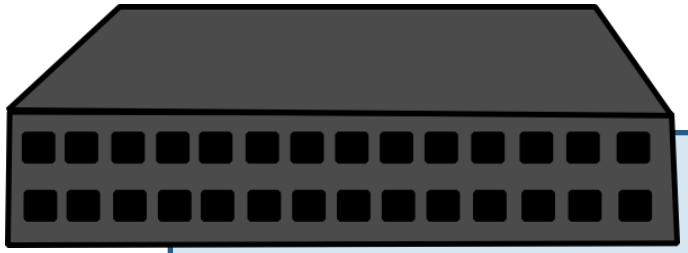
Riley: Simplifying the Data Center Switch

Sean Choi, Changhoon Kim, Robert Soulé, Jongkeun Lee
Milad Sharif, Xin Jin and Nick McKeown

BAREFOOT NETWORKS

Here is how a **switch** is built today.



Control Plane

Data Plane

Switch OS

Packet Processing

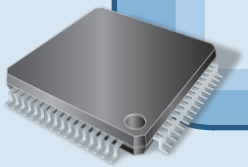
Routing Protocols

Forwarding Table

Runtime API

Registers

Driver



Control Plane

Routing Protocols

Runtime API

Driver

Switch OS

Data Plane

Packet Processing

Forwarding Table

Registers

How will a data center switch look like in **10 years?**

But first…

Let’s take a step back and see how we got here.

Characteristics of Internet and Enterprise Networks

- Unknown and/or unpredictable network topology

  Requires complex routing protocols

- Need to support legacy protocols

  More logic and resources required on a switch

Characteristics of Internet and Enterprise Networks

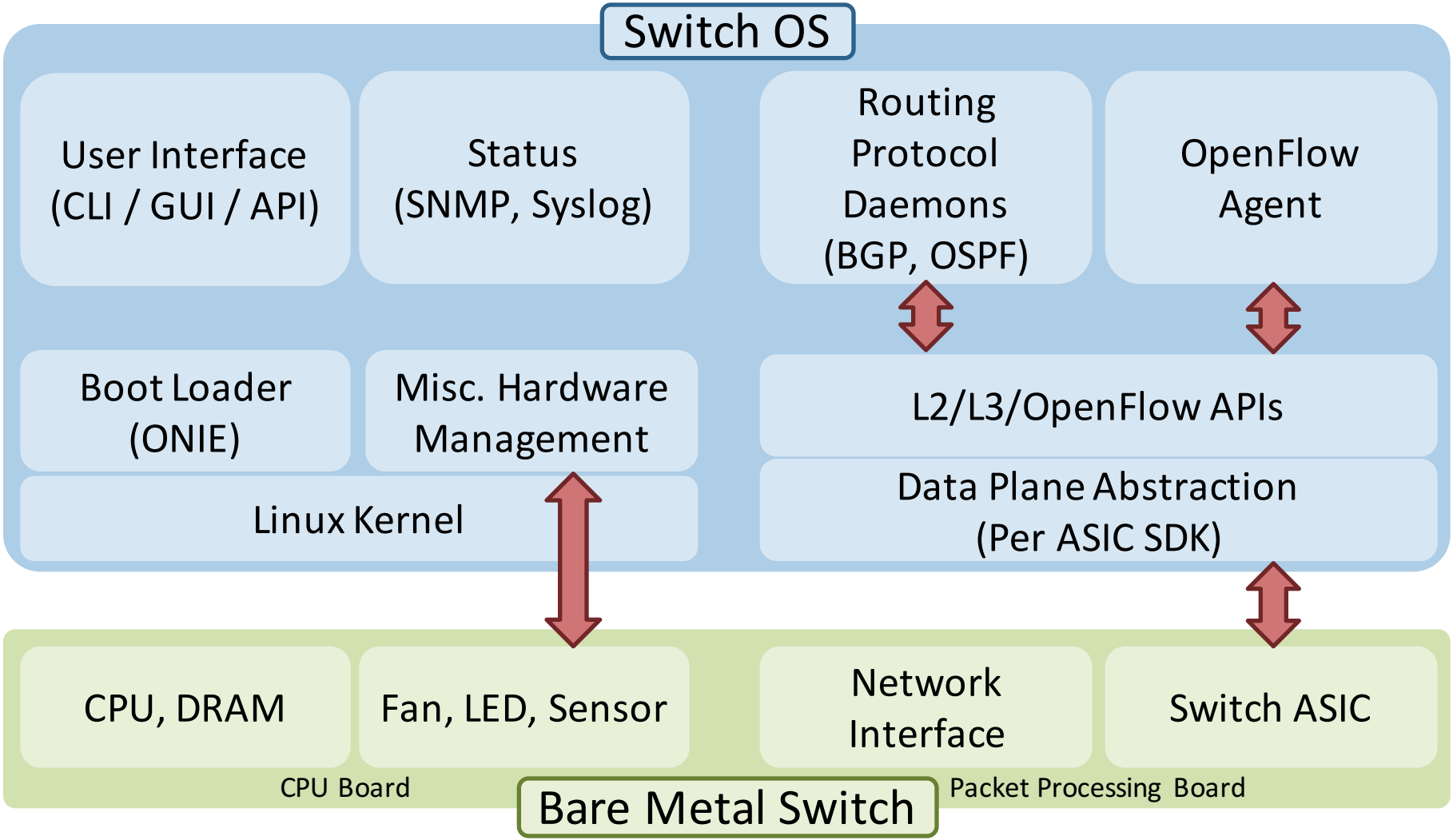
- No trust or control over the end-hosts

  Most states managed within the network

- Data plane could not be changed

  Custom logic must be handled outside the network

=> We need a Switch OS!


Switch OS
User Interface (CLI / GUI / API)
Status (SNMP, Syslog)
Routing Protocol Daemons (BGP, OSPF)
OpenFlow Agent
Boot Loader (ONIE)
Misc. Hardware Management
L2/L3/OpenFlow APIs
Linux Kernel
Data Plane Abstraction (Per ASIC SDK)
CPU, DRAM
Fan, LED, Sensor
Network Interface
Switch ASIC
CPU Board
Packet Processing Board
Bare Metal Switch

Switch OS	Open Source?	Number of Files	Lines of Code
Open Network Linux (ONL)	**Open**	**2129**	139317
Software for Open Networking in the Cloud (SONiC)	**Open**	**1092**	388574
Facebook Open Switching System (FBOSS)	**Open**	**499**	55299
PicOS	**Proprietary**	N/A	N/A
Cisco (IOS, NX-OS, CatOS)	**Proprietary**	N/A	N/A
Arista EOS	**Proprietary**	N/A	N/A

Switch OS	Open Source?	Number of Files	Lines of Code
Open Network Linux (ONL)	Open	2129	139317
Software for Open Networking in the Cloud (SONiC)	Open	1092	388574
Facebook Open Switching System (FBOSS)	Open	499	55299
PicOS	Proprietary	N/A	N/A
Cisco (IOS, NX-OS, CatOS)	Proprietary	N/A	N/A
Arista EOS	Proprietary	N/A	N/A

50+% of network failures happen in the control plane![1]

[1] The Zero Touch Network, Bikash Koley, CSNM 2016

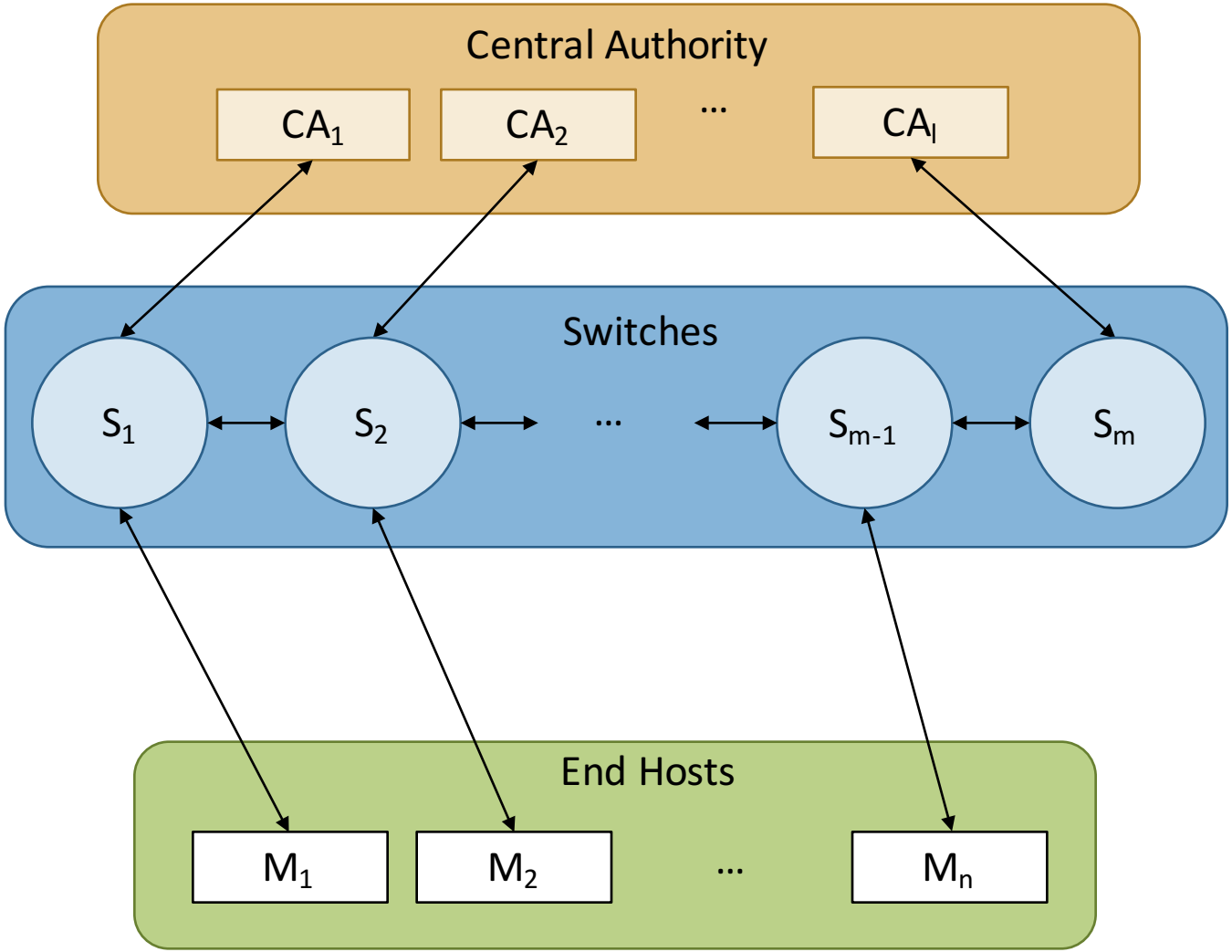
Switch	CPU	Ram	SSD
Barefoot Wedge 100	4 Core Intel Xeon-D	8GB DDR4	128GB M.2
Broadcom Open 1.0	2 Core AMD G-Series	2GB SDRAM	M.2
Mellanox MSX1710-OCP	Intel Ivy Bridge	8GB DDR3	128GB mSATA
SK Telecom CNA-SSX2RC	6 Core Intel Haswell	8GB DDR4	64GB MLC
Part Cost	$400~600	$50~100	$50~100
Power Consumption	100~500W	1~10W	1~10W

Switch	CPU	Ram	SSD
Barefoot Wedge 100	4 Core Intel Xeon-D	8GB DDR4	128GB M.2
Broadcom Open 1.0	2 Core AMD G-Series	2GB SDRAM	M.2
Mellanox MSX1710-OCP	Intel Ivy Bridge	8GB DDR3	128GB mSATA
SK Telecom [illegible]	[illegible] Intel Haswell	[illegible]	[illegible] GB MLC
Part Cost	$400~600	$50~100	$50~100
Power Consumption	100~500W	1~10W	1~10W

Expensive at Data Center Scale!

How to design the **simplest** data center switch?

How is a data center network **different**?


Central Authority
CA1
CA2
...
CAl
Switches
S1
S2
...
Sm-1
Sm
End Hosts
M1
M2
...
Mn

Characteristics of Data Center Network

- Existence of a central authority

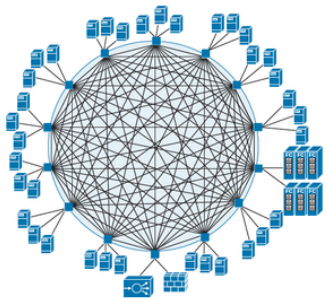
  Complex logic can be offloaded

- Known and generated network topology

  Can use simpler routing protocols

- Supports small unified set of features

  Less resources required on a switch



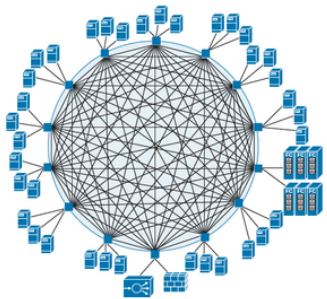
Characteristics of Data Center Network

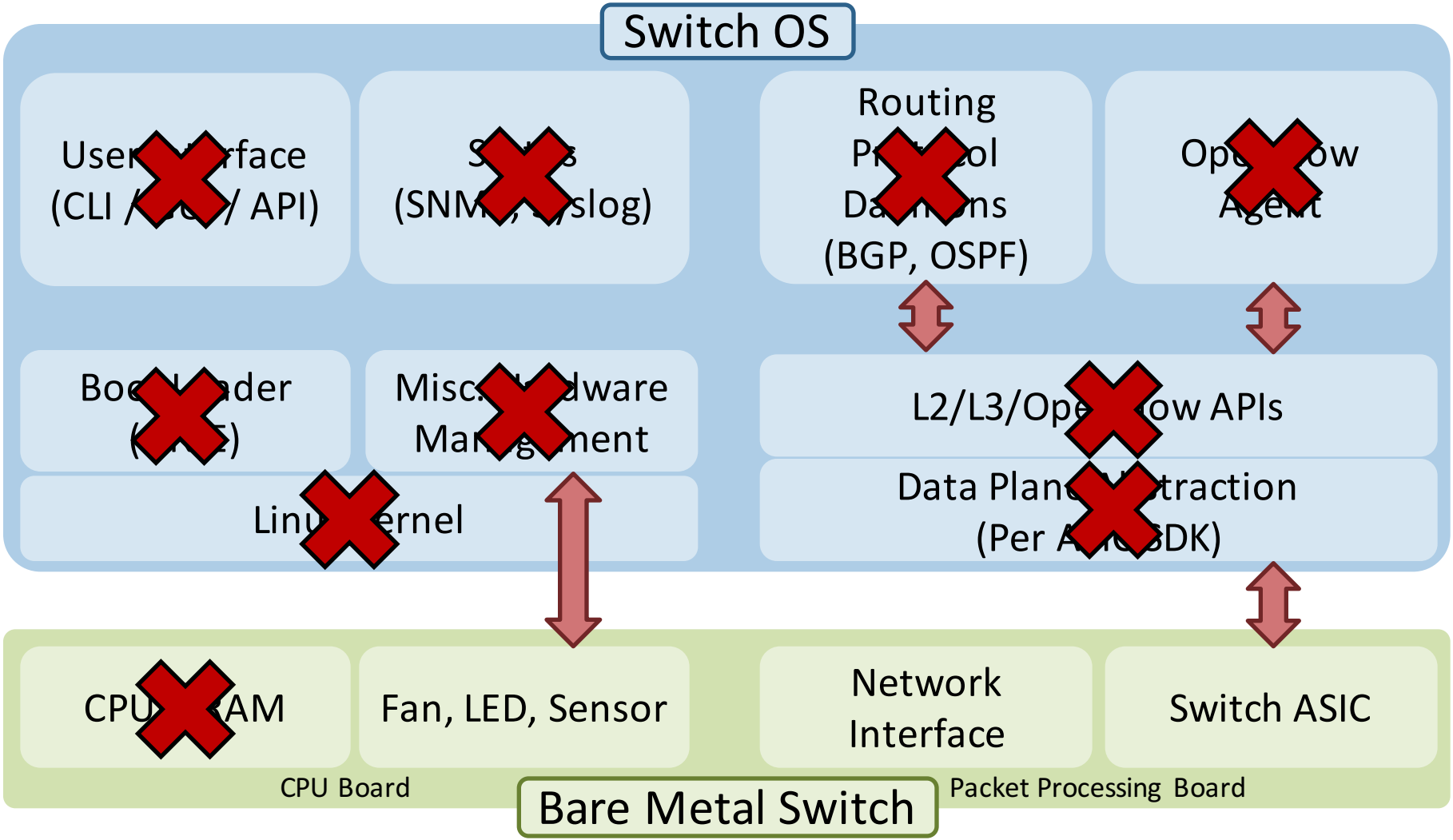
- More control over the end-hosts

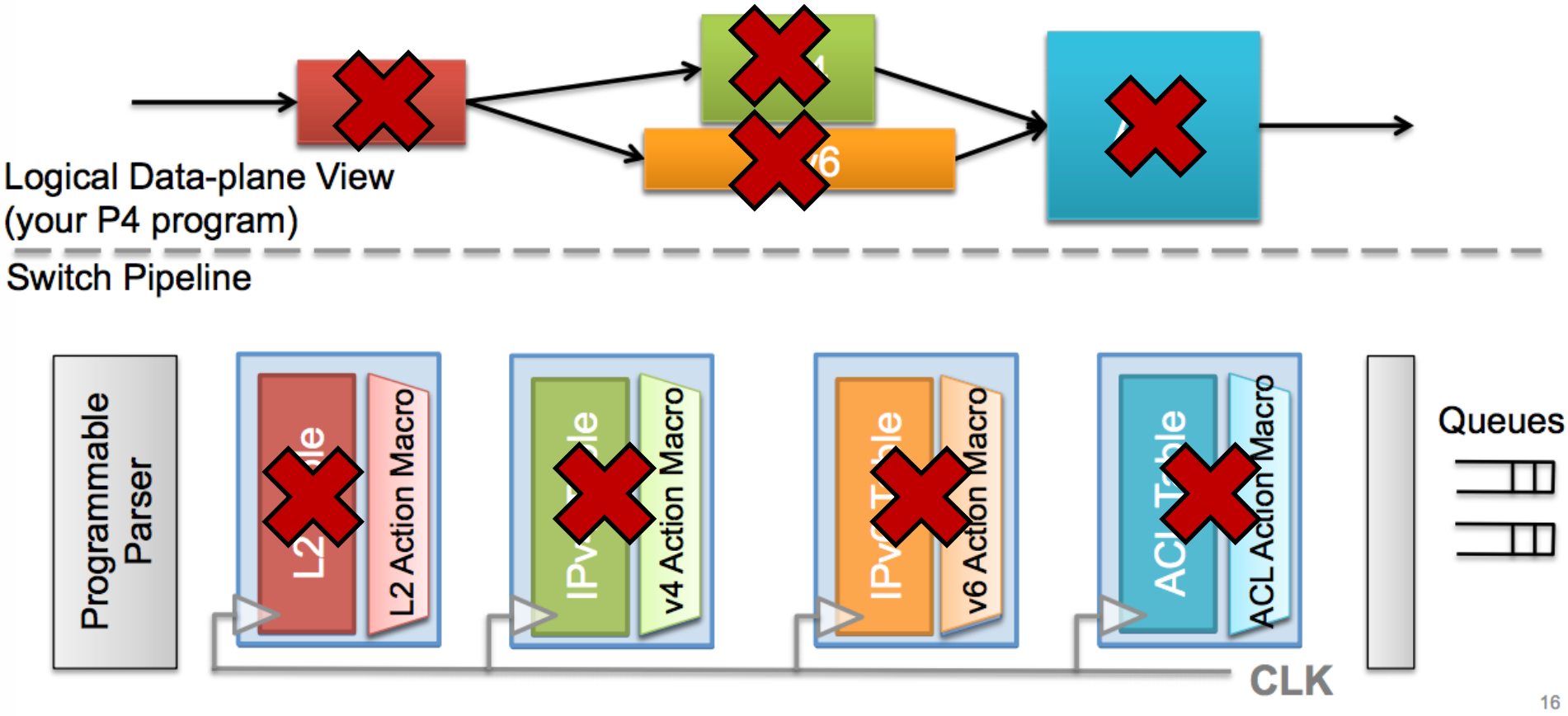
  Less states manage within the network

- Data plane can be customized

  Can directly communicate with the data plane




Switch OS
Routing
(BGP, OSPF)
L2/L3/
APIs
Linux
Fan, LED, Sensor
Network
Interface
Switch ASIC
CPU Board
Packet Processing Board
Bare Metal Switch

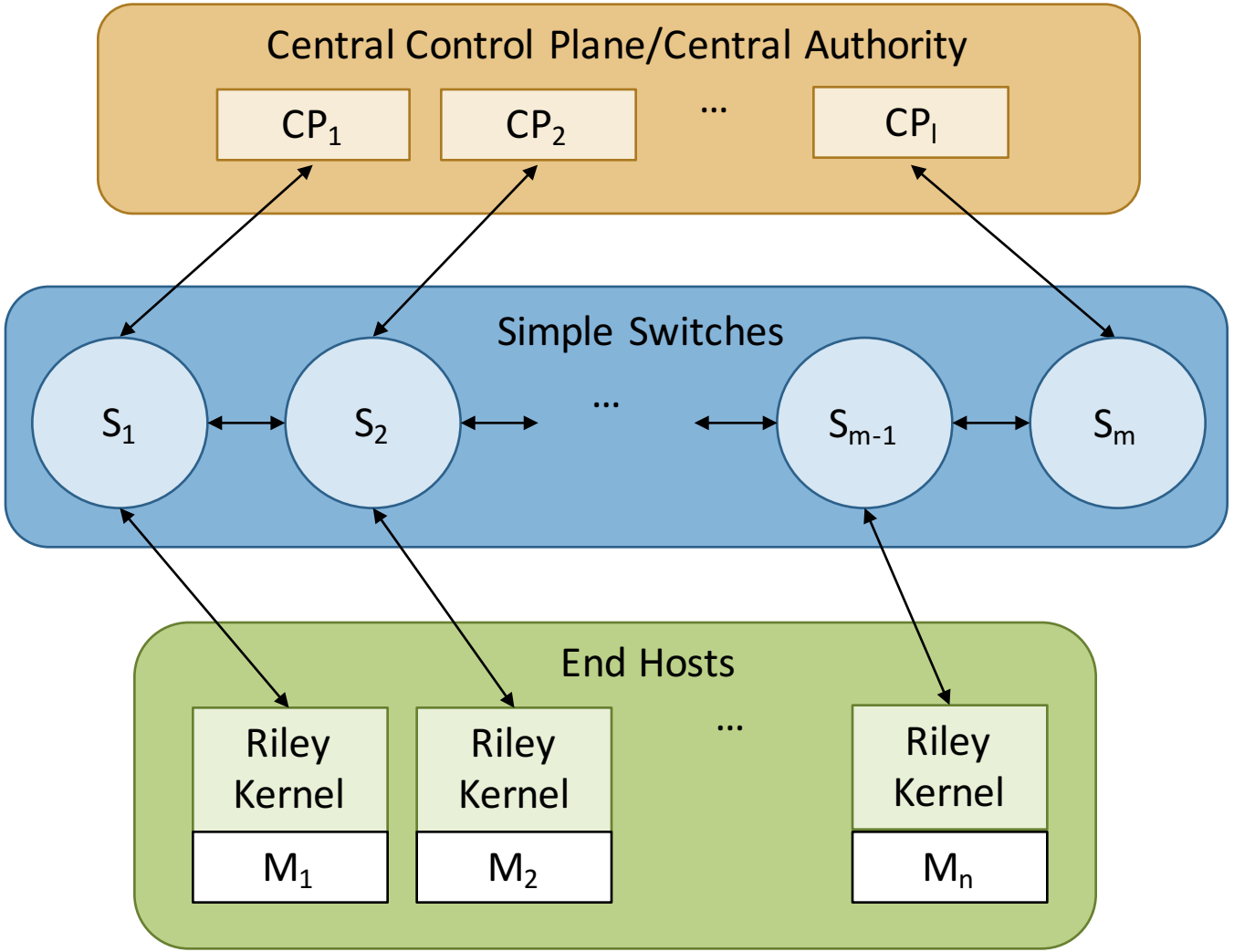

Logical Data-plane View
(your P4 program)
Switch Pipeline
Programmable Parser
L2 Action Macro
v4 Action Macro
v6 Action Macro
ACL Action Macro
Queues
CLK

16

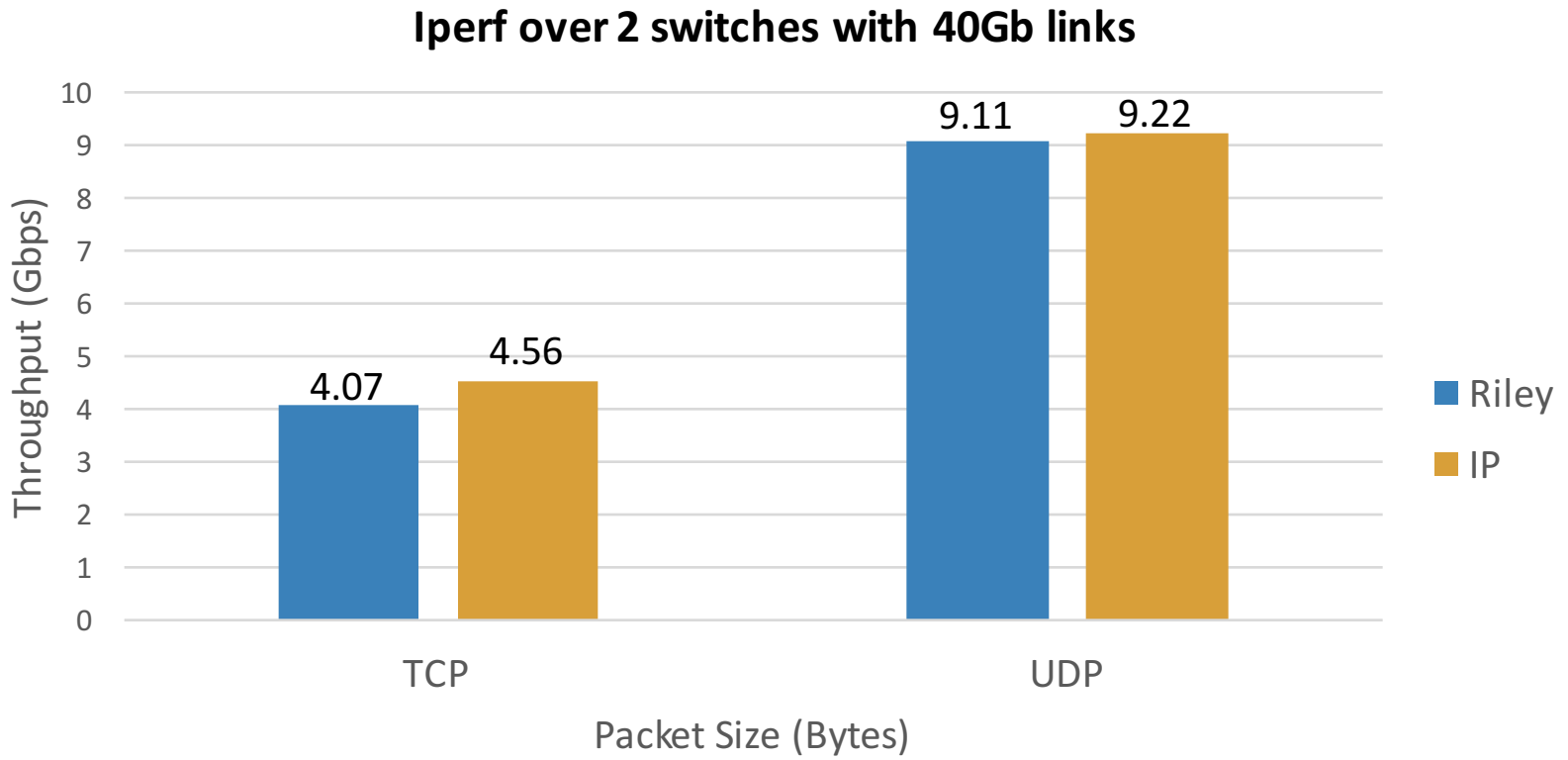
How to design the **simplest** data center switch?

Why do we need a Switch OS?

Riley is a network design using extremely simple switches with **NO** Switch OS


Central Control Plane/Central Authority
CP1
CP2
...
CPl
Simple Switches
S1
S2
...
Sm-1
Sm
End Hosts
Riley Kernel
M1
Riley Kernel
M2
...
Riley Kernel
Mn

Comparable Throughput Performance



Comparable Real-World Job Completion Times

Job Type	Completion Time Riley	Completion Time IP
Small File Transfer 72 MB	1 seconds	1 seconds
Large File Transfer 1031 MB	8 seconds	8 seconds
Spark PageRank (20 Iterations, ~4M Nodes, ~69M Edges)	11 minutes	11 minutes
Spark Logistic Regression (10000 Partitions)	38 seconds	40 seconds
Spark Pi Computation (10000 Points)	4.6 minutes	4.5 minutes

Comparable End-host Resource Usage

Protocol Type	CPU Userspace	CPU Kernel	Memory Usage
Riley	8.4	75.5	498.1MB
IP	8.6	71.3	495.3MB

Significant Less Switch Resource Usage

Protocol Type	TCAM	SRAM	MAU	CPU	Ram (25000 Forwarding Entries)
Riley	1	1	1	0%	0
IP	163.93	19.69	2.40	7%	392 MB

A data center network design with
simple switches,
comparable performance, and
significantly less switch resource usage

BAREFOOT NETWORKS