IEEE HPSR 2023
Albuquerque, NM

Do Switches Still Need to Deliver Packets in Sequence?

Ufuk Usubütün, Fraida Fund, Shivendra S. Panwar

NYU TANDON SCHOOL OF ENGINEERING
NYU WIRELESS
CENTER FOR ADVANCED TECHNOLOGY IN TELECOMMUNICATIONS

Conventional wisdom:

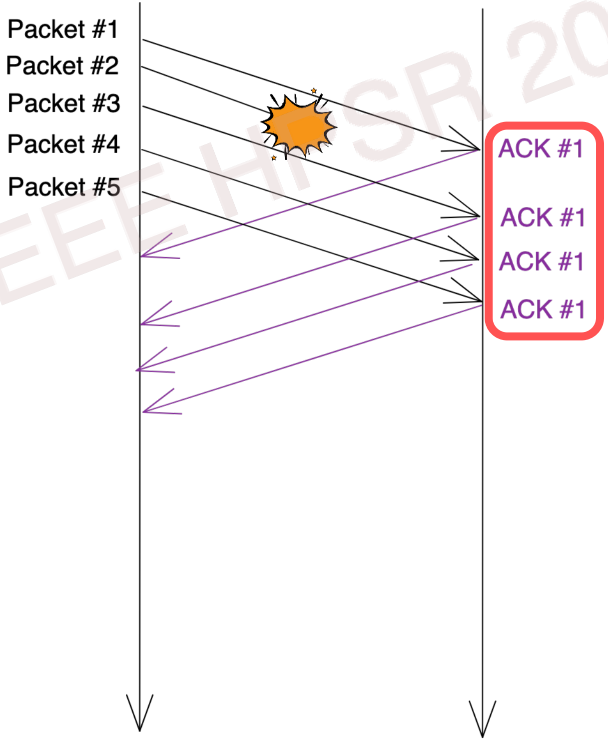
Packet reordering should be avoided wherever possible

Ufuk Usubutun et. al. - IEEE HPSR 2023

How did Classical TCP Detect Packet Loss?

Triple Duplicate ACK Rule

- Counts repeated ACKs



Ufuk Usubutun et. al. - IEEE HPSR 2023

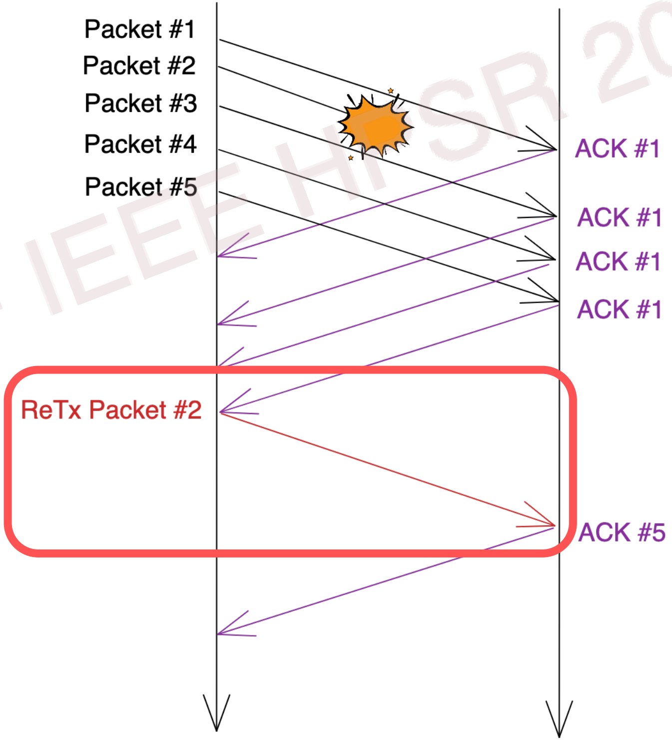
How did Classical TCP Detect Packet Loss?

Triple Duplicate ACK Rule

- Counts repeated ACKs

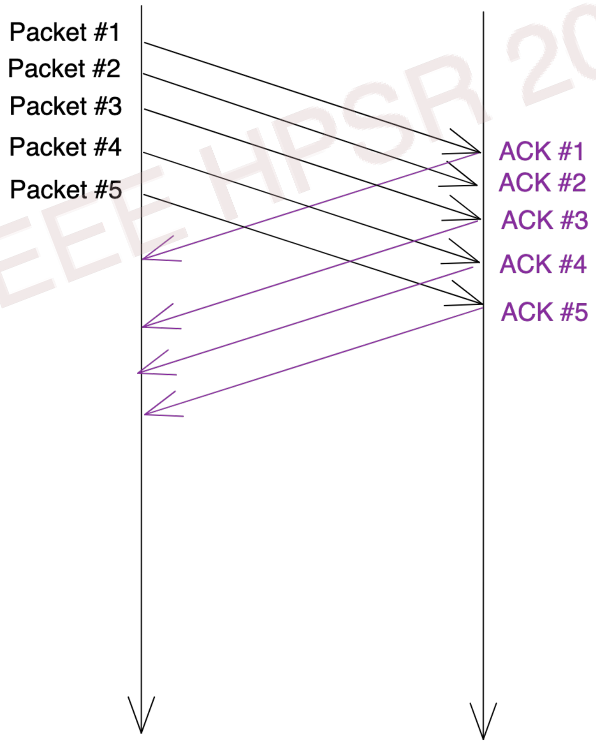
Loss detected:

- Retransmits lost packets
- Reduces *cwnd*



Ufuk Usubutun et. al. - IEEE HPSR 2023

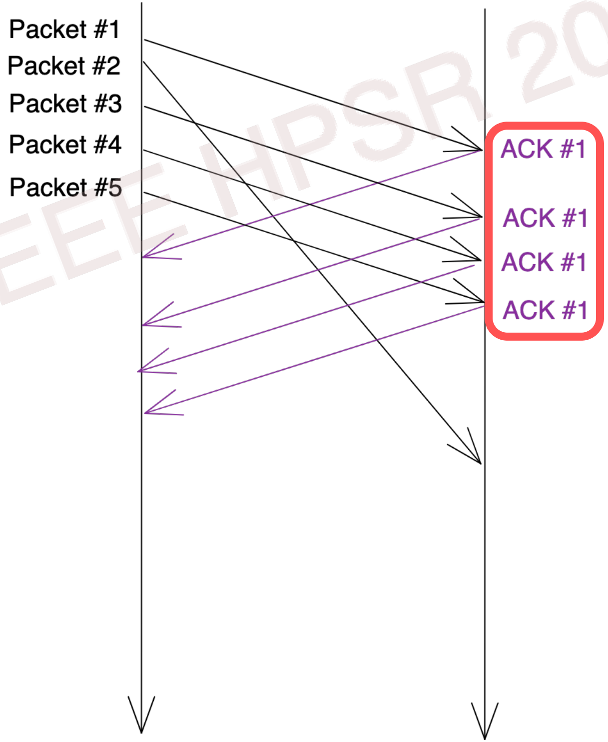
Assumption: Packets will arrive in sequence!



Ufuk Usubutun et. al. - IEEE HPSR 2023

Assumption: Packets will arrive in sequence!

- What happens if they don't?

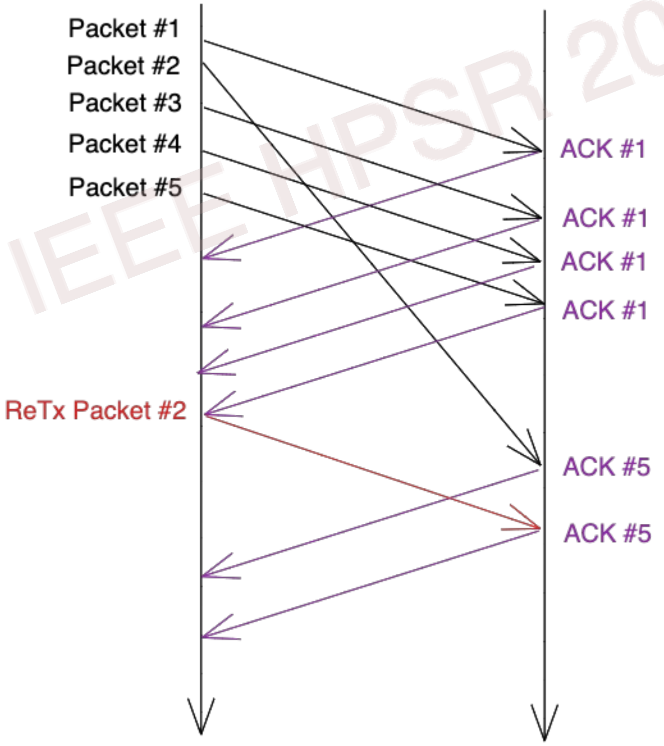


Ufuk Usubutun et. al. - IEEE HPSR 2023

Assumption: Packets will arrive in sequence!

- What happens if they don't?

Reordering is misinterpreted as loss!

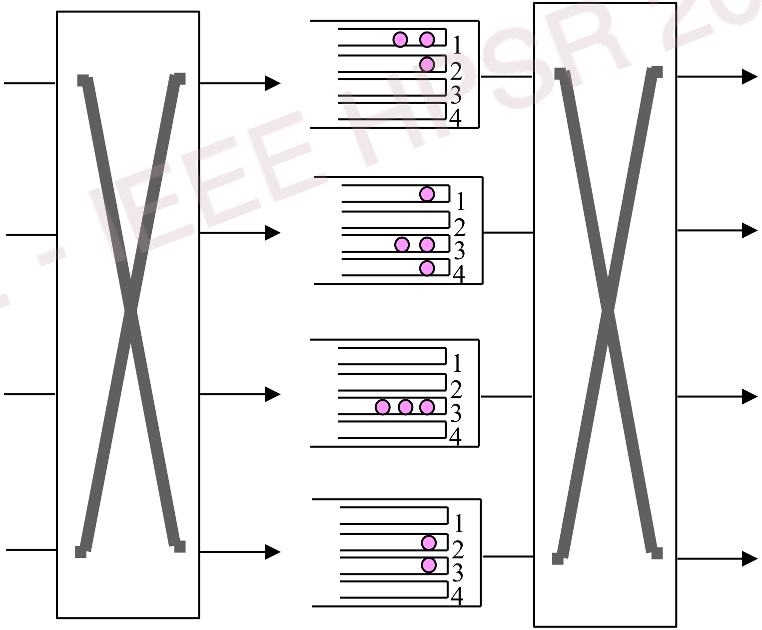


Ufuk Usubutun et. al. - IEEE HPSR 2023

Switches today are expected to deliver packets in-sequence

E.g.: Load-Balanced Birkhoff–von Neumann Switch:

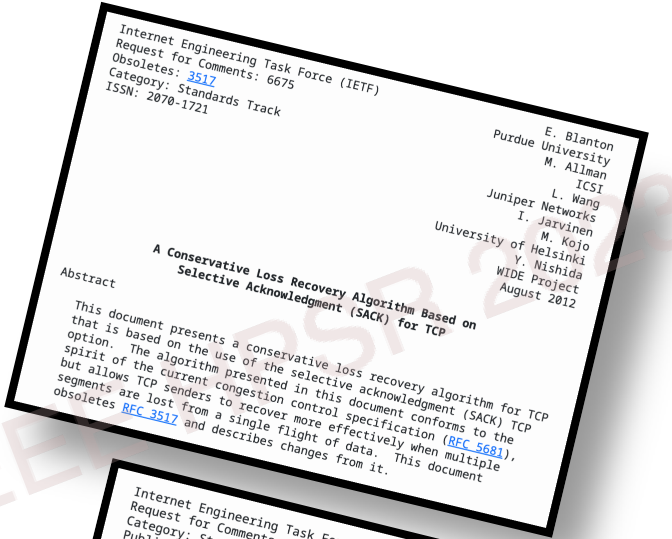
Was not adopted since it caused packet reordering



Ufuk Usubutun et. al. - IEEE HPSR 2023

In the past two decades: Two major changes

- Advanced loss detection algorithms for TCP widely deployed



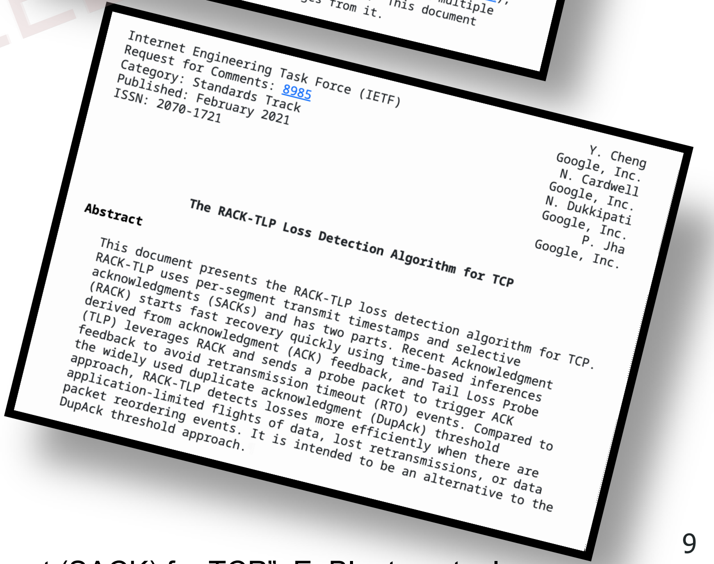
Internet Engineering Task Force (IETF)
Request for Comments: 6675
Obsoletes: 3517
Category: Standards Track
ISSN: 2070-1721

E. Blanton
Purdue University
M. Allman
ICSI
L. Wang
Juniper Networks
I. Jarvinen
M. Kojo
University of Helsinki
Y. Nishida
WIDE Project
August 2012

A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP

Abstract

This document presents a conservative loss recovery algorithm for TCP that is based on the use of the selective acknowledgment (SACK) TCP option. The algorithm presented in this document conforms to the spirit of the current congestion control specification (RFC 5681), but allows TCP senders to recover more effectively when multiple segments are lost from a single flight of data. This document obsoletes RFC 3517 and describes changes from it.



Internet Engineering Task Force (IETF)
Request for Comments: 8985
Category: Standards Track
Published: February 2021
ISSN: 2070-1721

Y. Cheng
Google, Inc.
N. Cardwell
Google, Inc.
N. Dukkipati
Google, Inc.
P. Jha
Google, Inc.

The RACK-TLP Loss Detection Algorithm for TCP

Abstract

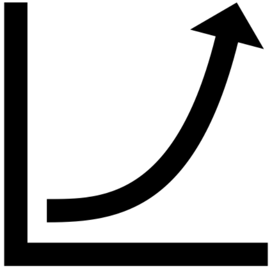
This document presents the RACK-TLP loss detection algorithm for TCP. RACK-TLP uses per-segment transmit timestamps and selective acknowledgments (SACKs) and has two parts. Recent Acknowledgment (RACK) starts fast recovery quickly using time-based inferences derived from acknowledgment (ACK) feedback, and Tail Loss Probe (TLP) leverages RACK and sends a probe packet to trigger ACK feedback to avoid retransmission timeout (RTO) events. Compared to the widely used duplicate acknowledgment (DupAck) threshold approach, RACK-TLP detects losses more efficiently when there are application-limited flights of data, lost retransmissions, or data packet reordering events. It is intended to be an alternative to the DupAck threshold approach.

Ufuk Usubutun et. al. - IEEE

RFC 8985, “The RACK-TLP Loss Detection Algorithm for TCP”, Y. Cheng et. al.
RFC 6675, “A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP”, E. Blanton et. al.

In the past two decades: Two major changes

- Advanced loss detection algorithms for TCP widely deployed
- Core network capacities grew from hundreds of Mbps to hundreds of Gbps



Ufuk Usubutun et. al. - IEEE HPSR 2023

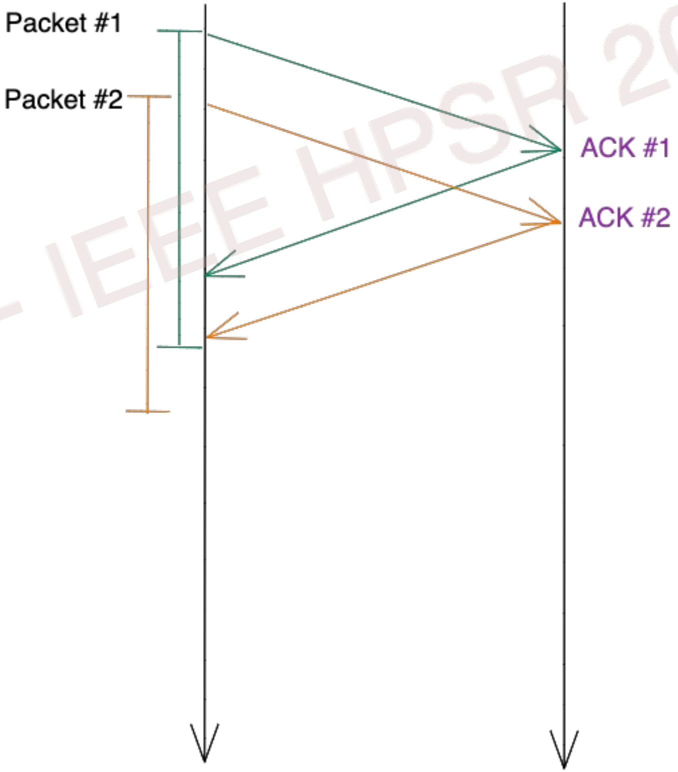
Advanced Loss Detection for TCP

Ufuk Usubutun et. al. - IEEE HPSR 2023

Temporal Loss Detection is now the default way!

RFC 8985 - *RACK*:

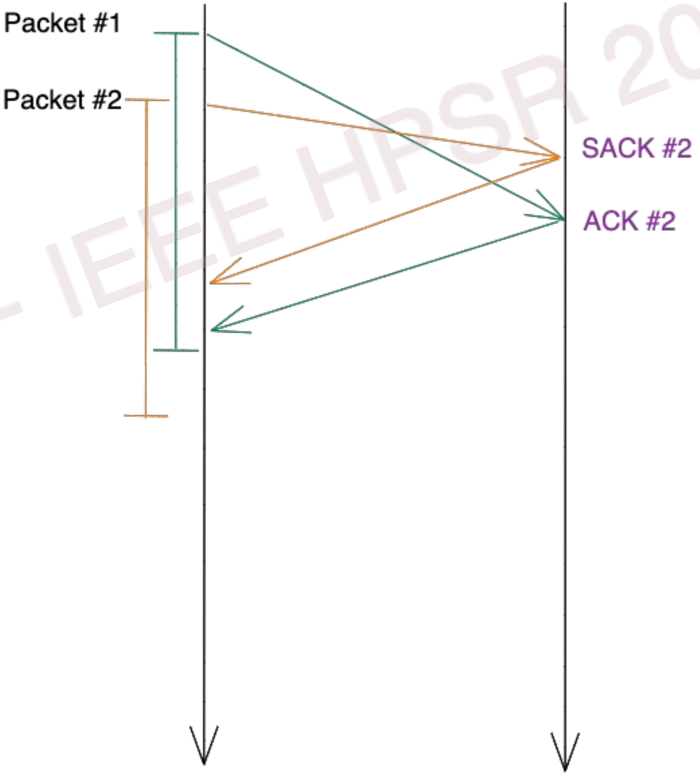
- Replaces ACK counting
- Lateness triggers loss detection.



Ufuk Usubutun et. al. - IEEE HPSR 2023

Arrival order is practically irrelevant

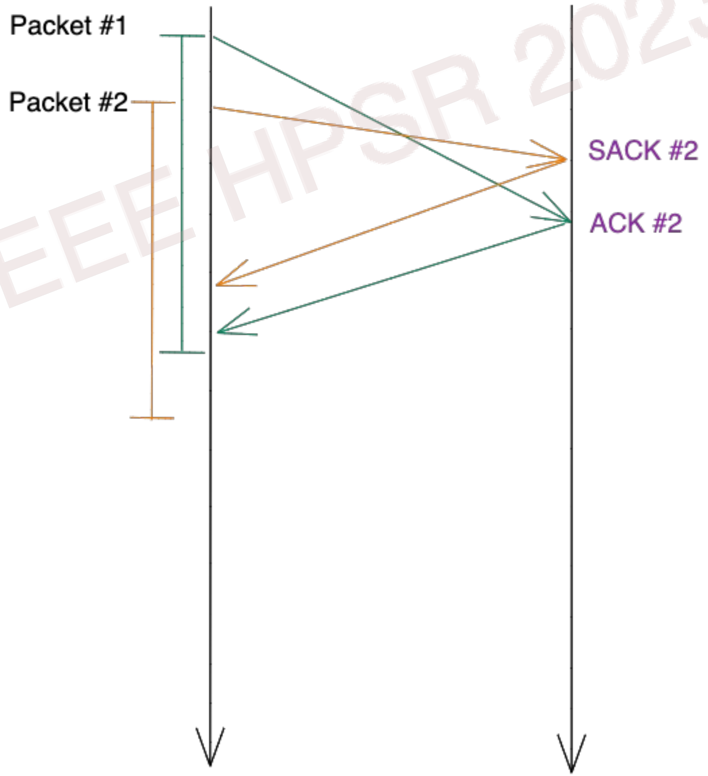
Tolerates reordering within time threshold!



Ufuk Usubutun et. al. - IEEE HPSR 2023

RACK Keeps a Dynamic Time Threshold

$$Time\ Threshold = SRTT + reo\_wnd$$

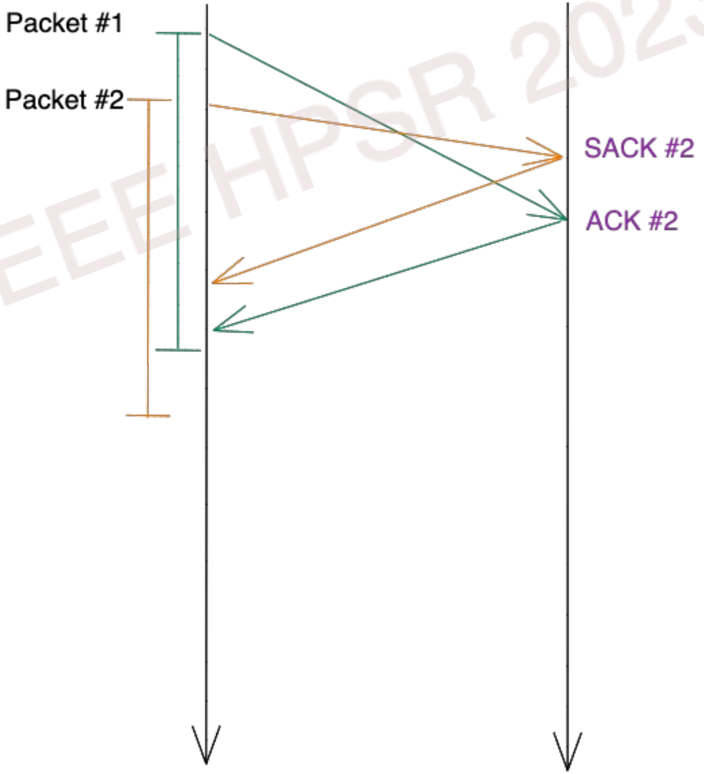


Ufuk Usubutun et. al. - IEEE HPSR 2023

RACK Keeps a Dynamic Time Threshold

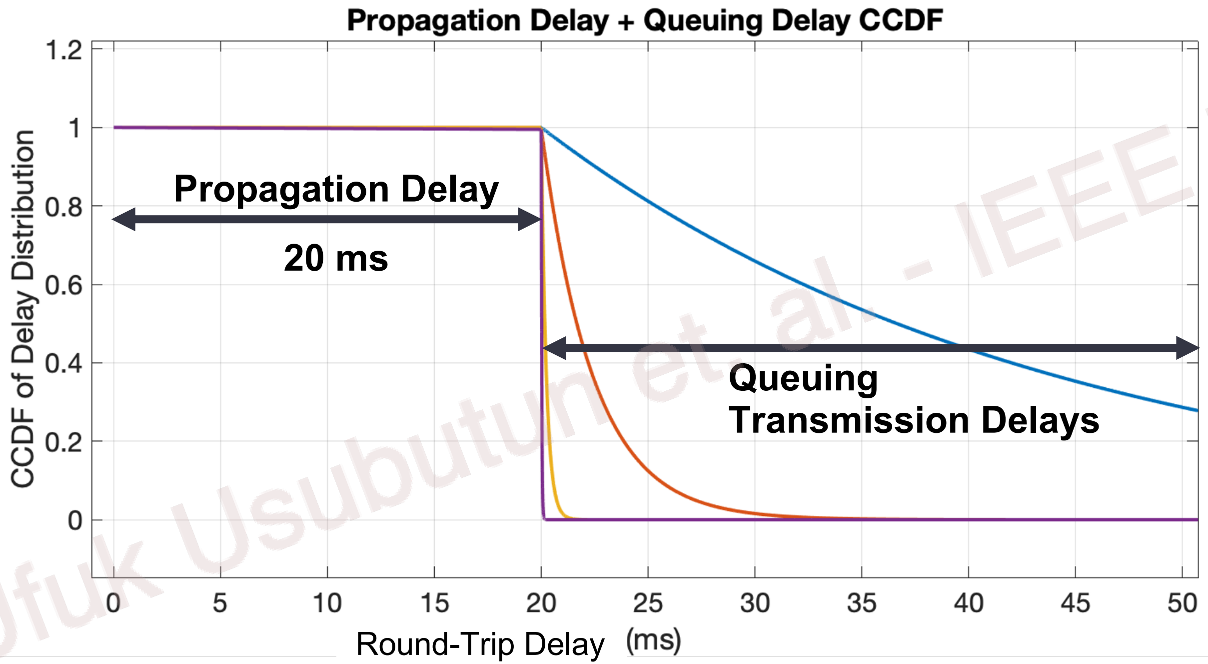
$$Time\ Threshold = SRTT + reo\_wnd$$

$$(\min\_RTT/4 < reo\_wnd < SRTT)$$



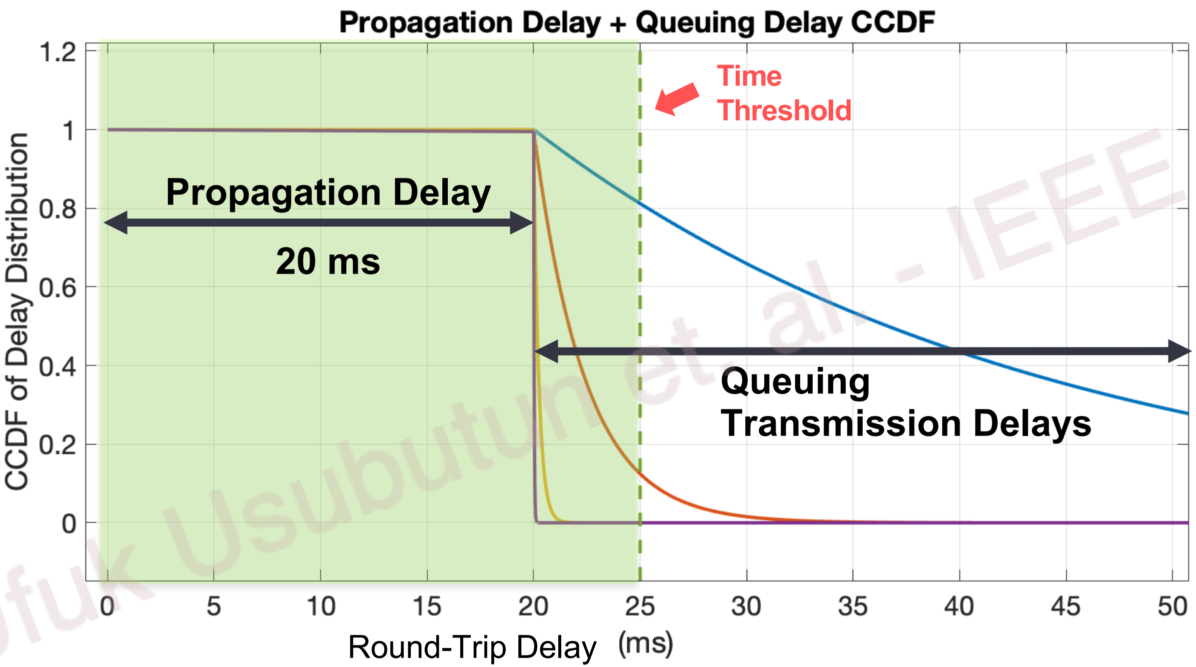
Ufuk Usubutun et. al. - IEEE HPSR 2023

Round-Trip Delay CCDF of Packets from 4 TCP Flows



Ufuk Usubutun et. al. - IEEE HPSR 2023

Round-Trip Delay CCDF of Packets from 4 TCP Flows

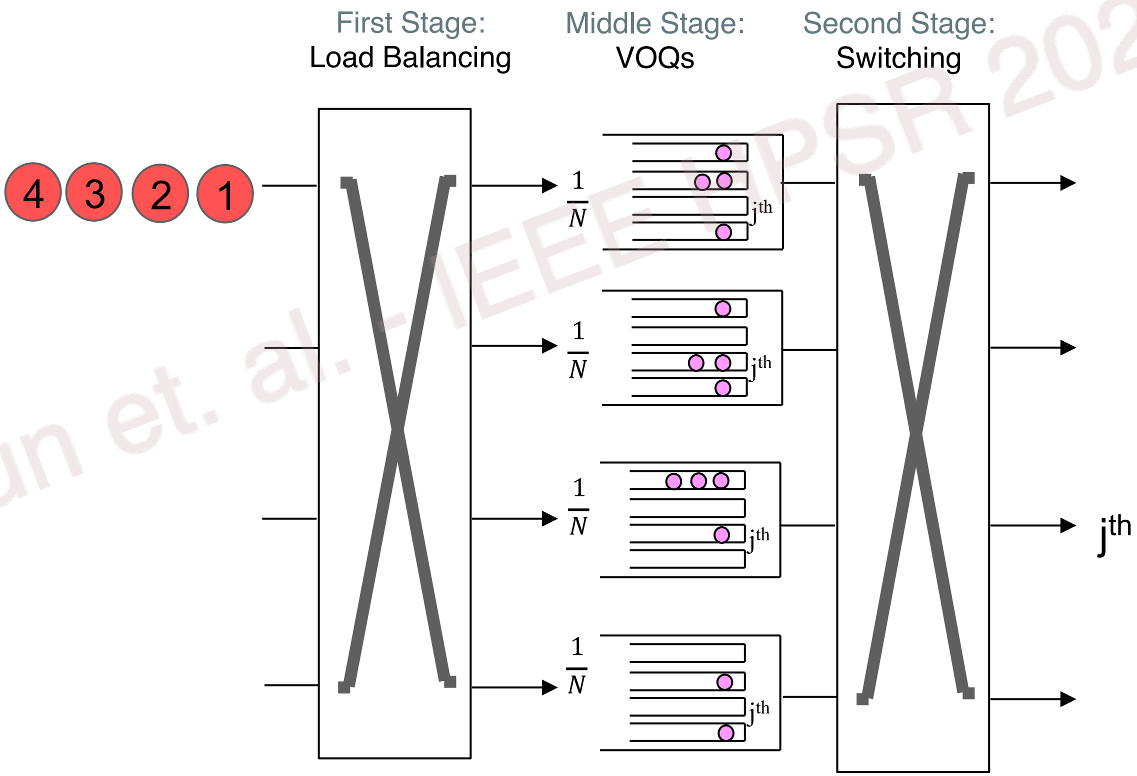


Low-Variation fits within the time window!

Ufuk Usubutun et. al. - IEEE HPSR 2023

Load-Balanced Switches

A load balancer spreads all incoming packets uniformly

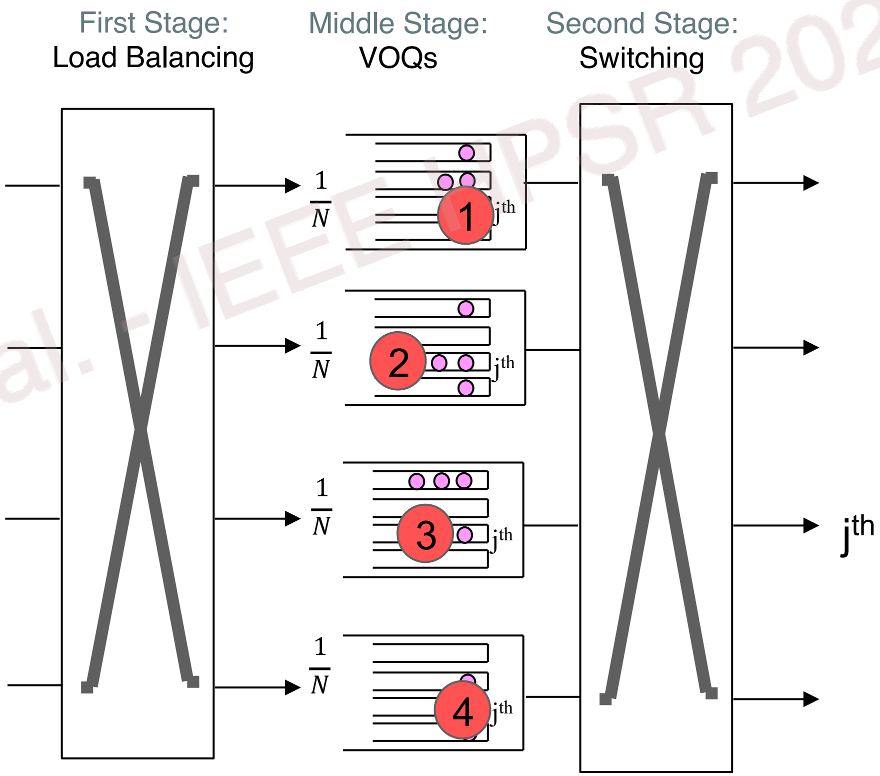


Ufuk Usubutun et. al. - IEEE HPSR 2023

Load-Balanced Switches

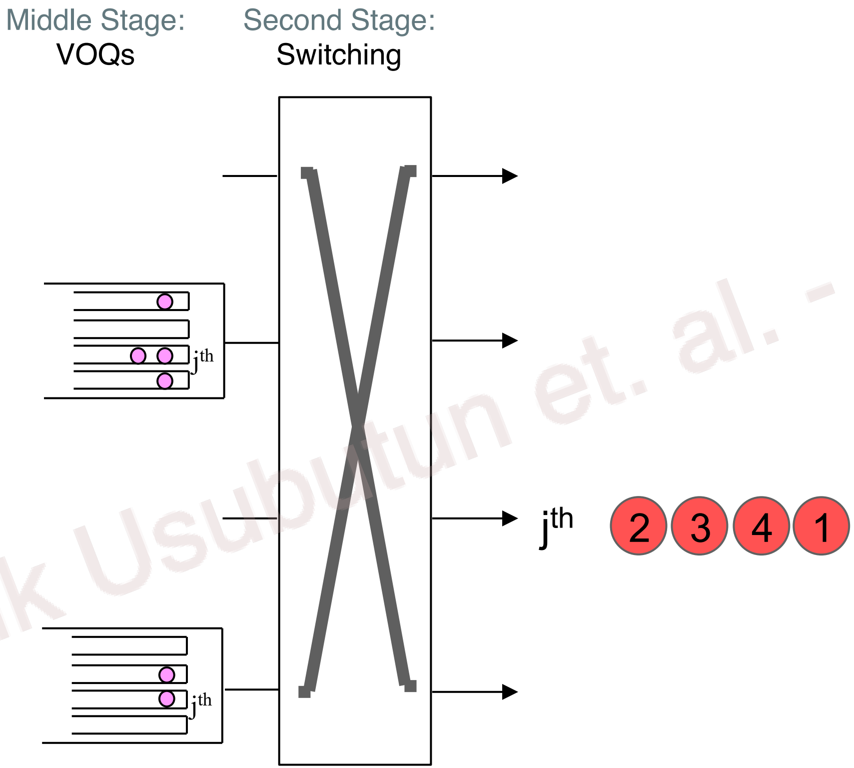
A load balancer spreads all incoming packets uniformly

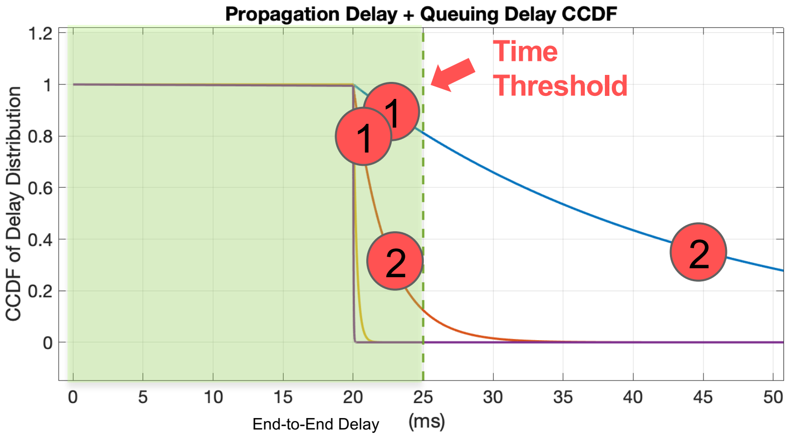
Packets from the same flow queue up at different VOQs



Ufuk Usubutun et. al. - IEEE HPSR 2023

Lateness determines loss detection performance





Ufuk Usubutun et. al. - IEEE HPSR 2023

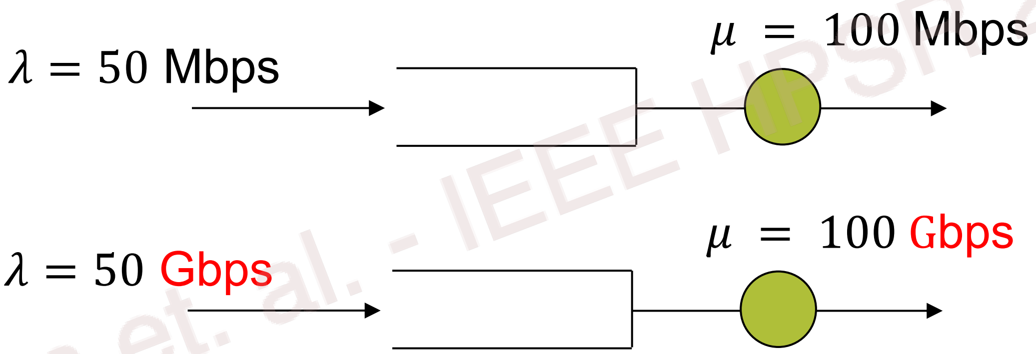
Effect of Network Capacity Growth

Ufuk Usubutun et. al. - IEEE HPSR 2023

Line rates increased from 100s of Mbps to 100s of Gbps

Then:

Now:



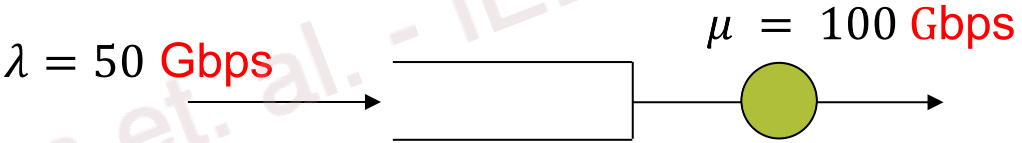
Both service rate μ and arrival rate λ are scaled up by a factor of $K = 1000$

Ufuk Usubutun et. al. - IEEE HPSR 2023

Line rates increased from 100s of Mbps to 100s of Gbps

Then:

Now:



Both service rate μ and arrival rate λ are scaled up by a factor of $K = 1000$

M/M/1 Queues: Mean Queue Occupancy does not change

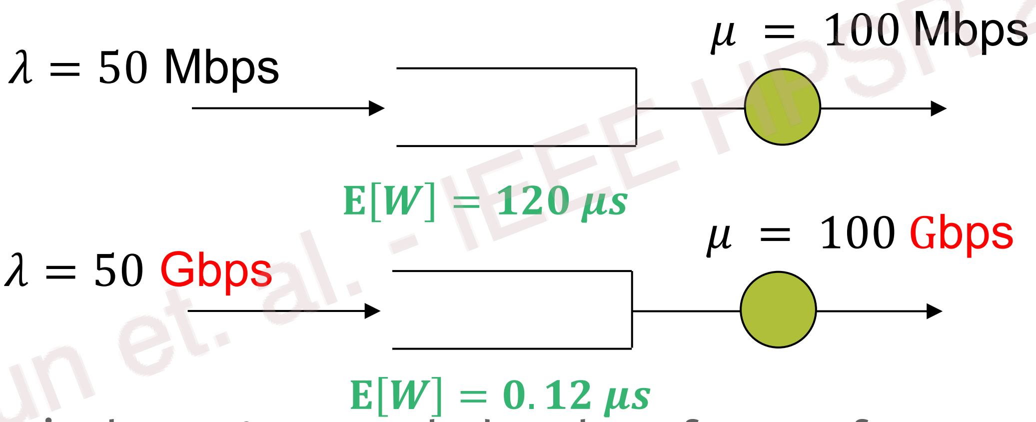
Mean Delay $\mathrm{E}[W]$: scaled down by K

Ufuk Usubutun et. al. - IEEE HPSR 2023

Line rates increased from 100s of Mbps to 100s of Gbps

Then:

Now:



Both service rate μ and arrival rate λ are scaled up by a factor of $K = 1000$

M/M/1 Queues: Mean Queue Occupancy does not change

Mean Delay $\mathrm{E}[W]$: scaled down by K

Ufuk Usubutun et. al. - IEEE HPSR 2023

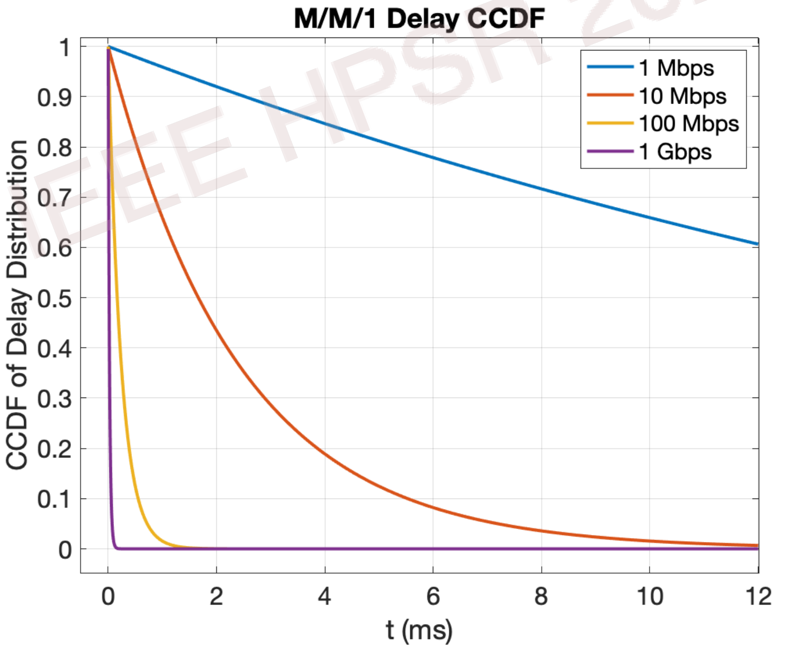
The delay distribution, too, shrinks with line rate increase

M/M/1 Queue
Tail Delay Probability:

$$P(W \geq \tau) = e^{-(\mu-\lambda)\tau} = e^{-K(\mu_0-\lambda_0)\tau}$$

If both μ and λ are scaled by a factor of K

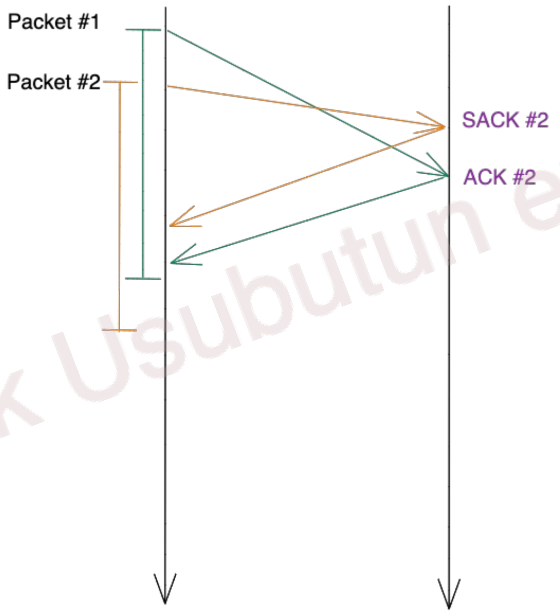
Tail probability is compressed by K



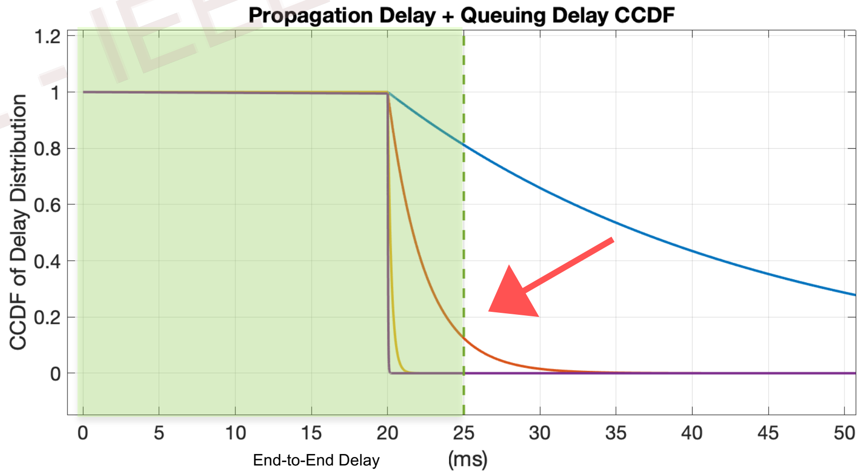
Ufuk Usubutun et. al. - IEEE HPSR 2023

RECAP: Two Things Have Changed

- Time based loss detection, based on lateness of individual packets, is widely deployed.



- Increasing line rates lead to smaller delay and delay variation at network core switches.



Ufuk Usubutun et. al. - IEEE HPSR 2023

Experimental Evaluation

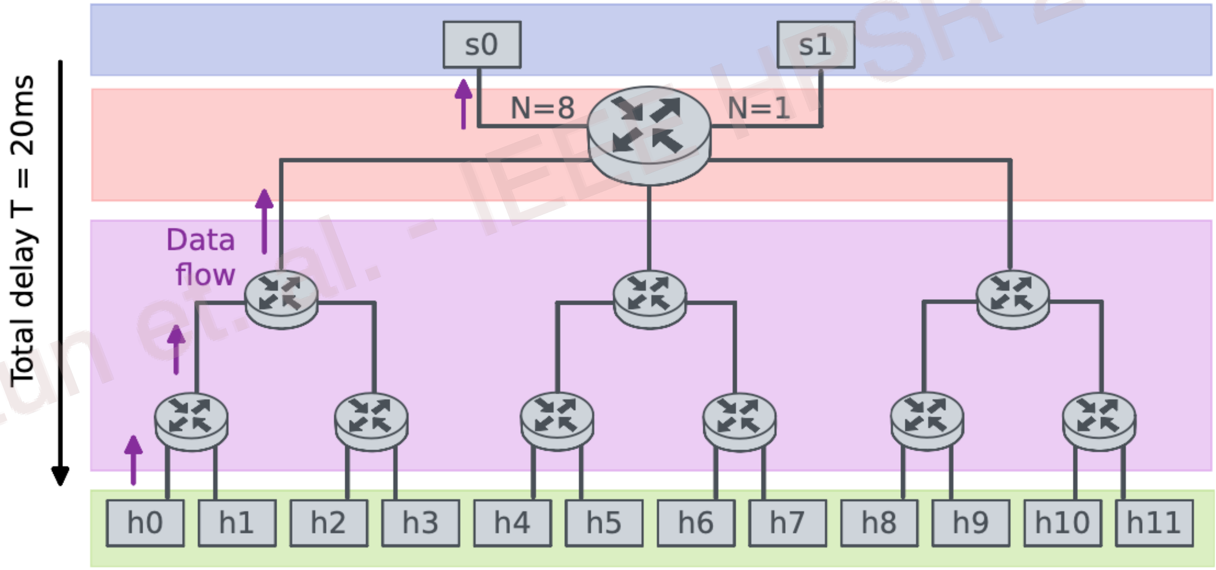
Ufuk Usubutun et. al. - IEEE HPSR 2023

GitHub Repo →

Testbed Experiments on Cloudlab

Goal:

Emulate a core network in a controlled environment



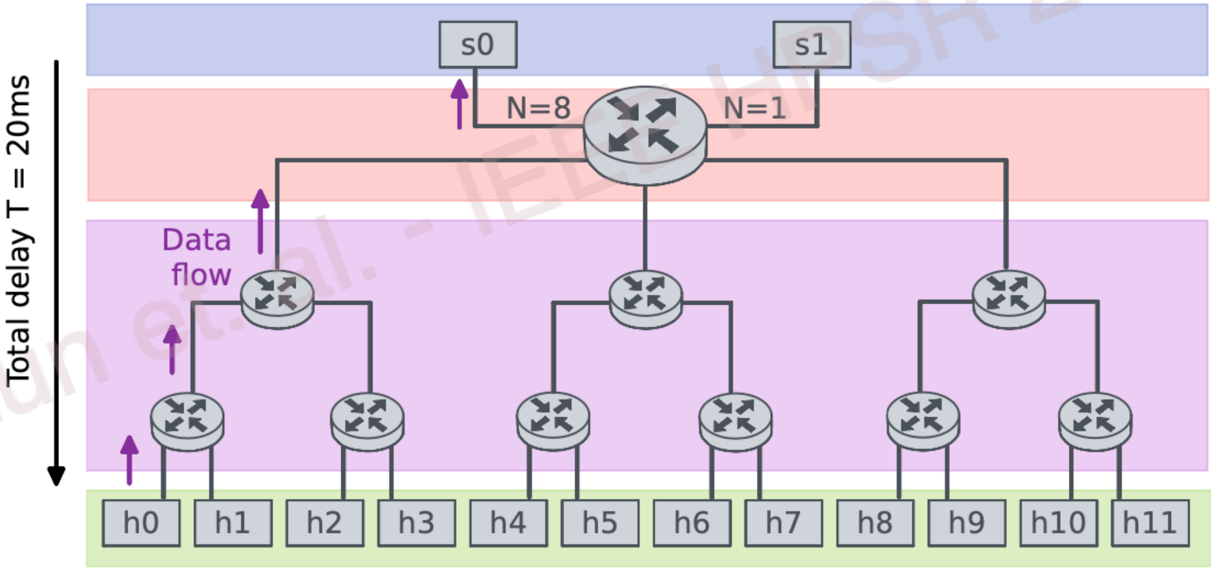
Ufuk Usubutun et al. - IEEE HPSR 2023

GitHub Repo →

Testbed Experiments on Cloudlab

***Goal*:**
Emulate a core network in a controlled environment

Thousands of TCP Cubic flows of different load sizes mixed through the topology

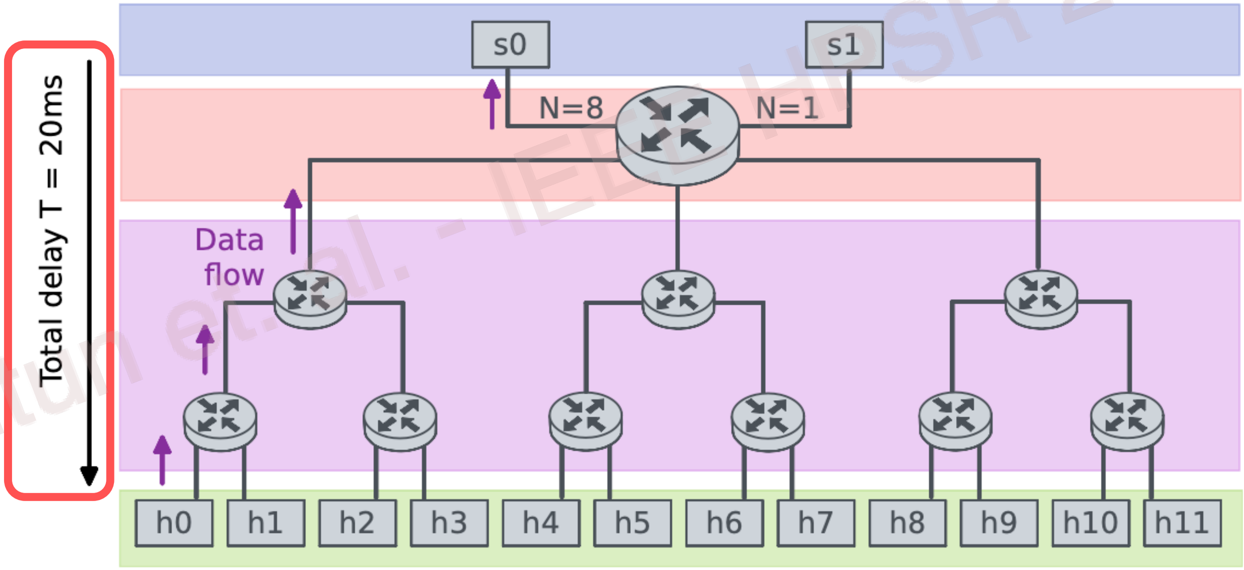


Ufuk Usubutun et al. - IEEE HPSR 2023

GitHub Repo →

Testbed Experiments on Cloudlab

Fixed base delay applied to reverse direction



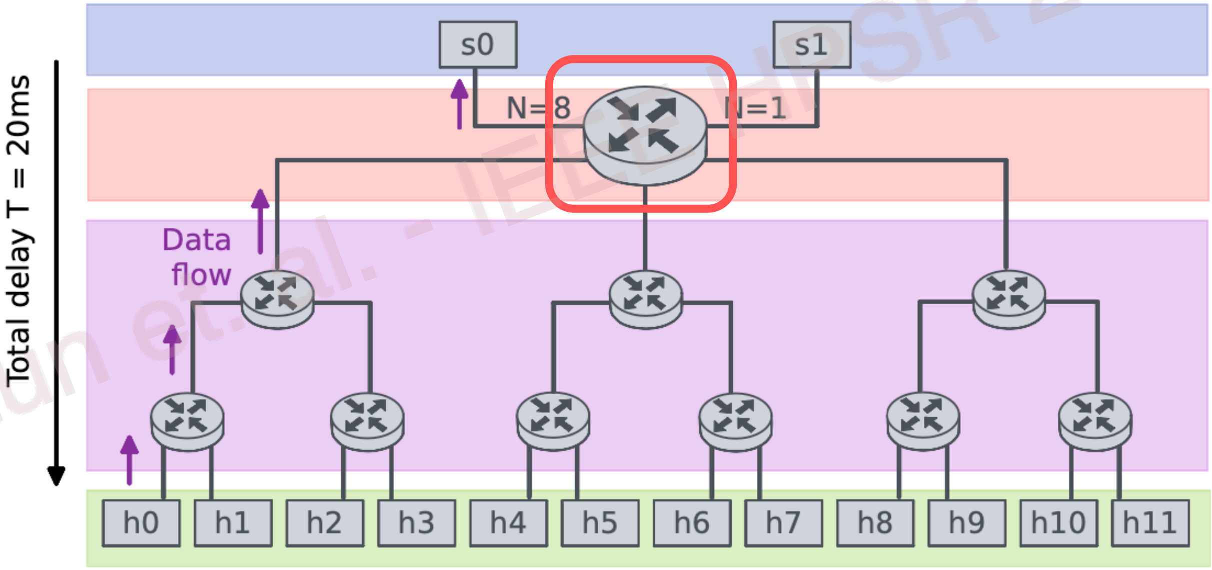
Ufuk Usubutun et al. - IEEE HPSR 2023

Testbed Experiments on Cloudlab

GitHub Repo →

Fixed base delay applied to reverse direction

Main switch was the bottleneck



Ufuk Usubutun et al. - IEEE HPSR 2023

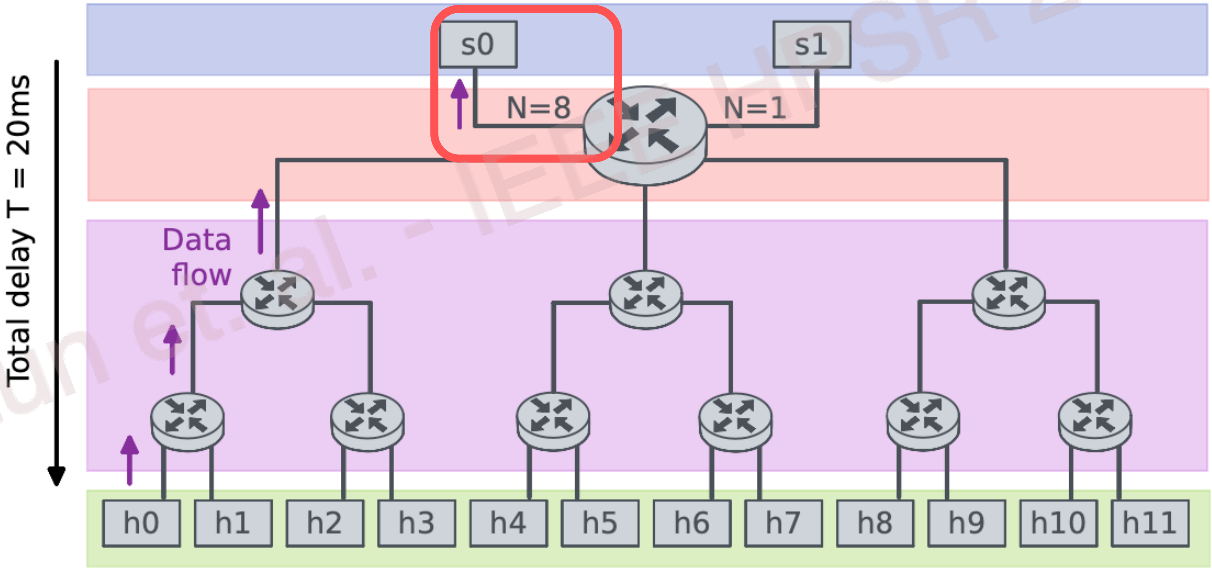
GitHub Repo →

Testbed Experiments on Cloudlab

Fixed base delay applied to reverse direction

Main switch was the bottleneck

The output to s0 had parallel queues



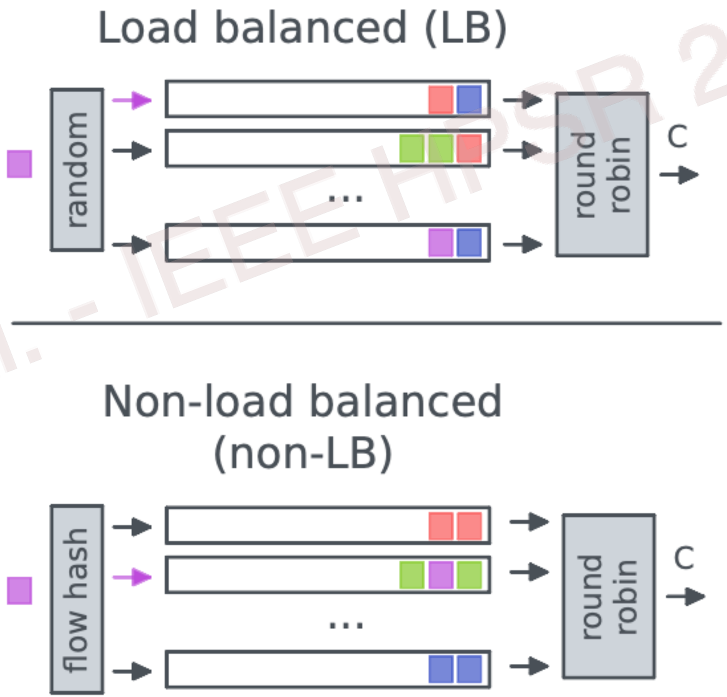
Ufuk Usubutun et al. - IEEE HPSR 2023

The experiment interface emulated a load balancer

***LB Configuration*:**
Probabilistic placement,
produces reordering

***non-LB Configuration*:**
Hashed placement,
No reordering

Both served in round-robin order.

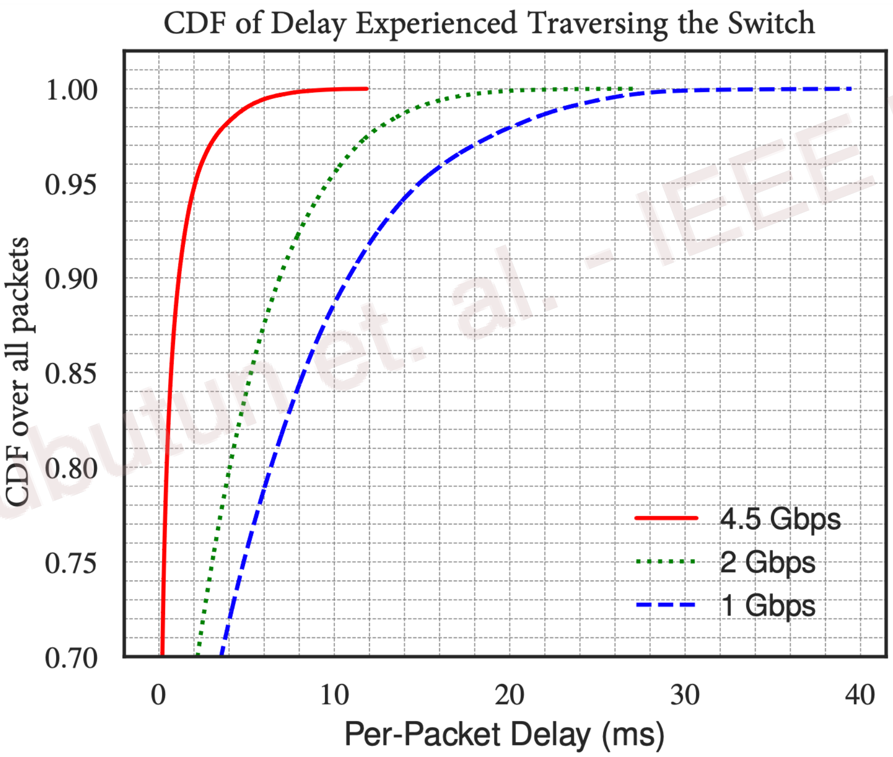


Ulukt Usubutun et. al. - IEEE HPSR 2023

Results

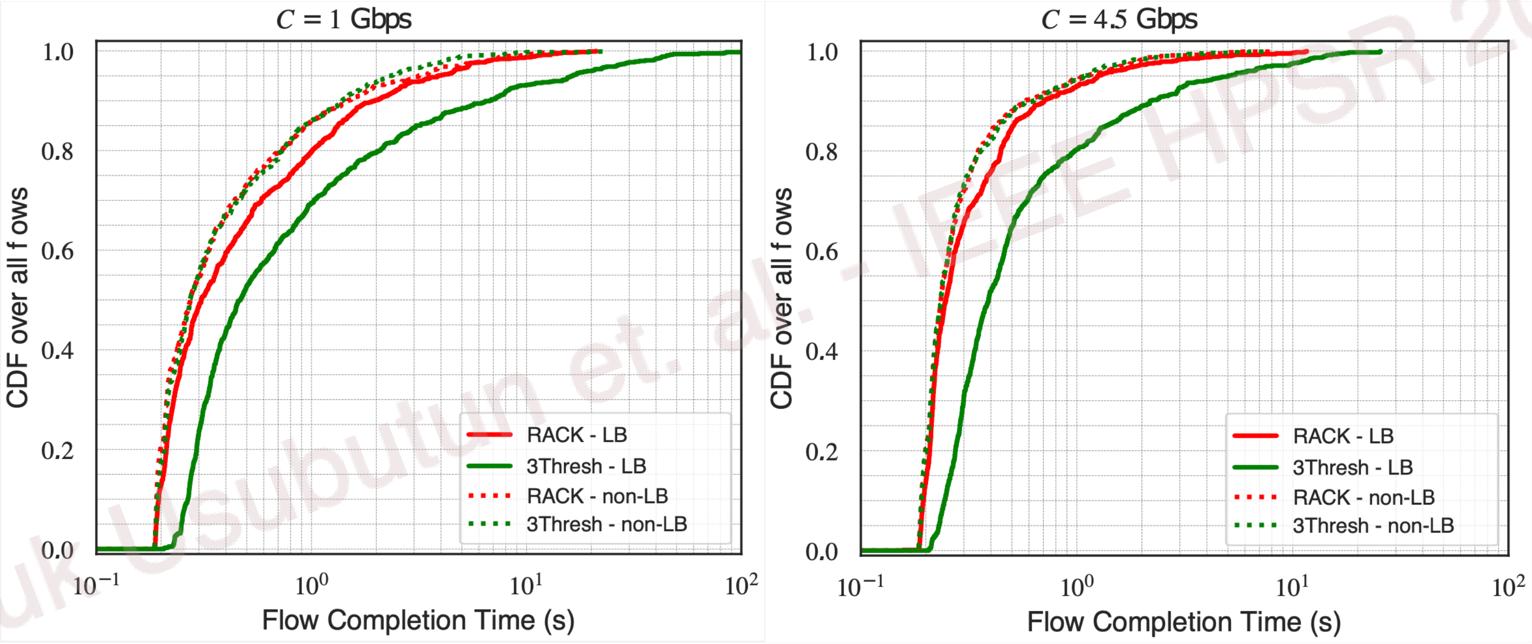
Ufuk Usubutun et. al. - IEEE HPSR 2023

Higher line rates result in narrower delay distributions through the reordering switch



Ufuk Usubutun et. al. - IEEE HPSR 2023

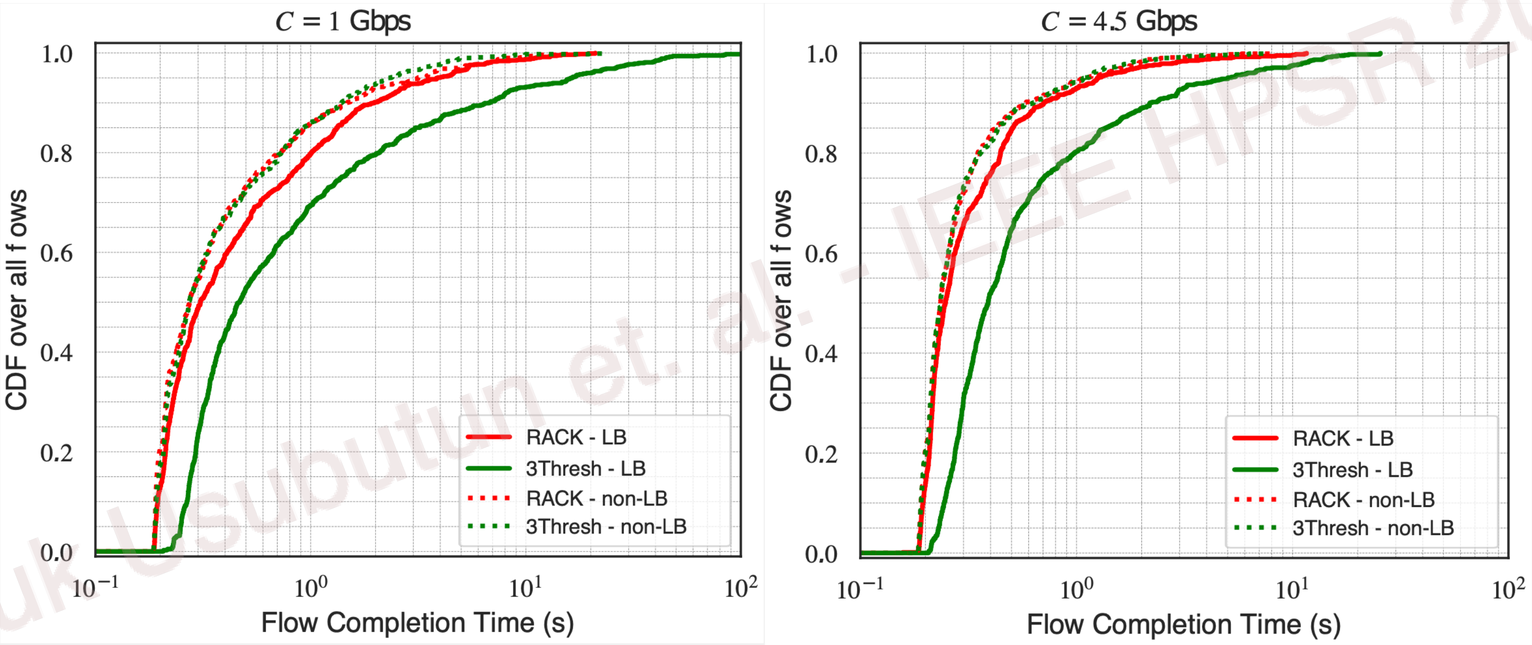
All algorithms perform the same without reordering (dashed lines)



Ufuk Usubutun et. al. - IEEE HPSR 2023

Approx (51 ± 4)% utilization at each scenario

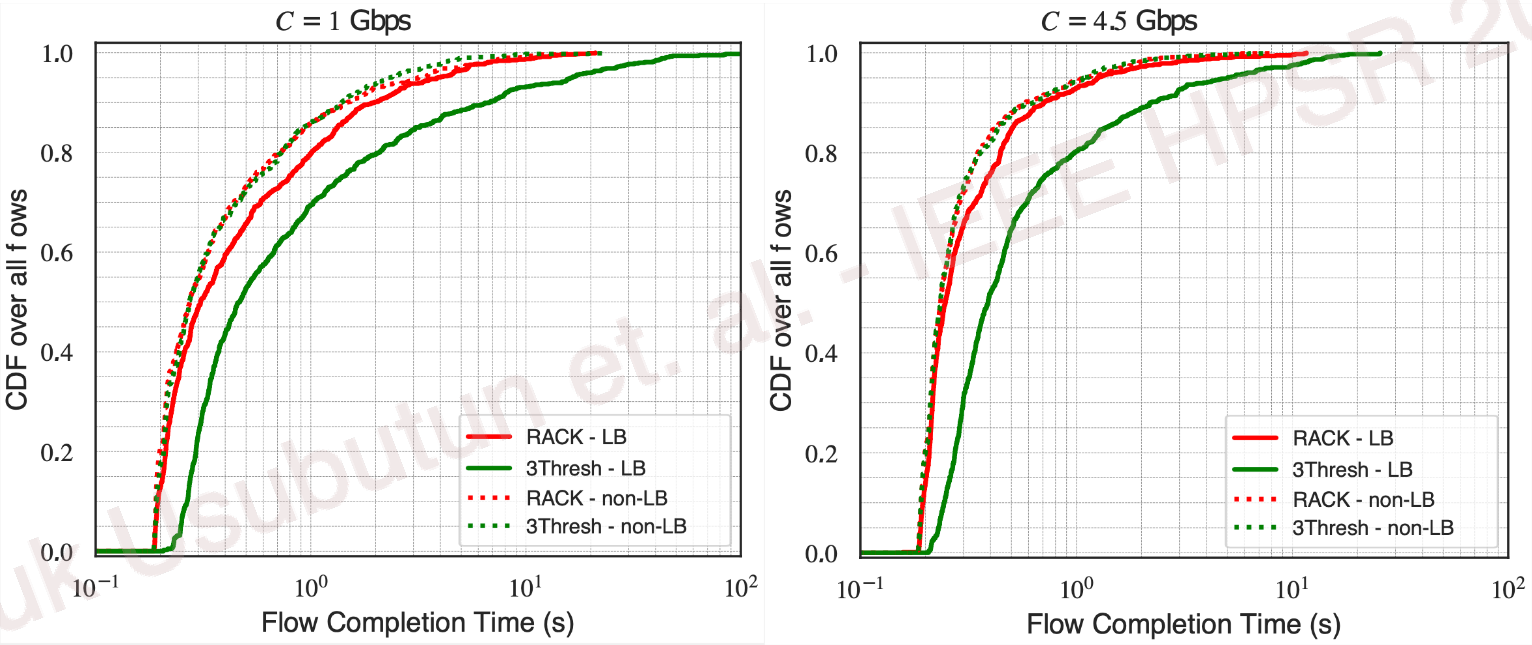
Triple Duplicate ACK does poorly under reordering



Ufuk Usubutun et. al. - IEEE HPSR 2023

Approx (51 ± 4)% utilization at each scenario

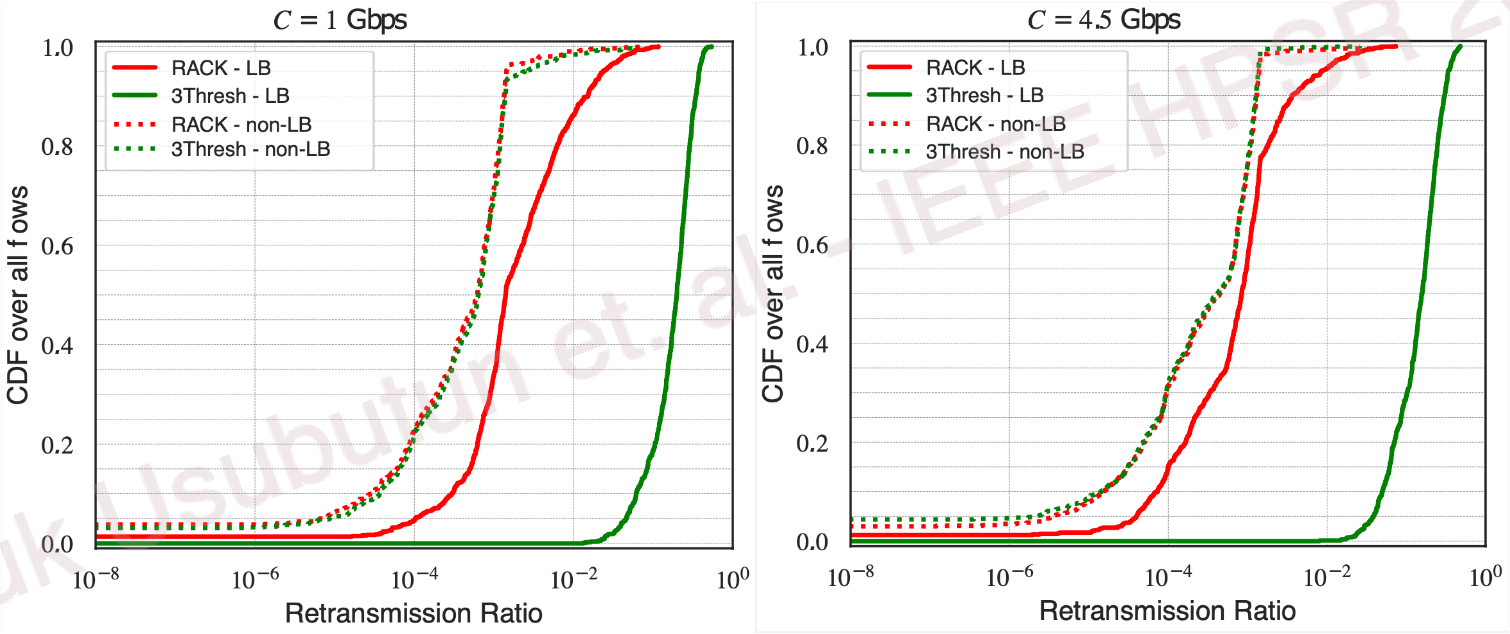
Higher line rates lead to better performance for RACK (time based)



Approx (51 ± 4)% utilization at each scenario

Ufuk Usubutun et. al. - IEEE HPSR 2023

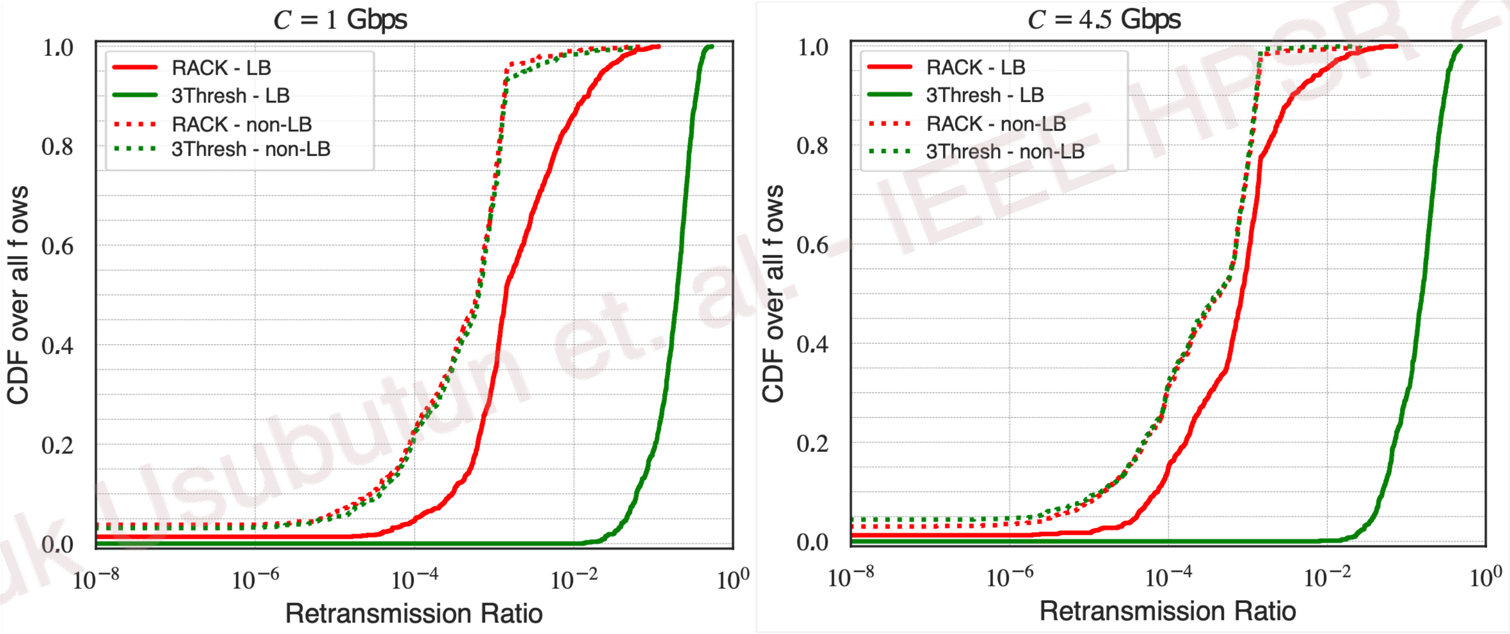
Triple Duplicate ACK retransmits significantly



Ufuk Usubutun et. al. - IEEE HPSR 2023

Approx (51 ± 4)% utilization at each scenario

High line rates results in less retransmissions for RACK (time based)



Ufuk Usubutun et. al. - IEEE HPSR 2023

Approx (51 ± 4)% utilization at each scenario

Conclusion

Ufuk Usubutun et. al. - IEEE HPSR 2023

Conclusions & Future Work

- Traditional wisdom on in-sequence delivery requirements for switches should be revisited

- This result also has implications for Data Center Networks and multi Radio Access Technology wireless systems.
- Similar implications for UDP based QUIC.
- The case of TCP BBR should be investigated

Ufuk Usubutun et. al. - IEEE HPSR 2023

GitHub Repo for Artifacts →

THANK YOU! QUESTIONS?

Ufuk Usubütün
usubutun@nyu.edu

Ufuk Usubutun et. al. - IEEE HPSR 2023

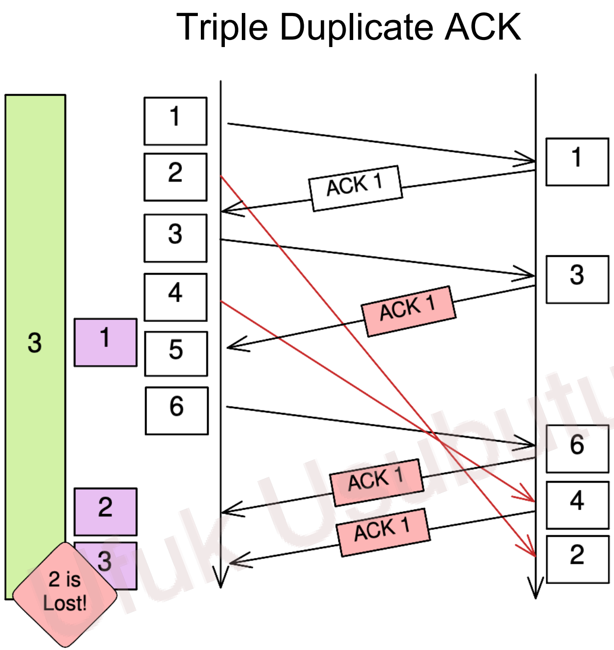
12/05/2019

Backup Slides

Ufuk Usubutun et. al. - IEEE HPSR 2023

NYU | TANDON SCHOOL OF ENGINEERING

A look into classical loss detection



Oguz Ucubutun et. al. - IEEE HPSR 2023

Evolved from 3 duplicate ACK

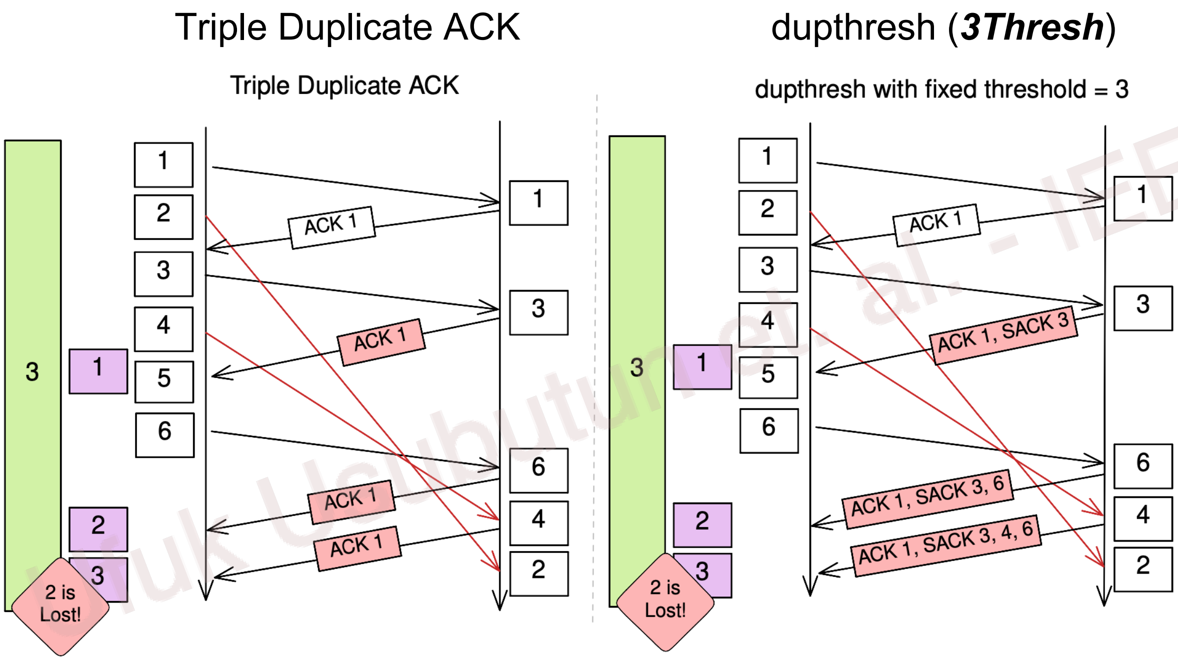
Selective ACK (SACK) and Duplicate SACK (DSACK) emerges.

- Expansion of the same ACK counting idea
  adaptive dupthresh

- Time based approach
  RACK

Ufuk Usubutun et. al. - IEEE HPSR 2023

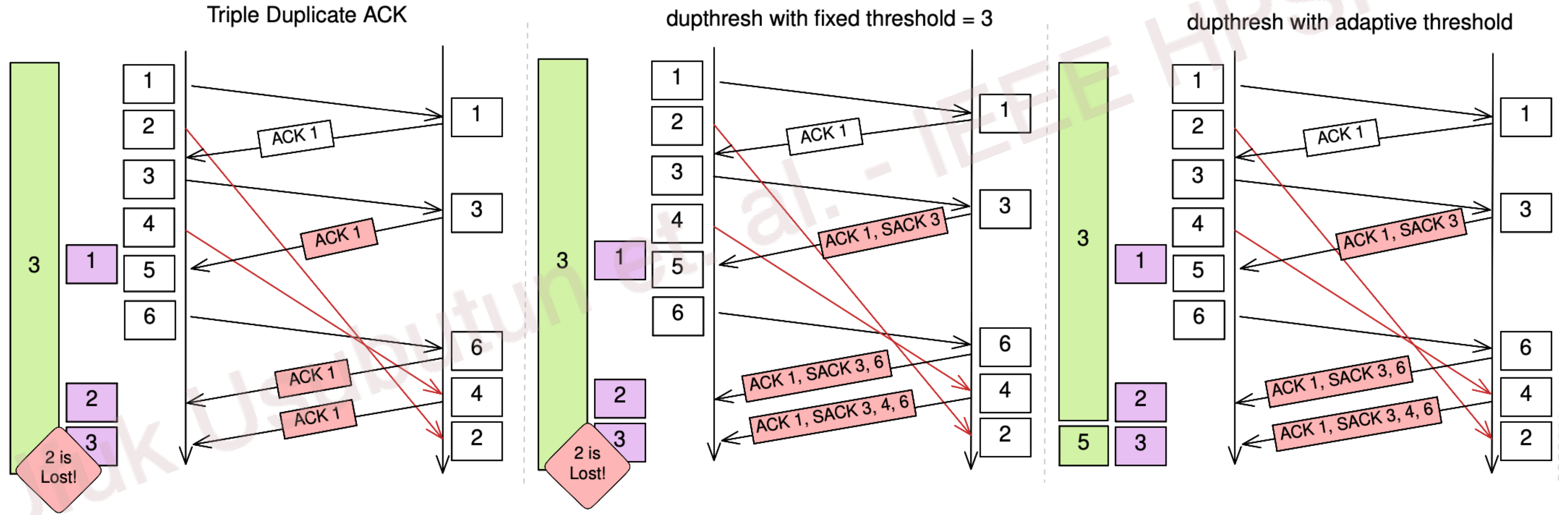
SACKs allow better knowledge of reception



Ufuk Usubutun et. al. - IEEE HPSR 2023

Adaptive mechanisms emerged

Triple Duplicate ACK | dupthresh (***3Thresh***) | adaptive dupthresh (***adapThresh***)



adapThresh gains resilience to reordering after an adaptation episode.

Ufuk Usubutun et al. - IEEE HPSR 2023

How to generate flows: Set up F flow generators at each node

At each generator:
Sample a flow size from a WAN TCP Traffic study (90% mice, 10% elephant)

At each generator:
Random wait between start of flows,
Always greater than flow completion time. Wait time scaled wrt capacity C.

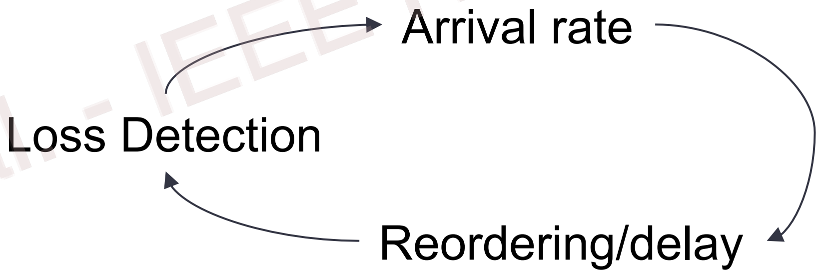
Aim: Keep the total bytes transferred independent of algorithm.

Ufuk Usubutun et. al. - IEEE HPSR 2023

Experimenting within a closed loop

We tested **different loss detection algorithms** at different line rates.

This results in a closed loop



We achieved Approx. (51 ± 4)% utilization at each scenario

Ufuk Usubutun et. al. - IEEE HPSR 2023