Engineering and Computing

UNIVERSITY OF SOUTH CAROLINA

Ph.D. Dissertation Defense

Leveraging Programmable Switches to Enhance the Performance of Networks: Active and Passive Deployments

Elie Kfoury

Integrated Information Technology Department
College of Engineering and Computing

Advisor:
- Dr. Jorge Crichigno

Committee Members:
- Dr. Elias Bou-Harb
- Dr. Robert Brookshire
- Dr. John Gerdes
- Dr. Neset Hikmet

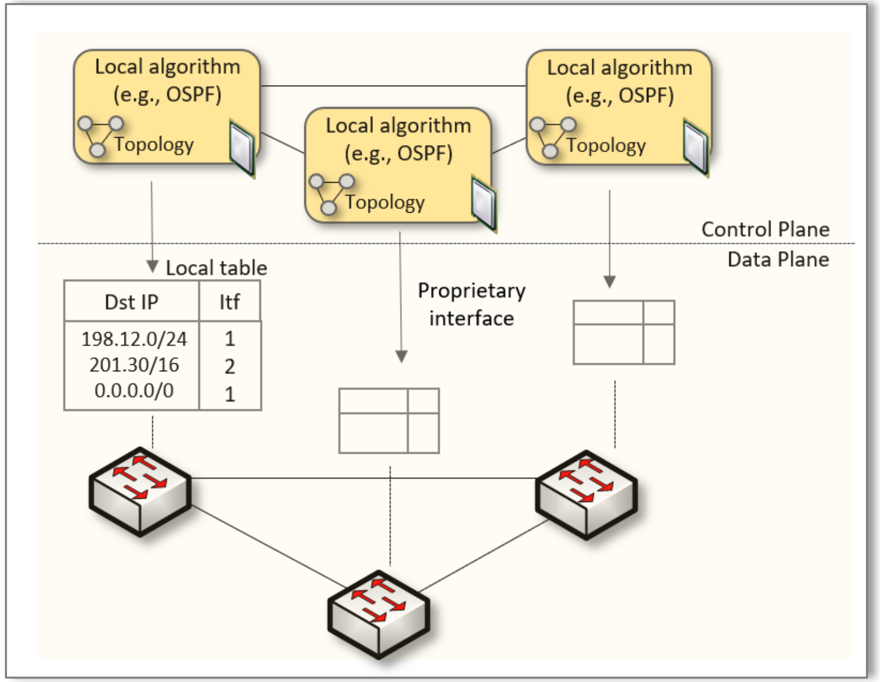
19 May 2023

Agenda

- Non-programmable Networks
- Software-defined Networking (SDN)
- P4 Programmable Switches
- Performance Issues in Networks Today
- Proposed System: Dynamic Buffer Sizing- Stanford
- Proposed System: P4BS
- Proposed System: P4CCI
- Proposed System: Media Offloading
- Proposed Architecture: P4Tune
- Conclusion

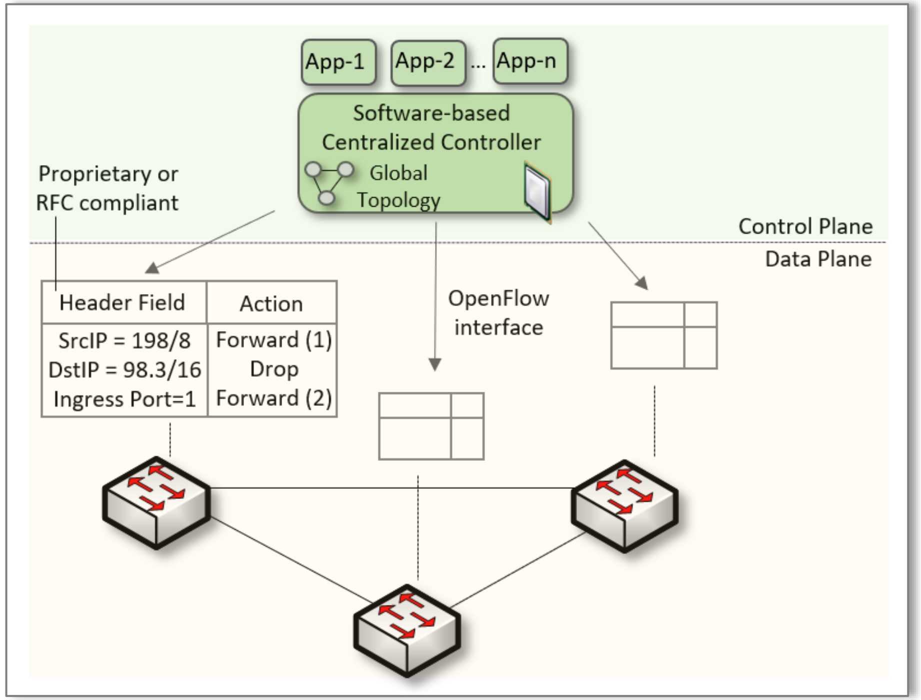
Non-programmable Networks

- Since the explosive growth of the Internet in the 1990s, the networking industry has been dominated by closed and proprietary hardware and software
- The interface between control and data planes has been historically proprietary
  - ➢ Vendor dependence: slow product cycles of vendor equipment, no innovation from network owners
  - ➢ A router is a monolithic unit built and internally accessed by the manufacturer only



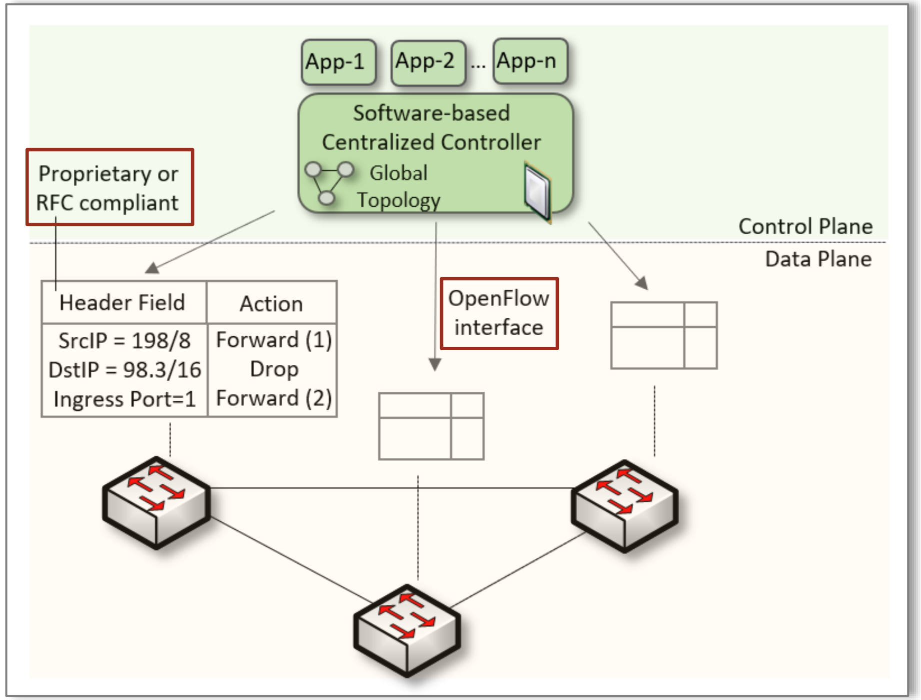
Software-defined Networking (SDN)

- Protocol ossification has been challenged first by SDN
- SDN explicitly separates the control and data planes, and implements the control plane intelligence as a software outside the switches
- The function of populating the forwarding table is now performed by the controller



SDN Limitation

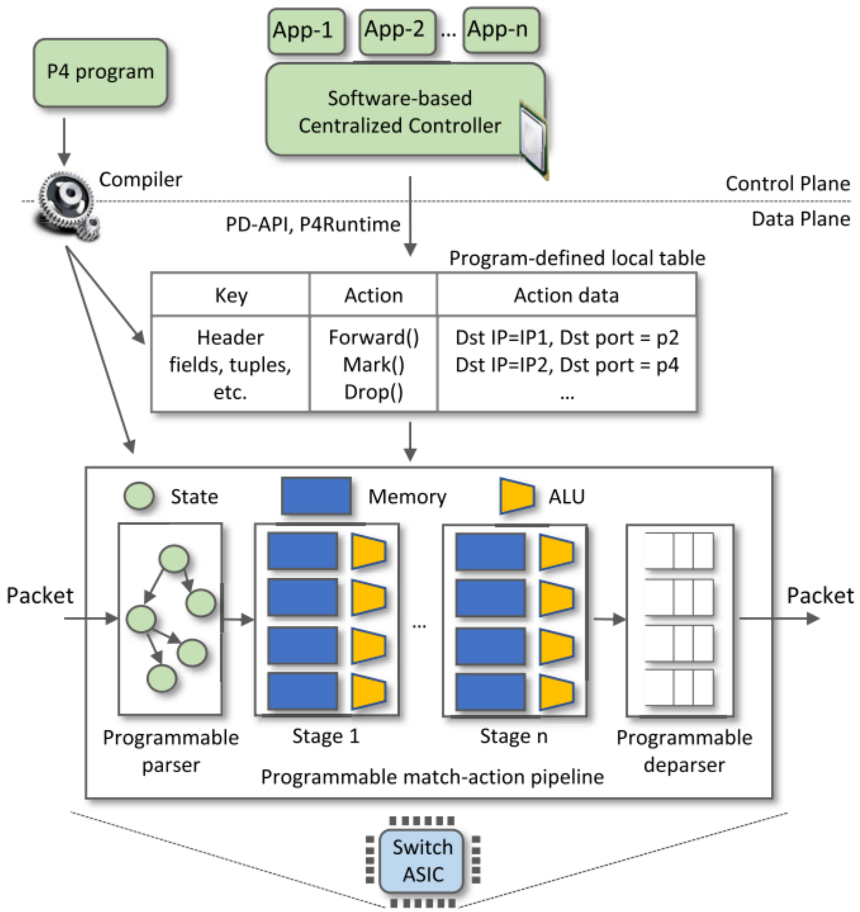
- SDN is limited to the OpenFlow specifications
  - Forwarding rules are based on a fixed number of protocols / header fields (e.g., IP, Ethernet)
- The data plane is designed with fixed functions (hard-coded)
  - Functions are implemented by the chip designer



P4 Programmable Switches

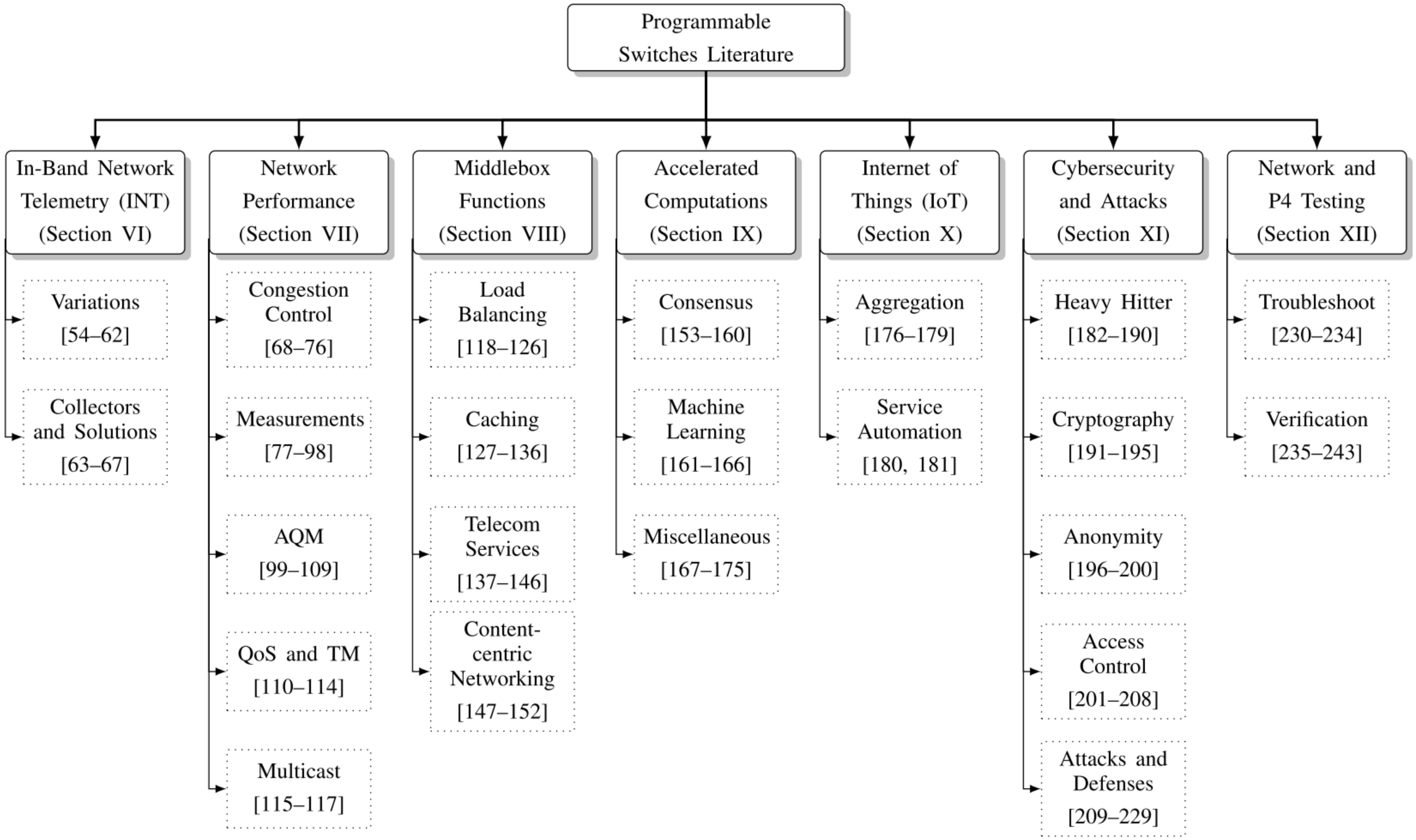
- P4[1] programmable switches permit a programmer to program the data plane
  - Define and parse new protocols
  - Customize packet processing functions
  - Measure events occurring in the data plane with high precision
  - Offload applications to the data plane

- Programmable Data Planes (PDPs)



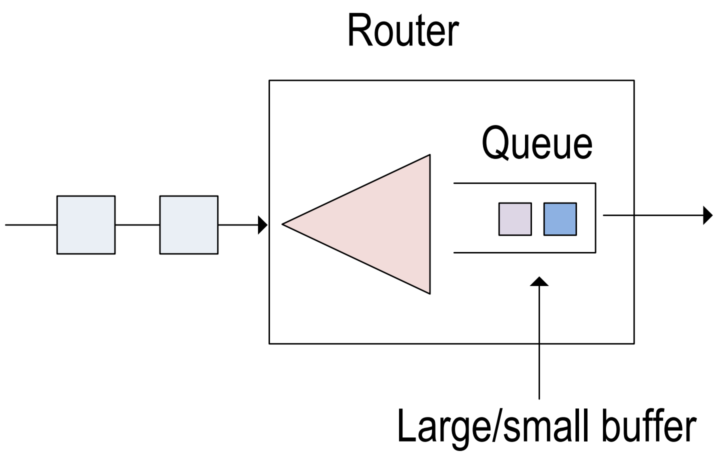
1. P4 stands for stands for Programming Protocol-independent Packet Processors

PDPs Applications



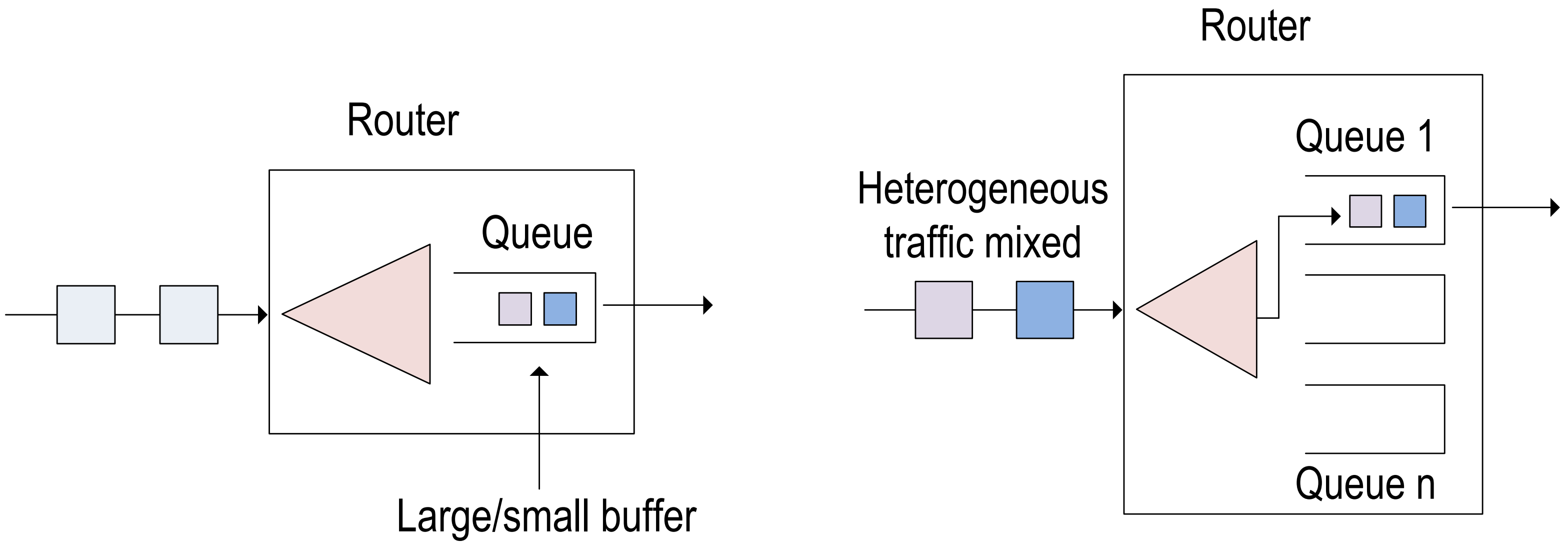
Performance Issues in Networks Today

- Three main issues affecting the performance of networks today:
  1. Networks are dominated by under/over buffered routers/switches



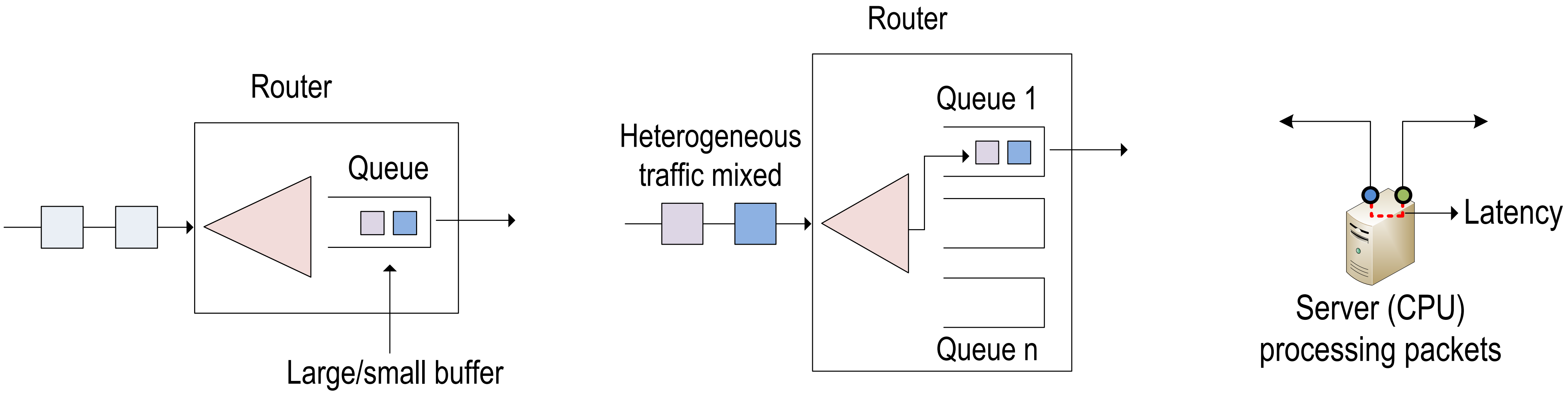
Performance Issues in Networks Today

- Three main issues affecting the performance of networks today:
  1. Networks are dominated by under/over buffered routers/switches
  2. Routers/switches configured with best-effort quality of service (heterogeneous traffic being mixed without any QoS measures)



Performance Issues in Networks Today

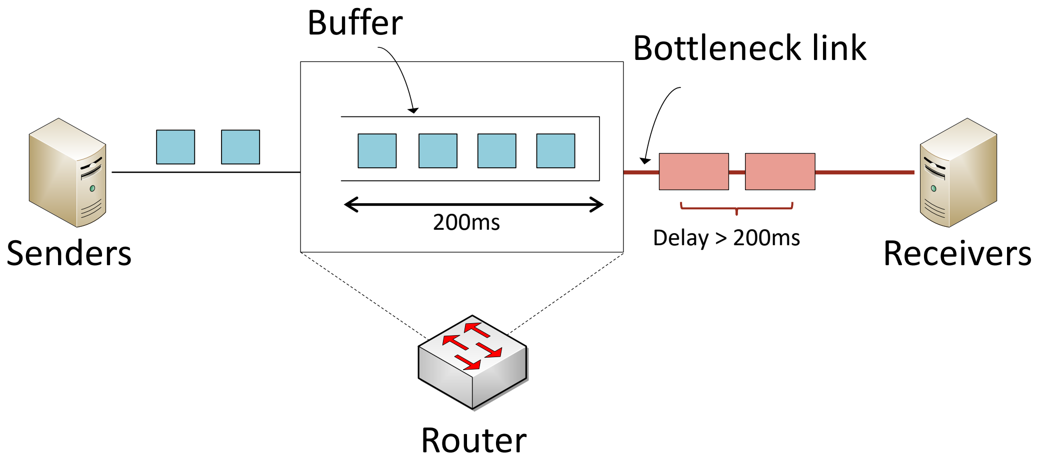
- Three main issues affecting the performance of networks today:
  1. Networks are dominated by under/over buffered routers/switches
  2. Routers/switches configured with best-effort quality of service (heterogeneous traffic being mixed without any QoS measures)
  3. CPU-based middlebox servers inducing latency and jitter and not keeping up with high traffic rates



Performance Problem #1:
Under/over Buffered Networks

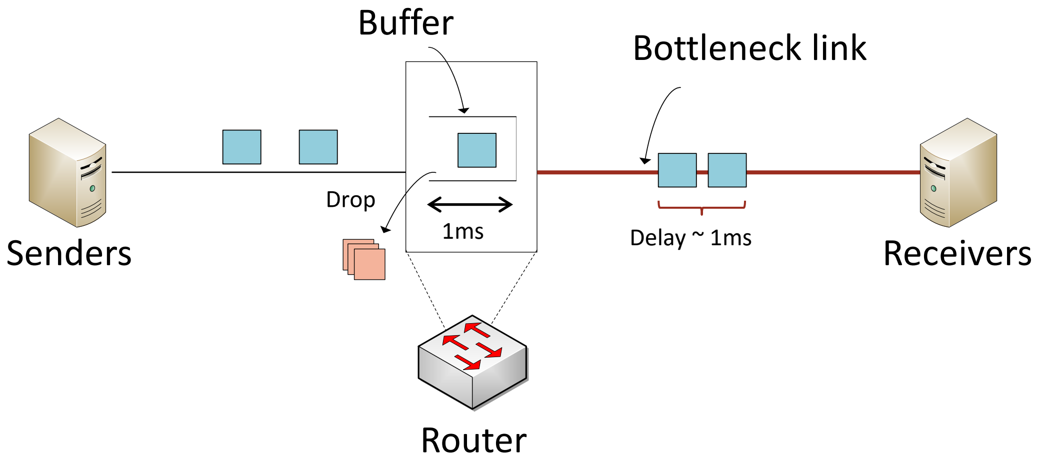
Buffer Size Problem

- Routers and switches have a memory referred to as packet buffer
- The size of the buffer impacts the network performance
  - Large buffers → TCP keeps the buffer full → excessive delays, Bufferbloat



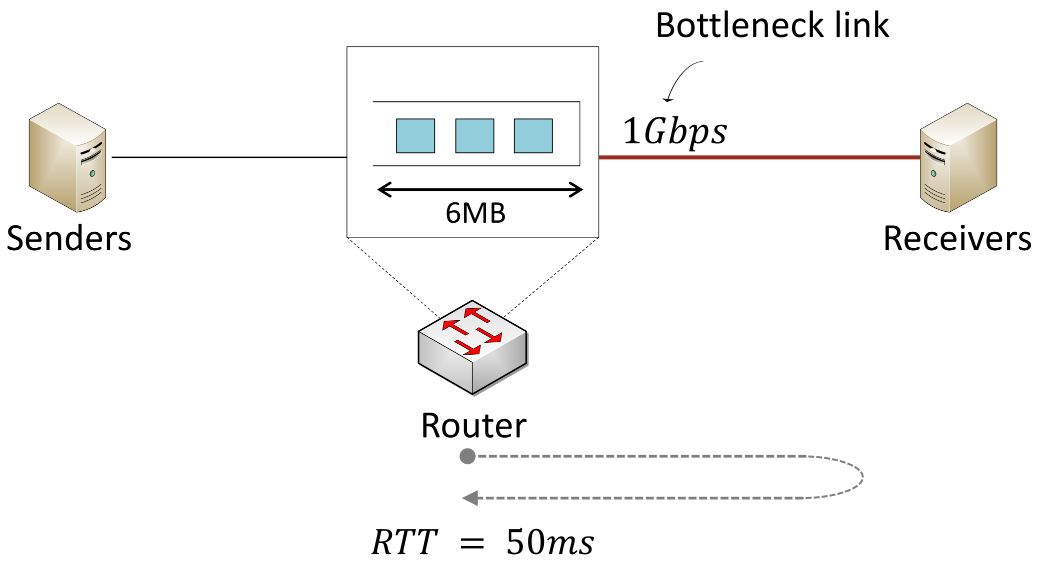
Buffer Size Problem

- Routers and switches have a memory referred to as packet buffer
- The size of the buffer impacts the network performance
  - ➢ Large buffers → TCP keeps the buffer full → excessive delays, Bufferbloat
  - ➢ Small buffers → packet drops → sender slows down → low link utilization



Buffer Sizing Rules: BDP

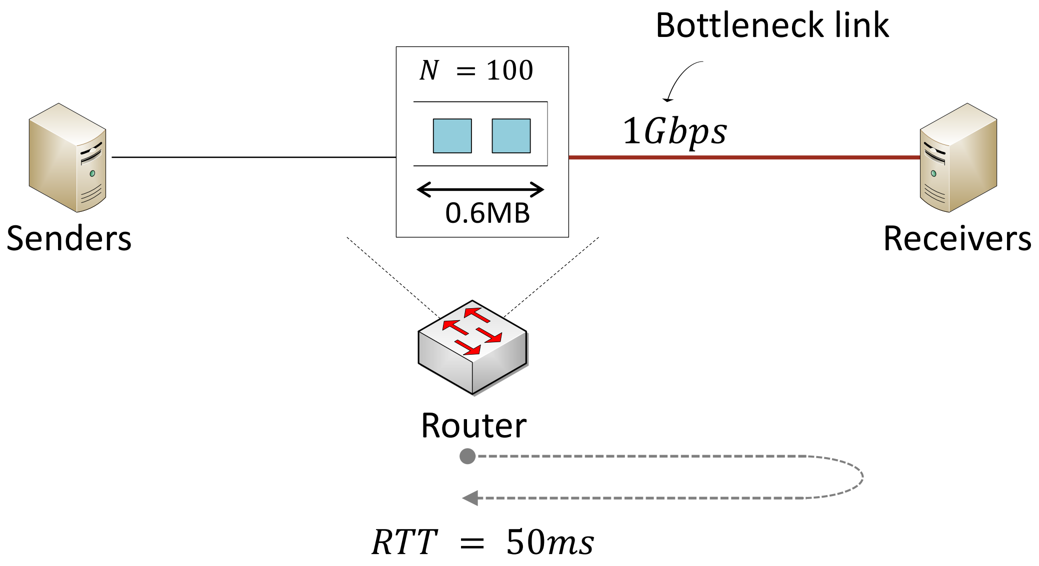
- General rule-of-thumb[1]: bandwidth-delay product (older rule)
  - $B = C \cdot RTT$
  - C is the capacity of the link and RTT is the average round-trip time
- Example: $C = 1Gbps$ with $RTT = 50ms$ → $B = 6\text{MB}$



1. C. Villamizar and C. Song. High performance TCP in ansnet. ACM Computer Communications Review, 24(5):45–60, 1994 199

Buffer Sizing Rules: Stanford

- Stanford rule[1]: smaller buffers are enough to get full link utilization
  - $B = \frac{C * RTT}{\sqrt{N}}$
  - N is the number of long (persistent over time) flows traversing the link
- Example: $C = 1Gbps$ with $RTT = 50ms$ and 100 flows → $B = 0.6\text{MB}$



1. Appenzeller, Guido, Isaac Keslassy, and Nick McKeown. "Sizing router buffers." *ACM SIGCOMM Computer Communication Review* 34.4 (2004)

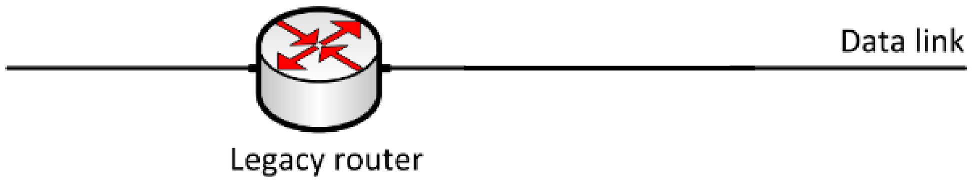
Buffer Size Rules

- Problem: RTT, N, and other metrics, continuously change over time
- Most networks today use very large buffers[1]
- Questions:
  - Is it possible to change the buffer size dynamically?
  - How to identify the small number of long flows from all the flows?
  - How to calculate the average RTT from millions of flows at line rate?
  - Are the existing buffer sizes adequate for all traffic scenarios?

1. McKeown, Nick, Guido Appenzeller, and Isaac Keslassy. "Sizing router buffers (redux)." ACM SIGCOMM Computer Communication Review (2019)

Proposed System

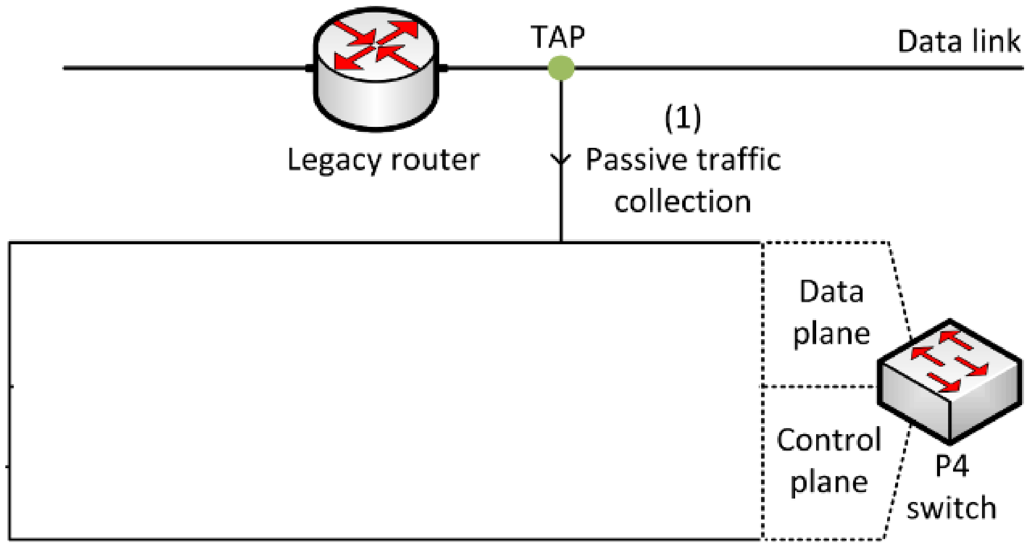
- The buffer size is dynamically modified and set to $B = \frac{C * RTT}{\sqrt{N}}$



Proposed System

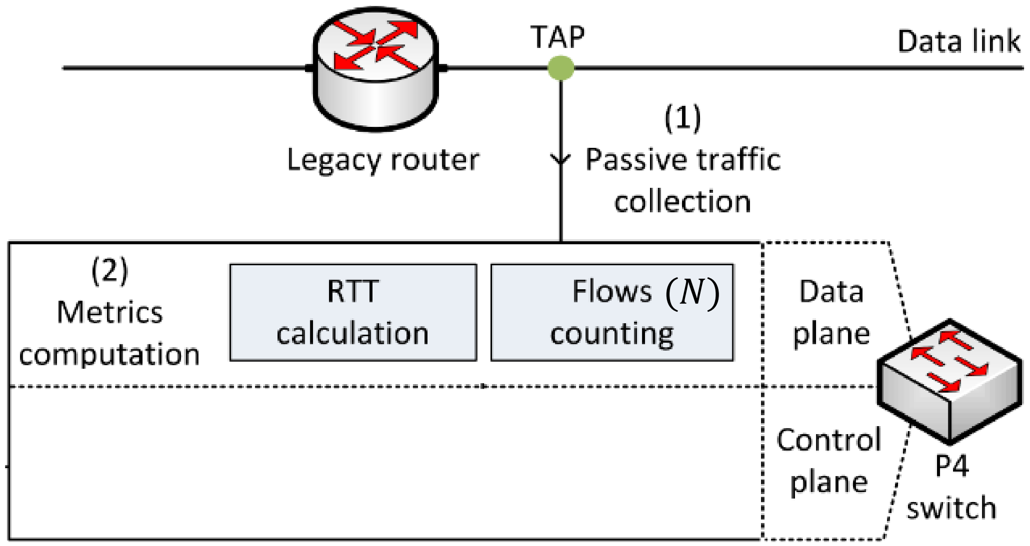
- The buffer size is dynamically modified and set to $B = \frac{C * RTT}{\sqrt{N}}$

1. Copy of the traffic is forwarded to a programmable switch using TAPs



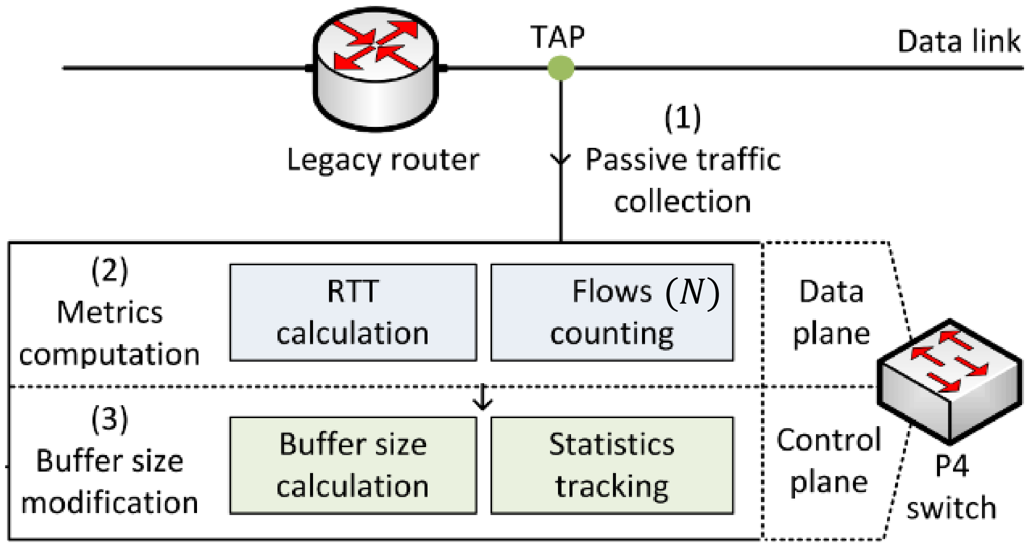
Proposed System

- The buffer size is dynamically modified and set to $B = \frac{C * RTT}{\sqrt{N}}$
  1. Copy of the traffic is forwarded to a programmable switch using TAPs
  2. The programmable switch identifies and track buffer sizing metrics



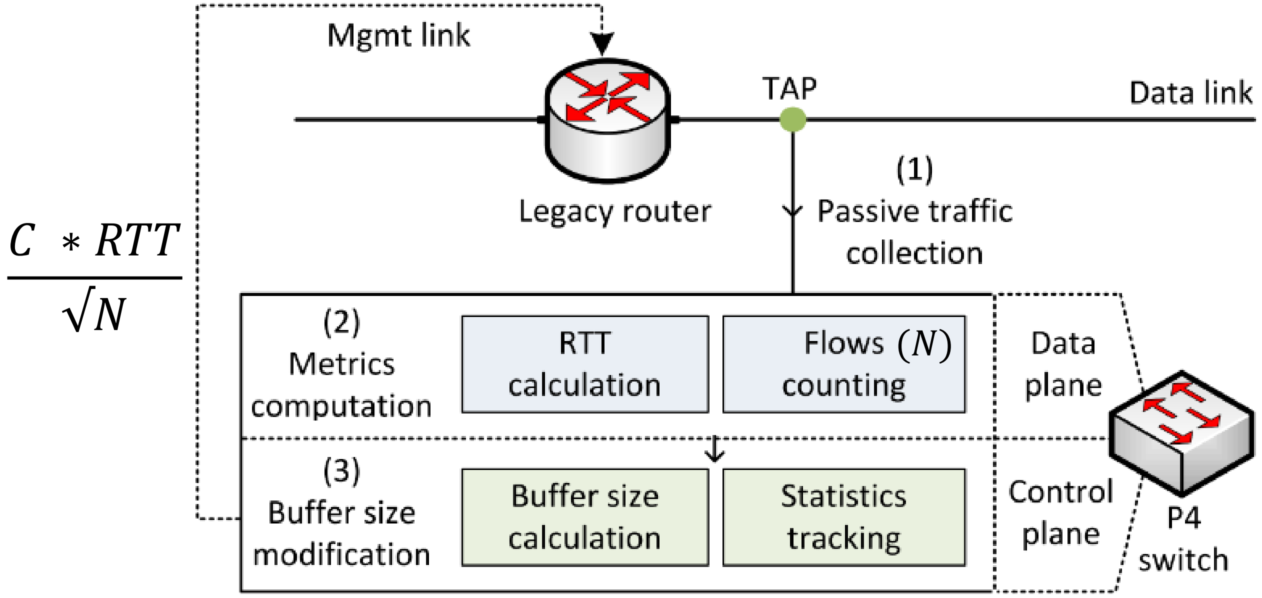
Proposed System

- The buffer size is dynamically modified and set to $B = \frac{C * RTT}{\sqrt{N}}$
  1. Copy of the traffic is forwarded to a programmable switch using TAPs
  2. The programmable switch identifies and track buffer sizing metrics
  3. The programmable switch modifies the legacy router's buffer size



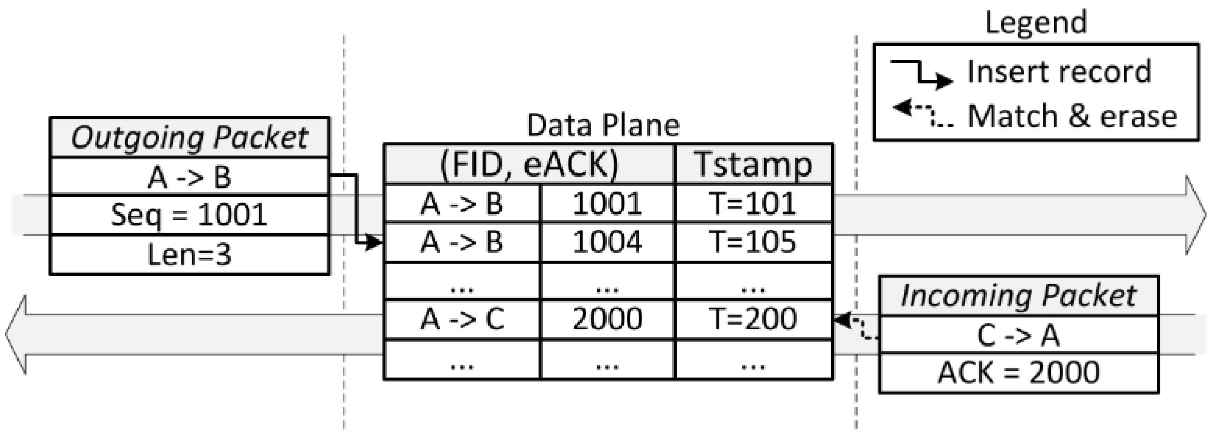
Proposed System

- The buffer size is dynamically modified and set to $B = \frac{C * RTT}{\sqrt{N}}$
  1. Copy of the traffic is forwarded to a programmable switch using TAPs
  2. The programmable switch identifies and track buffer sizing metrics
  3. The programmable switch modifies the legacy router's buffer size



RTT Calculation

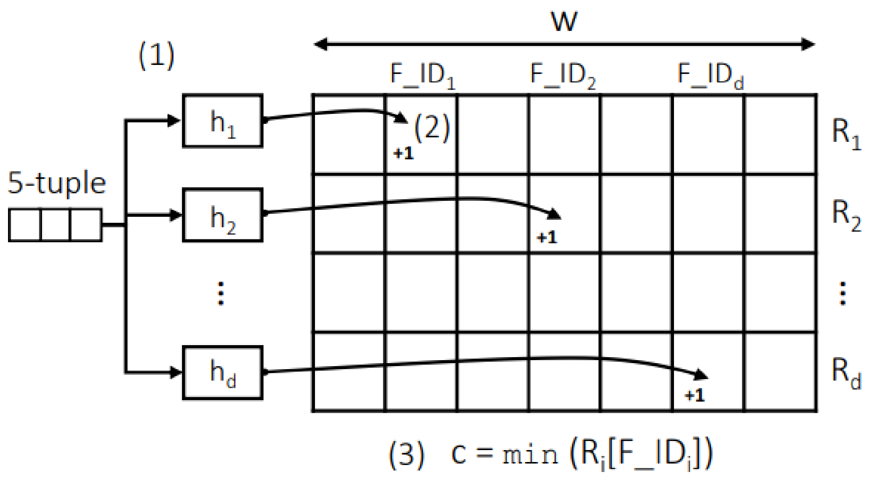
- Relate the TCP sequence (SEQ) and acknowledgement (ACK) numbers[1]
- The RTT is calculated as the time difference between the two packets



[1]Chen, Xiaoqi, et al. "Measuring TCP round-trip time in the data plane." Workshop on Secure Programmable Network Infrastructure. 2020.

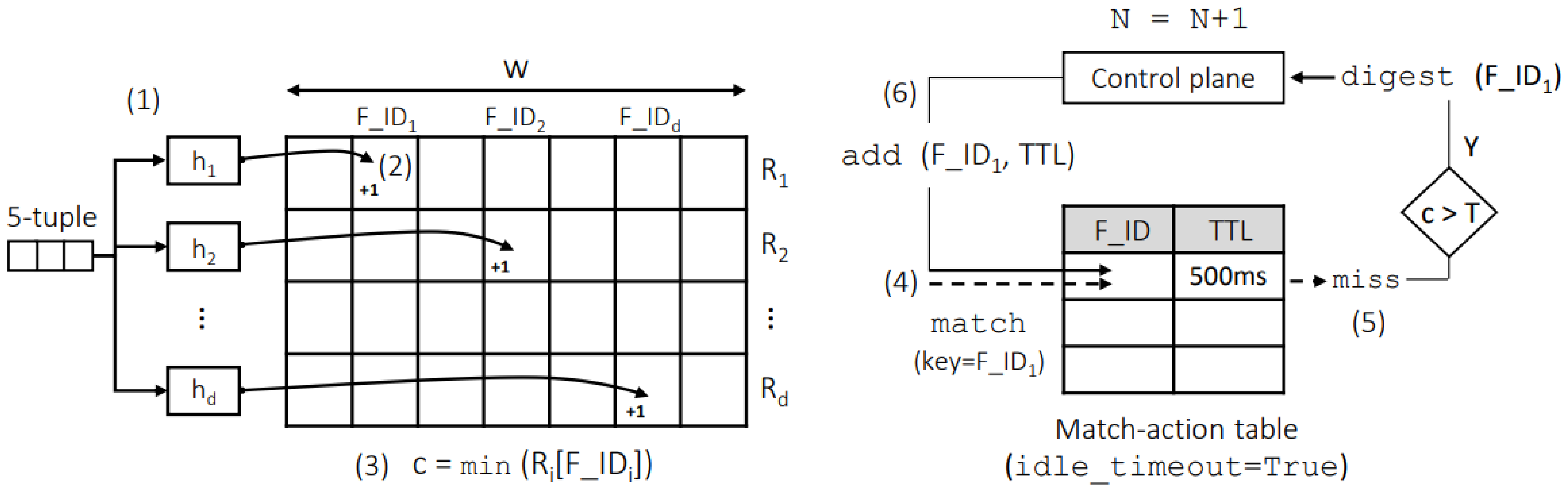
Long Flows Counting

- The Count-Min Sketch (CMS) is used to store the counts of the flows



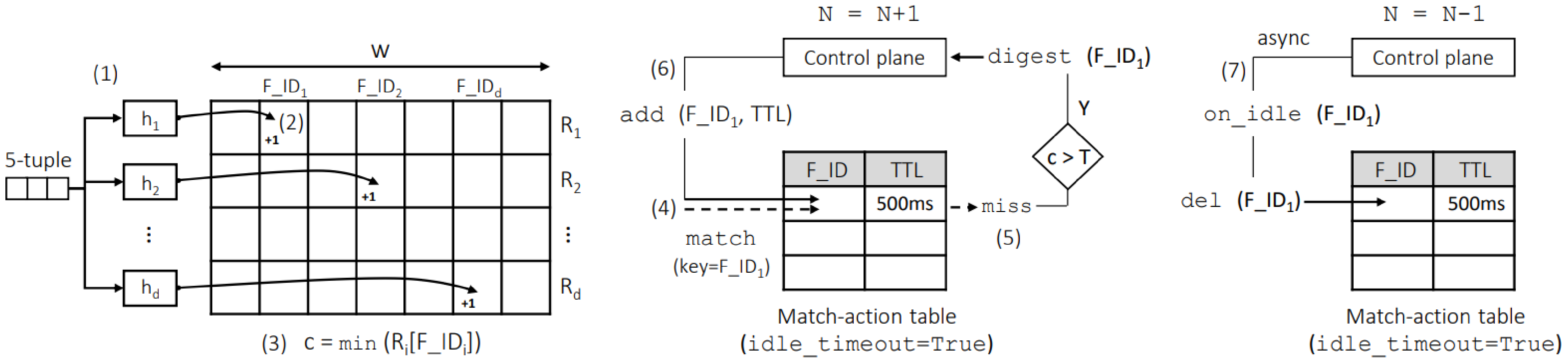
Long Flows Counting

- The Count-Min Sketch (CMS) is used to store the counts of the flows
- If the minimum exceeds a predefined threshold, the flow is identified as long



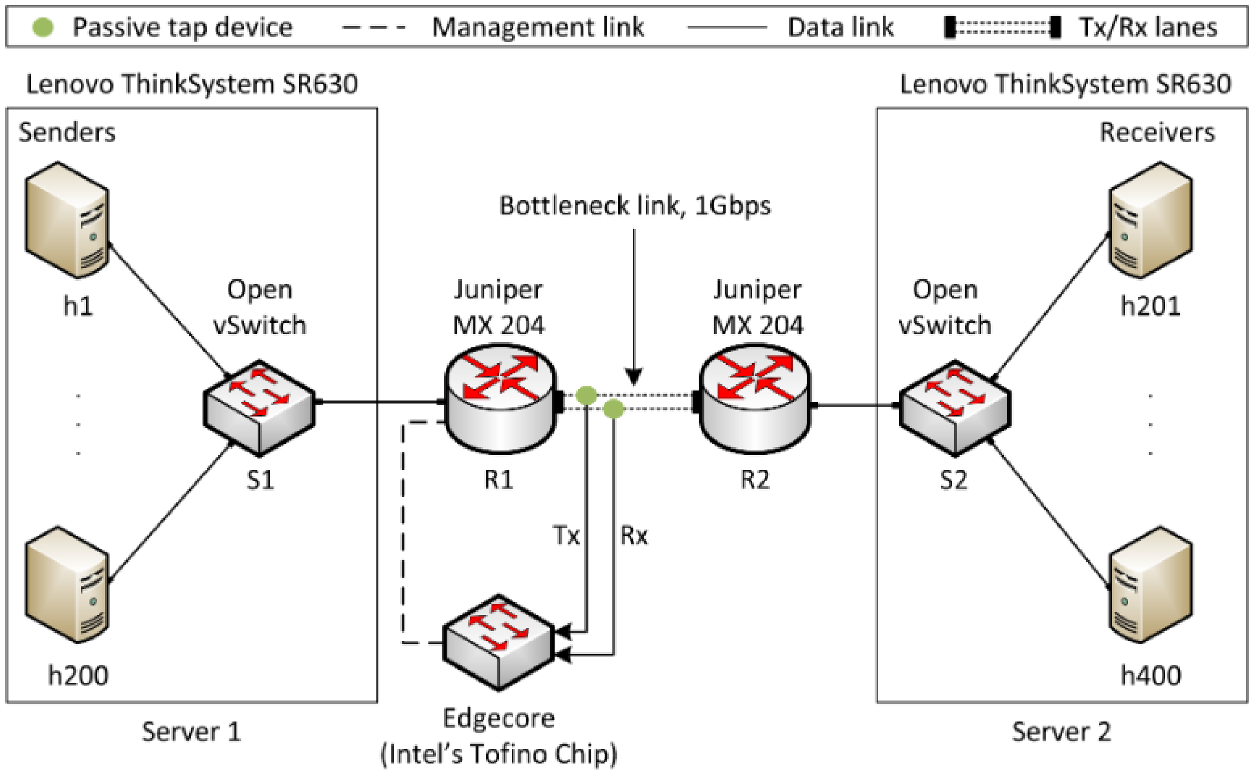
Long Flows Counting

- The Count-Min Sketch (CMS) is used to store the counts of the flows
- If the minimum exceeds a predefined threshold, the flow is identified as long
- Table timeouts are used to evict flows



Implementation and Evaluation

- Topology and experimental setup
- Different congestion control algorithms[1]
- iPerf3
- Default buffer size of the router is 200ms[2]
- Wedge100BF-32X, ASIC chip (Intel's Tofino)



[1]Mishra et al. "The great Internet TCP congestion control census," ACM on Measurement and Analysis of Computing Systems, 2019

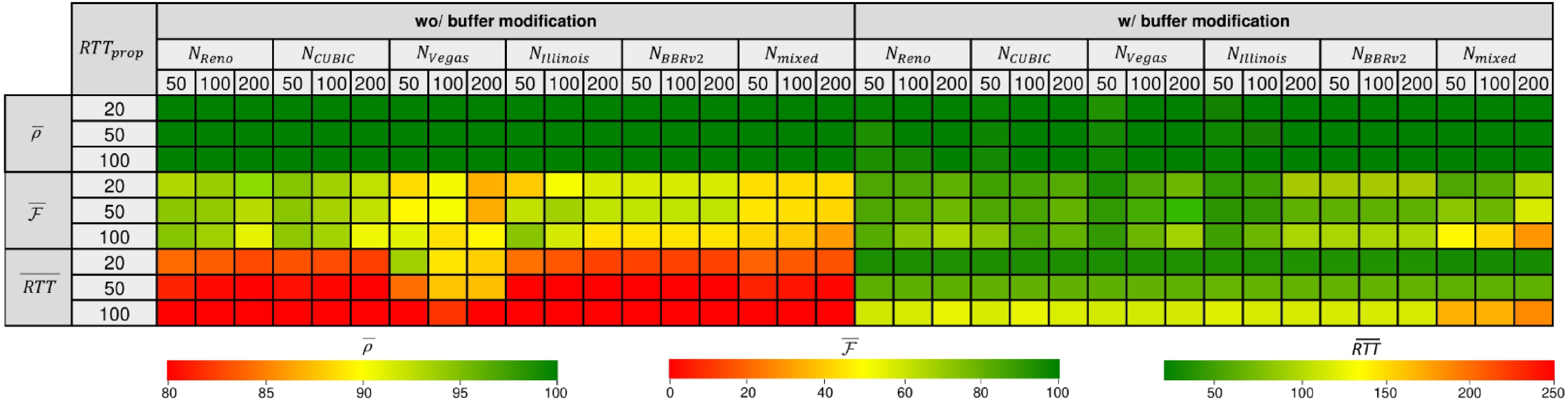
[2]N. McKeown et al. "Sizing router buffers (redux)," ACM SIGCOMM Computer Communication Review, vol. 49, no. 5

Implementation and Evaluation

- Two scenarios are considered:
  1. Default buffer size on the router, without any dynamic modification
  2. P4 switch measures and modifies the buffer size of the router

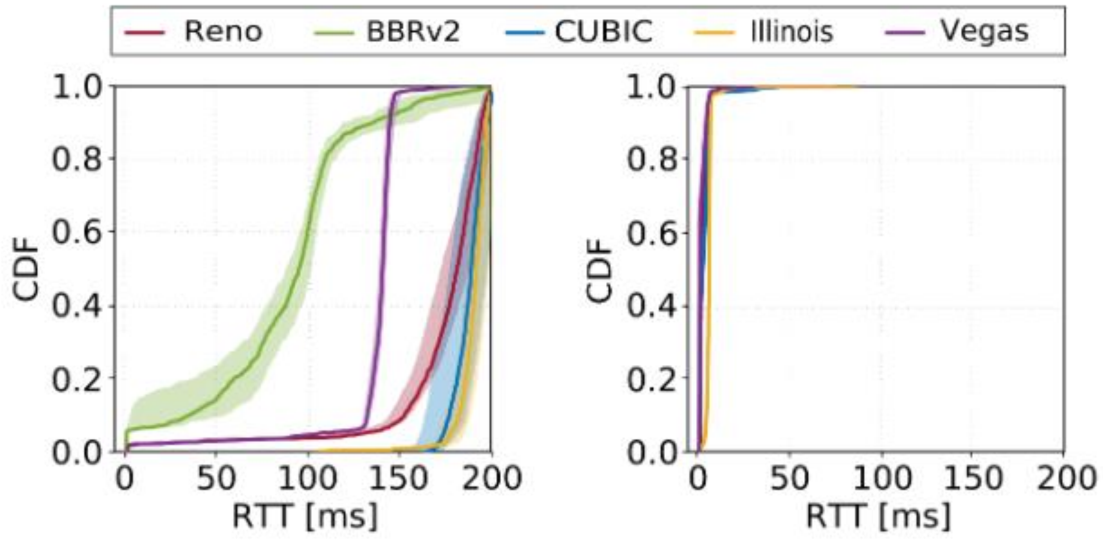
Results

- Various number of long flows, CCAs, and propagation delays
- Average link utilization $(\overline{\rho})$
- Average fairness index $(\overline{\mathcal{F}})$
- Average RTT $(\overline{RTT})$

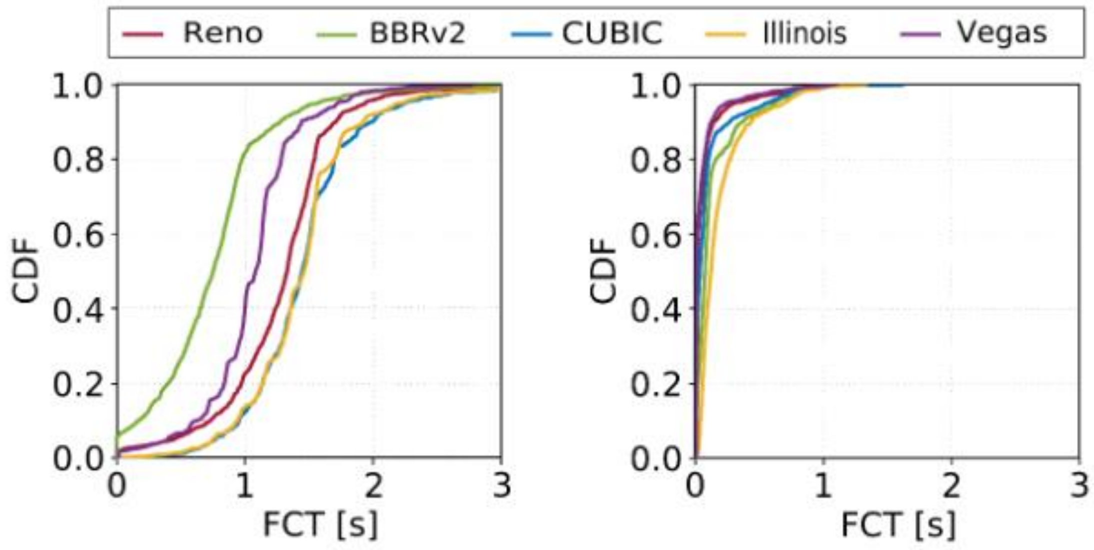


Results

- Performance of short flows sharing the bottleneck with long flows
- 1000 short flows are arriving according to a Poisson process
- Flow size distribution resembles a web search workload (10KB to 1MB)
- Background traffic: 200 long flows, propagation delay = 50ms



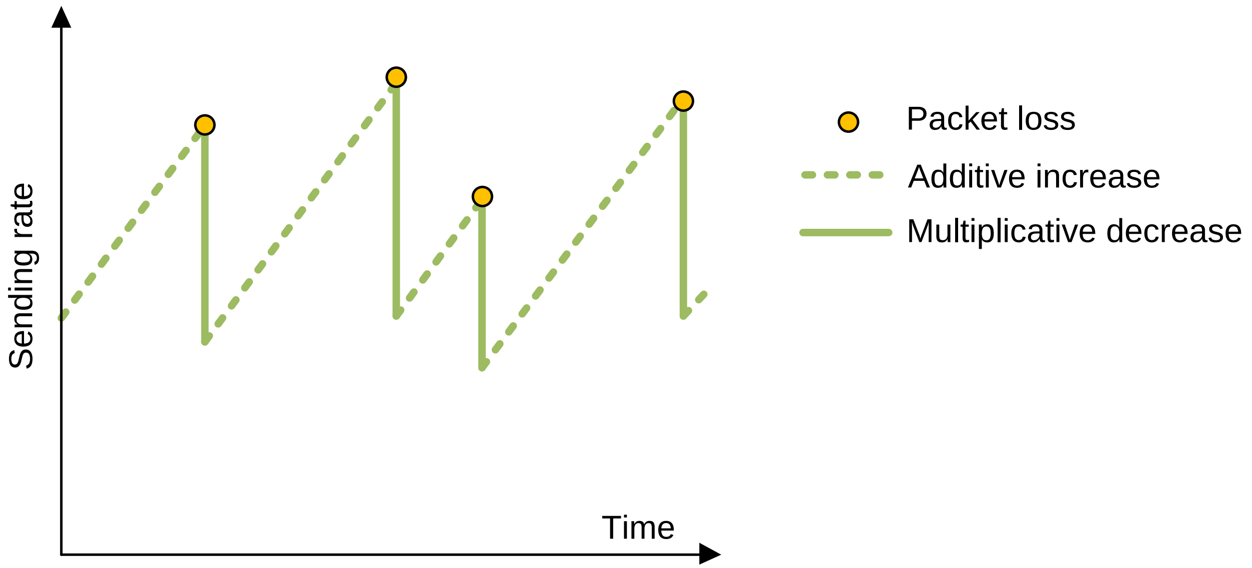
wo/ buffer modification w/ buffer modification



wo/ buffer modification w/ buffer modification

Discussions

- The proposed system improved:
  - The FCT and the RTT of short flows
  - The fairness and the RTT of long flows
- However, packet loss rates increased
- The buffer size assumes Additive Increase Multiplicative Decrease flows



Design Goals

- Dynamic adaptation to heterogeneous traffic
- Service-level Agreement (SLA) compliance
- Smooth integration in existing networks
- Extensibility

Design Goals

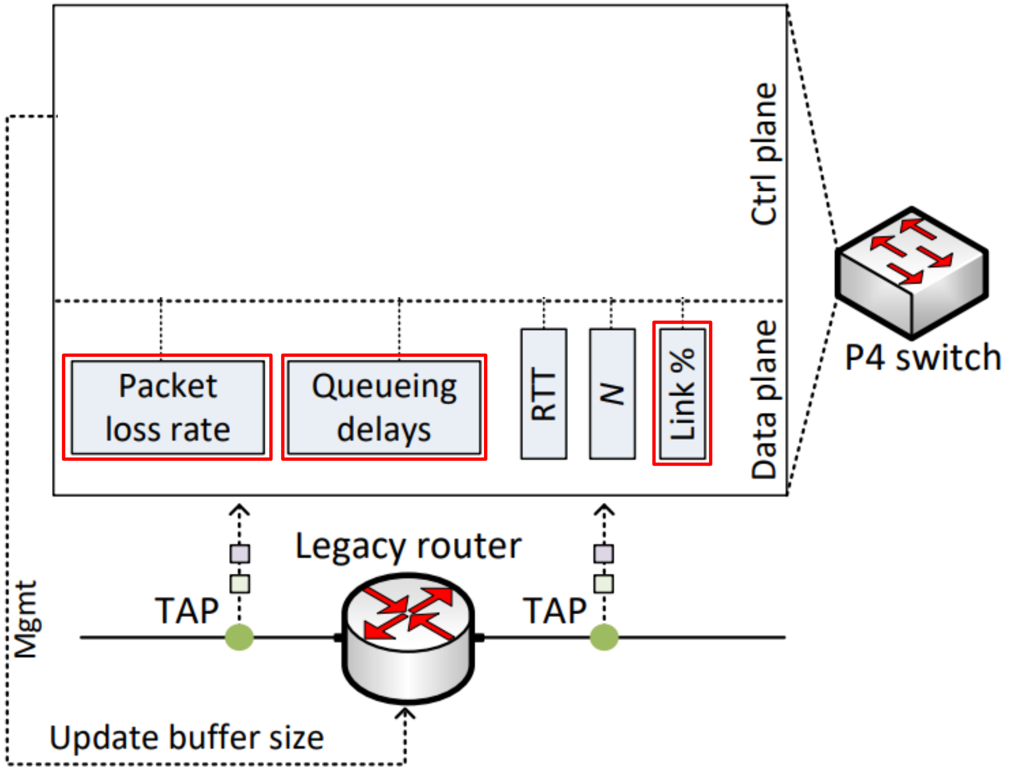
- Dynamic adaptation to heterogeneous traffic
- Service-level Agreement (SLA) compliance
- Smooth integration in existing networks
- Extensibility

Rule		Constraint			Deployability		Thresh	
Mode	**Name**	ρ	d	p	**CC-A**	**CR**	$\hat{d}$	$\hat{p}$
Static	BDP [45]	✓	×	×	×	✓	×	×
	Stanford [46]	✓	×	×	×	✓	×	×
	Tiny [82]	✓	×	×	×	✓	×	×
	BSCL [60]	✓	✓	✓	×	×	✓	✓
Dynamic	FPQ [62]	✓	✓	×	×	×	×	×
	ADT [84]	✓	×	×	×	×	×	×
	ABS [85]	✓	×	✓	✓	×	✓	✓
	P4BS	✓	✓	✓	✓	✓	✓	✓

ρ: Link utilization, d: queueing delay, p: packet loss, CC-A: congestion control-agnostic, CR: current routers, $\hat{d}$: delay bound, $\hat{p}$: loss bound

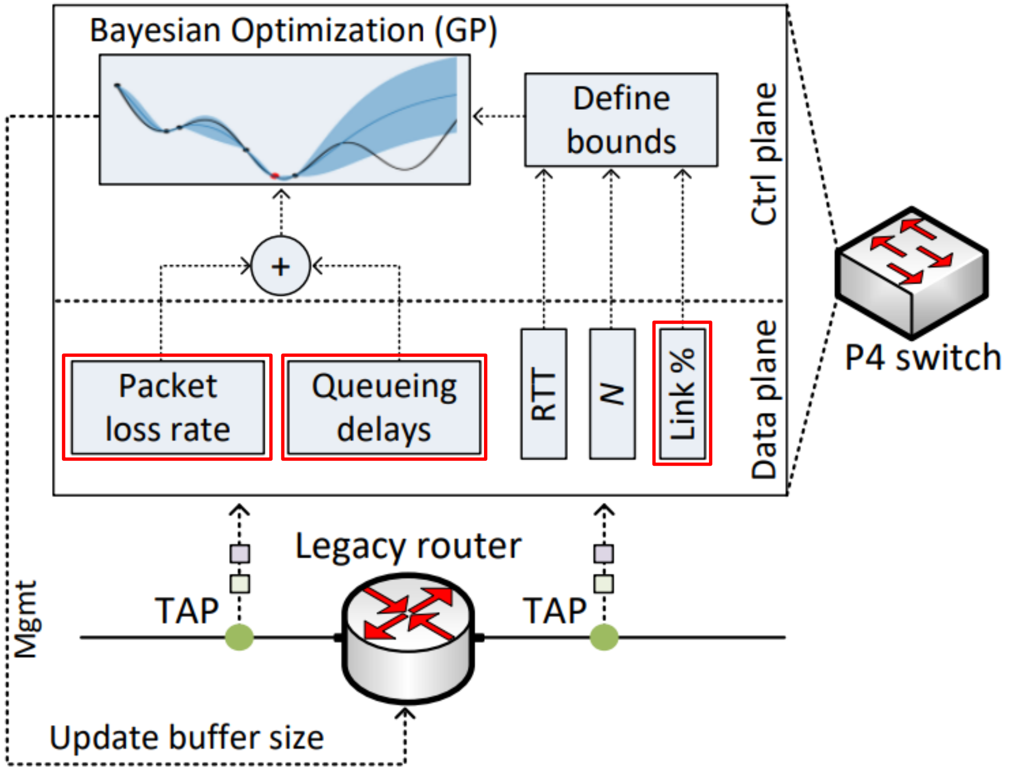
Proposed System

- The buffer size is dynamically modified
- A P4 switch is deployed passively to compute:
  - Number of long flows
  - Average RTT
  - Queueing delays ← New
  - Packet loss rates ← New
  - Link utilization ← New



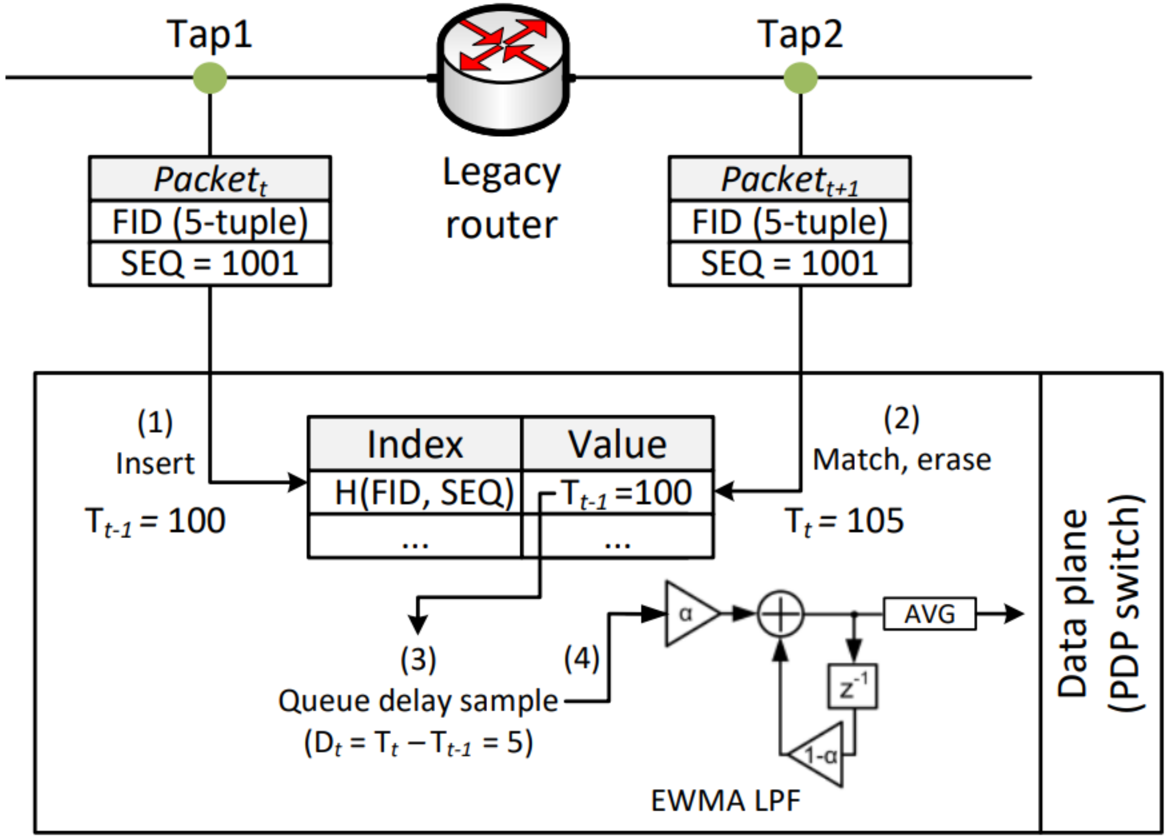
Proposed System

- The buffer size is dynamically modified
- A P4 switch is deployed passively to compute:
  - Number of long flows
  - Average RTT
  - Queueing delays ← New
  - Packet loss rates ← New
  - Link utilization ← New
- The control plane sequentially searches for a buffer that minimizes delays and losses
- The searching algorithm is Bayesian Optimization (BO) with Gaussian Processes



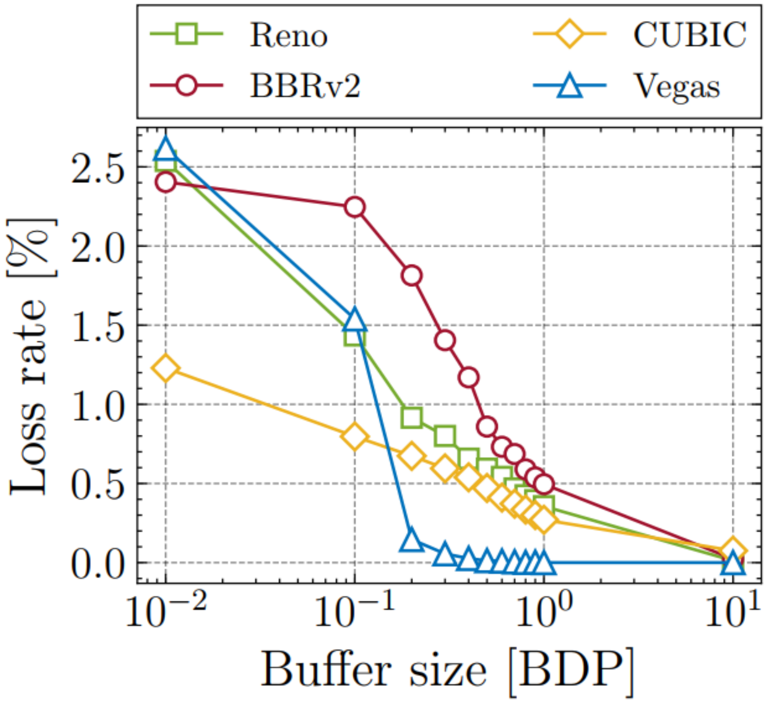
Queue Delay Calculation

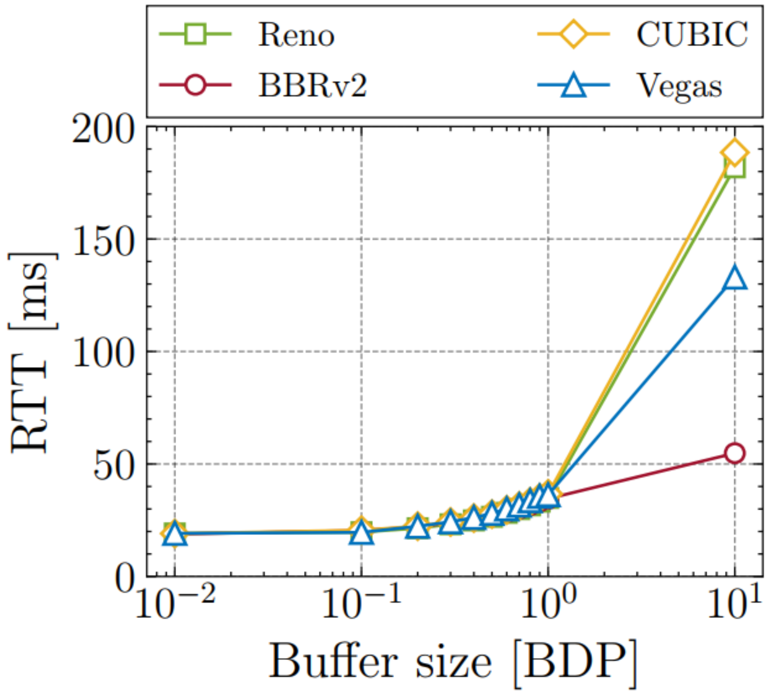
- The queueing delay is calculated by leveraging the precise timer of the hardware switch (nanosecond resolution)
- The queueing delay sample is fed to an Exponentially Weighted Moving Average (EWMA)



Impact of Buffer on Loss and Delay

- Intrinsic relationship between packet loss/delay and buffer
- Increasing the buffer → packet loss decreases, delay increases





Goal

- Find the buffer size that minimizes the losses and the delays
- Performance function: $f(.) = -[\, w_1 f_1(.) + w_2 f_2(.)\,]$ $\sum_{i=1}^{2} w_i = 1, w_i \in [0,1]$

  (f_1: Packet loss; f_2: Queueing delay)

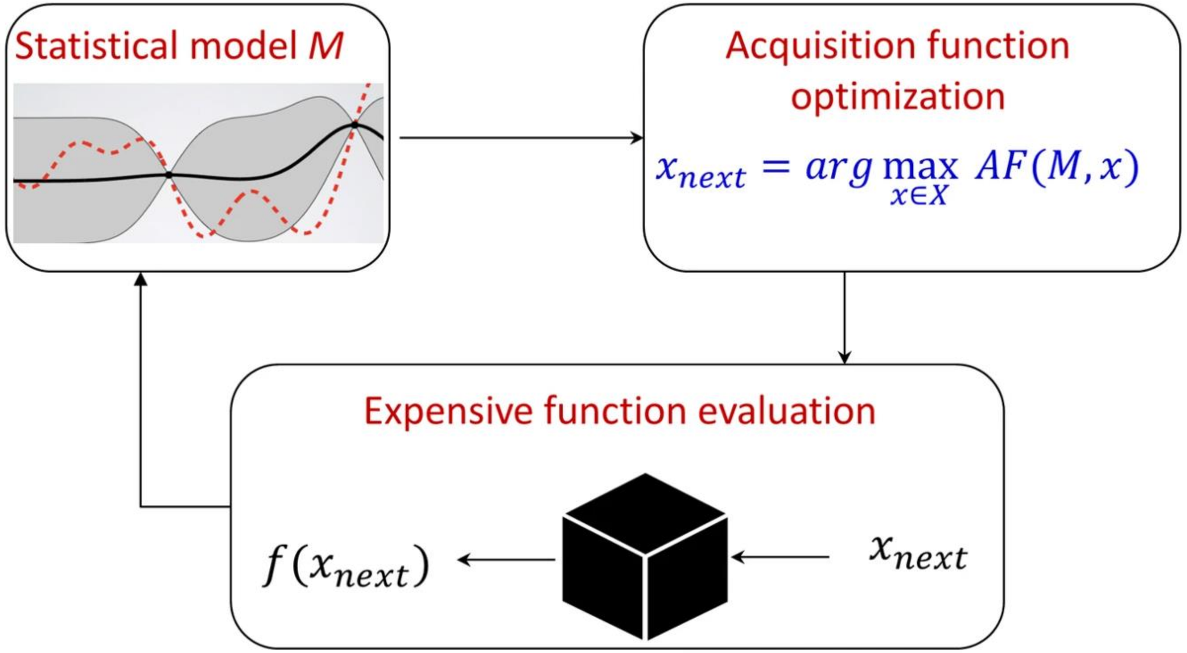
- Goal: finding a buffer size that maximizes the performance function

$$x^* = \arg\max_{x \in X} f(x)$$

- $f(.)$ is a **blackbox** function

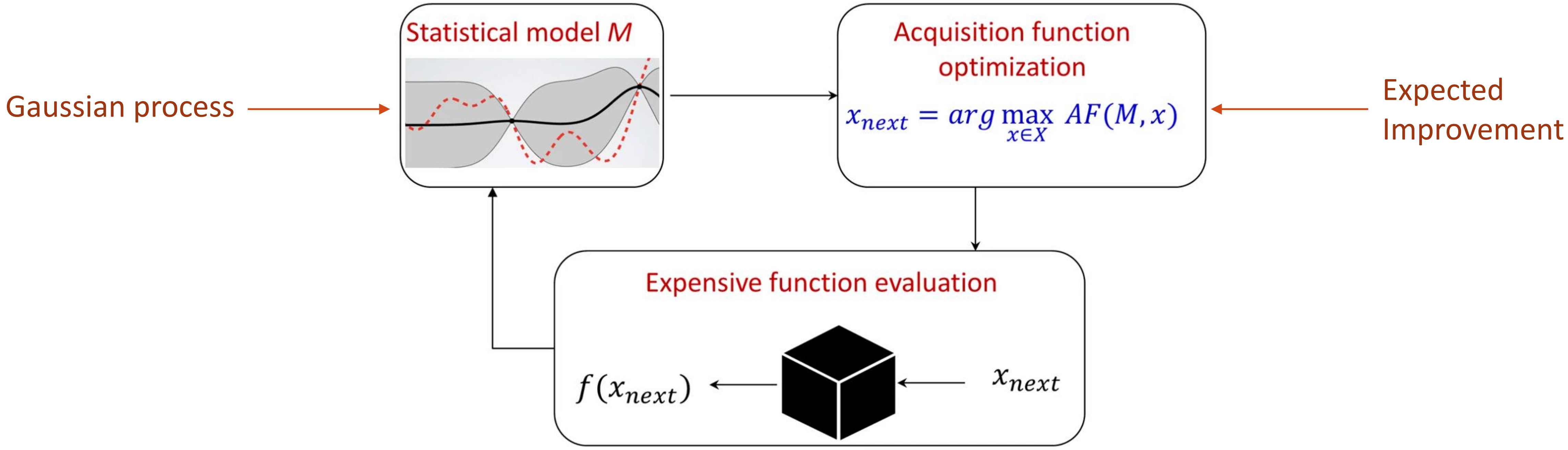
Bayesian Optimization

- Efficiently optimize expensive black-box functions
- Build a surrogate statistical model and use it to search the space
- Replace expensive queries with cheaper queries
- Use uncertainty of the model to select expensive queries

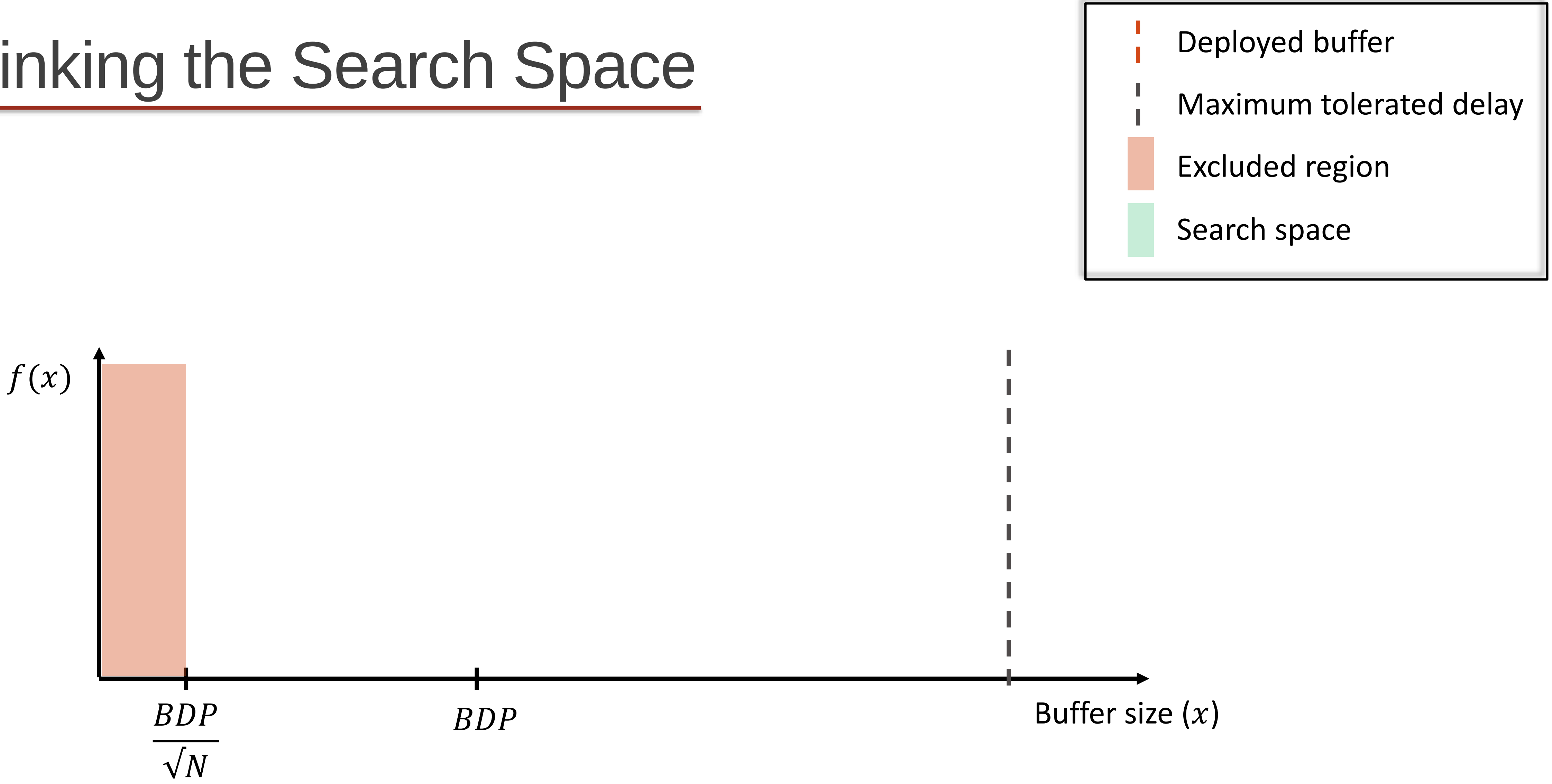


Bayesian Optimization

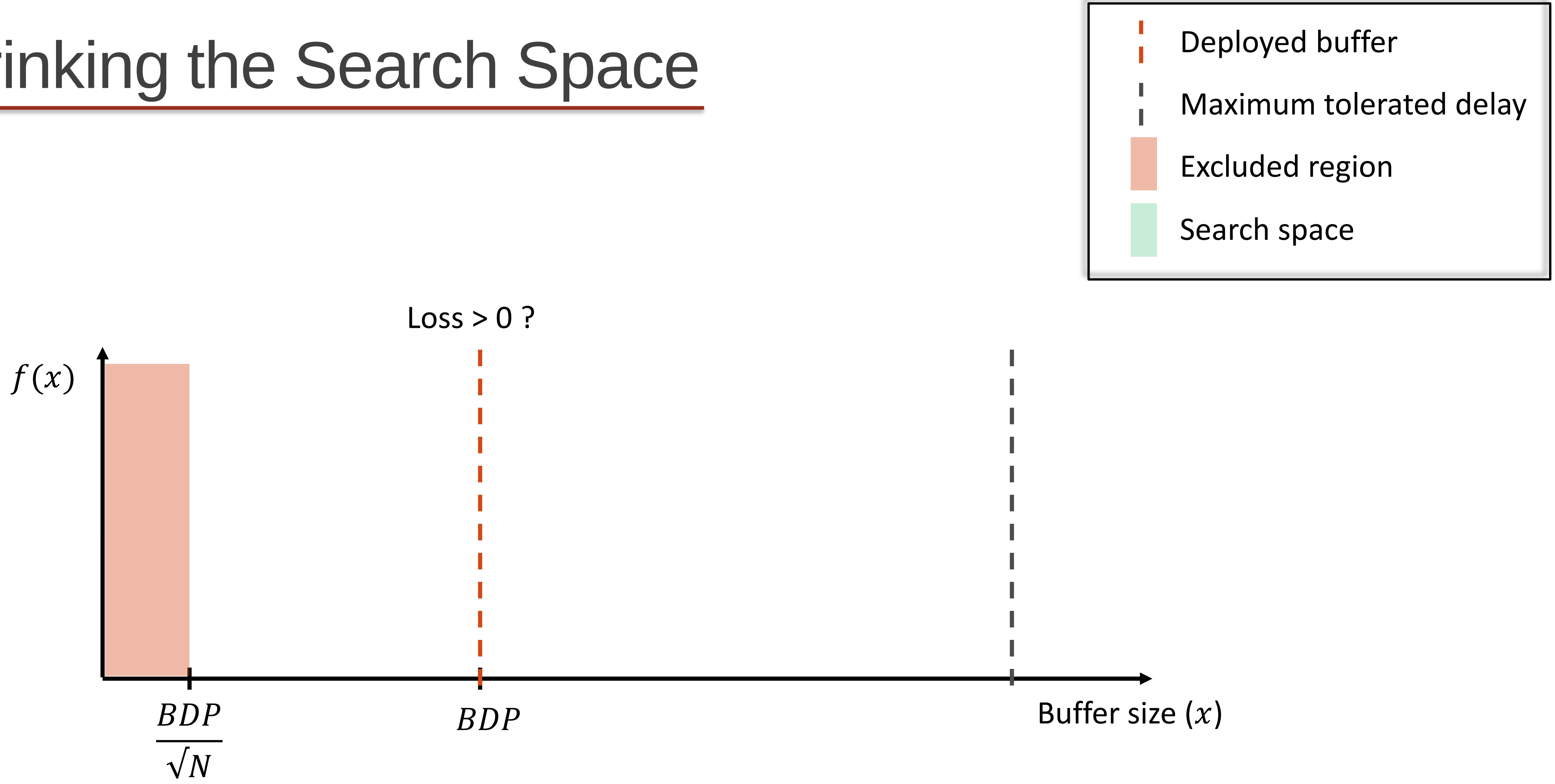
- Efficiently optimize expensive black-box functions
- Build a surrogate statistical model and use it to search the space
- Replace expensive queries with cheaper queries
- Use uncertainty of the model to select expensive queries



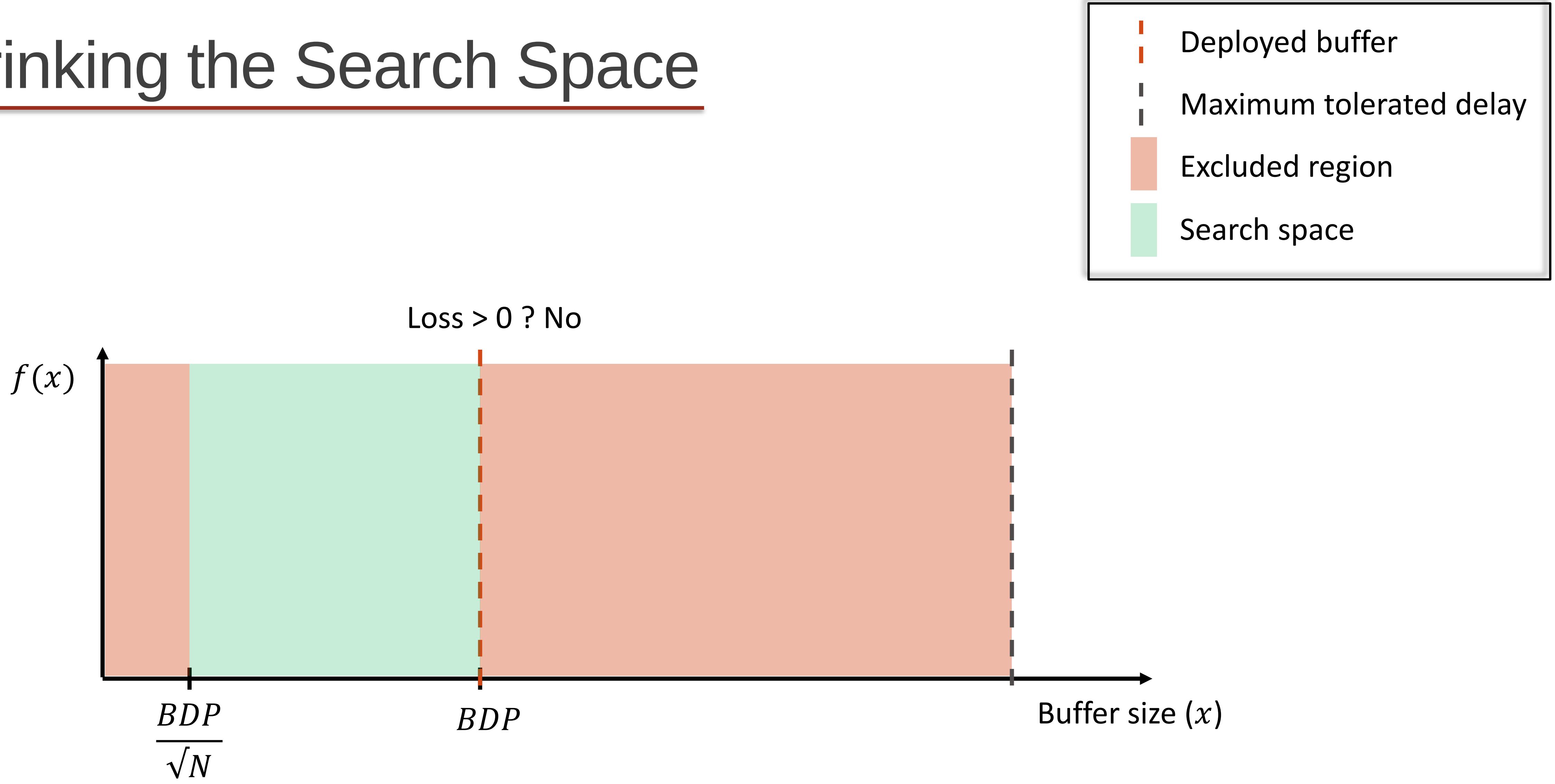
Shrinking the Search Space



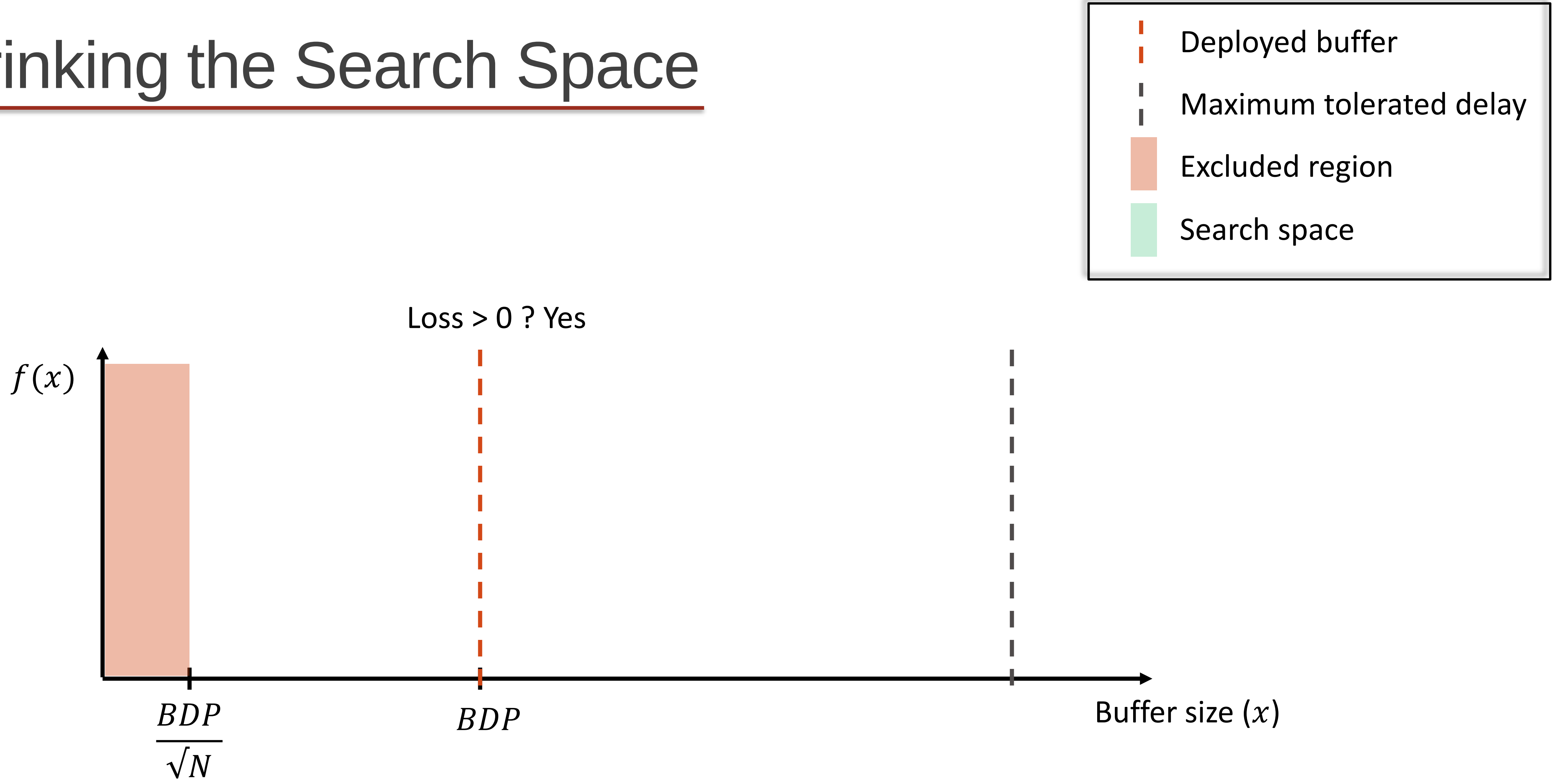
Shrinking the Search Space



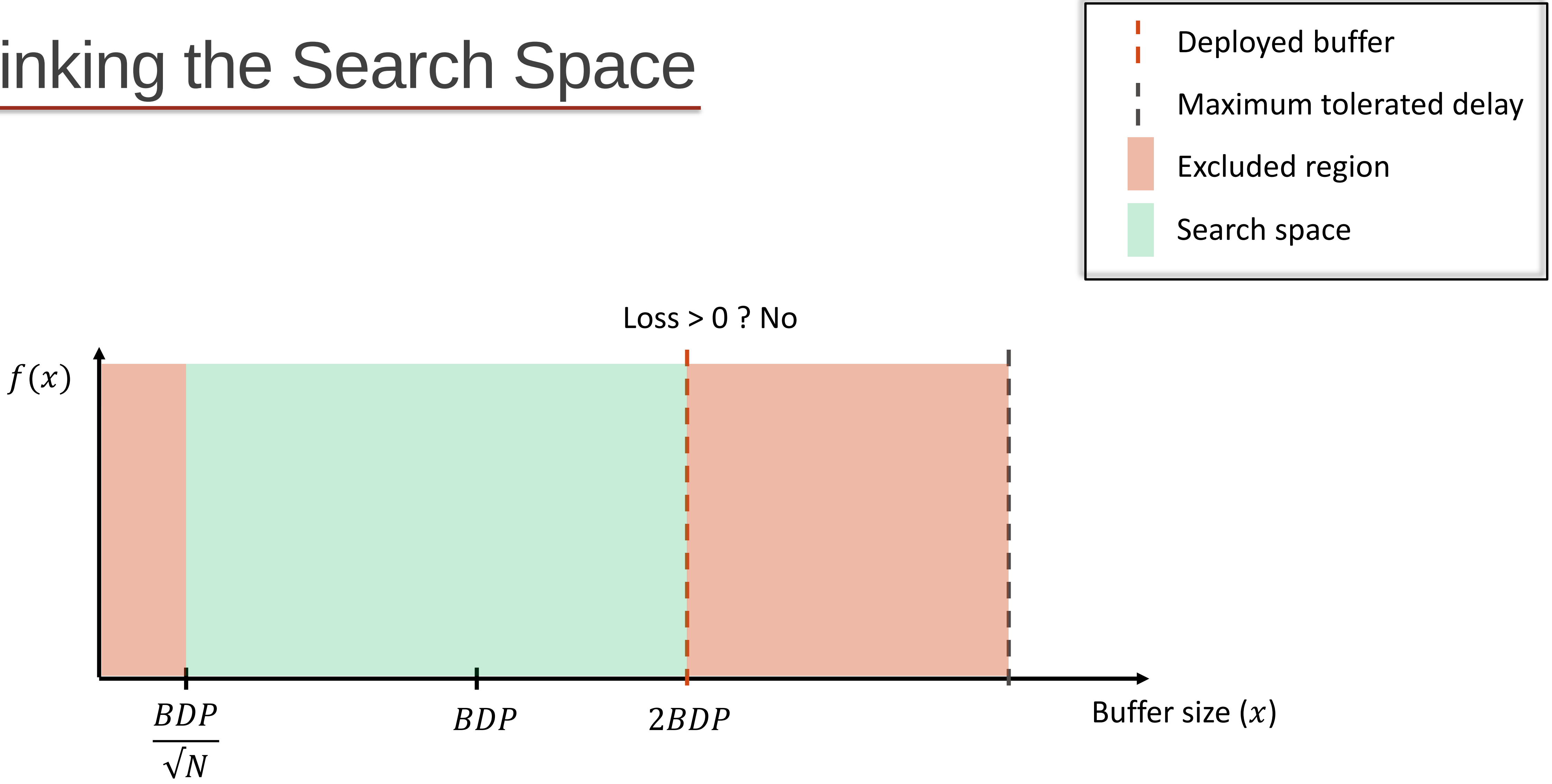
Shrinking the Search Space



Shrinking the Search Space

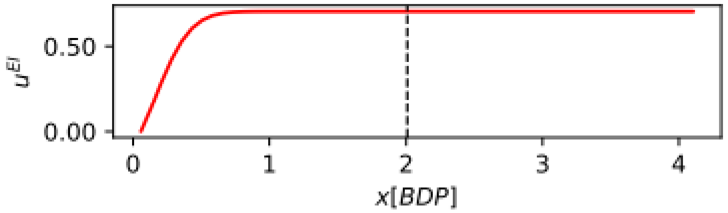


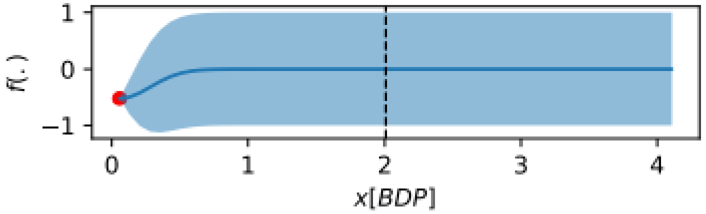
Shrinking the Search Space



Searching Example

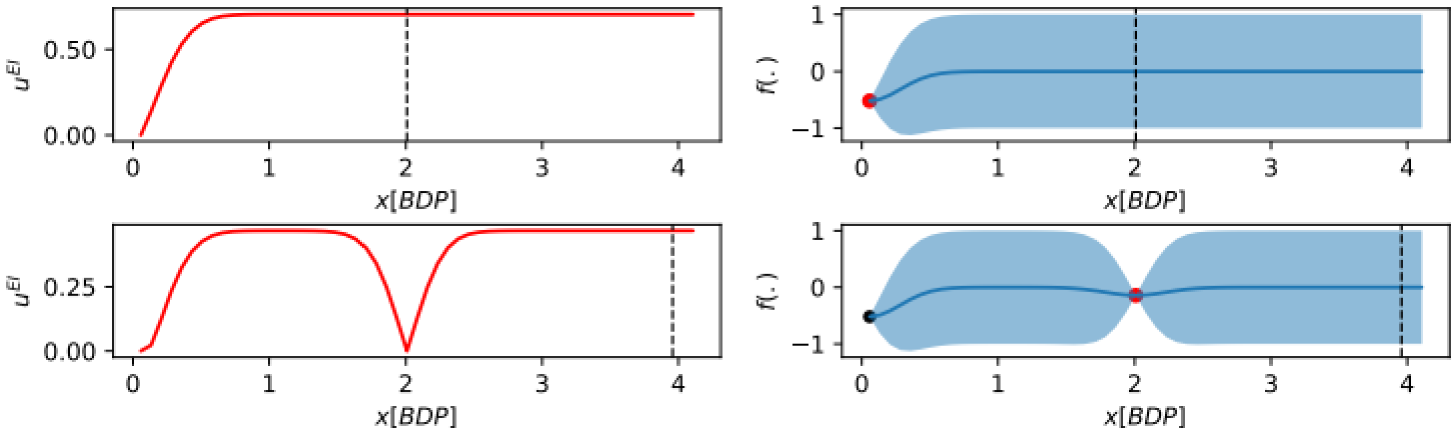
- 500 long flows
- Cubic CCA
- $C = 2.5 Gbps$





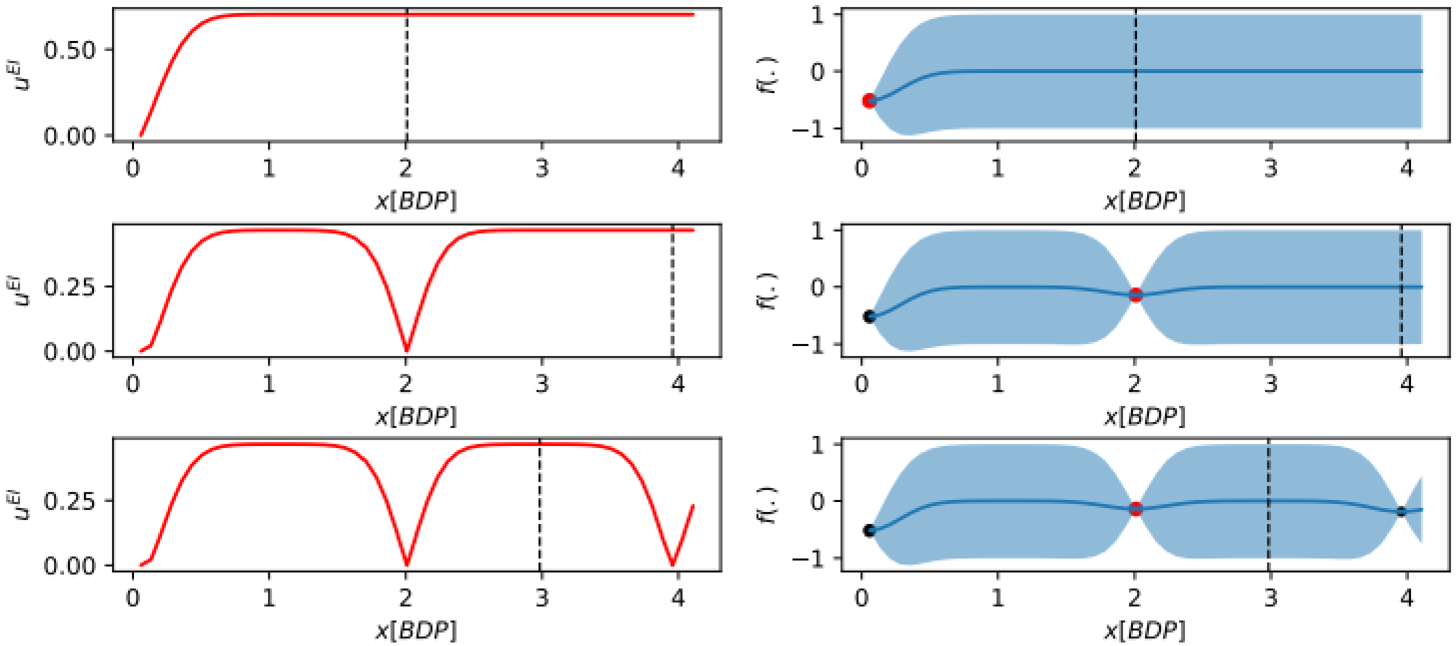
Searching Example

- 500 long flows
- Cubic CCA
- $C = 2.5Gbps$



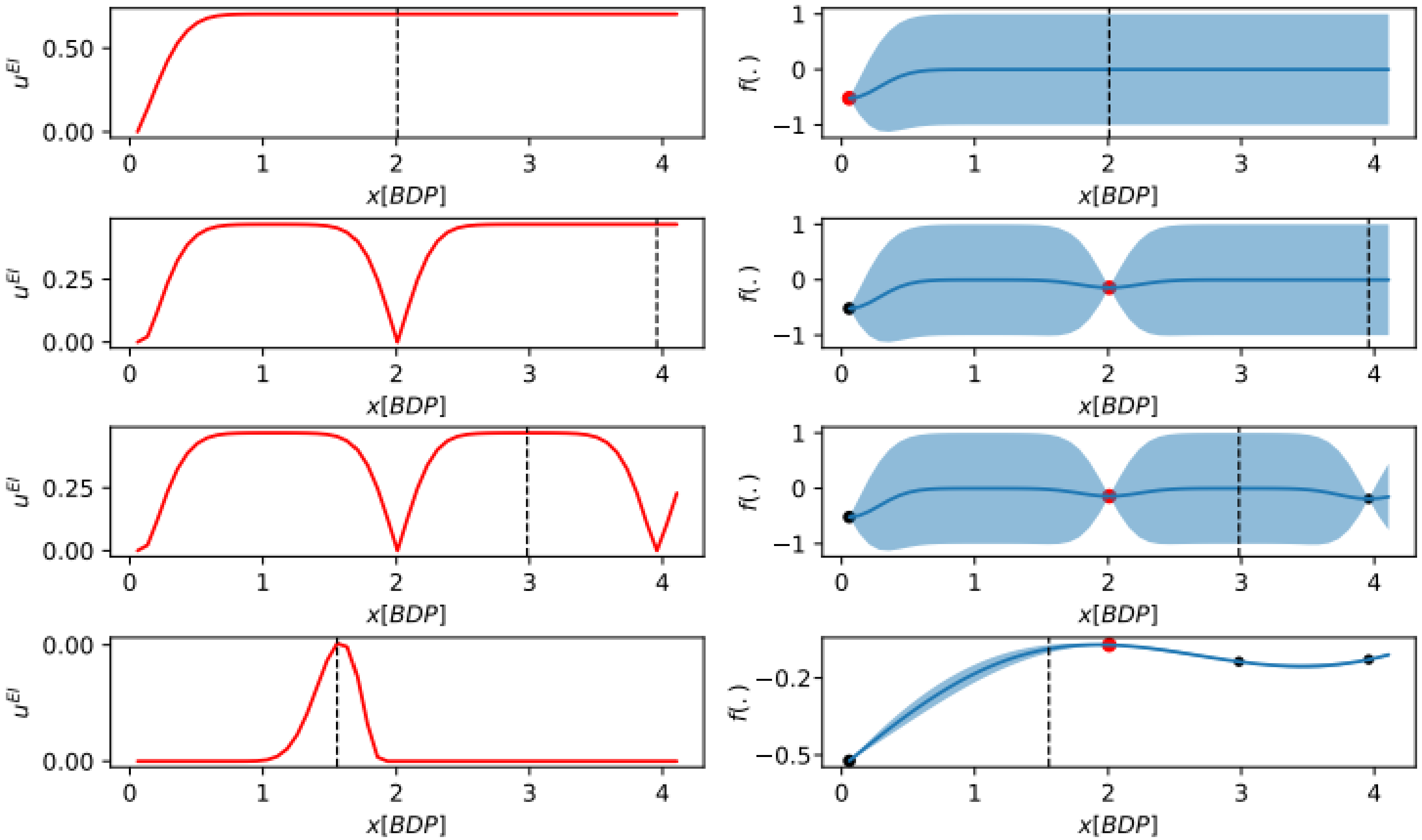
Searching Example

- 500 long flows
- Cubic CCA
- $C = 2.5Gbps$



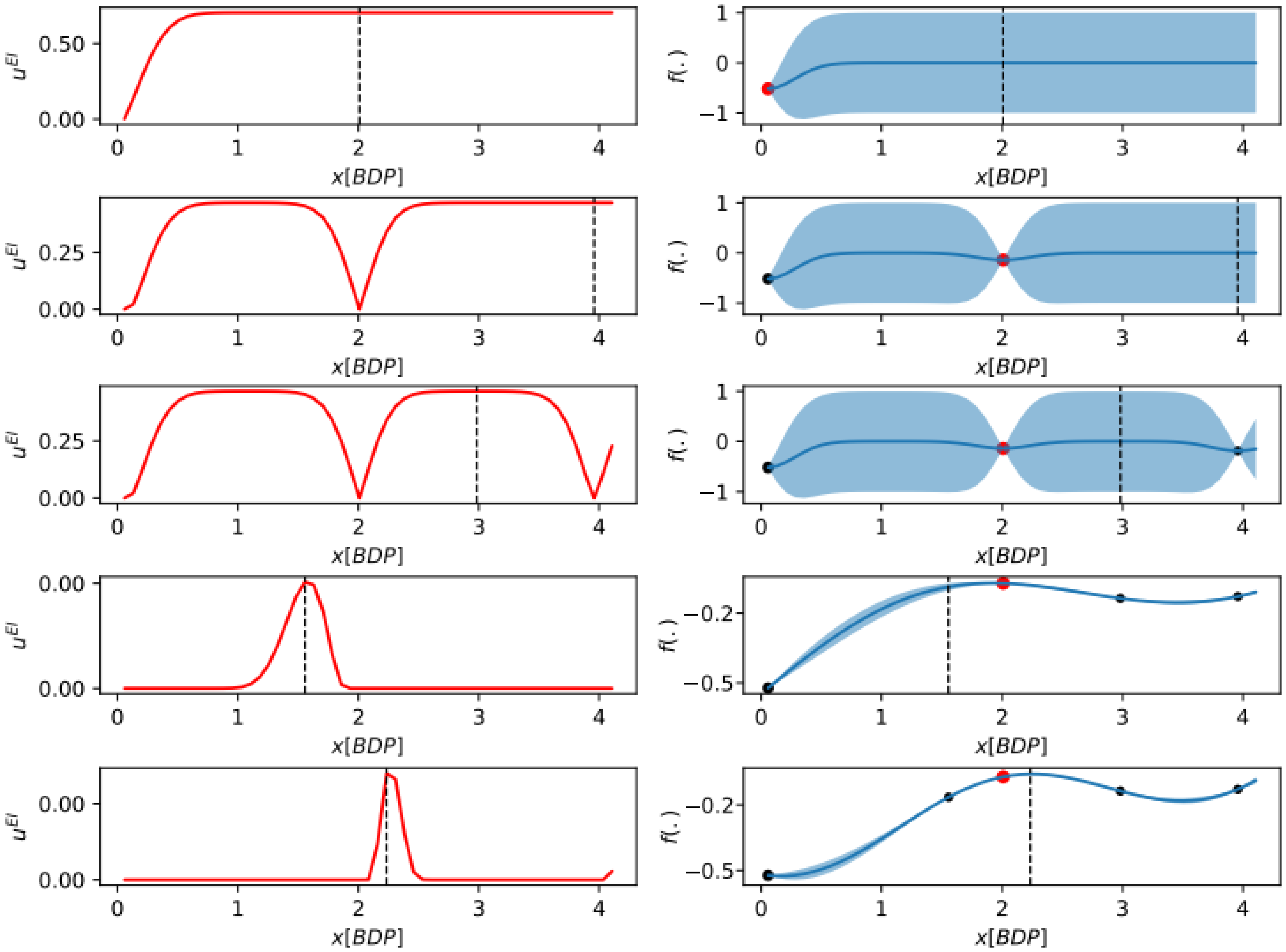
Searching Example

- 500 long flows
- Cubic CCA
- $C = 2.5Gbps$



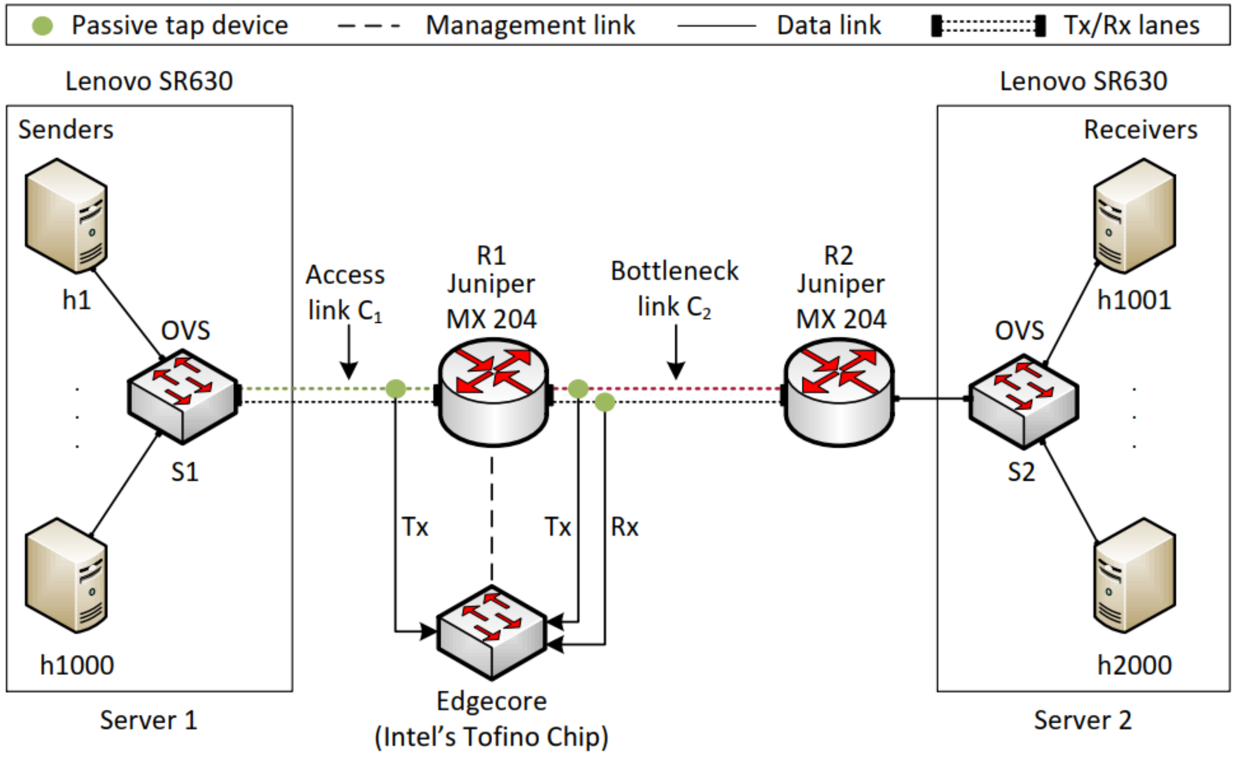
Searching Example

- 500 long flows
- Cubic CCA
- $C = 2.5Gbps$



Implementation and Evaluation

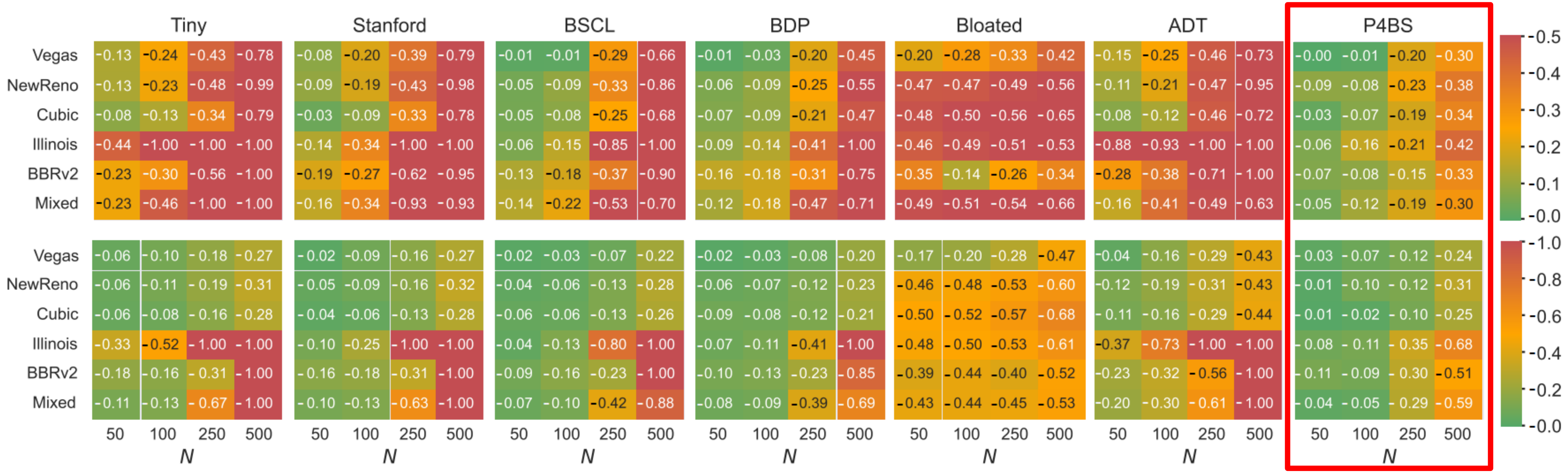
- Topology and experimental setup
- Different congestion control algorithms[1]
- iPerf3
- Access network:
  - C_1 = 40Gbps
  - C_2 = 1Gbps
- Core network:
  - C_1 = 10Gbps
  - C_2 = 2.4Gbps
- Wedge100BF-32X, ASIC chip (Intel's Tofino)



[1]Mishra et al. "The great Internet TCP congestion control census," ACM on Measurement and Analysis of Computing Systems, 2019

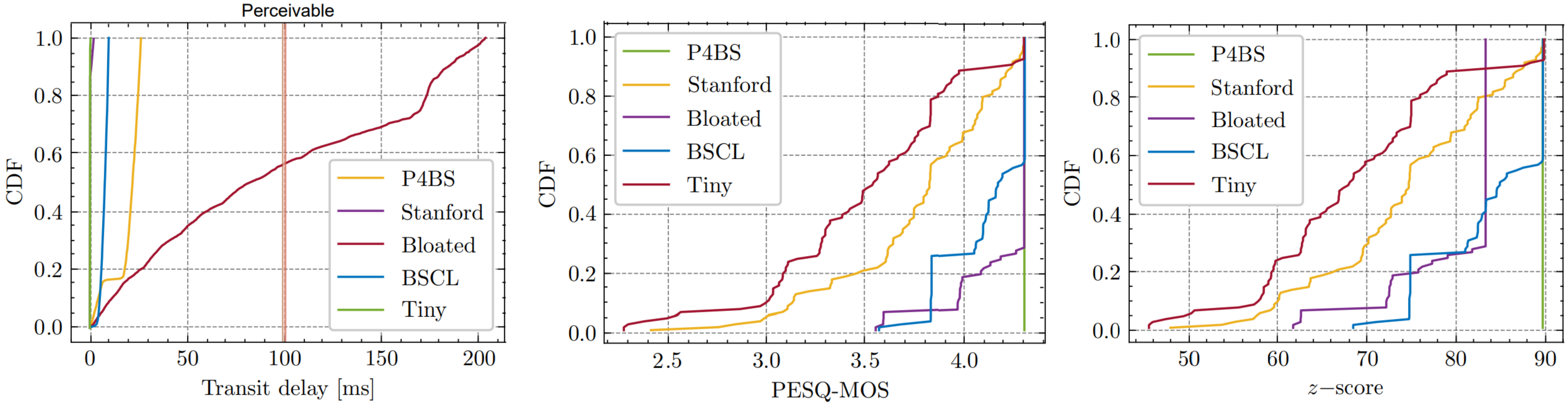
Results

- Average $f(.)$ over the test duration (higher/green is better)
- Top heatmaps: access network
- Bottom heatmaps: core network
- The Mixed scenario combines multiple congestion control algorithms



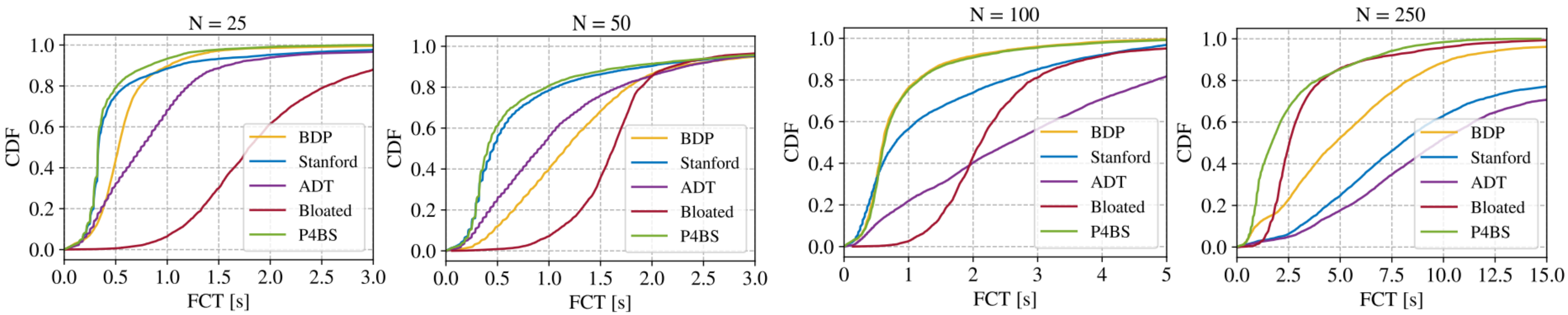
Results (VoIP)

- 100 VoIP calls playing 20 reference speech samples (G.711.a)
- PESQ compares an error-free audio signal to a degraded one (higher is better)
- The z-score considers both the delay and the PESQ (higher is better)



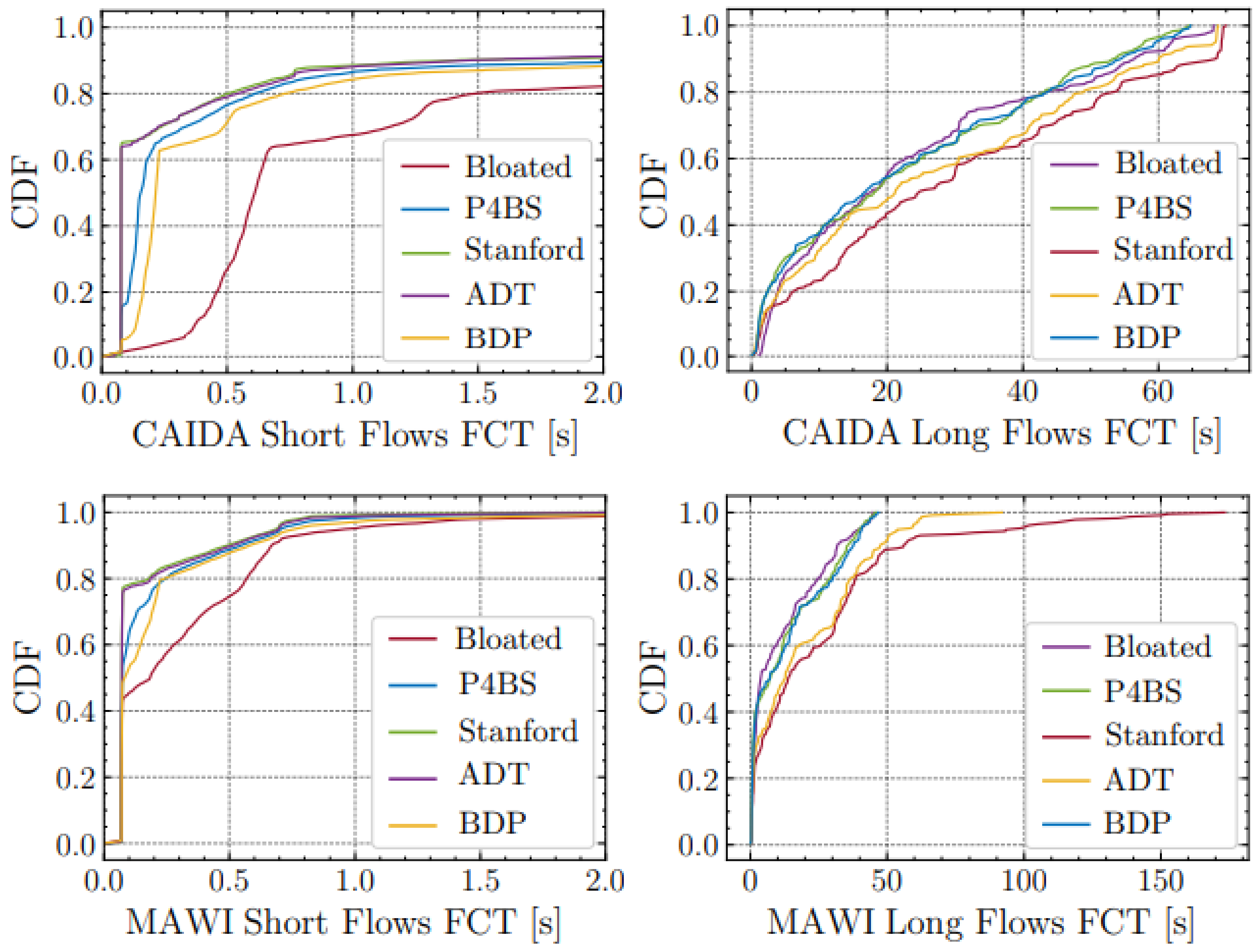
Results (Web Browsing)

- Web browsing traffic
- Background traffic is generated
  - ➢ The sizes of the web pages are in the range [15KB, 2.5MB]



Results (Real Traces)

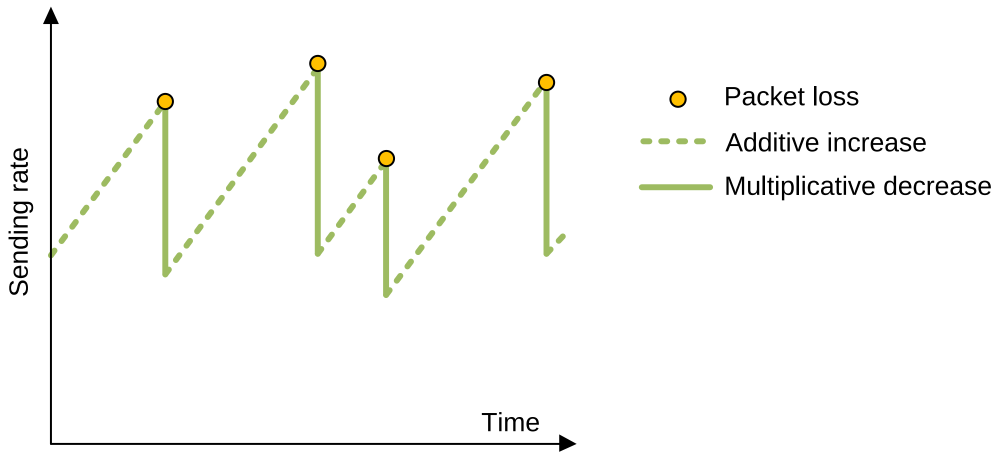
- Real traces
- Center for Applied Internet Data Analysis (CAIDA) traces from Equinix NYC
- MAWI traces from Widely Integrated Distributed Environment (WIDE)
- P4BS found a balance such that:
  - The FCT of long flows is close to that of the bloated buffer
  - The FCT of short flows is close to that of the Stanford buffer



Performance Problem #2:
Heterogeneous Traffic in Switches/Routers

TCP Traditional Congestion Control

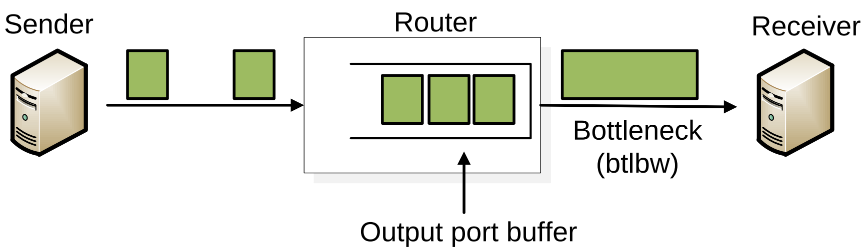
- The principles of window-based CC were described in the 1980s[1]
- Traditional CC algorithms follow the additive-increase multiplicative-decrease (AIMD) form of congestion control

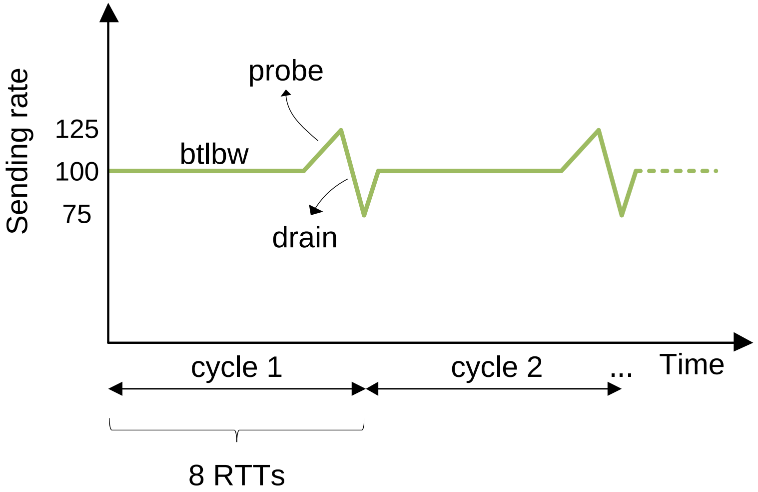


1. V. Jacobson, M. Karels, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review 18 (4) (1988).

BBR: Model-based CC

- TCP Bottleneck Bandwidth and RTT (BBR) is a rate-based congestion-control algorithm[1]
- BBR represented a disruption to the traditional CC algorithms:
  - is not governed by AIMD control law
  - does not the use packet loss as a signal of congestion
- At any time, a TCP connection has one slowest link bottleneck bandwidth (btlbw)

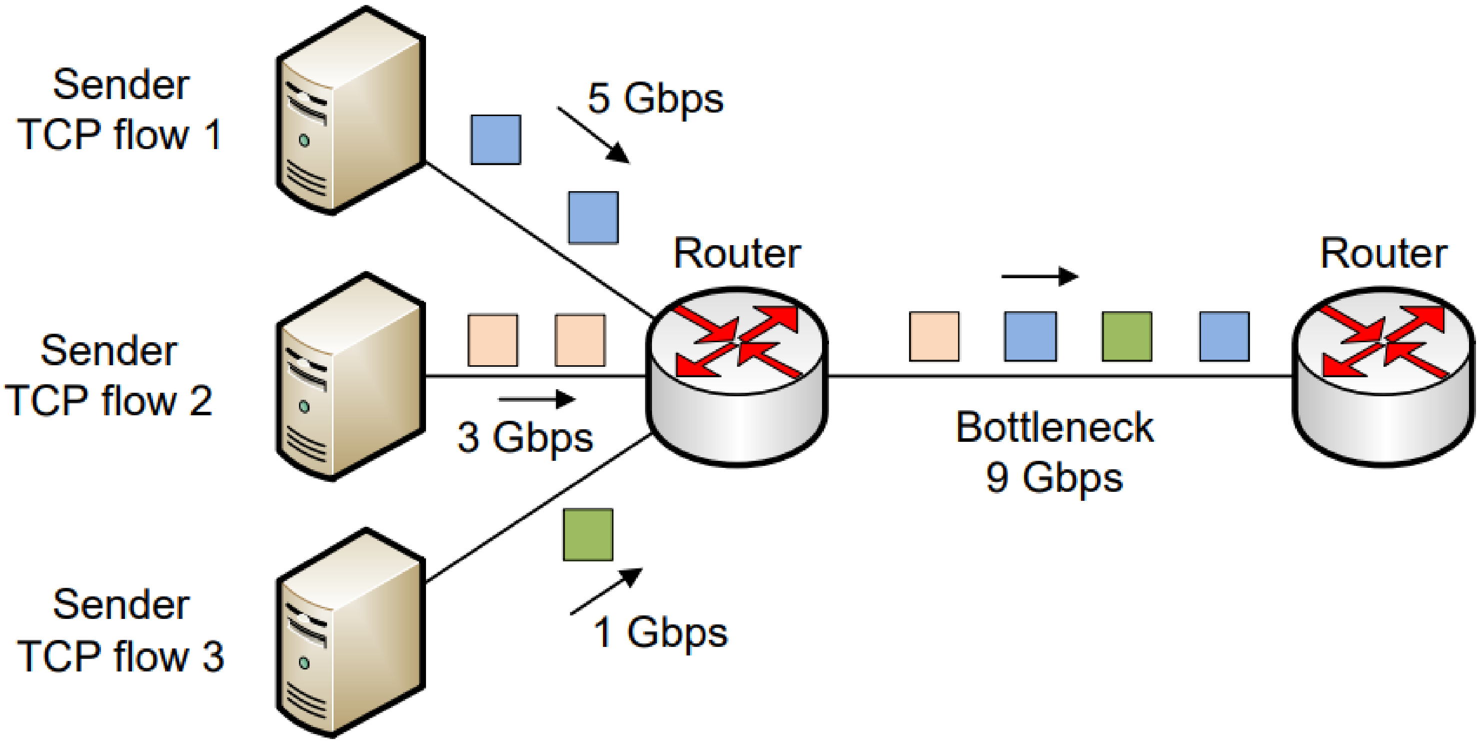




1. N. Cardwell et al. "BBR v2, A Model-based Congestion Control." IETF 104, March 2019.

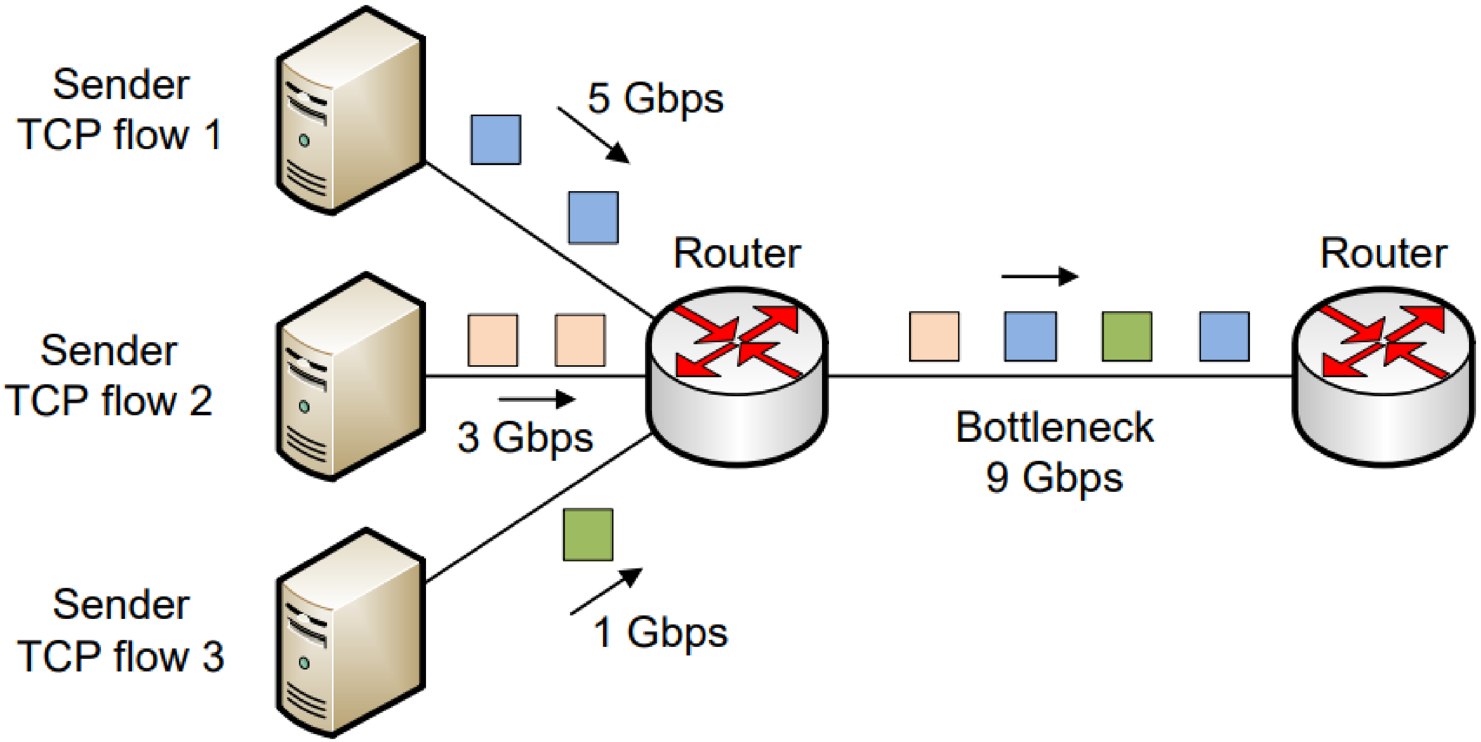
Fairness

- Fairness: how fair is the capacity of the link being divided among the competing flows
- Jain's fairness index: $I = \frac{(\sum_{i=1}^{n} T_i)^2}{n \sum_{i=1}^{n} {T_i}^2}$



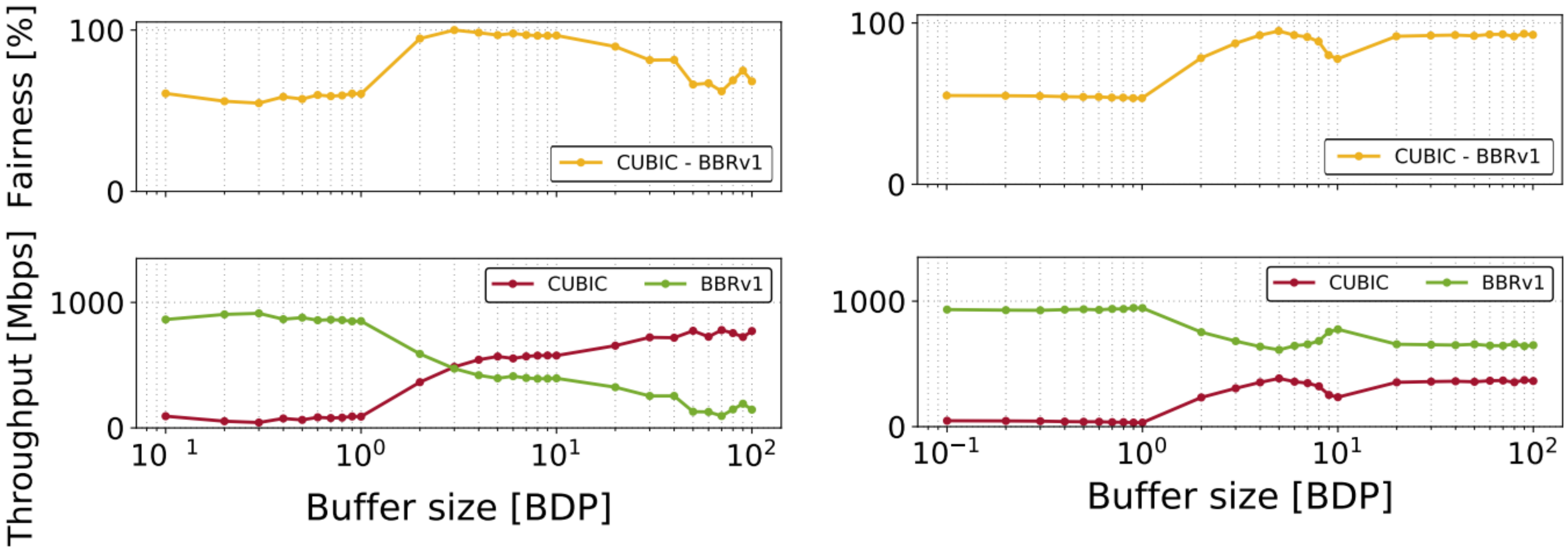
Fairness

- Fairness: how fair is the capacity of the link being divided among the competing flows
- Jain's fairness index: $I = \frac{(\sum_{i=1}^{3} T_i)^2}{3\ \sum_{i=1}^{3} {T_i}^2} = \frac{(5 \cdot 10^9 + 3 \cdot 10^9 + 1 \cdot 10^9)^2}{3\ \cdot ((5 \cdot 10^9)^2 + (3 \cdot 10^9)^2 + (1 \cdot 10^9)^2)} = \boxed{0.77}$



Fairness

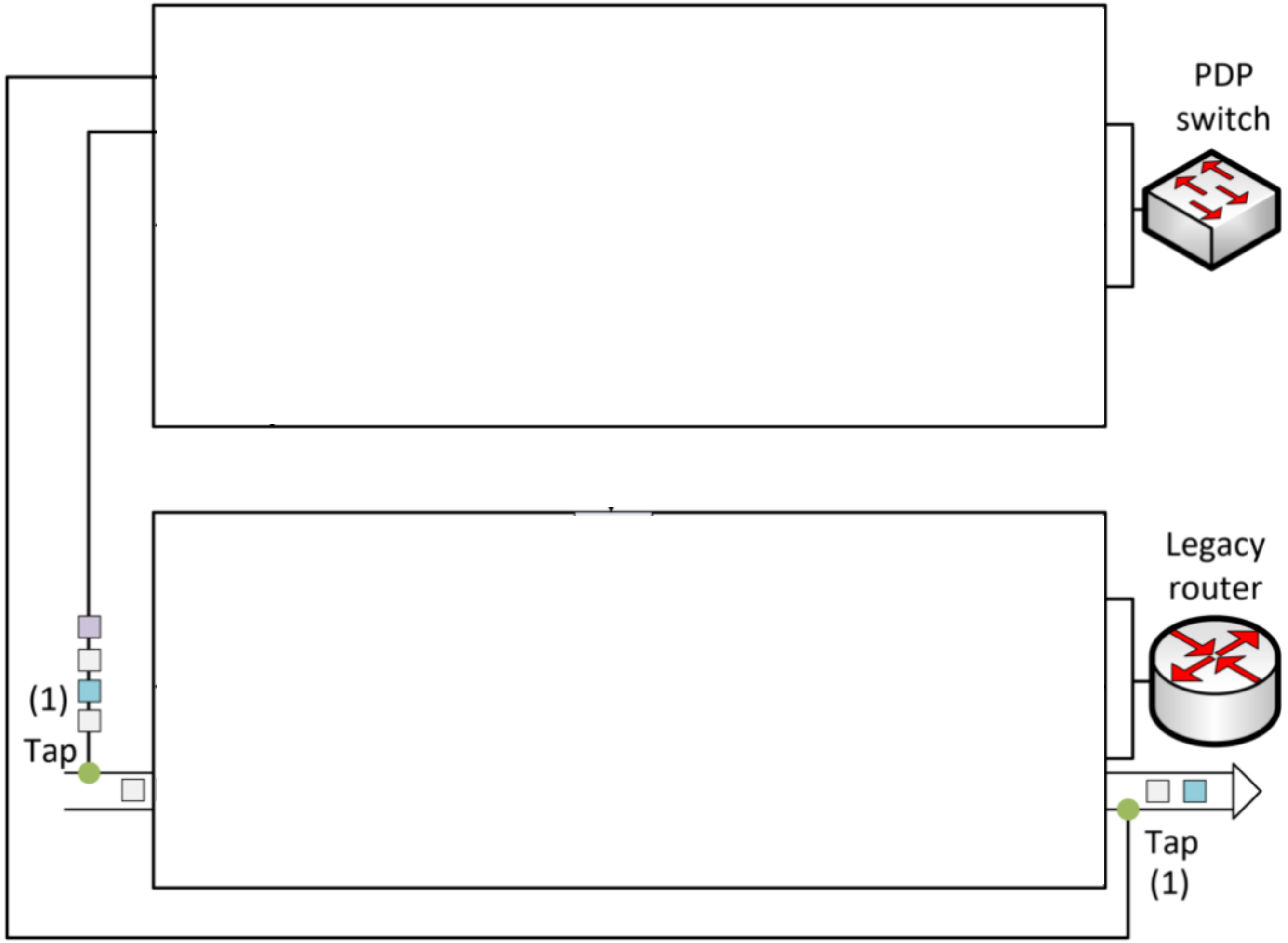
- The fairness between flows belonging to different CCAs is often low
- E.g., the fairness among Cubic and BBR flows[1]



(1 CUBIC, 1 BBR) (50 CUBIC, 50 BBR)

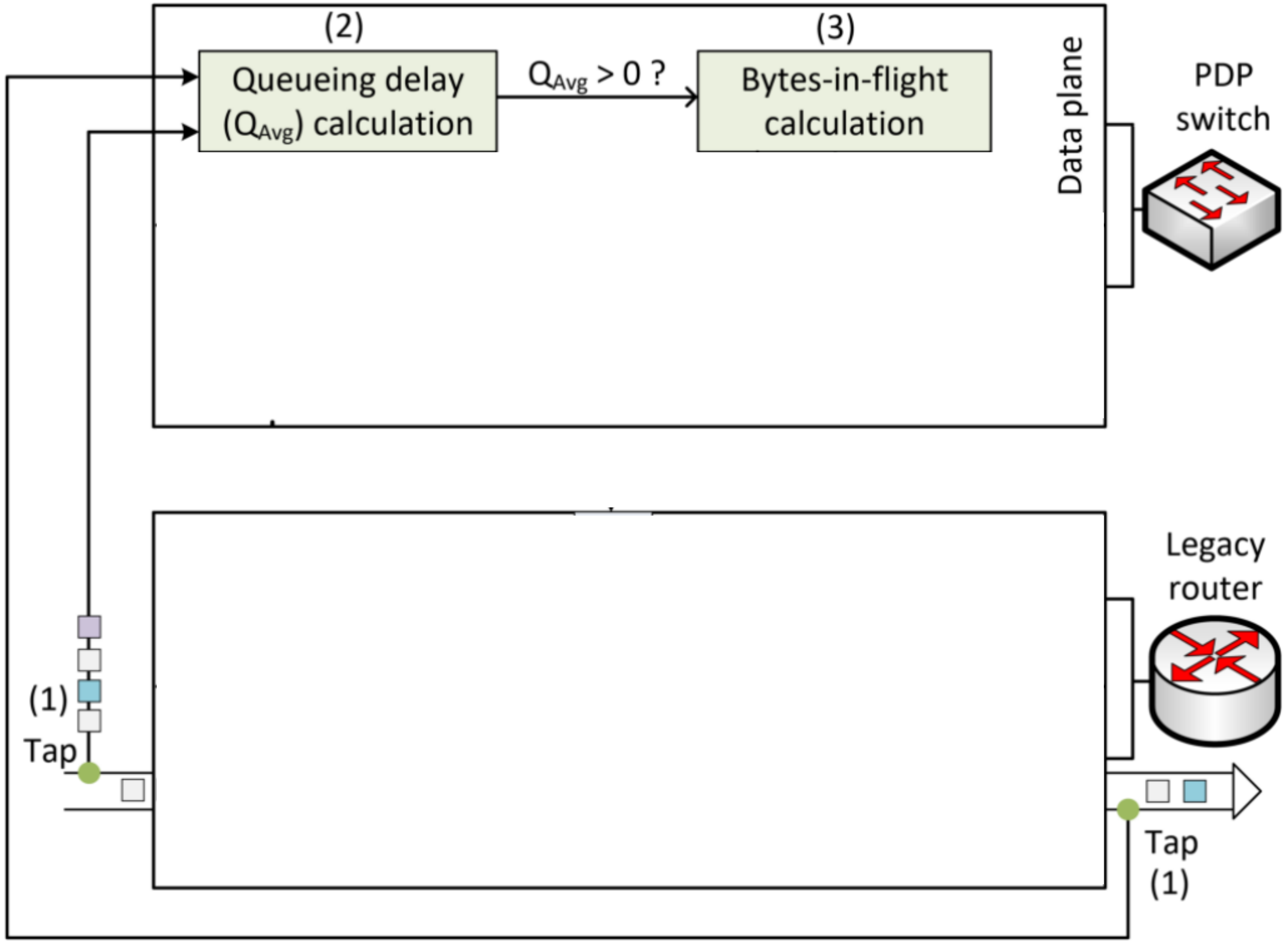
Proposed System

- Passive PDPs for congestion control algorithm (CCA) identification at line rate



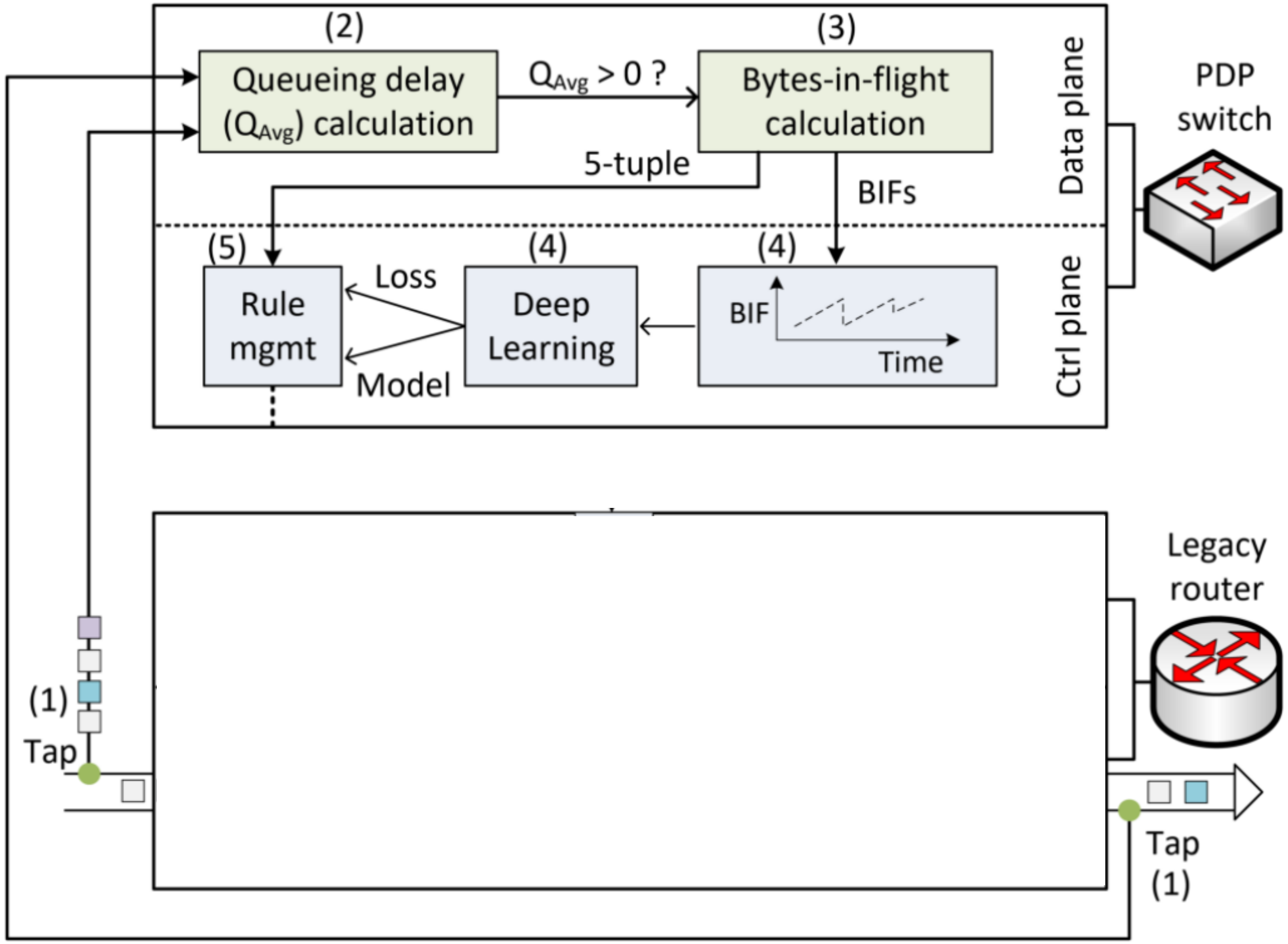
Proposed System

- Passive PDPs for congestion control algorithm (CCA) identification at line rate



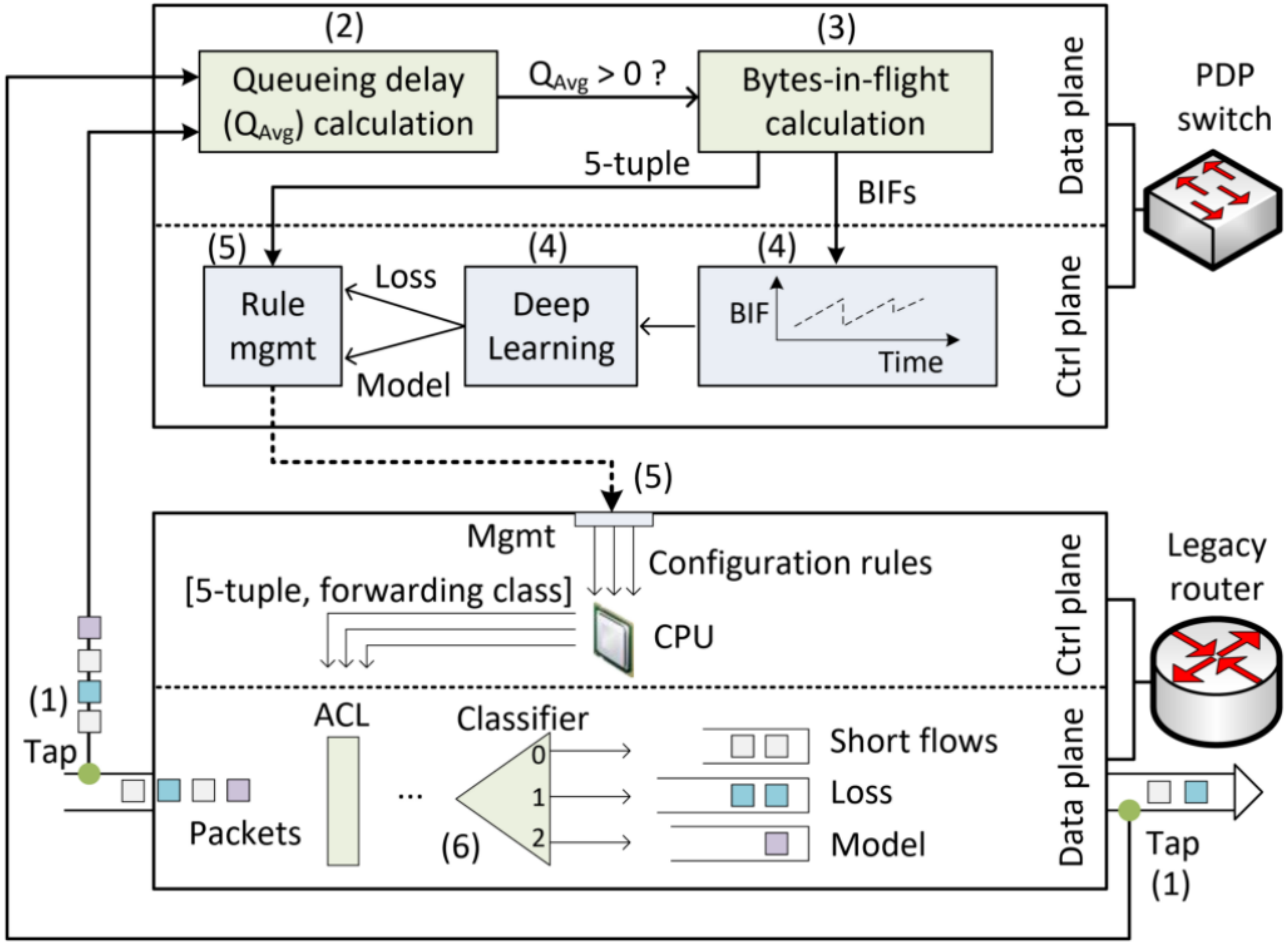
Proposed System

- Passive PDPs for congestion control algorithm (CCA) identification at line rate



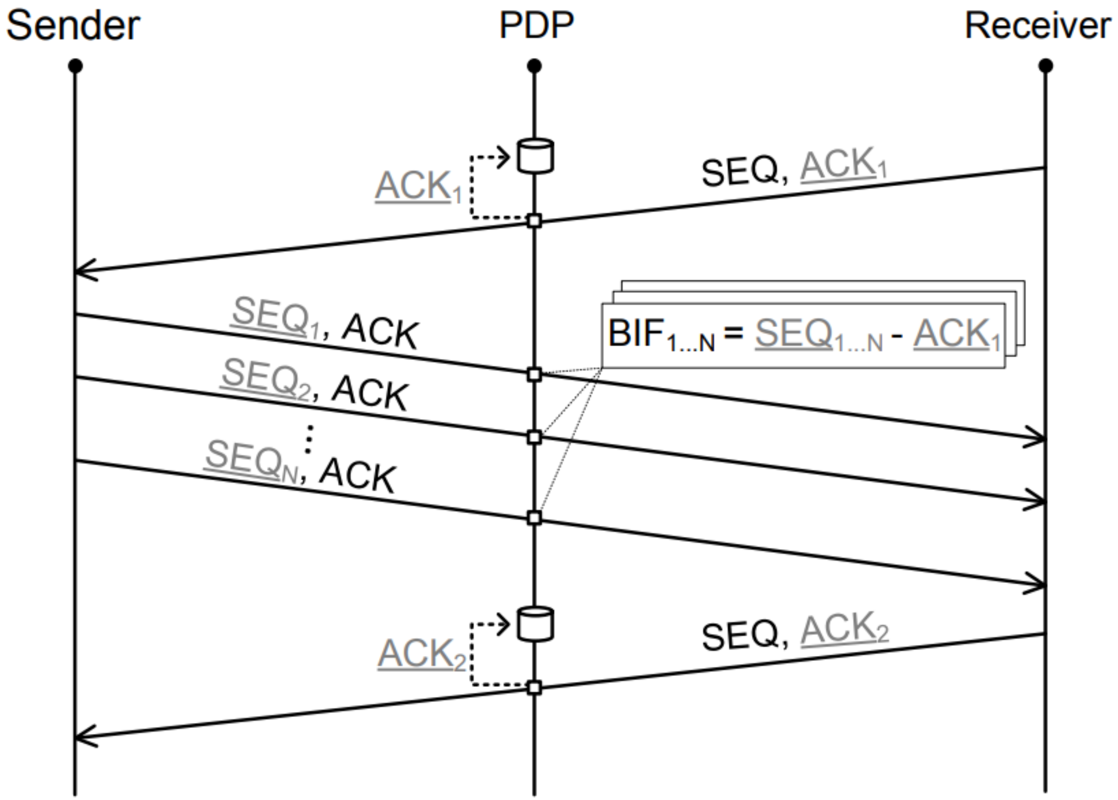
Proposed System

- Passive PDPs for congestion control algorithm (CCA) identification at line rate



Bytes-in-flight Calculation

- Bytes-in-flight (BIF) is the amount of data sent but not yet acknowledged
- BIF is correlated to the TCP congestion window



Model Training

- The model is trained on CAIDA's traffic for two minutes
- The model is also trained with synthetically generated traffic

TRAINING PARAMETERS FOR THE SYNTHETICALLY GENERATED DATASET

Flows	1, 2, 5, 10, 15, 20, 50, 100
Bandwidth [bps]	500M, 1G, 2G, 3G, 4G, 5G, 10G
CCAs	Loss (CUBIC, Reno), Model (BBR)
Packet loss rates [%]	0, 0.1, 0.25, 0.5
Propagation delays [ms]	0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Buffer sizes [ms]	10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Model Testing

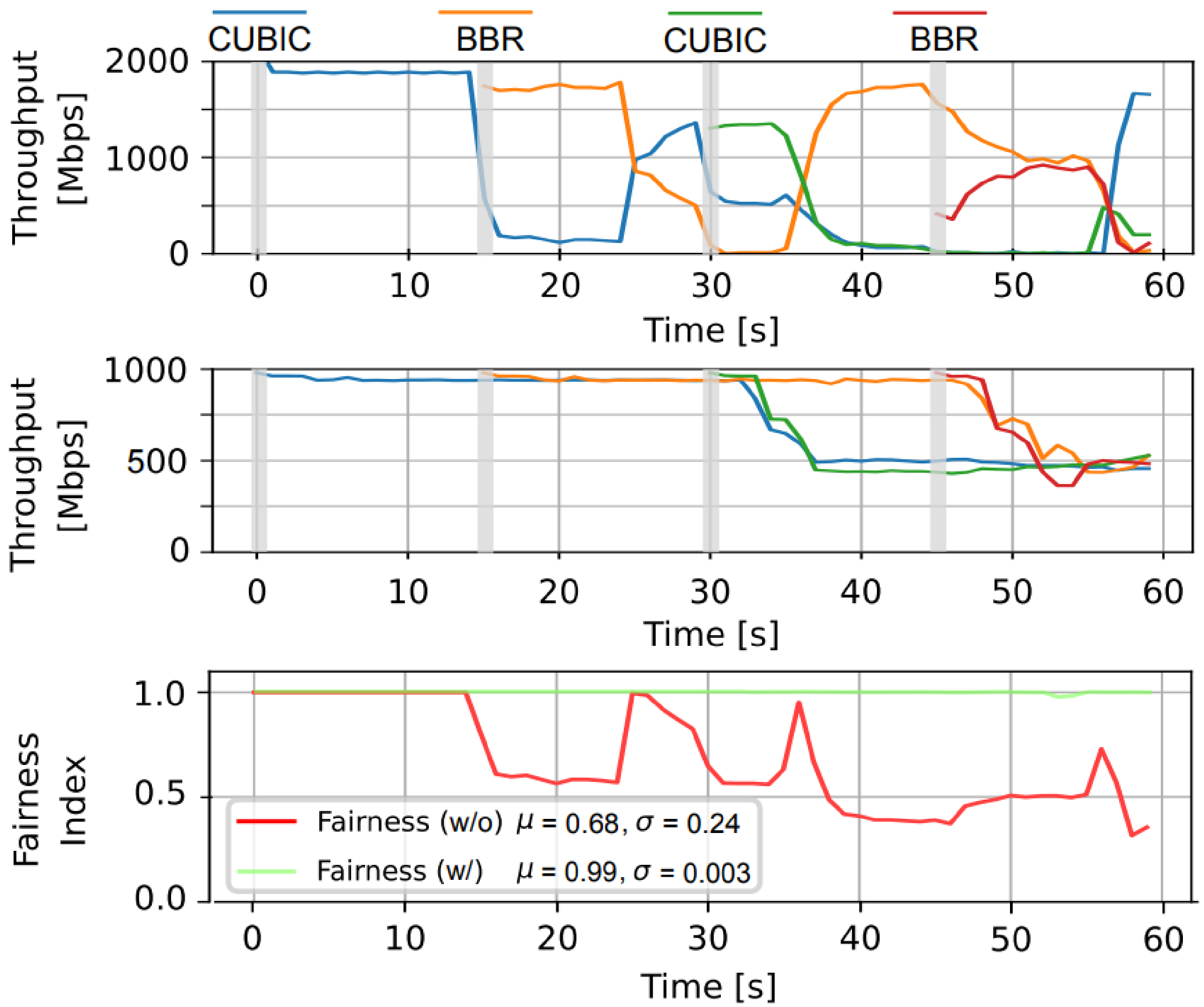
- The model was tested against 10 minutes of traffic from the remaining CAIDA dataset
- The bottleneck bandwidth was configured to 1Gbps, 1.5Gbps, 2Gbps, and 2.5Gbps
- Results outperformed the state-of-the-art CCA identification systems

CLASSIFICATION RESULTS.

Dataset	Classes	Precision	Recall	F1-score	Accuracy
CAIDA 1Gbps	Loss	96.2%	93.5%	94.8%	96.1%
	Model	96.0%	97.7%	96.8%	
CAIDA 1.5Gbps	Loss	95.2%	92.0%	93.1%	95%
	Model	95.6%	97.6%	96.6%	
CAIDA 2Gbps	Loss	92.0%	92.5%	92.3%	95.4%
	Model	96.9%	96.4	96.8%	
CAIDA 2.5Gbps	Loss	91.5%	91.0%	91.2%	95.6%
	Model	97.0%	97.1%	97.0%	
Synthetic	Loss	99.2%	99.5%	99.4%	99.4%
	Model	99.5%	99.2%	99.4%	

Fairness Evaluation

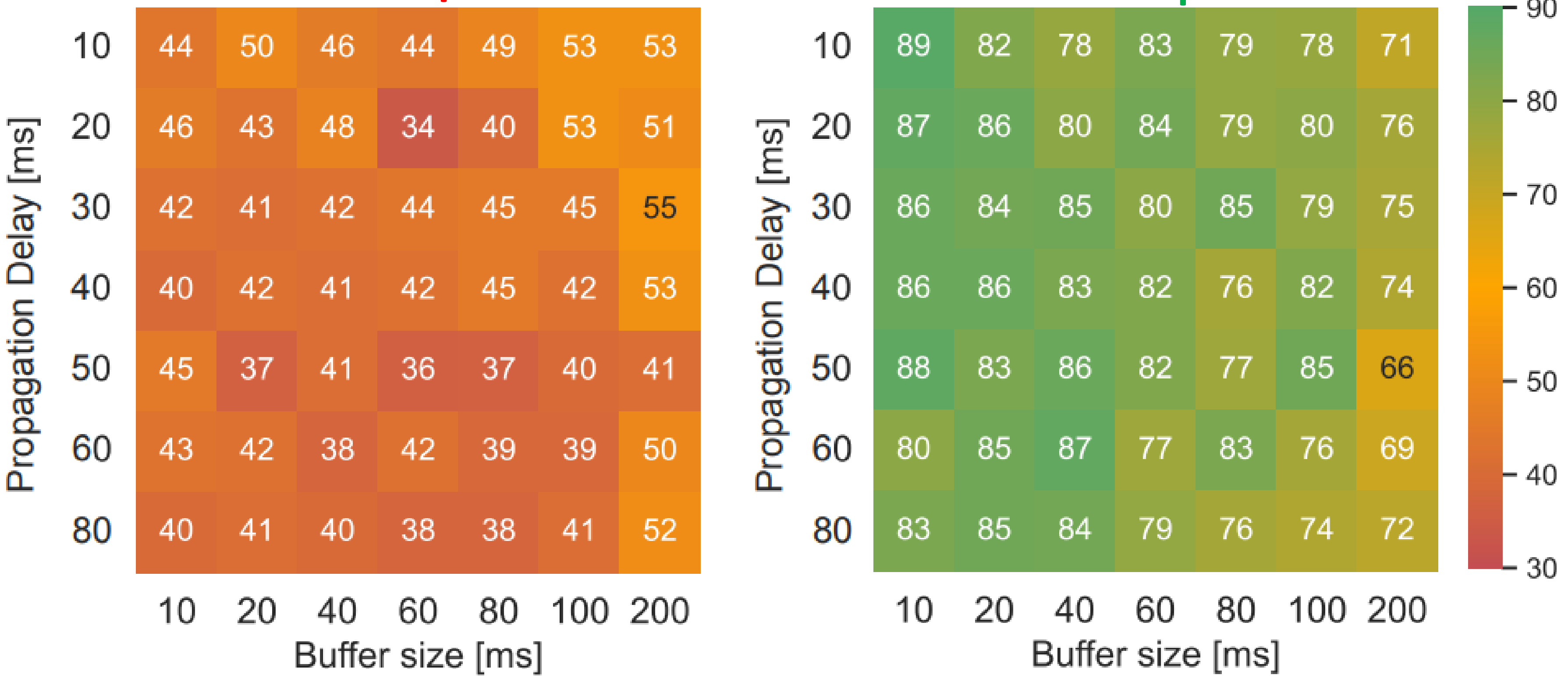
- Alternating flows joining every 15 seconds
- The system promptly identifies the CCA and assigns the flow
- Fairness is ~ 100%



Fairness Evaluation

- 10 long flows started at the same time, with alternating CCAs
  - Flow1 uses CUBIC, Flow2 uses BBR, Flow3 uses CUBIC, etc.
- Various propagation delays and various router buffer sizes are used

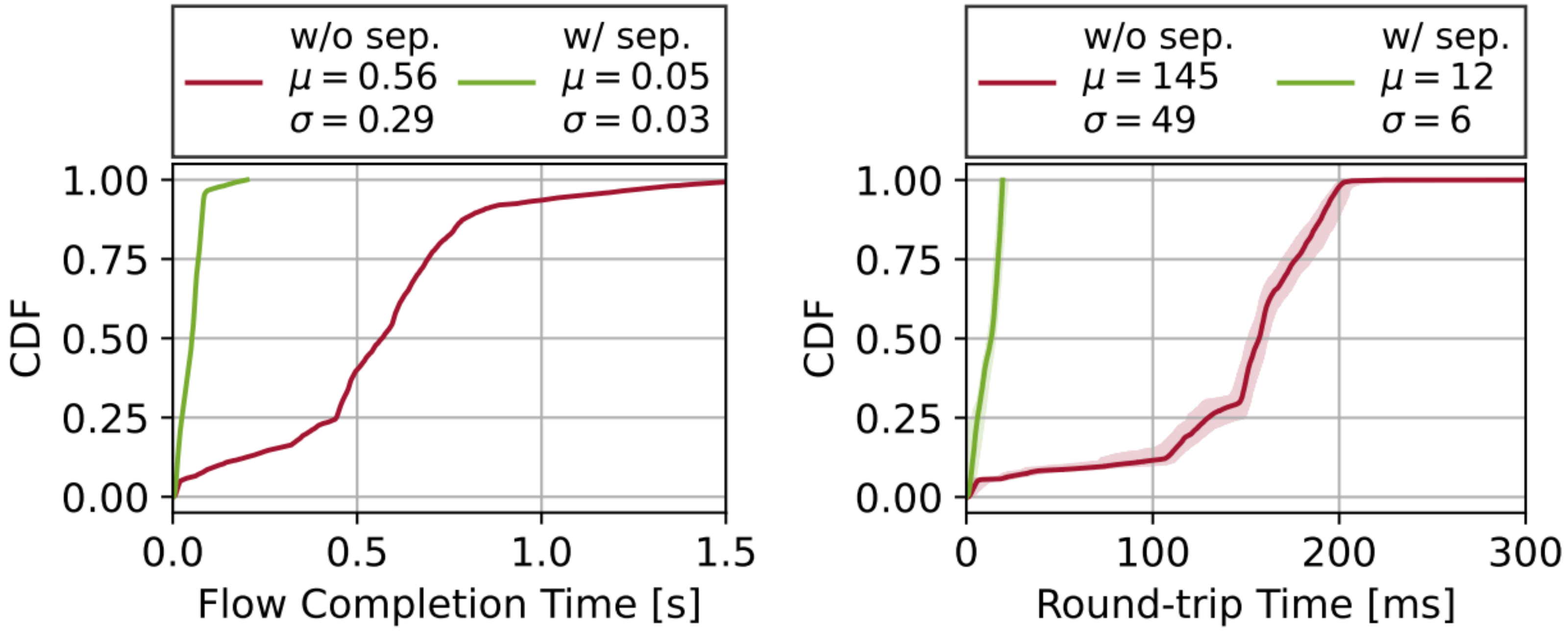
Without separation **With separation**



(a) (b)

Flow Completion Time (Short Flows)

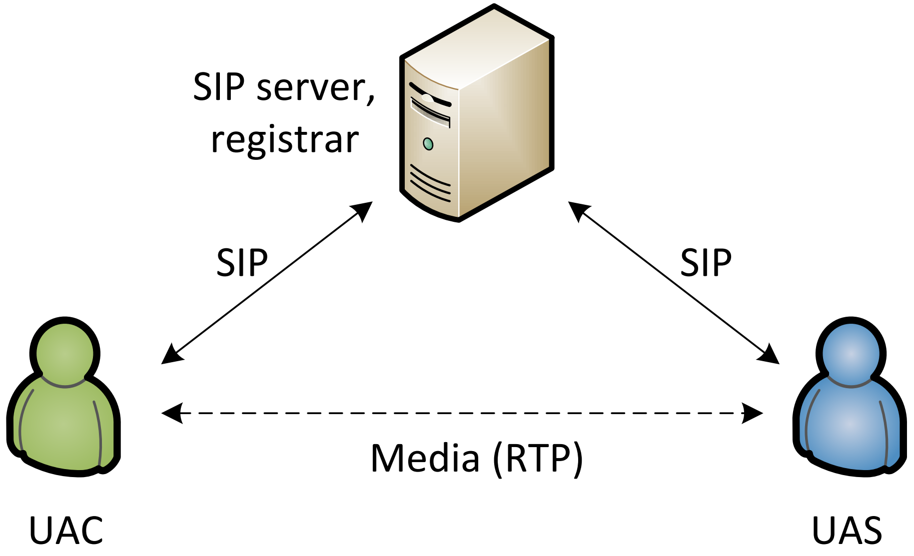
- 100 long flows (50% Cubic, 50% BBR) are generated over a bottleneck link of 3Gbps
- The queue size for the “w/o separation” scenario is 200ms
- 10,000 short flows, whose inter-connection times are generated from an exponential distribution with a mean of one second, are initiated



Performance Problem #3:
CPU-based Middleboxes Processing Packets

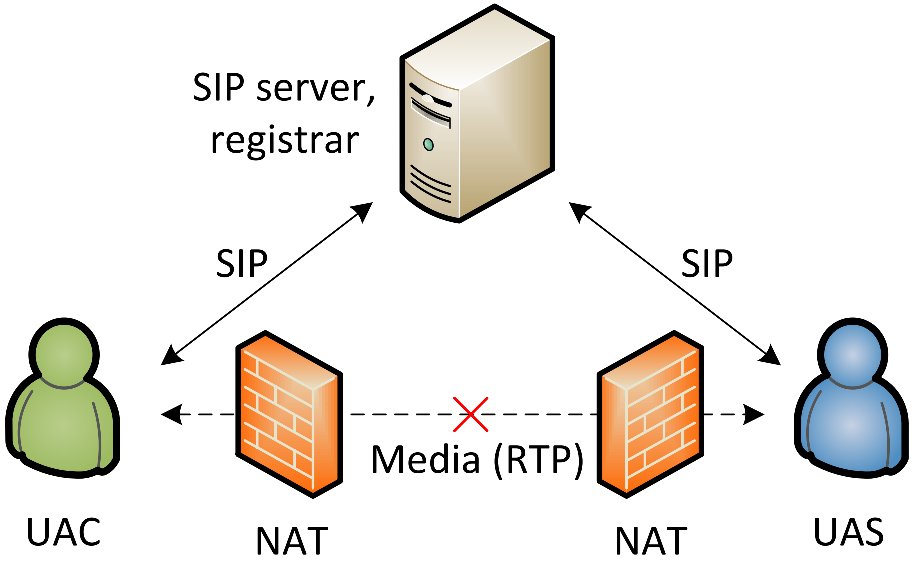
Voice over IP Use Case

- Signaling protocol (e.g., SIP) initiates, maintains, and terminates multimedia sessions between endpoints
  - User agent client (UAC)
  - User agent server (UAS)
- Media protocol (e.g., RTP) transports real-time data, such as audio and video



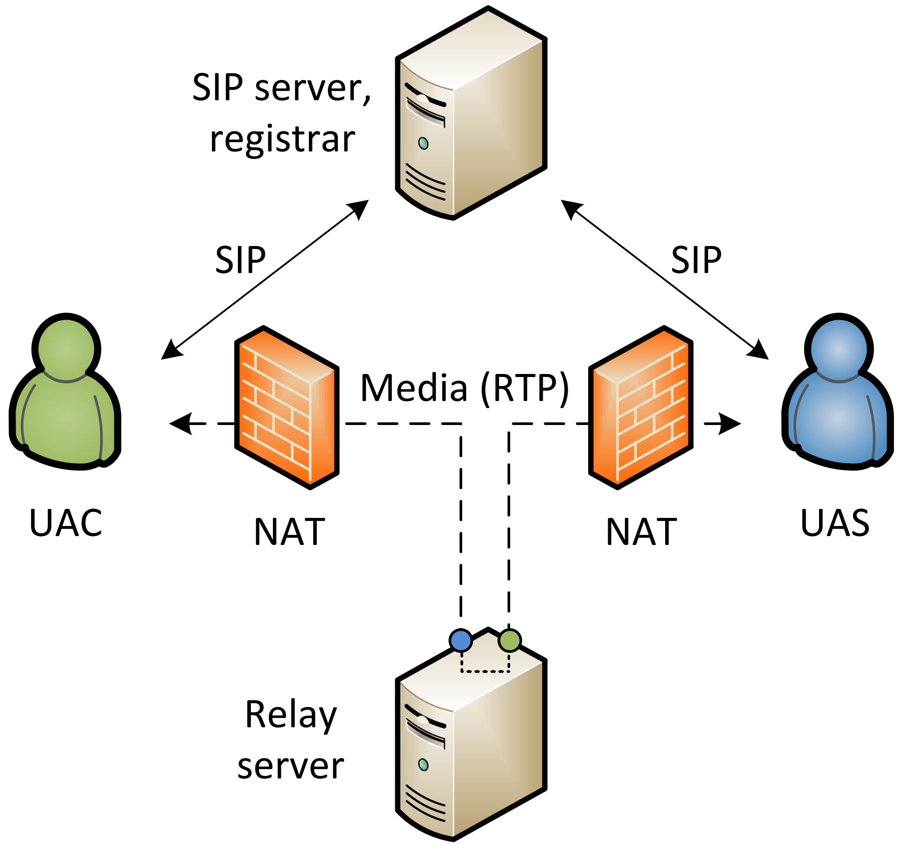
Voice over IP Use Case

- Communicating parties are often behind a Network Address Translation (NAT)
- Since they do not have public IP addresses, they cannot communicate directly



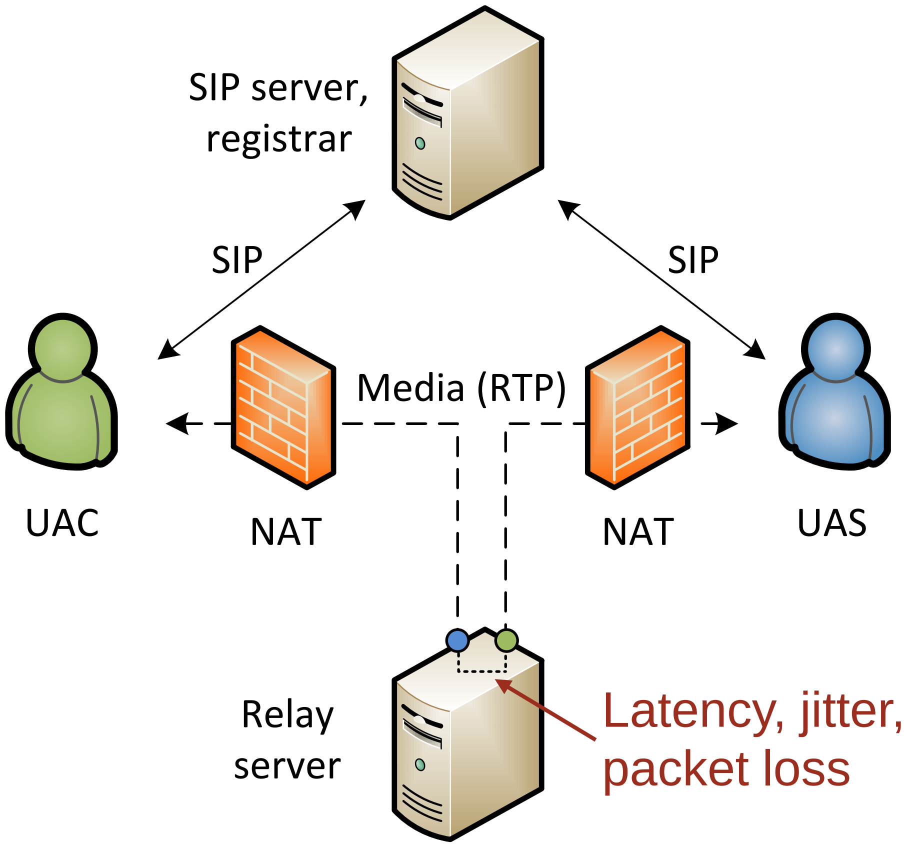
Voice over IP Use Case

- The most common solution is to use a relay server
- The server allocates ports to be used to receive RTP traffic on behalf of both users



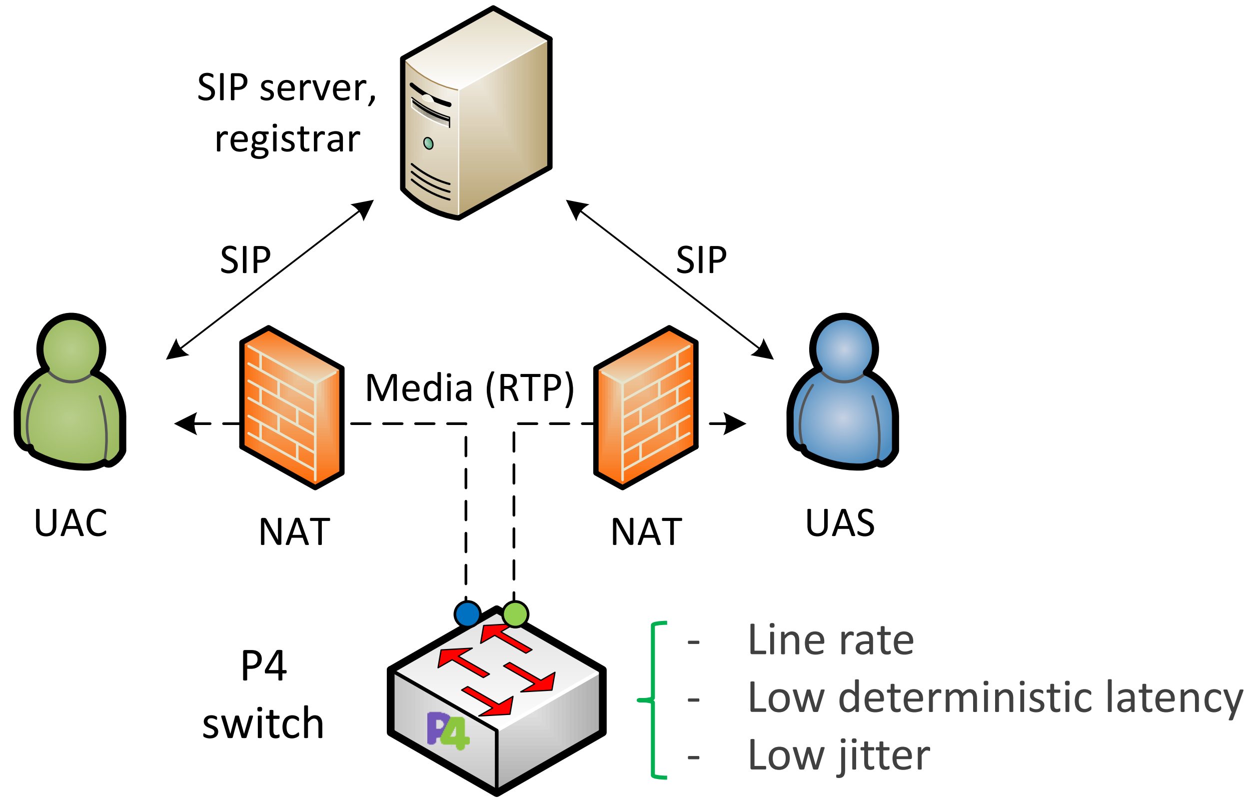
Voice over IP Use Case

- The most common solution is to use a relay server
- The server allocates ports to be used to receive RTP traffic on behalf of both users



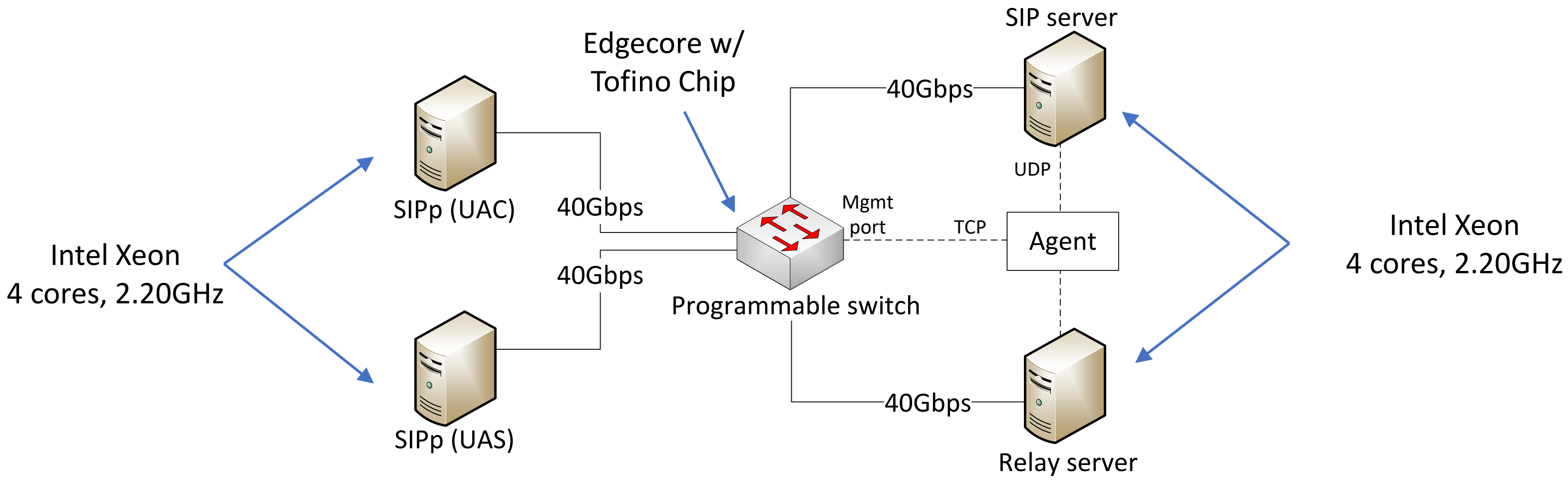
Proposed System

- Emulate the behavior of the relay server using programmable switch
- The switch parses packets and modifies headers to relay the traffic



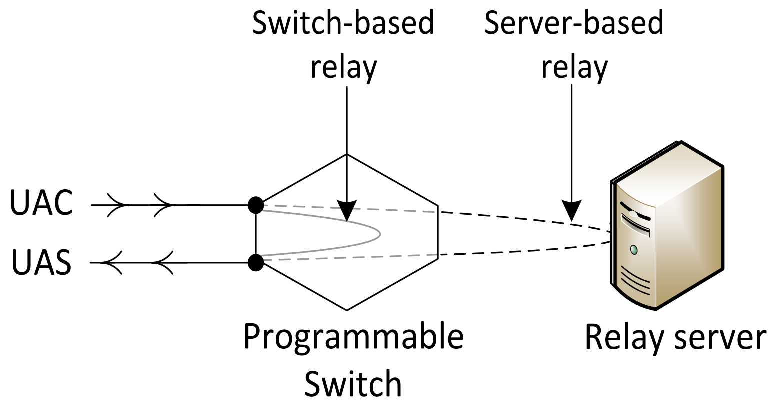
Implementation and Evaluation

- OpenSIPS, an open source implementation of a SIP server
- RTPProxy, a high-performance relay server for RTP streams
- SIPp: an open source SIP traffic generator that can establish multiple concurrent sessions and generate media (RTP) traffic
- Iperf3: traffic generator used to generate background UDP traffic
- Edgecore Wedge100BF-32X: programmable switch



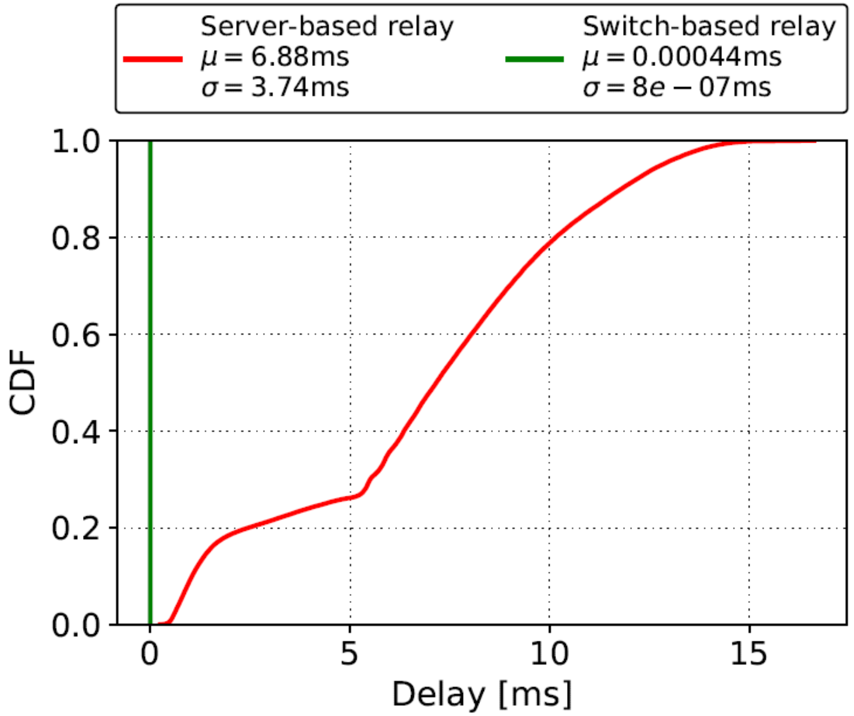
Implementation and Evaluation

- Two scenarios are considered:
  - “Server-based relay”: relay server is used to relay media between end devices
  - “Switch-based relay”: the switch is used to relay media
- UAC (SIPp) generates 900 media sessions, 30 per second
- The test lasts for 300 seconds
- G.711 media encoding codec (160 bytes every 20ms)



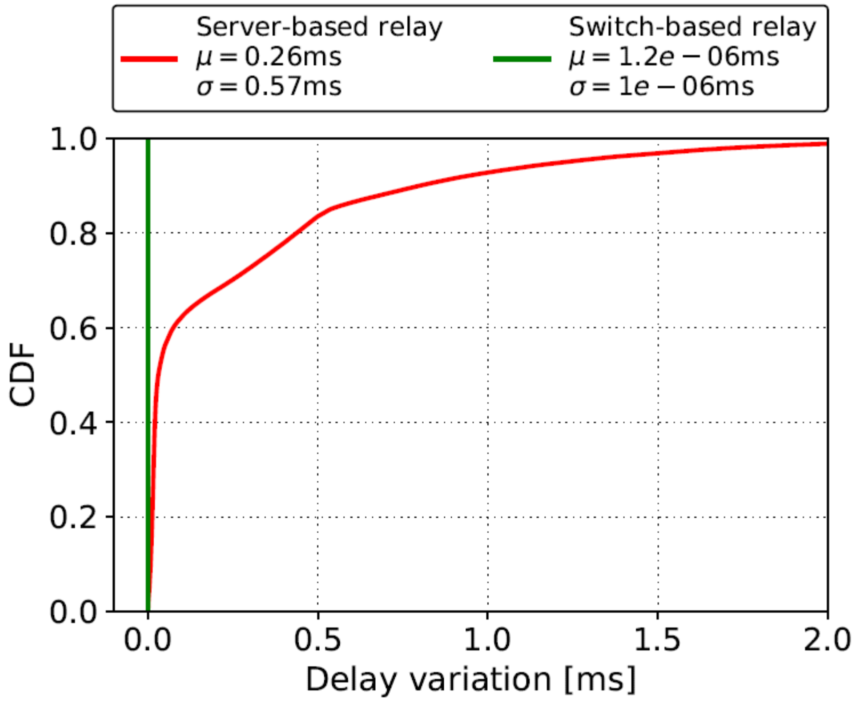
Results

- Delay contributions of the switch and the relay server



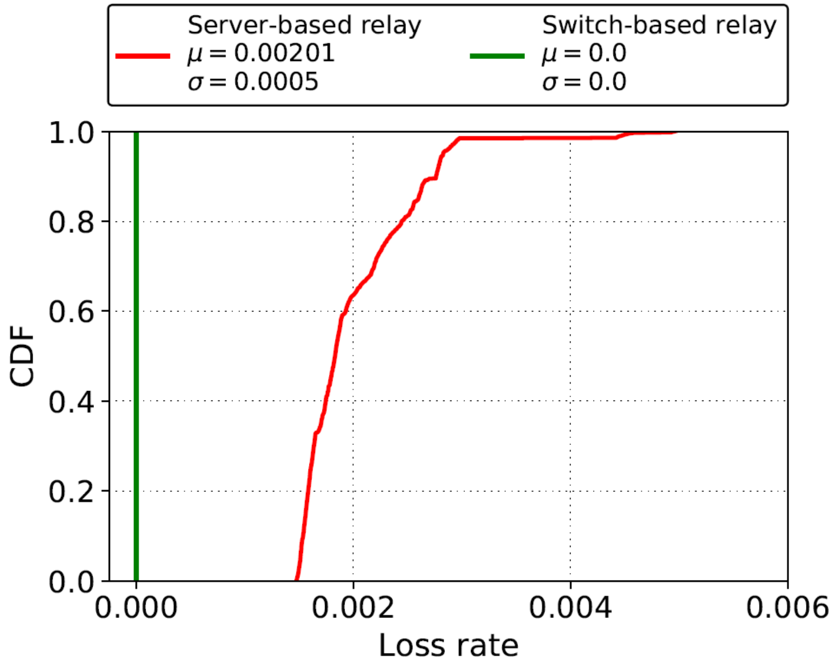
Results

- Delay variation: the absolute value of the difference between the delay of two consecutive packets
  - Analogous to jitter, as defined by RFC 4689



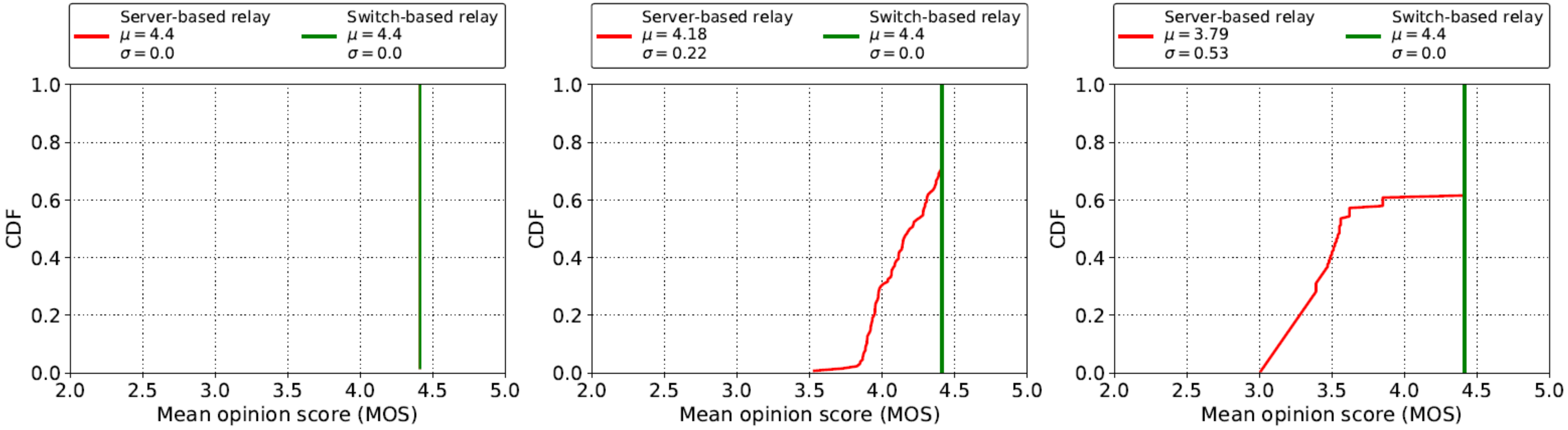
Results

- Loss rate: number of packets that fail to reach the destination
  - Calculation is based on the sequence number of the RTP header



Results

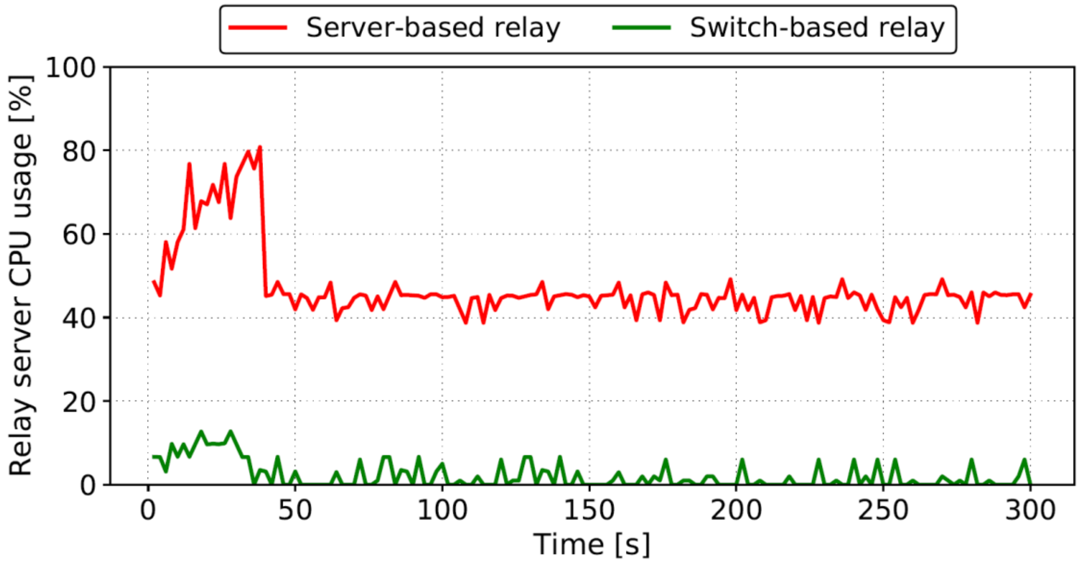
- Mean Opinion Score (MOS): estimation of the quality of the media session
  - A reference quality indicator standardized by ITU-T
  - Maximum for G.711 is ~4.4



(a) 750 simultaneous sessions. (b) 1500 simultaneous sessions. (c) 1800 simultaneous sessions.

Results

- CPU usage: the percentage of the CPU's capacity used by the relay server



Resource Consumption

- The prototype is implemented in two different scenarios:
  - On top of the baseline switch program (switch.p4): implements various features including Layer 2/3 functionalities, ACL, QoS, etc.
  - Standalone implementation

On top of switch.p4			
Table size	**SRAM**	**Hash Bits**	**TCAM**
32,000	+8.45%	+2.7%	+0%
64,000	+16.2%	+4.6%	+0%

Standalone program			
Table size	**SRAM**	**Hash Bits**	**TCAM**
500,000	——-	——-	——-
1,000,000	+97.84%	+86.4%	+0%
1,050,000	+107.5%	+89.8%	+0%

Additional hardware resources used when the solution is deployed on top of switch.p4 and as a standalone program

Resource Consumption

- The prototype is implemented in two different scenarios:
  - ➢ On top of the baseline switch program (switch.p4): implements various features including Layer 2/3 functionalities, ACL, QoS, etc.
  - ➢ Standalone implementation

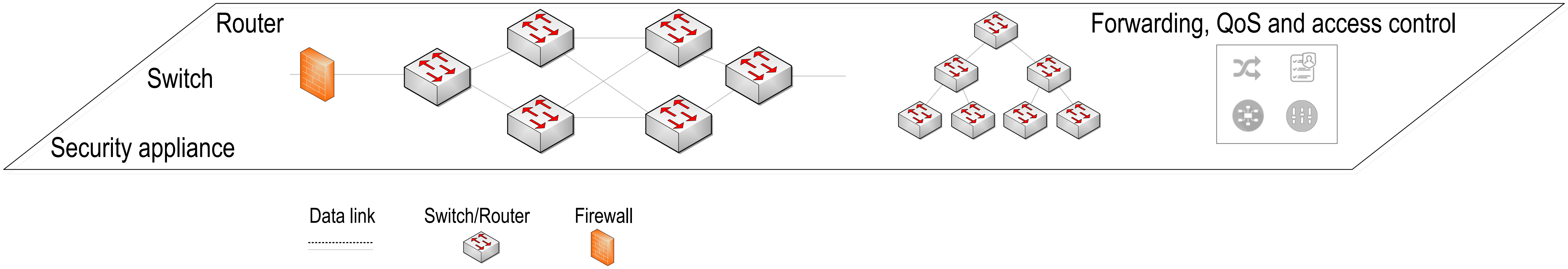
On top of switch.p4			
Table size	SRAM	Hash Bits	TCAM
32,000	+8.45%	+2.7%	+0%
64,000	+16.2%	+4.6%	+0%
Standalone program			
Table size	SRAM	Hash Bits	TCAM
500,000	--------	--------	--------
1,000,000	+97.84%	+86.4%	+0%
1,050,000	+107.5%	+89.8%	+0%

Additional hardware resources used when the solution is deployed on top of switch.p4 and as a standalone program

	Programmable Switch	General-purpose CPU
Cost	$6,000	$ 10,000 - 25,000
Capacity	Million connections per switch	~500 connections per core
Latency	400 nanoseconds	Tens to hundreds of milliseconds

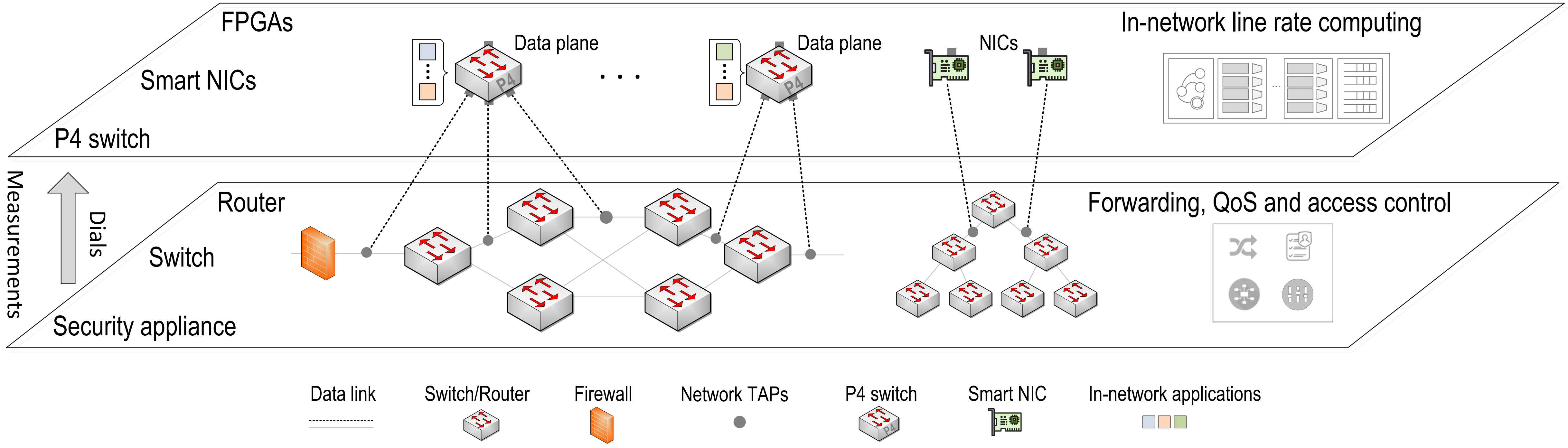
Architecture for Incremental Deployment of PDPs

Proposed Architecture

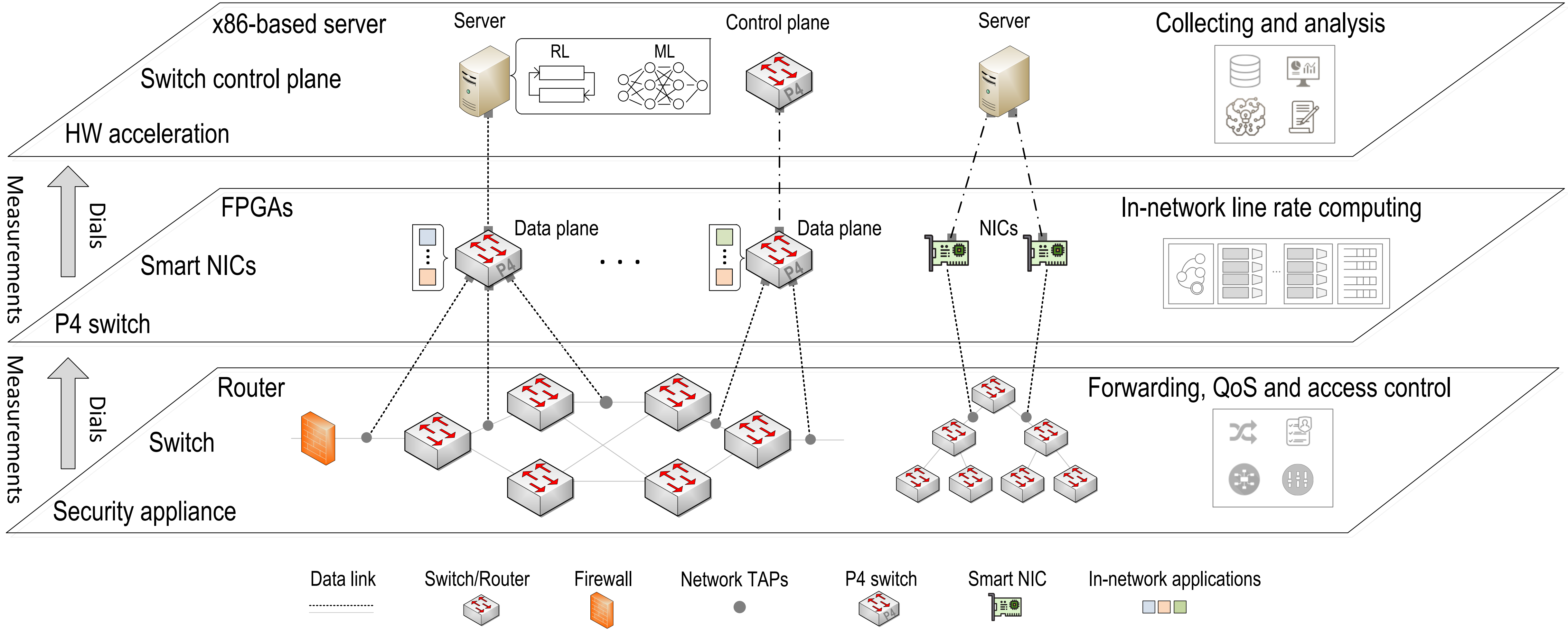


South Carolina

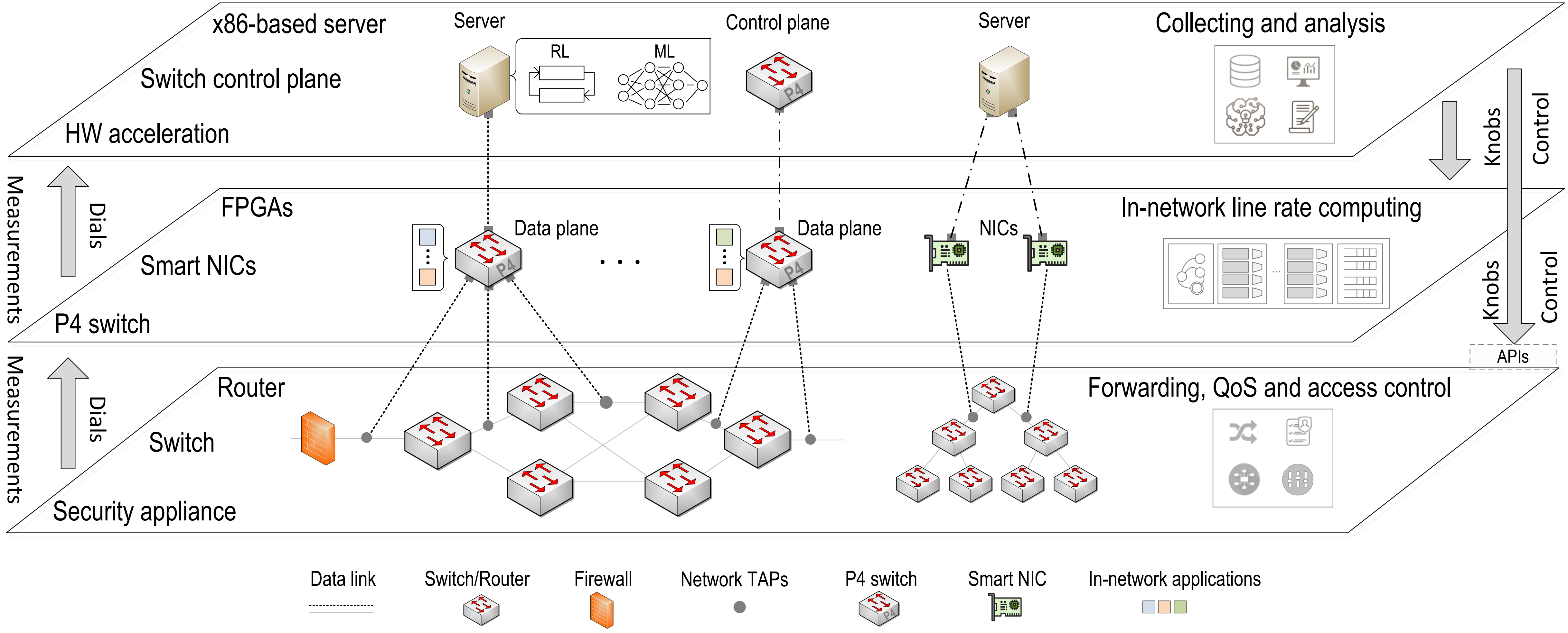
Proposed Architecture



Proposed Architecture

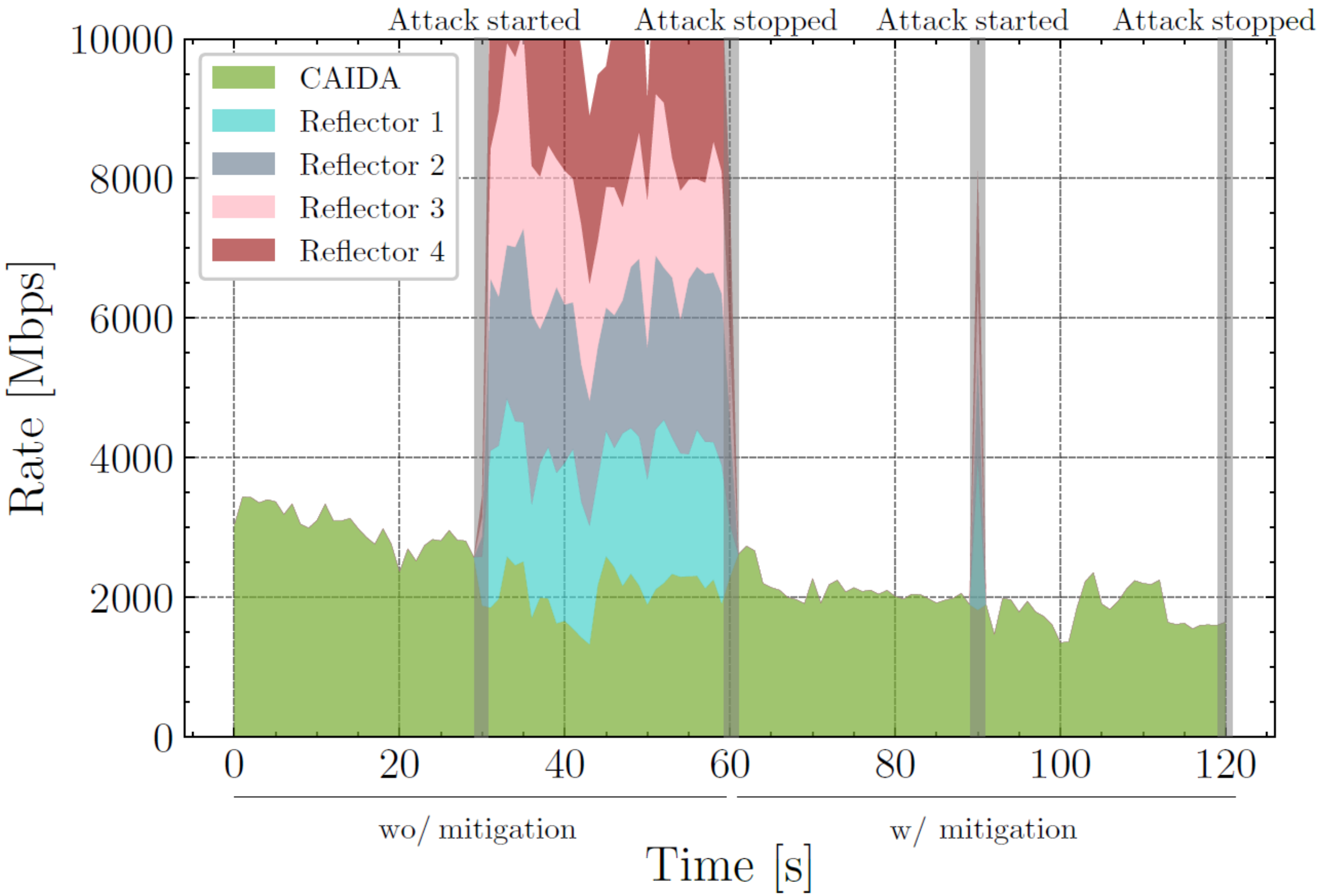


Proposed Architecture



Detecting DNS Amplification with P4

- CAIDA traffic replayed
- > 10Gbps DNS amplification attack generated
- Attack was mitigated in < 1s



South Carolina

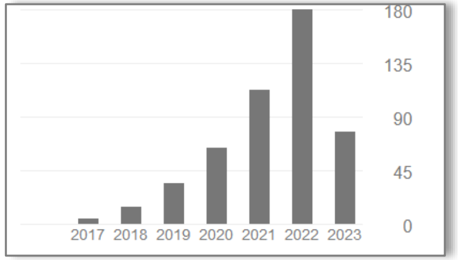
Conclusion

- Contributions:
  - Improving the QoS by **dynamically sizing the buffer**
  - Improving the QoS and fairness through **traffic classification and separation**
  - Scaling and optimizing media's QoS by **offloading packet processing** from CPUs to PDP
  - **Fostering PDP adoption** by proposing a passive PDP deployment architecture

- Future work:
  - Deploying and testing the systems on **FABRIC**, a US-based nation-wide testbed
  - Exploring network programmability on **SmartNICs**

Selected Publications

Cited by	All	Since 2018
Citations	493	487
h-index	12	12
i10-index	15	15



- Book:
  - J. Crichigno, E. Kfoury, E. Bou-Harb, N. Ghani, "High-Speed Networks: A Tutorial (Practical Networking)", Springer International Publishing, December 2021
- Journals:
  - E. Kfoury, J. Crichigno, and E. Bou-Harb. P4BS: Leveraging Passive Measurements from P4 Switches to Dynamically Modify a Router's Buffer Size, submitted to *IEEE Transactions on Network and Service Management*, 2023.
  - E. Kfoury, J. Crichigno, and E. Bou-Harb. P4Tune: Enabling Programmability in Non-Programmable Networks, *IEEE Communications Magazine*, 2023.
  - E. Kfoury, J. Crichigno, and E. Bou-Harb. An Exhaustive Survey on P4 Programmable Data Plane Switches: Taxonomy, Applications, Challenges, and Future Trends. *IEEE Access*, 2021.
  - E. Kfoury, J. Gomez, J. Crichigno, and E. Bou-Harb. An Emulation-based Evaluation of TCP BBRv2 Alpha for Wired Broadband. *Elsevier Computer Communications*, 2020.
  - E. Kfoury, J. Gomez, J. Crichigno, E. Bou-Harb, and D. Khoury. Decentralized Distribution of PCP Mappings Over Blockchain for End-to-End Secure Direct Communications. *IEEE Access*, 2019
  - E. Kfoury, J. Crichigno, E. Bou-Harb, "P4CCI: P4-based Online TCP Congestion Control Algorithm Identification for Traffic Separation", IEEE International Conference on Communications (ICC), Rome, Italy, June 2023.
- Conferences
  - E. Kfoury, J. Crichigno, E. Bou-Harb, G. Srivastava, "Dynamic Router's Buffer Sizing using Passive Measurements and P4 Programmable Switches", *IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, December 2021.
  - E. Kfoury, J. Crichigno, E. Bou-Harb, V. Gurevich, "Offloading Media Traffic to Programmable Data Plane Switches," *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, June 2020.
  - E. Kfoury, J. Crichigno, E. Bou-Harb, D. Khoury, G. Srivastava, "Enabling TCP Pacing using Programmable Data Plane Switches", 42nd *International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Hungary, July 2019
  - E. Kfoury, D. Khoury, A. AlSabeh, J.Gomez, J. Crichigno, E. Bou-Harb, "A Blockchain-based Method for Decentralizing the ACME Protocol to Enhance Trust in PKI", 43rd *International Conference on Telecommunications and Signal Processing (TSP)*, Milan, Italy, July 2020

Acknowledgement

- Thanks to the National Science Foundation (NSF)
- This work was supported by NSF, Office of Advanced Cyberinfrastructure (OAC), awards 1925484, 1829698, 1907821, and 2118311

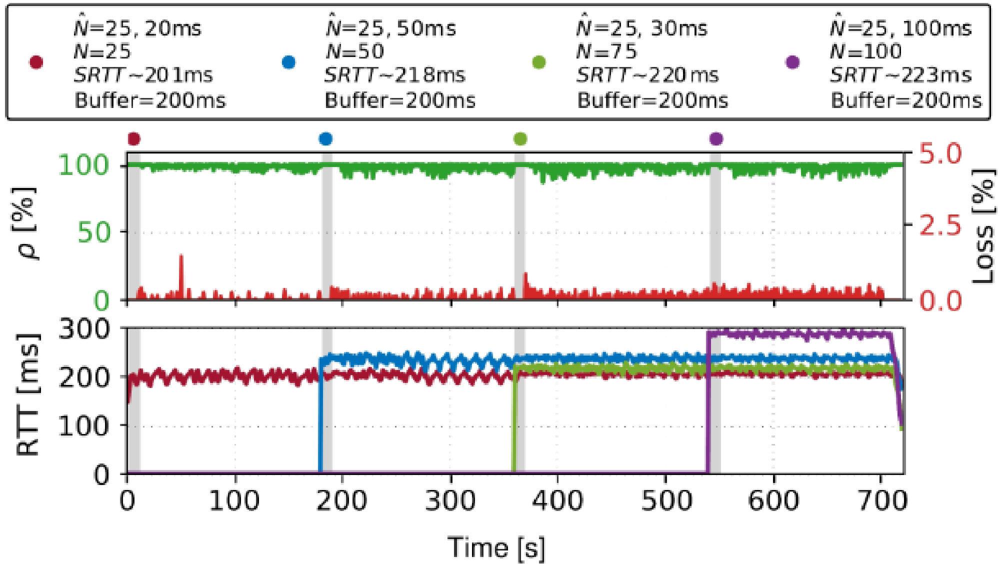
Thank you!

Questions?

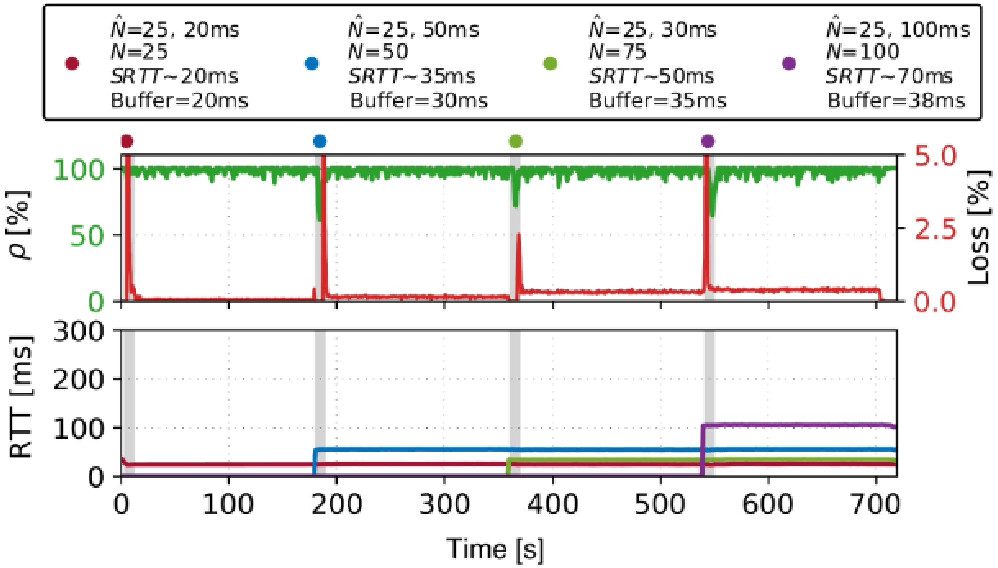
Results

- Long flows with different emulated propagation delays
- 100 long flows, divided into four groups of each 25 flows each
- Each group starts three minutes after the other
- CUBIC congestion control algorithm

wo/ buffer modification



w/ buffer modification

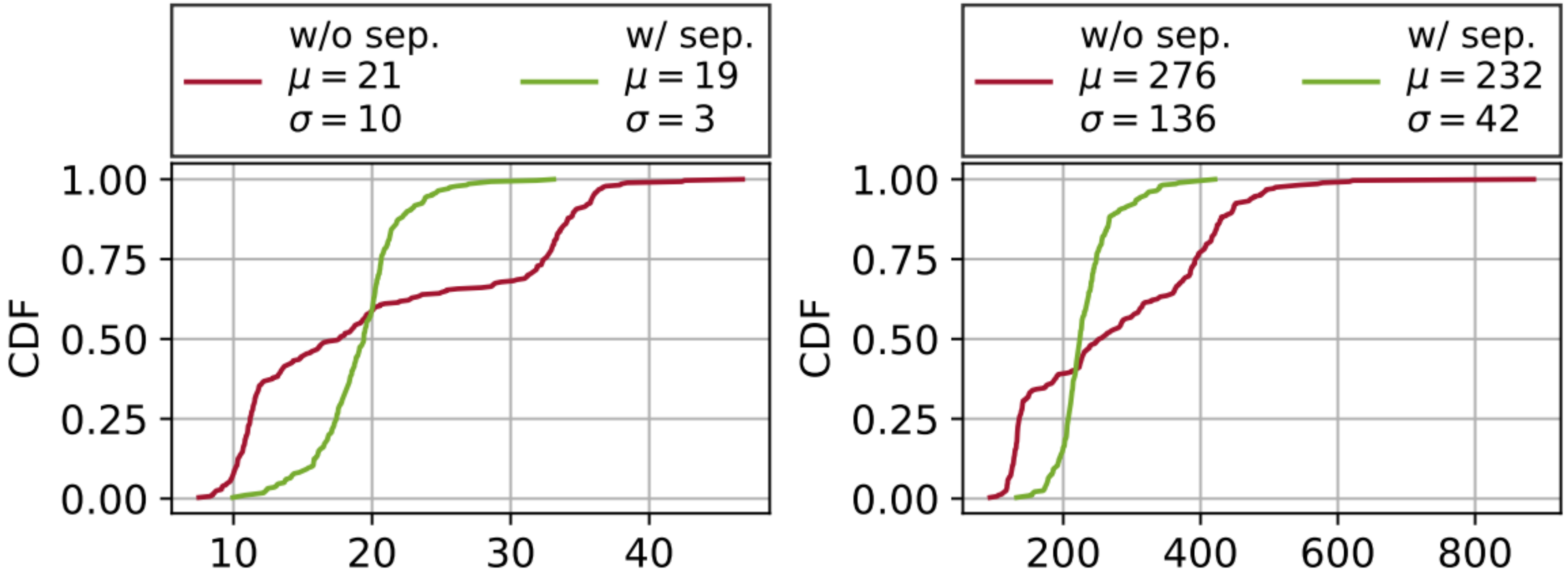


Time Series Preparation and Deep Learning

- BIF values are pushed to the control plane of the PDP switch during congestion
- A time series is constructed
- Two pre-processing steps:
  - **Outliers Rejection**: z-score method, which uses the MAD (Median Absolute Deviation), is used
  - **Normalization**: The time series is preprocessed using z-normalization
- Fully Convolutional Neural Networks (FCNs) used to classify the univariate time series

Flow Completion Time (Long Flows)

- 10 long flows started at the same time, with alternating CCAs
  - Flow1 uses CUBIC, Flow2 uses BBR, Flow3 uses CUBIC, etc.
- Each flow transfers a 500MB file
- In a fair network with a bottleneck of 2Gbps and 10 active flows:
  - Each flow is transferring at 200Mbps
  - FCT = 500MB / 200Mbps = 20s



Middlebox Devices

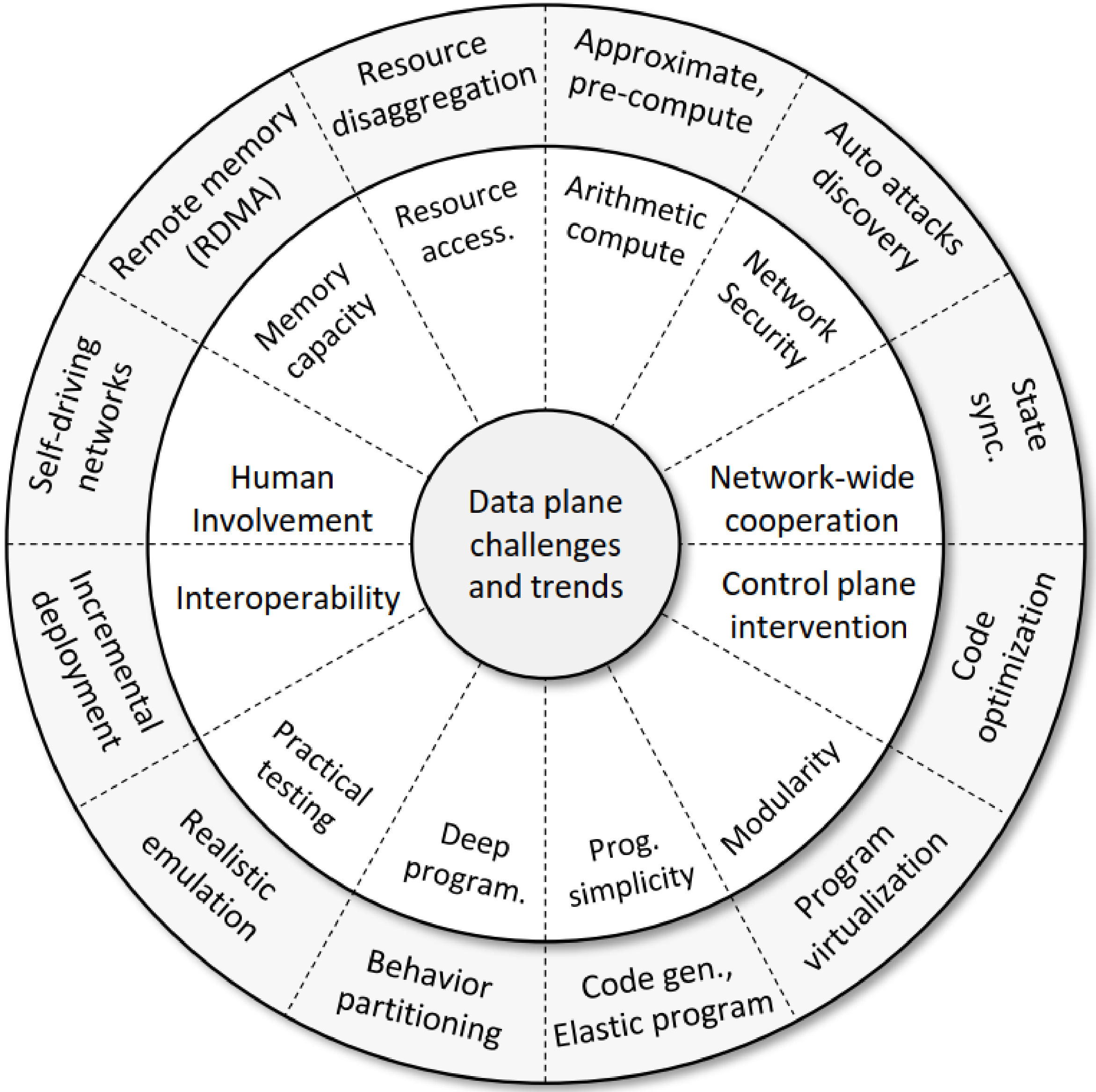
- RFC 3234[1] defines middlebox as a device that performs functions other than the standard functions of an IP router
- Legacy middleboxes are designed and implemented by manufacturers
- Examples:
  - Network address translators (NAT)
  - Firewalls
  - Intrusion Detection Systems (IDS)
  - Proxy servers (e.g., VoIP relay server)
- Legacy middleboxes are limited to the functions provided by the manufacturers
  - Expensive
  - Difficult to upgrade
  - Function-specific

[1]Carpenter, Brian, and Scott Brim. Middleboxes: Taxonomy and issues. RFC 3234, February 2002.

Performance Issues of Middleboxes

- The trend lately has been moving towards implementing middleboxes in servers
- Network Function Virtualization (NFVs)
- While this shift accelerated innovation, it induced performance issues (e.g., delay, jitter)
  - Operating systems' scheduling delays
  - Interrupt processing latency
  - Other low-level OS functions

PDP Challenges



P4 Switches Deployment Challenges

- Data plane programmability knowledge by operators
  - Operators only configure legacy devices (e.g., modify routing configuration, updating ACL)
  - Programming P4 targets is complex[1]
- Cost of replacing the existing infrastructure
  - Significant costs, time, and efforts spent in building the network and the existing equipment
  - Replacing these devices with P4 switches would incur significant costs
- Vendor support
  - The support in legacy devices is readily available
  - P4 switches are whiteboxes, with little to no support from vendors
- Network disruption
  - P4 programs might be potential sources of packet-processing error
  - Bugs can lead to network disruption, affecting the availability of the services

[1] The switch.p4 program, which contains the standard switch capabilities, has more than 10^{30} control paths