# Prism: Proxies without the Pain

Michio Honda

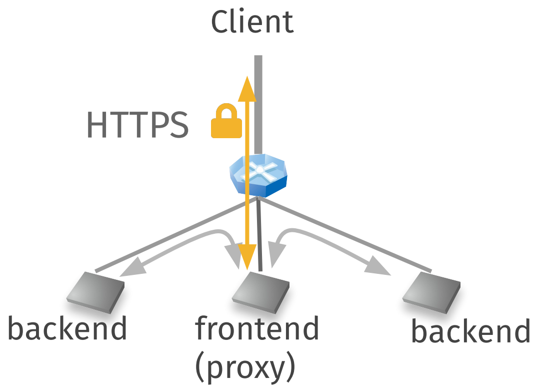
School of Informatics, University of Edinburgh

NGN Webinar, February 12, 2021

Reference: Y. Hayakawa, M. Honda, D. Santry and L. Eggert,
“Proxies without the Pain”, to appear in NSDI’21

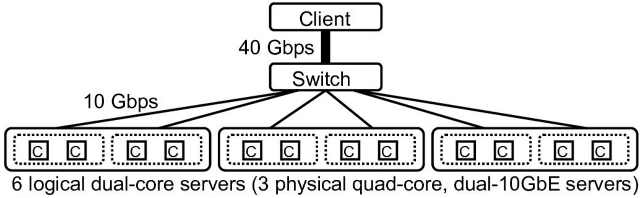
# Background

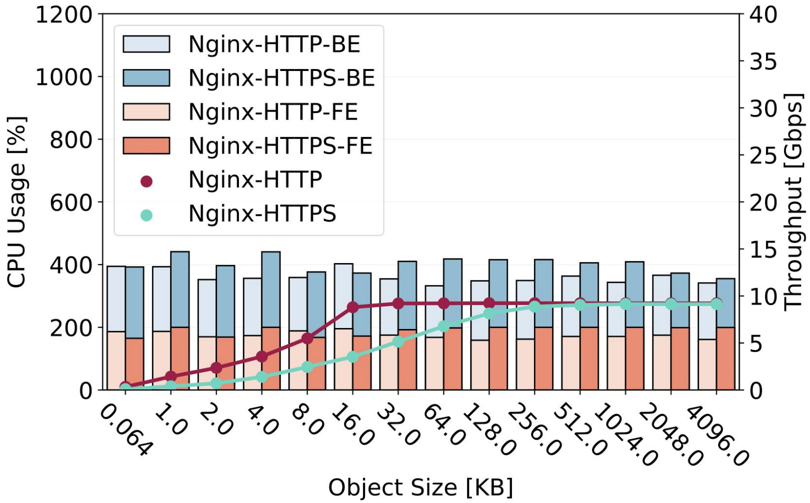
- Object storage (e.g., Amazon S3)
  - Flat namespace (URL)
  - HTTP(S)
- The role of frontend
  - L7 firewall
  - Backend selection
  - TCP/TLS termination



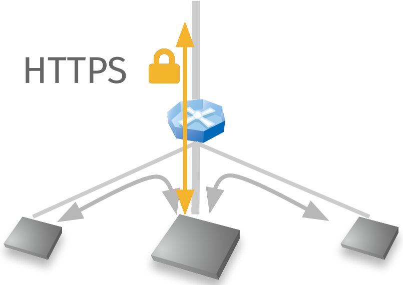
# Problem

- Bottleneck at the frontend
  - Attachment link
  - Encryption (TLS)
- Case study
  - 6-node `nginx` cluster


Client
40 Gbps
Switch
10 Gbps
C C C C C C C C C C C C
6 logical dual-core servers (3 physical quad-core, dual-10GbE servers)

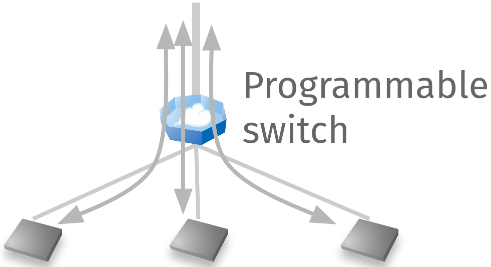


# Design Options



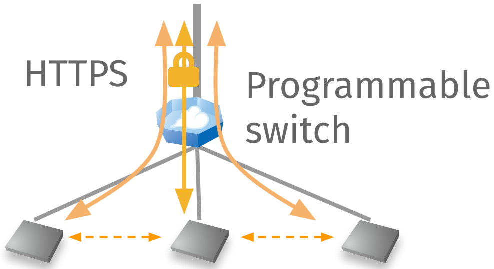
## Scale-up frontend

- Inflexible deployment



## Content-aware routing

- Infeasible for encrypted, multi-packet data
- SwitchKV[NSDI'16], Pegasus[OSDI'20]



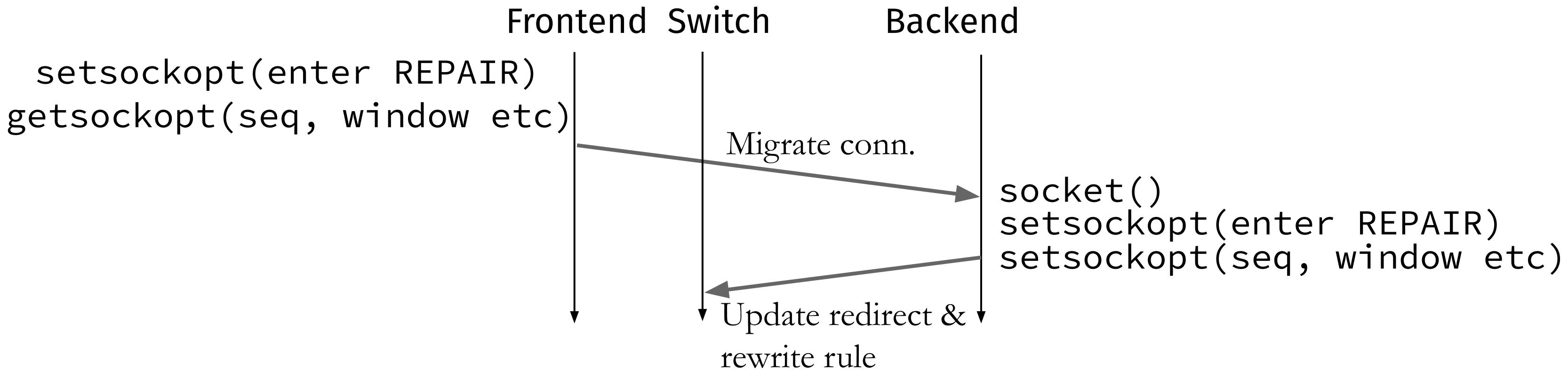
## Connection handoff

- Our choice

# TCP Connection Handoff

- Proposed 20 years ago (LARD[ASPLOS'98, ATC'00])
- Not used or featured since
  - Perhaps not needed
    - Bottleneck at disks
  - Perhaps too complex
    - Need for custom TCP stack and "programmable" switch
- Those circumstances have changed
  - Storage is fast (NVMe, Persistent memory)
  - We have Linux TCP serialization (`REPAIR`)
  - Programmable switches are available

# TCP Handoff in a Nutshell



```
setsockopt(enter REPAIR)
getsockopt(seq, window etc)
```

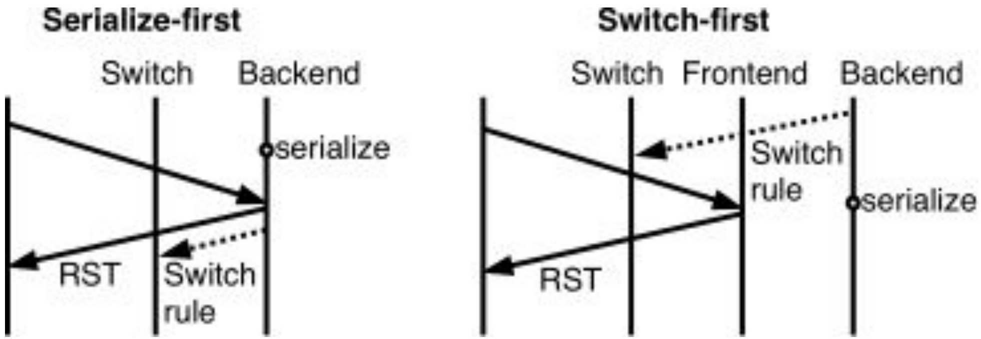
Migrate conn.

```
socket()
setsockopt(enter REPAIR)
setsockopt(seq, window etc)
```

Update redirect & rewrite rule

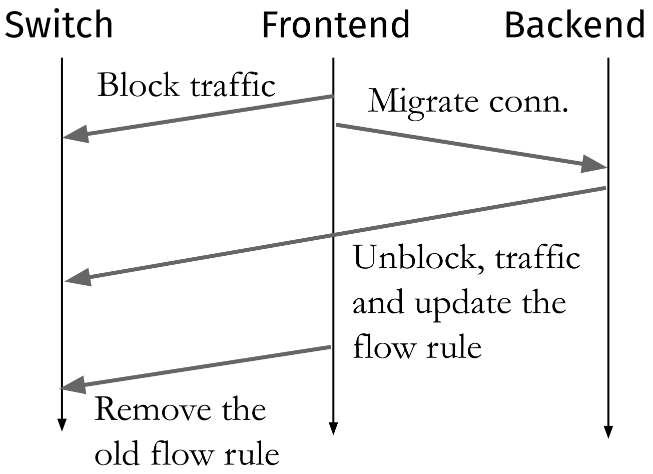
# The Packet Leak Problem

- Any packet arriving during the REPAIR mode resets the connection
  - To avoid ambiguity of connection state transition



Coordinating the switch update and handoff is difficult

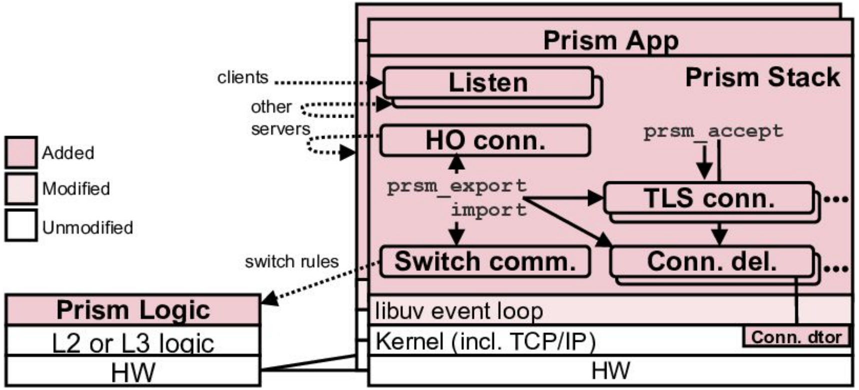
# Two-Phase Handoff Protocol



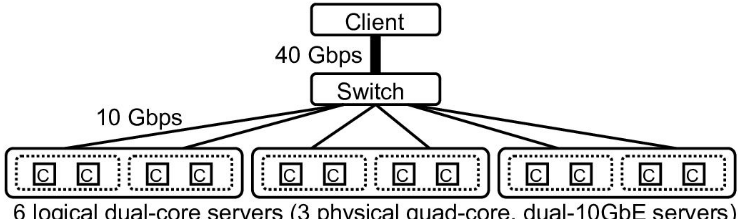
Dropping RSTs using a host firewall is not an option, as we need to manage connections at both the switch and host firewall

# The need for "Stack"

- Spans across the kernel and app
  - The kernel module detects in-kernel connection-state removal event (needed to withdraw the switch rule)
  - No need for kernel modification

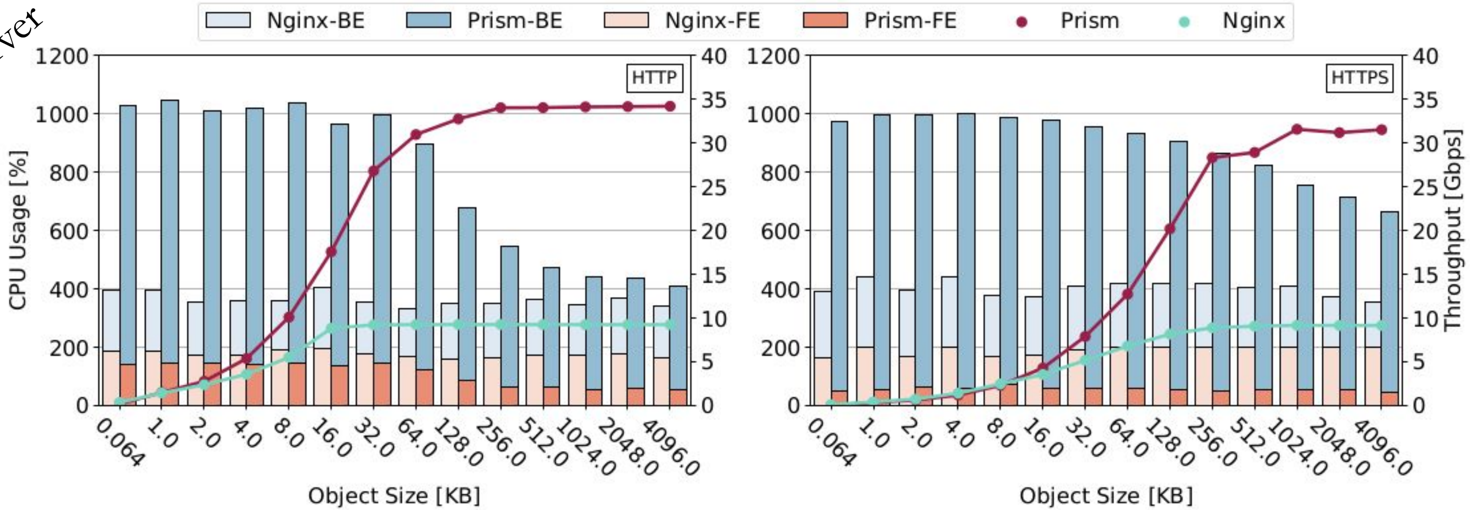


# Performance


Client
40 Gbps
Switch
10 Gbps
C C C C
C C C C
C C C C
6 logical dual-core servers (3 physical quad-core, dual-10GbE servers)

Connection handoff time is 232us

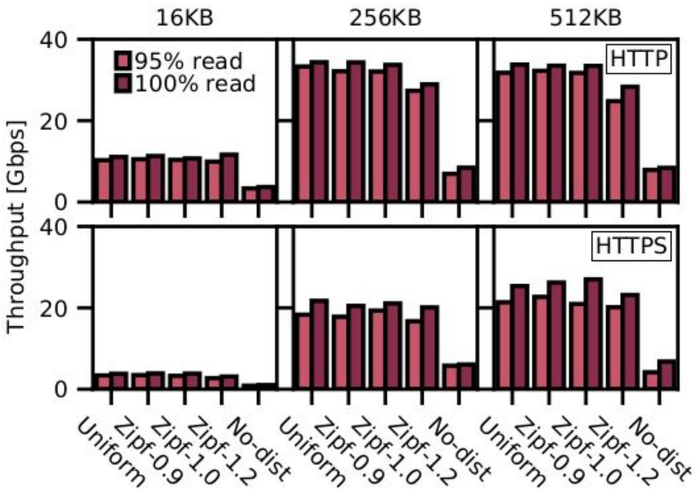
200% per server



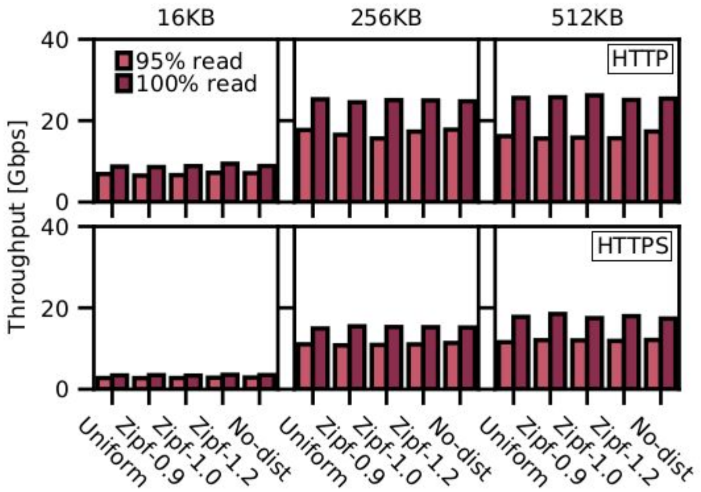
100 persistent TCP connections

# Use Cases

- Prism is useful to implement partitioned or replicated backends



**Partitioned backends**



**Replicated backends**

No-dist means the most "skewed" (all the reqs go to the same backend)

# Summary

- Time for TCP handoff has come
  - Storage is fast
  - TLS is everywhere
  - Programmable switch is available
- We don’t need kernel modification
  - We needed a small one, but we upstreamed it to Linux
- Check out our NSDI’21 paper for more details
  - https://micchie.net/files/prism-nsdi21.pdf