COMP3210: Advanced Computer Networks

Software Defined Networking

Professor Shiyan Hu
School of Electronics and Computer Science
University of Southampton

UNIVERSITY OF Southampton

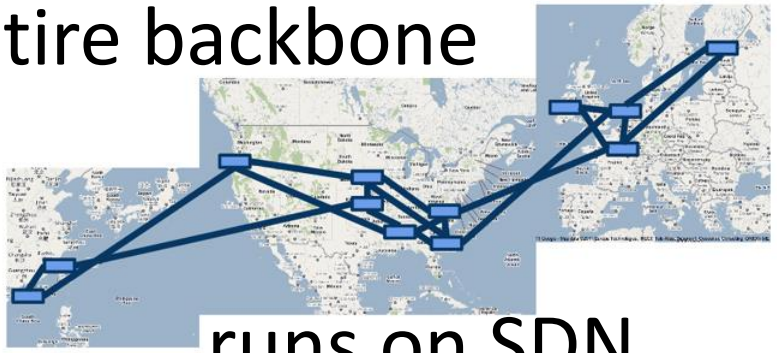
A Major Trend in Networking

OPEN NETWORKING FOUNDATION

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

Google

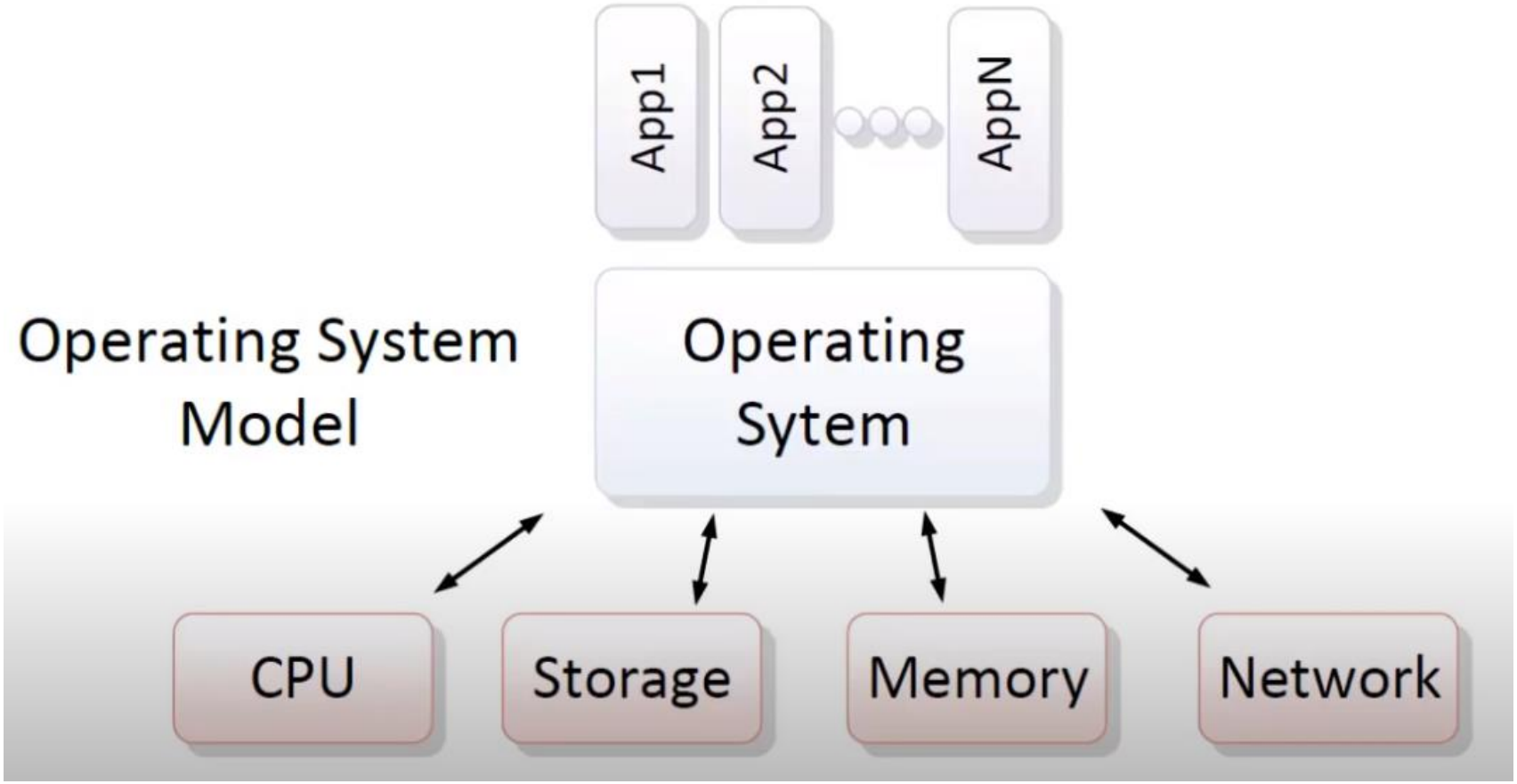
Entire backbone



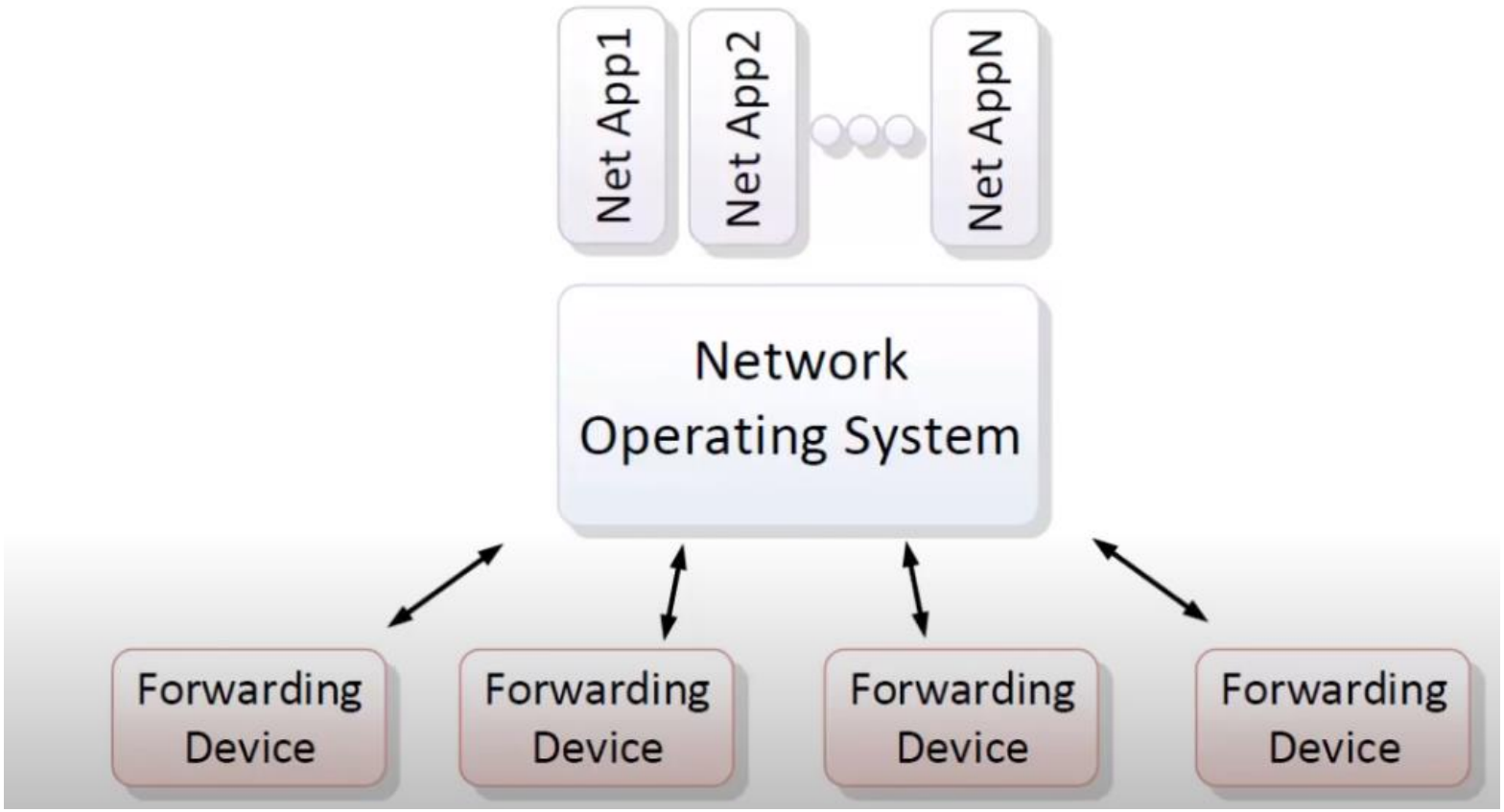
runs on SDN

nicira

Motivation (I)



Motivation (II)



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.

Traditionally

Data plane: processing and delivery of packets with local forwarding state

- Packet header + forwarding state → switch router looks at the packet header & forwarding state and make the forwarding decision (it is a local decision which can be performed very fast)

Control plane: computing the forwarding state in routers

- Determines how and where packets are forwarded (needs non-local information, not fast)

Management plane: the set of applications (such as firewalls, load balancers) that leverage the functions to implement network control and operation logic.

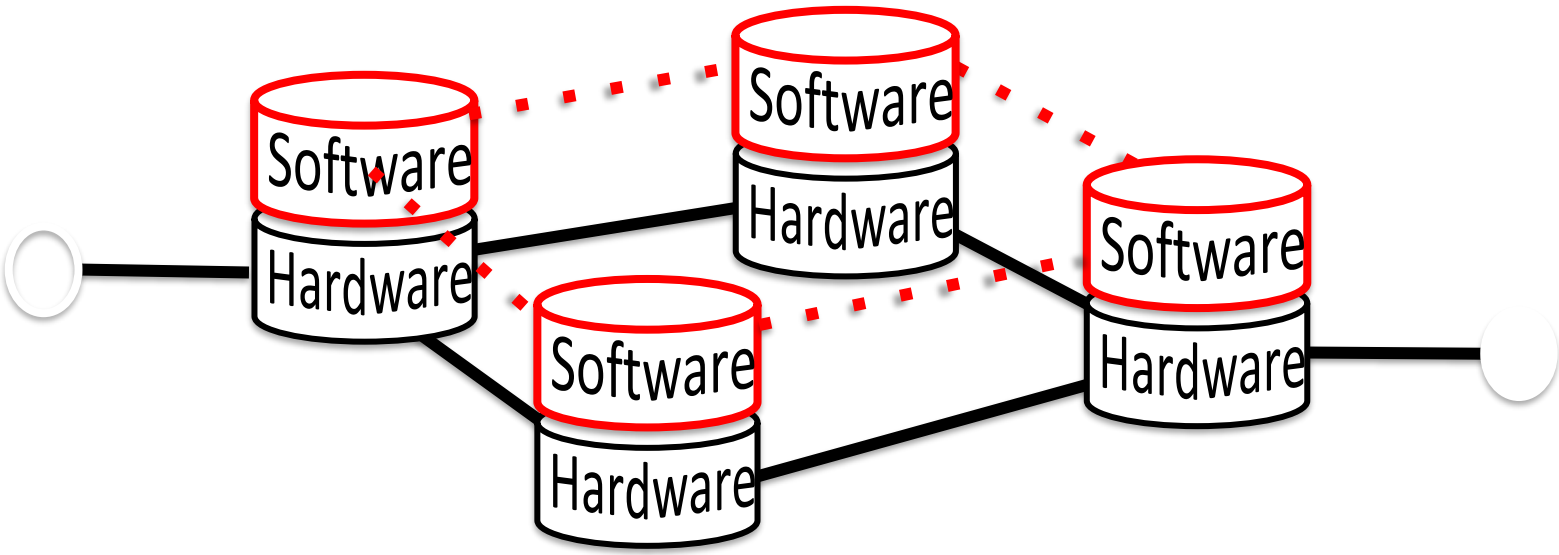
Control Plane

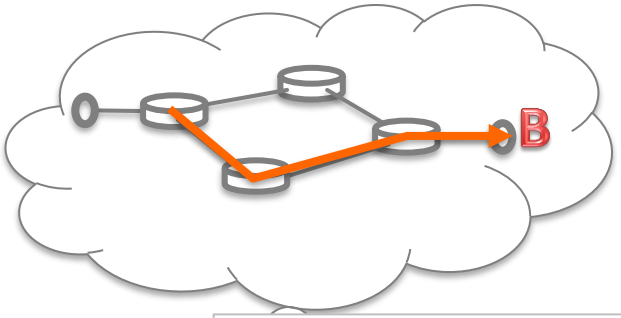
Compute paths the packets will follow

- Populate forwarding tables
- Traditionally, a distributed protocol

Example: Link-state routing (OSPF)

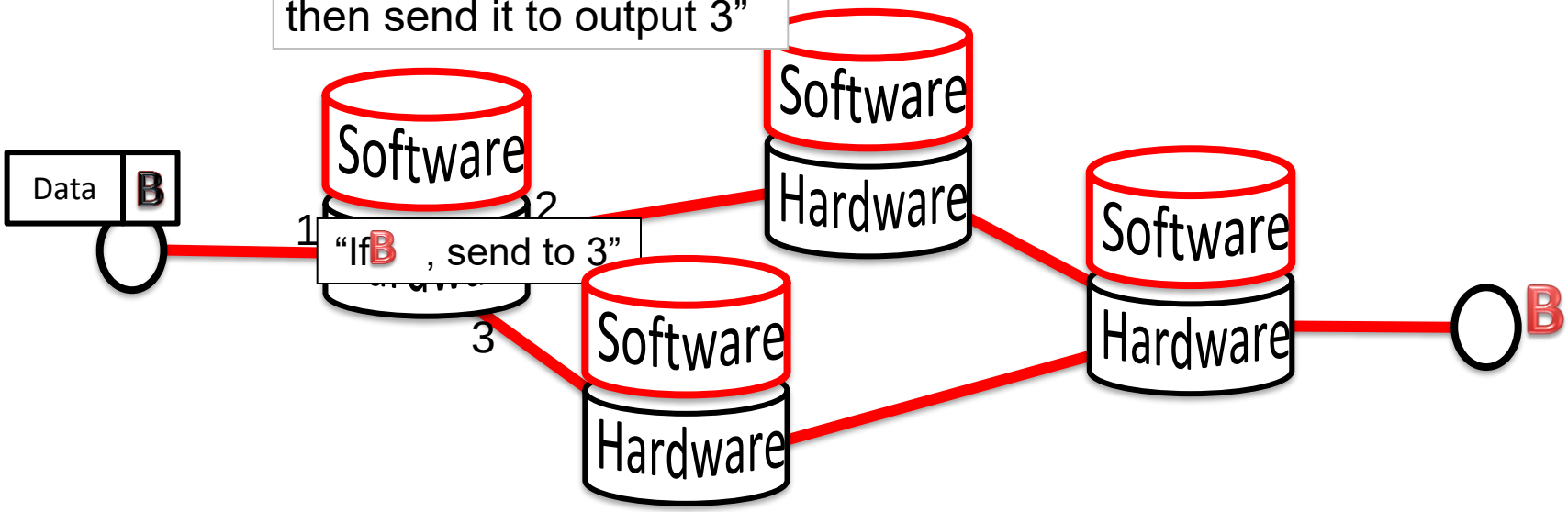
- Flood the entire topology to all nodes
- Each node computes shortest paths
- Dijkstra's algorithm


Software
Hardware
Software
Hardware
Software
Hardware
Software
Hardware



1. Figure out which routers and links are present.
2. Run Dijkstra's algorithm to find shortest paths.

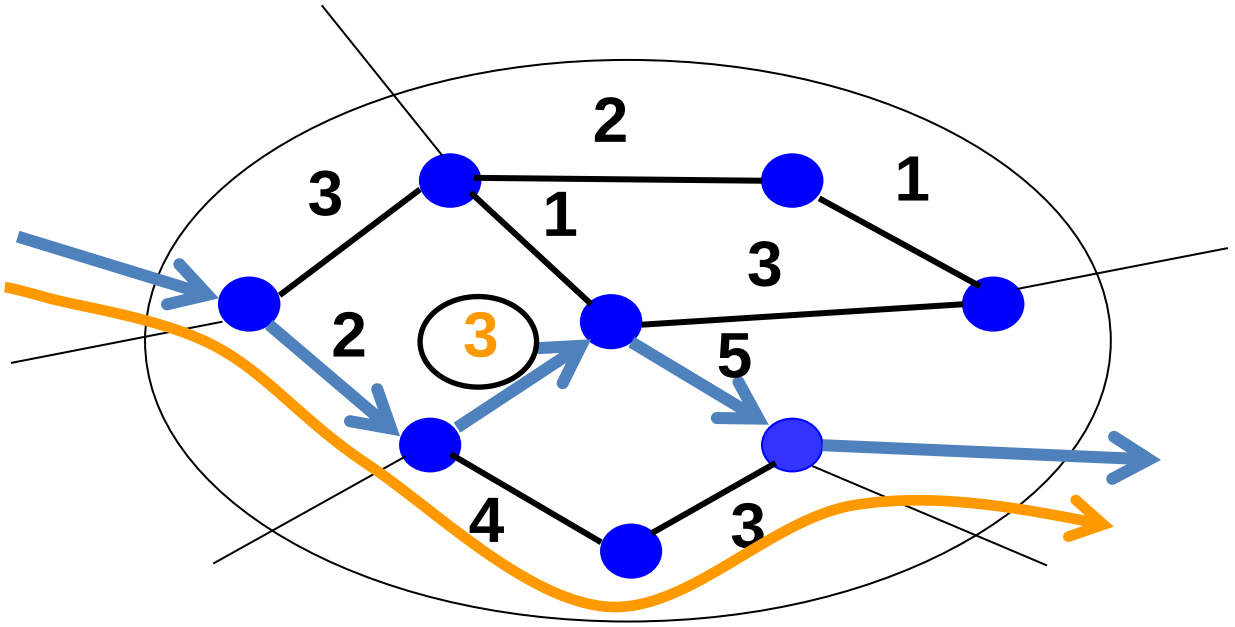
"If a packet is going to B, then send it to output 3"



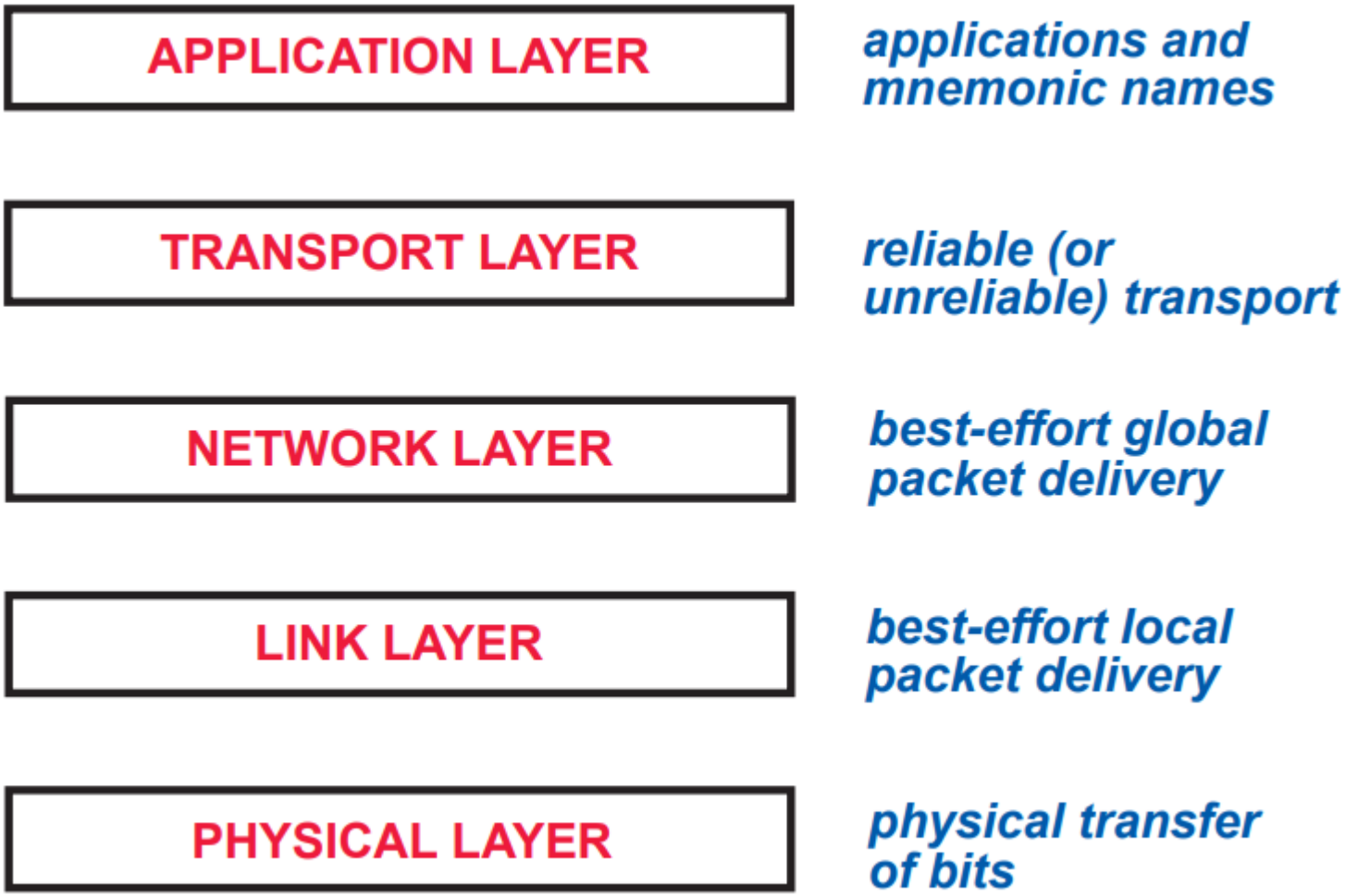
Management Plane

Traffic Engineering: setting the weights

- Inversely proportional to link capacity?
- Proportional to propagation delay?
- Network-wide optimization based on traffic?



Data Plane Abstractions: Layers



Layers are Key to Success

- Decompose the problem into tractable pieces
- Enable rapid innovation in each layer
- Control Plane: No abstraction, no layer, no building block, many mechanisms and goals
  - Routing: distributed routing algorithms
  - Traffic engineering: adjusting weights, etc

Motivation

- That is why we would like to have SDN, to introduce abstractions/modularity to control plane.
- Make the network open and programmable

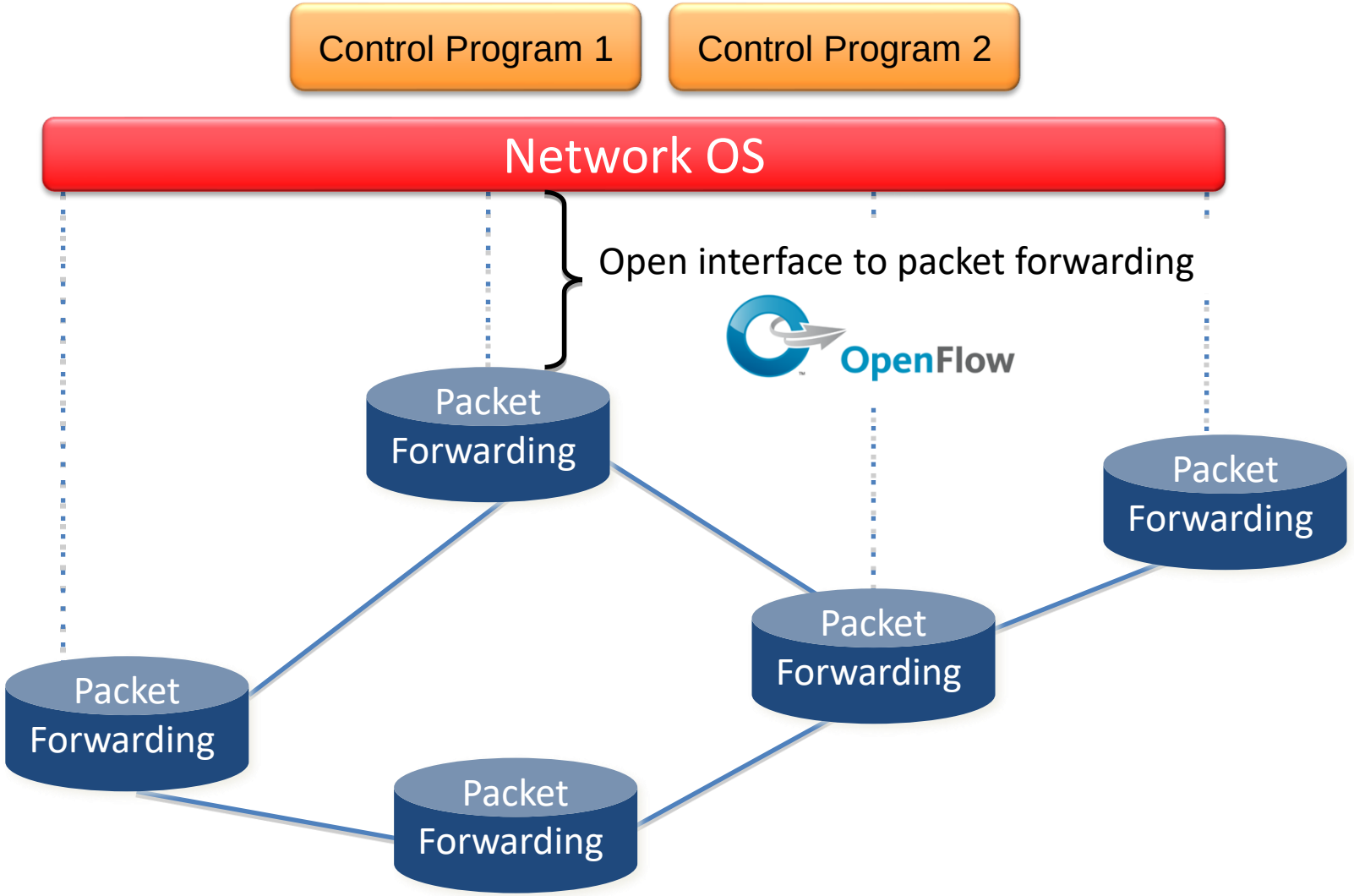
Abstractions for Control Plane

- Abstractions: identify reusable components
- The control plane needs to
  - Figure out the network topology (global network view)
  - Figure out how to accomplish routing on a given topology
  - Configure forwarding states for the switches
- What can be reused?
  - Figure out the network topology
  - Configure forwarding states for the switches

SDN: The two control plane abstractions

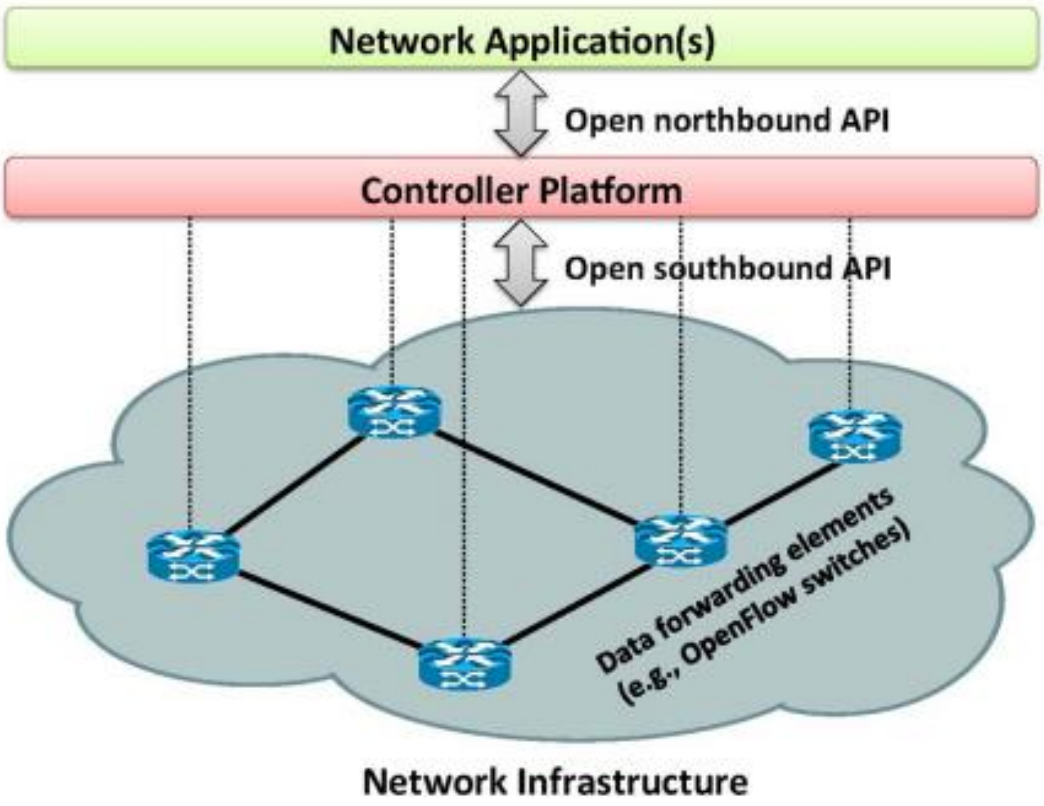
- Figure out the network topology
  - Provides global network view about the current network topology
  - Implemented with Network Operating System, using software running on (logically centralized) servers in the network
- Configure forwarding states for the switches
  - Provides standard way of defining forwarding state
  - This is OpenFlow: specification of <header, action> flow entries

Software Defined Network



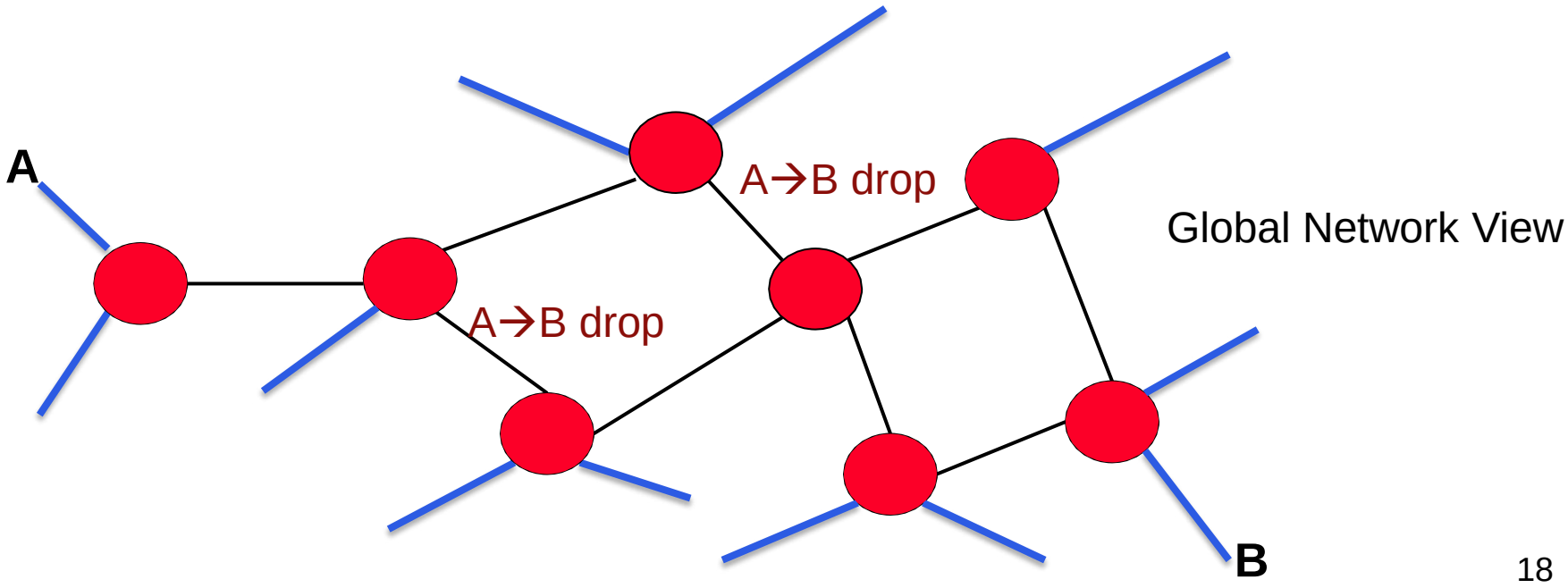
16

Simplified SDN Architecture



Example: Access Control

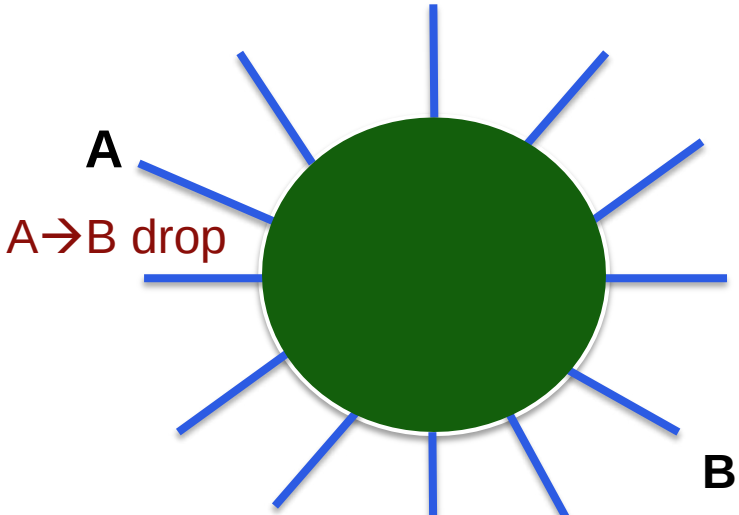
- Goal of operator: prevent A's packages from reaching B
- The control program can accomplish this
- It needs to respond to topology/routing changes



Network Virtualization

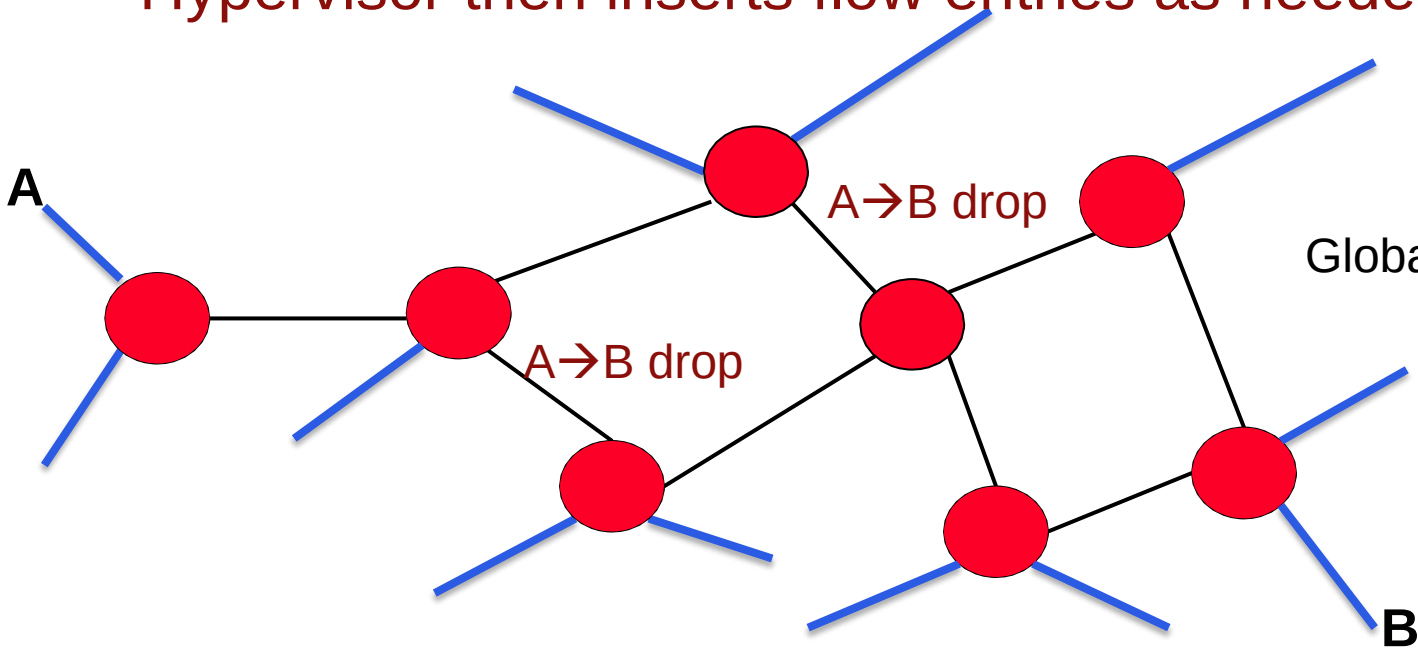
- Introduce new abstraction and a new SDN layer
- Abstraction: virtual topology
  - Allow operator to express requirements and policies
- Layer: network hypervisor
  - Implement those requirements into switch configurations
  - “Compiler” for virtual topologies

Virtualization Simplifies Control Program



Abstract Network View

Hypervisor then inserts flow entries as needed



Global Network View

Does SDN Simplify the Network?

Abstraction does not eliminate complexity

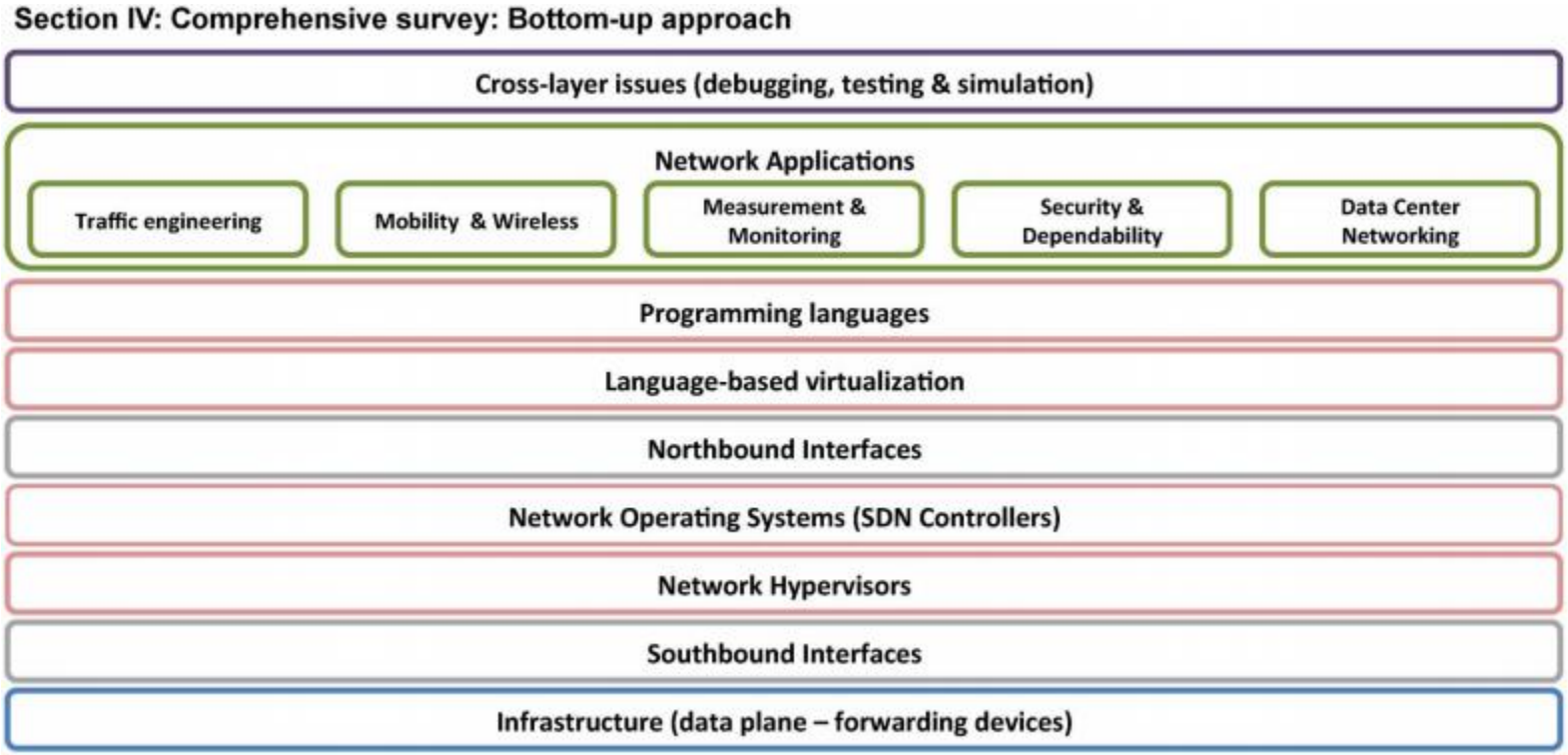
- Hypervisor are still complicated pieces of code

Main achievements of SDN

- Simplifies interface for control program (user-specific)
- Pushes complexity into reusable code (SDN platform)
- Hypervisor are “compilers”

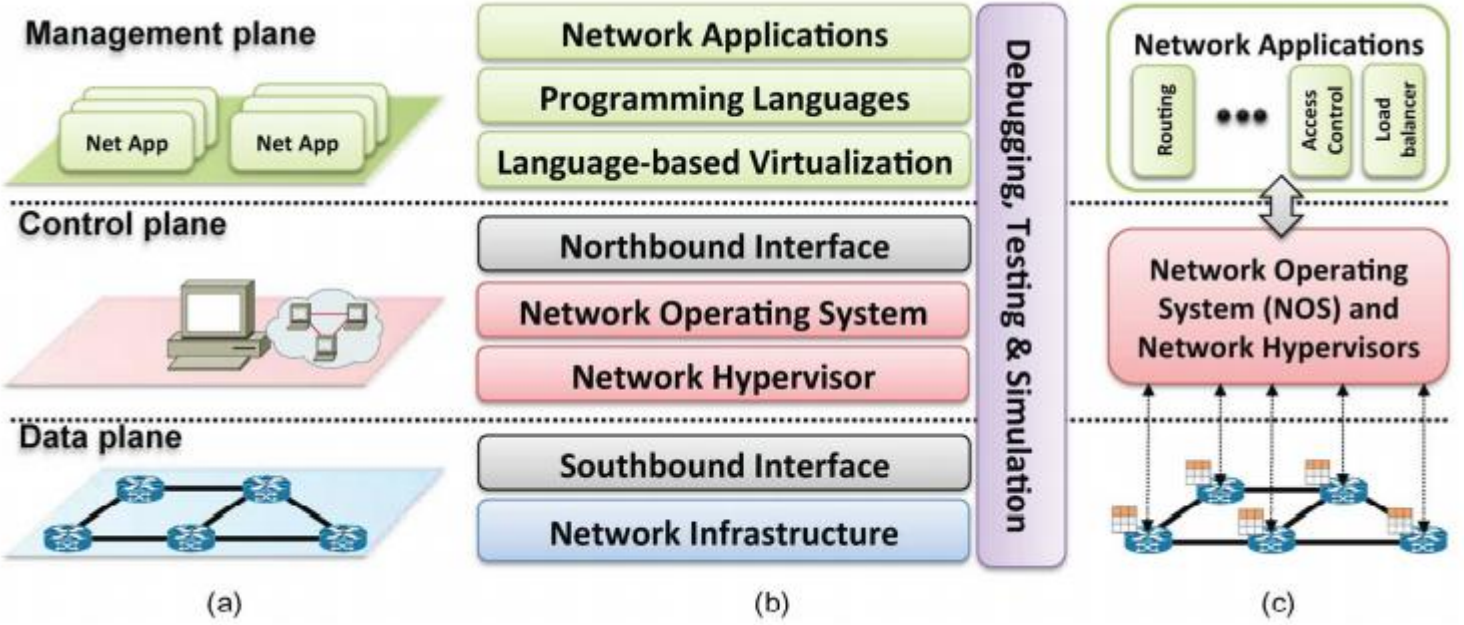
SDN

- SDN established control plane modularity
- Virtualization, implemented by hypervisor, decoupled control program from physical network

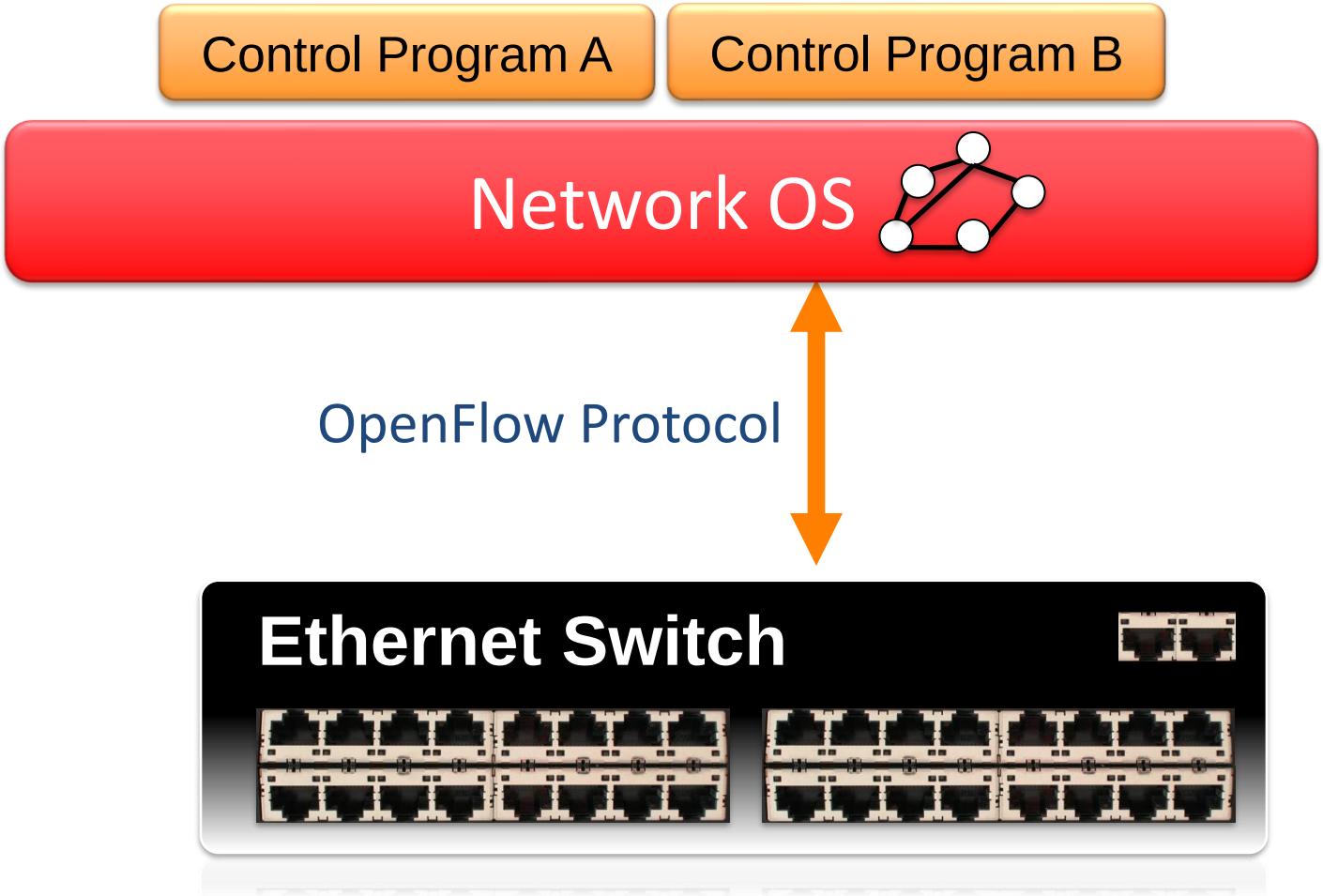


Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, Software-Defined Networking: A Comprehensive Survey, in Proceedings of the IEEE, Vol. 103, No. 1, pp 14-76, Jan. 2015

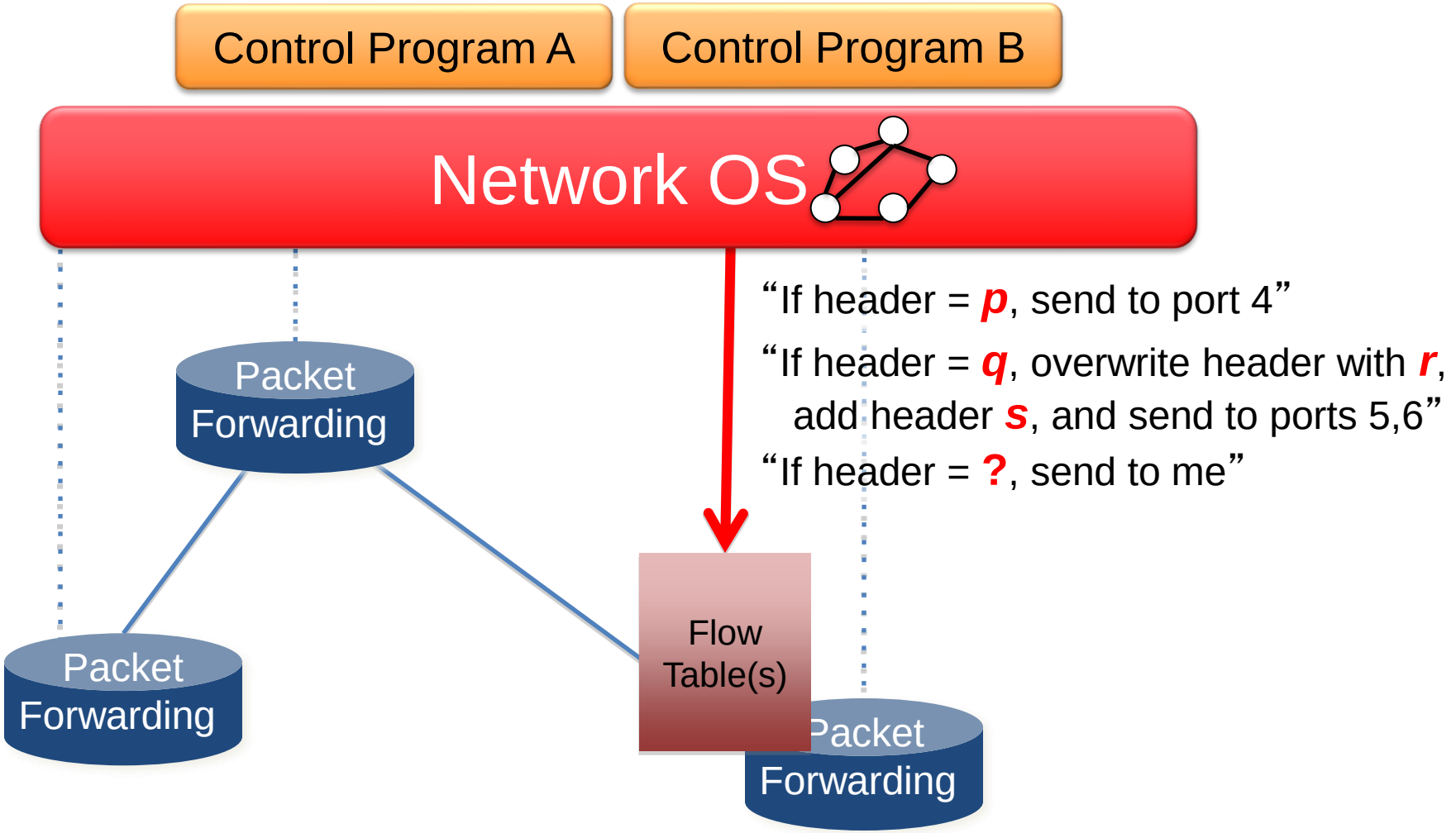
Planes, Layers, Architecture



OpenFlow Basics

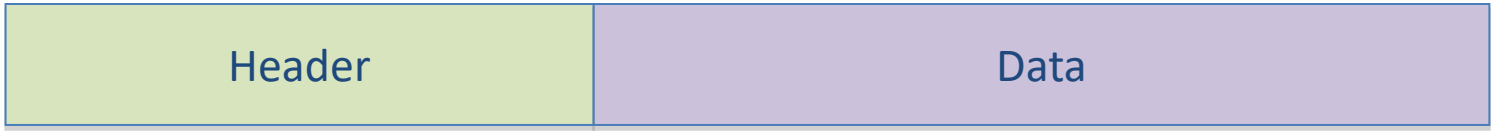


OpenFlow Basics



Primitives <Match, Action>

- **Match** arbitrary bits in headers:



Match: 1000x01xx0101001x

- **Action**
  - Forward to port(s), drop, send to controller
  - Overwrite header with mask
  - Forward at specific bit-rate

- OpenFlow compatible with existing vendors and hardware

Multi-Tenant Data Center (MTDC)

- A data center that rents physical space to multiple customers in a shared building
- Logistical services (physical security, power management, cooling) are available
- Size of the multi-tenant data center market worldwide in 2021 is projected to USD 26 billion according to Multi-tenant data center market by segment worldwide 2013-2021 | Statista

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

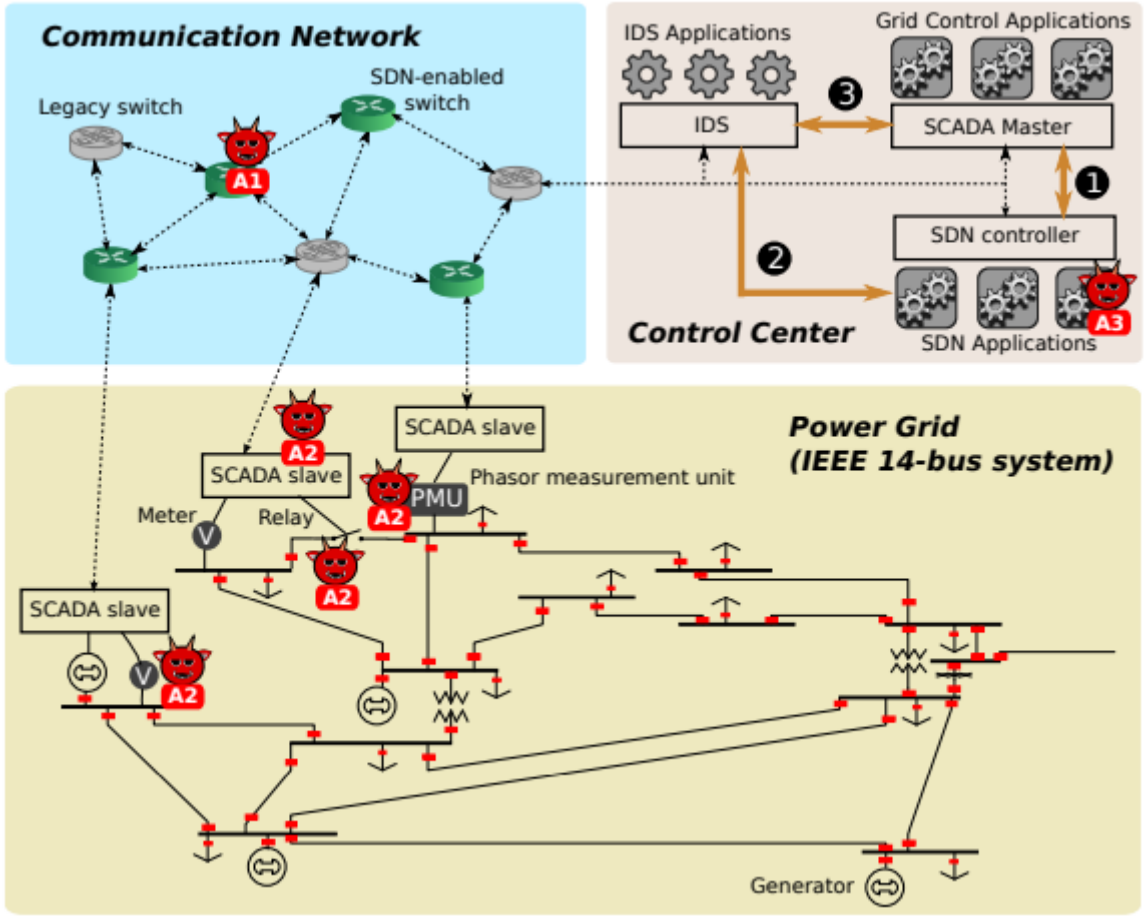
Datacenter's network hypervisor compiles these virtual topologies into set of switch configurations

- 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 capability to virtualize the network***

SDN for Smart Grid Resilience



Xinshu Dong, Hui Lin, Rui Tan, Ravishankar K. Iyer, Zbigniew Kalbarczyk, Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges, https://www.ideals.illinois.edu/bitstream/handle/2142/90446/UILU-ENG-15-2203.pdf

Opportunities for SDN to Enhance Smart Grid Resilience

- We can use SDN to establish dynamic routes from a control center to grid devices, right before the commands are to be transmitted.
  - Reduce the time window in which the attacker can inject malicious commands.
- We can use SDN to reset switches or recompute the routing if compromised switches are detected, to maintain grid control quality.
- We can use SDN to hot-swap certain grid communication channels from grid-owned communication networks to the public Internet with sufficient encryption, in the presence of attacks in the grid-owned networks.
- SDN can significantly raise the bar for attacks to be successful and provide fast network recovery for sustainable grid operations in the presence of attacks

What are the Security Risks

- What is unique when deploying SDN to smart grid compared to a standard network
  - Rerouting of a sensor/control data flow using a long-latency path may be valid from a pure networking perspective, but may reduce the operational quality of grid control systems.
- A compromised SDN controller could issue malicious SDN control messages to undermine network performance
  - We can examine each outgoing SDN control message by predicting its potential impact on the network and the grid.

Future Trend

Software-defined networking: Just the beginning

Over the last few years, SDN has offered a big step forward in enabling networkwide automation. SDN allows networking teams to manage networks as end-to-end systems, making management more efficient and flexible by separating the control and forwarding planes.

As a result, the control plane is directly programmable. It abstracts the underlying devices and infrastructure from applications and network services. Network intelligence is logically centralized through programmable SDN controllers.

SDN was initially introduced to simplify complex data center environments that needed to support portable, dynamic workload migrations and server-to-server traffic. The same principles underlie software-defined access (SD-Access), which helps secure user and device access more effectively, and software-defined WAN (SD-WAN), which can enable better user

Intent-based networking: Closing the loop

The primary objective of network teams is to continuously deliver application and service performance and protection for the business. So while SDN offers important advances in automation, it is only part of the solution. Organizations also need continuous network monitoring and optimization to support increasingly dynamic and digitally driven business models.

To achieve this, networks must understand the changing intent of the business and monitor dynamic network conditions so they can continuously accommodate that intent. According to an Internet Engineering Task Force (IETF) draft, "Intent constitutes declarative policy with a networkwide scope. A human operator defines 'what' is expected, and the network computes a solution meeting the requirements."[16]

Intent-based networking is a relatively new networking model that was first introduced to the market in 2017 and has since been adopted broadly by the networking industry.

Intent-Based Networking

- Intent-based networking (IBN) is an emerging technology that aims to apply a deeper level of intelligence and intended state insights to networking.
- Administrators can send a request to tell the network what outcome they want (their intent), while they do not need to code and execute individual tasks manually
- You give an IBN system your high-level business policies and it automatically configures the network to implement them.
- IBNs continuously monitor the state of the network to ensure your policies are being enforced, gather data and learn to adapt to your organization’s changing business demands
- Thanks to advances in machine learning and network automation, IBN systems will soon be a reality

Examples

- Build out services across a network with minimal human interaction. The system receives the administrator's intent to build a service with specific characteristics from point A to point B across a network with hundreds of distributed devices. The IBN calculates the possible paths for delivering on the requirements. Once a primary path and a fallback path are identified, the network takes responsibility for programming each of the network devices on each path to deliver the service. If the primary network path fails, the system automatically reprograms the devices to use the fallback route, without manually programming each of the devices.

- A user at a given location routinely downloads small amounts of information from a corporate server onto their laptop. The network establishes this as a baseline behavior for that user. If the same user using the same laptop from the same location now starts downloading gigabytes of data to an external server, the system perceives this as a threat. It possibly alerts the system administrator and quarantines the user from transmitting any additional data

Key Features

- Intent: IBNs understand commands from system administrators and translate them into actions. Administrators can define high-level business policies and IBNs translate those commands into actions. They validate whether a network policy can be implemented and then proactively manage resources to enforce the policy.
- Awareness: IBNs are constantly gathering data to monitor the state of the network, including traffic logs and streaming telemetry. They continuously assess the state of the network and determine the best way to implement the desired state. IBNs identify potential problems, react to changing network conditions, and take corrective action in real-time to solve issues without manual intervention.
- Context: IBNs gather information about who users are, where they are located, what resources they access and establishes baseline behavior. When the system perceives behavior outside the norm the administrator is notified which app, user, and the device is impacted or is going to be impacted. It learns over time, becomes more predictive, and provides suggestions for remediating potential problems before they impact users. IBNs use what they learn to adapt over time and customize experiences for specific users, improve network performance, and strengthen network defense against misuse and attacks.

IBN v.s. SDN

- SDN moves the focus of network infrastructure from hardware to software, from configurations to policies. It allows for more network programmability, improved automation, and reduced costs.
- IBN takes networking strategy to a higher level by combining automation with intelligence
- IBN = SDN + AI

Further Reading

- Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, Software-Defined Networking: A Comprehensive Survey, in Proceedings of the IEEE, Vol. 103, No. 1, pp 14-76, Jan. 2015, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6994333
- Xinshu Dong, Hui Lin, Rui Tan, Ravishankar K. Iyer, Zbigniew Kalbarczyk, Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges, https://www.ideals.illinois.edu/bitstream/handle/2142/90446/UILU-ENG-15-2203.pdf
- 2020 Global Networking Trends Report of Cisco, GLBL-ENG_NB-06_0_NA_RPT_PDF_MOFU-no-NetworkingTrendsReport-NB_rpten018612_5.pdf (cisco.com)

Summary

- Describe the basic idea of SDN
- What are the planes and layers in SDN
- Describe the deployment of SDN to Multi-Tenant Data Center
- Describe the deployment of SDN to smart grid
- Describe the basic idea of IBN