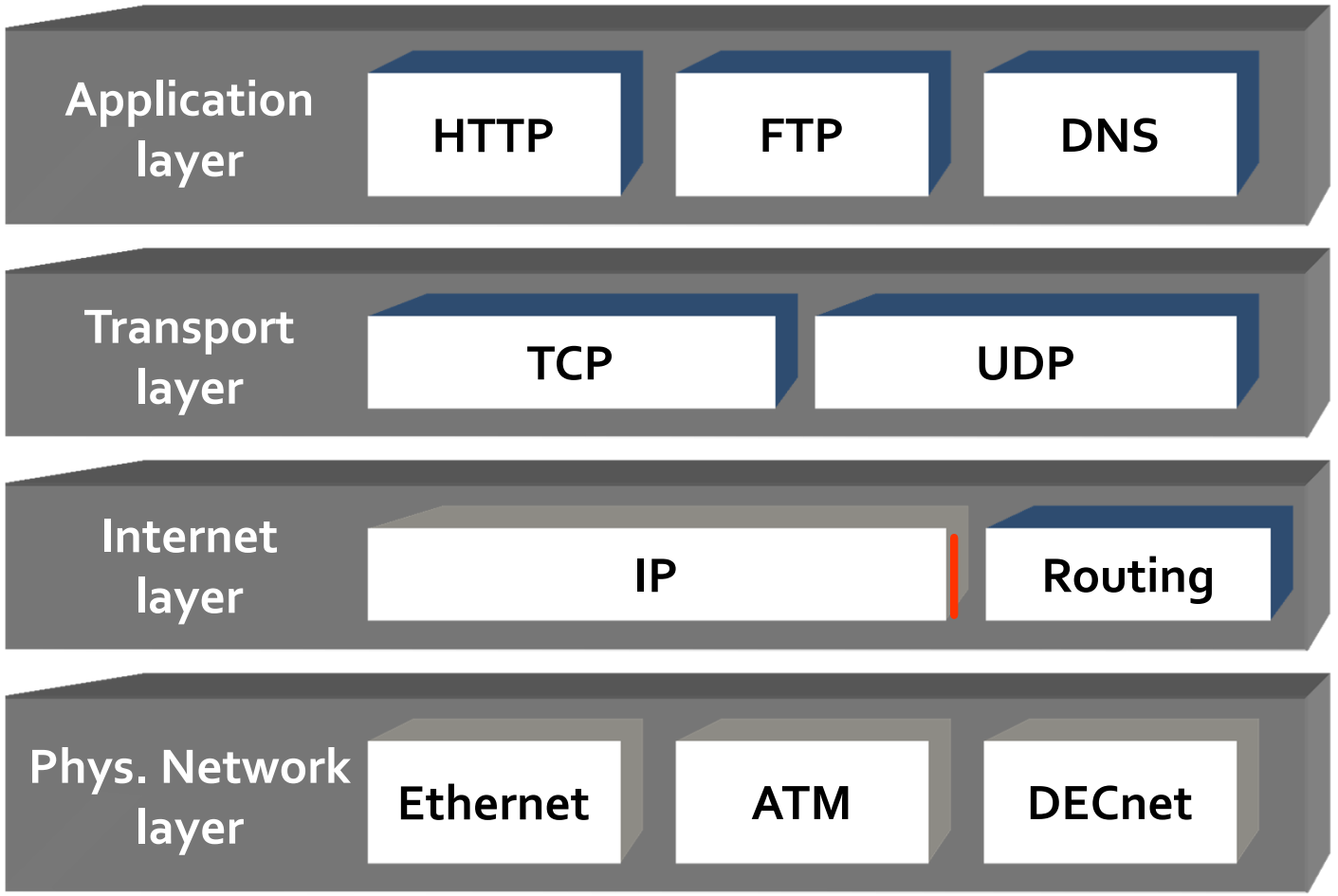
Software Defined Networks

Quick Recap of Internet Routing Architecture

IP Protocol Stack



Routing vs. forwarding

- **Routing (algorithm):**

  A successive exchange of connectivity information between routers. Each router builds its own routing table based on collected information.

- **Forwarding (process):**

  A switch- or router-*local* process which forwards packets towards the destination using the information given in the local routing table.

Routing algorithm

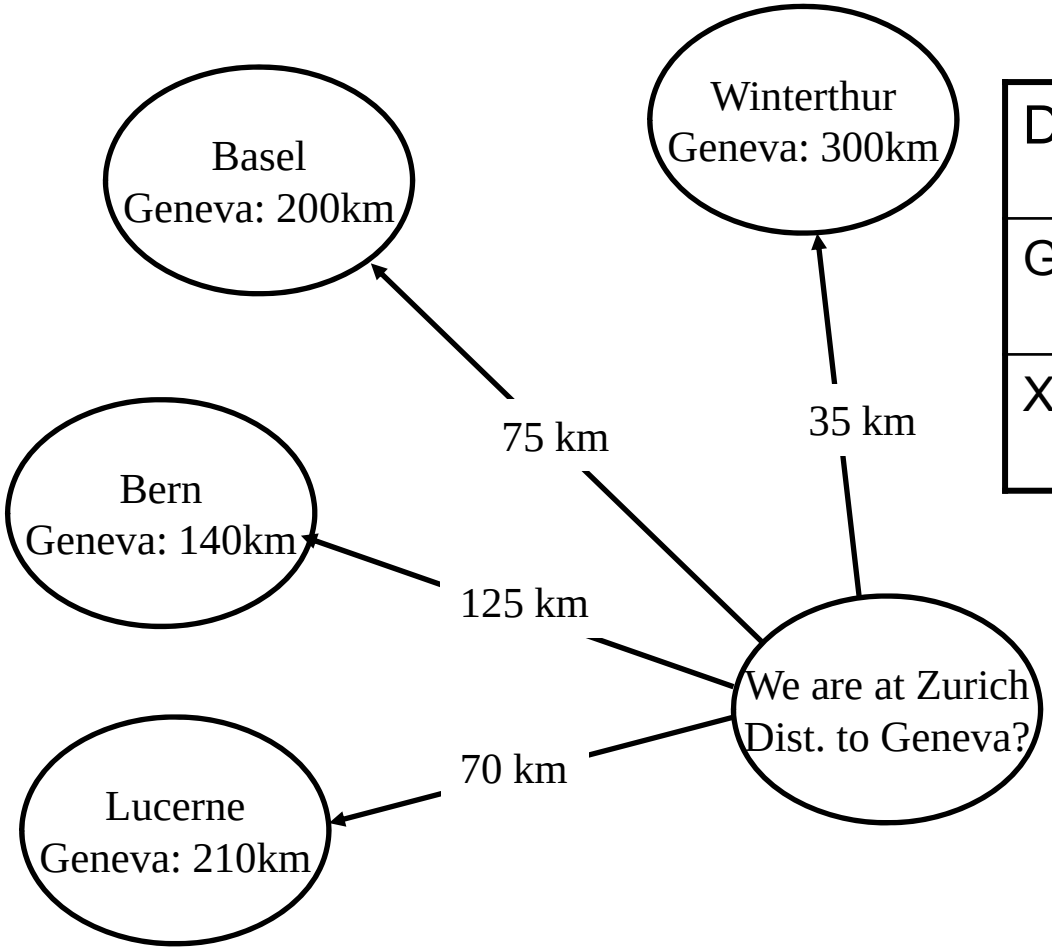
- A *distributed algorithm* executed among the routers which builds the routing tables. Path selection can be based on different metrics:
  - Quantative: #hops, bandwidth, available capacity, delay, delay jitter,…
  - Others: Policy, utilization, revenue maximization, politics,…
- Design and evaluation criteria:
  - Scalability of algorithm. How will *route information packets* (i.e. overhead) scale with an increased number of routers? Computational complexity?
  - Time to a common converged state.
  - Stability and robustness against errors and partial information
- Two important classes of routing algorithms
  - *Distance Vector* (also called Bellman-Ford or Ford-Fulkerson)
  - *Link State*

Richard Bellman: *On Routing Problem*, in Quarterly of Applied Mathematics, 16(1), pp.87-90, 1958.

Lestor R. Ford jr., D. R. Fulkerson: *Flows in Networks*, Princeton University Press, 1962.

Distance Vector Routing

Distance Vector Routing: Basic Idea



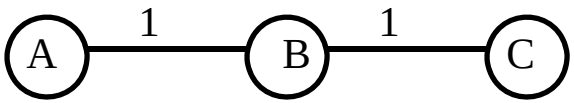
Dest	LU	BE	BS	W	opt
GE	280	265	275	335	BE
XY					

Distance Vector for GE as seen by the Zurich router: (GE, 265)

Distance Vector Routing - Description

- Each router reports a list of (directly or indirectly) *reachable destinations* and the *routing metric* (“distance vector”) to its neighbors
- Each router updates its internal tables according to the information received. If a *shorter distance* to a destination is received, this is recorded in the table.
- The distance vector is sent *periodically* or when the routing table is changed (e.g. interval 30 seconds)
- Packets containing distance vectors are called *routing updates.*

Count-to-infinity Problem

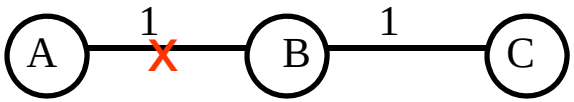


Node A		
Destination	Distance	Next node
B	1	B
C	2	B

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Noce C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

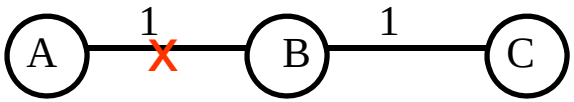


Node A		
Destination	Distance	Next node
B	1	B
C	2	B

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

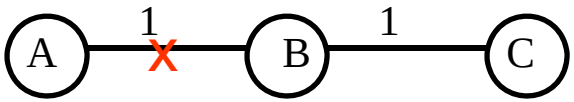


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	1	A
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

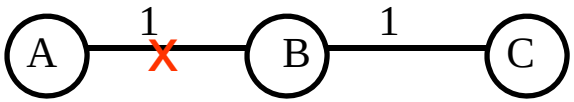


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	N.E.	
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

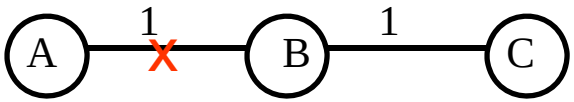


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	3	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	2	B

Count-to-infinity Problem

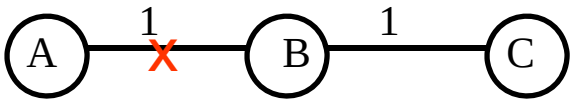


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	3	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	4	B

Count-to-infinity Problem

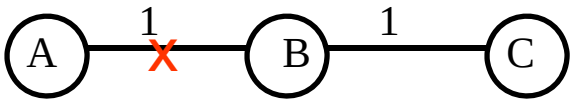


Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	5	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	4	B

Count-to-infinity Problem



Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

Node B		
Destination	Distance	Next node
A	5	C
C	1	C

Node C		
Destination	Distance	Next node
B	1	B
A	6	B

Bad news travel slow[ly]

Fixes

- Define infinity as finite
  - Maximum hop count is 15, ≥16 means infinite
- Split horizon
  - Never advertise a route out of the interface through which you learned it.
- Poison reverse
  - Advertise invalid routes as *unreachable*
- Split horizon with poison reverse
  - Once you learn of a route through an interface, advertise it as unreachable back through that same interface.
- Hold-down timer
- Report the entire path

Link State Routing

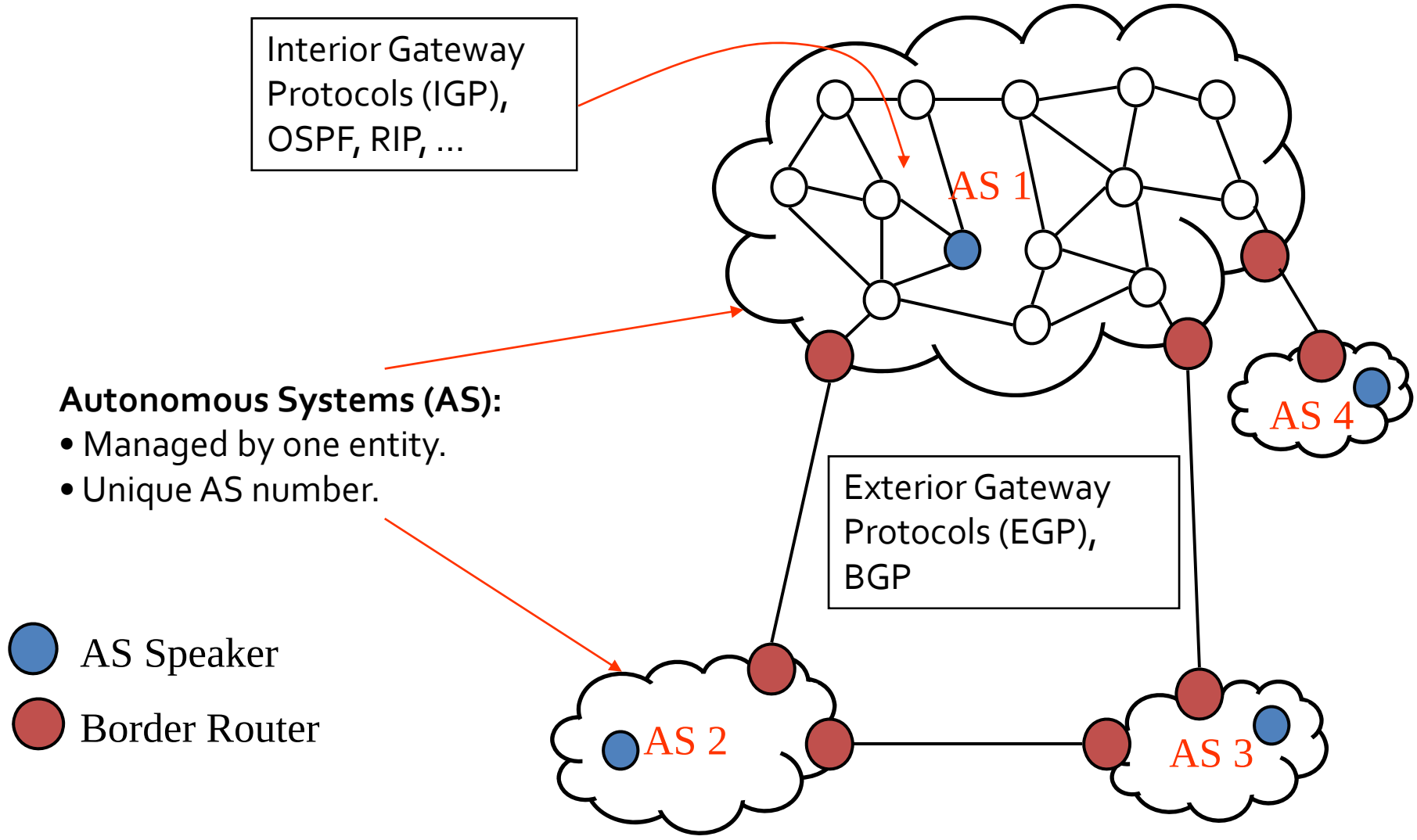
Link State Routing: Basic idea

- Each router compiles a list of *directly* connected neighbors with associated metric
- Each router participates in *flooding* these lists
- Convergence: With time, each router will get the *full topology* of the network.
- Routers compute the best route from a source (or themselves) to a destination using Dijkstra's Shortest Path First (SPF) algorithm

Motivation for *hierarchical routing*

- Scalability
  - Both algorithms (DV, LS) have poor scalability properties (memory and computational complexity).
  - DV also has some problem with number and size of routing updates.
- Administration may need more facilities, e.g.
  - Local routing policies
  - Specific metrics (hops, delay, traffic load, cost, …)
  - Medium-term traffic management
  - Different levels of trust (own routers / foreign routers)

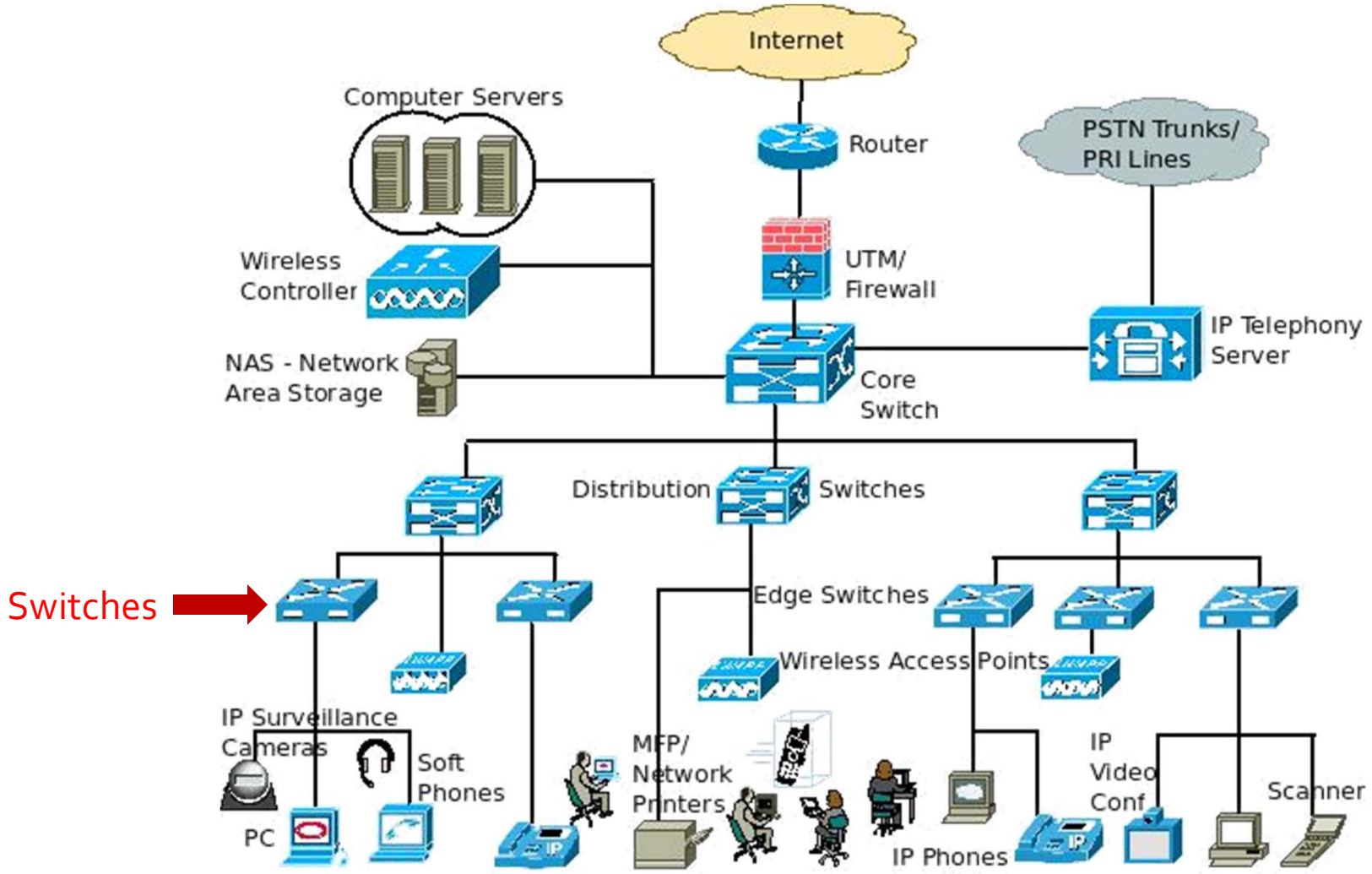
Hierarchical routing domains, AS



Internet intra-domain routing protocols

- Distance-Vector-type:
  - Routing Information Protocol (**RIP**), RFC 1058, 2453
- Link-State-type
  - Open Shortest Path First (**OSPF**), RFC 2328
  - Intermediate System-to-Intermediate System (IS-IS), an OSI protocol supported by most routers

Limitations of Current Networks



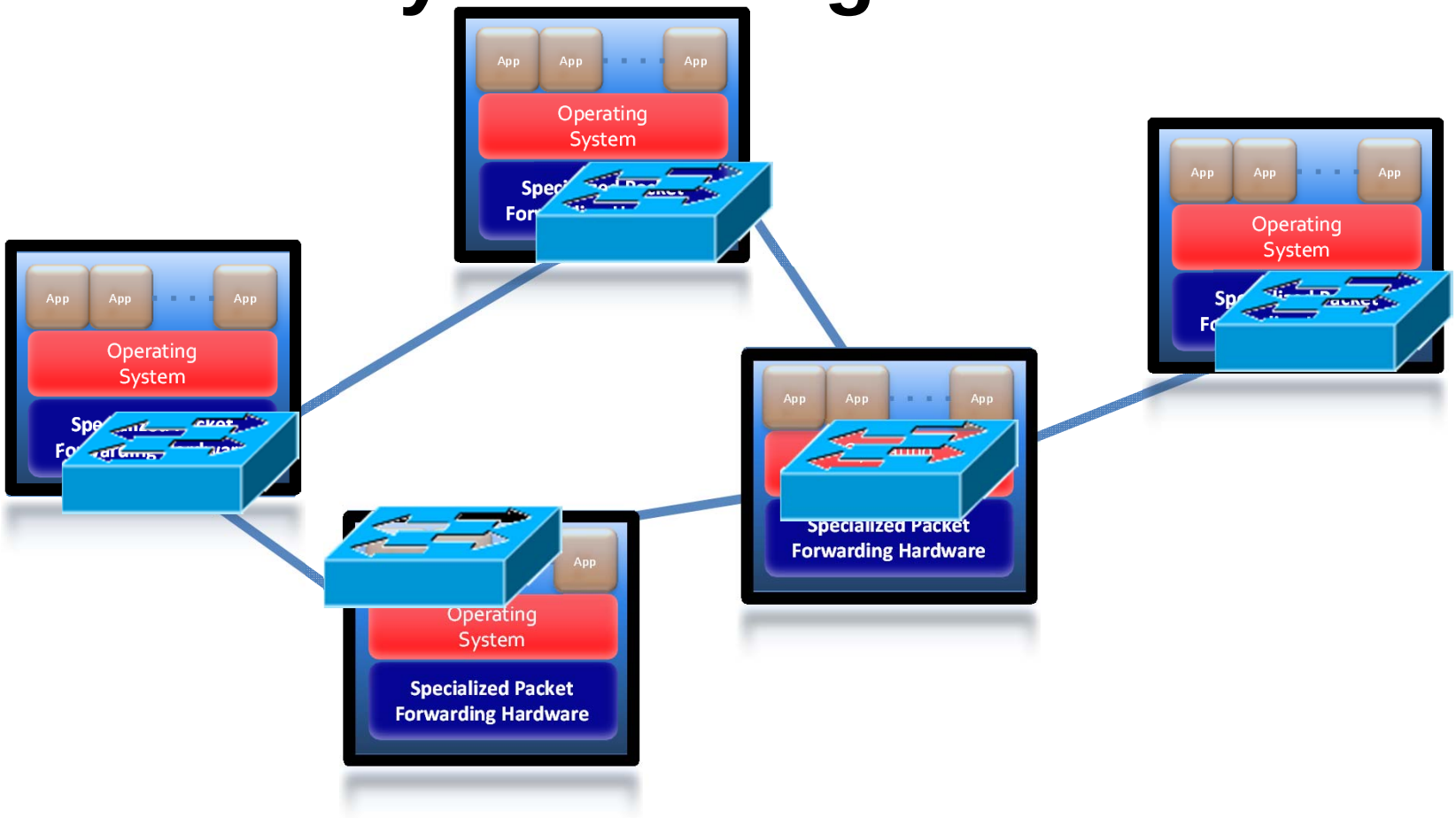
http://www.excitingip.net/27/a-basic-enterprise-lan-network-architecture-block-diagram-and-components/

Limitations of Current Networks

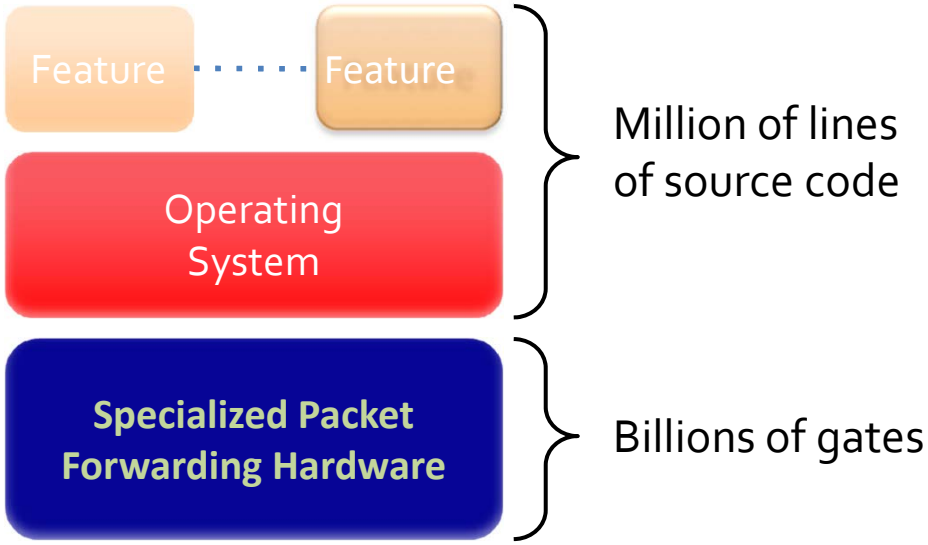
- **Enterprise networks are difficult to manage**
- **“New control requirements have arisen”:**
  - Greater scale
  - Migration of VMS
- **How to easily configure huge networks?**

Limitations of Current Networks

- **Old ways to configure a network**



Limitations of Current Networks



Many complex functions baked into infrastructure

OSPF, BGP, multicast, differentiated services, Traffic Engineering, NAT, firewalls, …

Cannot dynamically change according to network conditions

Limitations of Current Networks

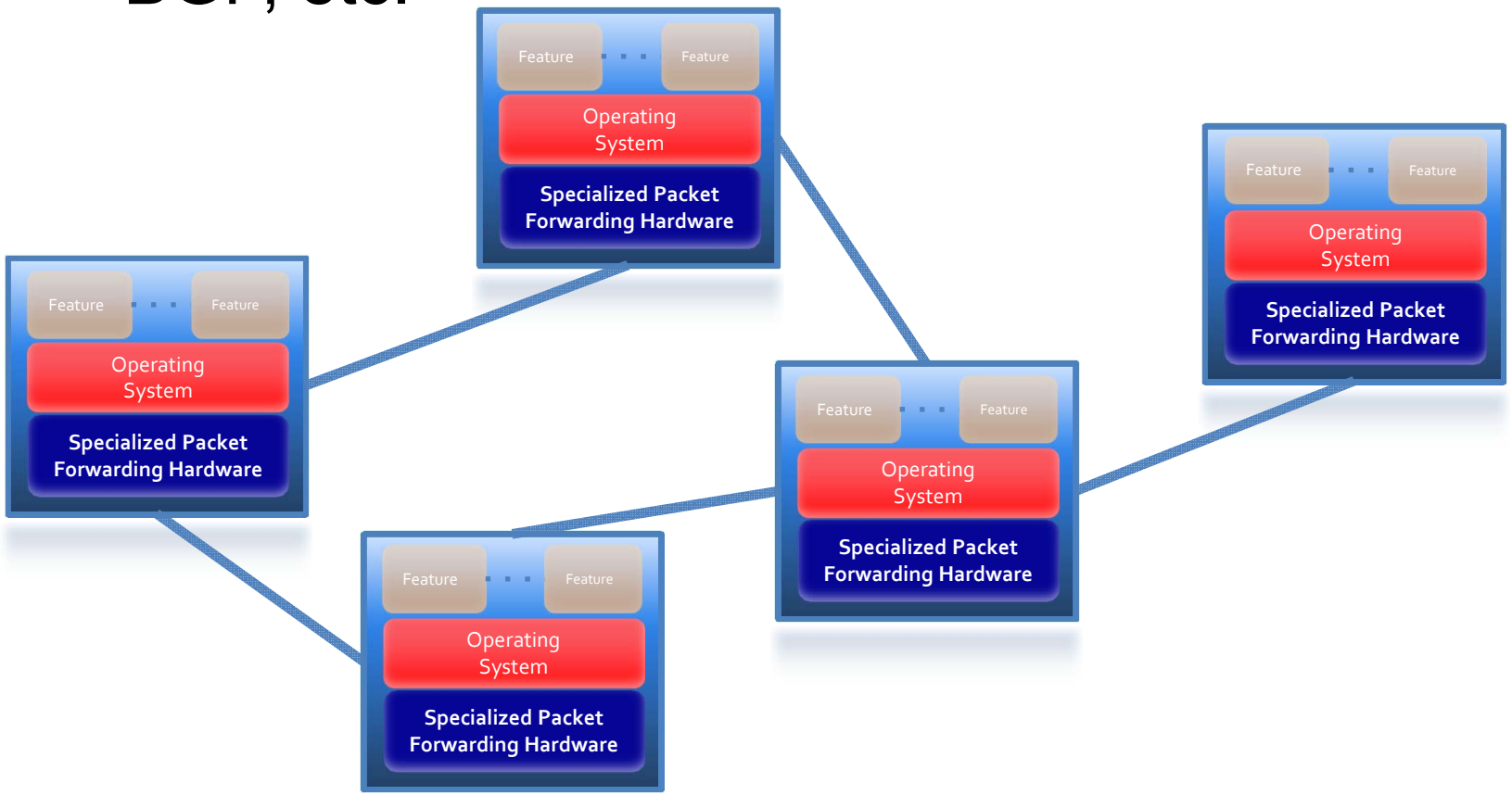
- **No control plane abstraction for the whole network!**

- **It's like ol when there was no OS…**

Wilkes with the EDSAC, 1949

Classical network architecture

- Distributed control plane
- Distributed routing protocols: OSPF, IS-IS, BGP, etc.



The Networking Industry (2007)

Routing, management, mobility management, access control, VPNs, …

Feature ····· Feature

Operating System

Specialized Packet Forwarding Hardware

Million of lines of source code — 5400 RFCs — Barrier to entry

Billions of gates — Complex — Power Hungry

Closed, vertically integrated, boated, complex, proprietary

Many complex functions baked into the infrastructure

OSPF, BGP, multicast, differentiated services,
Traffic Engineering, NAT, firewalls, MPLS, redundant layers, …

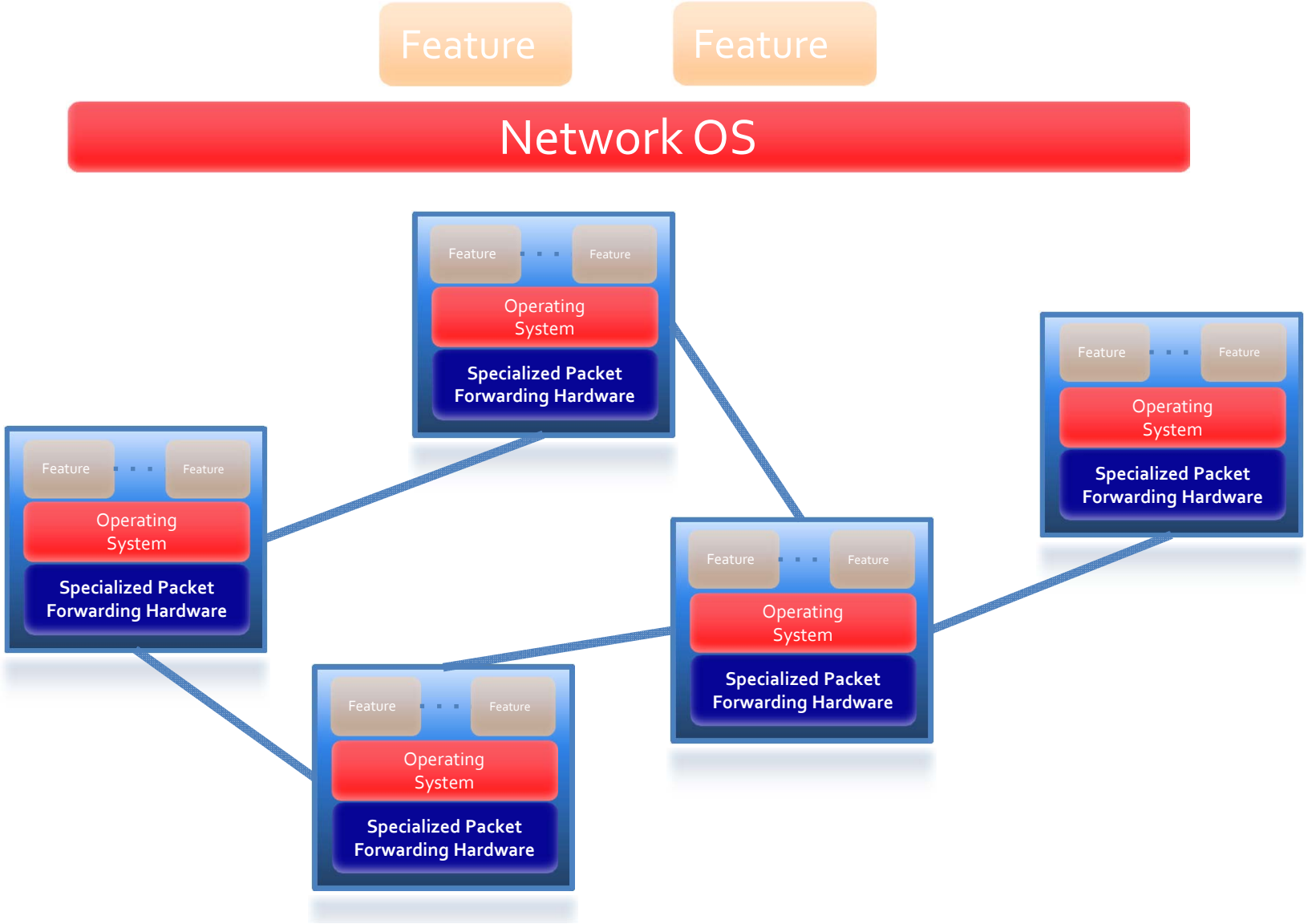
Little ability for non-telco network operators to get what they want

Functionality defined by standards, put in hardware, deployed on nodes

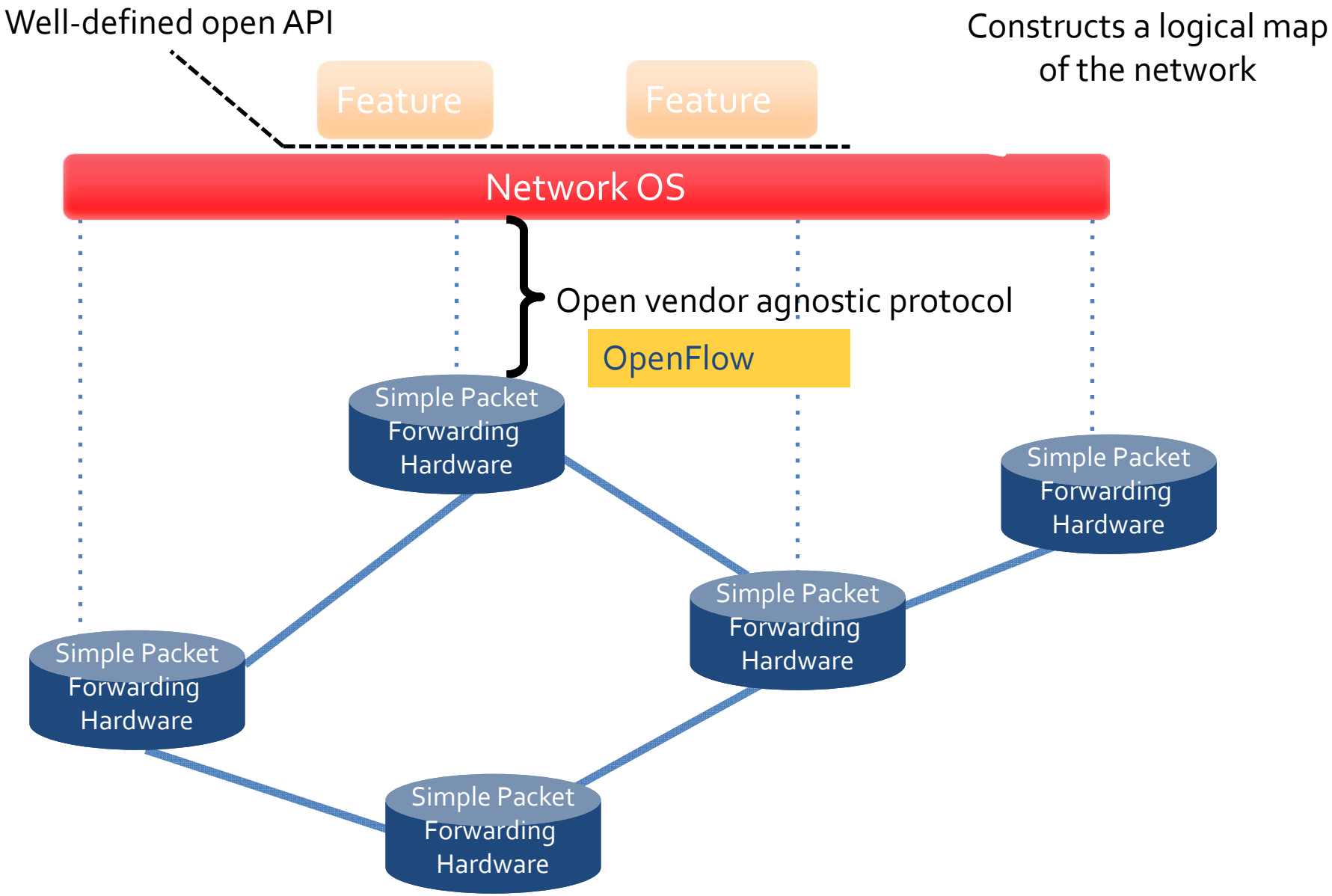
SDN

- Possible definitions:
  - SDN is a new network architecture:
    - that's makes it easier to program networks.
    - with the core idea that software remotely controls network hardware.
    - …

From Vertically Integrated to …



Software Defined Network



Network OS

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

- Runs on servers (controllers) in the network

Uses an open protocol to:

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

OpenFlow

- OpenFlow
  - is a protocol for remotely controlling the forwarding table of a switch or router
  - is one element of SDN

How does OpenFlow work?

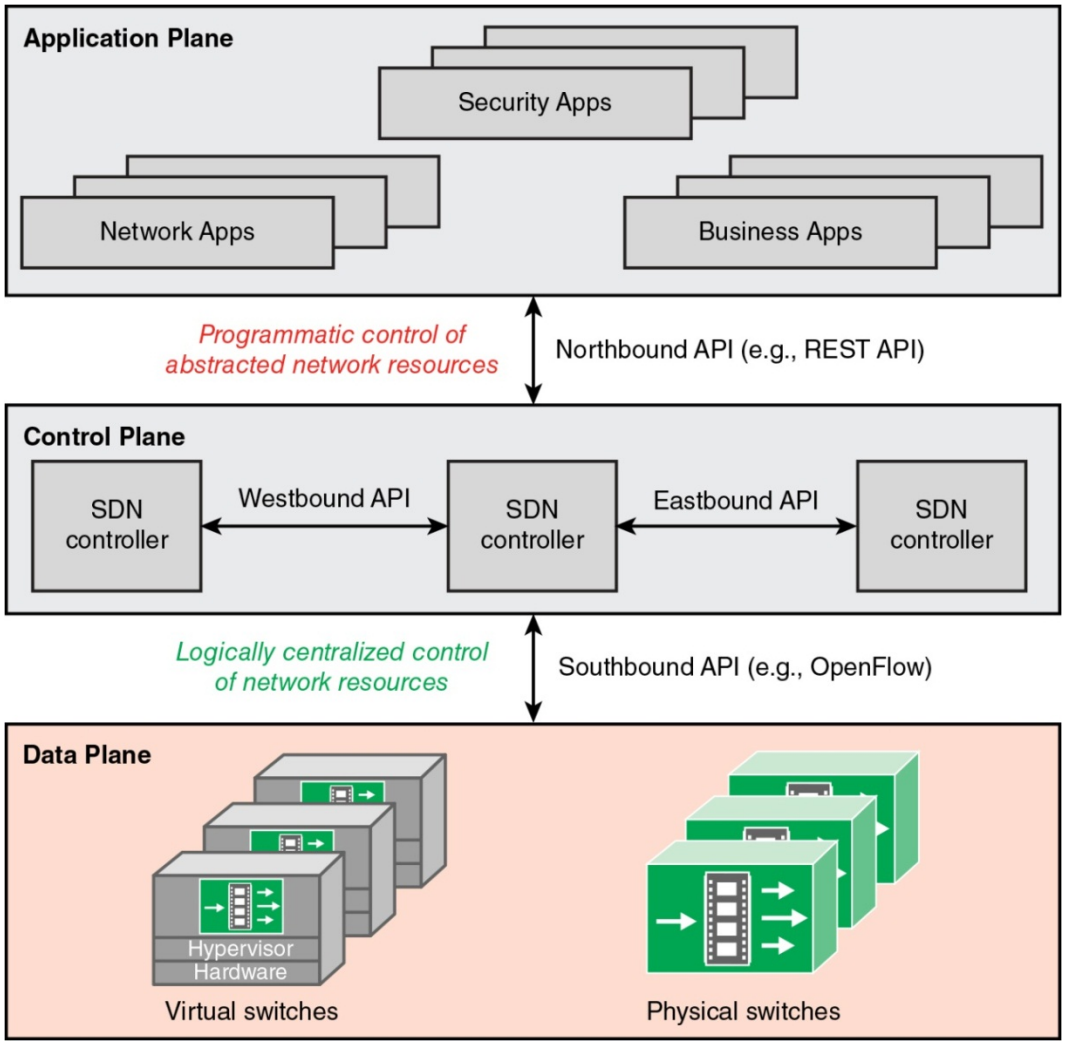


FIGURE 4.1 SDN Architecture

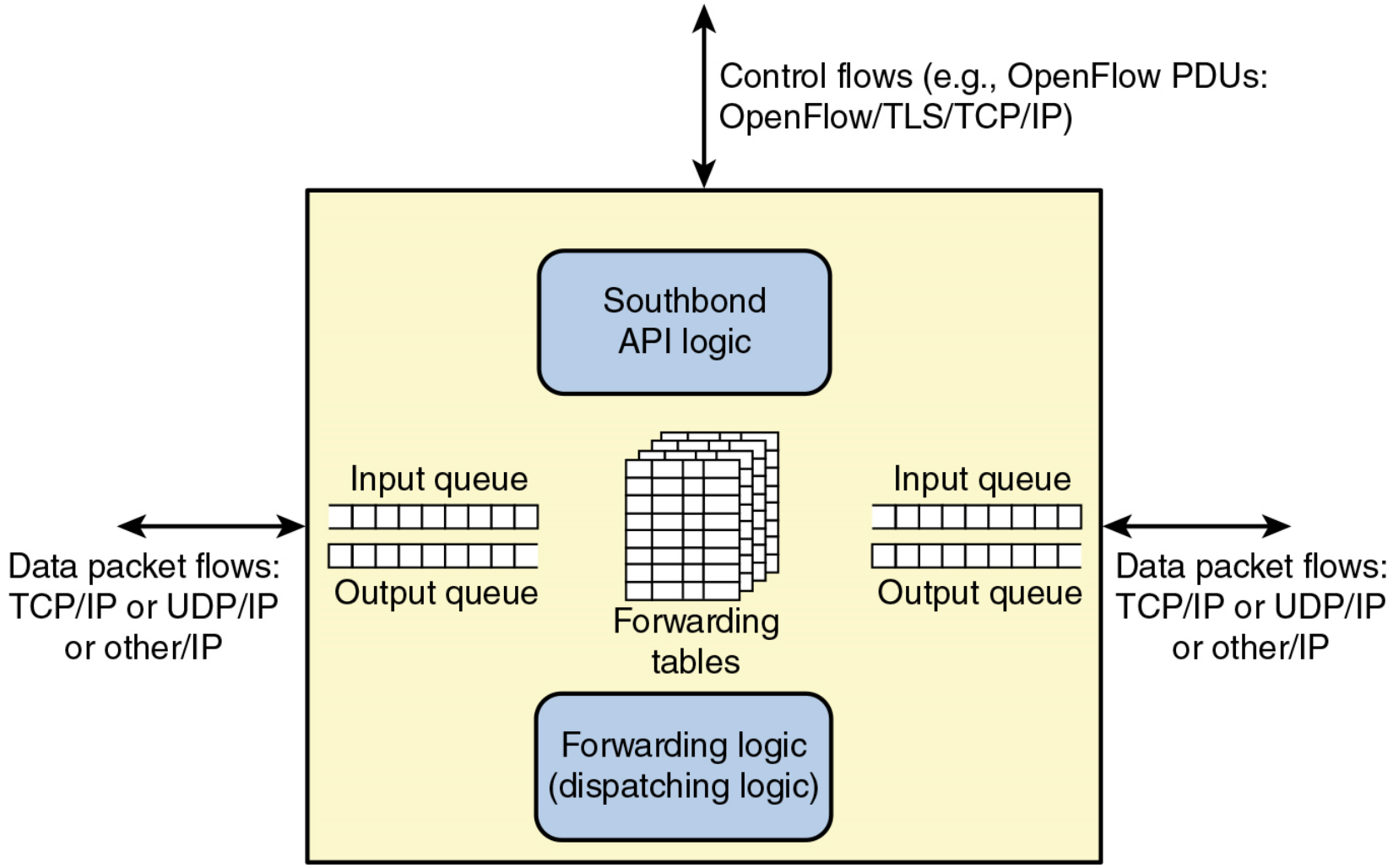
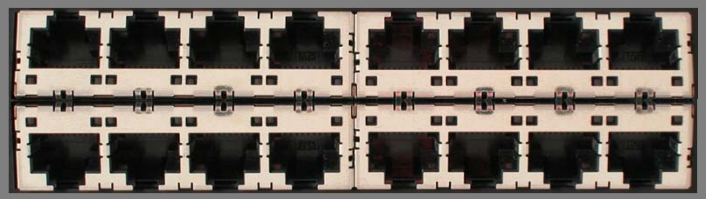
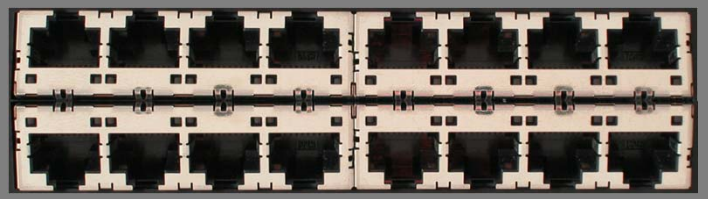
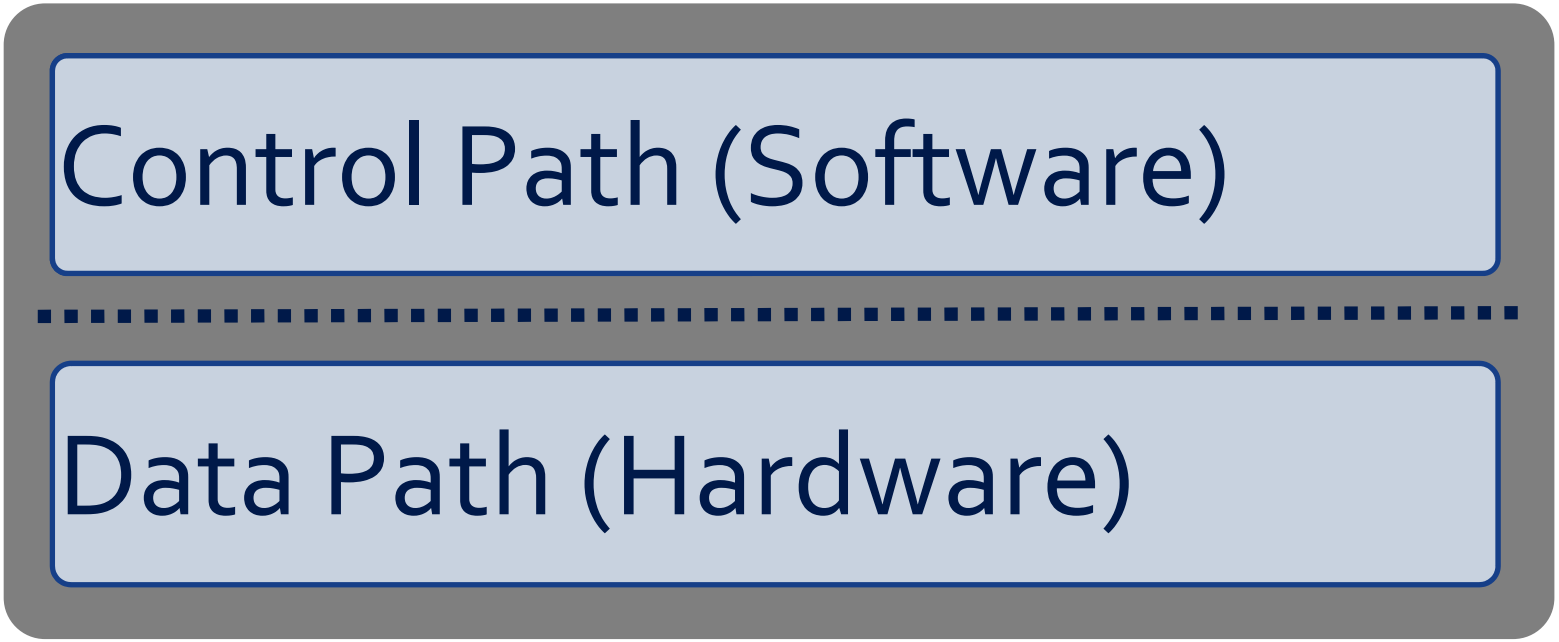


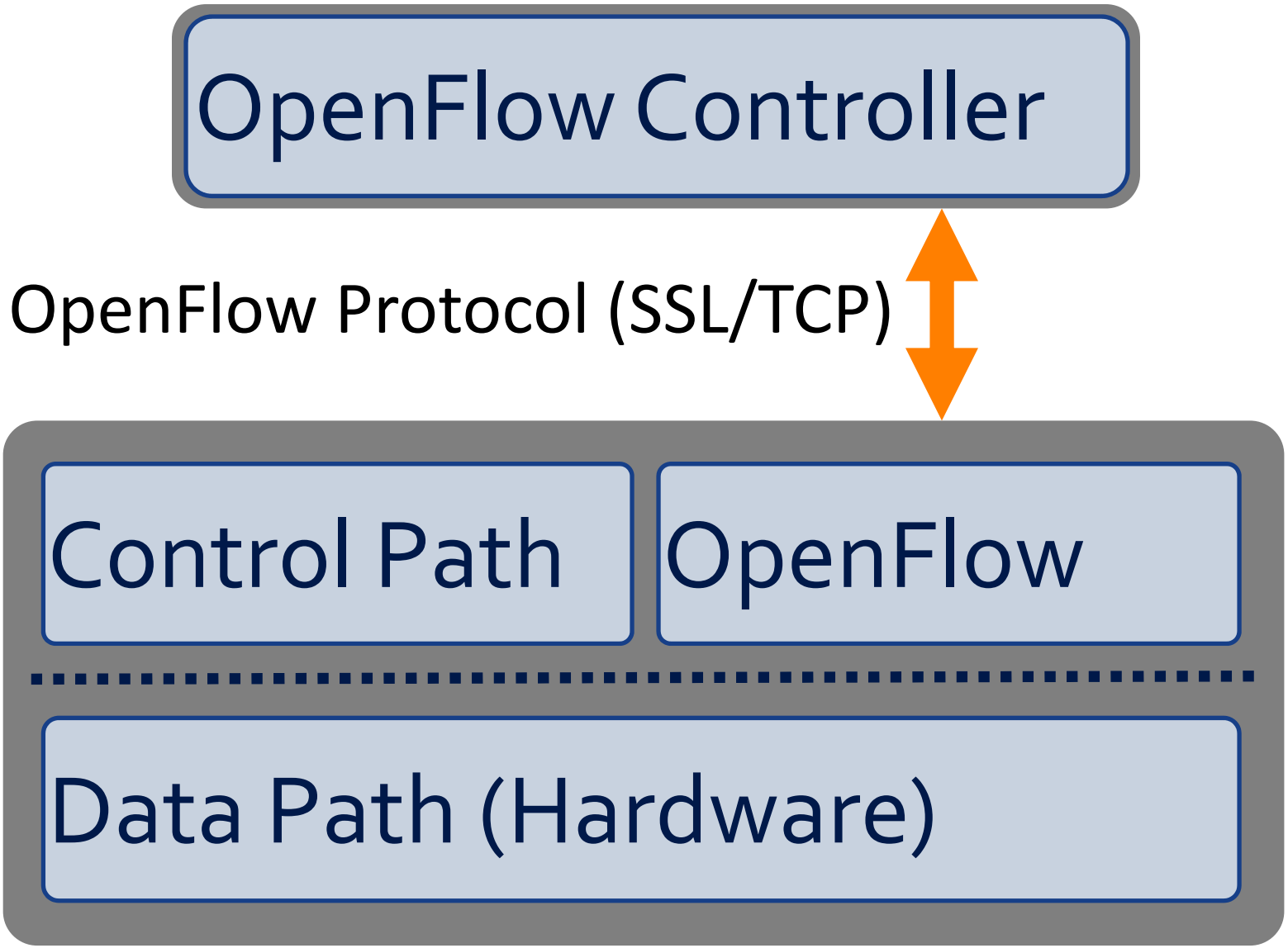
FIGURE 4.2 Data Plane Network Device

Ethernet Switch






Control Path (Software)
Data Path (Hardware)


OpenFlow Controller
OpenFlow Protocol (SSL/TCP)
Control Path
OpenFlow
Data Path (Hardware)

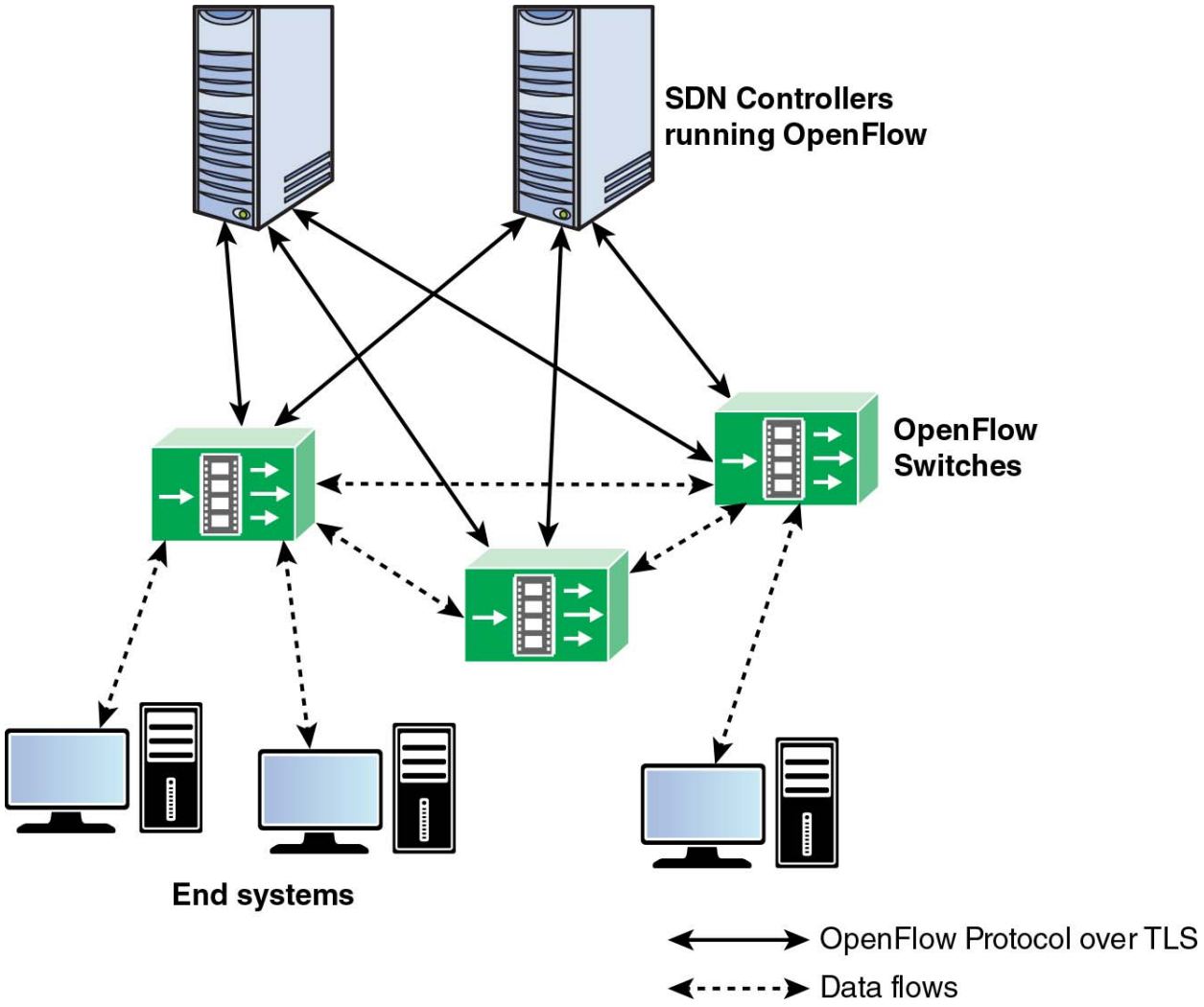


FIGURE 4.3 OpenFlow Switch Context

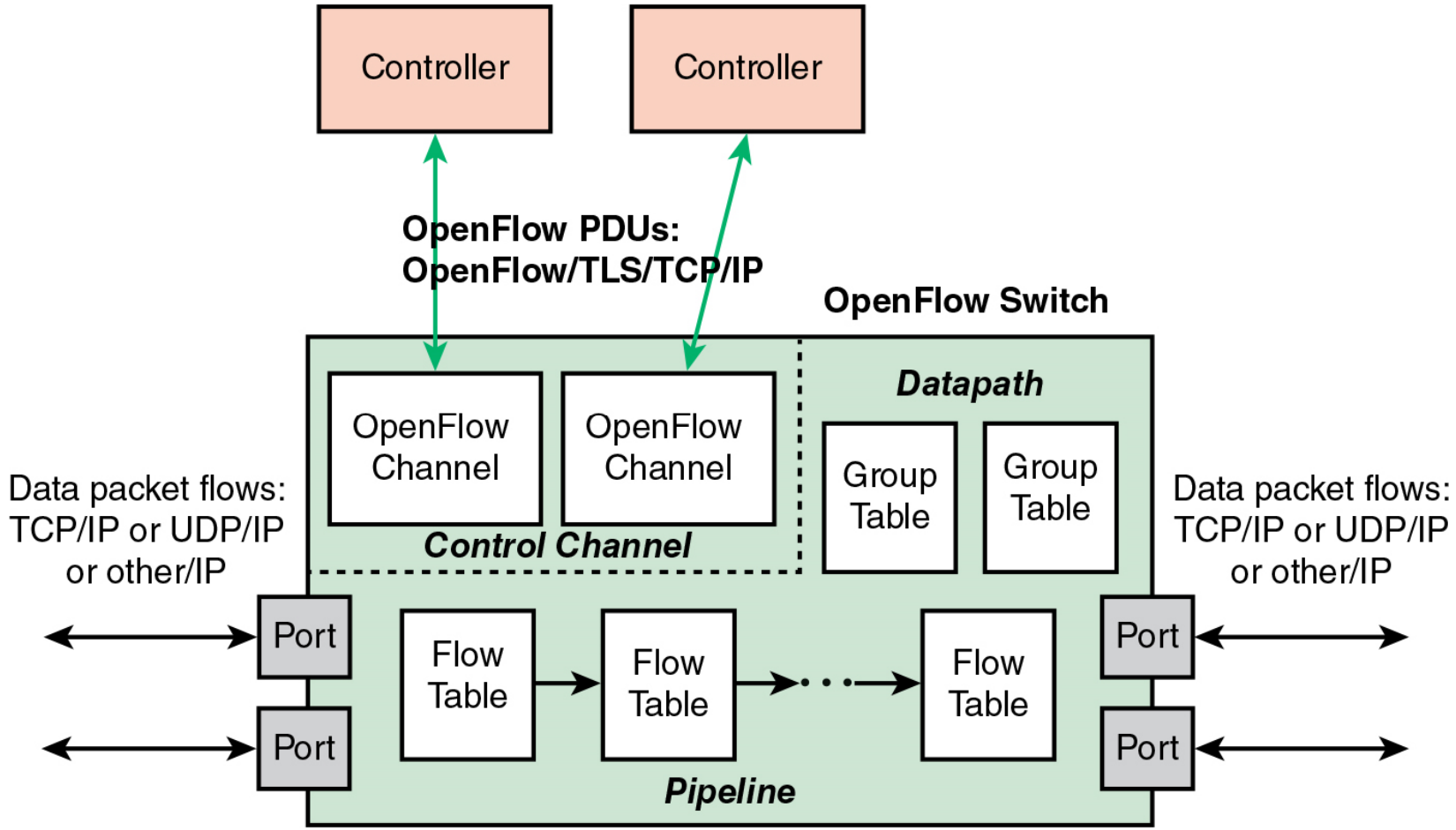
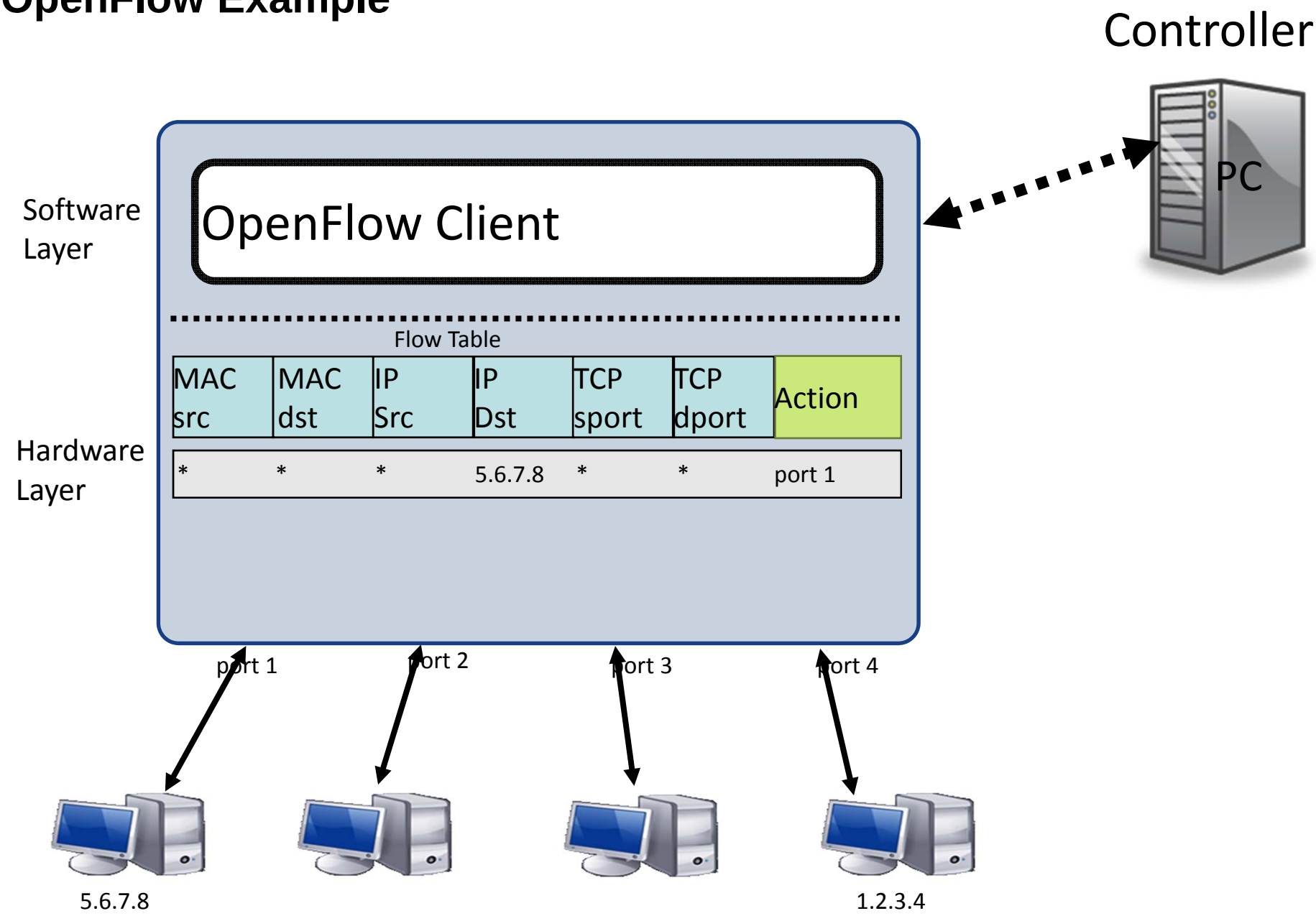


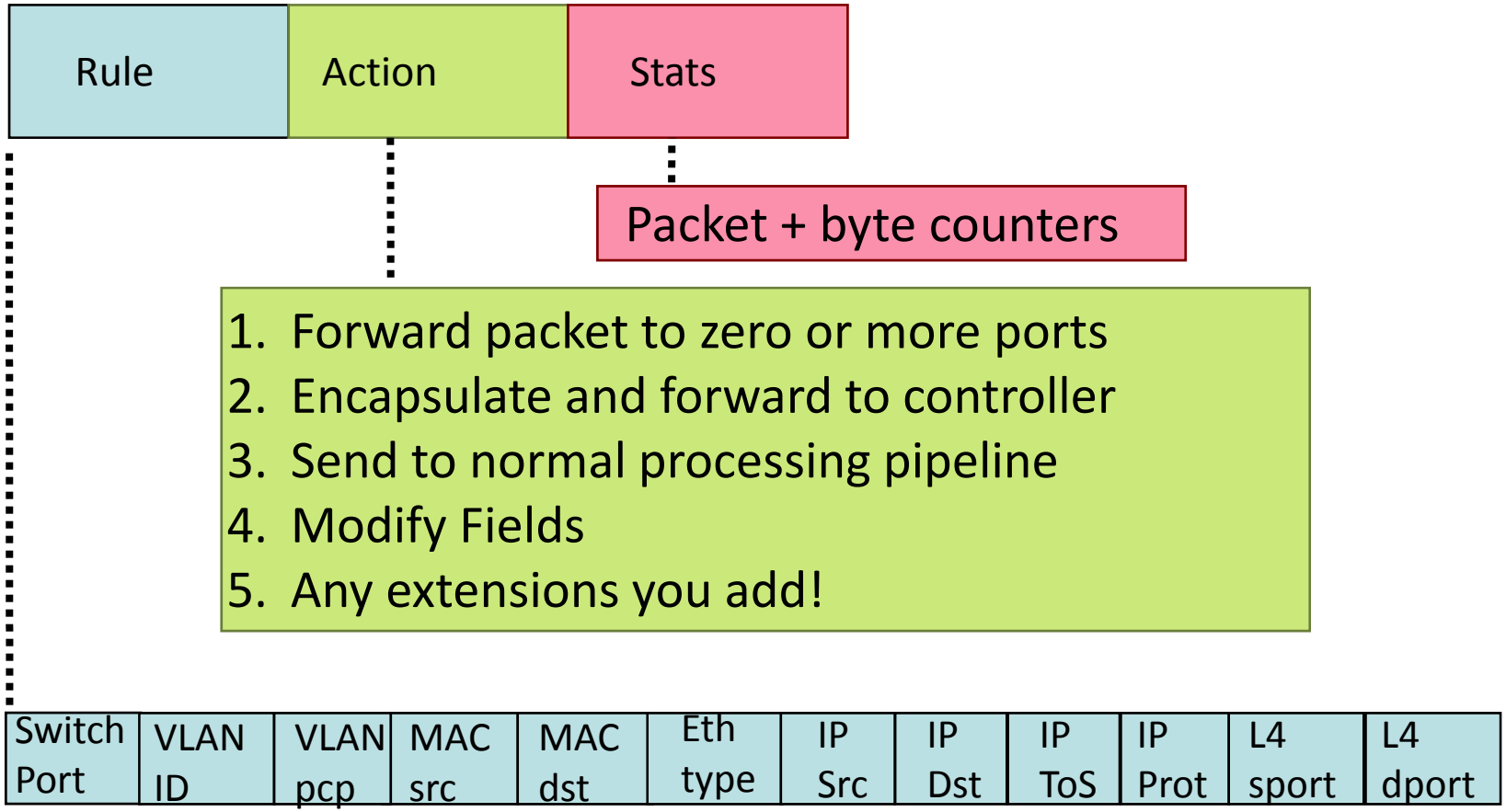
FIGURE 4.4 OpenFlow Switch

OpenFlow Example



OpenFlow Basics

Flow Table Entries



+ mask what fields to match

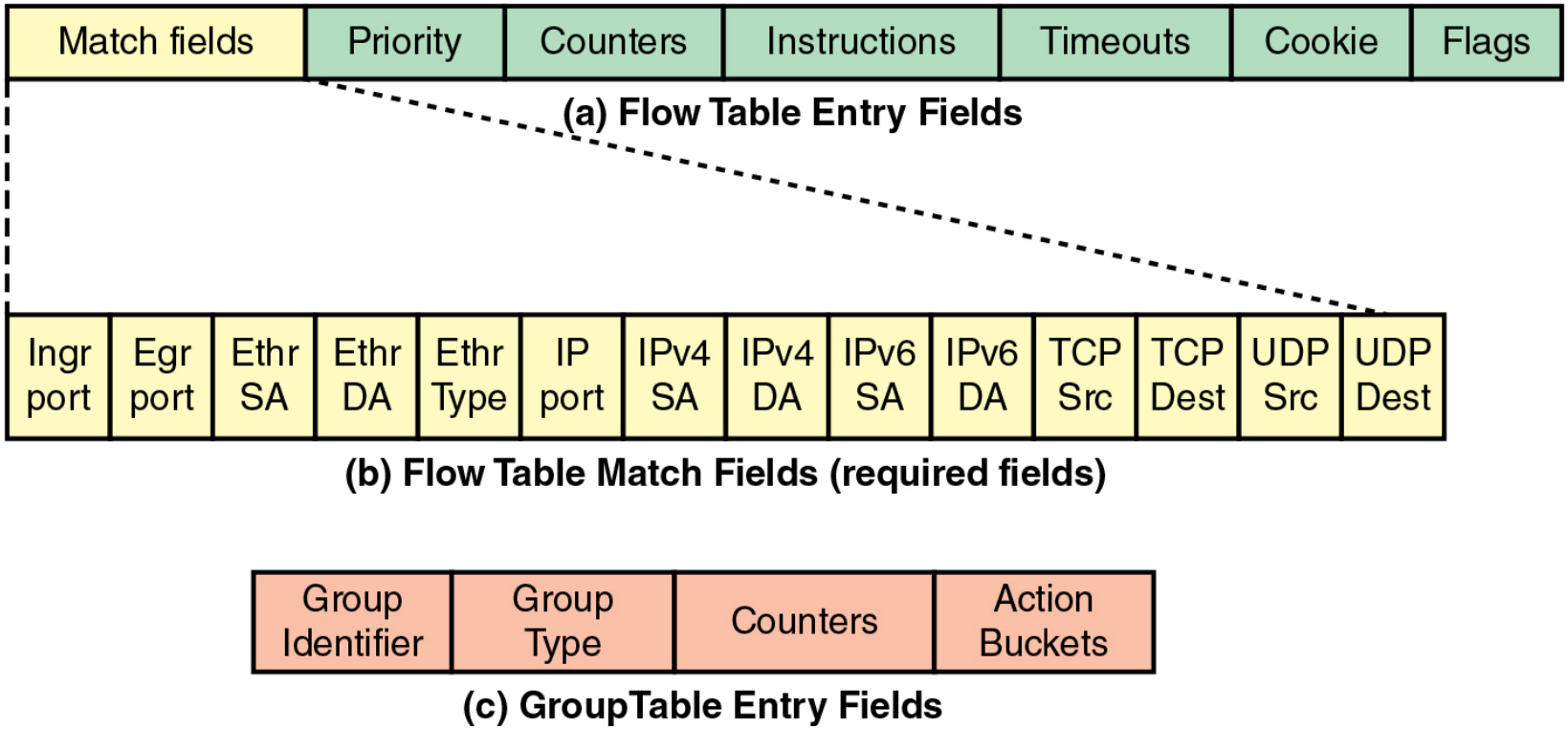


FIGURE 4.5 OpenFlow Table Entry Formats

Examples

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Examples

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

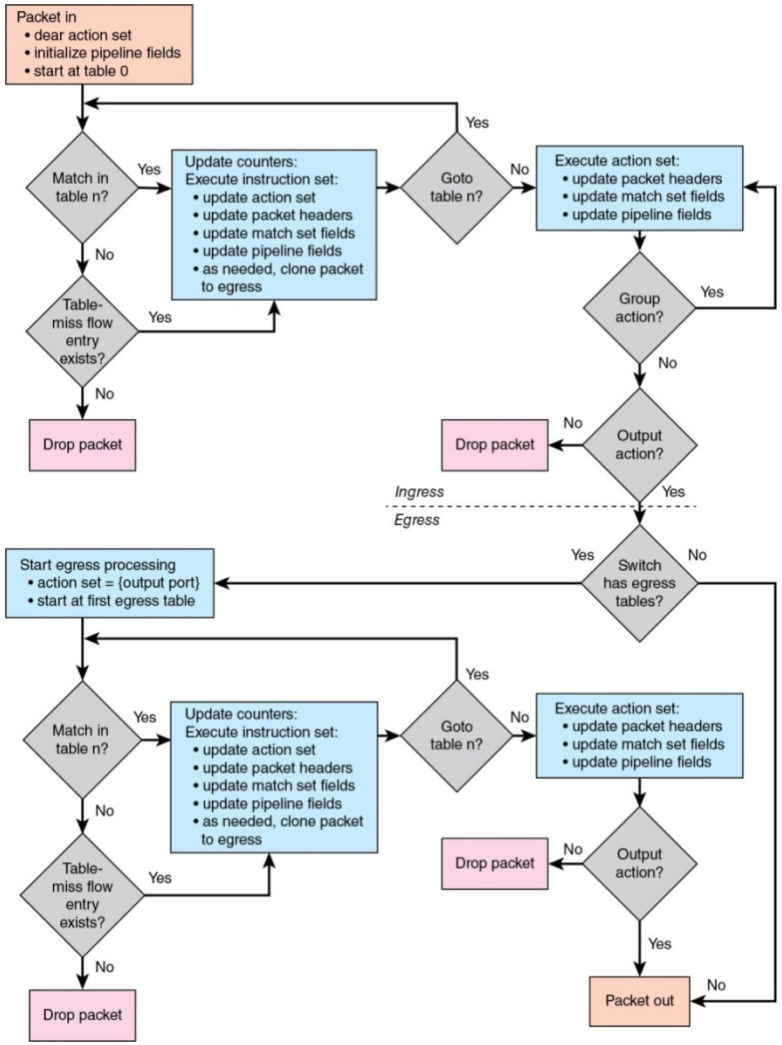


FIGURE 4.6 Simplified Flowchart Detailing Packet Flow Through an OpenFlow Switch

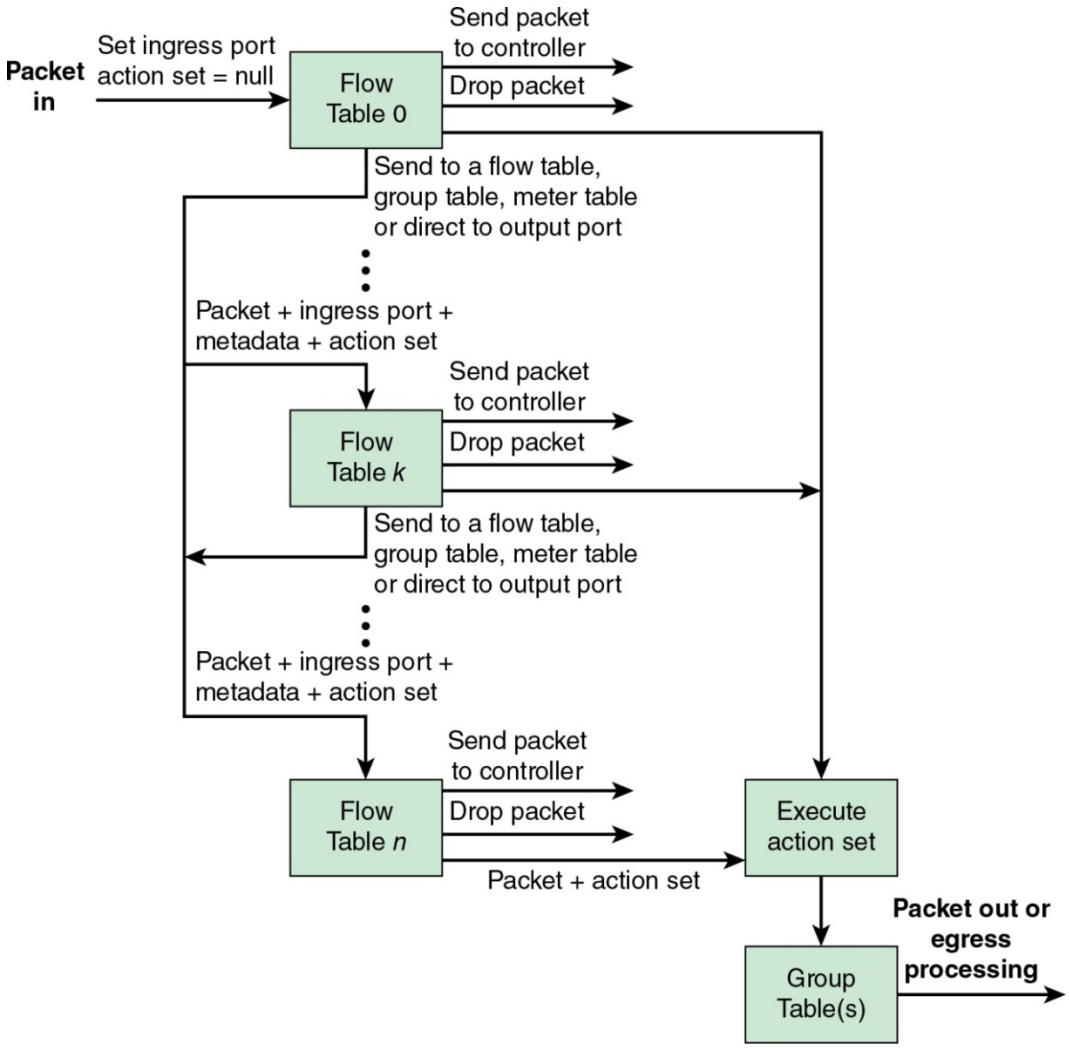


FIGURE 4.7 Packet Flow Through an OpenFlow Switch: Ingress Processing

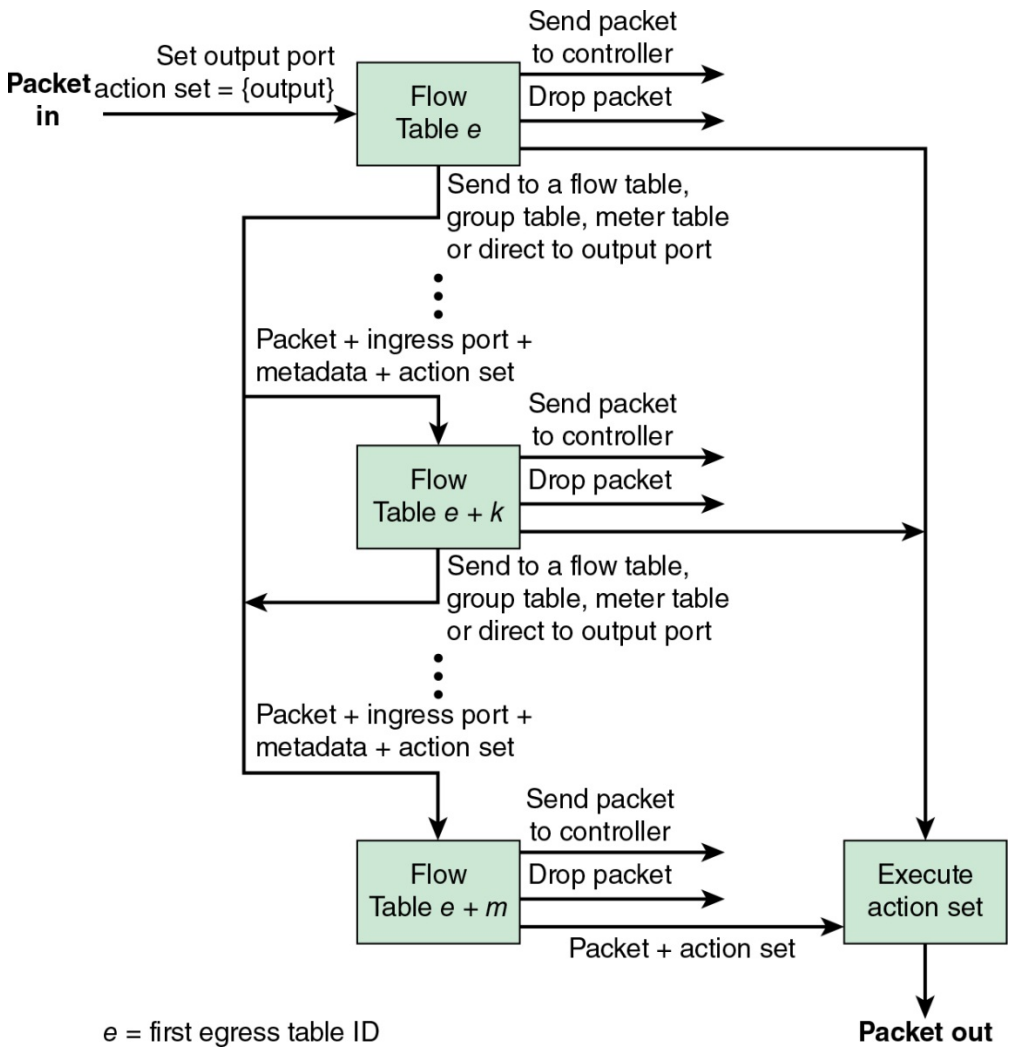


FIGURE 4.8 Packet Flow Through OpenFlow Switch: Egress Processing

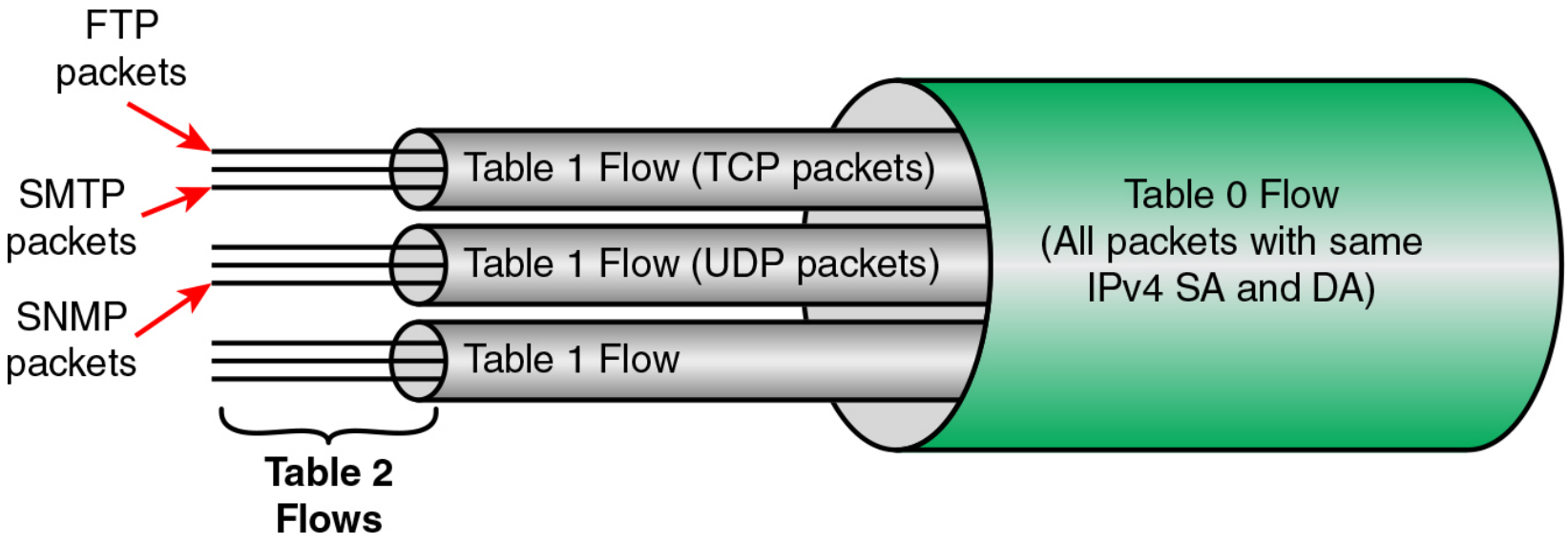


FIGURE 4.9 Example of Nested Flows

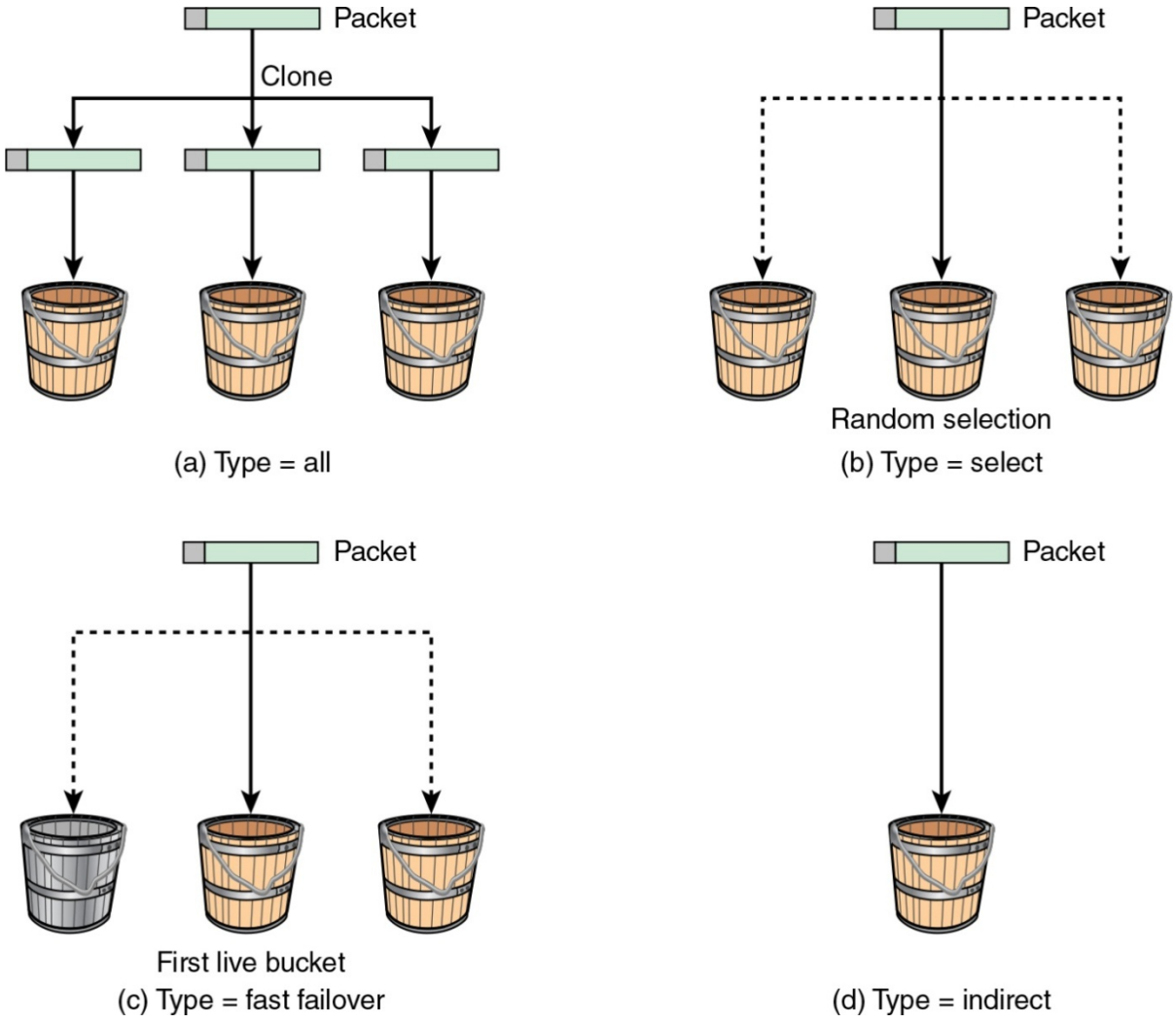
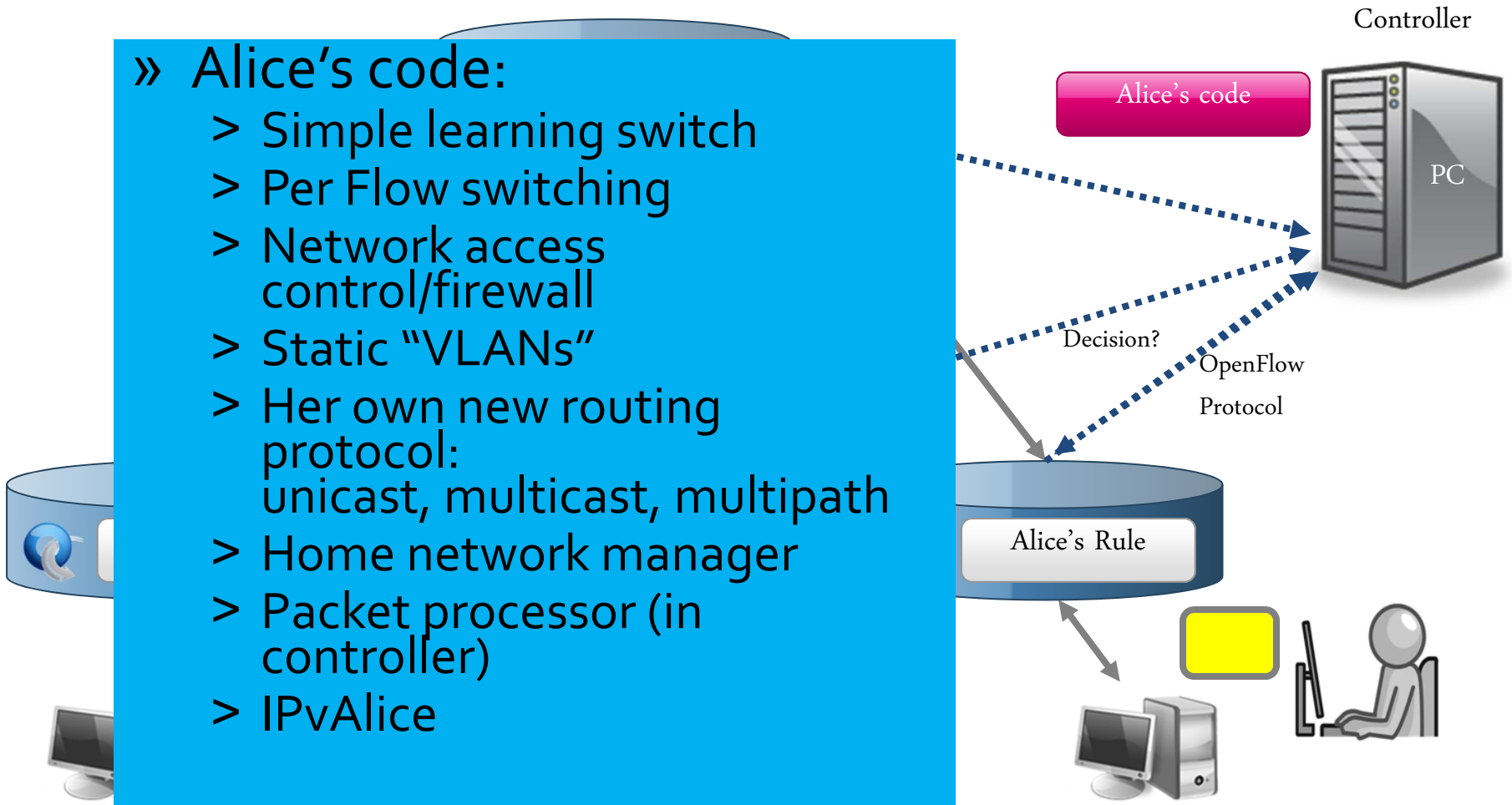


FIGURE 4.10 Group Types

OpenFlow Usage

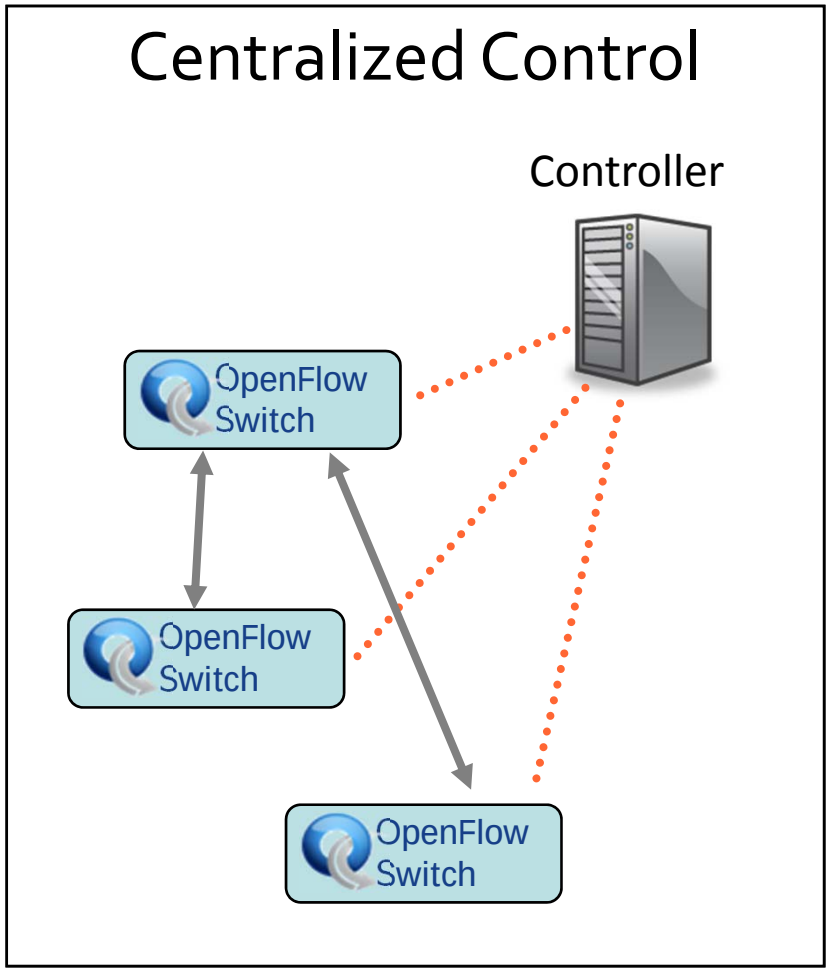
- Alice's code:
  - Simple learning switch
  - Per Flow switching
  - Network access control/firewall
  - Static "VLANs"
  - Her own new routing protocol: unicast, multicast, multipath
  - Home network manager
  - Packet processor (in controller)
  - IPvAlice

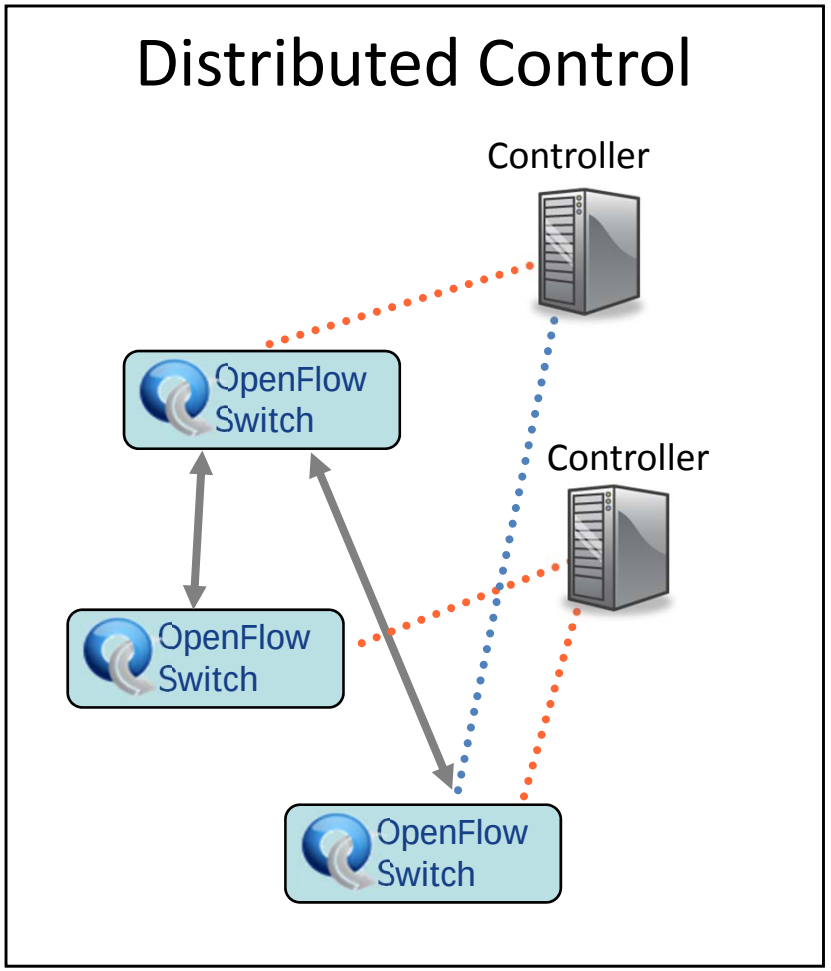


- Standard way to control flow-tables in commercial switches and routers
- Just need to update firmware
- Essential to the implementation of SDN

Centralized/Distributed Control

- **"Onix: A Distributed Control Platform for Large-scale Production Networks"**





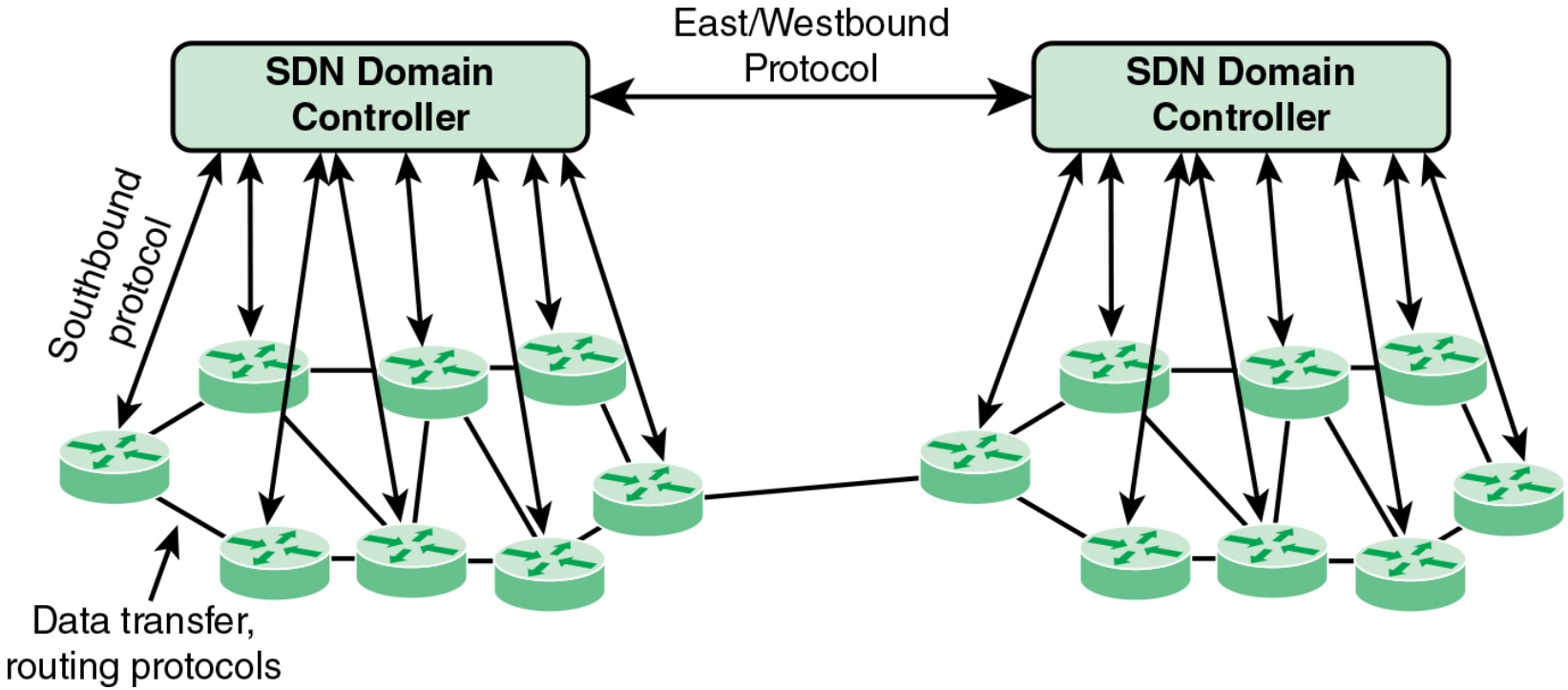


FIGURE 5.10 SDN Domain Structure

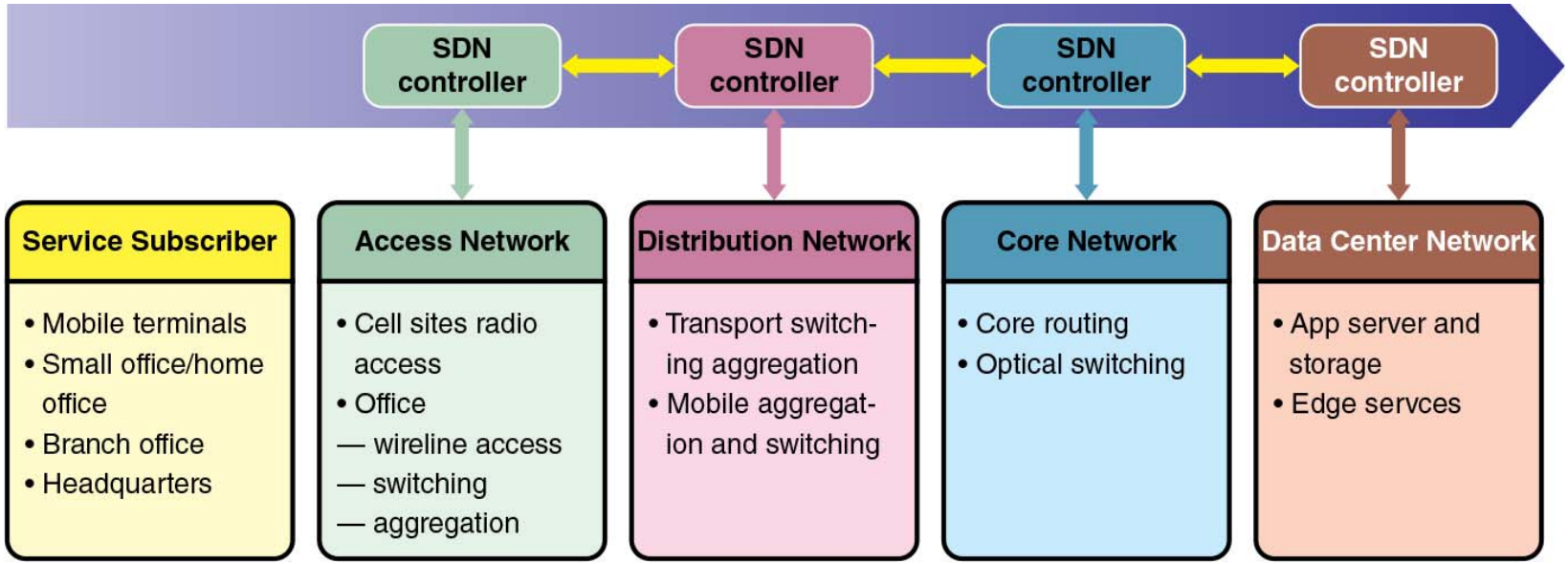


FIGURE 5.11 Federation of SDN Controllers [GUPT14]

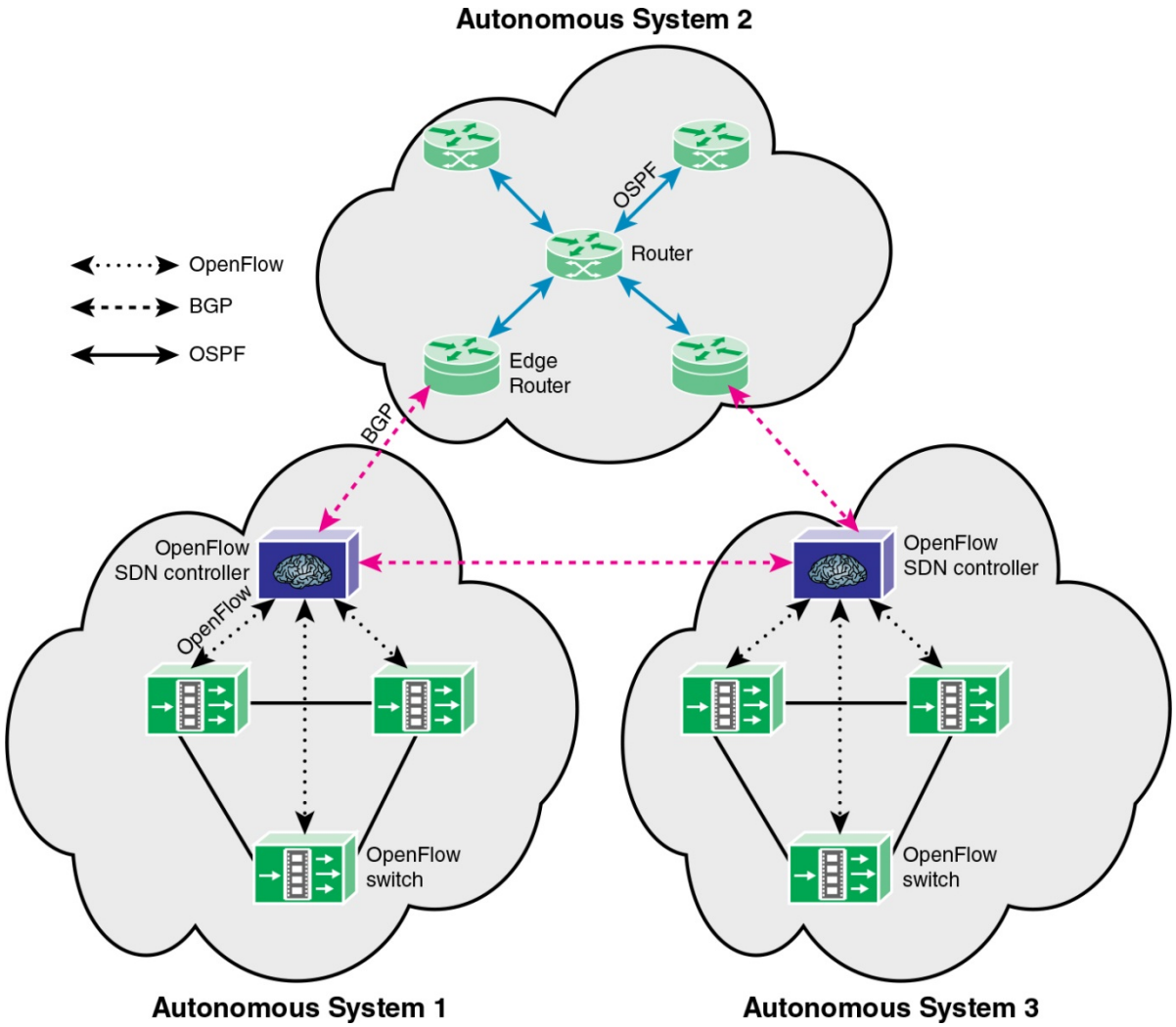


FIGURE 5.12 Heterogeneous Autonomous Systems with OpenFlow and Non-OpenFlow Domains

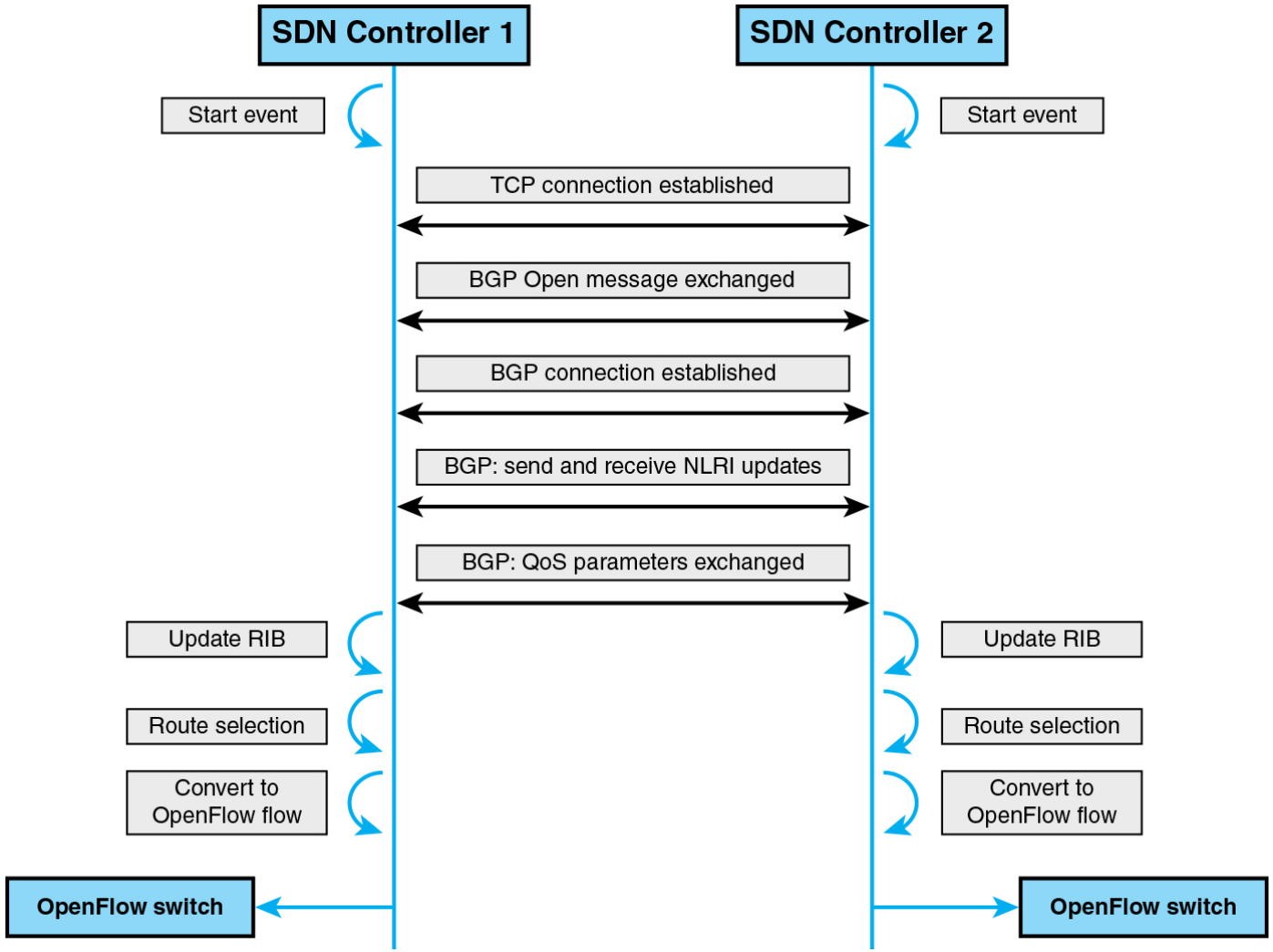


FIGURE 5.13 East-West Connection Establishment, Route, and Flow Setup

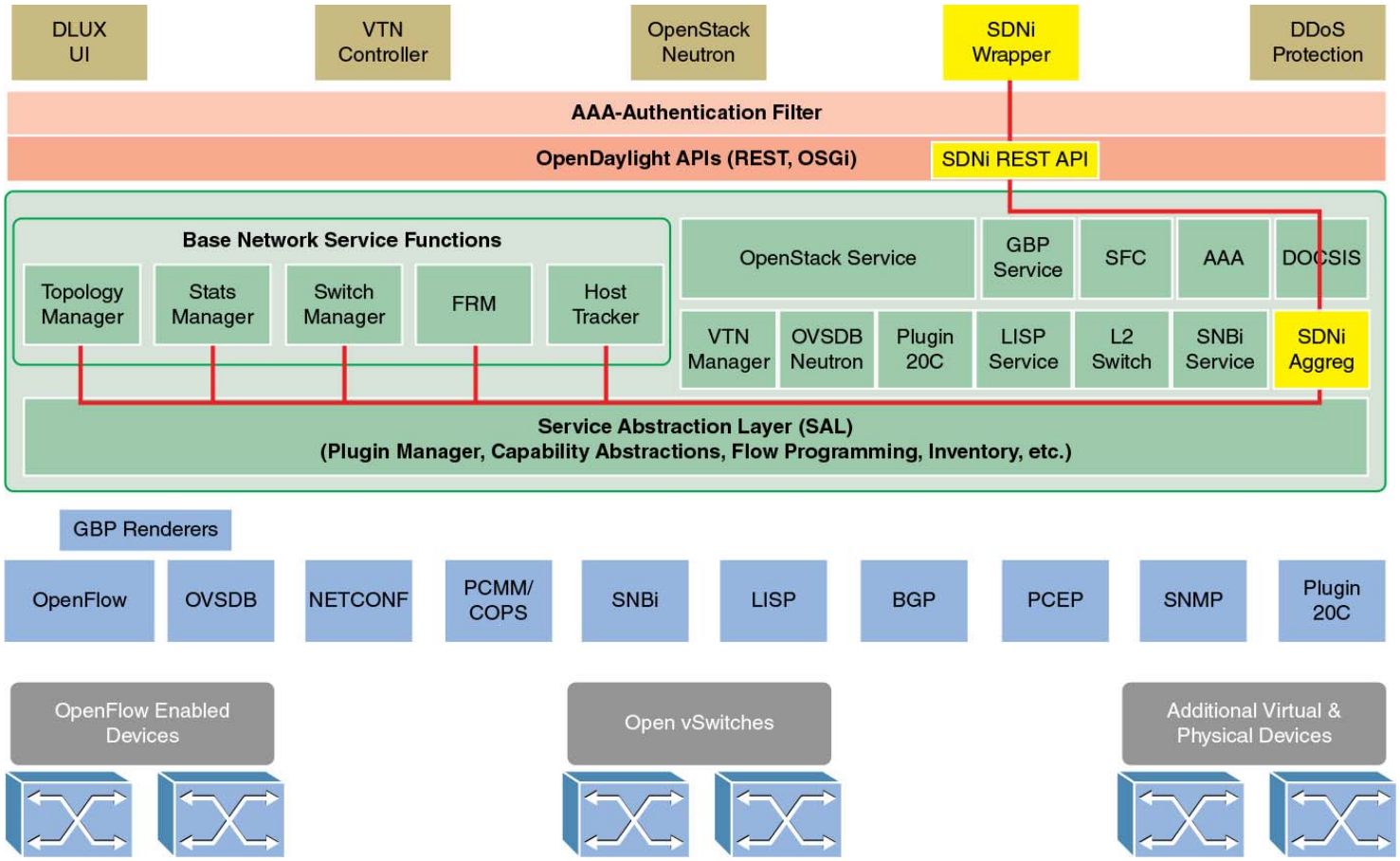


FIGURE 5.14 SDNi Components in OpenDaylight Structure (Helium)

Secure Channel

- SSL Connection, site-specific key
- Controller discovery protocol
- Encapsulate packets for controller
- Send link/port state to controller

Main Concepts of Architecture

- Separate data from control
  - A standard protocol between data and control
- Define a generalized flow table
  - Very flexible and generalized flow abstraction
  - Open up layers1-7
- Open control API
  - For control and management applications
- Virtualization of the data and control plane
- Backward compatible
  - Though allows completely new header

OpenFlow is not enough.

OpenFlow is not enough…

- Adds the ability to modify, experiment…
- But still harder than it should be to add features to a network
- Effectively assembly programming or an ISA

[OpenFlow is just a forwarding table management protocol]

SDN App example

OSPF and Dijkstra

OSPF

- RFC 2328: **245 pages**

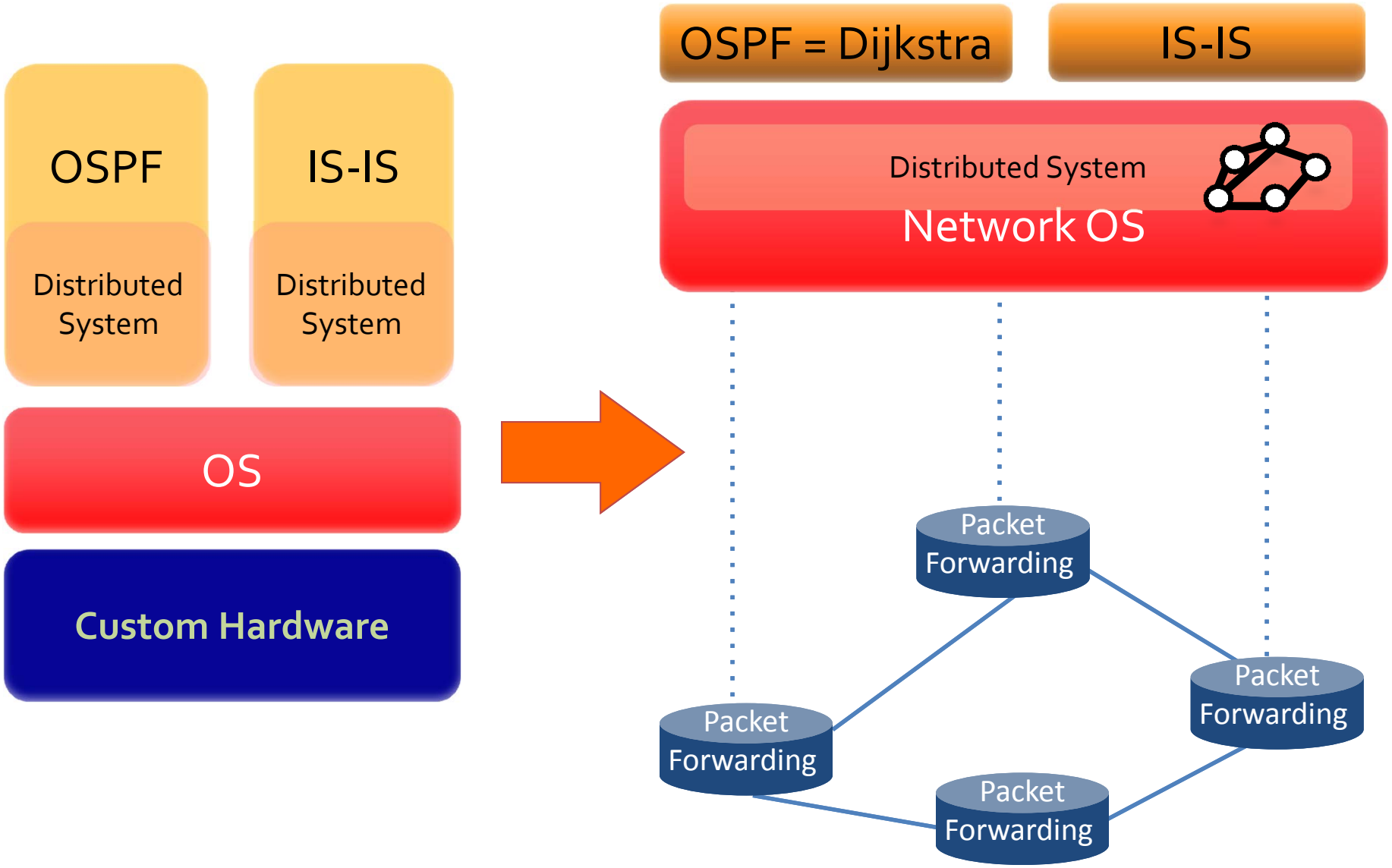
Distributed Protocol

- 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

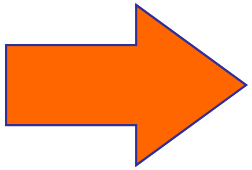
Other SDN Use Cases

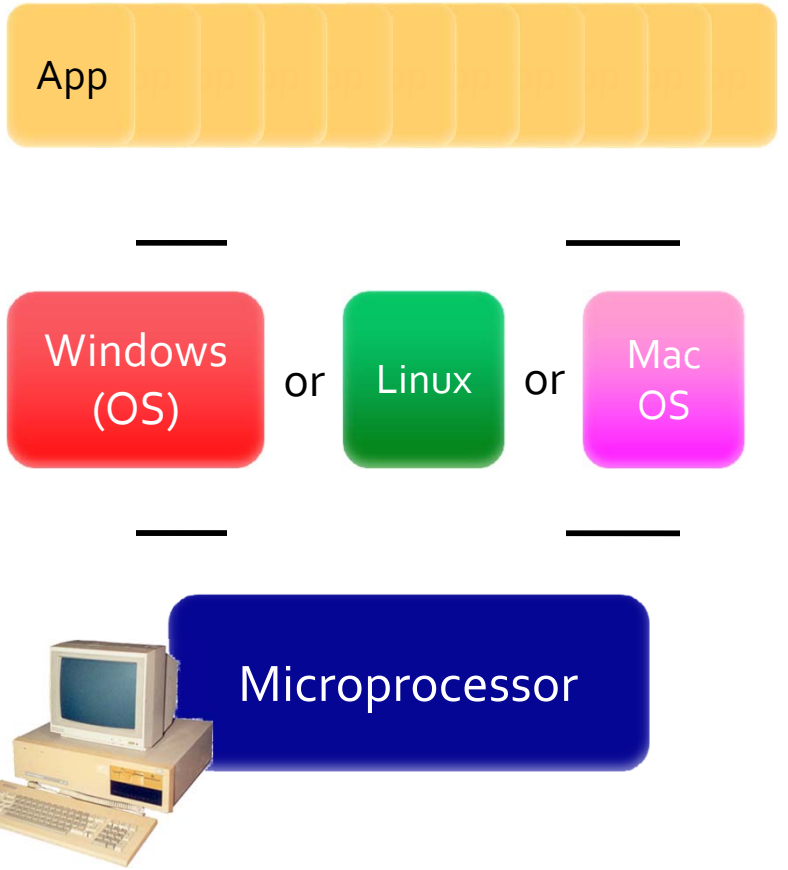
- Energy conservation, routing, and management in data centers
- Seamless use of diverse wireless networks
- Network based load balancing
- Traffic engineering
- Slicing and scalable remote control/management of home networks
- Experimentation with new approaches and protocols using selected production traffic
- Run virtual shadow network for traffic analysis and re-configuration
- And many more …

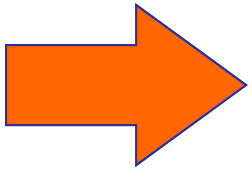
See http://www.openflow.org/videos/

A Helpful Analogy

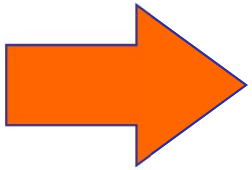
Mainframe industry in the 1980s: Vertically integrated
Closed, proprietary
Slow innovation
Small industry

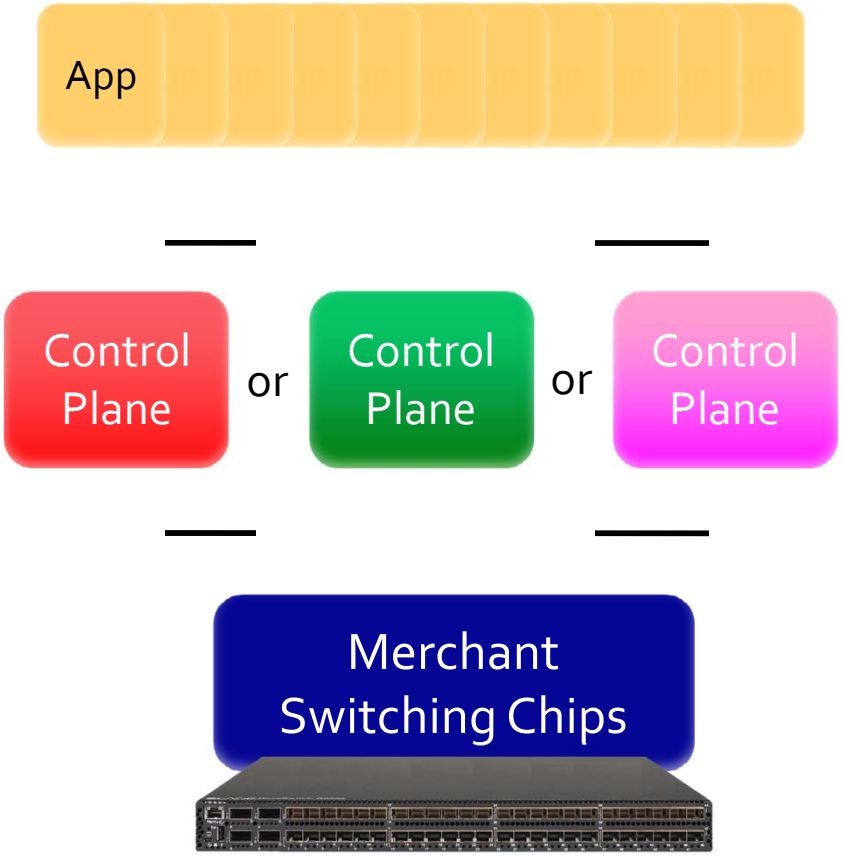




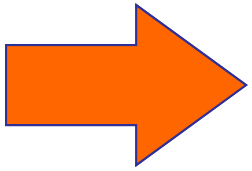


Horizontal
Open interfaces
Rapid innovation
Huge industry



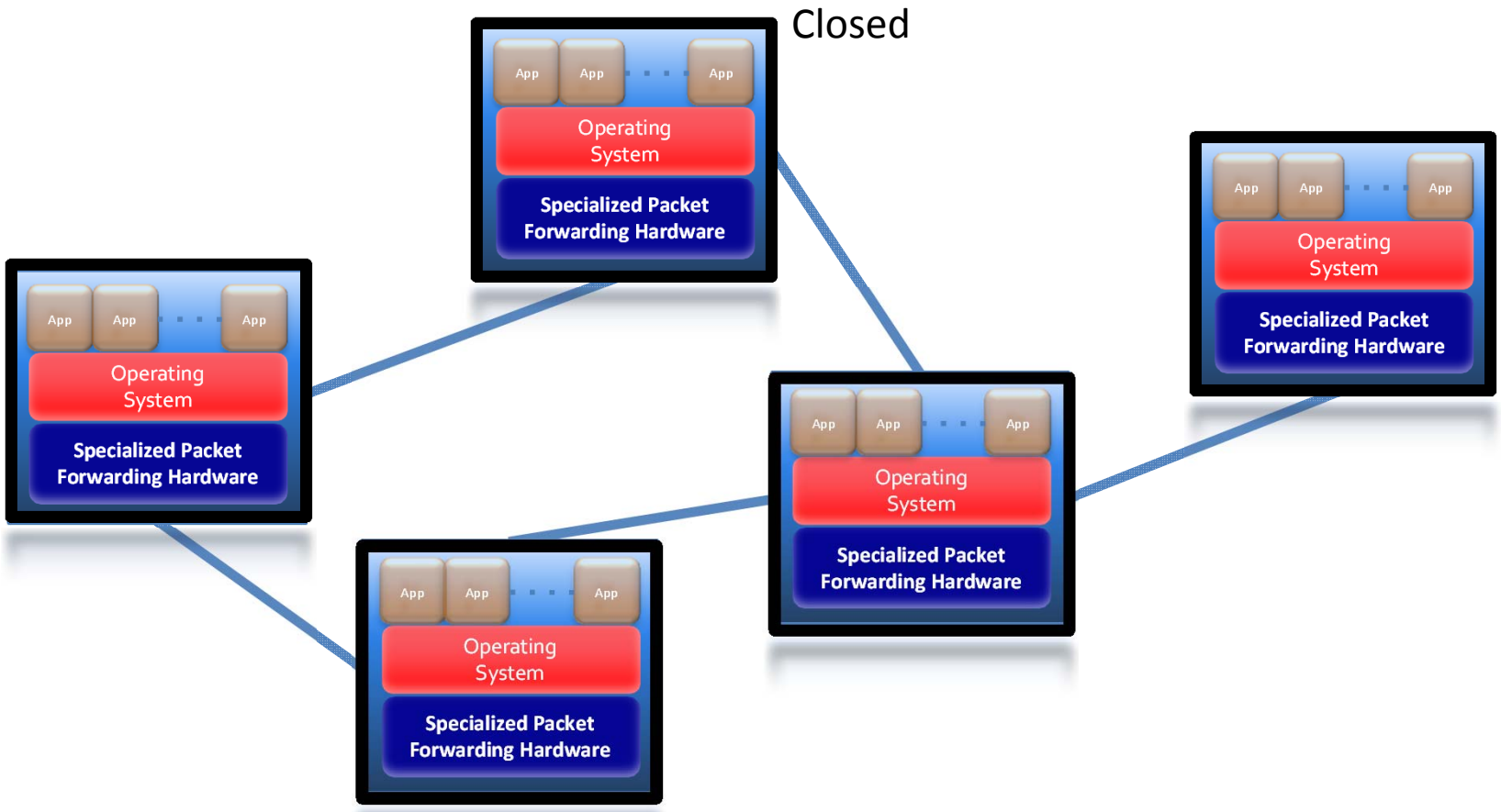


Networking industry in 2007: Vertically integrated
Closed, proprietary
Slow innovation

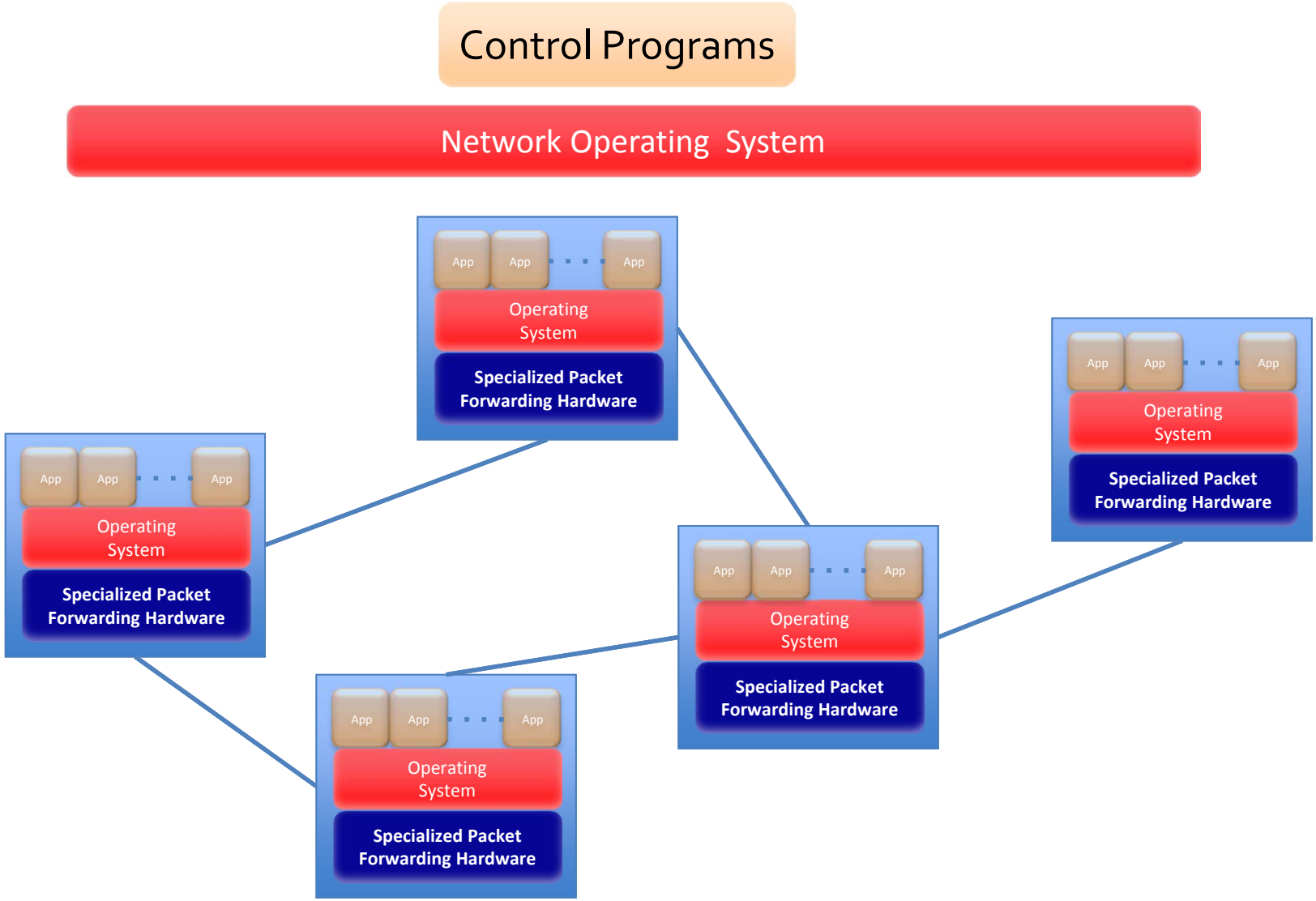


Horizontal
Open interfaces
Rapid innovation

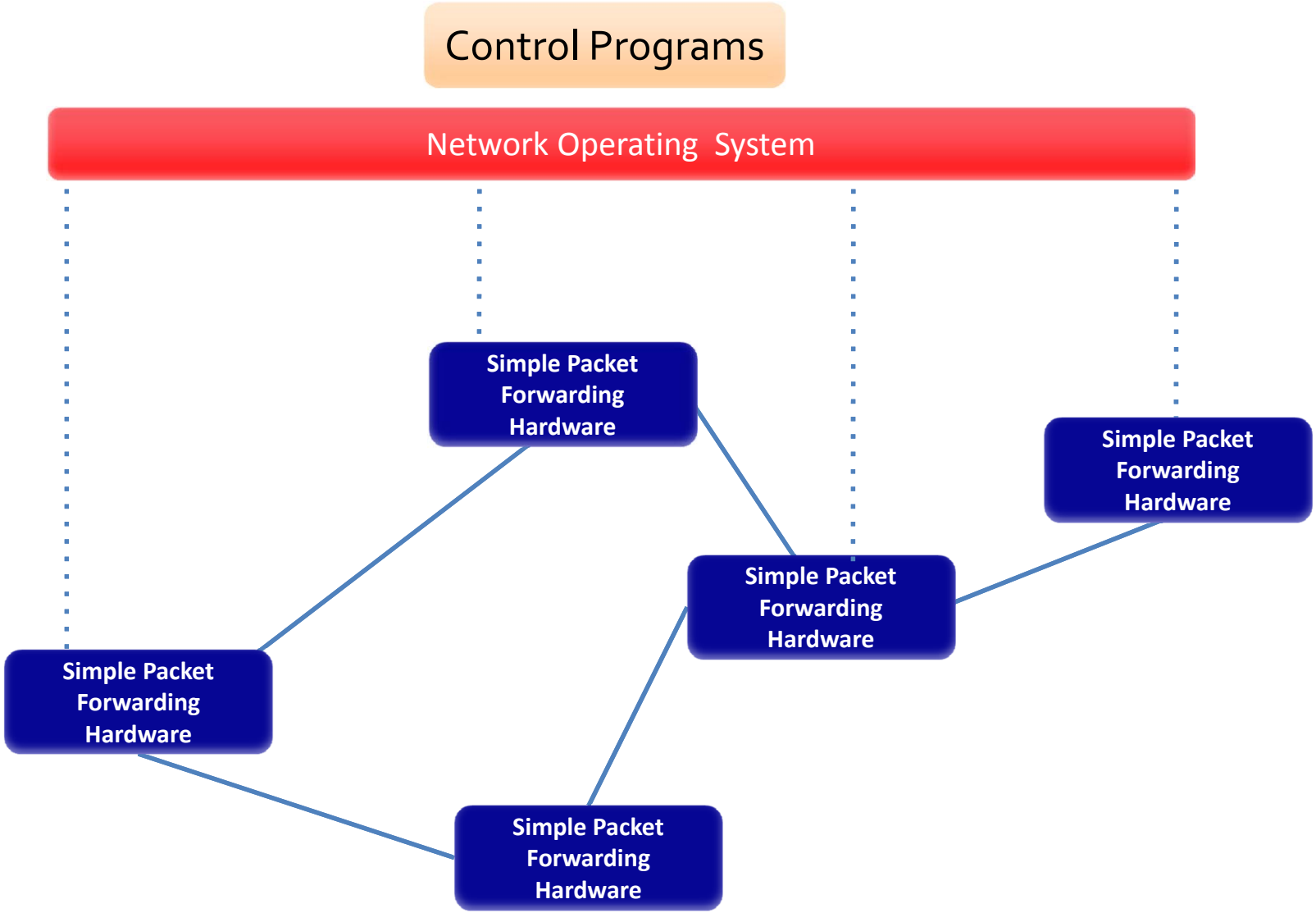
Idea: An OS for Networks



Idea: An OS for Networks



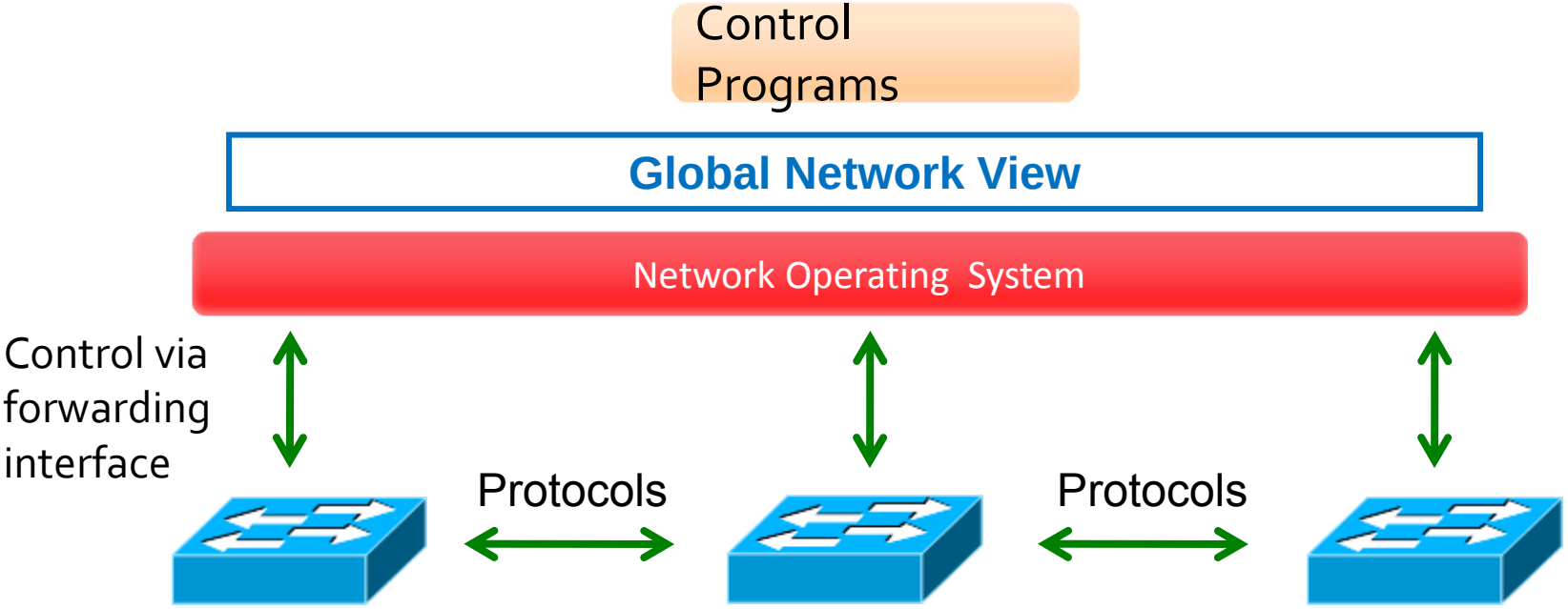
Idea: An OS for Networks



Idea: An OS for Networks

- **"NOX: Towards an Operating System for Networks"**

Software-Defined Networking (SDN)



The Future of Networking, and the Past of Protocols, Scott Shenker, *with Martin Casado, Teemu Koponen, Nick McKeown*

Software Defined Networking

- **No longer designing distributed control protocols**

- **Much easier to write, verify, maintain, …**
  - An interface for programming

Software Defined Networking

- **Examples**

  - Ethane: network-wide access-control
    - define a single networkwide fine-grain policy, and then enforces it directly. Ethane couples extremely simple flow-based Ethernet switches with a centralized controller that manages the admittance and routing of flows.

  - Power Management

Software Defined Networking

- **Questions:**

  - How to obtain global information?
  - What are the configurations?
  - How to implement?
  - How is the scalability?
  - How does it really work?

Current status of SDN

- Hardware support

Juniper MX-series

NEC IP8800

WiMax (NEC)

HP Procurve 5400

Netgear 7324

PC Engines

Pronto 3240/3290

Ciena Coredirector

More coming soon...

Current status of SDN

- Industry support
  - Google built hardware and software based on the OpenFlow protocol
  - VMware purchased Nicira for $1.26 billion in 2012
  - IBM, HP, NEC, Cisco and Juniper also are offering SDNs that may incorporate OpenFlow, but also have other elements that are specific to that vendor and their gear.

http://gigaom.com/2012/03/19/are-vendors-closing-openflow/
http://gigaom.com/2012/12/17/2012-the-year-software-defined-networking-sold-out/
http://www.extremetech.com/internet/140459-networking-is-getting-better-and-thats-partly-thanks-to-google

Future Focuses of SDN

- Research focuses
  - Mostly implementations of newly proposed systems, frameworks, or applications

Conclusions

- What is SDN?
  - A system-layered abstraction
  - Programmable, flexible, and extensible
- What is OpenFlow?
  - Interface between switches and controllers
  - Enabling SDN
- Future SDN
  - Enabling innovation