

Why this meeting?

Background

- Research papers on experiments and theories of buffer size.
- Yet, no universal agreement on how big router buffers should be, and why.
- Personal confession: I have no idea what the general answer is
 - Incast
 - Data centers
 - For specific environments, like financial networks, SLAs, HPC, ...

Our goal

- A workshop in October/November 2019: *“How Big Should Buffers be in Switches and Routers?”*
- Measurements: Invite operators of large networks to perform experiments in their networks.
- Theory: Invite researchers to develop theory explaining, supporting (challenging?) measurements.
- Report results publicly at workshop.
- Compare notes and write a report together, sharing our results to the world.

Organizers

1. Neda Beheshti
2. Christophe Diot
3. Tom Edsall
4. Nasser El-Aawar
5. Yashar Ganjali
6. Nick McKeown
7. Bruce Spang

Local logistics:
Andi Villanueva

Who we are

- Speakers from 14 companies and 2 universities
 - Network operators, cloud companies, router vendors, chip vendors
- Attendees from 22 companies and 2 universities
- Let's introduce ourselves...

Schedule for the day

10.30am – 1.30pm

Session 1: Network Operators

- Neda Beheshti Facebook
- Lincoln Dale Google
- TY Huang Netflix
- Hongqiang Liu Alibaba
- Ken Duell AT&T
- Joel Jaeggli Fastly

[12.00 – 12.30 Lunch]

- Simon Leinen Switch
- Bob Briscoe CableLabs
- Chuanxiong Guo Bytedance
- Igor Gashinsky Oath

1.45pm – 2.45pm

Session 2: Technology Providers

- Parvin Taheri Cisco
- Francois Labonte Arista
- Golan Schzukin Dune/BCM
- Chang Kim Barefoot

3.00pm – 4.00pm

Session 3: Discussion

- Conclusions Neda, Bruce, Nasser
- Actions and Next Steps Yashar, Nick

A brief history of buffer size

1988

1994

2019

Congestion Avoidance and Control

VJ & MK

High Performance TCP in ANSNET

CV & CS

Congestion Avoidance and Control*

Van Jacobson¹
Lawrence Berkeley Laboratory
Michael J. Karels²
University of California at Berkeley
November, 1988

Introduction

Computer networks have experienced an explosive growth over the past few years and with that growth have come severe congestion problems. For example, it is now common to see internet gateways drop 10% of the incoming packets because of local buffer overflows. Our investigation of some of these problems has shown that much of the cause lies in transport protocol implementations (not in the protocols themselves). The 'obvious' ways to implement a window-based transport protocol can result in exactly the wrong behavior in response to network congestion. We give examples of 'wrong' behavior and describe some simple algorithms that can be used to make right things happen. The algorithms are rooted in the idea of achieving network stability by forcing the transport connection to obey a 'packet conservation' principle. We show how the algorithms derive from this principle and what effect they have on traffic over congested networks.

In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and two BGP hops) dropped from 32 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad. In particular, we wondered if the 4.3BSD (Berkeley UNIX) TCP was mis-behaving or if it could be tuned to work better under abysmal network conditions. The answer to both of these questions was "yes".

*Note: This is a very slightly revised version of a paper originally presented at SIGCOMM '88 [1]. If you wish to reference this work, please cite [1].
This work was supported in part by the U.S. Department of Energy under Contract Number DE-AC02-80SF0008.
This work was supported by the U.S. Department of Commerce, National Bureau of Standards, under Grant Number 60NANB0093.

High Performance TCP in ANSNET

Chris Villamizar <cvillamizar@netsys.net>
Advanced Network & Services, Inc.
Chang Song <csong@vnet.ibm.com>
AdvanTis
September 12, 1994

Abstract

This report concentrates on specific experiments and gains of the research activities supported by ANSNET, but applies to any TCP-dominated high speed WAN and in particular those striving to support high speed read-to-read flow. Measurements have been made under random conditions to better understand performance barriers imposed by queuing capacities and queue drop strategies.

The IBM RS/6000 based system currently supporting ANSNET performed very well in these tests. Measurements have been made with the current software and performance enhanced software. Single TCP flows are able to achieve 40 Mbps and competing multiple TCP flows achieve over 11 Mbps link utilization on 147 Mbps DSL links with delay comparable to 40 error corrected ANSNET flows. Congestion collapse is demonstrated with intentionally reduced queuing capacity and using window sizes much larger than optimal.

A variation of Floyd and Jacobson's Random Early Detection (RED) algorithm [2] is tested. Performance improved with the use of RED for both arriving multiple flows. With RED and queuing capacity at or above the delay bandwidth product, congestion collapse is avoided, allowing the maximum window size to satisfy its self arbitrarily high.

Queuing capacity greater than or equal to the delay bandwidth product and RED are recommended. RED provides performance improvement in all but the single flow case, but cannot substitute for adequate queuing capacity, particularly at high speed flows or in burst periods.

Contents

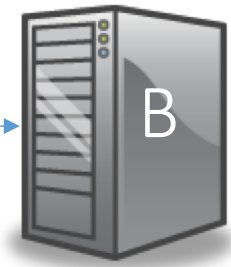
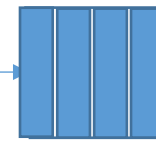
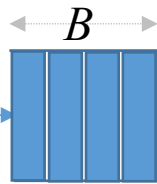
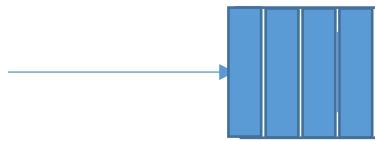
- 1 Introduction
- 2 TCP Segment Size
- 2.2 TCP Maximum Window Size
- 2.3 TCP Congestion Avoidance
- 2.4 Fast Retransmit and Recovery
- 2.5 Performance Details
- 3 Queue Size Requirements
- 3.1 Multiple TCP Flows
- 3.2 Effect of Queuing Capacity
- 4 Performance Testing
- 4.1 Test Network Conditions
- 4.2 Router Queuing Capacity
- 4.3 Traffic Sources
- 4.4 Summary of Test Conditions
- 5 Test Results
- 5.1 Single High Speed Flows
- 5.2 Multiple Flows
- 5.3 Reverse Flow
- 5.4 Random Early Detection
- 5.5 Fluctuation and Delay
- 5.6 Link Utilization Estimates
- 6 Recommendations
- 7 Other Considerations
- 8 Conclusions
- 9 Acknowledgments

$$B = 2T \times C$$

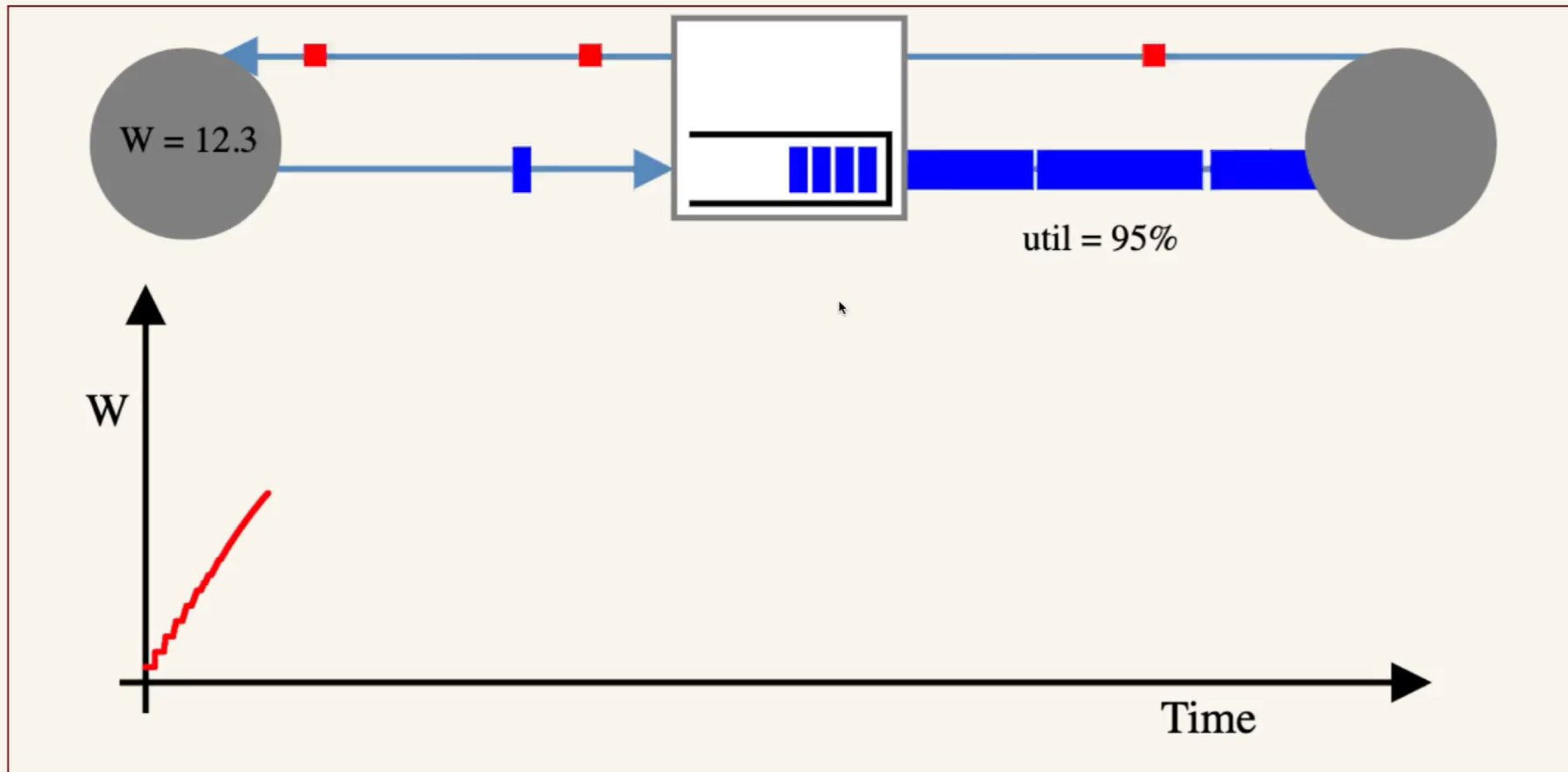
“Buffer size should equal the bandwidth delay product”

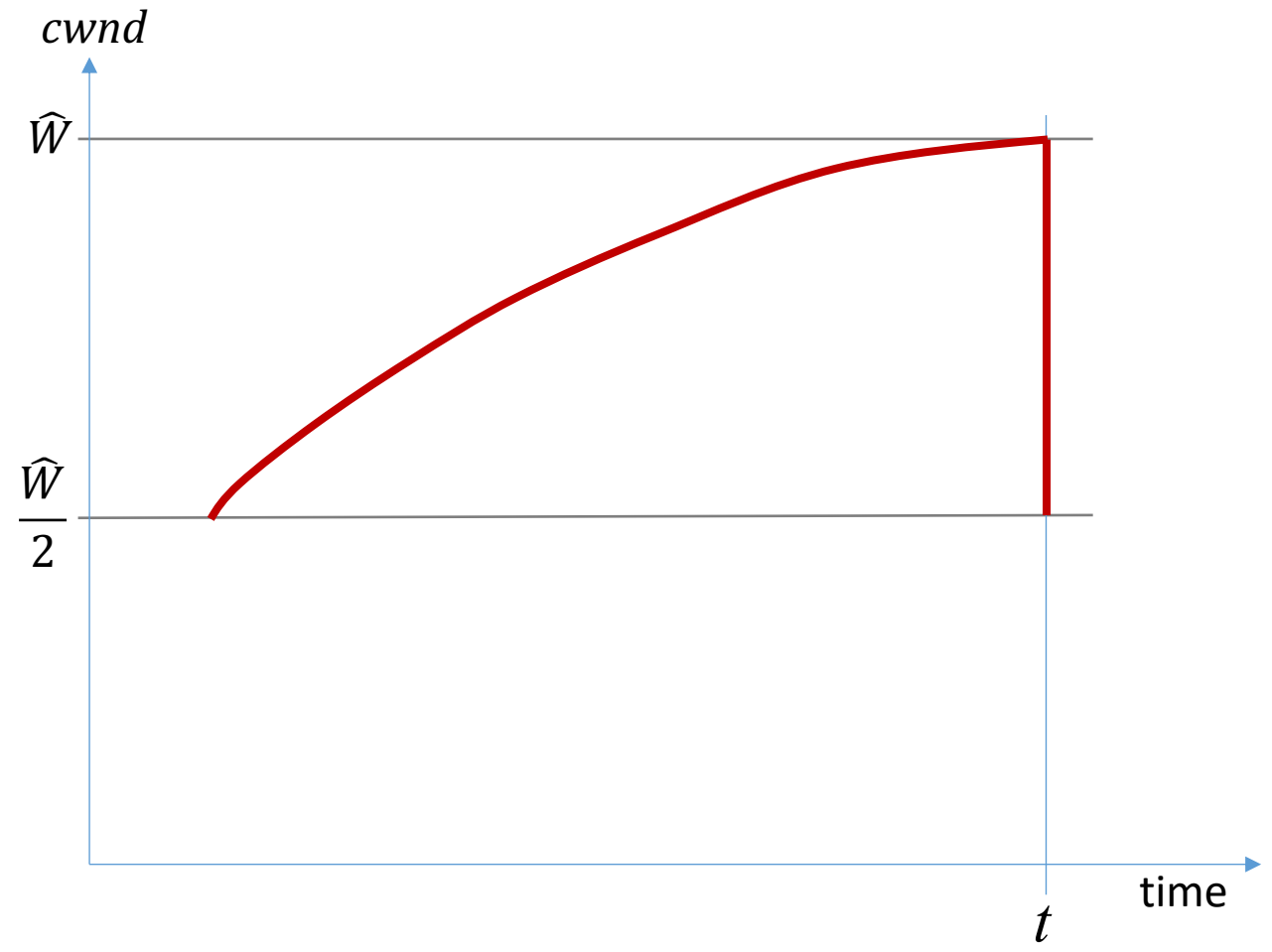
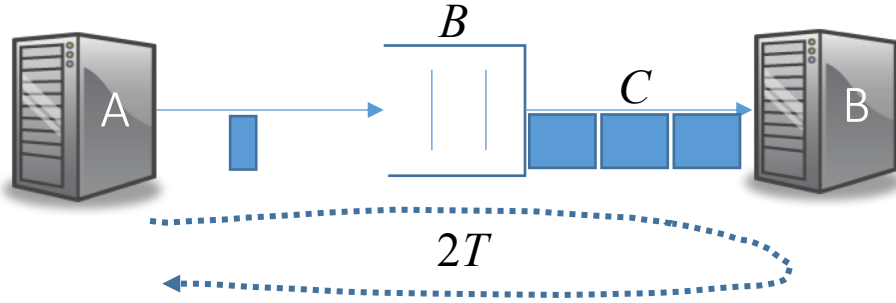
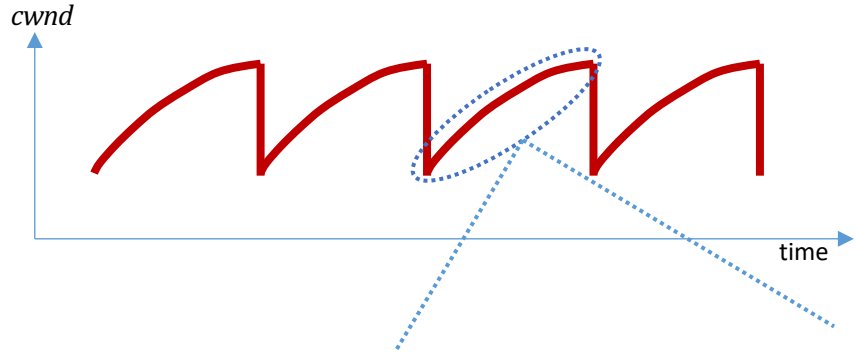
~~$$Max RTT = 2T + B/C = 4T$$~~

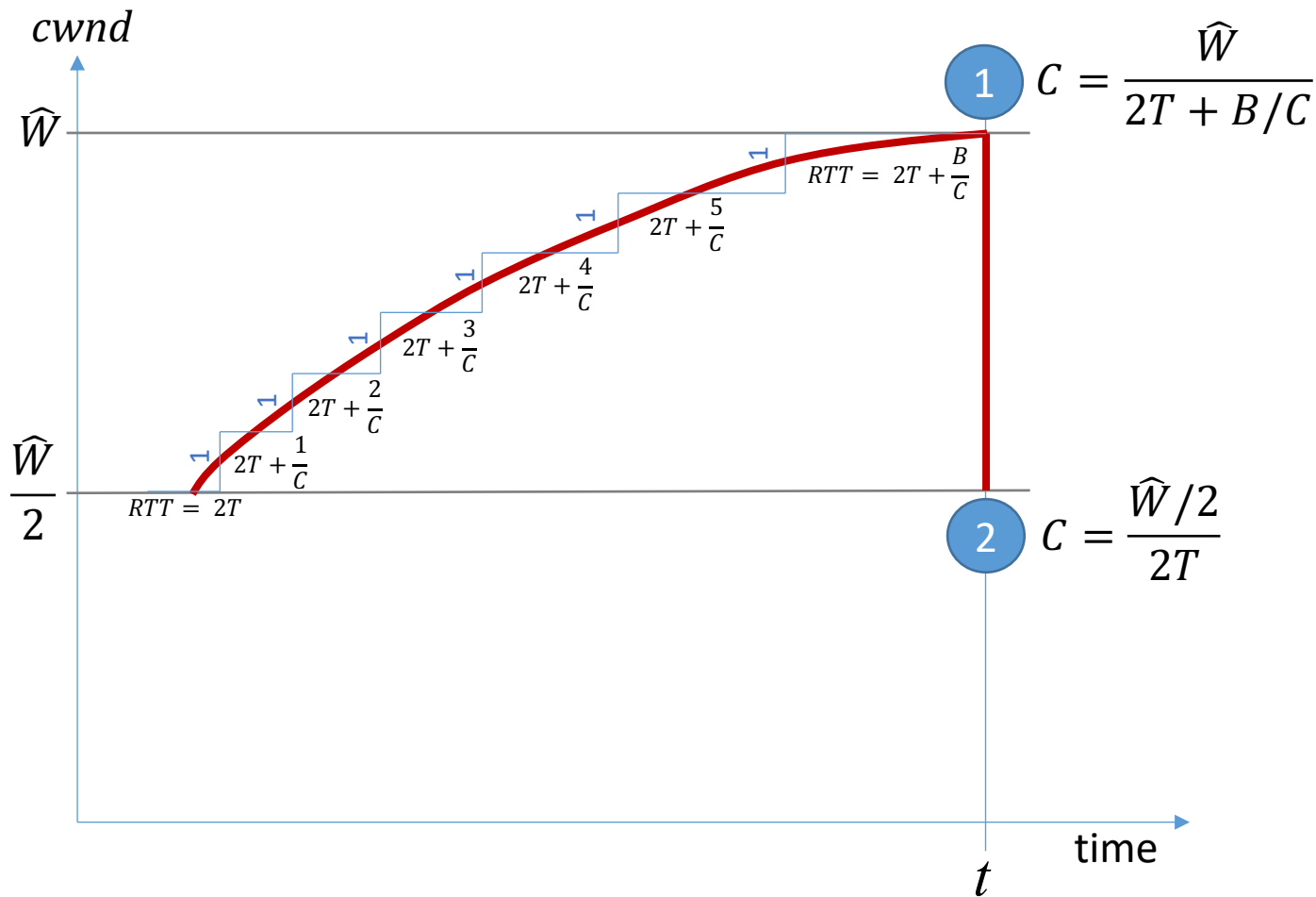
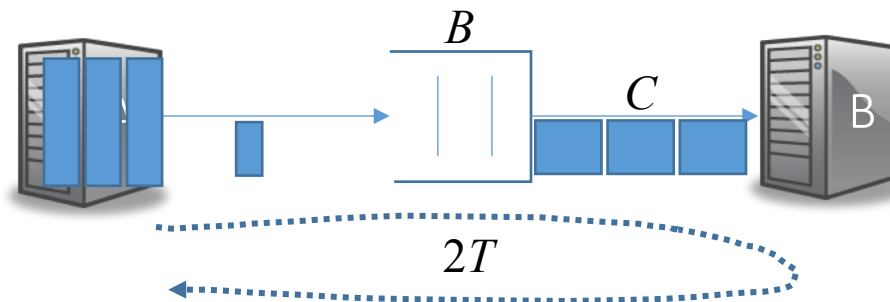
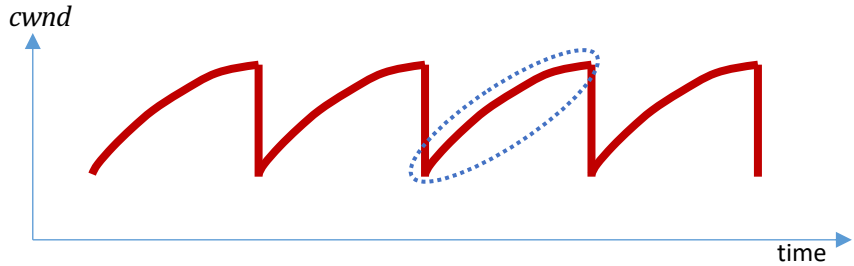
$$Max RTT = 2T + 4B/C = 10T$$



$$Min RTT = 2T$$



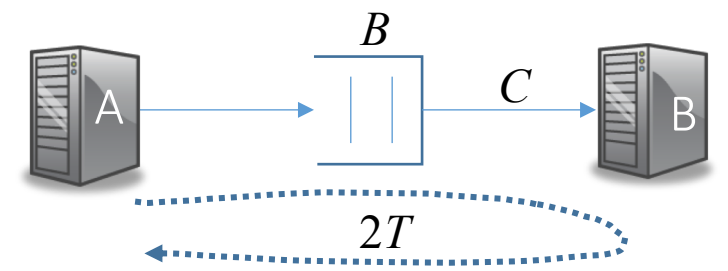




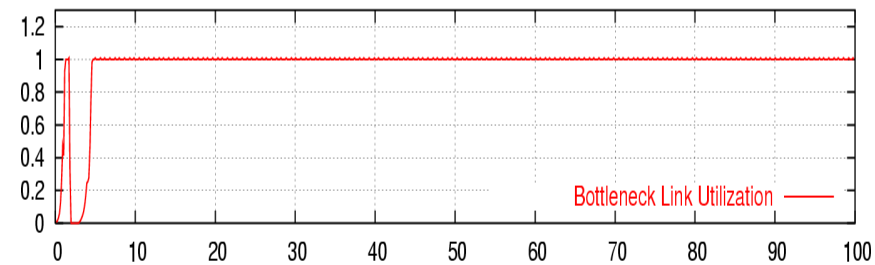
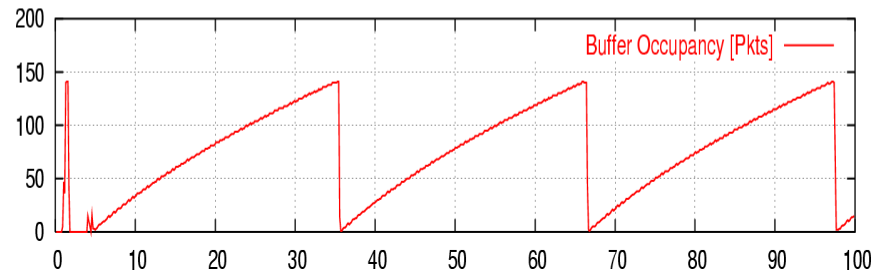
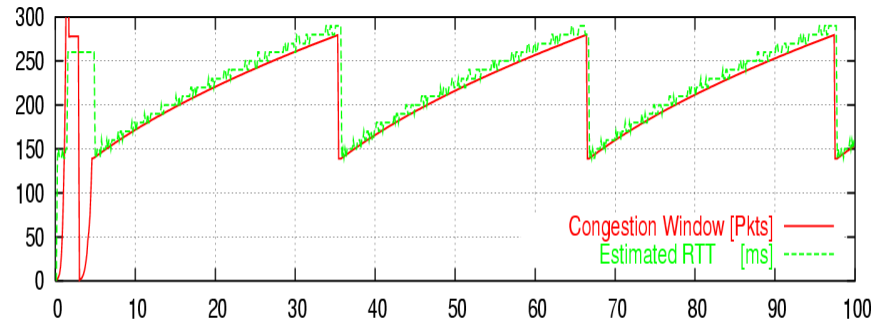
$$C = \frac{\hat{W}}{2T + B/C} = \frac{\hat{W}/2}{2T}$$

$$B = 2T \times C$$

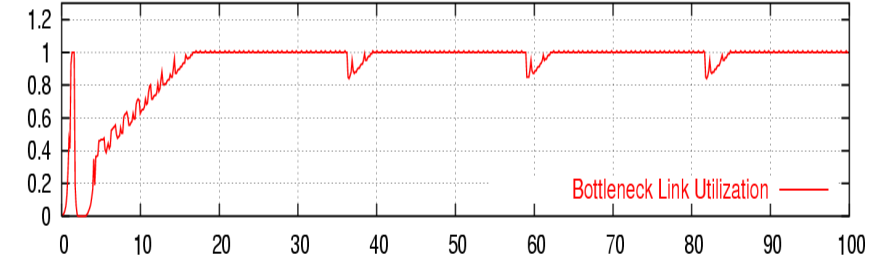
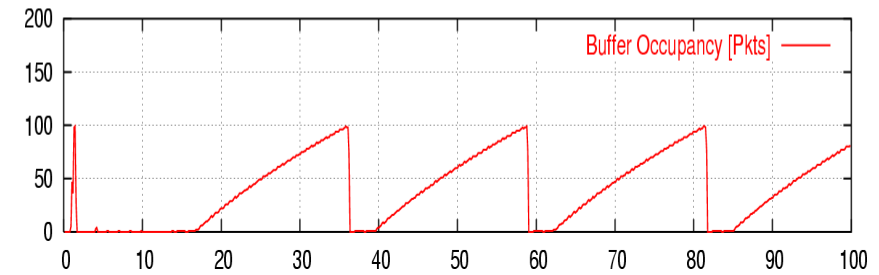
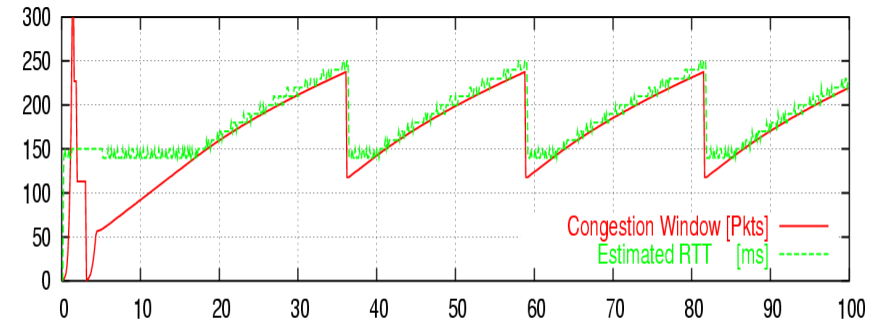
Time Evolution of a Single TCP Flow



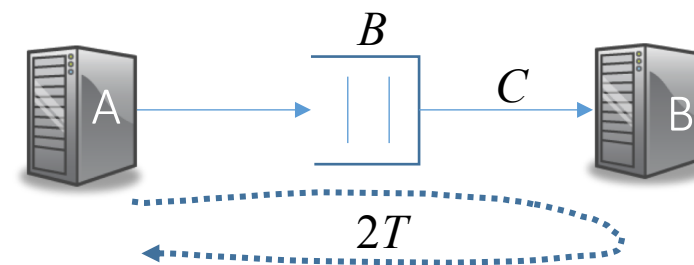
$$B = 2T \times C$$



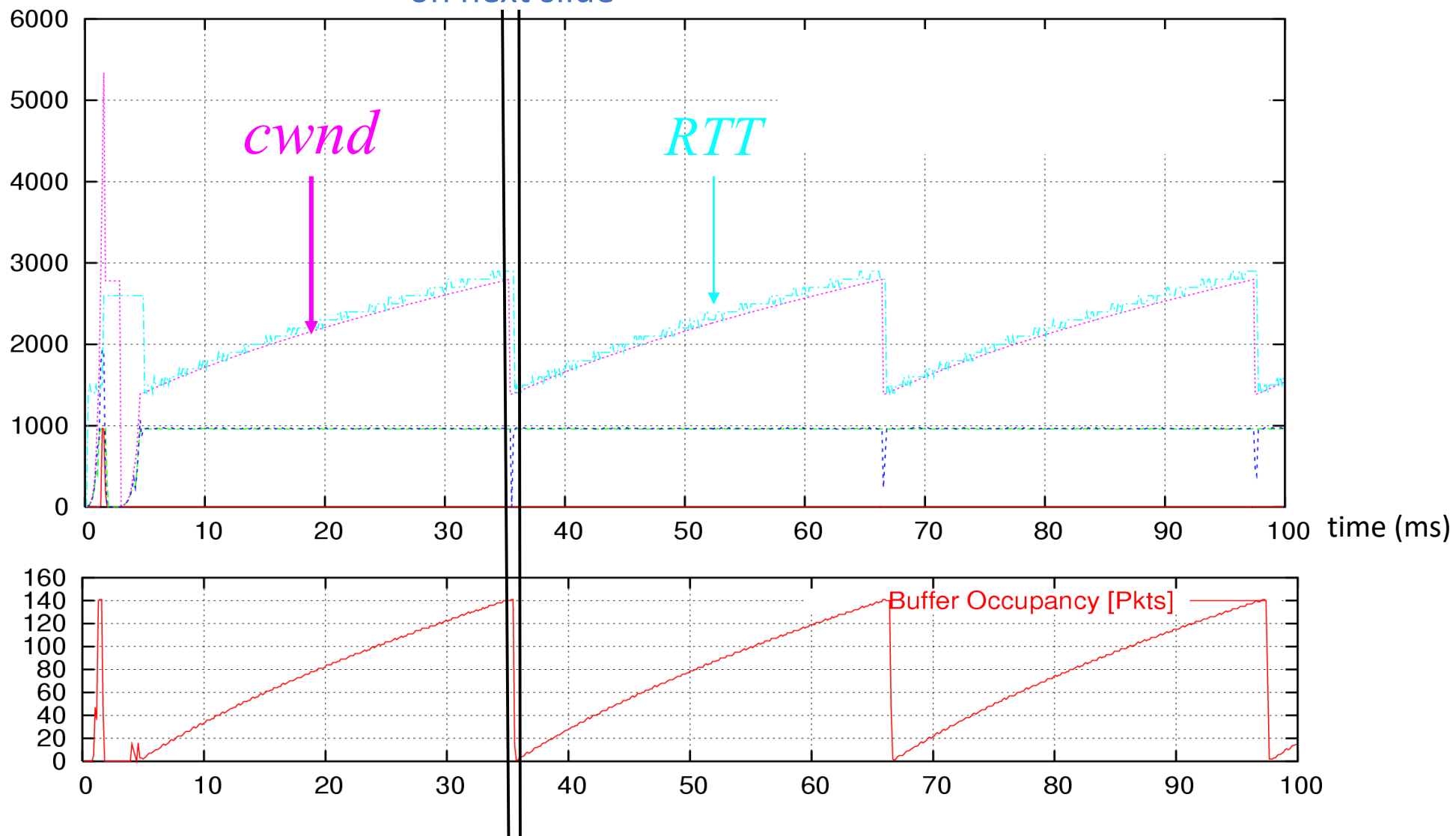
$$B < 2T \times C$$



$$B = 2T \times C$$

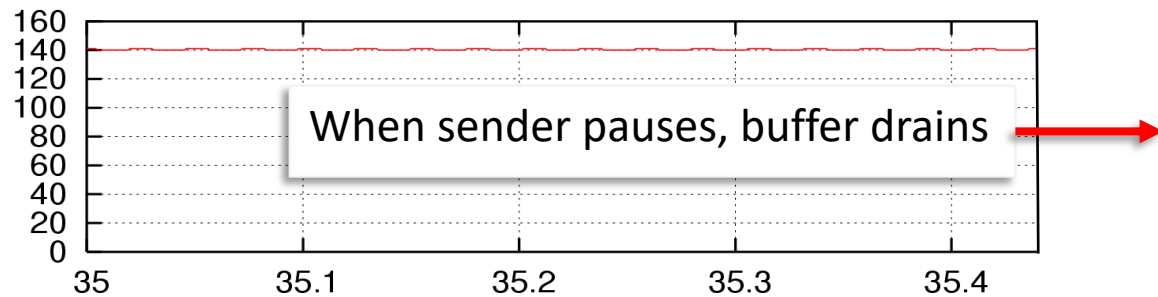
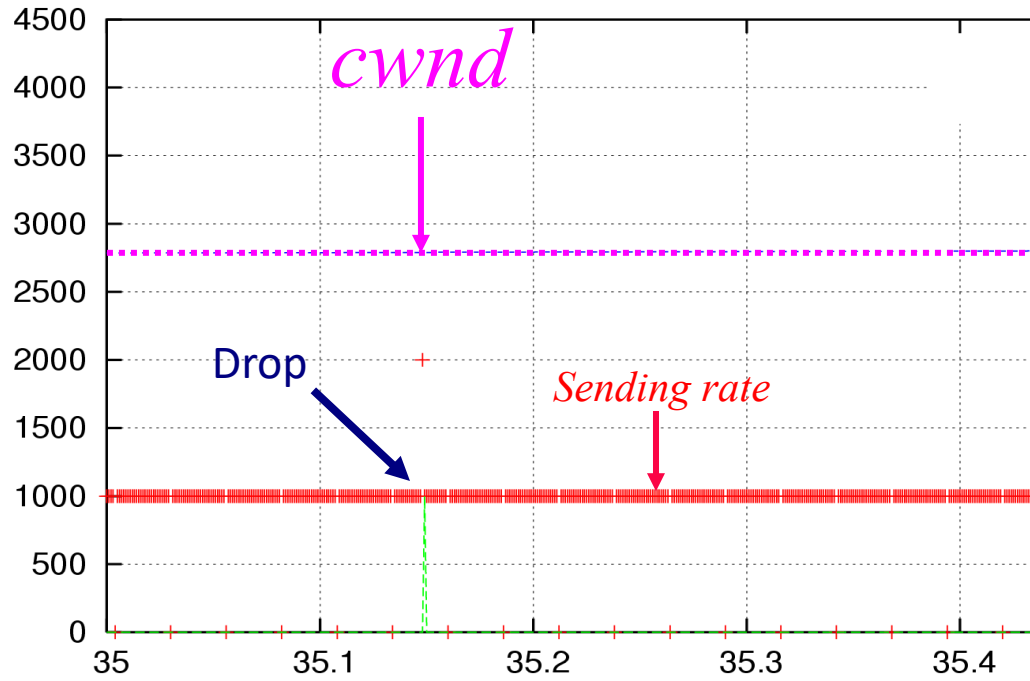
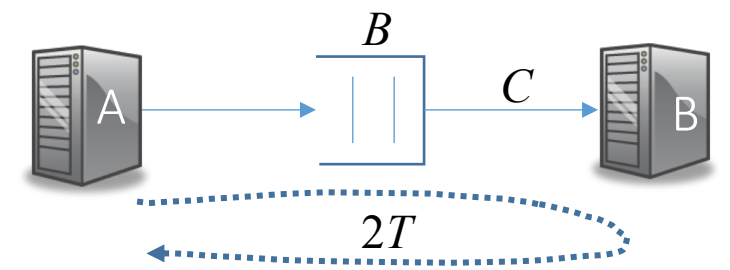


Interval magnified
on next slide



$$B = 2T \times C$$

Zoom View



Single AIMD flow: 100% Throughput

1. If $\hat{W} \rightarrow \frac{\hat{W}}{2}$ then $B \geq 2T \times C$

Example: $2T = 100ms, C = 10Gb/s$
 $B \geq 1Gbit$

2. If $\hat{W} \rightarrow \frac{\hat{W}}{k}$ then $B \geq 2T(k - 1) \times C$

Example: $k = 1.5$
 $B \geq 500Mbits$

Example: $k = 1.14$
 $B \geq 140Mbits$

3. If $k = 1 + \frac{a}{2T}$ then $B \geq aC$

Example: $a = \frac{1}{100}$
 $B \geq 50Mbits$

i.e. if end host knows $2T$, buffer size is independent of RTT

1988

1994

2004

2020

Congestion Avoidance and Control

VJ & MK

High Performance TCP in ANSNET

CV & CS

Sizing Router Buffers

GA, IK, NM

Congestion Avoidance and Control*

Van Jacobson¹
Lawrence Berkeley Laboratory
Michael J. Karels²
University of California at Berkeley
November, 1988

Introduction

Computer networks have experienced an explosive growth over the past few years and with that growth have come severe congestion problems. For example, it is now common to see internet gateways drop 10% of the incoming packets because of local buffer overflows. Our investigation of some of these problems has shown that much of the cause lies in transport protocol implementations (not in the protocols themselves). The 'obvious' ways to implement a window-based transport protocol can result in exactly the wrong behavior in response to network congestion. We give examples of 'wrong' behavior and describe some simple algorithms that can be used to make right things happen. The algorithms are rooted in the idea of achieving network stability by forcing the transport connection to obey a 'packet conservation' principle. We show how the algorithms derive from this principle and what effect they have on traffic over congested networks.

In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and two BGP hops) dropped from 32 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad. In particular, we wondered if the 4.3BSD (Berkeley UNIX) TCP was misbehaving or if it could be tuned to work better under abysmal network conditions. The answer to both of these questions was "yes".

*Note: This is a very slightly revised version of a paper originally presented at SIGCOMM '88 [12]. If you wish to reference this work, please cite [12].
This work was supported in part by the U.S. Department of Energy under Contract Number DE-AC02-80SF0080.
This work was supported by the U.S. Department of Commerce, National Bureau of Standards, under Grant Number 60NANB09030.

High Performance TCP in ANSNET

Chris Villanar court@lan.net
Advanced Network & Services, Inc.
Chang Song <chang@net.illm.com>
Advanis
September 12, 1994

Abstract

This report concentrates on specific experiments and gains of the research activities supported by ANSNET, but applies to any TCP-dominated high speed WAN and in particular those striving to support high speed real-time flows. Measurements have been made under real-time conditions to better understand performance barriers imposed by queuing capacities and queue drop strategies.

The IBM RS/6000 based routers currently supporting ANSNET performed very well in these tests. Measurements have been made with the current software and performance enhanced software. Single TCP flows are able to achieve 40 Mb/s and congestive multiple TCP flows achieve over 1 Mb/s link utilization on 1.5 Mb/s DSL links with delay comparable to 40 times current ANSNET delays. Congestion collapse is demonstrated with statistically reduced queuing capacity and using window sizes much larger than optimal.

A variation of Floyd and Jacobson's Random Early Detection (RED) algorithm [5] is tested. Performance improved with the use of RED for links arriving multiple flows. With RED and queuing capacity at or above the delay bandwidth product, congestion collapse is avoided, allowing the maximum window size to be set to be arbitrarily high.

Queuing capacity greater than we expect to the delay bandwidth product and RED are recommended. RED provides performance improvement in at least the single flow case, but cannot substitute for adequate queuing capacity, particularly at high speed flows so as to be supported.

Contents

1 Introduction

2 TCP Segment Size

2.2 TCP Maximum Window Size

2.3 TCP Congestion Avoidance

2.4 Fast Retransmit and Recovery

2.5 Performance Details

3 Queue Size Requirements

3.1 Multiple TCP Flows

3.2 Effects of Queuing Capacity

4 Performance Tuning

4.1 Test Network Conditions

4.2 Router Queuing Capacity

4.3 Traffic Sources

4.4 Summary of Test Conditions

5 Test Results

5.1 Single High-Speed Flows

5.2 Multiple Flows

5.3 Reverse Flow

5.4 Random Early Detection

5.5 Forward and Delay

5.6 Link Utilization Estimates

6 Recommendations

7 Other Considerations

8 Conclusions

9 Acknowledgments

Sizing Router Buffers

Guido Appenzeller
Stanford University
appenz@cs.stanford.edu

Isaac Keslassy
Stanford University
keslassy@yuba.stanford.edu

Nick McKeown
Stanford University
nickm@stanford.edu

ABSTRACT

All Internet routers contain buffers to hold packets during times of congestion. Today, the size of the buffers is determined by the dynamics of TCP's congestion control algorithm. In particular, the goal is to make sure that when a link congests, it is to have 20% of the capacity which is expected to be making use of the buffer when congestion occurs. A widely used related-claim states that each link needs a buffer of size $B = 2T \times C$, where T is the average round-trip time of a flow passing across the link, and C is the data rate of the link. For example, a 10Gbit/s router should need approximately 200ms \times 10Gbit/s = 2.0Gbit of buffers, and the amount of buffering grows linearly with the Internet.

Such large buffers are challenging for router manufacturers, who must use large, slow, off-chip DRAMs, and queuing delays can be long, have high variance, and may destabilize the congestion control algorithm. In this paper we argue that the related-claim ($B = 2T \times C$) is now outdated and incorrect for backbone routers. This is because of the large number of flows (TCP connections) multiplexed together on a single backbone link. Using theory, simulation and experiments on a network of real routers, we show that a link with N flows requires no more than $B = (2T \times C) / \sqrt{N}$, for long-lived or short-lived TCP flows. The consequences on router delays are enormous: a 2.0Gbit/s link carrying 10,000 flows could reduce its buffers by 90% with negligible difference in throughput, and a 10Gbit/s link carrying 10,000 flows requires only 100Mbit of buffering, which can easily be implemented using fast, on-chip SRAM.

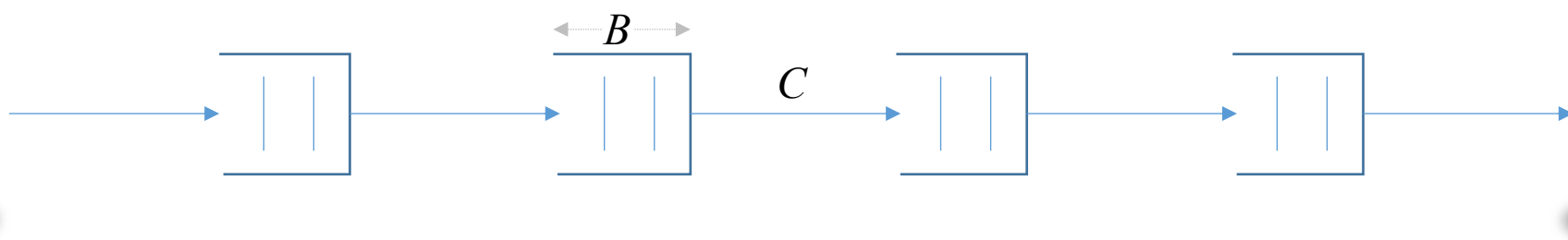
Categories and Subject Descriptors

C.2 [Internetworking]: Routers

*The authors are with the Computer Systems Laboratory at Stanford University. Some findings in our work with the Tekon network (based on building of Berkeley's backhaul). This work was funded in part by the *NSF and Networking Research Center*, the *Stanford Center for Integrated and Systems*, and a *Walter S. Guggenheim Graduate Fellowship*.

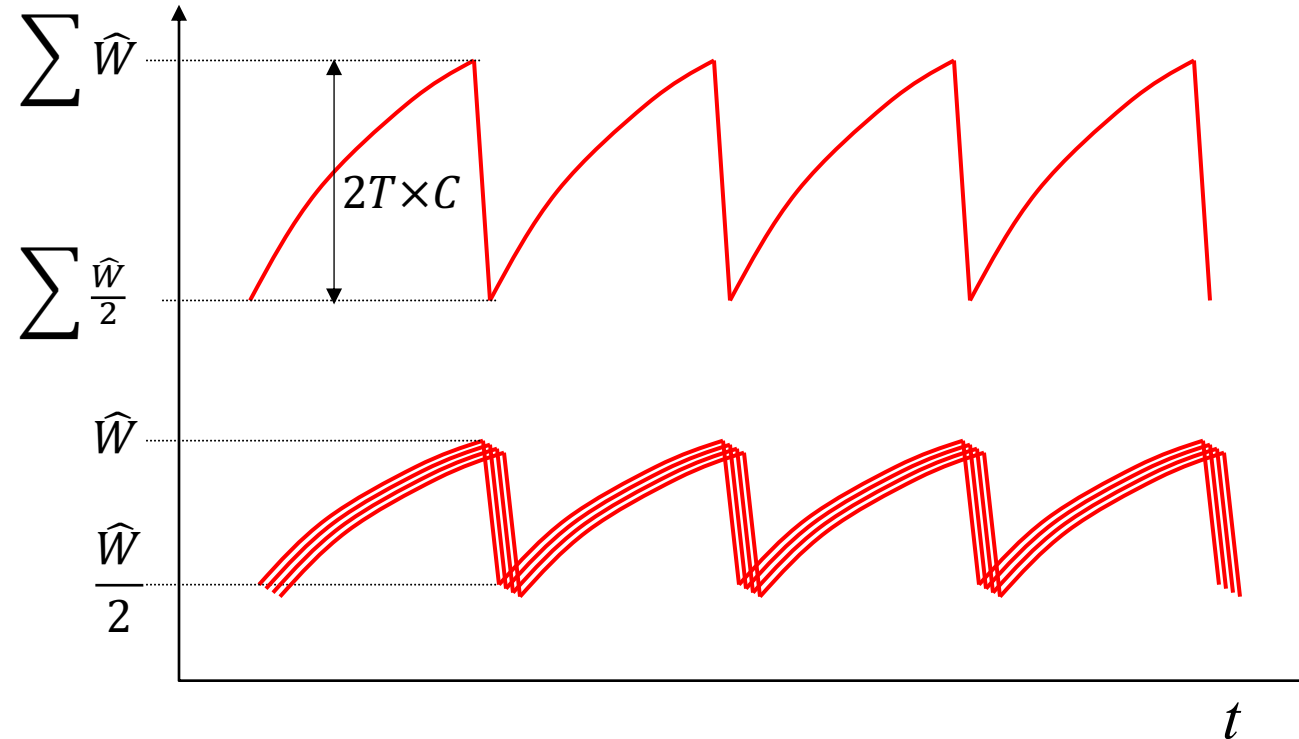
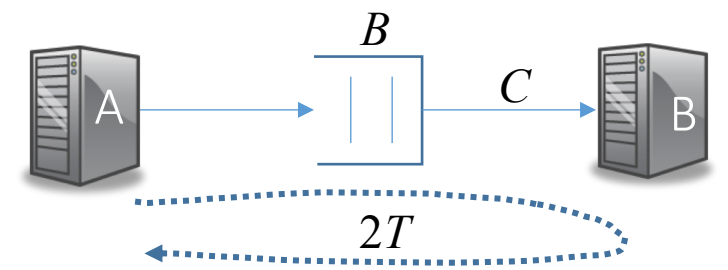
$$B \geq \frac{2T \times C}{\sqrt{N}}$$

where N is the number of long-lived flows



$$\text{Min RTT} = 2T$$

Synchronized Flows

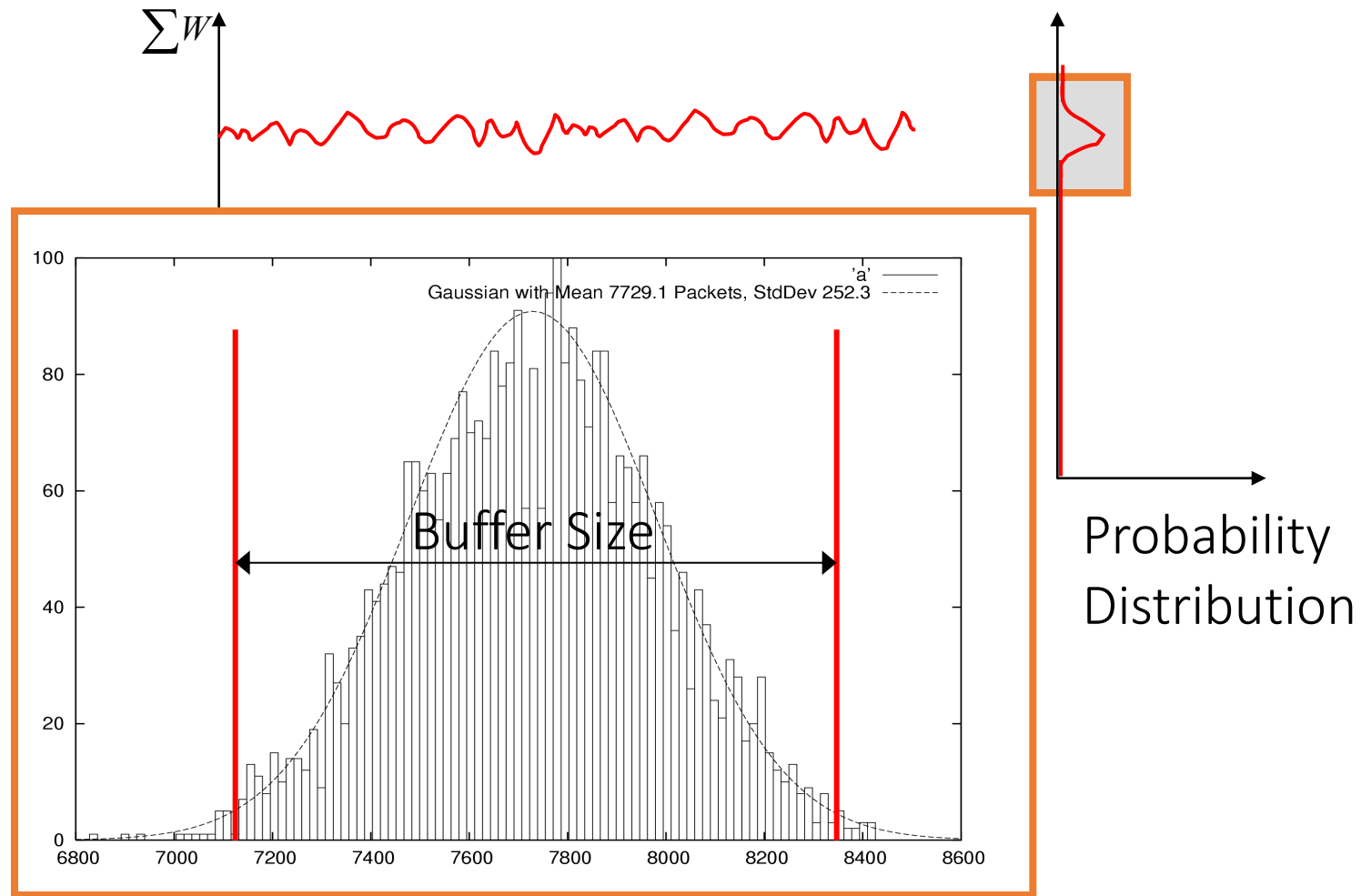


Aggregate window of all the flows has same dynamics

Therefore buffer occupancy has same dynamics

Rule-of-thumb $B \geq 2T \times C$ still holds.

Desynchronized TCP Flows



Many AIMD flows: 100% Throughput

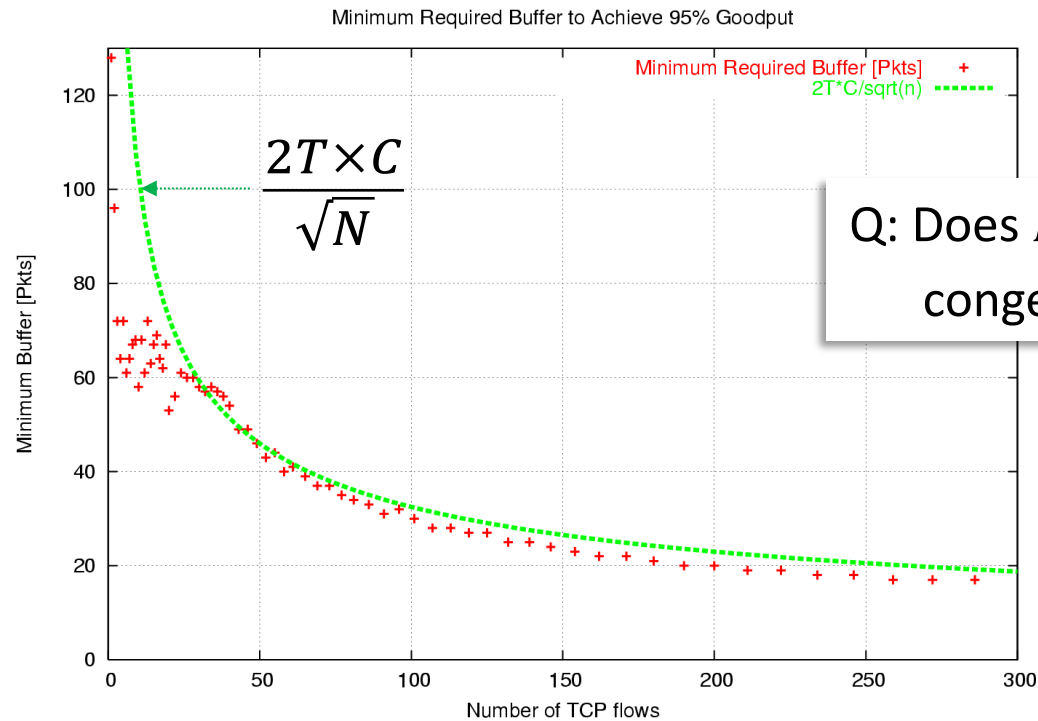
$$B \geq \frac{2T \times C}{\sqrt{N}}$$

Example: $2T = 100ms, C = 10Gb/s, N = 1$

$B \geq 1Gbit$

Example: $2T = 100ms, C = 10Gb/s, N = 10,000$

$B \geq 10Mbit$



Q: Does $B \geq \frac{2T \times C}{\sqrt{N}}$ hold for all popular congestion control schemes...?

1988

1994

2004

2006

2020

Congestion Avoidance and Control VJ & MK

High Performance TCP in ANSNET CV & CS

Sizing Router Buffers GA, IK, NM

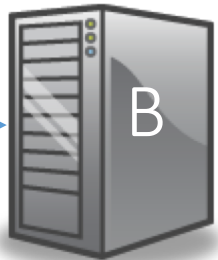
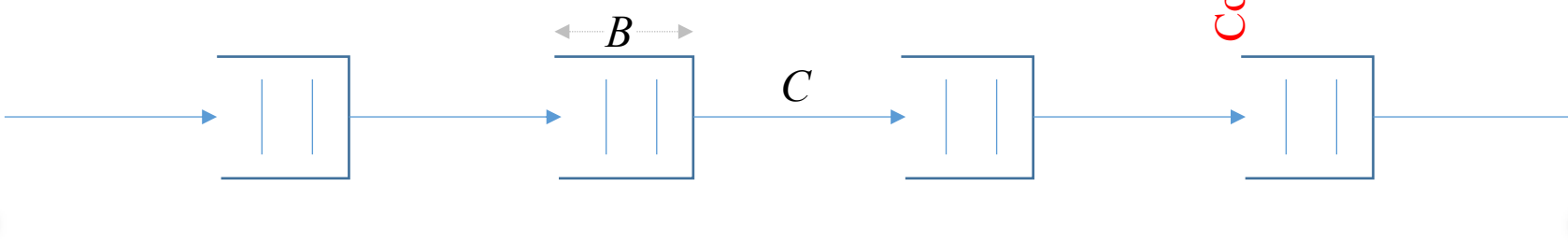
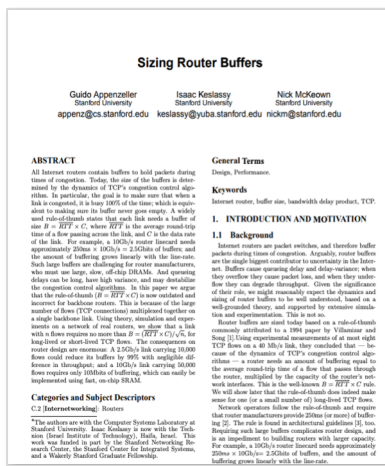
Routers with Very Small Buffers ME, YG, AG, NM, TR

B = O(log W)

- 1. Paced Traffic
2. Link utilization < 80%

20-50 packet buffers, all-optical routers.

Assumptions Consequences



Min RTT = 2T

1988

1994

2004

2006

2008

2020



Congestion Avoidance and Control

VJ & MK

High Performance TCP in ANSNET

CV & CS

Sizing Router Buffers

GA, IK, NM

Routers with Very Small Buffers

ME, YG, AG, NM, TR

Experimental Study of Router Buffer Sizing

NB, YG, MG, NM, GS

Congestion Avoidance and Control*

Van Jacobson[†]
Lawrence Berkeley Laboratory
Michael J. Karels[‡]
University of California at Berkeley
November, 1988

Introduction

Computer networks have experienced an explosive growth over the past few years and with that growth have come severe congestion problems. For example, it is now common to see internet gateways drop 10% of the incoming packets because of local buffer overflows. Our investigation of some of these problems has shown that much of the cause lies in transport protocol implementations (not in the protocols themselves). The 'obvious' ways to implement a window-based transport protocol can result in exactly the wrong behavior in response to network congestion. We give examples of 'wrong' behavior and describe some simple algorithms that can be used to make right things happen. The algorithms are rooted in the idea of achieving network stability by forcing the transport connection to obey a 'packet conservation' principle. We show how the algorithms derive from this principle and what effect they have on traffic over congested networks.

In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 miles and two BGP hops) dropped from 22 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad. In particular, we wondered if the 4.3BSD (Berkeley UNIX) TCP was mis-behaving in a way that could be tuned to work better under abnormal network conditions. The answer to both of these questions was "yes".

*This is a very slightly revised version of a paper originally presented at SIGCOMM '88 [12]. If you wish to reference this work, please cite [12].

[†]This work was supported in part by the U.S. Department of Energy under Contract Number DE-AC02-78SF0080.

[‡]This work was supported by the U.S. Department of Commerce, National Bureau of Standards, under Grant Number 60NANB09030.

High Performance TCP in ANSNET

Chris Villamor court@lanl.net
Advanced Network & Services, Inc.
Chang Song cchang@research.ibm.com
Advanis
September 12, 1994

Abstract

This report concentrates on specific experiments and gains of the research activities reported by ANSNET. It applies to any TCP dominated high speed WAN and in particular those striving to support high speed real-time flow. Measurements have been made under real-time conditions to better understand performance barriers imposed by queuing capacities and queue drop strategies.

The IBM RS/6000 based system currently supports ANSNET performed very well in these tests. Measurements have been made with the current software and performance enhanced software. Single TCP flows are able to achieve 40 Mb/s and complete multiple TCP flows achieve over 4 Mb/s link utilization on a 147 Mb/s DS3 link with delay comparable to 80 msec standard ANSNET flows. Congestion collapse is demonstrated with multi-destination routing capacity and using window size much larger than optimal.

A variation of Floyd and Jacobson's Random Early Detection (RED) algorithm [5] is tested. Performance improved with the use of RED for links arriving multiple flows. With RED and queuing capacity as slow as delay bandwidth product, congestion collapse is avoided, allowing the maximum window size to be set at arbitrarily high.

Queuing capacity greater than or equal to the delay bandwidth product and RED are recommended. RED provides performance improvement in at least the single flow case. But cannot substitute for adequate queuing capacity. Performance of high speed flows can be improved.

Contents

- 1 Introduction
- 2 TCP Segment Size
- 2.2 TCP Maximum Window Size
- 2.3 TCP Congestion Avoidance
- 2.4 Fast Retransmit and Recovery
- 2.5 Performance Details
- 3 Queue Size Requirements
- 3.1 Multiple TCP Flows
- 3.2 Effects of Queuing Capacity
- 4 Performance Tuning
- 4.1 Test Network Conditions
- 4.2 Router Queuing Capacity
- 4.3 Traffic Sources
- 4.4 Summary of Test Conditions
- 5 Test Results
- 5.1 Single High-Speed Flows
- 5.2 Multiple Flows
- 5.3 Random Early Detection
- 5.4 Random Early Detection [5] in Detail
- 5.5 Fluctuation and Delay
- 5.6 Link Utilization Estimates
- 6 Recommendations
- 7 Other Considerations
- 8 Conclusions
- 9 Acknowledgments

Sizing Router Buffers

Guido Appenzeller
Stanford University
appenz@stanford.edu

Isaac Keslassy
Stanford University
keslassy@yuba.stanford.edu

Nick McKeown
Stanford University
nickm@stanford.edu

ABSTRACT

All Internet routers contain buffers to hold packets during times of congestion. The buffers used to be large and are slowly being shrunk to small enough to use fast memory technologies such as SRAM or all-optical buffering. Unfortunately, a widely used rule-of-thumb says we need a bandwidth-delay product of buffers at each router, an rule we now question. This rule can be challenged. In a recent paper, Appenzeller et al. without making this claim, we show that the number of flows sharing the bandwidth can be significantly reduced even in a link with a large delay product. In our setting, if the TCP sources are not evenly spaced, then there can be many packet buffers are sufficient for high throughput. Specifically, we argue that $C \times RTT \times C/D$ is sufficient, where C is the number of flows, RTT is the round-trip time, and D is the delay of the link. We support our claim with analysis and a variety of simulations.

General Terms

Design, Performance.

Keywords

Internet router, buffer size, bandwidth-delay product, TCP.

1. INTRODUCTION AND MOTIVATION

1.1 Background

Internet routers are packet switches, and therefore buffer packets during times of congestion. Arguably, router buffers are the single largest contributor to uncertainty in the Internet. Buffers cause queuing delay and delay variance when they overflow their queue packets, and when they underflow they can degrade throughput. Given the significance of this role, we might reasonably expect the dimensioning and operation of buffers to be well understood, based on a well-grounded theory and supported by extensive simulation and experimentation. This is not the case. In fact, the most commonly attributed to a 1994 paper by Villamor and Song [2] using experimental measurements of at least eight TCP flows on a 40 Mb/s link, they concluded that "because of the dynamics of TCP's congestion control algorithm, a router needs an amount of buffering equal to the average round-trip time of a flow that passes through the router, multiplied by the capacity of the router's output network interface. This is the well known $C \times RTT \times C$ rule. We will later show that the rule-of-thumb does indeed make sense for one (or a small number) of long-lived TCP flows. Network operators follow the rule-of-thumb and require that router manufacturers provide 200ms for every 100Mbps link. The rule is based in architectural guidelines [3], too. Repeating such large buffer congestion router devices, and as implemented in building routers with larger capacity. For example, a 100Gbps router required needs approximately $200ms \times 100Gbps = 20GB$ of buffers, and the amount of buffering grows linearly with the line rate.

Routers with Very Small Buffers

Mihaila Staehle[†], Tadeu Chaves[†], Adnan Chari[†], Nick McKeown[†], and Tim Roughgarden[‡]
Stanford University
[†]Department of Electrical Engineering, Stanford University
[‡]Department of Management Science and Engineering, Stanford University
staehle@stanford.edu

ABSTRACT

Internet routers require buffers to hold packets during times of congestion. The buffers used to be large and are slowly being shrunk to small enough to use fast memory technologies such as SRAM or all-optical buffering. Unfortunately, a widely used rule-of-thumb says we need a bandwidth-delay product of buffers at each router, an rule we now question. This rule can be challenged. In a recent paper, Appenzeller et al. without making this claim, we show that the number of flows sharing the bandwidth can be significantly reduced even in a link with a large delay product. In our setting, if the TCP sources are not evenly spaced, then there can be many packet buffers are sufficient for high throughput. Specifically, we argue that $C \times RTT \times C/D$ is sufficient, where C is the number of flows, RTT is the round-trip time, and D is the delay of the link. We support our claim with analysis and a variety of simulations.

1. MOTIVATION AND INTRODUCTION

Internet routers are packet switches, and therefore buffer packets during times of congestion. Arguably, router buffers are the single largest contributor to uncertainty in the Internet. Buffers cause queuing delay and delay variance when they overflow their queue packets, and when they underflow they can degrade throughput. Given the significance of this role, we might reasonably expect the dimensioning and operation of buffers to be well understood, based on a well-grounded theory and supported by extensive simulation and experimentation. This is not the case. In fact, the most commonly attributed to a 1994 paper by Villamor and Song [2] using experimental measurements of at least eight TCP flows on a 40 Mb/s link, they concluded that "because of the dynamics of TCP's congestion control algorithm, a router needs an amount of buffering equal to the average round-trip time of a flow that passes through the router, multiplied by the capacity of the router's output network interface. This is the well known $C \times RTT \times C$ rule. We will later show that the rule-of-thumb does indeed make sense for one (or a small number) of long-lived TCP flows. Network operators follow the rule-of-thumb and require that router manufacturers provide 200ms for every 100Mbps link. The rule is based in architectural guidelines [3], too. Repeating such large buffer congestion router devices, and as implemented in building routers with larger capacity. For example, a 100Gbps router required needs approximately $200ms \times 100Gbps = 20GB$ of buffers, and the amount of buffering grows linearly with the line rate.

Experimental Study of Router Buffer Sizing

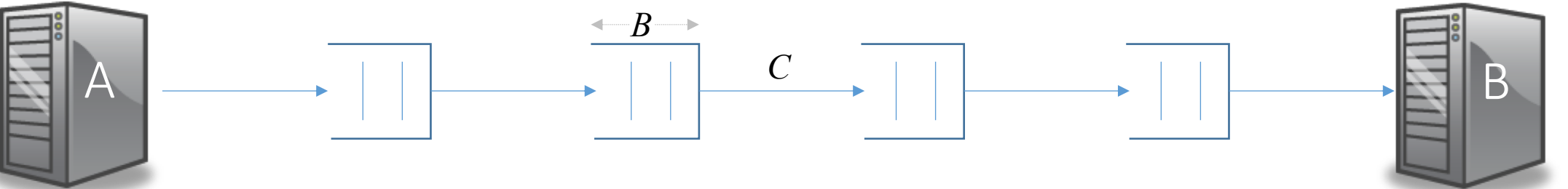
Neda Behrang[†], Tadeu Chaves[†], Adnan Chari[†], Nick McKeown[†], and Tim Roughgarden[‡]
Stanford University
[†]Department of Electrical Engineering, Stanford University
[‡]Department of Management Science and Engineering, Stanford University
nbeh@stanford.edu

ABSTRACT

During the past few years, several papers have proposed rules for sizing buffers in Internet core routers. Specifically, it is argued that a link needs a buffer of size $C \times RTT \times C$, where C is the capacity of the link and RTT is the round-trip time. In this paper, we experimentally study the impact of this rule on the performance of a network. We show that the rule is not always valid, and that the number of flows that can be supported by a link can be significantly larger than the rule suggests. We also show that the rule is not always valid, and that the number of flows that can be supported by a link can be significantly larger than the rule suggests.

1. MOTIVATION AND INTRODUCTION

Internet routers are packet switches, and therefore buffer packets during times of congestion. Arguably, router buffers are the single largest contributor to uncertainty in the Internet. Buffers cause queuing delay and delay variance when they overflow their queue packets, and when they underflow they can degrade throughput. Given the significance of this role, we might reasonably expect the dimensioning and operation of buffers to be well understood, based on a well-grounded theory and supported by extensive simulation and experimentation. This is not the case. In fact, the most commonly attributed to a 1994 paper by Villamor and Song [2] using experimental measurements of at least eight TCP flows on a 40 Mb/s link, they concluded that "because of the dynamics of TCP's congestion control algorithm, a router needs an amount of buffering equal to the average round-trip time of a flow that passes through the router, multiplied by the capacity of the router's output network interface. This is the well known $C \times RTT \times C$ rule. We will later show that the rule-of-thumb does indeed make sense for one (or a small number) of long-lived TCP flows. Network operators follow the rule-of-thumb and require that router manufacturers provide 200ms for every 100Mbps link. The rule is based in architectural guidelines [3], too. Repeating such large buffer congestion router devices, and as implemented in building routers with larger capacity. For example, a 100Gbps router required needs approximately $200ms \times 100Gbps = 20GB$ of buffers, and the amount of buffering grows linearly with the line rate.



$$Min RTT = 2T$$

Buffer Sizing Experiments Are Challenging

Testbed experiments:

- Generate realistic traffic with high accuracy
- Explore a very large space (load, traffic shape, ...)

Real network experiments:

- Packet drops *may* violate SLAs
- Adjusting buffers not straight forward (device limitations)

Both:

- Accurate measurement of performance metrics not straight forward

Buffer Sizing Experiments

Small Buffers

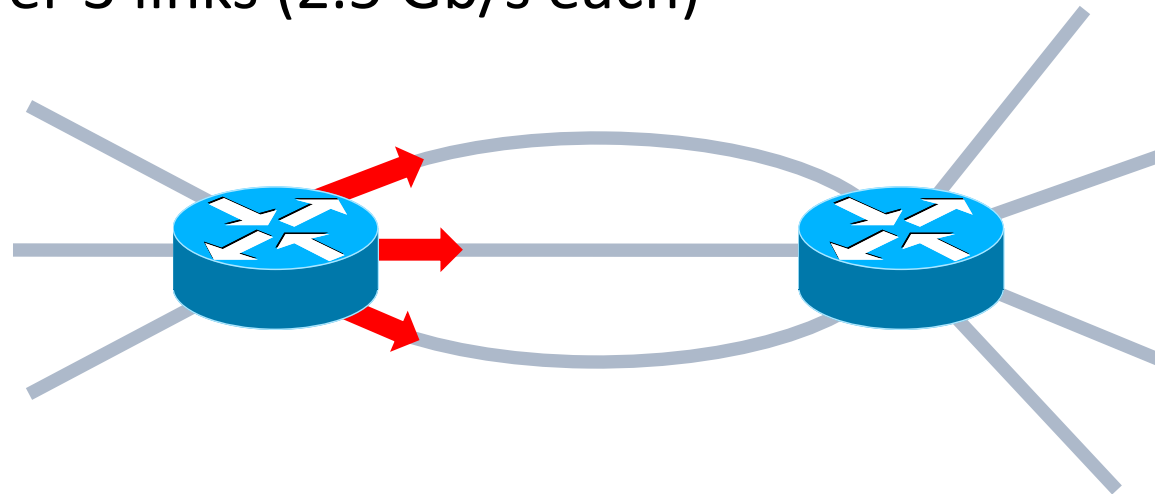
- Stanford University dorm network
- University of Wisconsin
- Internet2
- **Level 3 Communications**

Tiny Buffers

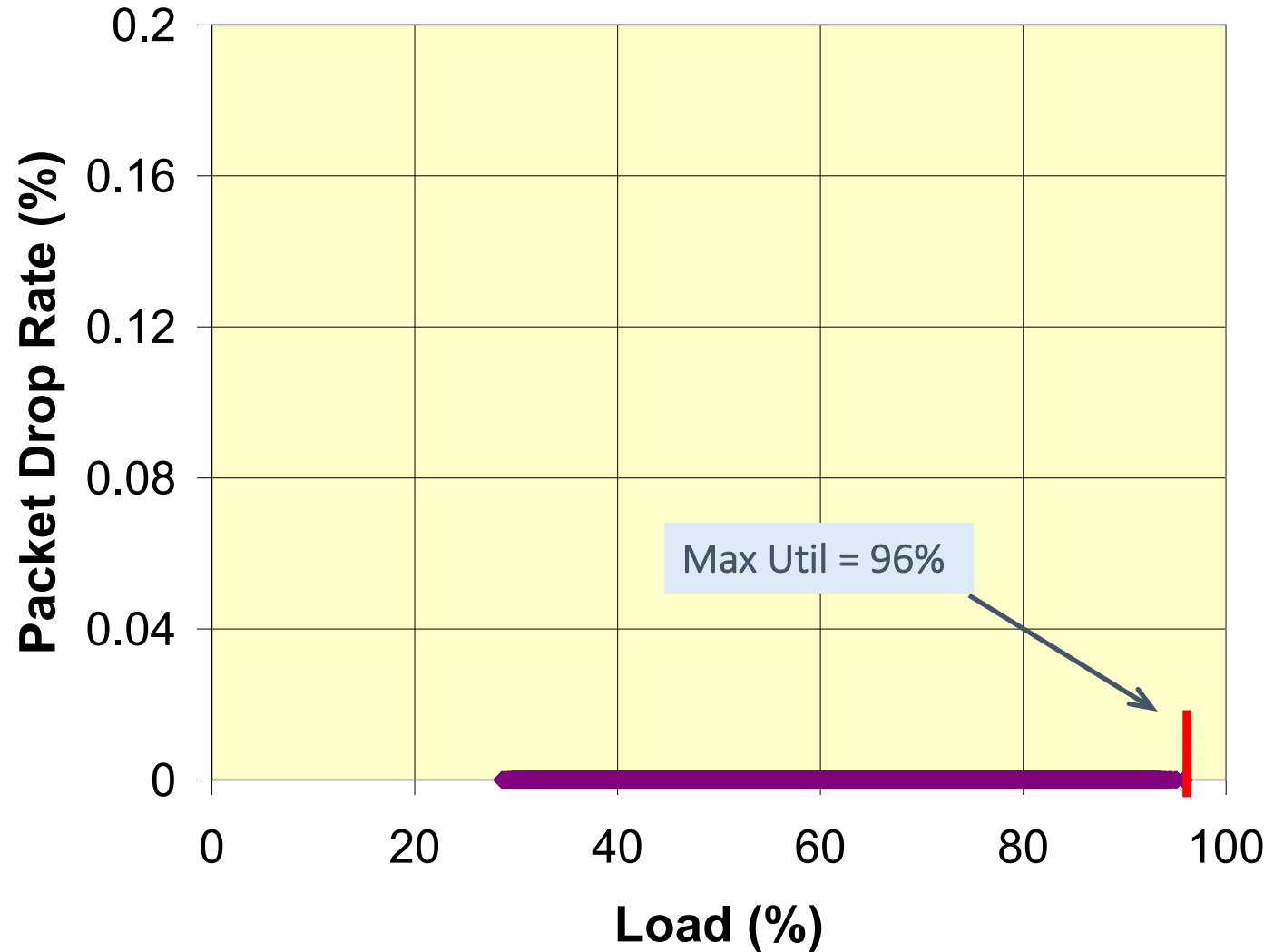
- Internet2
- Sprint Advanced Technology Lab
- University of Toronto

Level 3 Communications Experiments

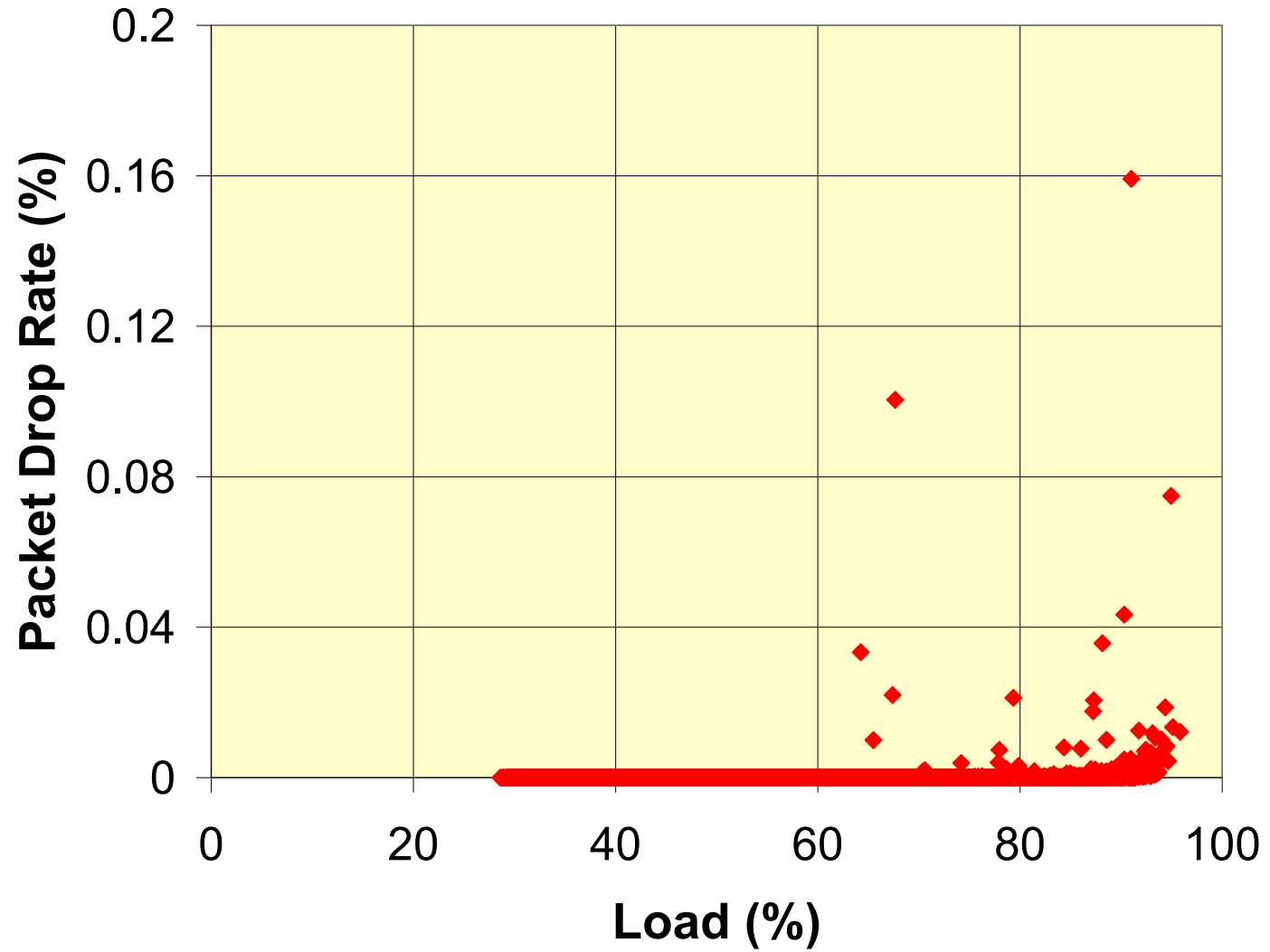
- High link utilization
- Long duration (about two weeks)
- Buffer sizes 190ms (250K packets), 10ms (10K packets), 2.5ms (2500 packets), 1ms (1000 packets)
- Load balancing over 3 links (2.5 Gb/s each)



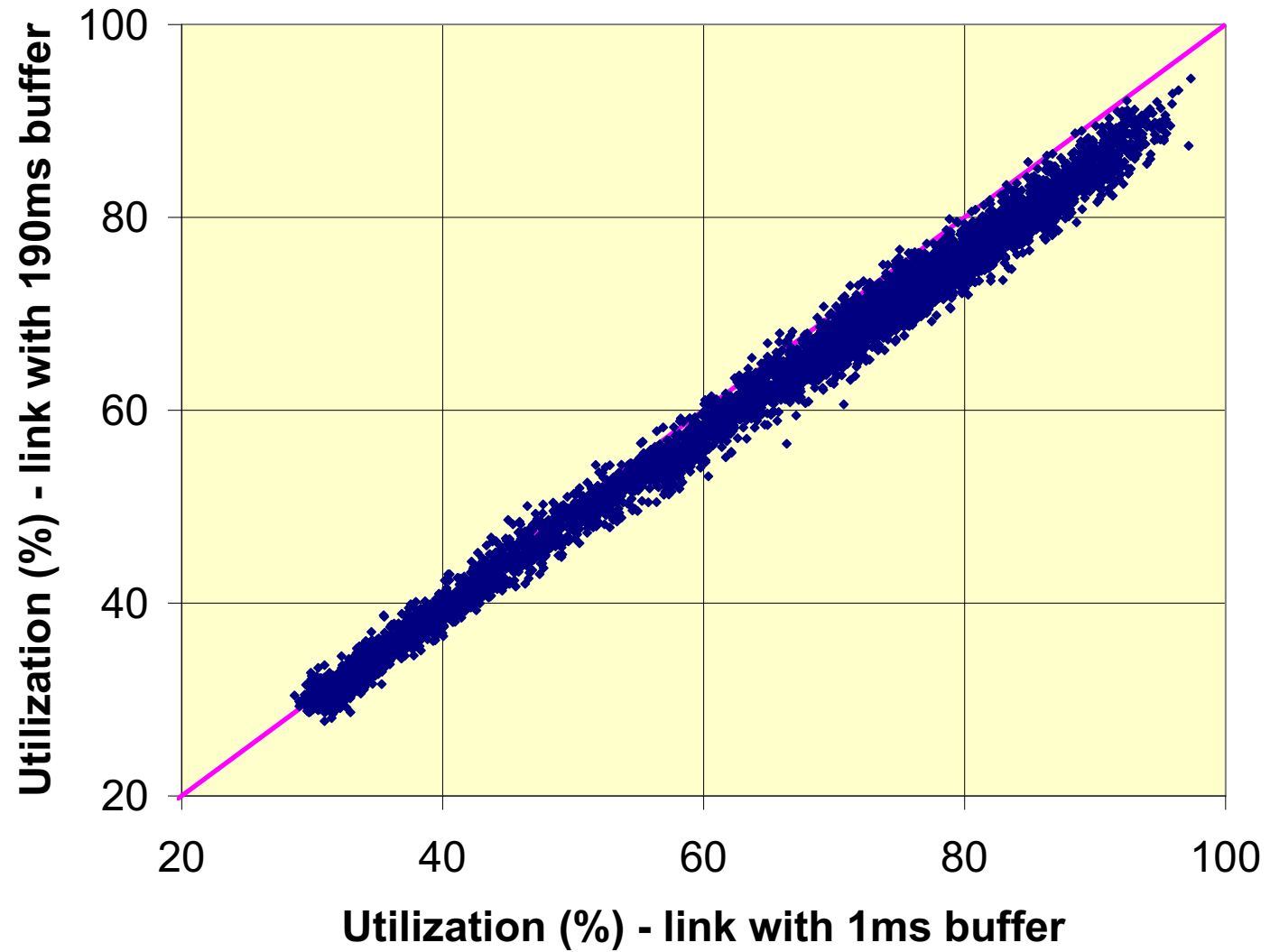
Drop vs. Load, Buffer = 190ms, 10ms



Drop vs. Load, Buffer = 1ms



Relative Link Utilization



Buffer Sizing Experiments

Small Buffers

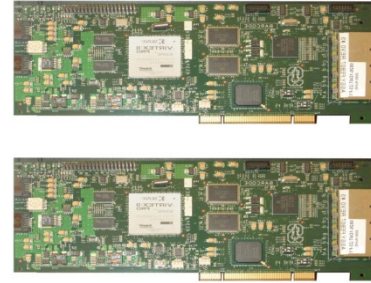
- Stanford University dorm network
- University of Wisconsin
- Internet2
- Level 3 Communications

Tiny Buffers

- Internet2
- Sprint Advanced Technology Lab
- **University of Toronto**

Tiny Buffers Experiments

- Network of NetFPGA-based switches (20-100 machines)
 - 4 GigE interfaces
 - Programmable
- Accurate packet injections
- Complete TCP stack
- Accurate buffer size control
- No hidden buffers
- Added feature to measure queue occupancy time series



Experiment Results

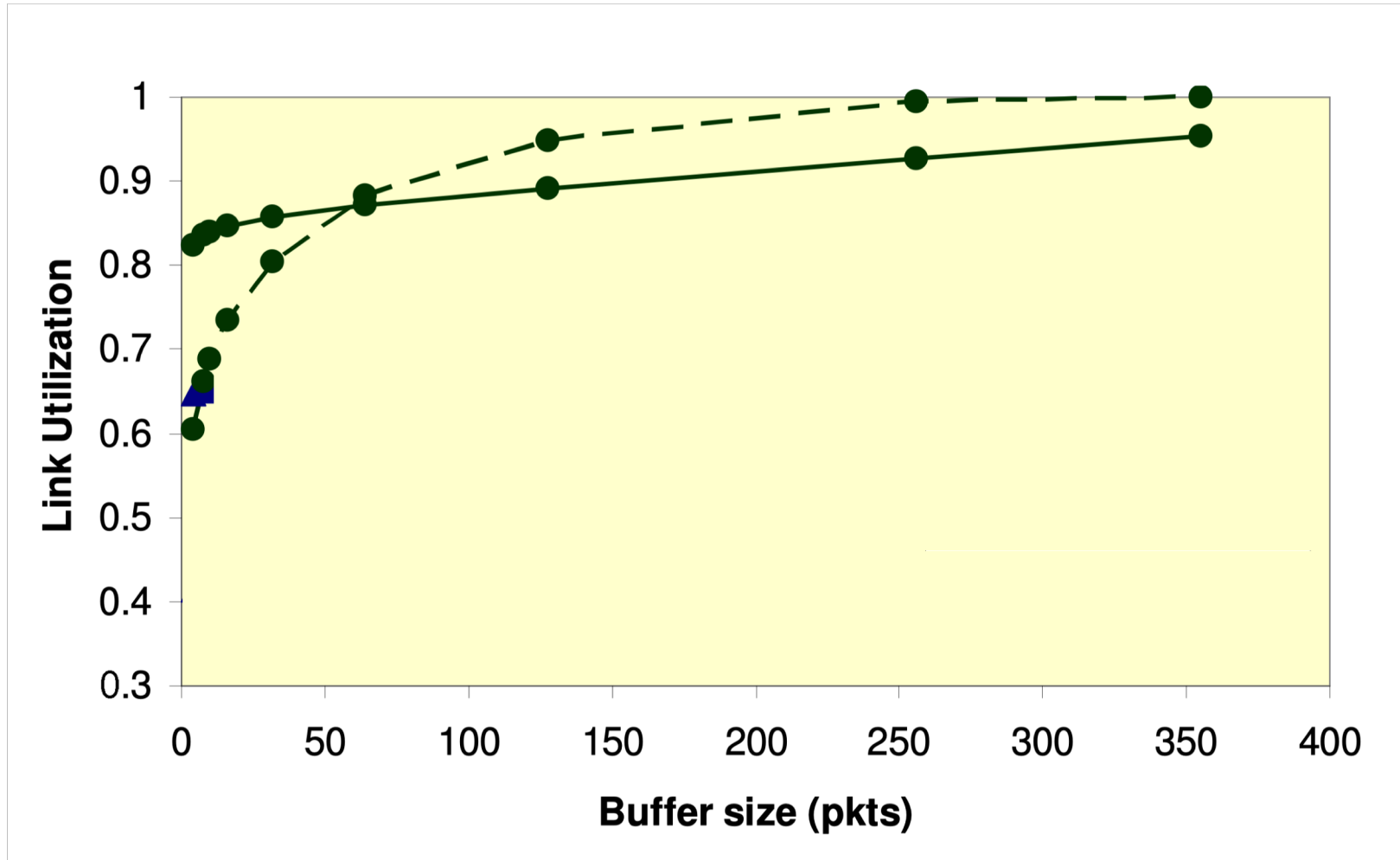
We measured:

- Throughput
- Flow completion times
- Packet drop rates
- ...

For various combinations of:

- Input traffic
- Delays
- Buffer sizes
- ...

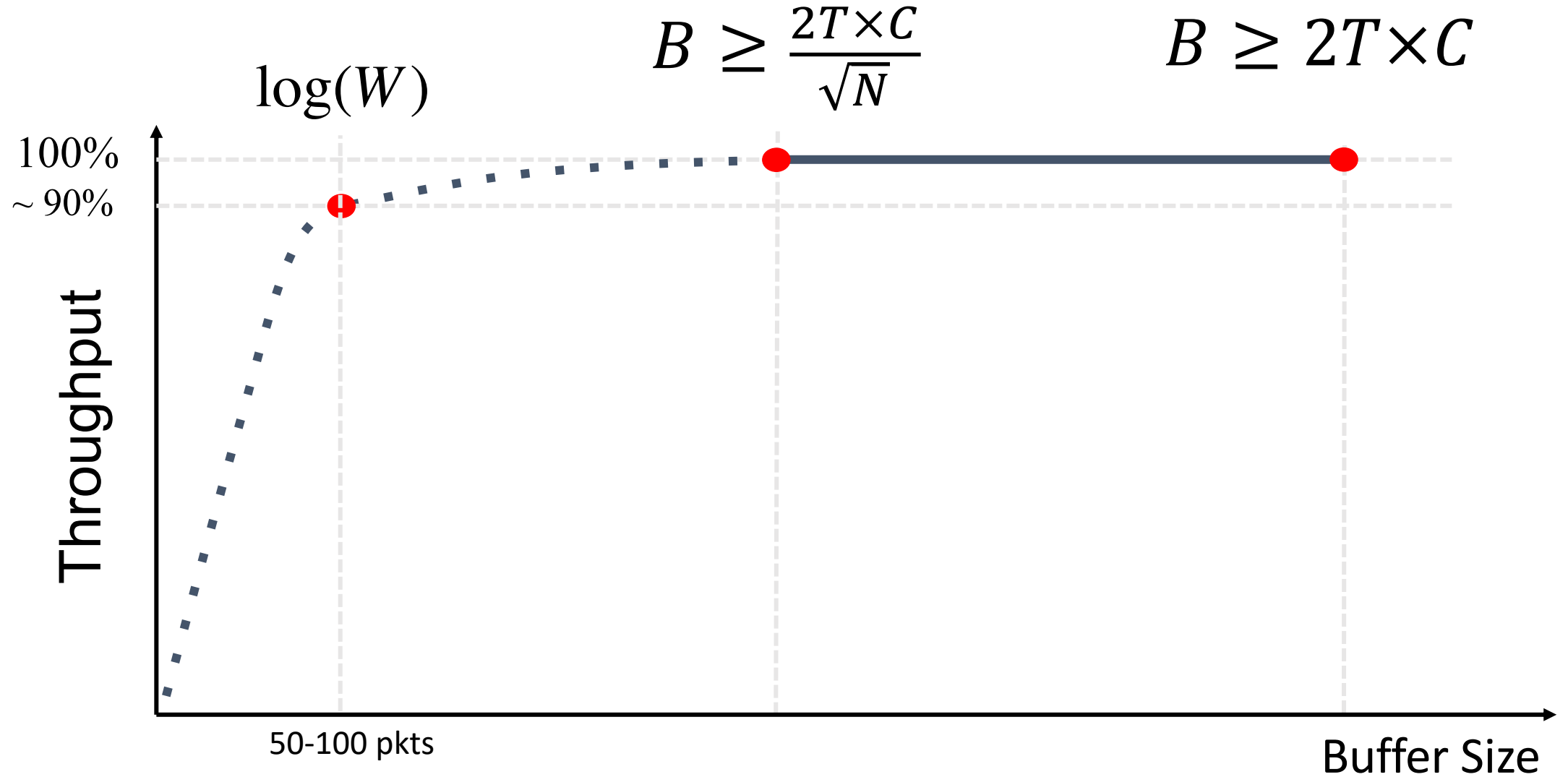
Results: Pacing and Buffer Size



Experiment Conclusions

- Small and tiny buffer **experiments inline with theoretical predictions**
- **Small buffers:** no change needed
- **Tiny buffers:** assumptions are extremely important
 - Necessary to guarantee them all over the network
 - We need support from network components (both software and hardware)

Summary



Some ground rules for the day

- 40 different experiences, 40 preconceived notions. Me too.
- Let's check preconceptions at the door: None of us know for sure.
- **Speakers:** Please keep you to 15 minutes, including Q&A
- This afternoon, two discussion sessions:
 1. **Conclusions:** What do we take away from today?
 2. **Actions:** What are the next steps?

Schedule for the day

10.30am – 1.30pm

Session 1: Network Operators

- Neda Beheshti Facebook
- Lincoln Dale Google
- TY Huang Netflix
- Hongqiang Liu Alibaba
- Ken Duell AT&T
- Joel Jaeggli Fastly

[12.00 – 12.30 Lunch]

- Simon Leinen Switch
- Bob Briscoe CableLabs
- Chuanxiong Guo Bytedance
- Igor Gashinsky Oath

1.45pm – 2.45pm

Session 2: Technology Providers

- Parvin Taheri Cisco
- Francois Labonte Arista
- Golan Schzukin Dune/BCM
- Chang Kim Barefoot

3.00pm – 4.00pm

Session 3: Discussion

- Conclusions Neda, Bruce, Nasser
- Actions and Next Steps Yashar, Nick