9-2011

Queue Length Based Pacing of Internet Traffic

Cai Yan
University of Massachusetts Amherst

QUEUE LENGTH BASED PACING OF INTERNET TRAFFIC

A Dissertation Presented

by

YAN CAI

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

DOCTOR OF PHILOSOPHY

September 2011

Department of Electrical and Computer Engineering

QUEUE LENGTH BASED PACING OF INTERNET TRAFFIC

A Dissertation Presented

by

YAN CAI

Approved as to style and content by:

Tilman Wolf, Co-chair

Weibo Gong, Co-chair

Don Towsley, Member

Lixin Gao, Member

Christopher V. Hollot, Department Chair
Department of Electrical and Computer Engineering

To my parents, my wife Wen Tang, our lovely son Evan and daughter Emma.

ACKNOWLEDGMENTS

This dissertation would not have been finished without the help from many people.

First, I own my heartfelt gratitude to my advisor, Professor Weibo Gong, for his support, encourage and guidance. With his enthusiasm in mathematics and keen vision to academic research, Professor Gong exemplified me what a distinguished professor looks like. His valuable advice will guide me through the rest of my life.

I would like to express my sincere appreciation towards my co-advisor, Professor Tilman Wolf. Without him, I could not have done my dissertation this way. His creative thinking and excellent research experiences inspired many brilliant ideas in this dissertation. His many valuable suggestions greatly improved the quality of this dissertation.

It is my honor to have Professor Don Towsley and Professor Lixin Gao serve in my committee. It is very fortunate for me to have chances to discuss research topics with Professor Don Towsley. Professor Lixin Gao provided me with incredible help during my studies periods.

Many thanks to my collaborators at other Universities and the fellow students in CSMCL Lab and Computer Network Research Group in the Computer Science Department: Professor Yong Liu, Professor Patrick P. C. Lee, Dr. Yong Huang, Dr. Jie Sun, Bo Jiang and Sheng Xiao. They gave me wonderful suggestions and created a rewarding and enjoyable academic environment.

I am deeply indebted to my parents and my brother for their support and love. I am grateful to my wife, Wen Tang, whose love, care and encouragement make my life so wonderful and meaningful. Our lovely son, Evan, and daughter, Emma, have

brought me the most precious moments and made me realize that giving love is of the same importance as being loved.

ABSTRACT

QUEUE LENGTH BASED PACING OF INTERNET TRAFFIC

SEPTEMBER 2011

YAN CAI

B.En., TSINGHUA UNIVERSITY, P.R. CHINA

M.Sc., TSINGHUA UNIVERSITY, P.R. CHINA

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf and Professor Weibo Gong

As the Internet evolves, there is a continued demand for high Internet bandwidth. This demand is driven partly by the widely spreading real-time video applications, such as on-line gaming, teleconference, high-definition video streaming. All-optical switches and routers have long been studied as a promising solution to the rapidly growing demand. Nevertheless, buffer sizes in all-optical switches and routers are very limited due to the challenges in manufacturing larger optical buffers. On the other hand, Internet traffic is bursty. The existence of burstiness in network traffic has been shown at all time scales, from tens of milliseconds to thousands of seconds. The widely existing burstiness has a very significant impact on the performance of small buffer networks, resulting in high packet drop probabilities and low link utilization.

There have been many solutions proposed in the literature to solve the burstiness issue of network traffic. Traffic engineering techniques, such as traffic shaping

and polishing, have been available in commercial routers/switches since the era of Asynchronous Transfer Mode (ATM) networks. Moreover, TCP pacing, as a natural solution to the TCP burstiness, has long been studied. Furthermore, several traffic conditioning and scheduling techniques are proposed to smooth core network traffics in a coordinated manner. However, all the existing solutions are inadequate to efficiently solve the burstiness issue of high-speed traffic.

In this dissertation we aim to tackle the burstiness issue in small buffer networks, which refer to the future Internet core network consisting of all-optical routers and switches with small buffers.

This dissertation is composed of two parts. In the first part, we analyze the impact of a general pacing scheme on the performance of a tandem queue network. This part serves as a theoretical foundation, based on which we demonstrate the benefits of pacing in a tandem queue model. Specifically, we use the Infinitesimal Perturbation Analysis (IPA) technique to study the impact of pacing on the instantaneous and average queue lengths of a series of nodes. Through theoretical analyses and extensive simulations, we show that under certain conditions there exists a linear relationship between system parameters and instantaneous/average queue lengths of nodes and that pacing improves the performance of the underlying tandem queue system by reducing the burstiness of the packet arrival process.

In the second part, we propose a practical on-line packet pacing scheme, named Queue Length Based Pacing (QLBP). We analyze the impact of QLBP on the underlying network traffic in both time and frequency domains. We also present two implementation algorithms that allow us to evaluate the performance of QLBP in real experimental and virtual simulation environments. Through extensive simulations, we show that QLBP can effectively reduce the burstiness of network traffic and hence significantly improve the performance of a small buffer network. More important, the network traffic paced with QLBP does not exhibit a weakened competition capa-

bility when competing with non-paced traffic, which makes the QLBP scheme more attractive for ISPs.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

There is a continued demand for high network bandwidth due to the rapidly growing volume of traffic on the Internet. At present, most of the Internet infrastructure is composed of electronic routers and switches that provide end-to-end connectivity. However, researchers are concerned with the possibility that some electronic "bottleneck" within electronic routers will eventually limit network capacity [67]. In addition, power consumption and heat dissipation problems are becoming a major issue in the deployment of large-scale electronic routers.

All-optical routers have long been studied as a promising solution to meet the rapidly surging demand on the Internet bandwidth and overcome the power dissipation and scaling problems in electric routers [9, 68, 81, 37, 17]. An important feature of an all-optical router is that packets are transmitted all the way through the router in optical form, which is referred to as optical transparency [68]. This requires packets to be buffered inside all-optical routers in the form of light. The most common techniques for implementing an optical buffer are fiber delay lines [9, 52, 80, 13] and slow-light delay lines [11, 72, 41]. Nevertheless, these techniques limit the sizes of optical buffers to be very small (e.g., a dozen of packets) [67].

The use of small buffers in core networks has been justified with theoretical analyses and empirical conclusions [22, 76, 59, 30, 45]. Enachescu et al. [22] argued that $O(\log W)$ buffers are sufficient for high throughput, where W is congestion window size of each flow, and router buffer can even be reduced to a few dozen packets if a small amount of link utilization is sacrificed. Gu et al. [30] demonstrated that a more

than 90% link utilization is achievable in a X Gbps bottleneck link with a buffer of 20 packets, where X is in the range from 1 to 10. Lakshmikantha et al. [45] further showed that $O(1)$ buffer sizes, on the order of 20 packets, are sufficient for good performance with no loss of link utilization when considering the impact of file arrivals and departures.

We observe that all high performance results obtained in [22, 30, 45] are achieved only when TCP sessions are paced by either some rate-control mechanism, known as TCP pacing, or access links have capacities much smaller than the bottleneck link. This indicates that in a small buffer network, pacing is a necessary technique to prevent high packet drop probabilities at optical routers/switches with small buffers.

There have been several solutions to the high packet drop rate issue in small buffer networks proposed in the literature [23, 82, 6, 64, 53, 1, 2]. Ordinary traffic shaping and policing techniques, which are widely available in ATM routers, seem to be adequate to circumvent the burstiness issue of network traffic. However, as explained in detail later, without carefully designed modifications, these existing traffic conditioning techniques are incapable of effectively reducing burstiness at small buffer routers. As a natural solution to TCP burstiness, TCP pacing finds its roots in the explicit rate control non-TCP protocols [82]. However, empirical results show [3] that paced TCP flows have lower shares of the bottleneck link than unpaced TCP flows. This weakened competition capability of paced TCP against non-paced TCP prevents the wide adoption of pacing-enabled TCPs [3]. The other approaches proposed in [64, 53, 1, 2] rely on either prior statistical knowledge of the underlying traffic or a global network-wide coordinated scheduling privilege.

The strong demand for a traffic shaping technique that can effectively reduce traffic burstiness in a core network motivates our work in this dissertation. To this end, an ideal technique should satisfy the following requirements. First, it is so simple that it can be implemented at a high-speed processing rate. Second, it does not require any

prior knowledge of traffic statistics. Third, it works in a distributed manner and an accumulative pacing effect can be achieved by deploying multiple such pacing systems within a network.

With ideal pacing techniques described above, we further advocate a packet pacing architecture for the next-generation Internet in which traffic bursts traversing multiple pacing nodes are smoothed out to nearly match constant bit-rate traffic. Using this pacing technology throughout access networks will help operate optical core networks more effectively.

1.1 Burstiness of Internet Traffic

Self-similarity of Internet traffic indicates that burstiness exhibits in a wide range of time scales, from tens of milliseconds to several minutes and even longer [46, 56, 74, 19]. The burstiness of Internet traffic is roughly categorized into two classes: "long-term" and "short-term." As pointed out in [24, 26], long-term and short-term burstiness are mainly contributed by user/session attributes at the macroscopic level and TCP congestion control mechanisms at the microscopic level, respectively. Specifically, long-term burstiness refers to the time scales from hundreds of milliseconds to tens of minutes and short-term burstiness the time scales below hundreds of milliseconds [26].

The time scales are very important in network management. In the context of IP Quality of Service (QoS), three timeframes are specified: $O(milliseconds)$, $O(100\ milliseconds)$ and $O(10\ seconds)$ and more [23]. The first timeframe is such that congestion is mainly caused by short-time bursts of individual traffic streams or of the aggregate traffic where the traffic volume exceeds the available bandwidth. QoS mechanisms relevant to this timeframe include queueing, scheduling and dropping techniques. The second timeframe defines network round-trip times (RTTs), which are important to TCP-based close-looped applications. Active queue manage-

ment (AQM) as a congestion control technique works within this timeframe. The third timeframe is relevant to the management of the long-term average network traffic rates and capacities, which is achieved through capacity planning and traffic engineering.

To capture the global scaling behavior (i.e., the long-term burstiness) reflected in local-area network (LAN) and wide-area network (WAN) traffics [74, 19], a TCP-based hierarchical HTTP traffic generator model is proposed in [24], where the user/session attributes are taken into account as the major contributors to burstiness. In particular, the hierarchy of the traffic model consists of a number of TCP sessions, each containing a number of pages, each of whom includes several objects to transmit. The user/session attributes in terms of inter-session time, pages per session, inter-page time, objects per page, inter-object time, and object time, are explicitly specified as parameters in this model. With appropriate settings, this model can successfully recreate the global scaling behavior, namely, the self-similar property exhibited in the captured network traffic data.

In this dissertation, we aim to tackle the short-term burstiness, rather than the long-term burstiness. On one hand, any effort to reduce the long-term burstiness will inevitably affect user experiences, causing longer delays that are on the order of hundreds of milliseconds. Such delays are too long to be tolerated by end users and applications at higher layers. On the other hand, since short-term burstiness usually takes place at time scales much lower than hundreds of milliseconds, the delays introduced by certain traffic smoothing techniques are not perceivable. This forms the guideline for the work presented in this dissertation.

1.1.1 Burstiness of TCP

Arguably speaking, the closed-loop congestion control mechanisms in TCP are the major cause for the short-term burstiness of IP traffic [24]. In this subsection, we

first briefly introduce the TCP congestion control mechanisms and then review how they contribute to short-term burstiness.

1.1.1.1 TCP Congestion Control

The following description of the congestion control mechanisms in TCP is mainly taken from Chapter 3.7 in [43]. For more details of TCP, see RFC 2581 [5]. TCP Reno is used for illustration purposes in the following. The terms "packets" and "segments", "connection" and "session" are used interchangeably.

Transmission control protocol (TCP) is a sliding window protocol. A TCP connection consists of a sender and a receiver at both sides of the connection. At any particular point of time, the sender maintains two state variables, congestion window size and receive window size, denoted by $CWND$ and $RWND$, respectively, where $RWND$ is advertised by the receiver. Upon receipt of an acknowledgement (ACK), the sender sends one or more segments, depending on which phase the TCP session is in. Upon receipt of a segment, the receiver sends back an acknowledgement that acknowledges the last cumulative segment. An cumulative segment is defined as one for which all preceding segments have successfully arrived at the receiver. Within each round-trip time (RTT), the amount of unacknowledged segments transmitted in flight is bounded by the minimum of $CWND$ and $RWND$. $RWND$ is used for flow control purposes [5]. By assuming that $RWND$ is so large that $CWND$ is always smaller than $RWND$ during the lifetime of a TCP session, the amount of unacknowledged segments is limited by $CWND$ solely.

The TCP congestion control algorithm has three major components: slow start, congestion avoidance, and reaction to timeout events.

Slow start

When a TCP connection begins, the value of $CWND$ is initialized to one. Here we assume that the unit of $CWND$ is equal to the size of a segment and all segments have

the same sizes. Upon receipt of an acknowledgement, $CWND$ is increased by one. The sender keeps increasing its $CWND$ this way until a loss event is perceived, at which time $CWND$ is cut in half and the slow start phase ends. During the slow start phase $CWND$ doubles every RTT, starting from one, which is known as the exponential growth of the congestion window. The TCP session enters the congestion avoidance phase right after the slow start phase ends. A packet loss event is identified with the receipt of three consecutive acknowledgements with the same acknowledged sequence number, known as duplicate ACKs.

Congestion Avoidance

During the congestion avoidance phase, upon the receipt of an acknowledgement, $CWND$ increases by $\frac{1}{CWND}$. Once a packet loss event is perceived, $CWND$ is reduced to $\frac{CWND}{2}$. The rules specifying the changes in the congestion window in response to an acknowledgement or a packet loss is known as the additive increase and multiplicative decrease (AIMD) rules. The term AIMD stems from the fact that under this mechanism, $CWND$ increases by one every RTT and decreases by a fixed fraction of $CWND$ in the face of a packet loss event. The TCP session keeps running within the congestion avoidance phase until ether a timeout event occurs or the connection is ended explicitly.

Reaction to Timeout Events

The TCP session sets up a timer for every packet it has sent and the timer starts as soon as the associated packet is sent out. If the acknowledgement to that packet is returned to the sender before the timer expires, the timer is deactivated and canceled. Otherwise, the timer will expire and a timeout event occurs, which will reduce $CWND$ back to one, no matter how large the current $CWND$ is. From that moment on, the TCP session enters the slow start phase, starting with $CWND$ of one.

1.1.1.2 Burstiness Inherent to TCP Congestion Control Mechanisms

It is widely pointed out that congestion control mechanisms of TCP can cause burstiness of TCP traffic (for details, see [24, 3]). In what follows we briefly review the work done in [3].

The authors presented in [3] three aspects of the TCP congestion control mechanism that cause TCP traffic to be burst. They are slow start, losses, and ACK compression. Before going into details, let us first briefly discuss how the TCP congestion control mechanism helps reduce the burstiness.

In a situation where the bottleneck link only serves one TCP session, the throughput of the TCP session is bounded by the bottleneck link rate. The bottleneck link can be saturated with packets of the TCP session. Thus, the acknowledgements are also sent back to the sender at the bottleneck link rate. As long as the associated buffer is not full, the congestion window of the TCP session can increase while the queue is being built up. In this case, the packets within one RTT are evenly spread over the entire RTT, and as a result, the TCP traffic shows no burstiness. This phenomenon is known as ACK-clocking. Even though ACK-clocking in effect smooths TCP traffic, it rarely occurs, because the bottleneck link rate is much higher than the available bandwidth of a single individual TCP connection owing to multiplexing.

Slow Start

During the slow start phase, upon receipt of an acknowledgement, the sender sends out two packets. Without loss of generality, suppose that at time t, $CWND = W$. Also suppose that at this time the throughput of the TCP session is lower than its available bandwidth B, that is, $\frac{W}{RTT} < B$, where RTT is the round trip time of the TCP session at time t. With the assumption that no packets are dropped as long as the buffer at the bottleneck link is not full, all W packets arrive at the receiver at rate B. As a result, all W acknowledgements are sent back to the sender also at rate B. Since each acknowledgement's arrival will trigger two packets to be sent, the

resulting $2W$ packets are sent out at a sending rate of $2B$. Note that even though these $2W$ packets are sent at rate $2B$, they arrive at the receiver at rate B because B is the bottleneck link rate. Within the same RTT, W packets are buffered at the bottleneck link while the other W packets are delivered to the sender at rate B. As the congestion window keeps increasing from RTT to RTT, the number of packets stored at the bottleneck link's buffer also increases. Since the buffer size at the bottleneck link is definitely finite in practice, the number of packets stored in the buffer can always reach the limit, and then a packet drop will occur with a new packet arriving at the full buffer. Such a packet drop ends the slow start stage. During the entire slow start phase packets are always sent at rate $2B$, which is a bursty behavior.

Losses

One way for the sender to detect losses is by receiving duplicate ACKs. Once the lost packet is successfully retransmitted, the receiver's next acknowledgement will acknowledge not only the lost packet but also other packets that had been successfully received by the receiver. When the acknowledgement arrives at the sender, the sender will be allowed to send a bunch of packets.

ACK Compression

ACK compression refers to as a situation in which a bunch of acknowledgements arrive at the sender in a bursty manner. Such a behavior can be caused by queueing acknowledgements at some intermediate routers due to congestion on the reverse path of the TCP connection. Even thought acknowledgements can ideally be spread evenly at the bottleneck link rate, such an ACK clocking effect might be weakened or even eliminated by the congestion that occurs quite often on the round trip path of a TCP connection.

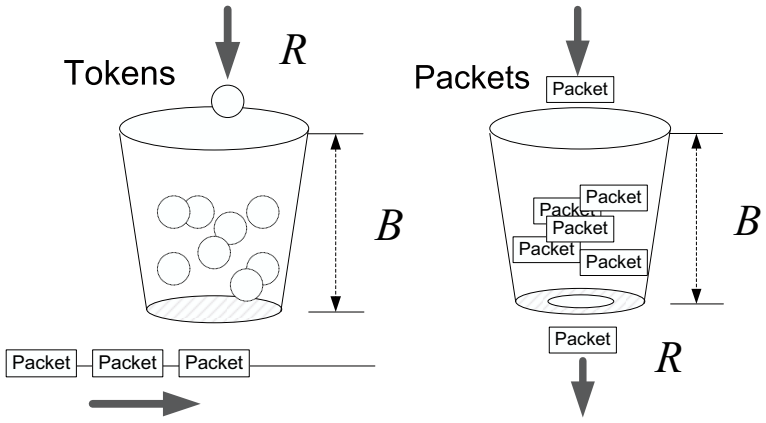
1.2 Solutions to Burstiness

IP QoS mechanisms are used to ensure that a network can provide certain levels of services that meet the requirements specified in service level agreements (SLA). Burstiness, as a characteristic of IP traffic, has a close relationship with the metrics defined in an SLA in that the extent to which the traffic is bursty significantly affects the values of the metrics. In IP networks, the metrics defined in the SLA are delay, jitter, packet loss, throughput, service availability and per flow sequence preservation. From knowledge of queueing theory, we know that given the same average rate, the burstier the traffic, the longer the queue. Consequently, packets passing through routers experience longer delays, larger jitters, higher packet drop rates and as a result, suffer from a lower throughput.

Since burstiness has a significant impact on SLA metrics, there have been a number of mechanisms and techniques designed to handle the burstiness of IP traffic, such as policing and shaping. The token bucket and leaky bucket algorithms are two main algorithms used to implement traffic policing and shaping techniques. Besides, since TCP is the main transport protocol on which most of the Internet traffic is carried, TCP pacing also has received great attention as a natural solution to TCP burstiness. However, they all come with their own limitations, as we will observe in what follows. In this section, we briefly introduce these existing solutions and illustrate their weaknesses in dealing with burstiness in the current Internet.

1.2.1 Token Bucket and Leaky Bucket Algorithms

Token bucket and leaky bucket algorithms are widely used as traffic policing or shaping techniques to enforce a maximum rate cap on the underlying traffic, which could be an individual flow, a traffic class consisting of multiple flows satisfying some classification criteria, or even the aggregate traffic over a link. However, the terms



(a) token bucket (b) leaky bucket

Figure 1.1. Token bucket and leaky bucket structures

"token bucket" and "leaky bucket" have confused and are often mistakenly used. For the sake of clarification, they are presented together. The reader is referred to [23].

1.2.1.1 Token Bucket Algorithm

The token bucket algorithm can be used in both a policer and a shaper to enforce a maximum rate to a traffic stream, even though a policer and a shaper differ fundamentally in their treatment of packets. In what follows we first introduce the concept of the token bucket algorithm and then illustrate how it is applied to implement a policer and a shaper. We conclude by pointing out the difference between a policer and a shaper.

Token Bucket Algorithm

In the token bucket algorithm there is a bucket with depth B (in bytes). Tokens with size of 1 byte are added to the bucket at rate R (in bytes per second). Tokens can be added either every time a packet is processed, or at regular intervals, depending on the particular implementations. In either case, the rate at which the tokens are added into the bucket needs to be equal to R. If a token arrives at a bucket full of tokens, the token will be dropped. Thus, it is guaranteed that the amount of tokens

(in bytes) is always less than or equal to B. Figure 1.1-(a) shows the structure of a token bucket.

When the token bucket algorithm is applied to a traffic stream, every packet belonging to the traffic stream asks the token bucket to grant it an amount of tokens equal to its size (in bytes). If there is a sufficient enough amount of tokens left in the bucket for the packet, then we say this packet has conformed to the token bucket definition. If there are fewer tokens left in the bucket, then we say this packet has exceeded the token bucket definition. For packets that have conformed to the token bucket definition, both the token bucket-based policer and shaper forward them immediately. For packets that have exceeded the token bucket definition, the token bucket-based policer just simply drops them while the token bucket-based shaper puts them into a dedicated queue. Whenever there are more tokens available in the bucket, the shaper will pull the packets out, assign tokens to them, and forward them. Here we can see that the difference between a policer and a shaper is whether or not the "excess" packets are stored for late transmission.

Note that the maximum rate R enforced by a token bucket-based policer or shaper is the maximum average rate, that is, over the long-term period, the maximum average rate of traffic policed or shaped by the token bucket algorithm is R. The instantaneous throughput of the policed or shaped traffic can be higher than R. For instance, if a burst of packets arrive at a bucket full of tokens in a back-to-back manner at a rate higher than R, then they will be forwarded out of the policer or the shaper at the same rate again as long as the sum of these packets' sizes is smaller than the bucket depth B.

1.2.1.2 Leaky Bucket Algorithm

In contrast with the token bucket algorithm, the leaky bucket algorithm is only used in a shaper and strictly limits the instantaneous throughput of the shaped traffic to R, which is the maximum rate at which packets can flow out from the bucket.

In the leaky bucket algorithm, there is also a bucket with depth B (in bytes). There is a hole at the bottom that allows packets to flow out at rate R (in bytes). If the bucket is full when a packet arrives at it , the packet is dropped. A leaky bucket-based shaper performs exactly like a first-in-first-out (FIFO) queue with outgoing rate R and queue limit B. Figure 1.1-(b) shows the structure of a leaky bucket-base shaper.

The differences between a token bucket-based shaper and a leaky bucket-based shaper are summarized as follows. First, instead of tokens stored in a token bucket, packets are stored in a leaky bucket. There is no concept of token in the case of the leaky bucket algorithm at all. Second, the maximum peak rate in a token bucket-based shaper can exceed R while the maximum peak rate in a leaky bucket-based shaper is strictly bounded by R.

The best known example of a leaky bucket algorithm is the Generic Cell Rate Algorithm (GCRA) used in traffic shaping of ATM networks [27].

In summary, the most obvious disadvantage of the token bucket-based or leaky bucket-based poliers/shapers is that they will do nothing if the peak rates of the underlying traffic are smaller than R. It will be shown in Chapter 3 that our proposed pacing scheme can always achieve a certain level of smoothing effect on the underlying traffic no matter how low its peak rates are.

1.2.2 TCP Pacing

As a natural solution to the burstiness of TCP traffic, TCP pacing has been studied for a while. TCP pacing was initially proposed by Zhang et al. in [82] to

correct for the acknowledgement compression due to cross traffic. Since then, research has followed, suggesting different usages of pacing in TCP for different purposes, such as compensation of ACK-clocking in slow start [7, 55] and fast recovery after a packet loss [33], reducing burstiness in asymmetric networks [8].

In this dissertation, we adopted the description of TCP pacing provided in [3]. In [3], pacing is implemented throughout the lifetime of a TCP session. Packet-sending is no longer triggered by arrivals of acknowledgements. Instead, with a given $CWND$ and a given RTT, packets are scheduled to be evenly sent out over the entire RTT, that is, the sender sends one packet per $RTT/CWND$ seconds. Upon the receipt of acknowledgements, $CWND$ is updated in the same way as in the ordinary TCP (Reno). Since a fine-grained estimate of RTT is required to calculate the interval $RTT/CWND$, the TCP timestamp option is enabled to get accurate RTT samples. Also, the RTT estimate is computed using the exponential weighted moving average (EWMA) algorithm. Every time either $CWND$ or RTT is updated, the interval is recalculated and then applied to the subsequent packet transmissions. Thus, instead of being sent in a bursty manner, the packets are evenly spread over the whole RTT, eliminating the short-term burstiness, which was inherent to the ordinary non-paced TCP.

The conclusions on the impact of TCP pacing on network performance have been controversial. On one hand, empirical studies conducted in [3] indicate that although TCP pacing can improve throughput and fairness in some situations, it degrades the performance of TCP in general. The poor performance of pacing is attributed mostly to "synchronized drops" and packet delays being misinterpreted as congestion. What's more, paced TCP sessions are less competitive when competing with non-paced TCP ones, because by spreading the packets evenly over the whole RTT, paced TCP sessions are more likely to encounter a packet drop then non-paced TCP ones [3]. On the other hand, TCP pacing is necessary for TCP sessions to achieve high

link utilization when the bottleneck router is equipped with small buffers [22, 76, 60, 30, 44].

From the above analyses, we summarize the conclusions on the impact of TCP pacing as follows.

1. Whether or not pacing helps improve the link utilization depends on the extent to which the packet drop rate is affected by the short-term burstiness. When the buffer size is moderate or large, short-term bursts are absorbed by the buffer and packet losses rarely occur. When the buffer size is small, packet drops occur very frequently due to buffer overflows. Since the short-term burstiness is reduced by TCP pacing, the packet drop probability decreases, and as a result, the link utilization increases.

2. Two factors prevent TCP pacing from widely being adopted in the current Internet: large buffer sizes and lower bandwidth shares of paced-TCP. Nowadays switches and routers on the Internet are equipped with buffers of a rule-of-thumb, that is, the bandwidth and delay product. Paced TCP does not outperform non-paced TCP in such a situation. Besides, considering the vast number of computers in the world and the diversity of operating systems running on these computers, it is unlikely to standardize and activate TCP pacing at all computers. Therefore, no one is willing to voluntarily enable pacing, suffering the weakened competition capability.

3. Pacing is critical for the future small buffer core networks to operate efficiently because in the case of small buffer networks the link utilization is significantly affected by the short-term burstiness.

The conclusions above motivate our work on proposing a blindly pacing algorithm that can be deployed at edges of small buffer core networks to ensure that the short-

term burstiness can be reduced to a certain low level for optical switches/routers to operate with high link utilizations.

1.3 Contributions Made in This Dissertation

The contributions made in this dissertation are two-fold.

First, we analyze the impact of burstiness on network performance from a queueing theory perspective. The framework developed in this part serves a theoretical foundation on which we show the benefits of pacing based. In particular, we show that there is a linear relationship between the parameters of the input process and the average queue lengths of a tandem queue system. Under certain mild conditions we show that decreasing inter-arrival times reduces average queue lengths.

Second, we propose a practical online packet-pacing algorithm, named queue length-based pacing (QLBP), to fulfill our goal towards a small buffer optical core network. This algorithm is designed to be applied on the aggregate traffic to reduce the short-term burstiness of the traffic. Unlike other existing policer or shaper suffering various restrictions, the proposed QLBP system is capable of reducing the burstiness of any traffic. It also overcomes the shortcomings of TCP pacing in that it is applied on the aggregate traffic, namely, it paces traffic blindly. With multiple QLBP pacers deployed on the Internet, network traffics are smoothed before they flow into the small buffer core networks. As a result, packet drop rates are reduced and link utilization is improved.

1.4 Organization of This Dissertation

The rest of this dissertation is organized as follows. In Chapter 2 we analyze the impact of the burstiness from the perspective of queueing theory. This chapter works as a theoretical foundation based on which the necessariness of pacing is justified. A practical online pacing algorithm, named queue length based pacing (QLBP), is pro-

posed in Chapter 3 as an effort towards the Internet-wide implementation of pacing. The effectiveness of the QLBP algorithm is analyzed via theoretical analyses and the benefit of pacing is demonstrated via simulation. In Chapter 4 we summarize the dissertation and present some interesting future work topics.

CHAPTER 2

LINEAR IMPACT OF PACING ON QUEUE LENGTHS

2.1 Introduction

In this chapter we analyze the impact of a general pacing scheme on the performance of a tandem queue network. The framework developed in this chapter serves a theoretical foundation on which we show the benefits of pacing based.

Classical queueing theory shows that bursty traffic degrades the performance of networks, increasing queueing delays, causing more packet drops and reducing link utilization [42]. It has been observed that the TCP (Transmission Control Protocol) congestion control mechanism can produce bursty traffic on high bandwidth-delay product link with heavily multiplexed flows [82].

The issue of bursty traffic becomes even more severe in the context of small-buffer networks (e.g., network with all-optical routers). On one hand, the growing demand for raw bandwidth motivates the interest in all-optical routers. Using fluid and queueing models, authors in [61] and [60] argue that small buffers provide greater network stability. Experimental studies and analytical results in [22] and [44] show that with buffer sizes of $\Theta(\log(W))$ the congested link can operate at 75% or higher utilization (where W is congestion window size). On the other hand, high link utilization is achieved through traffic pacing. By "traffic pacing" we mean a scheme that spreads packet bursts over surrounding idle periods while keeping long-term average rate unchanged. As pointed out in [22] and [44], to achieve high link utilization it is necessary to pace TCP flows either by exploiting explicit pacing schemes at end hosts or by limiting the speed of their access links. Our work is motivated by these

observations, and in this chapter we study the impact of arrival traffic burstiness on network performance.

Our focus is to show that there exists a linear relationship between traffic burstiness and queue length statistics for a tandem queue network with infinite buffers. Our approach is based on the analysis of the sample-path derivatives of the queue length with respect to system parameters, and it uses infinitesimal perturbation analysis (IPA) technique developed in the '80s [32].

The model used in this chapter is a tandem queue network with infinite buffers. We consider the aggregate traffic arriving at a network core router and model it as a marked point process. Each marked point arrival represents a sequence of back-to-back TCP packets. The size of this burst represents the workload of each marked point. The inter-arrival time and workload distributions have scale parameters. The average input load is defined as the product of the average arrival rate (i.e., the inverse of the average inter-arrival time) and the average workload. Under such a framework we study the impact of arrival bursts on the statistics of a tandem queue network. In particular, we show that there is a linear relationship between the queue lengths and the system parameters.

The impact of arrival bursts has been studied in the context of fluid models. Using Markov On-Off model, Brocket et al. derived the average queue lengths of a tandem queue in [12]. Later, the impact of the autocorrelation carried by a flow was studied with a hierarchical On-Off fluid model in [36]. Liu and Gong applied perturbation analysis on the statistics of a tandem queue network using fluid models [48]. The analytical results obtained in those works support our conclusion in their own problem settings.

The contributions of this chapter are two-fold:

- We show that a tandem queue network that is fed by a marked point process with inter-arrival time and workload distributions with scale parameters ex-

hibits a linear relationship between the queue lengths and the distribution scale parameters when the average input load is fixed.

- We derive the IPA estimator of the derivative of the average queue length with respect to the average inter-arrival time, and show that it is unbiased and strongly consistent under the assumption that the average input load is constant.

The rest of this chapter is organized as follows. In Section 2.2, we introduce the tandem queue network model and derive expressions for the instantaneous and average queue lengths based on a sample path of the arrival point process. We derive the IPA estimators of the derivatives of average queue lengths with respect to the average inter-arrival time, introduce the assumptions and conditions that form the basis of our analysis and show unbiasedness and strong consistency of the IPA estimators in Section 2.3. In Section 2.4 we show there is a linear relationship between instantaneous/average queue lengths and the inter-arrival parameters under the given assumptions and conditions. Finally, we draw our conclusions in Section 2.6.

2.2 Modeling and Analysis

Currently, there are two basic approaches to analyzing queueing models: classical queueing theory and stochastic fluid models. Classical queueing theory focuses on the packet-level behavior of a queueing system with certain assumptions on the distributions of inter-arrival time and service time. It is widely used in the performance evaluation of network protocols [15, 16, 28, 44]. Stochastic fluid models treat packet arrival bursts as a continuous fluid. As demonstrated in [12] and [36], Markov On-Off fluid models are able to capture the impact of source correlation on the average queue size.

In this chapter, we work with a combination of these two models. The input of the first queue is a point process in which every impulse carries a workload representing

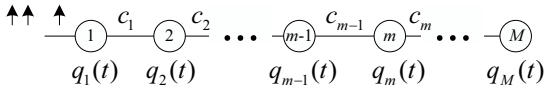


Figure 2.1. A M node tandem queue topology

a burst of packets. Whenever there are buffered packets, a queue has an outgoing flow at a constant rate. We further extend our study to consider tandem queues with infinite size buffers. The tandem queue topology is meaningful since a connection over Internet usually goes over multiple hops. The first portion of this path (before reaching the bottleneck link) encounters decreasing link capacities, which are often due to multiplexing with other flows. Pacing of traffic implies that system parameters may be perturbed due to the effect pacing, but the average input load remains unchanged.

In what follows we first introduce the notation and then derive expressions for the instantaneous and average queue lengths.

2.2.1 Network Model and Notation

Fig. 2.1 shows a tandem queue network fed with a point process in which the inter-arrival times and workloads follow scale parameters θ and ξ. In this model c_i $(1 \leq 1 \leq M-1)$ is the outgoing capacity of node i and $q_i(t)$ is the instantaneous queue length of node i at time t. We denote by X and Y the generally distributed intensity of an arrival impulse, i.e., the workload carried by a customer and the generally distributed inter-arrival time. When an impulse arrives at node 1, it contributes an instant queue increment. The nodes in the network are of decreasing capacities, which implies that whenever a node has a queue built up, so does its downstream nodes.

We define in Table 2.1 the variables and symbols that are used to describe the dynamics of the tandem queue network. To make them easier to understand, they are categorized according to the objects they serve.

Table 2.1. Major notation in Chapter 2

For the arrival point process	
i	index of customer $i \geq 1$
A_i	arrival time of i-th customer
D_i	departure time of i-th customer from node 1
X_i	i-th customer's workload, and $X =_{s.d.} X_i$ with $E[X] = \xi$
Y_i	inter-arrival time between $(i-1)$-th and i-th customers, and $Y =_{s.d.} Y_i$ with $E[Y] = \theta$
Z_i	i-th customer's service time at node 1 $(= X_i/c_1)$
Ξ	a compact set, which ξ belongs to
Θ	a compact set, which θ belongs to
For node $m = 1, ..., M$	
$q_m(t)$	instantaneous queue length of node m at t
c_m	capacity of node m
j	index of $q_m(t)$'s busy period
τ_j^m	duration time of $q_m(t)$'s first j busy periods[1]
L_j^m	integral of $q_m(t)$ over $[0, \tau_j^m)$
l_j^m	average queue length of node m over $[0, \tau_j^m)$
l^m	average queue length of node m in steady state
For busy period j of $q_m(t)$	
A_j^m	beginning time of j-th busy period of $q_m(t)$
D_j^m	end time of j-th busy period of $q_m(t)$
B_j^m	duration time of $q_m(t)$'s j-th strictly ascending phase (for $m > 1$)
I_j^m	duration time of $q_m(t)$'s j-th strictly descending phase (for $m > 1$)
$n_{C,j}^m$	index of last customer arrival within j-th busy period of $q_m(t)$ $(n_{C,0}^m = 0)$
$n_{B,j}^m$	index of last busy period of $q_m(t)$ covered within j-th busy period of $q_{m+1}(t)$ $(n_{B,0}^m = 0)$

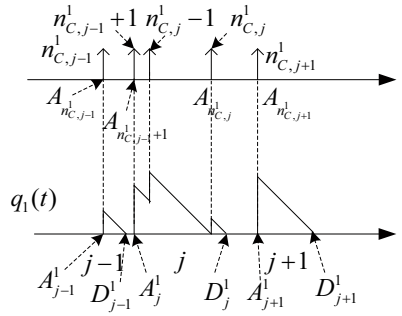


Figure 2.2. A sample path of $q_1(t)$

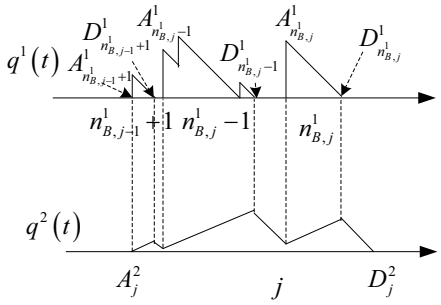


Figure 2.3. A sample path of $q_2(t)$

2.2.2 Instantaneous Queue Lengths of Nodes

We now derive expressions for instantaneous queue lengths based on a given sample path S, which is a sequence of pairs of (X_i, Y_i), denoted as $\{(X_i, Y_i), i > 0\}$, where X_i's and Y_i's are instants of X and Y, respectively.

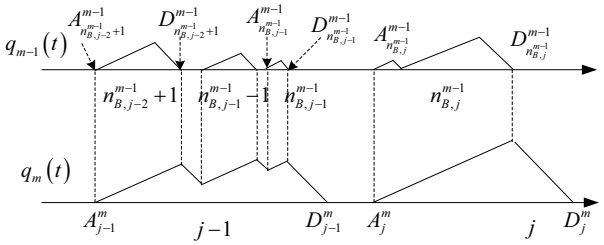


Figure 2.4. A sample path of $q_m(t)$

2.2.2.1 Instantaneous Queue Length $q_1(t)$

q_1 consists of alternate busy and idle periods and each busy period covers one or more customer arrivals. With the notation of $n^1_{C,j}$ (i.e., $n^m_{C,j}$ for $m = 1$), the first customer covered within $q_1(t)$'s j-th busy period is indexed $n^1_{C,j-1}+1$. Fig. 2.2 shows a piece of sample path $q_1(t)$. Thus, $q_1(t)$ is recursively expressed by

$$q_1(t) = \begin{cases} 0, & \text{for } t \in [0, A_1); \\ Z_{n^1_{C,j-1}+1}, & \text{for } t = A_{n^1_{C,j-1}+1}; \\ (\text{for } n^1_{C,j-1}+1 \le i \le n^1_{C,j}, j \ge 1) & \\ q_1(A_i) - c_1(t - A_i), & \text{for } t \in (A_i, A_{i+1}), \\ & n^1_{C,j-1}+1 \le i \le n^1_{C,j}-1; \\ q_1(A_i) - c_1 Y_{i+1} + Z_{i+1}, & \text{for } t = A_{i+1}, n^1_{C,j-1}+1 \le i \le n^1_{C,j}-1; \\ q_1(A_{n^1_{C,j}}) - c_1(t - A_{n^1_{C,j}}), & \text{for } t \in [A_{n^1_{C,j}}, D^1_{n^1_{C,j}}); \\ 0, & \text{for } t \in [D^1_{n^1_{C,j}}, A_{n^1_{C,j}+1}). \end{cases} \tag{2.1}$$

where

$$\begin{cases} A_i = \sum_{j=1}^{i} Y_j, \\ D_{n^1_{C,j}} = \sum_{j=1}^{n^1_{C,j-1}+1} Y_j + \sum_{j=n^1_{C,j-1}+1}^{n^1_{C,j}} Z_j. \end{cases}$$

2.2.2.2 Instantaneous Queue Length $q_m(t)$ for $m = 2, ..., M$

Fig. 2.3 and Fig. 2.4 illustrate how $q_m(t)$ changes as $q_{m-1}(t)$ goes. An important observation is that $q_m(t)$ increases during busy periods of $q_{m-1}(t)$ and decreases during idle periods of $q_{m-1}(t)$ until it becomes empty. Thus, $q_m(t)$ is recursively expressed by

$$q_m(t) = \begin{cases} 0, & \text{for } t \in [0, A_1^{m-1}]; \\ (\text{for } n_{B,j-1}^{m-1} + 1 \le i \le n_{B,j}^{m-1}, j \ge 1) \\ q_m(A_i^{m-1}) + (c_{m-1} - c_m)(t - A_i^{m-1}), \\ \quad \text{for } t \in [A_i^{m-1}, D_i^{m-1}), n_{B,j-1}^{m-1} + 1 \le i \le n_{B,j}^{m-1}; \\ q_m(D_i^{m-1}) - c_m(t - D_i^{m-1}), \\ \quad \text{for } t \in [D_i^{m-1}, A_{i+1}^{m-1}), n_{B,j-1}^{m-1} + 1 \le i \le n_{B,j}^{m-1} - 1; \\ q_m(D_i^{m-1}) - c_m(t - D_i^{m-1}), \\ \quad \text{for } t \in [D_i^{m-1}, \frac{q_m(D_i^{m-1})}{c_m}), i = n_{B,j}^{m-1}. \end{cases} \tag{2.2}$$

In the equations above $A_{(\cdot)}^{m-1}$ and $D_{(\cdot)}^{m-1}$ remain to be defined. $A_{(\cdot)}^{m-1}$ and $D_{(\cdot)}^{m-1}$ are recursively expressed in terms of $q_{m-1}(t)$, $A_{(\cdot)}^{m-2}$ and $D_{(\cdot)}^{m-2}$. Without loss of generality, we derive the expressions for $A_{(\cdot)}^{m}$ and $D_{(\cdot)}^{m}$ instead of $A_{(\cdot)}^{m-1}$ and $D_{(\cdot)}^{m-1}$. Another important observation is that the beginning times of the j-th busy period of $q_m(t)$ and the $n_{B,j-1}^{m-1}+1$-th busy period of $q_{m-1}(t)$ coincide with each other, i.e. $A_j^m = A_{n_{B,j-1}^{m-1}+1}^{m-1}$. Also $D_j^m = \frac{q_m\left(D_{n_{B,j}^{m-1}}^{m-1}\right)}{c_m}$ where $q_m\left(D_{n_{B,j}^{m-1}}^{m-1}\right)$ is determined by $c_m, c_{m-1}, A_j^{m-1}, D_j^{m-1}$ for $j < n_{B,j}^{m-1}$. Thus, A_j^m and D_j^m are given by

$$\begin{cases} A_j^m & = A_{n_{B,j-1}^{m-1}+1}^{m-1}, \\ D_j^m & = \frac{1}{c_m} q_m\left(D_{n_{B,j}^{m-1}}^{m-1}\right). \end{cases}$$

With $A_{(\cdot)}^1$ and $D_{(\cdot)}^1$ given by

$$\begin{cases} A_j^1 & = A_{n_{C,j-1}^1+1} = \sum_{k=1}^{n_{C,j-1}^1+1} Y_k, \\ D_j^1 & = \sum_{k=n_{C,j-1}^1+1}^{n_{C,j}^1} Z_k + \sum_{k=1}^{n_{C,j-1}^1+1} Y_k. \end{cases}$$

2.2.3 Average Queue Lengths of Nodes

Next we derive an expression of the average queue length of $q_m(t)$ over its first B busy periods, denoted as l_B^m. By definition, it is given by

$$l_B^m = \frac{L_B^m}{\tau_B^m}. \tag{2.3}$$

2.2.3.1 Average Queue Length l_B^1

The definition of average queue length leads to

$$l_B^1 = \frac{L_B^1}{\tau_B^1}, \tag{2.4}$$

where

$$L_B^1 = c_1 \sum_{b=1}^{B} \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i \Bigg(\sum_{j=n_{C,b-1}^1+1}^{i-1} Z_j - \sum_{j=n_{C,b-1}^1+2}^{i} Y_j \Bigg) + \frac{c_1}{2} \sum_{i=1}^{n_{C,B}^1} Z_i^2 \tag{2.5}$$

and

$$\tau_B^1 = \sum_{j=1}^{n_{C,B}^1+1} Y_j. \tag{2.6}$$

2.2.3.2 Average Queue Length l_B^m for $m \geq 2$

l_B^m is defined by

$$l_B^m = \frac{L_B^m}{\tau_B^m}, \tag{2.7}$$

where

$$L_B^m = \sum_{b=1}^{B} \sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2} \tag{2.8}$$

and

$$\tau_B^m = \sum_{j=1}^{n_{C,B}^m+1} Y_j. \tag{2.9}$$

In the equations above, v_j^m, p_j^m, B_j^m and I_j^m for $n_{B,b-1}^{m-1}+1 \leq j \leq n_{B,b}^{m-1}, b \geq 1$ are given by

$$\begin{cases} v_j^m = q_m(A_j^{m-1}), & n_{B,b-1}^{m-1}+1 \leq j \leq n_{B,b}^{m-1}; \\ v_{j+1}^m = 0, & j = n_{B,b}^{m-1} \\ p_j^m = q_m(D_j^{m-1}), & n_{B,b-1}^{m-1}+1 \leq j \leq n_{B,b}^{m-1}; \\ B_j^m = \sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}} Z_k, & n_{B,b-1}^{m-1}+1 \leq j \leq n_{B,b}^{m-1}; \\ I_j^m = \sum_{k=n_{C,j-1}^{m-1}+2}^{n_{C,j}^{m-1}+1} Y_k - B_j^m, & n_{B,b-1}^{m-1}+1 \leq j \leq n_{B,b}^{m-1}-1; \\ I_j^m = \frac{q_m(D_j^{m-1})}{c_m}, & j = n_{B,b}^{m-1}. \end{cases} \tag{2.10}$$

2.3 IPA on Derivatives of Average Queue Lengths with Respect to System Parameters

In this section we derive IPA estimators for the derivative of l_B^m (for $m = 1, ..., M$) with respect to system parameter θ. Throughout the rest of the chapter, the ratio between ξ and θ is fixed, which reflects the principle of pacing.

We first derive the IPA estimators and then prove their unbiasedness and strong consistency under the given assumptions and conditions.

2.3.1 IPA Estimators of l_B^m's Derivative with respect to System Parameters

2.3.1.1 For Node 1

We start with

$$
\begin{aligned}
&\frac{dl_B^1(\theta)}{d\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{\Delta l_B^1(\theta)}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{l_B^1(\theta+\Delta\theta) - l_B^1(\theta)}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{1}{\Delta\theta}\left(\frac{L_B^1(\theta+\Delta\theta)}{\tau_B^1(\theta+\Delta\theta)} - \frac{L_B^1(\theta)}{\tau_B^1(\theta)}\right) \\
&= \frac{\lim_{\Delta\theta\to 0}\frac{\Delta L_B^1(\theta)}{\Delta\theta}\tau_B^1(\theta) - L_B^1(\theta)\lim_{\Delta\theta\to 0}\frac{\Delta\tau_B^1(\theta)}{\Delta\theta}}{(\tau_B^1(\theta))^2}
\end{aligned}
$$

Let $\Delta Z_i = Z_i(\theta+\Delta\theta) - Z_i(\theta)$ and $\Delta Y_i = Y_i(\theta+\Delta\theta) - Y_i(\theta)$. Substituting them into the equation above, we obtain

$$
\begin{aligned}
\frac{dl_B^1}{d\theta} = \frac{c_1}{(\tau_B^1)^2}\Bigg[\sum_j^{n_{C,B}^1+1} Y_j \Bigg(&(\sum_b^B \sum_i^{n_{C,b}^1} \frac{dZ_i}{d\theta} \sum_j^{i-1} Z_j + \sum_b^B \sum_i^{n_{C,b}^1} Z_i \sum_j^{i-1} \frac{dZ_j}{d\theta}) - \\
&(\sum_b^B \sum_i^{n_{C,b}^1} \frac{dZ_i}{d\theta} \sum_j^{i} Y_j + \sum_b^B \sum_i^{n_{C,b}^1} Z_i \sum_j^{i} \frac{dY_j}{d\theta}) + \sum_i^{n_{C,B}^1} Z_i \frac{dZ_i}{d\theta}\Bigg) - \\
&\Bigg(\sum_b^B \sum_i^{n_{C,b}^1} Z_i \sum_j^{i-1} Z_j - \sum_b^B \sum_i^{n_{C,b}^1} Z_i \sum_j^{i} Y_j + \frac{1}{2}\sum_i^{n_{C,B}^1} Z_i^2\Bigg) \sum_j^{n_{C,B}^1+1} \frac{dY_j}{d\theta}\Bigg].
\end{aligned} \tag{2.11}
$$

Equation (2.11) is an IPA estimator of $\frac{dl_B^1}{d\theta}$.

2.3.1.2 For Nodes 2 to M

We start with

$$
\begin{aligned}
&\frac{dl_B^m(\theta)}{d\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{\Delta l_B^m(\theta)}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{l_B^m(\theta+\Delta\theta) - l_B^m(\theta)}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{1}{\Delta\theta}\left(\frac{L_B^m(\theta+\Delta\theta)}{\tau_B^m(\theta+\Delta\theta)} - \frac{L_B^m(\theta)}{\tau_B^m(\theta)}\right) \\
&= \frac{\lim_{\Delta\theta\to 0}\frac{\Delta L_B^m(\theta)}{\Delta\theta}\tau_B^m(\theta) - L_B^m(\theta)\lim_{\Delta\theta\to 0}\frac{\Delta\tau_B^m(\theta)}{\Delta\theta}}{(\tau_B^m(\theta))^2}
\end{aligned}
$$

Let $\Delta v_i^m = v_i^m(\theta+\Delta\theta) - v_i^m(\theta)$, $\Delta p_i^m = p_i^m(\theta+\Delta\theta) - p_i^m(\theta)$, $\Delta B_i^m = B_i^m(\theta+\Delta\theta) - B_i^m(\theta)$ and $\Delta I_i^m = I_i^m(\theta+\Delta\theta) - I_i^m(\theta)$. Substituting them into the equation above, we have

$$\begin{aligned}\frac{dl_B^m}{d\theta} = \frac{1}{(\tau_B^m)^2}\Bigg[\sum_j^{n_{C,B}^m+1} Y_j \Bigg(\sum_b^B \sum_j^{n_{B,b}^{m-1}} \cdot \frac{(\frac{d}{d\theta}v_j^m + \frac{d}{d\theta}p_j^m)B_j^m + (\frac{d}{d\theta}p_j^m + \frac{d}{d\theta}v_{j+1}^m)I_j^m}{2} + \\ \sum_b^B \sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)\frac{d}{d\theta}B_j^m + (p_j^m + v_{j+1}^m)\frac{d}{d\theta}I_j^m}{2} \Bigg) - \\ \Bigg(\sum_b^B \sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2} \Bigg) \sum_j^{n_{C,B}^m+1} \frac{d}{d\theta}Y_j \Bigg].\end{aligned} \tag{2.12}$$

Also the lower bound of every summation index in (2.12) is skipped, which can be retrieved by referring to (2.7), (2.8) and (2.9). Equation (2.12) is an IPA estimator of $\frac{dl_B^m}{d\theta}$.

2.3.2 Assumptions and Conditions

It is pointed out in [78] (e.g. see page 419) that for a GI/G/1 system with average arrival rate λ and average service rate μ to be stable, a sufficient condition is that $\lambda < \mu$ [47, 49]. For our tandem queue network to be stable, we have A. 2.3.1.

Assumption 2.3.1. ***Stability***. *It is assumed that* $c_m > \frac{\xi}{\theta}$, *for* $m = 1, ..., M$.

However, for the average queue length defined to exist, we need a stricter constraint, which is presented in A. 2.3.2. A similar assumption is made in [29] (e.g. see Theorem 8.3 in [29]).

Assumption 2.3.2. ***Regenerativeness***. *It is assumed that* $q_m(t)$ *is regenerative with a sequence* $\{\sigma_j^m, j \geq 1\}$, *where* σ_j^m *is the end time of the* j*-th busy period of* $q_m(t)$ *and it also holds that* $E[\sigma_{j+1}^m - \sigma_j^m] < \infty$, *and* $E[(\sigma_{j+1}^m - \sigma_j^m)^2] < \infty$.

Assumption 2.3.3. ***Continuity*** *For each $\xi \in \Xi$, the cumulative density function of X, $F_X(x, \xi)$ is a.s. continuous in x and zero at $x = 0$. For each $\theta \in \Theta$, the cumulative density function of Y, $F_Y(y, \theta)$ is a.s. continuous in y and zero at $y = 0$.*

Assumption 2.3.4. ***Differentiability*** *For each $\xi \in \Xi$ and each i, $X_i(\xi)$ is, with probability one, a continuously differentiable function of ξ in Ξ. For each $\theta \in \Theta$ and each i, $Y_i(\theta)$ is, with probability one, a continuously differentiable function of θ in Θ.*

Condition 2.3.1. ***Fixed Input Load Average****. The average input load is fixed, that is, $\frac{\xi}{\theta} = C$ (a constant).*

Condition 2.3.2. ***Scale Parameters****. The distributions of the inter-arrival times and the workloads have scale parameters θ and ξ, i.e. $\frac{dY}{d\theta} = \frac{Y}{\theta}$, and $\frac{dX}{d\xi} = \frac{X}{\xi}$.*

Two lemmas below are useful in proving unbiasedness and strong consistency of the IPA estimators.

Lemma 2.3.1. *Under C. 2.3.1 and C. 2.3.2, it holds that $\frac{dZ}{d\theta} = \frac{Z}{\theta}$, where $Z =_{s.d.} Z_i$.*

The proof is provided in Appendix A.

Lemma 2.3.2. *Under A. 2.3.3, A. 2.3.4, C. 2.3.1 and C. 2.3.2, it holds that for $m = 2, ..., M$ and $j \geq 1$, $\frac{dv_j^m}{d\theta} = \frac{v_j^m}{\theta}$, $\frac{dp_j^m}{d\theta} = \frac{p_j^m}{\theta}$, $\frac{dB_j^m}{d\theta} = \frac{B_j^m}{\theta}$, $\frac{dI_j^m}{d\theta} = \frac{I_j^m}{\theta}$.*

The proof is provided in Appendix A.

2.3.3 Unbiasedness and Strong Consistency

Even though we are aware that IPA does not give unbiased derivative estimators for traditional discrete-event multiple-class queues, the IPA estimators in our problem setting are unbiased. They are also strongly consistent.

Lemma 2.3.3. *Under A. 2.3.3, A. 2.3.4, C. 2.3.1 and C. 2.3.2, for any $\theta \in \Theta$, l_B^m is, with probability one, continuously differentiable in θ.*

The proof is provided in Appendix A.

With Lemma 2.3.3, we are ready to show the unbiasedness and strong consistency of the derived IPA estimators.

2.3.3.1 Unbiasedness

Theorem 2.3.1. *Under A. 2.3.1-2.3.4 and C. 2.3.1, C. 2.3.2, and with Lemmas 2.3.1-2.3.3, for any $\theta \in \Theta$, the IPA estimators of $\frac{dl^m(\theta)}{d\theta}$ given by* (2.11) *and* (2.12) *are unbiased, i.e.*

$$E[\frac{dl_B^m(\theta)}{d\theta}] = \frac{d}{d\theta}E[l_B^m(\theta)], for\ m = 1, ..., M. \tag{2.13}$$

The detailed proof is provided in Appendix A. Here we briefly sketch the proof procedure. For instance, when $m = 1$, we first shown l_B^1 is a.s. continuous under A. 2.3.3, A. 2.3.4 and C. 2.3.1. Then, with A. 2.3.2 we show $E[\sup_{\theta\in\Theta} |l_B^1(\theta)/\theta] < \infty$. Applying the Generalized Mean Value Theorem and the Dominated Convergence Theorem (e.g. see page 14 and 15 in [29]), we show the inter-exchange of expectation and derivative holds, which leads to Theorem 2.3.1.

2.3.3.2 Strong Consistency

Theorem 2.3.2. *With A. 2.3.1-2.3.4 and under C. 2.3.1, C. 2.3.2, Lemmas 2.3.1-2.3.3, for any $\theta \in \Theta$, the IPA estimators of $\frac{dl^m(\theta)}{d\theta}$ given by* (2.11) *and* (2.12) *are strongly consistent, i.e.*

$$\lim_{B\to\infty} \frac{dl_B^m(\theta)}{d\theta} = \frac{dl^m(\theta)}{d\theta}, for\ m = 1, ..., M. \tag{2.14}$$

The proof takes advantage of A. 2.3.1 and A. 2.3.2 to show the unbiasedness of derivatives of $\widetilde{L}_b^m$ and $\widetilde{T}_b^m$ with θ under C. 2.3.1 and C. 2.3.2, where $\widetilde{L}_b^m$ and $\widetilde{T}_b^m$ are defined as the integral and the duration time of $q_m(t)$'s b-th busy period. Based on their unbiasedness properties, the strong consistency is proven with A. 2.3.2. The proof is provided in Appendix A.

2.4 Linear Impact

An important conclusion from these results is that there is a linear relationship between queue lengths and system parameters under the given conditions. By "linear relationship" we mean that the queue lengths change linearly proportionally to the average inter-arrival time. This is revealed by showing that the derivative of queue lengths with respect to the average inter-arrival time is a constant.

Assume that function $f(x)$ is continuously differentiable in x. $df(x)/dx = f(x)/x$ if and only if $df(x)/dx = C$ (const.). A simple proof is as follows. First, it is trivial that $f(x) = Cx \Rightarrow df(x)/dx = f(x)/x$. Second, for $df(x)/dx = f(x)/x \Rightarrow f(x) = Cx$, we have $df(x)/dx = f(x)/x \Rightarrow df(x)/f(x) = dx/x \Rightarrow \int 1/f(x)df(x) = \int 1/x dx \Rightarrow \ln f(x) = \ln x + \ln C \Rightarrow f(x) = Cx$.

2.4.1 Linear Impact on Instantaneous Queue Lengths

Theorem 2.4.1. *Under A. 2.3.3, A. 2.3.4, C. 2.3.1 and C. 2.3.2, it holds that for* $i \geq 1$,

$$\frac{dq_1(A_i(\theta), \theta)}{d\theta} = \frac{q_1(A_i(\theta), \theta)}{\theta}.$$

The proof is provided in Appendix A.

Theorem 2.4.2. *Under A. 2.3.3, A. 2.3.4, C. 2.3.1 and C. 2.3.2, it holds that for* $m = 2, ..., M$ *and* $j \geq 1$,

$$\frac{dq_m(D_j^{m-1}(\theta), \theta)}{d\theta} = \frac{q_m(D_j^{m-1}(\theta), \theta)}{\theta}.$$

The proof is provided in Appendix A.

Remark: Theorems 2.4.1 and 2.4.2 indicate that under the given assumptions and conditions the sample path peak values change linearly proportionally to the average inter-arrival time.

2.4.2 Linear Impact on Average Queue Lengths

Theorem 2.4.3. *Under A. 2.3.3, A. 2.3.4, C. 2.3.1 and C. 2.3.2, we have for* $m = 1, ..., M$ *and* $B \geq 1$,

$$\frac{dl_B^m}{d\theta} = \frac{l_B^m}{\theta}. \tag{2.15}$$

It follows that for $m = 1, ..., M$,

$$\frac{dl^m}{d\theta} = \frac{l^m}{\theta}. \tag{2.16}$$

The proof is provided in Appendix A.

Remark: Equation (2.16) unveils the linear relationship between the average queue lengths and the inter-arrival time under the given assumptions and conditions.

If a pacing scheme is able to make such changes, i.e. to keep the same distribution but to generate smaller average inter-arrival time and workload, then consequently the instantaneous and average queue lengths are lower. In practice, that means that small-buffer networks can operate more efficiently when pacing is used.

2.5 Simulation Validation

2.5.1 Experiment Setup

In this section, we validate this linear relationship between the instantaneous/average queue lengths and the average inter-arrival time, using simulations. The topology used in experiments is the same as shown in Fig. 2.1, but with only three nodes. Their outgoing capacities are set $c_1 = 1$ units/s, $c_2 = 0.81$ units/s, $c_3 = 0.75$ units/s. The units of ξ and θ are "units/customer" and "seconds," respectively. We have developed a C program to simulate the dynamics of this tandem queue network and collect data.

2.5.2 Linear Impact on Average Queue Lengths

We first simulate the impact of parameter perturbation on the average queue lengths. We run simulations with the inter-arrival time and workload distributions with and without scale parameters. Each simulation run with the same parameter settings lasts for $100,000$ busy periods and repeats 30 times to obtain the average.

2.5.2.1 Exponential inter-arrival time and workload distributions

We run simulations with a point process with Exponential inter-arrival time and workload distributions. θ changes from 1.5s to 15s with a step of 1.5s, and $\xi = 2\theta/3$. Fig. 2.5 shows the derivative of $E[q_i]$ (either simulated or theoretical) with respect to θ is a constant, where $i = 1, 2, 3$, which confirms that this linear relationship exists. We also calculate an 95% confidential interval for each group of 30 simulated values, which shows very small variation around the average. For instance, simulated $E[q_3(\theta, t)]$ at $\theta = 15$ is 33.609 units with an 95% confidential interval of 0.606.

2.5.2.2 Triangular inter-arrival time and workload distributions

We next run simulation with a point process with Triangular inter-arrival time and workload distributions. Triangular distribution is briefly introduced below. For details of Triangular distribution, please refer to [66].

The cumulative density function (CDF) of Triangular distribution with lower limit a, mode c and upper limit b is given by

$$F_{X|a,b,c}(x) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)}, & \text{for } a \leq x \leq c \\ 1 - \frac{(b-x)^2}{(b-c)(b-a)}, & \text{for } c \leq x \leq b \\ 0, & \text{for } x < a \\ 1, & \text{for } x > b \end{cases}$$

Its first and second moments are given by

$$\begin{cases} E[X] = \frac{a+b+c}{3}; \\ E[X^2] = \frac{a^2+b^2+c^2-ab-bc-ca}{18} + \frac{(a+b+c)^2}{9}. \end{cases}$$

Whether Triangular distribution has the average as scale parameter or not depends on how to set up its parameters. A simple way to make it have scale parameter is to set $a = 0, b = 2c$. Suppose the Triangular inter-arrival time and workload distributions have parameters (a_ξ, b_ξ, c_ξ) and $(a_\theta, b_\theta, c_\theta)$, respectively. Our settings are as follows. $a_\xi = a_\theta = 0$, $c_\xi = 1$ and $c_\theta = 1.5$. $b_\xi = 2.2+0.2*i$ and $b_\theta = 8.3+0.3*i$ for $i = 1, ..., 10$, where i represents the index of each simulation run. With these settings, the inter-arrival and workload distributions do not have the average as scale parameters. Fig. 2.6 shows the derivative of the average queue length of node i with respect to θ for $i = 1, 2, 3$ is no longer a constant. To make the non-linearity more visible, we draw two dotted lines along $E[q_1]$, $E[q_2]$ and $E[q_3]$. The differences between the dotted lines and $E[q_1], E[q_2]$ and $E[q_3]$ indicate that $E[q_i]$ $i = 1, 2, 3$ are not straight lines. Also simulated $E[q_1(\theta, t)]$ at $\theta = 4.17$ is 0.508 units with an 95% confidential interval of $7.859 * 10^{-4}$.

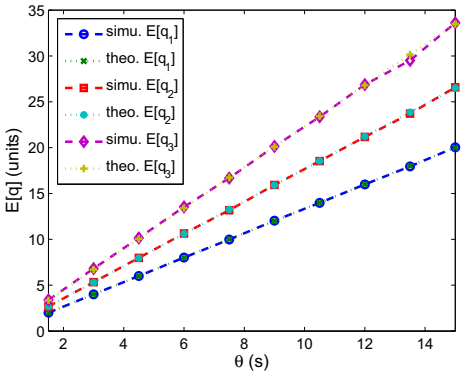


Figure 2.5. Average queue length with exponential inter-arrival time and workload distributions

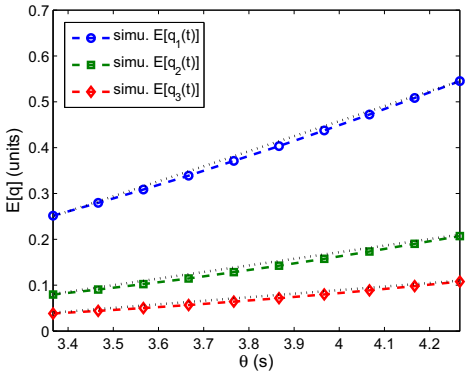


Figure 2.6. Average queue length with triangular inter-arrival time and workload distributions

2.5.3 Impact of Pacing Scheme

In this sub-section we simulate the impact of an ideal pacing scheme in which each marked point is divided into two equal pieces when it arrives at node 1. The first half is sent at its original arrival time while the second one is sent in the middle of the consecutive inter-arrival time. Such a pacing scheme is ideal in the sense that it can only be implemented in simulation with prior knowledge about the consecutive inter-arrival time.

2.5.3.1 Exponential inter-arrival time and workload distributions

The parameter settings are the same as those in the preceding experiments with Exponential inter-arrival time and workload distributions. Every marked point is paced by the pacing scheme at node 1 before being processed. Fig. 2.7 shows that with different θ's, this pacing scheme can reduce the average queue lengths by half.

2.5.3.2 Triangular inter-arrival time and workload distributions

The parameter settings are the same as those in the preceding experiments with Triangular inter-arrival time and workload distributions. Even for Triangular inter-arrival and workload distributions, the pacing scheme can still achieve the similar

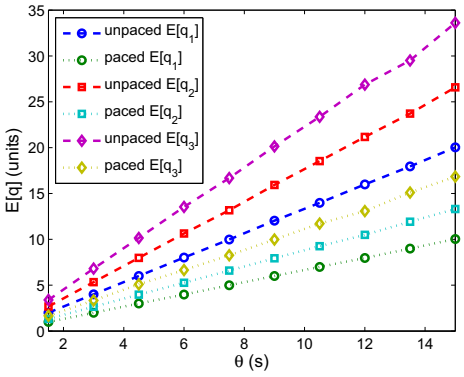


Figure 2.7. Comparison of $E[q]$ with exponential inter-arrival time and workload distributions

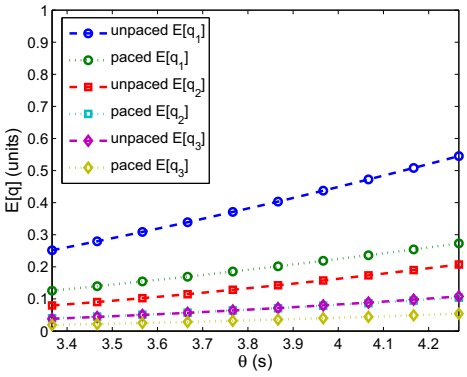


Figure 2.8. Comparison of $E[q]$ with angular inter-arrival time and workload distributions

performance improvement. Fig. 2.8 shows that under the pacing scheme, the average queue lengths are approximately reduced by half for various θ.

2.6 Summary

To meet the increasing demands for network bandwidth, optical core networks are being deployed. Due to technology limitations, buffering of traffic in all-optical routers is very costly. Therefore, router designs with small packet buffers are emerging as infrastructure components in next-generation networks. Network traffic pacing plays an important role in improving the operational efficiency and performance of these small-buffer networks.

In this chapter we investigated the potential benefits of traffic pacing by quantitatively studying the impact of traffic burstiness on the buffer occupancies of a tandem queue network fed with a point process. The results of our work are:

1. We derive an expressions for the instantaneous and average queue lengths of a tandem queue network from a sample-path perspective;

2. Under mild and reasonable assumptions on traffic arrivals and workload patterns, we develop the IPA estimators for average queue lengths and also show the unbiasedness and strong consistency of them;

3. We show under the given conditions that the arrival traffic burstiness has a linear impact on both instantaneous and average queue lengths of all queues in a tandem network, which demonstrates that traffic pacing has great potential to reduce buffer occupancies and largely improve the packet loss and delay performance in communication networks with small buffers.

CHAPTER 3

QUEUE LENGTH BASED PACING

3.1 Introduction

Many data communication networks use a layered network architecture, where each layer implements different networking protocols [39]. The separation of networking functionality into layers simplifies the design of network protocols as each layer can rely on the services provided by the underlying layer. This functional dependency also implies that the performance that can be achieved within a protocol layer is highly dependent on the performance achieved by underlying layers. Specifically, the performance of transport layer protocols, which provide process-to-process communication between end-systems, relies on the performance achieved by interface-to-interface packet delivery in the network layer.

In our work, we discuss how to improve the throughput performance of transport layer protocols by adjusting the operation of the network at the network layer. The main idea is to adjust the characteristics of network traffic at the edge of the network to ensure better performance in the core of the network. Specifically, we propose to introduce intentional delay in network layer transmissions to reduce the occurrence of traffic bursts, which have detrimental effects on transport layer performance as they can lead to packet loss due to buffer overflow. Our focus is on networks with small packet buffers (e.g., all-optical packet-switched networks, wireless networks with low-performance nodes) [63].

3.1.1 Packet Loss in Networks

One of the most problematic events for data transmissions in the network layer is a packet loss. The two main causes for packet loss in networks are:

- Bit errors in the physical layer: Bit errors in the physical layer most commonly occur in wireless transmissions due to interference, but can also occur in wired links. These bit errors cause checksums in the data link layer to fail, triggering a packet drop.

- Congestion in the network layer: Statistical multiplexing of network traffic implies that there are no guarantees about the available bandwidth on any given link. Thus, network traffic can congest the outgoing port of a router and cause transmission buffers to fill up. If a packet arrives at such a transmission queue when no more buffer space is available, then it is dropped.

While these causes of packet loss are fundamentally different, their effects result in the same performance degradation in the transport layer.

In practice, many applications require reliable (i.e., lossless) data transfer. While some applications can compensate for lost data in the application layer, lossy transmission are only useful in very specific application domains (e.g., video playback). To recover from a loss event, the transport layer initiates a retransmission of the lost packet. This is a problematic solution for applications where data needs to be delivered with low delay (e.g., cyber-physical control, online gaming, etc.), since retransmission of a packet can incur considerable delay (time to discover loss plus one round-trip time). Therefore, there is a considerable need to develop mechanisms that allow for reliable data communication while ensuring low delay.

3.1.2 Delay and Bandwidth Tradeoffs

There are several possible approaches to addressing the problem of reducing the impact of packet loss on the delay in transport layer communication. Figure 3.1

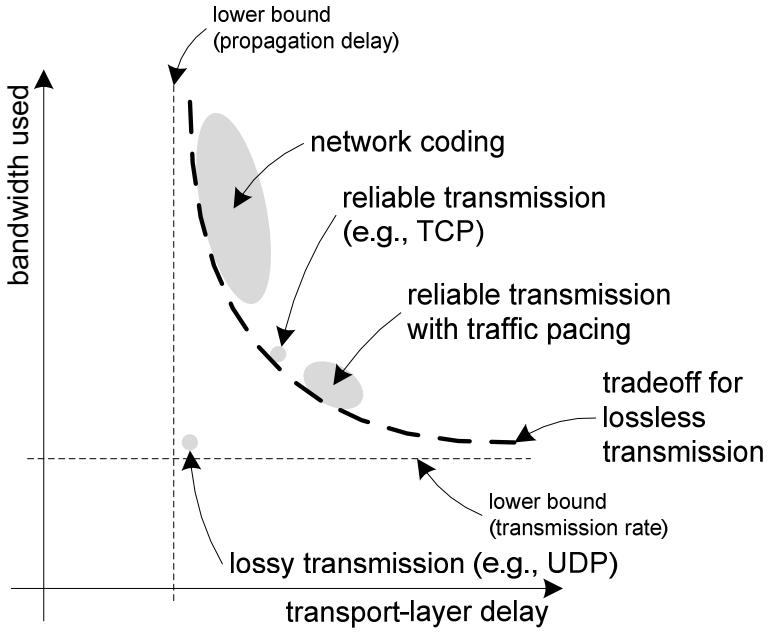


Figure 3.1. Tradeoff of delay and bandwidth consumption for different lossless transmission techniques.

illustrates how some of these techniques relate. The figure shows the amount of delay incurred at the transport layer versus the amount of bandwidth used at the transport layer. The main techniques noted in this figure are:

- Lossy transmission: Using lossy transmission protocols (e.g., User Datagram Protocol (UDP) [57]) places the bandwidth needs and delay close to the ideal lower bounds. Marginal amounts of additional bandwidth are necessary for packet headers and additional delay is incurred due to the packetized transmission of data. As discussed above, lossy transmission are not suitable for most applications.

- Reliable transmission: The baseline protocol for reliable transmission is the Transmission Control Protocol (TCP) [58]. Compared to UDP, TCP requires more bandwidth since some packets need to be retransmitted. It also incurs additional delay due to these retransmissions.

- Network coding: There are several coding techniques to reduce packet loss in networks. To reduce bit errors, error correction coding can be used [51]. To avoid packet losses, transmission information can be spread across multiple paths in the network using network coding [4]. These techniques require additional bandwidth since they rely on redundant transmission of information. They also exhibit increased delay over a lossy transmission due to the need for data reconstruction at the receiver. However, these techniques incur less delay than TCP.

- Traffic pacing: Traffic pacing is based on TCP, but uses traffic conditioning techniques in the network to reduce traffic bursts. By delaying some packet transmissions, less packet losses occur and thus less retransmissions are needed. Traffic pacing incurs a small additional delay, but uses less bandwidth than TCP since fewer retransmissions are necessary.

Overall, Figure 3.1 shows that there is a general tradeoff between bandwidth use and delay for lossless transmission in the transport layer.

While network coding and traffic pacing trade off bandwidth versus delay in different manners, it is interesting to note that they both target the same problem of packet loss. When considering a distribution of end-to-end packet delays in networks, it can be expected that most packets are transmitted successfully in the first attempt. However, packets that get lost and are retransmitted exhibit much longer delays. This "tail" of the packet delay distribution is the main problem for transport layer performance. When requiring lossless data transfers, long delays of a few packets limit overall throughput performance. Thus, it is critical to eliminate (or at least reduce) this tail in the delay distribution. In network coding, long packet delays are avoided by reducing the probability of packet loss through redundant coding of packet information. In traffic pacing, long delays are circumvented by reducing the probability of packet loss due to traffic bursts. Thus, network coding and traffic pacing can

be seen as two different approaches to tackling the same problem in transport layer transmissions.

3.1.3 Traffic Pacing in Networks

A key operational principle in the Internet is "best effort." Network resources are used when there is traffic to be sent and link schedulers on routers use "work-conserving" scheduling disciplines. This approach of not wasting opportunities to transmit packets intuitively seems to lead to the best possible network performance. However, a significant drawback is that best-effort forwarding propagates traffic bursts through the network and leads to potential buffer overflows (and thus packet loss). In contrast to best effort, several traffic pacing approaches have been proposed. In traffic pacing, transmission of some packets are intentionally delayed (despite link availability) to improve the characteristics of network traffic as a whole and thus reduce the probability of packet loss due to buffer overflows.

In our work, we present a traffic pacing technique that can reduce the burstiness of traffic and improve the throughput of transport layer TCP connections. The design of our traffic pacing system is particularly suitable for emerging network architectures for two reasons:

- Indiscriminate pacing does not require per-flow state: Many existing pacing techniques determine packet delays on a per-flow basis. This process requires computationally expensive packet classification and the maintenance of per-flow state on the router. For high-bandwidth links, this technique does not scale well. In our work, we pace packets indiscriminately regardless of what flows they belong to. Thus, we only need to maintain a single packet queue with one set of pacing parameters.

- Pacing algorithm improves operation of small-buffer networks: As we show in this work, the proposed pacing technique improves throughput in networks with

small packet buffers on routers. Since these small-buffer networks are expected to be deployed in the next-generation Internet [22], our solution presents an important contribution to the efficient operation of these networks.

The specific contributions of our work are:

- Queue Length Based Pacing (QLBP): We present a novel pacing algorithm that decreases the burstiness of network traffic by delaying packets based on the length of the local packet buffer.

- Analysis of QLBP: We present a formal analysis of QLBP that provides delay bounds and a quantitative understanding of the effect of traffic smoothing, and extends the analysis using a signal-processing approach.

- Simulation Results: We present simulation results that show the effectiveness of QLBP, its improvements in transport layer performance in small-buffer networks, and its impact on various kinds of Internet traffic.

We believe that these contributions present an important step towards more effective operation of networks, particularly when transport layer requirements demand high throughput with limited end-to-end delay.

The remainder of this chapter is organized as follows. Section 3.2 introduces the network architecture for pacing and details on the Queue Length Based Pacing algorithm. Analytical results are presented in Section 3.3. Simulation results on the effectiveness of QLBP are presented in Section 3.5. Section 3.6 discusses related work, and Section 3.7 summarizes and concludes this chapter.

3.2 Queue Length Based Pacing

The pacing technique that we propose in this work aims to reduce the burstiness of network traffic. Before detailing the pacing algorithm, we briefly discuss background

on TCP burstiness and an overview of a network architecture that uses our pacing technique.

3.2.1 TCP Burstiness

TCP is the most widely used transport layer protocol in the Internet. Its traffic characteristics have considerable impact on the operation of the network. As we discuss here, TCP traffic is inherently bursty due to the design of the protocol and can cause problems in networks with small buffers.

The TCP protocol can pace itself due to ACK-clocking, where acknowledgments are spaced out by the bottleneck link. As a result, packets sent in the congestion avoidance phase are spaced by acknowledgement arrivals. However, as pointed out by Aggarwal et al. in [3], several factors inherent to TCP can cause burstiness in the behavior of a TCP flow, such as slow start, lost packet retransmission, ACK-compression and multiplexing (for details, see [3]). Even though the impact of retransmissions of lost packets can somehow be mitigated by enabling TCP selective acknowledgement (SACK) options [50, 25], the negative impact of ACK-compression and multiplexing might become even worse in the future Internet with much larger bandwidth.

To illustrate this point, consider the detailed dynamics of TCP. (For simplicity, we only examine the TCP congestion avoidance phase.) For a long-lived TCP session, its available bandwidth is determined by the capacity of the bottleneck link. In particular, the available bandwidth is equal to the bottleneck link capacity divided by the number of long-lived TCP sessions that compete for the bottleneck link. (Here, we assume only long-lived TCP sessions exist.) If there are UDP sessions, then the bandwidth of the bottleneck link is equal to the total bandwidth minus the UDP sessions' bandwidth. We ignore the impact of short-lived TCP sessions because of their small congestion windows. Due to ACK-compression and multiplexing, all packets belonging to one congestion window can go through the bottleneck link in a back-to-

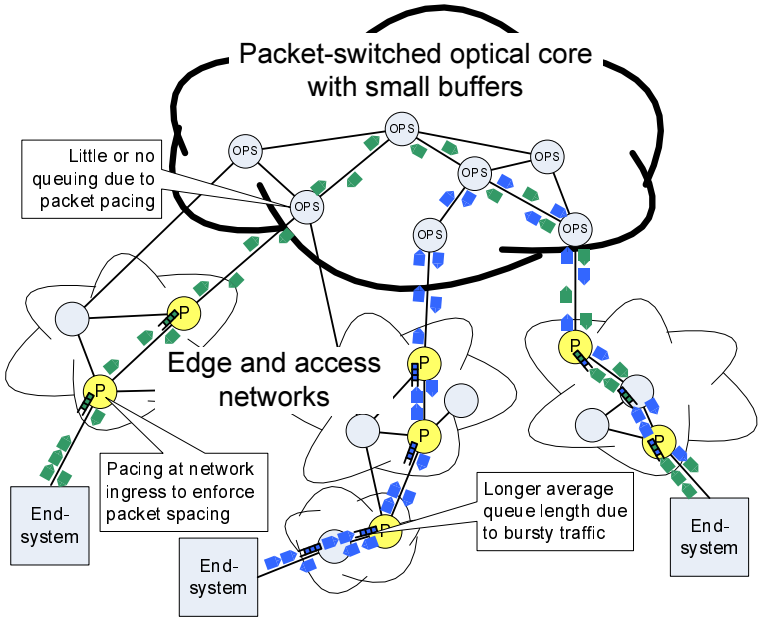


Figure 3.2. Network architecture with opportunistic pacing.

back manner. Thus, the transmission rate within a burst of packets is likely to be close to the line speed of the bottleneck link, which might be much higher than the long-term throughput of the underlying TCP session. This difference is the source of burstiness in the TCP session. As physical link speeds increase in the future Internet [30], this burstiness will be more severe.

3.2.2 Pacing Network Architecture

To reduce the burstiness of TCP traffic (and any other traffic), we propose a pacing technique that delays some packet transmissions. This pacing process can be implemented on the outgoing interfaces of routers. We envision an overall network architecture as shown in Figure 3.2. Pacing is deployed on several (but not necessarily all) nodes in the network. Since pacing cannot be practically implemented on optical packet switches, it is constrained to non-optical routers. These routers have sufficiently large buffers that allow moderate traffic bursts to be absorbed and paced without packet loss. At the network edge, routers with pacing capabilities reduce

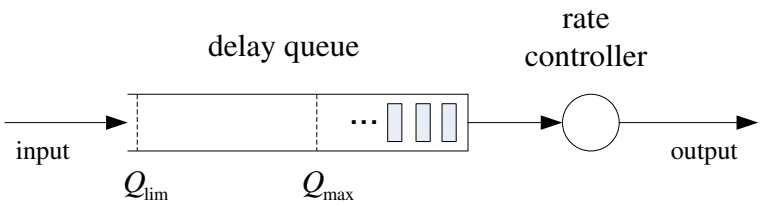


Figure 3.3. QLBP system for router buffer.

the burstiness of traffic before it enters the small-buffer network core. Within the network core, packet drops are reduced since non-bursty traffic is less likely to fill up router queues, even when they are small.

It is important to note that all traffic on an outgoing link uses only one queue and pacer. Thus, pacing is done *indiscriminately* and can be implemented efficiently for high-performance routers. Also, pacing can be performed *opportunistically*: the more pacing nodes traversed by traffic, the less bursty the traffic becomes.

3.2.3 Queue Length Based Pacing System

The general idea of Queue Length Based Pacing (QLBP) is to adjust the sending rate of a queue according to the queue length, rather than send packets at a constant rate. The structure of a QLBP system is shown in Figure 3.3, and the major notation used in this chapter is summarized in Table 3.1.

The figure shows a single input and output, but the concept can be applied to routers with any number of ports. A QLBP system includes a delay queue and a rate controller, and has three parameters: $\mu_{\max}, \mu_{\min}$ and $Q_{\max}$. The delay queue in Figure 3.3 is an ordinary FIFO queue. Packets arrive at a certain rate on the input link and are stored in the delay queue. If the queue is full (i.e. $q(t) = Q_{\text{lim}}$), the arriving packet is dropped. The output rate $\mu(t)$ is controlled by a rate controller according to the queue length $q(t)$: if $0 \leq q(t) \leq Q_{\max}$, $\mu(t)$ is calculated in a deterministic way (will be specifically introduced in the next sub-section); if $Q_{\max} < q(t) \leq Q_{\text{lim}}$, $\mu(t)$ is set to the capacity C of the outgoing link.

Table 3.1. Major notation in Chapter 3

Defined in Section 3.2.3	
$q(t)$	instantaneous length of the delay queue at time t
$\lambda(t)$	arrival rate of input traffic at time t
$\mu(t)$	output rate of the rate controller at time t
$\mu_{\max}$	maximum rate at which the rate controller transmits packets when pacing is enabled
$\mu_{\min}$	minimum rate at which the rate controller transmits packets when pacing is enabled
$Q_{\max}$	(pacing cutoff queue length) queue length beyond which no pacing delays are introduced by the pacer
$Q_{\lim}$	buffer size of the delay queue
C	capacity of the outgoing link
Defined in Section 3.3.2	
d	pacing delay
d_{pacer}	delay a packet experiences when passing through a QLBP pacer
d_{FIFO}	delay a packet experiences when passing through a FIFO queue
Defined in Section 3.3.3	
N_1	ON Poisson counter of the Markov ON-OFF modeled process
N_2	OFF Poisson counter of the Markov ON-OFF modeled process
r_1	rate of ON Poisson counter N_1
r_2	rate of OFF Poisson counter N_2
h	peak rate during ON periods

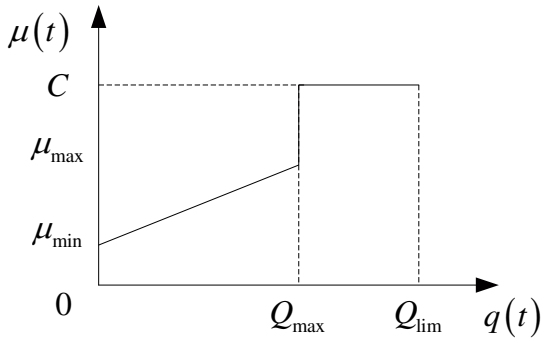


Figure 3.4. Pacing rate $\mu(t)$ vs. queue length $q(t)$.

Typically, QLBP would be used on an egress port of a router. In this case, the delay queue is the output queue of the egress port, and C is the link capacity of the egress port.

3.2.4 Pacing Delay

One of the key aspects of any pacing algorithm is how the inter-packet pacing delay is determined. In TCP pacing [3], the inter-packet pacing delay is roughly set to the ratio of the current RTT to the congestion window size. In the pacing scheme proposed by Sivaranman [64], the inter-packet pacing delay is calculated based on the packet arrival curve and the packet deadline curve within the same pacing interval. In QLBP, we determine this delay based on some very simple rules:

- If the pacing queue lengths increases due to a higher input traffic rate, QLBP intentionally lowers the introduced pacing delay. This rule ensures the link can be fully utilized under a heavy load.

- Packets that arrive at a rate lower than $\mu_{\min}$ are not delayed. This rule ensures that pacing is only activated when packets arrive at a high rate.

Based on these rules, we have designed the queue length dependent output rate $\mu(t)$ as follows:

$$\mu(t) = \begin{cases} \frac{\mu_{\max} - \mu_{\min}}{Q_{\max}} q(t) + \mu_{\min}, & 0 \leq q(t) \leq Q_{\max}, \\ C, & \text{otherwise.} \end{cases} \tag{3.1}$$

Figure 3.4 depicts the output rate $\mu(t)$ versus the instantaneous queue length $q(t)$.

A key question remaining to answer is *how one translates the pacing rate at a particular time to a pacing delay that is enforced between two consecutive packets to achieve the pacing effect*. We now answer this question. A QLBP system uses a variable $S(t)$ to record the packet size of the last transmitted packet before time t. Whenever a packet is forwarded out of the pacing queue, $S(t)$ is set to the packet size of that packet. The pacing delay is calculated by

$$d_p(t_0) = \frac{S(t_0)}{\mu(t_0)}, \tag{3.2}$$

where t_0 is the time at which the last packet departs from the pacing queue. Starting at t_0, the pacing queue is blocked for $d_p(t_0)$ seconds, that is, no packet is allowed to be served within $d_p(t_0)$ seconds after time t_0. The pacing delay $d_p(t_0)$ is called a penalty time. This penalty time will be adjusted whenever the length of the pacing queue changes, for instance, new packets arrive at the pacing queue. As time goes by and the queue length potentially increases due to packet arrivals, the penalty time can be much smaller than its initial setting at time t_0. Whenever the system clock exceeds $t_0 + d'_p(t_0)$, the pacing queue is unblocked and becomes ready to serve the packet at its head, if any. Note we use d'_p to denote the potentially updated pacing delay $d_p(t_0)$.

The above procedure is key to fulfilling rule 2 described at the beginning of this sub-section. Two points are worth emphasizing again. First, QLBP always uses the size of the last departed packet to calculate the pacing delay. Second, the pacing delay is dynamically updated whenever the length of the pacing queue changes.

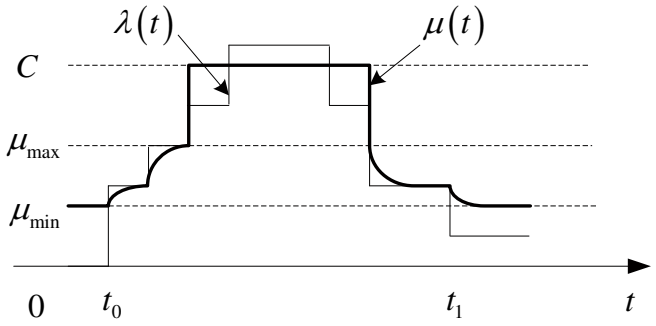


Figure 3.5. Relationship between $\mu(t)$ and $\lambda(t)$.

3.2.5 Example of Pacing

In what follows we use a simple example shown in Figure 3.5 to illustrate how a QLBP system paces packets. Suppose that at time t_0, $\lambda(t)$ is zero. From that moment on, $\lambda(t)$ begins to increase. Without loss of generality, $\mu_{\min}$ and $\mu_{\max}$ are set to $\frac{C}{a}$ and $\frac{C}{b}$, and $Q_{\max}$ is set to $\frac{Q_{\lim}}{c}$, where $a, b, c > 1$ and $a > b$.

When $\lambda(t) < \mu_{\min}$, $q(t) = 0$ and $\mu(t) = \mu_{\min}$ according to (3.1). As a result, no packets are paced and the actual output rate is still $\lambda(t)$. When $\lambda(t)$ exceeds $\mu_{\min}$ (i.e., $\mu(t)$), a queue begins to be built up, i.e., $q(t) > 0$, which causes $\mu(t)$ to increase to follow $\lambda(t)$. When the equilibrium is reached, $\mu(t) = \lambda(t)$, and the corresponding $q(t)$ is given by

$$q(t) = \frac{\lambda(t) - \mu_{\min}}{\mu_{\max} - \mu_{\min}} Q_{\max}.$$

As $\lambda(t)$ continues growing up to $\mu_{\max}$, $q(t)$ increases towards $Q_{\max}$, causing $\mu(t)$ to further increase. *When $\mu_{\max} < \lambda(t) \leq C$, $q(t)$ is equal to $Q_{\max}$ and $\mu(t)$ is C.*

It is possible for $\lambda(t)$ to be even larger than C (considering an egress port as an example). In this case, $q(t)$ will keep growing up to $Q_{\lim}$ and eventually overflow.

When $\lambda(t)$ decreases, a similar but reversed process follows.

Given the detailed description of QLBP, we now analyze its properties.

3.3 Analysis of QLBP

In this section we analyze the properties of QLBP. First, we illustrate that QLBP works in a non work-conserving mode. Moreover, we show that the pacing delay introduced by QLBP is upper bounded by a constant that depends only on system parameters. Furthermore, we demonstrate how QLBP achieves a pacing effect by analyzing its response time to the changes in the volume of network traffic and the resulting reduction of the auto-variance of the underlying traffic. Finally, we generalize our analysis to any kind of input traffic using a signal-processing approach.

3.3.1 Non Work-Conserving Property

Clearly, QLBP operates in a non work-conserving fashion, namely, the outgoing link could still be idle when the pacing queue is not empty. This non work-conserving behavior can potentially increase the packet drop probability at the pacer where the QLBP system is deployed. Compared to an ordinary first-in-first-out (FIFO) queue, in general an QLBP queue has a weaker capacity of absorbing traffic surges because of the combinational effect of a lower sending rate and a shorter available buffer space. More concretely, we compare a FIFO queue to an QLBP queue of the same size in the following scenario. Suppose that both the FIFO and QLBP queues are fed with an identical input process $\lambda(t)$ (> 0). For the FIFO queue, $q(t)$ is always zero because $\lambda(t) < C$, whereas, for the QLBP queue, $q(t)$ is always larger than zero. Besides, the sending rate, $\mu(t)$, of the QLBP queue is smaller than that of the FIFO queue, i.e., line speed C. Assume that at time t_0, $\lambda(t)$ jumps to a constant rate λ_0 ($> C$) and lasts for $\frac{Q_{\lim}}{\lambda_0 - C}$. Thus, at time $t_0 + \frac{Q_{\lim}}{\lambda_0 - C}$, the length of the FIFO queue is just equal to $Q_{\lim}$, and there is no drop in the FIFO queue between t_0 to $t_0 + \frac{Q_{\lim}}{\lambda_0 - C}$. However, such a surge can cause packet drops in the QLBP queue because of a smaller available buffer, i.e., $Q_{\lim} - q_0$, where q_0 is the length of the QLBP queue at time t_0. With the assumption of $\lambda(t) > 0$, we know for sure that $q_0 > 0$ at time t_0.

3.3.2 Guaranteed Pacing Delay

To obtain delay bounds, we first give a precise definition of pacing delay.

Definition 1. *The pacing delay d of a packet is defined as the difference $d_{pacer} - d_{FIFO}$, where d_{pacer} and d_{FIFO} represent the delay the packet experiences when passing through a QLBP queue and an ordinary FIFO (drop-tail) queue, respectively.*

Remark: This definition differentiates pacing delay from queuing delay. As the delay queue itself is the packet-storing queue, a packet might experience either queuing delay or pacing delay, or both when it passes through the delay queue. This extra amount of delay is counted as the pacing delay in that packets are not transmitted at a full line speed but, instead, at a pacing rate, which is smaller than or equal to the full line speed.

Given the definition of pacing delay, we now have the following theorem.

Theorem 3.3.1. *Given parameters $\mu_{\max}$, $\mu_{\min}$, and $Q_{\max}$, for an input traffic with rate λ, the pacing delay d at steady state depends on λ and is upper bounded by $\frac{Q_{\max}}{\mu_{\max}}$.*

The proof is provided in Appendix B.

Remark: For a 600Mbps OC-12 link equipped with a QLBP pacer of $Q_{\max}$ = 150KB (i.e., 100 of 1500 Byte packets) and $\mu_{\max} = 300Mbps$, the delay bound is 4ms. The delay bound reduces to 2ms when $\mu_{\max}$ is set to 600Mbps. In Theorem 3.3.1, we focus only on the steady state pacing delay. In practice, the incoming traffic rate changes over time. In this case, a more complicated analysis is required.

3.3.3 Reduction of Traffic Burstiness

We quantitatively analyze two aspects of the pacing effect of a QLBP system: (1) how quickly a QLBP system responds to the change in the input rate, (2) how a QLBP system smoothes the input traffic by reducing the auto-covariance. Even

though the modeling and analysis are established based on simple toy traffic models, they still unveil the fundamental nature of QLBP. To this end, our work can be viewed as the first step towards more realistic and sophisticated modeling and analysis.

In what follows we make the following assumption regarding the parameters of QLBP and the input rate $\lambda(t)$.

Assumption 1. *The parameters of the QLBP system are set as follows:* $\mu_{\min} = 0$, $\mu_{\max} = C$, $Q_{\max} = \frac{Q_{\lim}}{a}$, *where* a $(a > 1)$ *is an arbitrary real number, and for any* $t > 0$, $0 \leq \lambda(t) < C$.

This corresponds to a scenario where the QLBP system is applied to a campus edge router in which the input traffic rarely overflows the outbound link of capacity C.

3.3.3.1 Response Speed of QLBP

Under Assumption 1 the QLBP system can be described by the following equations,

$$\begin{cases} dq(t) & = (\lambda(t) - \mu(t))1_{(q>0)}dt, \\ \mu(t) & = \frac{\mu_{\max}-\mu_{\min}}{Q_{\max}}q(t) + \mu_{\min}, \end{cases} \tag{3.3}$$

where $1_{(X)}$ is an indicator function, which is 1 if predicate X is true, and 0 otherwise.

Now we examine how $\mu(t)$ responds when $\lambda(t)$ changes. Assume $\lambda(t)$ changes from 0 to λ_0 at time 0. $\lambda(t)$ can be expressed by $\lambda(t) = \lambda_0 U(t)$, where $U(t)$ is a step function. Also assume the initial condition $q(0) = 0$ (i.e., $\mu(0) = \mu_{\min}$). Then, we solve for $\mu(t)$ as follows,

$$\mu(t) = -(\lambda_0 - \mu_{\min})e^{-\frac{\mu_{\max}-\mu_{\min}}{Q_{\max}}t} + \lambda_0, \text{ for } t > 0.$$

Define the response constant α by

$$\alpha = \frac{\mu_{\max} - \mu_{\min}}{Q_{\max}}. \tag{3.4}$$

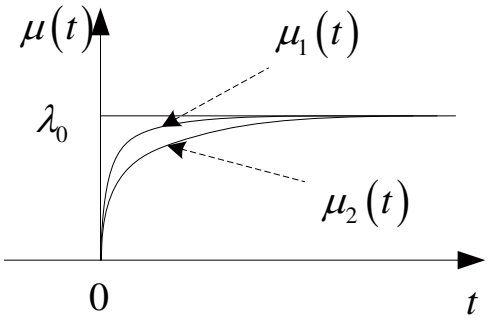


Figure 3.6. Relationship between $\mu(t)$ and changes to $\lambda(t)$.

The larger the value of α, the faster $\mu(t)$ converges to $\lambda(t)$, as shown in Figure 3.6. Under the same initial condition, $\mu_1(t)$, with a larger value of α, converges to λ_0 faster than $\mu_2(t)$ does.

3.3.3.2 Reduction of Auto-covariance

Next we propose a fluid model that describes the dynamics of the QLBP system. Our goal is to provide insights into how the QLBP system smoothes traffic in terms of reducing auto-covariance of network traffic rate. For a random process $X(t)$, its auto-covariance is defined by

$$\mathrm{Cov}(X(t_1), X(t_2)) = E[(X(t_1) - E[X(t_1)])(X(t_2) - E[X(t_2)]).$$

In this case, once the queue becomes nonempty, it remains so, though it may be very arbitrarily close to zero. Then, Equation (3.3) gives

$$\frac{d\mu(t)}{dt} = -\alpha\mu(t) + \alpha\lambda(t). \tag{3.5}$$

To investigate the impact of QLBP on the auto-covariance of the network traffic, we consider a special case where incoming traffic is modeled as a Markov ON-OFF process. The Markov ON-OFF process has been used to model voice data [31, 54]

and to show the impact of the auto-covariance of network traffic on buffer size [12, 36, 79, 35]. Also Willinger et al. [74, 75] characterized Ethernet LAN traffic as an aggregate of multiple ON-OFF processes and interpreted the measurements in terms of exponential and heavy-tailed distributed ON/OFF durations.

Now the input traffic is modeled as a Markov ON-OFF process, $\lambda(t)$, with peak rate h, ON and OFF Poisson counters N_1 and N_2 with arrival rates r_1 and r_2. Thus, $\lambda(t)$ is given by a Poisson Counter Driven Stochastic Differential Equation (PCSDE) [12]

$$\lambda(t) = hx(t),$$

where

$$dx(t) = (1 - x(t))dN_1(t) - x(t)dN_2(t).$$

Note that the average ON and OFF period durations are $1/r_2$ and $1/r_1$, respectively, and, as a result, $E[\lambda] = hE[x] = hr_1/(r_1 + r_2)$ (For details, see [12]).

Combining the above equations together, we have the following description of the QLBP system with a Markov ON-OFF input process,

$$\begin{cases} \lambda(t) & = hx(t), \\ dx(t) & = (1 - x(t))dN_1 - x(t)dN_2, \\ d\mu(t) & = -\alpha\mu(t)dt + \alpha\lambda(t)dt, \end{cases} \tag{3.6}$$

where α is given by (3.4).

Theorem 3.3.2. *Under Assumption 1, for a QLBP system described by Equation* (3.6)*, the steady-state auto-covariances of the input and output processes are given by*

$$C_{\lambda\lambda}(\tau) \triangleq \lim_{t\to\infty} \mathrm{Cov}(\lambda_{t+\tau}, \lambda_t) = \frac{h^2 r_1 r_2}{(r_1 + r_2)^2} e^{-(r_1+r_2)\tau}, \tag{3.7}$$

and

$$C_{\mu\mu}(\tau) \triangleq \lim_{t\to\infty} \mathrm{Cov}(\mu_{t+\tau}, \mu_t)$$
$$= \begin{cases} Ae^{-(r_1+r_2)\tau} + Be^{-\alpha\tau}, & if\, \alpha \neq r_1 + r_2, \\ \frac{h^2 r_1 r_2}{2(r_1+r_2)^2}(1+\alpha\tau)e^{-\alpha\tau}, & if\, \alpha = r_1 + r_2, \end{cases} \tag{3.8}$$

where

$$A = \frac{\alpha^2 h^2 r_1 r_2}{(r_1+r_2)^2(\alpha+r_1+r_2)(\alpha-r_1-r_2)},$$

and

$$B = -\frac{\alpha h^2 r_1 r_2}{(r_1+r_2)(\alpha+r_1+r_2)(\alpha-r_1-r_2)}.$$

The proof is provided in Appendix B.

Remark: Note that

$$C_{\mu\mu}(\tau) \approx \frac{\alpha}{\alpha+r_1+r_2}[1+(r_1+r_2)\tau]C_{\lambda\lambda}(\tau) < C_{\lambda\lambda}(\tau)$$

for small τ, the auto-covariance of $\mu(t)$ is smaller than that of $\lambda(t)$, indicating the short-term burstiness is reduced [12, 36]. The drawback is that the auto-covariance decays more slowly for large τ. However, since the decay is still exponential, this is not a great concern. When the buffer is small, a reduction in the short-term burstiness is more desirable.

These analytical results show that QLBP has a limited effect on the delay of packet transmissions, but can effectively reduce the short-term burstiness of traffic.

3.3.3.3 Pacing Impact in Frequency Domain

Pacing can reduce the burstiness of the incoming traffic, and hence lower the packet drop probability at downstream routers. We carry out the following calculation to demonstrate the burst-reducing effect at different frequencies of the underlying input traffic.

Assuming $\mu(0) = 0$ and taking Laplace transform on both sides of Eq. (3.5), we have

$$U(s) = \frac{\alpha}{s+\alpha}\Lambda(s), \tag{3.9}$$

where $U(s)$ and $\Lambda(s)$ are Laplace transforms of $\mu(t)$ and $\lambda(t)$, respectively. Thus, given an input traffic signal $\lambda(t)$, the frequency information of $\mu(t)$ can be completely determined using Eq. (3.9).

In what follows we use a toy traffic model to demonstrate the pacing effect of QLBP at different frequencies of an underlying network traffic. Here the incoming traffic $\lambda(t)$ is modeled by

$$\lambda(t) = c_0 u(t) + h_1 \sin(\omega_1 t) + h_2 \sin(\omega_2 t), \tag{3.10}$$

where $u(t)$ is the step function and $c, h_1, h_2, \omega_1, \omega_2$ satisfy the constraints below,

$$\begin{cases} c_0 > h_1 \gg h_2, \\ c_0 + h_1 + h_2 < C, \\ \omega_1 \ll \omega_2. \end{cases} \tag{3.11}$$

The traffic model (3.10) is motivated by an observation that Internet traffic is shown to oscillate at large time scales, namely, tens of minutes (see Fig. 1-(a) in [77]). In addition, Internet traffic changes quite dramatically at small time scales, for instance, less than hundreds of milliseconds [24]. Motivated by such observations, we use the terms $h_1 \sin(\omega_1 t)$ and $h_2 \sin(\omega_2 t)$ to model traffic burst at the two significantly different time scales ω_1 and ω_2. The top inequality in (3.11) guarantees that $\lambda(t) > 0$. The middle inequality ensures $\lambda(t) < C$. The last inequality reflects the fact that the two traffic components are far from each other in frequency domain.

Notice here that by no means we intend to make a claim that the Internet traffic exhibits any kind of periodicity as described by Eq. (3.10). Our only reason for

adopting such a model is to reflect the components at different frequencies of the traffic signal. Eq. (3.10) is an extremely simplified model in which the power of the incoming traffic signal is all concentrated at two frequencies ω_1 and ω_2.

For incoming traffic $\lambda(t)$ given by Eq. (3.10), its Laplace transform is given by

$$\Lambda(s) = \frac{c_0}{s} + \frac{h_1\omega_1}{s^2+\omega_1^2} + \frac{h_2\omega_2}{s^2+\omega_2^2}. \tag{3.12}$$

Substituting it into Eq. (3.9), we obtain

$$U(s) = \frac{\alpha}{s+\alpha}(\frac{c_0}{s} + \frac{h_1\omega_1}{s^2+\omega_1^2} + \frac{h_2\omega_2}{s^2+\omega_2^2}). \tag{3.13}$$

Taking the inverse Laplace transform for the equation above, we have

$$\begin{aligned}\mu(t) = c_0(u(t) - e^{-\alpha t}) \\ + \frac{h_1\alpha\omega_1}{\omega_1^2+\alpha^2}e^{-\alpha t} + \frac{h_1}{\sqrt{(\omega_1/\alpha)^2+1}}\sin(\omega_1 t - \gamma_1) \\ + \frac{h_2\alpha\omega_2}{\omega_2^2+\alpha^2}e^{-\alpha t} + \frac{h_2}{\sqrt{(\omega_2/\alpha)^2+1}}\sin(\omega_2 t - \gamma_2),\end{aligned}$$

where γ_1, γ_2 are given by

$$\gamma_i = \arcsin(\frac{1}{\sqrt{1+(\alpha/\omega_i)^2}}), \text{ for } i = 1, 2.$$

For $t \gg 0$, we have

$$\begin{aligned}\mu(t) = c_0 u(t) + \frac{h_1}{\sqrt{(\omega_1/\alpha)^2+1}}\sin(\omega_1 t - \gamma_1) \\ + \frac{h_2}{\sqrt{(\omega_2/\alpha)^2+1}}\sin(\omega_2 t - \gamma_2).\end{aligned} \tag{3.14}$$

For an α ($\omega_1 \ll \alpha < \omega_2$), $\omega_1/\alpha \approx 0$ and $\omega_2/\alpha > 1$, and, therefore, $\frac{h_1}{\sqrt{(\omega_1/\alpha)^2+1}} \approx h_1$ and $\frac{h_2}{\sqrt{(\omega_2/\alpha)^2+1}} < h_2$.

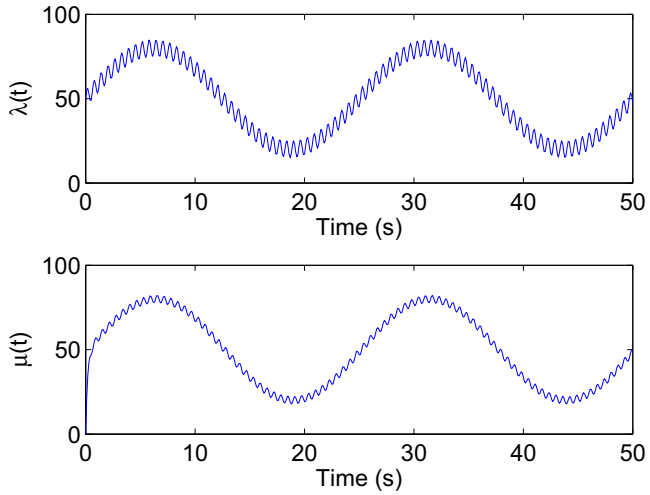


Figure 3.7. Pacing effect

Figure 3.7 shows an example of the pacing effect. The upper figure represents the trajectory of $\lambda(t)$ and the lower figure the trajectory of $\mu(t)$ (given by Eq. (3.14)). In this case, the parameters are set as follows: $c_0 = 50, h_1 = 30, h_2 = 5, \omega_1 = 0.2512, \omega_2 = 10, \alpha = 5$. It can be seen that the high frequent oscillation in $\lambda(t)$ is effectively suppressed, resulting in a smoother $\mu(t)$. In the meantime, the lower frequent component is little affected.

In reality, there is no evidence that the Internet traffic exhibits any periodicity at small time scales. However, the above analysis can be extended to account for real Internet traffic in the following way. Assume that α is chosen to curb burstiness at time scales of the order of 10ms. We take an input traffic long of thousands of seconds. By doing so, we eliminate the impact of boundary conditions. We make up a periodic signal by cascading duplicate copies of the input traffic together. Such a periodic signal can be expressed as a Fourier series, and the Fourier transform and inverse Fourier transform operations can be carried through. As a result, for these components whose frequency ω is larger than α, their amplitudes are reduced by a factor of $1/\sqrt{1+(\omega/\alpha)^2}$ and for these components whose frequencies are far smaller

than α, their amplitudes remain almost unchanged. Notice that since the coefficient $1/\sqrt{1+(\omega/\alpha)^2}$ is always smaller than one, the amplitudes at all frequencies are never amplified.

3.4 Implementations of QLBP

In this section we present two implementation algorithms of QLBP. The first algorithm is implemented as a service plug-in that can be deployed in real routers in the Open Network Laboratory test-bed [20], demonstrating the feasibility of implementing QLBP at high-speed routers. The second algorithm is implemented as a pacing queue in the network simulator 2, which is meant to evaluate the performance of QLBP in large-scale experiments.

3.4.1 QLBP Implementation in Routers

Based on Equation (3.1), we design an algorithm that can efficiently implement this pacing mechanism in the data path of a router. This algorithm is shown as Algorithm 1.

The algorithm includes two functions: `handle_packet` and `send_packets`. The `handle_packet` function is called by the operating system every time a packet arrives. The `send_packets` function uses `transmit_packet` to pass packets back to the operating system for transmission. To manage the pacing delay, the algorithm maintains a packet queue, q, and two global variables, t_{last} and t_{next}, which record the last time a packet was transmitted and the next time one could be transmitted, respectively.

Whenever a packet arrives, it is enqueued (line 5). Then the algorithm determines when the packet may be transmitted (line 6) while maintaining the delay defined in Equation (3.2). If the next transmission time is in the future, a callback is scheduled through the operating system (line 8). Otherwise, `send_packets` is called directly (line 10).

When `send_packets` is called (either directly from `handle_packet` or through a callback), the function enters a while loop (lines 15–20). While there are packets in the queue and the next transmission time has passed, packets are dequeued (line 16), transmitted (line 17), and their transmission time recorded (line 18). Then the next transmission time is determined based on the previous transmission time and the queue length (line 19). Once the while loop terminates, the next call back is scheduled if there are more packets to be transmitted.

We make several important observations about the QLBP algorithm:

- The delay (i.e., $t_{next} - t_{last}$) is updated every time the queue length changes. Thus, the pacing delay always considers the most recent state of the delay queue.

- The algorithm does not explicitly cancel scheduled callbacks that become unnecessary (e.g., when t_{next} is reduced due to arrival of another packet and the packet transmission is triggered by the `handle_packet` calling `send_packets`). The check for (`system_time()` $\geq t_{next}$) in line 15 ensures that "old" callbacks do not trigger premature transmissions.

- S_p refers to the size of the packet at the head of queue q, both in line 6 and line 19.

- The initialization of t_{last} to 0 (i.e., $t_{last} \ll$ `system_time()`) in line 2 ensures that the first packet traversing the node does not get delayed when calculating t_{next} in line 6.

3.4.2 QLBP Implementation in NS2

For our simulation study, we implemented QLBP that realizes Equation (3.1). To test the QLBP mechanism in a larger-scale network, we need ns2 as our simulation environment.

Algorithm 1 QLBP Algorithm in Open Network Laboratory

1: $q \leftarrow$ empty_queue()
2: $t_{last} \leftarrow 0$
3:
4: **function** handle_packet(p)
5: enqueue(q,p)
6: $t_{next} \leftarrow t_{last} + S_p/(\frac{\mu_{max}-\mu_{min}}{\text{max\_length}(q)} \cdot \text{length}(q) + \mu_{min})$
7: **if** $t_{next} >$ system_time() **then**
8: callback(t_{next},send_packets())
9: **else**
10: send_packets()
11: **end if**
12: **end function**
13:
14: **function** send_packets()
15: **while** (system_time() $\geq t_{next}$) $\wedge$ (length(q) > 0)
16: $p \leftarrow$ dequeue(q)
17: $S_p \leftarrow p.size()$
18: transmit_packet(p)
19: $t_{last} \leftarrow$ system_time()
20: $t_{next} \leftarrow t_{last} + S_p/(\frac{\mu_{max}-\mu_{min}}{\text{max\_length}(q)} \cdot \text{length}(q) + \mu_{min})$
21: **end while**
22: **if** length(q) > 0 **then**
23: callback(t_{next},send_packets())
24: **end if**
25: **end function**

The algorithm used in our work is described in detail as Algorithm 2. There are four functions: `handle_packet()`, `send_packet()`, `resume()` and `target()`. The `handle_packet()` function is triggered by a packet arrival event. The `send_packet()` function uses the `target()` function to deliver a packet to the link. After it delivers the packet to its associated link, the queue is blocked for a certain period equal to the transfer time, that is, S_p/C, where S_p is the size of the delivered packet. The `resume()` function is invoked when a queue is awakened by a timer expiration. The timer could be set by either the queue itself or its downstream link that receives the packet delivered by the queue.

In our ns2 implementation, we use t_{next} to control when a packet at the head of the delay queue is allowed to be transmitted. Variable t_{last} is used to keep track of the last packet's sending time. The difference of $t_{next} - t_{last}$ is the delay we intend to control to implement the pacing effect. A longer difference means a lower output rate of the rate controller.

We make several important observations about the QLBP algorithm:

- The delay (i.e., $t_{next} - t_{last}$) is updated every time the queue length changes. Thus, the pacing delay always considers the most recent state of the delay queue. Also the complexity of updating the delay is $O(1)$. The calculation of delay can be executed based on specific hardware.

- Whenever a packet arrives at the delay queue, it will be forwarded immediately if the queue is not blocked and $now() \geq t_{next}$. This behavior ensures that an *the first packet arriving after a timer expires (at t_{next}) does not get delayed, which is critical to the implementation of the adaptive pacing delay.* By "first", we mean such a packet that finds the queue empty and non-blocked when it arrives.

Algorithm 2 QLBP Algorithm in NS2: Part I

```
1: q ← empty_queue()
2: t_last ← 0
3:
4: function handle_packet(p)
5:    enqueue(q,p)
6:    if isblocked(q) then
7:       t_next ← t_last + S_p/((μ_max−μ_min)/Q_max · length(q) + μ_min)
8:       if q.timer.status() == PENDING then
9:          if now() ≥ t_next then
10:            q.timer.reschedule(now(), resume())
11:         else
12:            q.timer.reschedule(t_next, resume())
13:         end if
14:      end if
15:   else
16:      if now() ≥ t_next then
17:         if q.timer.status() == PENDING then
18:            q.timer.cancel()
19:         end if
20:         send_packet()
21:         block q
22:      else
23:         q.timer.schedule(t_next, resume())
24:      end if
25:   end if
26: end function
27:
28: function send_packet()
29:   p ← dequeue(q)
30:   S_p ← p.size()
31:   t_last ← now()
32:   t_next ← t_last + S_p/((μ_max−μ_min)/Q_max · length(q) + μ_min)
33:   target(q,p)
34: end function
```

Algorithm 3 QLBP Algorithm in NS2: Part II

```
1: function resume()
2:   if now() ≥ t_next then
3:     if q.timer.status() == PENDING then
4:       q.timer.cancel()
5:     end if
6:     if length(q) > 0 then
7:       send_packet()
8:     else
9:       unblock q
10:    end if
11:  else
12:    q.timer.reschedule(t_next, resume())
13:  end if
14: end function
15:
16: function target(q,p)
17:   target processes packet p
18:   target.timer.schedule(now(), q.resume())
19: end function
```

3.5 Simulation Results

The reduction of burstiness in network traffic translates into increased throughput for TCP traffic. In this section, we present results from a QLBP prototype implementation on the Open Network Laboratory (ONL) [20]. We also show results from simulation using larger-scale network configurations in ns-2 [65]. These results (1) show the pacing effect of QLBP on TCP and UDP flows, (2) validate the adaptive pacing delay introduced by QLBP, (3) quantitatively evaluate QLBP effectiveness on reducing burstiness of traffic in terms of the variance of the instantaneous traffic rate, (4) compare QLBP performance with TCP pacing in improving link utilization, and (5) show that the end-to-end delay distribution of paced traffic has a smaller tail than that of unpaced traffic.

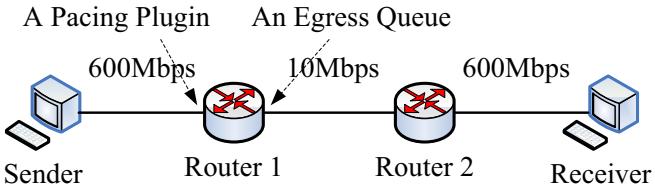


Figure 3.8. Network topology for single TCP flow.

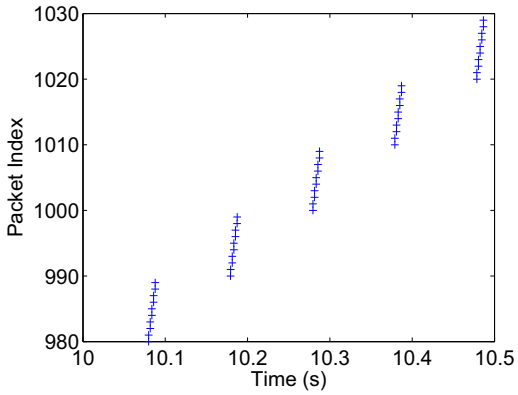


Figure 3.9. Arrival process of TCP packets without pacing.

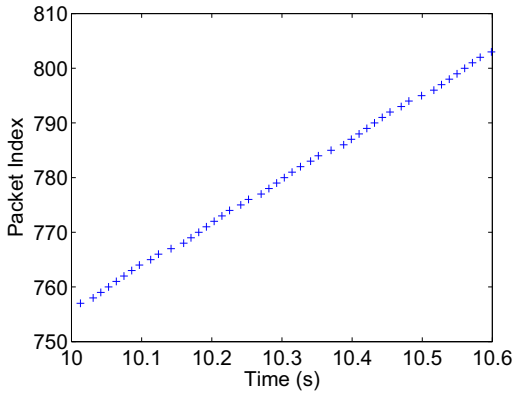


Figure 3.10. Arrival process of TCP packets with QLBP pacing.

3.5.1 Impact of QLBP on Single TCP and UDP Flows

This set of experiments is conducted using prototype implementation of QLBP in the Open Network Laboratory. More details on this implementation of QLBP can be found in [14]. The topology for these experiments is shown in Figure 3.8. A QLBP pacer is implemented as an ONL plugin and applied at the ingress port of router 1. A TCP or UDP flow is transmitted between the sender and the receiver.

The experimental setup is as follows: $\mu_{\max} = 200$Mbps, $\mu_{\min} = 1.2$Mbps, $Q_{\max} =$ 100pkts. The round-trip time (RTT) from the sender and the receiver is always 100ms. To create a RTT of 100ms, two 50ms `pdelay` plugins are installed at two egress ports of router 2. The buffer size of the egress queue at the 10Mbps link is 16 pkts.

The traffic over the 1Mbps link consists of a single TCP connection. Figure 3.9 shows that without pacing the packets within one RTT window are sent as a burst,

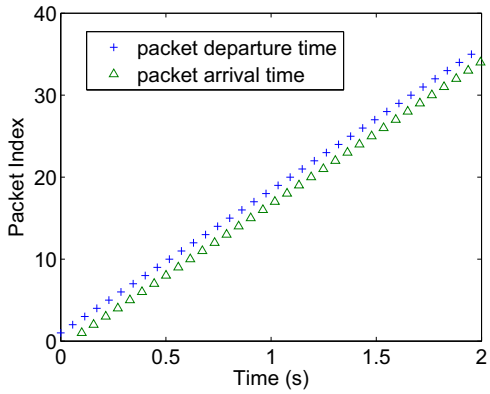


Figure 3.11. Arrival and departure time of 200Kbps CBR traffic.

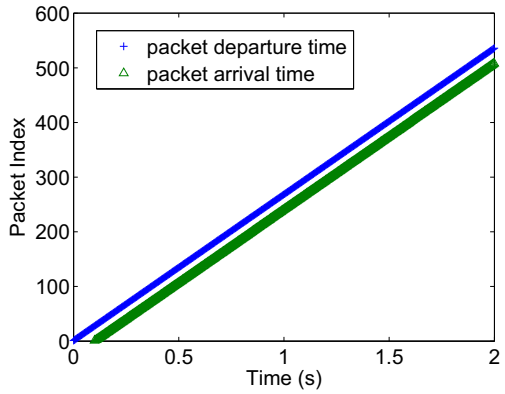


Figure 3.12. Arrival and departure time of 3Mbps CBR traffic.

e.g., a bunch of packets depart at the very beginning of each RTT period. In contrast, Figure 3.10 indicates that the QLBP pacing plugin creates a packet departure sequence that is much smoother that with no pacing.

When sending UDP traffic at a constant bit rate (CBR), we observe the packet arrival and departure processes shown in Figures 3.11 and 3.12. Figure 3.11 uses CBR traffic with a lower data rate of 200Kbps ($< \mu_{\min}$) and Figure 3.12 uses a higher data rate of 3Mbps ($> \mu_{\min}$). These figures show that QLBP does not affect the data rates of CBR traffic at steady state.

3.5.2 Sequence of Multiple QLBP Pacers

When deploying QLBP in a practical network, pacing may occur at any node within the network. Such indiscriminate pacing (independent of location or packet flow) simplifies the deployment as pacers can be installed opportunistically and without central coordination. A key question is how traffic is affected by a sequence of multiple pacers. From the earlier results in Figures 3.9 and 3.10, we see that bursty traffic gets smoothed in the limit.

To show the impact of multiple pacers in more detail, we use the topology shown in Figure 3.13. There are three pacing plugins in a row between the sender and the

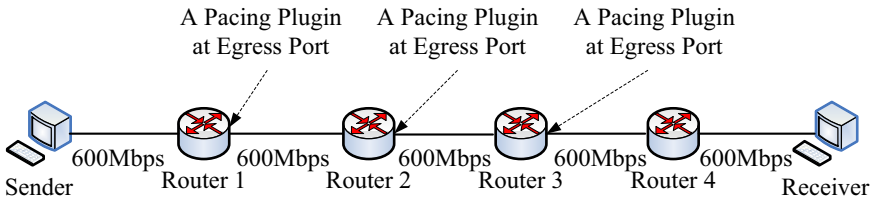


Figure 3.13. Network Topology with Multiple QLBP Pacers.

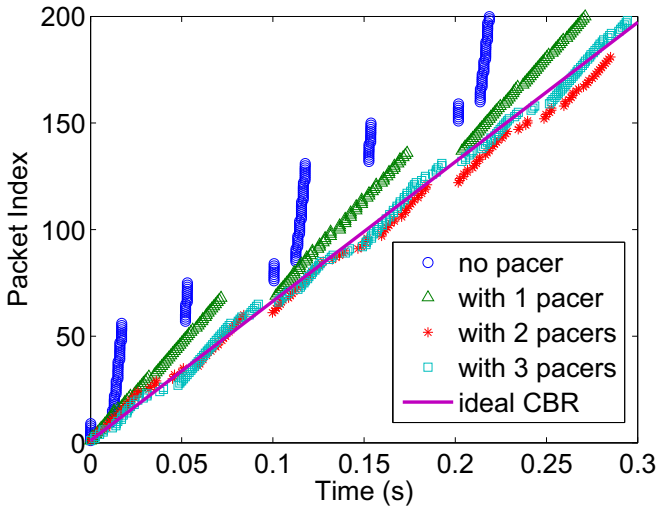


Figure 3.14. Arrival Process with Multiple Pacers

receiver. Every pacing plugin has the configuration as specified in the prior subsection. Starting with a single pacer, we enable an additional pacer in each experiment.

Figure 3.14 shows the arrival processes of a single TCP flow passing through 0, 1, 2 and 3 pacers, respectively. To allow for a comparison of changes in the packet arrival process, we show all four arrival sequences in one figure. The packet indices and arrival time have been shifted accordingly. From Figure 3.14, it can be observed that more pacers lead to smaller gaps between two consecutive RTT periods. And thus traffic approaches the properties of CBR after only 3 pacing steps.

3.5.3 Adaptive Pacing Delay

In this ns-2 experiment, we send CBR traffic through a QLBP pacer and examine the pacing queue length Q_p and the pacing delay D_p. Figure 3.15 shows the topology

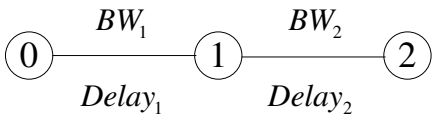


Figure 3.15. A three node topology.

Table 3.2. Pacing delay vs. input rate

λ (Mbps)	Q_p (pkts)	D_p (ms)
1	0	0
2	0	0
4	2	4
6	4	5.33
8	7	7
10	9	7.2
12	10	6.67
15	10	5.3

used in the experiment. A CBR traffic with rate λ flows from node 0 to node 2. A QLBP pacer is placed at node 1 to pace the traffic towards node 2. The parameters are set as follows. $BW_1 = BW_2 = 15$Mbps, and $Delay_1 = Delay_2 = 10$ms. $\mu_{\max} = 10$Mbps, $\mu_{\min} = 2$Mbps, $Q_{\max} = 10$pkts and $Q_{\lim} = 1000$pkts. UDP packet size is 1000 Bytes.

Table 3.2 shows pacing queue lengths and pacing delays for different CBR rates. When λ is smaller than or equal to $\mu_{\min}$, the pacing queue length is zero and no pacing delay is introduced. As λ increases while being still below $\mu_{\max}$, the pacing delay grows. When λ exceeds $\mu_{\max}$, the pacing delay stays at $Q_{\max}$. Since $\mu = \lambda$ in steady state, the pacing delay decreases as μ and λ increases. The relationship between λ, Q_p and D_p satisfies $D_p = Q_p/\lambda$. The delay bound in this case is 8ms (10 pkts * 8000 bits per packet / 10Mbps).

3.5.4 Pacing Effectiveness

We are interested in how QLBP affects traffic burstiness. The metric of concern in this ns-2 experiment is the coefficient of variation of the traffic rate, which is used in [64] to measure the extent to which traffic is bursty. There are two sets of

experiments. In the first set, we apply QLBP to an arrival process generated by a Markov ON-OFF modeled process. Using this model, we show how the pacing effect of QLBP can be enhanced by increasing $Q_{\max}$ or deploying multiple pacers. In the second set of experiments, we use an ns-2-integrated traffic generator, Tmix [73] to replicate a 3600 second Internet trace captured on a campus edge router of North Carolina State University. This traffic trace has been shown to be self-similar [73].

3.5.4.1 QLBP on Markov ON-OFF Modeled Process

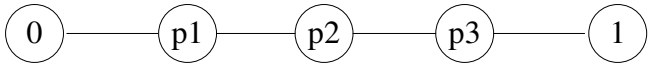


Figure 3.16. A tandem queue topology.

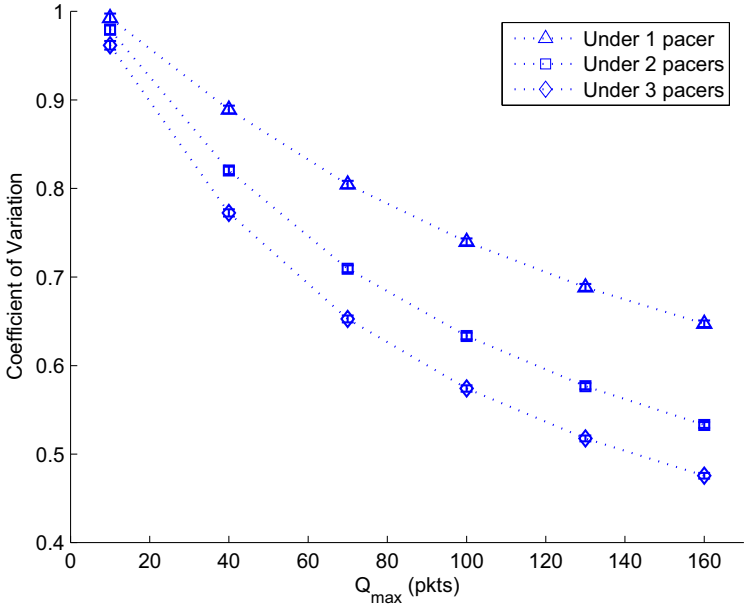


Figure 3.17. Pacing effect of QLBP on Markov ON-OFF modeled process.

Figure 3.16 shows a tandem queue topology. Traffic generated by a Markov ON-OFF process flows from node 0 to node 1. The flow rate in the ON state is h, and

0 otherwise. We run experiments with 1, 2, and 3 pacer nodes, respectively. Even though we draw all three pacer nodes in the figure, in an experiment with i pacer nodes ($1 \leq i \leq 3$), only P1 to Pi exist to pace traffic. Parameter settings are set as follows. All links have the same delay of 2ms and bandwidth of 10Mbps. $h = 2$Mbps. The average busy and idle periods are 100ms and 200ms, respectively. $\mu_{\max} = 10$Mbps and $\mu_{\min} = 10$Kbps. UDP packet size is 1000 Bytes. $Q_{\max}$ varies from 10 to 160 and the number of pacer nodes is 1, 2 or 3, respectively. We run a 1900 second long simulation with the same $Q_{\max}$ and the number of pacer nodes 10 times to obtain the average coefficient of variation over the ten runs. We analyze the trace file from [100s, 1900s]. We set 50ms as the interval and count the amount of bytes arriving at node 1 per interval. We obtain a time series $X = \{X_i\}$ where X_i represents the amount of bytes arriving at node 1 during the i-th interval.

Figure 3.17 shows the coefficient of variation of X as well as the 95% confidence interval. The x-axis is $Q_{\max}$ and the y-axis is the coefficient of variation divided by the coefficient of variation of the time series X that is generated without QLBP. Though not shown here, the average arrival rate of paced traffic (i.e., $E[X]$) is the same for all cases no matter whether and how many pacers are used, which implies that QLBP does not hurt the long-term throughput.

It is observed that a larger $Q_{\max}$ results in a smaller coefficient of variation, which is consistent with the analysis in Section 3.3.3. Also, deploying multiple pacers can further reduce the coefficient of variation.

3.5.4.2 QLBP on Self-similar Internet Traffic

It is interesting how QLBP affects burstiness of real Internet traffic. We make use of Tmix in ns-2 to replicate a piece of Internet trace file that has been show to be self-similar with Hurst parameter $H = 0.95$ [73].

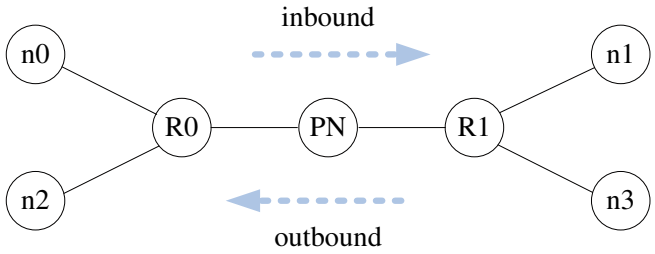


Figure 3.18. A Tmix topology

Figure 3.18 shows the topology used in this experiment. We use the same topology and parameters described in a TCL script that can be found in the ns-2 manual (for details, see Chapter 43 in the ns-2 manual [65]). The inbound and outbound connection vectors files (inbound.cvec and outbound.cvec) are provided by Weigle [38]. We slightly modify the script to insert a pacer (i.e., 'PN' node as shown in Figure 3.18) between two Tmix-Delaybox nodes (R0 and R1) to pace inbound traffic. All the links in this topology are 1Gbps. Inbound traffic is sent from n0 to n1 while outbound traffic is sent from n2 to n3. Figure 7 in [73] shows that inbound traffic rate varies from 10Mbps to 35Mbps with an average of 16Mbps. To better investigate the QLBP's effect on the inbound traffic, the parameters of the pacer node 'PN' are set as follows. $\mu_{\min} = 1$Mbps and $\mu_{\max} = 35$Mbps. $Q_{\max}$ varies from 5 to 320pkts.

Figure 3.19 shows the coefficient of variation, $CV(s)$, versus the time scale s on a log-log scale with base 2. The x-axis is the base 2 logarithm of s and the y-axis the base 2 logarithm of $CV(s)$. The basic time resolution is 5ms. A point x of coordinate $(\log_2(s_0), \log_2(CV(s_0))$ represents the base 2 logarithm of the coefficient of variation CV at time scale $5 * 2^{s_0}$ ms.

From Figure 3.19 we make the following observations. First, QLBP with a small $Q_{\max}$ (e.g., 5 or 10pkts) affects the coefficient of variation at small time scales. Comparing the plots of $\log_2(CV(s))$ with no pacing, $Q_{\max}$ of 5pkts and 10pkts, we see that QLBP with $Q_{\max}$ of 5 or 10pkts reduces the coefficient of variation by nearly 50% at time scale 5ms ($s = 0$). As s goes up, $\log_2(CV(s))$ with $Q_{\max}$ of 5 or 10 pkts

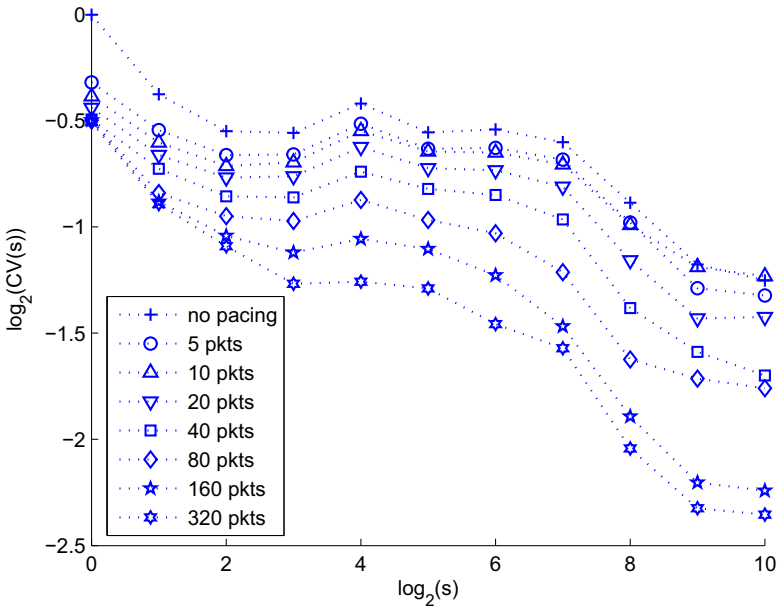


Figure 3.19. Pacing effect of QLBP on self-similar Internet traffic

converts to that with no pacing, indicating the fading impact of pacing. Second, the larger $Q_{\max}$, the wider the range of time scale in which QLBP has a significant impact on burstiness. A larger $Q_{\max}$ (e.g., 160 or 320pkts) results in a significant reduction at large time scales (e.g., 2.5s ($s = 512$) or 5s ($s = 1024$)). This is because a large value of $Q_{\max}$ makes the rate-controller of QLBP less sensitive to the changes in the instantaneous input rate.

3.5.5 Improvement on Link Utilization

In this sub-section we investigate the impact of short-term burstiness on a non-bottleneck link in terms of link utilization. This set of experiments is used in [22] to show the performance improvement of TCP pacing in small buffer networks. The topology used in this set of experiments is a dumbbell one, as shown in Figure 3.20.

Core router C0 is connected to four access routers Aj ($1 \leq j \leq 4$), each connecting ten sender nodes Si ($1 \leq i \leq 10$). Core router C1 is connected to ten receiver nodes

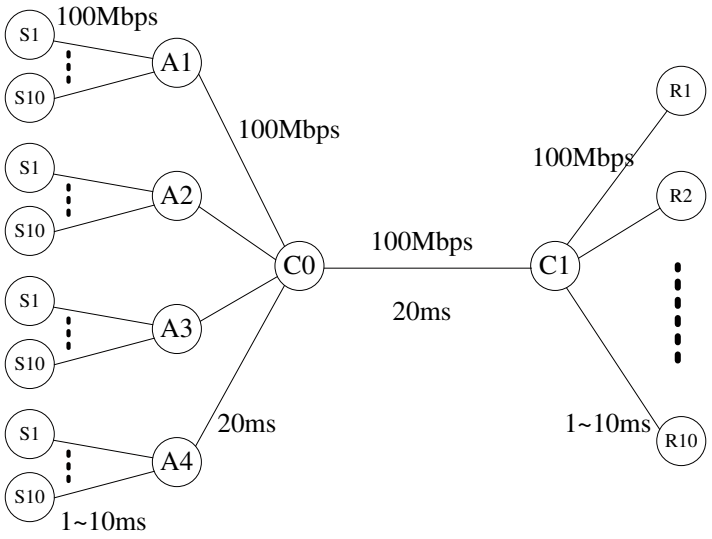


Figure 3.20. A dumbbell topology

Ri ($1 \leq i \leq 10$). The bandwidths of all links are 100Mbps. Delays between Aj ($j = 1, 2, 3, 4$) and C0 and between C0 and C1 are set to 20ms, and delays between sender nodes and access routers and between C1 to receiver nodes are uniformly distributed in [1 10ms] to reduce the impact of TCP synchronization. The average RTT is about 100ms. 40 long-lived TCP flows are sent from 40 senders to 10 receivers. For each TCP flow, the maximum congestion window is set to 32 packets and packet size is set to 1000Bytes. The maximum throughput of one TCP session on average is bounded by 2.5Mbps ($\approx 1000 Bytes/packet * 8 bits/byte * 32 packets/100 ms$). To reduce the impact of synchronization, the start times of 40 TCP sessions are uniformly distributed in [0 100s]. We apply four QLBP pacers on four access routers, each on the link Aj-C0 ($1 \leq j \leq 4$) with $\mu_{\max} = 100$Mbps and $\mu_{\min} = 1$Mbps. Buffer sizes of Aj ($1 \leq j \leq 4$) are set to be 2000pkts. $Q_{\max}$'s at four QLBP pacers are the same, varying from 10 to 160 packets. The buffer size at C0 varies from 1 to 100 packets. Each simulation run lasts one thousand seconds and the steady state starts at 200s. The metric is the normalized throughput (defined as the ratio of the total throughput to the link bandwidth) of link C0-C1 in steady state.

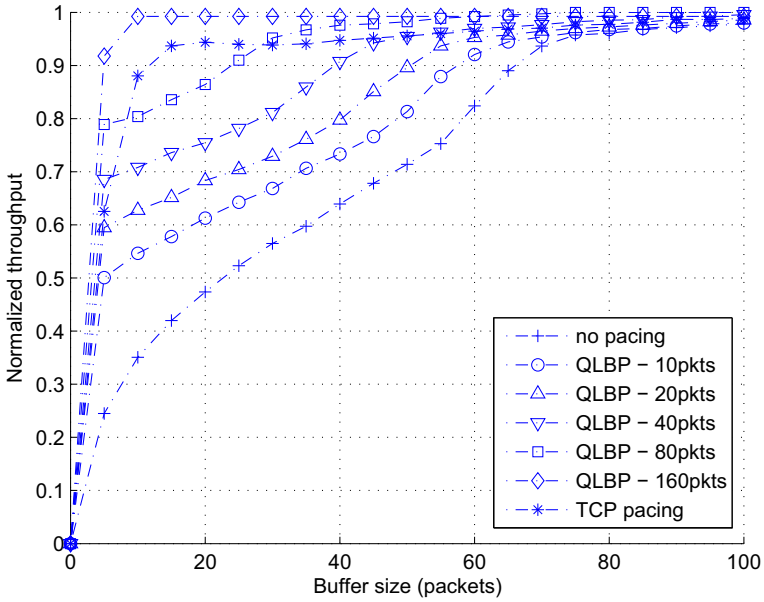
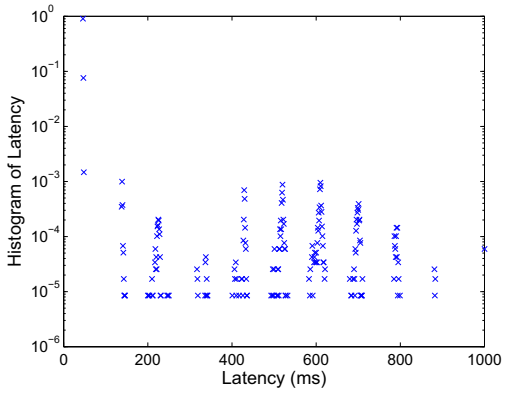


Figure 3.21. Link utilization vs. various buffer sizes.

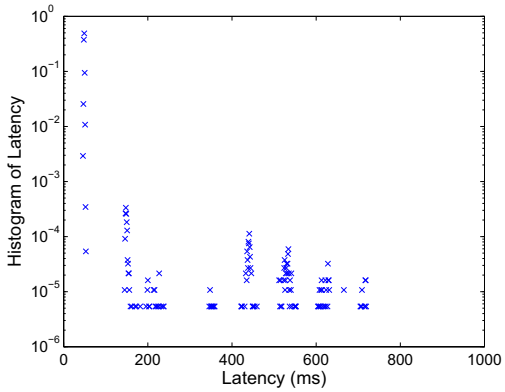
Figure 3.21 shows the normalized throughput (i.e., the link utilization) versus the buffer size at router C0. For a small buffer of 5 packets, QLBP with $Q_{\max}$ of 10 packets can improve link utilization by nearly 100%. QLBP with $Q_{\max}$ of 80 packets outperforms TCP pacing when the buffer size grows beyond 30 packets. QLBP with $Q_{\max}$ of 160 packets outperforms TCP pacing over the whole range of buffer size.

3.5.6 Delay Distribution

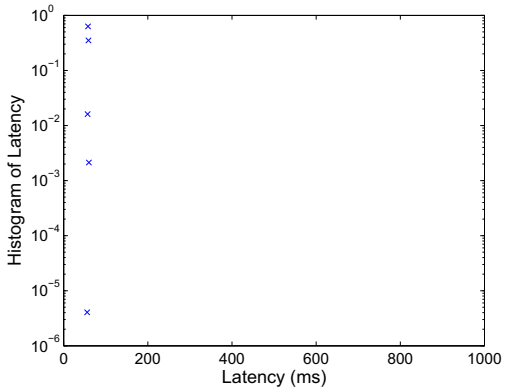
In the introduction to this chapter, we argued that a long tail in the delay distribution for packet transmission in the transport layer leads to poor performance. In Figure 3.22, we show the delay distributions for successful packet transmissions in TCP connections in ns-2 simulations. The different figures show the distribution for a network without pacing, for QLBP pacing with a small amounts of pacing ($Q_{\max}$=40 packets), and for QLBP pacing with a large amounts of pacing ($Q_{\max}$=160 packets). As expected, the tail of the distribution decreases with more pacing.



(a) No pacing.



(b) QLBP pacing (40 pkts).



(c) QLBP pacing (160 pkts).

Figure 3.22. End-to-end delay distribution for reliable packet transmissions. Long delays are caused by retransmissions in transport layer.

Table 3.3. Link utilization and delay for non-pacing and QLBP pacing.

Pacing technique	link utilization	delay		
		average	minimum	maximum
no pacing	47.57%	56.9ms	50.8ms	63.4ms
QLBP (40 pkts)	76.33%	51.9ms	46.3ms	57.5ms
QLBP (160 pkts)	98.48%	60.8ms	55.2ms	66.5ms

Table 3.3 shows the corresponding link utilization and average, minimum, and maximum packet delays. These results confirm that QLBP pacing meets the goals that we set in our work: we achieve better throughput performance (as indicated by higher link utilization) at the cost of a slightly larger delay (when comparing QLBP ($Q_{\max}$=160) with no pacing). Interestingly, QLBP ($Q_{\max}$=40) achieves both higher bandwidth and lower delay. This is accomplished by avoiding packet loss with only small amounts of additional delay.

3.5.7 Pacing Impact on a Mix of Long/Short-lived TCP Flows

In this subsection we study the impact of QLBP ono a mixture of long/short-lived TCP flows.

3.5.7.1 System Metrics

We first introduce the system metrics used in the simulation. The first metric is a so-called average flow completion time (AFCT), the use of which is justified in [21]. Different from link utilization, throughput and fairness, FCT reflects how promptly a network-based application, such as web-surfing and instant messages, responds to actions of end-users. We use it as a metric to quantitatively characterize the pacing impact on short-lived flows. The second is throughput, which is used for long-lived flows. This metric determines the underlying network's capacity to deliver real-time services, such as video on demand or teleconference. The third one is TCP fairness. A great concern for TCP pacing is the disadvantage of paced TCP flows in competing with non-paced TCP flows [3]. We use it to evaluate how the bottleneck bandwidth

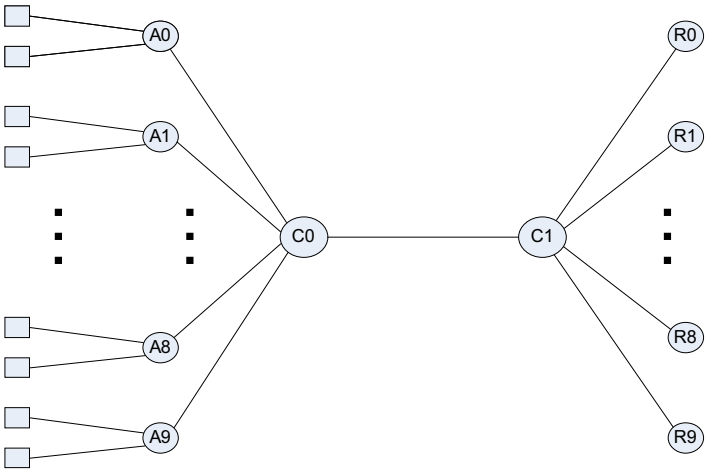


Figure 3.23. A modified dumbbell topology

is shared between paced and non-paced aggregate traffics. We use the Jain's fairness index [40] as a measure of fairness. It is defined as

$$J(\vec{x}) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}, \tag{3.15}$$

where $\vec{x} = (x_1, \cdots, x_n)$ is the throughput vector of n flows.

3.5.7.2 Experimental Setup

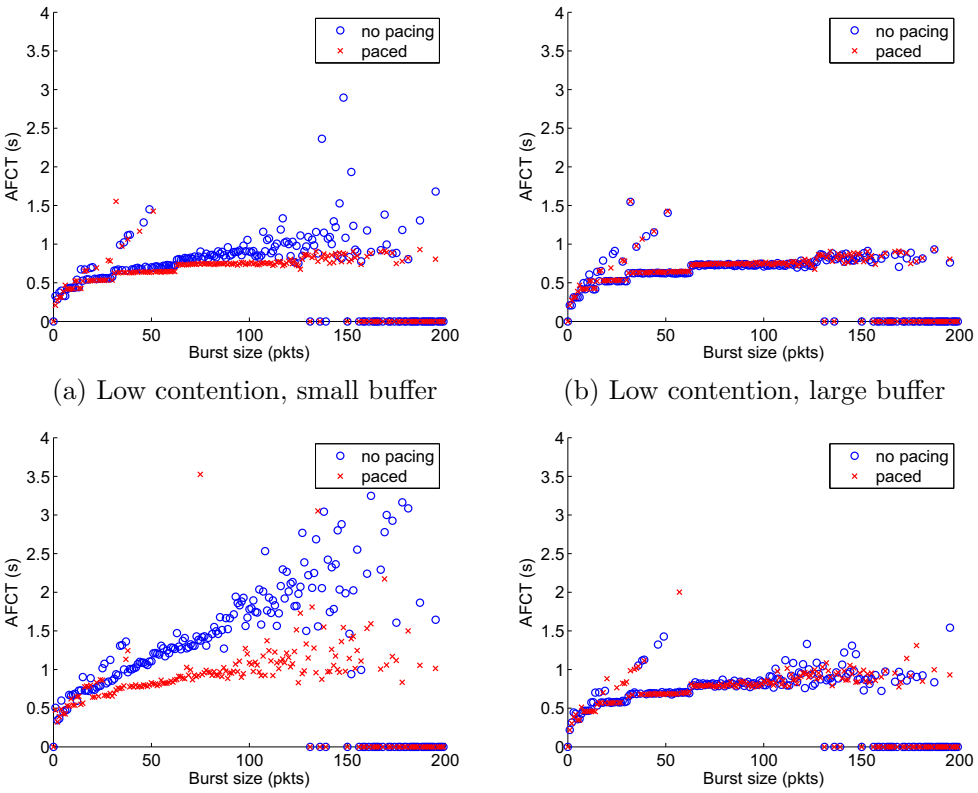
The entire set of simulations is run in ns2. Figure 3.23 shows the dumbbell topology used in simulation. All links have a bandwidth of 100Mbps. The transmission delays of all links are set such that the average round trip time between a sender/receiver pair is 100ms. The bottleneck link is between C0 and C1. Ten access routers A0 to A9 are attached to C0 while ten receivers R0 to R9 are attached to C1. Each access router is connected to two senders. Buffer sizes of all links except for the bottleneck link are set to 2000pkts. The buffer size of the bottleneck link varies from 0 to 100pkts. Thus, we ensure that all packet drops only occur at the bottleneck link, rather than anywhere else. In the case of pacing enabled, $\mu_{\max} = 100$Mbps, $\mu_{\min} = 1$Mbps, $Q_{\max} = 40$pkts.

The traffic model is a mix of long-lived and short-lived TCP flows. Each sender generates traffic from a number of long-lived TCP flows and a number of short-lived TCP flows. The TCP packet size is 1000Bytes. The maximum congestion window and receiver window of a long-lived TCP flow are 32pkts and 64pkts, respectively. The maximum congestion window and receiver window of a short-lived TCP flow are both 64pkts. A long-lived TCP flow is just an FTP application that can send data persistently during a run while a short-lived flow randomly sends a bunch of packets, called a packet burst. The average burst size is 20pkts and the average interval between two consecutive bursts is 2.25seconds. With such settings, the maximum throughput of a long-lived TCP flow is 2.5Mbps, the average throughout of a short-lived flow is 71Kbps (=20pkts*8Kbits/pkt / 2.25s), and a short-lived flow spends 99.9% of its active time in the slow-start phase, where the active time of a short-lived flow means the period of time during which it either sends packets or waits for acknowledgements.

All short-lived and long-lived flows start randomly in the first 100 seconds. Each run lasts 2000s and the transient state ends at 200s. All measures are measures taken from the interval of [200s, 2000s].

3.5.7.3 Average Flow Completion Time of Short-Lived Flows

We evaluate the impact of pacing on the performance of short flows in terms of average flow completion time (AFCT). The transmission delays are set as follows. The link delays between senders to access routers and C1 to receivers are uniformly distributed in [1ms 10ms]. The link delays between access routers are all 10ms and the link delay between C0 to C1 is 30ms. Such settings result in the average round trip transmission delays between a pair of sender and receiver of 100ms. The number of long-lived flows is either 0 or 40. The number of short-lived flows is 100. The



(a) Low contention, small buffer

(b) Low contention, large buffer

(c) High contention, small buffer

(d) High contention, large buffer

Figure 3.24. Impact of pacing on average flow completion time

Table 3.4. Parameter settings in Figure 3.24

Subfigure	# of l.f.	Buffer Size (pkts)	# of s.f.
(a)	0	5	100
(b)	0	100	100
(c)	40	5	100
(d)	40	100	100

buffer size of the bottleneck link is 5pkts or 100pkts. Parameter settings are given in Table 3.4.

Figure 3.24 shows the impact of pacing on AFCT under different conditions. In each subfigure, x-axis is the burst size and y-axis is the AFCT, where the AFCT of burst size s_0 is defined as the average completion time of all bursts long of s_0 packets. The dotted blue curves and the dashed red curves represent non-paced and paced short-lived flows, respectively. We make the following observations about Figure 3.24.

First, from Figure 3.24-(a) and (c), we can see that pacing is beneficial when the buffer size is small. In the case of Figure 3.24(a), although there are no long-lived TCP flows competing with short-lived flows, short-lived flows still experience packet drops occasionally due to a combinational effect of TCP burstiness and limited buffer size. In the case of Figure 3.24(c), the existence of long-lived flows makes the bottleneck link more congested, and as a result, short-lived flows experience drops more frequently. Due to the pacing effect, TCP burstiness is significantly reduced and AFCT is shortened (comparing non-paced and paced curves in Figure 3.24(a) and Figure 3.24(c)).

Second, when the buffer size is so large that TCP burstiness can be significantly absorbed, the AFCTs of paced and unpaced short-lived flows are quite similar, as shown in Figure 3.24-(b) and (d), indicating there is little impact of pacing on short-lived flows.

3.5.7.4 Throughput of Long-Lived Flows

Now we examine the impact of pacing on the performance of long-lived flows.

Figures 3.25 and 3.26 show how QLBP affects long-lived flows in terms of throughput. In Figure 3.25 and 3.26, the x-axis is the buffer size at the bottleneck link while the y-axis is the normalized total throughput of all long-lived flows, which is defined as the ratio of the total throughput to the capacity of the bottleneck link. In Figure 3.25 there are 20 long-lived flows. Since the maximum throughput of each long-lived flows is 2.5Mpbs, the total maximum throughput is roughly 50Mbps, corresponding to a light traffic load. In Figure 3.26 there are 40 long-lived flows, corresponding to a moderate traffic load [22]. Note that without contention, 100 and 900 short-lived flows contribute about 7.1Mbps and 64Mbps of traffic to the total traffic at the bottleneck link, respectively.

We make the following observations.

First, without pacing, long-lived TCP flows can't fully utilize the shared bandwidth when the buffer size at the bottleneck link is small. For example, with a buffer size smaller than 40pkts and the existence of 100 short-lived flows in Figure 3.25 (see the dotted curve with the square marks, the throughput of all long-lived flows is smaller than its theoretical maximum, i.e., 50Mbps. A similar trend exists for other cases.

Second, pacing improves the throughput of long-lived flows for small buffers. In Figure 3.25 and 3.26 we compare all cases with and without pacing, the improvement on throughput is shown by the gap between the pair of corresponding performance curves. For example, for a buffer size of 20pkts, pacing improves the total throughput by 50% (see curves of 900 short-lived flows in Figure 3.26) to nearly 100% (see curves of 900 short-lived flows in Figure 3.25).

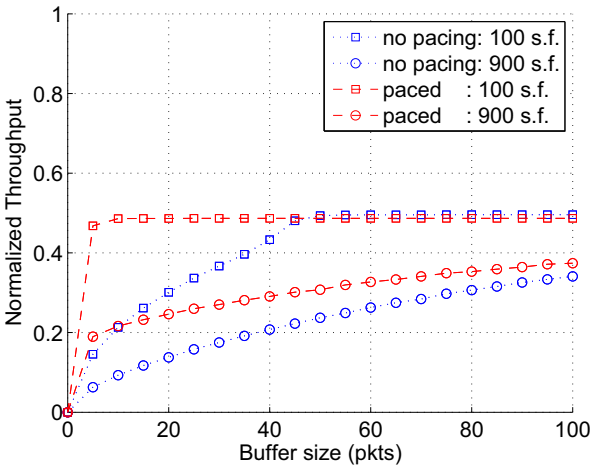


Figure 3.25. Light load: 20 long-lived flows

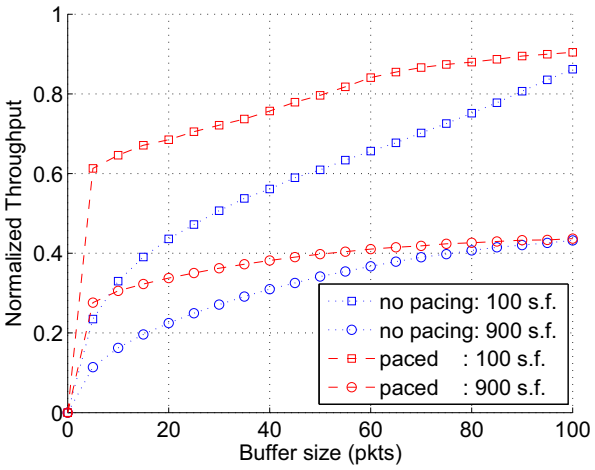


Figure 3.26. Light load: 40 long-lived flows

Third, as the buffer size at the bottleneck link gets larger, the benefit of pacing diminishes. This is because the impact of TCP burstiness that is what the pacing targets is weakened by the increased buffer size.

We conduct more experiments with different parameters. They all show the same trend.

3.5.7.5 TCP Fairness

We now study the fairness between non-paced and paced traffic. Each sender node establishes 2 long-lived flows and 5/45 short-lived flows. In total, there are 40 long-term flows and 100/900 flows. The average inter-burst time is set 10seconds. The rest of the settings are the same as described before. We disable QLBP at access routers A5 to A9 in Figure 3.23. Thus, the half of the traffic flows are paced, and rest are not.

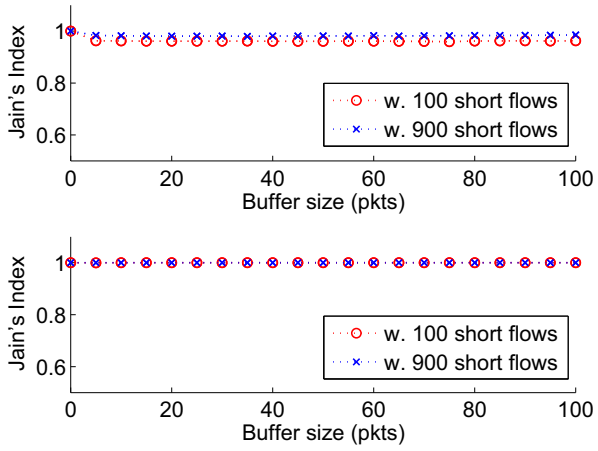


Figure 3.27. Fairness between paced and non-paced long-lived flows

The upper subfigure in Figure 3.27 shows the fairness $J(\vec{x})$ versus the various buffer size with different numbers of short-lived flows. In this case, the throughput vector $\vec{x}$ has 40 elements: the first 20 represent the throughput of paced long-lived flows while the second 20 the throughput of non-paced long-lived flows. We can see

the Jain's index $J(\vec{x})$ is quite close to 1, indicating the TCP fairness is established between these paced and non-paced flows.

The lower subfigure in Figure 3.27 also shows the fairness $J(\vec{x})$ as a function of buffer size for two populations of short-lived flows. Here $\vec{x} = (x_1, x_2)$ where x_1 and x_2 are the aggregate throughputs of the QLBP-paced flows and the unpaced flows, respectively. We use such a $\vec{x}$ to study if paced traffics are treated differently from non-paced traffic at the bottleneck link. From the figure we observe no matter how many short-lived flows exist and how large the bottleneck buffer size is, the bottleneck link give paced and non-paced traffic the same forwarding priority, indicating that paced traffic has the same priority at the bottleneck router as non-paced traffic.

3.6 Related Work

The impacts of small buffers on transport-layer network performance have been studied in the context of real-time traffic and TCP traffic [76, 60, 64, 22, 30, 44]. Interestingly, the results of these studies are not conclusive.

On one hand, it has been shown that small buffers significantly degrade network performance with ordinary TCP sessions by causing packet drop more frequently. Enachescu et al. [22] showed that a 80% workload consisting of long-lived TCP sessions only achieves a 20% link utilization when the buffer size of the shared link is 10 packets. Sivaramman et al. [64] demonstrated that "a 10Gbps optical packet switching (OPS) node with 10 to 20 packets can experience significant losses even at low (40%) to moderate (60% for long-range dependent or 80% for short-range dependent) traffic loads."

On the other hand, theoretical analyses and empirical results show that small buffers are feasible for core routers through which tens of thousands of TCP sessions flow [22, 76, 60, 30, 44]. Enachescu et al. [22] argued that $O(\log W)$ buffers are sufficient for high throughput, where W is congestion window size of each flow, and router

buffer can even be reduced to a few dozen packets if a small amount of link utilization is sacrificed. Gu et al. [30] demonstrated that more than 90% link utilization is achievable in a 1–10 Gbps bottleneck link with a buffer of 20 packets. Lakshmikantha et al. [44] further showed that $O(1)$ buffer sizes (20 packets) are sufficient for good performance with no loss of link utilization when considering the impact of file arrivals and departures. We note that all high performance results are achieved only when TCP sessions are paced by either some rate-control mechanism (i.e., TCP pacing) or access links with capacities much slower than the bottleneck link.

The main concern with the small buffer core networks is the high packet loss probability due to the small buffer size and the bursty behavior of TCP. Several techniques are proposed to lower the drop probability in small buffer networks by smoothing network traffic. Packet pacing finds its roots in the explicit rate control non-TCP protocols, which send data at a fixed rate irrespective of the receipt of acknowledgments [18, 10]. Pacing was used in the TCP context to correct the compression of acknowledgements due to cross traffic [82], to avoid slow start [7, 55], after packet loss [34], or when an idle connection resumes [71]. Aggarwal et al. [3] concluded that pacing improves throughput in some cases but in general decreases performance. The poor performance of pacing is attributed mostly to "synchronized drops" and packet delays being misinterpreted as congestion.

In addition to TCP pacing, there have been several proposals for resolving packet drops in small buffer networks [6, 64, 53, 1, 2]. The work by Alparslan et al. [6] shares a very similar idea with our, i.e., turning the pacing rate based on the buffer occupancy, which was originally proposed by Tzu-Ying Tung et al. [69], and the effect of the pacing is evaluated in a large-scale hypothetic network. The work by Sivaraman et al. [64] stems from previous works on traffic conditioners for video transmission, called traffic conditioning *off-line* [62]. They proposed an on-line version of traffic

conditioner based on this traffic conditioning *off-line*. The approaches in [53, 1, 2] rely on the global network-wide coordinated scheduling.

Unlike the above pacing-based approaches, Vishwanath et al. proposed to recover lost packets by using the packet-level forward error correction (FEC) scheme [70]. Their coding-based approach works based on an observation that "loss at core links is due to contention, not congestion." Through simulation they show the efficiency of the FEC-based approach.

3.7 Summary

Our work presents a novel view on the tradeoff between link bandwidth and packet delay. Instead of using an error correction or network coding approach where more bandwidth is used to avoid packet losses, we proposed to delay packet transmissions to reduce the burstiness of traffic and thus reduce packet losses in small-buffer networks. We present Queue Length Based Pacing, which is a pacing technique that uses a single pacing queue on router ports and adapts its sending rate based on the amount of traffic that is buffered at that port. Our analysis shows that pacing delay due to QLBP is bounded and that the variance of the instantaneous traffic rate is reduced. We show the effectiveness of QLBP through a prototype implementation and simulation. Specifically, we show that TCP connections in a small-buffer network with QLBP pacing achieve higher link utilization than in non-paced networks. Therefore, we believe that QLBP is an effective approach to improving the operation of networks and improving the effective bandwidth of connections at the cost of only small amounts of additional delay.

CHAPTER 4

CONCLUSIONS

In this chapter we summarize the work presented in this dissertation and present some interesting future work topics.

4.1 Summary

In this dissertation we (i) analyze the impact of burstiness on network performance in the context of a general queueing system model and (ii) propose a practical online packet pacing scheme, known as queue length based pacing (QLBP). The first part serves as a theoretical framework in which the benefit of pacing is demonstrated while the second part embodies our idea on the Internet-wide deployment of packet pacing.

In the first part we investigated the potential benefits of traffic pacing by quantitatively studying the impact of traffic burstiness on the buffer occupancies of a tandem queue network fed with a point process. First, we derive an expressions for the instantaneous and average queue lengths of a tandem queue network from a sample-path perspective. Second, we develop the IPA estimators for average queue lengths and also show the unbiasedness and strong consistency of them under mild and reasonable assumptions on traffic arrivals and workload patterns. Final, we show under the given conditions that the arrival traffic burstiness has a linear impact on both instantaneous and average queue lengths of all queues in a tandem network, which demonstrates that traffic pacing has great potential to reduce buffer occupancies and largely improve the packet loss and delay performance in communication networks with small buffers.

In the second part we propose an adaptive QLBP system. A QLBP system consists of a pacing queue associated with a rate controller that controls the sending rate of the pacing queue. Specially, the sending rate of the pacing queue is linearly proportional to the length of the pacing queue. Starting at a minimum sending rate, the queue begins to build if the input traffic rate to the pacing queue exceeds the current pacing rate, which in turn drives the pacing rate to increase. In contrast, when the input rate goes down, the length of the pacing queue shrinks, resulting in a lower pacing rate. Under certain conditions such a dynamic process is described by an ordinary differential question, which reveals that a QLBP system in effect performs as a low-pass filter, filtering out high frequency components in the input traffic signals.

In addition to introducing the QLBP mechanism, we further analyze its properties. Our analysis indicates that QLBP is non work conserving. We show that the pacing delay introduced by QLBP is upper bounded by a constant that depends only on system parameters of QLBP. The derivation on the paced traffic in the context of a Markov On-Off model and the analysis in frequency domain demonstrate the effectiveness of QLBP in reducing the burstiness in network traffic.

We evaluate the performance of QLBP via extensive simulations. First, we demonstrate the pacing effect of QLBP on single TCP and UDP flows as well as the cumulative pacing effect achieved with use of multiple QLBP pacers. Second, we verify that the upper bound of the pacing delay is consistent with the derived close form. Third, we compare QLBP with TCP pacing. Final, we investigate the impact of QLBP on performance of long-term and short-term TCP flows in terms of delay distribution, average flow completion time, throughput and TCP fairness. These simulation results confirm that in small buffer networks, QLBP can effectively reduce the TCP burstiness, and hence lower the packet drop probability at small buffer bottleneck links. As a result, the performance of the network and individual flows are significantly improved.

4.2 Future Work

We can continue the research work on the QLBP in the following directions.

Hardware-based Implementation of QLBP

Of interest is the feasibility of implementing QLBP at high speed routers. FPGA provides engineers with a better programming capability than ASIC chip board. So it is worth prototyping the QLBP algorithm in FPGA. One of the challenging designs in doing so is to make large delay queues on FPGA chips where buffer resource is scarce. Another challenge is to effectively implement pacing delays using the discrete time clocks on FPGA chips. Despite the existence of these technical challenges, FPGA-based QLBP implementation solution is still promising and sounds feasible.

Large Scale Simulation of Pacing

The second future work of interest is to evaluate QLBP in a large scale network test-bed with reasonable traffic patterns. Such a network setup should have two key characteristics: (1) the short-term burstiness can be effectively created so that the performance of traffic flows are significantly degraded and (2) the short-term burstiness should exhibit network-widely, namely, the congestion caused by the short-term burst overlapping occurs dynamically on multiple links, rather than a single link. The first characteristic requires the number of flows competing on one bottleneck to be so large that the small buffers can be "easily" overflowed. The second characteristic will result in a situation in which most of the traffic flows would be affected by congestion.

APPENDIX A

RELATED PROOFS ON LEMMAS AND THEOREMS IN CHAPTER 2

Proof of Lemma 2.3.1

Proof. Under C. 2.3.1 and C. 2.3.2, we have

$$\begin{aligned}\frac{dZ}{d\theta} &= \frac{1}{c_1}\frac{dX}{d\theta} \\ &= \frac{1}{c_1}\frac{dX}{d\xi}\frac{d\xi}{d\theta} \\ &= \frac{1}{c_1}\frac{X}{\xi}\frac{\xi}{\theta} \\ &= \frac{1}{c_1}\frac{X}{\theta} \\ &= \frac{Z}{\theta}\end{aligned}$$

Proof is over. □

Proof of Lemma 2.3.2

Proof. It is proven using induction. Under C. 2.3.1 and C. 2.3.2, for $m = 2, ..., M$ we have the following.

For any $n_{B,b-1}^{m-1} + 1 \leq j \leq n_{B,b}^{m-1}$ where $b \geq 1$,

1. $j = n_{B,b-1}^{m-1} + 1$: Since $v_j^m = 0$, then $\frac{dv_j^m}{d\theta} = \frac{v_j^m}{\theta}$. $\frac{dB_j^m}{d\theta} = \frac{d\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}} Z_k}{d\theta} = \frac{\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}} Z_k}{\theta} = \frac{B_j^m}{\theta}$. $\frac{dp_j^m}{d\theta} = \frac{d(c_{m-1}-c_m)B_j^m}{d\theta} = \frac{(c_{m-1}-c_m)B_j^m}{\theta} = \frac{p_j^m}{\theta}$. $\frac{dI_j^m}{d\theta} = \frac{d(\sum_{k=n_{C,j-1}^{m-1}+2}^{n_{C,j}^{m-1}+1} Y_k - B_j^m)}{d\theta} = \frac{\sum_{k=n_{C,j-1}^{m-1}+2}^{n_{C,j}^{m-1}+1} Y_k - B_j^m}{\theta} = \frac{I_j^m}{\theta}$.

2. $n_{B,b-1}^{m-1}+1<j<n_{B,b}^{m-1}$: Assume that it is true for $j=h-1$. Then when $j=h$, we have $\frac{dv_j^m}{d\theta}=\frac{d(p_{j-1}^m-c_mI_{j-1}^m)}{d\theta}=\frac{p_{j-1}^m-c_mI_{j-1}^m}{\theta}=\frac{v_j^m}{\theta}$. $\frac{dB_j^m}{d\theta}=\frac{d\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}}Z_k}{d\theta}=\frac{\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}}Z_k}{\theta}=\frac{B_j^m}{\theta}$. $\frac{dp_j^m}{d\theta}=\frac{d(v_j^m+(c_{m-1}-c_m)B_j^m)}{d\theta}=\frac{(v_j^m+(c_{m-1}-c_m)B_j^m)}{\theta}=\frac{p_j^m}{\theta}$. $\frac{dI_j^m}{d\theta}=\frac{d(\sum_{k=n_{C,j-1}^{m-1}+2}^{n_{C,j}^{m-1}+1}Y_k-B_j^m)}{d\theta}=\frac{\sum_{k=n_{C,j-1}^{m-1}+2}^{n_{C,j}^{m-1}+1}Y_k-B_j^m}{\theta}=\frac{I_j^m}{\theta}$.

3. $j=n_{B,b}^{m-1}$: we have $\frac{dv_j^m}{d\theta}=\frac{d(p_{j-1}^m-c_mI_{j-1}^m)}{d\theta}=\frac{p_{j-1}^m-c_mI_{j-1}^m}{\theta}=\frac{v_j^m}{\theta}$. $\frac{dB_j^m}{d\theta}=\frac{d\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}}Z_k}{d\theta}=\frac{\sum_{k=n_{C,j-1}^{m-1}+1}^{n_{C,j}^{m-1}}Z_k}{\theta}=\frac{B_j^m}{\theta}$. $\frac{dp_j^m}{d\theta}=\frac{d(v_j^m+(c_{m-1}-c_m)B_j^m)}{d\theta}=\frac{(v_j^m+(c_{m-1}-c_m)B_j^m)}{\theta}=\frac{p_j^m}{\theta}$. $\frac{dI_j^m}{d\theta}=\frac{dp_j^m/c_m}{d\theta}=\frac{p_j^m/c_m}{\theta}=\frac{I_j^m}{\theta}$. Since $v_{j+1}^m=0$, then $\frac{dv_{j+1}^m}{d\theta}=\frac{v_{j+1}^m}{\theta}$.

Proof is over. □

Proof of Lemma 2.3.3

Proof. **Case I:** for $m=1$, we have $l_B^1(\theta)=L_B^1(\theta)/\tau_B^1(\theta)$, where $L_B^1(\theta)$ and $\tau_B^1(\theta)$ are given by (2.5) and (2.6). Also $\tau_B^1(\theta)>0$ for any $B>1$.

Under A. 2.3.4 and C. 2.3.1, Y_i's and Z_i's are continuous in $\theta\in\Theta$. As a result, $L_B^1(\theta)$ and $\tau_B^1(\theta)$ are continuous in $\theta\in\Theta$.

Now we show $1/\tau_B^1(\theta)$ is continuous in $\theta\in\Theta$. Suppose $\{\theta_n\}$ is a sequence in Θ converges to θ. Since $\tau_B^1(\theta)$ is continuous, it is true that for any given $\epsilon\cdot(\inf_{\theta\in\Theta}\{\tau_B^1(\theta)\})^2>0$, $\exists N_1$ such that for any $n>N_1$, $|\tau_B^1(\theta_n)-\tau_B^1(\theta)|<\epsilon\cdot(\inf_{\theta\in\Theta}\{\tau_B^1(\theta)\})^2$. Thus, for any $\epsilon>0$, find N as N_1 such that for any $n>N$, $|\frac{1}{\tau_B^1(\theta_n)}-\frac{1}{\tau_B^1(\theta)}|=|\frac{\tau_B^1(\theta)-\tau_B^1(\theta_n)}{\tau_B^1(\theta_n)\tau_B^1(\theta)}|<|\frac{\tau_B^1(\theta)-\tau_B^1(\theta_n)}{(\inf_{\theta\in\Theta}\{\tau_B^1(\theta)\})^2}|<\epsilon$. Hence $1/\tau_B^1(\theta)$ is continuous in $\theta\in\Theta$.

Consequently, $l_B^1(\theta)$ is continuous in $\theta\in\Theta$.

Case II: for $m\geq 2$, we have $l_B^m(\theta)=L_B^m(\theta)/\tau_B^m(\theta)$, where $L_B^m(\theta)$ and $\tau_B^m(\theta)$ are given by (2.8) and (2.9). Also $\tau_B^m(\theta)>0$ for any $B>1$.

Lemma 2.3.2 implies that v_j^m's, p_j^m's, B_j^m's and I_j^m's are continuous in $\theta\in\Theta$. So $L_B^m(\theta)$ is continuous in $\theta\in\Theta$. $\tau_B^m(\theta)$ is also continuous in $\theta\in\Theta$. Following the same

procedure, it can be shown that $1/\tau_B^m(\theta)$ is continuous in $\theta \in \Theta$. As a result, $l_B^m(\theta)$ is continuous in $\theta \in \Theta$.

Proof is over. □

Proof of Theorem 2.3.1

Proof. C. 2.3.1 leads to a fact that there is no change in the order of events while one of system parameters is perturbing. Lemma 2.3.3 shows that l_B^m for $m = 1, ..., M$ is, with probability one, continuous in θ on Θ.

Case I: For $m = 1$, under Lemma 2.3.1 we have for any $\theta \in \Theta$

$$\begin{aligned}
\frac{dl_B^1}{d\theta} &= \frac{c_1}{(\tau_B^1)^2}\Bigg[\sum_j^{n^1_{C,B}+1} Y_j\Bigg((\sum_b^B\sum_i^{n^1_{C,b}}\frac{Z_i}{\theta}\sum_j^{i-1}Z_j + \sum_b^B\sum_i^{n^1_{C,b}} Z_i\sum_j^{i-1}\frac{Z_j}{\theta}) \\
&\quad - (\sum_b^B\sum_i^{n^1_{C,b}}\frac{Z_i}{\theta}\sum_j^{i}Y_j + \sum_b^B\sum_i^{n^1_{C,b}} Z_i\sum_j^{i}\frac{Y_j}{\theta}) + \sum_i^{n^1_{C,B}} Z_i\frac{Z_i}{\theta}\Bigg) \\
&\quad - \Bigg(\sum_b^B\sum_i^{n^1_{C,b}} Z_i\sum_j^{i-1}Z_j - \sum_b^B\sum_i^{n^1_{C,b}} Z_i\sum_j^{i}Y_j + \frac{1}{2}\sum_i^{n^1_{C,B}} Z_i^2\Bigg)\sum_j^{n^1_{C,B}+1}\frac{Y_j}{\theta}\Bigg] \\
&= \frac{l_B^1}{\theta}
\end{aligned}$$

It follows that

$$\begin{aligned}
\left|\frac{dl_B^1}{d\theta}\right| &= \left|\frac{l_B^1}{\theta}\right| \\
&= \left|\frac{1}{\theta\tau_B^1}\left[c_1\sum_{b=1}^B\sum_{i=n^1_{C,b-1}+1}^{n^1_{C,b}} Z_i\Bigg(\sum_{j=n^1_{C,b-1}+1}^{i-1} Z_j - \sum_{j=n^1_{C,b-1}+2}^{i} Y_j\Bigg) + \frac{c_1}{2}\sum_{i=1}^{n^1_{C,B}} Z_i^2\right]\right| \\
&\le \frac{c_1}{\theta\tau_B^1}\Bigg(\left|\sum_{b=1}^B\sum_{i=n^1_{C,b-1}+1}^{n^1_{C,b}} Z_i\sum_{j=n^1_{C,b-1}+1}^{i-1} Z_j\right| + \left|\sum_{b=1}^B\sum_{i=n^1_{C,b-1}+1}^{n^1_{C,b}} Z_i\sum_{j=n^1_{C,b-1}+2}^{i} Y_j\right| + \left|\frac{1}{2}\sum_{i=1}^{n^1_{C,B}} Z_i^2\right|\Bigg) \\
&\le \frac{c_1}{\theta\tau_B^1}\Bigg(\left|\sum_{b=1}^B\sum_{i=n^1_{C,b-1}+1}^{n^1_{C,b}} Z_i\tau_B^1\right| + \left|\sum_{b=1}^B\sum_{i=n^1_{C,b-1}+1}^{n^1_{C,b}} Z_i\tau_B^1\right| + \left|\frac{1}{2}\sum_{i=1}^{n^1_{C,B}} Z_i\tau_B^1\right|\Bigg) \\
&\le \frac{5c_1}{2\theta}\tau_B^1.
\end{aligned}$$

It follows that for any $\theta \in \Theta$,

$$E[\left|\frac{dl_B^1}{d\theta}\right|] \leq \frac{5c_1}{2\theta} E[\tau_B^1] \leq \frac{5c_1}{2\theta} BE[\sigma_{b+1}^1 - \sigma_b^1] < \infty,$$

which leads to

$$E[\sup_{\theta\in\Theta}\left|\frac{l_B^1(\theta)}{\theta}\right|] < \infty.$$

Furthermore, as $l_B^1(\theta)$ is continuously differential in θ, according to the Generalized Mean Value Theorem (e.g., refer to page 15 in [29]) for any $\theta, \theta + h \in \Theta$, we have

$$\left|\frac{l_B^1(\theta+h) - l_B^1(\theta)}{\theta}\right| \leq \sup_{\theta\in\Theta}\left|\frac{l_B^1(\theta)}{d\theta}\right|.$$

The right hand side is integrable, by hypothesis; the Dominated Convergence Theorem (e.g., see page 14 in [29]) applies, and for any $\theta \in \Theta$, we have

$$\begin{aligned} E\left[\frac{dl_B^1(\theta)}{d\theta}\right] &= E\left[\lim_{h\to 0}\frac{l_B^1(\theta+h) - l_B^1(\theta)}{h}\right] \\ &= \lim_{h\to 0} E\left[\frac{l_B^1(\theta+h) - l_B^1(\theta)}{h}\right] \\ &= \lim_{h\to 0}\frac{E[l_B^1(\theta+h)] - E[l_B^1(\theta)]}{h} \\ &= \frac{d}{d\theta}E[l_B^1(\theta)] \end{aligned}$$

Case II: For $m = 2, ..., M$, we have the following.

$$\begin{aligned} \left|\frac{dl_B^m}{d\theta}\right| &= \left|\frac{l_B^m}{\theta}\right| \\ &= \frac{1}{\theta\tau_B^m}\left|\sum_{b=1}^{B}\sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}}\frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2}\right|. \end{aligned}$$

Since $v_j^m, p_j^m \leq (c_{m-1} - c_m)[\sigma_{b+1}^m - \sigma_b^m] \leq (c_{m-1} - c_m)\tau_B^m$, where b is the index of the busy period to which j belongs, we have

$$\begin{aligned}
\left|\frac{dl_B^m}{d\theta}\right| &\leq \frac{1}{\theta\tau_B^m}\left|\sum_{b=1}^{B}\sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}}(c_{m-1}-c_m)\tau_B^m\frac{(1+1)B_j^m+(1+1)I_j^m}{2}\right| \\
&\leq \frac{(c_{m-1}-c_m)}{\theta}\left|\sum_{b=1}^{B}\sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}}(B_j^m+I_j^m)\right| \\
&\leq \frac{(c_{m-1}-c_m)}{\theta}\tau_B^m.
\end{aligned}$$

It follows that for any $\theta \in \Theta$,

$$E[\left|\frac{dl_B^m}{d\theta}\right|] \leq \frac{c_{m-1}-c_m}{\theta}BE[\sigma_{b+1}^m - \sigma_b^m] < \infty,$$

which means that

$$E[\sup_{\theta\in\Theta}\left|\frac{l_B^m(\theta)}{d\theta}\right|] < \infty.$$

Furthermore, $l_B^m(\theta)$ is continuously differential in θ, according to the Generalized Mean Value Theorem for any $\theta, \theta + h \in \Theta$, we have

$$\left|\frac{l_B^m(\theta+h) - l_B^m(\theta)}{\theta}\right| \leq \sup_{\theta\in\Theta}\left|\frac{l_B^m(\theta)}{\theta}\right|.$$

The right hand side is integrable; the Dominated Convergence Theorem applies, and for any $\theta \in \Theta$, we have

$$\begin{aligned}
E\left[\frac{dl_B^m(\theta)}{d\theta}\right] &= E\left[\lim_{h\to 0}\frac{l_B^m(\theta+h) - l_B^m(\theta)}{h}\right] \\
&= \lim_{h\to 0}E\left[\frac{l_B^m(\theta+h) - l_B^m(\theta)}{h}\right] \\
&= \lim_{h\to 0}\frac{E[l_B^m(\theta+h)] - E[l_B^m(\theta)]}{h} \\
&= \frac{d}{d\theta}E[l_B^m(\theta)]
\end{aligned}$$

Proof is over. □

Proof of Theorem 2.3.2

Proof. For $m = 1$, we have the following.

Let $\widetilde{L}_b^1$ and $\widetilde{T}_b^1$ denote the area over the b^{th} busy period of $q_1(t)$ and its duration time. They are given by

$$\widetilde{L}_b^1 = c_1 \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i \Big(\sum_{j=n_{C,b-1}^1+1}^{i-1} Z_j - \sum_{j=n_{C,b-1}^1+2}^{i} Y_j \Big) + \frac{c_1}{2} \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i^2,$$

and

$$\widetilde{T}_b^1 = \sum_{j=n_{C,b-1}^1+2}^{n_{C,b}^1+1} Y_j.$$

As Z_i's and Y_i's are continuously differentiable at θ, so are $\widetilde{L}_b^1$ and $\widetilde{T}_b^1$.

Under C. 2.3.2 and Lemma 2.3.1, we further have

$$\frac{d\widetilde{L}_b^1}{d\theta} = 2\frac{\widetilde{L}_b^1}{\theta} \text{ and } \frac{d\widetilde{T}_b^1}{d\theta} = 2\frac{\widetilde{T}_b^1}{\theta}.$$

For $E[|\widetilde{L}_b^1|]$ and $E[|\widetilde{T}_b^1|]$, we have

$$\begin{aligned}
E[|\widetilde{L}_b^1|] &\le E[c_1 \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i \sum_{j=n_{C,b-1}^1+1}^{i-1} Z_j + c_1 \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i \sum_{j=n_{C,b-1}^1+2}^{i} Y_j + \frac{c_1}{2} \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i^2] \\
&\le E[c_1 \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i(\sigma_{b+1}^1 - \sigma_b^1) + c_1 \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i(\sigma_{b+1}^1 - \sigma_b^1) + \frac{c_1}{2} \sum_{i=n_{C,b-1}^1+1}^{n_{C,b}^1} Z_i(\sigma_{b+1}^1 - \sigma_b^1)] \\
&\le \frac{5c_1}{2} E[(\sigma_{b+1}^1 - \sigma_b^1)^2] \\
&< \infty
\end{aligned}$$

and

$$
\begin{aligned}
E[|\widetilde{T}_b^1|] &= E[|\sum_{j=n_{C,b-1}^1+2}^{n_{C,b}^1+1} Y_j|] \\
&\leq E[(\sigma_{b+1}^1 - \sigma_b^1)] \\
&< \infty
\end{aligned}
$$

Therefore, for $\theta \in \Theta$, we have

$$
E[\left|\frac{d\widetilde{L}_b^1}{d\theta}\right|] = 2E[\left|\frac{\widetilde{L}_b^1}{\theta}\right|] < \infty \text{ and } E[\left|\frac{d\widetilde{T}_b^1}{d\theta}\right|] = 2E[\left|\frac{\widetilde{T}_b^1}{\theta}\right|] < \infty.
$$

It follows that

$$
E[\sup_{\theta\in\Theta}\left|\frac{d\widetilde{L}_b^1}{d\theta}\right|] < \infty \text{ and } E[\sup_{\theta\in\Theta}\left|\frac{d\widetilde{T}_b^1}{d\theta}\right|] < \infty.
$$

Considering that $\widetilde{L}_b^1$ and $\widetilde{T}_b^1$ are continuously differentiable in θ, the Generalized Mean Value Theorem leads to that for any $\theta, \theta + h \in \Theta$ we have

$$
\left|\frac{\widetilde{L}_b^1(\theta+h) - \widetilde{L}_b^1(\theta)}{h}\right| \leq \sup_{\theta\in\Theta}\left|\frac{d\widetilde{L}_b^1(\theta)}{d\theta}\right|,
$$

and

$$
\left|\frac{\widetilde{T}_b^1(\theta+h) - \widetilde{T}_b^1(\theta)}{h}\right| \leq \sup_{\theta\in\Theta}\left|\frac{d\widetilde{T}_b^1(\theta)}{d\theta}\right|.
$$

Applying the Dominated Convergence Theorem, we have that for $\theta \in \Theta$,

$$
\begin{aligned}
E\left[\frac{d\widetilde{L}_b^1}{d\theta}\right] &= E\left[\lim_{h\to 0}\frac{\widetilde{L}_b^1(\theta+h) - \widetilde{L}_b^1(\theta)}{h}\right] \\
&= \lim_{h\to 0} E\left[\frac{\widetilde{L}_b^1(\theta+h) - \widetilde{L}_b^1(\theta)}{h}\right] \\
&= \lim_{h\to 0}\frac{E[\widetilde{L}_b^1(\theta+h)] - E[\widetilde{L}_b^1(\theta)]}{h} \\
&= \frac{d}{d\theta}E[\widetilde{L}_b^1]
\end{aligned}
$$

and

$$
\begin{aligned}
E\left[\frac{d\widetilde{T}_b^1}{d\theta}\right] &= E\left[\lim_{h\to 0}\frac{\widetilde{T}_b^1(\theta+h)-\widetilde{T}_b^1(\theta)}{h}\right] \\
&= \lim_{h\to 0} E\left[\frac{\widetilde{T}_b^1(\theta+h)-\widetilde{T}_b^1(\theta)}{h}\right] \\
&= \lim_{h\to 0}\frac{E[\widetilde{T}_b^1(\theta+h)]-E[\widetilde{T}_b^1(\theta)]}{h} \\
&= \frac{d}{d\theta}E[\widetilde{T}_b^1]
\end{aligned}
$$

Now we suffice to show the strong consistence. We start with

$$
\begin{aligned}
\lim_{B\to\infty}\frac{dl_B^1}{d\theta} &= \lim_{B\to\infty}\frac{d}{d\theta}\frac{L_B^1}{\tau_B^1} \\
&= \lim_{B\to\infty}\frac{d}{d\theta}\frac{\sum_{b=1}^B \widetilde{L}_b^1}{\sum_{b=1}^B \widetilde{T}_b^1} \\
&= \lim_{B\to\infty}\frac{\sum_{b=1}^B \widetilde{L}_b^{1\prime}}{\sum_{b=1}^B \widetilde{T}_b^1} - \frac{\sum_{b=1}^B \widetilde{L}_b^1}{\sum_{b=1}^B \widetilde{T}_b^1}\frac{\sum_{b=1}^B \widetilde{T}_b^{1\prime}}{\sum_{b=1}^B \widetilde{T}_b^1} \\
&= \frac{E[\widetilde{L}_b^{1\prime}]}{E[\widetilde{T}_b^1]} - \frac{E[\widetilde{L}_b^1]}{E[\widetilde{T}_b^1]}\frac{E[\widetilde{T}_b^{1\prime}]}{E[\widetilde{T}_b^1]} \\
&= \frac{E[\widetilde{L}_b^1]'}{E[\widetilde{T}_b^1]} - \frac{E[\widetilde{L}_b^1]}{E[\widetilde{T}_b^1]}\frac{E[\widetilde{T}_b^1]'}{E[\widetilde{T}_b^1]} \\
&= \Big(\frac{E[\widetilde{L}_b^1]}{E[\widetilde{T}_b^1]}\Big)' \\
&= \frac{dl^1}{d\theta}
\end{aligned}
$$

For $m = 2, ..., M$, we have the following.

Let $\widetilde{L}_b^m$ and $\widetilde{T}_b^m$ denote the area over the b^{th} busy period of $q_m(t)$ and its duration time. They are given by

$$
\widetilde{L}_b^m = \sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2},
$$

and

$$\widetilde{T}_b^m = \sum_{j=n_{C,b-1}^m+2}^{n_{C,b}^m+1} Y_j.$$

As Z_i's and Y_i's are continuously differentiable at θ, so are $\widetilde{L}_b^m$ and $\widetilde{T}_b^m$.

Under C. 2.3.2 and Lemma 2.3.2, we further have

$$\frac{d\widetilde{L}_b^m}{d\theta} = 2\frac{\widetilde{L}_b^m}{\theta} \text{ and } \frac{d\widetilde{T}_b^m}{d\theta} = 2\frac{\widetilde{T}_b^m}{\theta}.$$

For $E[|\widetilde{L}_b^m|]$ and $E[|\widetilde{T}_b^m|]$, we have

$$\begin{aligned} E[|\widetilde{L}_b^m|] &\leq E[\sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2}] \\ &\leq E[(c_{m-1} - c_m)(\sigma_{b+1}^m - \sigma_b^m) \sum_{j=n_{B,b-1}^{m-1}+1}^{n_{B,b}^{m-1}} (B_j^m + I_j^m)] \\ &\leq (c_{m-1} - c_m)E[(\sigma_{b+1}^1 - \sigma_b^1)^2] \\ &< \infty \end{aligned}$$

and

$$\begin{aligned} E[|\widetilde{T}_b^m|] &= E[|\sum_{j=n_{C,b-1}^m+2}^{n_{C,b}^m+1} Y_j|] \\ &\leq E[(\sigma_{b+1}^m - \sigma_b^m)] \\ &< \infty \end{aligned}$$

Therefore, for $\theta \in \Theta$, where Θ is a compact set, we have

$$E[\left|\frac{d\widetilde{L}_b^m}{d\theta}\right|] = 2E[\left|\frac{\widetilde{L}_b^m}{\theta}\right|] < \infty \text{ and } E[\left|\frac{d\widetilde{T}_b^m}{d\theta}\right|] = 2E[\left|\frac{\widetilde{T}_b^m}{\theta}\right|] < \infty.$$

It follows that

$$E[\sup_{\theta\in\Theta}\left|\frac{d\widetilde{L}_b^m}{d\theta}\right|] < \infty \text{ and } E[\sup_{\theta\in\Theta}\left|\frac{d\widetilde{T}_b^m}{d\theta}\right|] < \infty.$$

Considering that $\widetilde{L}_b^m$ and $\widetilde{T}_b^m$ are continuously differentiable in θ, the Generalized Mean Value Theorem leads to that for any $\theta, \theta + h \in \Theta$ we have

$$\left| \frac{\widetilde{L}_b^m(\theta + h) - \widetilde{L}_b^m(\theta)}{h} \right| \leq \sup_{\theta \in \Theta} \left| \frac{d\widetilde{L}_b^m(\theta)}{d\theta} \right|,$$

and

$$\left| \frac{\widetilde{T}_b^m(\theta + h) - \widetilde{T}_b^m(\theta)}{h} \right| \leq \sup_{\theta \in \Theta} \left| \frac{d\widetilde{T}_b^m(\theta)}{d\theta} \right|.$$

Applying the Dominated Convergence Theorem, we have that for $\theta \in \Theta$,

$$\begin{aligned} E\left[\frac{d\widetilde{L}_b^m}{d\theta}\right] &= E\left[\lim_{h \to 0} \frac{\widetilde{L}_b^m(\theta + h) - \widetilde{L}_b^m(\theta)}{h}\right] \\ &= \lim_{h \to 0} E\left[\frac{\widetilde{L}_b^m(\theta + h) - \widetilde{L}_b^m(\theta)}{h}\right] \\ &= \lim_{h \to 0} \frac{E[\widetilde{L}_b^m(\theta + h)] - E[\widetilde{L}_b^m(\theta)]}{h} \\ &= \frac{d}{d\theta} E[\widetilde{L}_b^m] \end{aligned}$$

and

$$\begin{aligned} E\left[\frac{d\widetilde{T}_b^m}{d\theta}\right] &= E\left[\lim_{h \to 0} \frac{\widetilde{T}_b^m(\theta + h) - \widetilde{T}_b^m(\theta)}{h}\right] \\ &= \lim_{h \to 0} E\left[\frac{\widetilde{T}_b^m(\theta + h) - \widetilde{T}_b^m(\theta)}{h}\right] \\ &= \lim_{h \to 0} \frac{E[\widetilde{T}_b^m(\theta + h)] - E[\widetilde{T}_b^m(\theta)]}{h} \\ &= \frac{d}{d\theta} E[\widetilde{T}_b^m] \end{aligned}$$

Now we suffice to show the strong consistence. We start with

$$
\begin{aligned}
\lim_{B\to\infty}\frac{dl_B^m}{d\theta} &= \lim_{B\to\infty}\frac{d}{d\theta}\frac{L_B^m}{\tau_B^m} \\
&= \lim_{B\to\infty}\frac{d}{d\theta}\frac{\sum_{b=1}^{B}\widetilde{L}_b^m}{\sum_{b=1}^{B}\widetilde{T}_b^m} \\
&= \lim_{B\to\infty}\frac{\sum_{b=1}^{B}\widetilde{L}_b^{m\prime}}{\sum_{b=1}^{B}\widetilde{T}_b^m} - \frac{\sum_{b=1}^{B}\widetilde{L}_b^m}{\sum_{b=1}^{B}\widetilde{T}_b^m}\frac{\sum_{b=1}^{B}\widetilde{T}_b^{m\prime}}{\sum_{b=1}^{B}\widetilde{T}_b^m} \\
&= \frac{E[\widetilde{L}_b^{m\prime}]}{E[\widetilde{T}_b^m]} - \frac{E[\widetilde{L}_b^m]}{E[\widetilde{T}_b^m]}\frac{E[\widetilde{T}_b^{m\prime}]}{E[\widetilde{T}_b^m]} \\
&= \frac{E[\widetilde{L}_b^m]'}{E[\widetilde{T}_b^m]} - \frac{E[\widetilde{L}_b^m]}{E[\widetilde{T}_b^m]}\frac{E[\widetilde{T}_b^m]'}{E[\widetilde{T}_b^m]} \\
&= \big(\frac{E[\widetilde{L}_b^m]}{E[\widetilde{T}_b^m]}\big)' \\
&= \frac{dl^m}{d\theta}
\end{aligned}
$$

Proof is over. □

Proof of Theorem 2.4.1

Proof. It is proven using induction.

For any $n_{C,b-1}^1 + 1 \le j \le n_{C,b}^1$, we have the following.

1. $j = n_{C,b-1}^1 + 1$: Since $q_1(A_{n_{C,b-1}^1+1}) = 0$, we have $\frac{dq_1(A_{n_{C,b-1}^1+1})}{d\theta} = \frac{q_1(A_{n_{C,b-1}^1+1})}{\theta}$.

2. $n_{C,b-1}^1 + 1 < j \le n_{C,b}^1$: Assume that it is true when $j = h-1$. When $j = h$, we have $\frac{dq_1(A_j)}{d\theta} = \frac{d(q_1(A_{j-1})-c_1Y_j+Z_j)}{d\theta} = \frac{d}{d\theta}(q_1(A_{j-1}) - c_1Y_j + Z_j) = \frac{1}{\theta}(q_1(A_{j-1}) - c_1Y_j + Z_j) = \frac{q_1(A_j)}{\theta}$.

Proof is over. □

Proof of Theorem 2.4.2

Proof. Since $q_{m-1}(t)$'s j-th busy period corresponds the j-th strictly ascending phase of $q_m(t)$, $q_m(D_j^{m-1}(\theta), \theta) = p_j^m$. Also since $\frac{dp_j^m}{d\theta} = \frac{p_j^m}{\theta}$, then $\frac{dq_m(D_j^{m-1}(\theta),\theta)}{d\theta} = \frac{q_m(D_j^{m-1}(\theta),\theta)}{\theta}$.

Proof is over. □

Proof of Theorem 2.4.3

Proof. For $m = 1$, it has been proven in proof of Theorem 2.3.1.

For $m = 2, ..., M$, under Lemma 2.3.2, Equation (2.12) leads to

$$\begin{aligned}
\frac{dl_B^m}{d\theta} &= \frac{1}{(\tau_B^m)^2}\Bigg[\sum_j^{n_{C,B}^m+1} Y_j\Bigg(\sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(\frac{d}{d\theta}v_j^m + \frac{d}{d\theta}p_j^m)B_j^m + (\frac{d}{d\theta}p_j^m + \frac{d}{d\theta}v_{j+1}^m)I_j^m}{2} + \\
&\quad \sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)\frac{d}{d\theta}B_j^m + (p_j^m + v_{j+1}^m)\frac{d}{d\theta}I_j^m}{2}\Bigg) - \\
&\quad \Bigg(\sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2}\Bigg)\sum_j^{n_{C,B}^m+1}\frac{d}{d\theta}Y_j\Bigg] \\
&= \frac{1}{(\tau_B^m)^2}\Bigg[\sum_j^{n_{C,B}^m+1} Y_j\Bigg(\sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(\frac{1}{\theta}v_j^m + \frac{1}{\theta}p_j^m)B_j^m + (\frac{1}{\theta}p_j^m + \frac{1}{\theta}v_{j+1}^m)I_j^m}{2} + \\
&\quad \sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)\frac{1}{\theta}B_j^m + (p_j^m + v_{j+1}^m)\frac{1}{\theta}I_j^m}{2}\Bigg) - \\
&\quad \Bigg(\sum_b^B\sum_j^{n_{B,b}^{m-1}} \frac{(v_j^m + p_j^m)B_j^m + (p_j^m + v_{j+1}^m)I_j^m}{2}\Bigg)\sum_j^{n_{C,B}^m+1}\frac{d}{d\theta}Y_j\Bigg] \\
&= \frac{l_B^m}{\theta}.
\end{aligned}$$

Proof is over. □

APPENDIX B

RELATED PROOFS ON THEOREMS IN CHAPTER 3

Proof of Theorem 3.3.1

Proof. According to the amplitude of λ (i.e., the input rate in steady state), we prove Theorem 3.3.1 in four cases. Note that $d_{FIFO} = 0$ for $\lambda \leq C$.

Case 1: $\lambda \leq \mu_{\min}$

$d = d_{pacer} - d_{FIFO} = 0 < \frac{Q_{\max}}{\mu_{\max}}$.

Case 2: $\mu_{\min} < \lambda \leq \mu_{\max}$

Without loss of generality, let $\lambda = \beta\mu_{\max} + (1-\beta)\mu_{\min}$, where $0 < \beta \leq 1$. Thus, we have

$$\begin{aligned} d = d_{pacer} - d_{FIFO} &= \frac{q_\lambda}{\mu} - 0 \\ &= \frac{Q_{\max}\frac{\lambda-\mu_{\min}}{\mu_{\max}-\mu_{\min}}}{\lambda} = \frac{Q_{\max}}{\mu_{\max} + \frac{1-\beta}{\beta}\mu_{\min}} \\ &\leq \frac{Q_{\max}}{\mu_{\max}}. \end{aligned}$$

Case 3: $\mu_{\max} \leq \lambda \leq C$

In this case, the pacing queue length stays at $Q_{\max}$, as demonstrated in Section 3.5.3. $d = d_{pacer} - d_{FIFO} = \frac{Q_{\max}}{\lambda} - 0 < \frac{Q_{\max}}{\mu_{\max}}$.

Case 4: $\lambda > C$

In this case the input traffic saturates the bottleneck link and overflows the router buffer. For the packets who successfully pass the delay/FIFO queue, we have $d = d_{pacer} - d_{FIFO} = \frac{Q_{\lim}}{C} - \frac{Q_{\lim}}{C} = 0 < \frac{Q_{\max}}{\mu_{\max}}$.

Thus, we always have $d \leq \frac{Q_{\max}}{\mu_{\max}}$ no matter how big λ is. Hence, Theorem 3.3.1 is proved.

□

Proof of Theorem 3.3.2

Proof. For the sake of clarity, we will use the subscript notations, i.e. write x_t for $x(t)$, etc. In steady state, the expectation of x_t is $E[x] \triangleq \lim_{t\to\infty} E[x_t] = \frac{r_1}{r_1+r_2}$ and its auto-covariance is $C_{xx}(\tau) \triangleq \lim_{t\to\infty} \mathrm{Cov}(x_t, x_{t+\tau}) = \frac{r_1 r_2}{(r_1+r_2)^2} e^{-(r_1+r_2)\tau}$. Therefore,

$$E[\lambda] \triangleq \lim_{t\to\infty} E[\lambda_t] = \lim_{t\to\infty} hE[x_t] = \frac{hr_1}{r_1+r_2},$$

and

$$C_{\lambda\lambda}(\tau) \triangleq \lim_{t\to\infty} \mathrm{Cov}(\lambda_t, \lambda_{t+\tau}) = \frac{h^2 r_1 r_2}{(r_1+r_2)^2} e^{-(r_1+r_2)\tau}.$$

Moreover,

$$E[\mu] \triangleq \lim_{t\to\infty} E[\mu_t] = E[\lambda] = \frac{hr_1}{r_1+r_2}.$$

Next we compute the steady-state cross-covariance $C_{x\mu}(\tau)$. Note that $d(x_t\mu_t) = \mu_t(1-x_t)dN_1 - \mu_t x_t dN_2 - \alpha x_t \mu_t dt + \alpha h x_t dt$. Taking expectations gives

$$E[x\mu] \triangleq \lim_{t\to\infty} E[\mu_t x_t] = \frac{hr_1(r_1+\alpha)}{(r_1+r_2)(r_1+r_2+\alpha)}.$$

Note also that $d(x_t\mu_s) = \mu_s(1-x_t)dN_1 - \mu_s x_t dN_2$. where s is held constant. Taking expectations gives

$$\frac{d}{dt}E[x_t\mu_s] = r_1 E[\mu_s] - (r_1+r_2)E[x_t\mu_s],$$

which yields

$$\begin{aligned} E[x_t\mu_s] =& \frac{r_1}{r_1+r_2}E[\mu_s] \\ &+ \left(E[x_s\mu_s] - \frac{r_1}{r_1+r_2}E[\mu_s]\right) e^{-(r_1+r_2)(t-s)}. \end{aligned}$$

Letting $t, s \to \infty$ such that $t - s = \tau$ is constant, we have

$$\begin{aligned} C_{x\mu}(\tau) &= \lim_{s\to\infty} E[x_{s+\tau}\mu_s] - E[x]E[\mu] \\ &= \frac{\alpha h r_1 r_2}{(r_1+r_2)^2(r_1+r_2+\alpha)} e^{-(r_1+r_2)\tau}. \end{aligned}$$

Finally, we compute the auto-covariance $C_{\mu\mu}(\tau)$. Note that $d\mu_t^2 = -2\alpha\mu_t^2 dt + 2\alpha h x_t \mu_t dt$. Taking expectations, we have

$$E[\mu^2] \triangleq \lim_{t\to\infty} E[\mu_t^2] = hE[x\mu] = \frac{h^2 r_1(r_1+\alpha)}{(r_1+r_2)(r_1+r_2+\alpha)}.$$

Note also that $d(\mu_t\mu_s) = -\alpha\mu_t\mu_s dt + \alpha h x_t \mu_s dt$, which, upon taking expectations, gives

$$\frac{d}{dt}E[\mu_t\mu_s] = -\alpha E[\mu_t\mu_s] + \alpha h E[x_t\mu_s].$$

Plugging in the formula for $E[x_t\mu_s]$ and solving for $E[\mu_t\mu_s]$,

$$E[\mu_t\mu_s] = \frac{hr_1}{r_1+r_2}E[\mu_s] + A(s)e^{-(r_1+r_2)(t-s)} + B(s)e^{-\alpha(t-s)}.$$

where $A(s) = \frac{\alpha h}{\alpha - r_1 - r_2}\left(E[x_s\mu_s] - \frac{r_1}{r_1+r_2}E[\mu_s]\right)$ and $B(s) = E[\mu_s^2] - \frac{hr_1}{r_1+r_2}E[\mu_s] - A(s)$, assuming $\alpha \neq r_1 + r_2$. Letting $t, s \to \infty$ such that $t - s = \tau$ is constant, we have

$$\begin{aligned} C_{\mu\mu}(\tau) &= \lim_{s\to\infty} E[\mu_{s+\tau}\mu_s] - (E[\mu])^2 \\ &= Ae^{-(r_1+r_2)\tau} + Be^{-\alpha\tau}. \end{aligned}$$

where A and B are as in the theorem. When $\alpha = r_1 + r_2$, l'Hôpital's rule gives

$$C_{\mu\mu}(\tau) = \frac{h^2 r_1 r_2}{2(r_1+r_2)^2}[1 + (r_1+r_2)\tau]e^{-(r_1+r_2)\tau}.$$

□

BIBLIOGRAPHY

[1] Adler, M., Khanna, S., Rajaraman, R., and Rosen, A. Time-constrained scheduling of weighted packets on trees and meshes. *Algorithmica 36*, 2 (2003), 123–152.

[2] Adler, M., Rosenberg, A. L., Sitaram, R. K., and Unger, W. Scheduling time-constrained communication in linear networks. *Theoretical Comp. Sc 35*, 6 (2002), 559–623.

[3] Aggarwal, Amit, Savage, Steve, and Anderson, Thomas. Understanding the performance of TCP pacing. In *Proc. of IEEE INFOCOM 2000* (Tel Aviv, Israel, Mar. 2000), pp. 1157–1165.

[4] Ahlswede, Rudolf, Cai, Ning, Li, Shuo-Yen Robert, and Yeung, Raymond W. Network information flow. *IEEE Transactions on Information Theory 46*, 4 (July 2000), 1204–1216.

[5] Allman, M., Paxson, V., and Stevens, W. RFC 3439: TCP Congestion Control.

[6] Alparslan, Onur, Arakawa, Shin'ichi, and Murata, Masayuki. Node pacing for optical packet switching. In *Profc. of Photonics in Switching, 2008* (Sapporo, Aug. 2008).

[7] Aron, Mohit, and Druschel, Peter. TCP: Improving startup dynamics by adaptive timers and congestion control. Technical Report TR98-318, Rice University, 1998.

[8] Balakrishnan, Hari, Padmanabhan, Venkata N., and Katz, Randy H. The effects of asymmetry on tcp performance. In *Mobicom 1997* (Sept. 1997).

[9] Blumenthal, D.J., Prucnal, P.R., and Sauer, J.R. Photonic packet switches: architectures and experimental implementations. *Proceedings of the IEEE 82*, 11 (nov 1994), 1650 –1667.

[10] Bonomi, F, and Fendick, K. The rate based flow control framework for the available bit rate ATM service. *IEEE Network Magazine* (1998), 25–39.

[11] Boyd, R.W., Bigelow, M.S., Lepeshkin, N., Schweinsberg, A., and Zerom, P. Fundamentals and applications of slow light in room temperature solids. In *Lasers and Electro-Optics Society, 2004. LEOS 2004. The 17th Annual Meeting of the IEEE* (nov. 2004), vol. 2, pp. 835 – 836 Vol.2.

[12] Brockett, Roger W., Gong, Weibo, and Guo, Yang. Stochastic analysis for fluid queueing systems. In *IEEE CDC '99* (Phoenix, AZ, Dec. 1999).

[13] Burmeister, E.F., and Bowers, J.E. Integrated gate matrix switch for optical packet buffering. *Photonics Technology Letters, IEEE 18*, 1 (Jan. 2006), 103 –105.

[14] Cai, Yan, Hanay, Sinan, and Wolf, Tilman. Practical packet pacing in small-buffer networks. In *ICC '09* (Dresden, Germany, June 2009).

[15] Chang, Cheng-Shang, Lee, Duan-Shin, and Jou, Yi-Shean. Load balanced Birkhoff-von neumann switches, part I: one-stage buffering. *Computer Communication 25* (2002), 611–622.

[16] Chang, Cheng-Shang, Lee, Duan-Shin, and Lien, Ching-Ming. Load balanced Birkhoff-von neumann switches, part II: multi-stage buffering. *Computer Communication 25* (2002), 623–634.

[17] Chiaroni, D. Packet switching matrix: a key element for the backbone and the metro. *Selected Areas in Communications, IEEE Journal on 21*, 7 (sept. 2003), 1018 – 1025.

[18] Clark, David D, Lambert, Mark M, and Zhang, Lixia. NETBLT: A high throughput transport protocol. *ACM SIGCOMM Comp. Comm. Rev. 17* (Aug. 1987), 353–359.

[19] Crovella, M., and Bestavros, A. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking 5* (Dec. 1997), 835–846.

[20] DeHart, John, Kuhns, Fred, Parwatikar, Jyoti, Turner, Jonathan, Wiseman, Charlie, and Wong, Ken. The open network laboratory: a resource for networking research and education. *ACM SIGCOMM Computer Communication Review 35*, 5 (Oct. 2005), 75–78.

[21] Dukkipati, Nandita, and McKeown, Nick. Why flow-completion time is the right metric for congestion control. *SIGCOMM Comput. Commun. Rev. 36* (January 2006), 59–62.

[22] Enachescu, Mihaela, Ganjali, Yashar, Goel, Ashish, Mckeown, Nick, and Roughgarden, Tim. Routers with very small buffers. In *Proc. of INFOCOM 2006* (Barcelona, Spain, Apr. 2006).

[23] Evans, John William, and Filsfils, Clarence. *Deploying IP and MPLS QoS for Multiservice Networks: Theory & Practice*. Morgan Kaufmann, 2007.

[24] Feldmann, A., Gilbert, A., Huang, P., and Willinger, W. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *Proc. of ACM SIGCOMM '99* (Aug. 1999), pp. 301–313.

[25] Floyd, S., Mahdavi, J., Mathis, M., and Romanow, A. RFC 2883: An extension to the selective acknowledgement (SACK) option for tcp.

[26] Floyd, S., and Paxson, V. Difficulties in simulation the Internet. *IEEE/ACM Transactions on Networking 9*, 4 (Aug. 2001), 392 – 403.

[27] Forum, ATM. *ATM User-Network Interface Specification*. Prentice Hall, 1993.

[28] Fredj, S. Ben, Bonald, T., Proutiere, A., Régnié, G., and Roberts, J.W. Statistical bandwidth sharing: A study of congestion at flow level. In *Proc. of ACM SIGCOMM 2001* (San Diego, CA, Aug. 2001).

[29] Glasserman, Paul. *Gradient Estimation Via Perturbation Analysis*. Kluwer, Boston, 1991.

[30] Gu, Yu, Towsely, Don, Hollot, Chris V., and Zhang, Honggang. Congestion control for small buffer high speed networks. In *Proc. of IEEE INFOCOM 07* (Anchorage, Alaska, May 2007), pp. 1037–1045.

[31] Heffes, H., and D.Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. In *IEEE Journal on selected areas in communications* (Sept. 1986), pp. 856–868.

[32] Ho, Y. C., Cao, Xiren, and Cassandras, Christos. Infinitesimal and finite perturbation analysis for queueing networks. In *IEEE CDC '82* (Dec. 1982).

[33] Hoe, J. Start-up dynamics of TCP's congestion control and avoidance schemes. Masterthesis, MIT, June 1995.

[34] Hoe, J. Start-up dynamics of TCP's congestion control and avoidance schemes. Masterthesis, MIT, June 1995.

[35] Hollot, C V., Liu, Yong, Misra, Vishal, and Towsley, Don. Unresponsive Flows and AQM Performance. In *Proc. of IEEE INFOCOM* (Apr 2003).

[36] Huang, Yong, Liu, Yong, Gong, Weibo, and Towsley, Don. Two-level Stochastic Fluid Tandem Queuing Model for Burst Impact Analysis. In *IEEE CDC '07* (New Orleans, LA, Dec. 2007).

[37] Hunter, D.K., and Andonovic, I. Approaches to optical internet packet switching. *Communications Magazine, IEEE 38*, 9 (sep 2000), 116 –122.

[38] inbound.cvec and outbound.cvec. http://www.cs.odu.edu/ netsim/ TrafGen/Traces-tmix-ccr06.

[39] International Organization for Standardization / International Electrotechnical Commission. *International Standard ISO/IEC 7498-1*, second ed. Geneve, Switzerland, Nov. 1994.

[40] Jain, R., Chiu, D., and Hawe, W. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report TR301, DEC, 1994.

[41] Khurgin, Jacob B. Optical buffers based on slow light in electromagnetically induced transparent media and coupled resonator structures: comparative analysis. *Journal of the Optical Society of America B 22*, 5 (oct. 2005), 1062–1074.

[42] Kleinrock, L. *Queueing Theory*. John Wiley, New York, 1975.

[43] Kurose, James F., and Ross, Keith W. *Computer Networking: A Top-Down Approach Featuring the Internet*, 4th ed. Addison Wesley, 2002.

[44] Lakshmikantha, A., Srikant, R., and Beck, C. Impact of file arrivals and departures on buffer sizing in core routers. In *Proc. of IEEE INFOCOM 08* (Pheonix, AZ, Apr. 2008), pp. 86–90.

[45] Lakshmikantha, Ashvin, Srikant, R., and Beck, Carolyn. Impact of File Arrivals and Departures on Buffer Sizing in Core Routers. In *Proc. of IEEE INFOCOM* (2008).

[46] Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking 2*, 1 (Feb. 1994), 1–15.

[47] Lindley, D. V. The theory of queues with a single server. *Proc. Cambridge Philos Soc. 48* (1952), 277–289.

[48] Liu, Yong, and Gong, Weibo. Perturbation Analysis for Stochastic Fluid Queueing Systems. In *IEEE CDC '99* (Dec. 1999), pp. 4440–4445.

[49] Loynes, R. M., and Walker, A. M. The stability of a queue with non-independent inter-arrival and service times. *Proc. Cambridge Philos Soc. 58*, 3 (1962), 497–520.

[50] Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. RFC 2018: Tcp selective acknowledgement options.

[51] Moon, Todd K. *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, June 2005.

[52] Mughal, S.S., Tucker, R.S., and Hinton, K. A taxonomy of optical buffer architectures. In *Optical Internet and Next Generation Network, 2006. COIN-NGNCON 2006. The Joint International Conference on* (july 2006), pp. 232 –234.

[53] Naor, J., Rosen, A., and Scalosub, G. Online time-constrained scheduling in linear networks. In *Proc. of IEEE INFOCOM 05* (Miami, FL, Mar. 2005).

[54] Nikolaidis, I., and Akyildiz, I. Source characterization and statistical multiplexing in ATM networks. Technical Report GIT-CC 92-24, Georgia Tech., 1992.

[55] Padmanabhan, Venkat N, and Katz, Randy H. TCP Fast Start: A technique for speeding up web transfers. In *Proc. of IEEE GLOBECOMM* (Sydney, Australia, Nov. 1998).

[56] Paxson, V., and Floyd, S. Wide-area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking 3*, 3 (June 1995), 226–244.

[57] Postel, Jon. User Datagram Protocol. RFC 768, Information Sciences Institute, Aug. 1980.

[58] Postel, Jon. Transmission Control Protocol. RFC 793, Information Sciences Institute, Sept. 1981.

[59] Raina, G, Towsely, D, and Wischik, D. Part II: Control theory for buffer sizing. *ACM SIGCOMM Comput Commun Rev* (July 2005), 79–82.

[60] Raina, G, Towsely, D, and Wischik, D. Part II: Control theory for buffer sizing. *ACM SIGCOMM Comput Commun Rev* (July 2005), 79–82.

[61] Raina, G, and Wischik, D. Buffer sizes for large multiplexers:TCP queuing theory and instabillity analysis. In *Proc. of EuroNGI* (Rome, Italy, Apr. 2005), pp. 173–180.

[62] Salehi, J. D., Zhang, Z., Kurose, J., and Towsley, D. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking, 6*, 4 (1998), 397–410.

[63] Shifrin, Mark, and Keslassy, Isaac. Small-buffer networks. *Computer Networks 53*, 14 (Sept. 2009), 2552–2565.

[64] Sivaraman, Vijay, Elgindy, Hossam, Moreland, David, and Ostry, Diethelm. Packet pacing in short buffer optical packet switched networks. In *Proc. of IEEE INFOCOM 06* (Spain, Apr. 2006).

[65] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[66] Triangular distribution. http://en.wikipedia.org/wiki/Triangular _distribution.

[67] Tucker, Rodney S. The role of optics and electronics in high-capacity routers. *Journal of Lightwave Technology 24*, 12 (2006), 4655–4673.

[68] Tucker, Rodney S., and Zhong, Wen De. Photonic packet switching: An overview. *IEICE Trans. Communi. E82-B*, 2 (1999), 254 –264.

[69] Tung, Tzu-Ying, Chen, Yin-Jieh, and Chang, Jin-Fu. Design and analysis of rc traffic shaper. *IEICE Transactions on Communications E81-B*, 1 (Jan 1998), 1–12.

[70] Vishwanath, Arun, Sivaraman, Vijay, Thottan, Marina, and Dovrolis, Constantine. Enabling a bufferless core network using edge-to-edge packet-level fec. In *Proc. of IEEE INFOCOM 10* (San Diego, CA, Mar. 2010).

[71] Visweswaraiah, V, and Heidermann, J. Improving restart of idle TCP connections. Technical Report TR97-661, University of Southern California, 1997.

[72] Vlasov, Yurii A., O'Boyle, Martin, Hamann, Hendrik F., and McNab, Sharee J. Active control of slow light on a chip with photonic crystal waveguides. *nature 438*, 7065 (nov. 2005), 65 –69.

[73] Weigle, Michele C., Adurthi, Prashanth, Jeffay, Felix Hernandez-Campos Kevin, and Smith, F. Donelson. Tmix: A tool for generating realistic tcp application workloads in ns-2. *SIGCOMM Computer Communication Review 36*, 3 (2006), 67–76.

[74] Willinger, W., Taqqu, M., Sherman, R., and Wilson, D. Self-similarity through highvariability: statistical analysis of ethernet lan traffic at the source level. 100–113.

[75] Willinger, Walter, Taqqu, Murad S., Sherman, Robert, and Wilson, Daniel V. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking 5* (1997), 71–86.

[76] Wischik, D, and McKeown, N. Part I: Buffer sizes for core routers. *ACM SIGCOMM Comput Commun Rev* (July 2005), 75–78.

[77] Wolf, Tilman, Cai, Yan, Kelly, Patrick A., and Gong, Weibo. Stochastic sampling for Internet traffic measurement. In *Proc. of 10th IEEE Global Internet Symposium* (Anchorage, AK, May 2007).

[78] Wolff, Ronald W. *Stochastic Modeling and The Theory of Queues*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[79] Wu, Yujing, Gong, Weibo, and Towsley, Don. Analysis of abstract simulation via stochastic differential equation models. In *IEEE CDC '03* (Dec 2003).

[80] Yang, Haijun, and Yoo, S.J.B. All-optical variable buffering strategies and switch fabric architectures for future all-optical data routers. *Lightwave Technology, Journal of 23*, 10 (oct. 2005), 3321 – 3330.

[81] Yao, Shun, Mukherjee, B., and Dixit, S. Advances in photonic packet switching: an overview. *Communications Magazine, IEEE 38*, 2 (feb 2000), 84 –94.

[82] Zhang, L, Shenker, S, and Clark, David D. Observations on the dynamics of a congestion control algorithm: the effects of two way traffic. In *Proc. of ACM SIGCOMM 91* (Zurich, Switzerland, Sept. 1991), pp. 133–147.