Remote TCP Connection Offload with XO

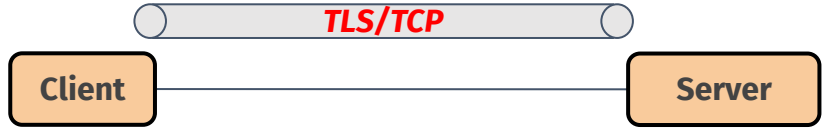
Shuo Li*, Steven W.D. Chien*, Tianyi Gao, Michio Honda
University of Edinburgh

Coseners 2025

THE UNIVERSITY of EDINBURGH
informatics

TCP in Scale-out Systems

- TCP is (still) dominant in the cloud
  - OS enhancements (zero-copy, I/O batching etc)
  - NIC offloading
    - Segmentation offload
    - TLS offload

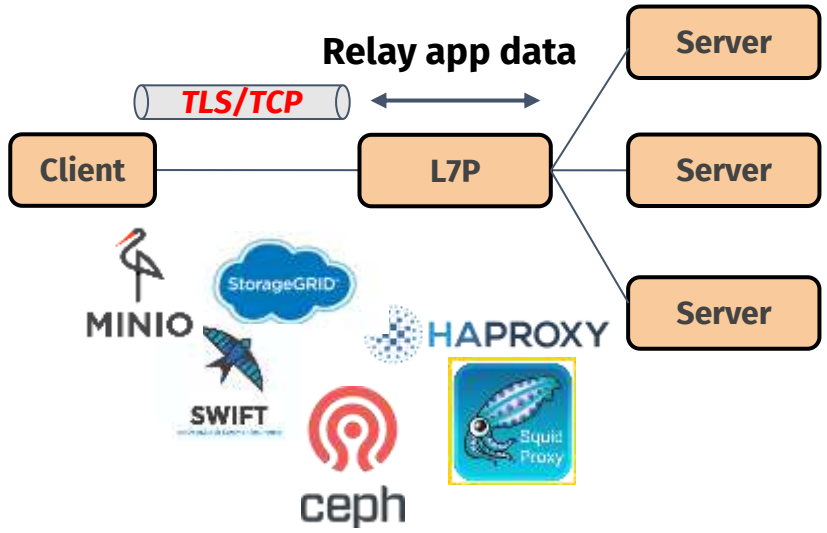


- TCP servers constitute Scale-out System

L7LB in Scale-out Systems

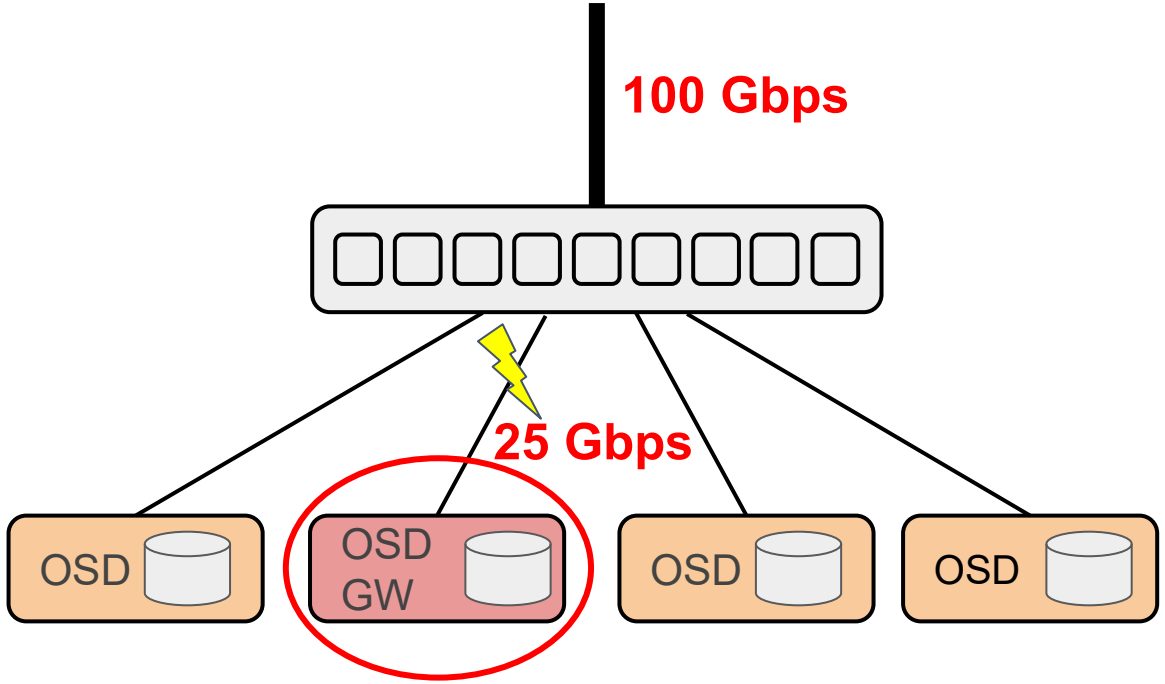
- A scale-out system
  - Higher service throughput with a load balancer
  - Higher storage capacity with a storage gateway

- Layer 7 Proxy (L7P)
  - terminate a client connection
  - select a server
  - relay data between the client and server



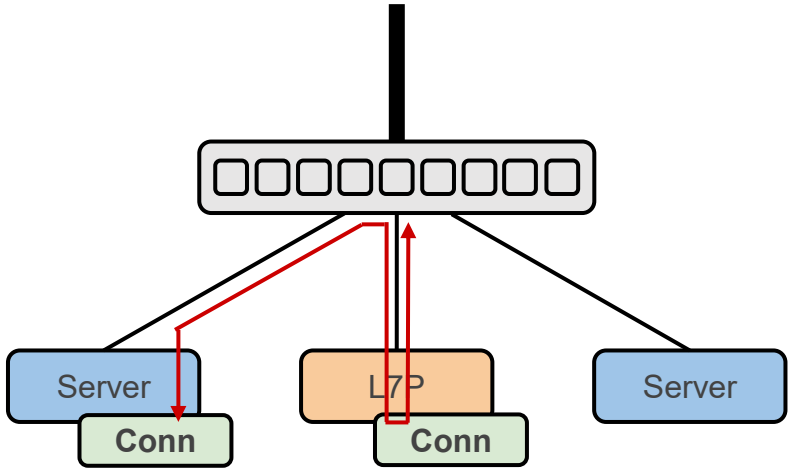
L7P Introduces Bottenecks

- Ceph study
  - Bandwidth bottleneck
  - CPU bottleneck
  - Underutilizes server resources



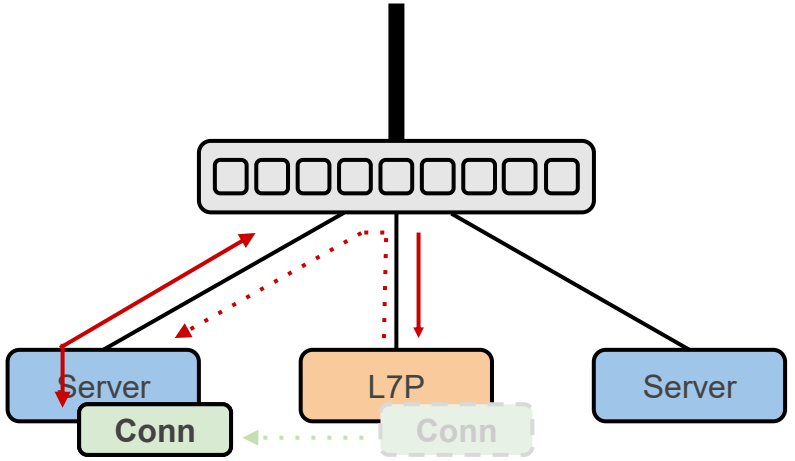
Existing Solutions

Connection Splicing
(AccelTCP [NSDI'20])



- Pros
  - Offload data relay to the kernel or NIC
- Cons
  - Static L7P-server binding

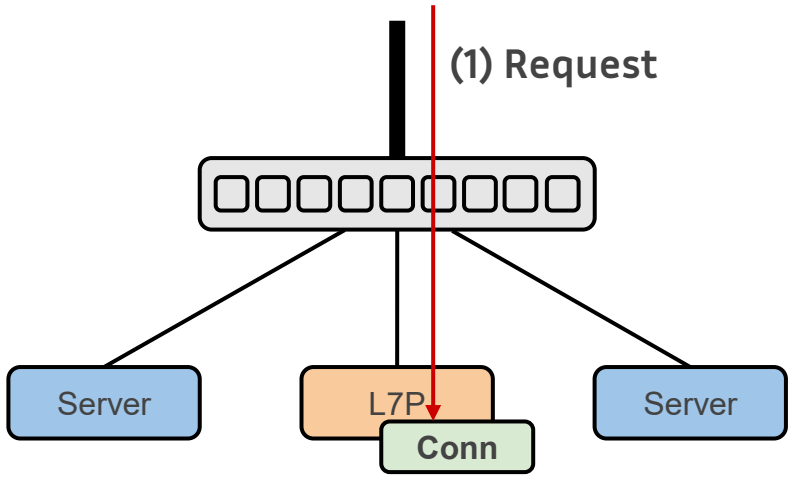
Connection Migration
(Prism [NSDI'21] & Capybara [ApSys'23])



- Pros
  - Bypass the app-level data relay
- Cons
  - Need programmable switch
  - Servers must be in the same rack

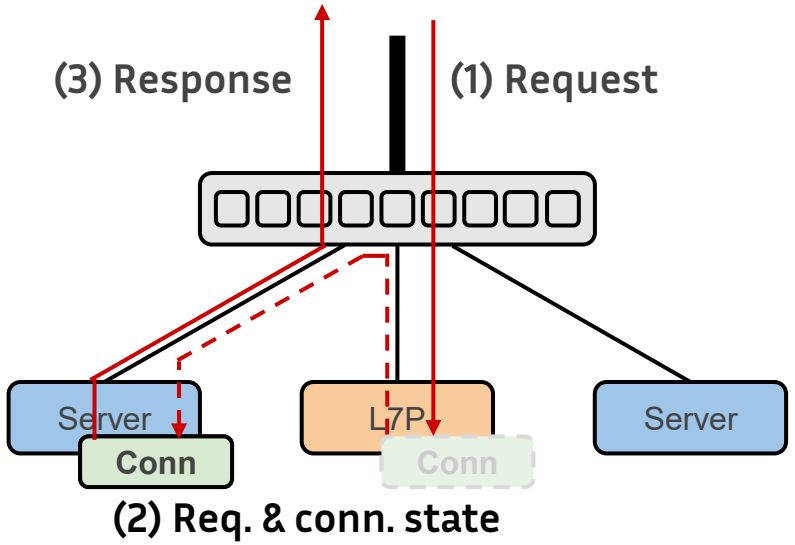
XO Approach

- TCP connection migration without programmable switch



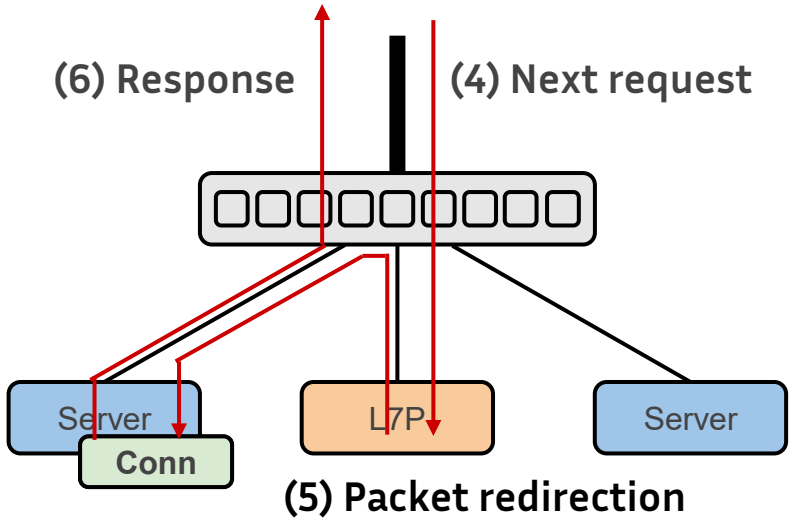
XO Approach

- TCP connection migration without programmable switch
  - Connection migration (although challenging or new protocol, see later) as usual



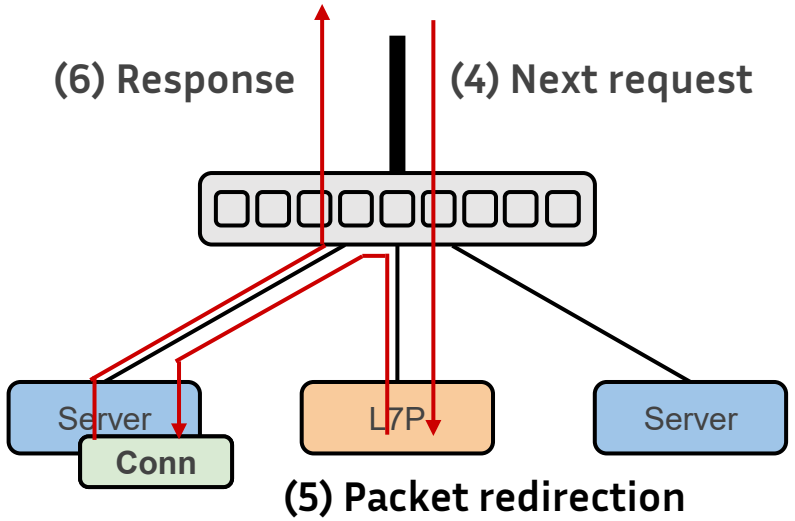
XO Approach

- TCP connection migration without programmable switch
  - Connection migration (although challenging or new protocol, see later) as usual
  - Flow-granularity packet redirection at the host



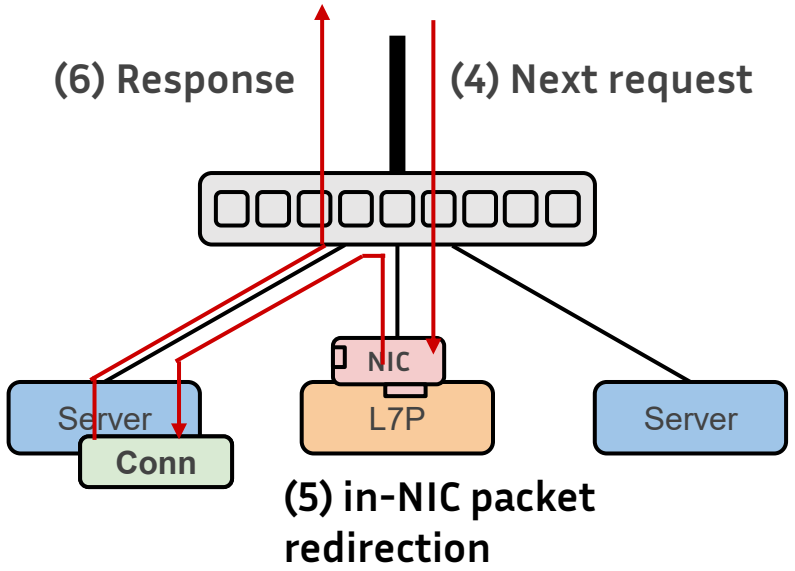
XO Approach

- TCP connection migration without programmable switch
  - Connection migration (although challenging or new protocol, see later) as usual
  - Flow-granularity packet redirection at the host
- ☹ Host-based redirection is not as efficient as switch-based redirection



XO Approach

- TCP connection migration without programmable switch
  - Connection migration (although challenging or new protocol, see later) as usual
  - Flow-granularity packet redirection at the host
- ☹ Host-based redirection is not as efficient as switch-based redirection
- 😀 NIC offloading (redirection behind the PCIe bus)
  - tc-flower



XO Building Blocks

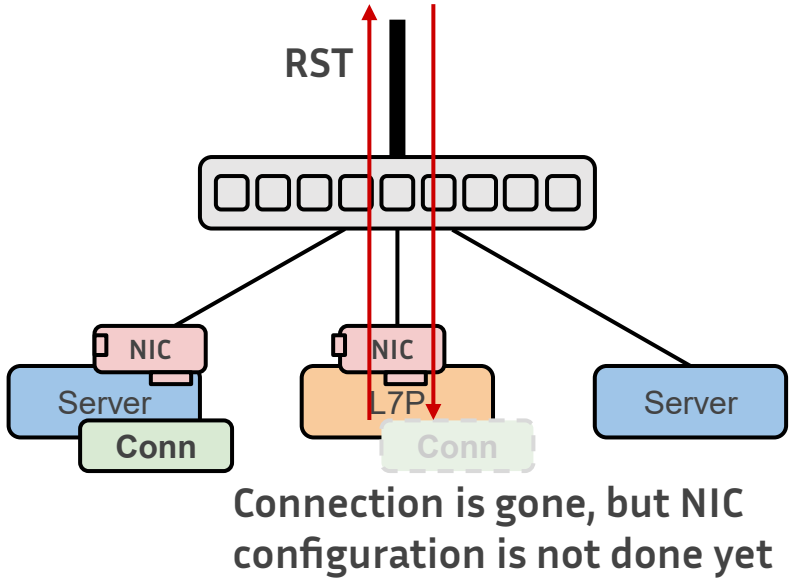
- New connection handoff protocol for robust host-based TCP connection migration
- New HW-SW hybrid packet redirection for efficient use of hw-based packet redirection
- A User space queue to manage rule insertion/deletion commands

XO Building Blocks

- New connection handoff protocol for robust host-based TCP connection migration
- New HW-SW hybrid packet redirection for efficient use of hw-based packet redirection
- A User queue to manage rule insertion/deletion commands

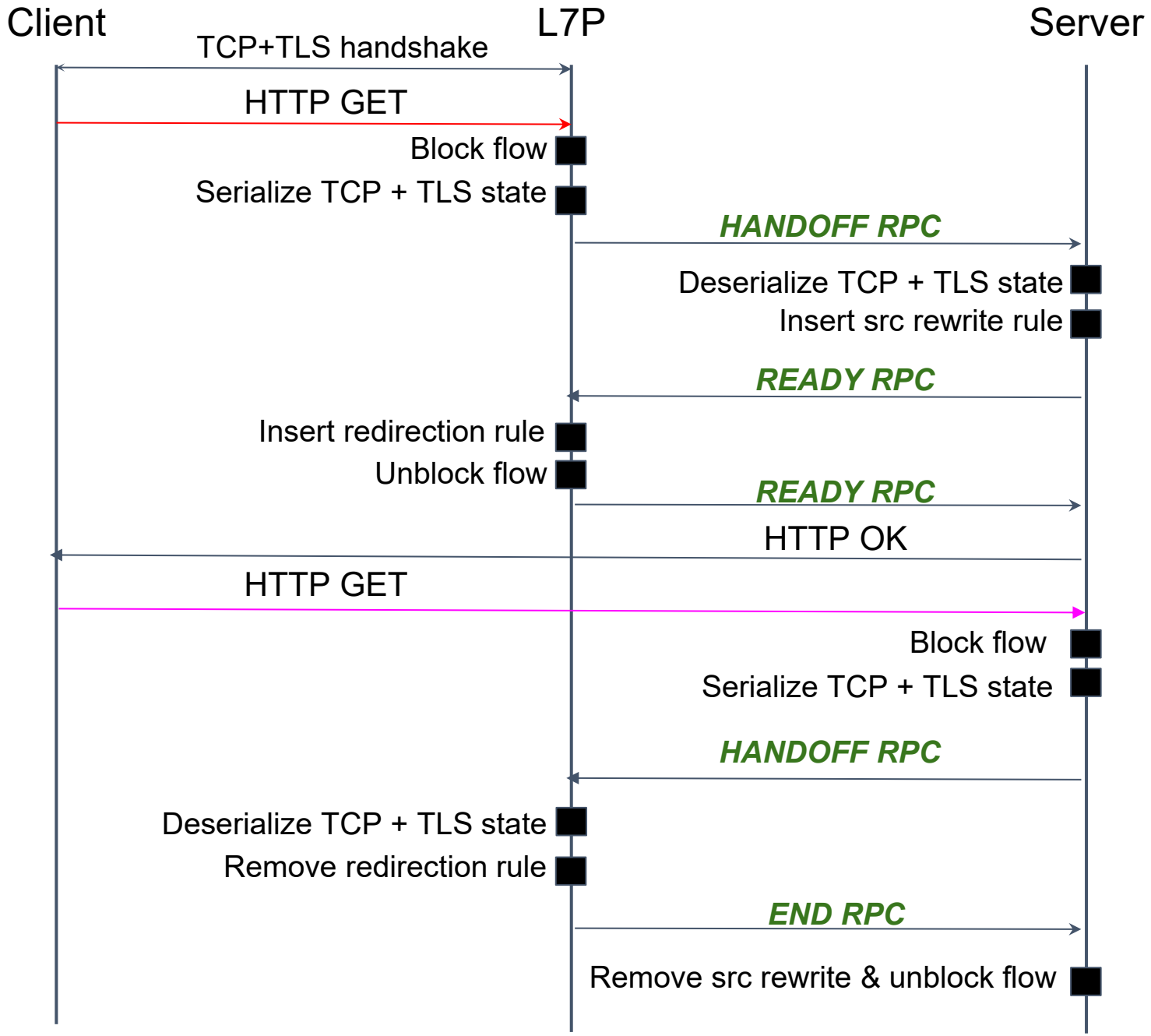
Host-based TCP connection handoff

- Many non-atomic operations
  - TCP/TLS connection serialization (many syscalls)
  - NIC configuration (many syscalls and device configuration)
  - Inter-host signalling (many RPCs)
- Ingress and egress packets during those operations break the connection



XO Handoff Protocol

- To avoid failure triggered by packet leaking

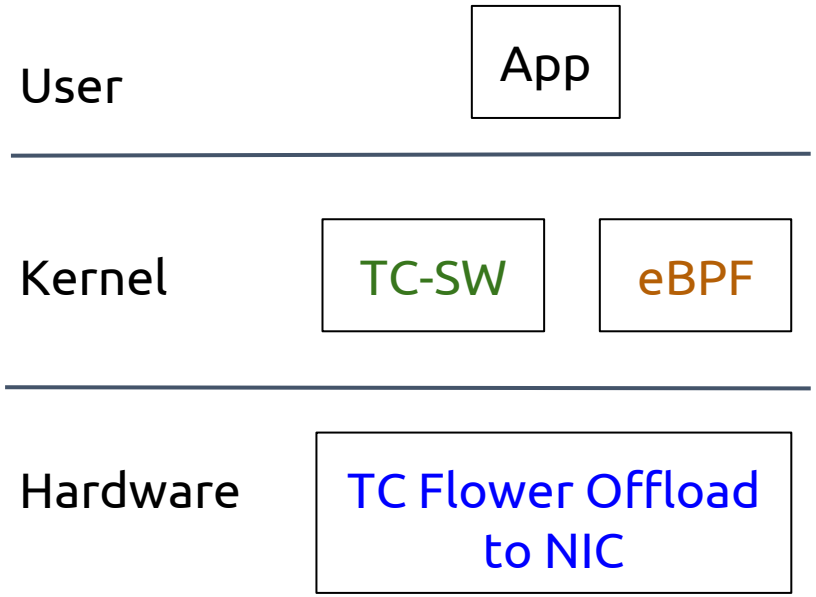


XO Building Blocks

- New connection handoff protocol for robust host-based TCP connection migration
- New HW-SW hybrid packet redirection for efficient use of hw-based packet redirection
- A User space queue to manage rule insertion/deletion commands

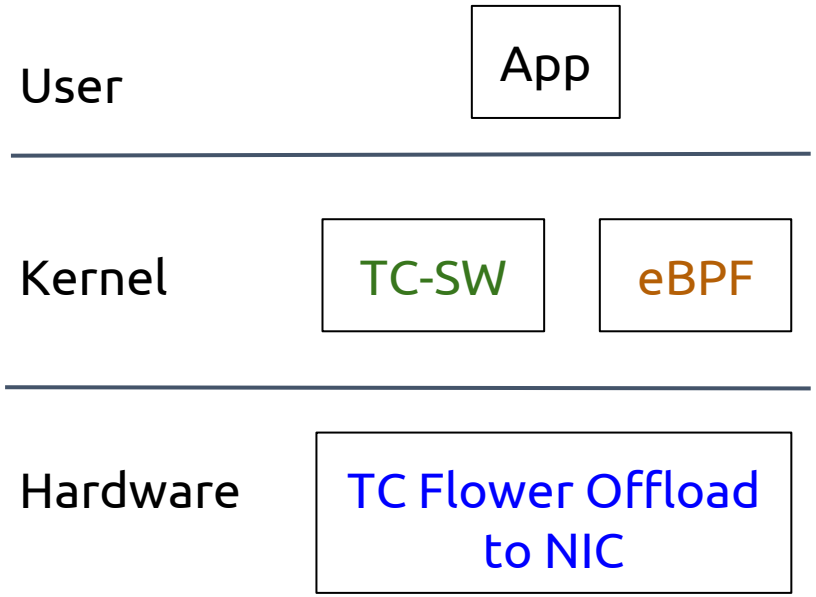
HW-SW Performance Tradeoff

- What packet redirection method should we use?



HW-SW Performance Tradeoff

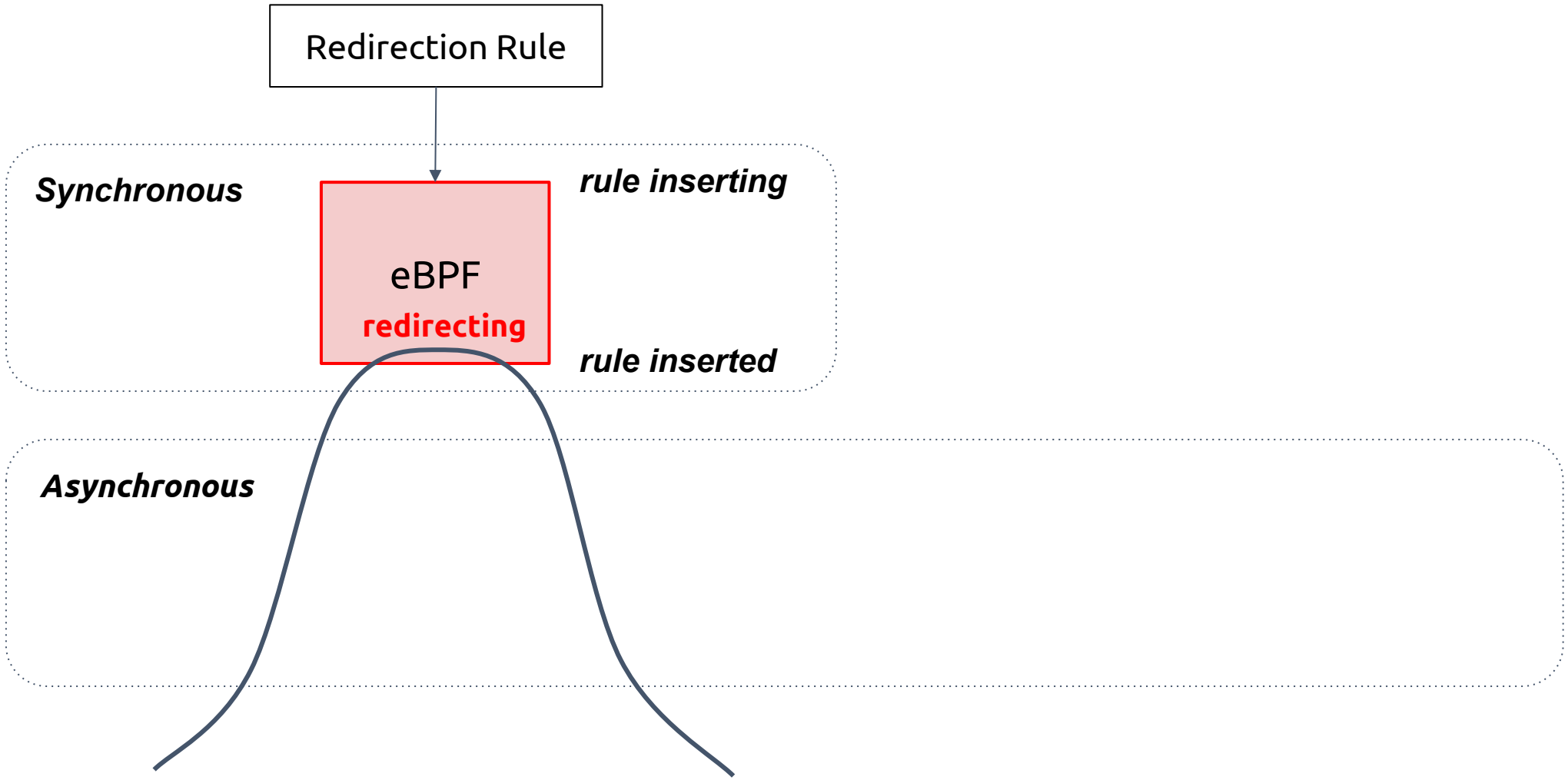
- What packet redirection method should we use?



- eBPF rule is fast to install but no offload
- TC's forwarding is much faster

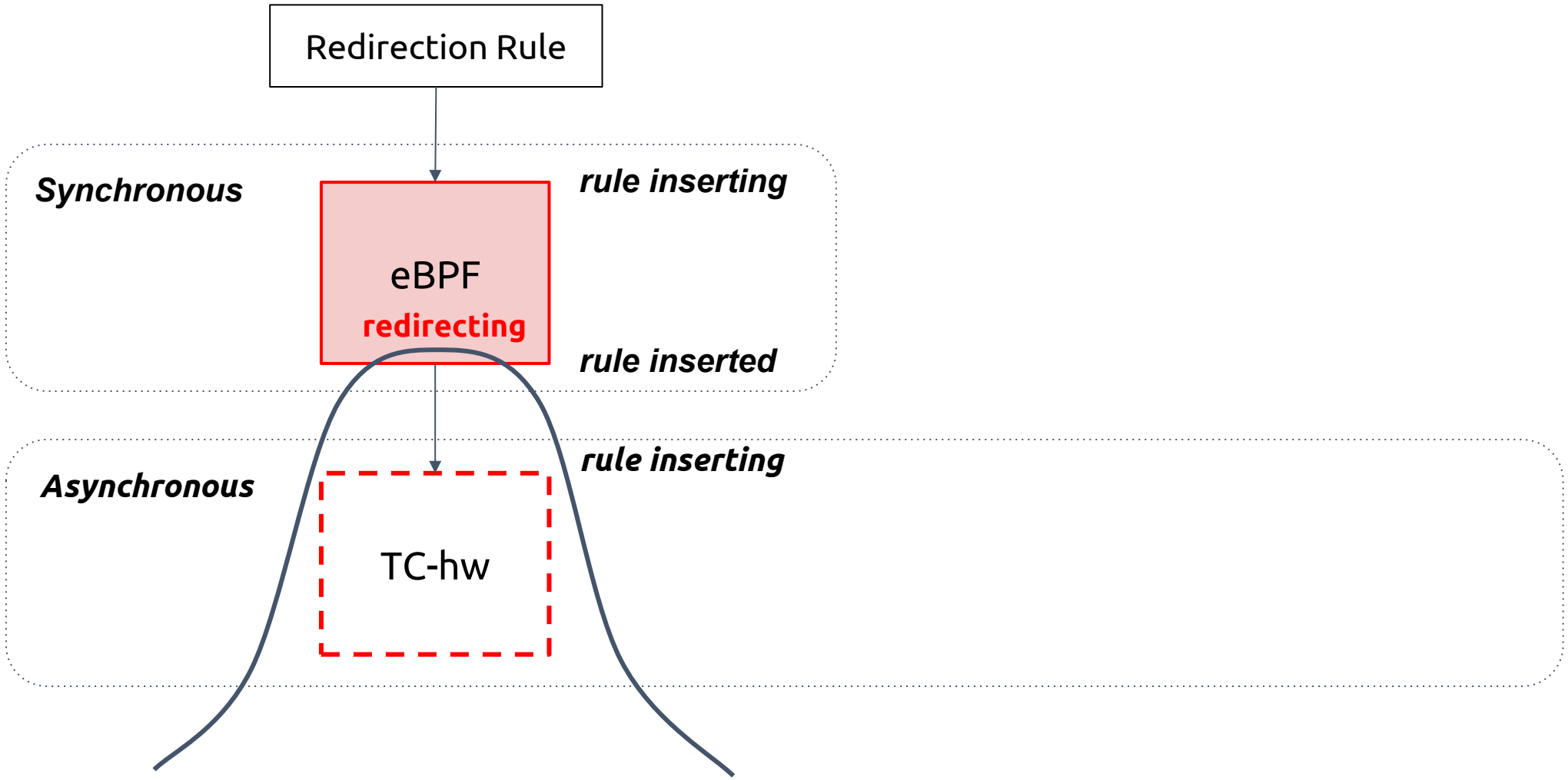
	Operation (µs)		Rate (Mpps)		Latency (µs)	
	Insert	**Remove**	**64B**	**1500B**	**64B**	**1500B**
eBPF (tc)	**4.01**	**3.77**	**0.79**	**0.78**	21.06	22.42
eBPF (XDP)	38.31	7.41	6.65	2.07	16.52	18.45
TC (CX5)	**476**	**404**	**33.01**	**2.07**	8.26	9.89
TC (CX7)	**2143**	**1134**	**33.08**	**2.07**	8.41	9.97
TC (Agilio)	68	65	22.12	2.07	19.77	20.58

HW-SW Hybrid Packet Redirection



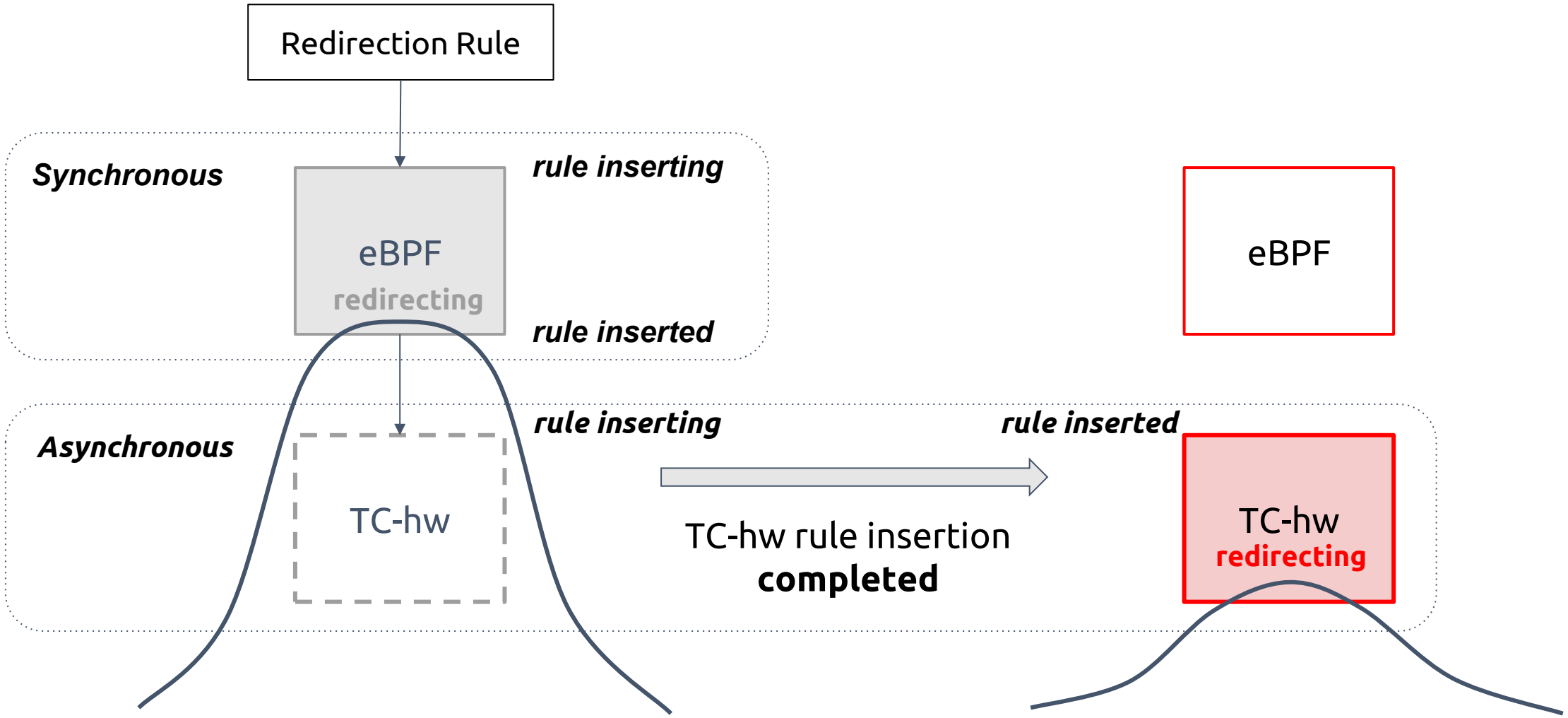
HW-SW Hybrid Packet Redirection

Use eBPF-based redirection until the HW one is activated



HW-SW Hybrid Packet Redirection

Use eBPF-based redirection until the HW one is activated

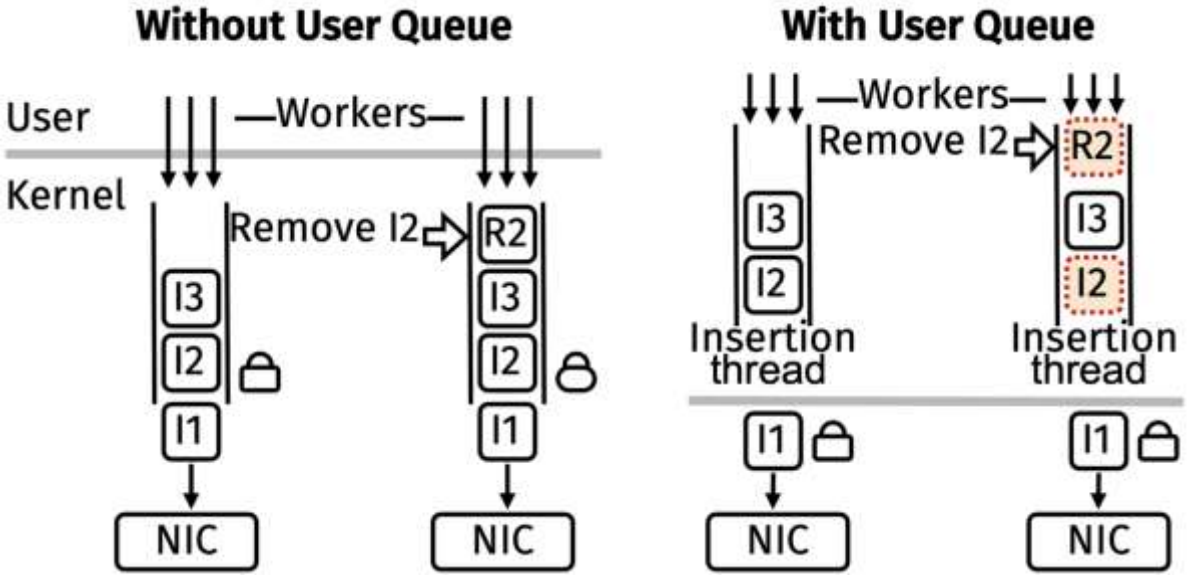


XO Building Blocks

- New connection handoff protocol for robust host-based TCP connection migration
- New HW-SW hybrid packet redirection for efficient use of hw-based packet redirection
- A User space queue to manage rule insertion/deletion commands

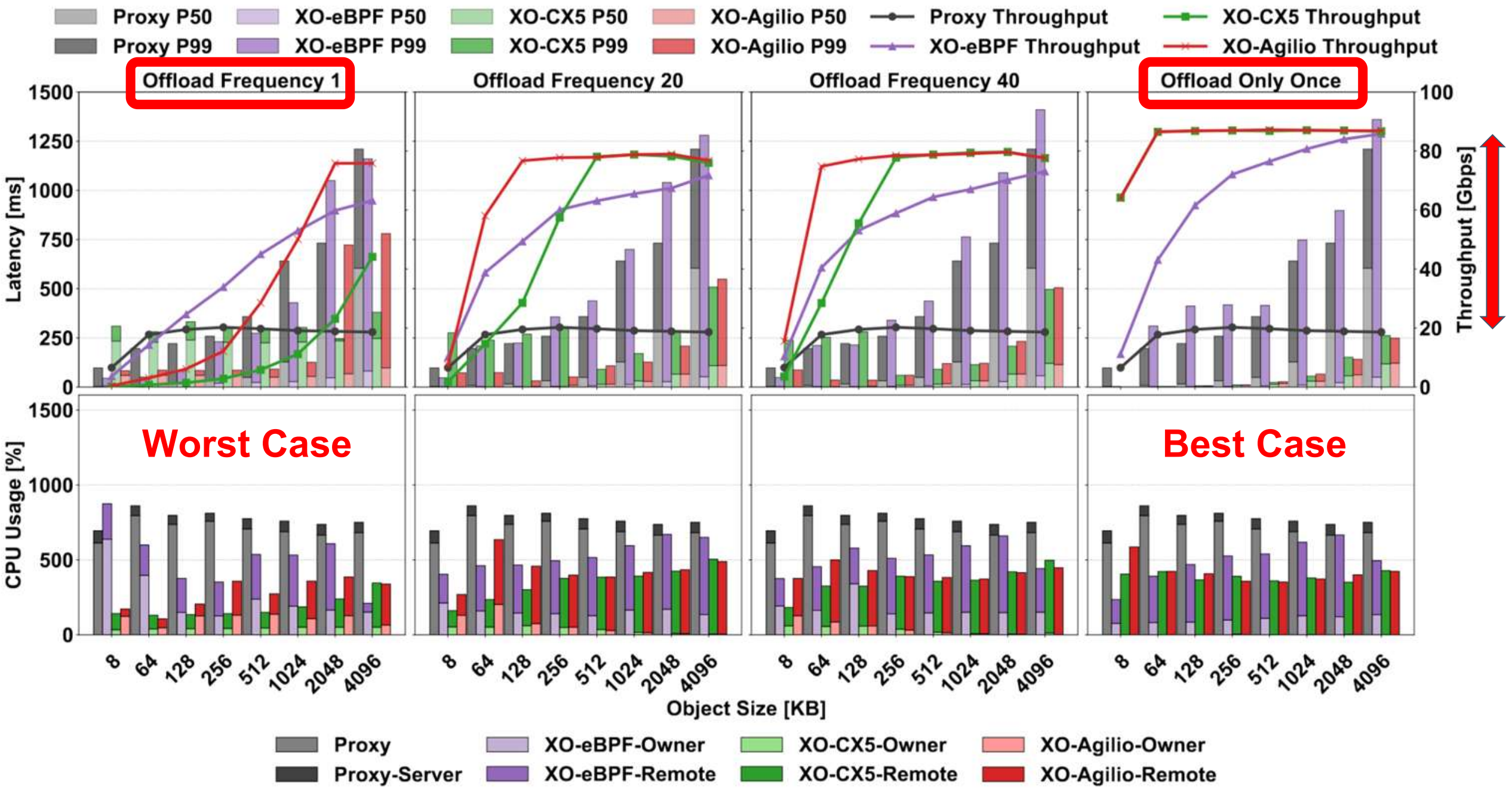
User space queue

- Observation
  - Rule insertion/deletion commands create backlog on the kernel over locks
- Problem
  - Latency unpredictability
  - Unnecessary command execution
- Solution
  - Moving the queue to the user space
    - Bounded command latency
    - Execution cancellation when no longer needed



Experiment Setup

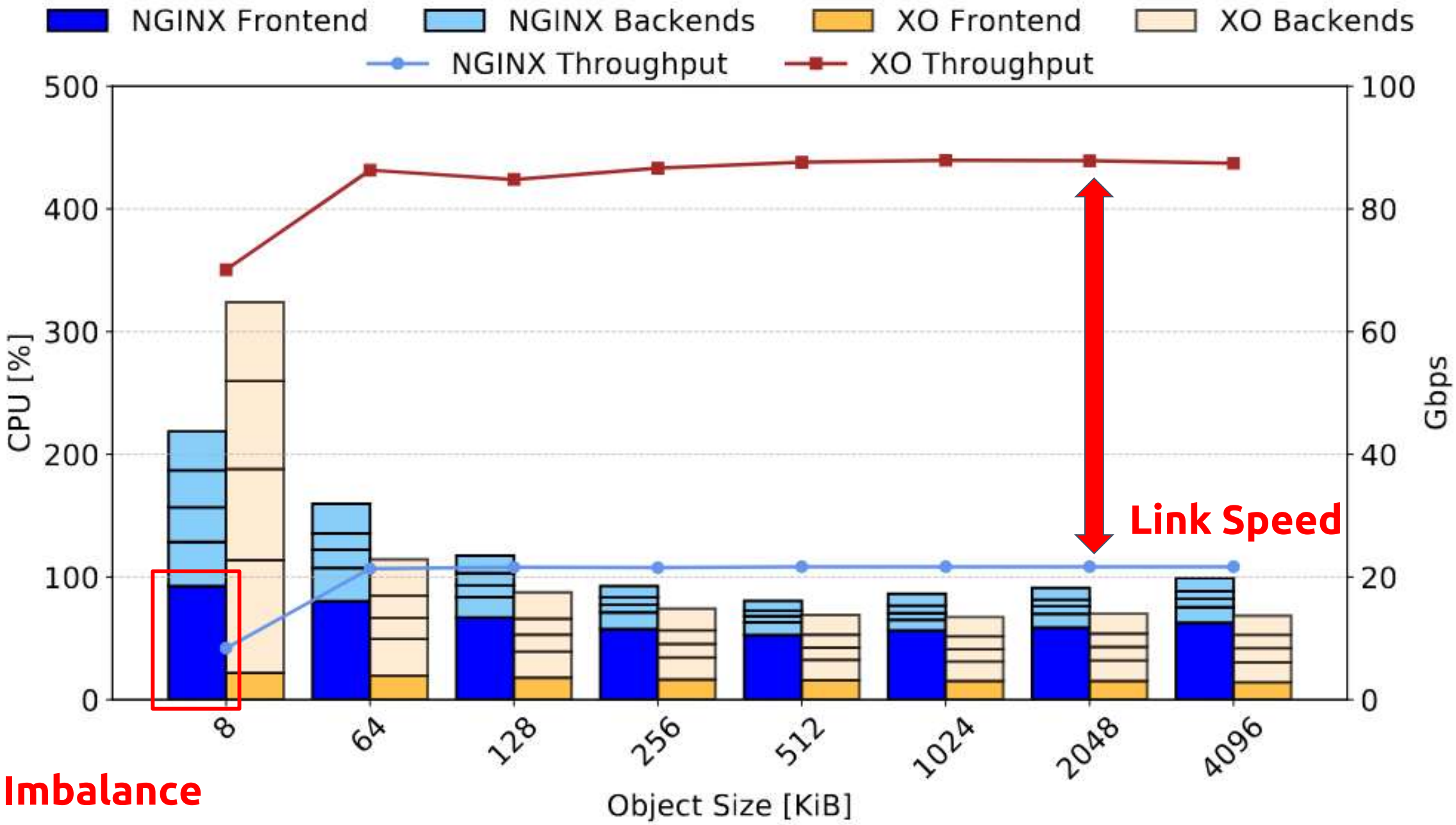
- 6-machine cluster
  - 1 client connects to a switch over 100Gbps link
  - 1 frontend with 25Gbps NICs
    - NVIDIA/Mellanox ConnectX-5
    - Netronome Agilio
  - 4 backends with 25Gbps NICs


Proxy P50
Proxy P99
XO-eBPF P50
XO-eBPF P99
XO-CX5 P50
XO-CX5 P99
XO-Agilio P50
XO-Agilio P99
Proxy Throughput
XO-eBPF Throughput
XO-CX5 Throughput
XO-Agilio Throughput
Offload Frequency 1
Offload Frequency 20
Offload Frequency 40
Offload Only Once
Latency [ms]
Throughput [Gbps]
CPU Usage [%]
Worst Case
Best Case
Object Size [KB]
Proxy
Proxy-Server
XO-eBPF-Owner
XO-eBPF-Remote
XO-CX5-Owner
XO-CX5-Remote
XO-Agilio-Owner
XO-Agilio-Remote

Real World Application Integration

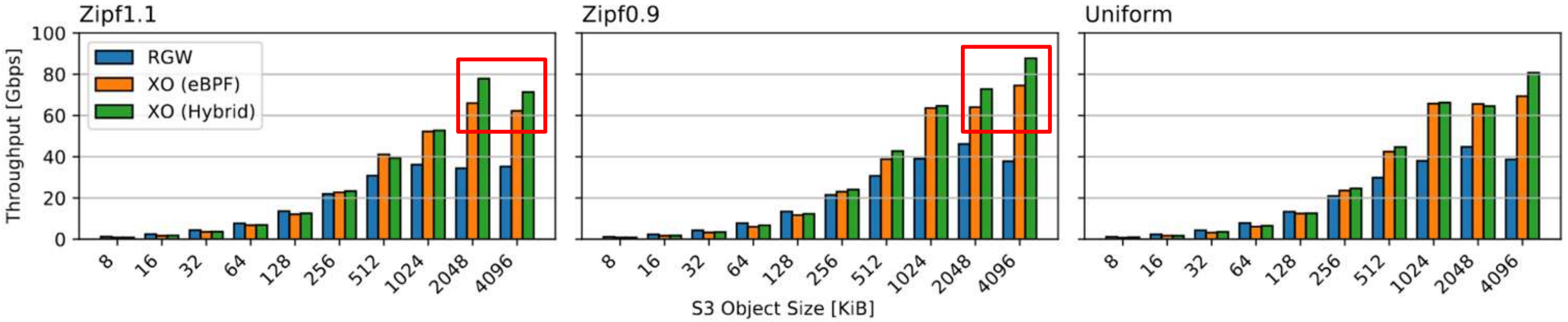
NGINX & Ceph

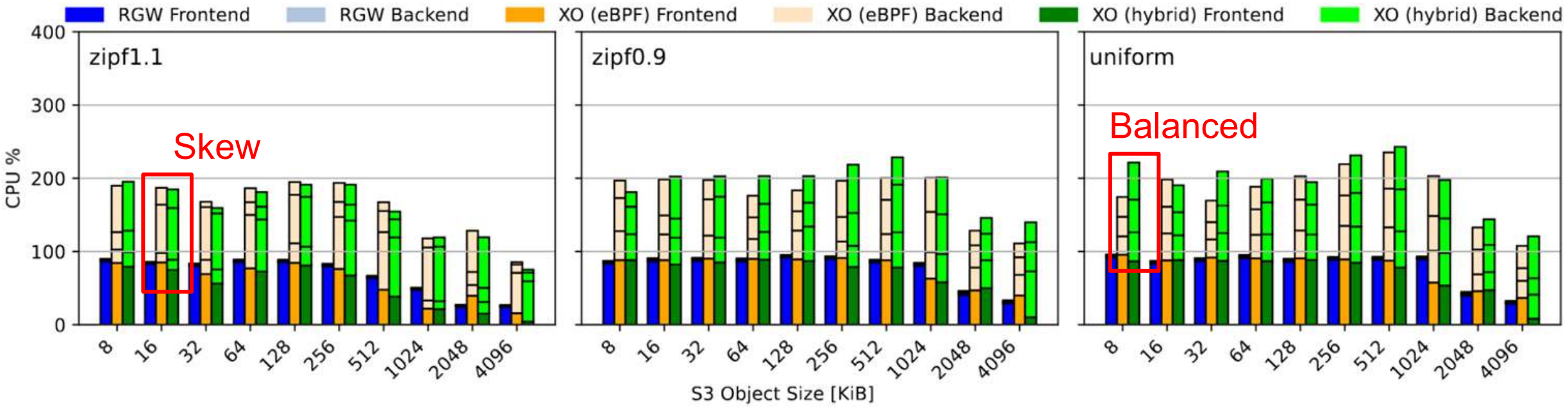
NGINX


NGINX Frontend
NGINX Backends
XO Frontend
XO Backends
NGINX Throughput
XO Throughput
CPU [%]
Gbps
Object Size [KiB]
8
64
128
256
512
1024
2048
4096
0
100
200
300
400
500
0
20
40
60
80
100
Link Speed
Imbalance

Ceph

17% better with hybrid rule insertion





Summary

- **XO**: Combining L4LB efficiency with L7LB flexibility
  - Support both replicated servers (e.g., nginx) and shareded servers (e.g., ceph)
  - Hardware-software hybrid traffic steering using commodity NIC features
  - First connection-migration-based approach integrated with real applications (nginx and Ceph)

Thanks!