

Sizing router buffers

CS 244: Spring '22

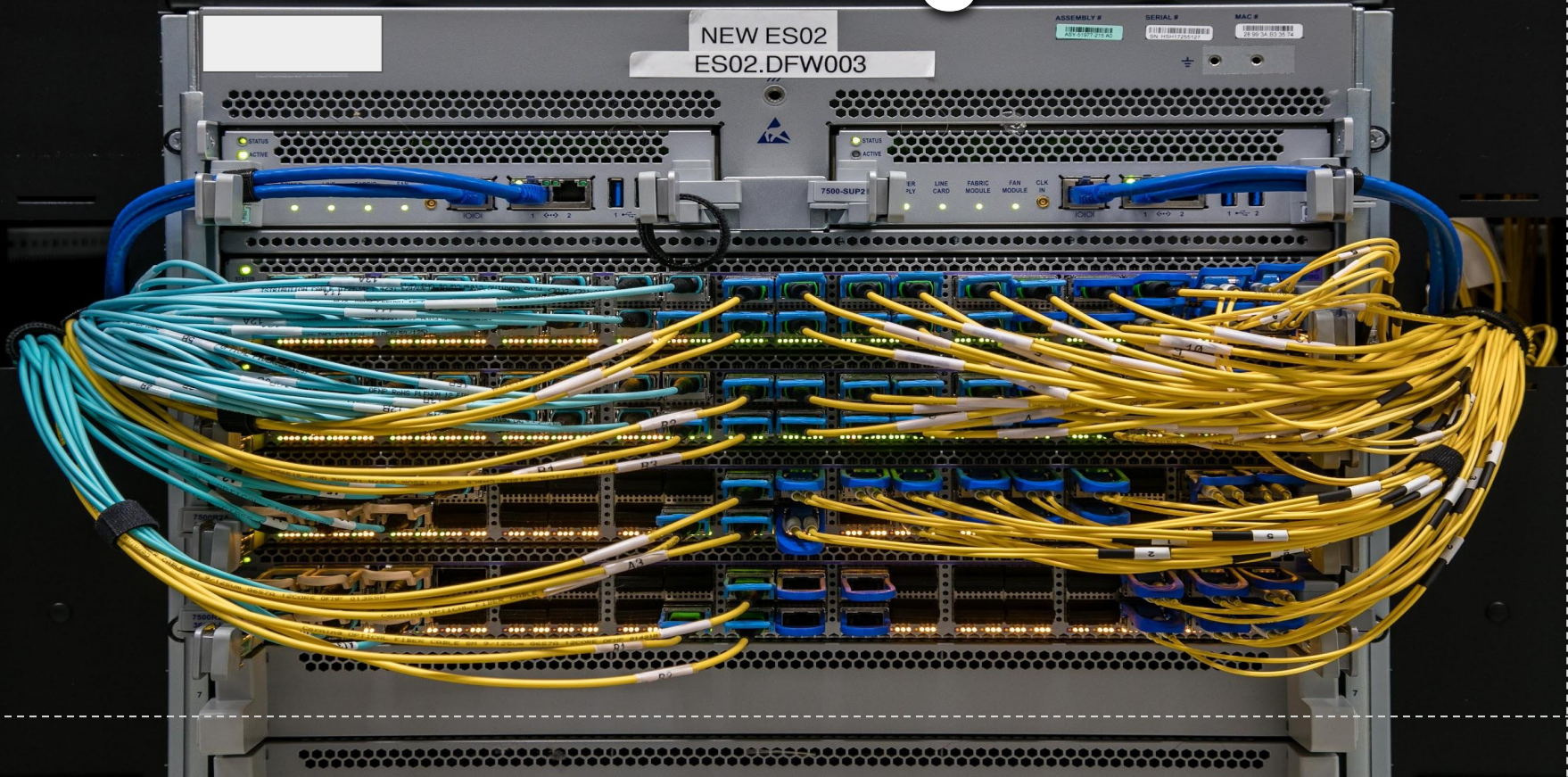
Outline

- What is this buffer sizing question?
- Prove the BDP result
- Discuss the BDP/\sqrt{n} result

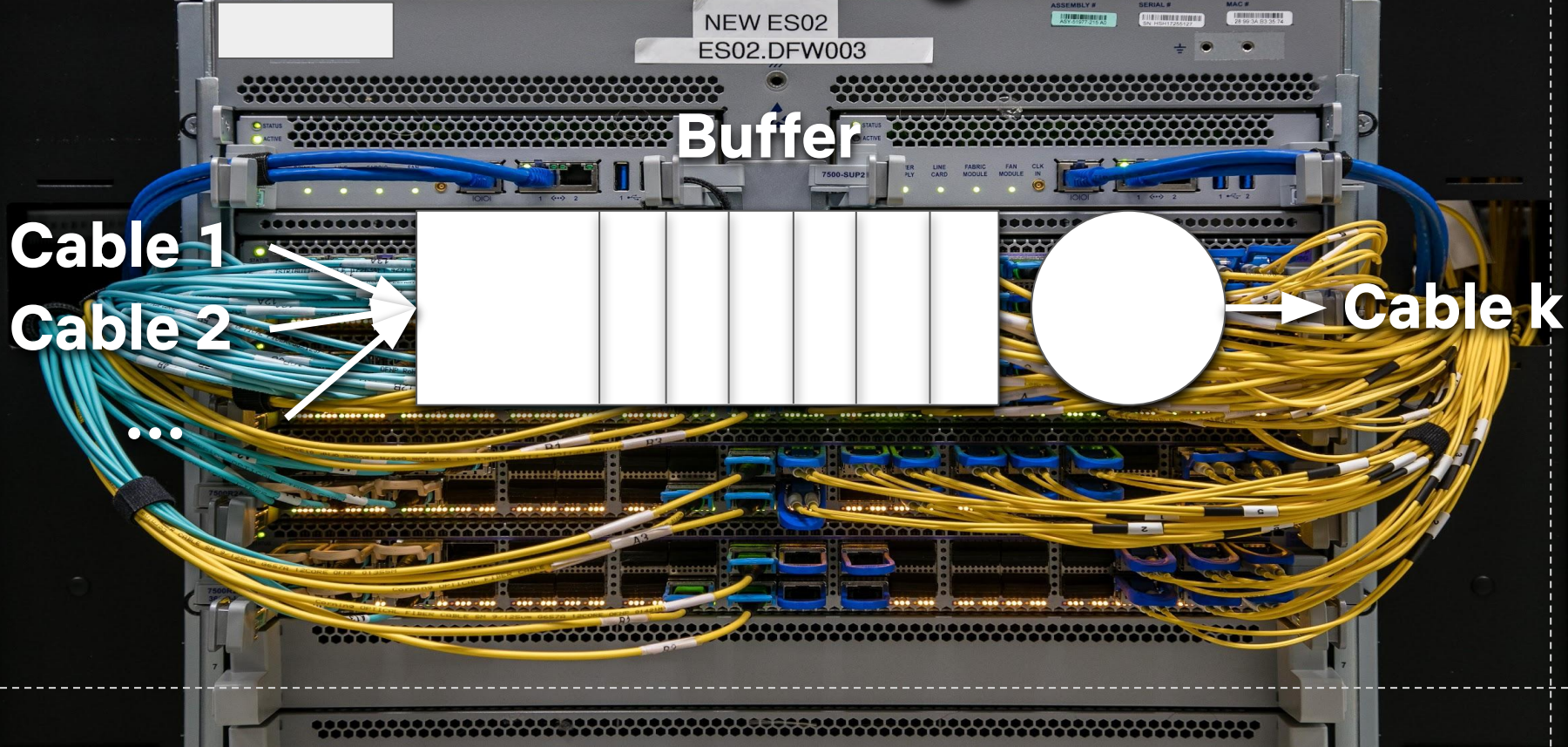
What I hope you will take away

1. An idea about the question/results
2. A sense of all the different parties involved here, and why it's a tricky problem to “solve”
3. A feeling that buffer size is determined by application/congestion control decisions

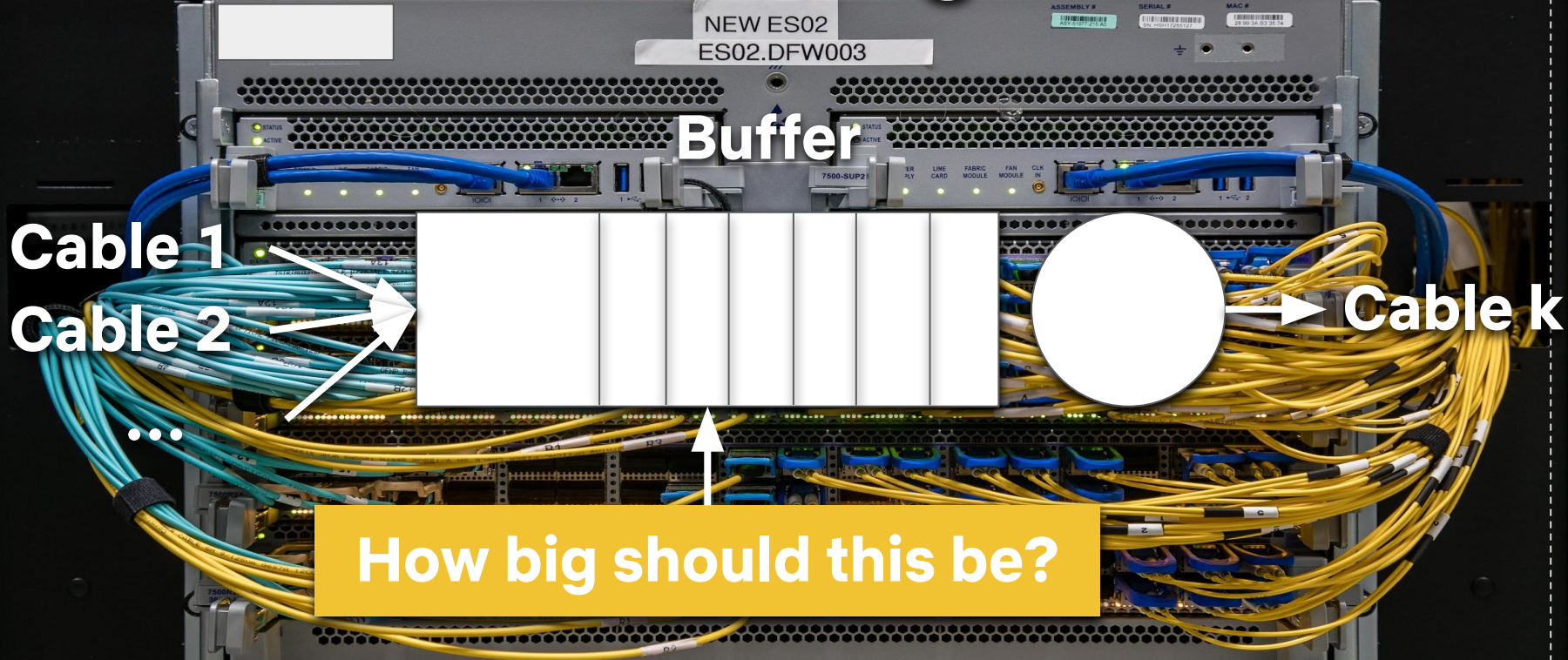
What are we talking about?



What are we talking about?



What are we talking about?



Application side

Think about the extremes:

One packet: if a packet arrives slightly too fast, it is dropped

Very large: packets will wait a long time to be sent

Network operator side

- Need to have good performance for network users
- Need to buy routers
 - Smaller commodity routers
 - Cheaper, programmable, etc...
 - Cost: Thousands to tens of thousands of dollars
 - Larger routers specifically for ISPs
 - Large buffers, routing tables
 - Cost: hundreds of thousands of dollars

Router manufacturer side

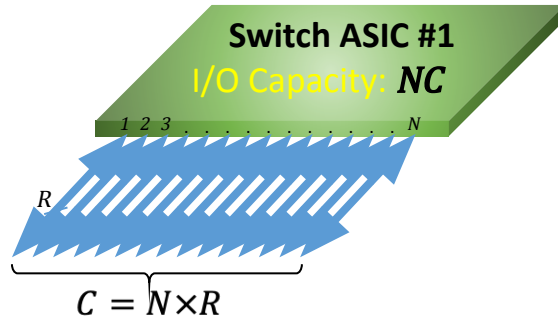
- Need to pick a size
- Can't increase later, but pretty easy to decrease later
- Balance selling routers w/ cost
 - On-chip vs off-chip
 - SRAM – Routers used 80% of global SRAM ca. 2004
 - Large markup for ISP routers – 90% ca. 2004

Off chip vs. on chip buffers

With on-chip buffers we can build higher capacity ASICs

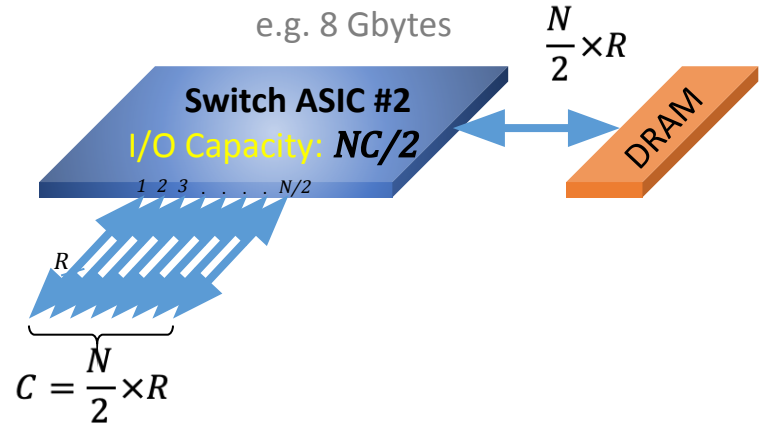
Switch Chips are Limited by Serial I/O Capacity

Single chip switch ASIC
Small on-chip buffering
e.g. 64 Mbytes

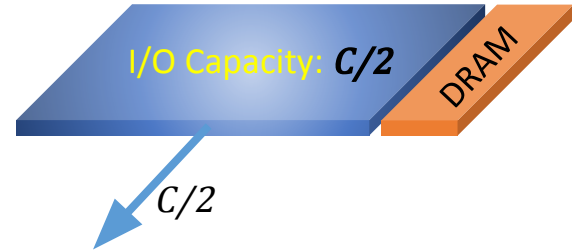
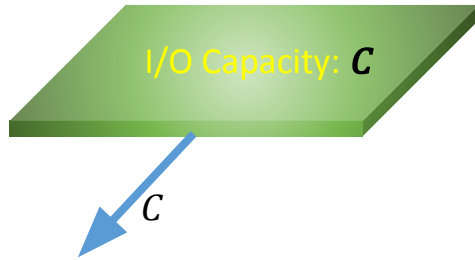


e.g. 12.8 Tb/s = 128 x
100Gb/s

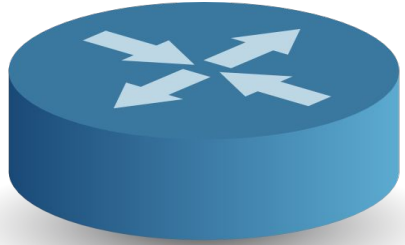
Switch ASIC with external memory
Large off-chip buffering
e.g. 8 Gbytes



Switch Chips are Limited by Serial I/O Capacity

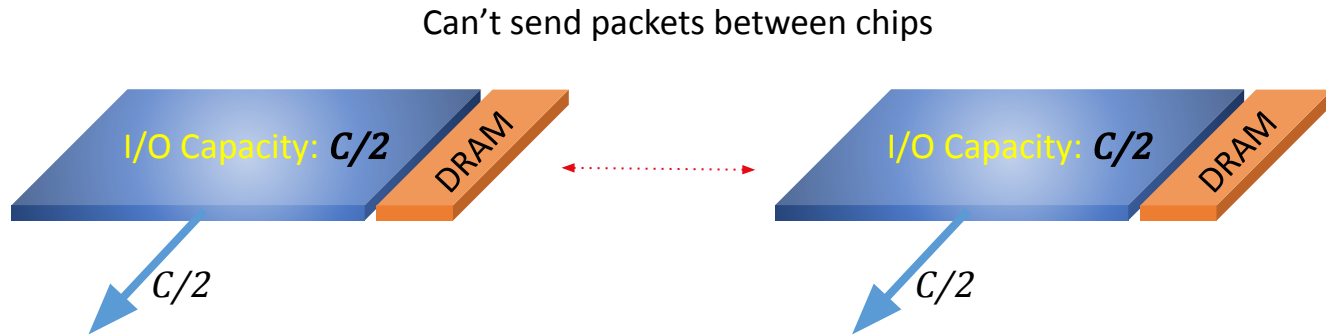


C

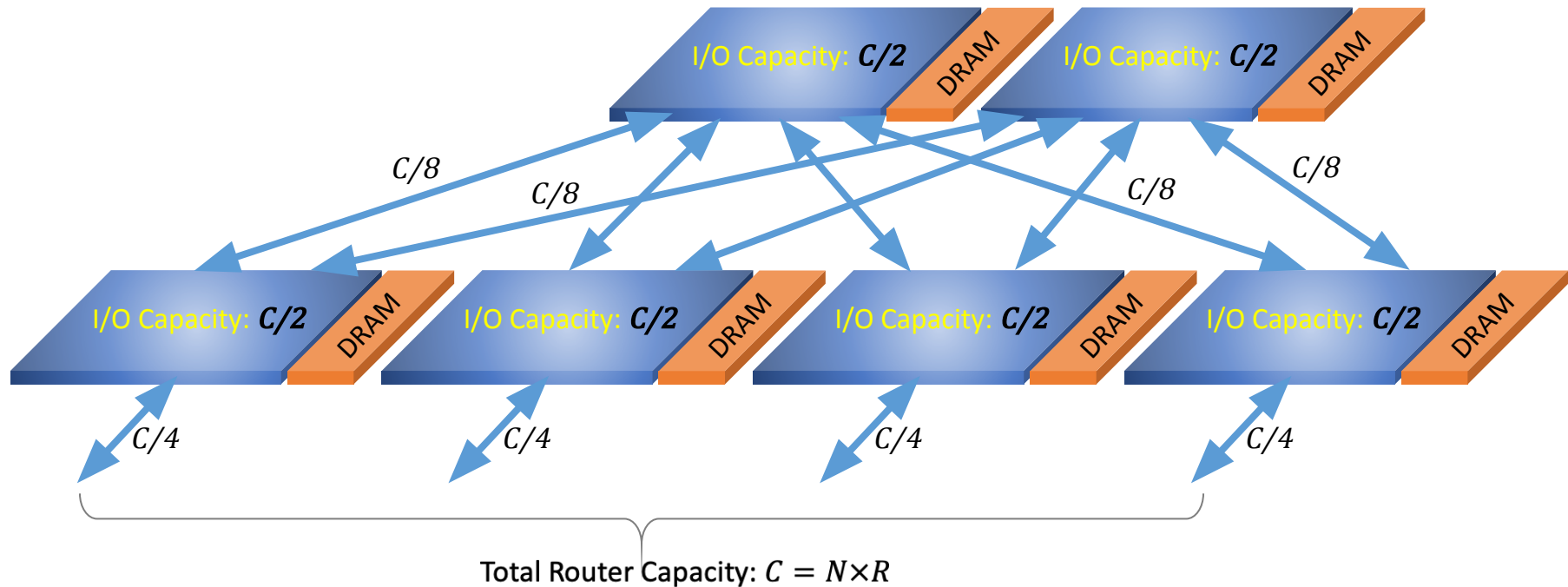


How many switch chips with capacity $C/2$ do we need to make a router with capacity C ?

It's not two!



We need **6** ASICs with capacity $C/2$



Takeaway

- Buffer size matters for performance
- Buffer size matters to network operators
- Buffer size matters when building a router

So how big should a buffer be?

You asked...

- “Can you speak to [...] whether learned methods are replacing brittle rules of thumb and dynamically adapting to each scenario?”
- “Given that the proper buffer size is dependent on factors like the number of flows, are there cases where we might want a more dynamic adjustment of buffer size?”
- “Also could in-network telemetry be used to dynamically size buffers?”

How to answer this question?

1. Come up with theoretical guidelines
 - Be useful for hardware manufacturers/network operators
 - *Not* experimental or algorithmic
2. Focus on the performance part of the trade-off
 - People want larger buffers because they think they perform well
 - How small can we get buffers w/o sacrificing performance?

Worst-case scenario #1

Can you come up with an example that requires large buffers?

Algorithm:

If a *single* packet is lost, we are very unhappy.

Implies that a buffer needs to be large enough to absorb any burst.

Aside: this is how [AT&T sizes buffers](#)

Worst-case scenario #2

Can you come up with some example that requires small buffers?

Algorithm:

If delay is *ever* larger than X ms, we are very unhappy.

Restricts buffer to $< X * C$ packets

Aside, [data from 2006](#): adding 500ms reduced Google traffic by 20%

Strategy in this line of work

Come up with a more definite result by:

1. Looking at one specific protocol: TCP
2. And look at one specific metric: link utilization
3. During one specific type of setting: congestion

Q: How big does a buffer need to be so that TCP Reno will fully utilize a link?

1988

1994

2004

2006

2008

2021

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 Buffers NB, YG, MG, NM, GS

Updating the Theory of Buffer Sizing BS, SA, 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, in a two continent test environment gateway drop 10% on 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 due to the protocols themselves. The "obvious" way 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 things happen. The algorithms are tested in the lab and achieve network stability by forcing the transport connections to obey a "packet conservation" principle. We show how the algorithm derives 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 U.S. to UC Berkeley (links separated by 4000 miles and two NSF hops) dropped from 15 Kbps to 1 Kbps. We were fascinated by this sudden factor-of-ten drop in bandwidth and embarked on an investigation of why things had gotten so bad. In particular, we considered the *Choke* (Berkley) (CDS) test was bandwidth of 100 packets per second was "OK" for a 100 Kbps link. This was a factor of 100 times the bandwidth per second of a 100 Kbps link. This was a factor of 100 times the bandwidth per second of a 100 Kbps link. This was a factor of 100 times the bandwidth per second of a 100 Kbps link.

*This work was supported in part by the U.S. Department of Energy under Contract Number DE-AC33-78-OR01464.

©1988 by the authors. All rights reserved. This work is distributed under the Creative Commons Attribution License.

High Performance TCP in ANSNET

Ching-Yi Yeh¹, Ching-Shyan Chen²
Advanced Network & Services, Inc.
Cheng-Sheng Chang³, David D. Stanculescu⁴
Advanet
September 12, 1994

Abstract

This paper introduces an explicit congestion control and packet scheduling algorithm for ANSNET. The algorithm is designed to improve the performance of TCP in ANSNET. It is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments. The algorithm is designed to be implemented in a wide range of network environments. The algorithm is designed to be implemented in a wide range of network environments.

1. Introduction

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper describes a new algorithm for congestion control and packet scheduling that is designed to improve the performance of TCP in ANSNET. The algorithm is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

2. Background

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper describes a new algorithm for congestion control and packet scheduling that is designed to improve the performance of TCP in ANSNET. The algorithm is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

3. The Algorithm

The algorithm is based on the principle of packet conservation. It is designed to be implemented in a wide range of network environments. The algorithm is designed to be implemented in a wide range of network environments.

4. Performance Testing

The algorithm was tested in a wide range of network environments. The results show that the algorithm is designed to be implemented in a wide range of network environments.

5. Conclusions

The algorithm is designed to be implemented in a wide range of network environments. The results show that the algorithm is designed to be implemented in a wide range of network environments.

6. Acknowledgments

The authors would like to thank the following people for their help and support:

7. References

[1] Van Jacobson, "Congestion Control and Packet Scheduling," *Proceedings of the ACM SIGCOMM Conference*, 1988.

8. Contact Information

Ching-Yi Yeh: cyeh@ansnet.com
Ching-Shyan Chen: cshchen@ansnet.com
Cheng-Sheng Chang: cchang@ansnet.com
David D. Stanculescu: ddstancu@ansnet.com

Sizing Router Buffers

Guoqing Xu¹, Hui Kang², Niek Meunander³
Shanghai University
University of Groningen
University of Groningen

Abstract

All Internet routers contain buffers to hold packets during their transmission. This buffer is used to store packets that are waiting to be transmitted over the link. The size of the buffer is a critical parameter in determining the performance of the router. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

1. Introduction

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

2. Background

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

3. The Problem

The problem of sizing router buffers is a complex one. It involves determining the optimal buffer size for a given link and for a given network. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

4. The Solution

The solution is based on the principle of packet conservation. It is designed to be implemented in a wide range of network environments. The solution is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

5. Performance Testing

The solution was tested in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

6. Conclusions

The solution is designed to be implemented in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

7. Acknowledgments

The authors would like to thank the following people for their help and support:

8. References

[1] Van Jacobson, "Congestion Control and Packet Scheduling," *Proceedings of the ACM SIGCOMM Conference*, 1988.

9. Contact Information

Guoqing Xu: gxu@shu.edu.cn
Hui Kang: hkang@shu.edu.cn
Niek Meunander: n.meunander@azc.rug.nl

Routers with Very Small Buffers

Mihalis Anastasiadis¹, Yael Gafni², Adnan Gani³, Niek Meunander⁴, and Tom Bragdon⁵
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University

Abstract

Internet routers contain buffers to hold packets during their transmission. This buffer is used to store packets that are waiting to be transmitted over the link. The size of the buffer is a critical parameter in determining the performance of the router. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

1. Introduction

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

2. Background

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

3. The Problem

The problem of sizing router buffers is a complex one. It involves determining the optimal buffer size for a given link and for a given network. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

4. The Solution

The solution is based on the principle of packet conservation. It is designed to be implemented in a wide range of network environments. The solution is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

5. Performance Testing

The solution was tested in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

6. Conclusions

The solution is designed to be implemented in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

7. Acknowledgments

The authors would like to thank the following people for their help and support:

8. References

[1] Van Jacobson, "Congestion Control and Packet Scheduling," *Proceedings of the ACM SIGCOMM Conference*, 1988.

9. Contact Information

Mihalis Anastasiadis: m.anastasiadis@stanford.edu
Yael Gafni: ygafni@stanford.edu
Adnan Gani: agani@stanford.edu
Niek Meunander: n.meunander@azc.rug.nl
Tom Bragdon: tbragdon@stanford.edu

Experimental Study of Router Buffer Sizing

Yael Gafni¹, Niek Meunander², and Tom Bragdon³
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University

Abstract

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

1. Introduction

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

2. Background

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

3. The Problem

The problem of sizing router buffers is a complex one. It involves determining the optimal buffer size for a given link and for a given network. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

4. The Solution

The solution is based on the principle of packet conservation. It is designed to be implemented in a wide range of network environments. The solution is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

5. Performance Testing

The solution was tested in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

6. Conclusions

The solution is designed to be implemented in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

7. Acknowledgments

The authors would like to thank the following people for their help and support:

8. References

[1] Van Jacobson, "Congestion Control and Packet Scheduling," *Proceedings of the ACM SIGCOMM Conference*, 1988.

9. Contact Information

Yael Gafni: ygafni@stanford.edu
Niek Meunander: n.meunander@azc.rug.nl
Tom Bragdon: tbragdon@stanford.edu

Updating the Theory of Buffer Sizing

Yael Gafni¹, Niek Meunander², and Tom Bragdon³
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University
Department of Electrical Engineering, Stanford University

Abstract

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

1. Introduction

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

2. Background

The Internet has become a critical part of our lives. It is used for everything from e-mail to video conferencing. As a result, the demand for high performance network services is increasing. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

3. The Problem

The problem of sizing router buffers is a complex one. It involves determining the optimal buffer size for a given link and for a given network. This paper discusses the problem of sizing router buffers and presents a method for determining the optimal buffer size. The method is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

4. The Solution

The solution is based on the principle of packet conservation. It is designed to be implemented in a wide range of network environments. The solution is based on the principle of packet conservation and is designed to be implemented in a wide range of network environments.

5. Performance Testing

The solution was tested in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

6. Conclusions

The solution is designed to be implemented in a wide range of network environments. The results show that the solution is designed to be implemented in a wide range of network environments.

7. Acknowledgments

The authors would like to thank the following people for their help and support:

8. References

[1] Van Jacobson, "Congestion Control and Packet Scheduling," *Proceedings of the ACM SIGCOMM Conference*, 1988.

9. Contact Information

Yael Gafni: ygafni@stanford.edu
Niek Meunander: n.meunander@azc.rug.nl
Tom Bragdon: tbragdon@stanford.edu

We will talk about these results!

Using data/arguments from here

Results

BDP=Bandwidth x Delay (Round trip time)



BDP: Jacobson 90, Villamizar and Song 1994

BDP/ \sqrt{n} : Appenzeller, McKeown, Keslassy 2004

Sizing buffers for one TCP Reno flow

Fact: TCP sends one window's worth of data every RTT

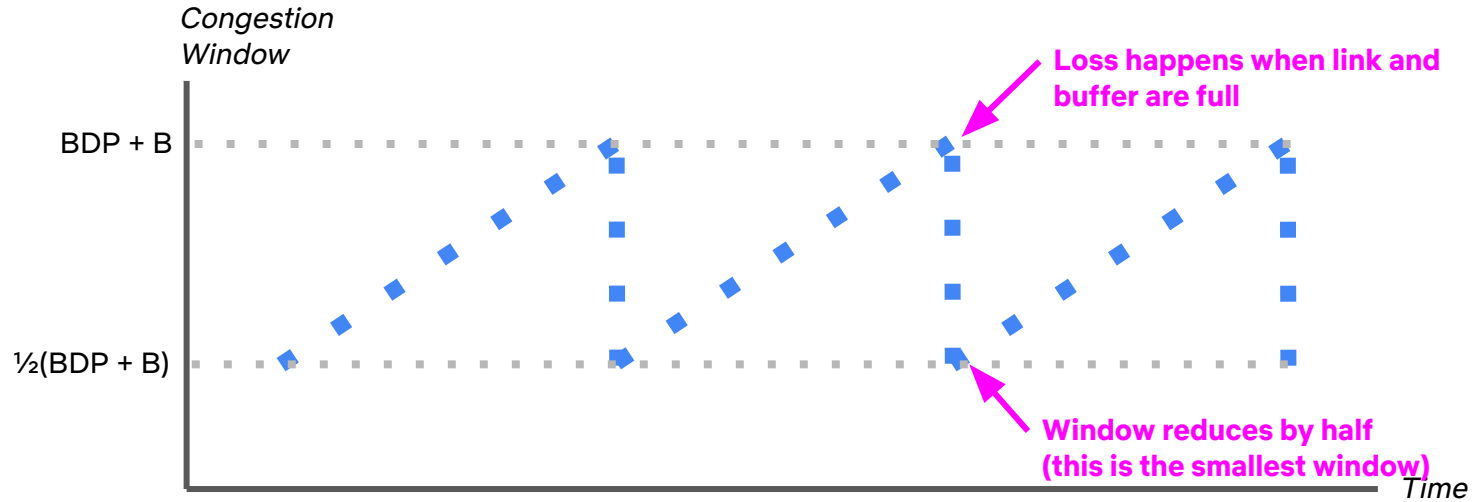
So if $\text{window} < \text{BDP} = \text{Capacity} * \text{RTT}$, then TCP sends at a rate of

$$\text{Window} / \text{RTT} < \text{BDP} / \text{RTT} < \text{Capacity}$$

Another fact: (This one takes a little more work, and a few assumptions)

Link is fully utilized over an RTT if and only if $\text{window} \geq \text{BDP}$.

Sizing buffers for one TCP Reno flow



Sizing buffers for one TCP Reno flow

All of this together:

1. If $\text{min window} \geq \text{BDP}$, then we have full link utilization.
2. $\text{Min window} = \frac{1}{2}(\text{BDP} + B)$

So for $\text{min window} \geq \text{BDP}$, we need:

$$\frac{1}{2}(\text{BDP} + B) \geq \text{BDP},$$

$$\text{BDP} + B \geq 2 \text{BDP},$$

$$B \geq \text{BDP}.$$

Aside about overhead

- TCP requires \geq BDP packets in flight to keep the link fully utilized during any RTT
- TCP reno requires *another* \geq BDP packets in the buffer to keep the link fully utilized over all RTTs
 - This is like 100% overhead!

You asked...

“How do new congestion control protocols like SWIFT or DCTCP affect ideal buffer size?”

Nice and flexible result

In a recent paper, we looked at:

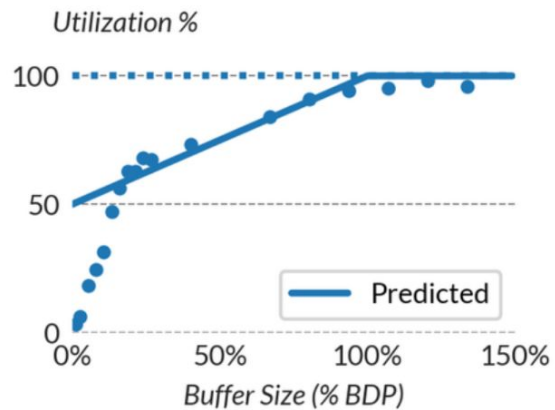
- What if you decrease the window by 30% (like Cubic) or $\frac{1}{8}$ (like Scalable TCP)?
- What if you want 90% link utilization?
- etc...

Buffer requirements for a single flow

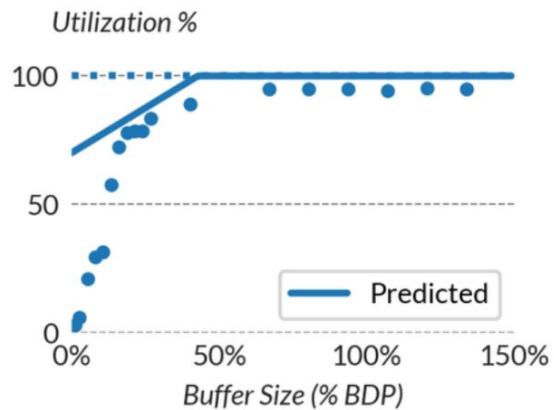
Algorithm	Full Utilization	90% Utilization
Reno	BDP	0.80 BDP
Cubic	0.42 BDP	0.28 BDP
BBR	0.25 BDP	0.15 BDP
Scalable	0.14 BDP	0.03 BDP
DCTCP	0.14 BDP	0.03 BDP

Experimental results

Reno



Cubic



Multiple TCP connections

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 is congested, it is busy 100% of the time; which is equivalent to making sure its buffer never goes empty. A widely used rule-of-thumb states that each link needs a buffer of size $B = \overline{RTT} \times C$, where \overline{RTT} 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 10Gb/s router linecard needs approximately $250\text{ms} \times 10\text{Gb/s} = 2.5\text{Gbits}$ of buffers; and the amount of buffering grows linearly with the line-rate. Such large buffers are challenging for router manufacturers, who must use large, slow, off-chip DRAMs. And queueing delays can be long, have high variance, and may destabilize the congestion control algorithms. In this paper we argue that the rule-of-thumb ($B = \overline{RTT} \times C$) is now outdated and incorrect for backbone routers. This is because of the large number of flows (TCP connections) multiplexed together on

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 biggest contributor to uncertainty in the Internet. Buffers cause queueing delay and delay-variance; when they overflow they cause packet loss, and when they underflow they can degrade throughput. Given the significance of their role, we might reasonably expect the dynamics and sizing of router buffers to be well understood, based on a well-grounded theory, and supported by extensive simulation and experimentation. This is not so.

Sizing buffers for multiple Reno flows

[Appenzeller, McKeown, Keslassy '04]

Paper context:

- Based on the previous argument, idea was that we needed buffers \geq BDP for core internet routers
- For 250ms RTT, 10 Gbps link (reasonable at the time), we would need 300 MB of buffers!

Main insights

- Buffer is there to absorb variability in packet arrivals
- n TCP connections behave very differently than one TCP connection
 - In particular, variability of their aggregate window is much lower
- This means that large internet routers can have smaller buffers

Theorem [Appenzeller, McKeown, Keslassy 2004]

If buffer is $\geq \text{BDP}/\sqrt{n}$ and [conditions apply] then link will be fully utilized with high probability

Conditions: TCP windows are

1. Uniformly distributed between $c_1 (\text{BDP}+B)/n$ and $c_2 (\text{BDP}+B)/n$
2. Independent

Proof

- [Do it on the board]
- Windows W_i are iid uniform, $2/3(\text{BDP} + B)/n \leq W_i \leq 4/3(\text{BDP} + B)/n$
- Useful inequality:

Bounds on sums of independent variables [\[edit\]](#)

Main articles: Hoeffding's inequality, Azuma's inequality, McDiarmid's inequality, Bennett's inequality, and Bernstein inequalities (probability theory)

Let X_1, X_2, \dots, X_n be independent random variables such that, for all i :

$$a_i \leq X_i \leq b_i \text{ almost surely.}$$

$$c_i := b_i - a_i$$

$$\forall i : c_i \leq C$$

Let S_n be their sum, E_n its expected value and V_n its variance:

$$S_n := \sum_{i=1}^n X_i$$

$$E_n := \mathbb{E}[S_n] = \sum_{i=1}^n \mathbb{E}[X_i]$$

$$V_n := \text{Var}[S_n] = \sum_{i=1}^n \text{Var}[X_i]$$

It is often interesting to bound the difference between the sum and its expected value. Several inequalities can be used.

1. Hoeffding's inequality says that:

$$\Pr[|S_n - E_n| > t] < 2 \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right) < 2 \exp\left(-\frac{2t^2}{nC^2}\right)$$

Multiple Reno flows [SAM21]

If n connections share a link and [conditions apply] then:

1. If buffer is $\geq \text{BDP}/\sqrt{n}$, link will be fully utilized
2. Utilization is at least $1 - \Omega(1/\sqrt{n})$, independent of buffer size

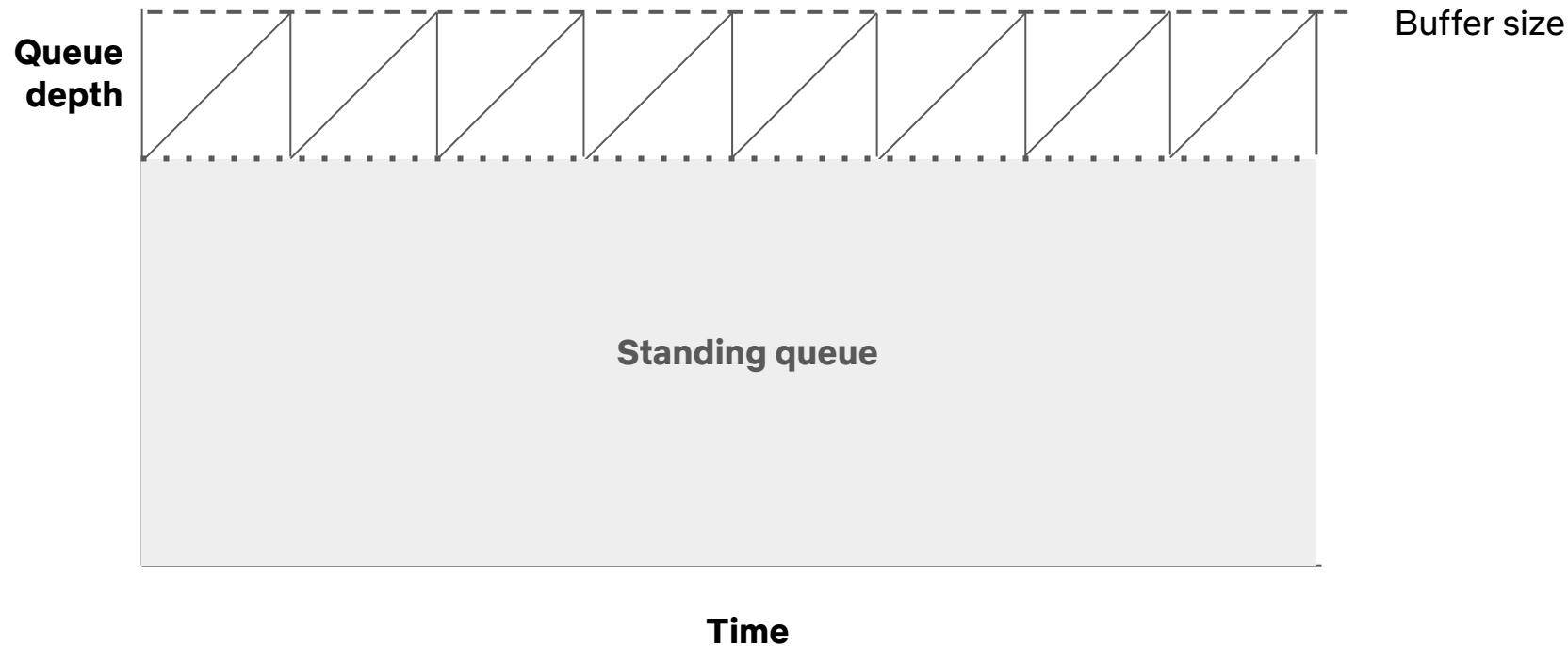
Conditions:

1. **Fair:** connections send roughly same amount of data
2. **Desynchronized:** only a few connections decrease windows at same time

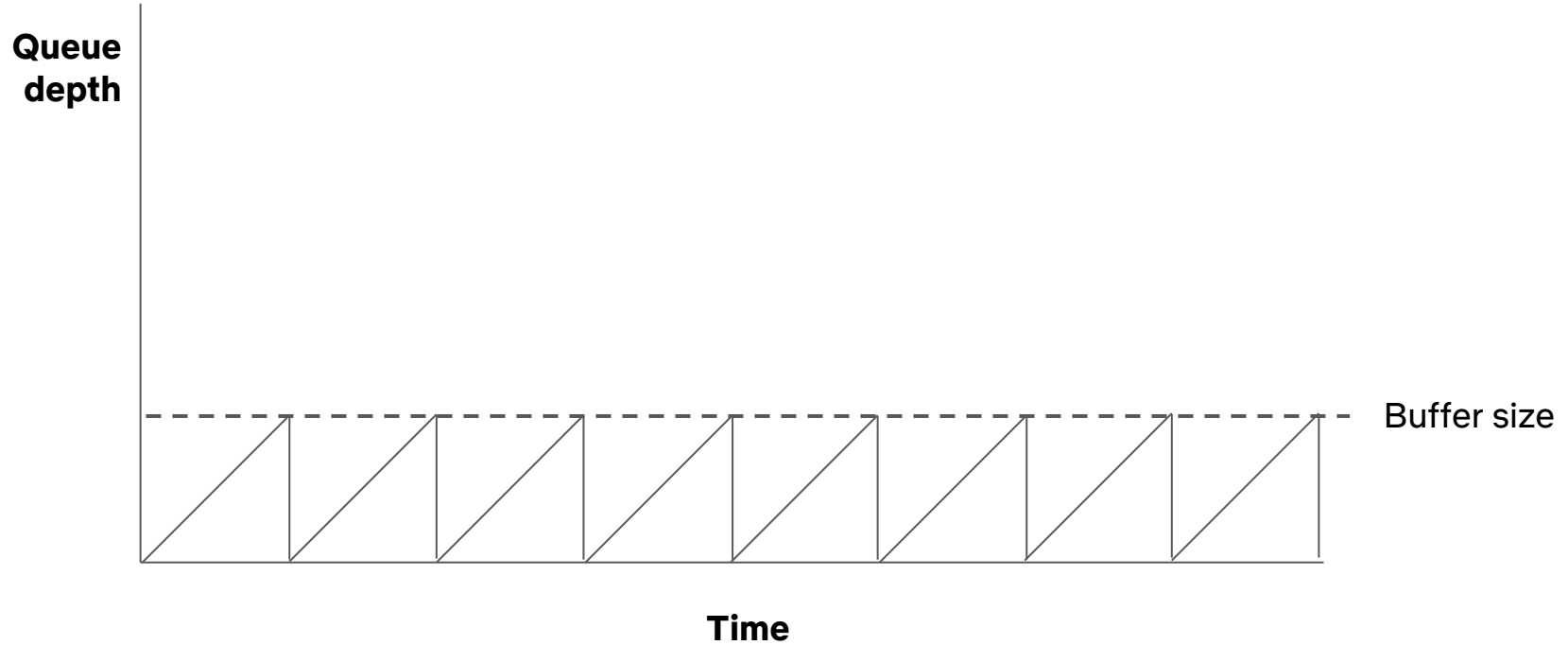
Why do we need the two conditions:

1. Fairness
2. No synchronization

Intuition: buffer only needs to handle variability

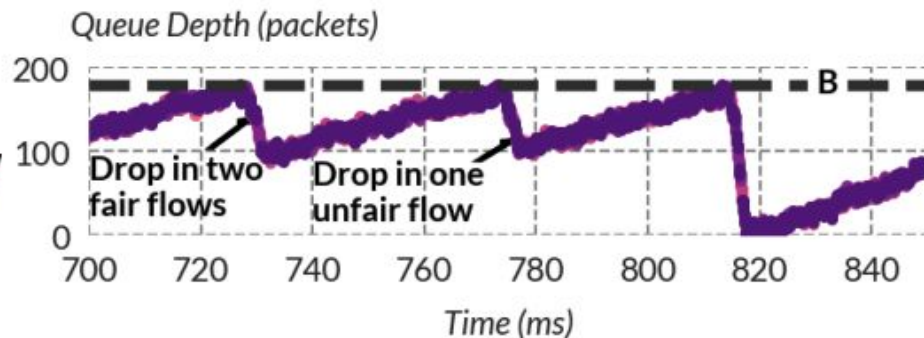
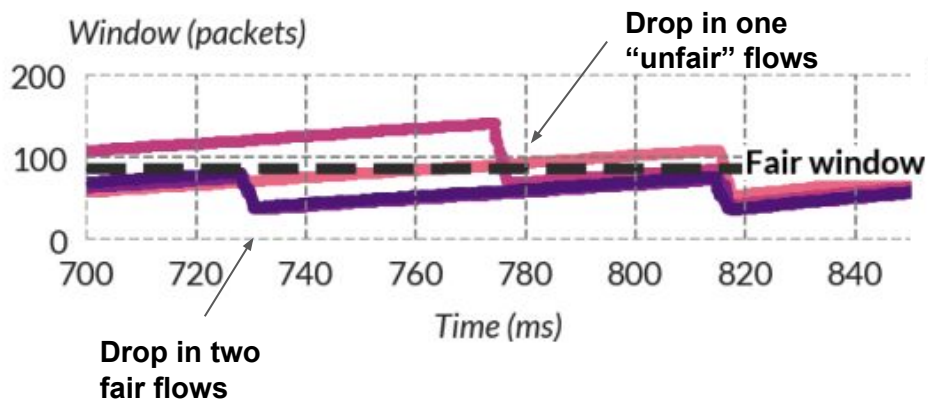


Intuition: buffer only needs to handle variability



Unfairness increases queue variability

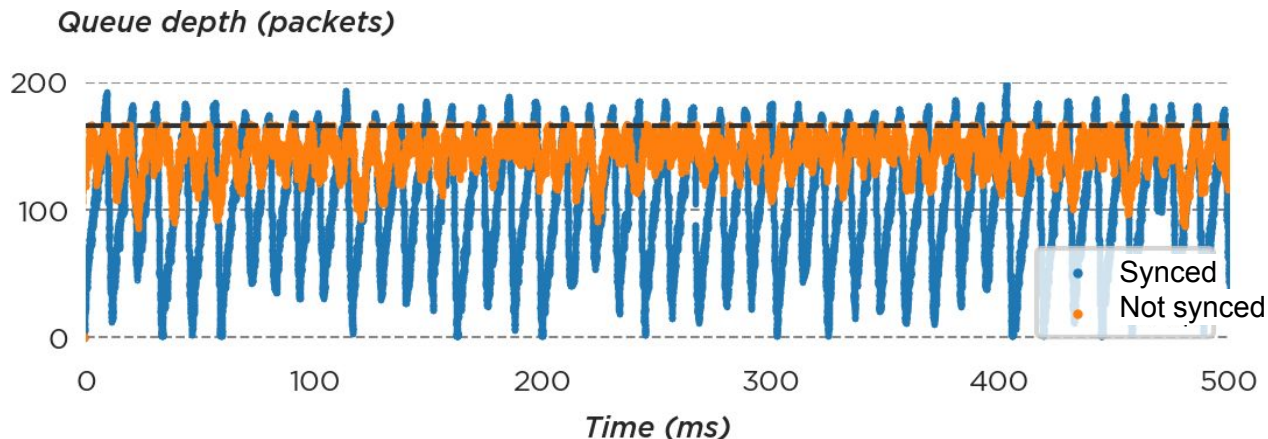
If a TCP flow has more data in flight, it will back off more, causing a larger drop in queue depth (and larger required buffer)



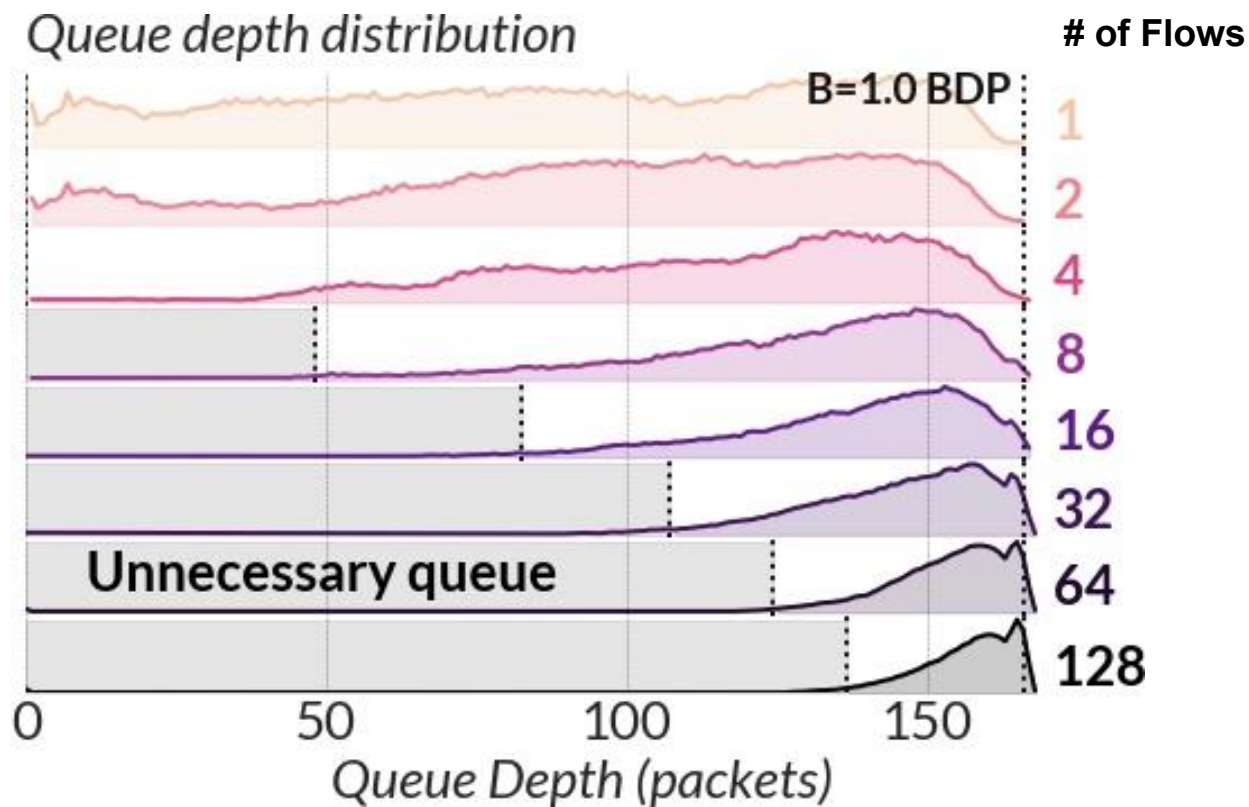
Synchronization increases queue variability

If everyone decreases windows at once, queues will fluctuate more

Note: congestion control designers can reduce synchronization, e.g. by randomly decreasing windows



Experimental results



Experiments

You asked...

“I appreciated a brief discussion in the conclusion, describing why router vendors might be uncomfortable switching to a model with fewer buffers, especially when operators would be more likely to buy the router with larger buffers. I think this problem of bringing research into practice when the results are almost too promising is pretty universal across many fields! How should a researcher go about convincing operators of a surprising research result, outside of the evaluation performed in the paper?”

Experimental results

Fair number of experiments have been run:

- [Experimental Study of Router Buffers](#) [NB, YG, MG, NM, GS 2008]
- [Buffer sizing and Video QoE Measurements at Netflix](#) [**BS**, BW, TH, TR, JL, NM 2019]
- [Buffer sizing experiments at Facebook](#) [NB, PL, YG 2019]

Typically:

- Reduce buffer size of one pair of a load-balanced link, see what happens
- Can get away with smaller buffers, but not clear how it maps to theory

Conclusion

Why is this tricky?

- Buffer sizes are a product of congestion control choices
 - Changing TCP can reduce buffer size requirements
- Congestion control is done by content providers, who don't control or measure buffers
- ISPs don't get to control or measure traffic performance
 - But buffer size only matters when links are congested
- Router manufacturers have to build to the worst case

We should only need small buffers

Modern TCP requires smaller buffers than Reno

Relationship between buffers and utilization is a consequence of congestion control choices

Should be able to get away with buffers of 10-100 packets.

Thanks!

bspang@stanford.edu