Transmitting Traffic in Circuit-Switched Networks

ROB MCGUINNESS

Talk Outline

Introduction

Circuit-Switched Endhost Networking

- Kernel module
- Kernel-bypass

Conclusion

Talk Outline

Introduction

Circuit-Switched Endhost Networking

- Kernel module
- Kernel-bypass

Conclusion

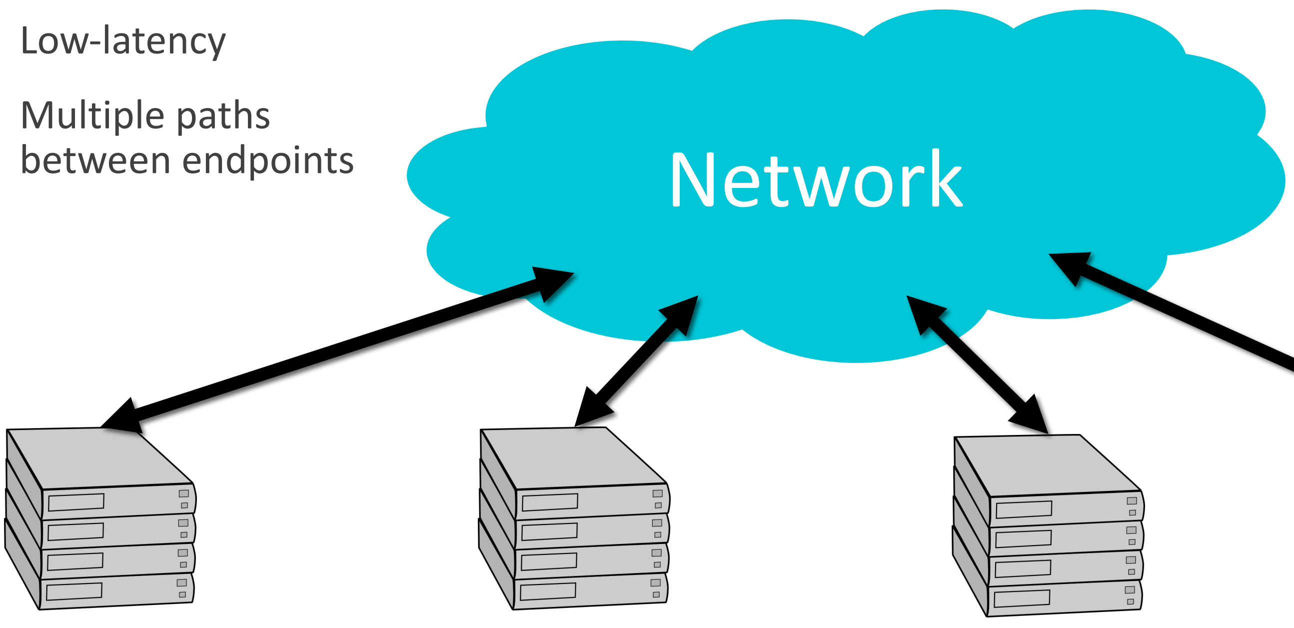
Modern-Day Datacenters

Modern-Day Datacenters

High-bandwidth

Low-latency

Multiple paths between endpoints



Datacenter growth

Datacenter traffic doubling approx. every year[1]

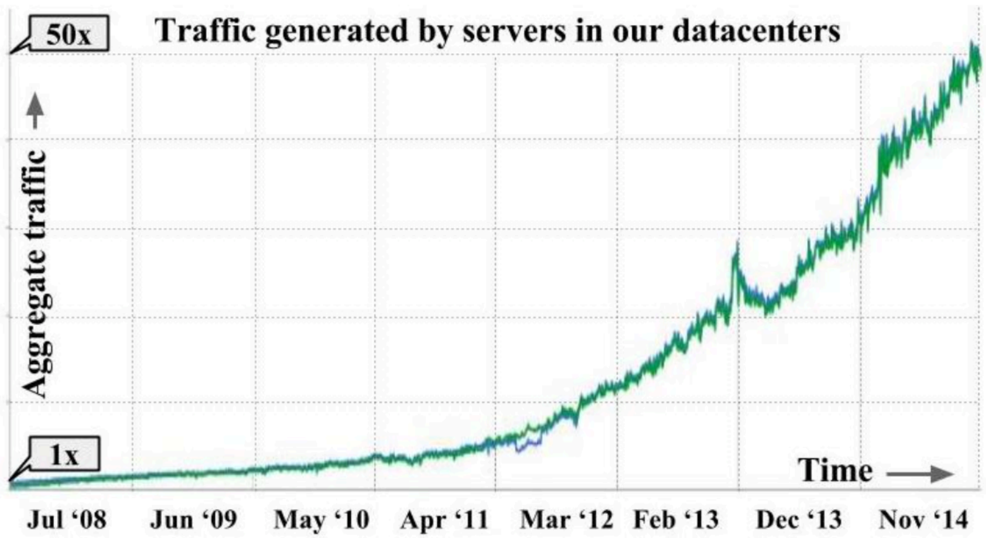
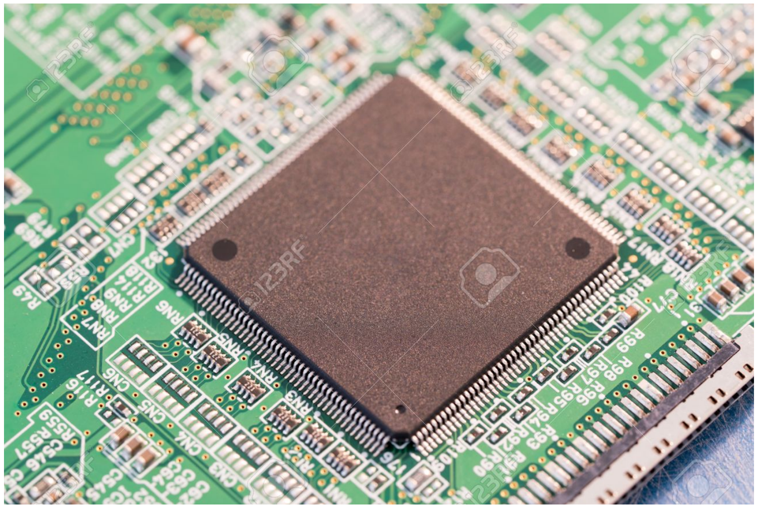


Figure 1: Aggregate server traffic in our datacenter fleet.

Higher-speed networks are costly

Moore's law applies to silicon-based packet switch chips[2]

High-speed packet switches will eventually become prohibitively expensive[3]



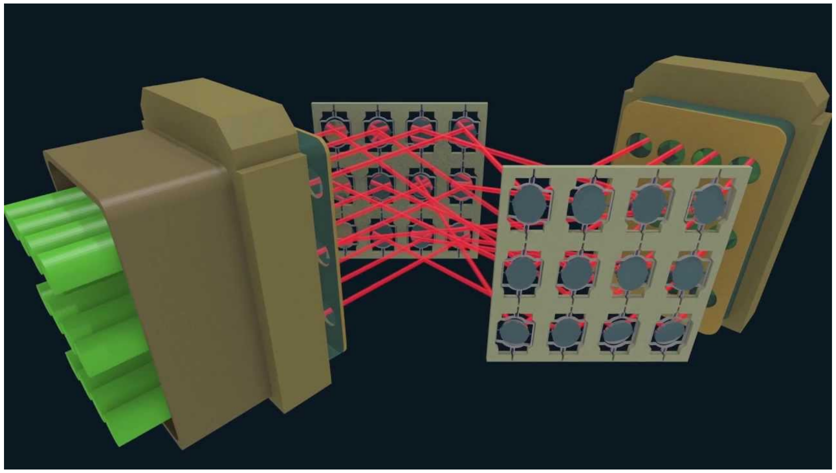
[2]: M. Taylor et al, Is dark silicon useful?, DAC '12.

[3]: N. Farrington et al, Helios, SIGCOMM '10.

Optical circuit-switches offer a solution

High-bandwidth optical switches are cheaper and save energy[4]

Used commonly in wide-area networking



[4]: W. Mellette et al, RotorNet, SIGCOMM '17.

Optical circuit-switches offer a solution

High-bandwidth optical switches are cheaper and save en

Us
ne

Effectively utilizing circuit switches is hard

What is a packet switch?

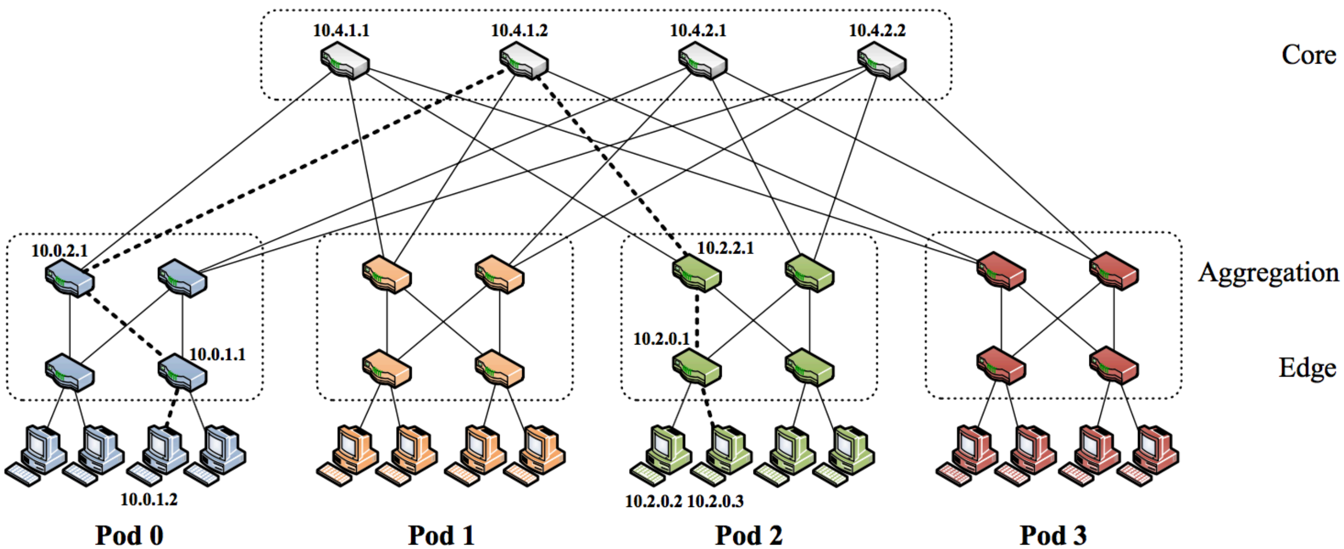


Figure 3: Simple fat-tree topology. Using the two-level routing tables described in Section 3.3, packets from source 10.0.1.2 **to destination** 10.2.0.3 **would take the dashed path.**

Image: M. Al-Fares et al, A Scalable Commodity Datacenter Network Architecture, SIGCOMM ‘08.

What is a packet switch?

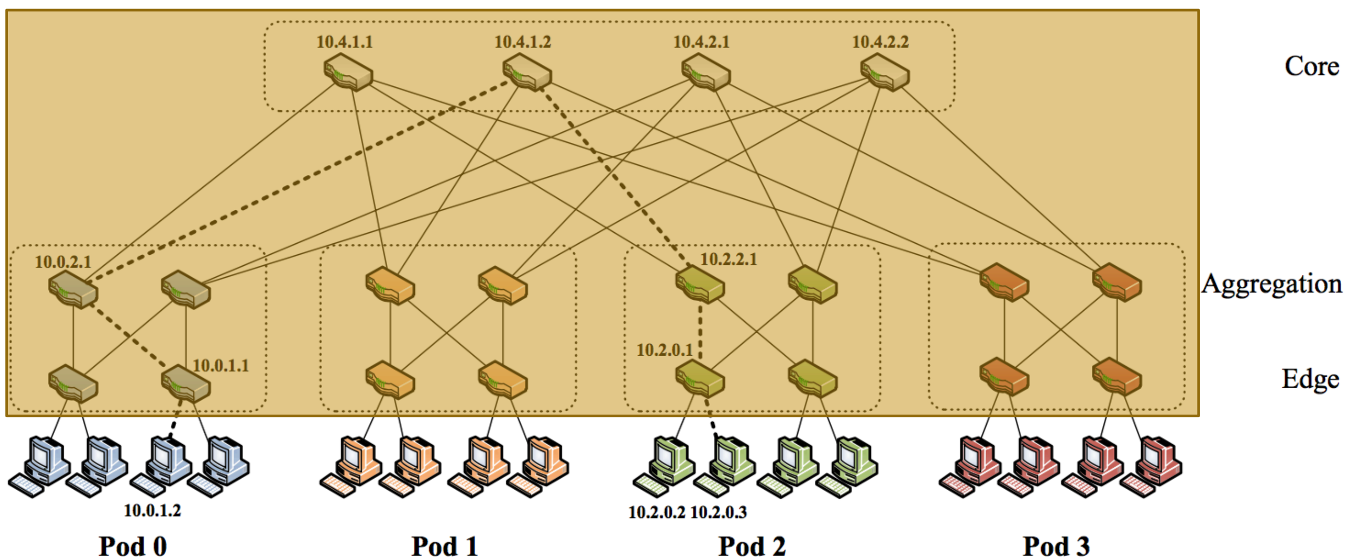
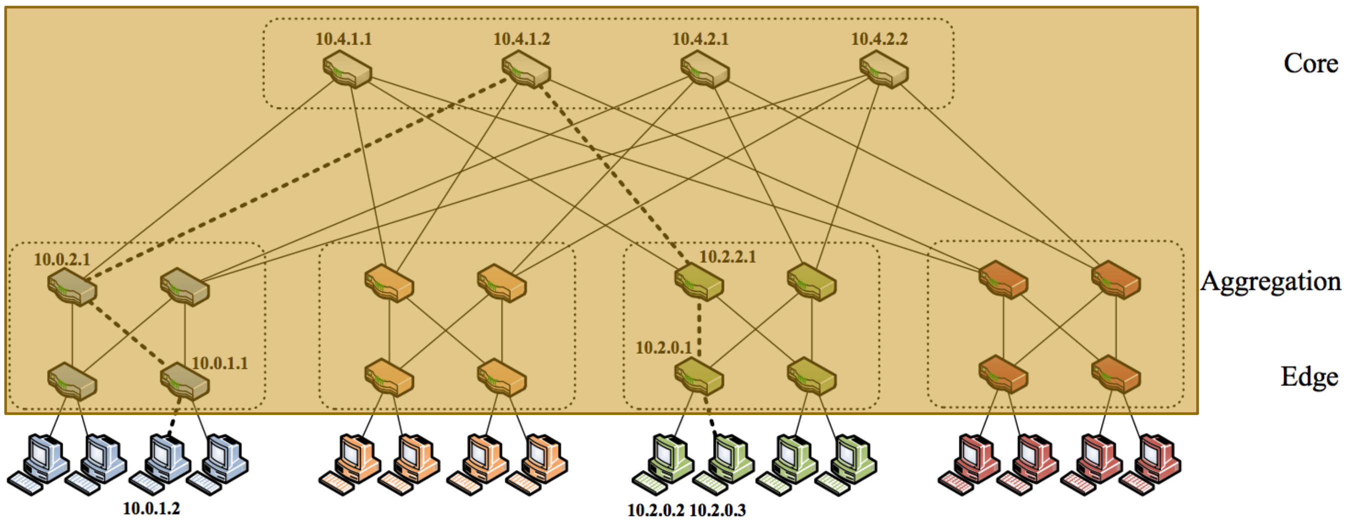


Figure 3: Simple fat-tree topology. Using the two-level routing tables described in Section 3.3, packets from source 10.0.1.2 **to destination** 10.2.0.3 **would take the dashed path.**

Image: M. Al-Fares et al, A Scalable Commodity Datacenter Network Architecture, SIGCOMM '08.

What is a packet switch?



Packet switches connect servers together in a topology

Figure 3: Simple fat-tree topology. Using the two-level routing tables described in Section 3.3, packets from source 10.0.1.2 **to destination** 10.2.0.3 **would take the dashed path.**

Image: M. Al-Fares et al, A Scalable Commodity Datacenter Network Architecture, SIGCOMM '08.

What is a packet switch?

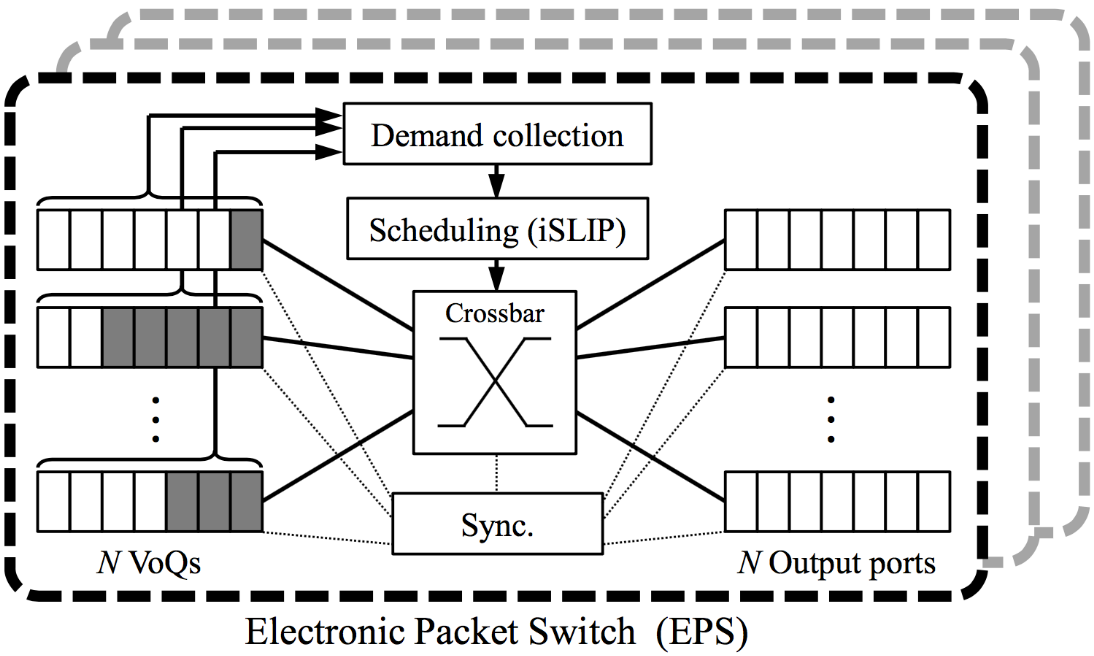


Image: W. Mellette et al, RotorNet, SIGCOMM '17.

What is a circuit switch?

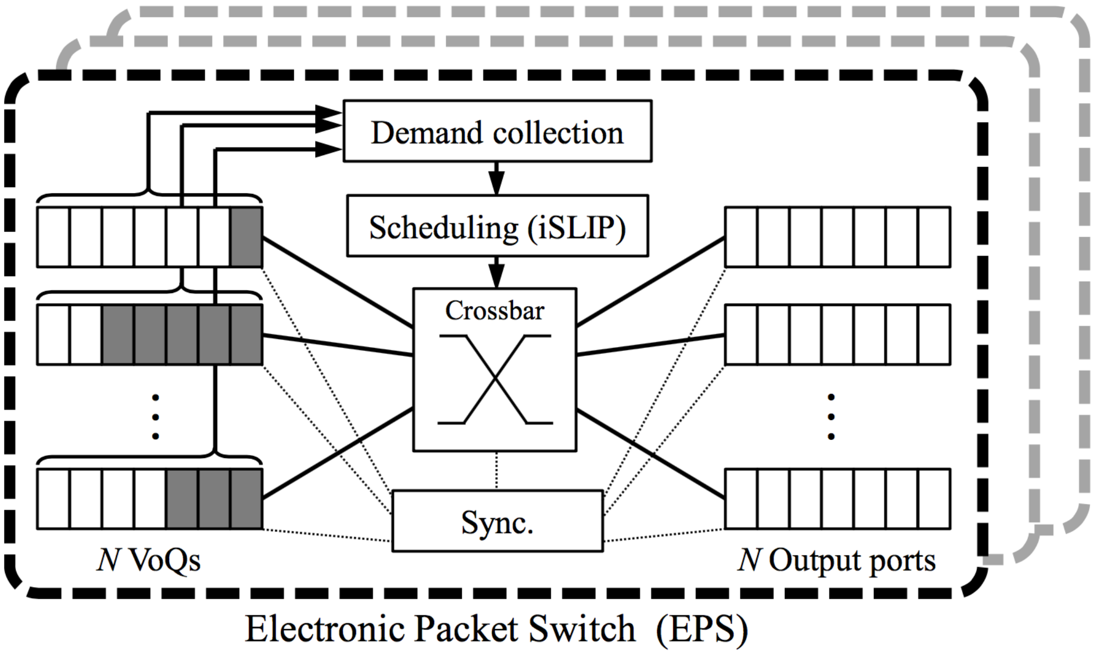


Image: W. Mellette et al, RotorNet, SIGCOMM '17.

What is a circuit-switch?

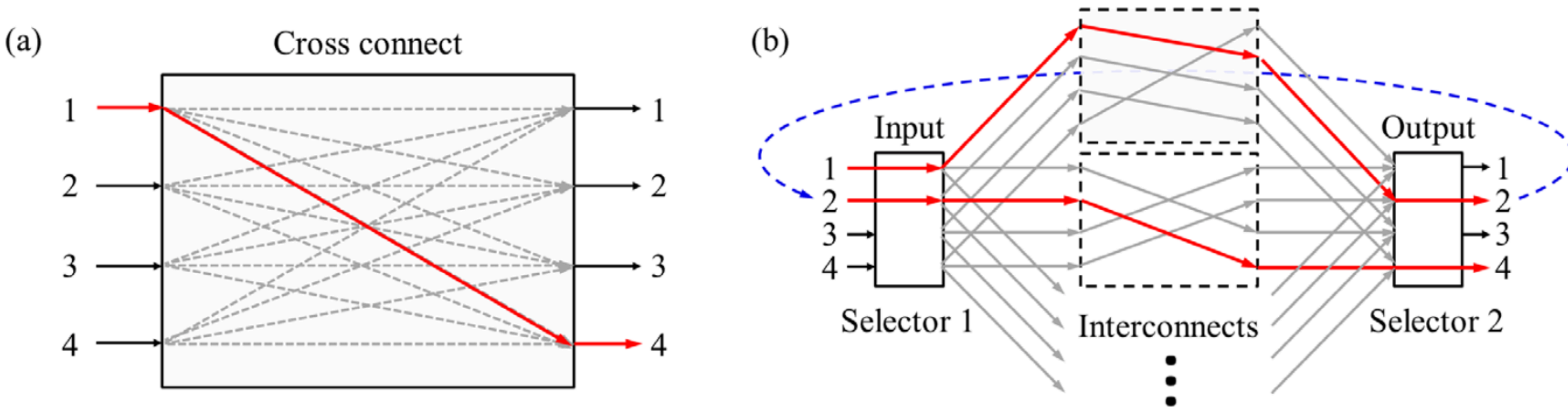


Image: W. Mellette et al, 61 Port 1 x 6 Selector Switch for Data Center Networks, OFC ‘16.

What is a circuit-switch?

Can't send to any destination anytime

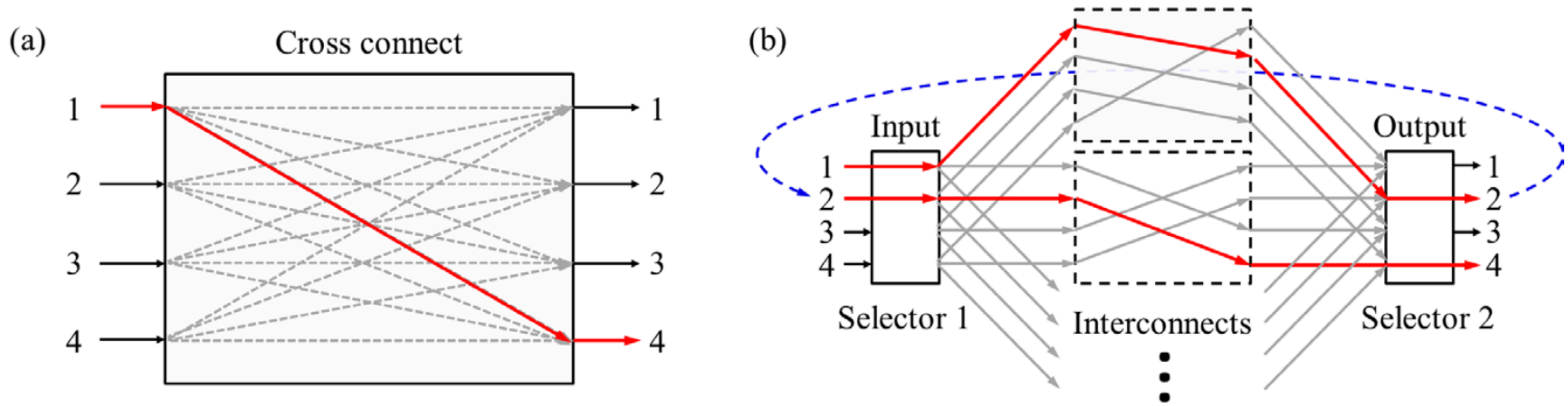
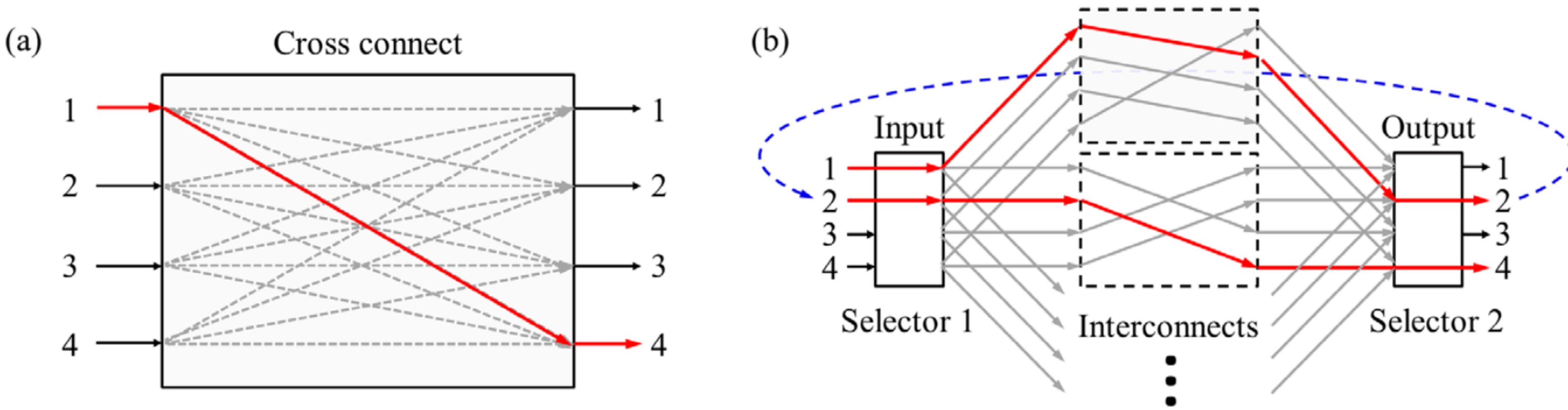


Image: W. Mellette et al, 61 Port 1 x 6 Selector Switch for Data Center Networks, OFC '16.

What is a circuit-switch?

Can’t send to any destination anytime



Switches take time to reconfigure

Image: W. Mellette et al, 61 Port 1 x 6 Selector Switch for Data Center Networks, OFC ‘16.

Circuit switching challenges

New hardware

New topologies

New protocols

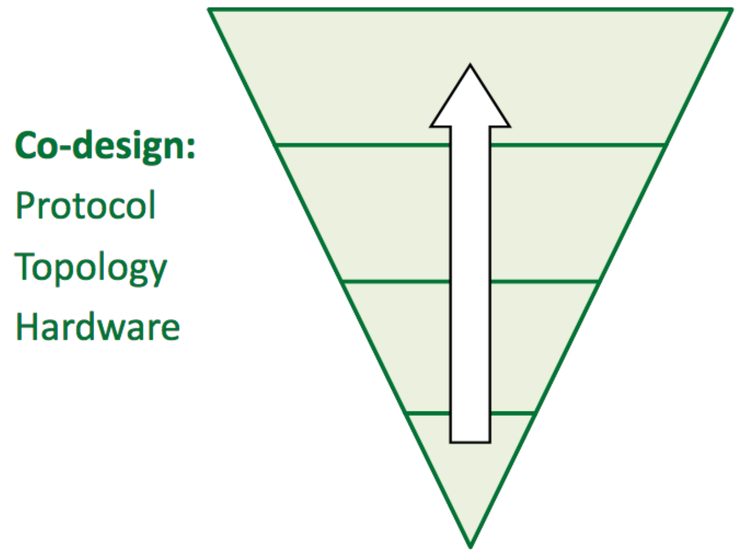


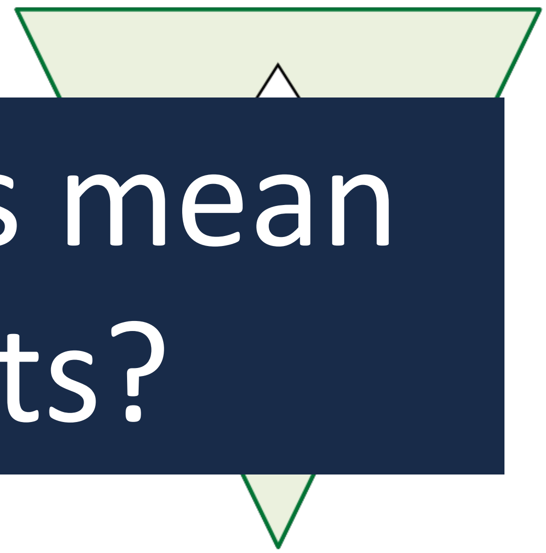
Image: W. Mellette et al, RotorNet, SIGCOMM ‘17.

Circuit switching challenges

New hardware

Ne

Ne



What does this mean for endhosts?

Packet switching and endhost annimation

Link to Packet/Circuit Annimation

Circuit switching and endhosts

Goal: to design new endpoint protocols to effectively leverage the new hardware and topologies in circuit-switched networks

Talk Outline

Introduction

Circuit-Switched Endhost Networking

- Kernel module
- Kernel-bypass

Conclusion

Endhost networking comparison

PACKET SWITCHED NETWORKS

+ Can send any packet at any time

+ Endhosts don't need to be network-aware

- Links may be congested

- Need to ACKnowledge packets

CIRCUIT SWITCHED NETWORKS

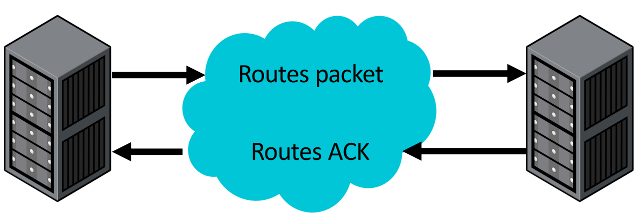
- Must wait for links to become available

- Endhosts store/interact with network state

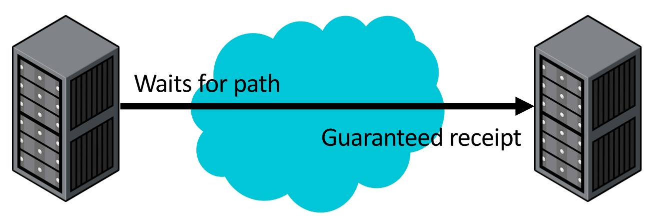
+ Links are dedicated to a single pair of nodes

+ Guaranteed receipt of packets

Packet-Switched Network



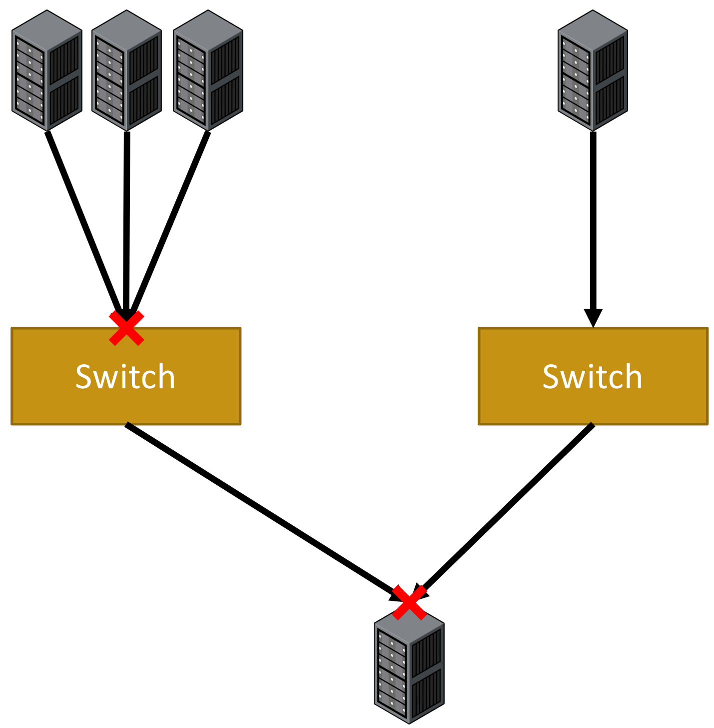
Circuit-Switched Network



Networking flow control

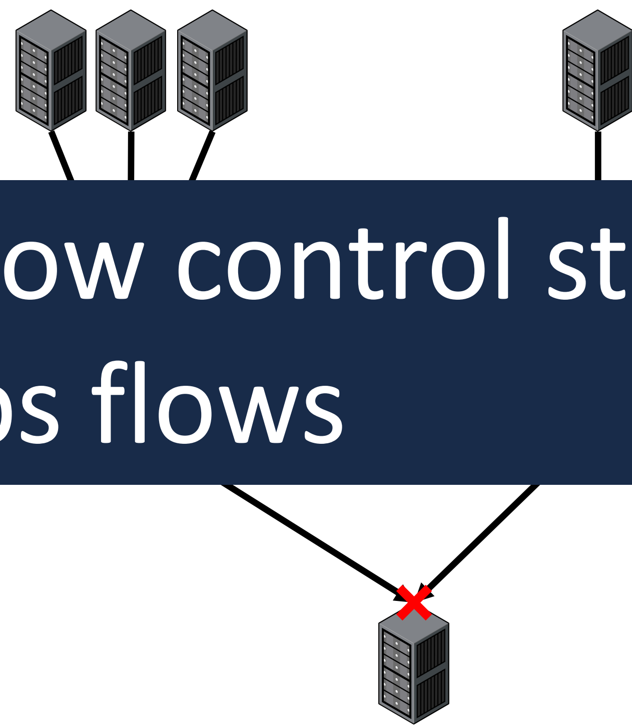
Flows contest for network bandwidth

Flow control reduces packet loss and increases throughput



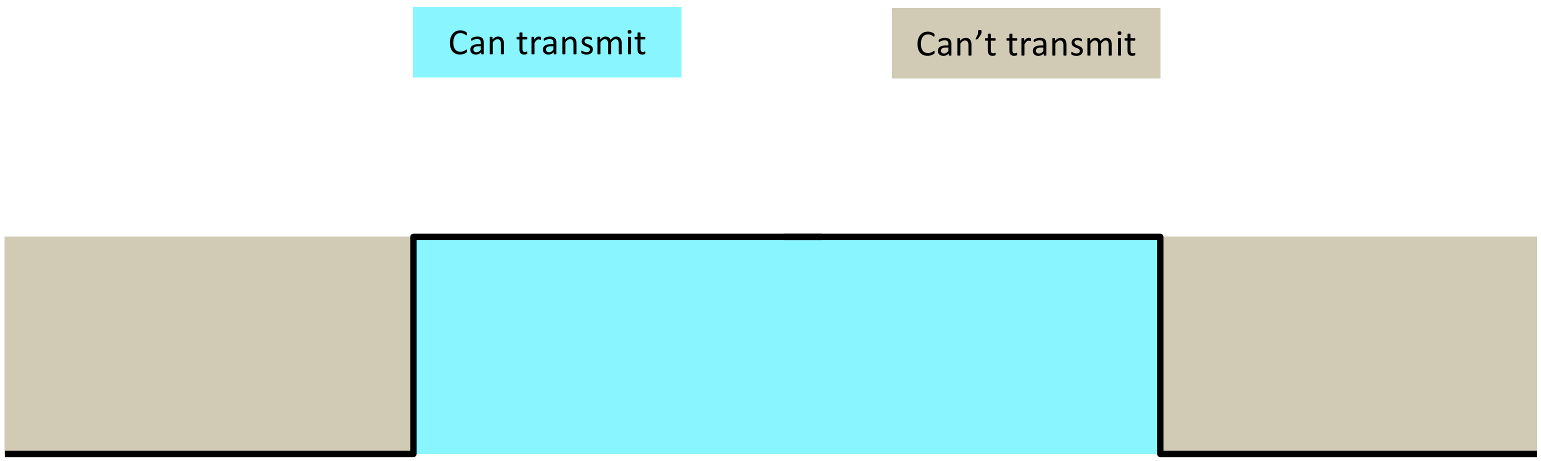
Networking flow control

Flows contest for network bandwidth

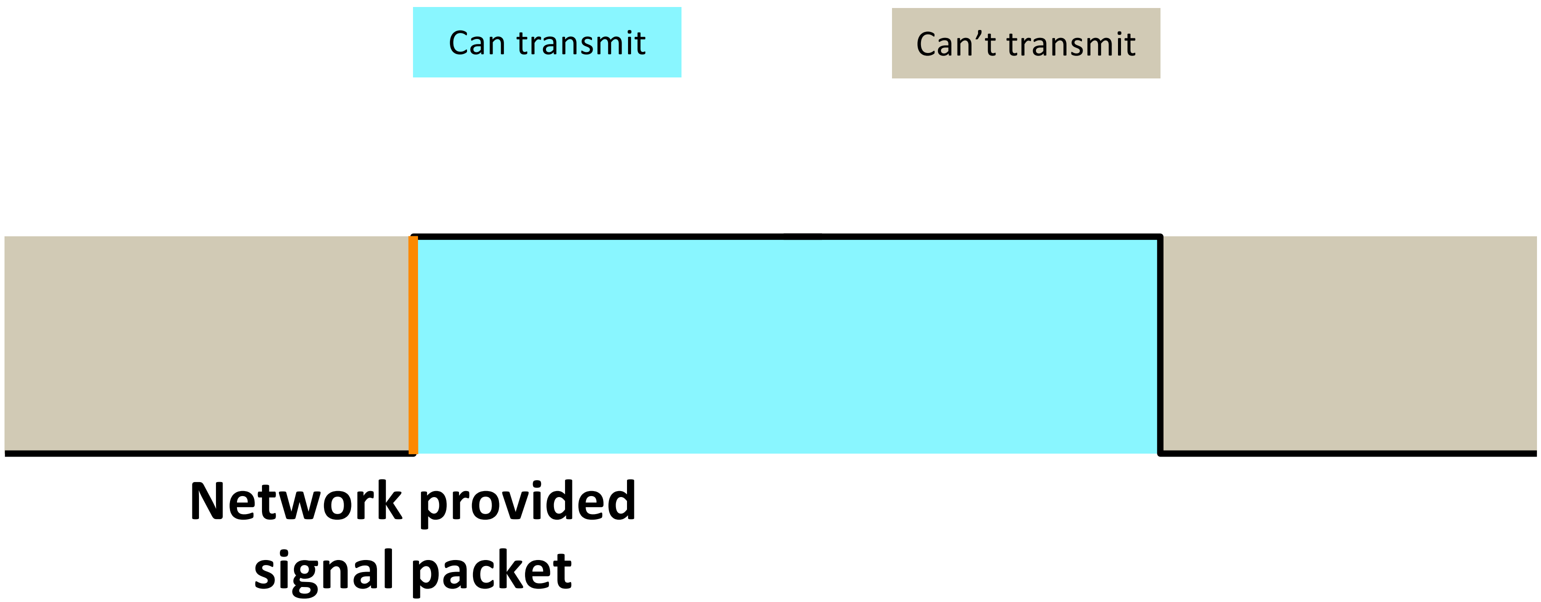


Circuit-switched flow control starts and stops flows

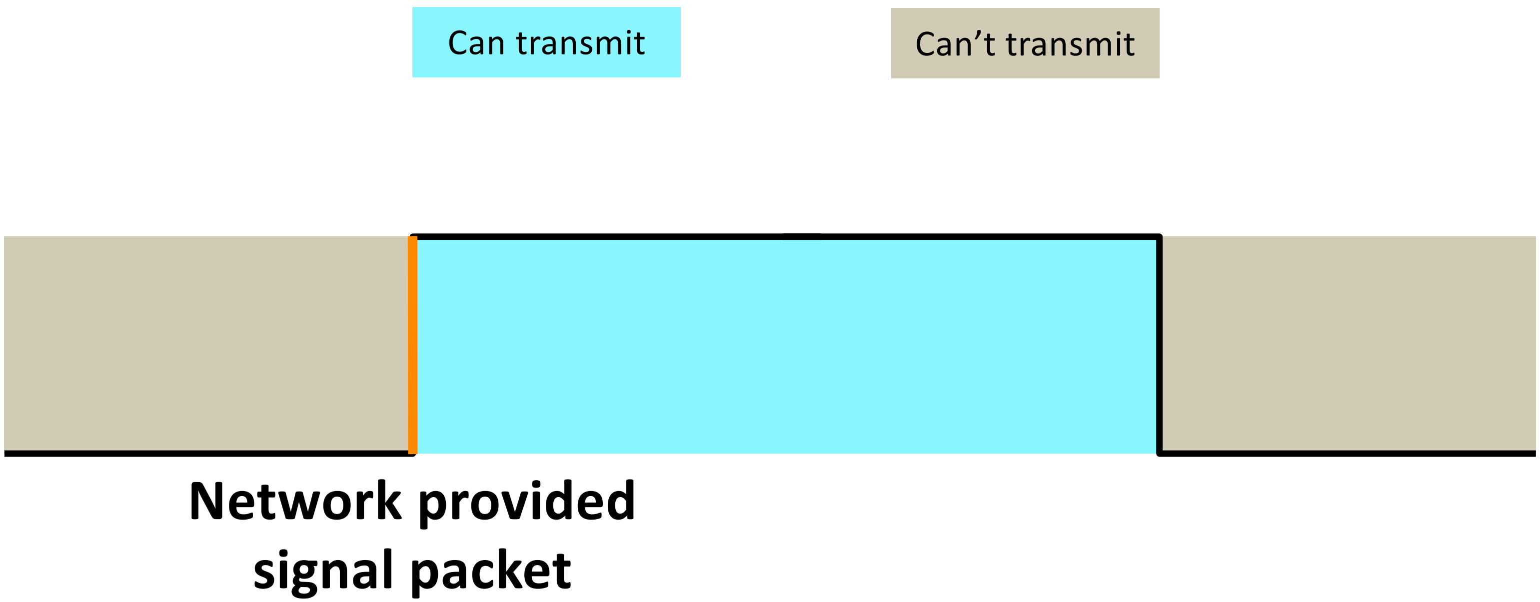
TDMA Flow Control



TDMA Flow Control



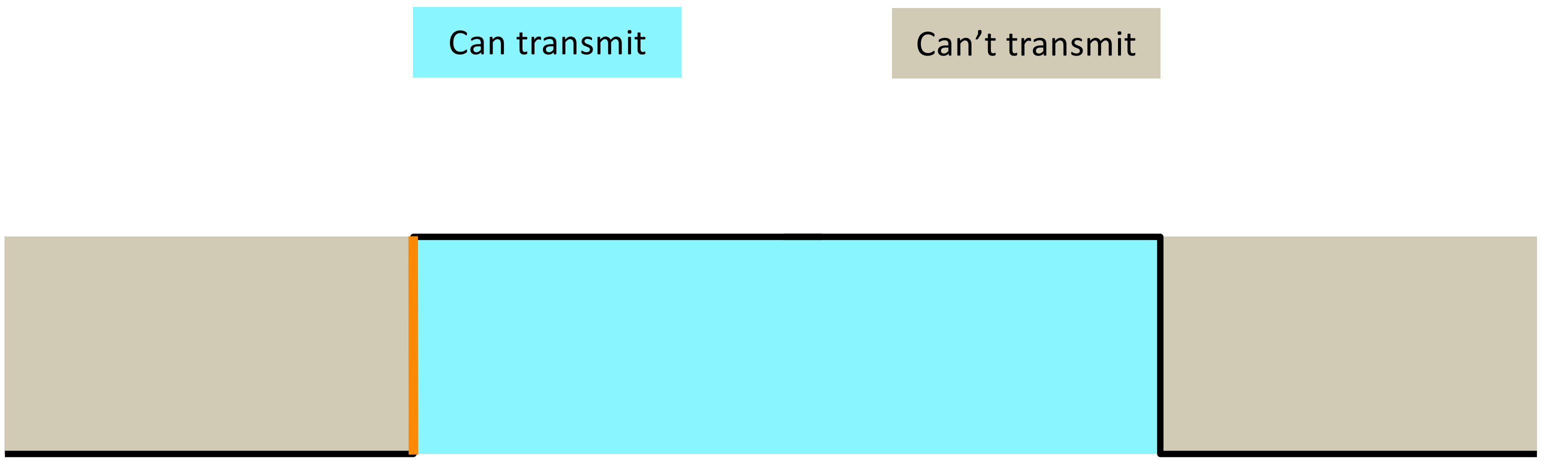
TDMA Flow Control



Uptime + downtime = **Cycle time**

(downtime/cycle time) = **Duty cycle**

TDMA Flow Control



Network provided signal packet

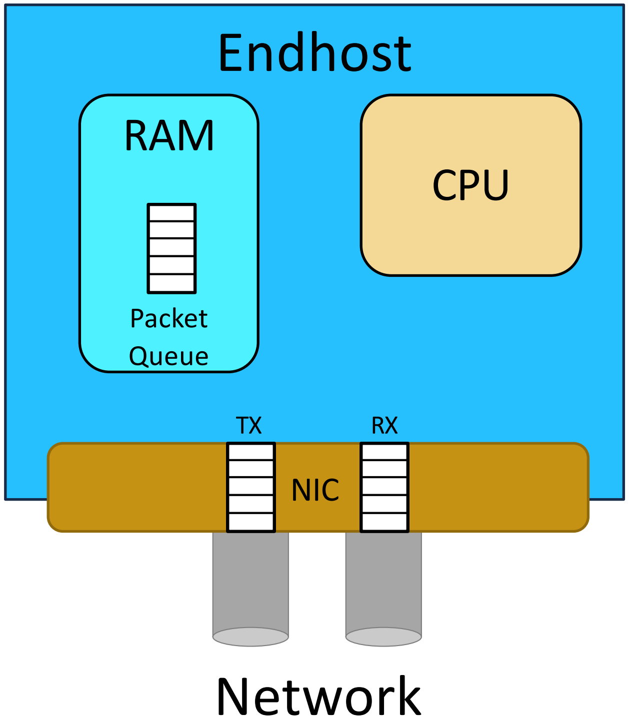
One cycle = **Timeslot**

Uptime + downtime = **Cycle time**

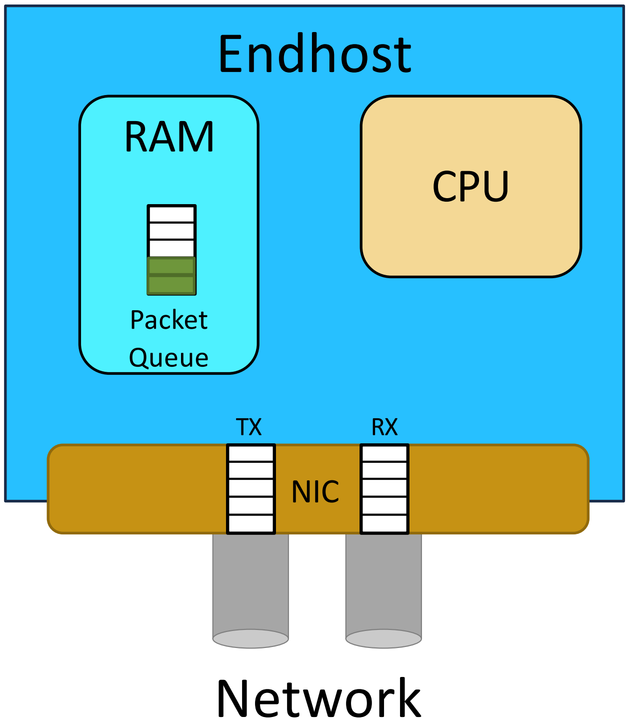
(downtime/cycle time) = **Duty cycle**

How can we implement TDMA flow control at endhosts to provide the best performance to optical datacenter networks?

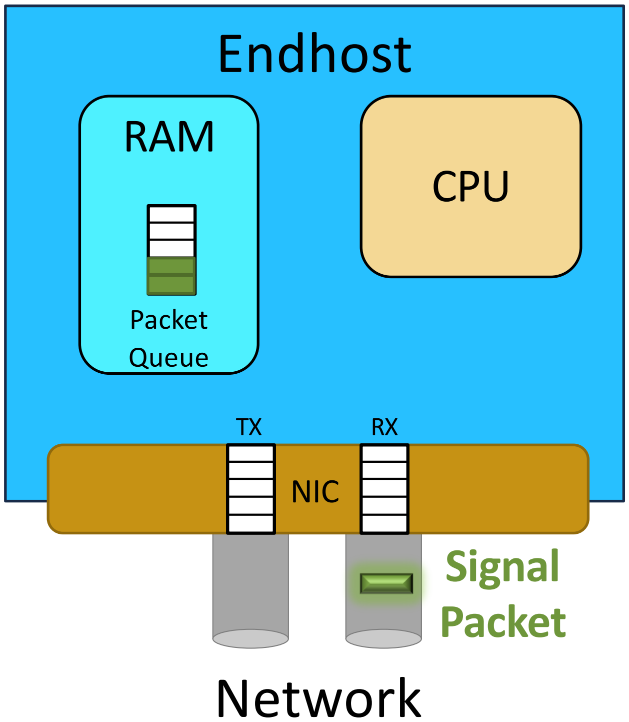
Handling TDMA network signals



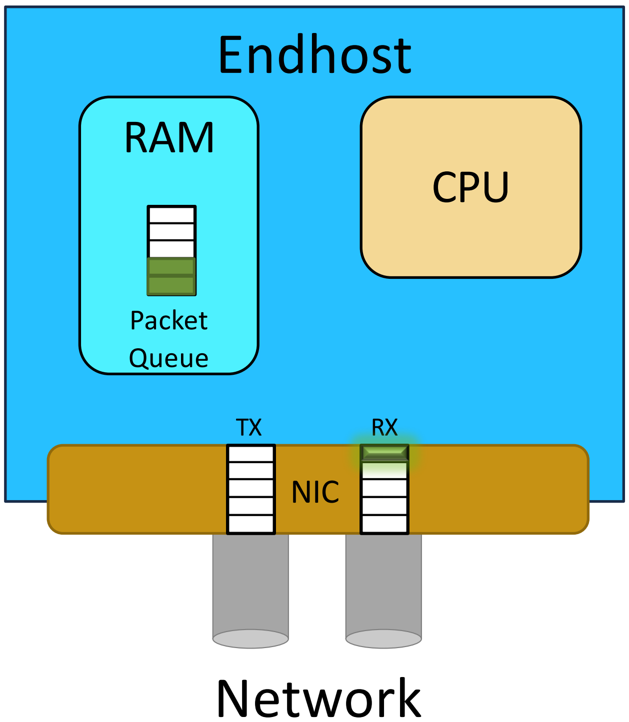
Handling TDMA network signals



Handling TDMA network signals

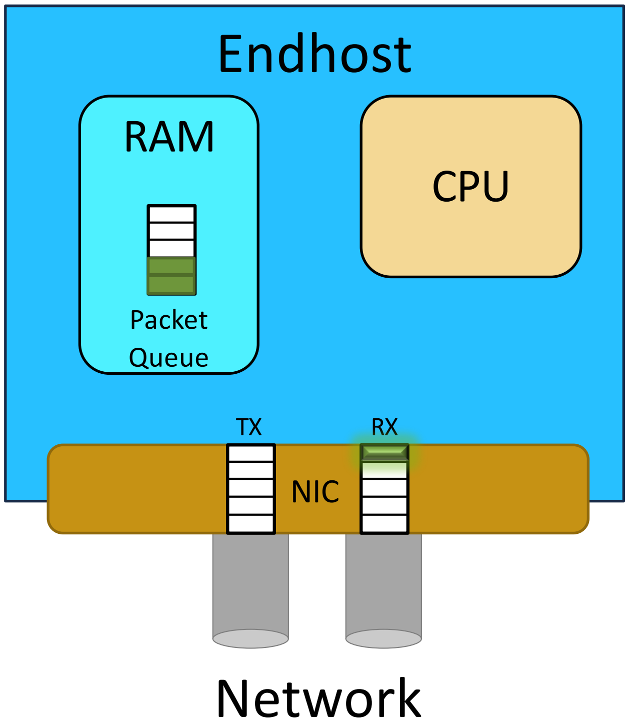


Handling TDMA network signals



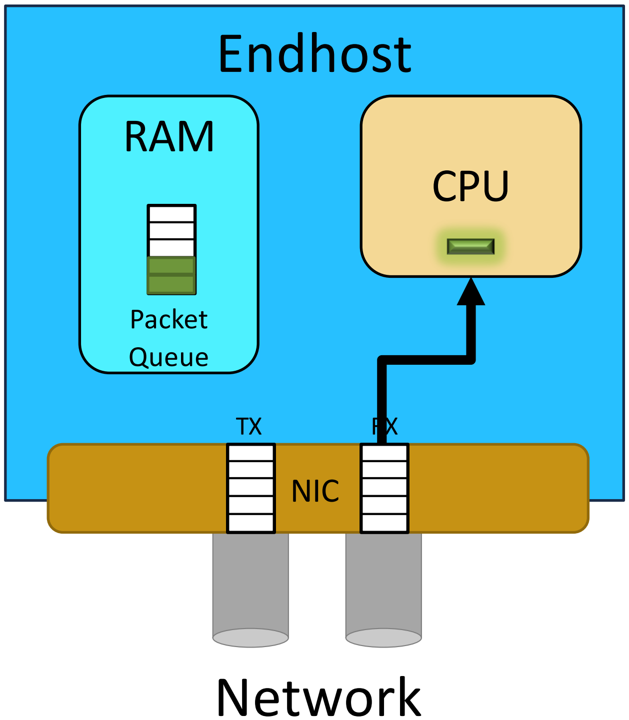
Handling TDMA network signals

1. CPU gets an interrupt from the NIC



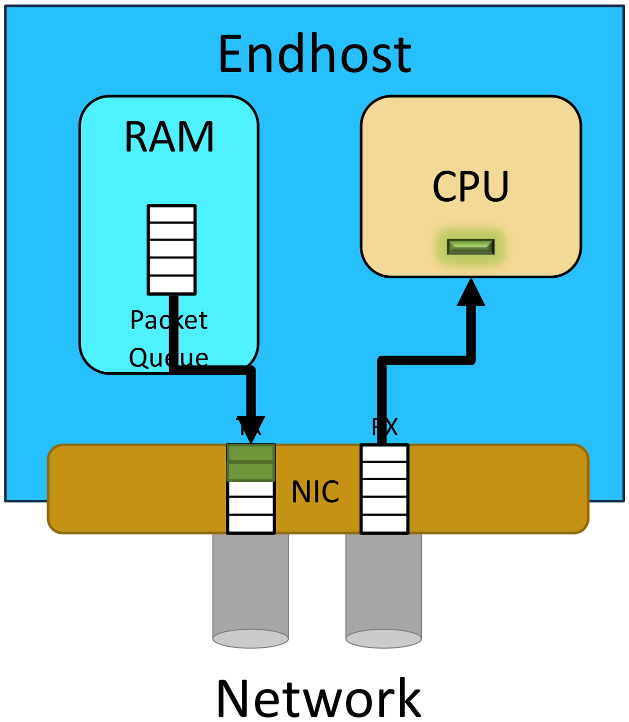
Handling TDMA network signals

1. CPU gets an interrupt from the NIC
2. CPU pulls the signal packet from the NIC



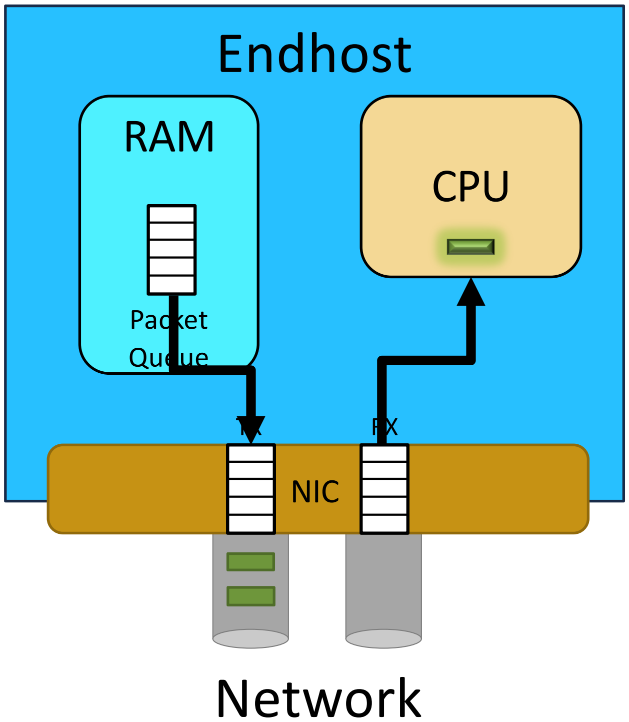
Handling TDMA network signals

1. CPU gets an interrupt from the NIC
2. CPU pulls the signal packet from the NIC
3. CPU copies outgoing packets to TX queue



Handling TDMA network signals

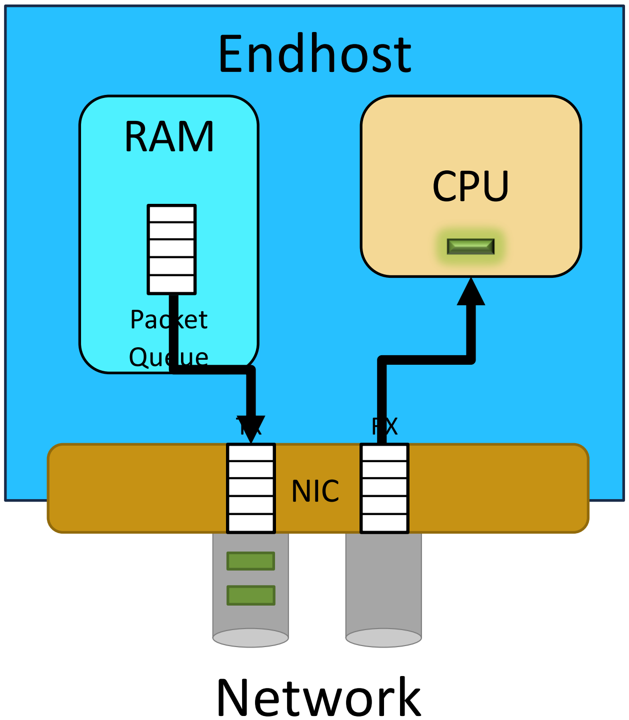
1. CPU gets an interrupt from the NIC
2. CPU pulls the signal packet from the NIC
3. CPU copies outgoing packets to TX queue
4. NIC sends packets to the network



Handling TDMA network signals

1. CPU gets an interrupt from the NIC
2. CPU pulls the signal packet from the NIC
3. CPU copies outgoing packets to TX queue
4. NIC sends packets to the network

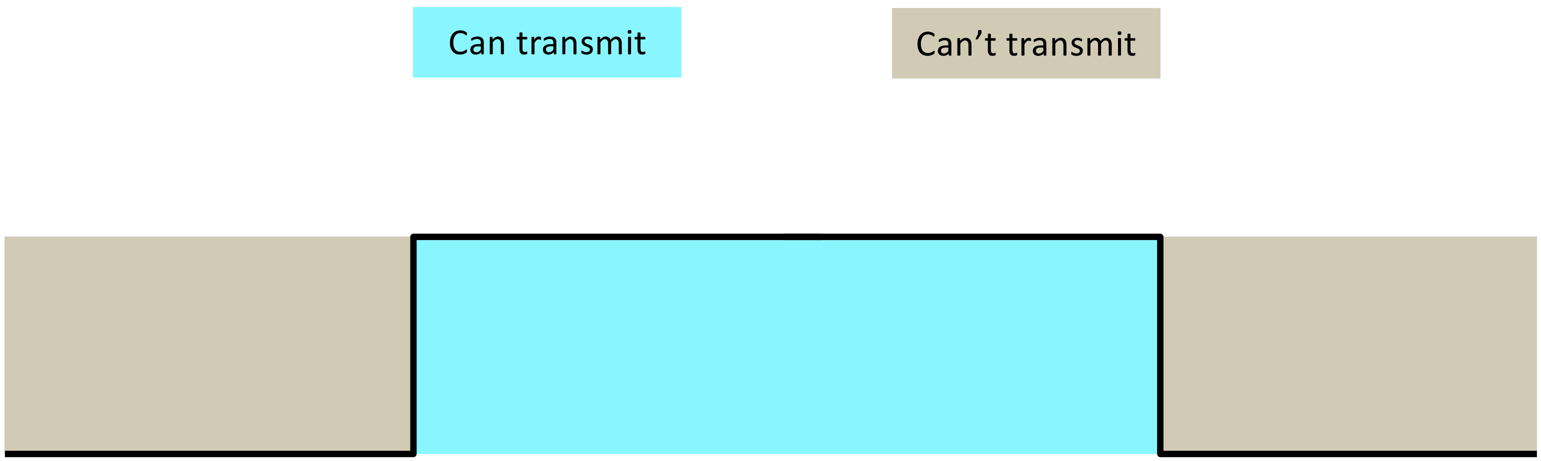
All these operations have variance!



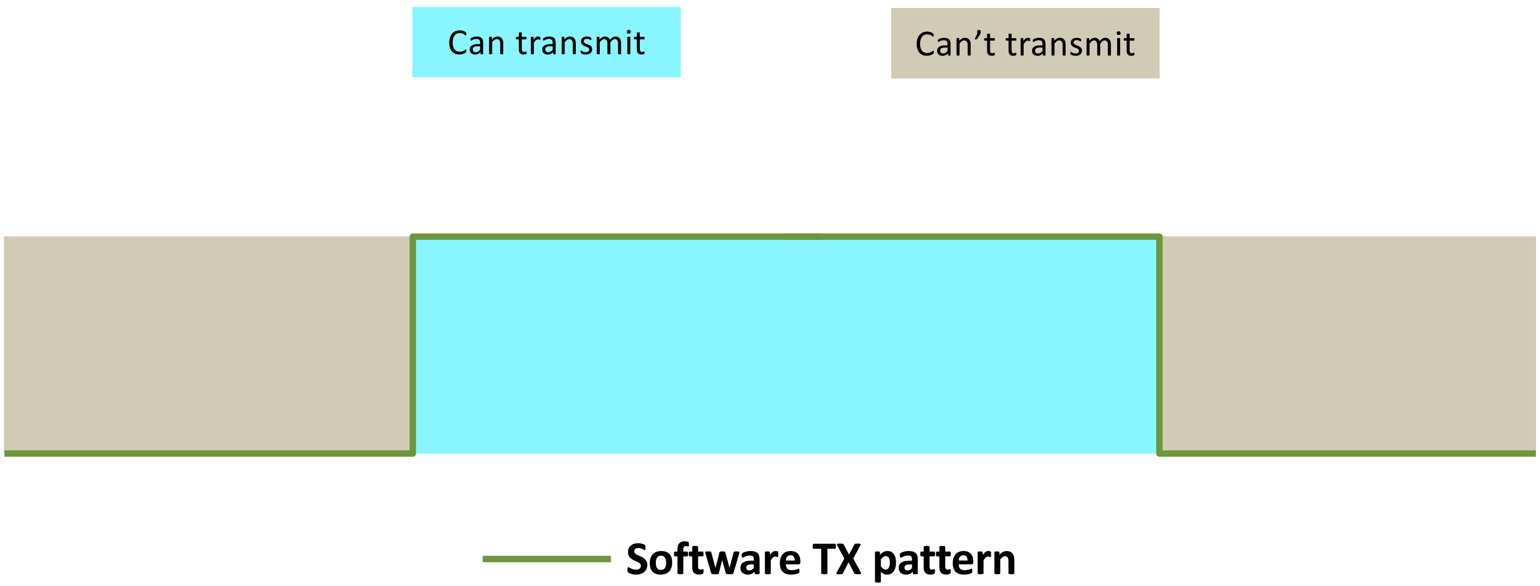
Endhost Variance Annimation

Link to End Host Variance Annimation

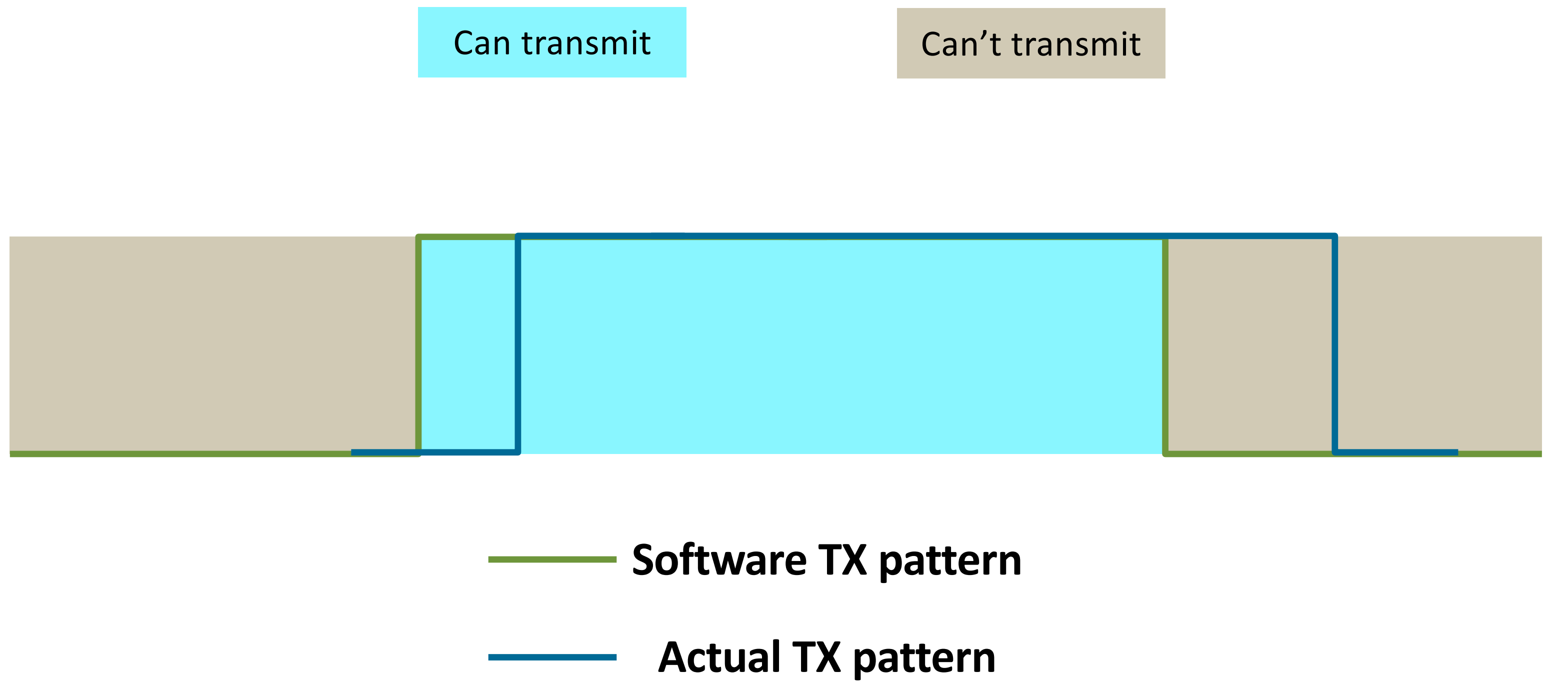
Handling variance in TDMA



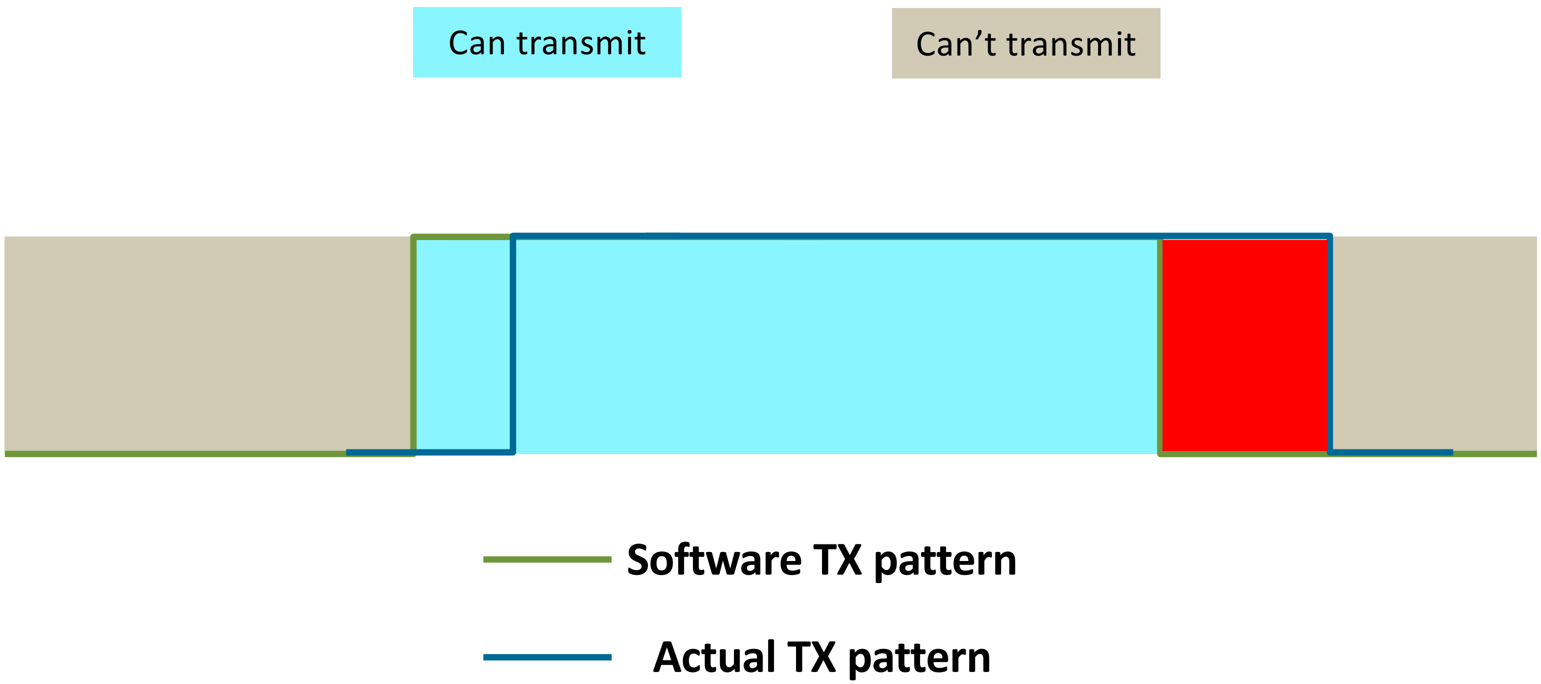
Handling variance in TDMA



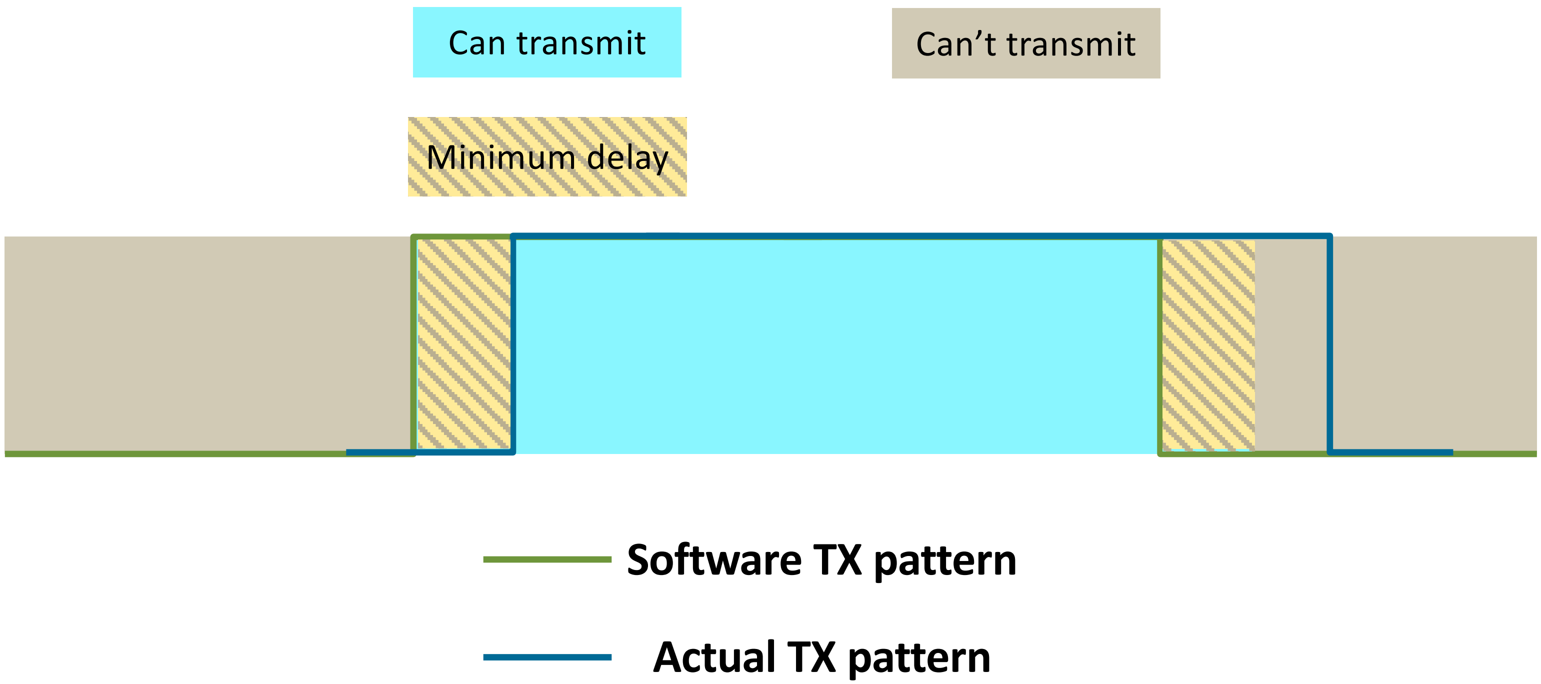
Handling variance in TDMA



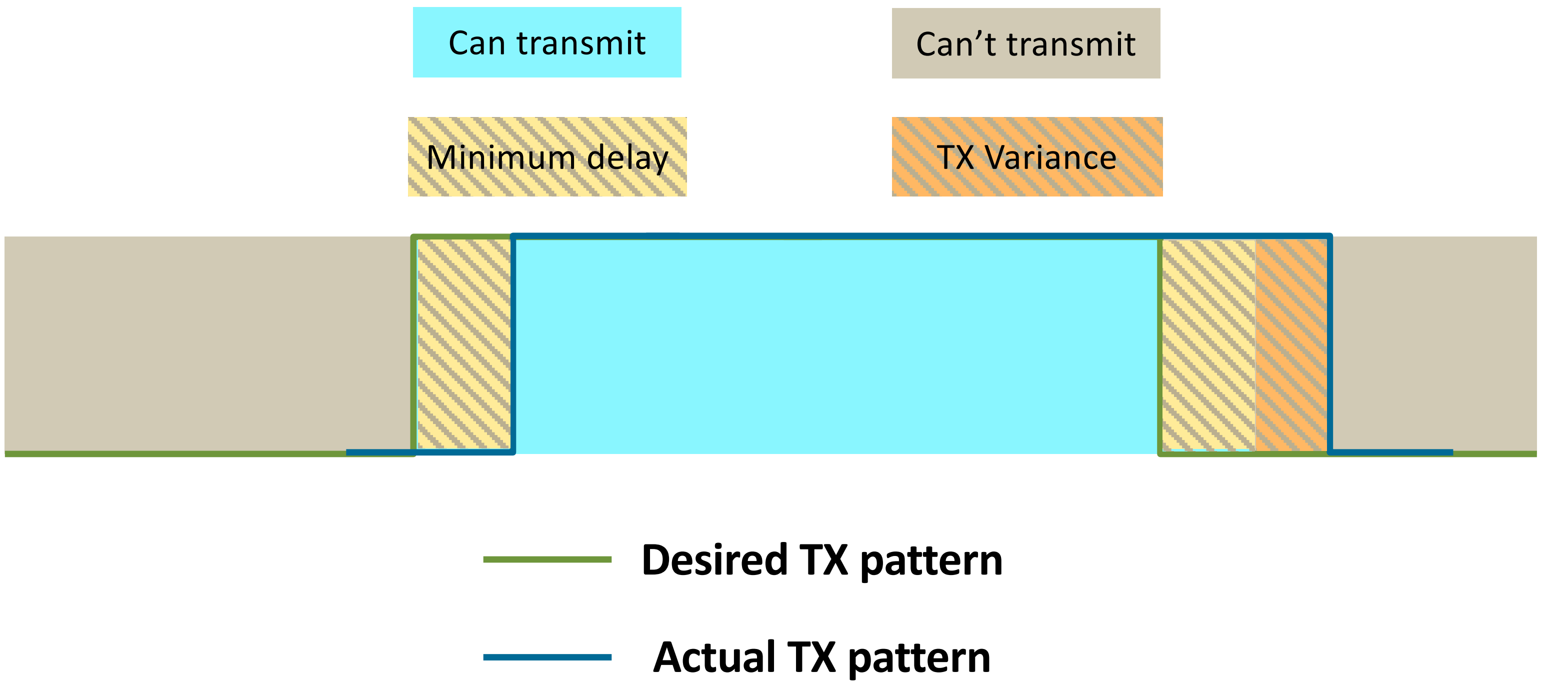
Handling variance in TDMA



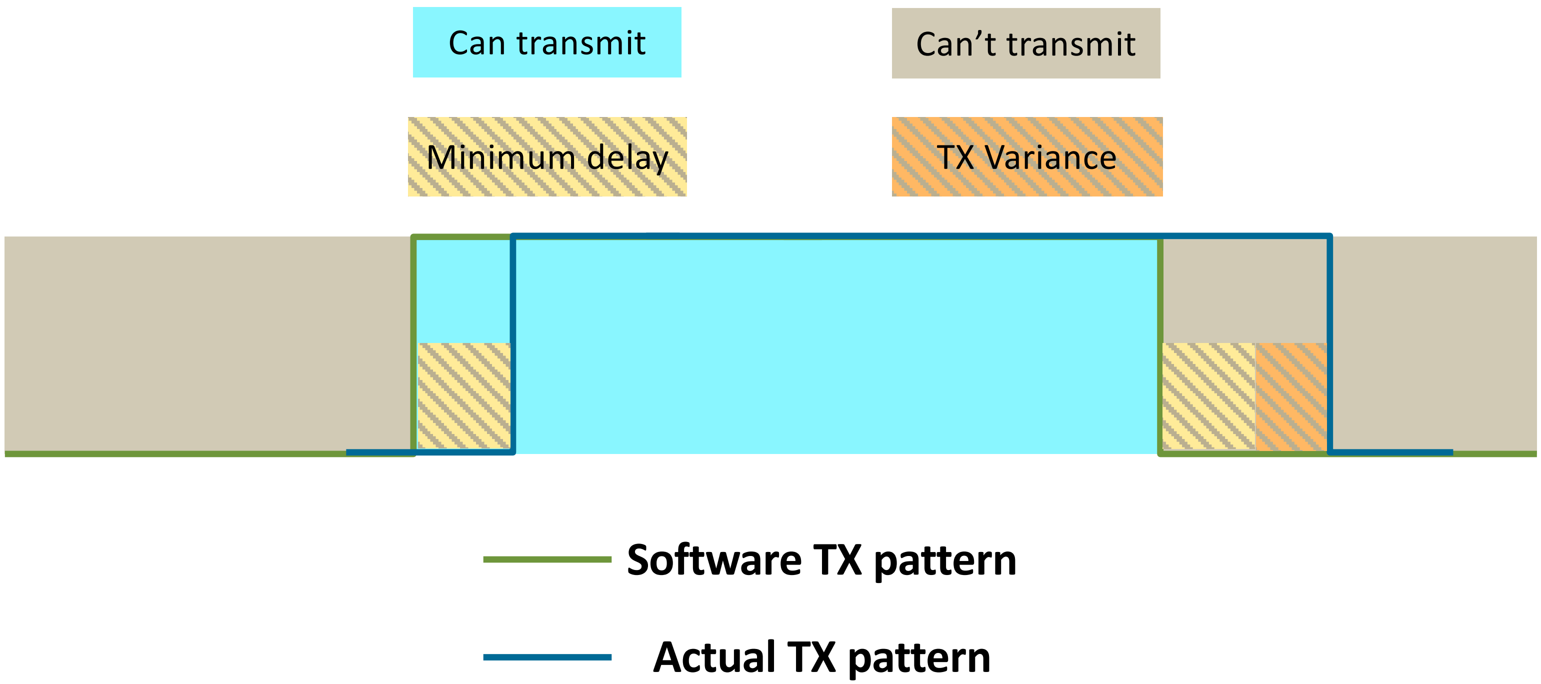
Handling variance in TDMA



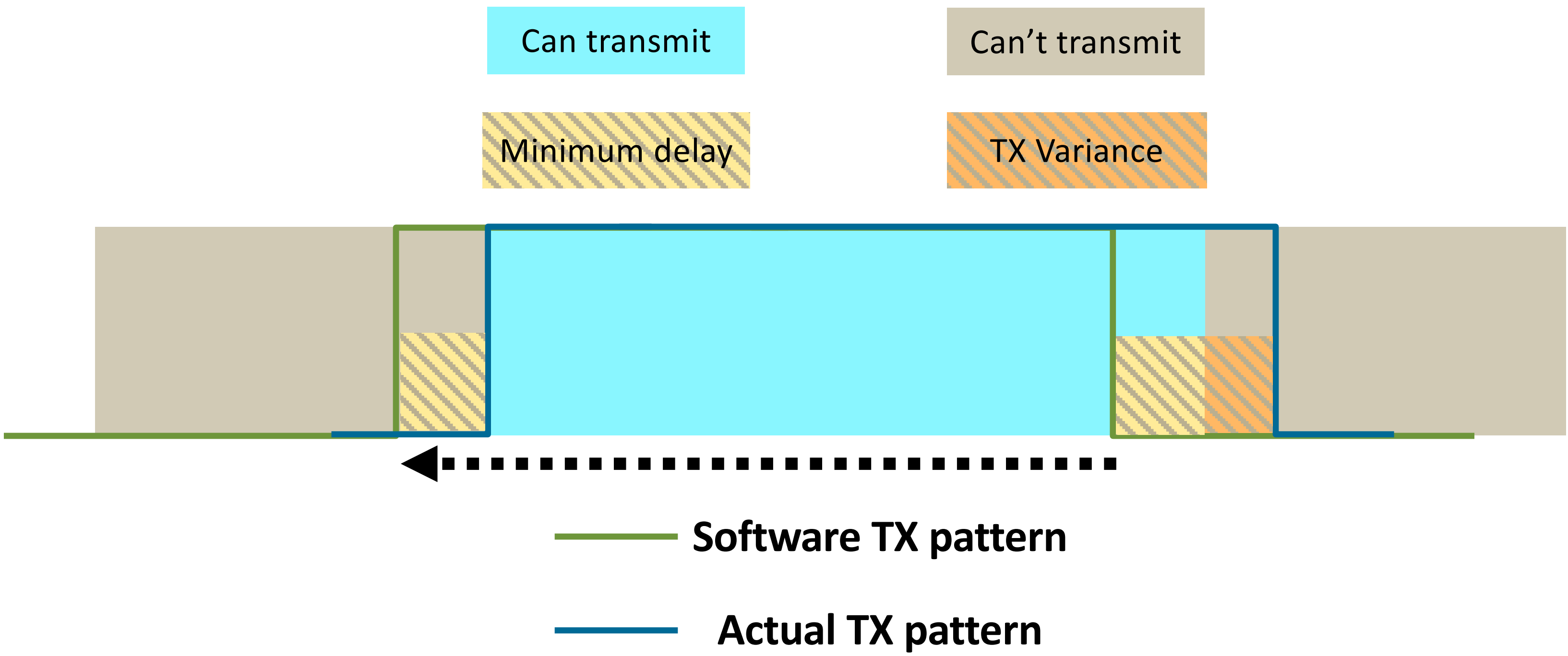
Handling variance in TDMA



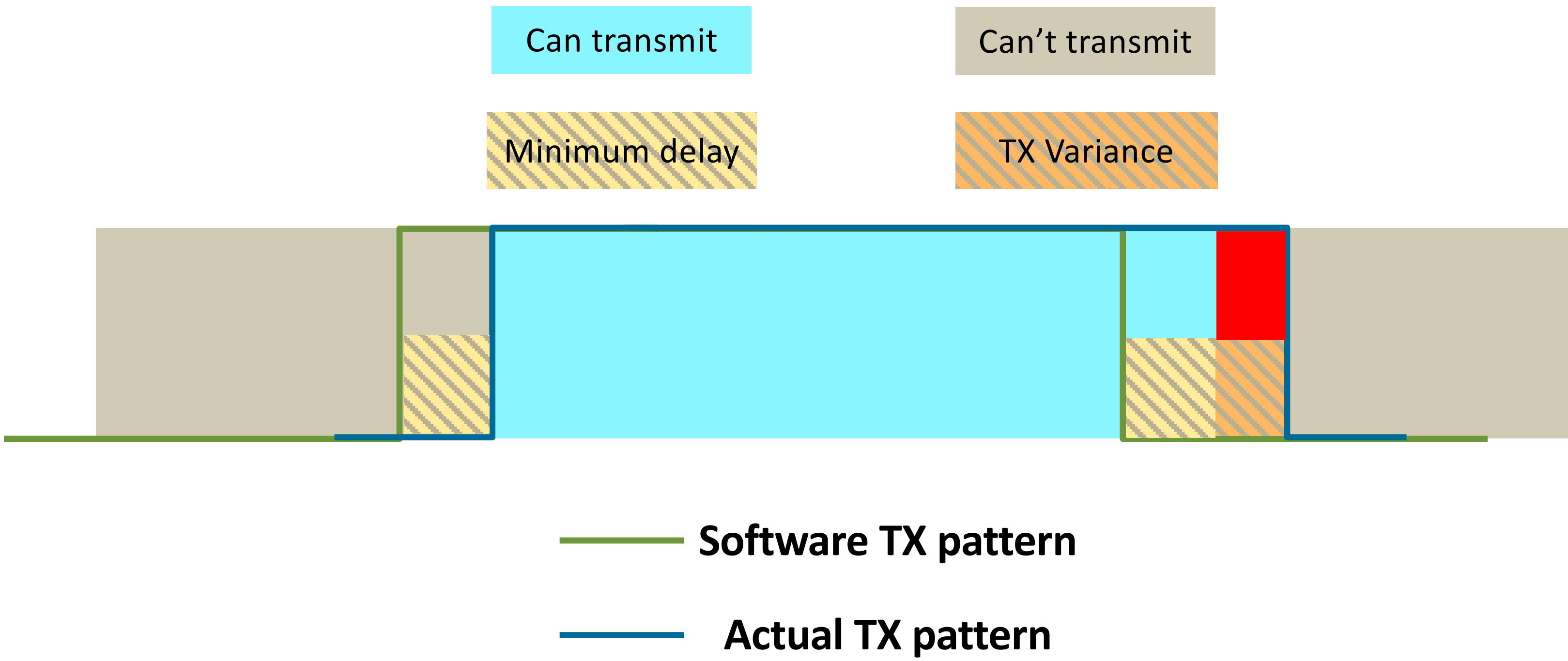
Handling variance in TDMA



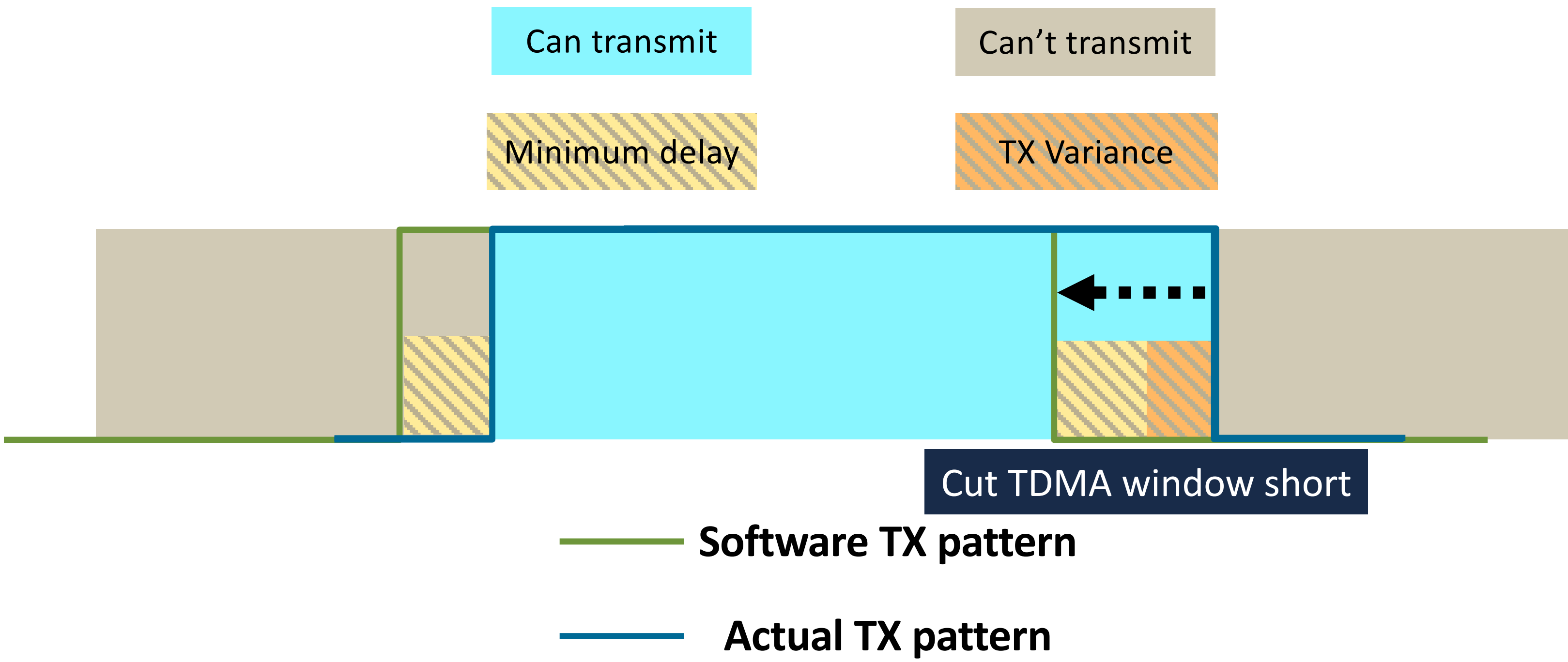
Handling variance in TDMA



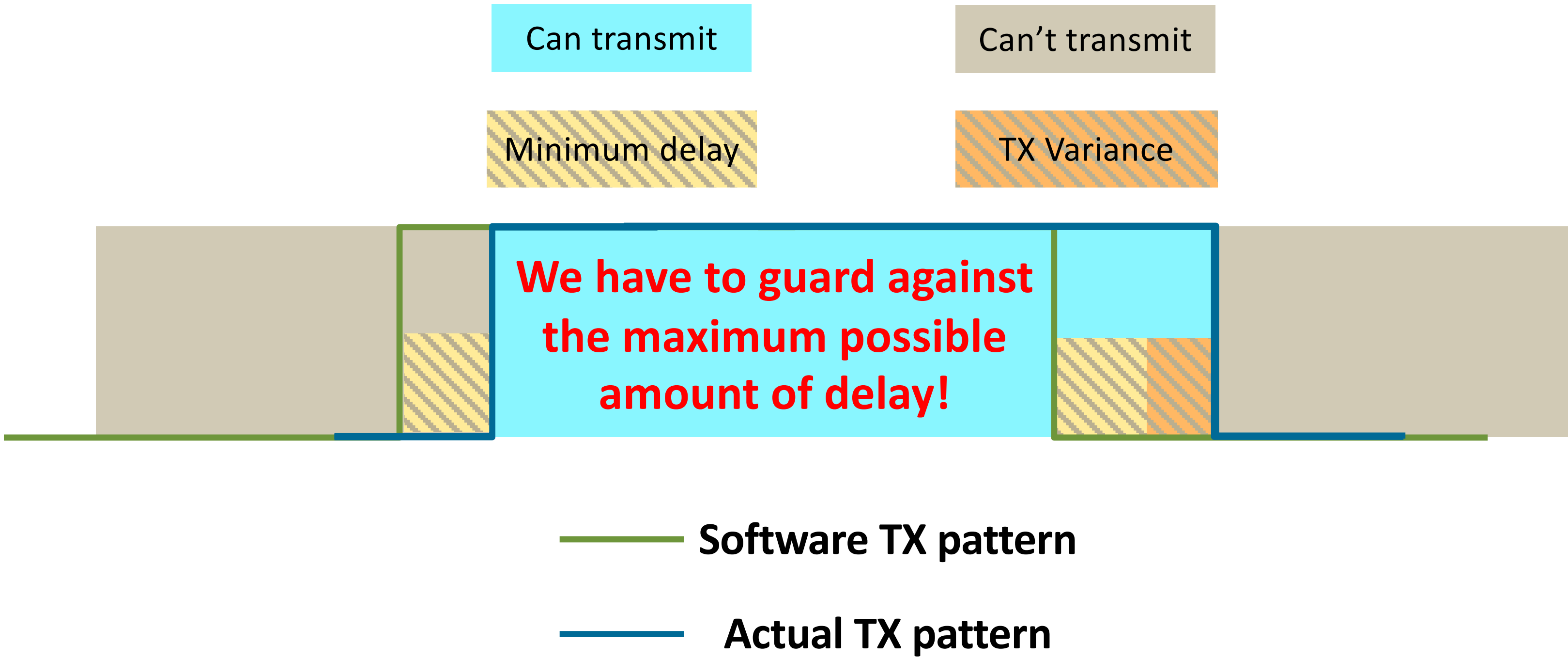
Handling variance in TDMA



Handling variance in TDMA



Handling variance in TDMA



How can we implement TDMA flow control at endhosts to provide the best performance to optical datacenter networks?

How can we implement TDMA flow control at endhosts to provide the best performance to optical datacenter networks?

1. Maximizing the raw bandwidth sent over the networks when enforcing TDMA traffic control

2. By ensuring a sufficiently small minimum delay and cutting down on transmission variance

Talk Outline

Introduction

Circuit-Switched Endhost Networking

- **Kernel module (SelectorNet/RotorNet, SIGCOMM ‘17)**
- Kernel-bypass

Conclusion

Prior circuit-switched networks

Primary shortcomings:

- Long cycle times
- Requiring a secondary packet-switched network
- Difficult to scale

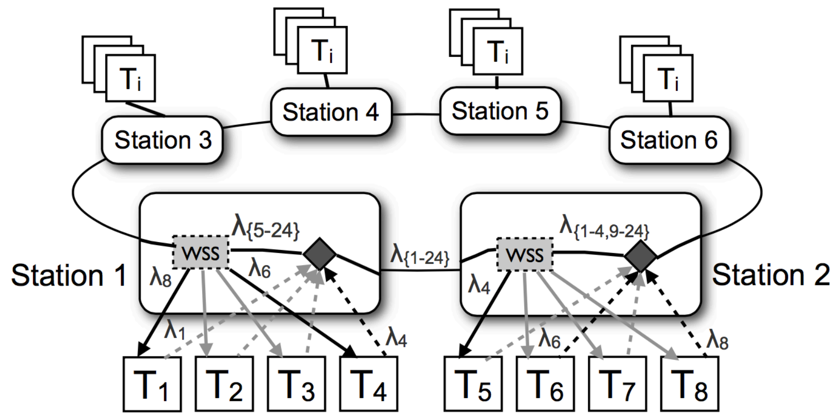
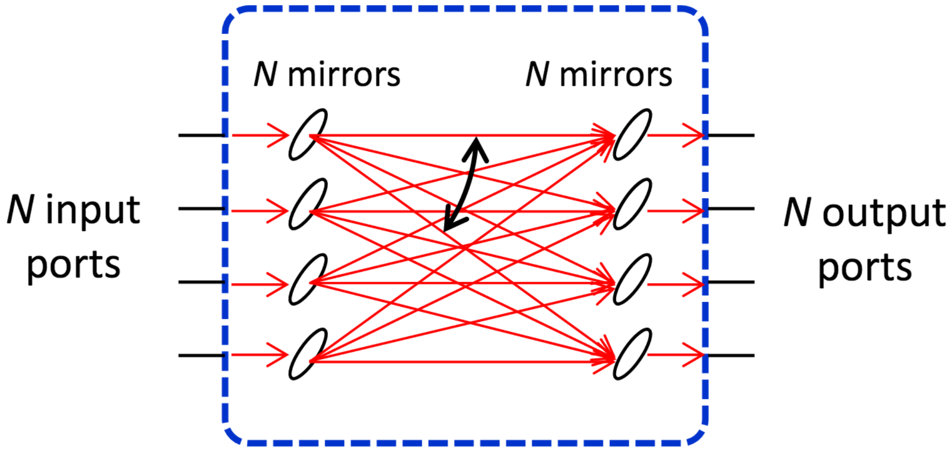


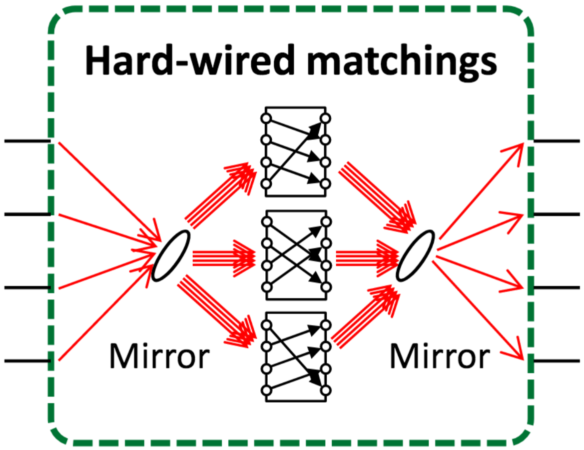
Image: G. Porter et al, Integrating Microsecond Circuit Switching into the Data Center, SIGCOMM ‘13.

Introducing RotorNet

Optical Crossbar:



Optical Rotor switch:



Introducing RotorNet

Low number of matchings

Matchings are hard-wired

Cycle time & duty cycle are constant

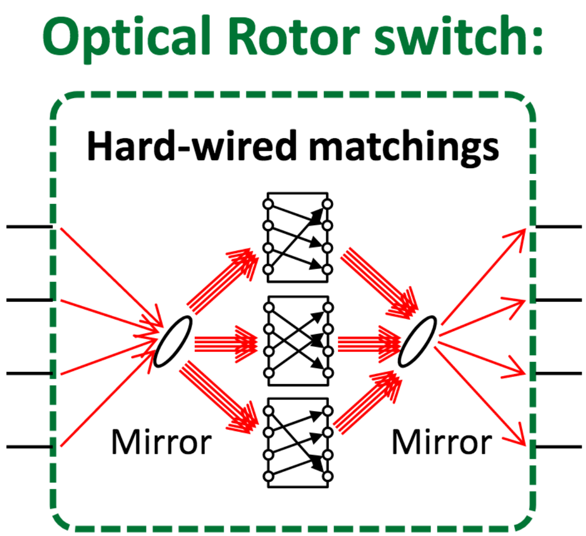


Image: W. Mellette et al, RotorNet, SIGCOMM '17.

Introducing RotorNet

Low number of matchings
- Cycle time should be short
  - Low-variance TDMA is paramount

Matchings are hard-wired
- Less network state

Cycle time & duty cycle are constant
- Endhosts can be preprogrammed

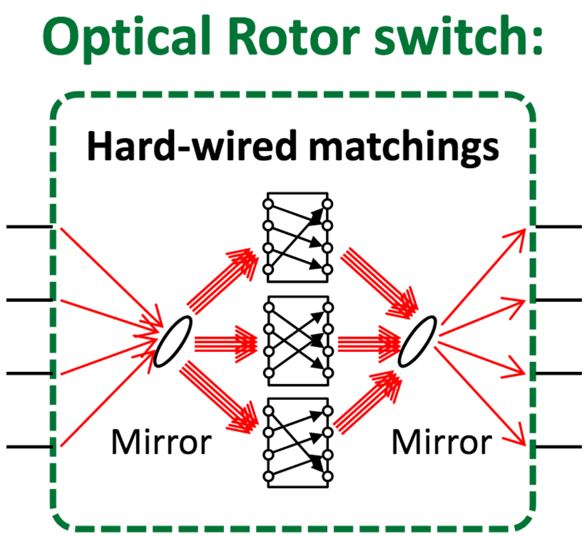


Image: W. Mellette et al, RotorNet, SIGCOMM ‘17.

TDMA Controller: Fastpass

TDMA-based networking for packet-switched networks

- Could we use this for RotorNet?

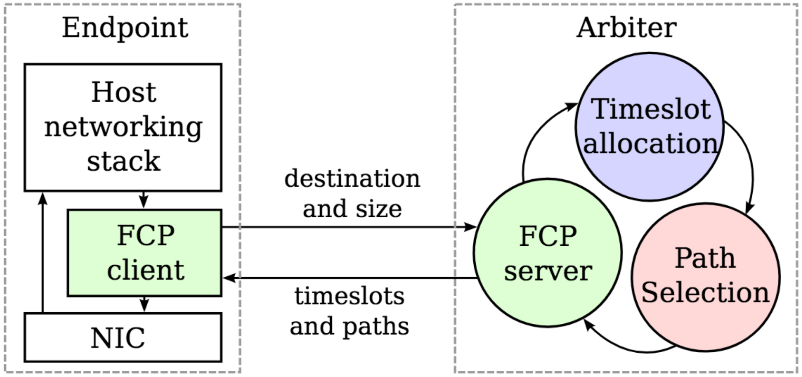


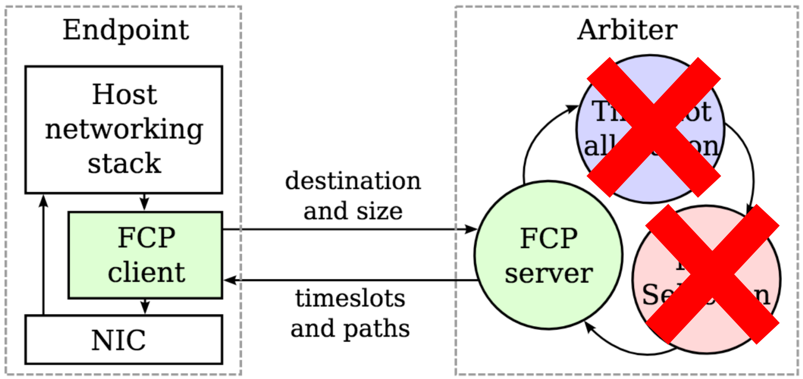
Image: J. Perry et al, Fastpass, SIGCOMM '14.

TDMA Controller: Fastpass

TDMA-based networking for packet-switched networks

- Could we use this for RotorNet?

Overcomplex controller



TDMA Controller: Fastpass

TDMA-based networking for packet-switched networks

- Could we use this for RotorNet?

Overcomplex controller

Low TDMA precision at endhost

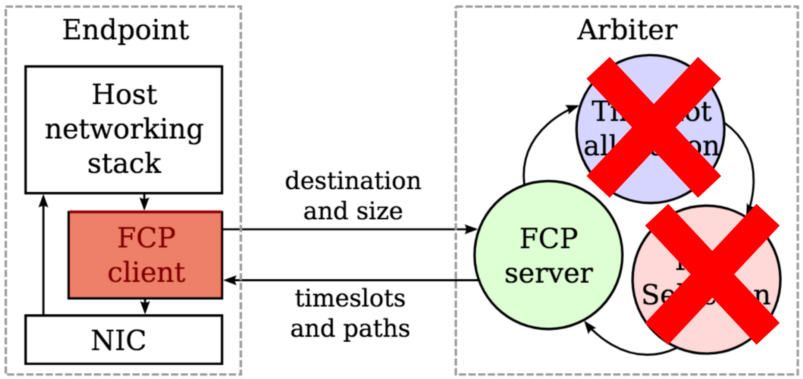


Image: J. Perry et al, Fastpass, SIGCOMM ‘14.

RotorNet & VLB Forwarding

RotorNet switches do not connect all hosts together

Endhosts forward traffic through other hosts

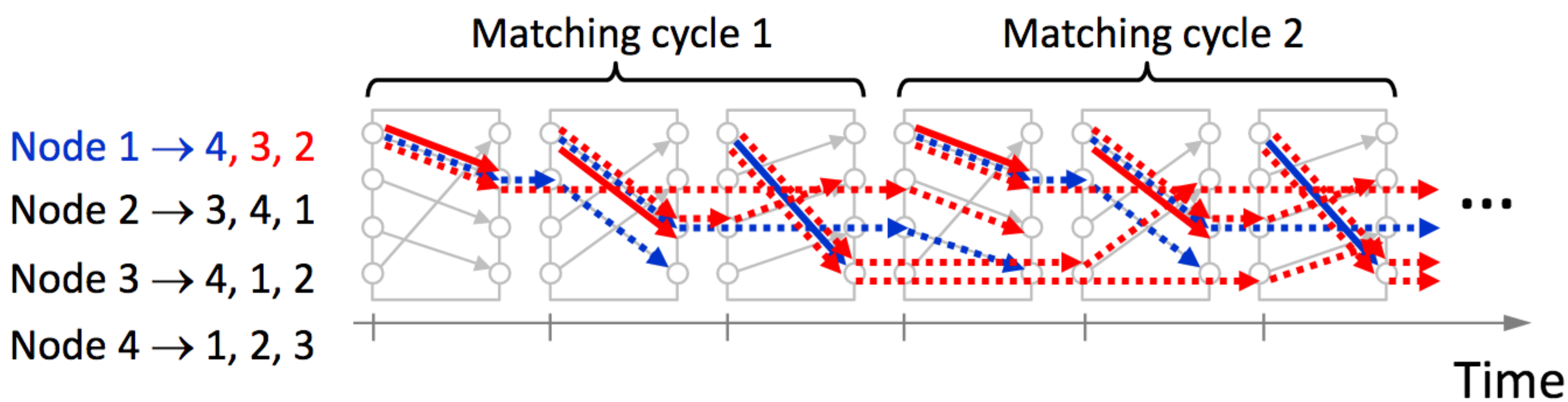
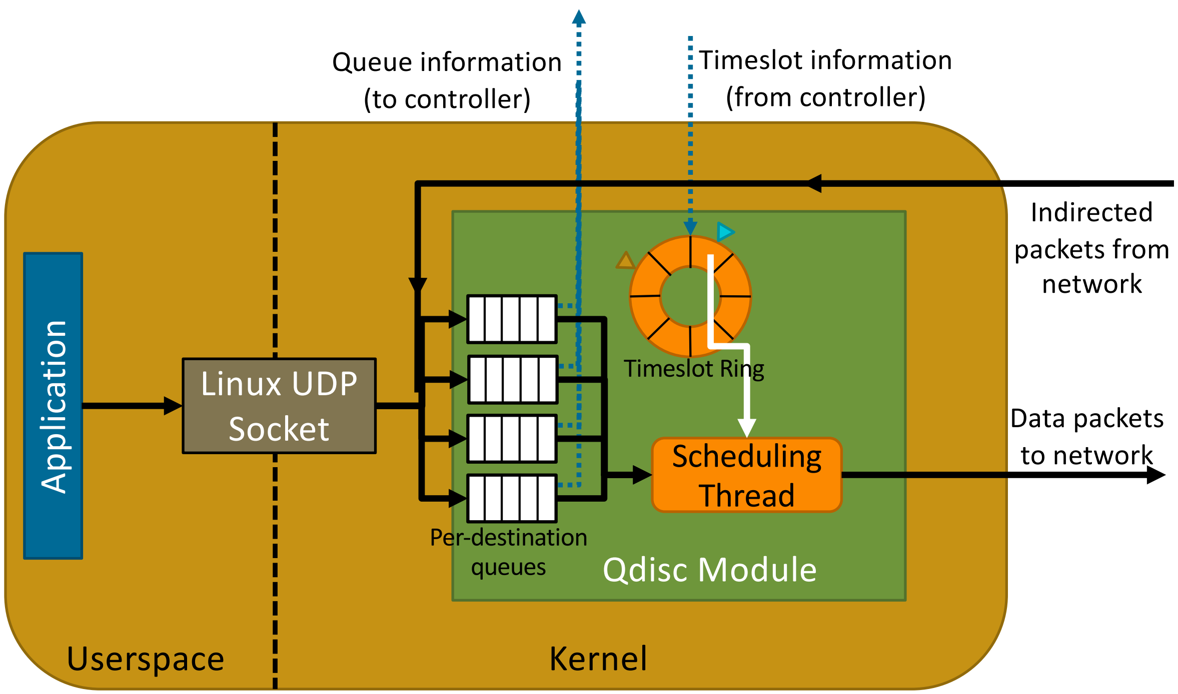


Image: W. Mellette et al, RotorNet, SIGCOMM '17.

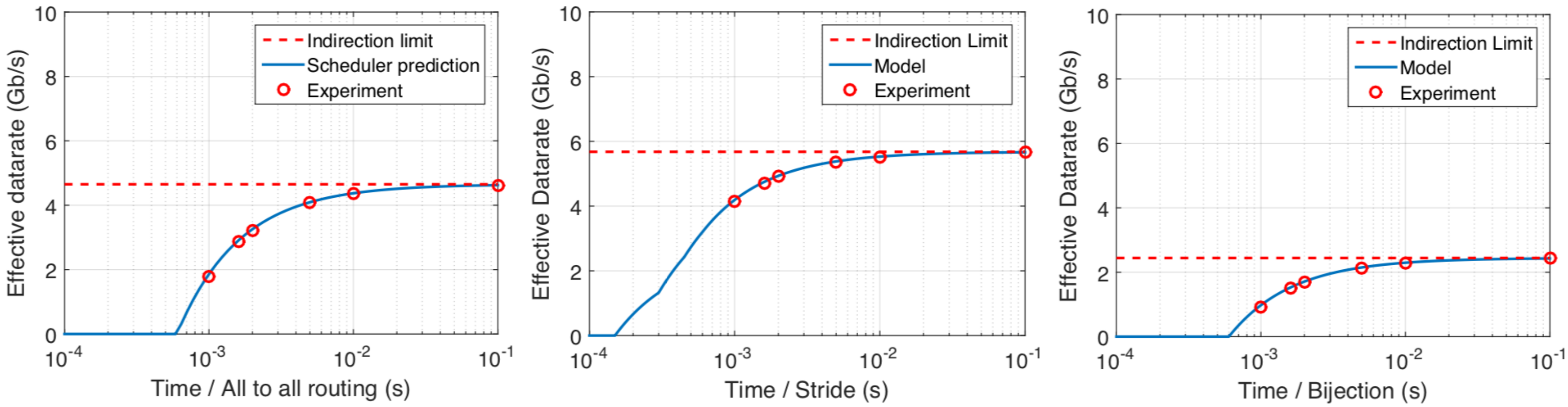
TDMA Queuing discipline (Qdisc)



TDMA Qdisc results

Worked with somewhat long (millisecond) cycle times

Sufficient throughput for 10G experiments with significant indirection



(a) An all-to-all workload. (b) A "rolling stride" workload. (c) A time-varying random bijection workload.

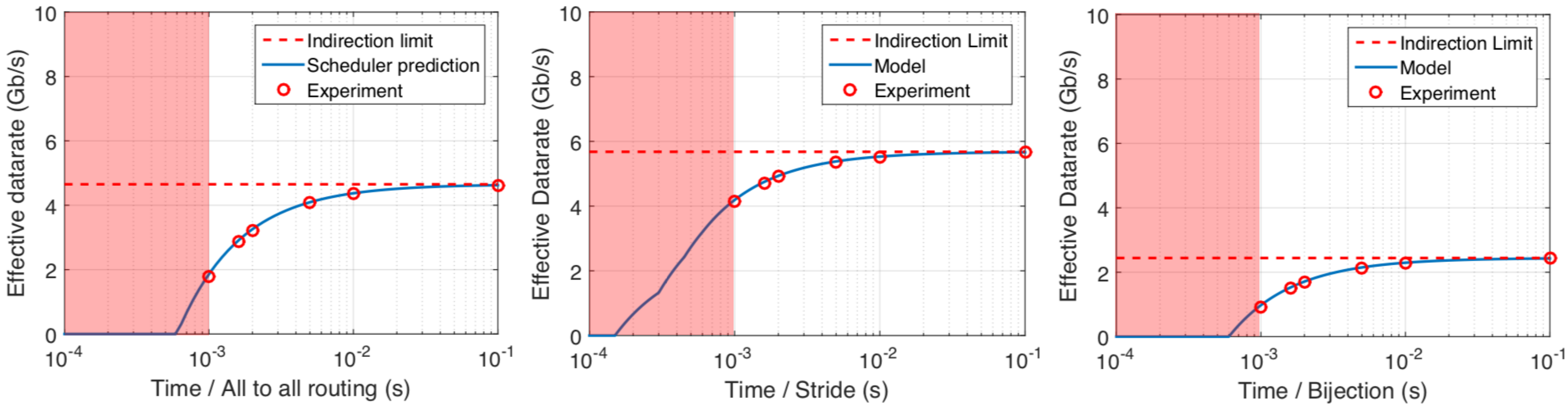
Figure 7: A comparison of the scheduler-predicted vs. experimentally observed throughputs for three workloads.

Image: W. Mellette et al, SelectorNet, unpublished.

TDMA Qdisc results

Worked with somewhat long (millisecond) cycle times

Sufficient throughput for 10G experiments with significant indirection

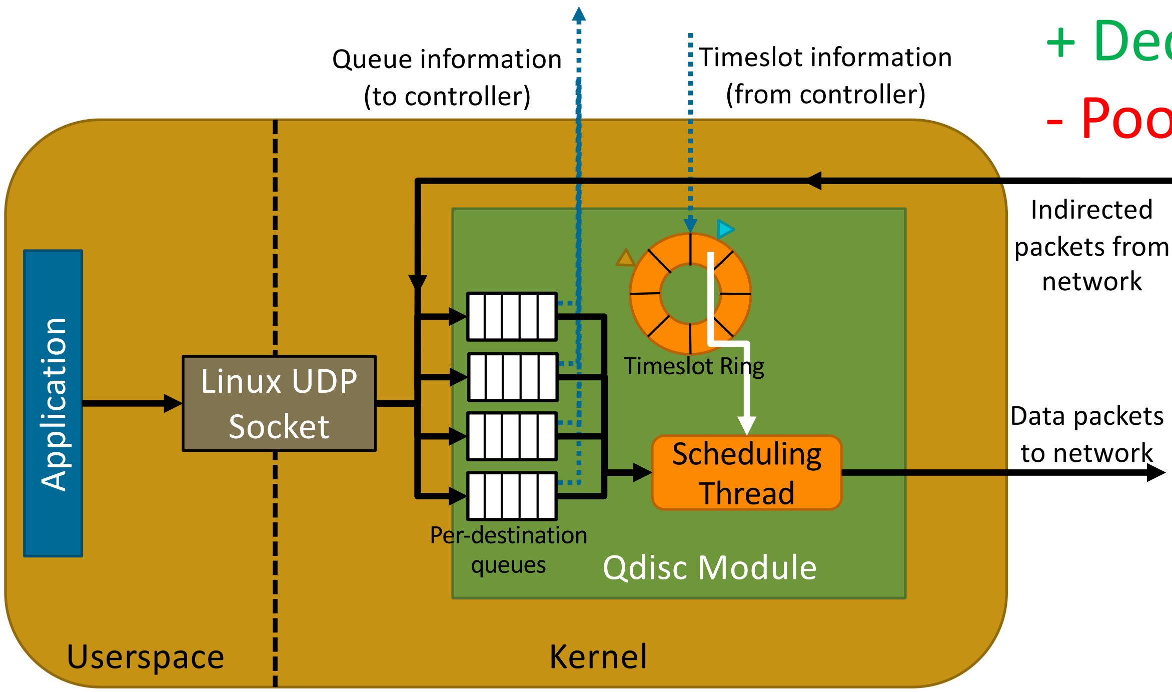


(a) An all-to-all workload. (b) A "rolling stride" workload. (c) A time-varying random bijection workload.

Figure 7: A comparison of the scheduler-predicted vs. experimentally observed throughputs for three workloads.

Image: W. Mellette et al, SelectorNet, unpublished.

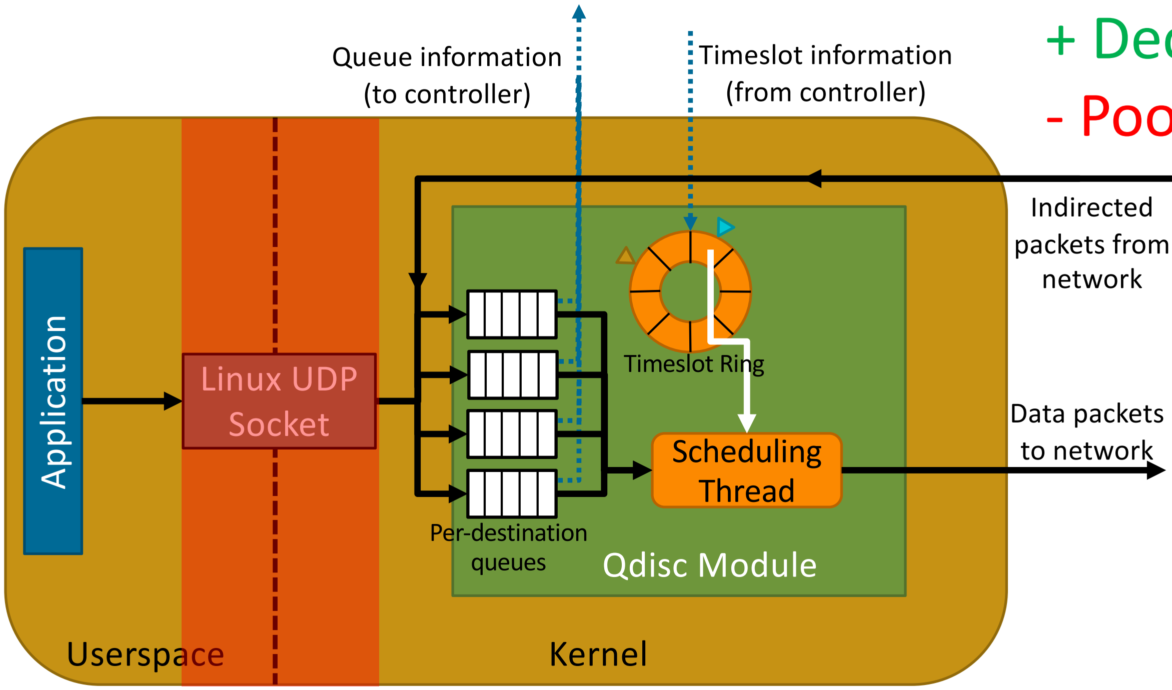
TDMA Queuing discipline (Qdisc)



+ Decent TDMA accuracy

- Poor data throughput

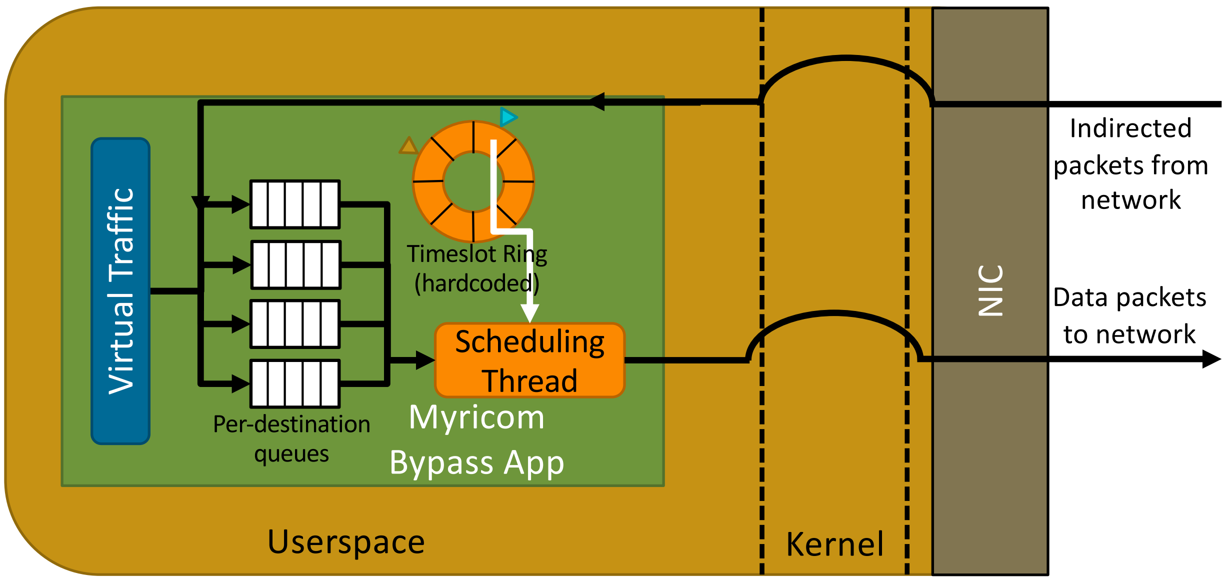
TDMA Queuing discipline (Qdisc)



+ Decent TDMA accuracy

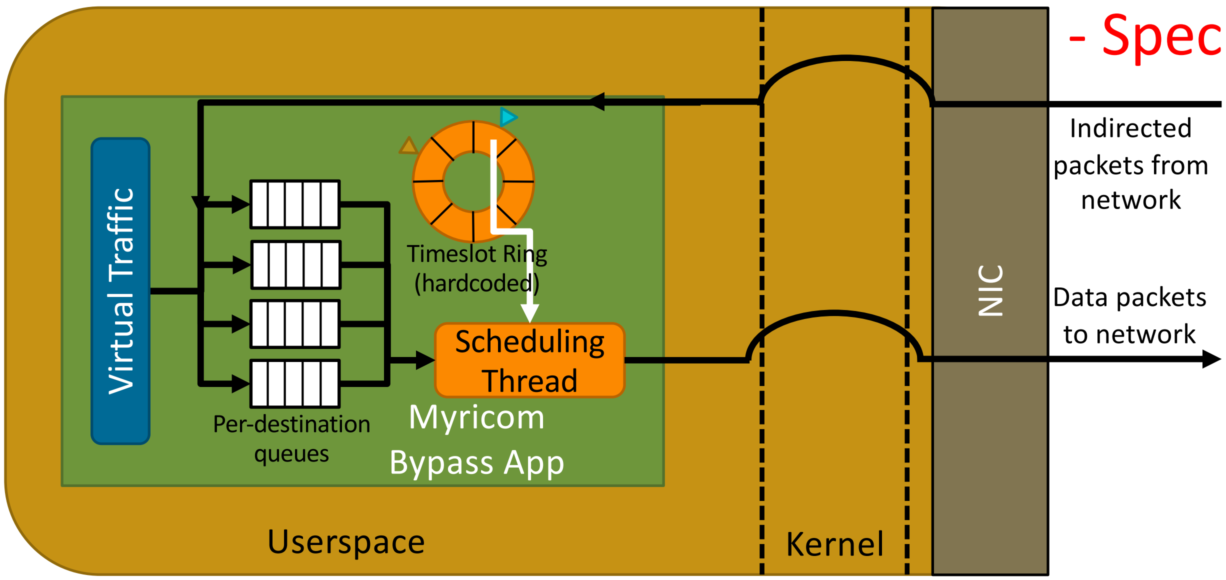
- Poor data throughput

Kernel-bypass TDMA (Myricom)



Kernel-bypass TDMA (Myricom)

+ Saturated Network Links
- Virtualized Traffic
- Specific to one NIC



RotorNet Simulation Results

Kernel-bypass method fit RotorNet model

Better throughput

Slightly better TDMA accuracy

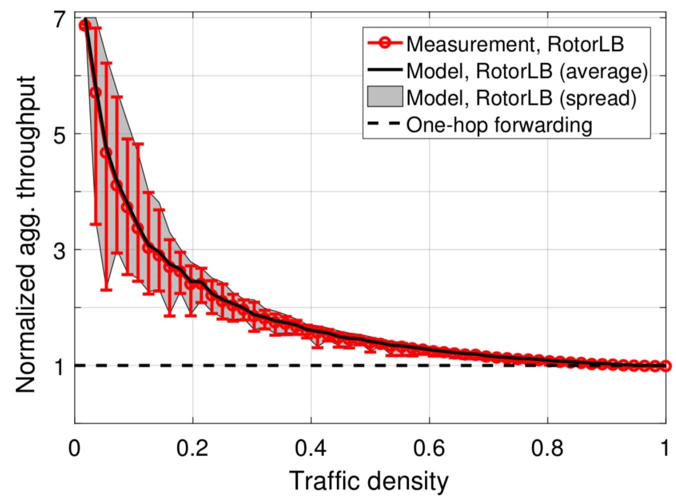


Figure 7: Measured and modeled throughput under RotorLB relative to that using one-hop forwarding. Circles represent the average througput over 32 random traffic patterns; error bars show the maximum and mimimum.

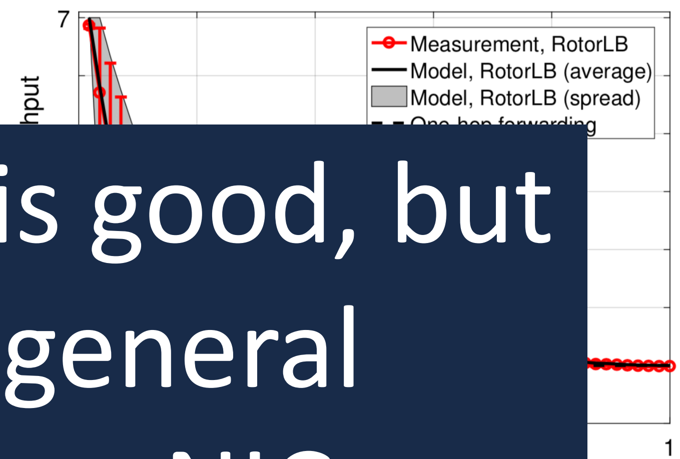
Image: W. Mellette et al, RotorNet, SIGCOMM '17.

RotorNet Simulation Results

Kernel-bypass method fit RotorNet model

Better

Slightl



RotorLB

represent

the average througput over 32 random traffic patterns; error bars show the maximum and mimimum.

Kernel-bypass is good, but we need a general API/software NIC

Talk Outline

Introduction

Circuit-Switched Endhost Networking

- Kernel module
- **Kernel-bypass (BESS Analysis, ANCS '18)**

Conclusion

[1]: University of California, Berkeley, '15

BESS[1], a kernel-bypass Software NIC

Software NIC using DPDK

DPDK works with many NICs

Allows many forms of flow control

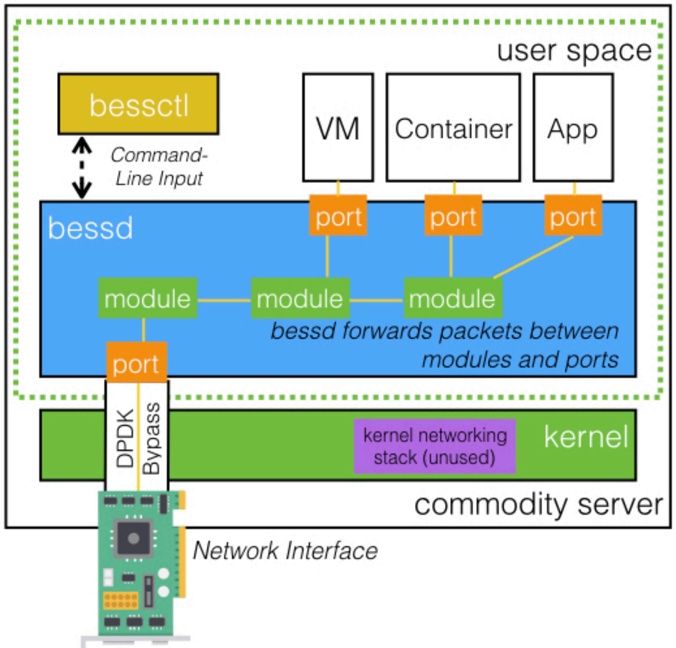


Image: S. Han et al, BESS Overview, GitHub.

BESS Module Chains

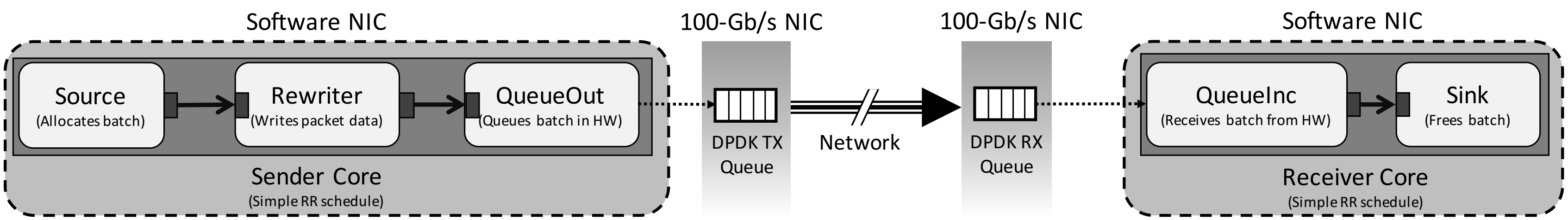


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

BESS Module Chains

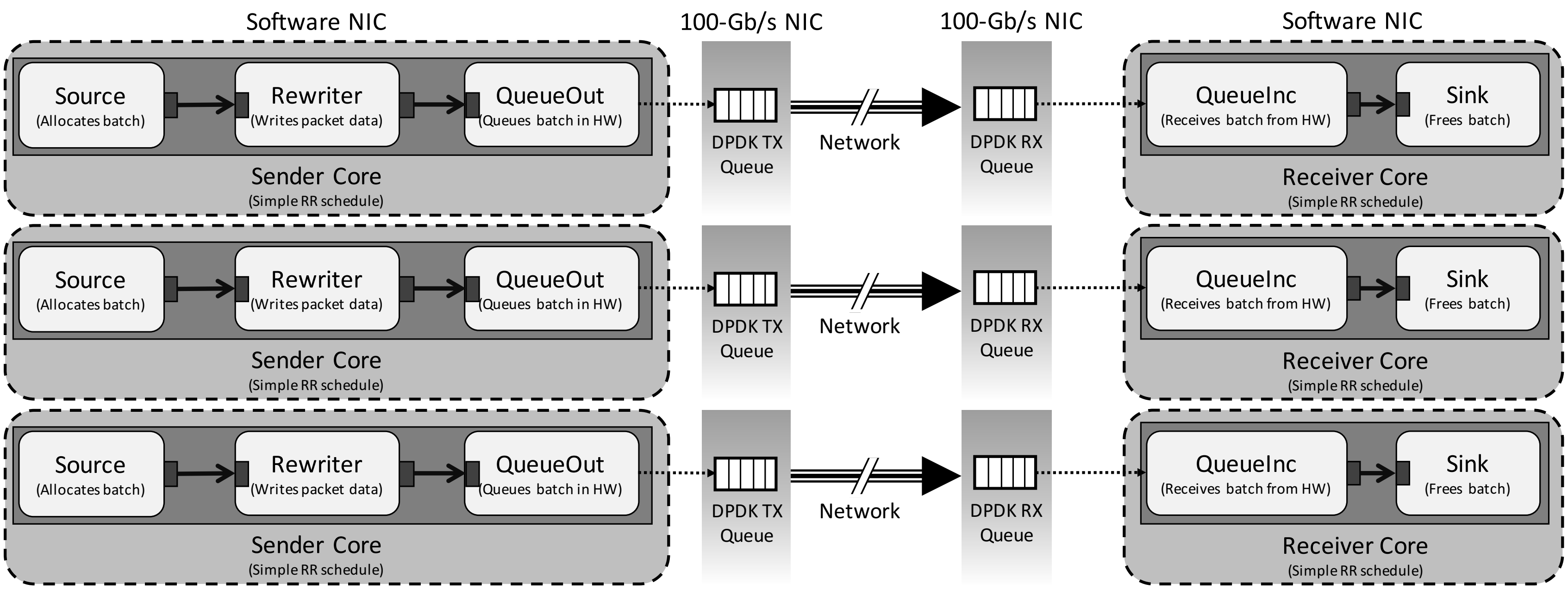
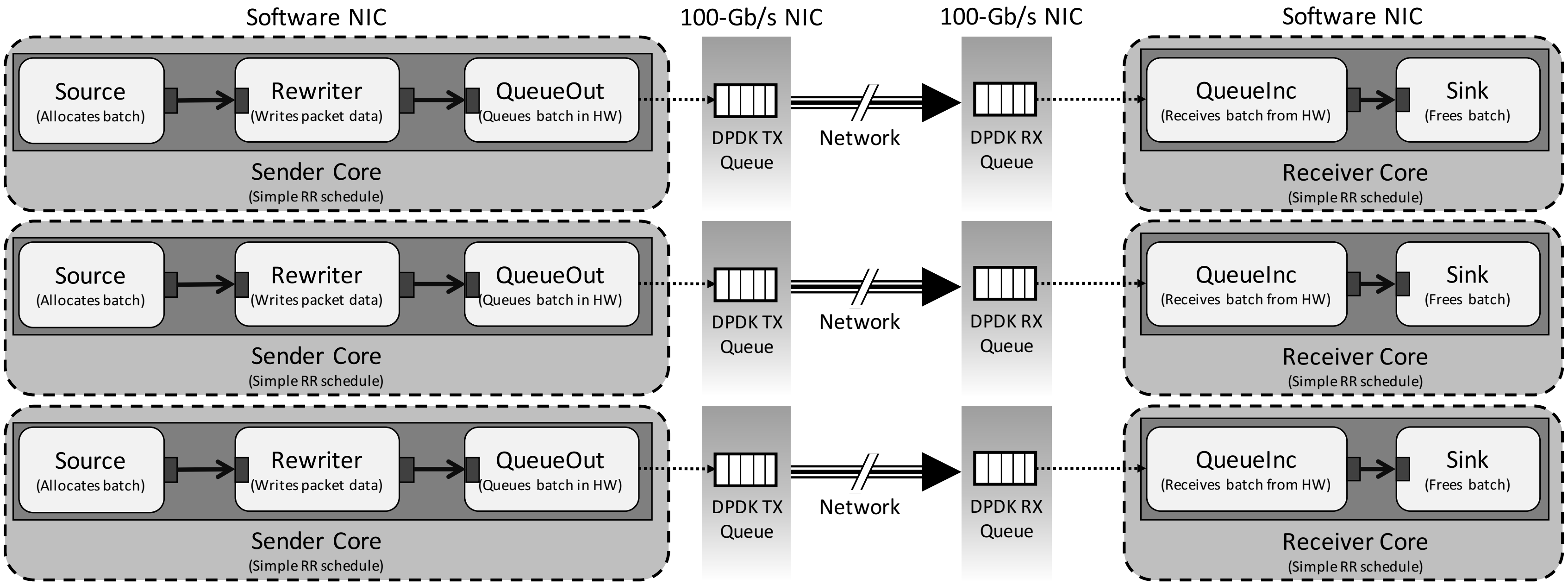


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

BESS Module Chains



Multiple cores used to create independent interfaces

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Can SoftNICs implement TDMA well in optical datacenter networks?

What do we want from SoftNICs?

High throughput

Rate limiting

Flow scheduling

Low processing latency

What do we want from SoftNICs?

High throughput

Rate limiting

Flow scheduling

Low processing latency

SoftNIC throughput

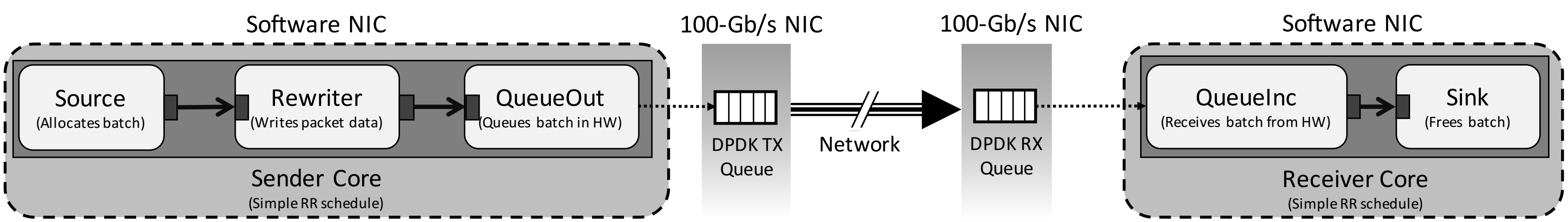


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

SoftNIC throughput

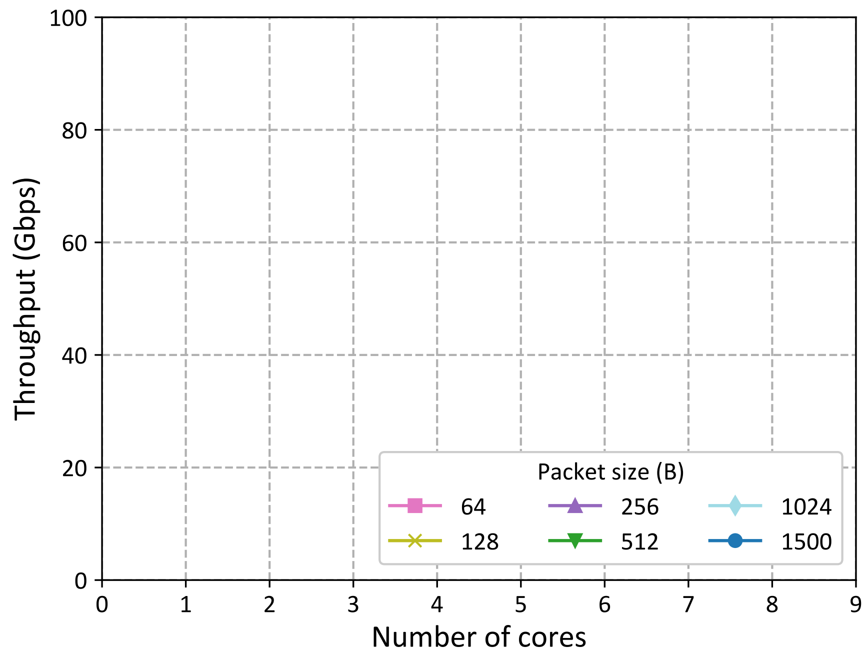


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

SoftNIC throughput

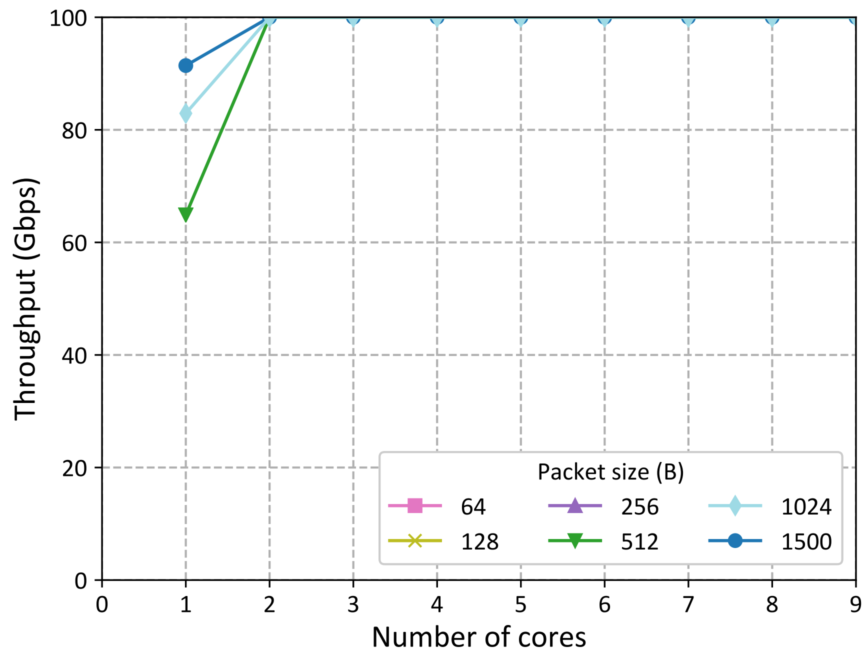


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

SoftNIC throughput

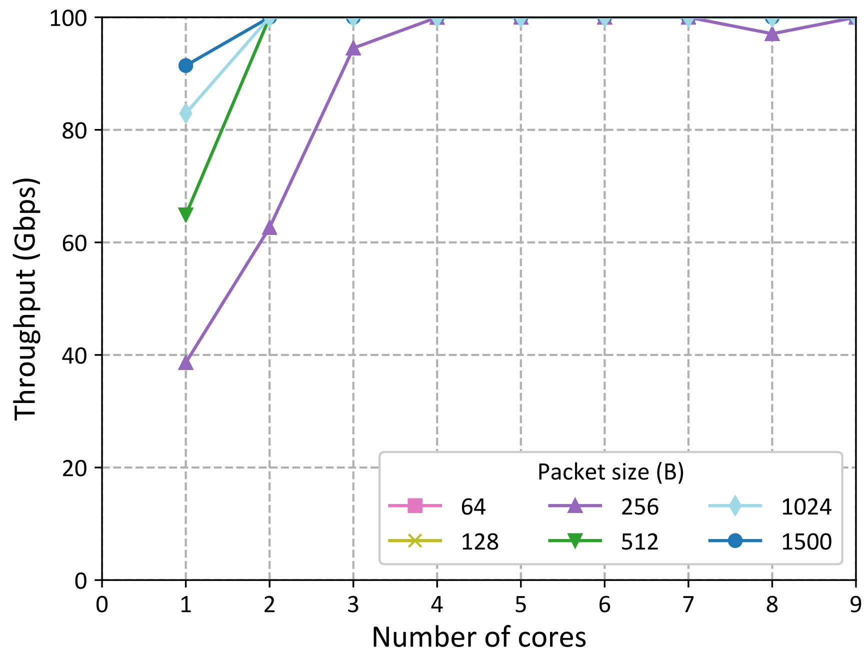


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

SoftNIC throughput

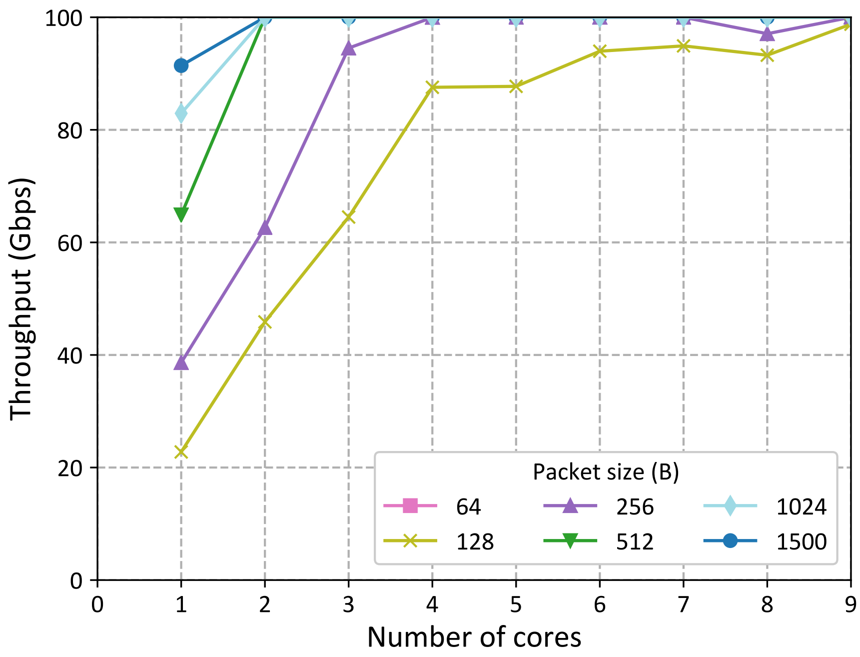


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

SoftNIC throughput

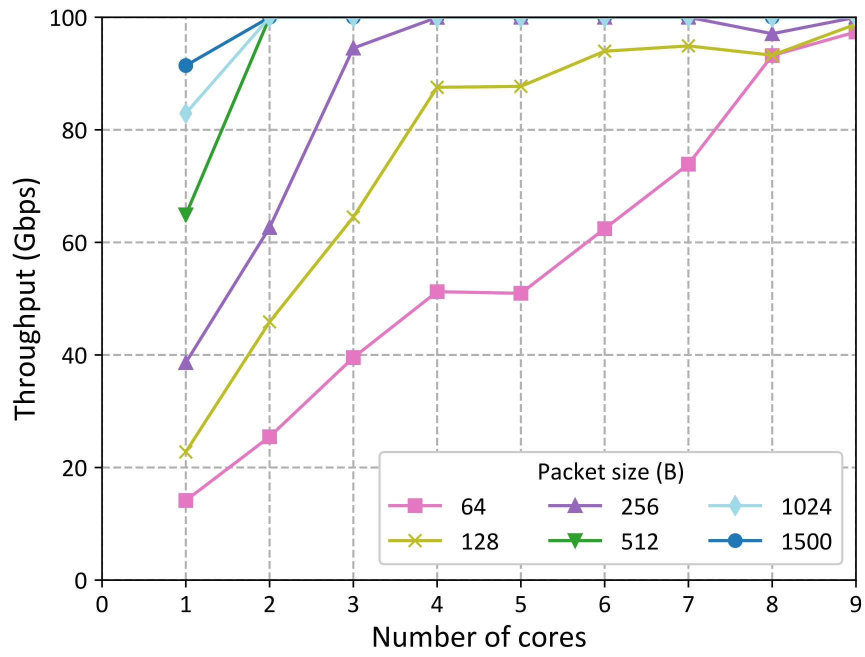
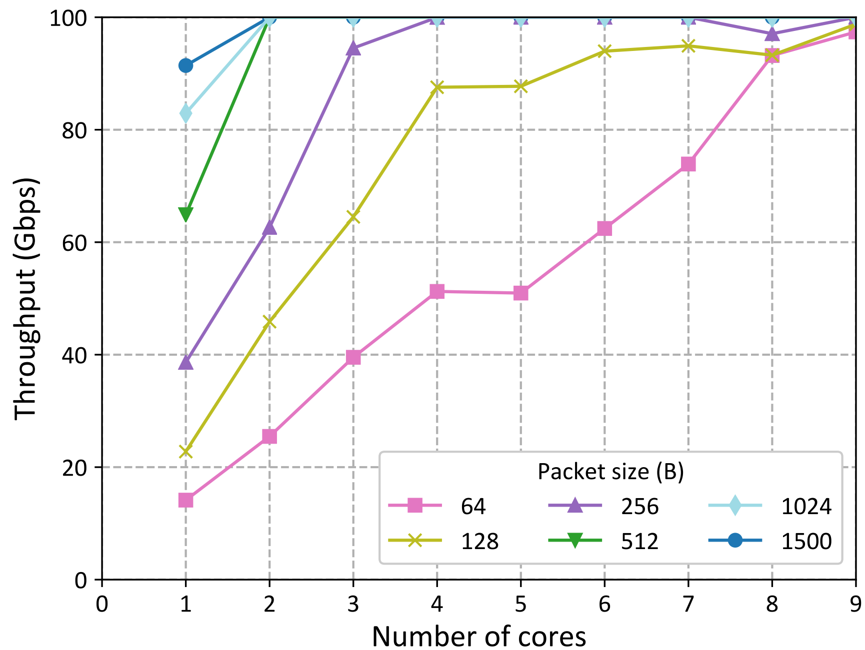


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

SoftNIC throughput



100-Gb/s links requires either big packets or big CPUs

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

What does a SoftNIC need to do?

High throughput

Rate limiting

Flow scheduling

Low processing latency

Rate limiting

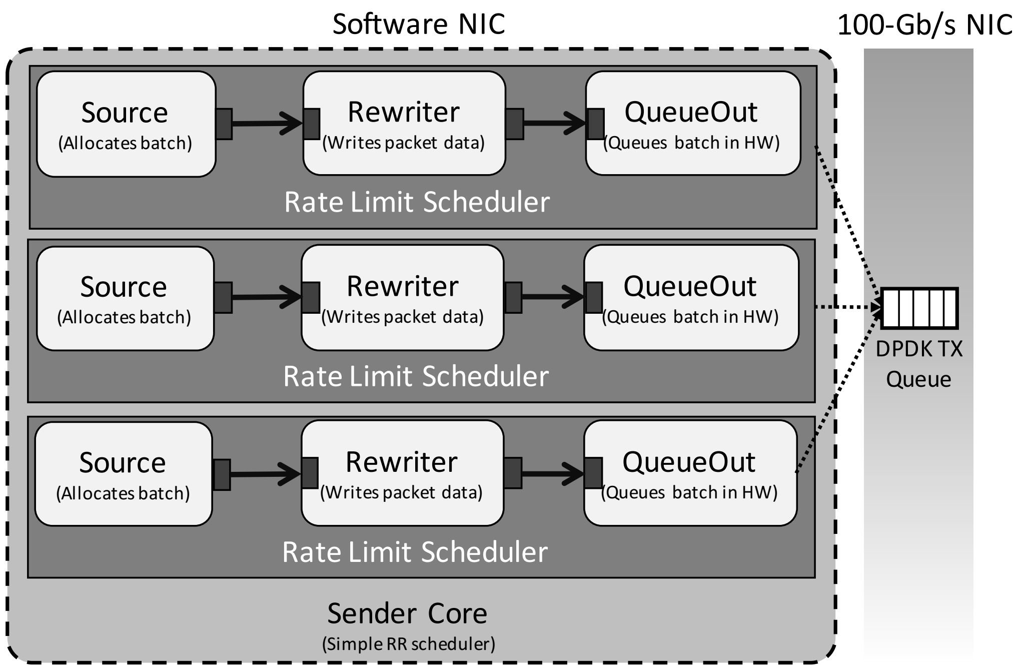


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Rate limiting

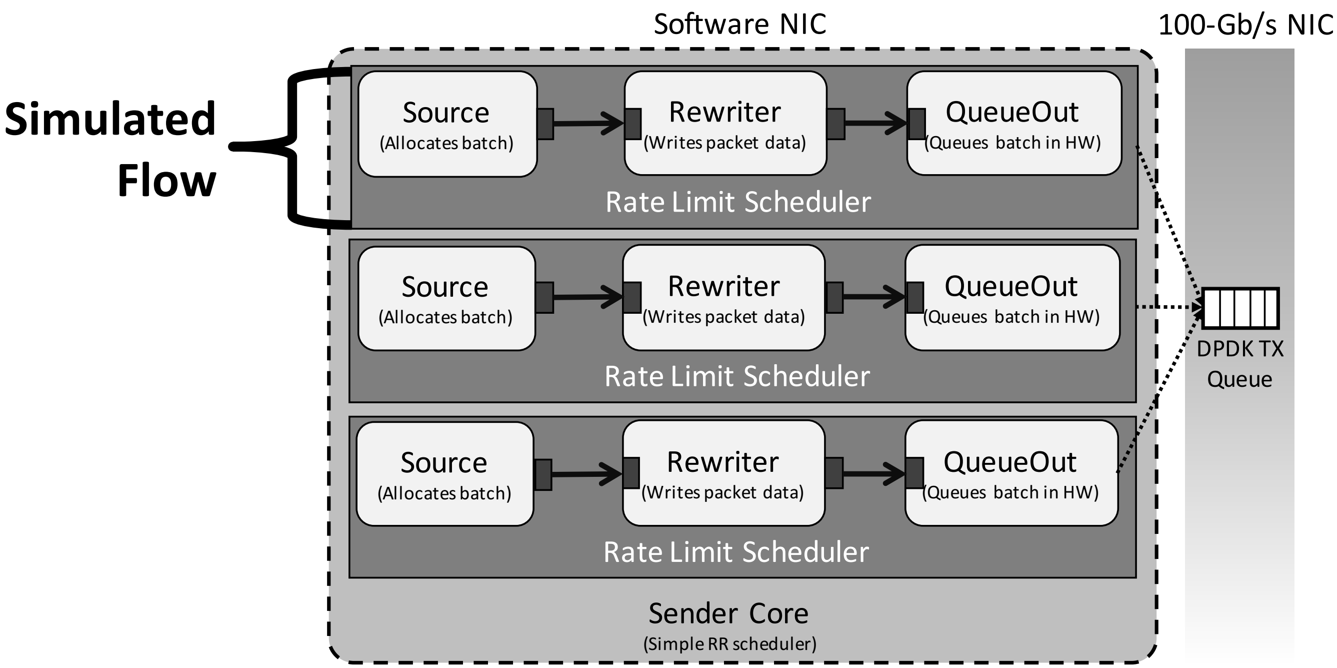


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Rate limiting

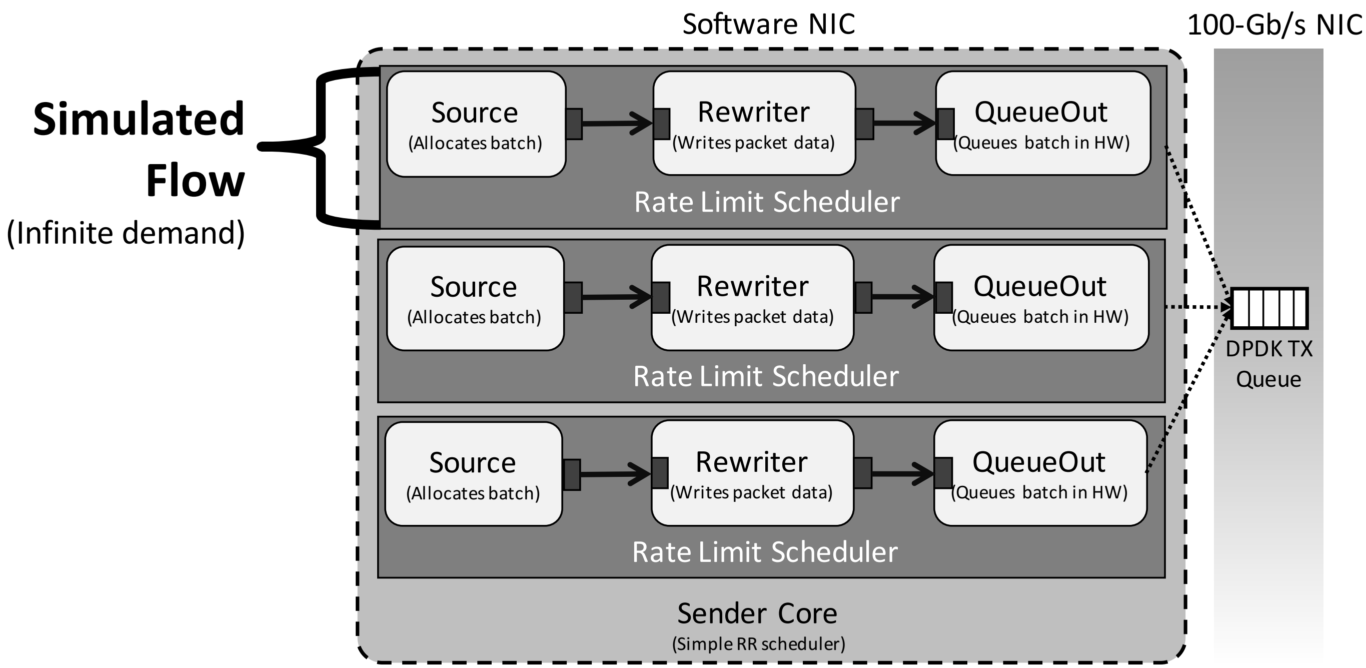
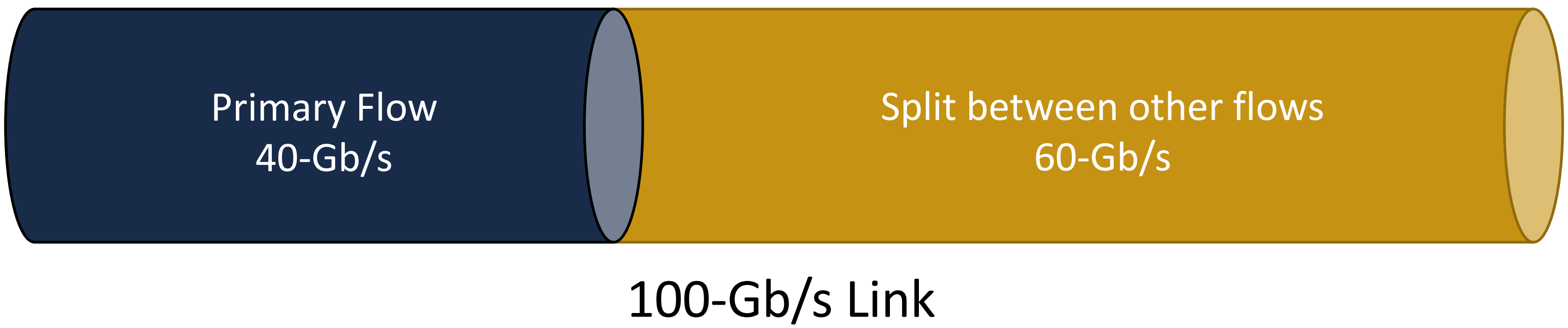


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Bandwidth allocation

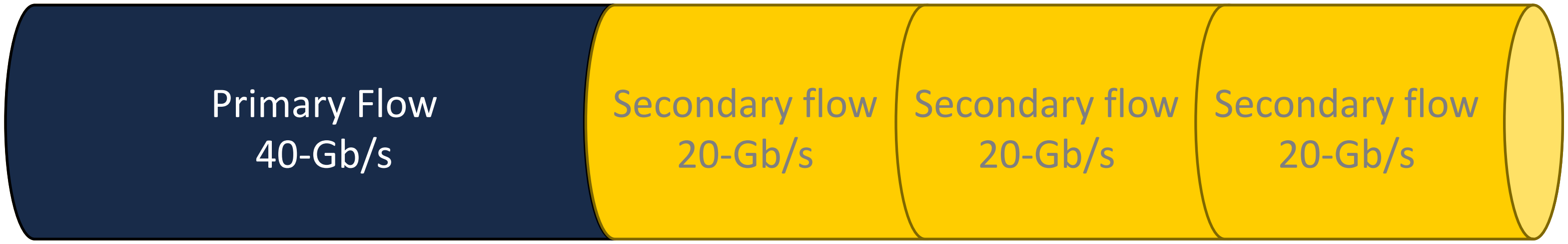
100-Gb/s Link

Bandwidth allocation



Bandwidth allocation

Four flow experiment



100-Gb/s Link

Rate limiting

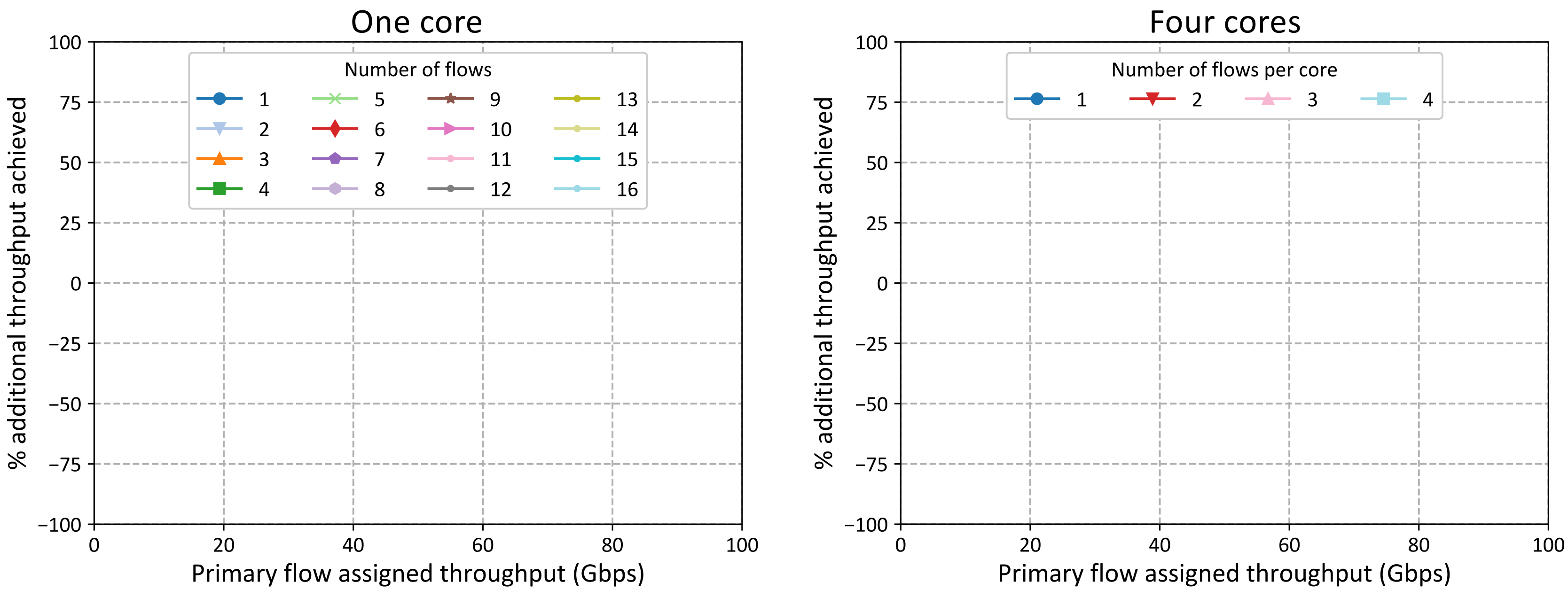


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Rate limiting

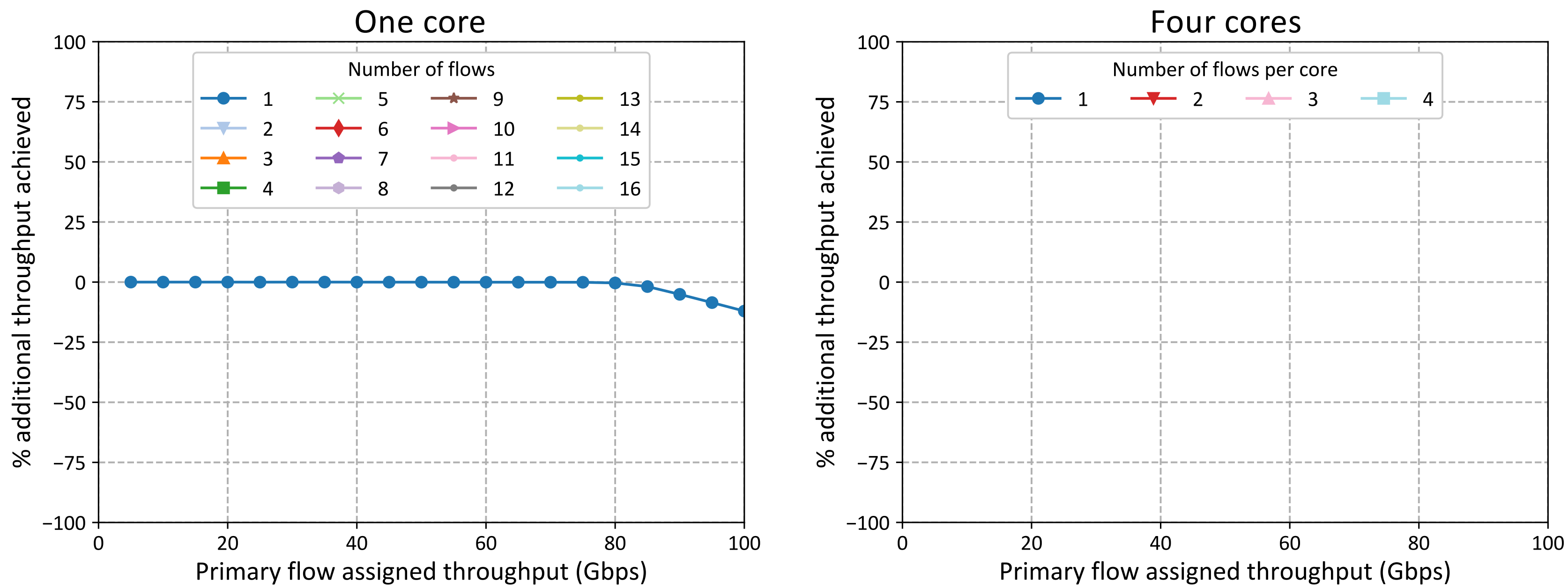


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Rate limiting

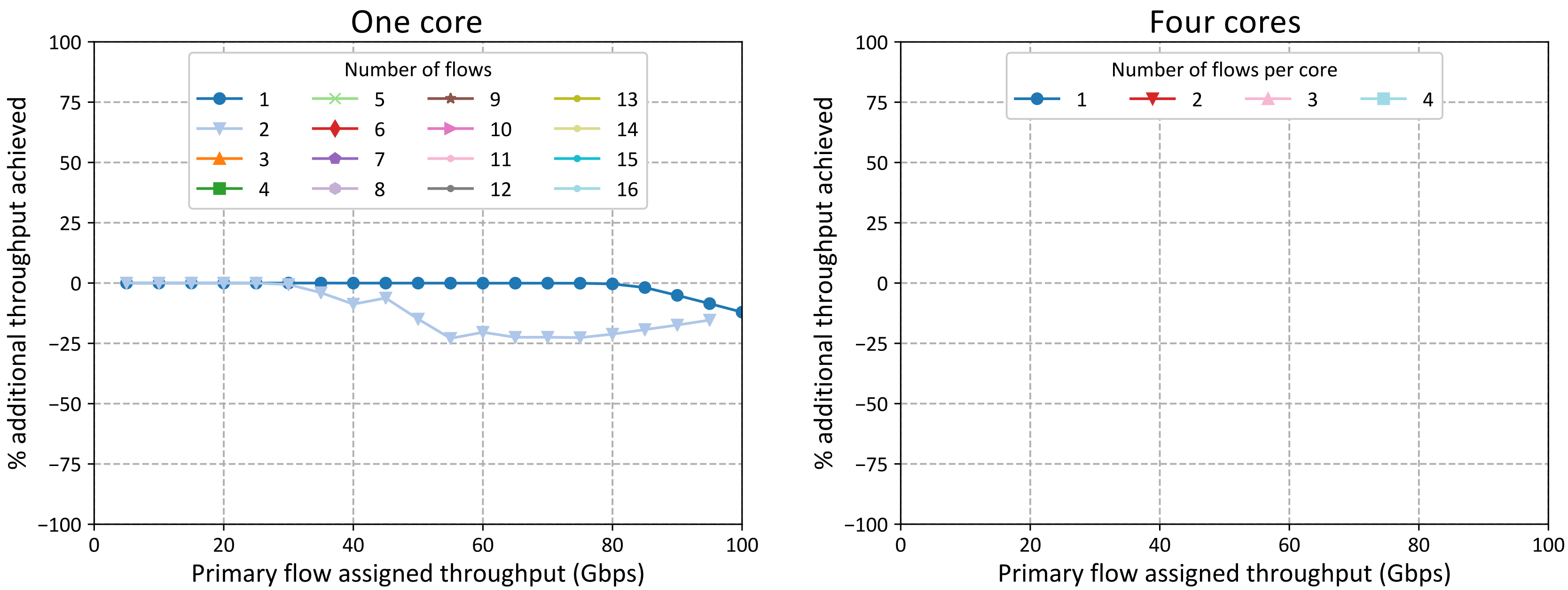


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Rate limiting

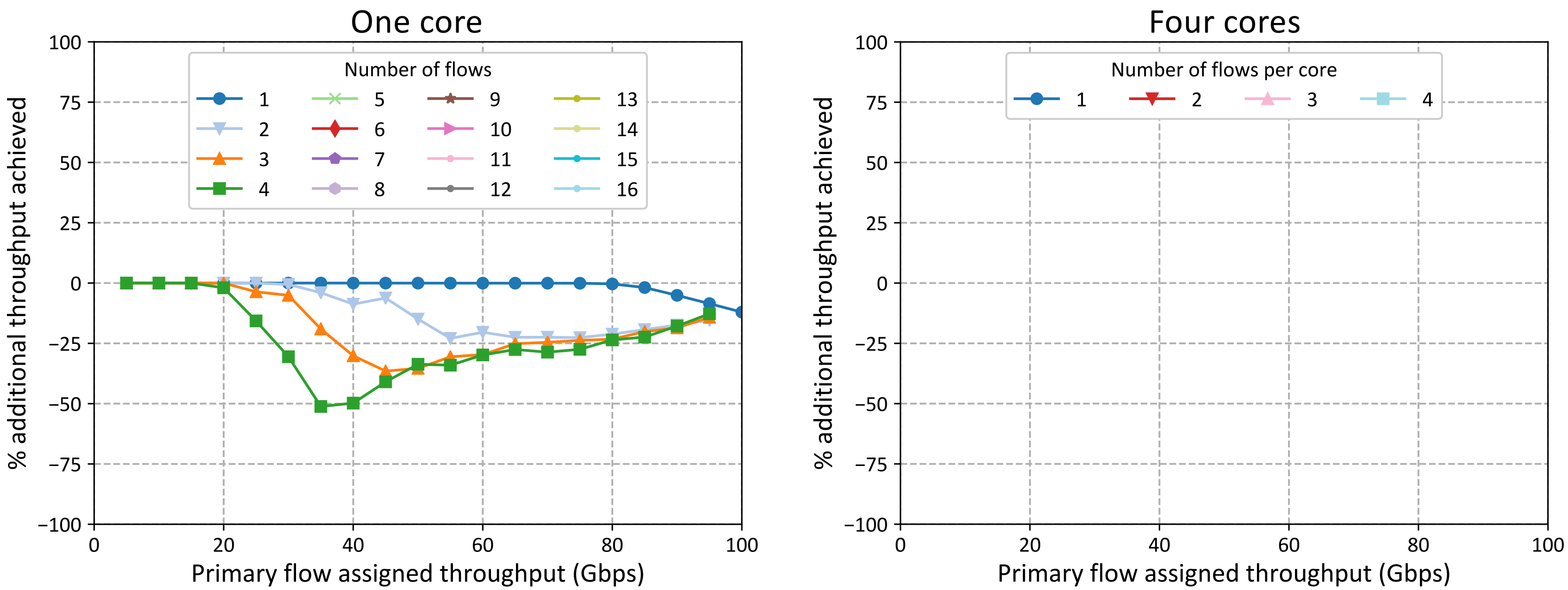


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Rate limiting

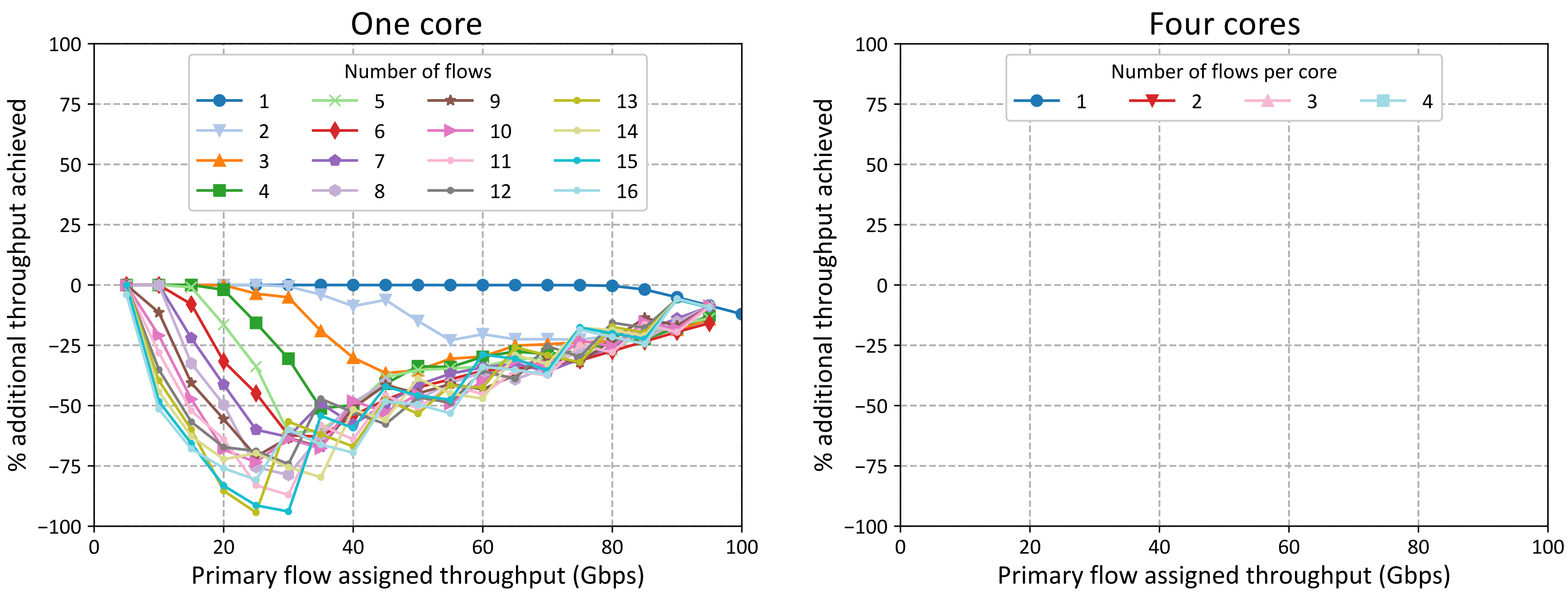


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Rate limiting

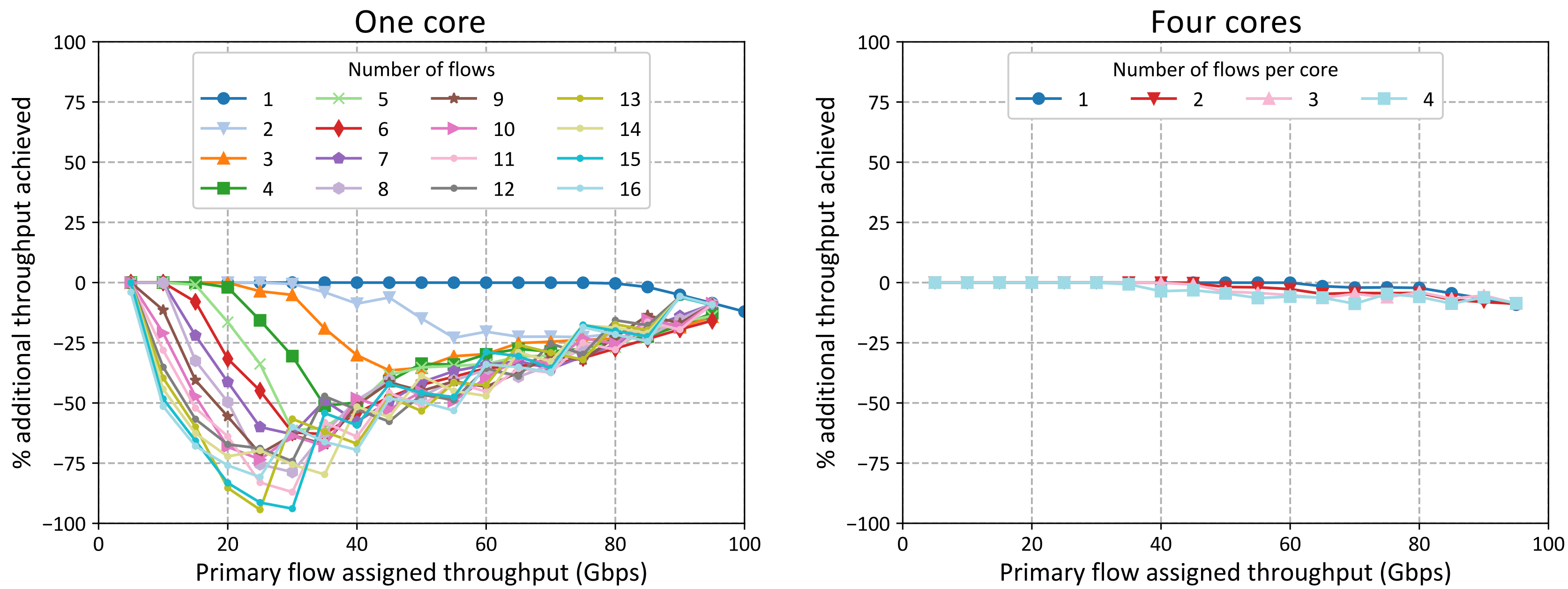
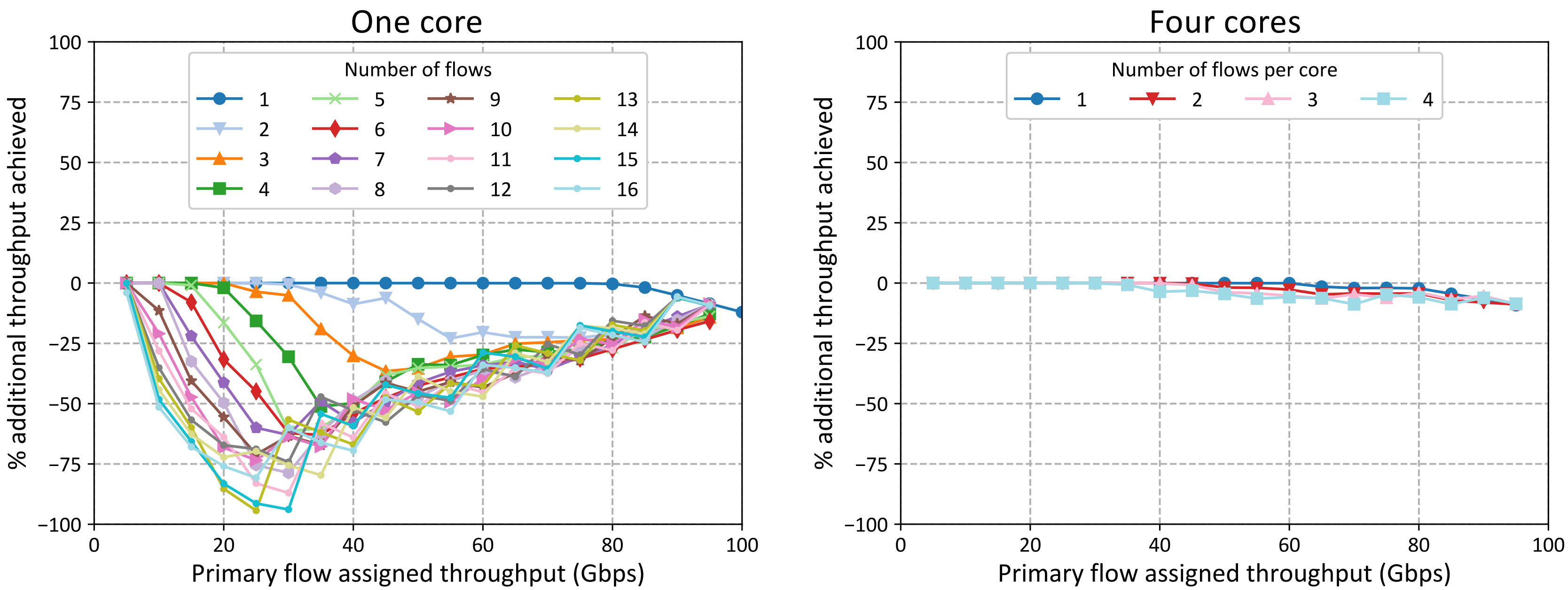


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Rate limiting



We need multiple cores to rate limit flows at 100-Gb/s

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

What does a SoftNIC need to do?

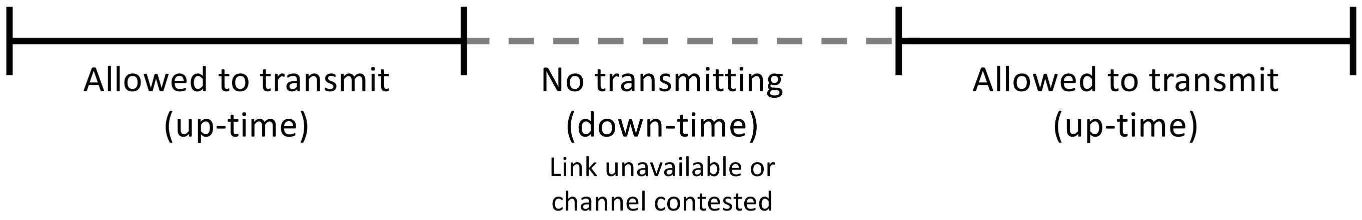
High throughput

Rate limiting

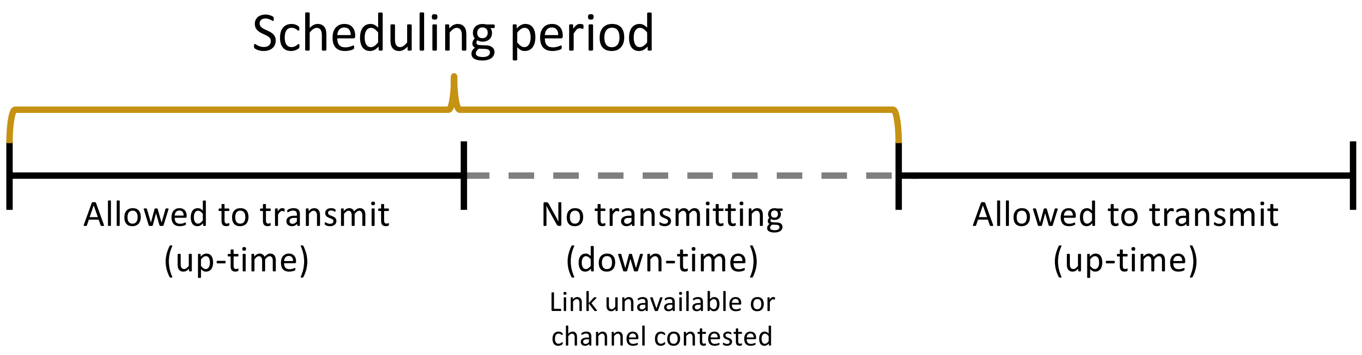
Flow scheduling

Low processing latency

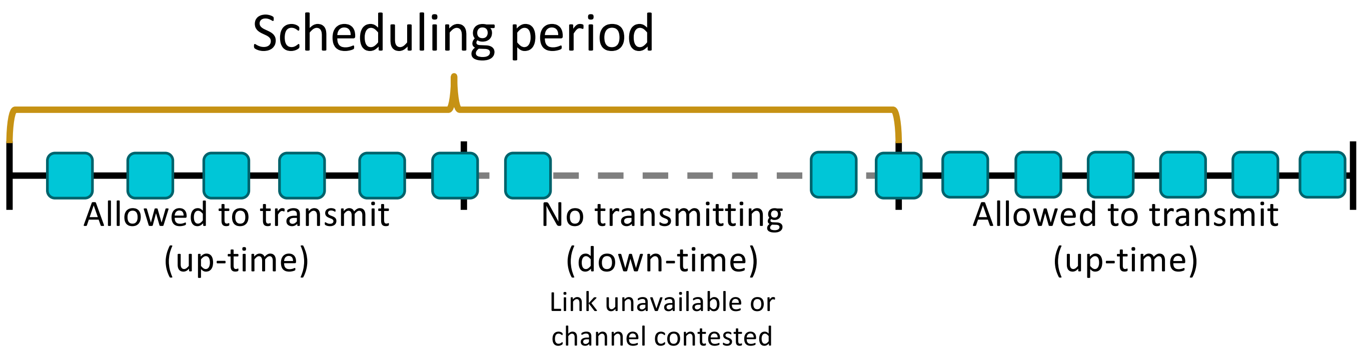
Flow Scheduling



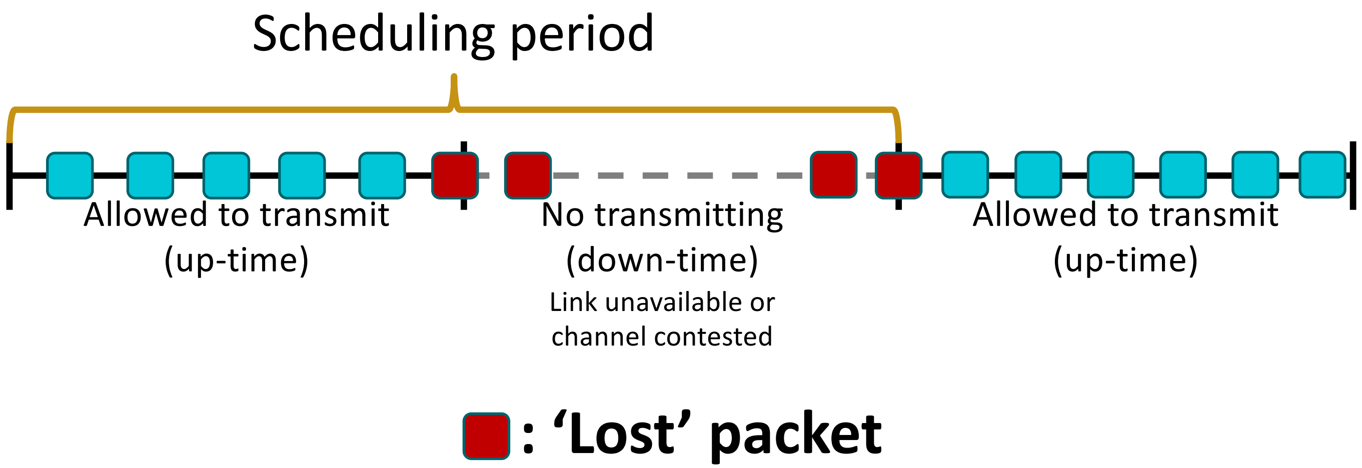
Flow Scheduling



Flow Scheduling



Flow Scheduling



Flow Scheduling

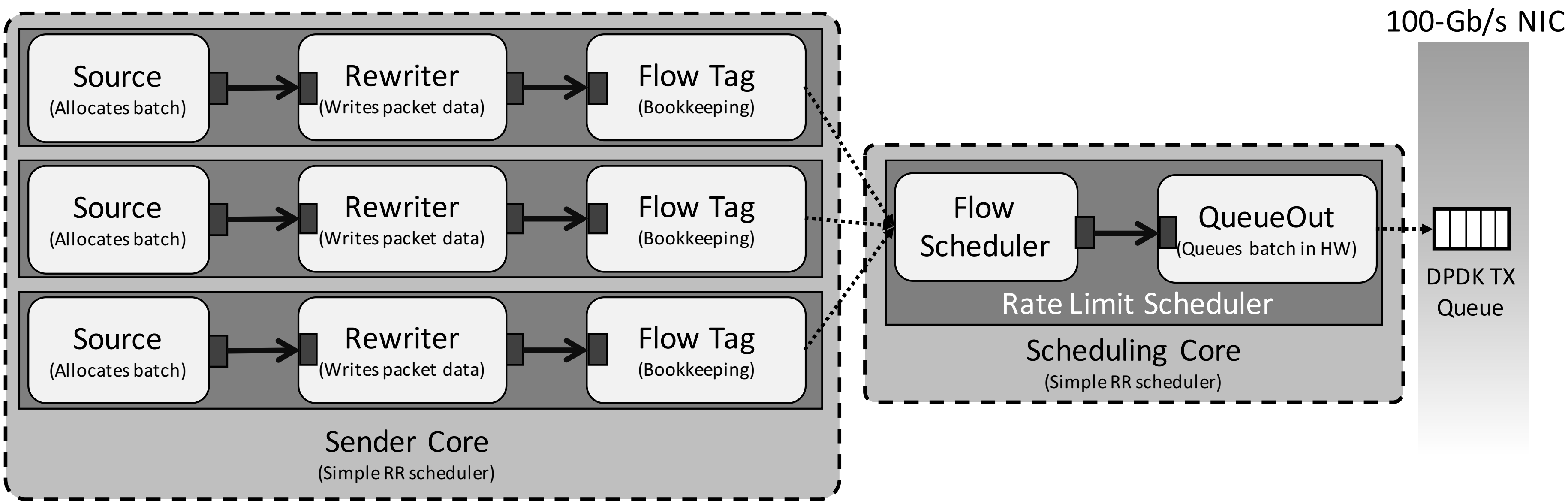
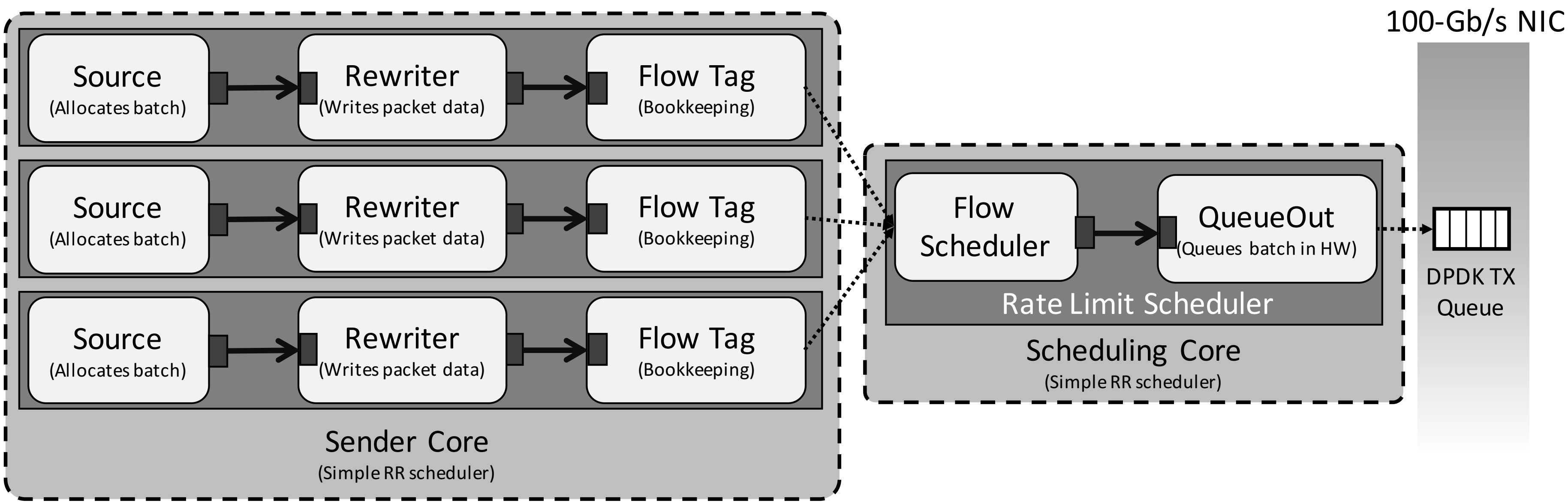


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Flow Scheduling



flows per scheduling core = number of scheduling cores on an endhost

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Flow scheduling loss, 25-Gb/s per scheduling core

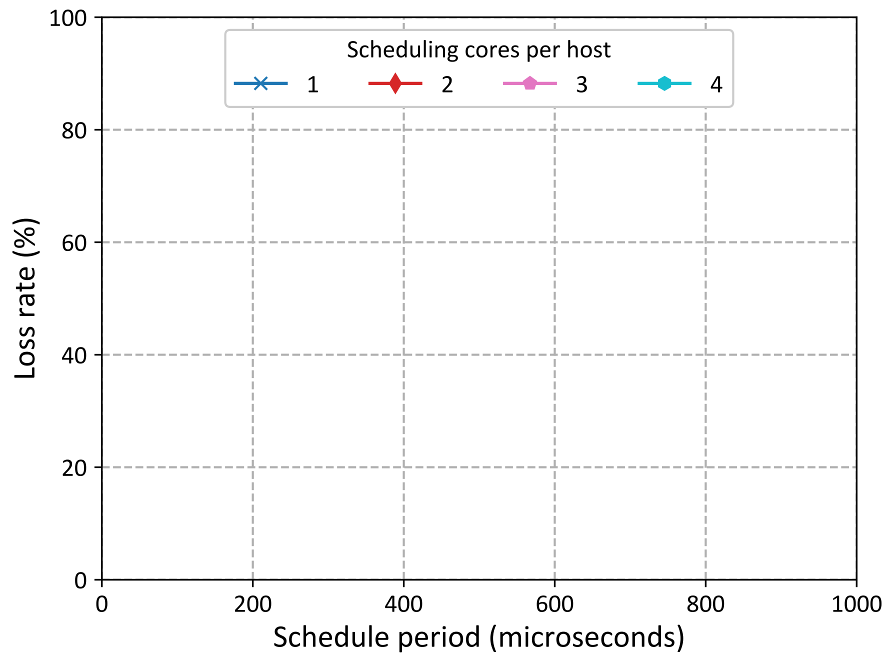


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Flow scheduling loss, 25-Gb/s per scheduling core

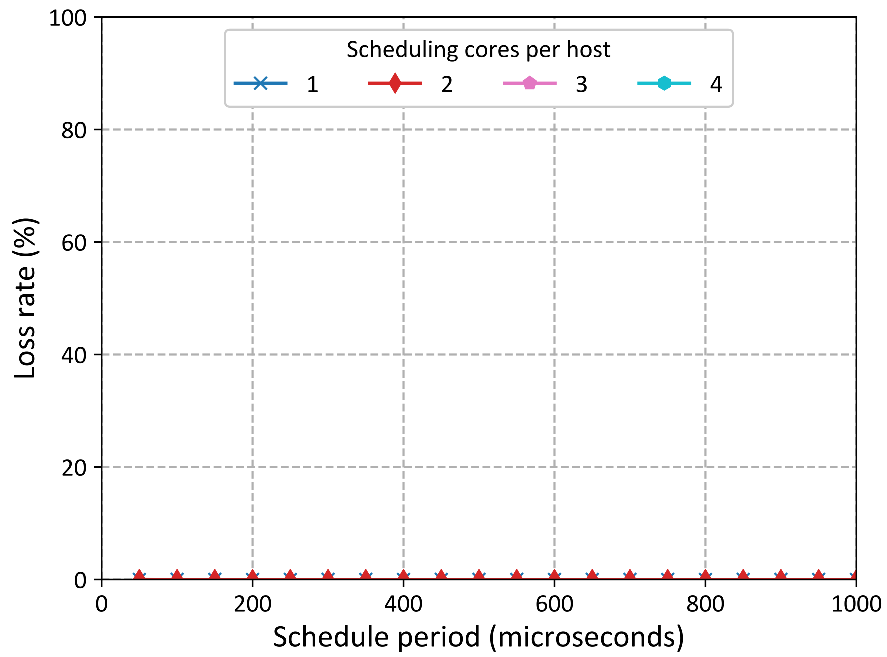


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Flow scheduling loss, 25-Gb/s per scheduling core

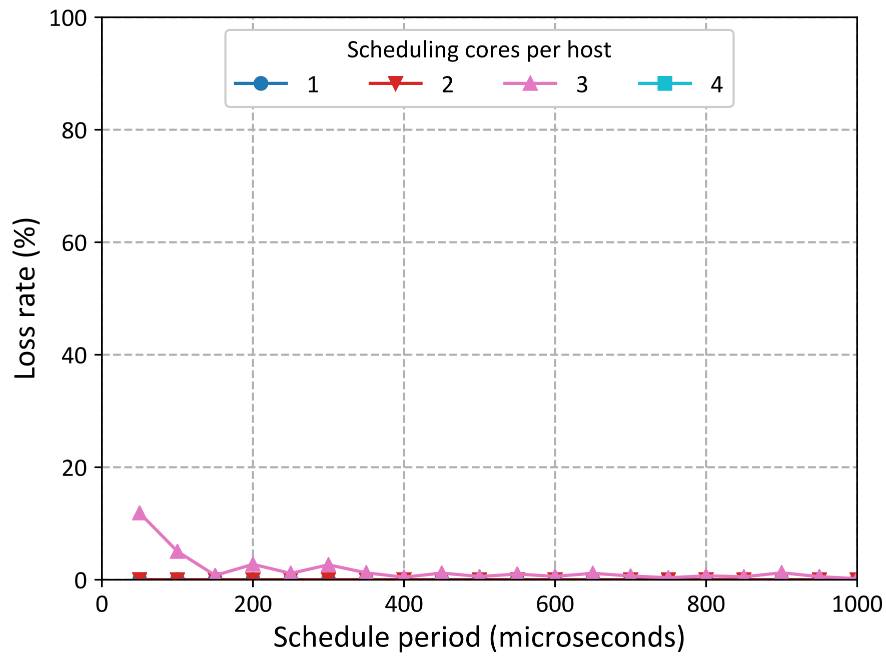


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Flow scheduling loss, 25-Gb/s per scheduling core

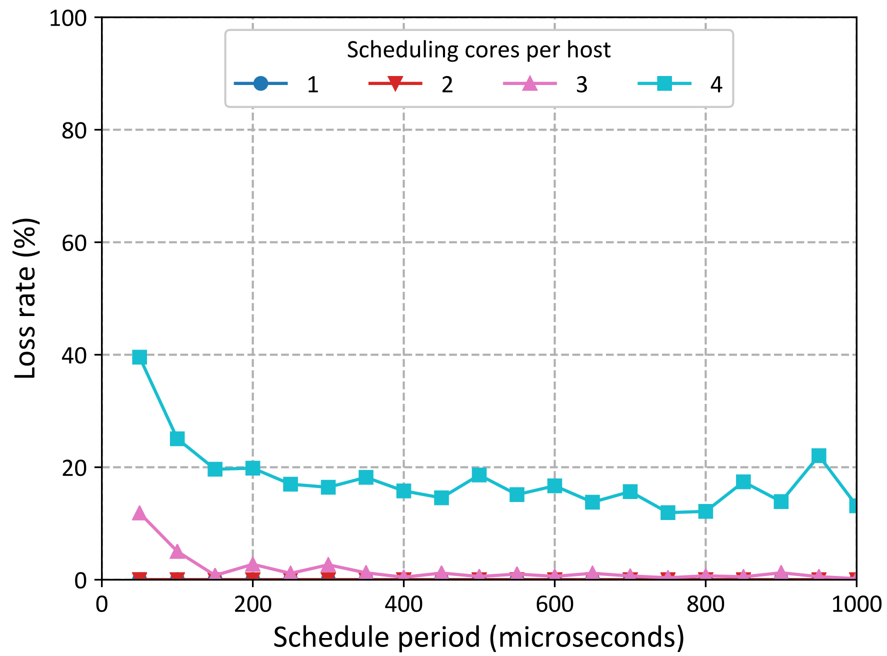
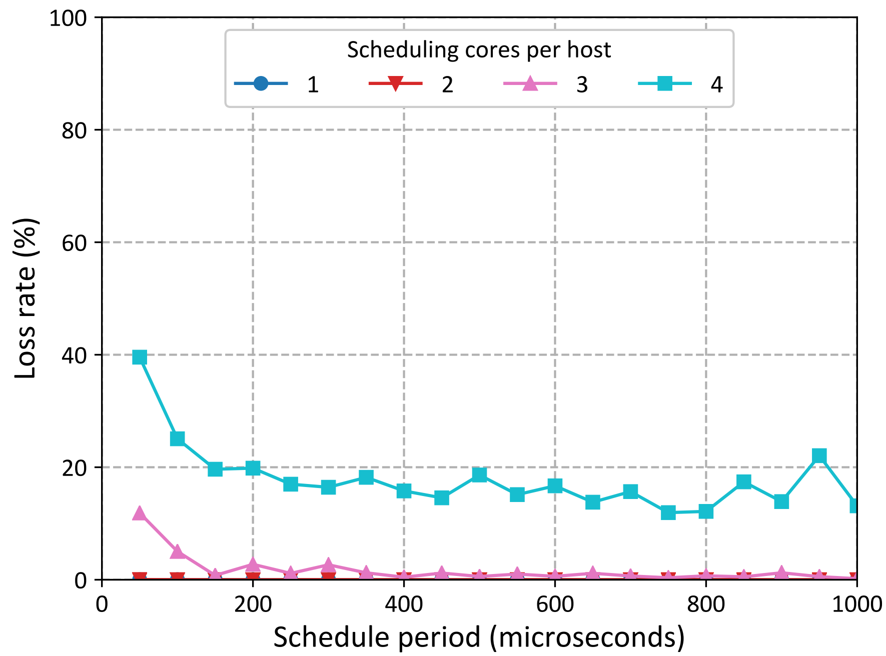


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

Flow scheduling loss, 25-Gb/s per scheduling core



Packet transmission become less precise at higher speeds

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS '18.

What does a SoftNIC need to do?

High throughput

Rate limiting

Flow scheduling

Low processing latency

Generic Routing Encapsulation (GRE)

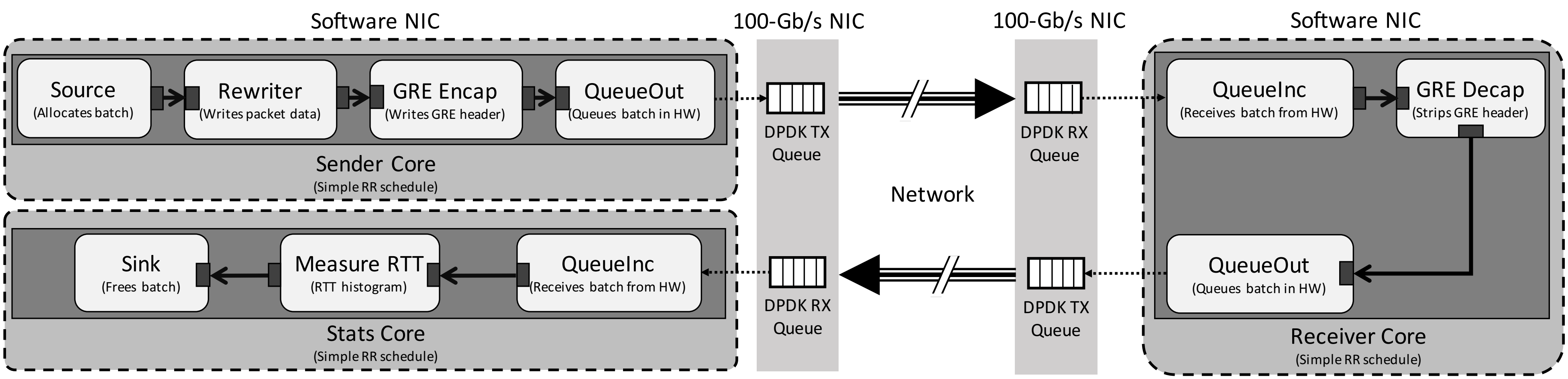


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

GRE forwarding latency

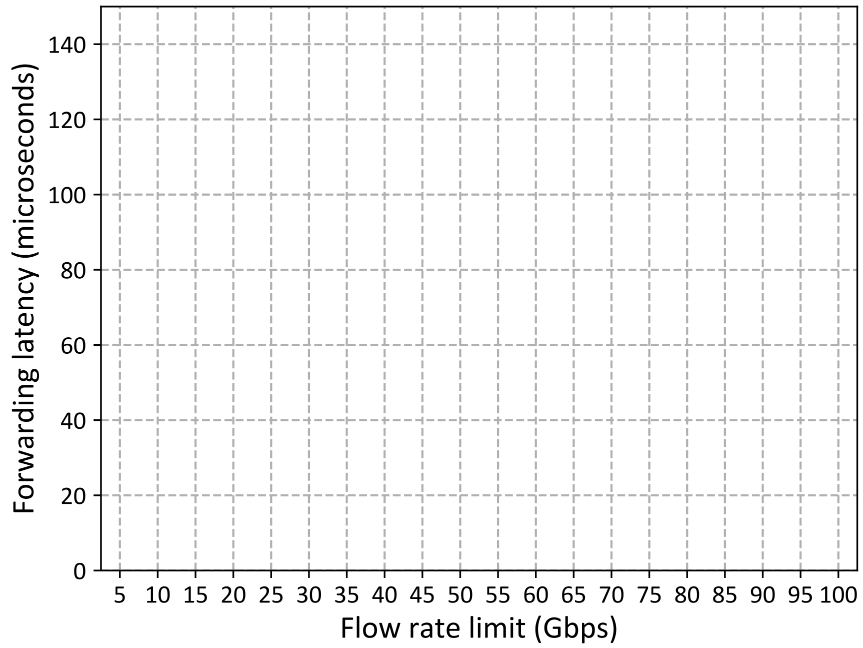


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

GRE forwarding latency

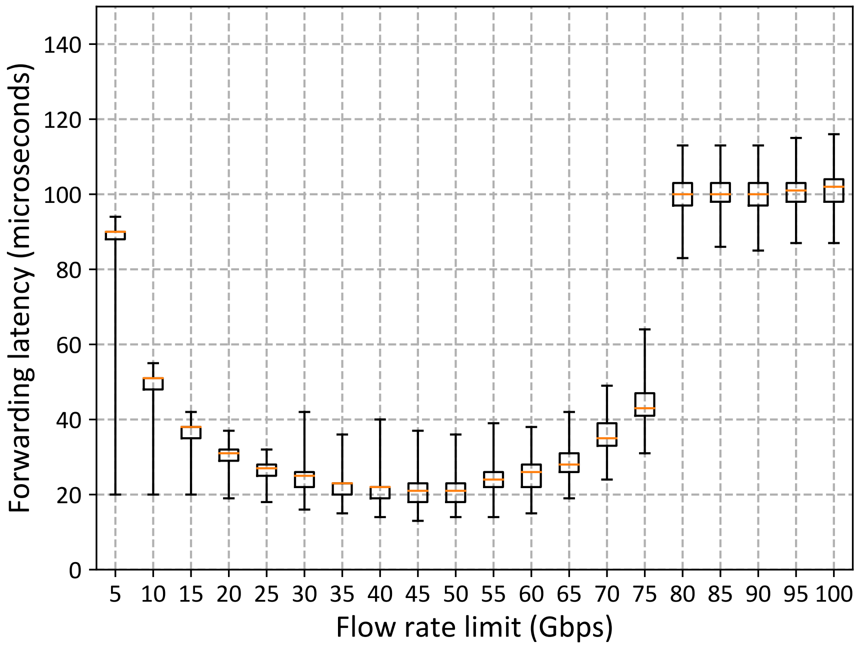
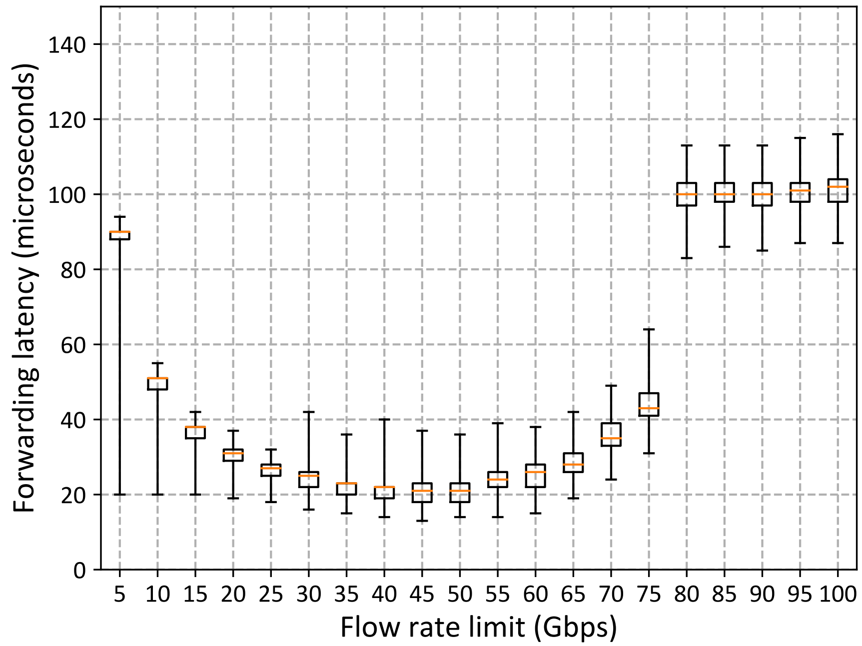


Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

GRE forwarding latency



BESS can’t process and forward packets quickly

Image: R. McGuinness et al, Evaluating the performance of software NICs for 100-gb/s datacenter traffic control, ANCS ‘18.

Can SoftNICs implement TDMA well in optical datacenter networks?

✓ 40-Gb/s ? ~80-Gb/s X 100-Gb/s

Can SoftNICs implement TDMA well in optical datacenter networks?

✓ 40-Gb/s

? ~80-Gb/s

X 100-Gb/s

But optical networks primarily target 100Gb/s+ speeds!

Networks don’t care about precision

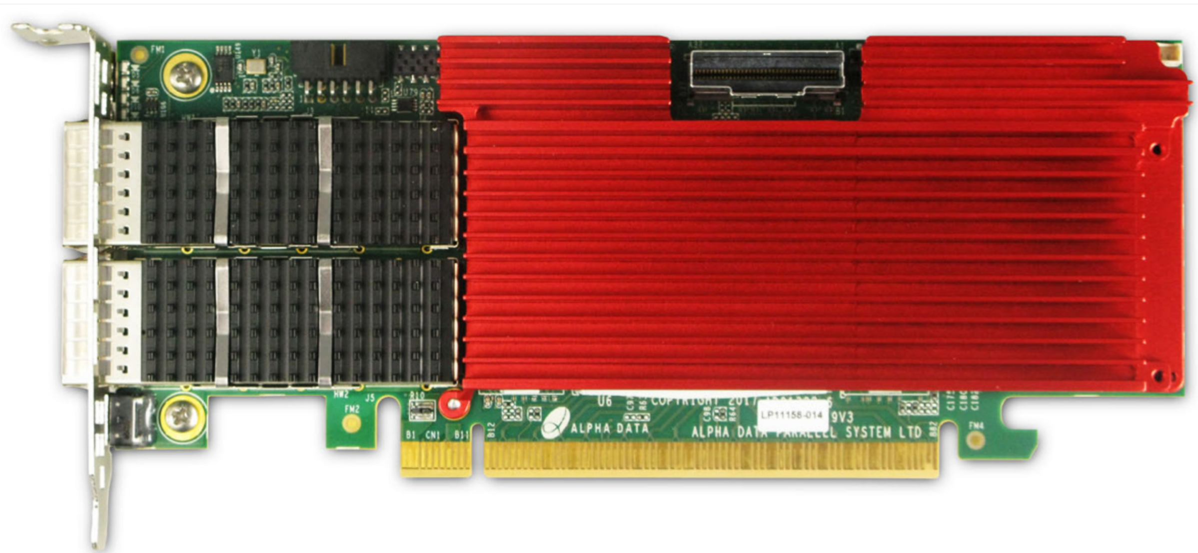
Packet switched networks don’t care about precision

Software *and* hardware follow this trend

What can we do to solve this?

FPGA-based NIC

Real FPGA NIC provides microsecond TDMA precision for end-to-end applications



[1]: A. Kalia et al, Datacenter RPCs can be General and Fast, NSDI ‘19. [2]: S. Han et al, SoftNIC: A Software NIC to Augment Hardware, 2015.

Making endhosts work with circuit-switched networks requires reevaluating both software and hardware

Making endhosts work with circuit-switched networks requires reevaluating both software and hardware

Implementing one without the other has lead to compromises in performance

Making endhosts work with circuit-switched networks requires reevaluating both software and hardware

Implementing one without the other has lead to compromises in performance

Leveraging hardware features in intelligent software design will provide solutions for circuit-switched networking