GENERATEPOILOSS(*pendingLITs*, $LLIs$, $GLLI$, T_l, W_{loss}, $p(t)$, τ)
▷ LLT: last loss time at congestion point
▷ *pendingLITs*: a list of pending loss indication times and corresponding flows
Initialization:
$GNPLI \leftarrow pendingLITs.nextLossTime$ ▷ next (global) pending loss indication time
$LF \leftarrow pendingLITs.nextFlow$ ▷ the corresponding flow of the next loss event
$T_{BDP} \leftarrow$ time when sum of *cwnd*'s reaches BDP
$t_0 \leftarrow \max(T_{BDP}, T_l)$

$lossTime \leftarrow$ COMPUTET$(LLIs, GLLI, W_{loss}, t_0, \tau, p(t))$

while $lossTime \geq GNPLI$ **do**
▷ next loss occurs after $GNPLI$, so:
▷ (1) determine duration of current congestion epoch for flow LF:
$I \leftarrow GNPLI - LLIs(f)$
▷ (2) the window function is changed at $GNPLI$, and we are looking at a new congestion epoch, so update relevant variables
$W_{loss} \leftarrow W_{LF}(I)$ ▷ get the W_{loss} value of the next congestion epoch for flow LF
$GLLI \leftarrow GNPLI$
$LLIs(LF) \leftarrow GLLI$
if $NPLI.isEmpty$ **then**
$NPLI \leftarrow \infty$
else
$GNPLI \leftarrow pendingLITs.nextLossTime$
$LF \leftarrow pendingLITs.nextFlow$
end if
▷ (3) generate a new loss event at congestion point
Recompute T_{BDP}
$t_0 \leftarrow \max(T_{BDP}, GLLI)$
$lossTime \leftarrow$ COMPUTET$(LLIs, GLLI, W_{loss}, t_0, \tau, p(t))$
end while

▷ schedule the next loss indication event
$pendingLITs.add(lossTime + \tau)$
return $(lossTime, pendingLITs)$
end function

function COMPUTET($LLIs$, $GLLI$, W_{loss}, t_0, τ, $p(t)$)
$u \leftarrow rand()$ ▷ generate a number from uniform distr.
construct $W_f(t),\ \forall f \in \{1, \ldots, N\}$ using W_{loss}
$t_{0,f} \leftarrow t_0 - LLIs(f),\ \forall f \in \{1, \ldots, N\}$
▷ to generate the next loss interval:
Use Equation (A.3) to compute T, keep only real, positive roots
$lossTime \leftarrow GLLI + t_0 + T$
end function

have been scheduled after T_0). Since $l_1 < \infty$, we exit the loop and have a new loss time of l_1 and pending loss indication $T_1 = l_1 + \tau$.

3. The main procedure iterates until it reaches the interval containing l_1, at which point `GeneratePoiLoss` is called. The latter function generates l_2, and since l_2 occurs after the next pending loss indication time (T_1), we re-generate l_2 using the same procedure as for l_1.

4. The main procedure continues until it reaches the interval containing T_1, at which point the loss indication is processed (the *cwnd* is halved and there is a new W_{loss}).

5. Loss events l_2 and l_3 and pending loss indications T_2 and T_3 are processed similarly.

6. At loss event l_4, a new loss time l_5 is generated. Since it appears before T_4, we simply schedule a pending loss at T_5 (no need to go through the `while` loop in `GeneratePoiLoss` as we did for the other losses).

7. At loss event l_5, l_6 is generated, but it occurs after the pending loss indication at T_4, which has not been processed yet. Hence, the `while` loop is triggered.

A.2 Stability Analysis of TCP CUBIC

A.2.1 Proof of Claim 2.4.4

Proof. Consider the un-expanded $\dot{x}_1$ and assume that $\tilde{p}_\tau > 0$ in the region where we are considering this function:

$$\begin{aligned}\dot{x}_1 &= \left(\Psi - x_1 - \hat{W}\right)\frac{\Psi_\tau}{\tau}\tilde{p}_\tau \\ &= \frac{c}{\tau}\left(x_2 + \hat{s} - \sqrt[3]{\frac{b(x_1+\hat{W})}{c}}\right)^3\left(c\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau}+\hat{W})}{c}}\right)^3 + x_{1_\tau} + \hat{W} - C\tau\right).\end{aligned}$$

Recall that when expanding this function about $[x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}] = \mathbf{0}$, the zero-, first-, and second-order terms are zero. The fourth-order terms are:

$$\frac{b^4x_1^4}{27c^3\hat{s}^{10}} + \frac{cx_2^3x_{1_\tau}}{\tau} - \frac{2b^3x_1^3x_2}{9c^2\hat{s}^8} + \frac{b^2x_1^2x_2^2}{3c\hat{s}^6} - \frac{bx_1x_2^2x_{1_\tau}}{\hat{s}^2\tau} - \frac{b^3x_1^3x_{1_\tau}}{27c^2\hat{s}^6\tau} + \frac{b^2x_1^2x_2x_{1_\tau}}{3c\hat{s}^4\tau}.$$

We see that there are no terms that depend on x_{2_τ} above. The terms that contain x_{1_τ} are not problematic: we can take their absolute values and replace $|x_{1_\tau}|$ by an expression that depends on $|x_1|$ and x_2^2 using Razumikhin's Theorem, to obtain an upper-bound for these terms. We conclude that these terms have power at least four in $[x_1\ x_2]$.

Next, consider the fifth-order terms in the expansion:

$$\frac{13b^4x_1^4x_2}{81c^3\hat{s}^{11}} - \frac{8b^5x_1^5}{243c^4\hat{s}^{13}} - \frac{5b^3x_1^3x_2^2}{27c^2\hat{s}^9} + \frac{b^4x_1^4x_{1_\tau}}{27c^3\hat{s}^9\tau} - \frac{2b^3x_1^3x_2x_{1_\tau}}{9c^2\hat{s}^7\tau} + \frac{b^2x_1^2x_2^2x_{1_\tau}}{3c\hat{s}^5\tau}.$$

Again, there are no terms that depend on x_{2_τ}. Terms that contain x_{2_τ} only begin to show up in the sixth-order partial derivatives evaluated at $\mathbf{0}$. However, these terms contain at most $x_{2_\tau}^3$, since taking the derivative of $\dot{x}_1$ with respect to x_{2_τ} four times yields zero. The rest of the variables in such a term is any cubic combination of x_1, x_2, and x_{1_τ}. Hence, the minimum combined power of such a term (after taking the absolute value and bounding using Razumikhin's Theorem) is $3 + 3(1/2) = 3 + 1.5 = 4.5 > 4$ in $[x_1\ x_2]$.

Finally, we can use Proposition 2.4.1 to bound the remainder. For some positive constant M_1,

$$|R_{\mathbf{0},6}| \leq \frac{M_1}{7!}(|x_1| + |x_2| + |x_{1_\tau}| + |x_{2_\tau}|)^7.$$

After substituting the expressions for the upper-bounds of $|x_{1_\tau}|$ and $|x_{2_\tau}|$ using Razumikhin's Theorem, the lowest-order term will have $|x_1|^{7/2} = |x_1|^{3.5}$. □

A.2.2 Proof of Claim 2.4.5

Proof. Consider the un-expanded $\dot{x}_2$ and assume that $\tilde{p}_\tau > 0$ in the region where we are considering this function:

$$\begin{aligned}\dot{x}_2 &= 1 - (x_2 + \hat{s})\frac{\Psi_\tau}{\tau}\tilde{p}_\tau \\ &= 1 - \frac{(x_2 + \hat{s})}{\tau}\left(c\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)^3 + x_{1_\tau} + \hat{W} - C\tau\right).\end{aligned}$$

Recall that when expanding this function about $[x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}] = \mathbf{0}$, the zero-order term is zero. The second-order terms do not depend on x_{2_τ}. To see this, let $g = \dot{x}_2$ and consider all second-order partial derivatives of g with respect to x_{2_τ}:

$$\begin{aligned}g_{x_{2_\tau}x_{2_\tau}} &= -\frac{(x_2 + \hat{s})}{\tau}(6c)\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right) \implies g_{x_{2_\tau}x_{2_\tau}}(\mathbf{0}) = 0, \\ g_{x_2x_{2_\tau}} &= -\frac{3c}{\tau}\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)^2 \implies g_{x_2x_{2_\tau}}(\mathbf{0}) = 0, \\ g_{x_{1_\tau}x_{2_\tau}} &= \frac{(x_2 + \hat{s})}{\tau}(2b)\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)\left(\frac{b(x_{1_\tau} + \hat{W})}{c}\right)^{-2/3} \\ &\implies g_{x_{1_\tau}x_{2_\tau}}(\mathbf{0}) = 0, \\ g_{x_1x_{2_\tau}} &= 0.\end{aligned}$$

Therefore, the second-order terms have combined powers of at least two in $[x_1\ x_2]$. Next, using Proposition 2.4.1, we can bound the remainder: for some positive constant M_2,

$$|R_{\mathbf{0},2}| \leq \frac{M_2}{3!}(|x_1| + |x_2| + |x_{1_\tau}| + |x_{2_\tau}|)^3.$$

The lowest-order term above is $|x_1|^{3/2} = |x_1|^{1.5}$. □

A.2.3 Proof of Claim 2.4.6

Proof. To prove our claim, we will show that for every $\epsilon > 0$, there exists a $\delta > 0$ so that whenever $0 < \sqrt{x_1^2 + x_2^2} < \delta$, $|f(x_1, x_2)| < \epsilon$. The higher-order terms are a sum of terms that have format $c_0 x_1^{c_1} x_2^{c_2}$, where c_0 is either a constant or a function of $x_{1_{\theta+\tau}}$ and $x_{2_{\theta+\tau}}$, as in the case of the higher-order terms that arise from the $x_2^3 \dot{x}_{1_\theta} \tau$ term of $\dot{V}$. We showed that in all cases, $|c_0|$ is bounded above by a positive constant. In addition, we showed that $c_1 + c_2 \geq 4.5$. Let n be the number of higher-order terms (note: n is finite). Then we can write $|f(x_1, x_2)|$ as follows:

$$|f(x_1, x_2)| = \frac{1}{||\mathbf{x}||_2^4} \left| \sum_{i=1}^{n} c_0^{(i)} x_1^{c_1^{(i)}} x_2^{c_2^{(i)}} \right|,$$

where the superscript (i) corresponds to the ith higher-order term. Clearly,

$$|f(x_1, x_2)| \leq \frac{1}{||\mathbf{x}||_2^4} \sum_{i=1}^{n} \left| c_0^{(i)} x_1^{c_1^{(i)}} x_2^{c_2^{(i)}} \right|.$$

Consider any term $|c_0 x_1^{c_1} x_2^{c_2}|$. Factor out any combination $|x_1|^a |x_2|^b$ such that $a + b = 4$. We know that

$$|x_1|, |x_2| \leq \sqrt{x_1^2 + x_2^2}.$$

Then

$$|x_1|^a |x_2|^b \leq \left(\sqrt{x_1^2 + x_2^2} \right)^4 = ||\mathbf{x}||_2^4,$$

$$|c_0 x_1^{c_1} x_2^{c_2}| \leq |c_0 g(x_1, x_2)| ||\mathbf{x}||_2^4$$

where $g(x_1, x_2)$ is defined s.t. $g(x_1, x_2) x_1^a x_2^b = x_1^{c_1} x_2^{c_2}$. Hence,

$$|f(x_1, x_2)| \leq \frac{1}{||\mathbf{x}||_2^4} \sum_{i=1}^{n} \left| c_0^{(i)} g^{(i)}(x_1, x_2) \right| ||\mathbf{x}||_2^4 = \sum_{i=1}^{n} \left| c_0^{(i)} g^{(i)}(x_1, x_2) \right|.$$

Each function $g(x_1, x_2)$ necessarily has the form

$$g(x_1, x_2) = x_1^{l_1} x_2^{l_2},\ l_1, l_2 \geq 0,\ l_1 + l_2 \geq \frac{1}{2}.$$

Hence, we can bound the absolute value of each of these functions by a function of δ.

$$\begin{aligned}
|x_{1/2}| &\leq \sqrt{x_1^2 + x_2^2} < \delta, \\
|x_{1/2}|^{l_{1/2}} &\leq \left(\sqrt{x_1^2 + x_2^2}\right)^{l_{1/2}} < \delta^{l_{1/2}}, \\
|g(x_1, x_2)| &\leq \left(\sqrt{x_1^2 + x_2^2}\right)^{l_1 + l_2} < \delta^{l_1 + l_2}.
\end{aligned}$$

This gives us

$$|f(x_1, x_2)| \leq \sum_{i=1}^{n} \left| c_0^{(i)} \right| \delta^{l_1^{(i)} + l_2^{(i)}}.$$

We would like the sum above to be less than a given $\epsilon > 0$. We can always find a $\delta > 0$ small enough to make this happen. □

A.3 Limiting behavior of H-TCP's fixed point

First, let us show that for H-TCP, $\Delta_0/Q = \mathcal{O}((C^2\tau^4)^{1/3})$. This is because

$$\frac{\Delta_0}{Q} = \frac{2^{1/3}\Delta_0}{(\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}} = \frac{(\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}}{2^{1/3}} < \Delta_1^{1/3} = \mathcal{O}(C^{2/3}\tau^{4/3}).$$

Next,

$$\begin{aligned}
\frac{\Delta_0}{Q} + Q &= \frac{(\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3} + (\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}}{2^{1/3}} \\
&< \frac{2(\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}}{2^{1/3}} = \mathcal{O}((C^2\tau^4)^{1/3}).
\end{aligned}$$

We now examine the limiting behavior of q'/S. Letting $c_0 = \Gamma + (57^3 + 228(105))/8$,

$$\begin{aligned}\frac{q'}{S} &= \frac{6C\tau^2 - c_0}{\frac{1}{2}\sqrt{\frac{2}{3}p' + \frac{1}{3}\left(Q + \frac{\Delta_0}{Q}\right)}} < \frac{12\sqrt{3}C\tau^2}{\left(Q + \frac{\Delta_0}{Q}\right)^{1/2}} \\ &= \frac{12\sqrt{3}\sqrt[6]{2}C\tau^2}{\left((\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3} + (\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}\right)^{1/2}} \\ &< \frac{12\sqrt{3}\sqrt[6]{2}C\tau^2}{\Delta_1^{1/6}} < \frac{25C\tau^2}{(C^2\tau^4)^{1/6}},\end{aligned}$$

where the last inequality holds for large C since the coefficient of $C^2\tau^4$ in Δ_1 is 972. It is now easy to see that $q'/S = \mathcal{O}((C^2\tau^4)^{1/3})$. Finally, it remains to show that (2.35) is real and positive for large C and $C \gg \tau$. Clearly, under these assumptions, $S > 57/4$. It then remains to show that the quantity inside the square root is positive, or

$$2p' + q'/S > 4S^2 = \frac{2}{3}p' + \frac{1}{3}\left(Q + \frac{\Delta_0}{Q}\right).$$

Defining $y \coloneqq Q + \Delta_0/Q$, multiplying both sides by S and simplifying yields

$$6\sqrt{3}q' > (y - 4p')(2p' + y)^{1/2}.$$

Both sides of the inequality are positive for large C, so we may square them to obtain, after simplifying,

$$108(q')^2 > y^3 - 6p'y^2 + 32(p')^3.$$

For large C, it is easy to see that $y^3 - 6p'y^2 + 32(p')^3 < y^3$, so it is sufficient to prove that $108(q')^2 > y^3$, or

$$216(q')^2 > \left[(\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3} + (\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3})^{1/3}\right]^3.$$

We first focus on the right-hand side of this inequality. It is easy to see that

$$\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3} < 2\Delta_1.$$

For large C and $C \gg \tau$, it is also true that $\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3} < \Delta_1/8$. To see that this is true, note that this claim is equivalent to that of

$$7\Delta_1/8 < \sqrt{\Delta_1^2 - 4\Delta_0^3}, \quad \text{or} \quad 15\Delta_1^2/64 > 4\Delta_0^3,$$

and recall that Δ_1 grows with $(C\tau^2)^2$ while Δ_0 grows with $C\tau^2$. Hence, it is sufficient for us to prove that

$$216(q')^2 > \left[(\Delta_1/8)^{1/3} + (2\Delta_1)^{1/3}\right]^3 = \Delta_1(2^{1/3} + 1/2)^3.$$

Noting that $(2^{1/3} + 1/2)^3 < 6$, it is sufficient to show that $36(q')^2 > \Delta_1$, or, after expanding both sides,

$$1296(C\tau^2)^2 - 432c_0C\tau^2 + 36c_0^2 > 972(C\tau^2)^2 - 348574.3C\tau^2 - 571698\tau^2 + 2070623.3,$$

which easily holds for large C and $C \gg \tau$.

APPENDIX B

ON THE STOCHASTIC ANALYSIS OF A QUANTUM ENTANGLEMENT DISTRIBUTION SWITCH

B.1 Capacity for Heterogeneous Systems with $B = \infty$

B.1.1 Proof of the last equality in Eq. (3.1)

From the first part of this equation, we have

$$
\begin{aligned}
C &= q\sum_{l=1}^{k}\sum_{j=1}^{\infty}\pi_l^{(j)}(\gamma-\mu_l) = q\sum_{l=1}^{k}\sum_{j=1}^{\infty}\pi_0\rho_l^j(\gamma-\mu_l)\\
&= q\pi_0\sum_{l=1}^{k}\frac{(\gamma-\mu_l)\rho_l}{1-\rho_l} = q\pi_0\sum_{l=1}^{k}\left(\frac{\gamma}{2}\frac{\rho_l}{1-\rho_l}+\left(\frac{\gamma}{2}-\mu_l\right)\frac{\rho_l}{1-\rho_l}\right)\\
&= q\pi_0\sum_{l=1}^{k}\left(\frac{\gamma}{2}\frac{\rho_l}{1-\rho_l}+\left(\frac{\gamma-2\mu_l}{2}\right)\frac{\mu_l(\gamma-\mu_l)}{(\gamma-\mu_l)(\gamma-2\mu_l)}\right)\\
&= q\pi_0\sum_{l=1}^{k}\left(\frac{\gamma}{2}\frac{\rho_l}{1-\rho_l}+\frac{\mu_l}{2}\right) = q\pi_0\frac{\gamma}{2}\left(\sum_{l=1}^{k}\frac{\rho_l}{1-\rho_l}+1\right) = \frac{q\gamma}{2}.
\end{aligned}
$$

B.1.2 Proof that $C_l = q\mu_l$

Letting $B \to \infty$ in Eq. (3.3),

$$
\begin{aligned}
C_l &= q\pi_0\left((\gamma-\mu_l)\frac{\rho_l}{1-\rho_l}+\mu_l\sum_{\substack{m=1,\\ m\neq l}}^{k}\frac{\rho_m}{1-\rho_m}\right)\\
&= q\pi_0\mu_l\left(\frac{1}{1-\rho_l}+\sum_{\substack{m=1,\\ m\neq l}}^{k}\frac{\rho_m}{1-\rho_m}+\frac{\rho_l}{1-\rho_l}-\frac{\rho_l}{1-\rho_l}\right)\\
&= q\pi_0\mu_l\left(1+\sum_{m=1}^{k}\frac{\rho_m}{1-\rho_m}\right) = q\mu_l.
\end{aligned}
$$

B.2 Decoherence

B.2.1 Homogeneous, Infinite Buffer

For this system, the balance equations are as follows:

$$\begin{aligned}
&\pi_0 k\mu = \pi_1(\alpha + (k-1)\mu),\\
&\pi_{i-1}\mu = \pi_i(i\alpha + (k-1)\mu), \quad i = 2, 3, \ldots,\\
&\sum_{i=0}^{\infty} \pi_i = 1.
\end{aligned}$$

Solving for the stationary distribution, we have:

$$\begin{aligned}
\pi_1 &= \frac{k\mu}{(k-1)\mu + \alpha}\pi_0,\\
\pi_2 &= \frac{\mu\pi_1}{(k-1)\mu + 2\alpha} = \frac{k\mu^2\pi_0}{((k-1)\mu + 2\alpha)((k-1)\mu + \alpha)},
\end{aligned}$$

and so on. In general, for $i = 1, 2, \ldots$ we can write

$$\pi_i = \frac{\pi_0 k\mu^i}{\prod_{j=1}^{i}((k-1)\mu + j\alpha)} = \pi_0 k \prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)}.$$

Using the normalizing condition, we have

$$\begin{aligned}
&\pi_0 + k\pi_0 \sum_{i=1}^{\infty}\prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)} = 1, \quad \text{so that}\\
&\pi_0 = \left(1 + k\sum_{i=1}^{\infty}\prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)}\right)^{-1}.
\end{aligned}$$

The capacity and $E[Q]$ can be computed numerically using the following formulas:

$$\begin{aligned}
C &= \sum_{i=1}^{\infty} \pi_i(k-1)\mu = (k-1)\mu(1-\pi_0),\\
E[Q] &= \sum_{i=1}^{\infty} i\pi_i = \pi_0 k \sum_{i=1}^{\infty} i \prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)}.
\end{aligned}$$

B.2.2 Homogeneous, Finite Buffer

The derivations are very similar to the previous case, with the only difference being that the balance equations are truncated at state $i = B$. The resulting expressions are almost identical to those above, with the exception of i being in $\{1, \ldots, B\}$ instead of $\{1, 2, \ldots\}$:

$$\pi_0 = \left(1 + k \sum_{i=1}^{B} \prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)}\right)^{-1},$$
$$C = \sum_{i=1}^{B} \pi_i (k-1)\mu = (k-1)\mu(1-\pi_0),$$
$$E[Q] = \sum_{i=1}^{B} i\pi_i = \pi_0 k \sum_{i=1}^{B} i \prod_{j=1}^{i} \frac{\mu}{((k-1)\mu + j\alpha)}.$$

B.2.3 Heterogeneous, Infinite Buffer

The balance equations are:

$$\pi_0 \mu_l = \pi_l^{(1)}(\gamma - \mu_l + \alpha),\ l \in \{1, \ldots, k\},$$
$$\pi_l^{(j-1)} \mu_l = \pi_l^{(j)}(\gamma - \mu_l + j\alpha),\ l \in \{1, \ldots, k\},\ j \in \{2, 3, \ldots\},$$
$$\pi_0 + \sum_{l=1}^{k} \sum_{j=1}^{\infty} \pi_l^{(j)} = 1.$$

For $j = 1, 2, \ldots$, we can write

$$\pi_l^{(j)} = \pi_0 \prod_{i=1}^{j} \frac{\mu_l}{\gamma - \mu_l + i\alpha}.$$

Using the normalizing condition, we obtain

$$\pi_0 = \left(1 + \sum_{l=1}^{k} \sum_{j=1}^{\infty} \prod_{i=1}^{j} \frac{\mu_l}{\gamma - \mu_l + i\alpha}\right)^{-1}.$$

The capacity and $E[Q]$ can be computed numerically using

$$\begin{aligned} C =& \sum_{l=1}^{k}\sum_{j=1}^{\infty}\pi_l^{(j)}(\gamma-\mu_l) = \pi_0\sum_{l=1}^{k}\sum_{j=1}^{\infty}(\gamma-\mu_l)\prod_{i=1}^{j}\frac{\mu_l}{\gamma-\mu_l+i\alpha}, \\ E[Q] =& \sum_{j=1}^{\infty} jP(Q=j) = \sum_{j=1}^{\infty} j\sum_{l=1}^{k}\pi_l^{(j)} \\ =& \pi_0\sum_{j=1}^{\infty} j\sum_{l=1}^{k}\prod_{i=1}^{j}\frac{\mu_l}{\gamma-\mu_l+i\alpha}. \end{aligned}$$

B.2.4 Heterogeneous, Finite Buffer

The derivations are similar to the previous case, with the only difference being that j is now in $\{1,\dots,B\}$ instead of in $\{1,2,\dots\}$. The resulting relevant expressions are:

$$\begin{aligned} \pi_0 &= \left(1+\sum_{l=1}^{k}\sum_{j=1}^{B}\prod_{i=1}^{j}\frac{\mu_l}{\gamma-\mu_l+i\alpha}\right)^{-1}, \\ C &= \pi_0\sum_{l=1}^{k}\sum_{j=1}^{B}(\gamma-\mu_l)\prod_{i=1}^{j}\frac{\mu_l}{\gamma-\mu_l+i\alpha}, \\ E[Q] &= \pi_0\sum_{j=1}^{B} j\sum_{l=1}^{k}\prod_{i=1}^{j}\frac{\mu_l}{\gamma-\mu_l+i\alpha}. \end{aligned}$$

□

APPENDIX C

ON THE EXACT ANALYSIS OF AN IDEALIZED QUANTUM SWITCH

C.1 Stationary Distribution

C.1.1 Proof of Eq. (4.5)

Introducing the value of $\pi_i = \beta^{k-1}\pi_1$ into Eq. (4.3) yields

$$\beta^{i-1}\pi_1 = \beta^{i-2}\pi_1 P_f + \beta^{i-1}\pi_1 P_s + \beta^{i-1}\pi_1 \sum_{j=1}^{k-1} \beta^j P_{(j)}$$

or equivalently

$$\begin{aligned}
\beta &= P_f + \beta P_s + \beta \sum_{j=1}^{k-1} \beta^j P_{(j)} \\
&= \beta(\overline{p}^k + (k-1)p^2\overline{p}^{k-2}) + \beta \sum_{j=1}^{k-1} \binom{k-1}{j} (\beta p)^j \overline{p}^{k-j} \\
&\quad + \frac{1}{\beta} \sum_{j=1}^{k-2} \binom{k-1}{j+1} (\beta p)^{j+2} \overline{p}^{k-j-2} + p\overline{p}^{k-1}. \qquad \text{(C.1)}
\end{aligned}$$

With $\sum_{j=1}^{k-1} \binom{k-1}{j} (\beta p)^j \overline{p}^{k-j} = \overline{p}\left((\beta p + \overline{p})^{k-1} - \overline{p}^{k-1}\right)$ and

$$\sum_{j=1}^{k-2}\binom{k-1}{j+1}(\beta p)^{j+2}\overline{p}^{k-j-2} = \sum_{i=2}^{k-1}\binom{k-1}{i}(\beta p)^{i+1}\overline{p}^{k-1-i}$$
$$= \beta p\left((\beta p+\overline{p})^{k-1} - (k-1)\beta p\overline{p}^{k-2} - \overline{p}^{k-1}\right),$$

Eq. (C.1) becomes

$$\begin{aligned}
\beta &= \beta(\overline{p}^k + (k-1)p^2\overline{p}^{k-2}) + \beta\overline{p}\left((\beta p+\overline{p})^{k-1} - \overline{p}^{k-1}\right) \\
&\quad + p\left((\beta p+\overline{p})^{k-1} - (k-1)\beta p\overline{p}^{k-2} - \overline{p}^{k-1}\right) + p\overline{p}^{k-1} \\
&= p\overline{p}^{k-1} + \beta\overline{p}^k + \beta(k-1)p^2\overline{p}^{k-2} + \beta\overline{p}(\beta p+\overline{p})^{k-1} \\
&\quad - \beta\overline{p}^k + p(\beta p+\overline{p})^{k-1} - \beta(k-1)p^2\overline{p}^{k-2} - p\overline{p}^{k-1} \\
&= (\beta p+\overline{p})^{k-1}(p+\beta\overline{p}).
\end{aligned}$$

Hence, β satisfies the equation $f(\beta) = 0$ with

$$f(\beta) \coloneqq (\beta p+\overline{p})^{k-1}(p+\beta\overline{p}) - \beta. \qquad \square$$

C.1.2 Proof that Eq. (4.5) has a unique solution in $(0,1)$

Let us first consider the case $k = 2$. We have

$$\begin{aligned}
f(\beta) &= (\beta p+\overline{p})(p+\beta\overline{p}) - \beta \\
&= \beta p^2 + \beta^2 p\overline{p} + p\overline{p} + \beta\overline{p}^2 - \beta \\
&= \beta^2 p\overline{p} - 2\beta p\overline{p} + p\overline{p}.
\end{aligned}$$

Setting this expression equal to zero yields

$$\beta^2 - 2\beta + 1 = 0,$$

which has a unique solution $\beta = 1$. Since this is outside the interval $(0, 1)$, we conclude that the DTMC is not stable when $k = 2$. Now, consider the case $k \geq 3$. We have

$$f'(\beta) = (k-1)p(\beta p + \overline{p})^{k-2}(p + \beta\overline{p}) + \overline{p}(\beta p + \overline{p})^{k-1} - 1$$

and $f''(\beta)$ is given by

$$\begin{aligned}&(k-1)(k-2)p^2(\beta p + \overline{p})^{k-3}(p + \beta\overline{p}) + 2(k-1)p\overline{p}(\beta p + \overline{p})^{k-2} \\ &\quad = (k-1)p(\beta p + \overline{p})^{k-3}\left[(k-2)p + 2(k-1)\overline{p}(\beta p + \overline{p})\right] > 0.\end{aligned}$$

This shows that the mapping $\beta \to f'(\beta)$ is strictly increasing in $[0, 1]$. On the other hand,

$$f'(0) = (k-1)p^2\overline{p}^{k-2} + \overline{p}^k - 1$$

and $f'(1) = (k-1)p + \overline{p} - 1 = (k-2)p > 0$. Let us show that $f'(0) < 0$. Define $g(p) = (k-1)p^2\overline{p}^{k-2} + \overline{p}^k - 1 = f'(0)$. We find

$$g'(p) = -\overline{p}^{k-3}(p^2k^2 + 2p(1-2k) + k).$$

Define $h(p) = p^2k^2 + 2p(1-2k) + k$ so that $g'(p) = -\overline{p}^{k-3}h(p)$. We have $h'(p) = 2(pk^2 + 1 - 2k)$, which vanishes for $p = p_0 \coloneqq (2k-1)/k^2$. Also, $h''(p) = 2k^2 > 0$. We deduce from this that $h(p)$ decreases in $[0, p_0)$ and increases in $(p_0, 1]$. Therefore, $h(p)$ is minimized in $[0, 1]$ for $p = p_0$. We have $h(p_0) = (-(2k-1)^2 + k^3)/k^2$ which is easily seen to be strictly positive for all $k \geq 3$. This shows that $h(p) > 0$ for $p \in [0, 1]$, which implies that $g'(p) < 0$ for $p \in [0, 1]$, so that $g(p) < g(0) = 0$ for $p \in (0, 1]$ and, finally, $f'(0) < 0$.

From $f'(0) < 0$, $f'(1) > 0$ and the fact that the continuous mapping $\beta \to f'(\beta)$ is strictly increasing in $[0, 1]$, we deduce that there exists $\beta_0 \in (0, 1)$ such that $f'(\beta) < 0$

for $\beta \in [0, \beta_0)$, $f'(\beta_0) = 0$ and $f'(\beta) > 0$ for $\beta \in (\beta_0, 1]$. This in turn shows that $\beta \to f(\beta)$ is strictly decreasing in $[0, \beta_0)$ and strictly increasing in $(\beta_0, 1]$. But since $f(0) > 0$ and $f(1) = 0$, this implies that f has a unique zero in $(0, 1)$. This zero is actually located in $(0, \beta_0)$. □

C.1.3 Equivalence of Eqs (4.1) and (4.2)

We start by rearranging (4.1):

$$\begin{aligned}
\sum_{i=0}^{k-1} \pi_i P_{i,0} &= \pi_0, \\
\sum_{i=1}^{k-1} \pi_i P_{i,0} &= \pi_0 (1 - P_{0,0}), \\
\pi_1 \sum_{i=1}^{k-1} \beta^{i-1} P_{i,0} &= \pi_0 P_{0,1}, \\
\sum_{i=1}^{k-1} \beta^{i} P_{i,0} &= \frac{\beta \pi_0}{\pi_1} P_{0,1}.
\end{aligned}$$

Then, we rearrange (4.2) in a similar fashion:

$$\begin{aligned}
\sum_{i=0}^{k} \pi_i P_{i,1} &= \pi_1, \\
\pi_0 P_{0,1} + \pi_1 \sum_{i=1}^{k} \beta^{i-1} P_{i,1} &= \pi_1, \\
\pi_0 P_{0,1} &= \pi_1 \left(1 - \frac{1}{\beta} \sum_{i=1}^{k} \beta^i P_{i,1} \right), \\
\frac{\beta \pi_0}{\pi_1} P_{0,1} &= \beta - \sum_{i=1}^{k} \beta^i P_{i,1}.
\end{aligned}$$

Hence, to show that one of (4.1) and (4.2) is redundant, it suffices to show that

$$\sum_{i=1}^{k-1} \beta^i P_{i,0} = \beta - \sum_{i=1}^{k} \beta^i P_{i,1}, \tag{C.2}$$

or equivalently,

$$\sum_{i=1}^{k-1} \beta^i (P_{i,0} + P_{i,1}) + \beta^k P_{k,1} = \beta. \tag{C.3}$$

Before we continue, we derive a few useful expressions. The first is as follows:

$$P_e(i, k-1) + P_o(i, k-1) = \sum_{j=i}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j}.$$

Next, we have

$$P_e(i, k-1) - P_o(i, k-1) = \sum_{j=i}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} (-1)^j.$$

Finally,

$$P_o(i, k-1) - P_e(i, k-1) = -\sum_{j=i}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} (-1)^j.$$

Now, consider the left side of Eq. (C.2): $\sum_{i=1}^{k-1} \beta^i P_{i,0}$ is equal to

$$
\begin{aligned}
&\sum_{i=1}^{k-1} \beta^i \left[\left(\frac{1+(-1)^i}{2} \right) (\overline{p} P_e(i,k-1) + p P_o(i+1,k-1)) \right. \\
&\quad \left. + \left(\frac{1-(-1)^i}{2} \right) (\overline{p} P_o(i,k-1) + p P_e(i+1,k-1)) \right] \\
&= \frac{\overline{p}}{2} \sum_{i=1}^{k-1} \beta^i \left[\sum_{j=i}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} (1 + (-1)^i (-1)^j) \right] \\
&\quad + \frac{p}{2} \sum_{i=1}^{k-2} \beta^i \left[\sum_{j=i+1}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} (1 - (-1)^i (-1)^j) \right] \\
&= \frac{\overline{p}}{2} \sum_{j=1}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} \sum_{i=1}^{j} \beta^i (1 + (-1)^i (-1)^j) \\
&\quad + \frac{p}{2} \sum_{j=2}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} \sum_{i=1}^{j-1} \beta^i (1 - (-1)^i (-1)^j) \\
&= \frac{1}{2} \left[\frac{\beta}{1-\beta} - \frac{2\beta}{1-\beta^2} (p\beta + \overline{p})^{k-1} (p + \overline{p}\beta) - (\overline{p} - p)^k \frac{\beta}{1+\beta} \right].
\end{aligned}
$$

Next, we look at $\sum_{i=1}^{k-1} \beta^i P_{i,1}$, which is equal to

$$
\begin{aligned}
&\sum_{i=1}^{k-1} \beta^i \left[\left(\frac{1+(-1)^i}{2} \right) (\overline{p} P_o(i-1,k-1) + p P_e(i,k-1)) \right. \\
&\quad \left. + \left(\frac{1-(-1)^i}{2} \right) (\overline{p} P_e(i-1,k-1) + p P_o(i,k-1)) \right] \\
&= \frac{\overline{p}}{2} \sum_{i=1}^{k-1} \beta^i \left[\sum_{j=i-1}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} \left(1 - (-1)^i (-1)^j\right) \right] \\
&\quad + \frac{p}{2} \sum_{i=1}^{k-1} \beta^i \left[\sum_{j=i}^{k-1} \binom{k-1}{j} p^j \overline{p}^{k-1-j} \left(1 + (-1)^i (-1)^j\right) \right] \\
&= \beta\overline{p}((\beta p + \overline{p})^{k-1} - (\beta p)^{k-1}) \\
&\quad + \frac{1}{2} \left(\frac{\beta}{1-\beta} - \frac{2\beta}{1-\beta^2} (p\beta + \overline{p})^k + \frac{\beta}{1+\beta} (\overline{p} - p)^k \right).
\end{aligned}
$$

Summing these two expressions, we obtain $\sum_{i=1}^{k-1} \beta^i (P_{i,0} + P_{i,1})$,

$$
\frac{\beta}{1-\beta} - \frac{\beta}{1-\beta} (p\beta + \overline{p})^{k-1} (p + \overline{p}\beta) - \beta\overline{p}(\beta p)^{k-1}.
$$

Next, we compute

$$\begin{aligned}
P_{k,1} &= \left(\frac{1+(-1)^k}{2}\right)\overline{p}P_o(k-1,k-1) + \left(\frac{1-(-1)^k}{2}\right)\overline{p}P_e(k-1,k-1) \\
&= \overline{p}\left(\left(\frac{1+(-1)^k}{2}\right)\left(\frac{1-(-1)^{k-1}}{2}\right)p^{k-1}\right. \\
&\quad \left. + \left(\frac{1-(-1)^k}{2}\right)\left(\frac{1+(-1)^{k-1}}{2}\right)p^{k-1}\right) = \overline{p}p^{k-1}.
\end{aligned}$$

Finally, the left side of Eq. (C.3) becomes

$$\begin{aligned}
&\sum_{i=1}^{k-1}\beta^i(P_{i,0}+P_{i,1}) + \beta^k P_{k,1} \\
&\quad = \frac{\beta}{1-\beta} - \frac{\beta}{1-\beta}(p\beta+\overline{p})^{k-1}(p+\overline{p}\beta) - \overline{p}\beta^k p^{k-1} + \beta^k\overline{p}p^{k-1} \\
&\quad = \frac{\beta}{1-\beta} - \frac{\beta}{1-\beta}(p\beta+\overline{p})^{k-1}(p+\overline{p}\beta).
\end{aligned}$$

Recall from (C.3) that the expression above must equal to β. Using Eq. (4.5), we know that

$$(p\beta+\overline{p})^{k-1}(p+\overline{p}\beta) = \beta,$$

and therefore,

$$\sum_{i=1}^{k-1}\beta^i(P_{i,0}+P_{i,1}) + \beta^k P_{k,1} = \frac{\beta}{1-\beta} - \frac{\beta^2}{1-\beta} = \beta.$$

□

C.2 Proof of Capacity

For simplicity, let us first derive C with the assumption that $q = 1$. Since q simply scales the capacity, we will multiply the resulting expression by q at the end. Consider

the first term of Eq. (4.10):

$$\begin{aligned}
\sum_{i=0}^{k-2}\pi_i\sum_{j=0}^{i} j\binom{k-1}{j}p^j\overline{p}^{k-1-j} &= \sum_{j=1}^{k-2} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\sum_{i=j}^{k-2}\pi_i \\
&= \sum_{j=1}^{k-2} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\pi_1\sum_{i=j}^{k-2}\beta^{i-1} \\
&= \frac{\pi_1}{\beta}\sum_{j=1}^{k-2} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\left(\sum_{i=0}^{k-2}\beta^i-\sum_{i=0}^{j-1}\beta^i\right) \\
&= \frac{\pi_1}{\beta}\sum_{j=1}^{k-2} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\left(\frac{1-\beta^{k-1}}{1-\beta}-\frac{1-\beta^j}{1-\beta}\right) \\
&= \frac{\pi_1}{\beta}\sum_{j=1}^{k-2} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\left(\frac{\beta^j-\beta^{k-1}}{1-\beta}\right) \\
&= \frac{\pi_1}{\beta}\sum_{j=1}^{k-1} j\binom{k-1}{j}p^j\overline{p}^{k-1-j}\left(\frac{\beta^j-\beta^{k-1}}{1-\beta}\right) \\
&= \frac{\pi_1}{\beta(1-\beta)}\left((k-1)(\beta p+\overline{p})^{k-2}\beta p-\beta^{k-1}(k-1)p\right) \\
&= \frac{\pi_1(k-1)p}{\beta(1-\beta)}\left((\beta p+\overline{p})^{k-2}\beta-\beta^{k-1}\right).
\end{aligned}$$

Next, keeping in mind that $k-1\geq 2$, the last term of Eq. (4.10) is

$$\begin{aligned}
(k-1)p\sum_{i=k-1}^{\infty}\pi_i &= \frac{(k-1)p\pi_1}{\beta}\sum_{i=k-1}^{\infty}\beta^i \\
&= \frac{(k-1)p\pi_1}{\beta}\left(\sum_{i=0}^{\infty}\beta^i-\sum_{i=0}^{k-2}\beta^i\right) \\
&= \frac{(k-1)p\pi_1}{\beta}\left(\frac{1}{1-\beta}-\frac{1-\beta^{k-1}}{1-\beta}\right)=\frac{(k-1)p\pi_1}{\beta}\frac{\beta^{k-1}}{1-\beta}.
\end{aligned}$$

Hence, so far,

$$
\begin{aligned}
C &= \sum_{i=0}^{k-2} \pi_i \left(i\binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1} + \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} \right) \\
&\quad + \frac{\pi_1 (k-1) p}{\beta(1-\beta)} \left((\beta p + \overline{p})^{k-2} \beta - \beta^{k-1} \right) + \frac{(k-1) p \pi_1}{\beta} \frac{\beta^{k-1}}{1-\beta} \\
&= \frac{\pi_1 (k-1) p}{(1-\beta)} (\beta p + \overline{p})^{k-2} \\
&\quad + \sum_{i=0}^{k-2} \pi_i \left(i\binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1} + \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} \right). \qquad \text{(C.4)}
\end{aligned}
$$

Next in Eq. (C.4) we have the term

$$
\begin{aligned}
\sum_{i=0}^{k-2} \pi_i i\binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1} &= \pi_1 \sum_{i=1}^{k-2} \beta^{i-1} i\binom{k-1}{i+1} p^{i+1} \overline{p}^{k-i-1} \\
&= \pi_1 \sum_{j=2}^{k-1} \beta^{j-2} (j-1)\binom{k-1}{j} p^{j} \overline{p}^{k-j} \\
&= \frac{\overline{p}\pi_1}{\beta^2} \left((k-1)(\beta p + \overline{p})^{k-2} \beta p - (\beta p + \overline{p})^{k-1} + \overline{p}^{k-1} \right).
\end{aligned}
$$

Substituting this into Eq. (C.4), we have

$$
\begin{aligned}
C &= \sum_{i=0}^{k-2} \pi_i \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} + \frac{\pi_1 (k-1) p}{(1-\beta)} (\beta p + \overline{p})^{k-2} \\
&\quad + \frac{\overline{p}\pi_1}{\beta^2} \left((k-1)(\beta p + \overline{p})^{k-2} \beta p - (\beta p + \overline{p})^{k-1} + \overline{p}^{k-1} \right) \\
&= \sum_{i=0}^{k-2} \pi_i \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} \\
&\quad + \frac{\overline{p}\pi_1}{\beta^2} \left(\overline{p}^{k-1} - (\beta p + \overline{p})^{k-1} \right) + \pi_1 (k-1) p (\beta p + \overline{p})^{k-2} \frac{(\beta p + \overline{p})}{\beta(1-\beta)} \\
&= \sum_{i=0}^{k-2} \pi_i \sum_{l=2}^{k-i} \left(\left\lfloor \frac{l}{2} \right\rfloor + i \right) \binom{k}{i+l} p^{i+l} \overline{p}^{k-i-l} \\
&\quad + \frac{\overline{p}\pi_1}{\beta^2} \left(\overline{p}^{k-1} - (\beta p + \overline{p})^{k-1} \right) + \pi_1 (k-1) p \frac{(\beta p + \overline{p})^{k-1}}{\beta(1-\beta)}. \qquad \text{(C.5)}
\end{aligned}
$$

Consider the remaining sum above. Let $m = i + l$. Then

$$\sum_{i=0}^{k-2}\pi_i\sum_{l=2}^{k-i}\left(\left\lfloor\frac{l}{2}\right\rfloor+i\right)\binom{k}{i+l}p^{i+l}\overline{p}^{k-i-l}=\sum_{m=2}^{k}\binom{k}{m}p^m\overline{p}^{k-m}\left(\sum_{i=0}^{m-2}\pi_i\left(\left\lfloor\frac{m-i}{2}\right\rfloor+i\right)\right)$$
$$=\sum_{m=2}^{k}\binom{k}{m}p^m\overline{p}^{k-m}\left(\sum_{i=0}^{m}\pi_i\left(\left\lfloor\frac{m-i}{2}\right\rfloor+i\right)-(m-1)\pi_{m-1}-m\pi_m\right)\coloneqq S. \qquad \text{(C.6)}$$

The inner sum above can be rewritten as follows:

$$\begin{aligned}
&\sum_{i=0}^{m}\pi_i\left(\left\lfloor\frac{m-i}{2}\right\rfloor+i\right)\\
&\quad=\sum_{i=0}^{m}\pi_i\left(i+\left(\frac{m-i}{2}\right)\frac{1+(-1)^{m-i}}{2}+\left(\frac{m-i-1}{2}\right)\frac{1-(-1)^{m-i}}{2}\right)\\
&\quad=\sum_{i=0}^{m}\pi_i\left(i+\frac{m-i}{2}-\frac{1}{2}\frac{(1-(-1)^{m-i})}{2}\right)\\
&\quad=\sum_{i=0}^{m}\pi_i\left(\frac{2m-1}{4}+\frac{i}{2}+\frac{(-1)^{m-i}}{4}\right)\\
&\quad=\frac{\pi_1}{\beta}\sum_{i=1}^{m}\beta^i\left(\frac{2m-1}{4}+\frac{i}{2}+\frac{(-1)^{m-i}}{4}\right)+\pi_0\left(\frac{2m-1+(-1)^m}{4}\right)\\
&\quad=\frac{\pi_1}{\beta}\left(\frac{2m-1}{4}\left(\frac{1-\beta^{m+1}}{1-\beta}-1\right)+\frac{1}{2}\sum_{i=1}^{m}i\beta^i+\frac{(-1)^m}{4}\left(\frac{1-(-\beta)^{m+1}}{1+\beta}-1\right)\right)\\
&\qquad+\pi_0\left(\frac{2m-1+(-1)^m}{4}\right)\\
&\quad=\pi_1\left(\frac{2m-1}{4}\left(\frac{1-\beta^m}{1-\beta}\right)+\frac{1}{2}\frac{(m\beta^{m+1}-(m+1)\beta^m+1)}{(1-\beta)^2}+\frac{\beta^m-(-1)^m}{4(1+\beta)}\right)\\
&\qquad+\pi_0\left(\frac{2m-1+(-1)^m}{4}\right).
\end{aligned}$$

Now, we can use the fact that $\pi_0+\pi_1/(1-\beta)=1$ to obtain

$$\pi_1\frac{2m-1}{4}\frac{1}{1-\beta}+\frac{2m-1}{4}\pi_0=\frac{2m-1}{4}.$$

Using the same relation, we have

$$\frac{(-1)^m}{4}\left(\pi_0-\frac{\pi_1}{1+\beta}\right)=\frac{(-1)^m}{4}\left(1-\frac{2\pi_1}{1-\beta^2}\right).$$

Therefore,

$$\begin{aligned}
&\sum_{i=0}^{m} \pi_i \left(\left\lfloor \frac{m-i}{2} \right\rfloor + i \right) \\
&= \pi_1 \left(\frac{2m-1}{4} \left(\frac{-\beta^m}{1-\beta} \right) + \frac{m\beta^{m+1} - (m+1)\beta^m + 1}{2(1-\beta)^2} + \frac{\beta^m}{4(1+\beta)} \right) \\
&\quad + \frac{2m-1}{4} + \frac{(-1)^m}{4} \left(1 - \frac{2\pi_1}{1-\beta^2} \right) \\
&= -\pi_1 \beta^m \frac{\beta}{(1-\beta)^2(1+\beta)} - \pi_1 \frac{m\beta^m}{1-\beta} + \frac{m}{2} + \frac{(-1)^m}{4} \left(1 - \frac{2\pi_1}{1-\beta^2} \right) + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4}.
\end{aligned}$$

From Eq. (4.9), we know that $\pi_1 = (1-\beta^2)/2$. Using this,

$$\sum_{i=0}^{m} \pi_i \left(\left\lfloor \frac{m-i}{2} \right\rfloor + i \right) = -\pi_1 \beta^m \frac{\beta}{(1-\beta)^2(1+\beta)} - \pi_1 \frac{m\beta^m}{1-\beta} + \frac{m}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4}.$$

Next, we can write

$$\begin{aligned}
&\sum_{i=0}^{m} \pi_i \left(\left\lfloor \frac{m-i}{2} \right\rfloor + i \right) - (m-1)\pi_{m-1} - m\pi_m \\
&= \frac{-\pi_1 \beta^m \beta}{(1-\beta)^2(1+\beta)} - \pi_1 \frac{m\beta^m}{1-\beta} + \frac{m}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} - \frac{\pi_1}{\beta} m\beta^m \left(\frac{1}{\beta} + 1 \right) + \frac{\pi_1}{\beta^2} \beta^m \\
&= \frac{m}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} + \pi_1 \beta^m \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) - \frac{\pi_1 m \beta^m}{\beta^2(1-\beta)}.
\end{aligned}$$

Hence, Eq. (C.6) becomes

$$
\begin{aligned}
S &= \sum_{m=2}^{k} \binom{k}{m} p^m \overline{p}^{k-m} \left(\pi_1 \beta^m \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) \right. \\
&\quad \left. - \frac{\pi_1 m \beta^m}{\beta^2(1-\beta)} + \frac{m}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} \right) \\
&= \pi_1 \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) (p\beta + \overline{p})^k - \frac{\pi_1 kp\beta (p\beta + \overline{p})^{k-1}}{\beta^2(1-\beta)} \\
&\quad - \overline{p}^k \left(\pi_1 \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} \right) \\
&\quad - kp\overline{p}^{k-1} \left(\pi_1 \beta \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) - \frac{\pi_1 \beta}{\beta^2(1-\beta)} + \frac{1}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} \right) \\
&\quad + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} \\
&= \pi_1 \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) (p\beta + \overline{p})^k - \frac{\pi_1 kp (p\beta + \overline{p})^{k-1}}{\beta(1-\beta)} \\
&\quad - \overline{p}^k \left(\frac{\pi_1}{\beta^2} + \frac{\pi_1}{2(1-\beta^2)} - \frac{1}{4} \right) - kp\overline{p}^{k-1} \left(\frac{-\pi_1}{2(1-\beta^2)} + \frac{1}{4} \right) + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4}.
\end{aligned}
$$

Substituting $\pi_1 = (1-\beta^2)/2$ above and simplifying yields

$$
\begin{aligned}
S &= \pi_1 \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) (p\beta + \overline{p})^k \\
&\quad - \frac{\pi_1 kp}{\beta(1-\beta)} (p\beta + \overline{p})^{k-1} + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} - \overline{p}^k \frac{\pi_1}{\beta^2}.
\end{aligned}
$$

Finally, substituting this result into Eq. (C.5), C becomes

$$
\begin{aligned}
C &= \frac{\overline{p}\pi_1}{\beta^2} \left(\overline{p}^{k-1} - (\beta p + \overline{p})^{k-1} \right) + \pi_1 (k-1) p \frac{(\beta p + \overline{p})^{k-1}}{\beta(1-\beta)} \\
&\quad + \pi_1 \left(\frac{1}{\beta^2} - \frac{\beta}{(1-\beta)^2(1+\beta)} \right) (p\beta + \overline{p})^k \\
&\quad - \frac{\pi_1 kp}{\beta(1-\beta)} (p\beta + \overline{p})^{k-1} + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} - \overline{p}^k \frac{\pi_1}{\beta^2} \\
&= \frac{-\pi_1 (\beta p + \overline{p})^{k-1} (\overline{p}\beta + p)}{(1-\beta)^2(1+\beta)} + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4}.
\end{aligned}
$$

We know from Eq. (4.5) that

$$
(\beta p + \overline{p})^{k-1} (p + \beta \overline{p}) - \beta = 0.
$$

Using this above, we obtain

$$\begin{aligned} C &= -\frac{\pi_1 \beta}{(1-\beta)^2(1+\beta)} + \frac{kp}{2} + \frac{\pi_1}{2(1-\beta)^2} - \frac{1}{4} \\ &= \frac{\pi_1}{2(1-\beta^2)} - \frac{1}{4} + \frac{kp}{2}. \end{aligned}$$

Recall that $\pi_1 = (1-\beta^2)/2$. Hence,

$$\begin{aligned} C &= \frac{1-\beta^2}{2} \frac{1}{2(1-\beta^2)} - \frac{1}{4} + \frac{kp}{2}, \\ C &= \frac{kp}{2}. \end{aligned}$$

Finally, recall that we earlier assumed $q = 1$. Removing this assumption, we obtain

$$C = \frac{qkp}{2}. \qquad \square$$

BIBLIOGRAPHY

[1] Allcock, William, Bresnahan, John, Kettimuthu, Rajkumar, Link, Michael, Dumitrescu, Catalin, Raicu, Ioan, and Foster, Ian. The Globus Striped GridFTP Framework and Server. In *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing* (Washington, DC, USA, 2005), SC '05, IEEE Computer Society, pp. 54–.

[2] Bao, Wei, Wong, V.W.S., and Leung, V.C.M. A Model for Steady State Throughput of TCP CUBIC. In *GLOBECOM 2010*.

[3] Bateman, Martin, Bhatti, Saleem, Bigwood, Greg, Rehunathan, Devan, Allison, Colin, Henderson, Tristan, and Miras, Dimitrios. A comparison of TCP behaviour at high speeds using ns-2 and Linux. In *Proceedings of the 11th communications and networking simulation symposium* (2008), ACM, pp. 30–37.

[4] Bennett, C. H., Brassard, G., and Mermin, N. D. Quantum cryptography without Bell's theorem. *Physical review letters 68*, 5 (1992), 557.

[5] Bennett, Charles H, Brassard, Gilles, Crépeau, Claude, Jozsa, Richard, Peres, Asher, and Wootters, William K. Teleporting an Unknown Quantum State Via Dual Classical and Einstein-Podolsky-Rosen Channels. *Physical review letters 70*, 13 (1993), 1895.

[6] Bennett, Charles H, Brassard, Gilles, Popescu, Sandu, Schumacher, Benjamin, Smolin, John A, and Wootters, William K. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical review letters 76*, 5 (1996), 722.

[7] Bennett, Charles H, and Wiesner, Stephen J. Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states. *Physical review letters 69*, 20 (1992), 2881.

[8] Bhaskar, Mihir K, Riedinger, Ralf, Machielse, Bartholomeus, Levonian, David S, Nguyen, Christian T, Knall, Erik N, Park, Hongkun, Englund, Dirk, Lončar, Marko, Sukachev, Denis D, et al. Experimental demonstration of memory-enhanced quantum communication. *Nature 580*, 7801 (2020), 60–64.

[9] Callegari, Christian, Giordano, Stefano, Pagano, Michele, and Pepe, Teresa. Behavior analysis of TCP Linux variants. *Computer Networks 56*, 1 (2012), 462–476.

[10] Cardwell, Neal, Cheng, Yuchung, Gunn, C Stephen, Yeganeh, Soheil Hassas, and Jacobson, Van. BBR: Congestion-based congestion control. *Queue* (2016).

[11] Crowcroft, Jon, and Oechslin, Philippe. Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP. *ACM SIGCOMM Computer Communication Review 28*, 3 (1998), 53–69.

[12] Dorf, Richard C, and Bishop, Robert H. *Modern Control Systems*. Pearson, 2011.

[13] Ekert, Artur K. Quantum Cryptography Based on Bell's Theorem. *Physical review letters 67*, 6 (1991), 661.

[14] El Khoury, R, Altman, E, and El Azouzi, R. Analysis of scalable TCP congestion control algorithm. *Computer Communications* (2010).

[15] Eldredge, Z., Foss-Feig, M., Gross, J. A., Rolston, S. L., and Gorshkov, A. V. Optimal and secure measurement protocols for quantum sensor networks. *Physical Review A 97*, 4 (2018), 042337.

[16] ESnet. ESnet Fasterdata Knowledge Base. `http://fasterdata.es.net/`. Accessed: 2018-10-31.

[17] Ewert, F., and van Loock, P. 3/4-Efficient Bell Measurement with Passive Linear Optics and Unentangled Ancillae. *Physical review letters* (2014).

[18] Floyd, Sally. HighSpeed TCP for large congestion windows.

[19] Folland, GB. Higher-Order Derivatives and Taylor's Formula in Several Variables. `https://sites.math.washington.edu/~folland/Math425/taylor2.pdf`, 2005. Accessed: 2018-01-10.

[20] Genin, Daniel, and Marbukh, Vladimir. Bursty fluid approximation of TCP for modeling Internet congestion at the flow level. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2009), IEEE, pp. 1300–1306.

[21] Giovannetti, Vittorio, Lloyd, Seth, and Maccone, Lorenzo. Quantum-enhanced measurements: beating the standard quantum limit. *Science 306*, 5700 (2004), 1330–1336.

[22] Grasselli, Federico, Kampermann, Hermann, and Bruß, Dagmar. Finite-key effects in multipartite quantum key distribution protocols. *New Journal of Physics 20*, 11 (2018), 113014.

[23] Grice, W. P. Arbitrarily Complete Bell-State Measurement Using Only Linear Optical Elements. *Physical Review A* (2011).

[24] Gu, Keqin, Chen, Jie, and Kharitonov, Vladimir L. *Stability of Time-Delay Systems*. Springer Science & Business Media, 2003.

[25] Gu, Yunhong, and Grossman, Robert L. UDT: UDP-based Data Transfer for High-speed Wide Area Networks. *Computer Networks* (2007).

[26] Guha, S., Krovi, H., Fuchs, C. A., Dutton, Z., Slater, J. A., Simon, C., and Tittel, W. Rate-loss analysis of an efficient quantum repeater architecture. *Physical Review A 92*, 2 (2015), 022357.

[27] Guok, Chin, Engineer, EN, and Robertson, David. ESnet On-Demand Secure Circuits and Advance Reservation System (OSCARS). *Internet2 Joint 92* (2006).

[28] Ha, Sangtae, Kim, Yusung, Le, Long, Rhee, Injong, and Xu, Lisong. A step toward realistic performance evaluation of high-speed TCP variants. In *Fourth International Workshop on Protocols for Fast Long-Distance Networks* (2006).

[29] Ha, Sangtae, and Rhee, Injong. Taming the elephants: New TCP slow start. *Computer Networks 55*, 9 (2011), 2092–2110.

[30] Ha, Sangtae, Rhee, Injong, and Xu, Lisong. CUBIC: a New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating Systems Review* (2008).

[31] Hao, Jiu-Cang, Li, Chuan-Feng, and Guo, Guang-Can. Controlled dense coding using the Greenberger-Horne-Zeilinger state. *Physical Review A 63*, 5 (2001), 054301.

[32] Herbauts, Isabelle, Blauensteiner, B, Poppe, A, Jennewein, T, and Huebel, Hannes. Demonstration of Active Routing of Entanglement in a Multi-User Network. *Optics express 21*, 23 (2013), 29013–29024.

[33] Hollot, Chris V, Misra, Vishal, Towsley, Don, and Gong, Wei-Bo. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)* (2001), vol. 3, IEEE, pp. 1726–1734.

[34] Hollot, CV, and Chait, Yossi. Nonlinear stability analysis for a class of TCP/AQM networks. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)* (2001), vol. 3, IEEE, pp. 2309–2314.

[35] Hollot, CV, Misra, Vishal, Towsley, Don, and Gong, Wei-Bo. A Control Theoretic Analysis of RED. In *INFOCOM 2001.* (2001).

[36] Hopp, W. J., and Simon, J. T. Bounds and heuristics for assembly-like queues. *Queueing systems 4*, 2 (1989), 137–155.

[37] Huang, Xiaomeng, Chuang, LIN, and Fengyuan, REN. Generalized Modeling and Stability Analysis of Highspeed TCP and Scalable TCP. *IEICE transactions on communications* (2006).

[38] Jamal, Habibullah, and Sultan, Kiran. Performance analysis of TCP congestion control algorithms. *International journal of computers and communications 2*, 1 (2008), 18–24.

[39] Jiang, L., Taylor, J. M., Sørensen, A. S., and Lukin, M. D. Distributed quantum computation based on small quantum registers. *Physical Review A 76*, 6 (2007), 062323.

[40] Kelly, Frank P, Maulloo, Aman K, and Tan, David KH. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research society* (1998).

[41] Kelly, Tom. Scalable TCP: Improving performance in highspeed wide area networks. *ACM SIGCOMM computer communication Review* (2003).

[42] Kleinrock, Leonard. *Queueing Systems, Volume I: Theory.* Wiley New York, 1975.

[43] Lee, Yuan, Bersin, Eric, Dahlberg, Axel, Wehner, Stephanie, and Englund, Dirk. A quantum router architecture for high-fidelity entanglement flows in multi-user quantum networks. *arXiv preprint arXiv:2005.01852* (2020).

[44] Leith, D. H-TCP: TCP Congestion Control for High Bandwidth-Delay Product Paths. Tech. rep., April 2008.

[45] Leith, DJ, and Shorten, RN. On RTT Scaling in H-TCP. *Discussion note, Hamilton Institute*.

[46] Leith, Douglas, Shorten, R, and Lee, Y. H-TCP: A framework for congestion control in high-speed and long-distance networks. In *PFLDnet Workshop* (2005).

[47] Leith, Douglas, and Shorten, Robert. H-TCP: TCP for high-speed and long-distance networks. In *Proceedings of PFLDnet* (2004).

[48] Li, Yee-Ting, Leith, Douglas, and Shorten, Robert N. Experimental evaluation of TCP protocols for high-speed networks. *Networking, IEEE/ACM Transactions on 15*, 5 (2007), 1109–1122.

[49] Mbarek, Rachid, Othman, Mohamed Tahar Ben, and Nasri, Salem. Performance Evaluation of Competing High-Speed TCP Protocols. *IJCSNS 8*, 6 (2008).

[50] Misra, Vishal, Gong, Wei-Bo, and Towsley, Don. Stochastic differential equation modeling and analysis of TCP-windowsize behavior. In *Proceedings of PERFORMANCE* (1999), vol. 99, Citeseer.

[51] Misra, Vishal, Gong, Wei-Bo, and Towsley, Don. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. *SIGCOMM Comput. Commun. Rev.* (2000).

[52] Morris, Robert. TCP behavior with many flows. In *Proceedings of International Conference on Network Protocols* (1997), IEEE.

[53] Muralidharan, Sreraman, Li, Linshu, Kim, Jungsang, Lütkenhaus, Norbert, Lukin, Mikhail D, and Jiang, Liang. Optimal Architectures for Long Distance Quantum Communication. *Scientific reports 6* (2016), 20463.

[54] Nain, Philippe, Vardoyan, Gayane, Guha, Saikat, and Towsley, Don. On the Analysis of a Multipartite Entanglement Distribution Switch. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* (2020).

[55] Nielsen, Michael A, and Chuang, Isaac. Quantum Computation and Quantum Information, 2002.

[56] NS-3 Development Team. NS-3 Network Simulator. `https://www.nsnam.org/`. Accessed: 2018-01-10.

[57] Pan, R., Natarajan, P., Baker, F., and White, G. Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem. Tech. rep., 2017.

[58] Pan, Rong, Prabhakar, Balaji, and Laxmikantha, Ashvin. QCN: Quantized Congestion Notification. *IEEE802* (2007).

[59] Pant, Mihir, Krovi, Hari, Towsley, Don, Tassiulas, Leandros, Jiang, Liang, Basu, Prithwish, Englund, Dirk, and Guha, Saikat. Routing Entanglement in the Quantum Internet.

[60] Pirandola, S., Laurenza, R., Ottaviani, C., and Banchi, L. Fundamental limits of repeaterless quantum communications. *Nature communications 8*, 1 (2017), 1–15.

[61] Poojary, Sudheer, and Sharma, Vinod. An Asymptotic Approximation of TCP CUBIC. *arXiv preprint arXiv:1510.08496* (2015).

[62] Ramachandran, S., and Delen, D. Performance analysis of a kitting process in stochastic assembly systems. *Computers & Operations Research 32*, 3 (2005), 449–463.

[63] Rao, Nageswara SV, Imam, Neena, Hanley, Jessey, and Oral, Sarp. Wide-Area Lustre File System Using LNet Routers. In *2018 Annual IEEE International Systems Conference (SysCon)* (2018), IEEE, pp. 1–6.

[64] Schmid, Christian, Kiesel, Nikolai, Weber, Ulrich K, Ursin, Rupert, Zeilinger, Anton, and Weinfurter, Harald. Quantum teleportation and entanglement swapping with linear optics logic gates. *New Journal of Physics 11*, 3 (2009), 033008.

[65] Schoute, Eddie, Mancinska, Laura, Islam, Tanvirul, Kerenidis, Iordanis, and Wehner, Stephanie. Shortcuts to Quantum Network Routing.

[66] Shchukin, Evgeny, Schmidt, Ferdinand, and van Loock, Peter. Waiting time in quantum repeaters with probabilistic entanglement swapping. *Physical Review A 100*, 3 (2019), 032322.

[67] Sinclair, Neil, Saglamyurek, Erhan, Mallahzadeh, Hassan, Slater, Joshua A, George, Mathew, Ricken, Raimund, Hedges, Morgan P, Oblak, Daniel, Simon, Christoph, Sohler, Wolfgang, et al. Spectral Multiplexing for Scalable Quantum Photonics Using an Atomic Frequency Comb Quantum Memory and Feedforward Control. *Physical review letters 113*, 5 (2014), 053603.

[68] Som, P., Wilhelm, W. E., and Disney, R. L. Kitting process in a stochastic assembly system. *Queueing Systems 17*, 3-4 (1994), 471–490.

[69] Srikant, Rayadurgam. *The Mathematics of Internet Congestion Control*. Springer Science & Business Media, 2012.

[70] Takeoka, M., Guha, S., and Wilde, M. M. Fundamental rate-loss tradeoff for optical quantum key distribution. *Nature communications 5*, 1 (2014), 1–7.

[71] Towns, John, Cockerill, Timothy, Dahan, Maytal, Foster, Ian, Gaither, Kelly, Grimshaw, Andrew, Hazlewood, Victor, Lathrop, Scott, Lifka, Dave, Peterson, Gregory D, et al. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering 16*, 5 (2014), 62–74.

[72] Van Meter, Rodney. *Quantum Networking*. John Wiley & Sons, 2014.

[73] Wang, Guodong, Wu, Yulei, Dou, Ke, Ren, Yongmao, and Li, Jun. AppTCP: The design and evaluation of application-based TCP for e-VLBI in fast long distance networks. *Future Generation Computer Systems 39* (2014), 67–74.

[74] Wei, David X, Jin, Cheng, Low, Steven H, and Hegde, Sanjay. FAST TCP: motivation, architecture, algorithms, performance. *IEEE/ACM transactions on Networking 14*, 6 (2006), 1246–1259.

[75] Xu, Lisong, Harfoush, K., and Rhee, Injong. Binary increase congestion control (BIC) for fast long-distance networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* (March 2004), vol. 4, pp. 2514–2524 vol.4.

[76] Yu, Se-young, Brownlee, Nevil, and Mahanti, Aniket. Comparative Analysis of Big Data Transfer Protocols in an International High-Speed Network.

[77] Zhuang, Quntao, Zhang, Zheshen, and Shapiro, Jeffrey H. Distributed quantum sensing using continuous-variable multipartite entanglement. *Physical Review A 97*, 3 (2018), 032329.

[78] Zukowski, M., Zeilinger, A., Horne, M. A., and Ekert, A. K. "Event-ready-detectors"Bell experiment via entanglement swapping. *Physical Review Letters 71* (1993), 4287–4290.