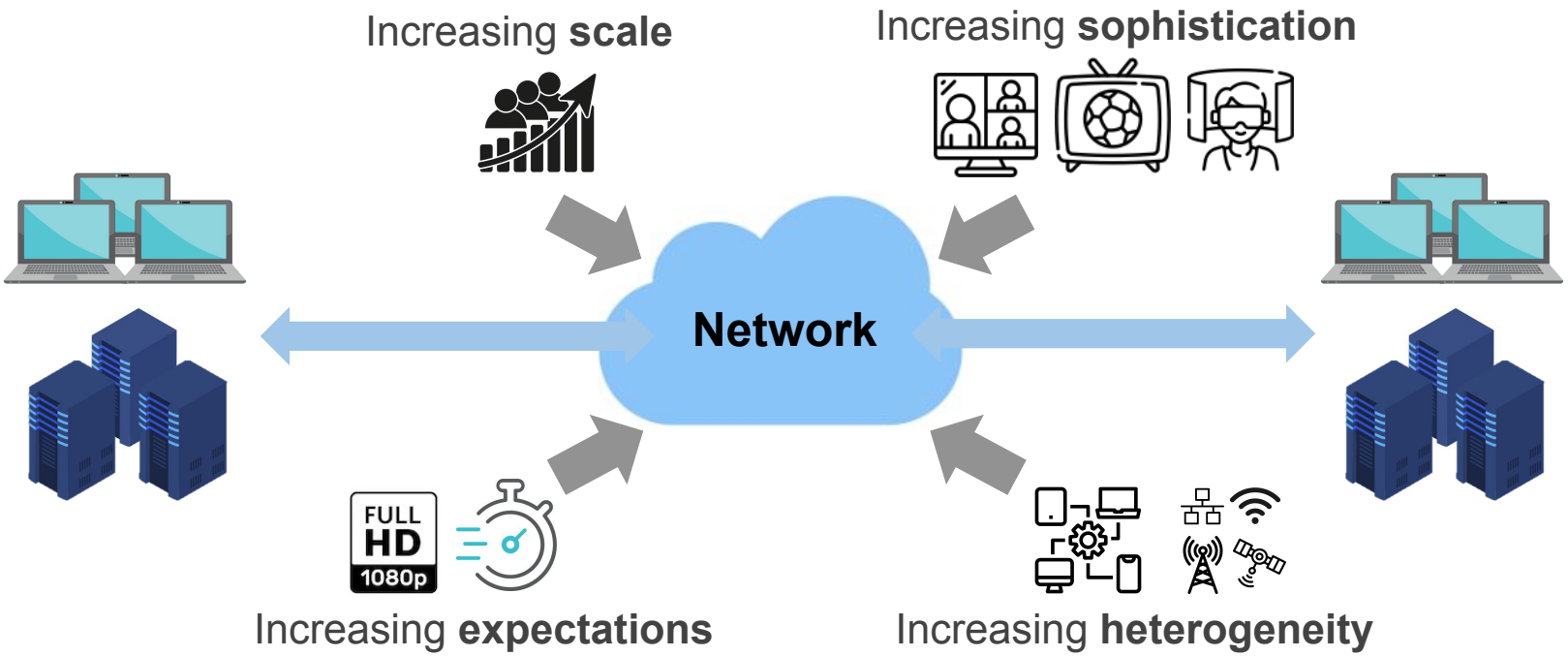
Network-Boosted Applications

Satadal “Sata” Sengupta

Advisor: Jennifer Rexford

Collaborators: Oliver Michel, Hyojoon Kim, Daniel Jubas, Maria Apostolaki, Ravi Netravali

Explosion of Demand



Network Falls Short

Best-Effort Conduit

End-to-End Protocols

Layer in Stack	Protocol
7 APPLICATION	WebRTC
4 TRANSPORT	TCP
3 NETWORK	BGP

Control plane

X Visibility, speed

General-Purpose S/W

e.g., middleboxes, VNFs

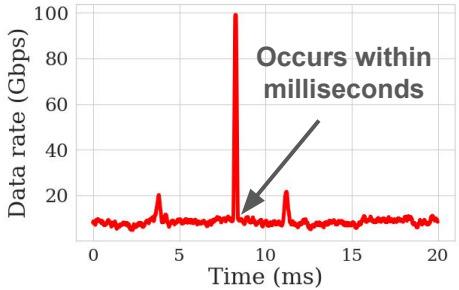
X CPU contested

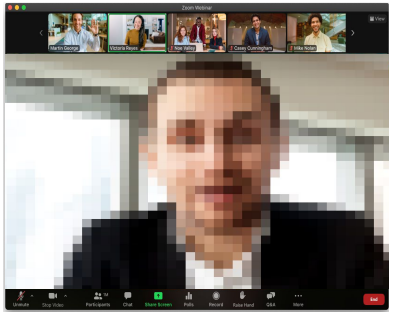
X High latency, jitter

X Expensive

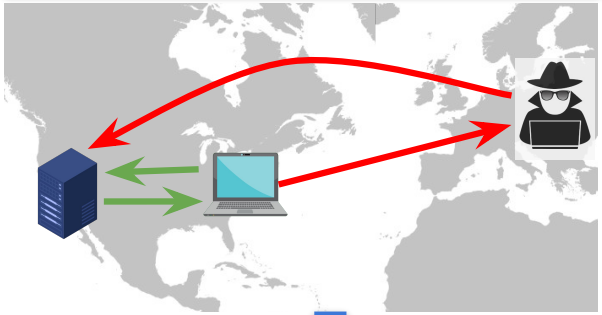
Network Problems (Will) Persist

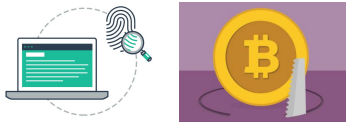
Performance Problems





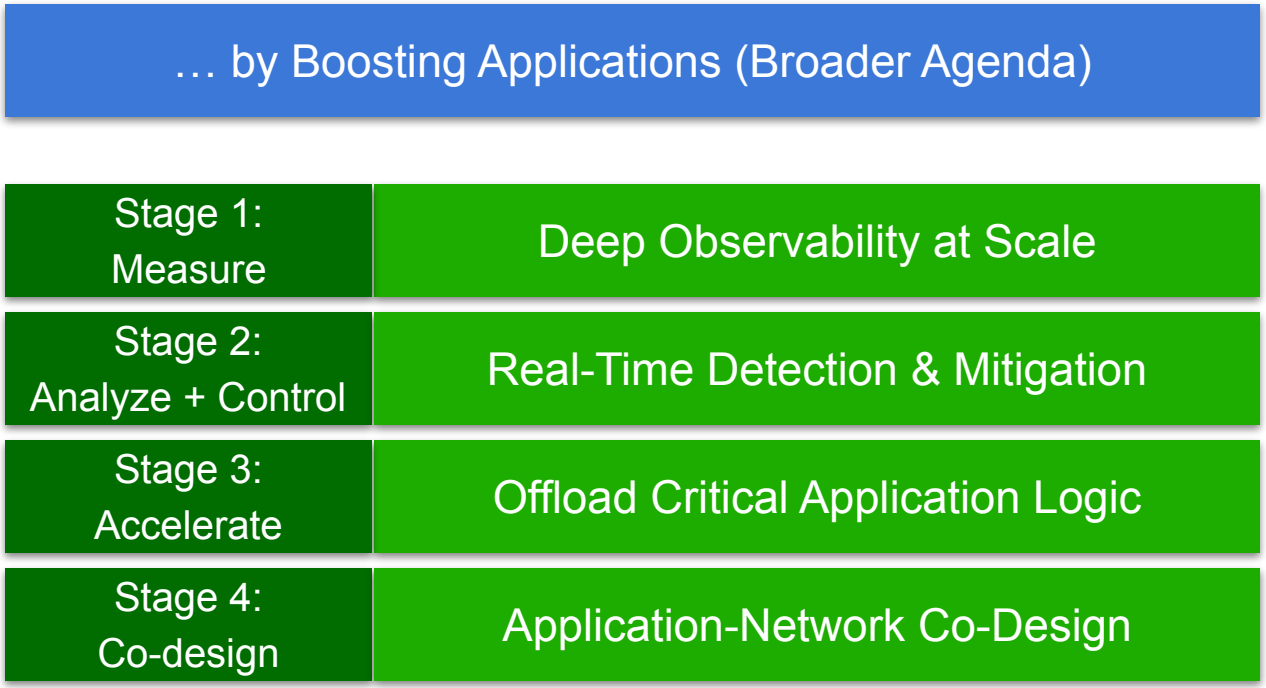
Security Incidents





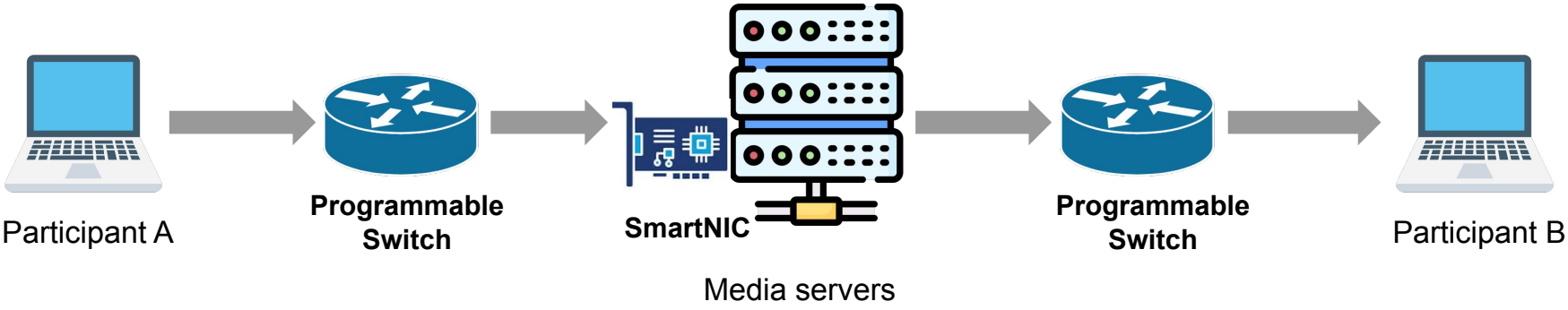
Scalability Issues

Network Needs to Step Up



Enabler … with Constraints

State-of-the-art network platforms



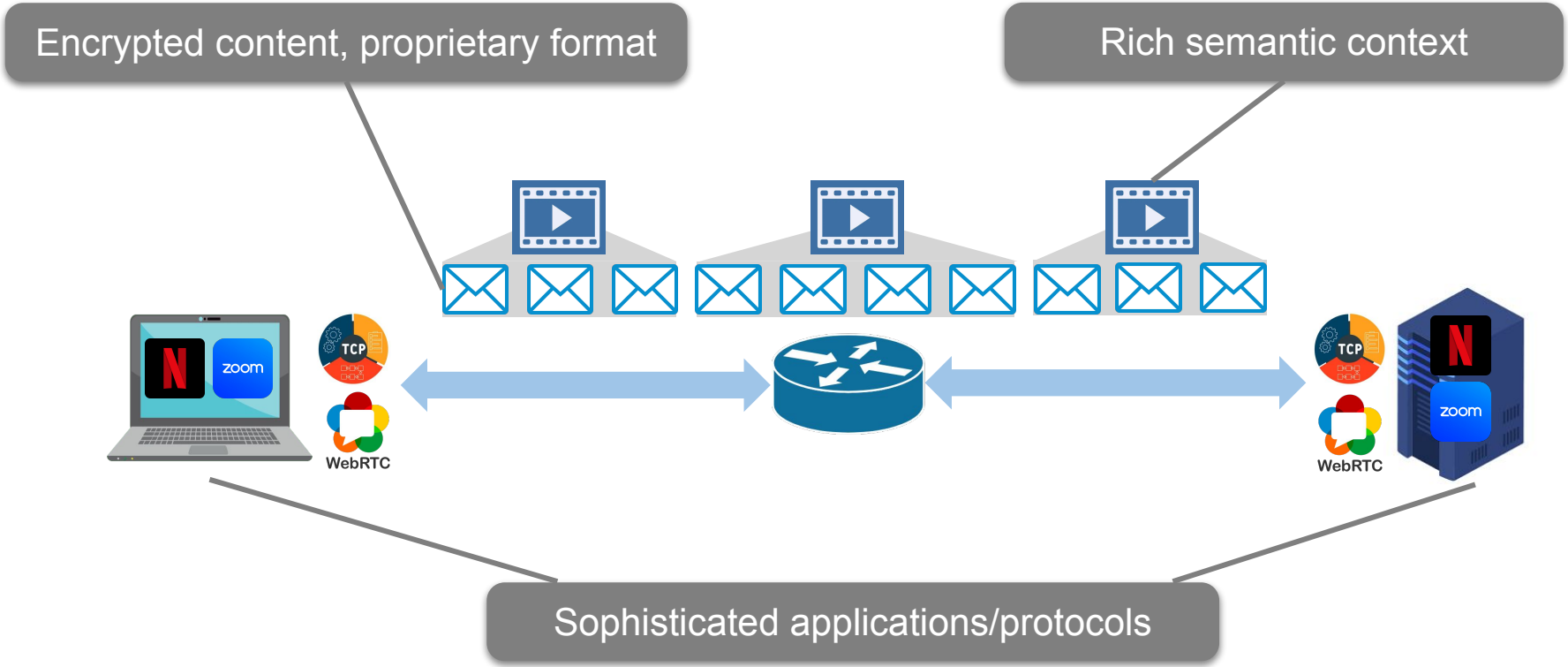
Advantages

- Production-scale (Tbps) speed
- Programmability (P4)

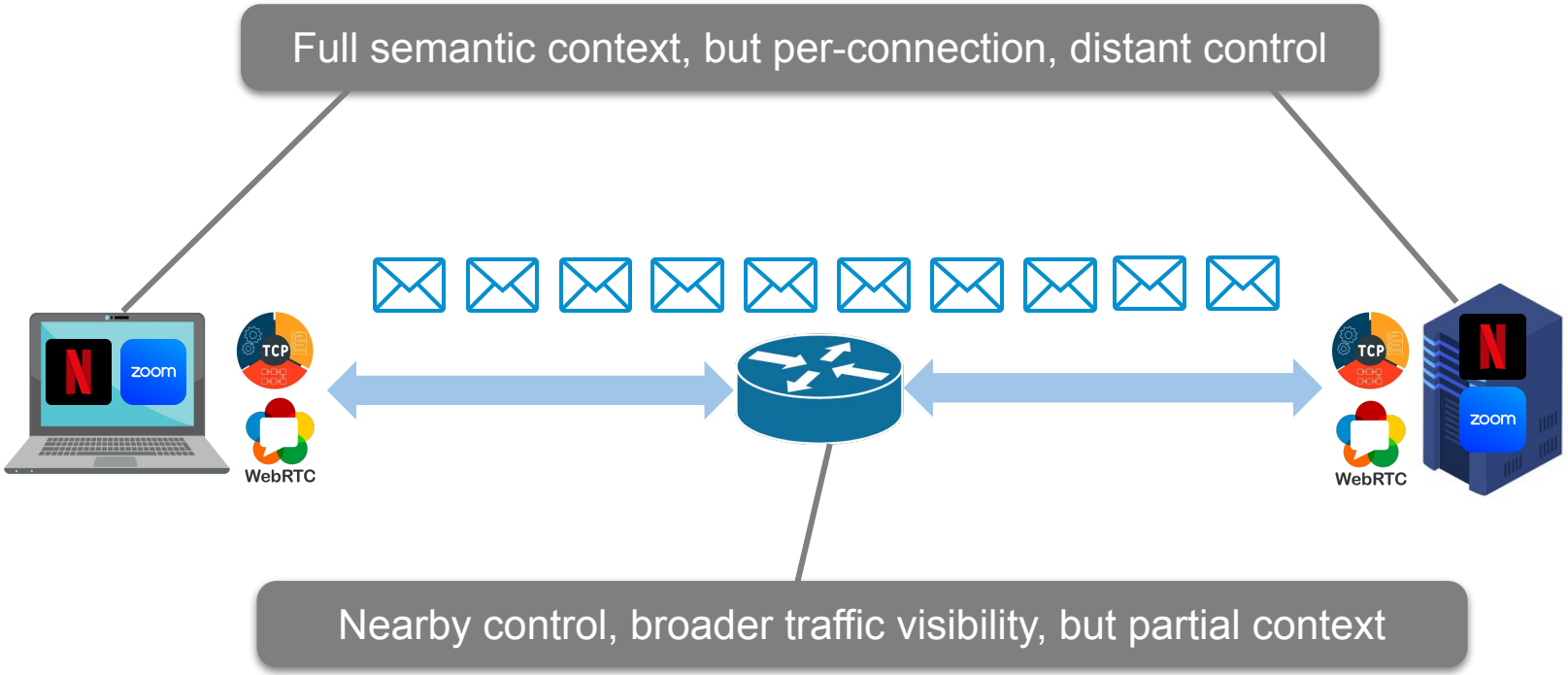
Limitations

- Memory size (# connections)
- Memory bandwidth (touches per pkt.)
- Compute (stages, loops, math)

Challenge: Application–Network Gap



Challenge: Application–Network Gap



Balancing Act

Partial semantic context + resource scarcity of programmable hardware

In-network boosting of sophisticated applications

Balancing Act

Application-aware, hardware-amenable systems

Partial semantic context + resource scarcity of programmable hardware

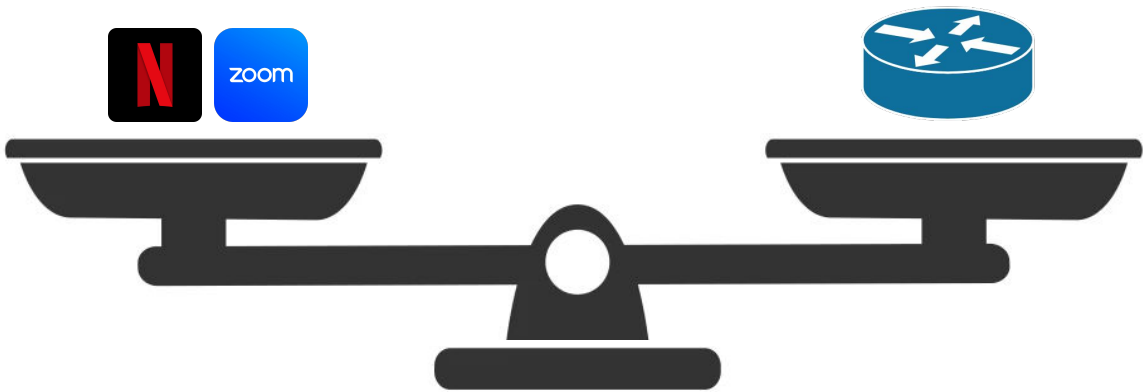
In-network boosting of sophisticated applications

Balancing Act

Application-aware, hardware-amenable systems

Partial semantic context + resource scarcity of programmable hardware

In-network, packet-level, boosting of sophisticated applications



Research Contributions

Important sub-problems within network-boosted applications

Stage 0: Understand	Demystifying important apps./workloads	Zoom Measurement Study (IMC 2022)
Stage 1: Measure	Deep Observability at Scale	Continuous Round-Trip Time Monitoring (SIGCOMM 2022)
Stage 2: Analyze + Control	Real-Time Detection & Mitigation	Delay-Based Routing-Attack Detection (NINeS 2026)
Stage 3: Accelerate	Offload Critical Application Logic	Scalable Video Conferencing Infra. (SIGCOMM 2025)
Stage 4: Co-design	Application-Network Co-Design	Lessons and Future Work

Analysis of production traffic

Hardware prototypes, artifacts

Today…

Important sub-problems within network-boosted applications

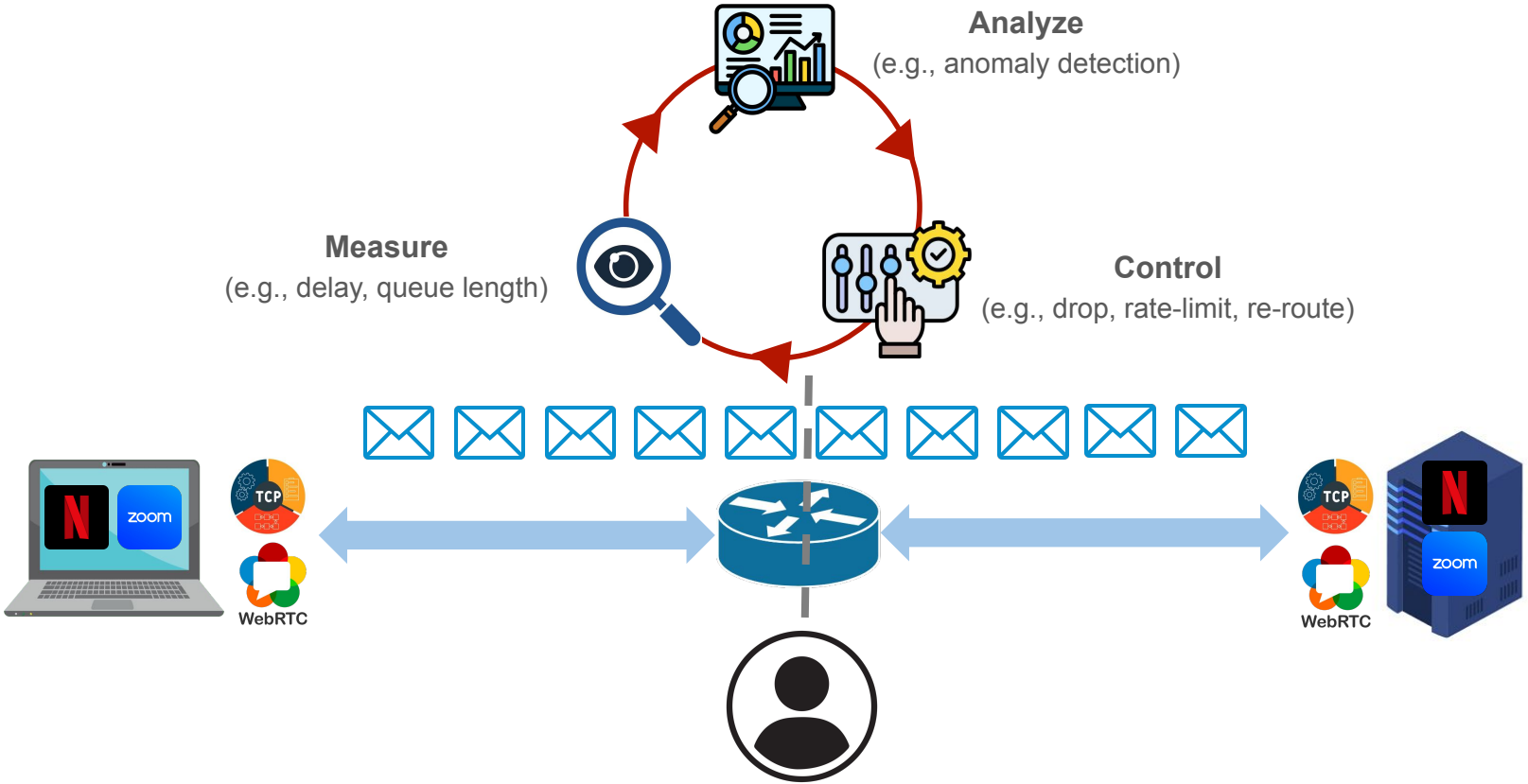
Stage 0: Understand	Demystifying important apps./workloads	Zoom Measurement Study (IMC 2022)
Stage 1: Measure	Deep Observability at Scale	Continuous Round-Trip Time Monitoring (SIGCOMM 2022)
Stage 2: Analyze + Control	Real-Time Detection & Mitigation	Delay-Based Routing-Attack Detection (NINeS 2026)
Stage 3: Accelerate	Offload Critical Application Logic	Scalable Video Conferencing Infra. (SIGCOMM 2025)
Stage 4: Co-design	Application-Network Co-Design	Lessons and Future Work

Analysis of production traffic

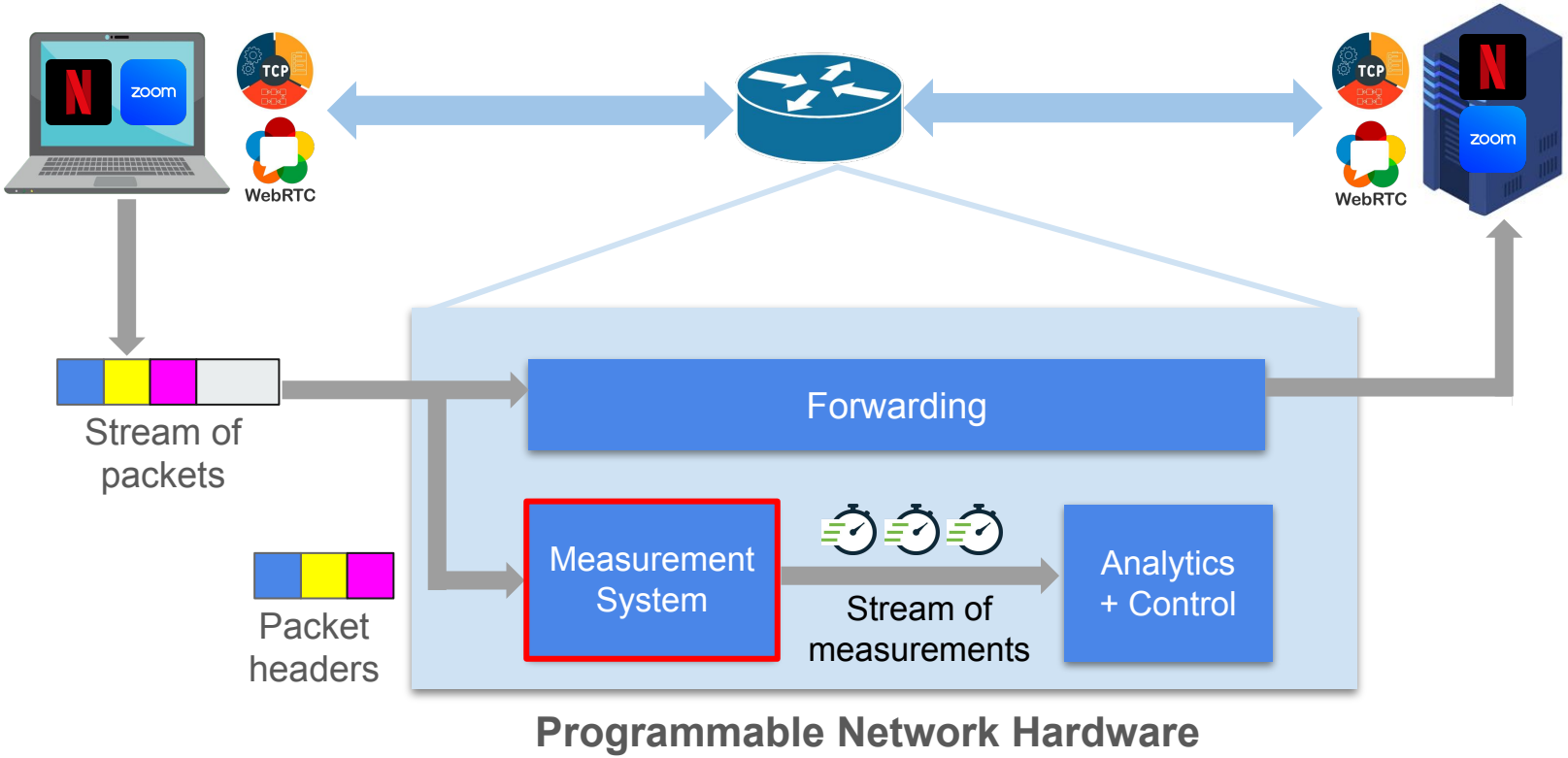
Hardware prototypes, artifacts

Stage 1: Measure

Deep Observability at Scale



System Vision



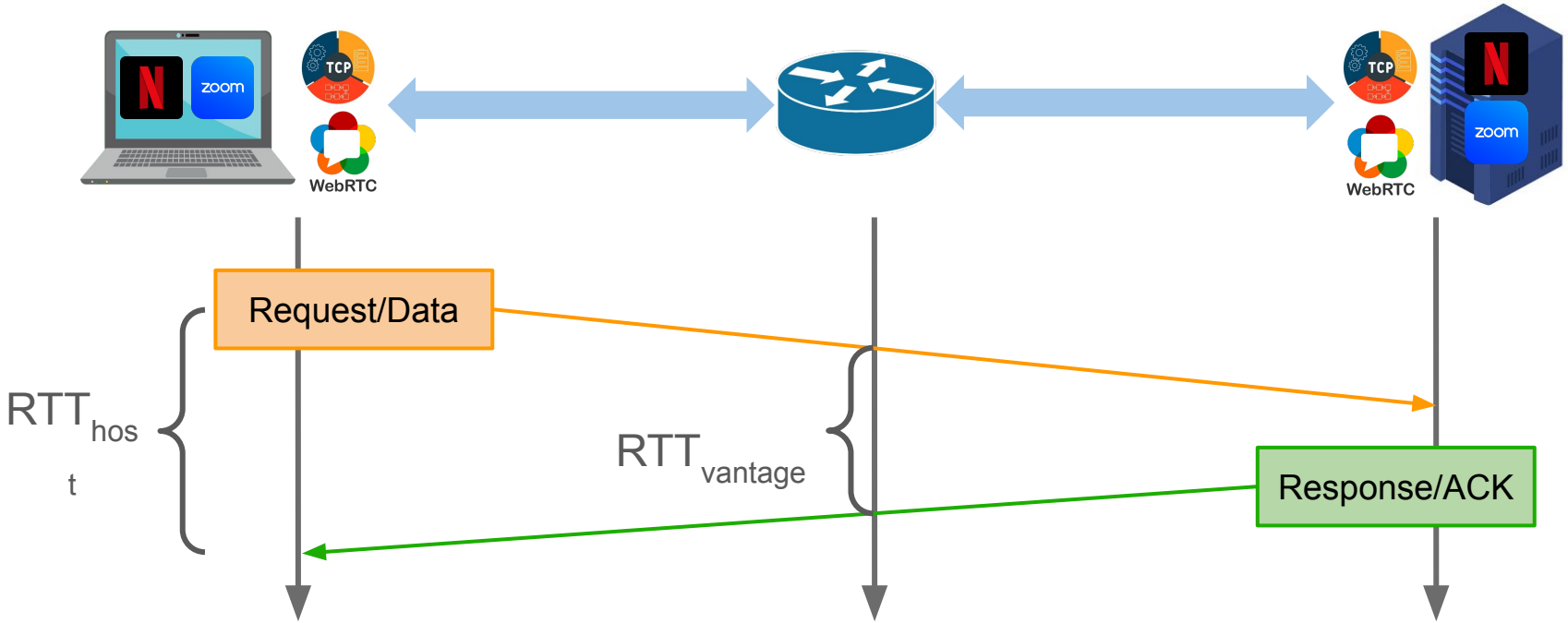
Stage 1: Measure	Deep Observability at Scale	Applied to on-demand traffic (e.g., browsing, video streaming)

DART: Continuous TCP Round-Trip Time Monitoring

with Hyojoon Kim and Jennifer Rexford

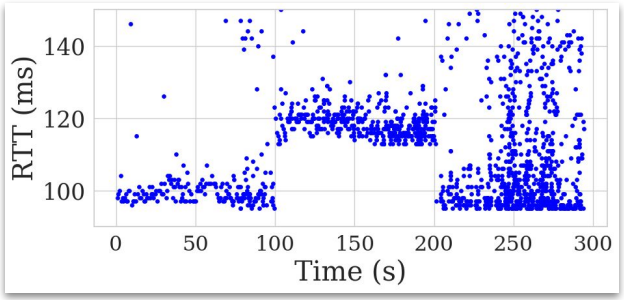
ACM SIGCOMM 2022

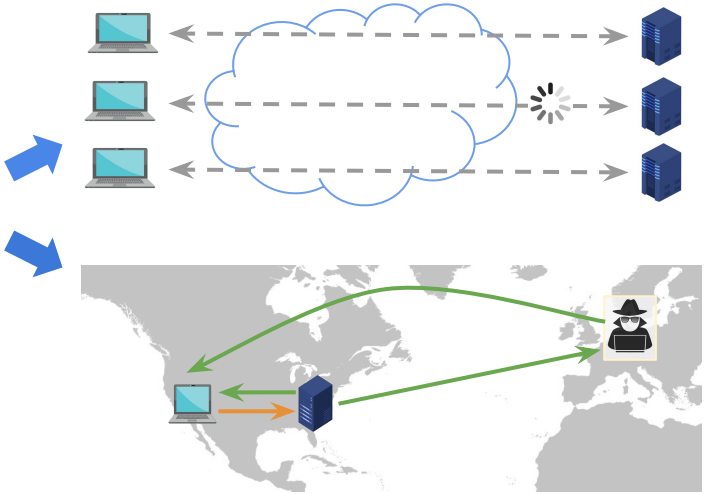
Round-Trip Time (RTT): Measure of Latency/Delay



Important indicator of network performance, user QoE

RTT for Network Control



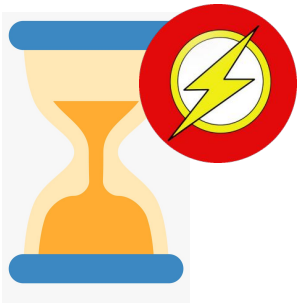


Rate control (e.g., queue management), re-routing

Raise alarm, block traffic

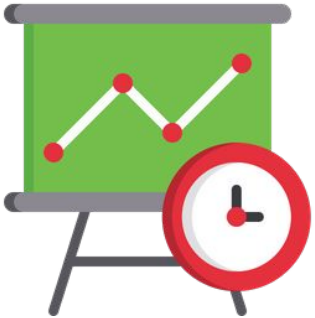
Effective RTT Measurement

Real-time RTT
(not active probing)



- Risk increases with time

Continuous RTT
(not just conn. establishment)



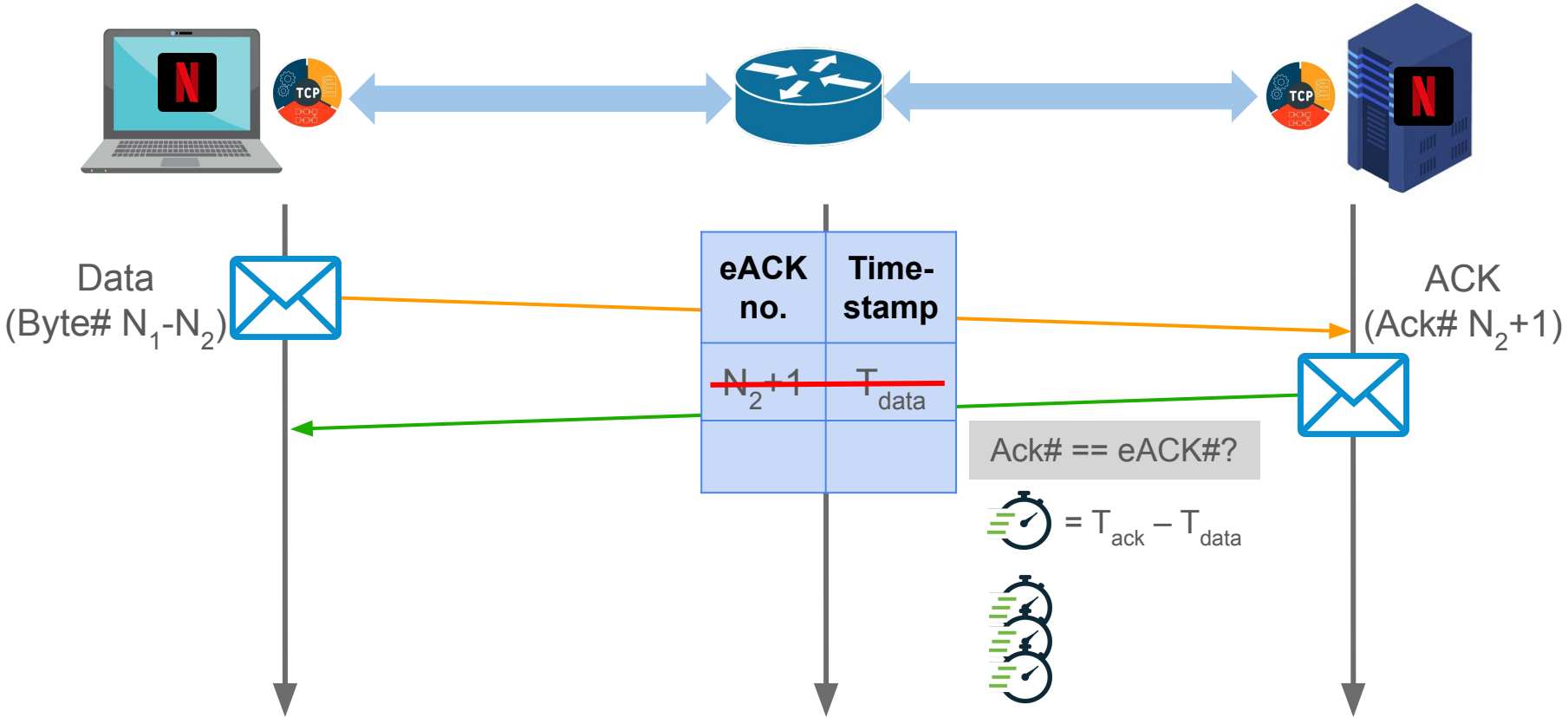
- Conditions may change rapidly

Accurate RTT
(not ignore ambiguities)

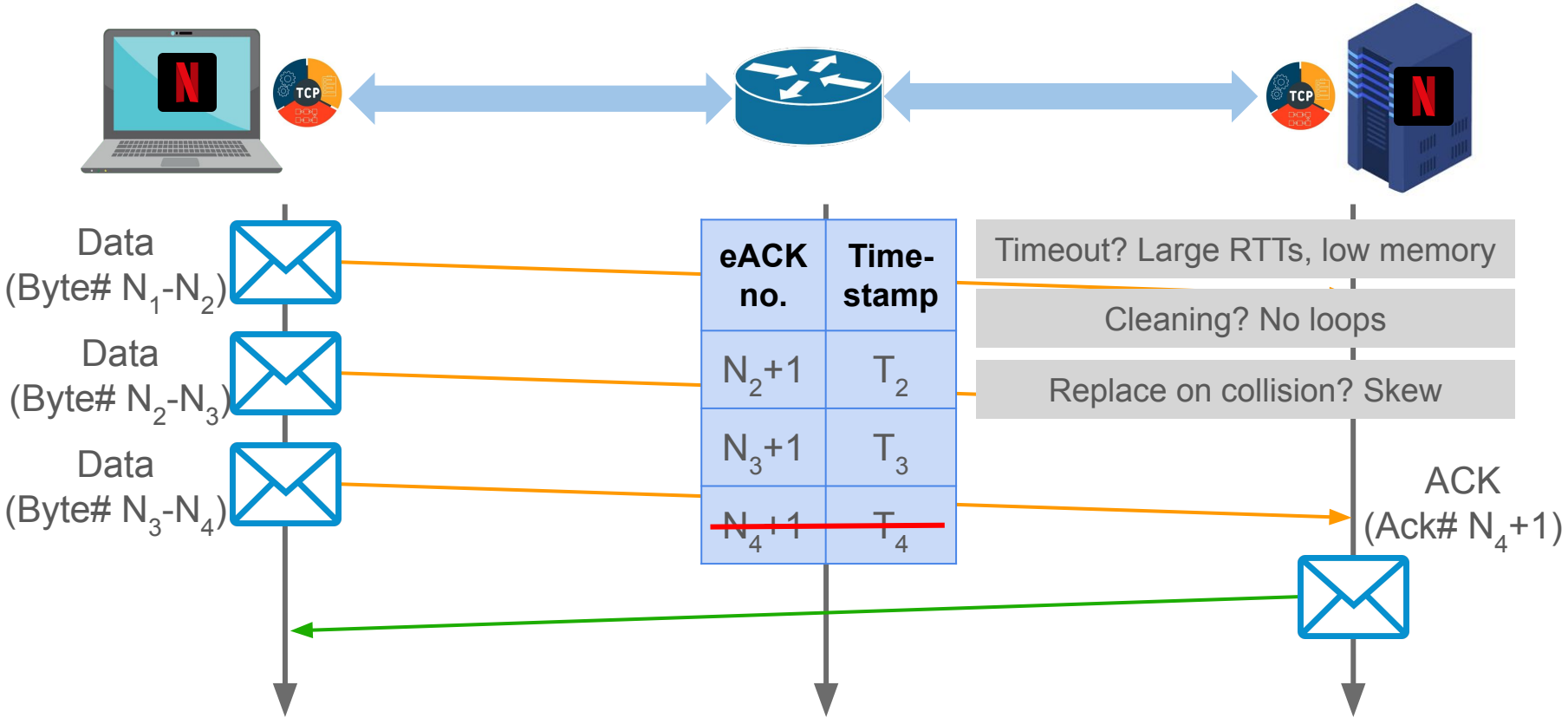
- False positives and negatives are costly

Opportunity: TCP's reliable delivery mechanism

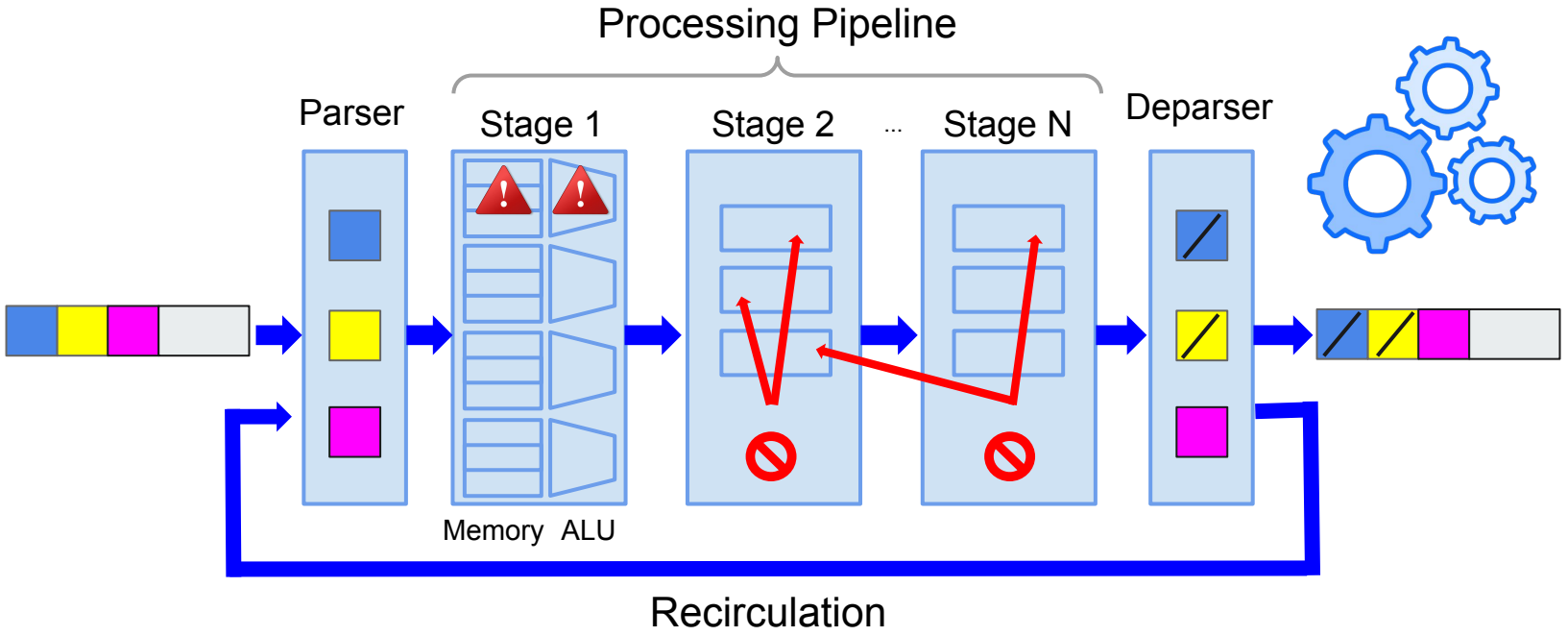
Continuous RTT Monitoring for TCP



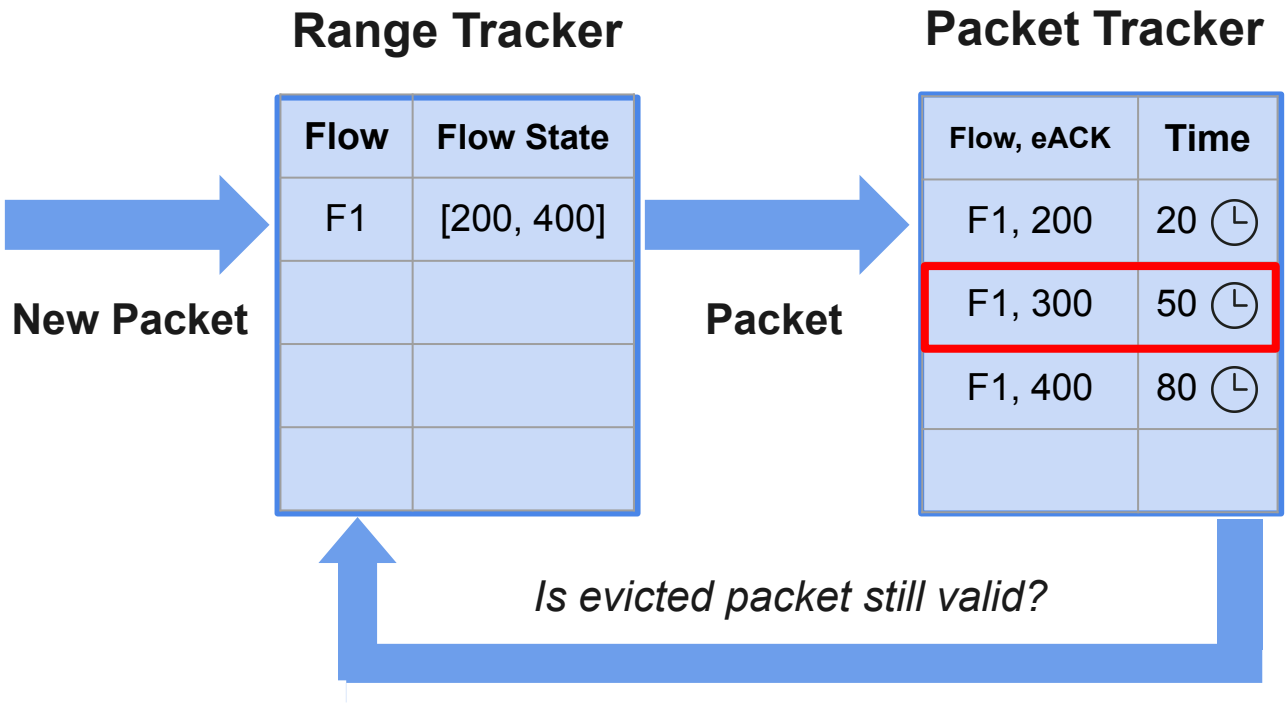
Challenge: Efficient ACKing in TCP



Background: Pipeline Architecture

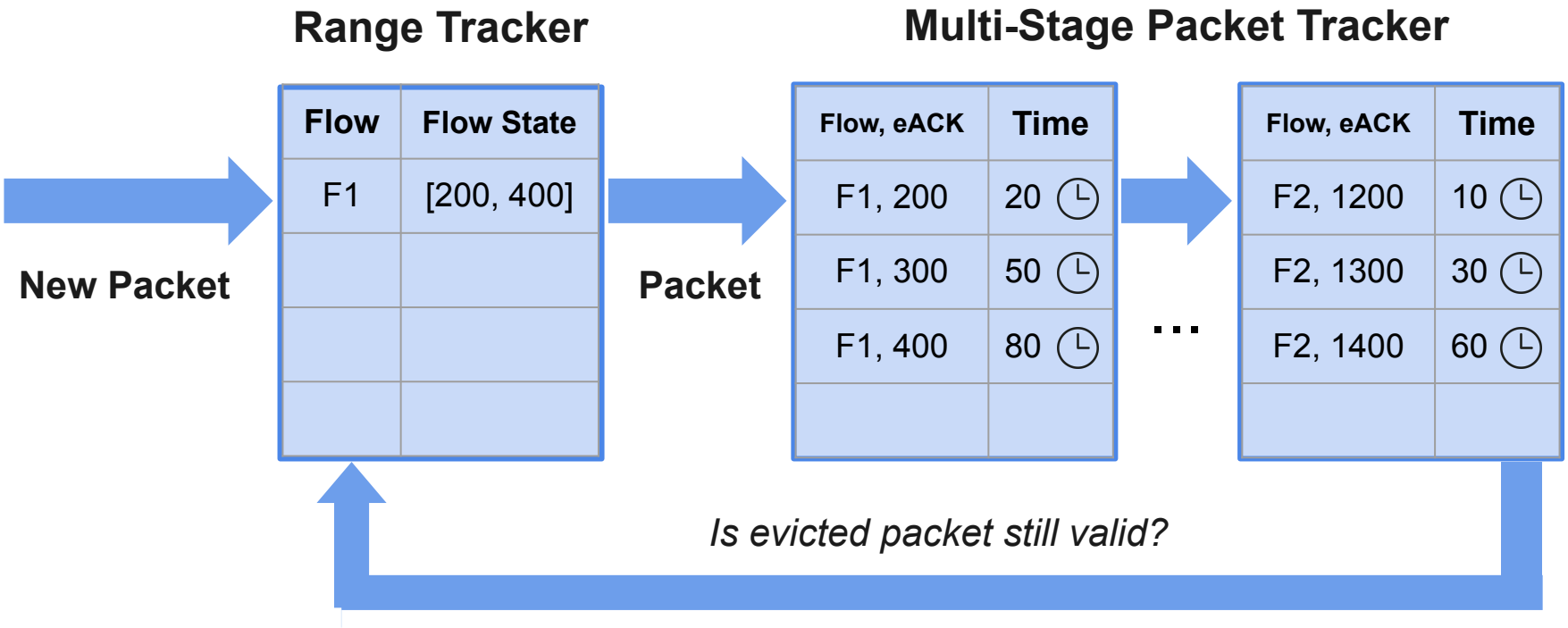


Solution: Cuckoo Hashing with Lazy Eviction

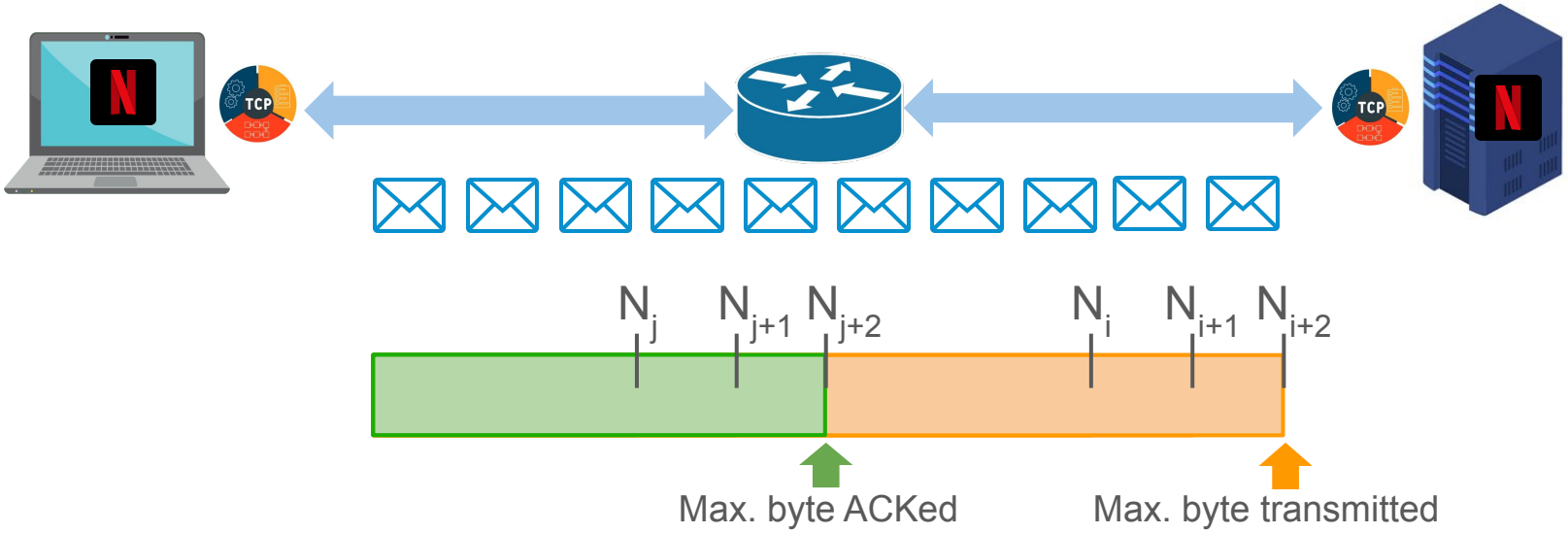


Pagh, Rasmus, and Flemming Friche Rodler. "Cuckoo hashing." *Journal of Algorithms* 51, no. 2 (2004): 122-144.

Solution: Multi-Stage Cuckoo Hashing

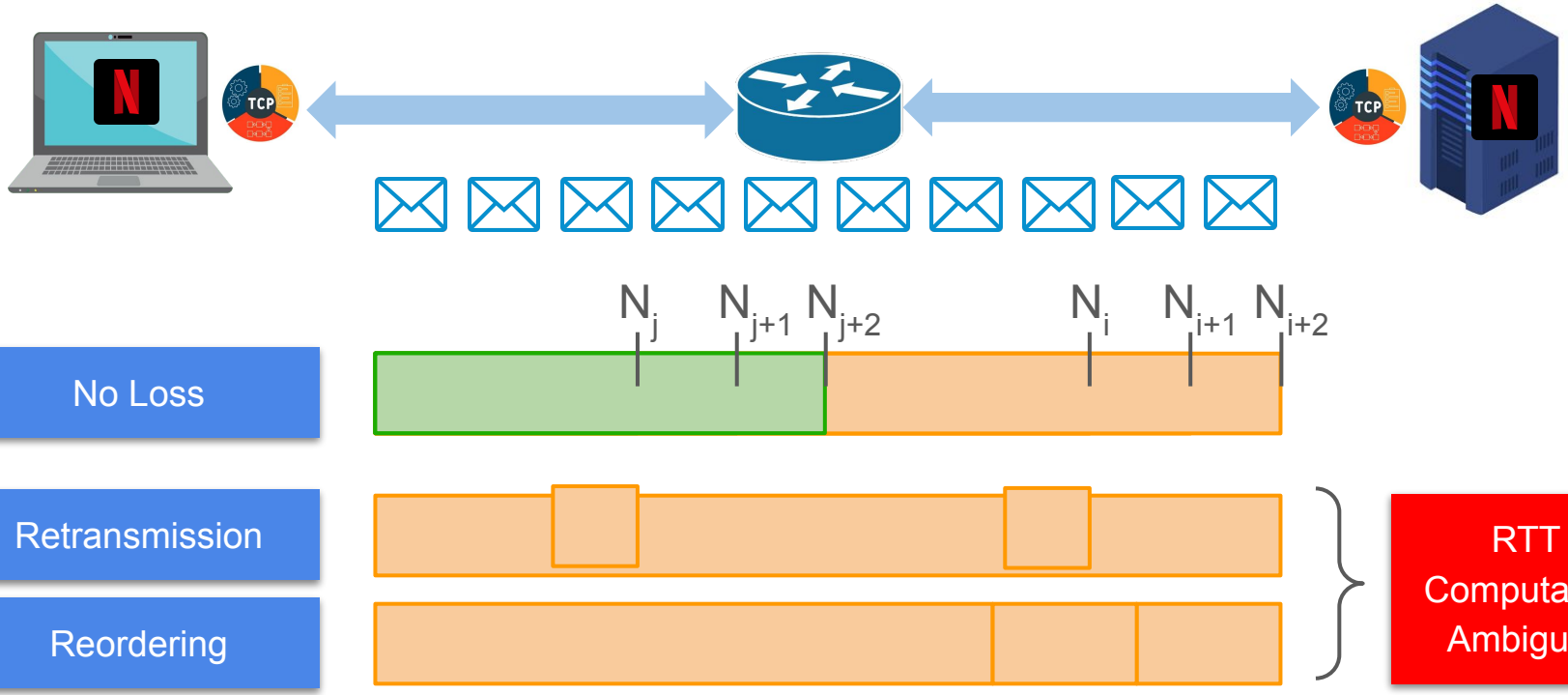


Keeping Flow State



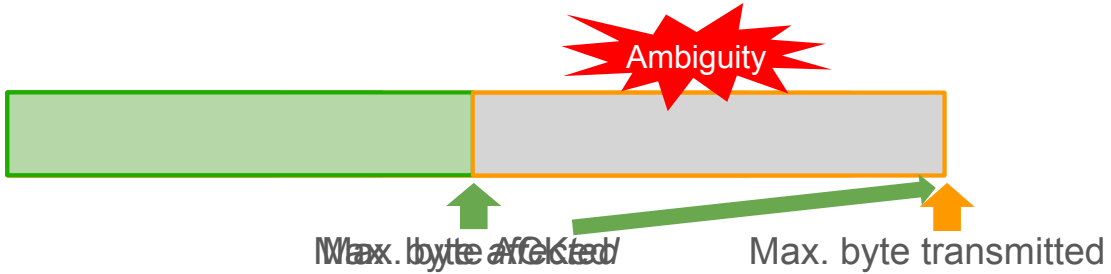
Measurement Range = [Max. byte ACKed, Max. byte transmitted]

Challenge: Loss & Reordering Handling in TCP



Solution: Addressing Ambiguity

Measurement Range = [Max. byte ACKed, Max. byte transmitted]



Measurement Range = [Max. byte *affected*, Max. byte transmitted]

Measurement Range = [Max. byte *ACKed* or *affected*, Max. byte transmitted]

Evaluation

Implementation

- In P4 for Tofino 1 & 2
- Python simulator

Dataset

PRINCETON
UNIVERSITY

- 15 mins
- 1.4M TCP conns.
- 241K packets/sec.

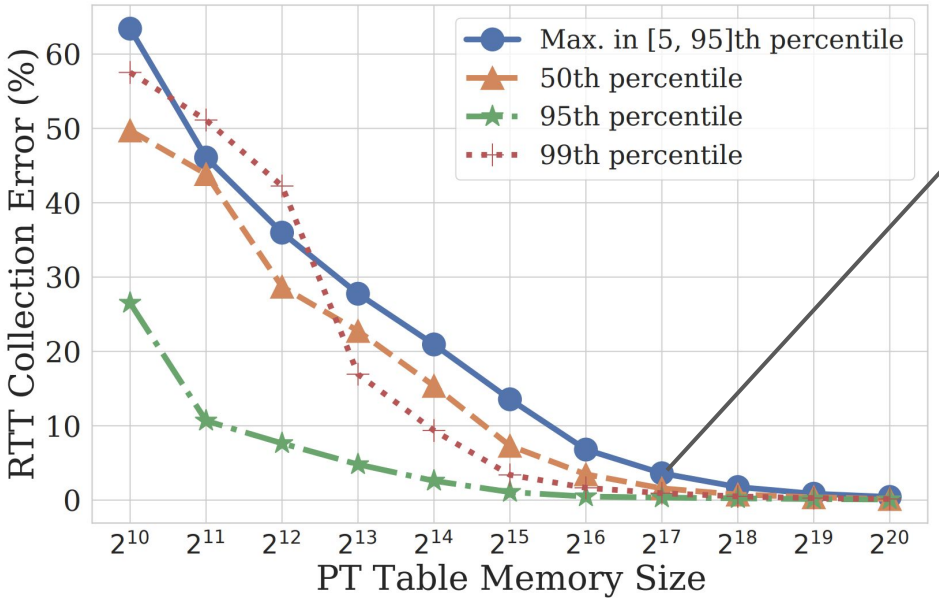
Baseline

tcptrace

- Software-based
- Post-processing tool
- Unlimited memory

Evaluation: Packet Tracker Sizing

Metrics: Error at pth percentile, Fraction of RTTs captured



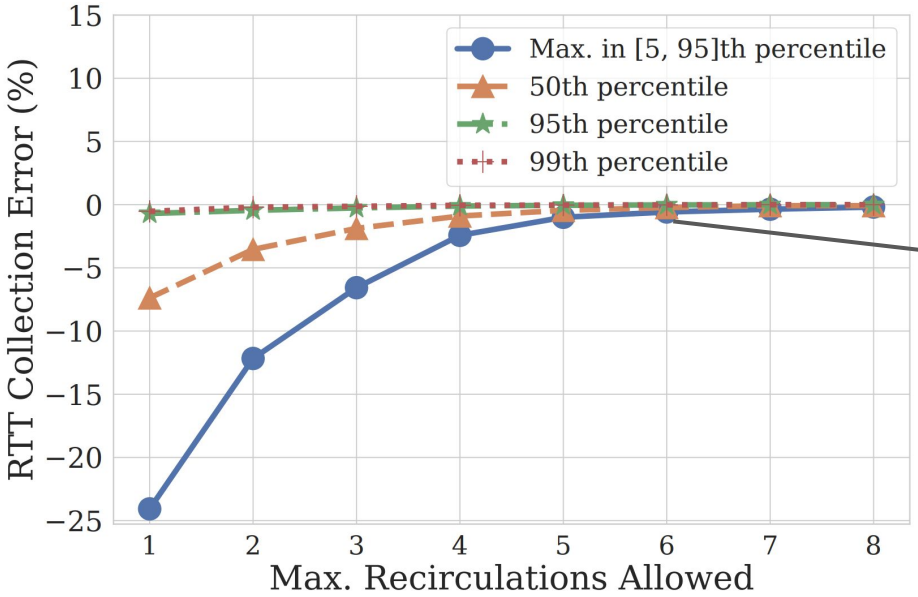
24% Tofino SRAM

<5% error, >99% samples

10% records recirculated

Evaluation: Multi-Stage Packet Tracker

Setting: 24% SRAM divided into 8 stages



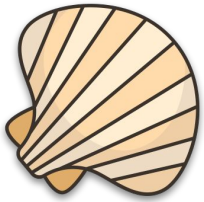
<1% error, ~100% samples

16% records recirculated

Conclusion and Lessons

- **Novelty:** Accurate, continuous in-network RTT measurement
- **Demsytification:** Understand parts of TCP relevant to RTT computation
- **Hardware-amenable design:** Cuckoo hashing with lazy recirculation
- **Key innovation:** Leverage interplay between per-flow state and per-packet state for memory efficiency
- **Key lesson:** General design to perform joins between request-response style traffic

Stage 3: Accelerate	Offload Critical Application Logic	Applied to interactive traffic (e.g., video conferencing)



Scallop: Scalable Video-Conferencing Using SDN Principles

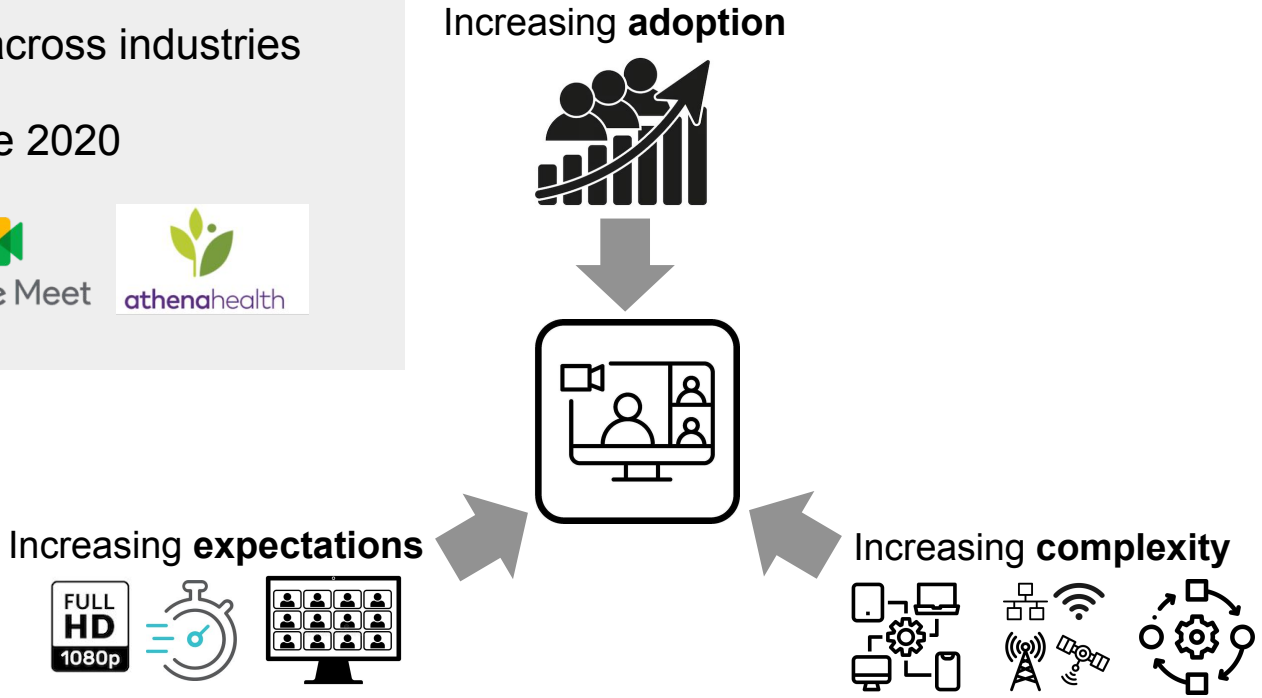
with Oliver Michel, Hyojoon Kim, Ravi Netravali and Jennifer Rexford

ACM SIGCOMM 2025

The Future of Video Conferencing

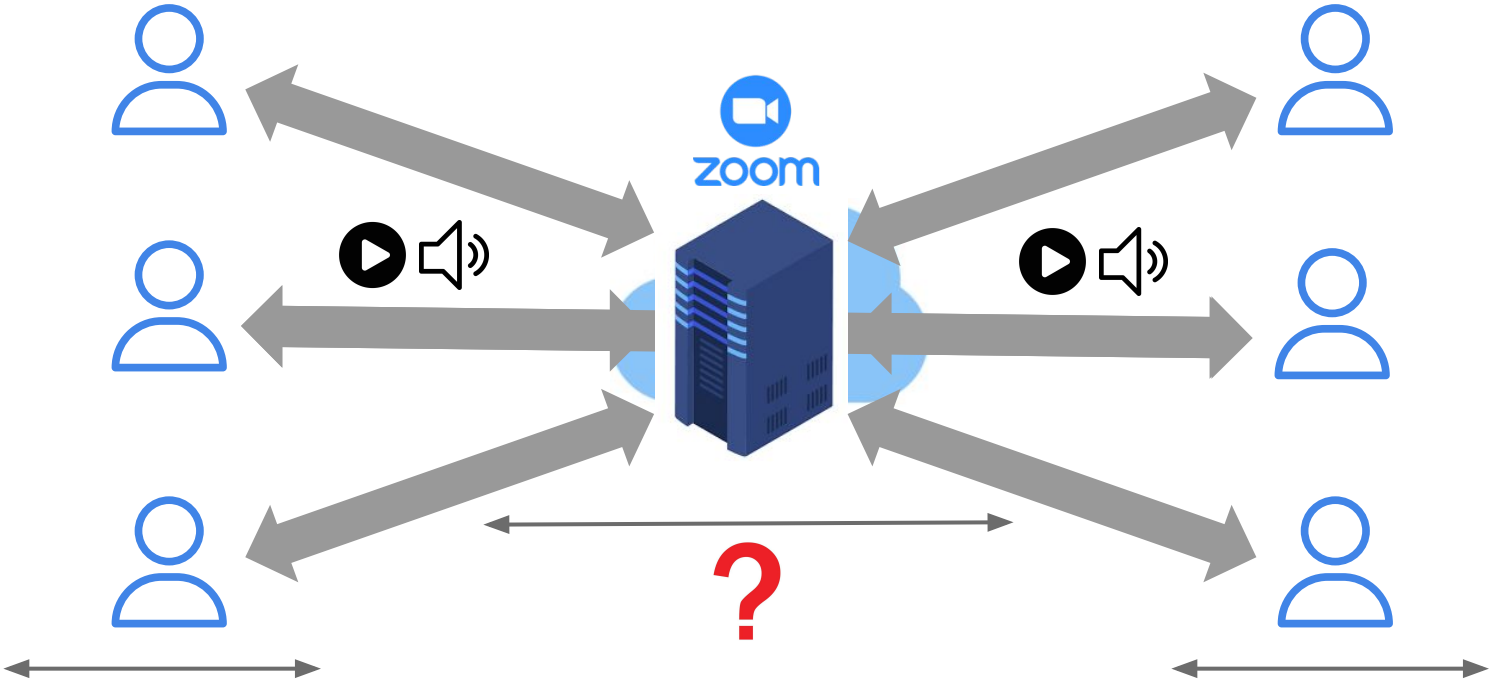
- Essential application across industries
- Explosive growth since 2020

zoom Microsoft Teams Google Meet athenahealth



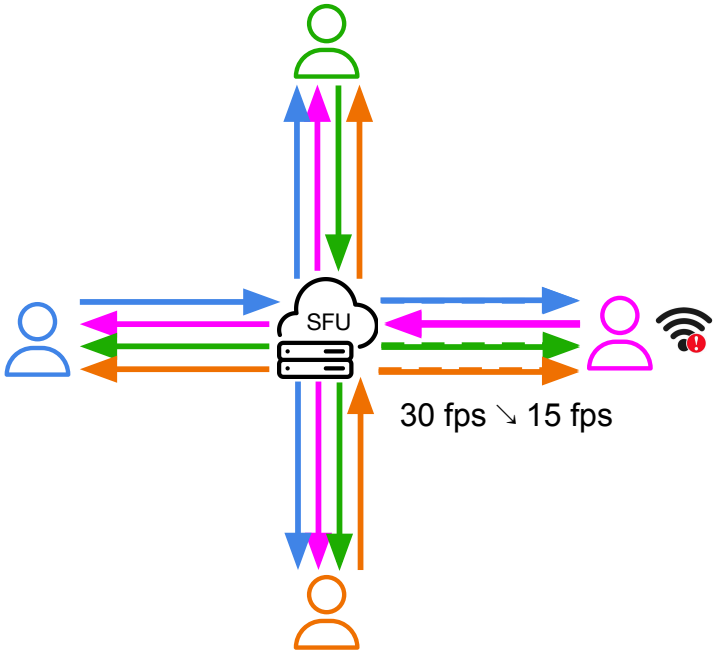
Challenge: Provide consistently high quality at scale

The Missing Middle



In this study: The application operators' perspective

Selective Forwarding Unit (SFU)



SFU roles:

(1) Relay audio and video streams

(2) Monitor and adapt media signals

SFUs hard to scale:

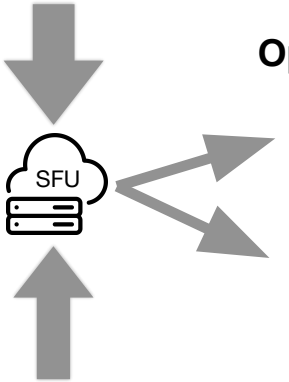
(1) Workload hard to predict

(2) Quadratic scaling

3 → 4 parts. ⇒ 9 → 16 streams

The SFU Scaling Challenge

Dynamic, high-volume workload

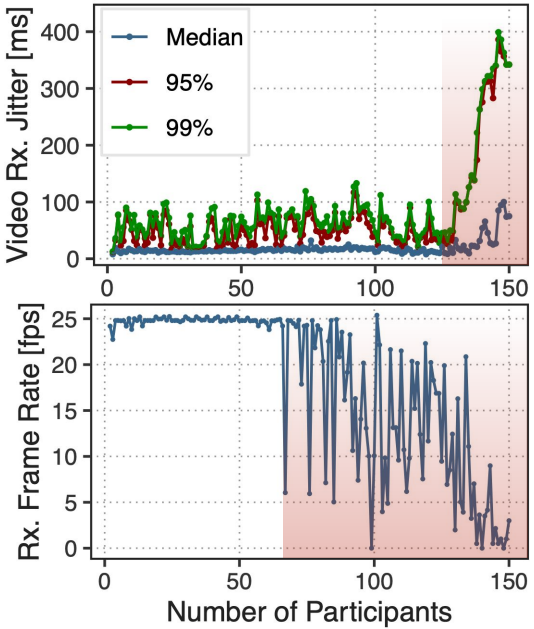


Underprovisioning can affect quality massively

Operators left with two options:

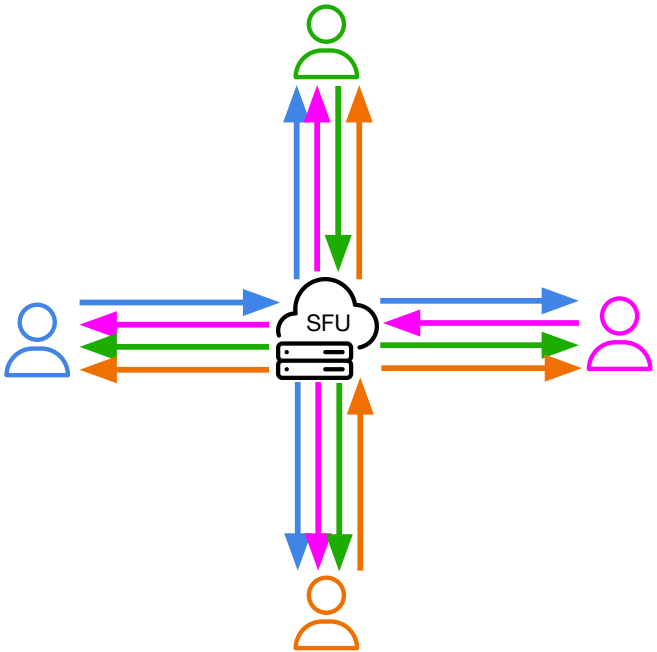
- Massively over-provision
  → costly and wasteful
- Reactively autoscale
  → risk harming QoE for users

QoE in MediaSoup when increasing SFU load:



SFUs as Packet Processors

SFU operation is strikingly similar to traditional packet processing



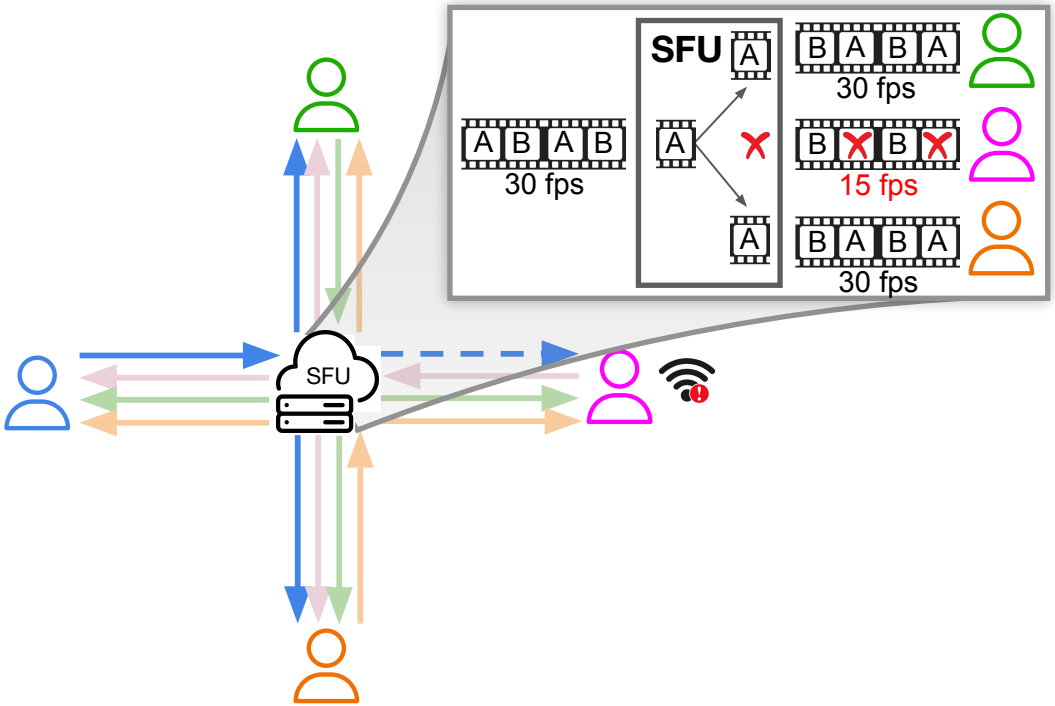
(1) Relay audio and video streams

↓

Replicate traffic *(Multicast)*

SFUs as Packet Processors

SFU operation is strikingly similar to traditional packet processing



(1) Relay audio and video streams

Replicate traffic *(Multicast)*

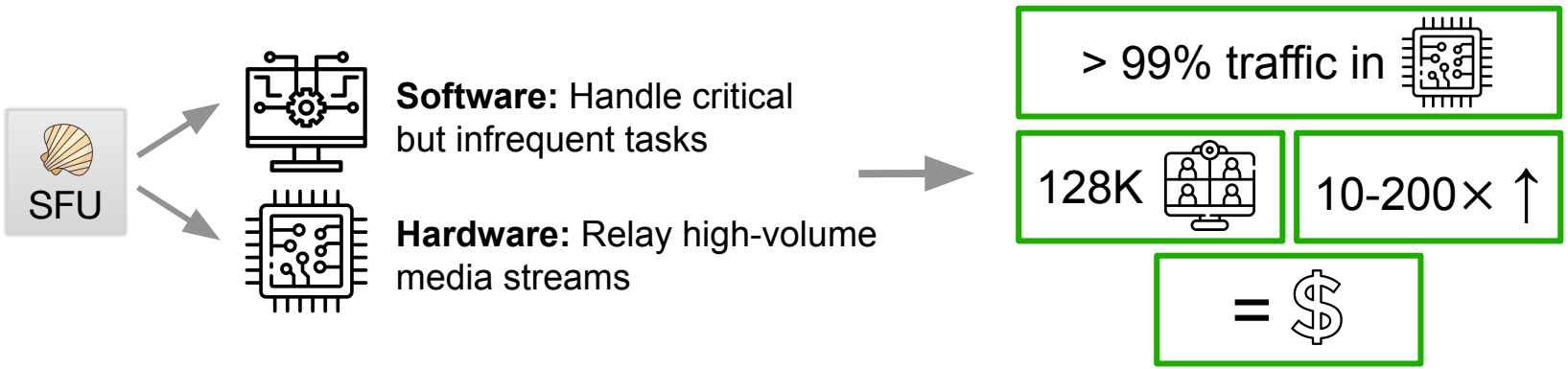
(2) Monitor and adapt media signals

Selectively forward traffic *(Firewall)*

Scallop

Fundamental rethink of video conferencing infrastructure
to support long-term traffic forecasts

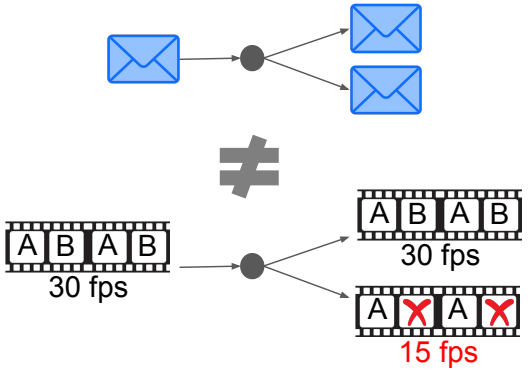
A novel hardware/software SFU co-design inspired by SDN



Challenges

Monolithic Software Architecture

Complex Multicast Requirements



Misaligned with Widely-Deployed Standard

WebRTC

Existing systems → split-proxy

- Proxy architecture?
- Feedback semantics?
- Transparent rate adaptation?

1

How to disaggregate into control and data planes?

2

How to realize and scale application-layer multicast?

3

How to make Scallop interoperable with WebRTC?

1 SDN-Inspired Disaggregation

SFU workload is amenable to a control/data-plane split

(1) >10ms latency, every few mins.

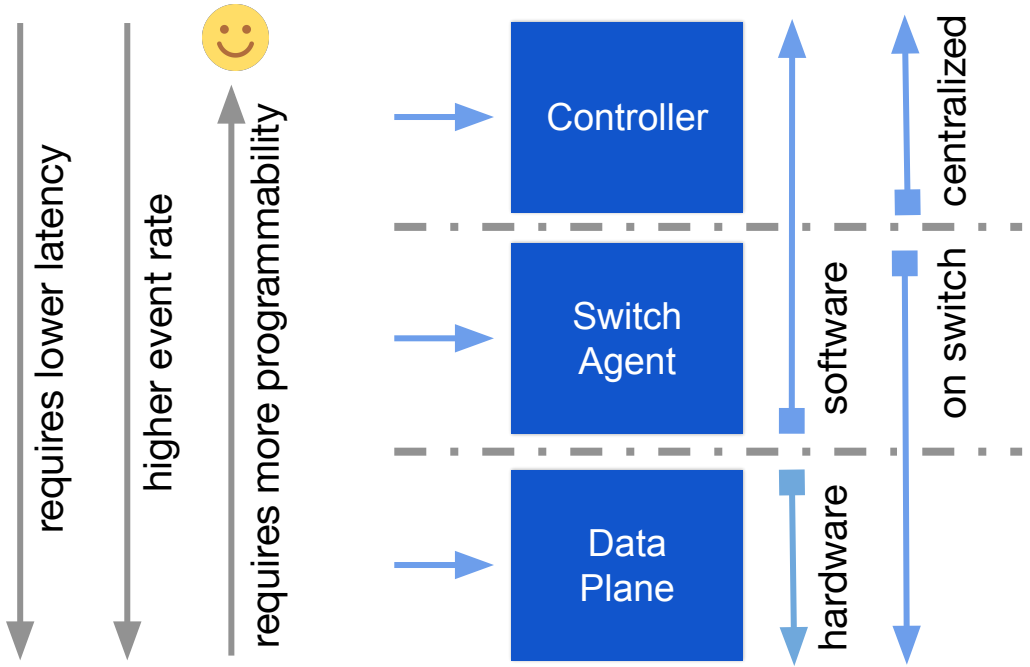
e.g., session management, signaling

(2) 1 < t < 10ms latency, 2-3 per sec.

e.g., handling feedback messages and connectivity checks

(3) <1ms latency, 100s per sec.

replication and selective forwarding of media packets

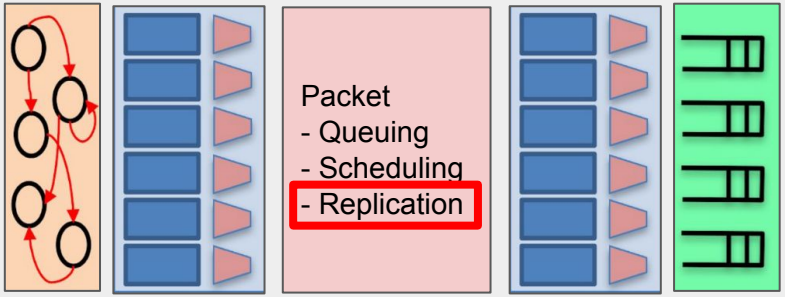


2 Scalable Application-Layer Multicast

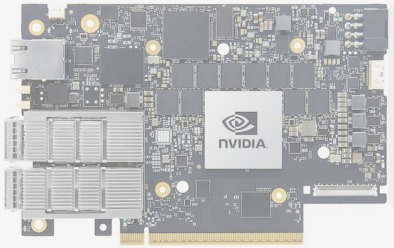
Hardware-native packet copying capabilities can be leveraged for SFU-style replication

Programmable Switches

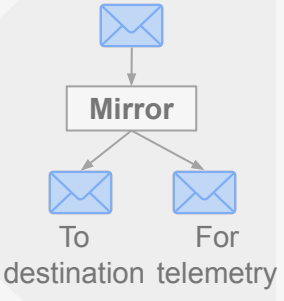
Intel Tofino2



SmartNICs/DPUs

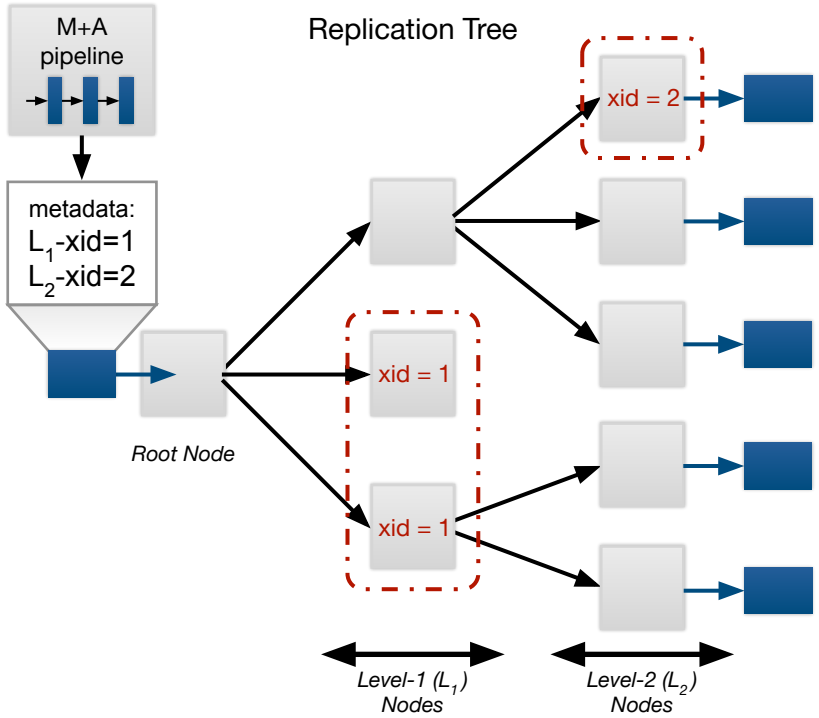


NVIDIA BlueField-3



2 Scalable Application-Layer Multicast

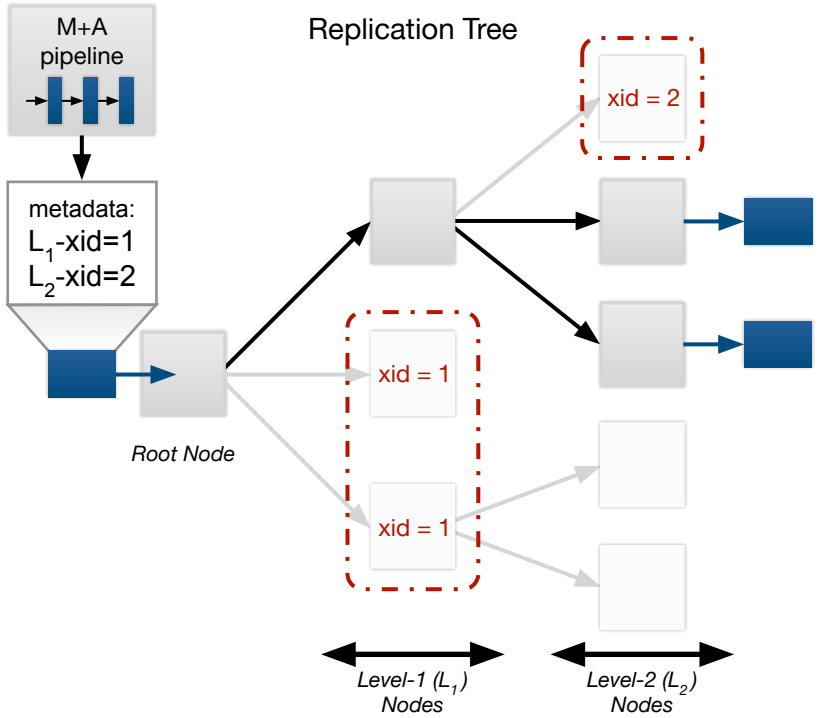
Background on Tofino's Packet Replication Engine (PRE)



- Abstraction: *replication tree* with *level-1 nodes* and *level-2 nodes*
- Supports *dynamic tree pruning*
  - Each node can be associated with an exclusion ID (xid)
  - The ingress match+action pipeline can associate individual packets with xids

2 Scalable Application-Layer Multicast

Background on Tofino's Packet Replication Engine (PRE)



- Abstraction: *replication tree* with *level-1 nodes* and *level-2 nodes*
- Supports *dynamic tree pruning*
  - Each node can be associated with an exclusion ID (xid)
  - The ingress match+action pipeline can associate individual packets with xids
  - No replication along edges leading to excluded nodes

2 Scalable Application-Layer Multicast

Packet Replication in Scallop: Challenges

PRE-to-VC Mapping?

- PRE entities:
  - Root, L1 + L2 nodes
  - L1 + L2 xids
- VC entities:
  - Meetings, participants
  - Quality layers

Different Meeting Configurations

- Rate adaptation status
- Two-party vs. multiparty
- Can change dynamically

Limited Resources

- 64,000 replication trees
- 2^{24} L1 nodes

Q

How to (i) correctly and (ii) efficiently map VC entities to PRE entities for each meeting configuration?

2 Scalable Application-Layer Multicast

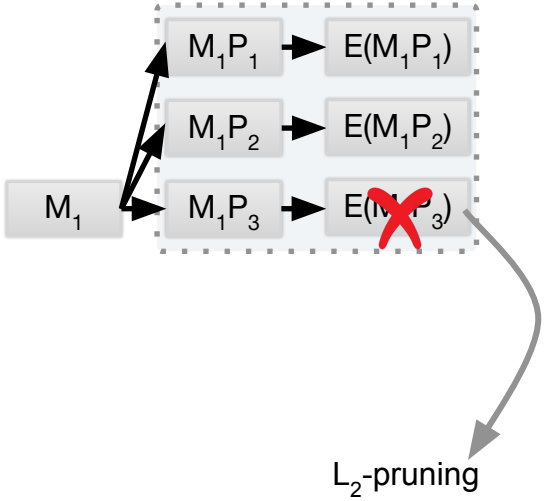
Packet Replication in Scallop: Solution

Dynamic Migration

Optimal Designs

- Non-Rate-Adapted (NRA)
- Rate-Adapted (RA)
  - Receiver (RA-R)
  - Sender-Receiver (RA-SR)
- Two-Party (2P)

Non-Rate-Adapted Design



2 Scalable Application-Layer Multicast

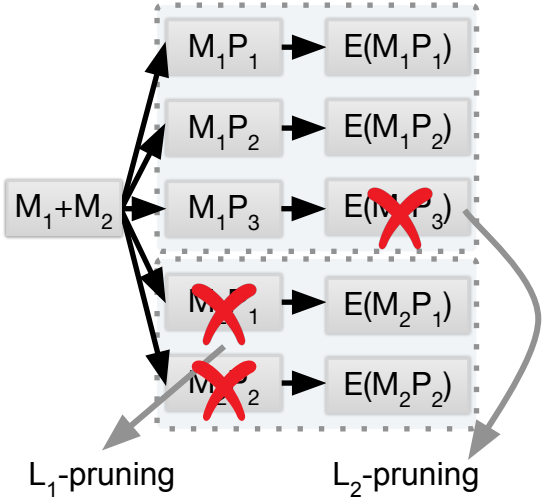
Packet Replication in Scallop: Solution

Dynamic Migration

Optimal Designs

- Non-Rate-Adapted (NRA)
- Rate-Adapted (RA)
  - Receiver (RA-R)
  - Sender-Receiver (RA-SR)
- Two-Party (2P)

Non-Rate-Adapted Design



Supports up to **128K concurrent meetings** and **2^{24} concurrent participants**

Evaluation

Scallop processes > 99% traffic in hardware

Protocol/Type	Packets	%	Per sec.
RTP	170,870	94.5	284.3
- Audio	29,746	16.46	49.49
- Video	141,124	78.09	234.81
- AV1 DS*	5	«	0.008
RTCP	9,153	5.06	15.22
- SR/SDES	3,456	1.91	5.75
- RR*	240	0.39	0.13
- RR/REMB*	5,457	3.02	9.07
STUN*	**695**	**0.38**	**1.15**

Evaluation

Scallop processes > 99% traffic in hardware

Protocol/Type	Packets	%	Per sec.
RTP	170,870	94.5	284.3
- Audio	29,746	16.46	49.49
- Video	141,124	78.09	234.81
- AV1 DS*	5	«	0.008
RTCP	9,153	5.06	15.22
- SR/SDES	3,456	1.91	5.75
- RR*	240	0.39	0.13
- RR/REMB*	5,457	3.02	9.07
STUN*	**695**	**0.38**	**1.15**

Evaluation

Scallop processes > 99% traffic in hardware

Protocol/Type	Packets	%	Per sec.
RTP	170,870	94.5	284.3
- Audio	29,746	16.46	49.49
- Video	141,124	78.09	234.81
- AV1 DS*	5	«	0.008
RTCP	9,153	5.06	15.22
- SR/SDES	3,456	1.91	5.75
- RR*	240	0.39	0.13
- RR/REMB*	5,457	3.02	9.07
STUN*	**695**	**0.38**	**1.15**
Control Plane	6,397	3.54	10.64
Data Plane	174,326	**96.46**	290.06
Total	180,718	100	300.69

Evaluation

Scallop processes > 99% traffic in hardware

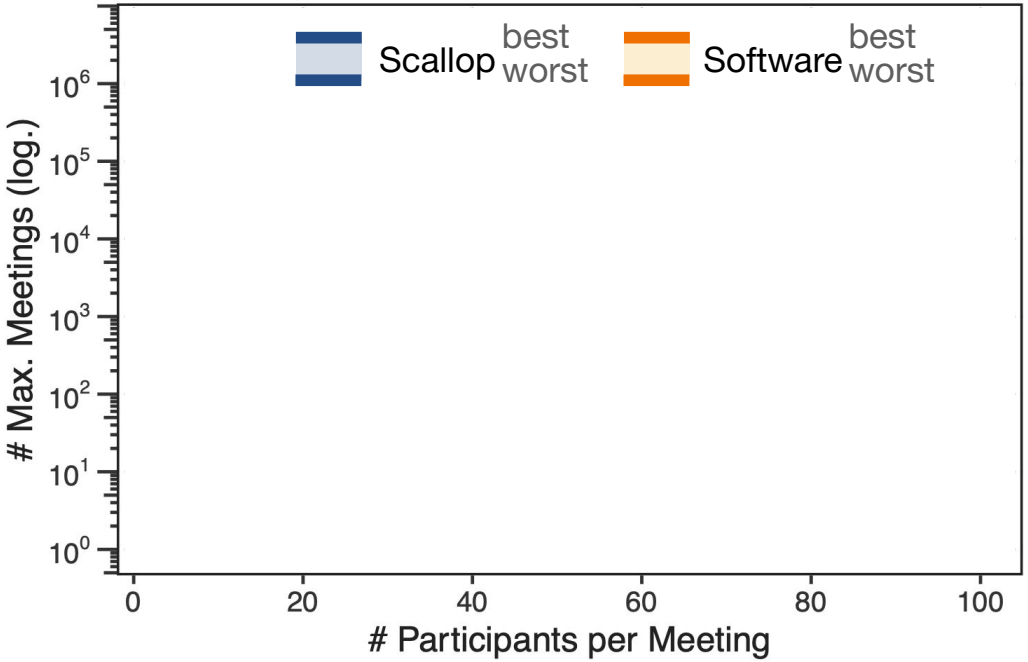
Protocol/Type	Packets	%	Per sec.	KBytes	%
RTP	170,870	94.5	284.3	166,762	99.47
- Audio	29,746	16.46	49.49	3826	2.28
- Video	141,124	78.09	234.81	162,935	97.19
- AV1 DS*	5	«	0.008	6	«
RTCP	9,153	5.06	15.22	801	0.48
- SR/SDES	3,456	1.91	5.75	304	0.18
- RR*	240	0.39	0.13	15	0.01
- RR/REMB*	5,457	3.02	9.07	482	0.29
STUN*	**695**	**0.38**	**1.15**	89	0.05
Control Plane	6,397	3.54	10.64	593	0.35
Data Plane	174,326	**96.46**	290.06	167,066	**99.65**
Total	180,718	100	300.69	167,653	100

96.46% packets

99.65% bytes

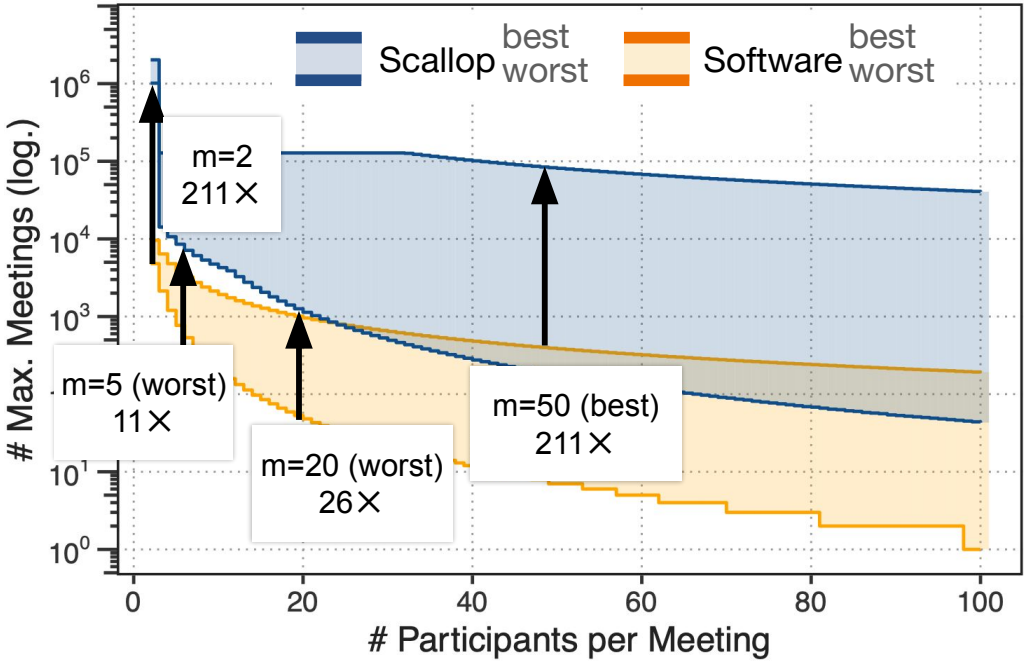
Evaluation

Scallop improves scalability over software by 7× to 211×



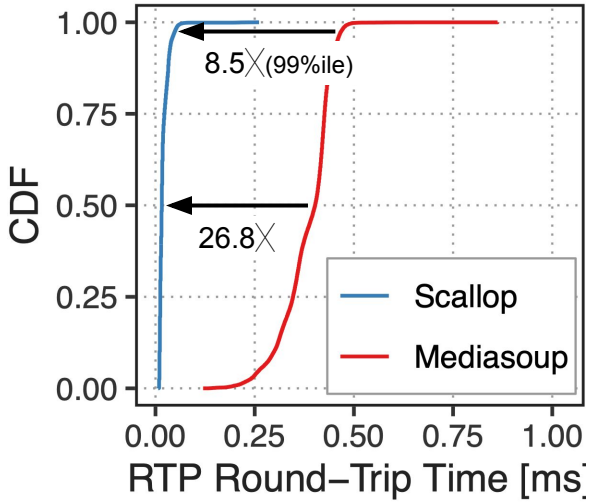
Evaluation

Scallop improves scalability over software by 7× to 211×



Evaluation

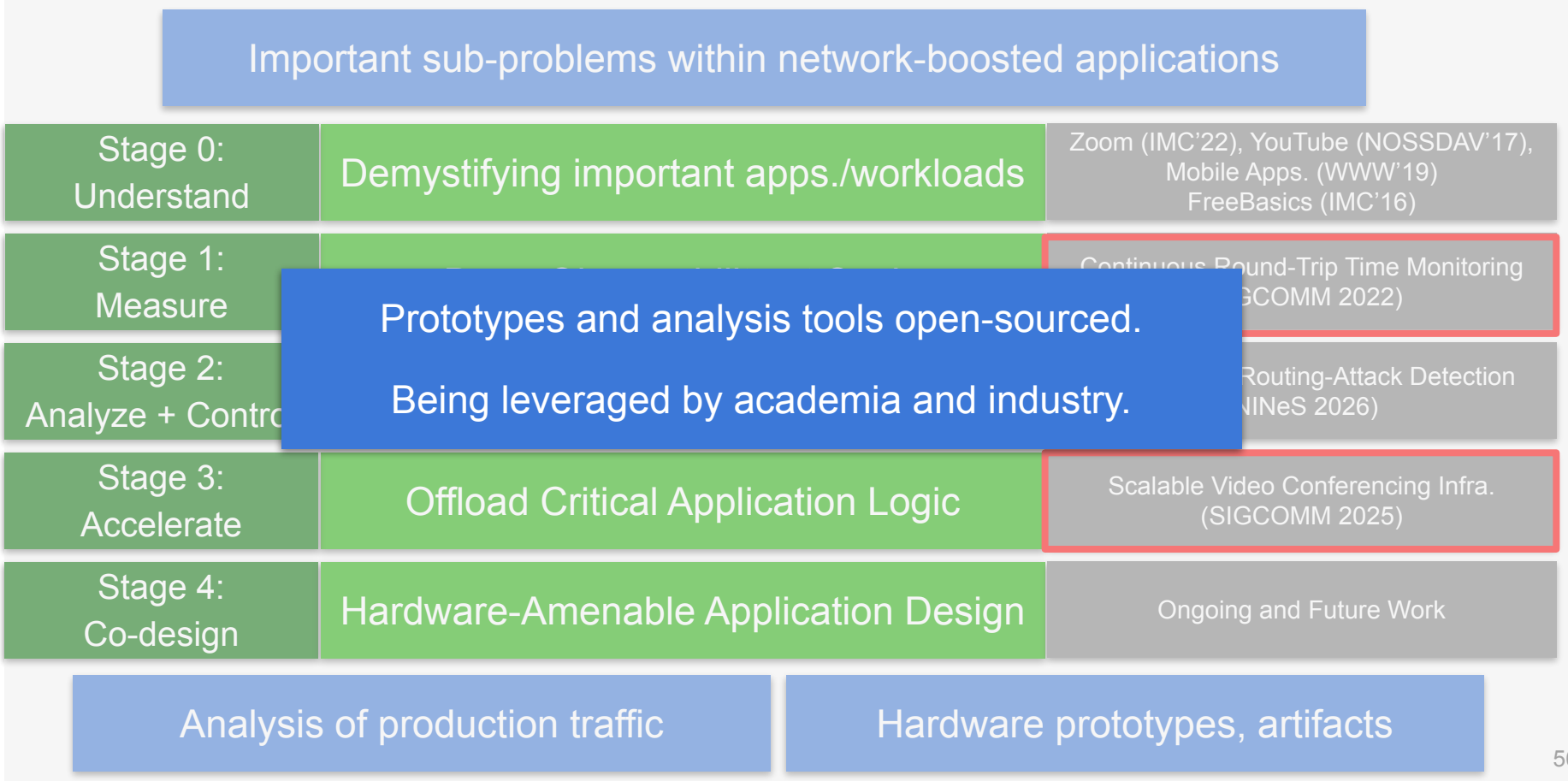
Scallop reduces SFU-induced latency: median by 27× and tail by 8×



Conclusion and Lessons

- **Novelty:** Hardware-software co-designed architecture for video conferencing
- **Demsytification:** Understand components of SFU operation
- **Hardware-amenable design:** Adapting PRE to achieve media replication
- **Key innovation:** Delineate fast-path operations from slow-path operations
- **Key lesson:** Monolithic implementations can reveal unique opportunities for network offloading

Recap: In This Talk



Future Work: Co-design Opportunities

Stage 4: Co-design	Hardware-Amenable Application Design

- Lessons learned about application-layer protocols (TCP, WebRTC) → easier to boost with hardware-amenable design

- Co-design future applications with the network

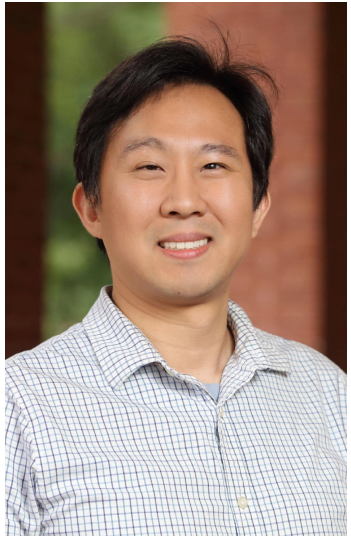
Acknowledgments

FPO Committee

Jennifer Rexford

Maria Apostolaki

Ravi Netravali

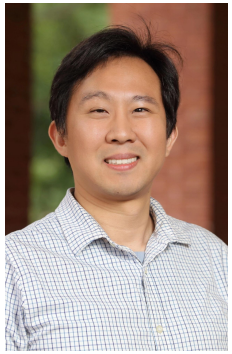


Hyojoon Kim

David Walker

Collaborators

Oliver Michel



Hyojoon Kim

Sophia Yoo

Daniel Jubas

Jennifer Rexford

Maria Apostolaki

Ravi Netravali

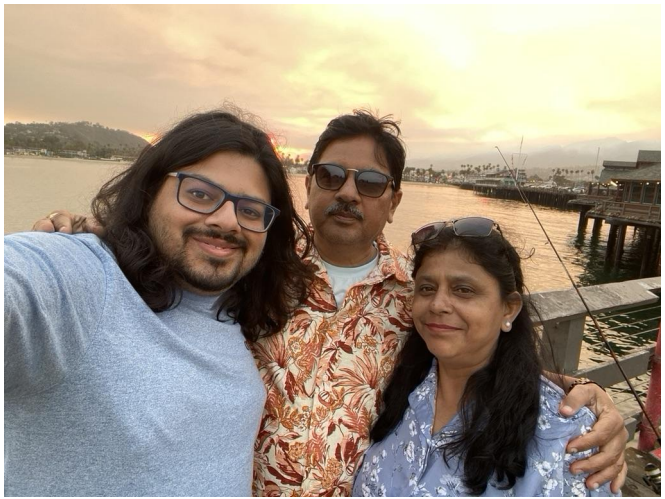
David Hay

Advisor

Jennifer Rexford

A Tale of Four Families

The Given Family



The Chosen Family



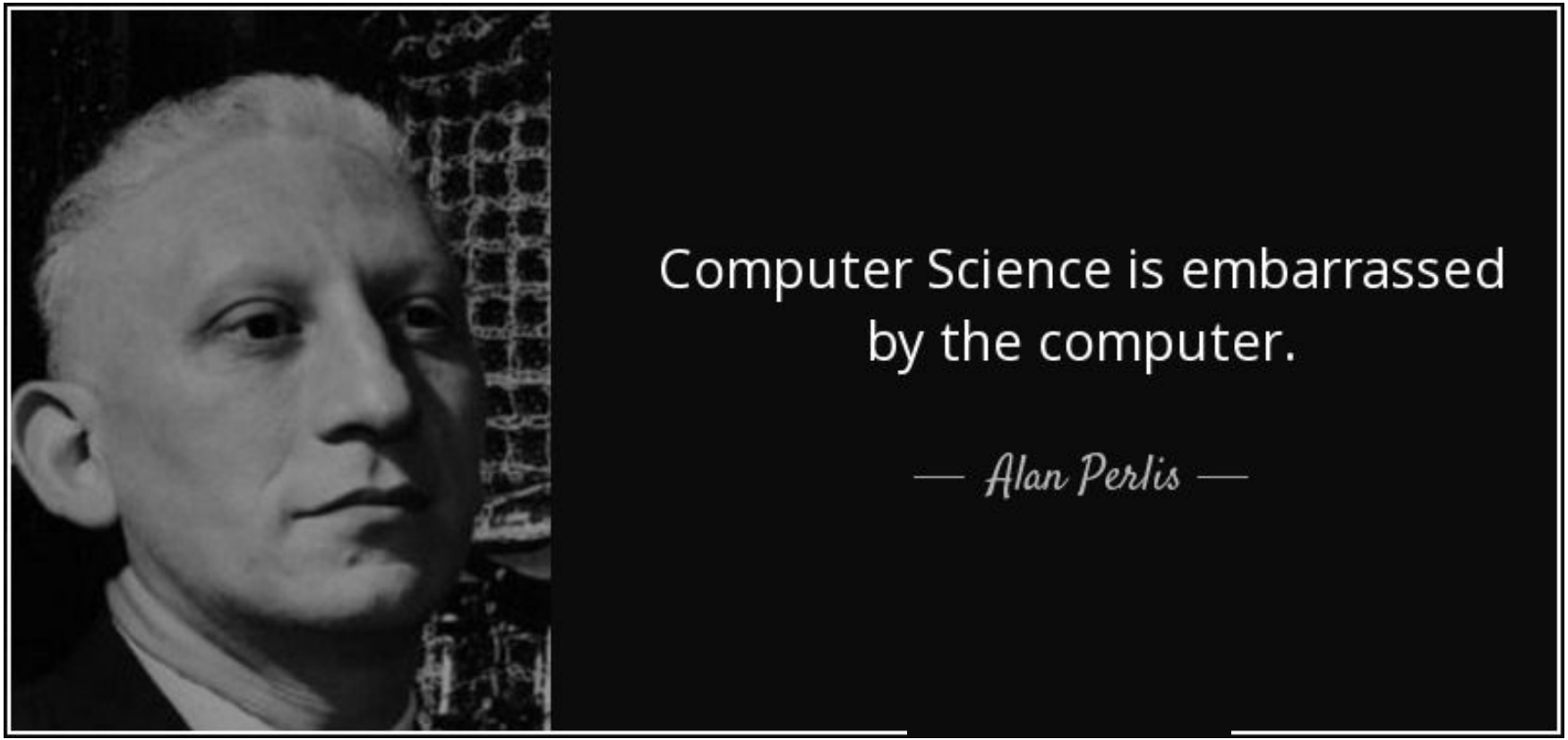
The Academic Family

The Future Family

Summarizing Time during Ph.D.

Summarizing Time during Ph.D.

Final Remarks



Alan J. Perlis: Pioneering work in PL, first Turing awardee
“Epigrams on Programming”, ACM SIGPLAN Notices, 1982