CPSC 416: Distributed Systems

Software Defined Networking

Fabian Ruffy and Nodir Kodirov

Slides are based on M. Alizadeh's lecture on SDN at MIT
people.csail.mit.edu/alizadeh/courses/6.888/slides/lecture14.pdf

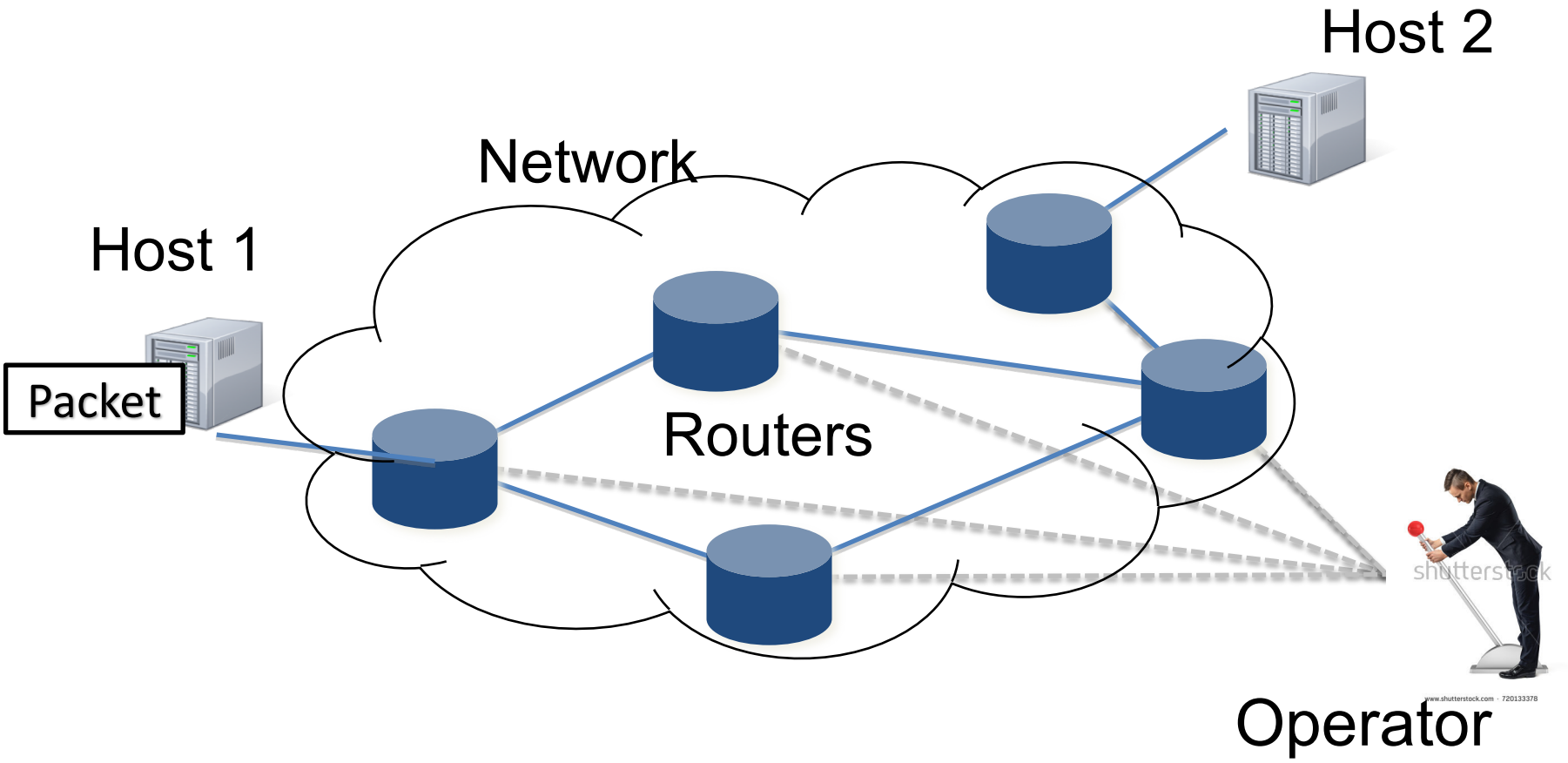
Context Switch

Let’s talk about another distributed system

Networks

Refresher

This is a network:



Two Important Types of Networks

Wide Area Network

- High latency, extremely high throughput
- Geographically dispersed
- Operator is constrained
  - negotiates with other autonomous systems
  - has to deal with adversarial peers

Datacenter Network

- Low latency, high bandwidth,
- All elements are in close proximity
- Operator has a full control (single AS)

How are networks managed?

Outline

(Traditional) Network Planes

Software Defined Networking

Fabian Ruffy

OpenFlow basics

Network virtualization

Nodir Kodirov

The “Planes” of the Network

Data plane: processes packets based on local forwarding table

- Forwarding state + packet header → forwarding decision
- Filtering, buffering, scheduling

Control plane: fills the forwarding table in routers

- Determines how and where packets are forwarded
- Routing, traffic engineering, failure detection/recovery, …

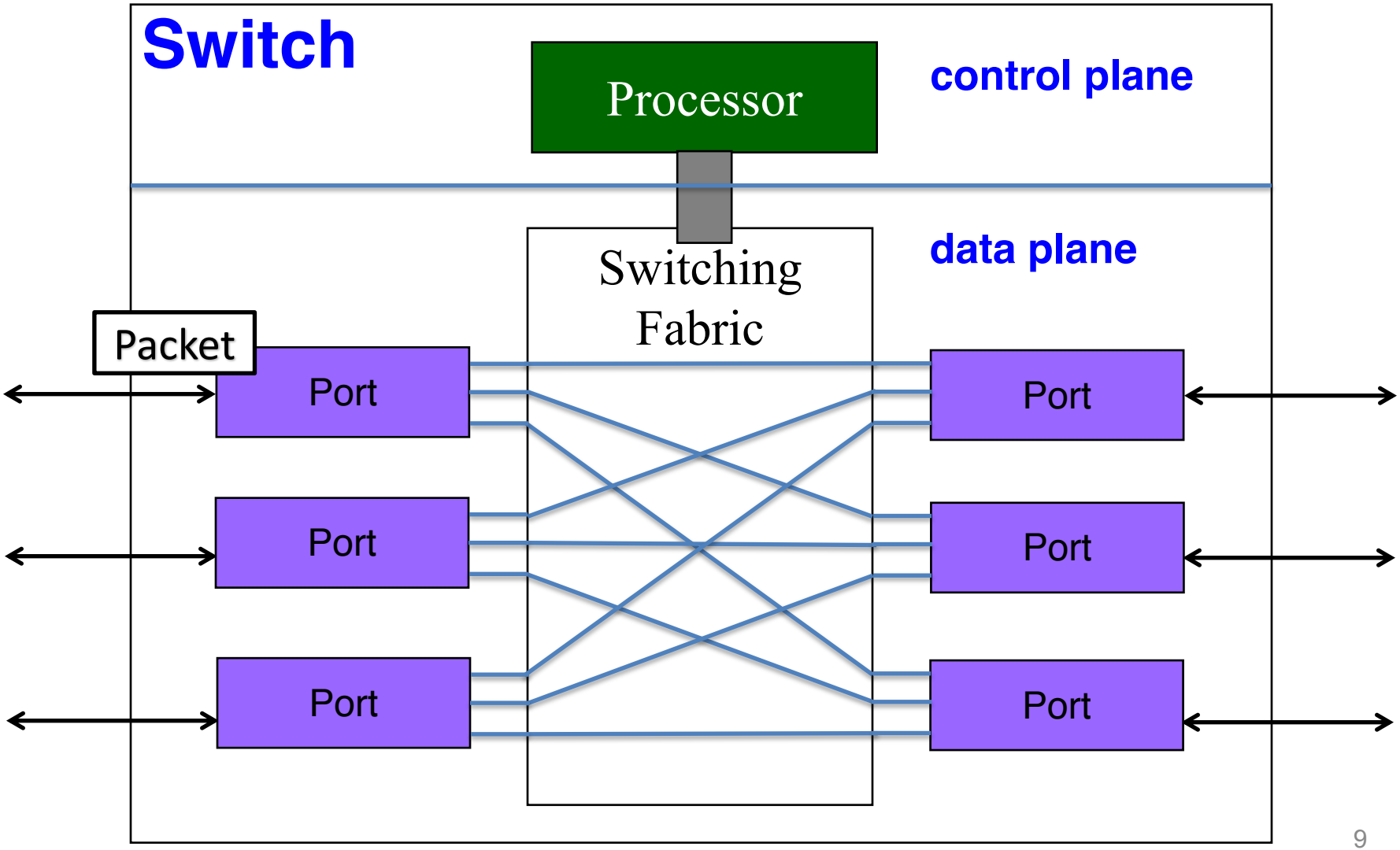
Management plane: configures and tunes the network

- Traffic engineering, ACL config, device provisioning, …

Timescales

	Data	Control	Management
Trigger	Packet	Event	Human
Scale	Nanoseconds	Milliseconds	Min to hours
Location	Network Hardware	Router software	Humans or scripts

Data and Control Planes

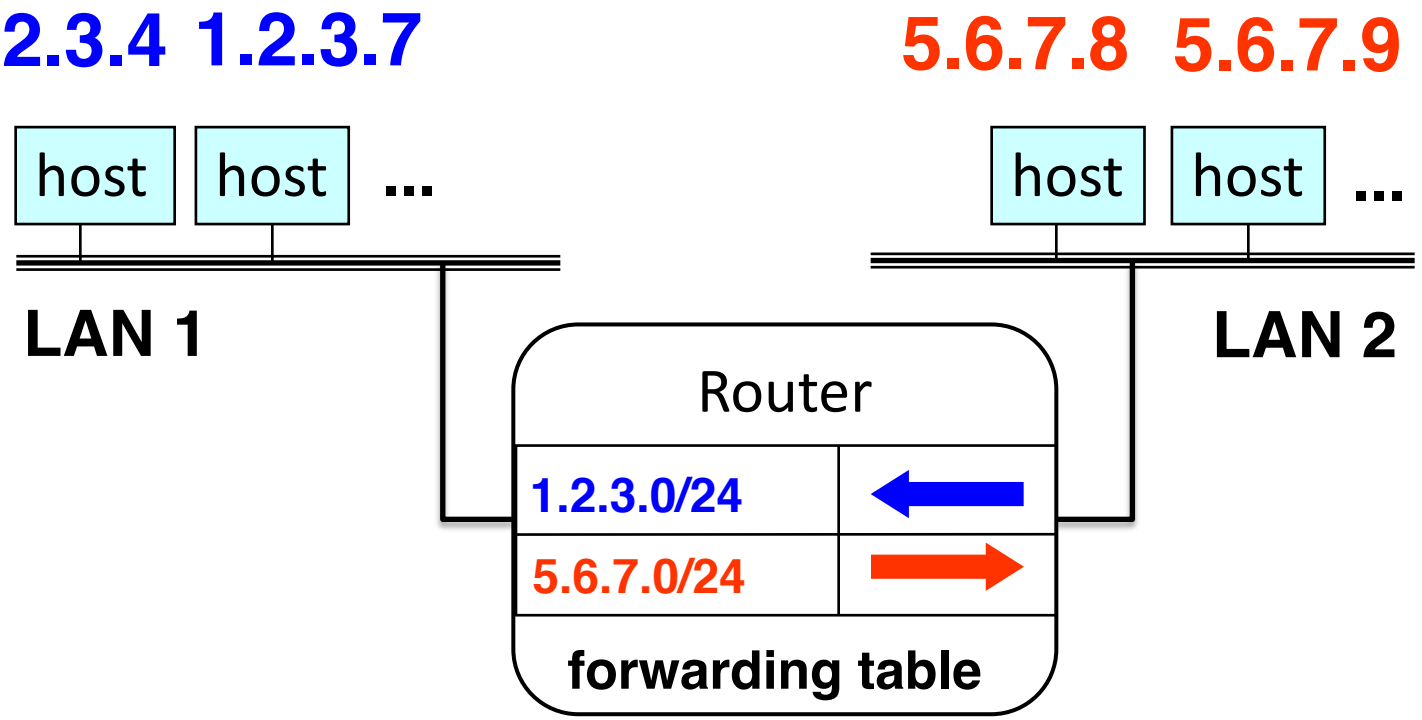


Data Plane

A match-action pipeline

- **Match** fixed bits against entries in table
- Perform **Action** on hit

Example: IP Forwarding



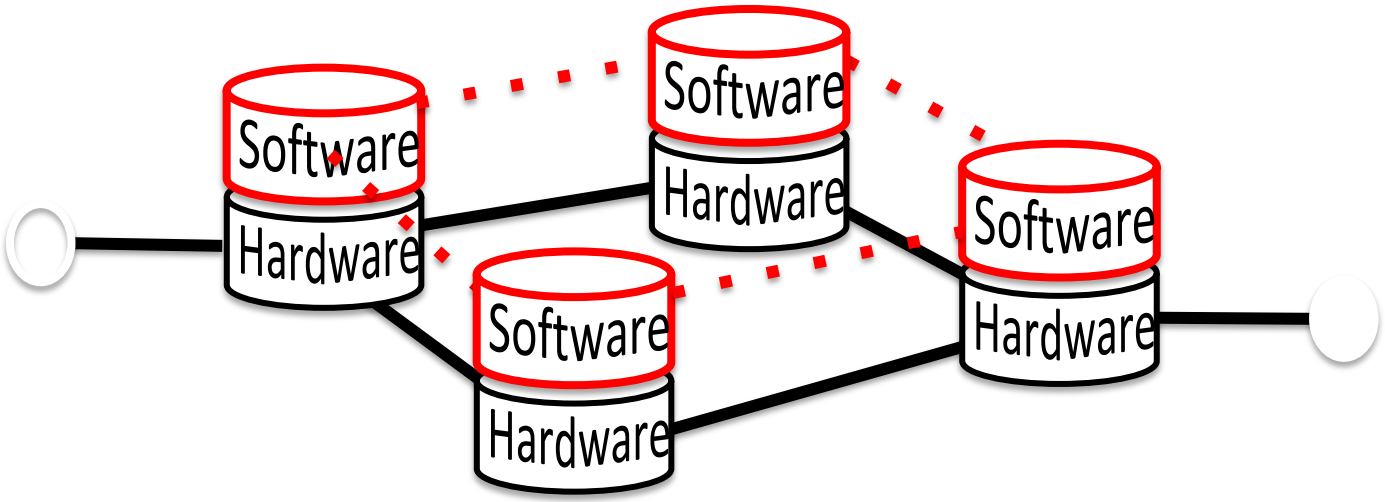
Control Plane

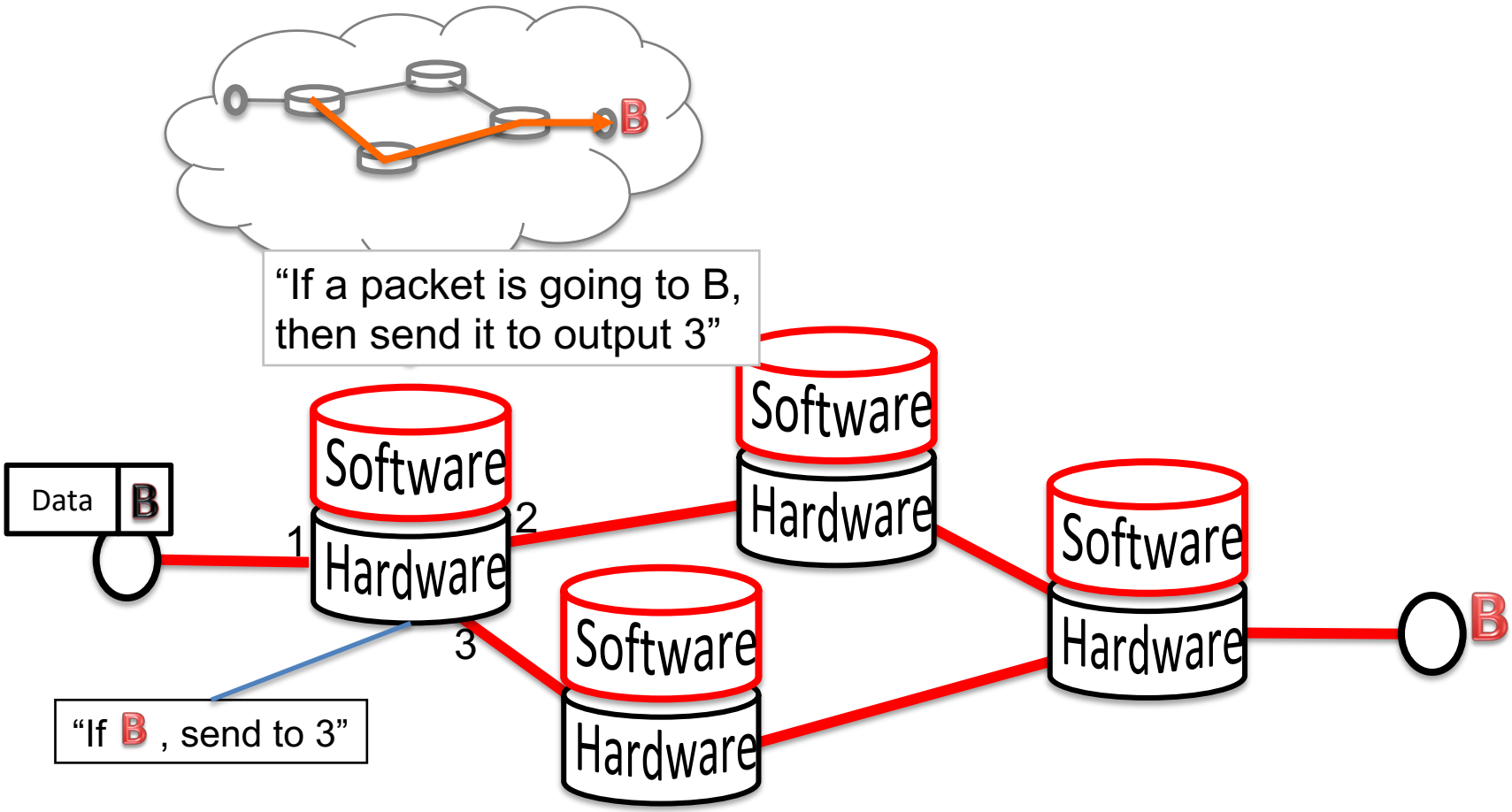
Computes decisions how to handle a packet

- Populates the match-action tables with entries
- Traditionally, a distributed protocol

Example: Link-state routing (OSPF, IS-IS)

- Flood the entire topology to all nodes
- Each node computes shortest paths



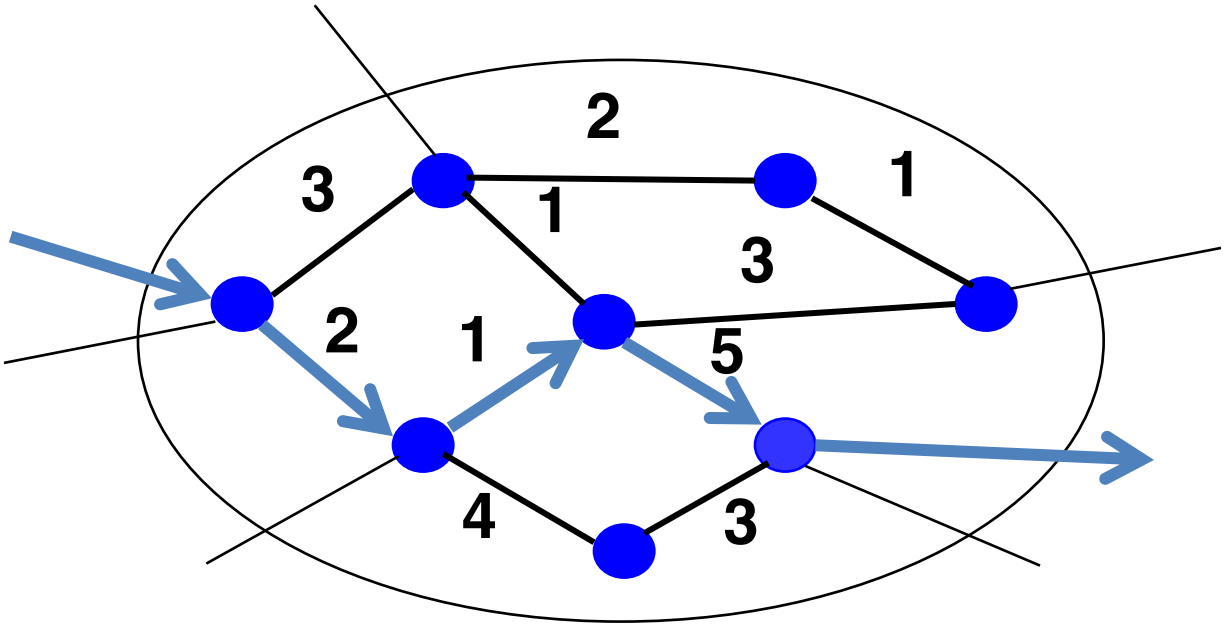


1. Figure out which routers and links are present.
2. Run algorithm (e.g., Dijkstra) to find shortest paths.

Management Plane

Traffic Engineering: Configures the control plane

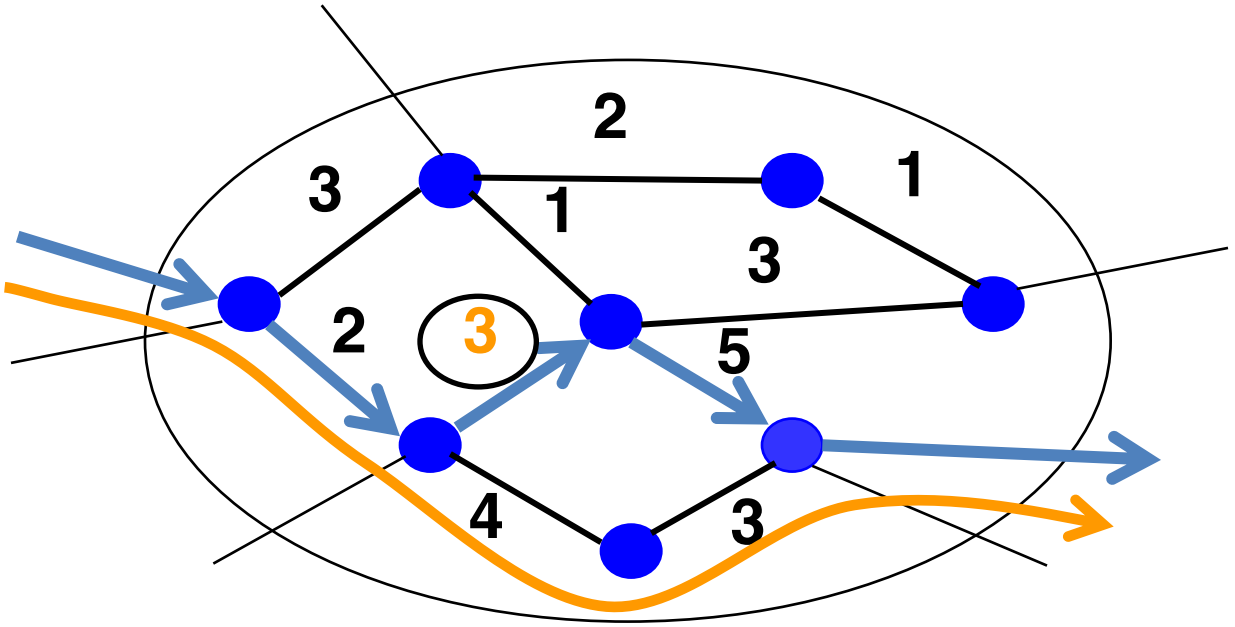
- specify cost of paths
- choose which protocols to use
- manage route preferences



Management Plane

Traffic Engineering: Configures the control plane

- specify cost of paths
- choose which protocols to use
- manage route preferences



The Big Flaw of Networks

"Big bag of protocols"

- Interaction of protocols defines system behavior
- Each task is defined as individual protocol
  - Routing, addressing, access control, QoS
- Operator has only indirect control
  - e.g., specifies weights of routes, updates parameters

A distributed system without central coordination

- Each network element has own control/data plane
- Operator has to manage everything individually
- Upgrades require compatibility with existing protocols

The Big Flaw of Networks

"Big bag of protocols"

- Int…or
- Ea…
  - …
- Op…
  - …s

A distri…tion

- Each network element has own control/data plane
- Operator has to manage everything individually
- Upgrades require compatibility with existing protocols

The network is

- Hard to reason about
- Hard to evolve
- Expensive

Complexity leads to problems…

LILY HAY NEWMAN SECURITY 11.06.17 05:00 PM

HOW A TINY ERROR SHUT OFF THE INTERNET FOR PARTS OF THE US

Data Centre ▸ Networks

Google routing blunder sent Japan's Internet dark on Friday

NEWS

BGP errors are to blame for Monday's Twitter outage, not DDoS attacks

No, your toaster didn't kill Twitter, an engineer did

Programming Analogy

What if programmers had to:

- specify where each value is stored in memory
- deal with all internal process communication
- use a very constricted language

We need a better abstraction of the network!

Software Defined Network

A network in which the control plane is **physically separate** from the data plane.

and

A single (**logically centralized**) control plane controls several forwarding devices.

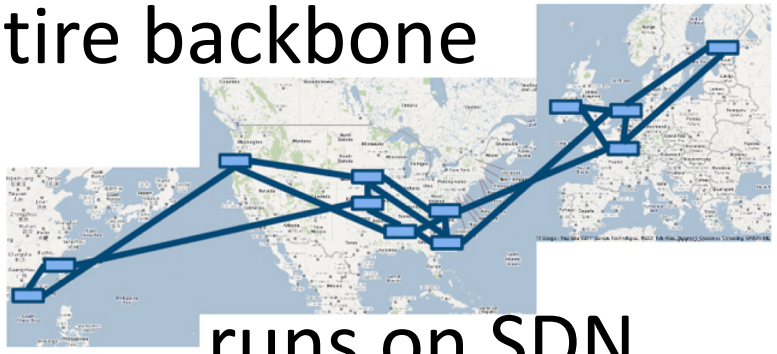
Not just an academic trend…

OPEN NETWORKING FOUNDATION

Deutsche Telekom · facebook · Goldman Sachs · Google · Microsoft · NTT Communications · verizon · YAHOO!

Google

Entire backbone



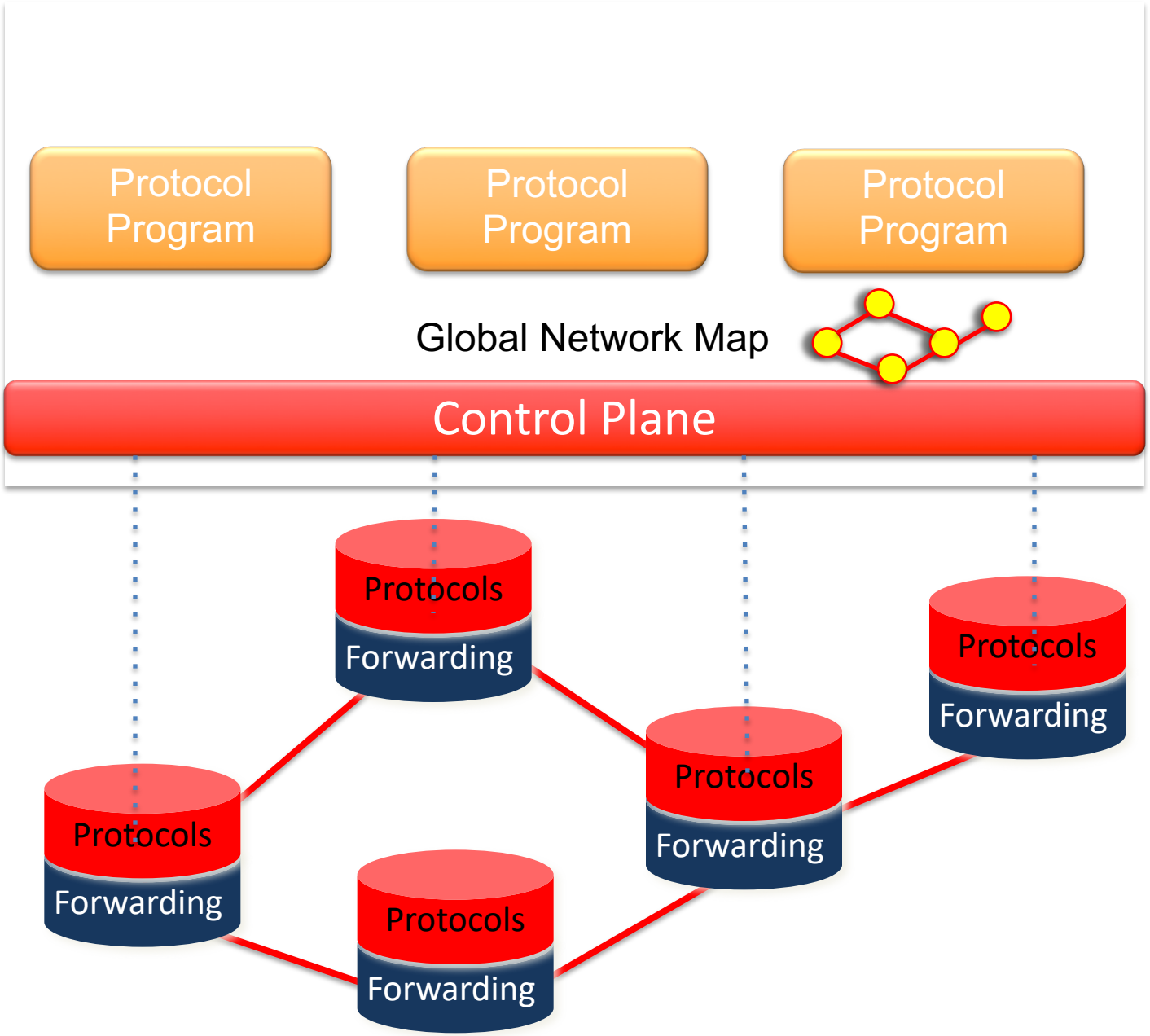
runs on SDN

nicira

Bought for **$1.2 billion by**

vmware®

How SDN changes the network



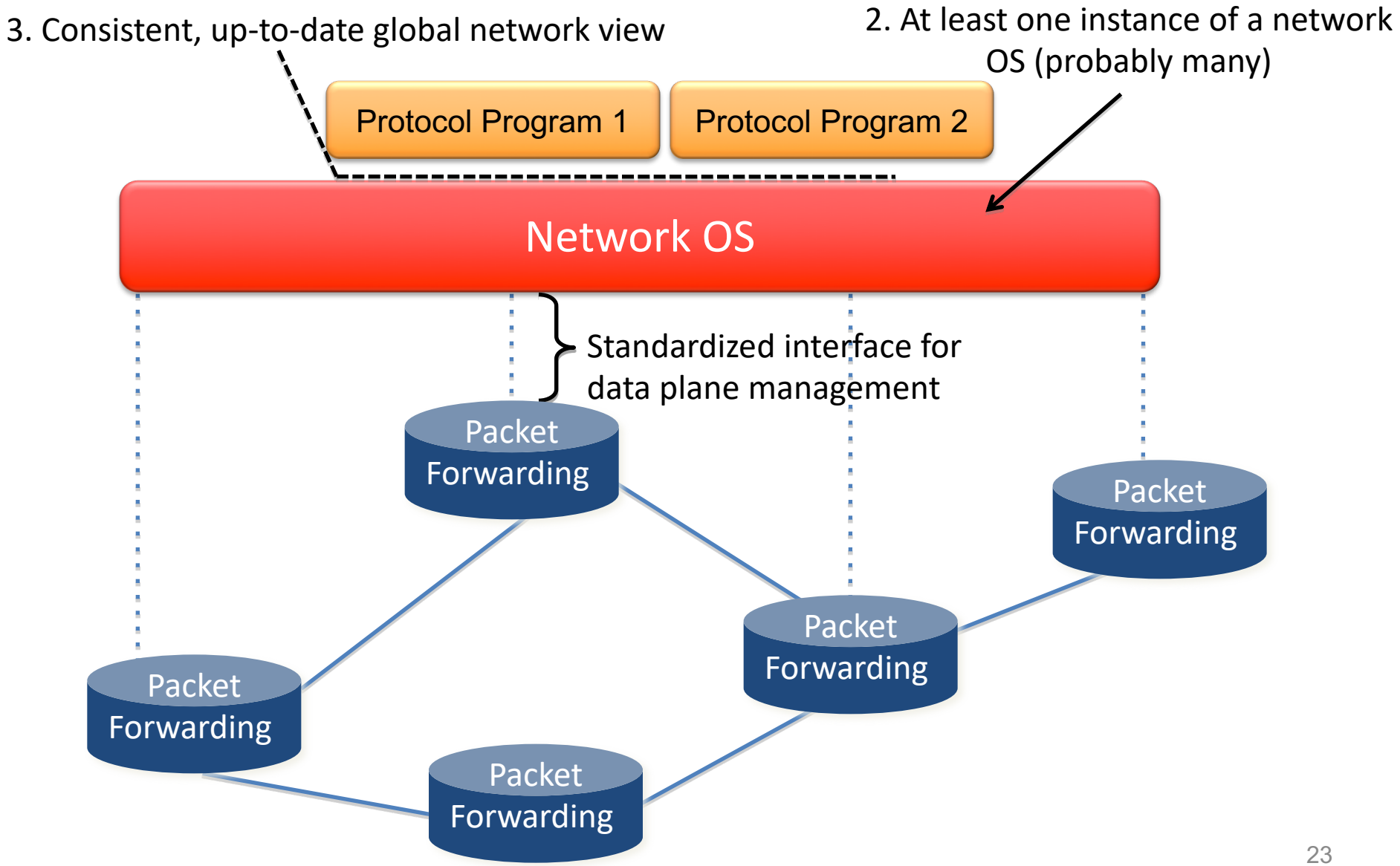
The network operating system

Network OS

- Manages all network nodes
- Establishes a global view of the network topology
- Provides an interface to write applications
- Examples: ONOS, OpenDaylight

How does this compare to your typical OS?

The Network Operating System



Control Program

Control programs operate on a **global** view of network

- **Input**: network view (graph/database), policy
- **Output**: configuration of each network device

A control program is **not** a distributed system

- It **operates** on one
- All distributed state is hidden away
- Programmer does not need to know about network details

Forwarding Abstraction

Purpose: API between network OS and the network

- Flexible
  - All behavior specified by control plane
  - Able to perform all operations based on primitives
- Minimalistic
  - Streamlined for speed and low-power
  - Open-source and standardized to encourage interoperability

OpenFlow is an example of such an abstraction

Outline

(Traditional) Network Planes

Software Defined Networking

Fabian Ruffy

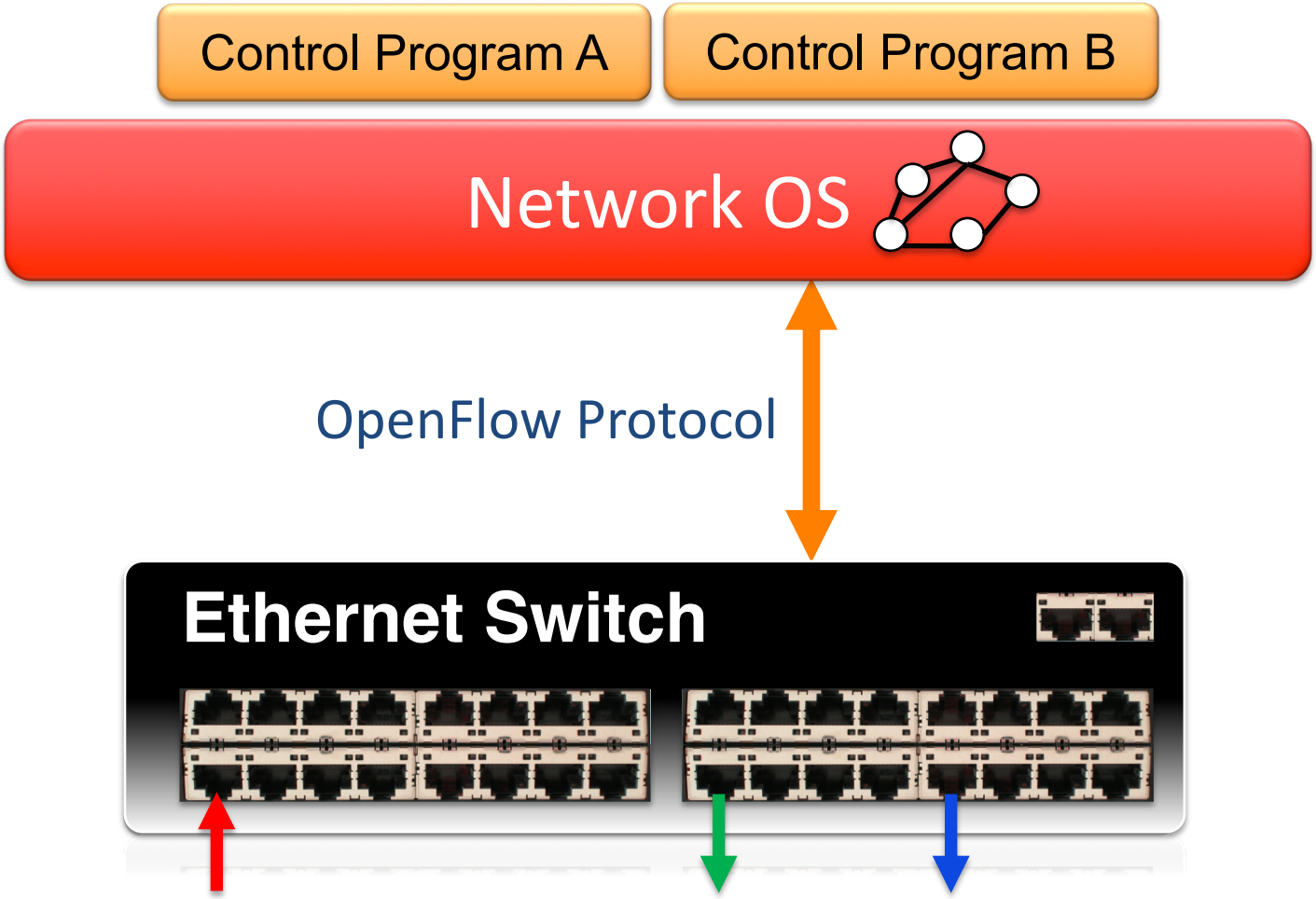
OpenFlow basics

Network virtualization

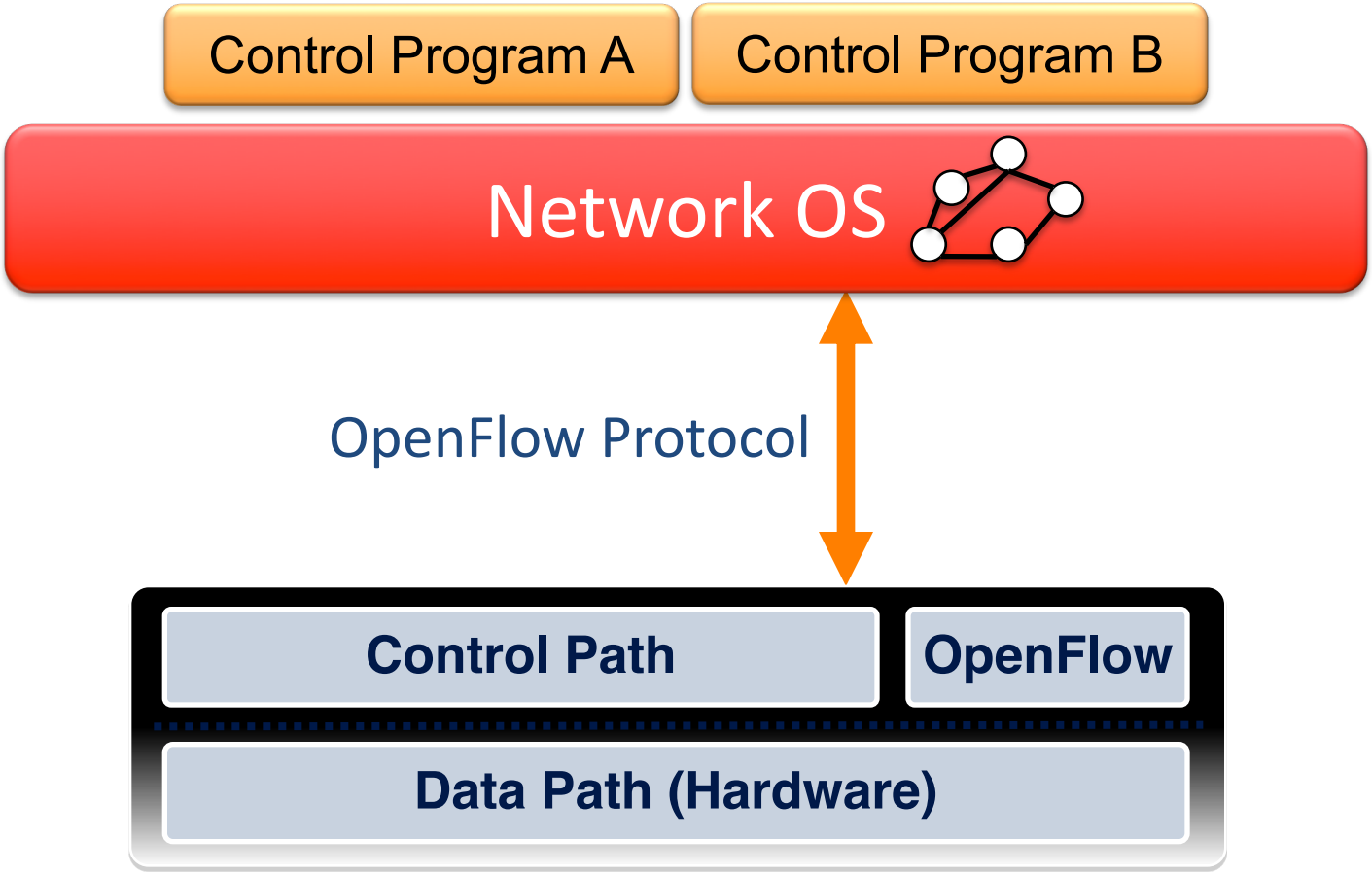
Nodir Kodirov

OpenFlow Basics

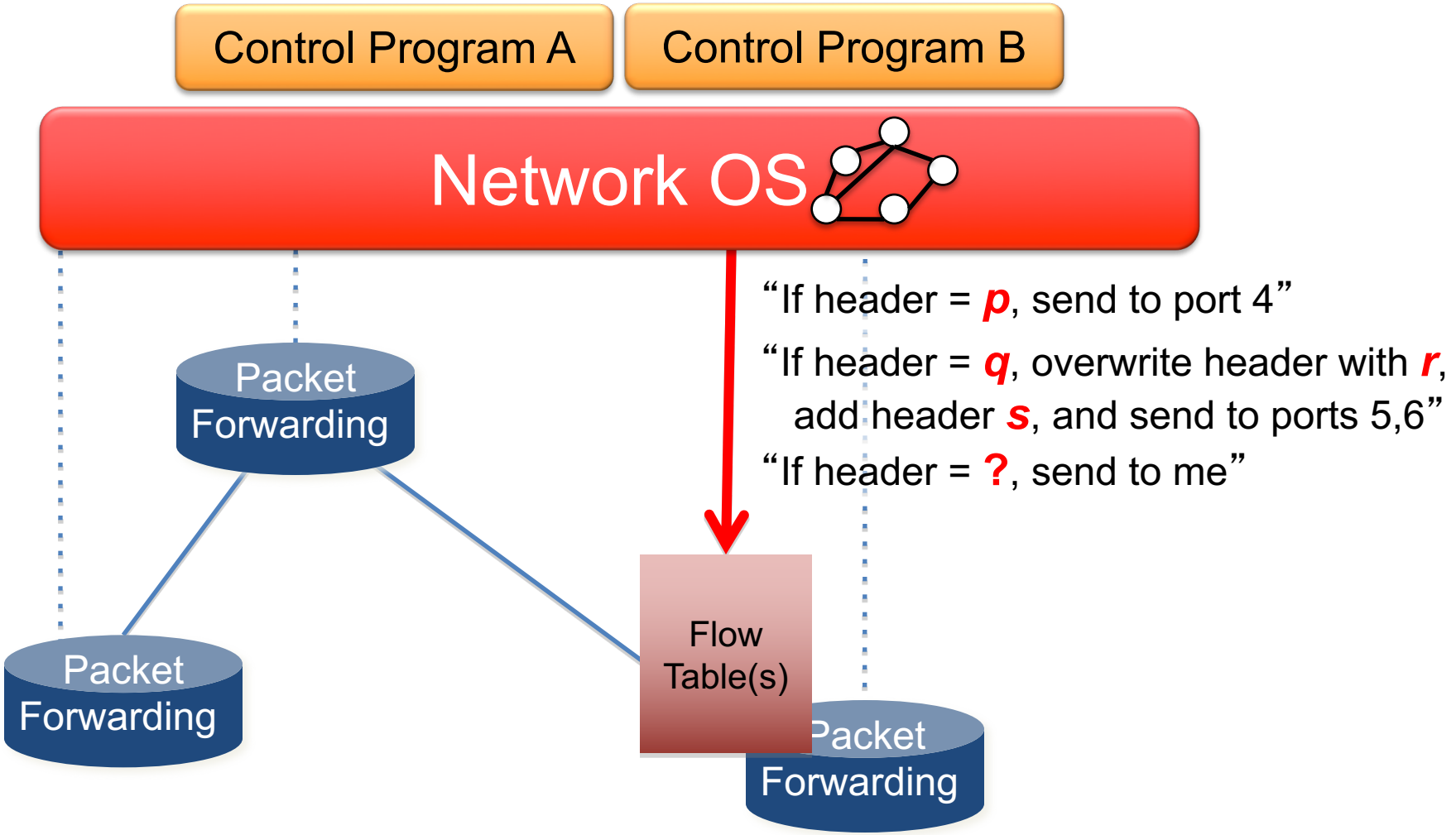
OpenFlow Basics



OpenFlow Basics



OpenFlow Basics



Primitives <Match, Action>

Match arbitrary bits in headers:

Match: 1000x01xx0101001x

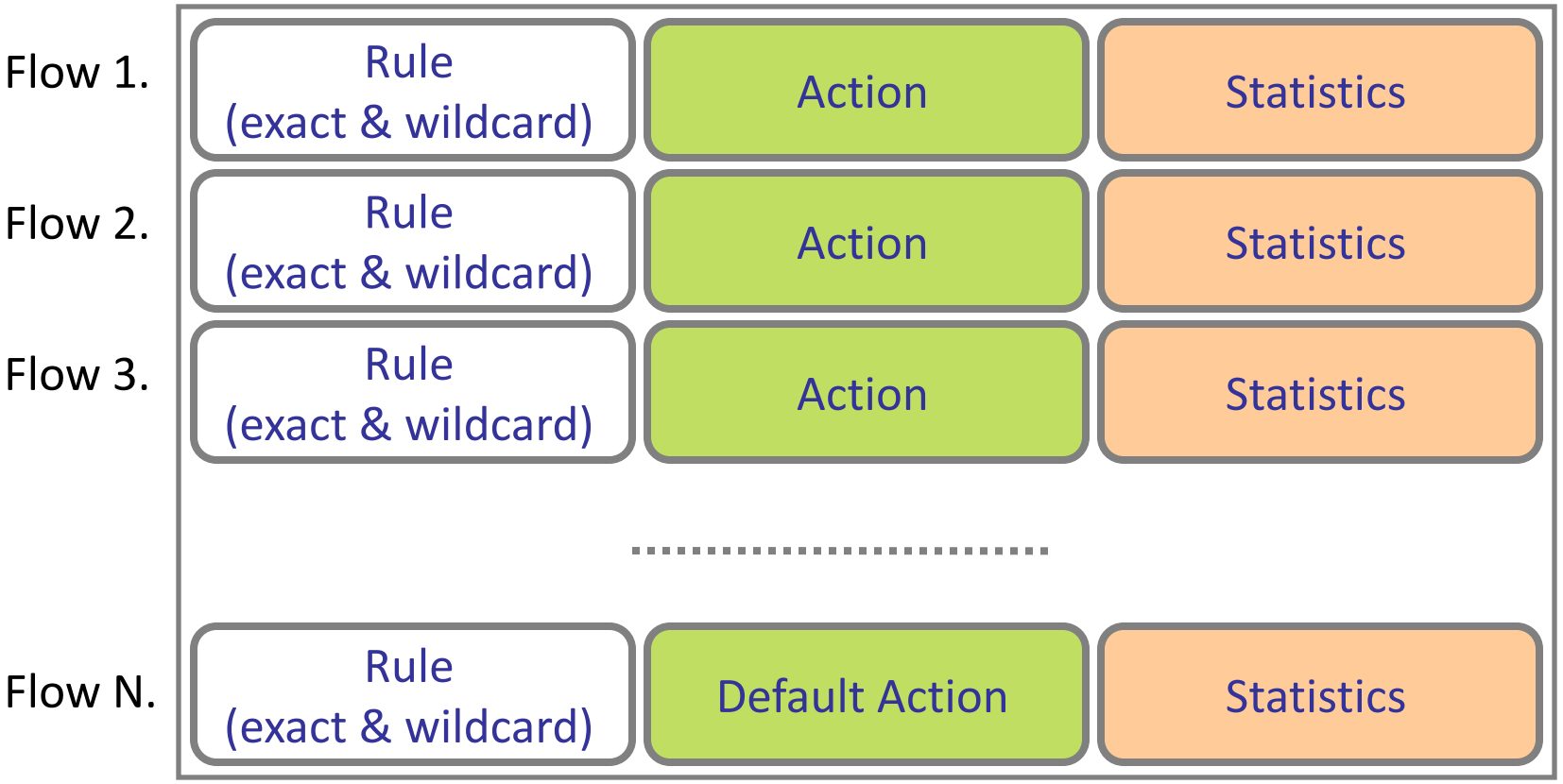
- Match on any header, or new header
- Allows any flow granularity

Action

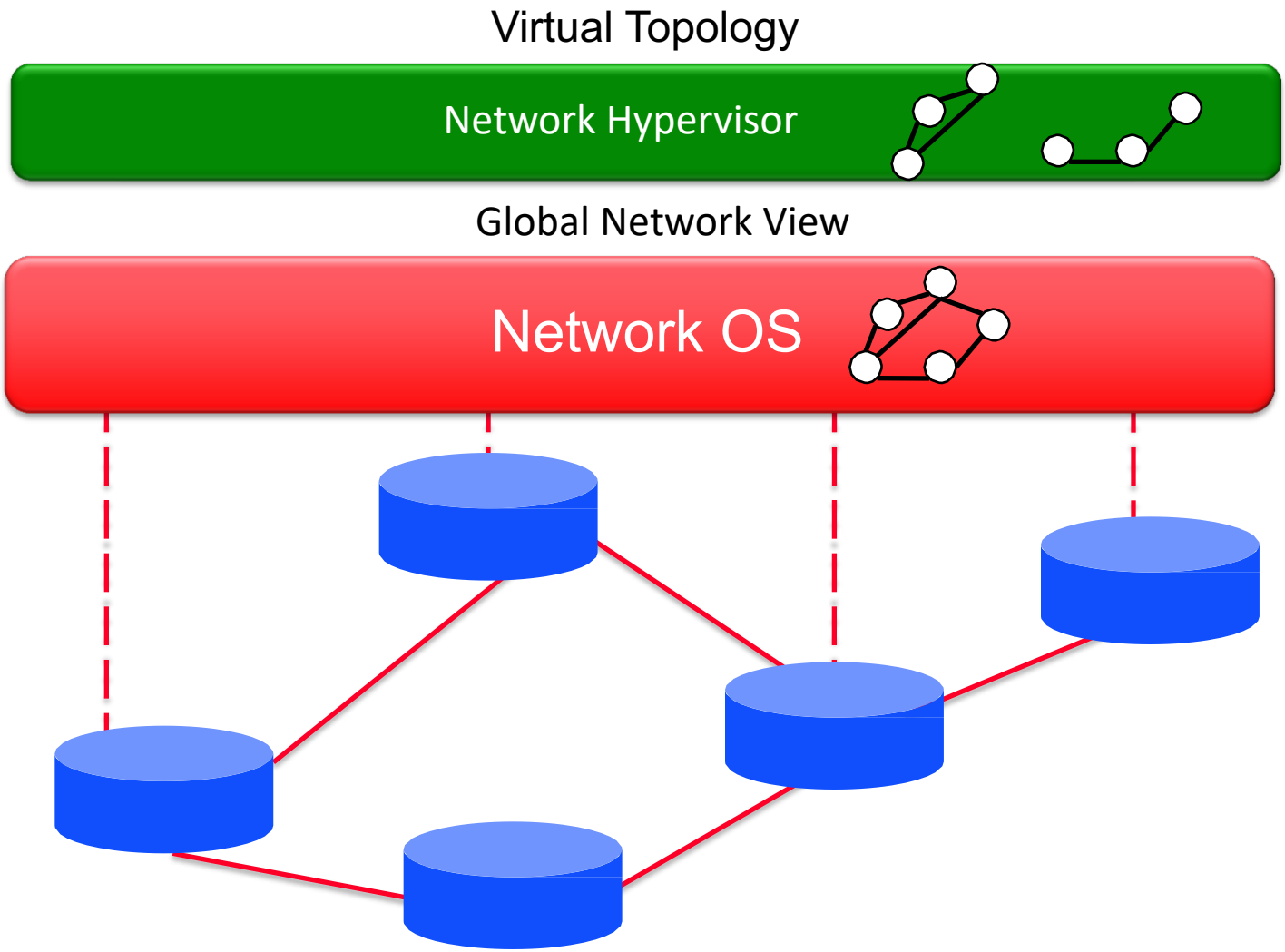
- Forward to port(s), drop, send to controller
- Overwrite header with mask, push or pop
- Forward at specific bit-rate

OpenFlow Rules

Exploit the flow table in switches, routers, and chipsets

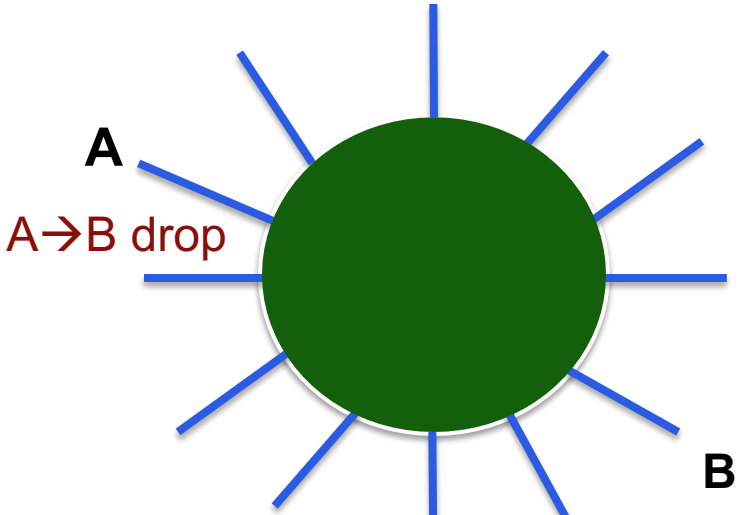


Software Defined Network



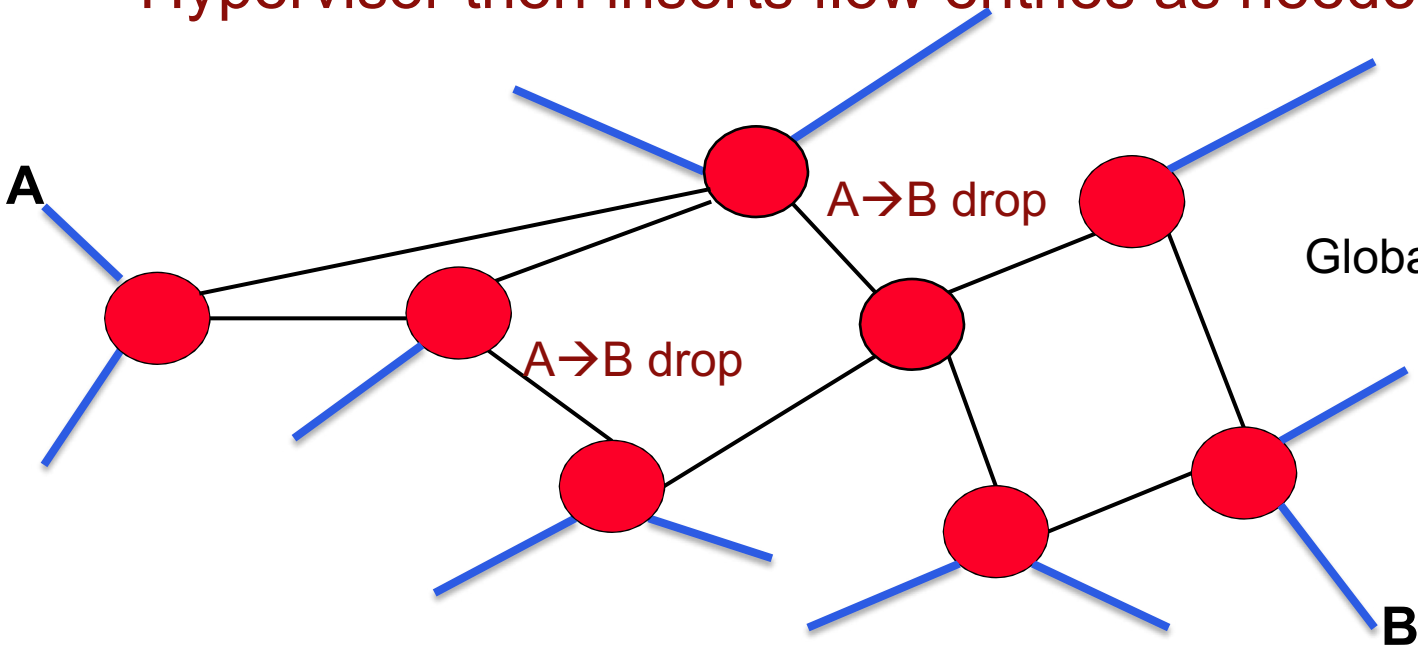
Network Virtualization

Virtualization Simplifies Control Program



Abstract Network View

Hypervisor then inserts flow entries as needed



Global Network View

Does SDN Simplify the Network?

Abstraction doesn't eliminate complexity

- NOS, Hypervisor are still complicated pieces of code

SDN main achievements

- Simplifies interface for control program (user-specific)
- Pushes complexity into reusable code (SDN platform)

Just like compilers….

Virtualization is Killer App for SDN

Consider a multi-tenant datacenter

- Want to allow each tenant to specify virtual topology
- This defines their individual policies and requirements

Datacenter's network hypervisor compiles these virtual topologies into set of switch configurations

- Takes 1000s of individual tenant virtual topologies
- Computes configurations to implement all simultaneously

This is what people are paying money for….

- ***Enabled by SDN's ability to virtualize the network***

Practical Challenges of SDN

Scalability
- Decision elements responsible for many routers

Reliability
- Surviving failures of decision elements and routers

Response time
- Delays between decision elements and routers

Consistency
- Ensuring multiple decision elements behave consistently

Security
- Network vulnerable to attacks on decision elements

Interoperability
- Legacy routers and neighboring domains

Practical Challenges of all distributed systems

Scalability
- Decision elements responsible for many routers

Reliability
- Surviving failures of decision elements and routers

Response time
- Delays between decision elements and routers

Consistency
- Ensuring multiple decision elements behave consistently

Security
- Network vulnerable to attacks on decision elements

Interoperability
- Legacy routers and neighboring domains

Takeaways

- Any network is a distributed system
  - Data Centre has a single AS: allows to reason about networking end-to-end
- SDN accelerates network evolution by
  - Decoupling control plane and data plane
    - Data plane is on firmware (only local state)
    - Control place faces distributed system challenges
  - Defining an open protocol (OpenFlow) between control plane and data plane
- Network virtualization is a killer SDN app
  - Tenant isolation in cloud

Backup slides

Why is SDN happening now?

The Road to SDN

Active Networking: 1990s

- First attempt make networks programmable
- Demultiplexing packets to software programs, network virtualization, …

Control/Dataplane Separation: 2003-2007

- ForCes [IETF],
  RCP, 4D [Princeton, CMU],
  SANE/Ethane [Stanford/Berkeley]
- Open interfaces between data and control plane, logically centralized control

OpenFlow API & Network Oses: 2008

- OpenFlow switch interface [Stanford]
- NOX Network OS [Nicira]

N. Feamster et al., "The Road to SDN: An Intellectual History of Programmable Networks", ACM SIGCOMM CCR 2014.

Drivers for SDN Adoption

Rise of merchant switching silicon

- Democratized switching
- Vendors eager to unseat incumbents

Cloud / Data centers

- Operators face real network management problems
- Extremely cost conscious; desire a lot of control

The right balance between vision & pragmatism

- OpenFlow compatible with existing hardware

A “killer app”: Network virtualization

Example 1: Inter-domain Routing

Today's inter-domain routing protocol, BGP, artificially constrains routes

- Routing only on **destination IP address blocks**
- Can only influence **immediate neighbors**
- Very difficult to incorporate other information

Application-specific peering

- Route video traffic one way, and non-video another

Blocking denial-of-service traffic

- Dropping unwanted traffic further upstream

Inbound traffic engineering

- Splitting incoming traffic over multiple peering links