SQR

In-network packet loss recovery from link failures for high-reliability datacenter networks

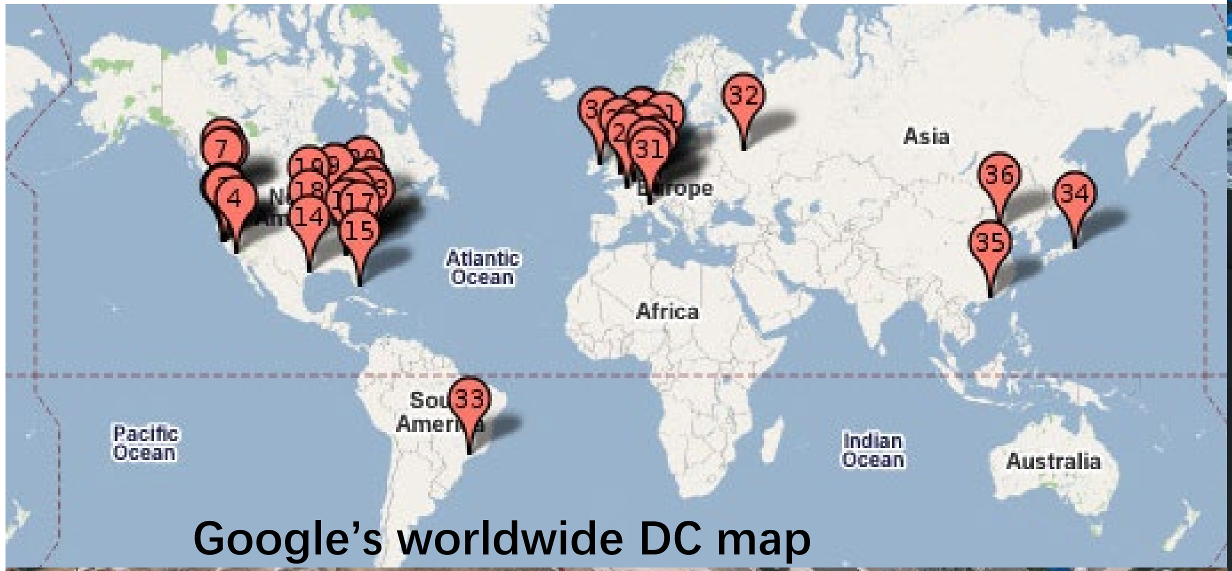
Ting Qu[1, 2] Raj Joshi[2] **Mun Choon Chan**[2]

Ben Leong[2] Deke Guo[1] Zhong Liu[1]

[1] 國防科学技術大学
National University of Defense Technology

[2] NUS
National University of Singapore

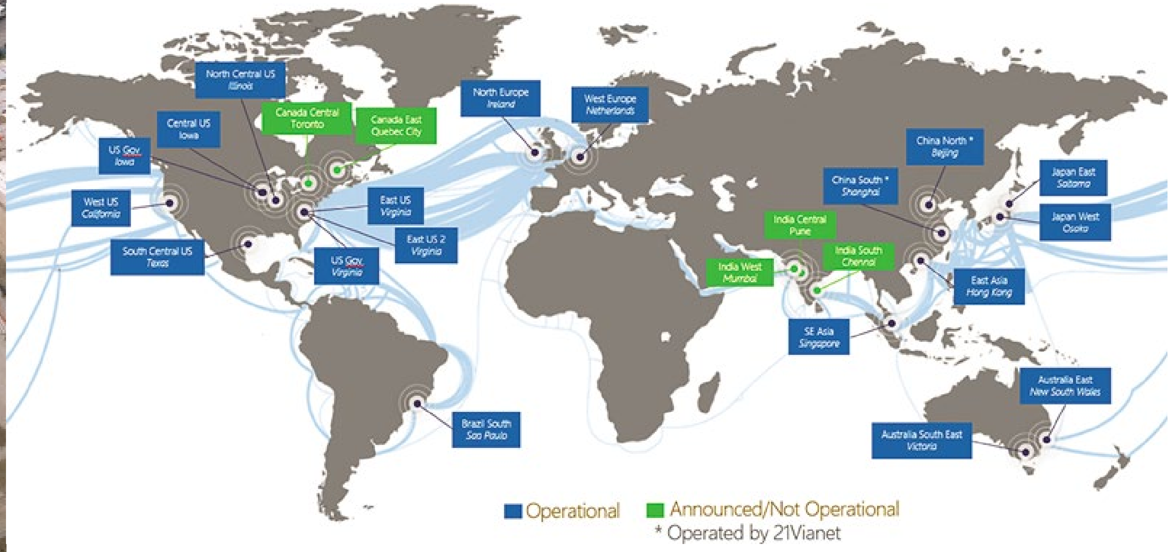
Data centers around the world



Google's worldwide DC map

Facebook DC interior

Microsoft's DC in Dublin, Ireland



Global Microsoft Azure DC Footprint

Low latency is a key requirement

Web search

e-commerce

database

cache

Low latency for short messages

Better app performance & user experience

Improve Flow Completion Time (FCT)

- DCTCP (sigcomm’10)
- D^3 (sigcomm’11)
- HULL (nsdi’12)
- pFabric (sigcomm’13)
- PASE (sigcomm’14)
- TIMELY (sigcomm’15)
- FUSO (atc’16)
- Homa (sigcomm’18)
- HPCC (sigcomm’19)

…

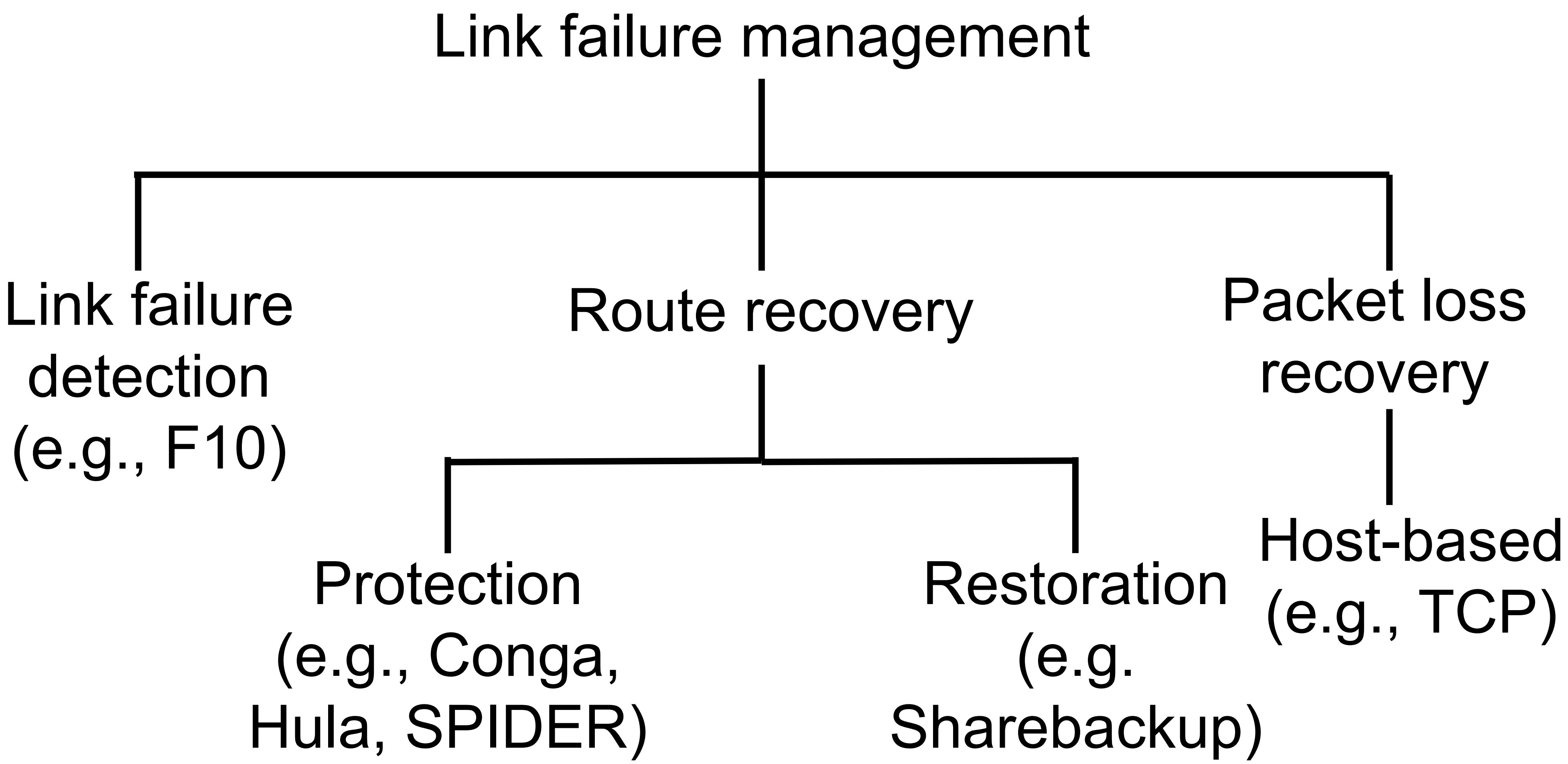
But very few work specifically address how link failures impact FCT

Link failures are common

- Gill et al. [1] reported:
    - Link failure are common and can cause loss of a large number of small packets.
    - The 95th percentile value of link failure is 136 times per day during their measurement period.

[1] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. 2011. Understanding network failures in data centers: measurement, analysis, and implications. In Proceedings of SIGCOMM.

Link failure management



Link failure management

Link failure detection (e.g., F10)

Route recovery

Protection (e.g., Conga, Hula, SPIDER)

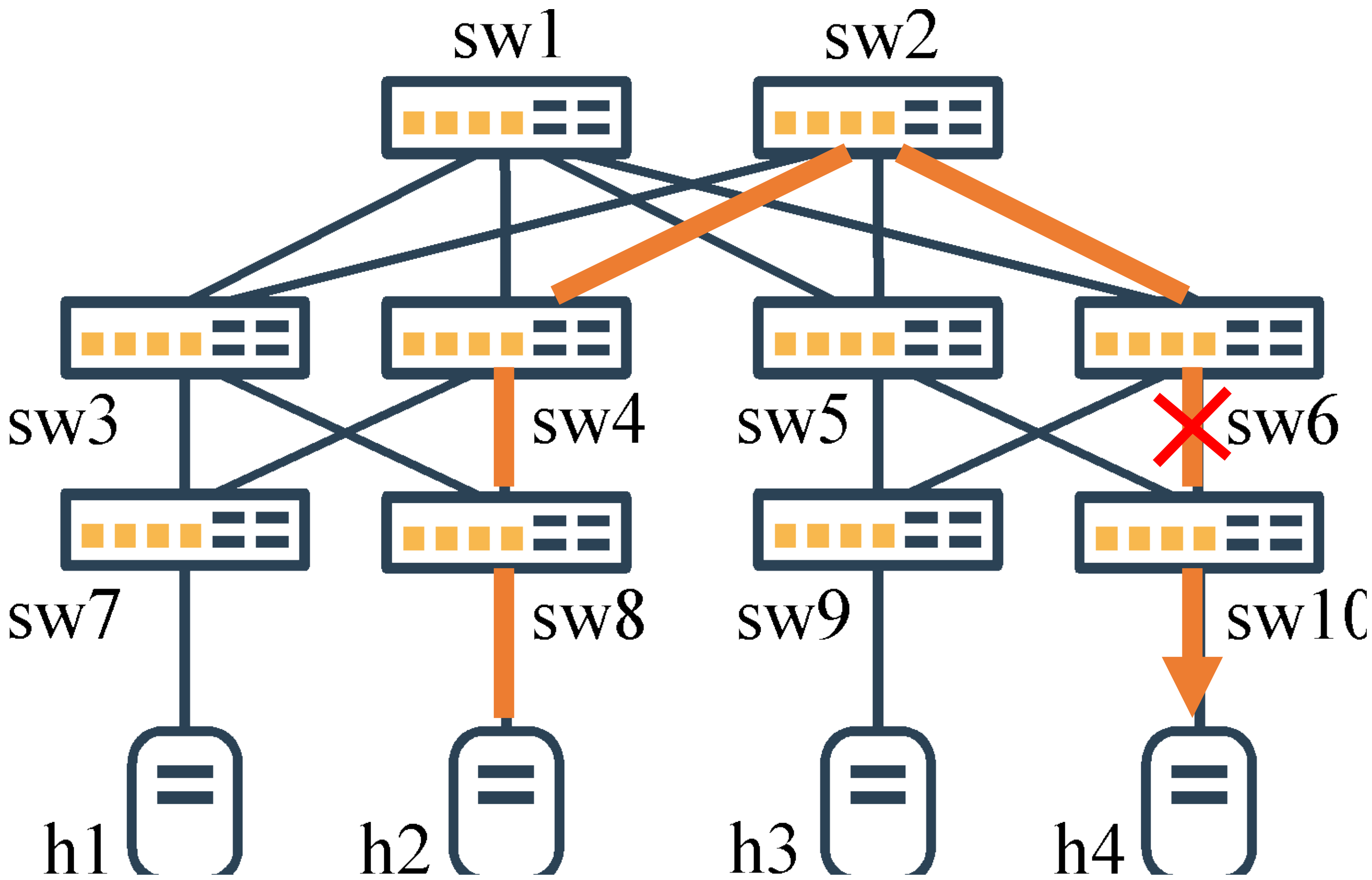
Restoration (e.g. Sharebackup)

Packet loss recovery

Host-based (e.g., TCP)

Host-based pkt loss recovery can lead to much longer flow completion time (FCT) for short flows

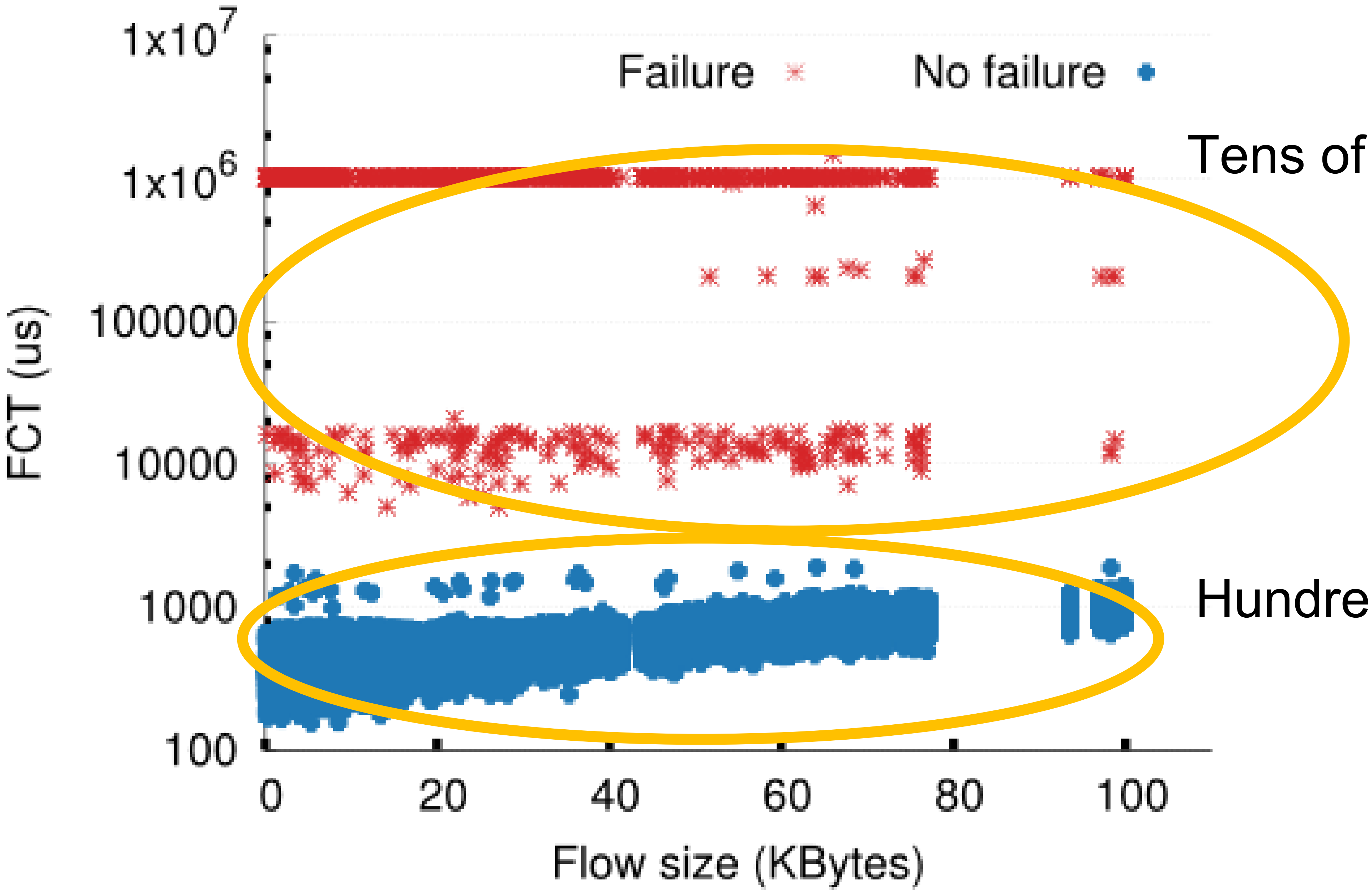
Link failure case



Link detection time 30us (F10, NSDI'13)	+	route reconfiguration time 730us (ShareBackup, sigcomm'18)	=	760us

Long FCT under link failure

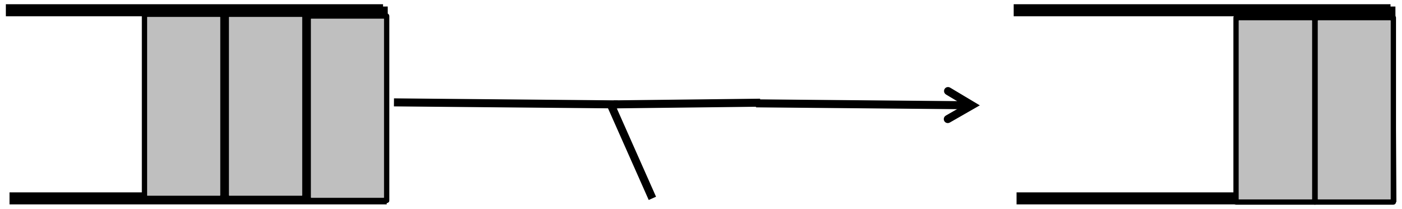
Host based recovery is a major contributor to the large increase in FCT



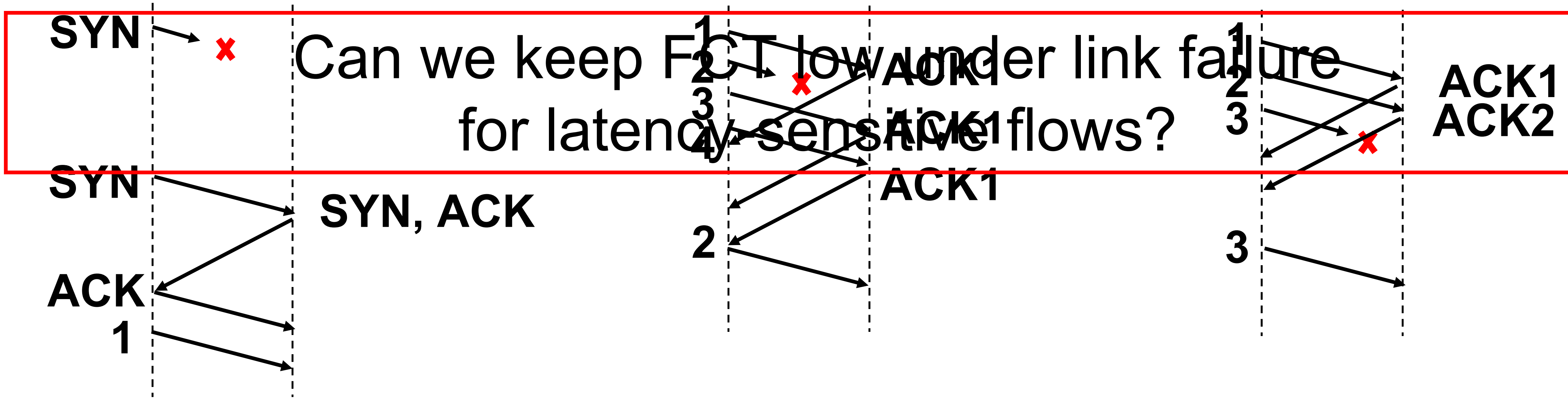
Tens of ms to 1s

Hundreds of μs

Why does host-based recovery increase FCT significantly?

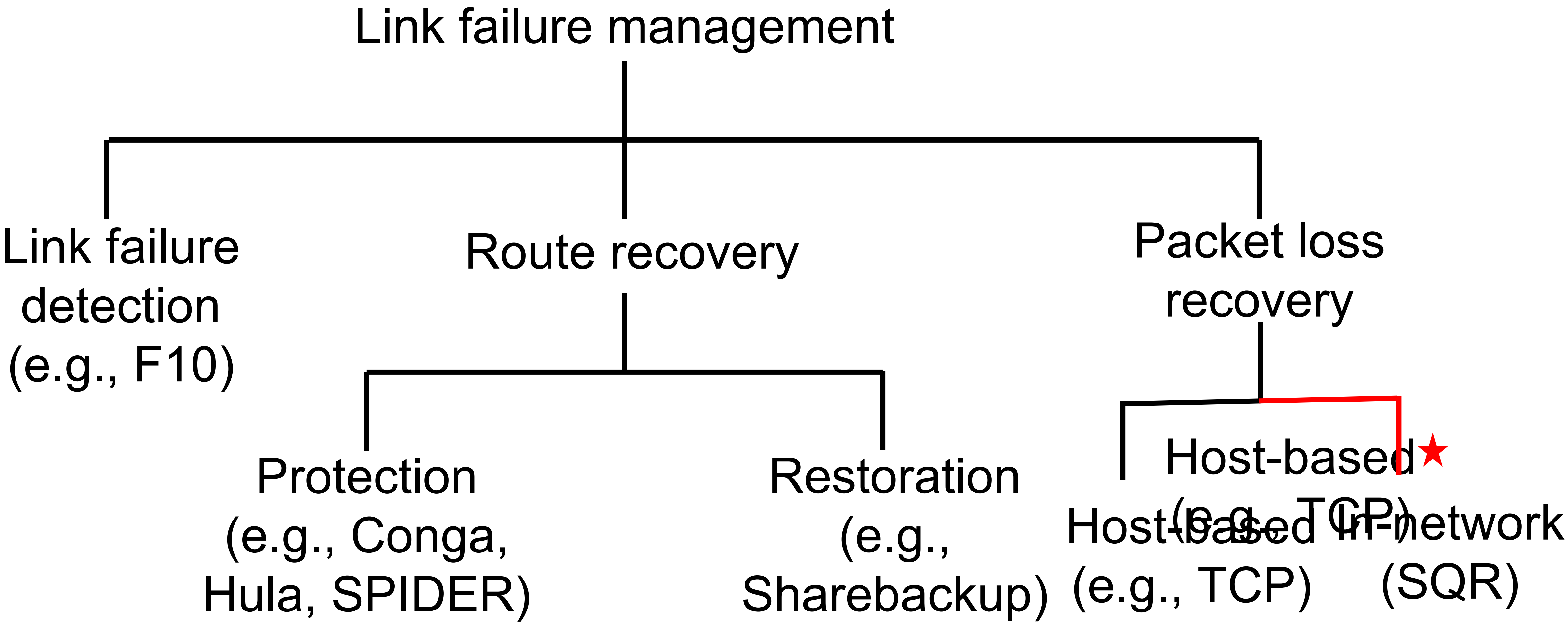


- Packet losses in the TCP three-way handshake
  - Wait at least 1s and retransmit
- Packet losses in the middle of a cwnd
  - Fast retransmission: 1RTT (100s of us)
- Packet losses at the tail of a cwnd
  - Retransmission timeout: several ms



Can we keep FCT low under link failure for latency-sensitive flows?

Our solution: SQR



Link failure management

Link failure detection (e.g., F10)

Route recovery

Packet loss recovery

Protection (e.g., Conga, Hula, SPIDER)

Restoration (e.g., Sharebackup)

Host-based★ (e.g., TCP)

Host-based (e.g., TCP)

In-network (SQR)

The network is the "right" place to perform packet loss recovery

How does SQR keeps FCT low when there is link failure?

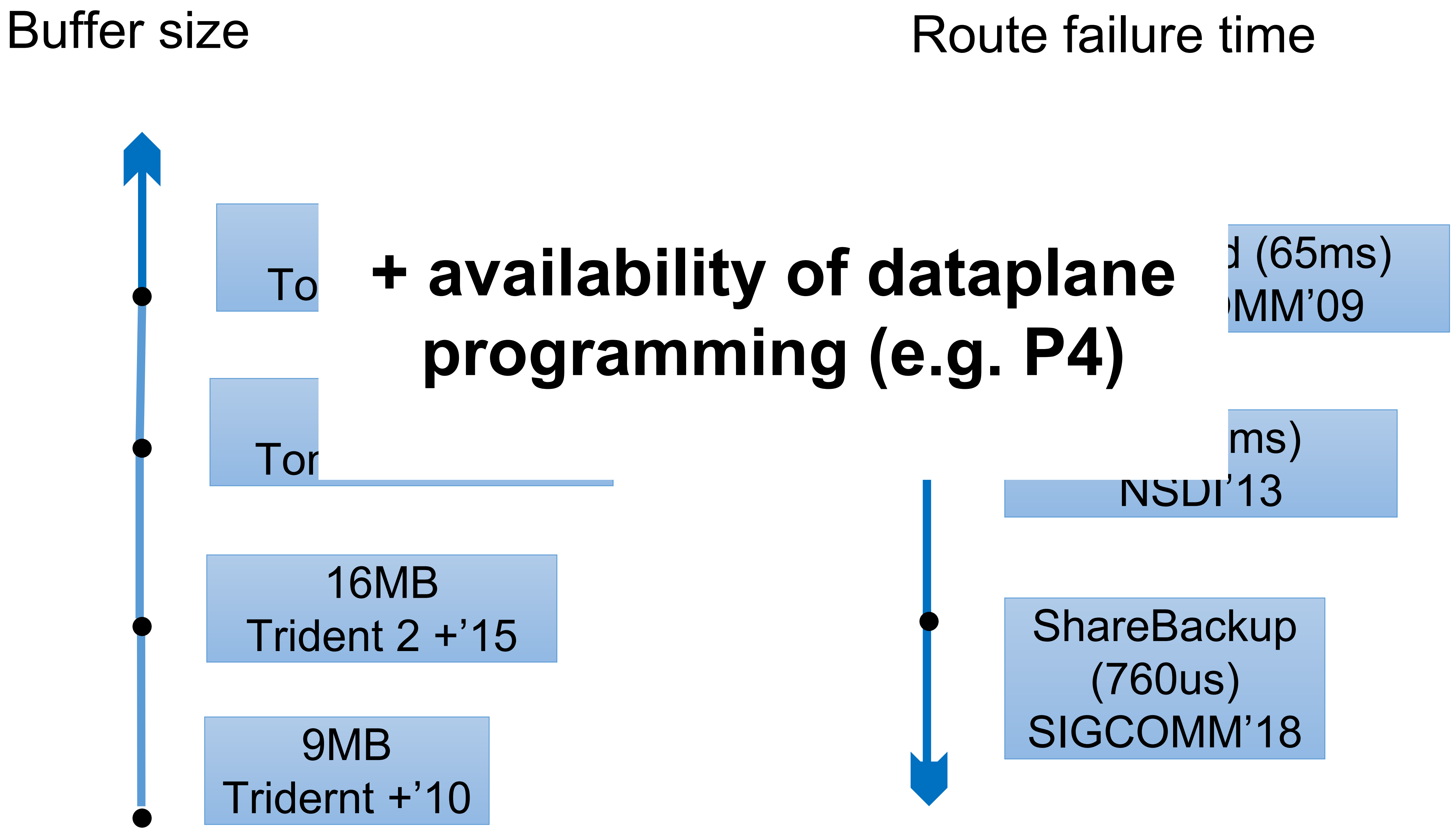
Objective:

- Mask the effect of packet loss from the end-points during link failure detection time and route reconfiguration time (route failure time).

Key idea:

- **Continuously cache recently sent packet in the switch for a duration equal to the route failure time**

Is it feasible to cache pkts on switch?



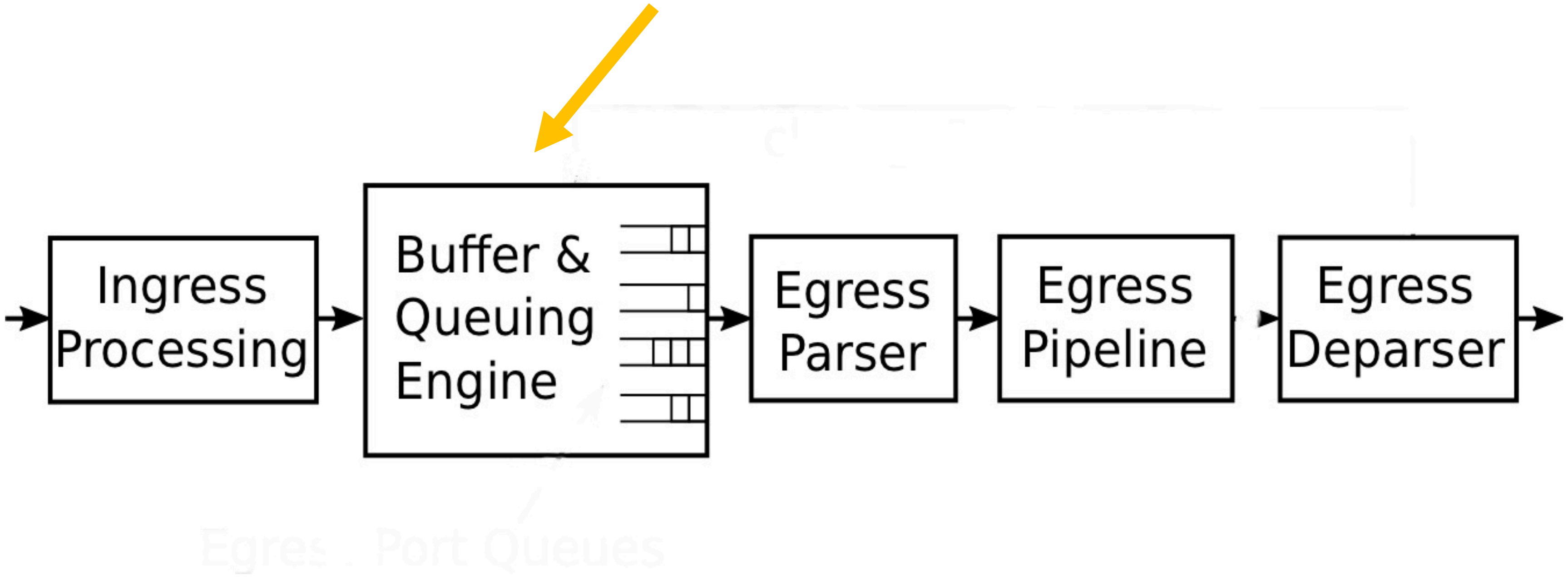
+ availability of dataplane programming (e.g. P4)

Where and how to cache?

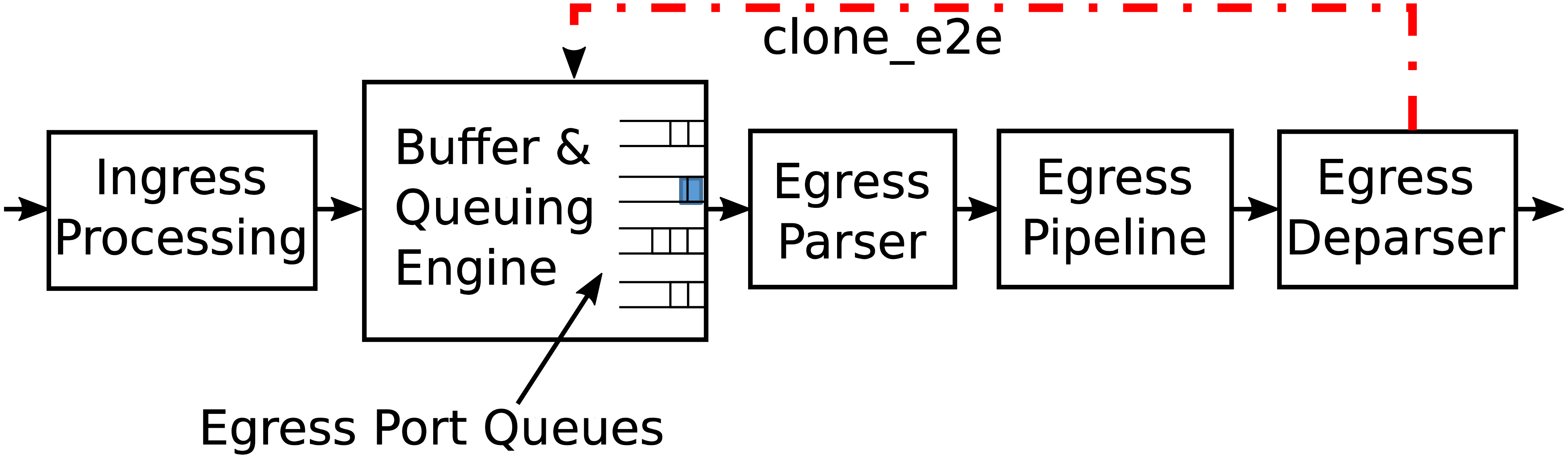
- In a switch dataplane, the packets can only be stored in the packet buffer within the buffer & queuing engine (BQE).

◆Challenges

- The default FIFO queues send out packets as fast as possible.
- No BQE today readily provides the queuing discipline required to realize packets caching with a fixed time.
- BQE does not support custom packet scheduling algorithms.

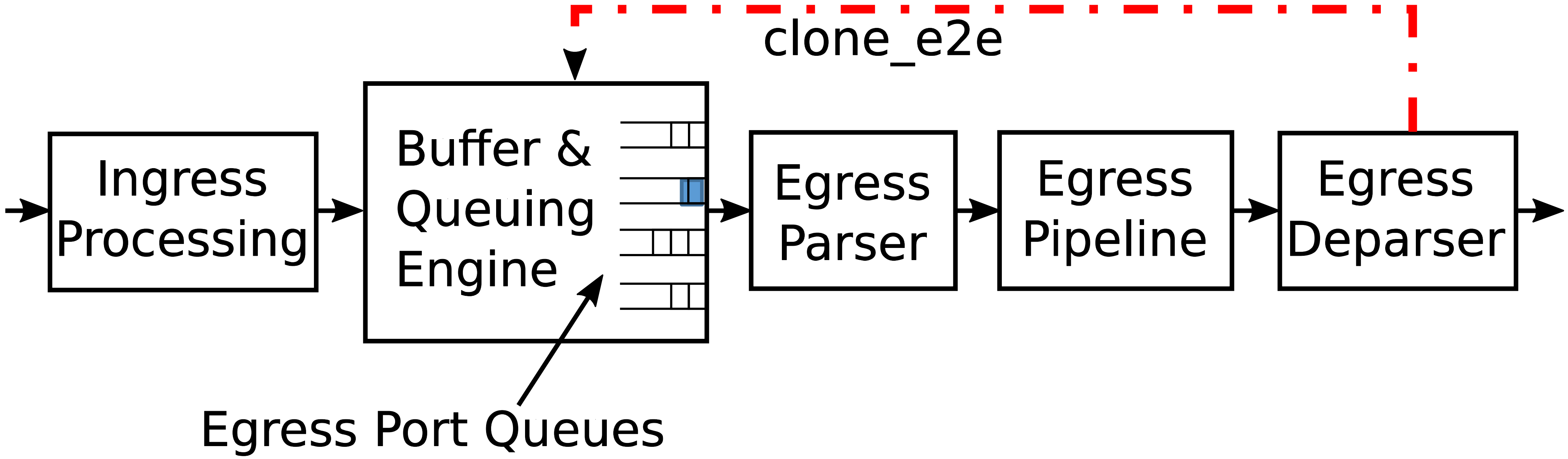


Solution



- Keep recent copies of transmitted packets by cloning and then recirculating cloned packets to BQE.
- ❑ Supported by the Portable Switch Architecture (PSA)
- Packets are cached for durations sufficiently long to detect link failure and perform route recovery.
- Resend cached packets to new route when it is available.

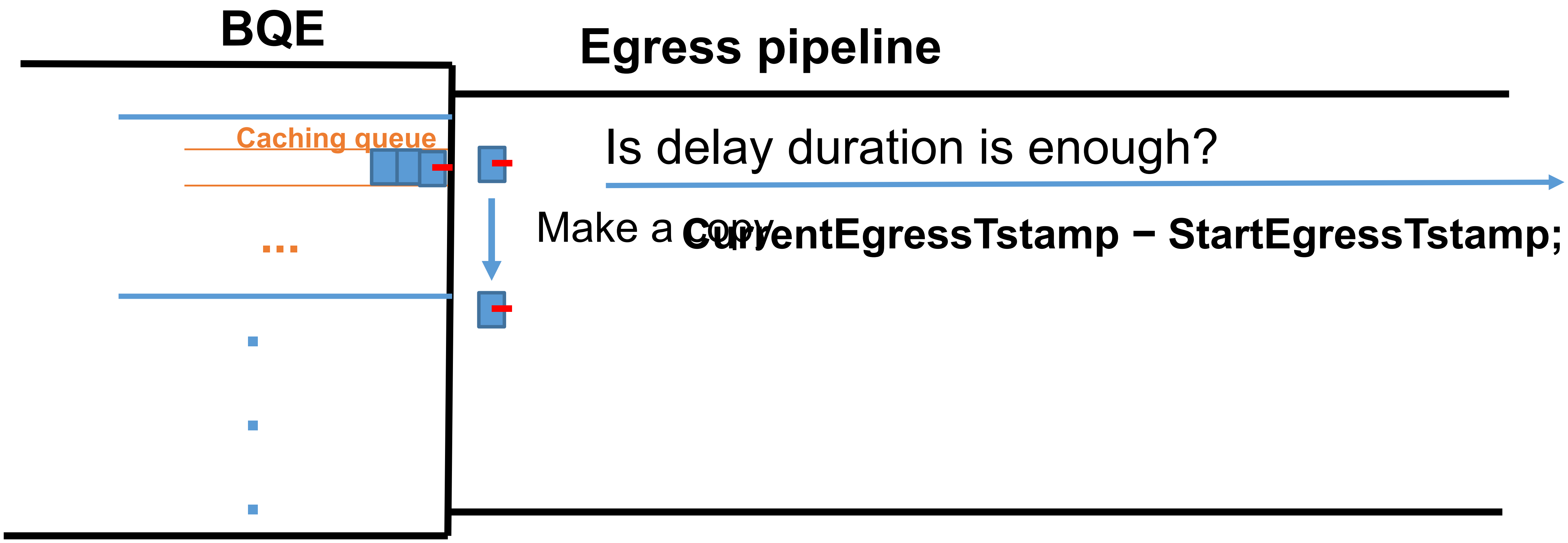
Challenges



- □ “Aging” of packets
- □ Load balancing of circulating packets
- □ Handle packet reordering

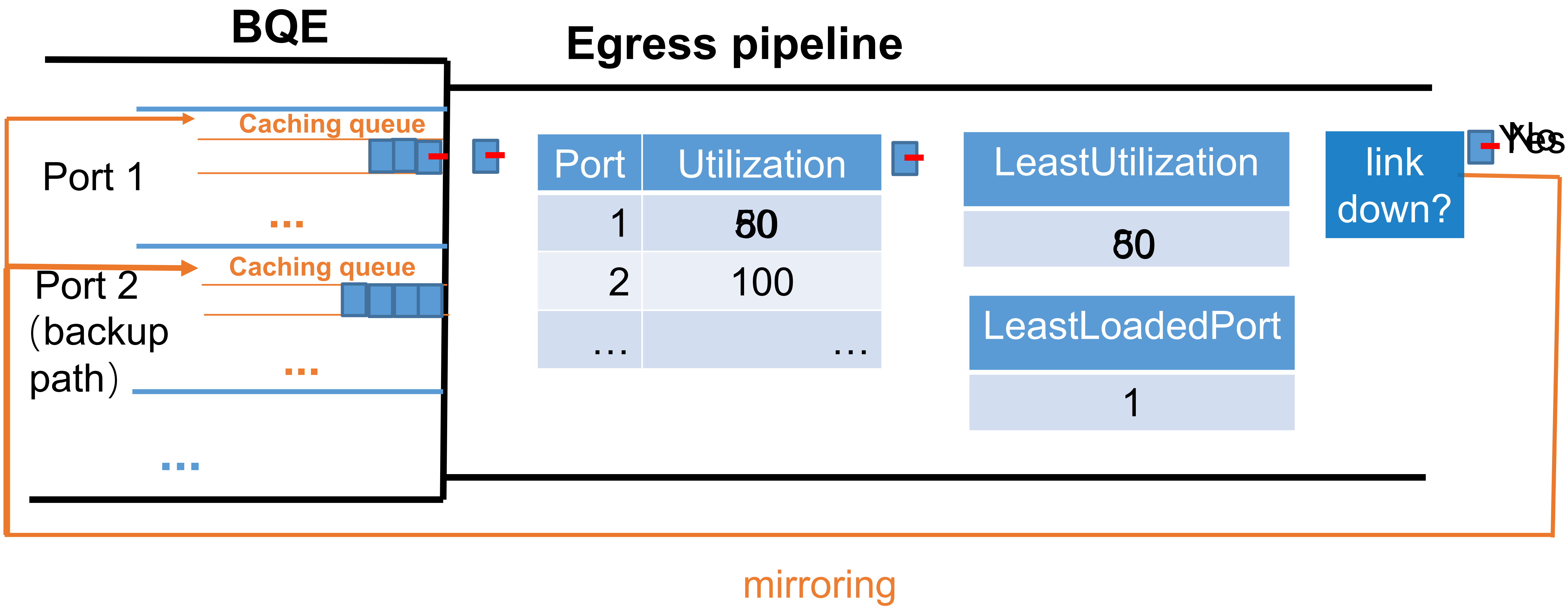
Delay timer

Transmit packet if this is the first/original packet


BQE
Egress pipeline
Caching queue
...
Is delay duration is enough?
Make a copy
CurrentEgressTstamp − StartEgressTstamp;

Packet is dropped if it has been cached greater than link detection time

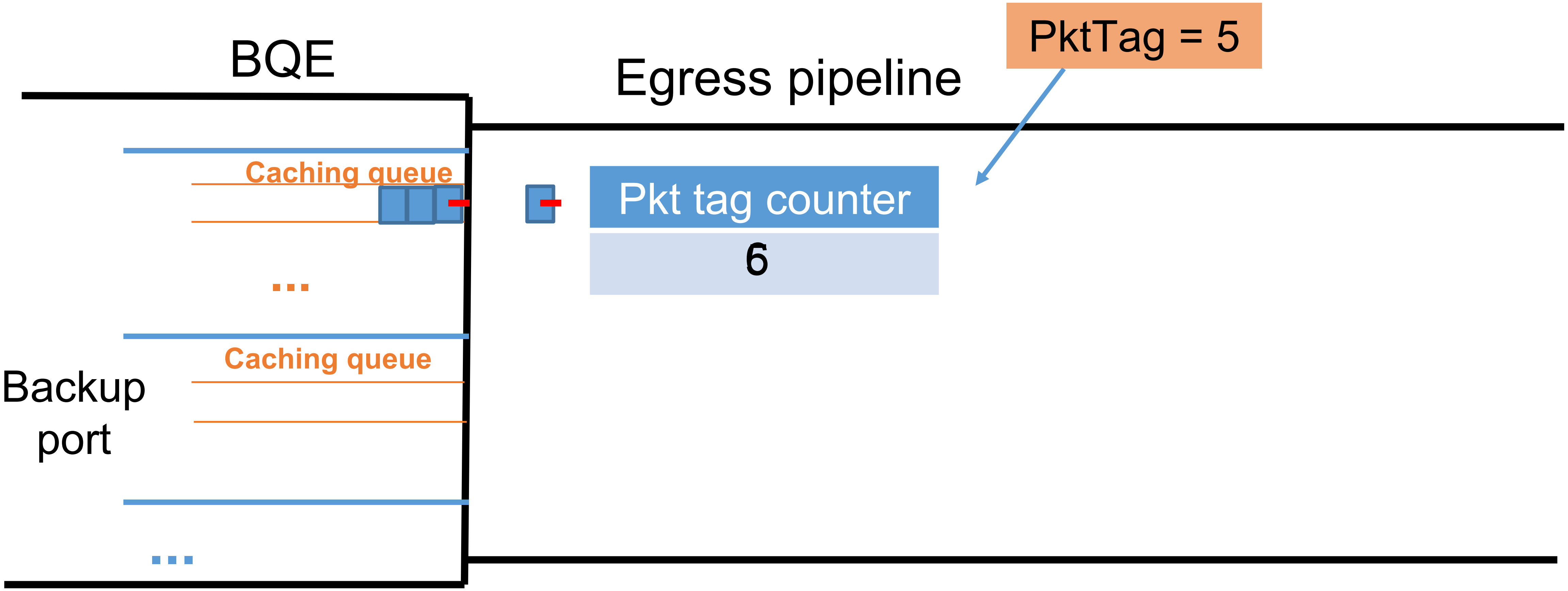
Dynamic queue selection



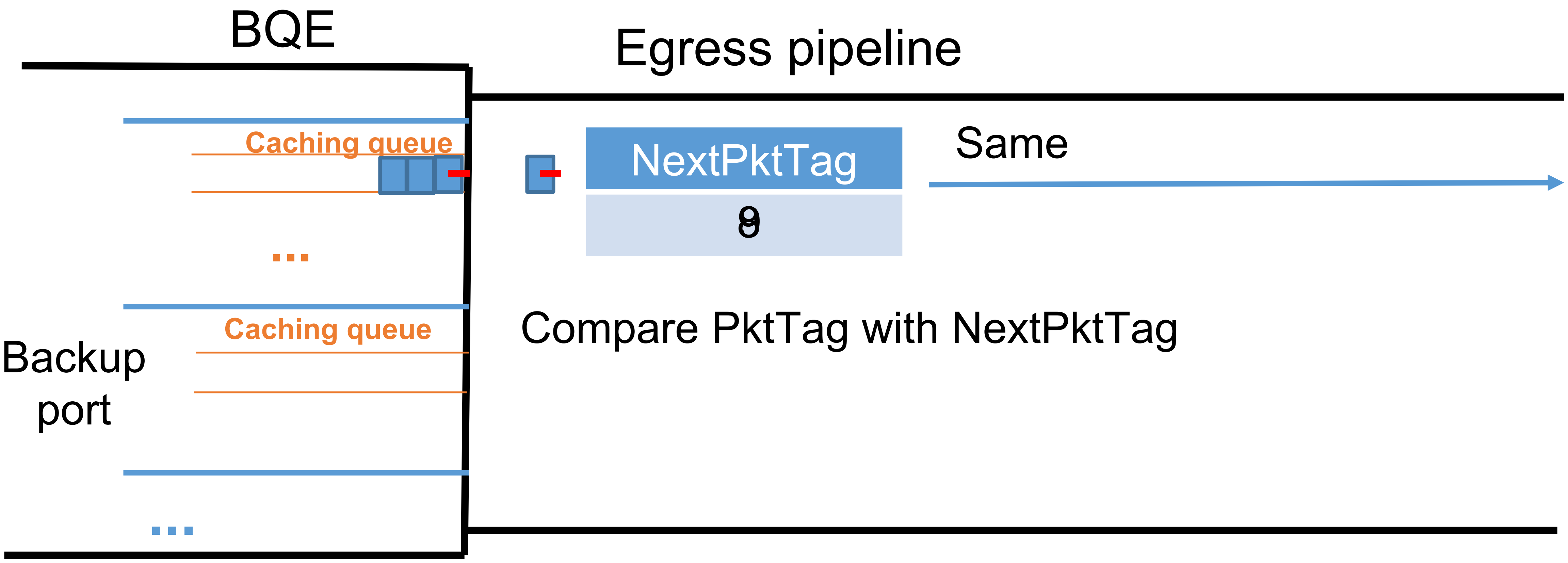
Port	Utilization
1	50 80
2	100
…	…

Packets from same flow can be cached on different queues

Packet order logic

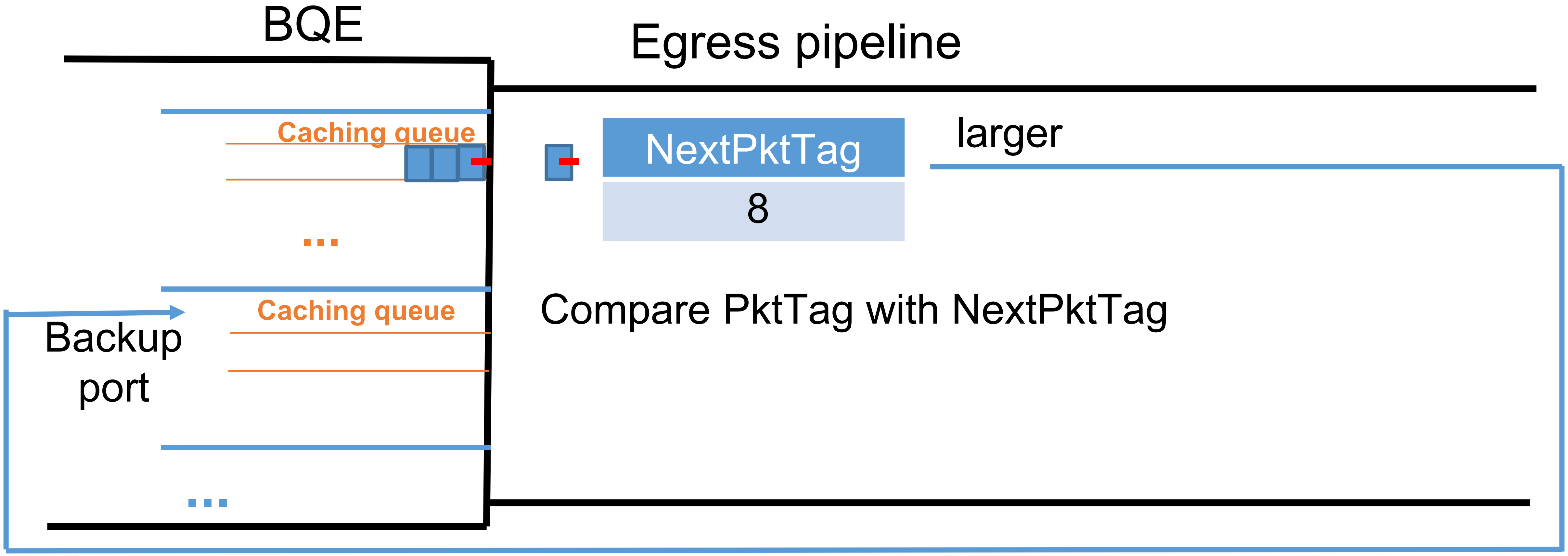


Packet order logic



Compare PktTag with NextPktTag

Packet order logic



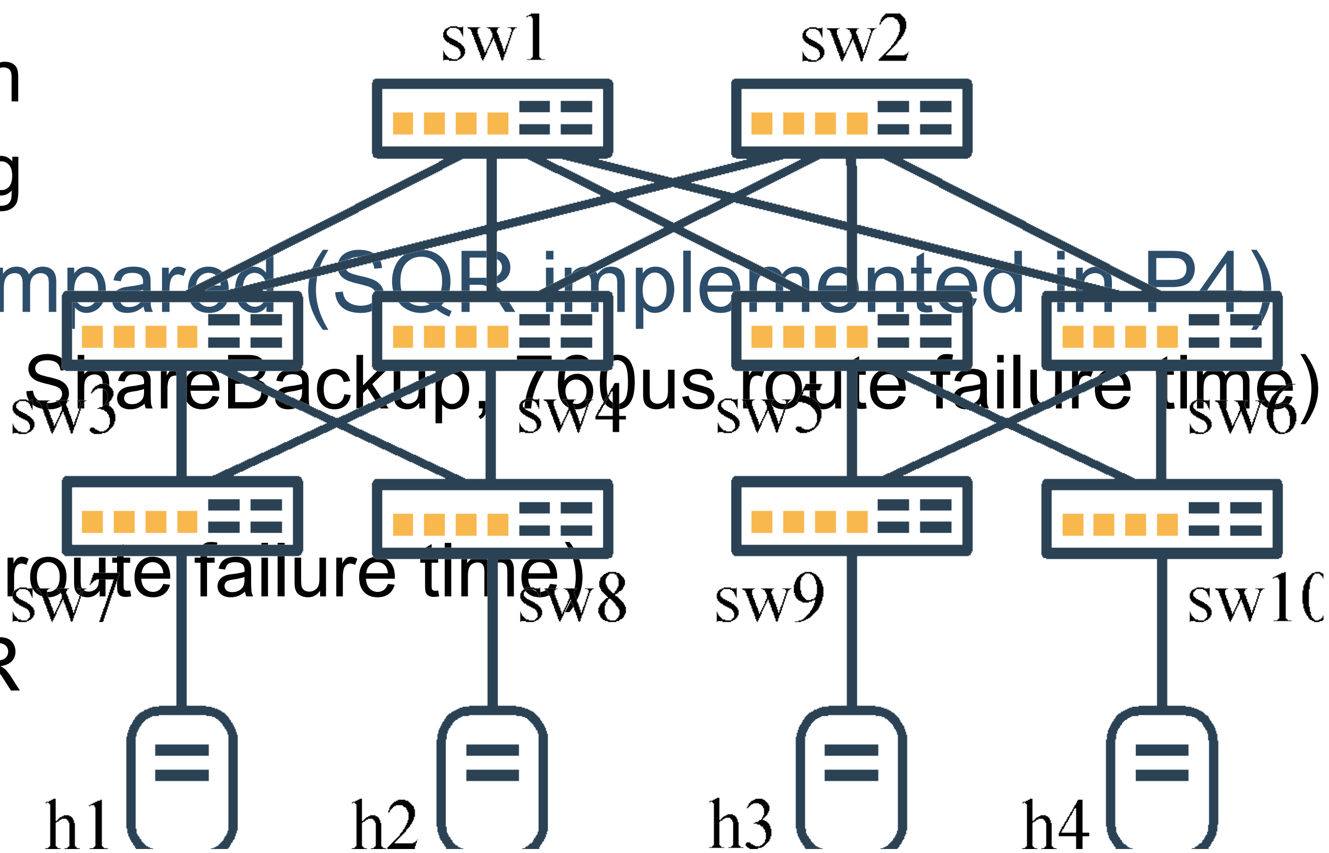
Compare PktTag with NextPktTag

Why it works

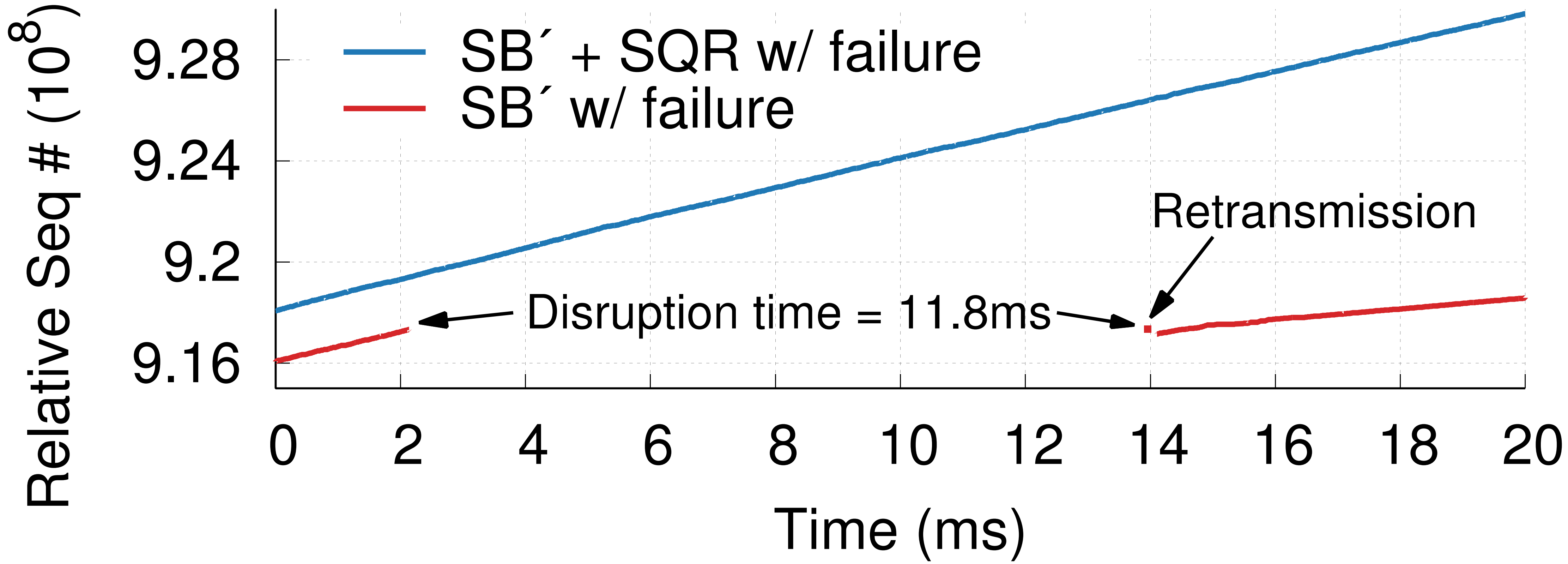
- No packet loss
  - ✓ Cache a copy of sent packets for a duration at least equal to the route failure time
  - ✓ Pkt is sent to backup port if new route is ready
- Packets in order
  - ✓ Recover lost pkts based on pkt tag
- Minimize egress processing delays on other flows going through the switch
  - ✓ Select caching queue from multiple ports
  - ✓ Dynamic least loaded port selection
- Complements existing methods of link failure detection and route reconfiguration

Evaluation

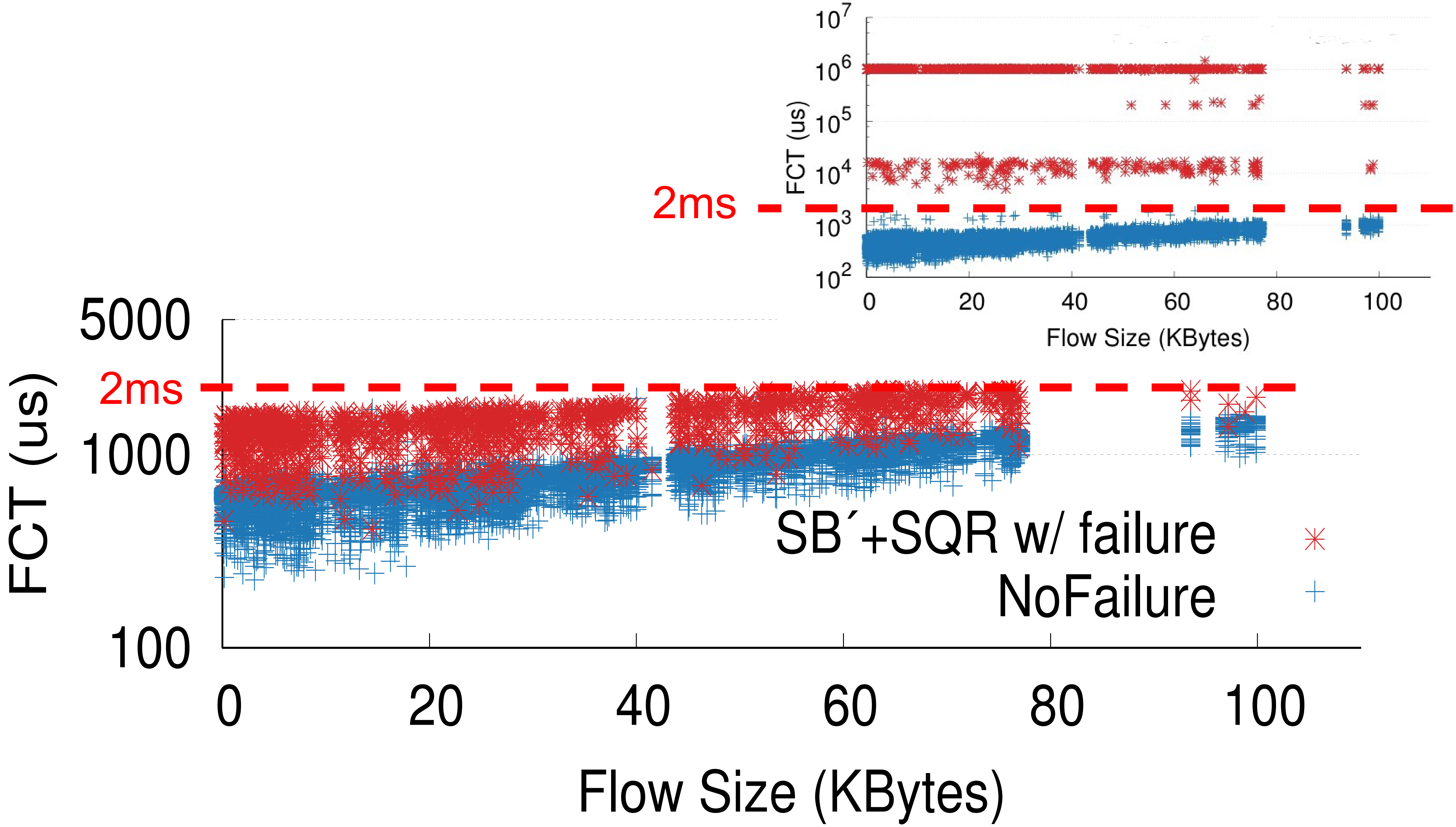
- Hardware Testbed
  - Barefoot Tofino switch
  - Intel Xeon servers equipped with Intel X710 NICs
- Trace
  - Web search
  - Data mining
- Schemes compared (SQR implemented in P4)
  - SB' (simple ShareBackup, 760us route failure time)
  - SB' + SQR
  - LRR (30us route failure time)
  - LRR + SQR



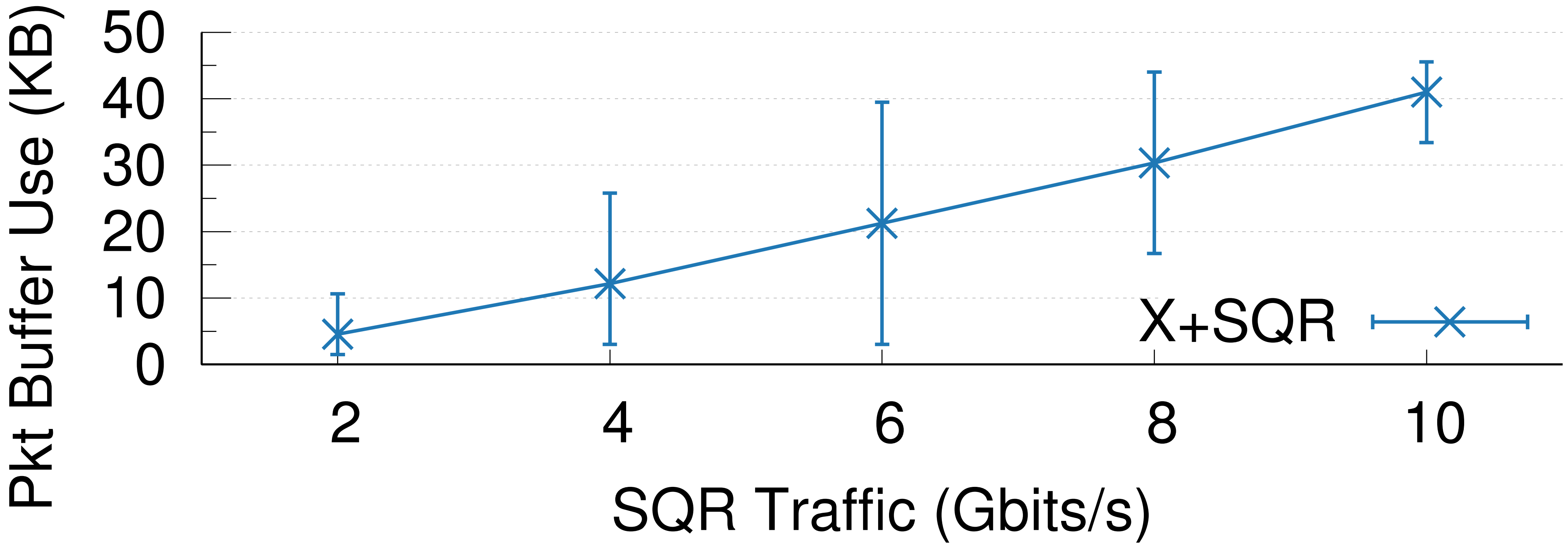
SQR masks link failures from end-point transport



SQR achieves low FCT under link failure



Overhead: Buffer size



Steady-state packet buffer consumption with 30us link failure detection time

Conclusion

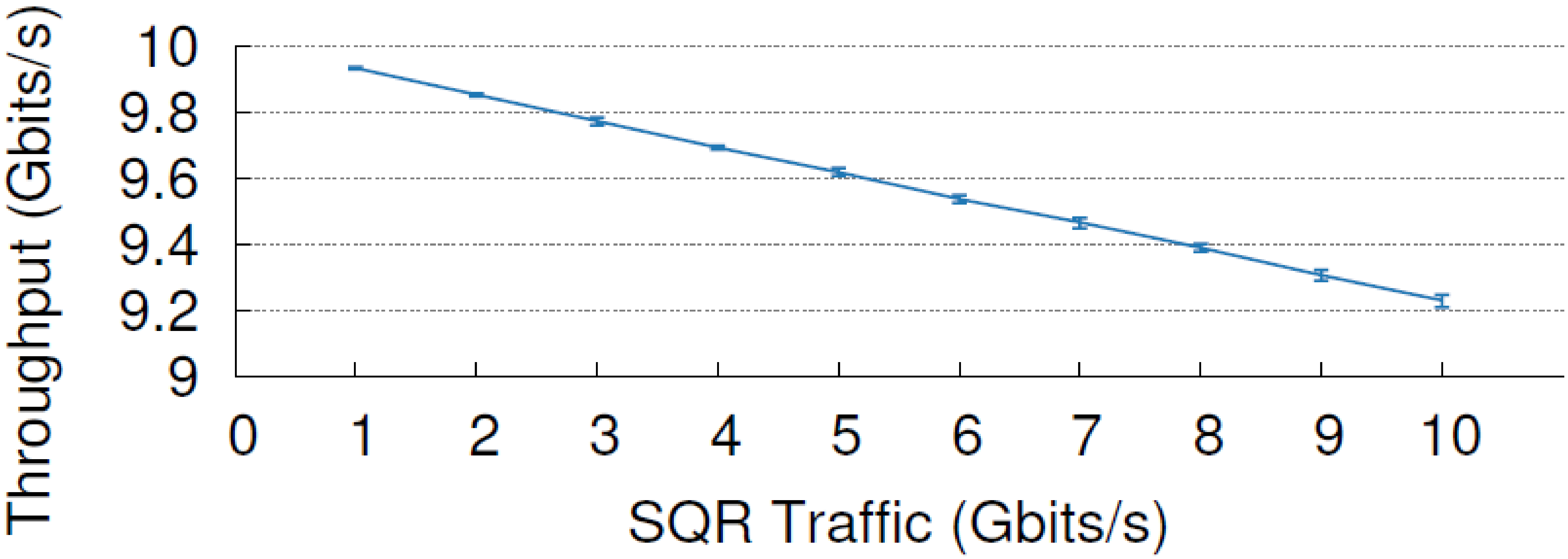
- Design SQR an In-Network packet loss recovery method which keeps FCT low for latency-sensitive flows when there is link failure.

- Eliminate packet loss during link failures and enables handing-off flows seamlessly to alternative paths.

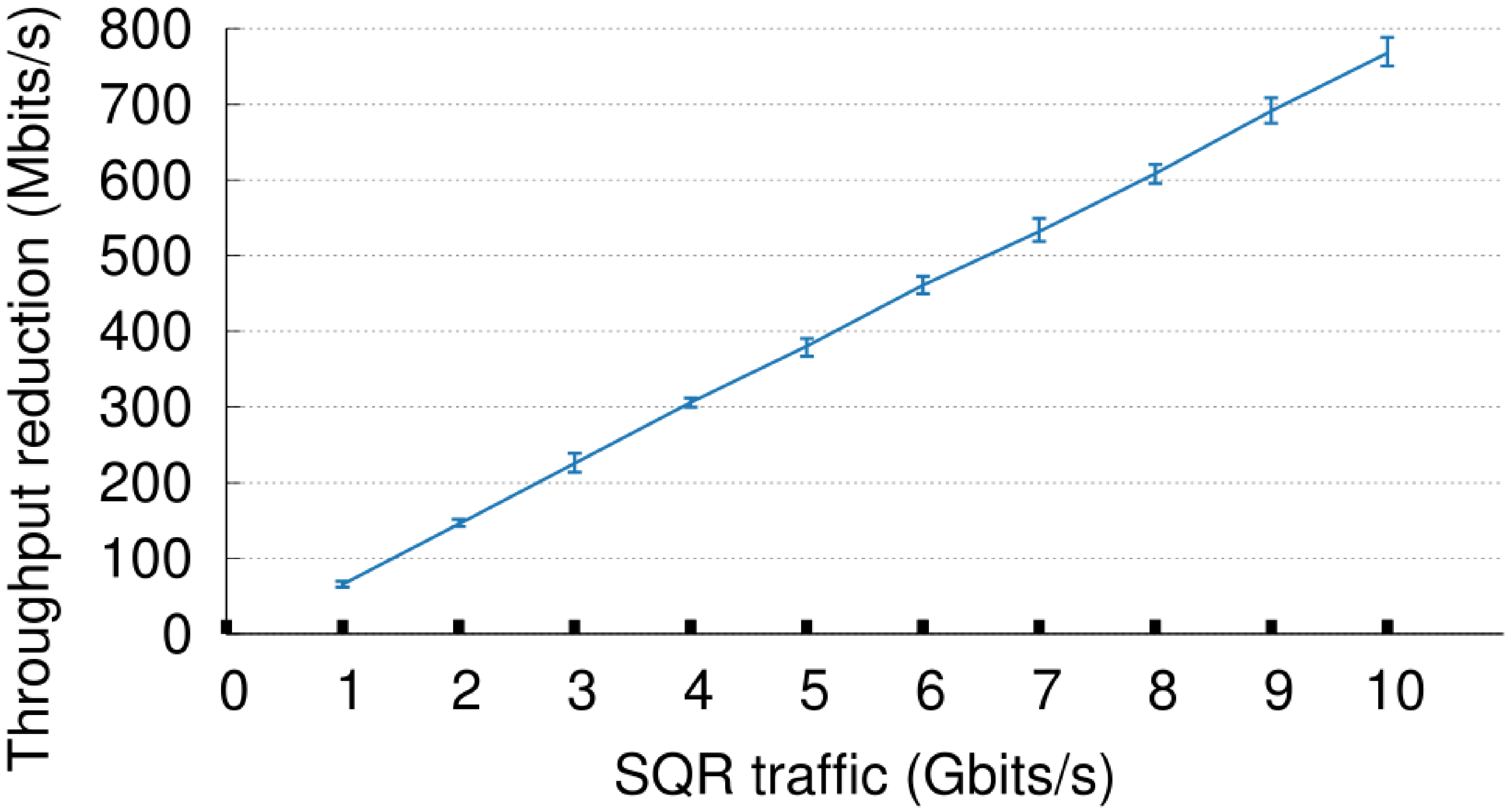
- SQR can be implemented on any programmable ASIC based on Portable Switch Architecture (PSA)

Thank
You
köszönöm
Kiitos
Asante
Shukria
Vinaka
Maake
Dhanyavadagalu
감사합니다
Dankscheen
спасибо
Kam Sah Hammida
شكرا
Manana
Dankon
Diolch i Chi
Terima Kasih
Matondo
Taiku
Tack
Dank Je
Mauruuru
Biyan
Blagodaram
Ngiyabonga
Dziekuje
Chokrane
Arigato
Grazie
Juspaxar
Mochchakkeram
Gracias
Paldies
Tingki
நன்றி
धन्यवाद
cảm ơn bạn
Kop Khun Khap
Gratias Tibi
Ua Tsaug Rau Koj
Bedankt
Ďakujem
Grazas
Di Ou Mèsi
Kia Ora
Obrigado
Děkuji
Nirringrazzjak
ありがとう
Suksama
Rahmat
Matur Nuwun
谢谢
хвала
Hvala
Welalin
Djiere Dieuf
Danke
Merci
Go Raibh Maith Agat
Najis Tuke
Eskerrik Asko
Misaotra
Salamat
ขอขอบคุณคุณ

Impact of SQR Traffic



Overhead: Egress processing


Throughput reduction (Mbits/s)
SQR traffic (Gbits/s)