Lesson 13: Software-defined networks

Van-Linh Nguyen

Fall 2024

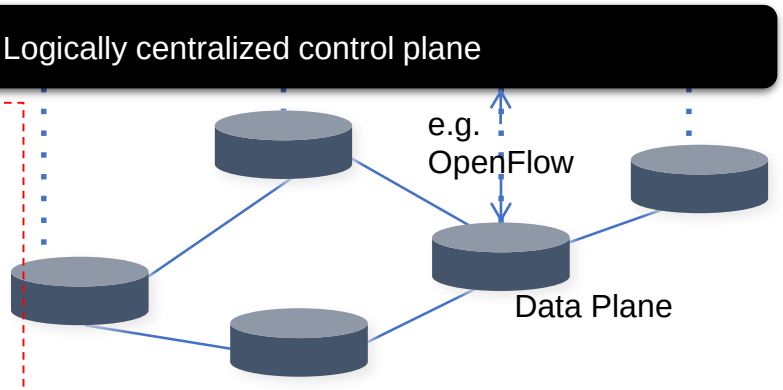
國立中正大學
Cybersecurity Lab

Outline

- Software defined networking
  1. Conventional networks vs SDN
  2. SDN labs

What is SDN?

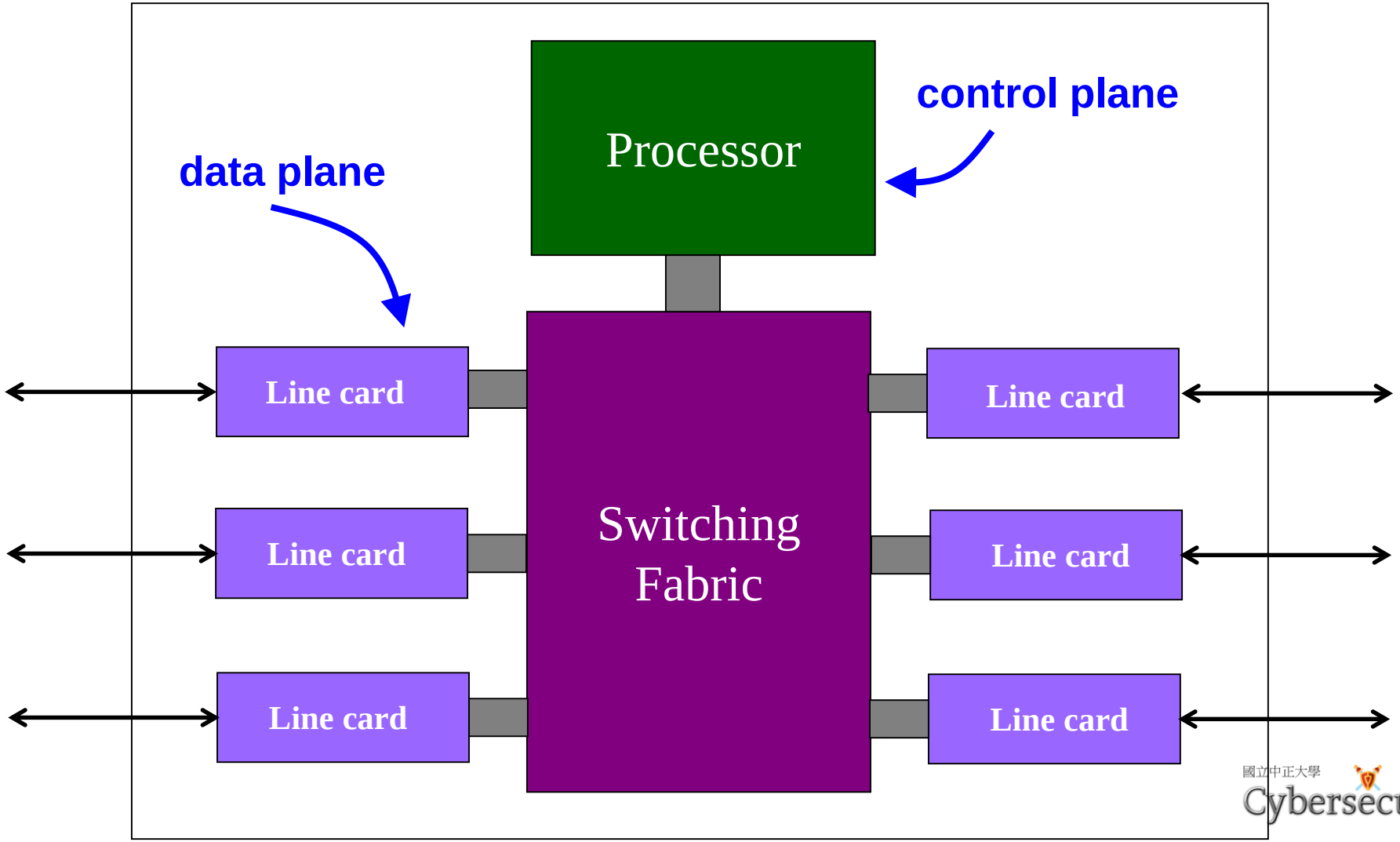
- Definition:
  - SDN (software-defined networking) is the separation of control and data planes
  - The separation allows control topology to be independent of physical network topology
- The more interesting question is:
  - Why would anyone want to do this?
  - That question has a lot of answers…



E.g., developers for software control and physical devices can be independent from each other that can encourage the faster innovation and more players than the traditional model → reduce price → ISPs like AT&T like this model

For example, start ups/ISP can join to develop routing/network control functions while the manufacturers like TSMC, Asus, can focus on producing high-performance hardware devices

Data and Control Planes



Outline

- History of SDN
  - Challenges faced by IP networks
- SDN architecture
- Case Studies:
  - Network Virtualization
  - Traffic Engineering
  - SD-WAN
  - Bare metal switching

A Revolution in Networking

MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2009

NEWS // WEB

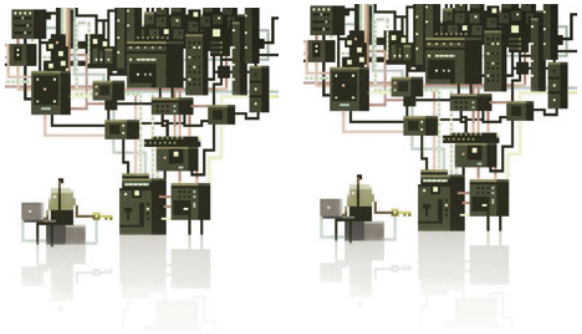
TR10: Software-Defined Networking

Nick McKeown believes that remotely controlling network hardware with software can bring the Internet up to speed.

4 comments

KATE GREENE

Tuesday, February 24, 2009



Superbrothers

For years, computer scientists have dreamed up ways to improve networks' speed, reliability, energy efficiency, and security. But their schemes have generally remained lab projects, because it's been impossible to test them on a large enough scale to see if they'd work: the routers and switches at the core of the Internet are locked down, their software the intellectual property of companies such as Cisco and Hewlett-Packard.

Frustrated by this inability to fiddle with Internet routing in the real world, Stanford computer scientist Nick McKeown and colleagues developed a standard called OpenFlow that essentially opens up the Internet to researchers, allowing them to define data flows using software–a sort of "software-defined networking." Installing a small piece of OpenFlow firmware (software embedded in hardware) gives engineers access to flow tables, rules that tell switches and routers how to direct network traffic. Yet it protects the proprietary routing instructions that differentiate one company's hardware from another.

Foundations of SDN

- 4D, Greenberg et al. – part of a broader set of “Clean Slate” initiatives
- Ipsilon General Switch Management Protocol – RFC 2297 (1996)
- IETF Forces WG (2001-2015!!)
- Ethane (2007)

Challenges with IP networks

- Lack of abstractions
- Inability to express intent
- Unpredictable outcome from complex distributed algorithms
- Interactions among protocols (e.g. IGP & EGP)
- Can’t manage a device unless it’s properly configured
  - bootstrap issue – control & management plane dependent on correct data plane
  - Fragility, risk of change
- Glacial pace of innovation

Evolution of network provisioning: 1996-2016

1996

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

Terminal Protocol: **Telnet**

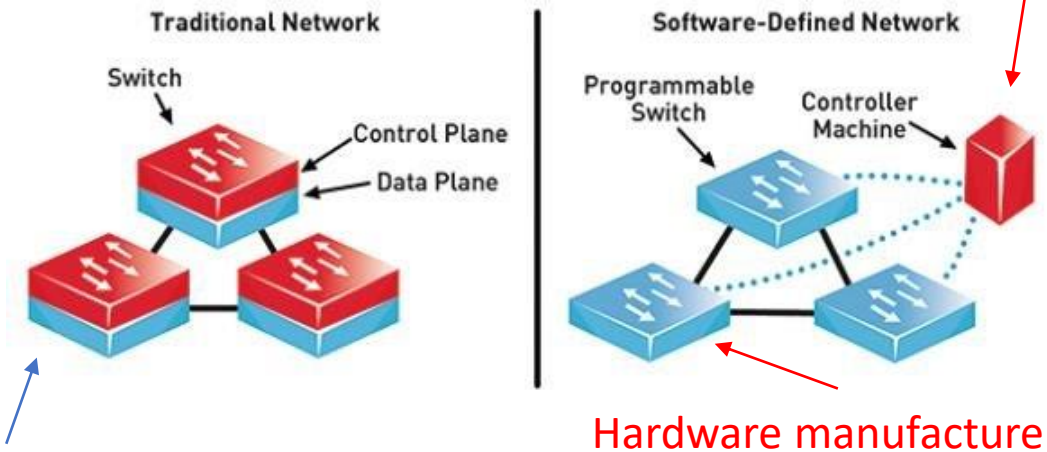
2016

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

Terminal Protocol: **SSH**

Key SDN Insights

- Centralizing the control plane enables more powerful abstractions
  - E.g. X and Y should be able to communicate
  - Express intent network-wide
- Distributed systems techniques to make central control scalable and fault tolerant
- Central control means a single API for the network, rather than an API per box
- Networks provisioned by software, not humans
- Disaggregation → innovation
- Network-wide intent → better security



Software developers

Hardware manufacturers

Cisco/Juniper/Huawei

Disaggregation of computing Industry

App App App App App App

Open Interface

Windows or Linux or Mac OS

Open Interface

Microprocessors

Disaggregation of networking Industry

App App App App App App

Open Interface

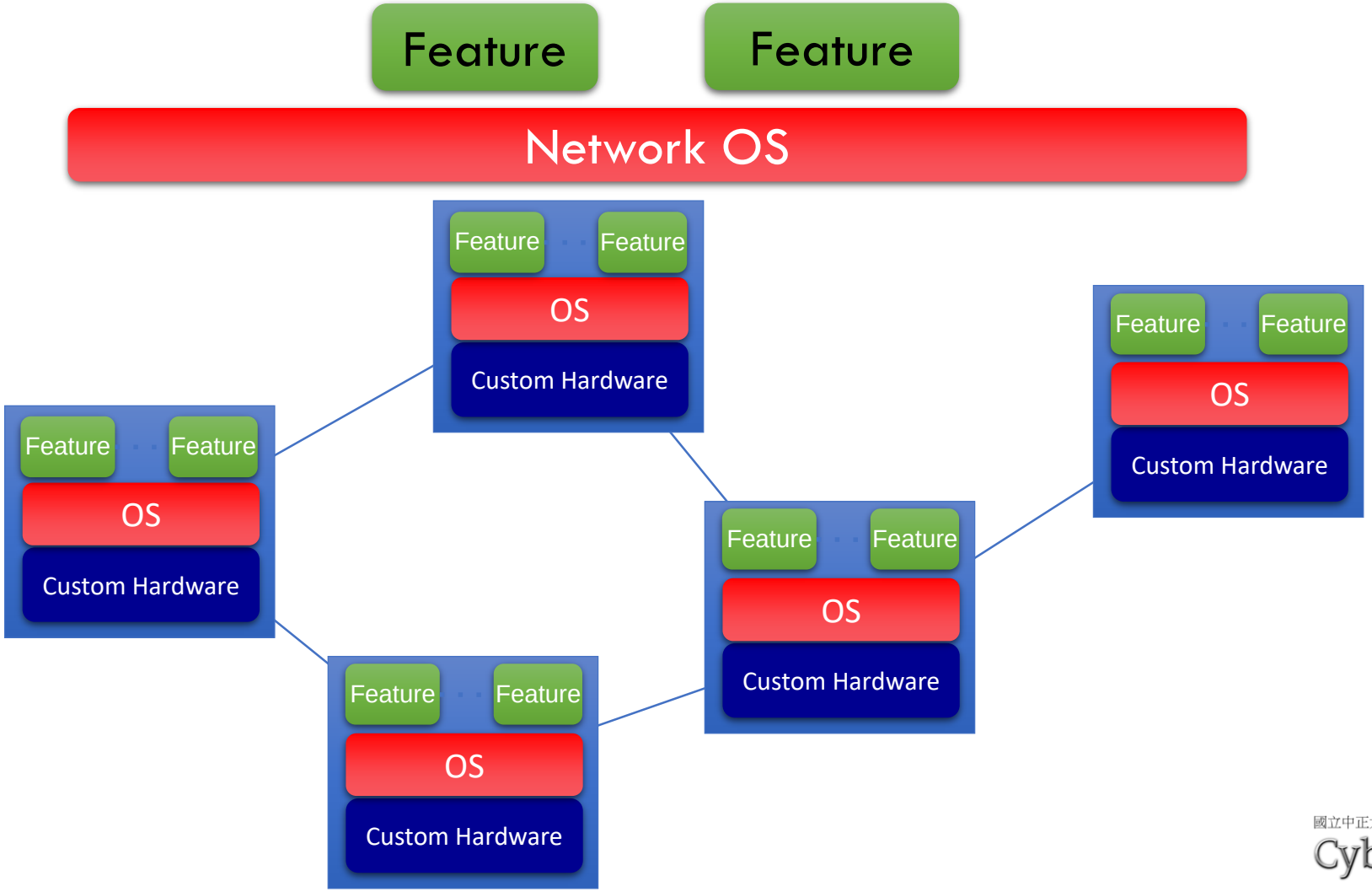
Network OS or Network OS or Network OS

Open Interface

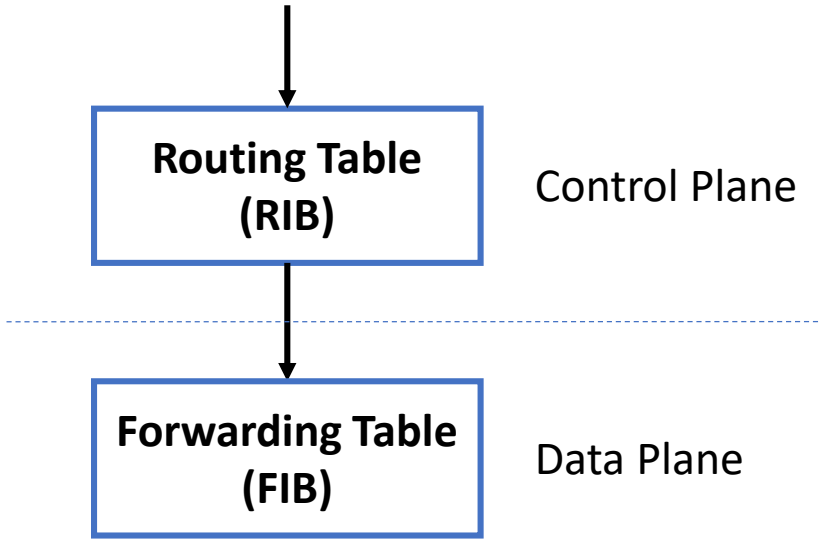
Merchant Silicon Switching Chips

國立中正大學 Cybersecurity Lab

The network is changing



Traditional Control and Data Planes



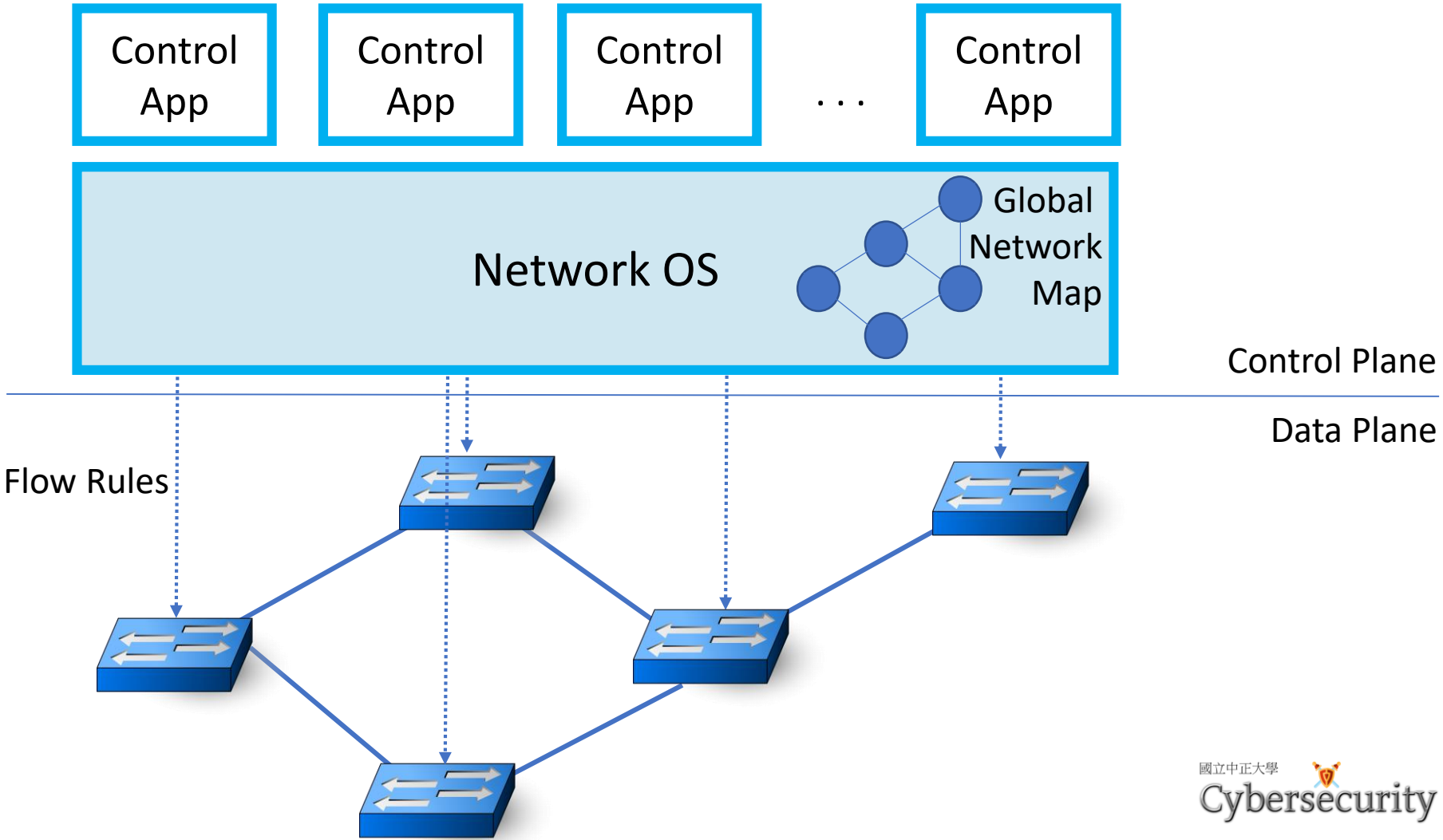
Control Plane

- Protocols: BGP, OSPF, RIP
- RIB: Collection of Link/Path Attributes
- Northbound *Configuration* Interface
  - e.g., Cisco CLI

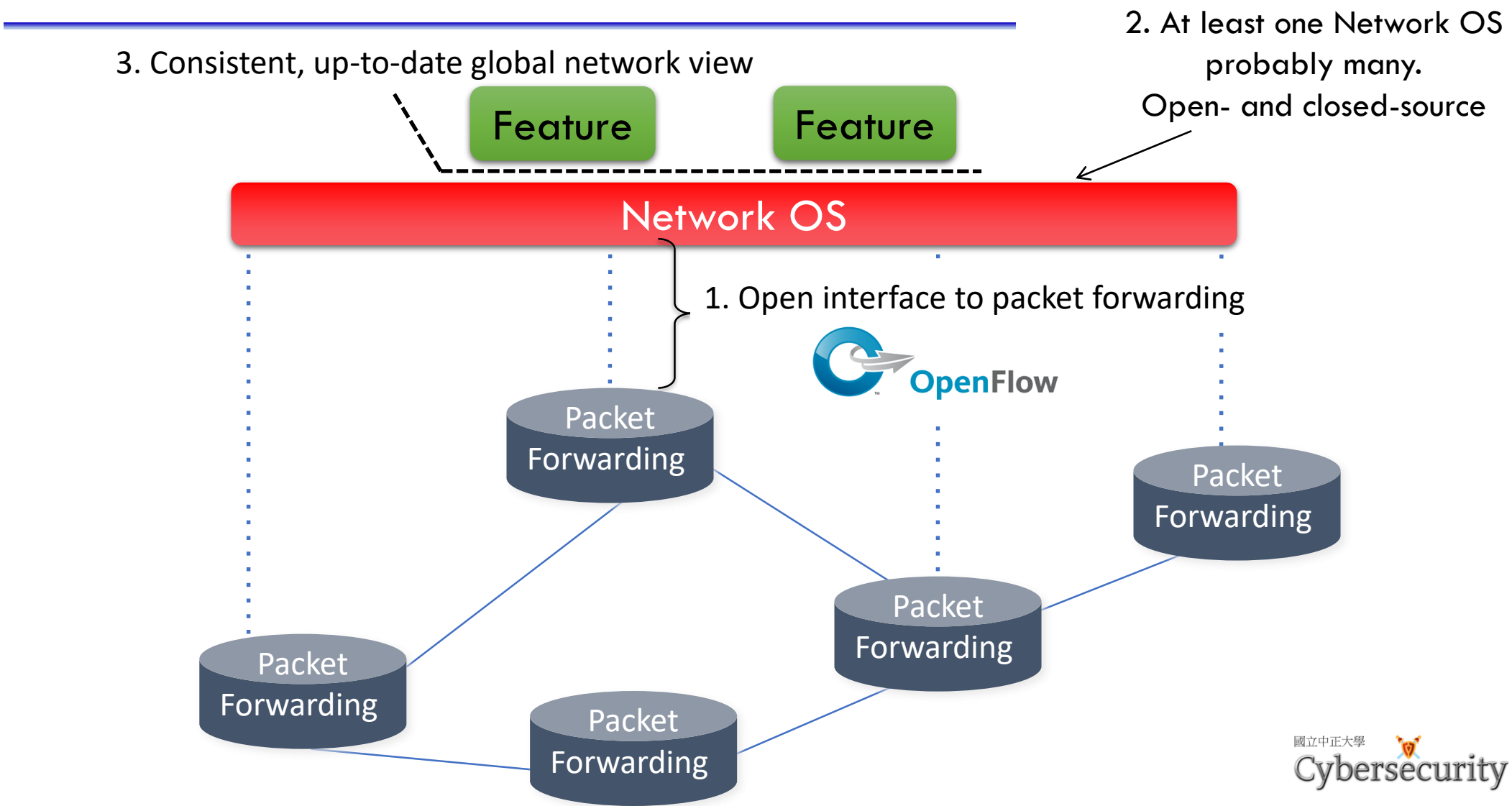
Data Plane

- Protocols: IP
- FIB: Optimized for Fast Lookup
- Northbound *Control* Interface
  - Historically Private/Internal

SDN Control and Data Planes



Software Defined Network (SDN)



Network OS

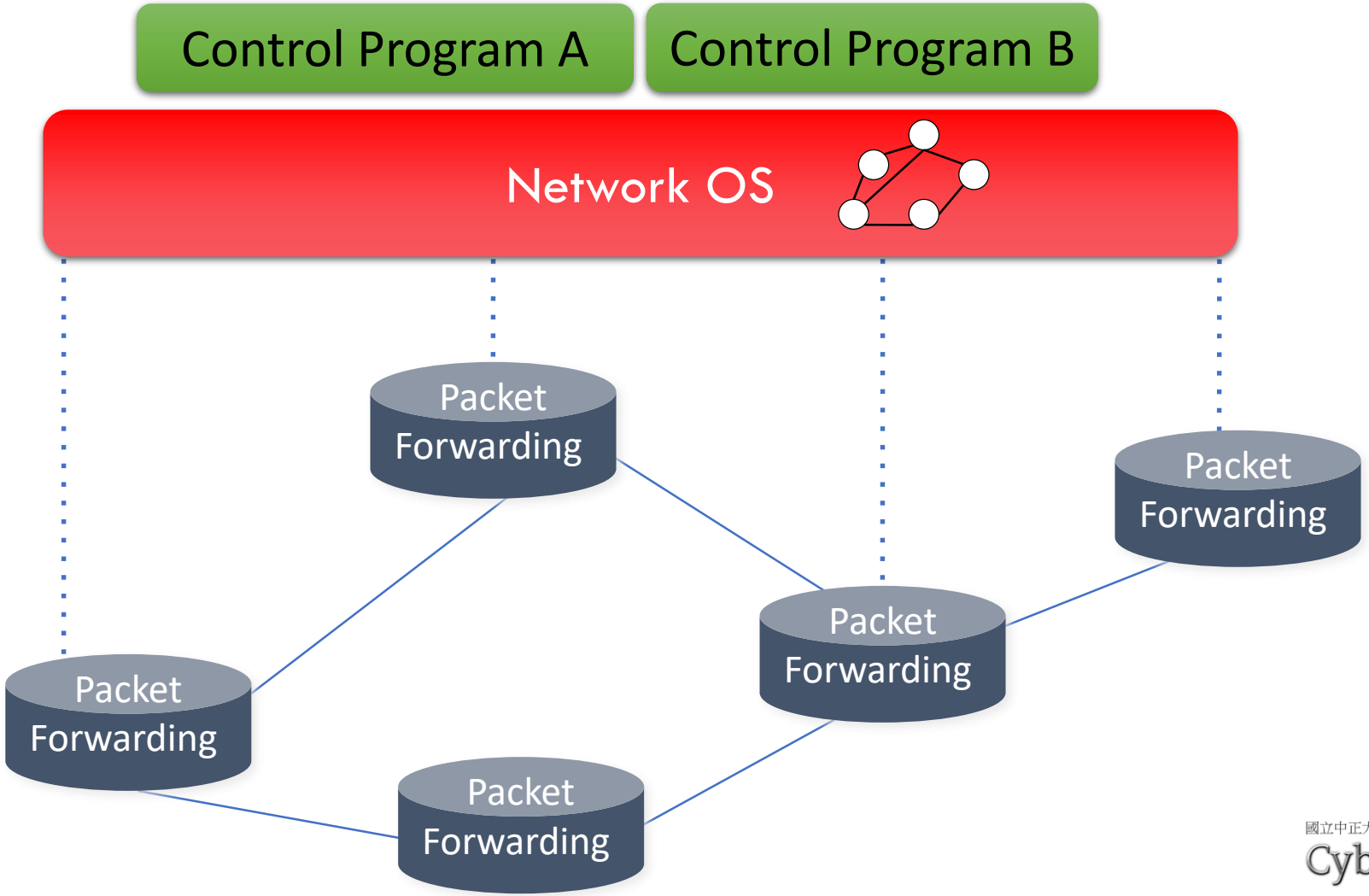
Network OS: distributed system that creates a consistent, up-to-date network view

- Runs on servers (controllers) in the network
- NOX, ONIX, Trema, Beacon, Maestro, … + more

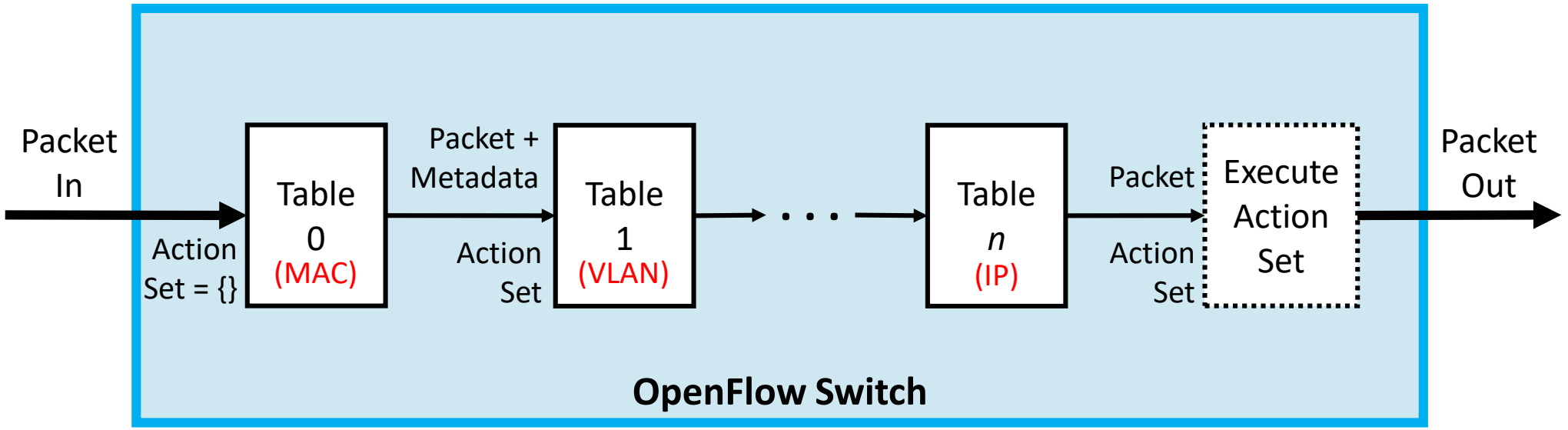
Uses forwarding abstraction to:

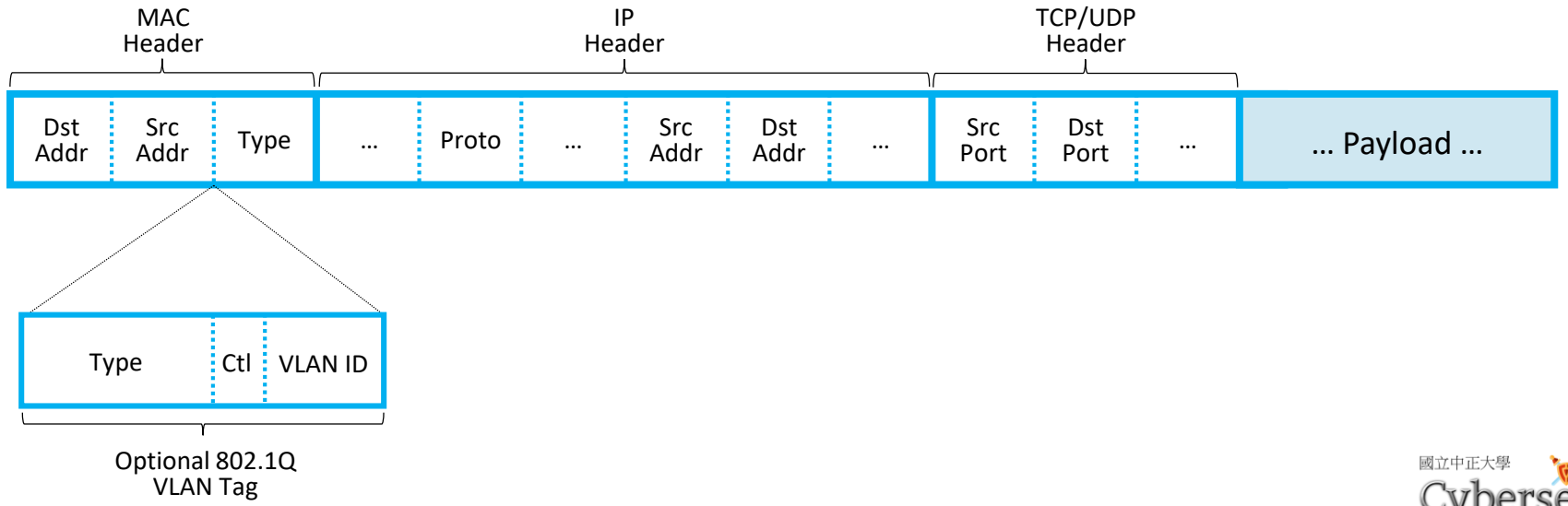
- Get state information **from** forwarding elements
- Give control directives **to** forwarding elements

Software Defined Network (SDN)

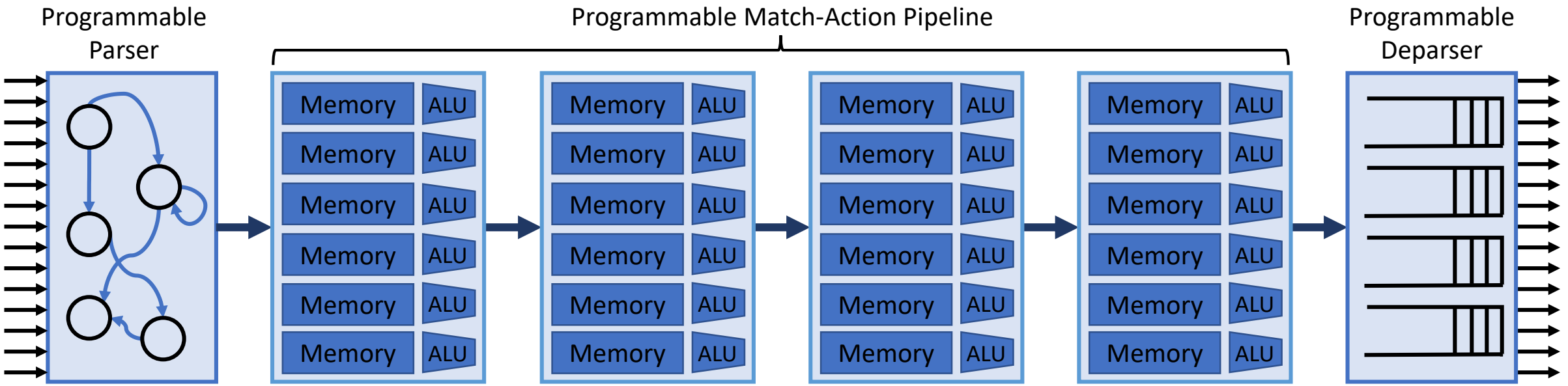


OpenFlow-style data plane

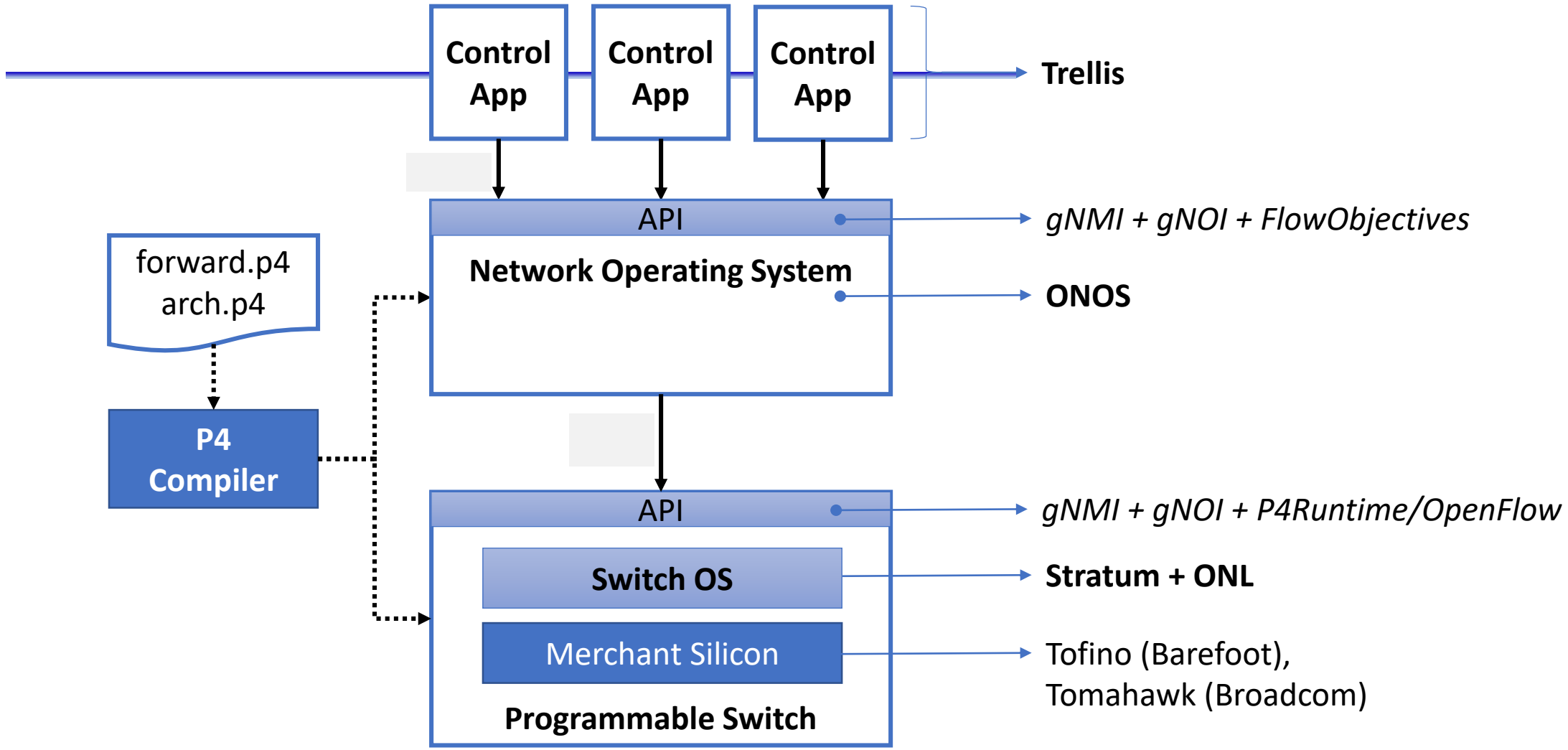




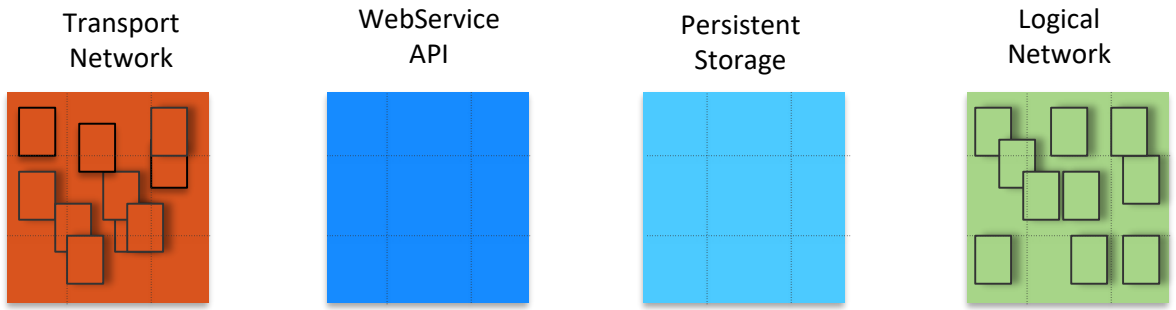
PISA: Protocol Independent Switching Architecture

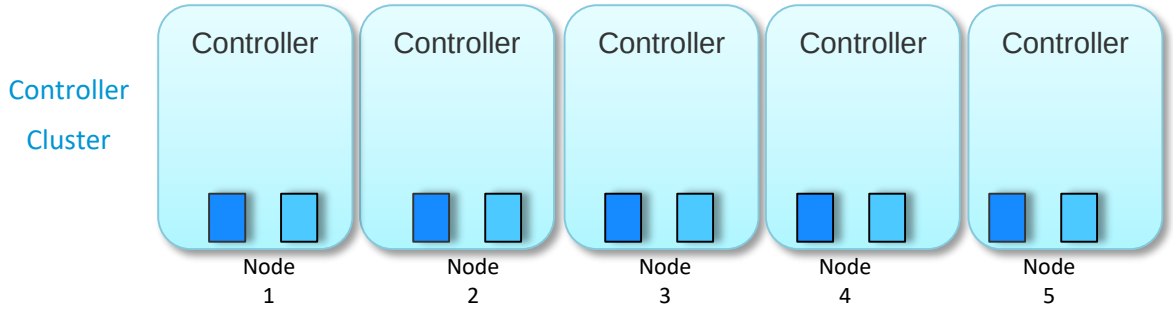


SDN Software Stack



Scaling the Central Control Plane





Summary

Definition of SDN

A network in which the control plane is physically separate from the forwarding plane, and a single control plane controls several forwarding devices. – *Nick McKeown (2013)*

Dimensions

- Disaggregated Control and Data planes
- Centralized vs Decentralized Control Plane
- Fixed-Function vs Programmable Data Plane

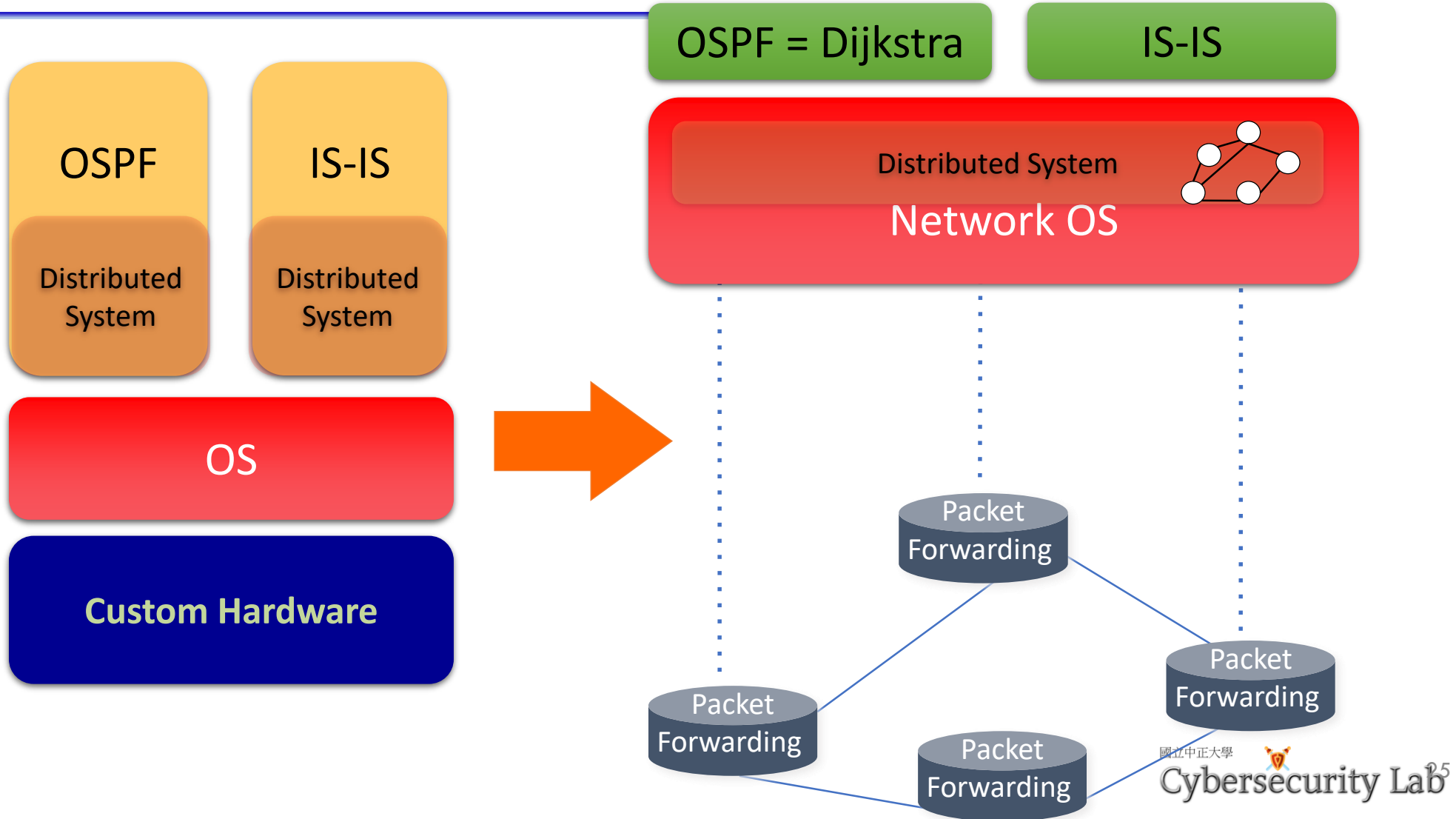
Phases of SDN

- Phase 1: Network operators took ownership of the control plane.
- Phase 1a: Non-traditional entrants to the networking business (via disaggregation)
- Phase 2: Network operators are taking ownership of the data plane.

Example 1: OSPF and Dijkstra

- OSPF
  - RFC 2328: **245 pages**
- Distributed System
  - Builds consistent, up-to-date map of the network: **101 pages**
- Dijkstra's Algorithm
  - Operates on map: **4 pages**

Example


OSPF
IS-IS
Distributed System
Distributed System
OS
Custom Hardware
OSPF = Dijkstra
IS-IS
Distributed System
Network OS
Packet Forwarding
Packet Forwarding
Packet Forwarding
Packet Forwarding