Software Defined Networks

H. Khanmirza

Assistant Professor

K. N. Toosi University of Technology

Outline

- Part I
  - Rise of SDN
  - SDN Basics
- Part II
  - Research Problems
- Part III
  - My Current Research

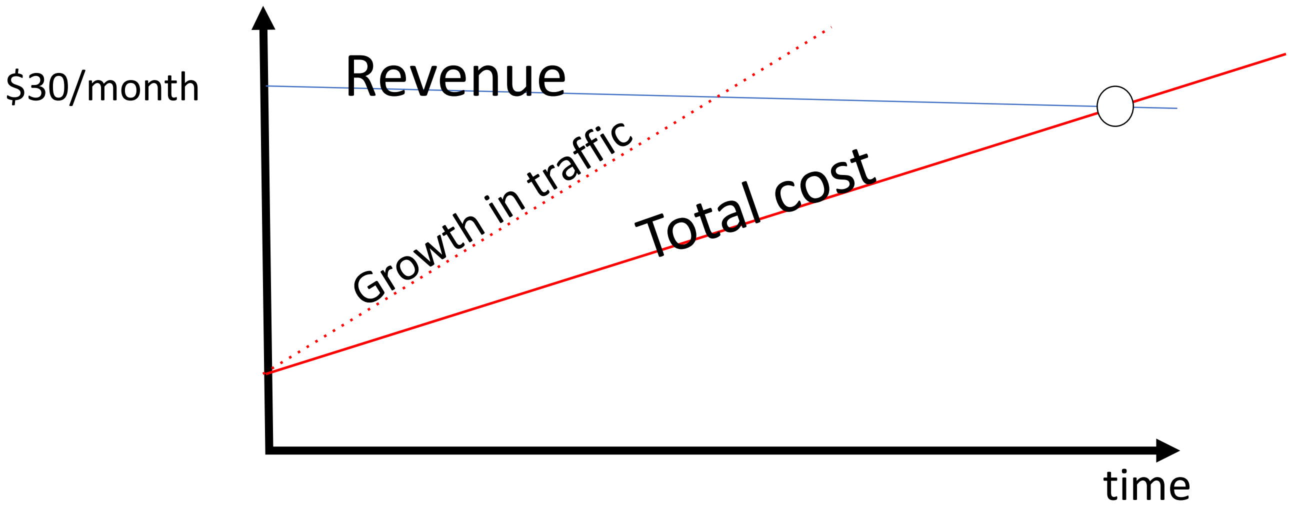
Rise of SDN

"Software will eat the world"

- Marc Andreessen

Why We Need a New Network Paradigm?

- IP Traffic grows 40-50% per year
- While bills remains unchanged!!
- Then, costs must be reduced 40-50%
  - In practice it reduces <20%



Solution?

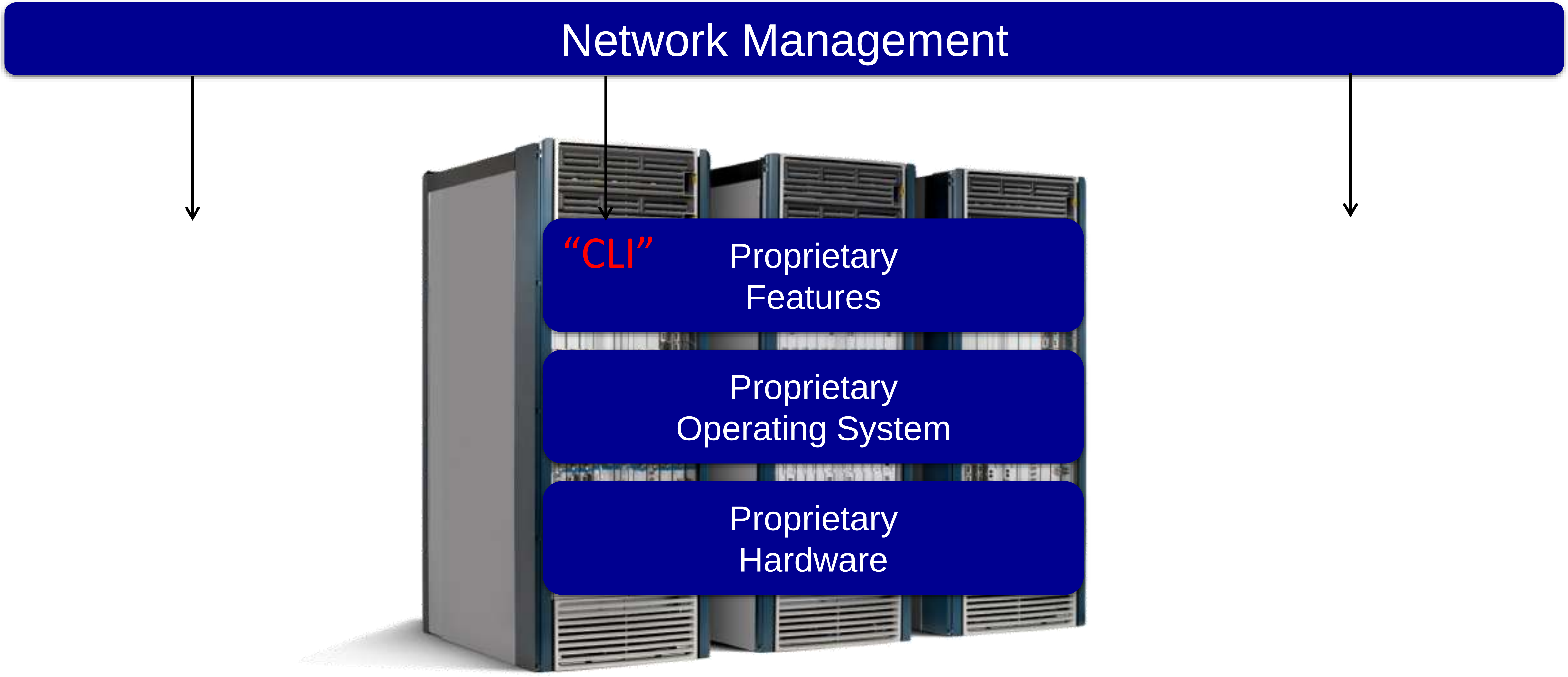
- Reduce operational costs
  - Just support and run protocols you need, software and features
- Reduce Capital Costs
  - move to simpler switches + open source control plan + apps
- Increase the price → But How??
  - by providing differentiated software service
    - in this part we all win because we get better services! if we can define.
    - Is it possible now?? NO, with hardware this is not possible.

Internet Business Issue

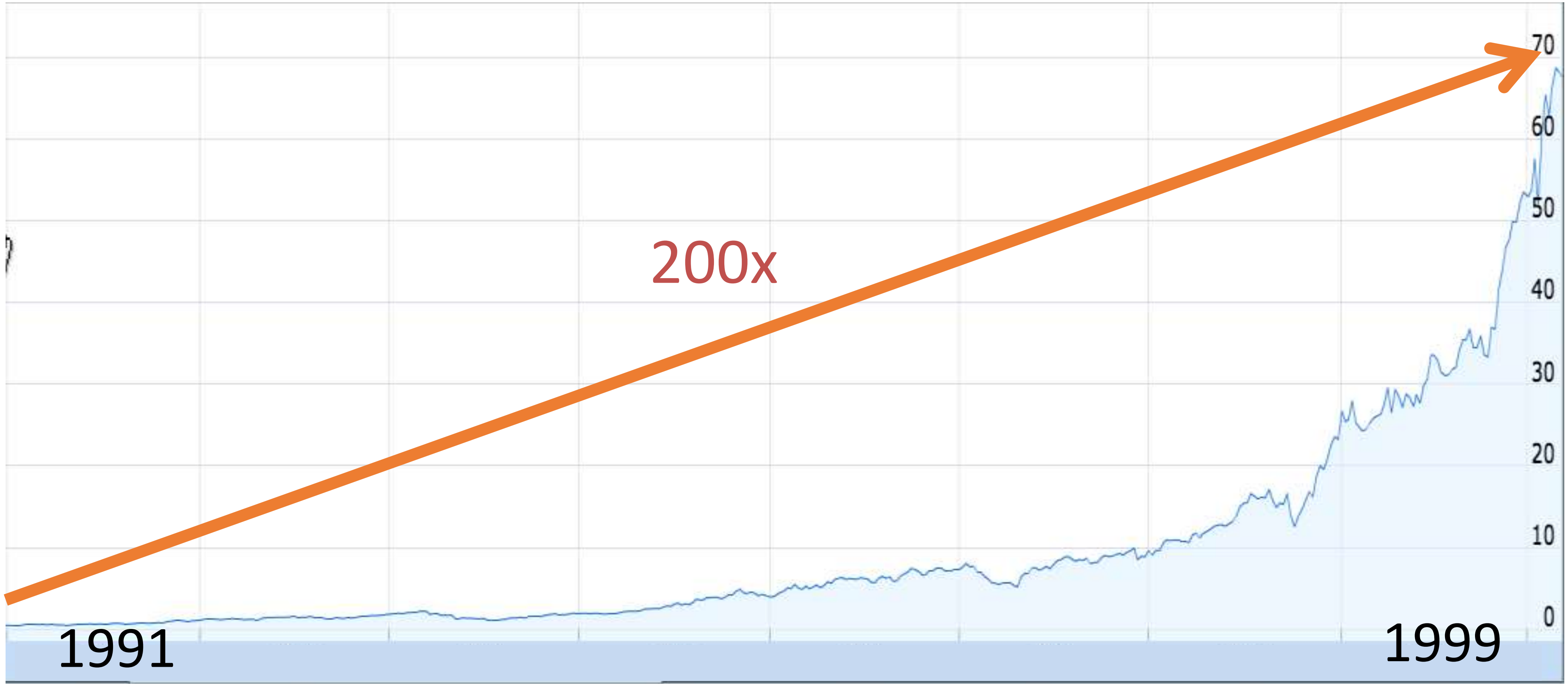
- The Internet was successful because it was simple and dumb
- Explosive growth of the internet → led to great businesses that **sell routers** —>
- Only **few** companies could survive and were successful and this is not sustainable!!

Business vs. Simplicity?

Strategy 1: Vertical integration

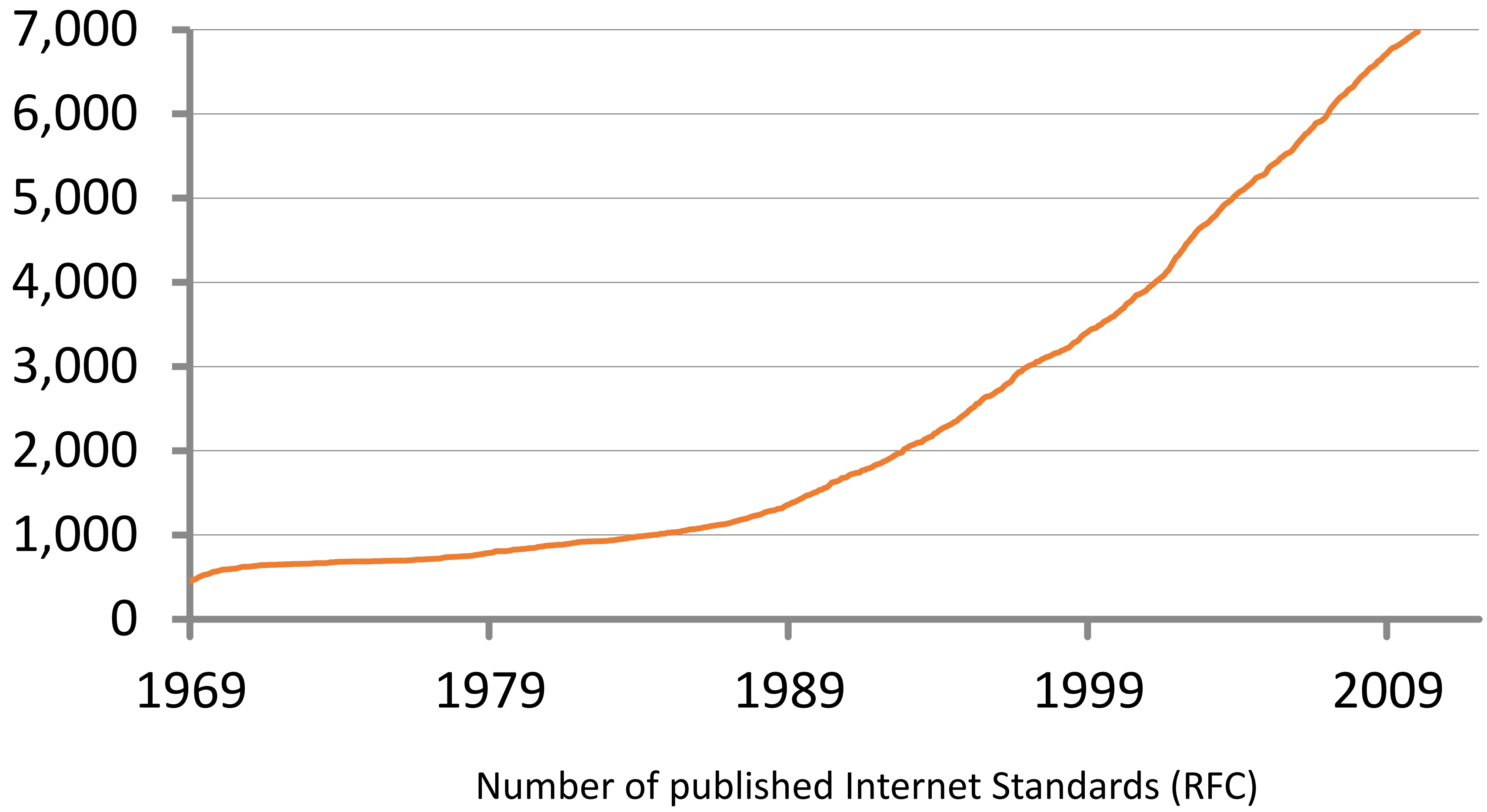


Cisco Stock Price



Google Finance

Strategy 2: Make Internet Complicated



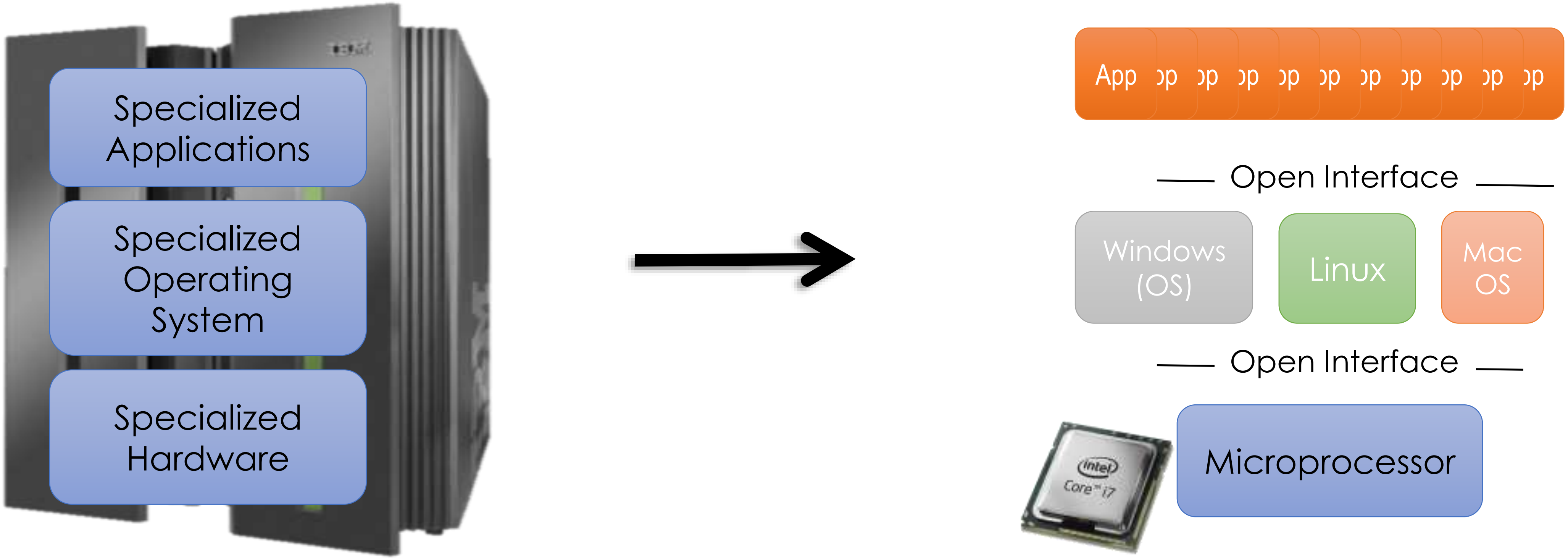
Solution

- Push innovation to service providers
  - They will be able to create new services fast and make distinction
- SDN (Software Defined Network)
  - Rise of merchant switching silicon (whitebox switches)
  - COTS (Common Off The Shelf) hardware
- Make a standard and common layer for network
  - Now competition is providing better service
  - This creates backend for cleaner race

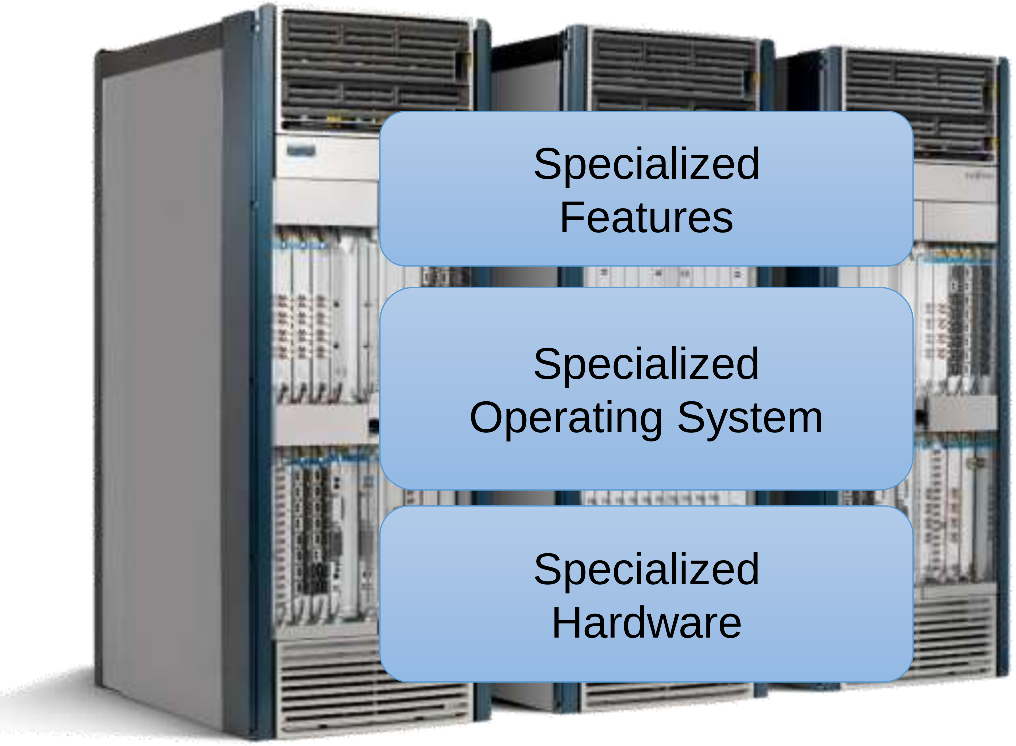
Accelarators

1. Rise of Linux.
2. Rise of simpler servers and switches.
3. NFV: Rise of virtualization.
4. SDN: Rise of merchant switching silicon.

Computer Industry



Networking Industry



Specialized Features

Specialized Operating System

Specialized Hardware

→

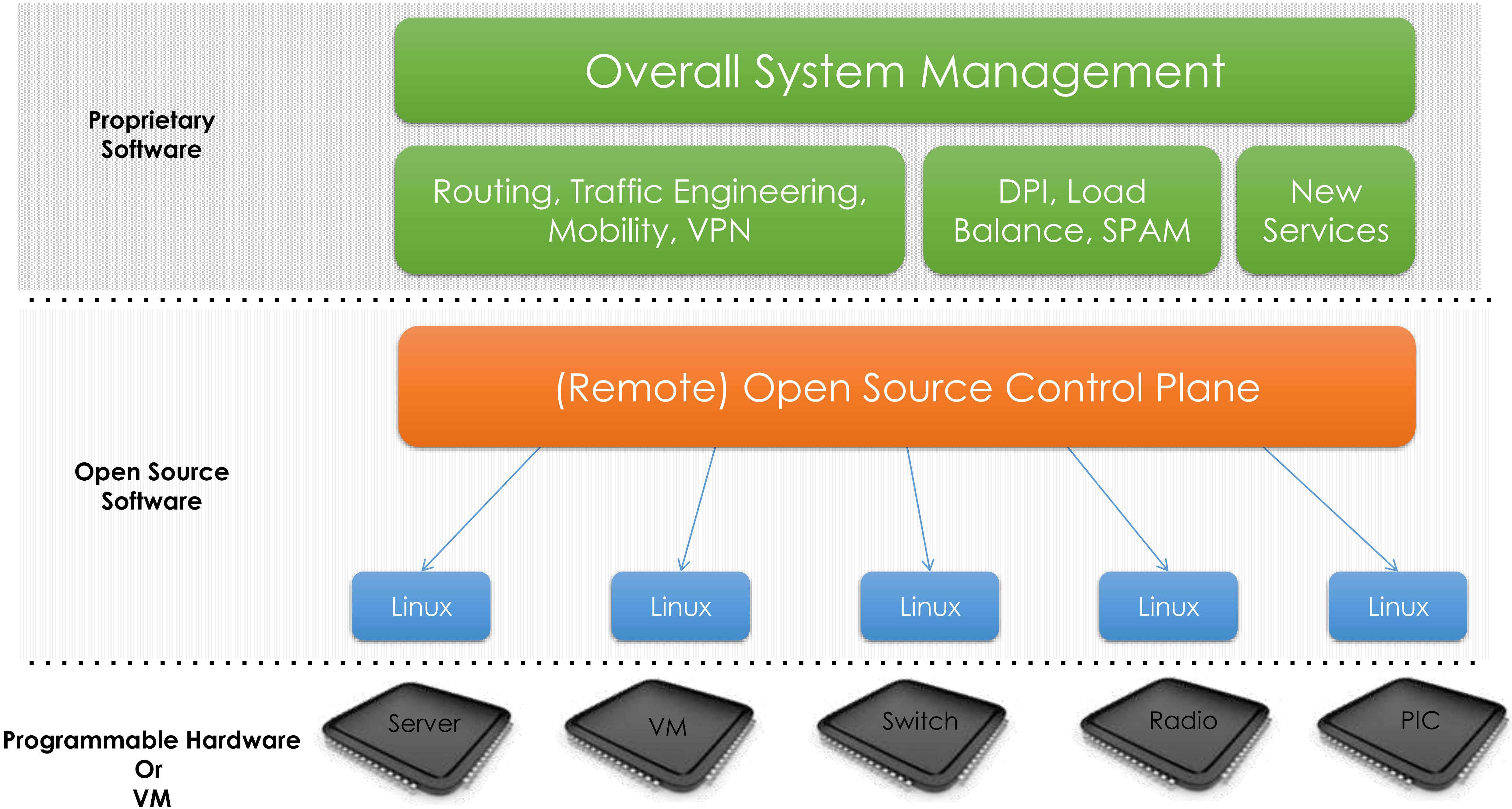
App | pp | pp | pp | pp | pp | pp | pp | pp | pp | pp

Open Interface

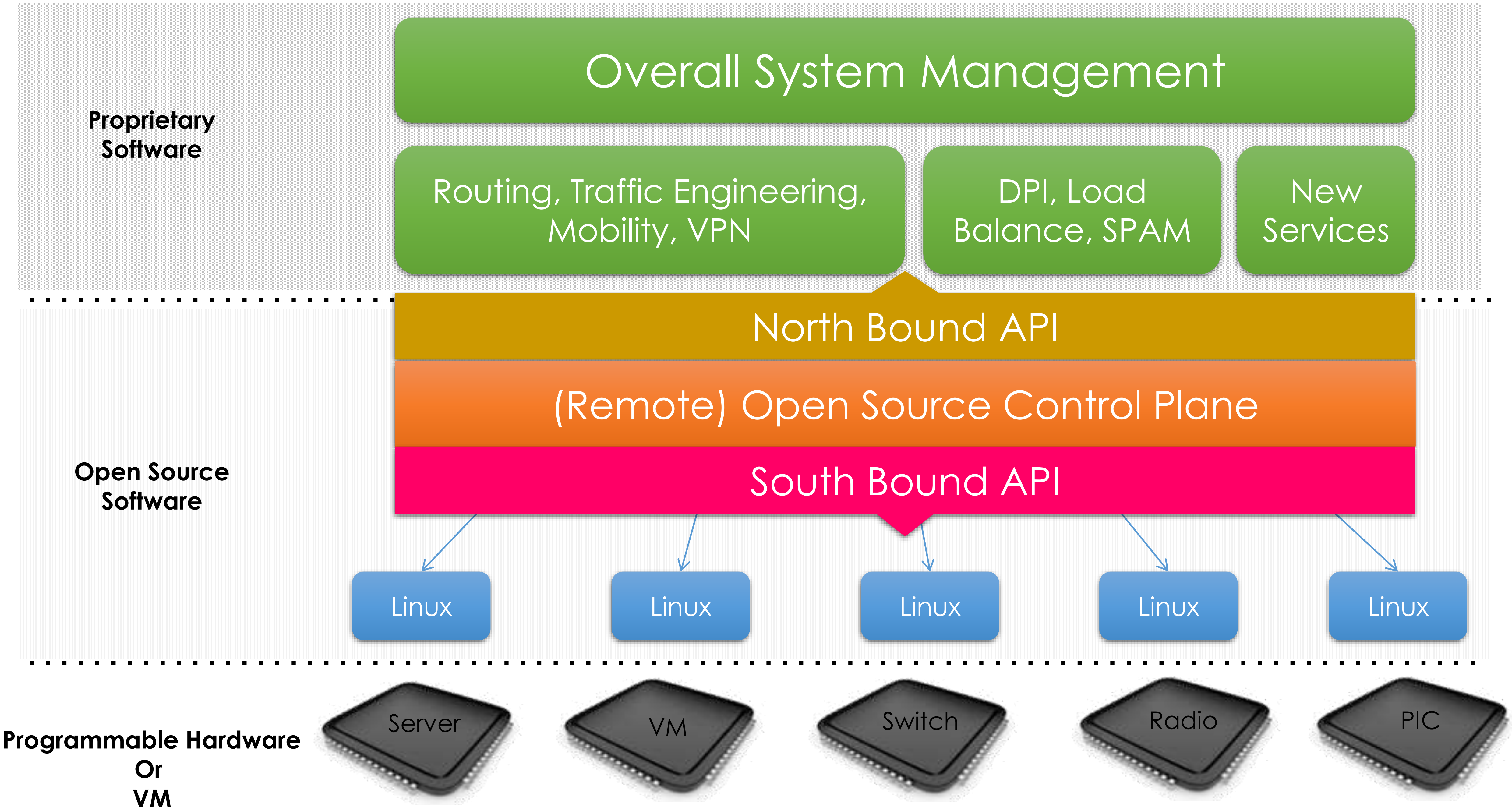
NOX | Beacon | ONIX | POX | ONOS | Floodlight | Trema | ODL | Ryu

Open Interface

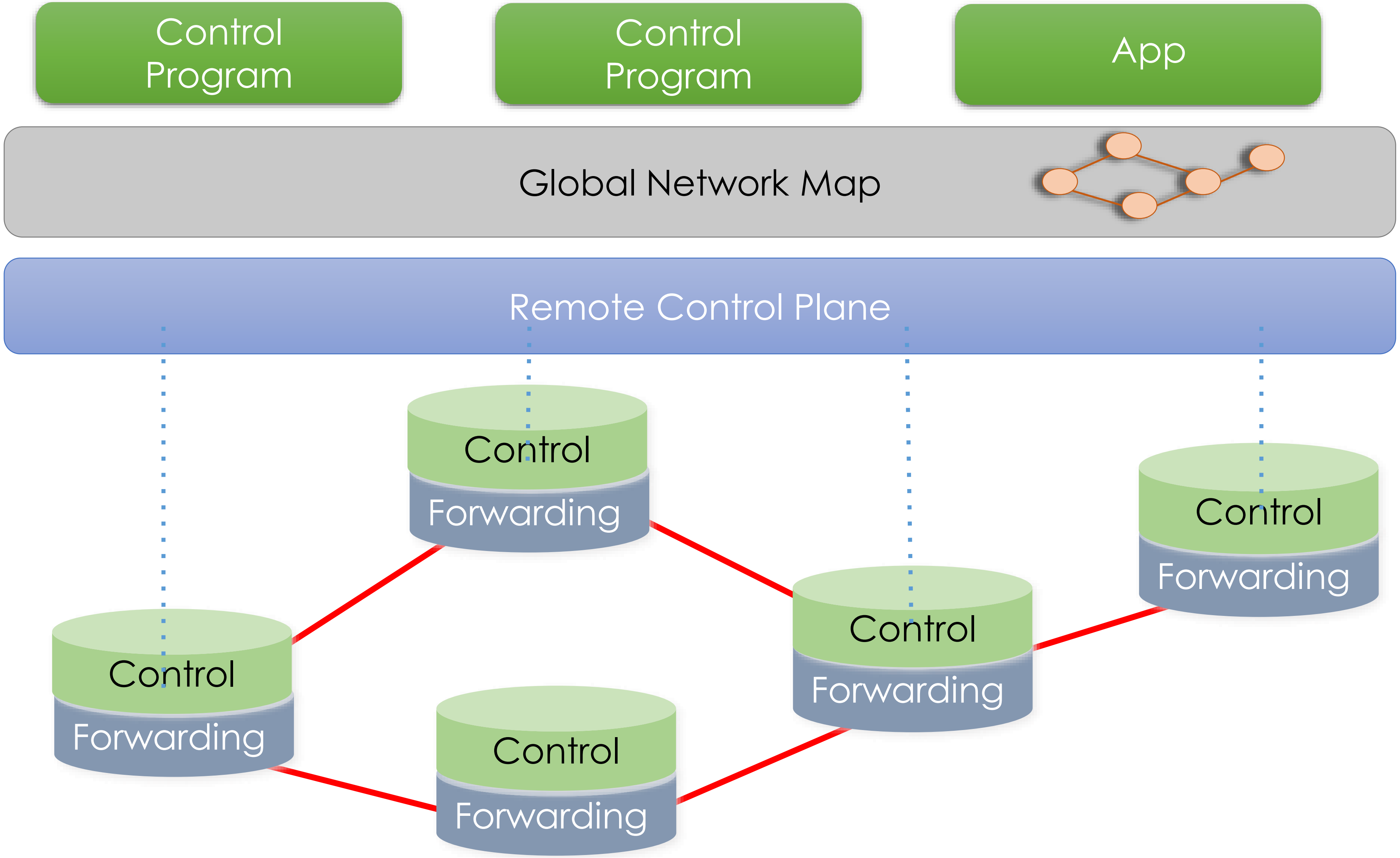
Switch Chips



Proprietary Software
Overall System Management
Routing, Traffic Engineering, Mobility, VPN
DPI, Load Balance, SPAM
New Services
Open Source Software
(Remote) Open Source Control Plane
Linux
Linux
Linux
Linux
Linux
Programmable Hardware Or VM
Server
VM
Switch
Radio
PIC


Proprietary Software
Open Source Software
Programmable Hardware Or VM
Overall System Management
Routing, Traffic Engineering, Mobility, VPN
DPI, Load Balance, SPAM
New Services
North Bound API
(Remote) Open Source Control Plane
South Bound API
Linux
Linux
Linux
Linux
Linux
Server
VM
Switch
Radio
PIC

Software Defined Network (SDN)



SDN History

- SDN is not completely new idea
  - Active Networks: programmable networks
  - Click Router: modular router
  - IETF ForCES: decoupling routing from forwarding
  - Cisco Nexus 7000: modularization of control & forwarding
- SDN uniqueness: provides programmability through decoupling of control and data planes.

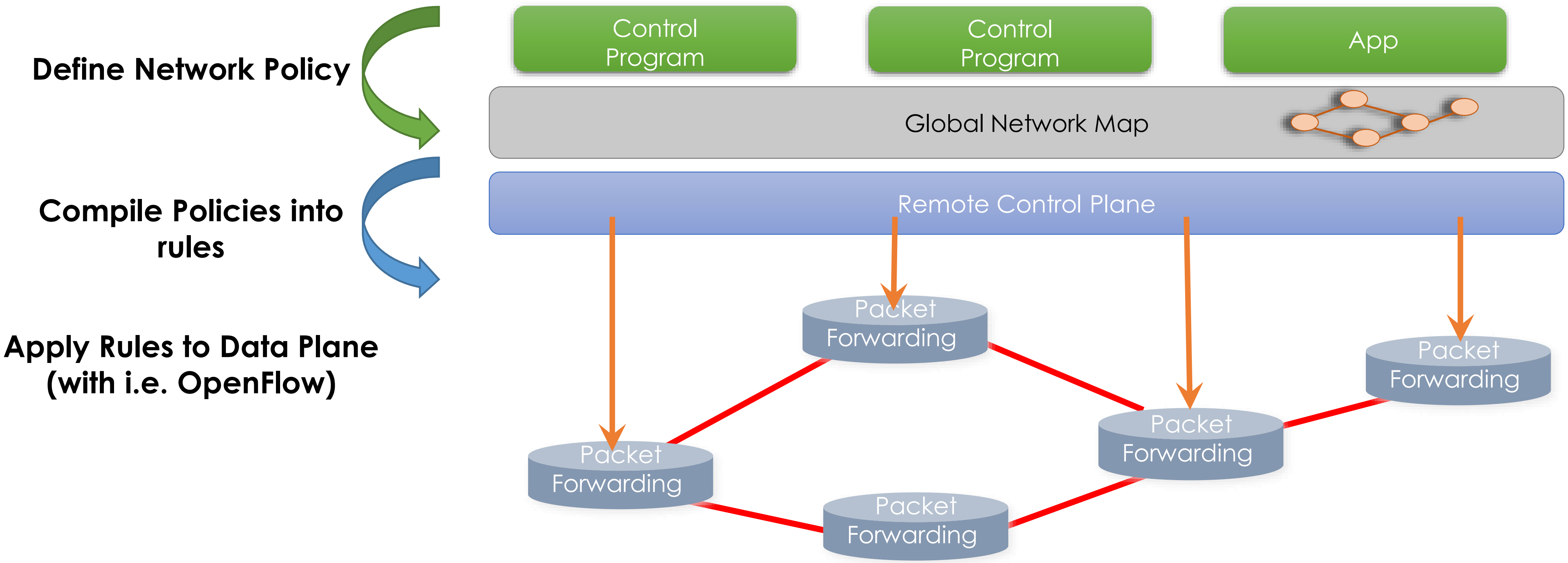
SDN Benefits

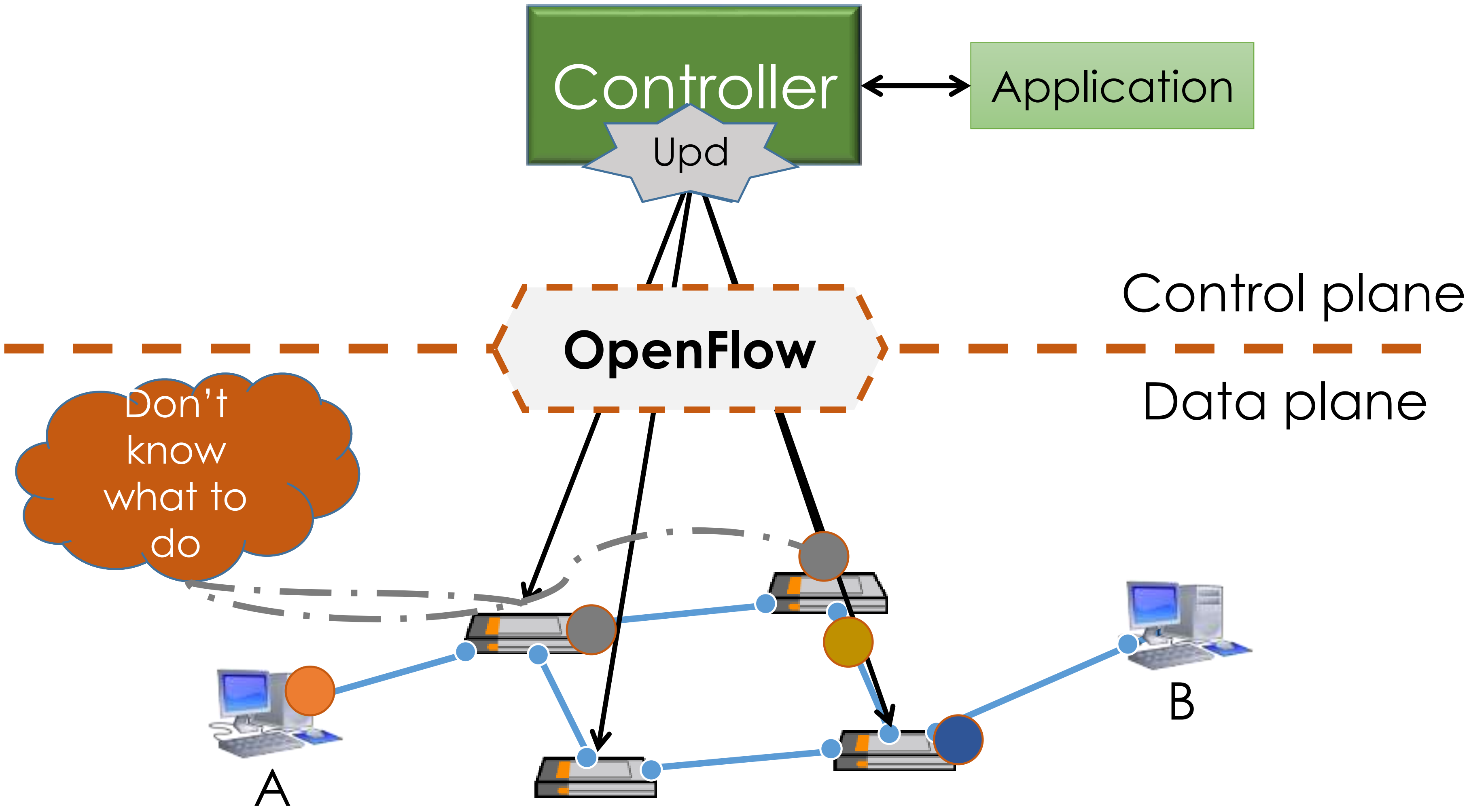
- Programmability
  - One central place for solving problems
  - VS. one protocol for each problem!
- Configuration
  - Automated & centralized validation
  - VS. manual error-prone config
- Performance
  - Dynamic global control with cross-layer information
  - VS. limited information

SDN Benefits

- Innovation
  - Easy software implementation for new ideas
  - Affordable, easily achievable test environment
  - Quick deployment with software upgrade
  - VS.
  - Difficult & lengthy hardware implementation
  - Limited test environment
  - Long standardization process

Proactive Rule Placement



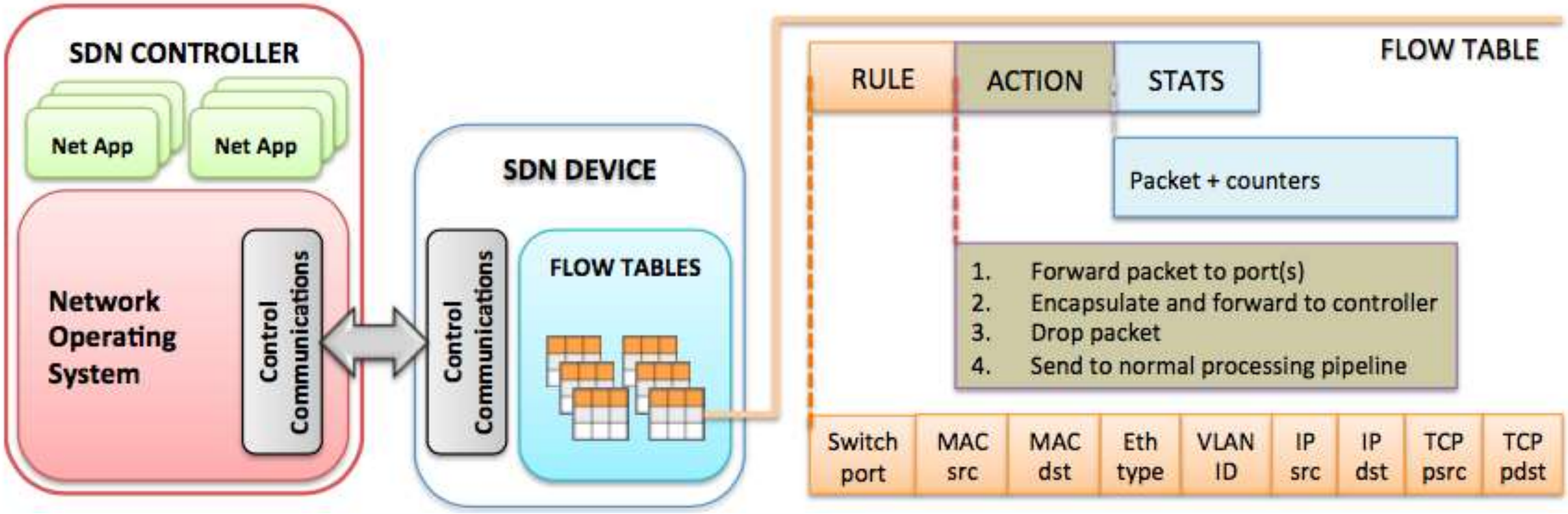

Controller
Upd
Application
Control plane
OpenFlow
Data plane
Don't know what to do
A
B

Reactive Rule Placement

- Why Reactive paradigm?
  - Switches have limited TCAM for storing rules
  - More dynamicity
- However
  - On-the-fly installation process may take time and yield delay spikes.
  - If a high number of new flows are aggregated at the end of switches, significant overhead can be yielded at both the control plane and data plane.

Research Challenges

Research Challenges – Data Plane

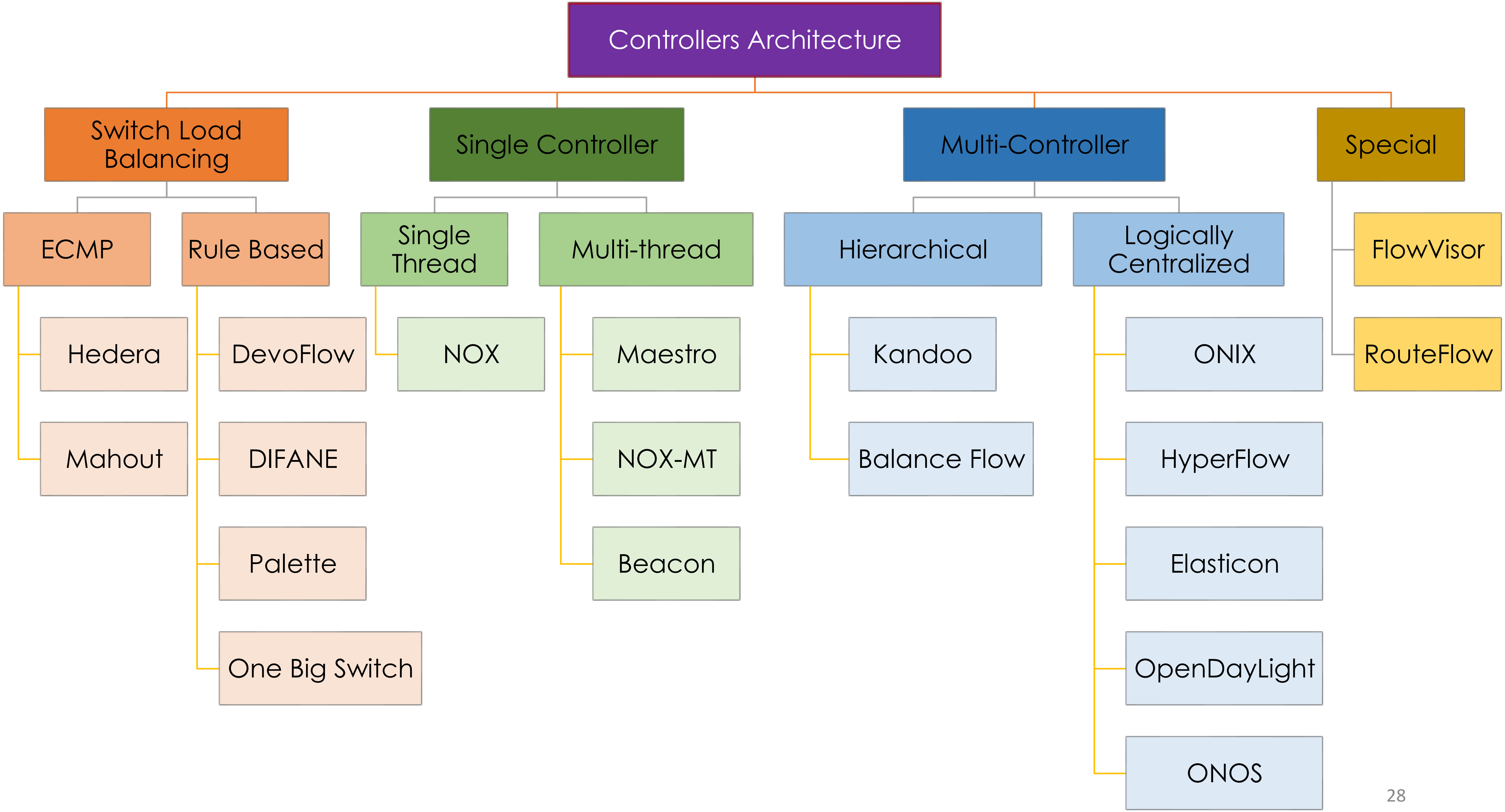


Research Challenges – Data Plane

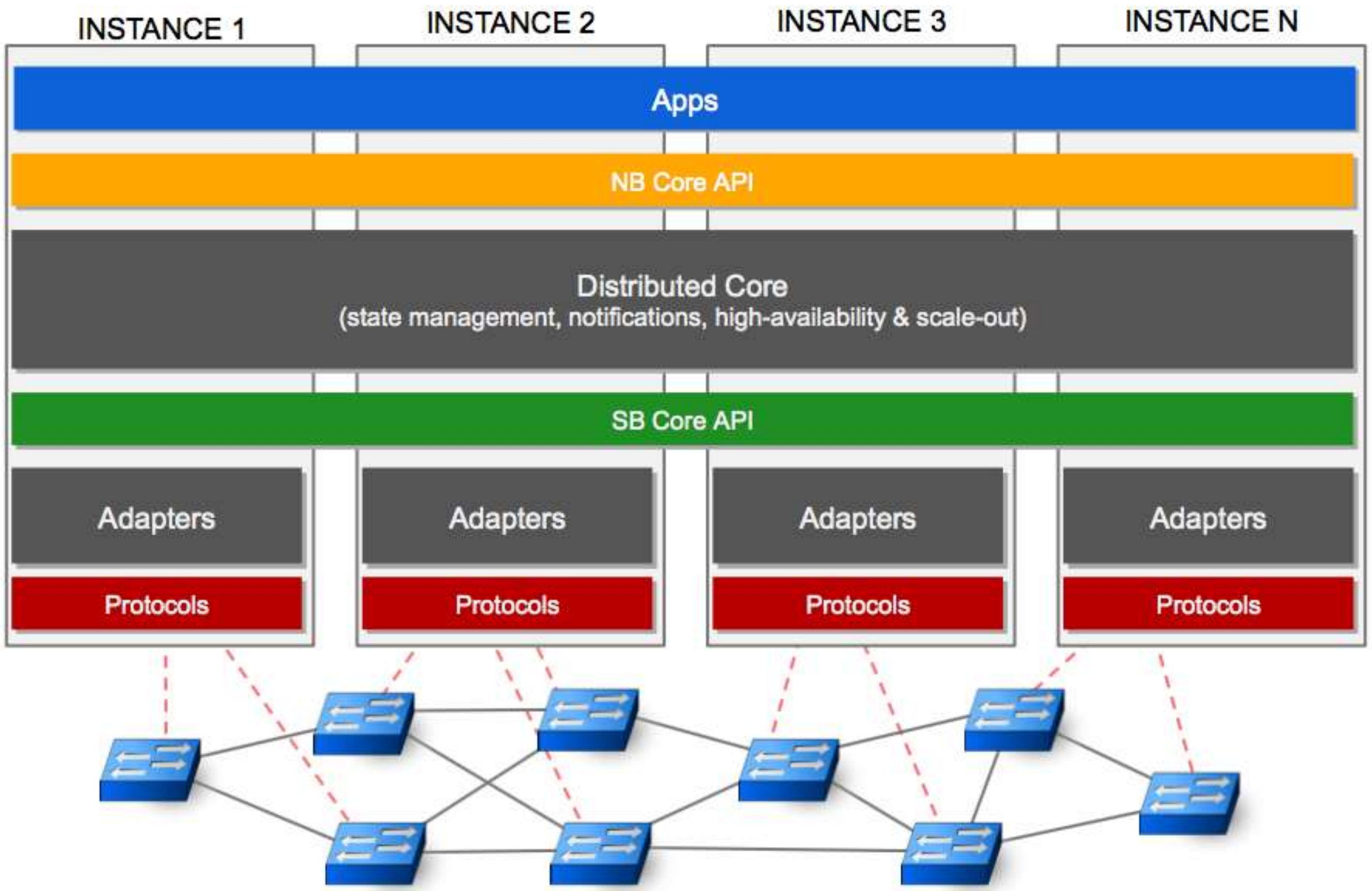
- Data Plane Performance
  - SDN rules can use up-to 13 factors of passing packets
  - Rules may contain wildcards (?, *)
- Solutions
  - Design native white-box switches with special CPU or accelerators
    - Pipelining, extended finite state machines, …
  - Use TCAM
    - TCAM (Ternary CAM) are small (4K-32K), expensive and power hungry
    - Compress rules to efficiently use TCAM limited space
  - Caching
    - Which rules? How many rules? Replacement Strategy?
  - Distributing rules in several switches

Research Challenges – Control Plane

- Performance
  - Better IO libraries
  - Compressing rules
  - Multithreading
- Scalability
  - Reactive or Proactive
    - Reactivity introduce remarkable challenges
      - Delays, buffering next packets, load of controller, …
  - Architectural consideration
    - Hierarchical
    - One logical controller, multiple consistent physical controllers
    - Distribute rules in switches


Controllers Architecture
Switch Load Balancing
Single Controller
Multi-Controller
Special
ECMP
Rule Based
Single Thread
Multi-thread
Hierarchical
Logically Centralized
FlowVisor
RouteFlow
Hedera
Mahout
DevoFlow
DIFANE
Palette
One Big Switch
NOX
Maestro
NOX-MT
Beacon
Kandoo
Balance Flow
ONIX
HyperFlow
Elasticon
OpenDayLight
ONOS

ONOS - Logically Centralized Controller



Research Challenges – Control Plane

- Scalability (cont.)
  - Elasticity
    - Reacting to imbalanced loads, delays, faults, …
      - Elasticon, Pratyaastha, our research
- Performance Evaluation
  - Measuring different aspects of a network in operation
    - Comparing architectures
      - Based packet sizes, rule models and sizes, reactivity, impact on delay, CPU load
    - Comparing accelerators
    - Comparing different configurations
    - Comparing different scenarios with traditional networks
    - Special data center considerations

Research Challenges – Control Plane

- Performance Evaluation (cont.)
  - Measuring different aspects of a network in operation
    - Designing and using benchmarks
      - OFLOPS, cbench
    - Sampling methods
    - Finding evidences from gathered data (data mining)
    - Monitoring networks at scale
  - Controller Placement Problem
    - Within different topologies
  - Developing analytical models
    - Few works on this, but growing

Research Challenges – Control Plane

- Interoperability
  - Designing High-level specific languages (Network DSL)
- Resiliency
  - When a controller fails? Link or port down
  - Architecture-specific solutions
- Load Balancing
  - Between controllers
  - Between switches

Research Challenges – Control Plane

- QoS
    - Shaping, metering, scheduling, multi-layer
    - Bandwidth on demand
    - Adaptive routing
    - Resource migration (VMs in cloud)
- Routing
- Supporting legacy IPv4 networks
- Supporting legacy protocols
    - MPLS over SDN

Research Challenges – Security

Threat vectors	**Specific to SDN?**	**Consequences in software-defined networks**
Vector 1	no	Open door for DDoS attacks.
Vector 2	no	Potential attack inflation.
Vector 3	yes	Exploiting logically centralized controllers.
Vector 4	yes	Compromised controller may compromise the entire network.
Vector 5	yes	Development and deployment of malicious applications on controllers.
Vector 6	no	Potential attack inflation.
Vector 7	no	Negative impact on fast recovery and fault diagnosis.

Research Challenges – Security

- OpenFlow networks are subject to
  - spoofing
  - Tampering
  - Repudiation
  - Information disclosure
  - DoS
  - Elevation of privileges
  - Assumption that all applications are gentle and will not affect SDN operation
  - The lack of isolation, protection, access control

Research Challenges – Security

Attack	Security Property	Examples
Spoofing	Authentication	MAC and IP address spoofing, forged ARP and IPv6 router advertisement
Tampering	Integrity	Counter falsification, rule installation, modification affecting data plane.
Repudiation	Non-repudiation	Rule installation, modification for source address forgery.
Information disclosure	Confidentiality	Side channel attacks to figure out flow rule setup.
Denial of service	Availability	Flow requests overload of the controller.
Elevation of privilege	Authorization	Controller take-over exploiting implementation flaws.

Research Challenges – Security

- Measuring security issues in various controllers
- Attack detection mechanisms
- Devising defending strategies against various threat vectors
- Comparing controllers in defensing capabilities
- Comparing various defense strategies
- Designing and proposing access control models and/or hardening guidelines
- Evaluating and finding new threat vectors

Research Challenges – SDE

- Software-Defined Environment
  - Offer a programmatic access to the environment as a whole
    - selecting the best available resources based on the current status of the infrastructure
    - Enforcing defined policies independent of the technology and underlying network
    - Workloads can be easily and automatically assigned to the appropriate IT resources based on application characteristics, security and service level policies
  - Building Block
    - Software-defined Network (SDN)
    - Software-Defined Storage (SDS)
    - Software-Defined Compute (SDC)
    - Software-Defined Management (SDM)

Research Challenges

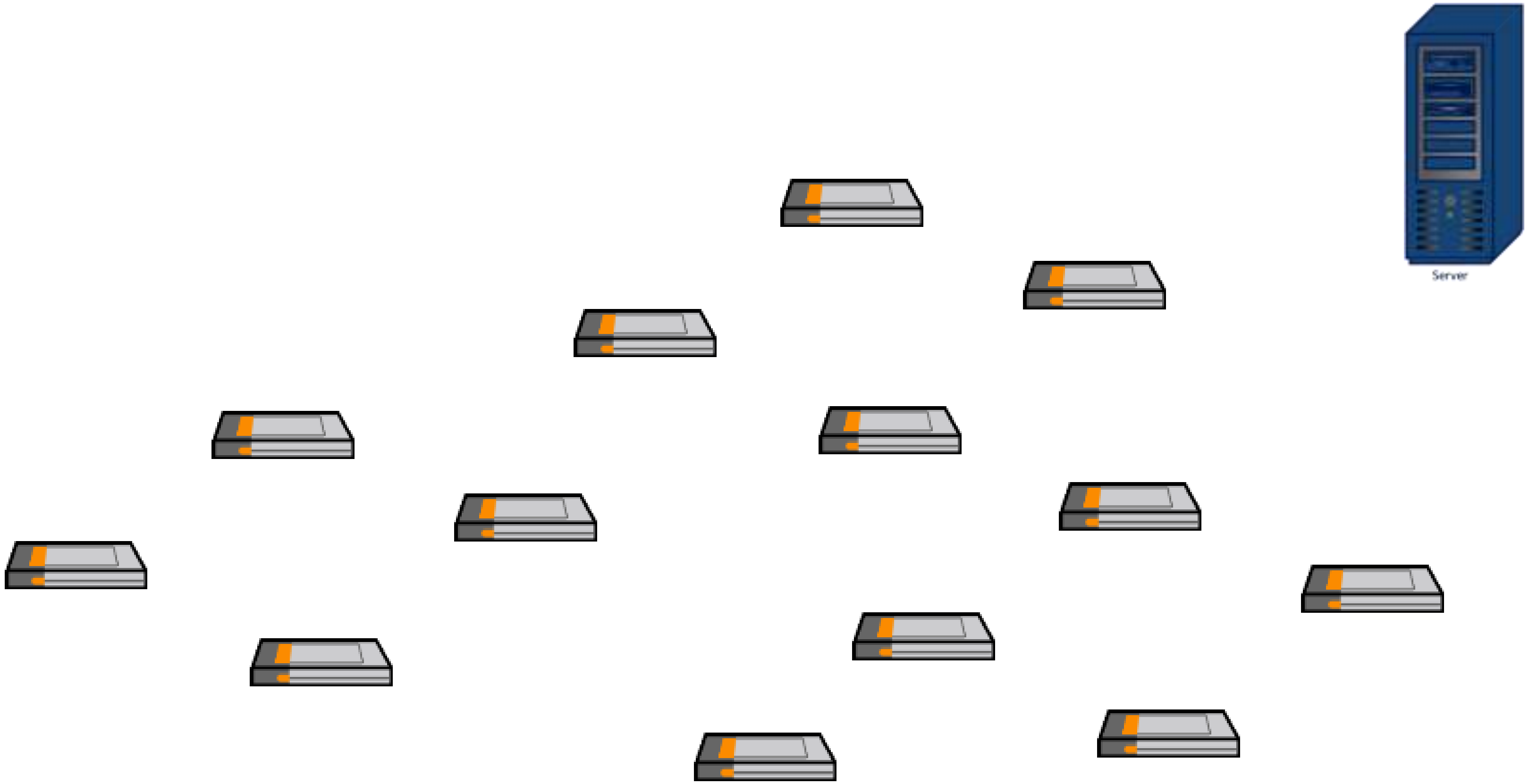
- Green Networking
  - Energy efficient protocols, mechanisms, applications, ...
- SDN for Sensor Networks
  - SDN-Wise
- SDN for IoT
- SDN for Vehicular Adhoc Networks (VANET)
- SDN for Mobile Adhoc Networks (MANET)

Current Research

Preliminaries

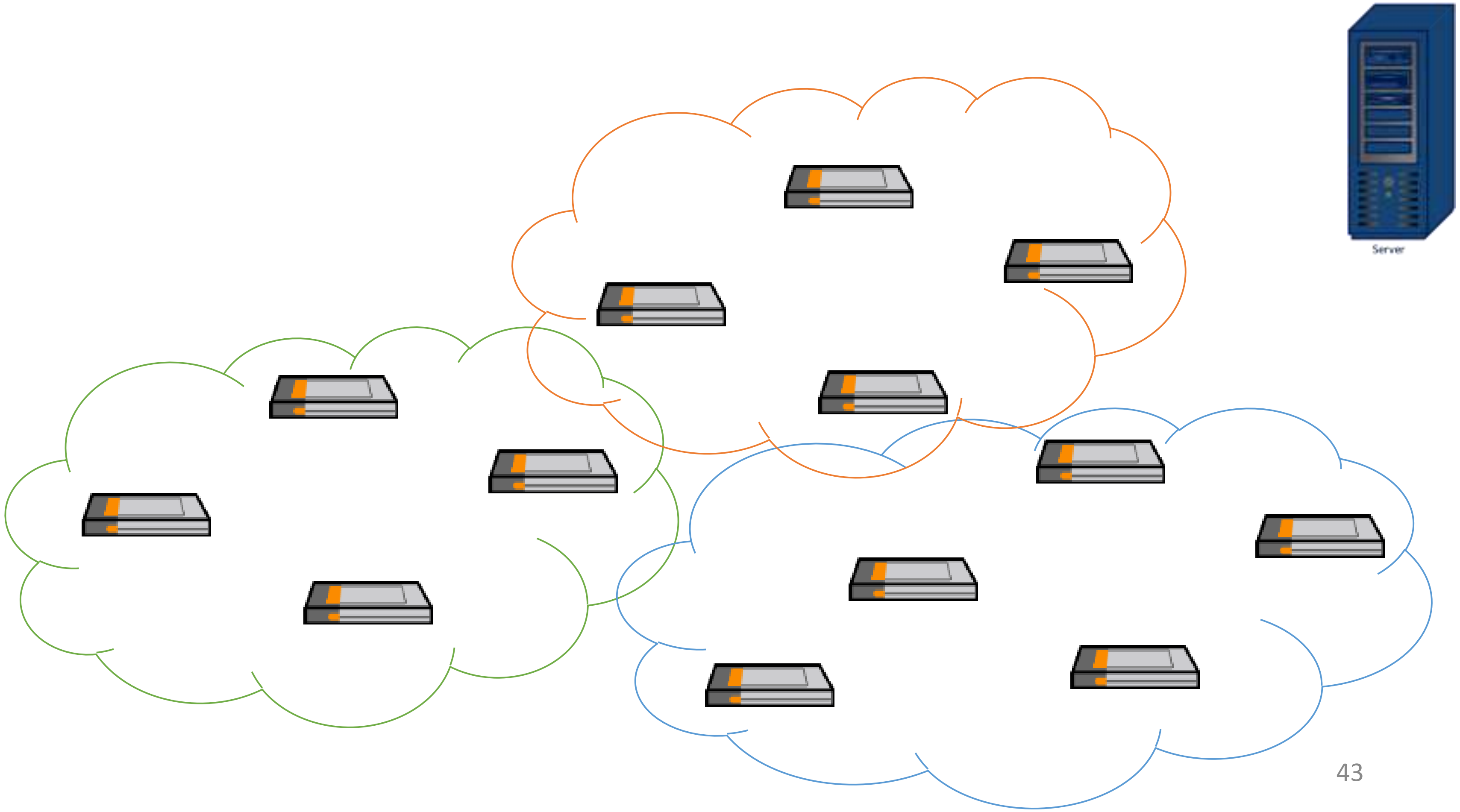
- Large networks cannot be designed based on reactive operations
- Controller must be used for
  - Declaring policies and applying to network
  - Reacting to incidents
  - Rules should be distributed on switches
    - Switches do the routine routing operation
- Issues
  - How rules should be distributed among switches?
  - What if the number of rules exceed the TCAM size of a switch?
  - How break rule table?
  - What if load of a switch become higher than others (load balancing)

Hierarchical Elastic Scalable SDN



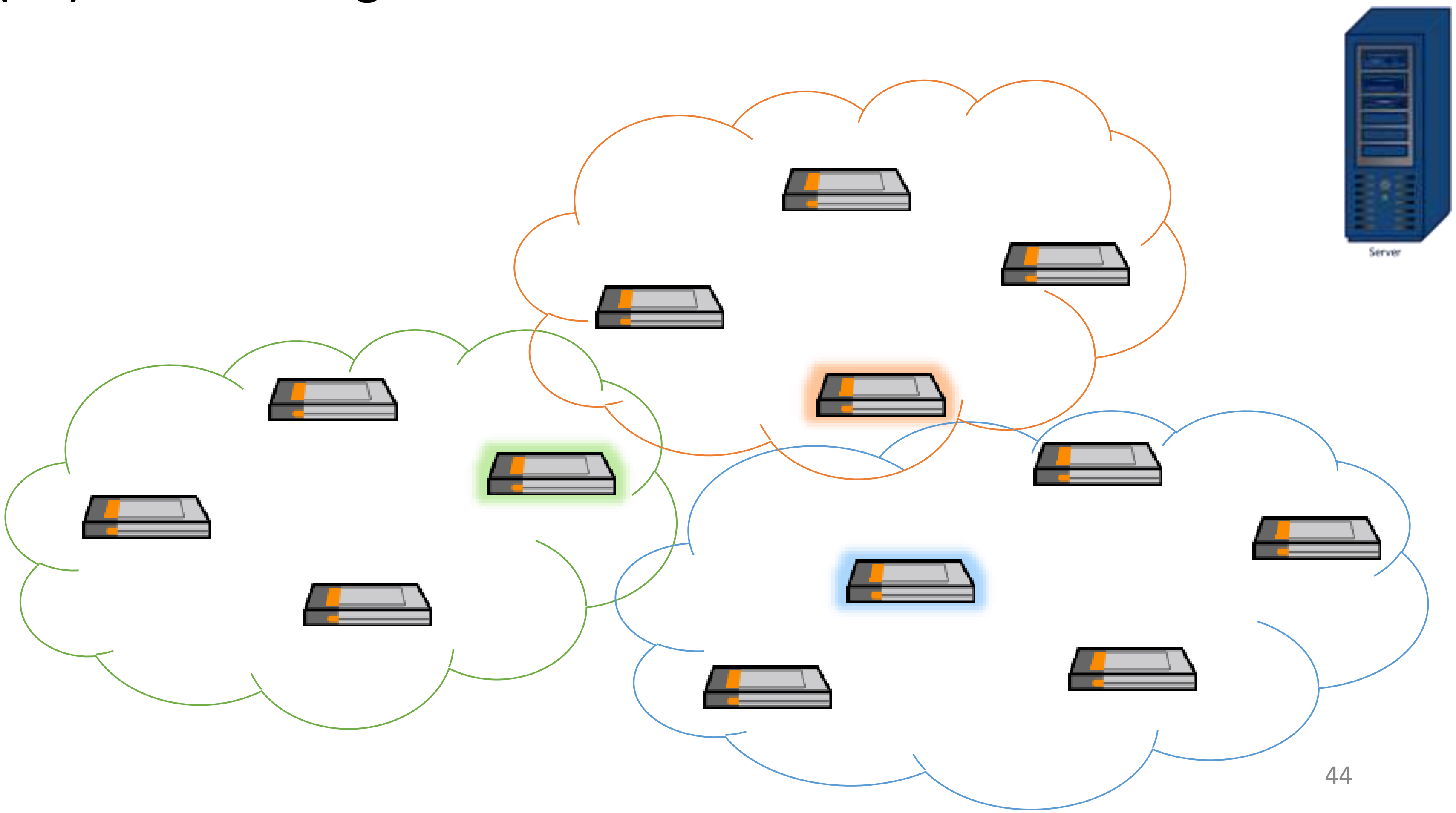
Hierarchical Elastic Scalable SDN

- Group switches in regions



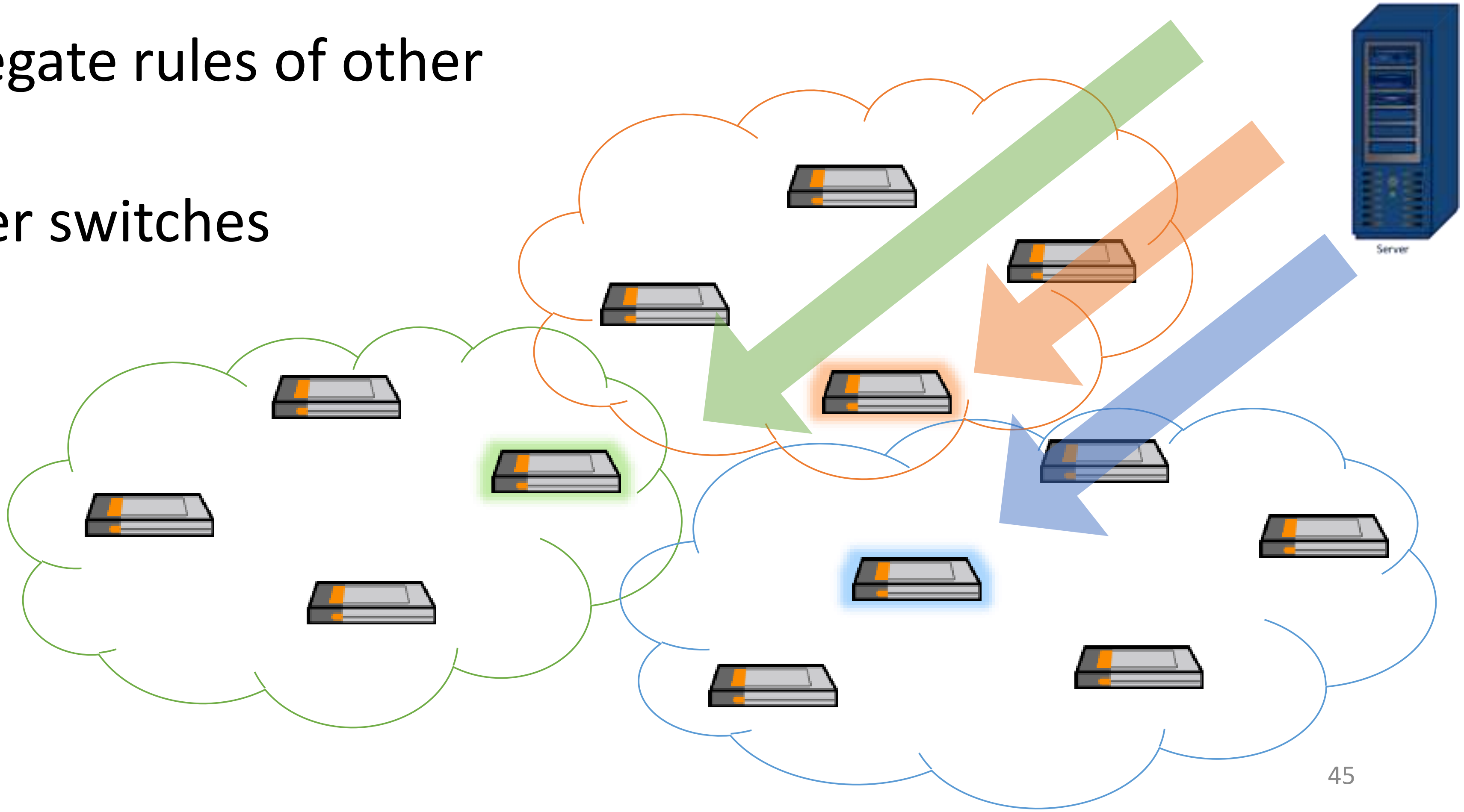
Hierarchical Elastic Scalable SDN

- Select Leader Switch (LS) in each region



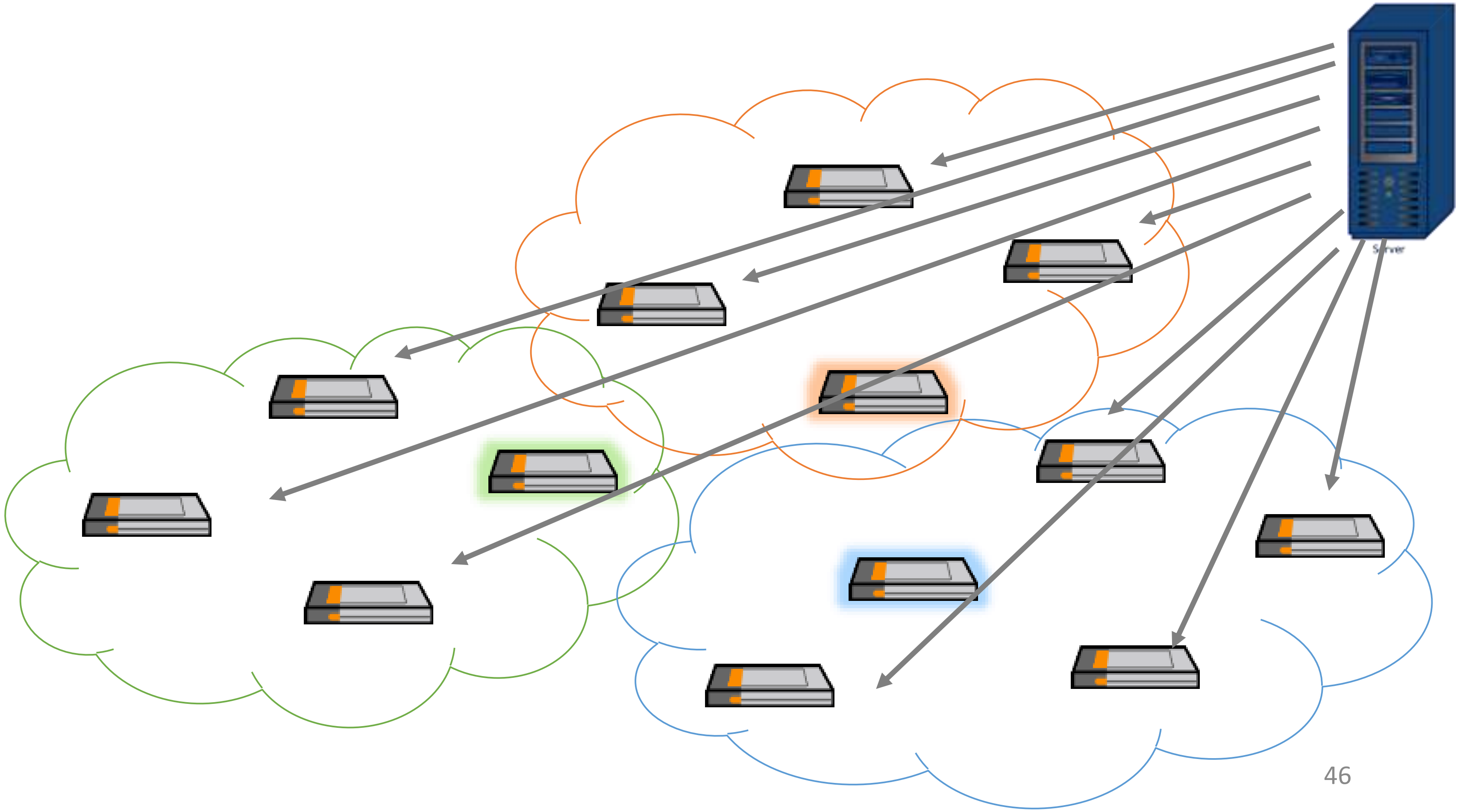
Hierarchical Elastic Scalable SDN

- Push Aggregate rules to LS
- LS knows about aggregate rules of other regions
- LS acts like BGP border switches



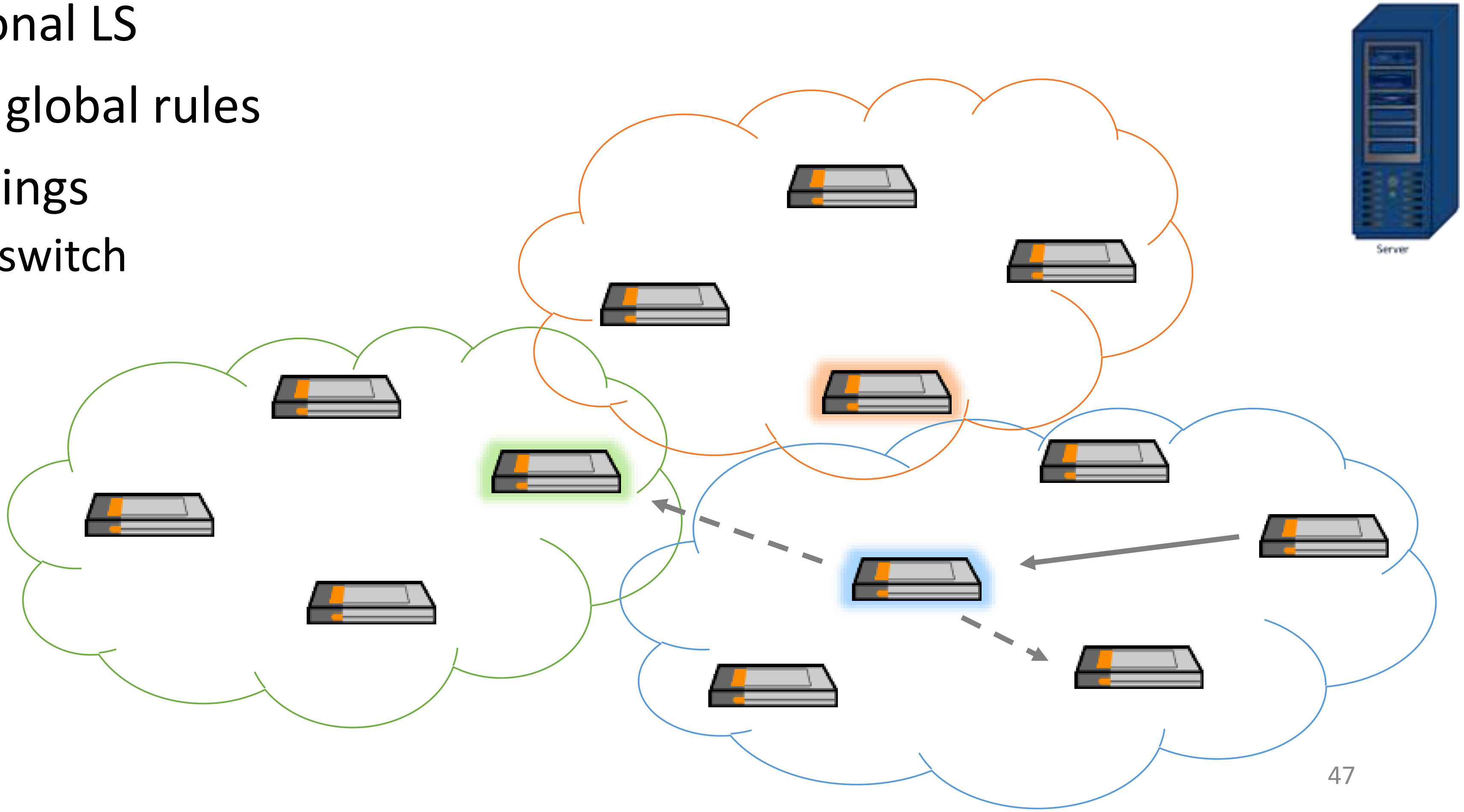
Hierarchical Elastic Scalable SDN

- Push fine-grained rules to normal switches (NS)



Hierarchical Elastic Scalable SDN

- When a NS could not match a packet with its rule table sends it to the regional LS
- LS knows the aggregated global rules
- LS may do one the followings
  - Sends it back to a local switch
  - Sends it another LS

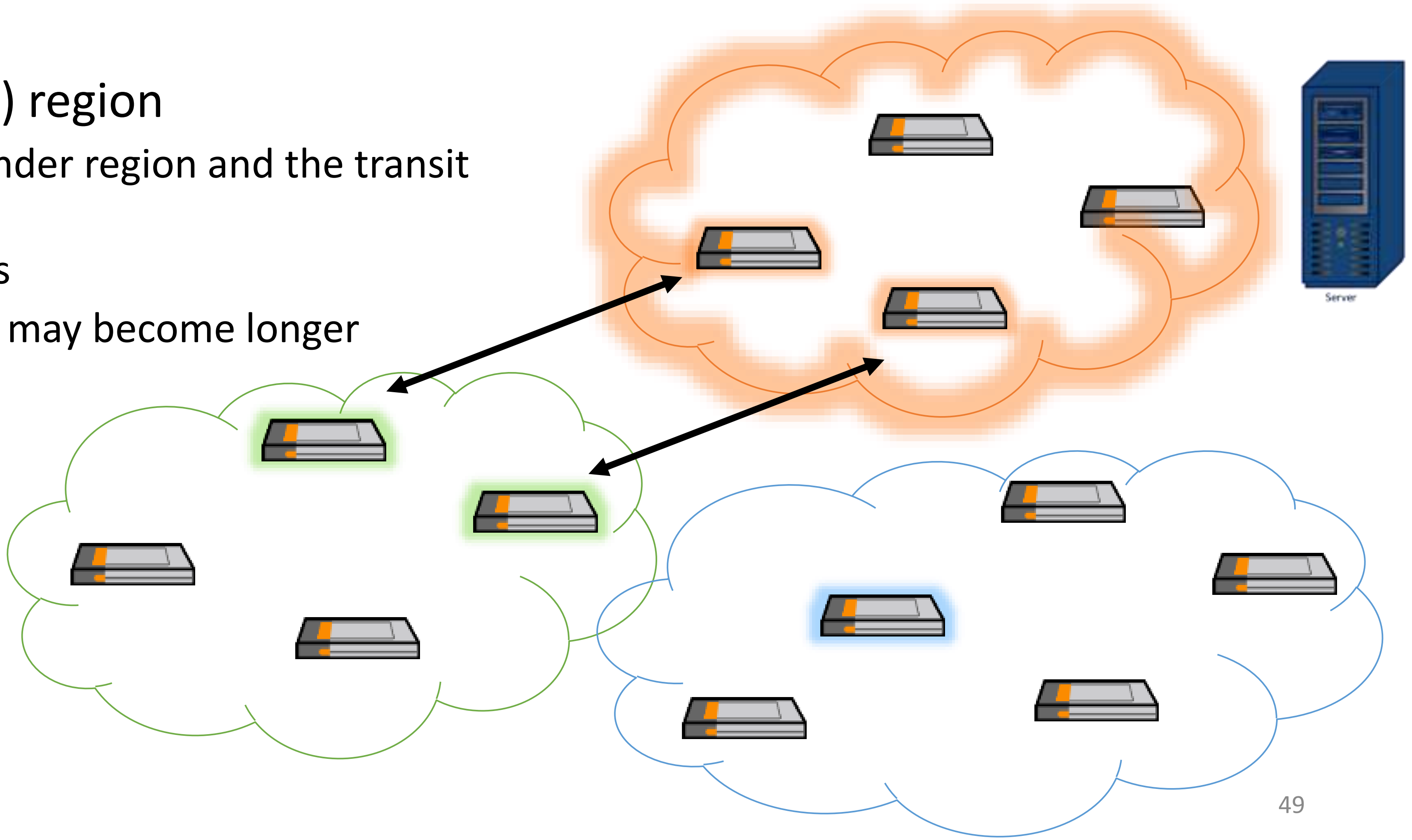


Hierarchical Elastic Scalable SDN

- Issues
  - Small TCAMs in switches
    - Break rules into smaller tables and distribute among several NSes
  - High load on a LS
    - Controller always monitors the network
    - When a LS get overloaded
      - Add another LS
      - Change table of some of NSes to direct their traffic toward the new LS

Hierarchical Elastic Scalable SDN

- Issues (cont.)
  - High load in a (transit) region
    - Add new LS in the sender region and the transit region
    - Connect the new Lses
    - Communication path may become longer



Hierarchical Elastic Scalable SDN

- Issues (cont.)
  - Most of the traffic use long paths
    - Add or replace LS
  - LS placement
    - Place LS to minimize
      - Path length
      - Delay
- Evaluation
  - We tested the ideas on mininet + floodlight controller

Distributing Tables in Switches

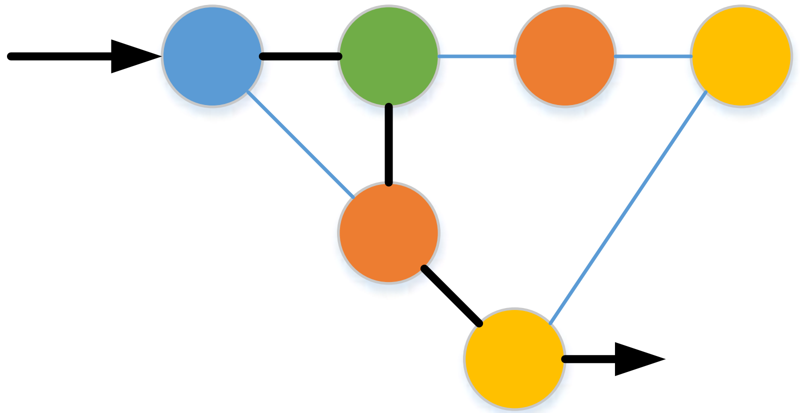
- We can break a big rule table into smaller parts
- However, breaking table is not easy
  - Longest prefix matching rule is the main issue
- Breaking Algorithm
  - Determine α points between 1 and 32
  - $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_\alpha\} \quad and \quad \alpha_1 < \alpha_2 < \cdots < \alpha_\alpha$
  - Expand all rules with prefix length $< \alpha$ to the length α
  - Break the big rule table into $\alpha + 1$ sub-tables
- Expanding Example:
  - 11* & $\alpha_1 = 4$ → we obtain 4 rules: 1100*, 1101*, 1110*, 1111*

Distributing Tables in Switches

- Determining α
  - Choose α in a way that traffic load of sub-tables become balanced
- Distribute sub-tables among switches in a way that packets visit all sub-tables
  - When a packet matches with rule in the sub-table with longest prefix (α_α) then packet destination is determined
  - In fact, all packets *MUST* visit this table

Distributing Tables in Switches

- This is a variant of graph coloring problem
- Rainbow Path Problem
  - For a flow-set $F = \{f_1, \dots, f_n\}$ and α colors
  - Color all paths with α colors in a way that all paths contain all colors



References

- Nick McKoewn Talks, http://yuba.stanford.edu/~nickm/talks/
- Xia, Wenfeng, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. “A Survey on Software-Defined Networking.” *IEEE Communications Surveys & Tutorials* 17, no. 1 (2015): 27–51.
- Kreutz, Diego, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. “Software-Defined Networking: A Comprehensive Survey.” *Proceedings of the IEEE* 103, no. 1 (2015): 14–76.

Thank You