Lecture 8: Software Defined Networking

CS 234 / NetSys 210: Advanced Computer Networks

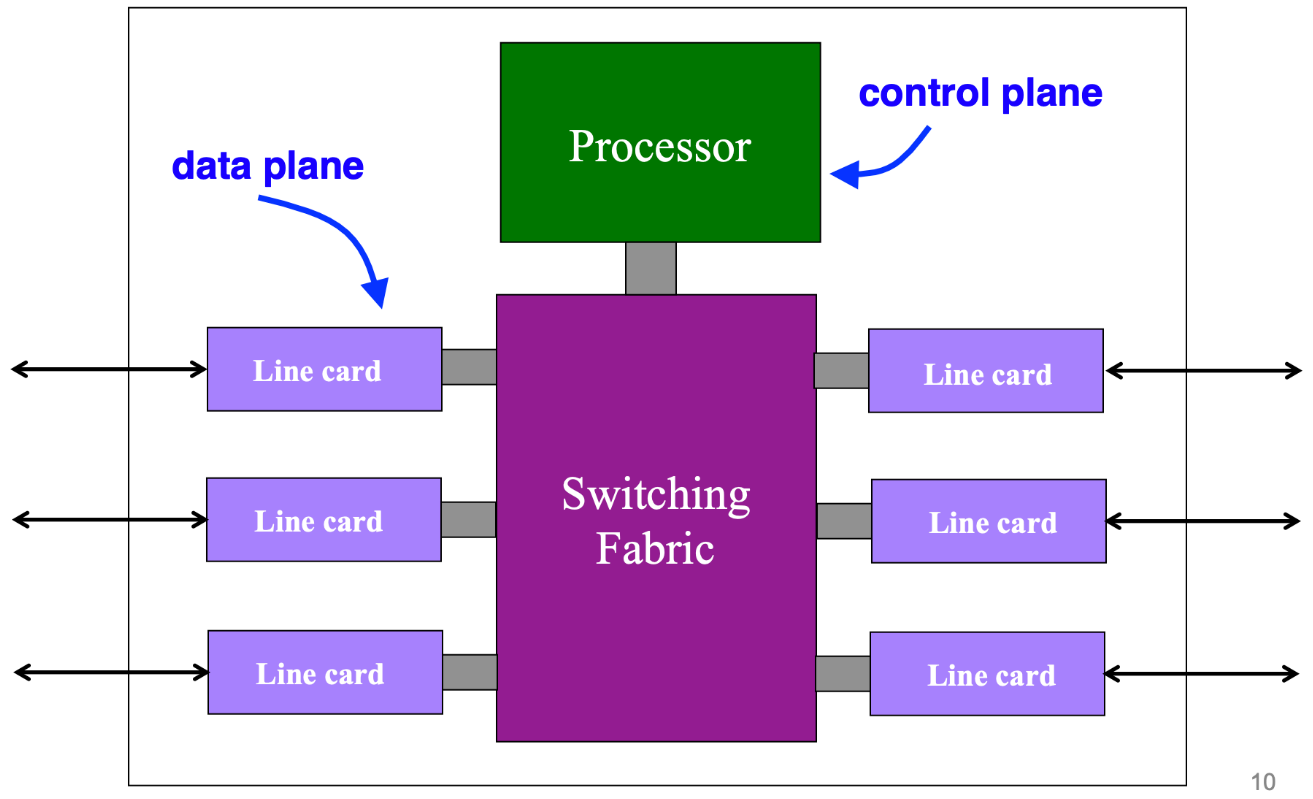
Sangeetha Abdu Jyothi

UCIRVINE

This lecture uses material from Radhika Mittal and Mohammad Alizadeh

Two Planes of Networking

- Data plane: forwarding packets
  - Based on local forwarding state
- Control plane: computing that forwarding state
  - Involves coordination with rest of system



Processor

data plane

control plane

Line card

Line card

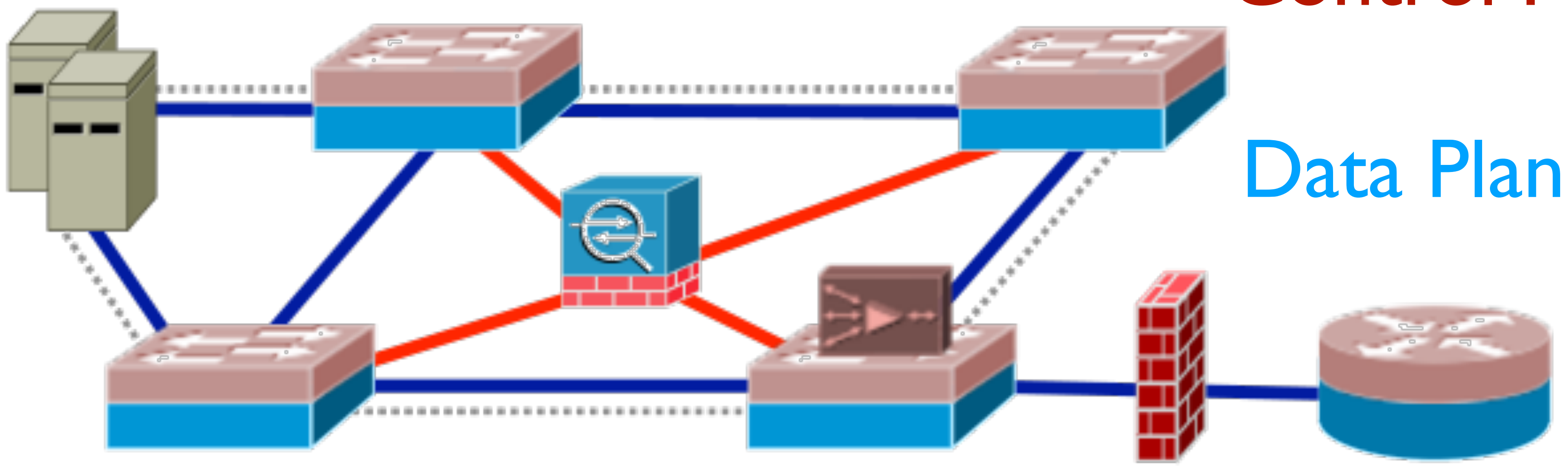
Line card

Switching Fabric

Line card

Line card

Line card



Control Plane

Data Plane

Traditional Networks

Original goals for the control plane

- Basic connectivity: route packets to destination
  - Local state computed by routing protocols
  - Globally distributed algorithms
- Interdomain policy: find policy-compliant paths
  - Done by globally distributed BGP
- For long time, these were the only relevant goals!
  What other goals are there in running a network?

Other Goals of Control Plane

- Isolation
    - Want multiple logical LANs on a physical network
- Access Control
    - Operators want to limit access to various hosts using Access Control List (ACL)
- Traffic Engineering
    - Want to avoid persistent overloads on links
    - Choose routes to spread traffic load across links
    - Two main methods
        - Setting up MPLS tunnels
        - Adjusting weights in OSPF

Network Management has many Goals

- Achieving these goals is job of the control plane which currently involves many mechanisms

- Globally distributed: Routing algorithms

- Manual/scripted configuration: ACLs, VLANs

- Centralized computation: Traffic engineering

Managing networks is extremely complicated!

- Many different control plane mechanisms

- Each designed from scratch for their intended goal

- Encompassing a wide variety of implementations (Distributed, manual, centralized,…)

- And none of them particularly well designed

Other Complications

- When running distributed algorithms and protocols
    - we need to ensure standardization and interoperability.
- When configuring individual network devices
    - Interface varies across vendors and protocols.
- Indirect control
    - Policy specification had to workaround existing routing mechanisms.

How did network operators manage?

- Until 2000s, network admins miraculously mastered the complexity!
- Then it became nearly impossible
    - Large data centers
    - Multi-tenancy
- Need for a simpler, more systematic design
- **So how do you "extract simplicity"?**
    - **Abstractions and Layering!**

Network Abstraction

- Consider the data and control planes separately

- Different tasks, so naturally different abstractions

Control Plane Task: Compute Forwarding State

- Be compatible with low-level hardware/software

  Need an abstraction for general forwarding model

- Make decisions based on entire network

  Need an abstraction for network state

- Compute configuration of each physical device

  Need an abstraction that simplifies configuration

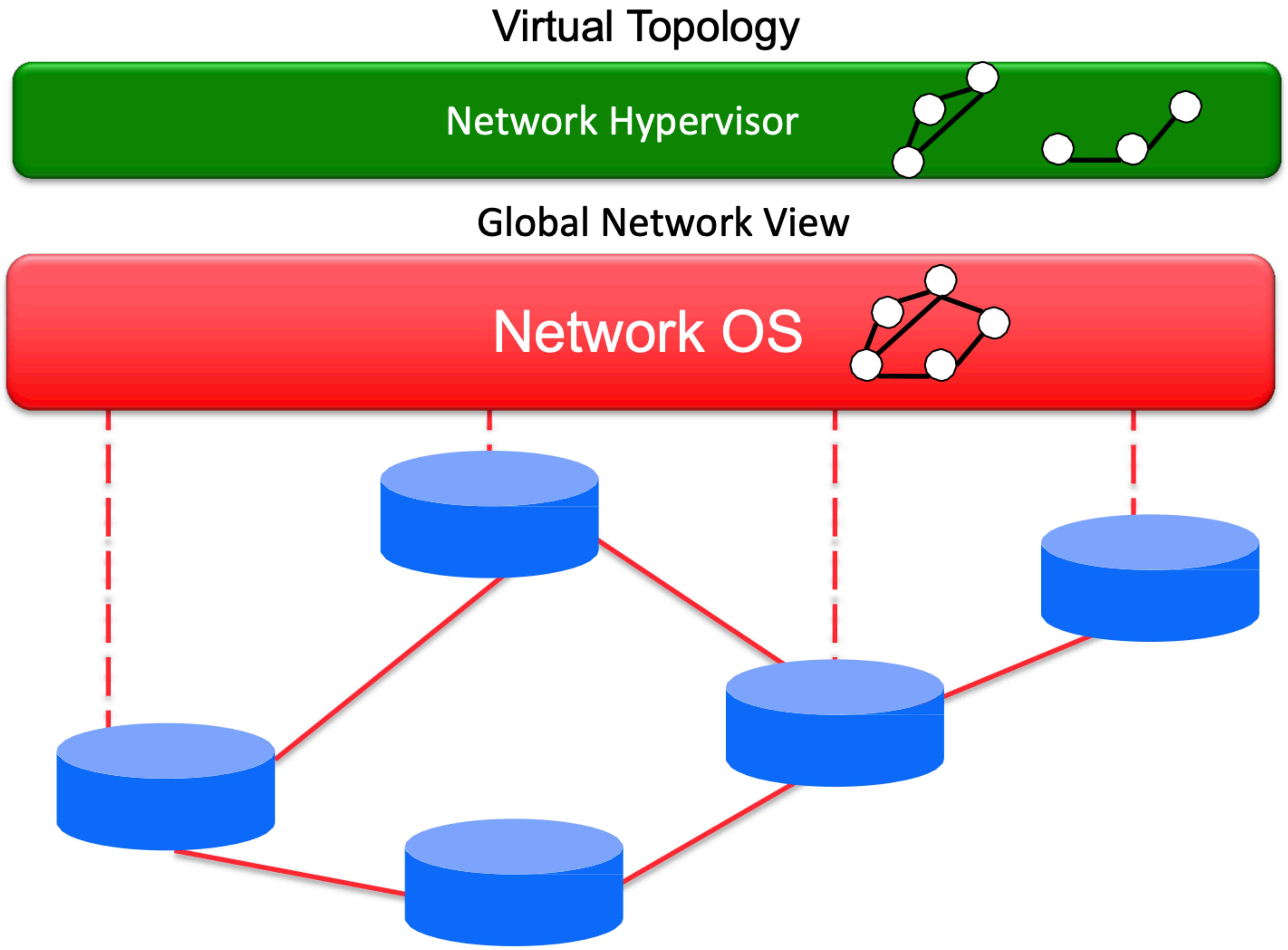
Abstraction 1: Forwarding Abstraction

- Express intent independent of implementation
    - Don’t want to deal with proprietary HW and SW
- OpenFlow is current proposal for forwarding
    - Standardized interface to switch

Abstraction 2: Network State Abstraction

- Abstract away various distributed mechanisms
- Abstraction: **global network view**
  - Annotated network graph provided through an API
- Implementation: "Network Operating System"
  - Runs on servers in network ("controllers")
  - Logically centralized
- Information flows both ways
  - Information from routers/switches to form "view"
  - Configurations to routers/switches to control forwarding

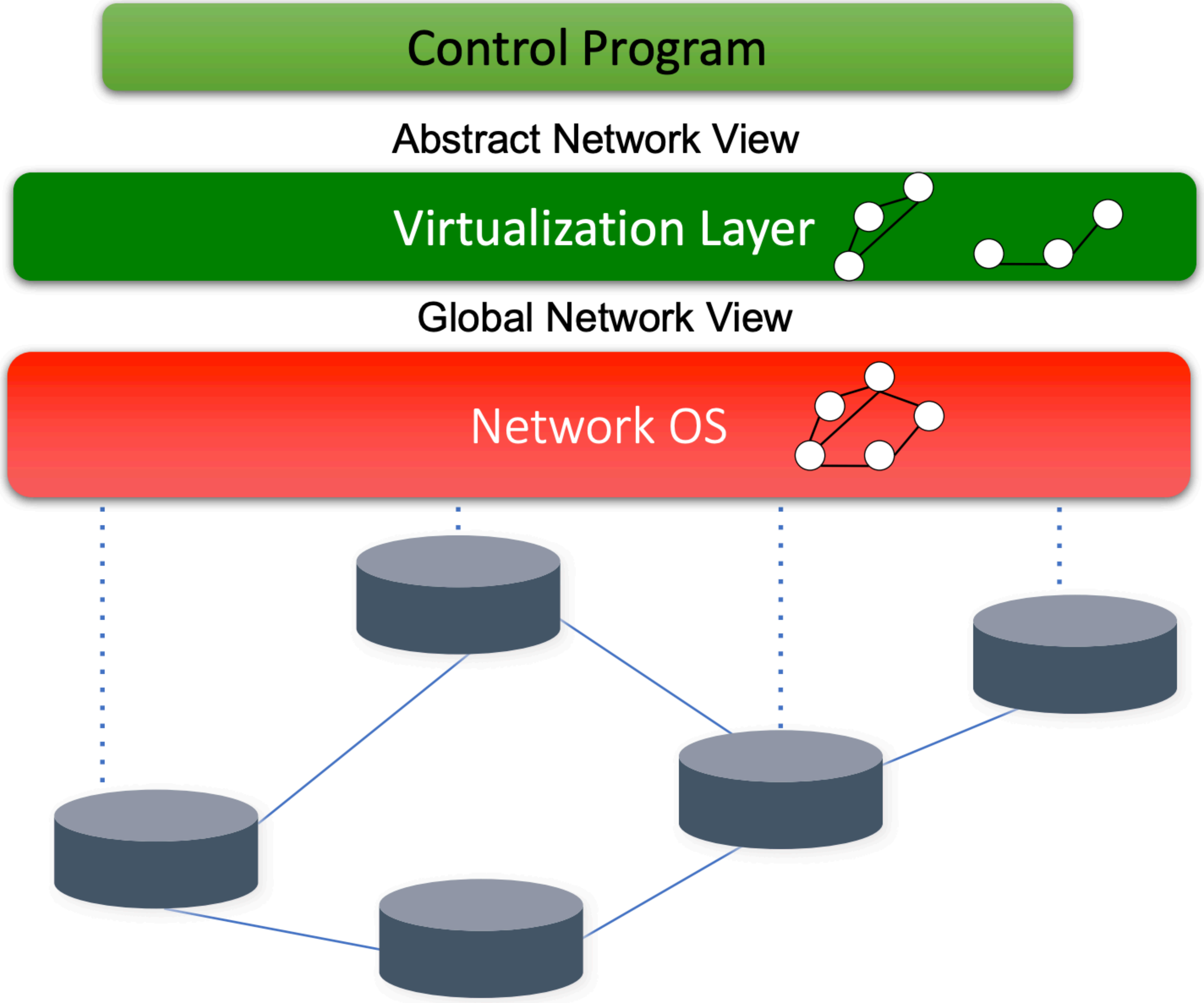
Software Defined Network



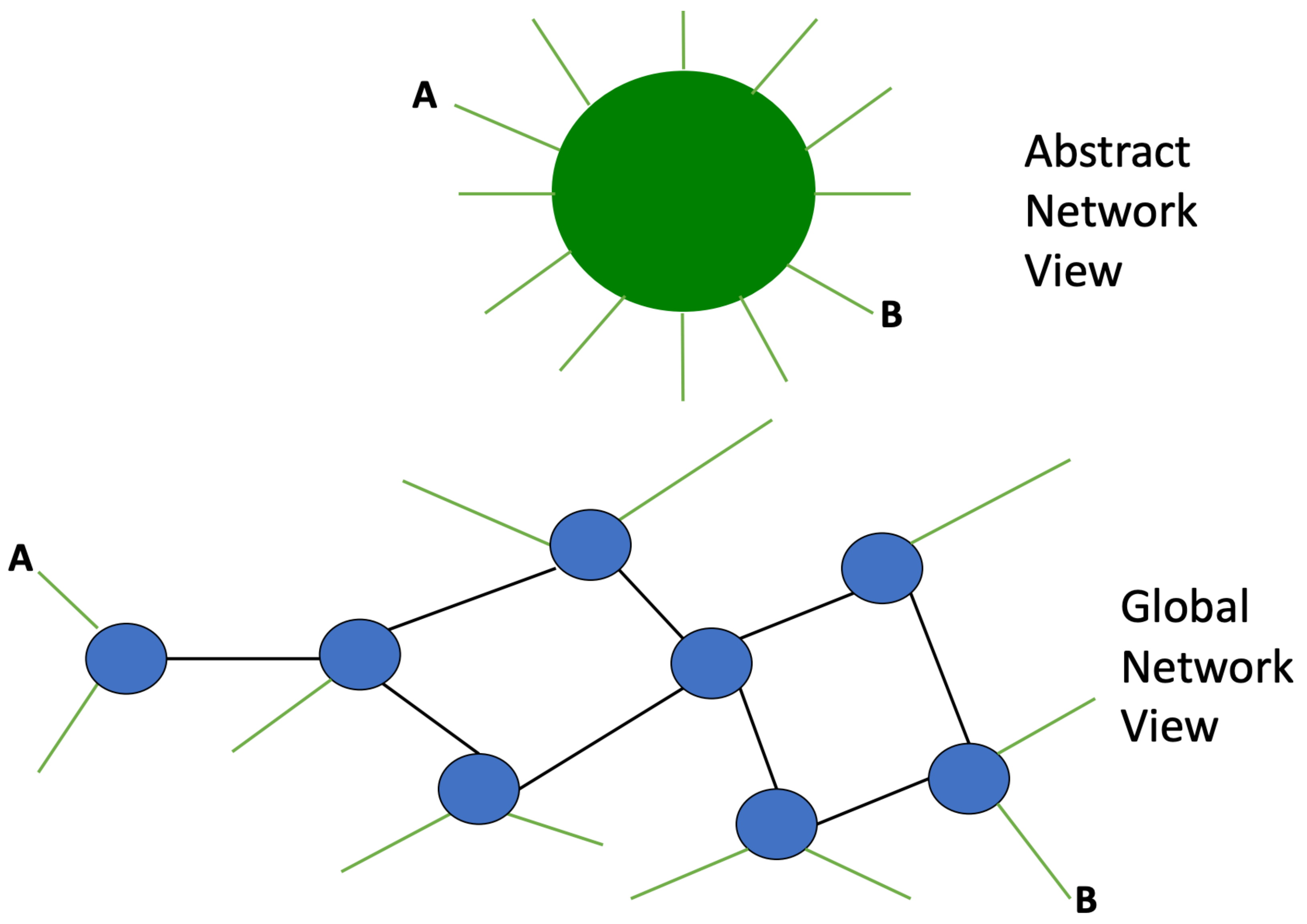
Abstraction 3: Specification Abstraction

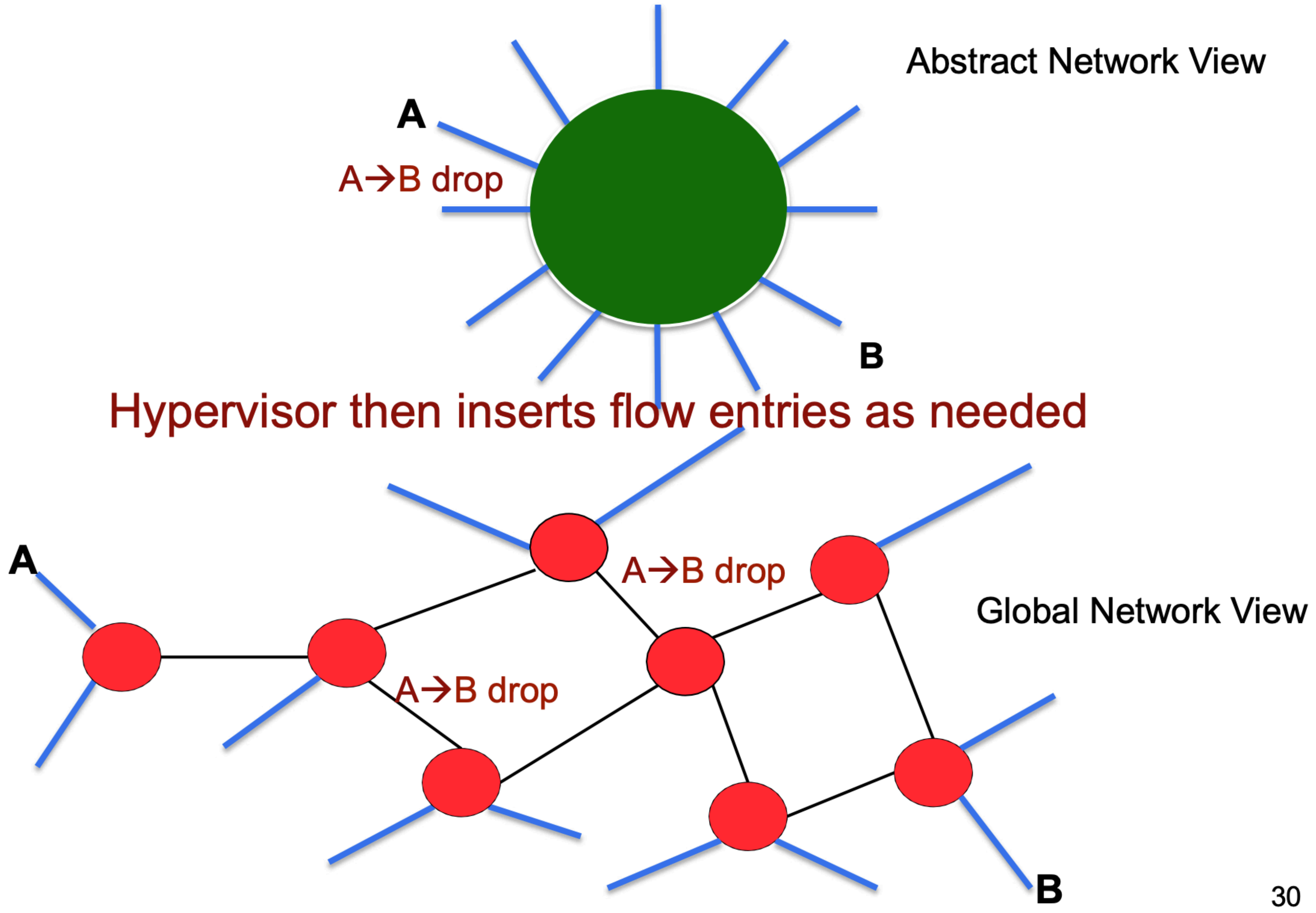
- Control mechanism expresses desired behavior
  - Whether it be isolation, access control, or QoS
- It should not be responsible for *implementing* that behavior on physical network infrastructure
  - Requires configuring the forwarding tables in each switch
- Proposed abstraction: abstract view of network
  - Abstract view models only enough detail to specify goals
  - Will depend on task semantics

SDN: Layers of the Control Plane



Simple Example: Access Control





Hypervisor then inserts flow entries as needed

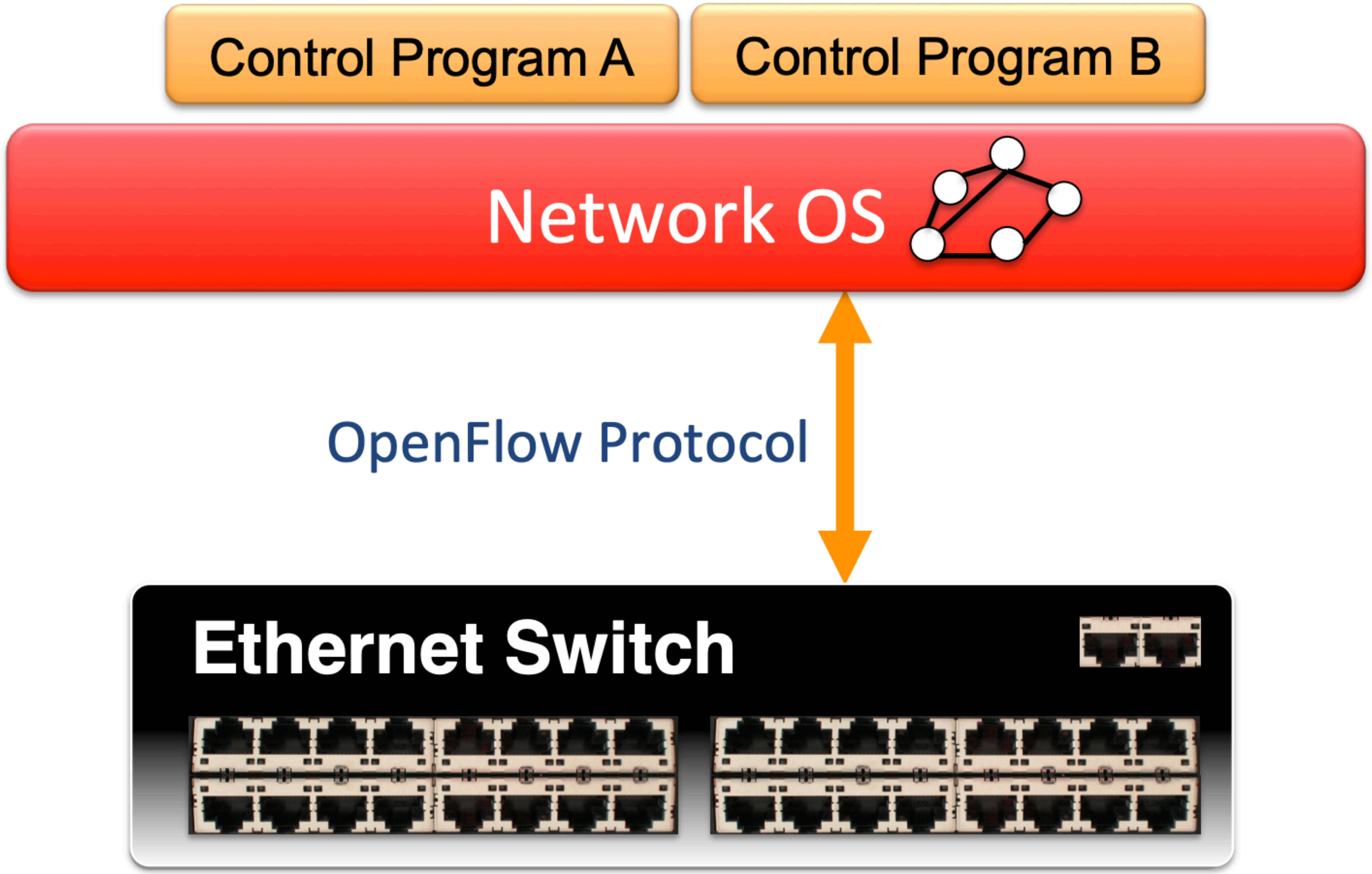
Does SDN Simplify the Network?

- 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)

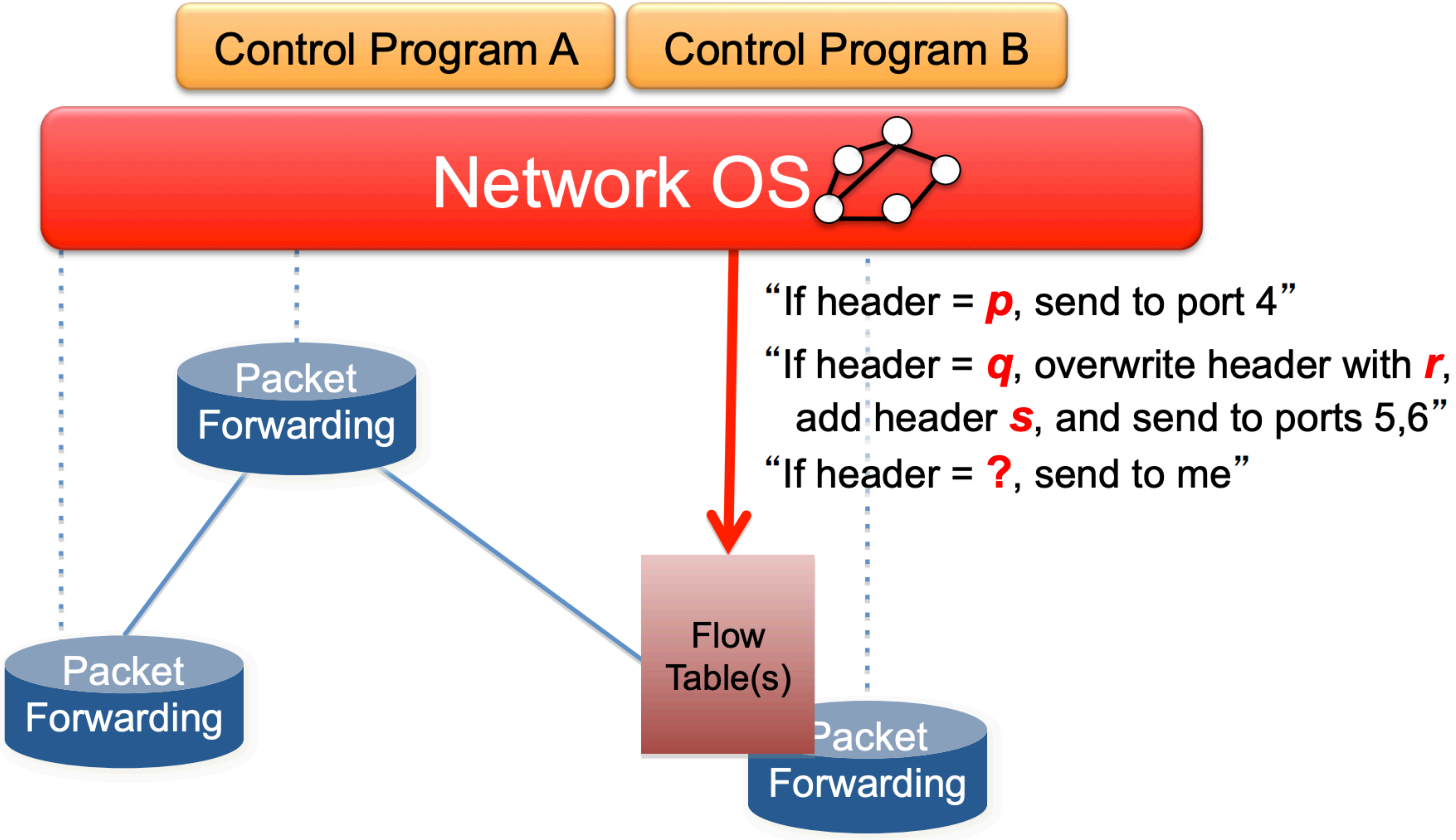
Key Ideas behind SDN

- Separate control plane from data plane
- Data plane: hardware that handles packet forwarding on individual switches
- Control plane: centralized software that remotely and directly controls switch hardware.
    - Use software design principles to modularize the control plane.

OpenFlow Basics



OpenFlow Basics



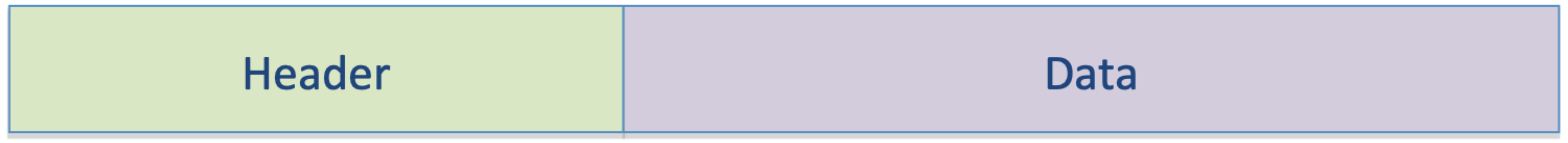
"If header = *p*, send to port 4"

"If header = *q*, overwrite header with *r*, add header *s*, and send to ports 5,6"

"If header = ?, send to me"

Primitives <Match, Action>

- Match:
  - 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



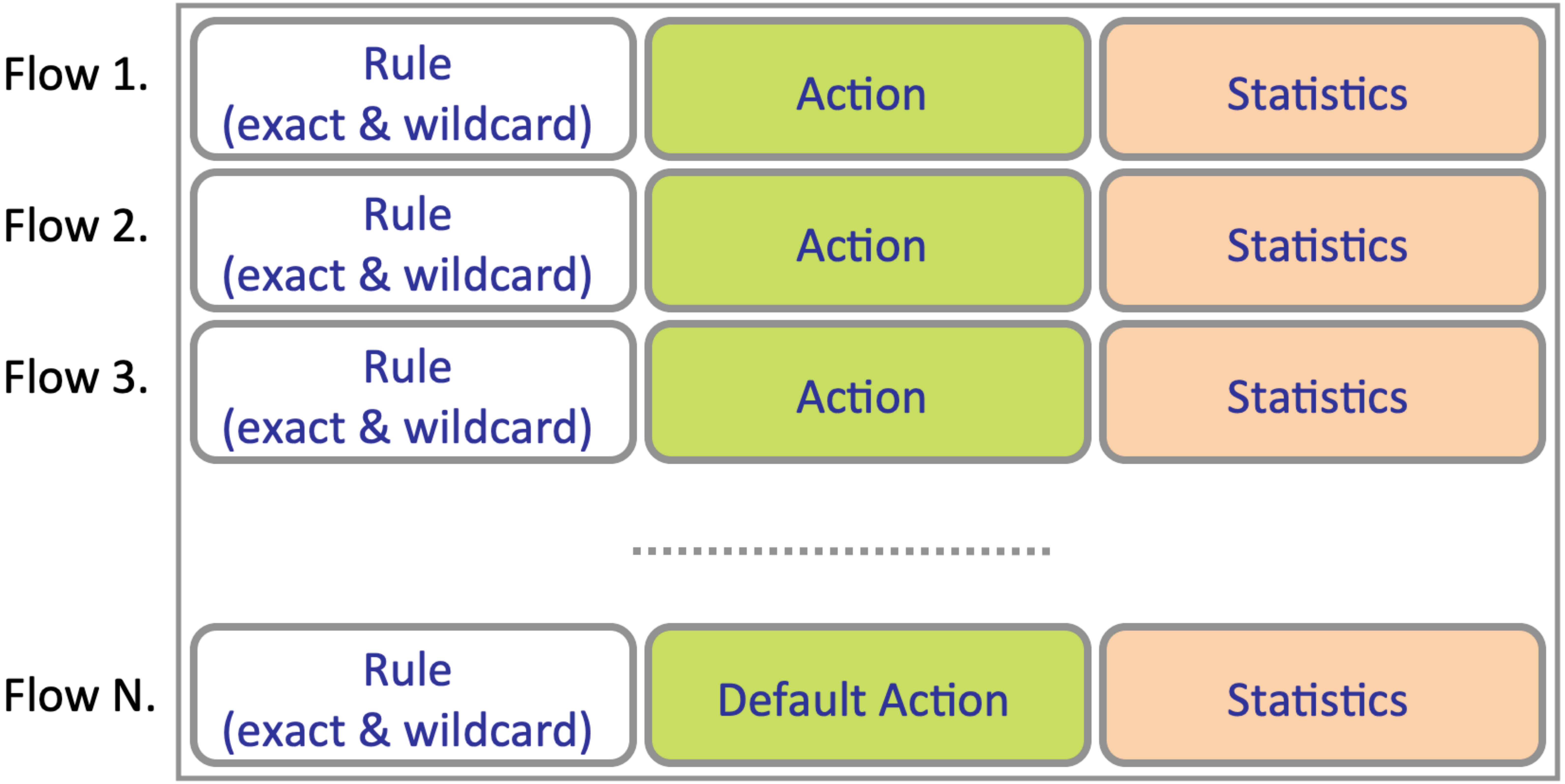
Header

Data

Match: 1000x01xx0101001x

OpenFlow Rules

Exploit the flow table in switches, routers, and chipsets



Flow 1.	Rule (exact & wildcard)	Action	Statistics
Flow 2.	Rule (exact & wildcard)	Action	Statistics
Flow 3.	Rule (exact & wildcard)	Action	Statistics
		....	
Flow N.	Rule (exact & wildcard)	Default Action	Statistics

Key Challenges with 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

Comments from students

- Security and reliability of controller - multiple students
- Networks beyond campus networks - multiple students
- “The paper does not consider how OpenFlow can interoperate with legacy devices and protocols that do not support OpenFlow.” - Krishna Teja Gettuboina
- “Real-time traffic management: OpenFlow enables dynamic network configuration, but the process of creating and deploying flow rules can be time-consuming. Future work could focus on developing real-time traffic management solutions that are more efficient and can respond to changes in traffic in real-time.” - Maganth Seetharaman
- “Integration with other networking technologies: OpenFlow can be combined with other networking technologies, such as Network Function Virtualization (NFV), to create more efficient and flexible network architectures.” - Rahul Jois

Quiz 1

- May 3
- Multiple Choice Questions and 1-2 line answer questions
- Beginning of the class: 5:05 - 5:25 pm

Thanks!