Applications of Software-Defined Networking (SDN) in Power System Communication Infrastructure: Benefits and Challenges

Jasson Casey and Alex Sprintson

Texas A&M University

(jasson.casey@tamu.edu and spalex@tamu.edu)

PSERC Webinar
February 17, 2015

About Flowgrammable

- What we do
  - Advocate and educate on SDN technologies
  - Mentor students
  - Conduct original research
  - Create/maintain open source projects & training

TEXAS A&M UNIVERSITY

- Who we are
  - Students: undergraduates, masters, PhDs
  - Researchers: profs, post docs
  - Engineers: hardware, software, systems, network

The University of Akron

Flowgrammable
Driving the Next SDN Generation

SDN Opportunity

- Global IT spending ~ 3.8T USD in 2014 (Gartner)
  - Telecom spend ~ 1.6T USD
  - SDN has the potential to affect all aspects of telecom
  - SDN is in early adoption in the telecom industry

- SDN simplifies network design, integration, operations
  - Commoditizes network hardware
  - Standardizes new network service development
  - Reduces planning risk
  - Widens the labor pool for engineering services

Networking's Stubborn Pressure on Productivity and IT Budgets

- TCP/IP is 40 years old
- Ethernet is 40 years old
- Most network hardware is COTS
- Five ODMs control 80%+ of designs
- Gross margins are still 60%+

Traditional Networks

- Packets are directed according to forwarding rules
  - Determined by distributed algorithms
    - Such as OSPF
  - Blackbox switches with pre-determined protocols

Drivers of SDN

- High cost of development
  - Large multi-disciplinary teams
  - Large scale custom software development
  - Difficult to verify and validate
- Increased risk of failure
  - Complex development increases the cost of change
  - Custom architecture discourages industry adoption
- Nontrivial time investment
  - Typically multi-year activities while need is immediate
  - Commits your resources for extended periods of time

Drivers of SDN

- Complex modern network services
  - VoIP, IPTV, Mobile WiFi, 3G/4G
  - Data Centers, Private/Public Cloud Data Centers

- Multi-device multi-vendor architectures
  - Many device types necessary
  - Multi-vendor options reduce supplier risk and price
  - Enabled by open network protocols and architecture

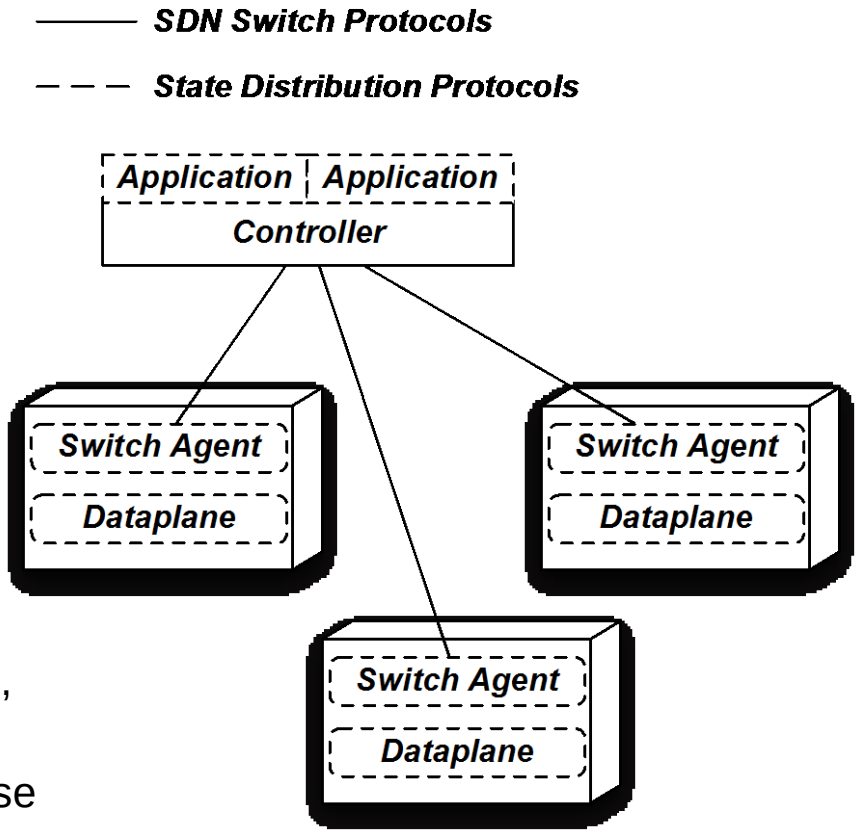
Drivers for SDN

- Poor utilization of physical resources
  - Specialized equipment forces topologies/hierarchies
  - Fragmentation of equipment resources
  - Choices in scaling are limited

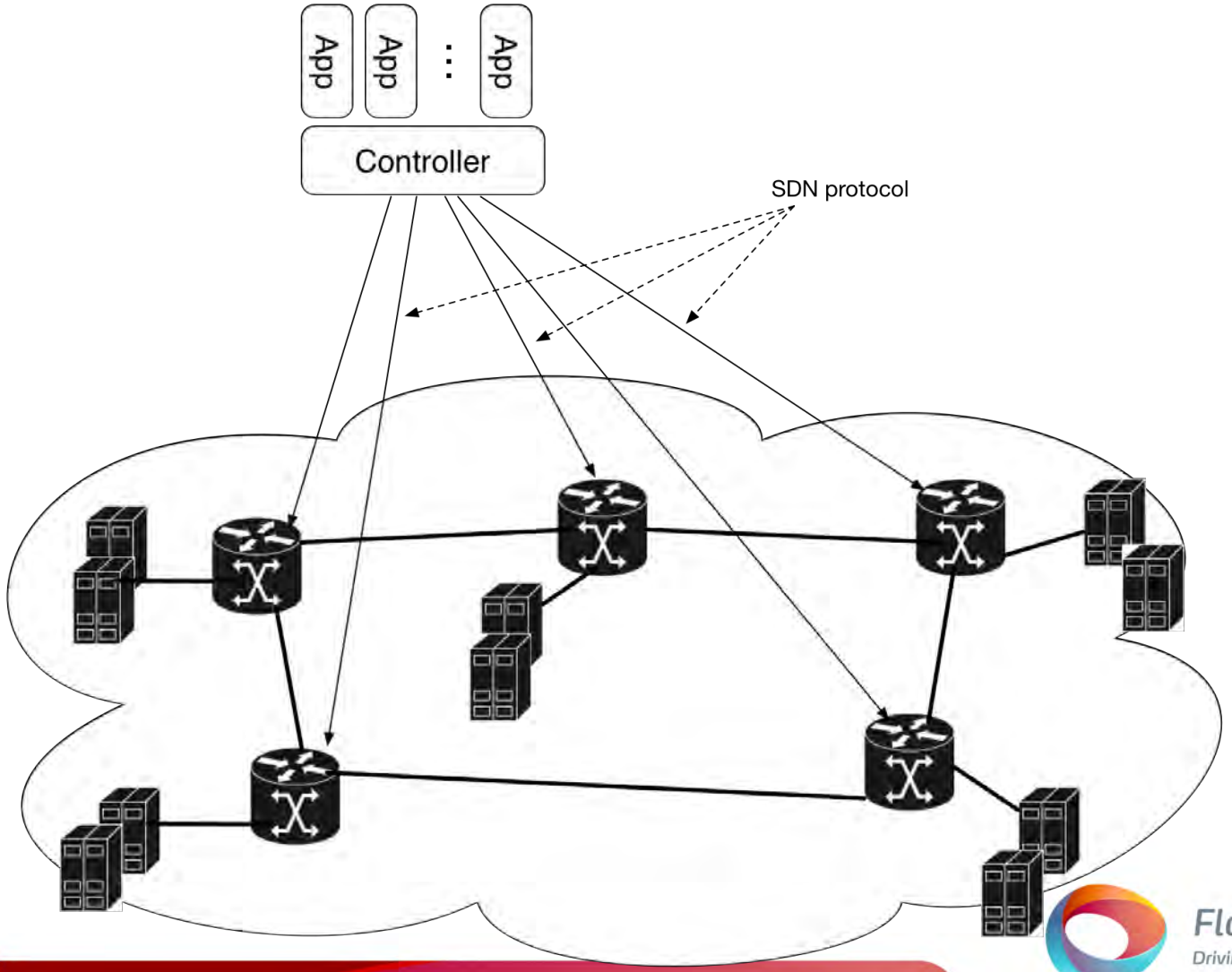
- Traffic monitoring is an afterthought
  - Uniform monitoring not possible across all equipment
  - Poor flow granularity
  - Monitoring can severely degrade performance
  - Specialized monitoring equipment changes the architecture

Initial SDN Architecture

- Whitebox switch
  - Commodity Ethernet switch
  - Programmable data plane
  - Control plane proxy (Switch Agent)
- Controller
  - Centralized control plane
  - Manages whitebox switches
  - Hosts network applications
- Application
  - View of network wide state: capabilities, configuration, and statistics
  - Modify network configuration in response to changing network wide state
  - Protected environment for programming



Initial SDN Architecture



Software Defined Networking (SDN)

- SDN provides …
  - Unified interface for network control
  - User definable network behaviors
  - Elimination of proprietary configuration languages
  - Increases the accessibility of network operations

- Enables Organizational Agility
  - Enables rapid network service prototyping
  - Simplifies large scale system integration
  - Reduces vendor dependence for custom behaviors

Operations Environment Tradeoffs

Non SDN Environment

- Each vendor and device type introduces a unique control interface and operational process
- Network behaviors are defined by the vendor and not operator modifiable
- SMEs with deep vertical specialization are required to design, deploy, and operate complex networks

SDN Environment

- All vendors and device types support a standardized interface and operational process
- Network operators can define custom network behaviors as well as extend Vendor specified behaviors
- Technology is more accessible to the larger labor pool of networking generalists

Energy Communications Networks

- Enable traffic engineering and security applications
  - Dynamic re-routing of flows
    - Based on load or failure scenarios
  - Security inspection of certain flows
  - Help improve reliability and robustness
- Flow prioritization
  - Priority for real-time messages (e.g., GOOSE traffic)

Use Case: Substation Networks

- Legacy solutions: copper wires and proprietary communication protocols
  - High costs, lack of flexibility
  - Difficult to maintain
- Ethernet-based solutions
  - Network-enabled IEDs
  - Difficult to make changes in the standards when new technology is needed
  - Hard to predict future requirements (as with IEC 61850)
  - Security concerns

Use Case: Substation Networks

- Challenge – a substation can maintain hundreds of different IEDs
- Increased network complexity
  - IEC 61850, PTP, DNP 3.0, proprietary
  - Reliance on Layer-3 broadcast
- SDN benefits
  - Streamlines configuration and management
    - Removes the need for multiple VLAN
  - Traffic monitoring
  - Security
    - Through link isolation
  - Traffic engineering and congestion avoidance

Virtualization

- Take advantage of data center technology
- Use virtual IEDs run on a commodity hardware
- Simple interface to manage large numbers of IEDs
  - E.g., ERCOT – 100K IEDS on a 3,600 substation networks
  - Significant effort to manage manually
  - Error-prone
  - SDN approach provides a simple interface to configure IES

Cahn el al. Software-Defined Energy Communication Networks: From Substation Automation to Future Smart Grids

Flowgrammable
Driving the Next SDN Generation

Software Defined Networks (SDN)

- Two rough categories of SDN
  - Service Configuration
  - Service Definition

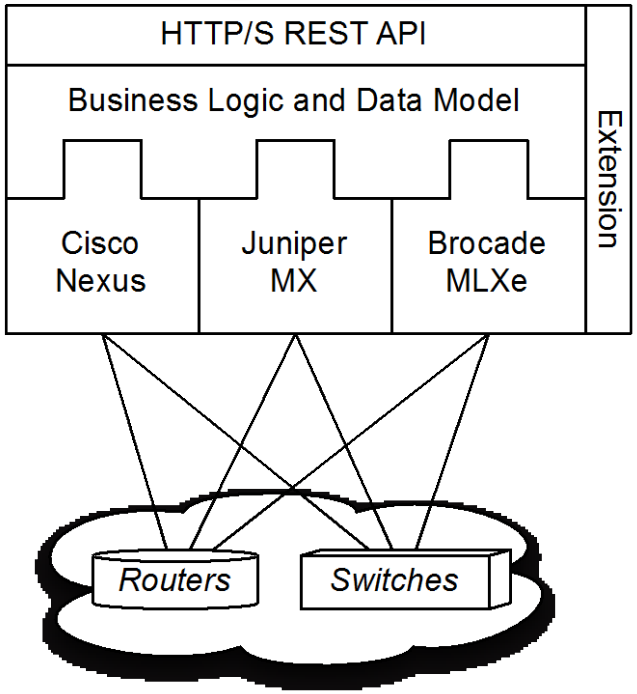
- Service Configuration - simplify process of configuring and operating complex networks

- Service Definition - enable the process of defining new network behaviors

Service Configuration

- Network abstractions
  - switch, router, firewall, load balancer
- Interface for manipulating abstractions

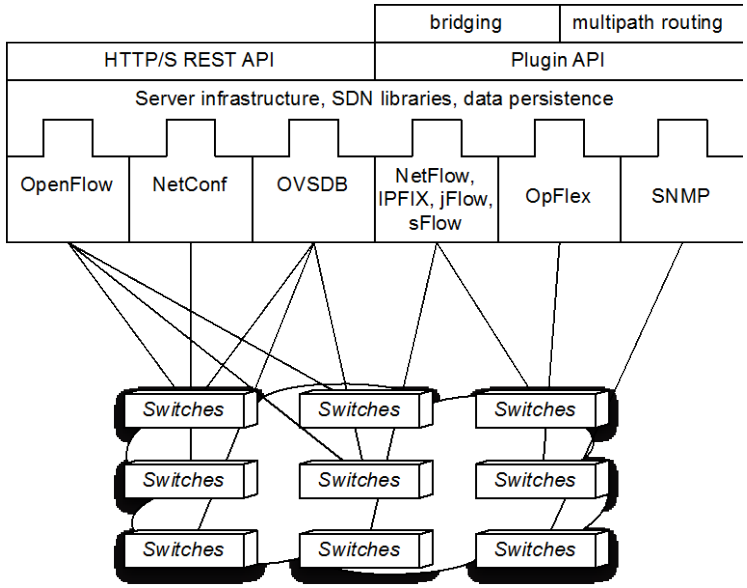
- Plugins that model network abstractions
- Libraries for common operations activities
  - Address allocation, key generation, etc.
- Standardized abstraction API
  - Operations exposed with HTTP/S REST API
  - Uniform authentication, authorization, and accounting
- Served from a configuration controller
  - OpenStack: Neutron
  - Juniper Contrails
  - Cisco onePK



Service Definition

- Data plane abstractions
  - port, flow table, meter, group
- Interface for manipulating abstractions

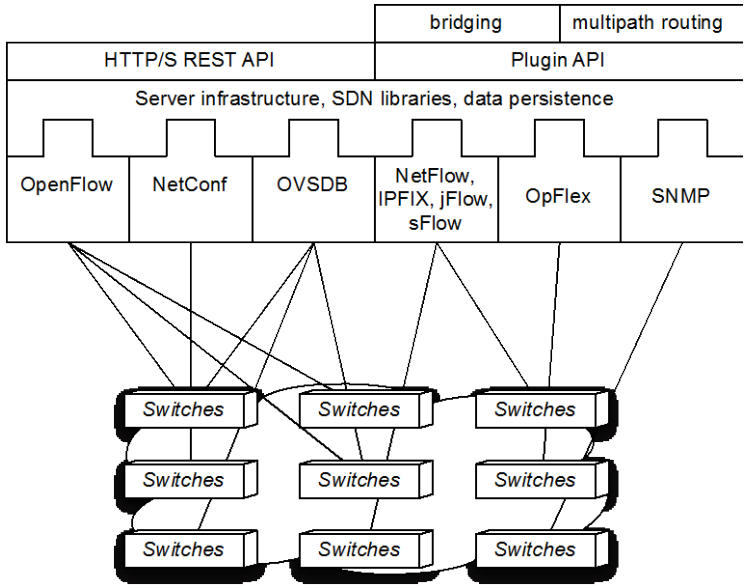
- Controller for hosting applications
- Plugins that model data plane abstractions
- Libraries for common activities
  - Topology discovery, bridging, routing, etc.



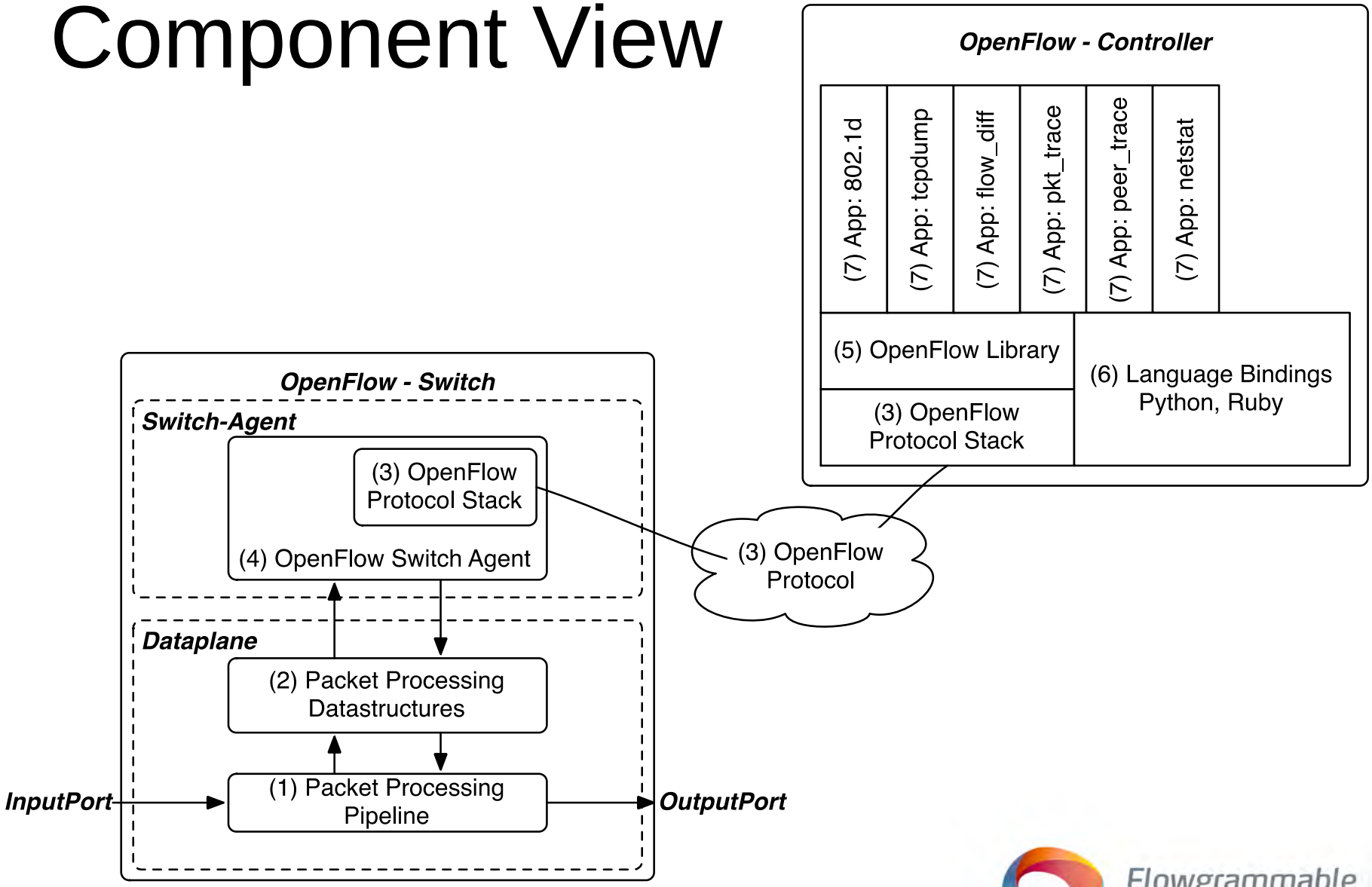
Service Definition

- Data plane abstractions
  - port, flow table, meter, group
- Interface for manipulating abstractions

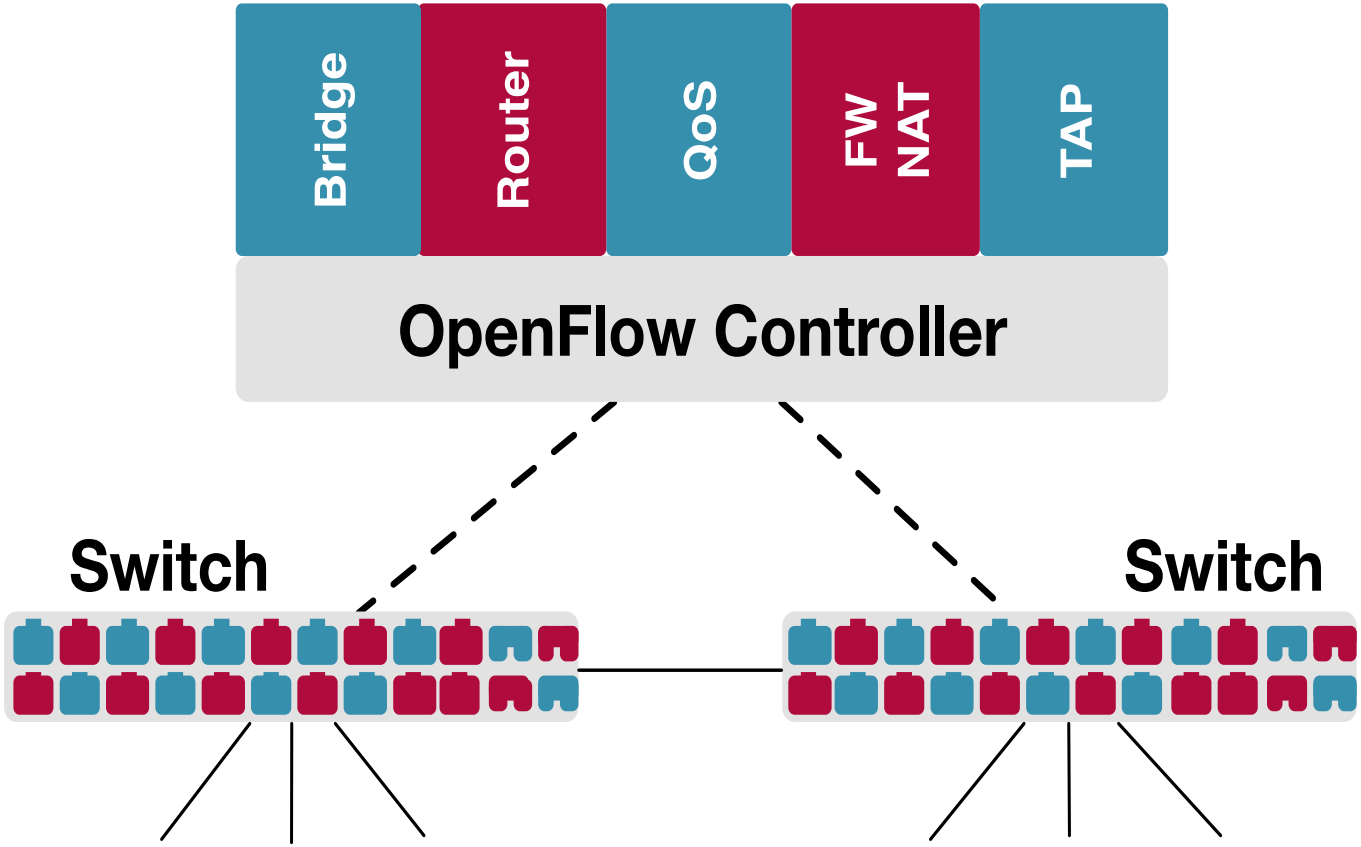
- Controller for hosting applications
- Plugins that model data plane abstractions
- Libraries for common activities
  - Topology discovery, bridging, routing, etc.



Component View



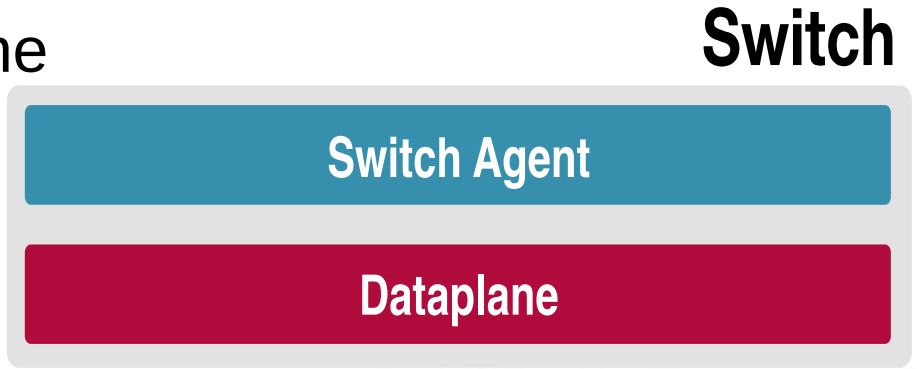
OpenFlow Architecture



Anatomy of a Whitebox Switch

- Switch Agent
  - Communicates with controller
  - Manages the dataplane
  - Provides feature offload

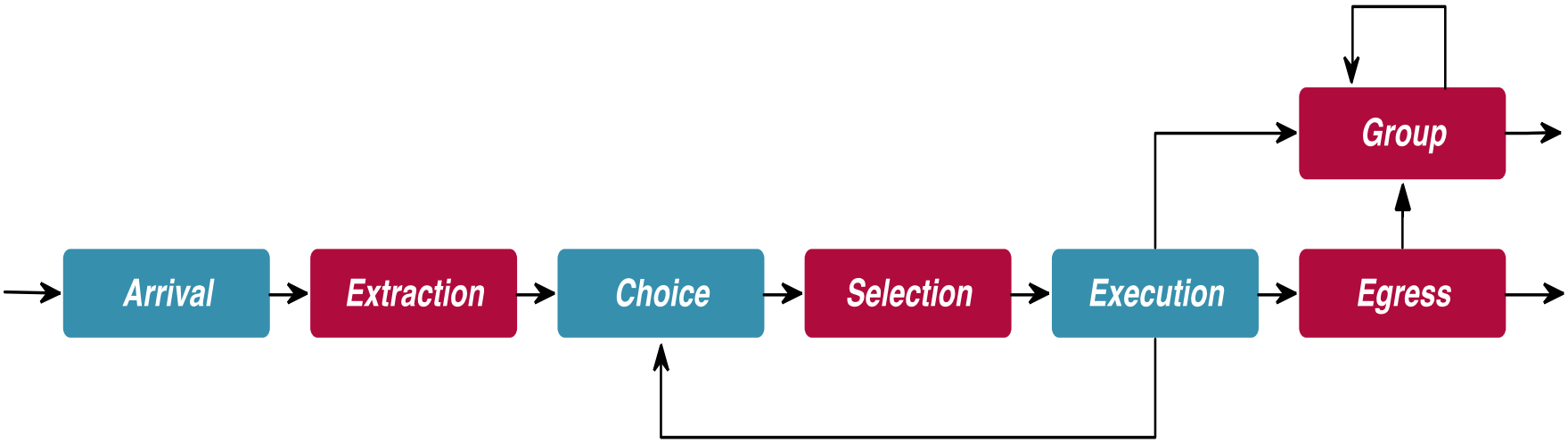
- Dataplane
  - Packet processing engine
  - Fast and efficient



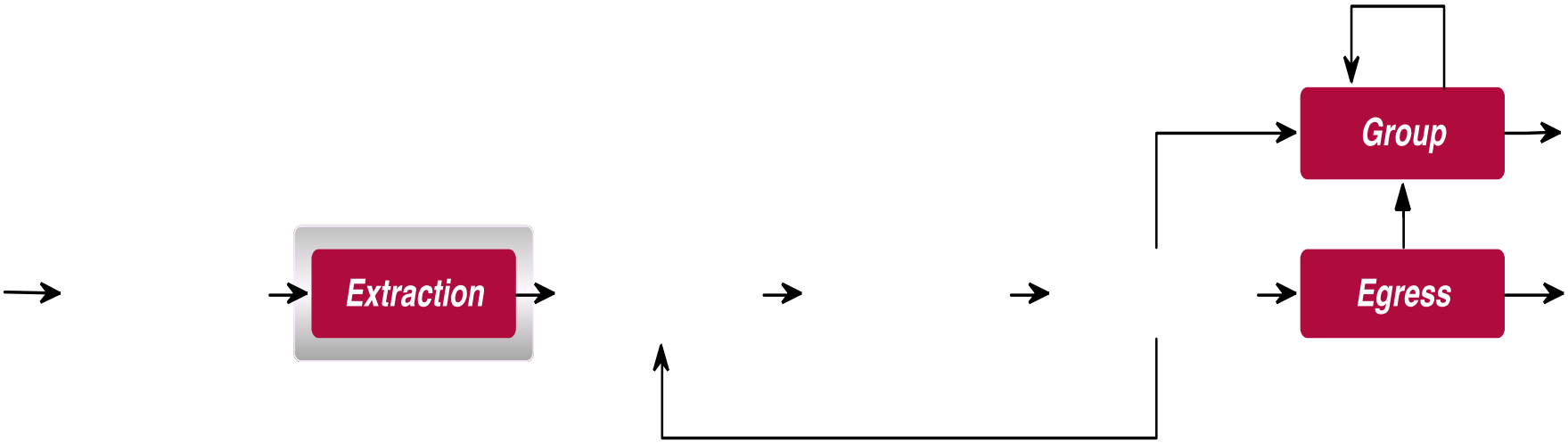
Applications for Power Systems

- Enables traffic engineering and security applications
  - Dynamic re-routing of flows
    - Based on load or failure scenarios
  - Security inspection of certain flows
  - Help improve reliability and robustness
- Flow prioritization
  - Priority for real-time messages (e.g., GOOSE traffic)

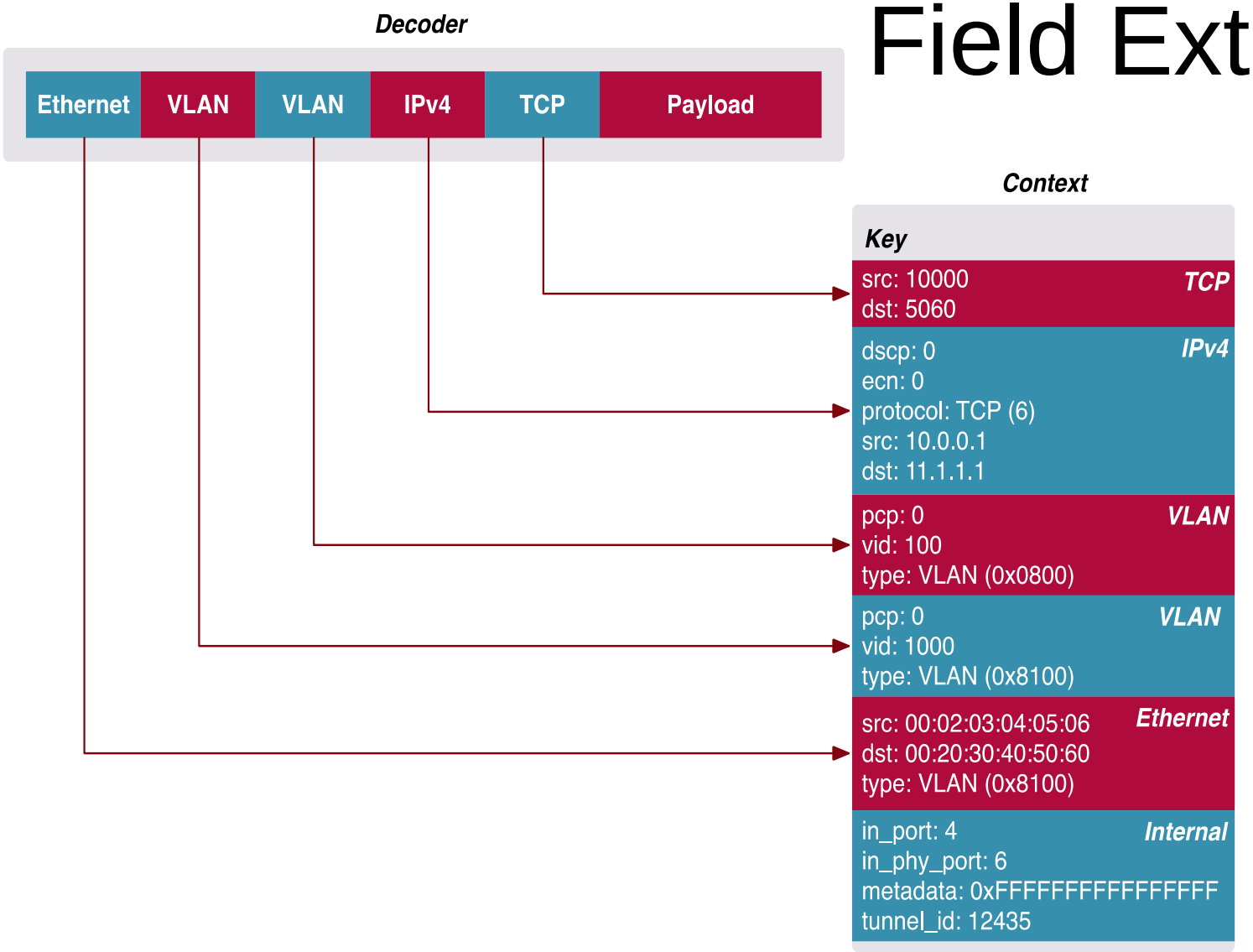
Dataplane Pipeline



Dataplane Pipeline - Extraction



Field Extraction



Dataplane Pipeline - Choice

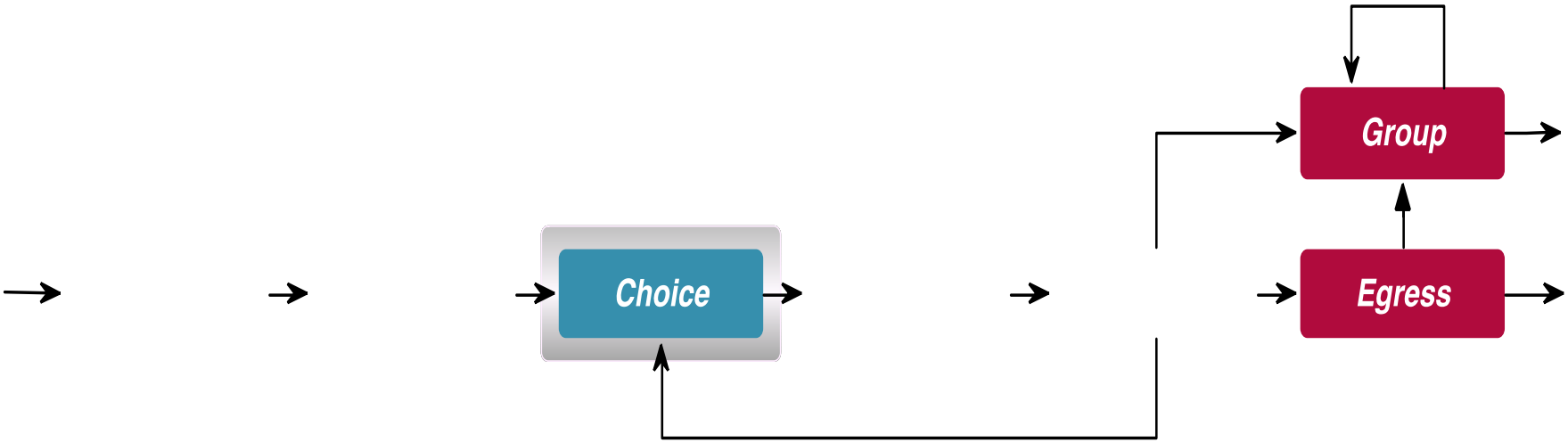
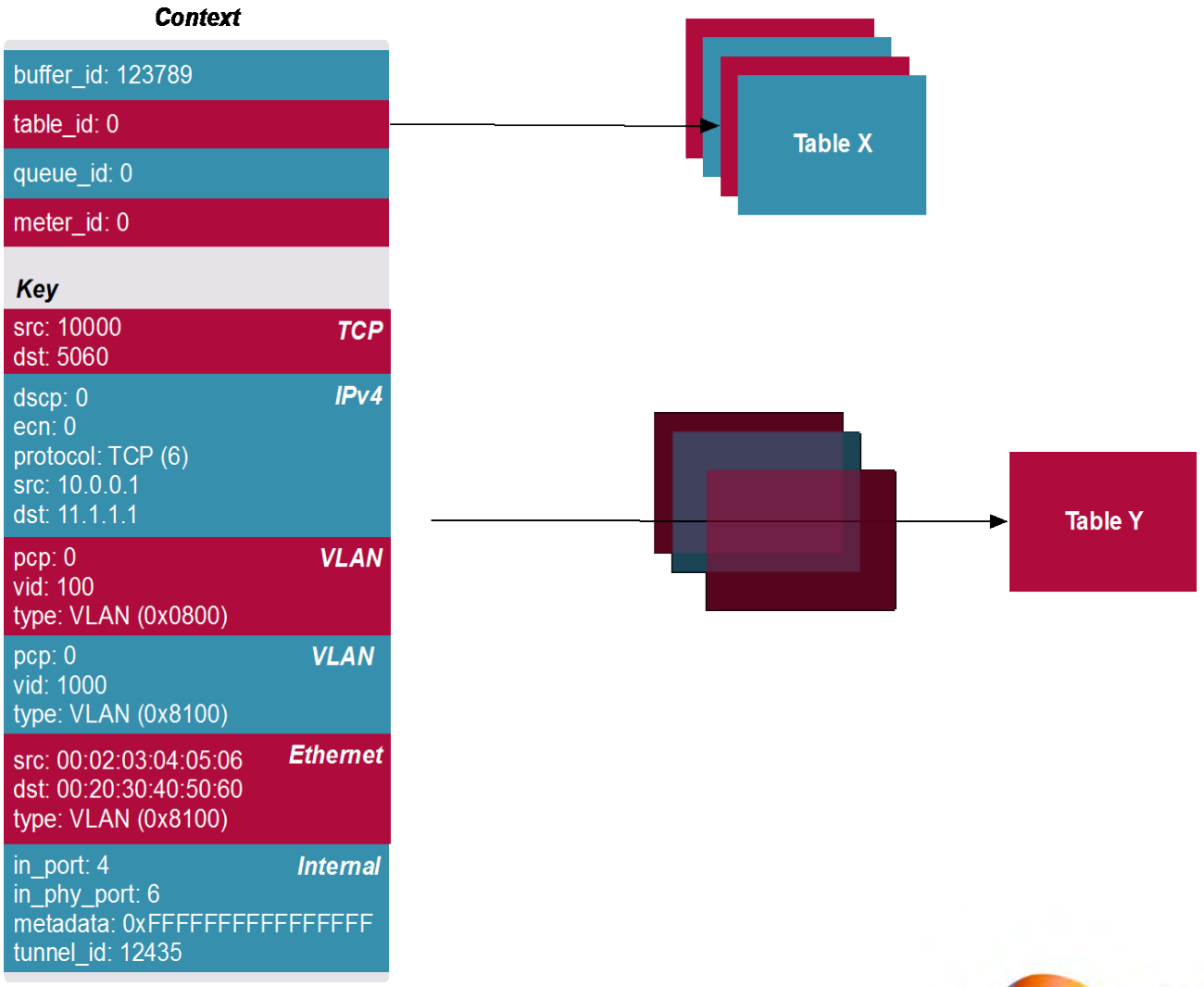
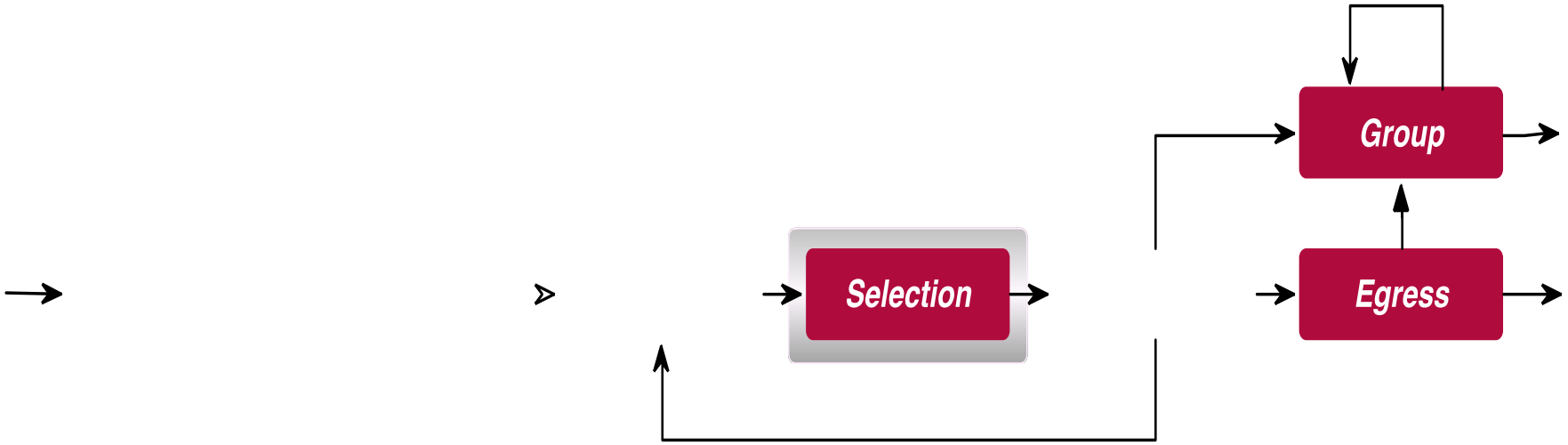


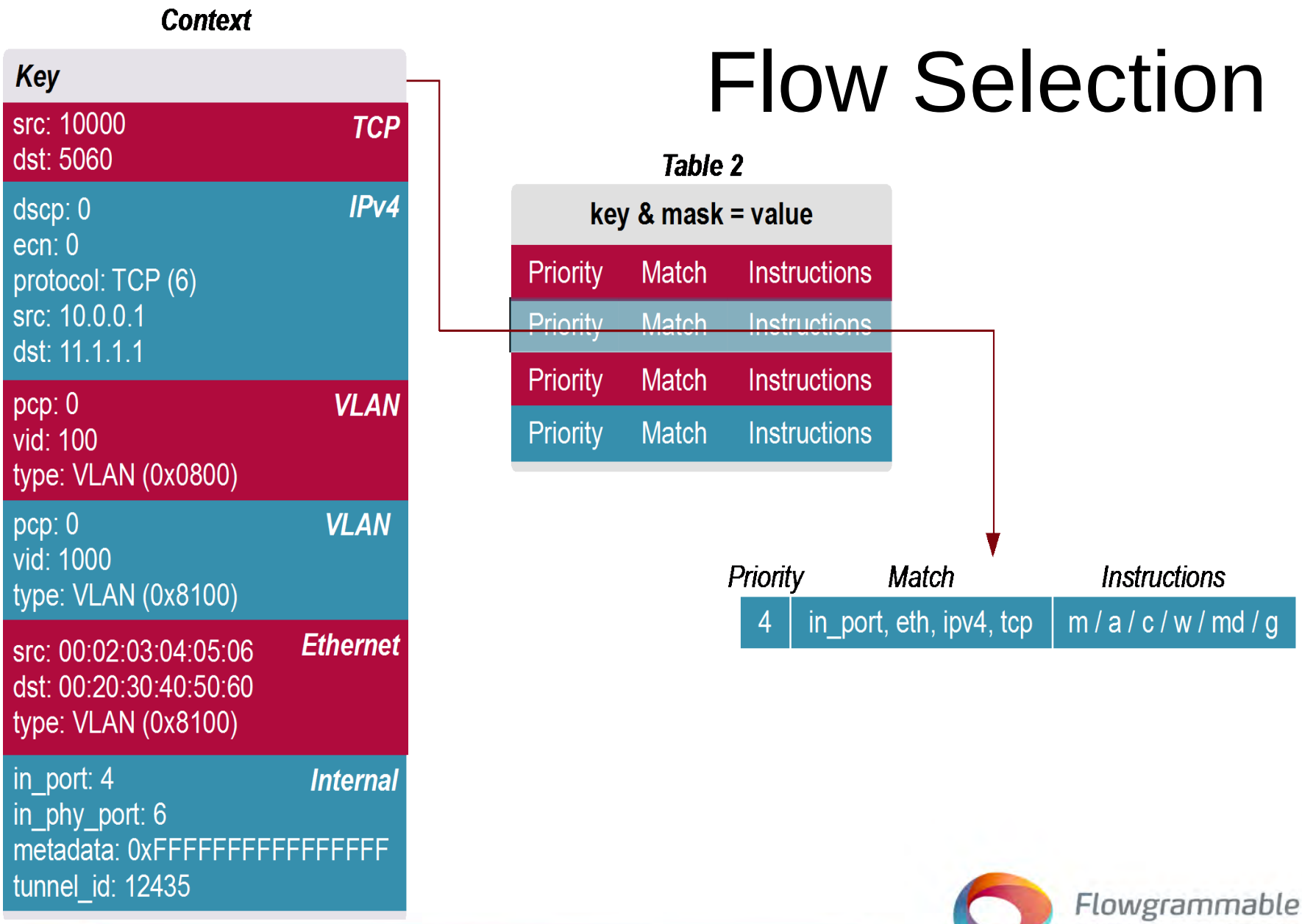
Table Choice/Selection



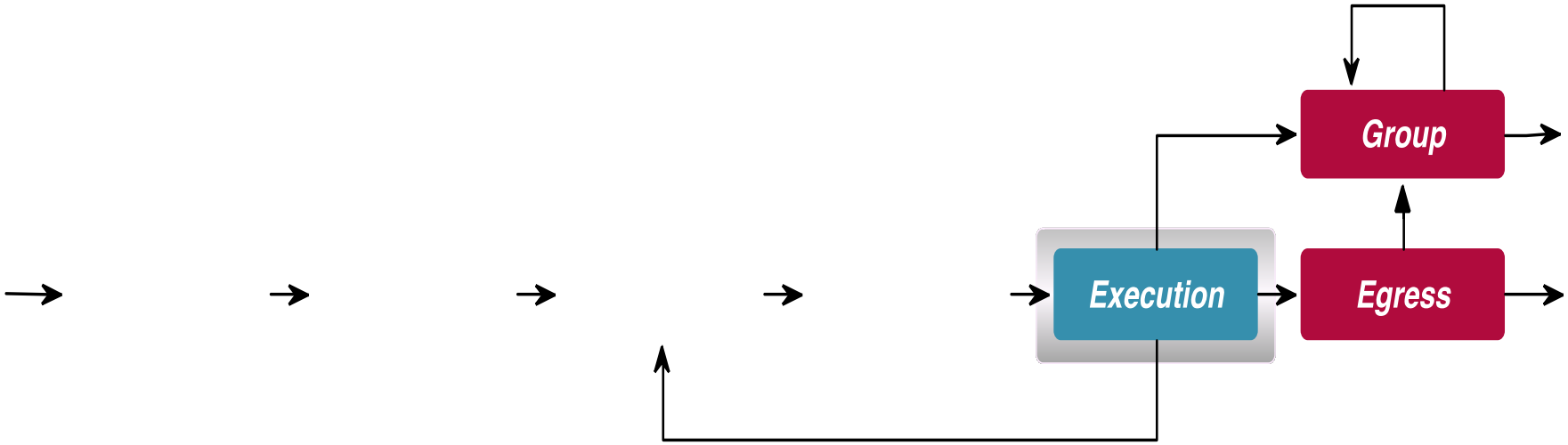
Dataplane Pipeline - Selection

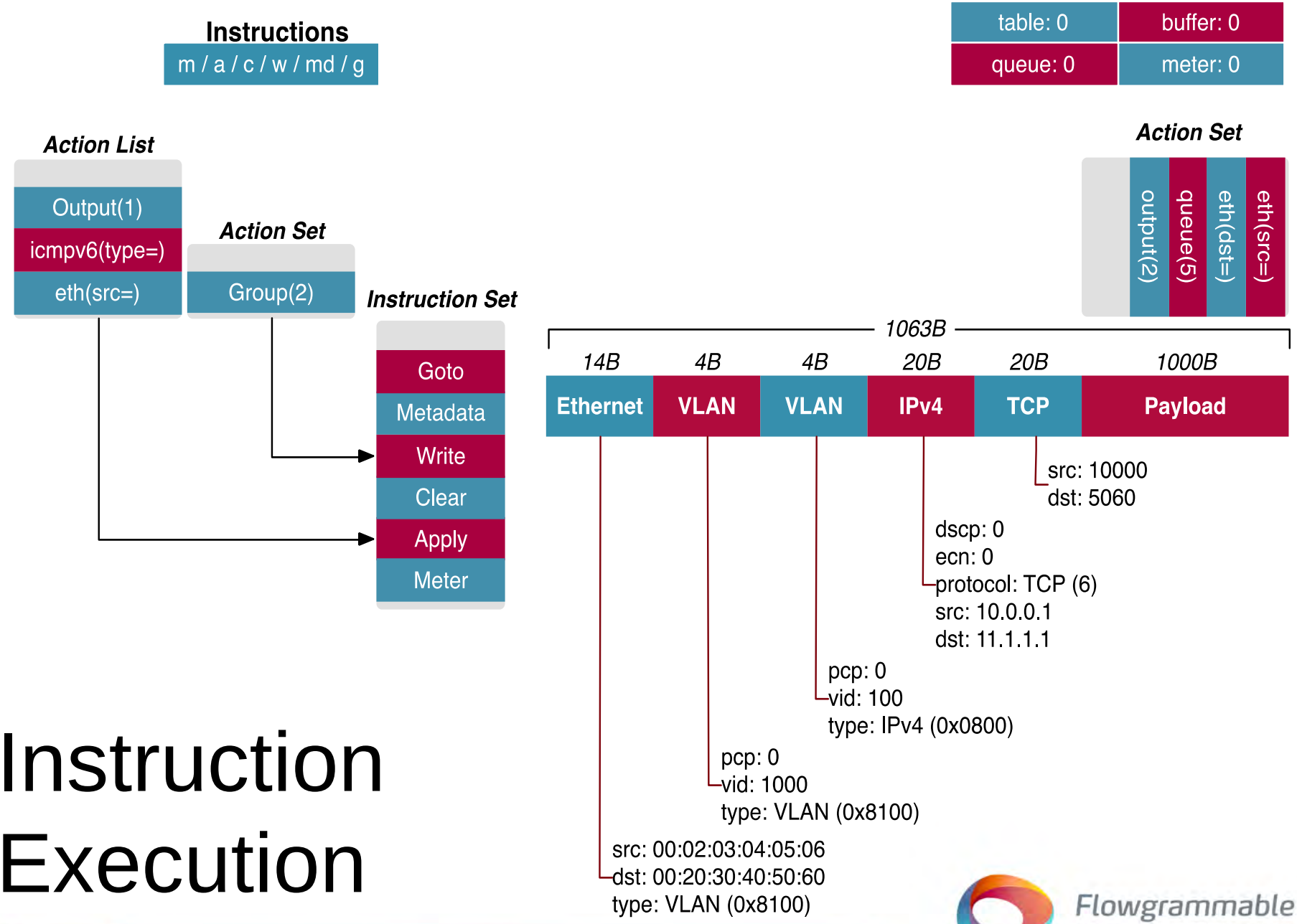


Flow Selection



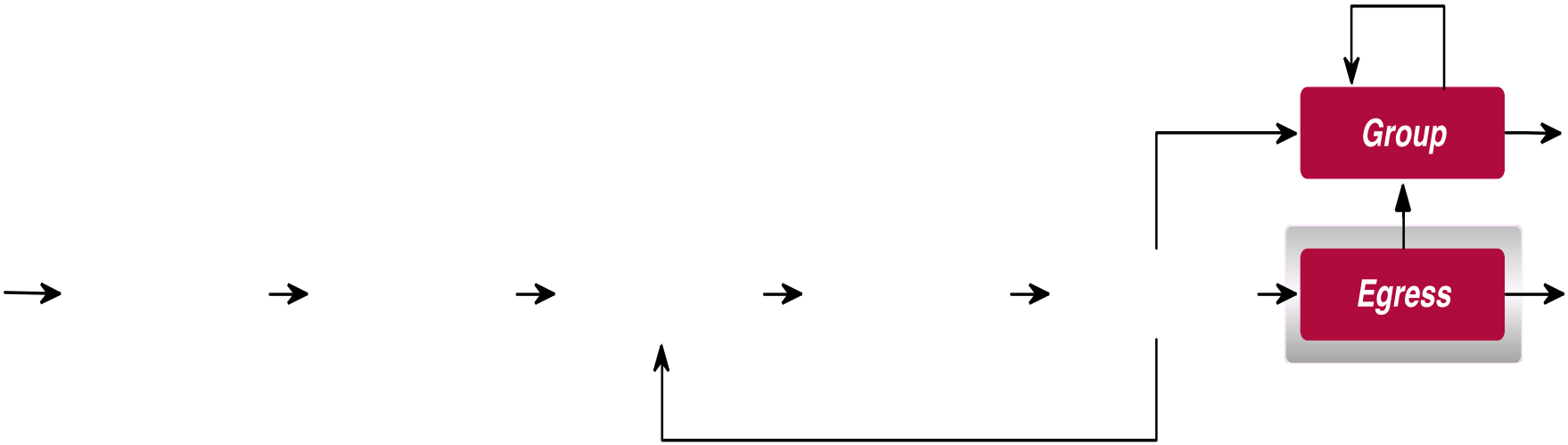
Dataplane Pipeline - Execution



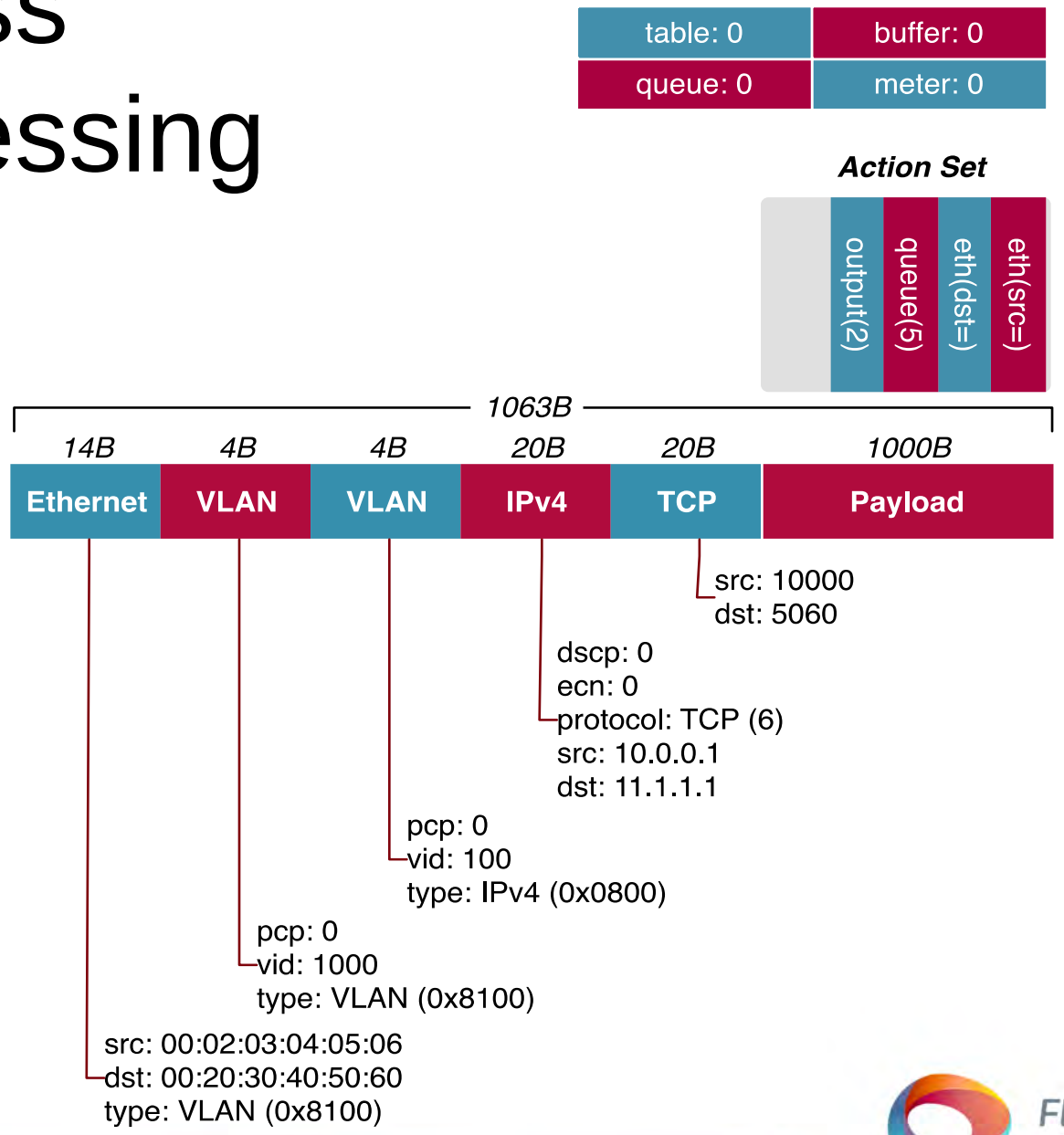


Instruction Execution

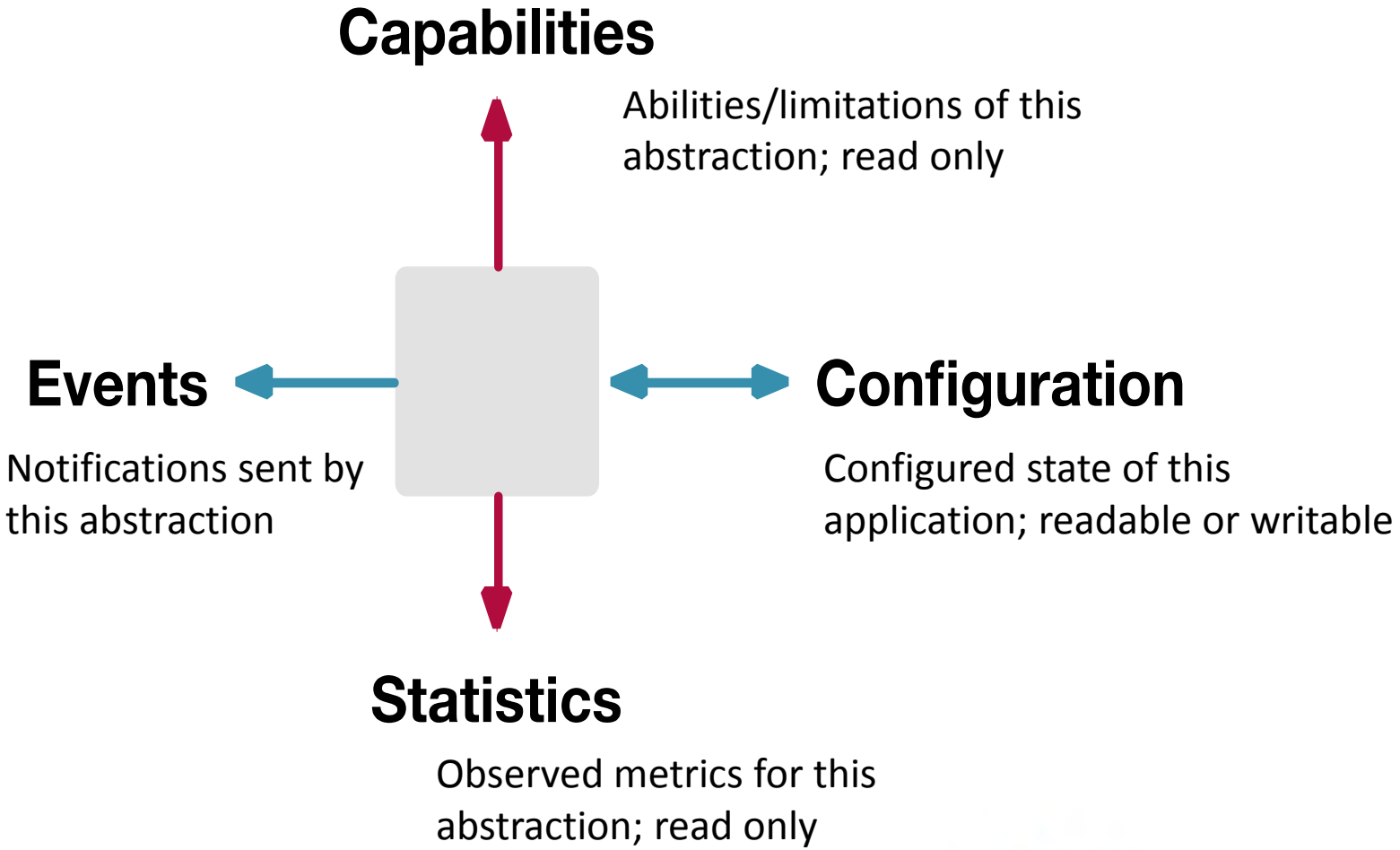
Dataplane Pipeline - Egress



Egress Processing

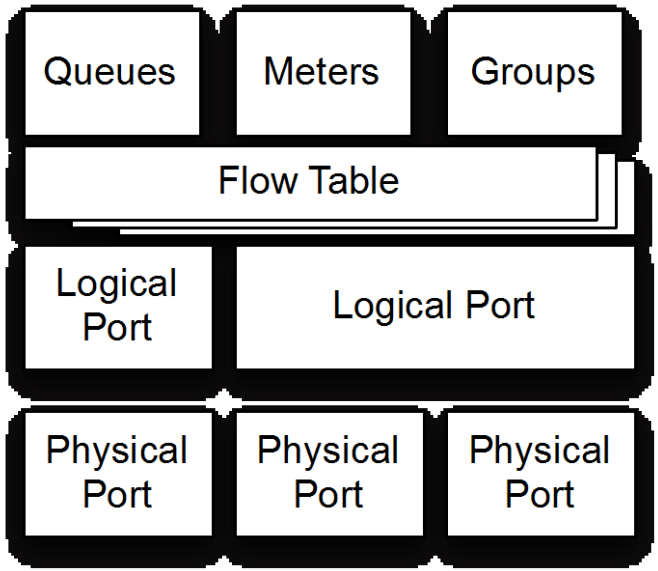


Definition of an Abstraction



What are data plan abstractions?

- Port – sources and sinks packets
- Flow Table – match packet to a flow and apply flow policy
- Meter – polices or shapes packet flows
- Group – provides egress processing

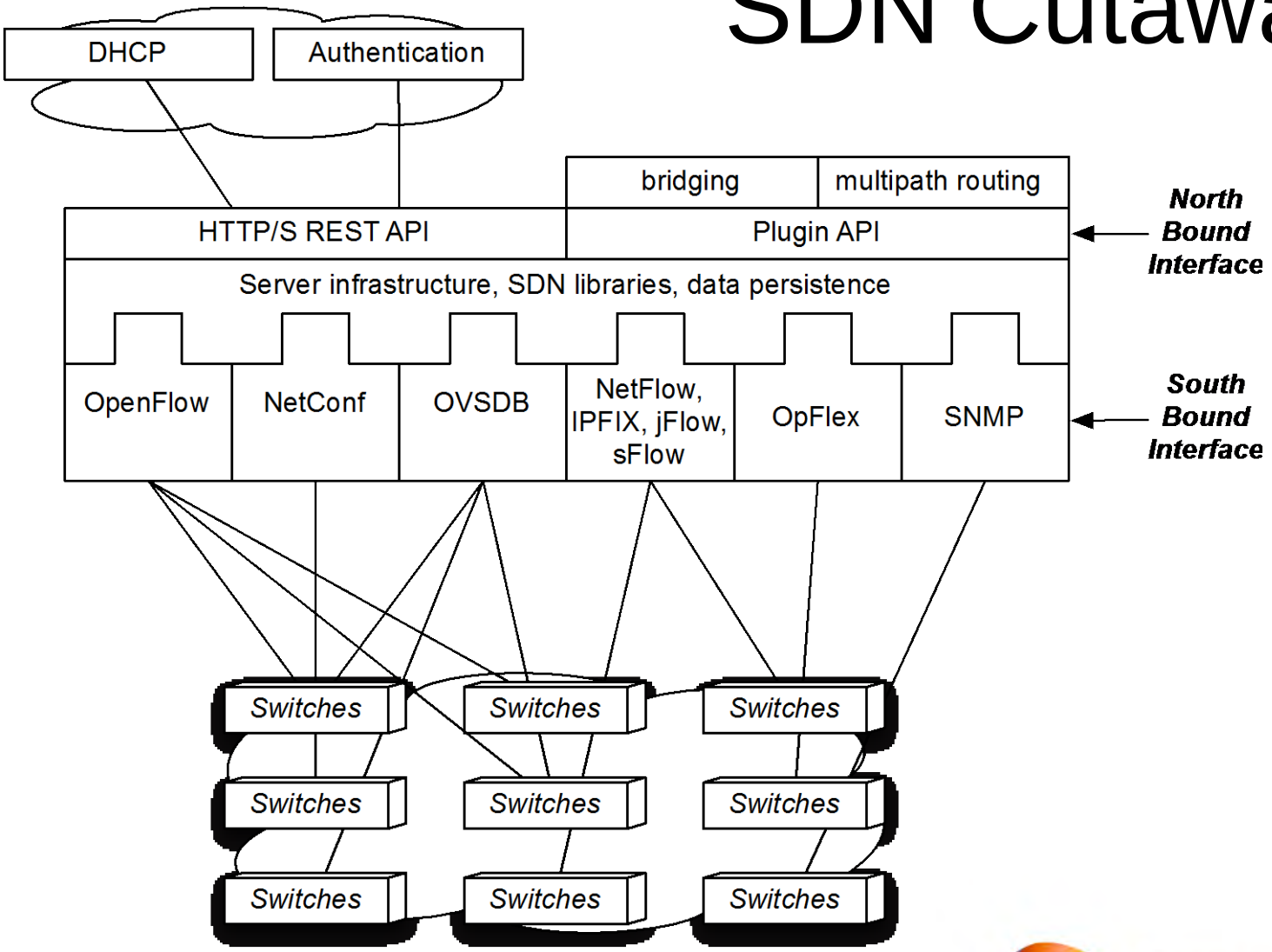


Emergence of SDN Applications

- Data plane abstractions
  - Device elements: Ports, Queues, Meters
  - Flow resources: Classifiers, Instructions, Actions
  - Interface elements: Tunnels, Certificates, Keys, Addresses
  - Abstractions exposed through South Bound Interface (SBI)
    - TLS/OpenFlow, OVSDB, OpFlex, Netconf/Yang, SNMP
    - NetFlow, IPFIX, jFlow, sFlow

- Data plane API
  - Controller manages remote data planes using SBI
  - Operations exposed through North Bound Interface (NBI)

SDN Cutaway



Flowgrammable
Driving the Next SDN Generation

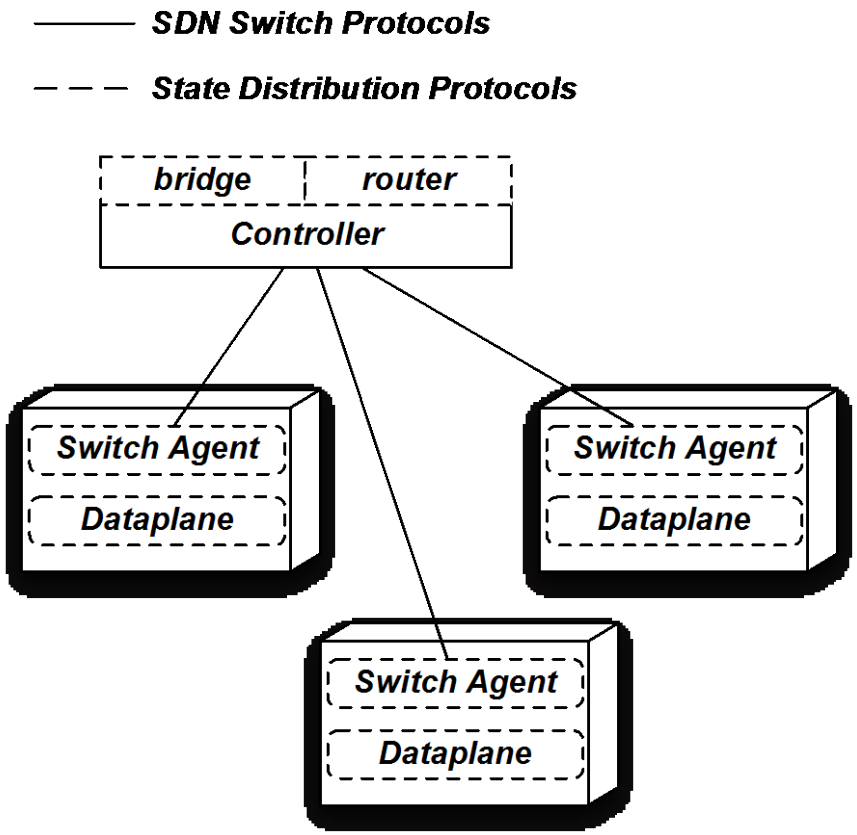
Centralized Controller

Applications

- Run as a single instance
- Hosted no a single logical controller
- No concern for distributed synchronization

Controller

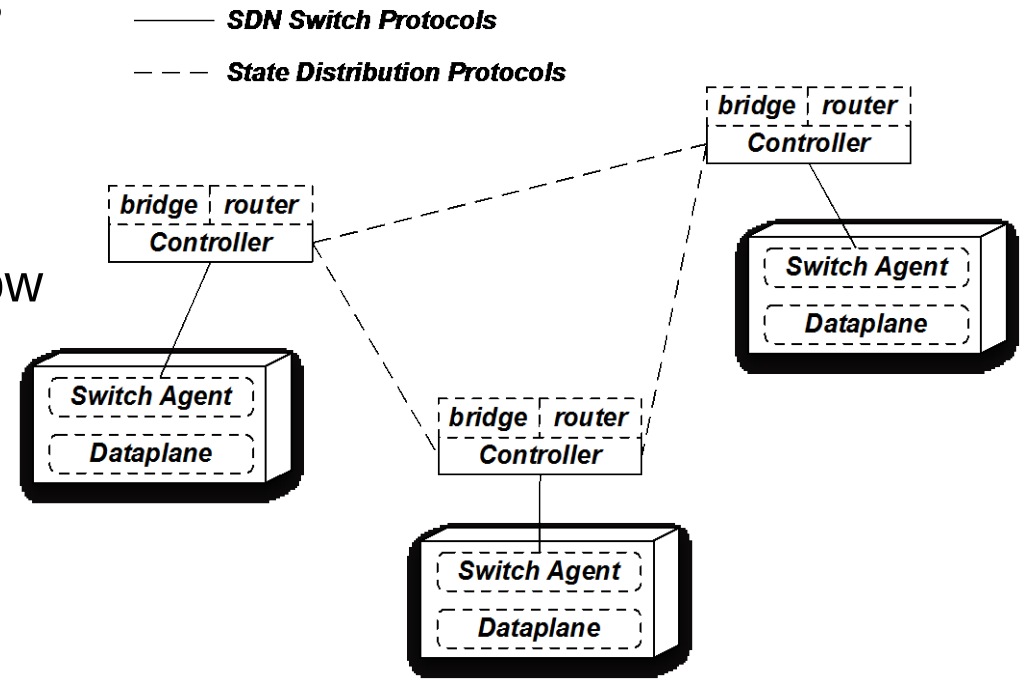
- Single logical instance
- Comprised of several physical controllers providing HA



Regionally Distributed Controller

Applications

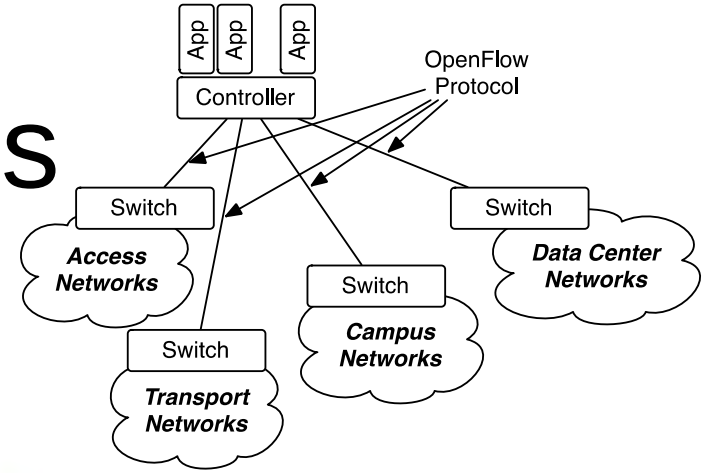
- Run as distributed instances
- Hosted multiple logical controller
- Distributed synchronization necessary for shared dataflow



Controller

- Multiple logical instances
- Several physical controllers provide distributed HA

Component Suppliers



freescale™

Ryu

Project Floodlight

hp

OPEN DAYLIGHT

IBM

NEC

Controller Vendors

Switch Vendors

CISCO

hp

PLURIBUS NETWORKS

BROCADE

NoviFlow
la commutation évoluée

ARISTA

DELL

IBM

vmware

OPEN VSWITCH
An Open Virtual Switch

centec networks

NEC

extreme networks

Flowgrammable
Driving the Next SDN Generation

SDN Customer Value

- Commodity switches
  - Greater supply of interchangeable network devices
  - Reduced equipment cost
  - Lower operating cost
- Standardized network service development
  - Rapid application development
  - Lower opportunity cost
  - Lower operating cost
  - Reduced risk developing complex network services
- Larger engineering labor pools
  - Lower network service development and operating cost
  - Reduced operational risk

SDN Problems

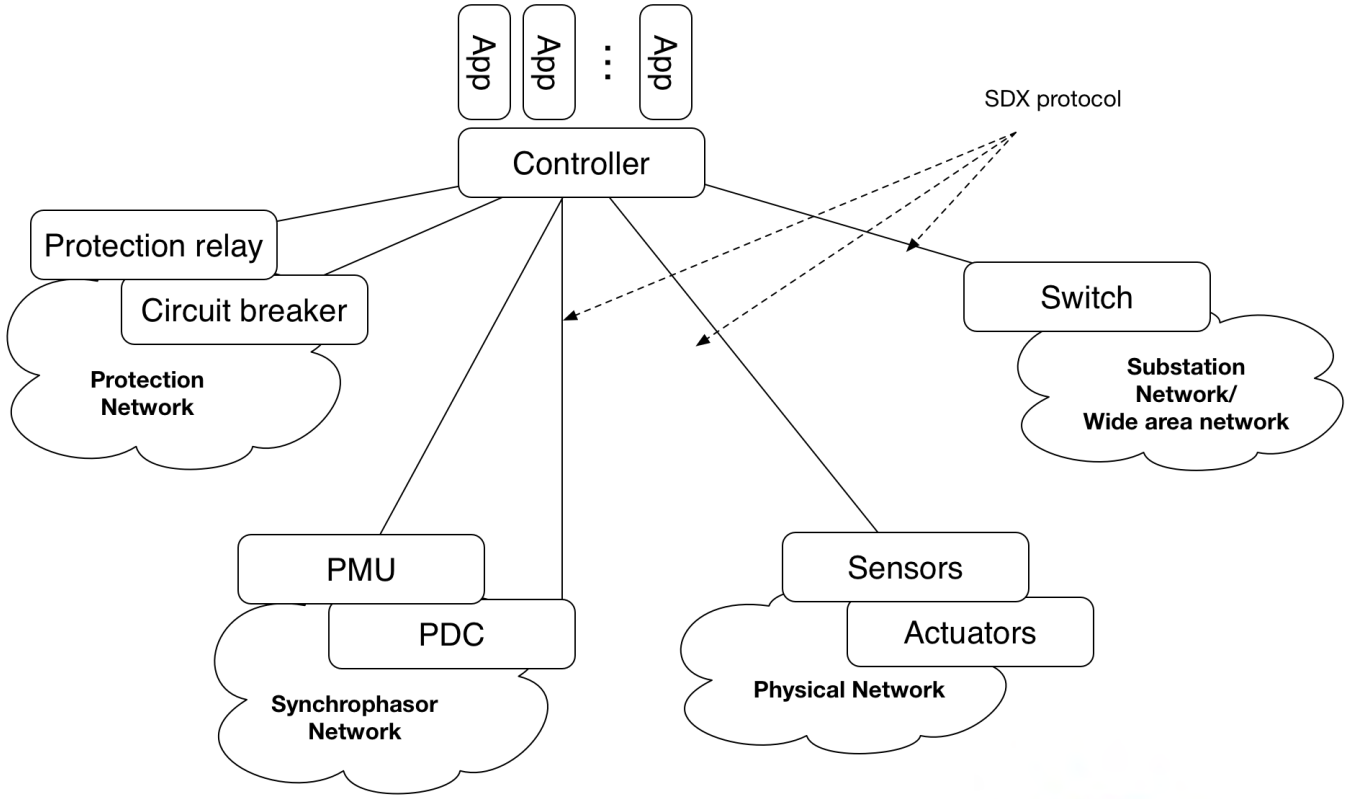
- New technology
  - Many competing standards
  - Low operational experience and maturity
  - Many non-interoperable systems are being built by vendors
- Complex technology
  - High performance packet processing data plane organization
  - High performance highly available controller software
  - Robust failure resistant application development
- Market noise
  - Many SDN startup companies (most focused on Data Centers)
  - Many advocating their own standards
  - Message is focused on ‘Program the Network’
  - Alienates non-programmers (Network Planning/Ops, CCIEs, etc)

The OpenFlow/SDN Problems

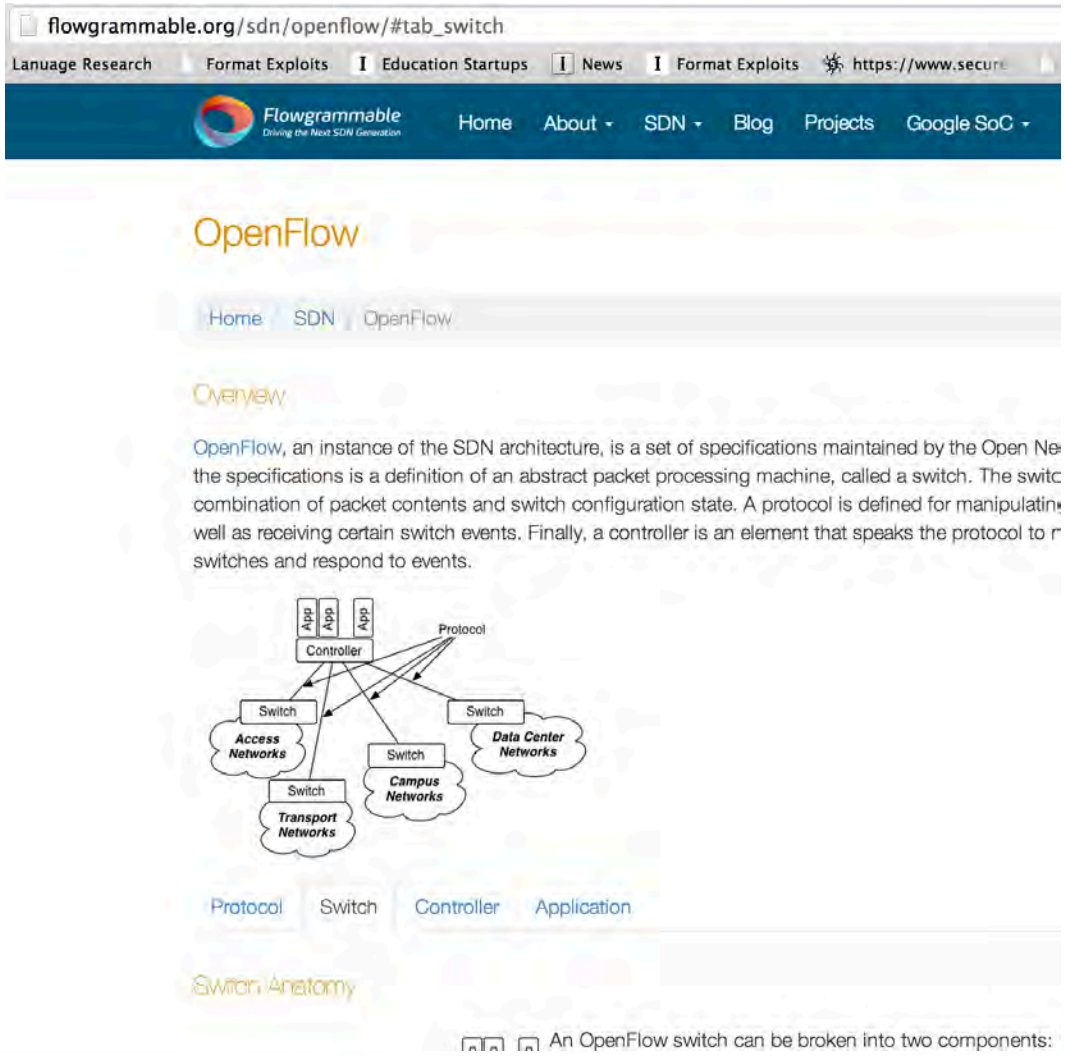
- Many versions of OpenFlow (more to come)
- Widely varying capabilities of switch vendors
- Controllers are difficult to program correctly
- No application portability across controllers
- Little operational experience with applications
- Existing networking staff does not program
- Limited development and test tools

Research Issues

- SDX – SDN for power systems application

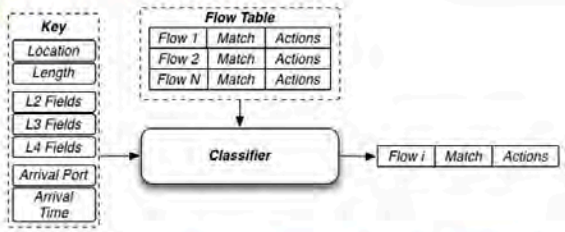


OpenFlow Knowledge Base



Dissects the Specifications

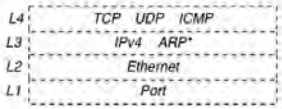
Home / SDN / OpenFlow / Classification



Classifiers are the fundamental component of an OpenFlow switch. Each flow, or row, in the flow table contains a classifier. The first flow whose classifier matches a packet becomes the active flow entry for that packet. The flow entry also contains an action set which will be applied to all packets matched. A classifier is a sequence of partial or full field match elements from various protocols. Each subsequent version of OpenFlow expands the number of protocols and fields supporting classification.

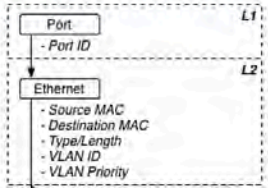
OpenFlow 1.0 OpenFlow 1.1 OpenFlow 1.2 OpenFlow 1.3 OpenFlow 1.4

Classifier Stack



This version of OpenFlow can classify against port ID and up to 6 protocols. It is important to note that support for classifying on ARP values is not mandatory. A controller will only know if a switch supports the optional classifier for ARP, based on a FeatureRes. The feature_capabilities will indicate ARP matching is supported if the switch is capable. Classifiers installed in a switch by the FlowMod message, which carriers the classifier in the Match Structure.

Classifier Dependencies



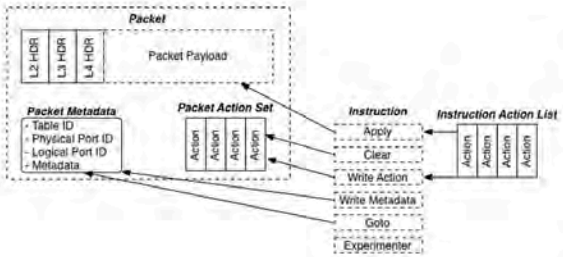
Classifier Fields

Protocol	Dependency	Name	Match Field	Mask Type
Port	none	Port ID	in_port	wildcard
Ethernet	none	Source MAC	dl_src	wildcard
		Destination MAC	dl_dst	wildcard

Flowgrammable
Driving the Next SDN Generation

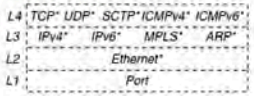
Explains the Protocol in Images

processing. The experimenter instructions provides a structure for custom extensions to instructions. Apply and write are the only instructions that apply actions in some way, as input they both contain action lists.



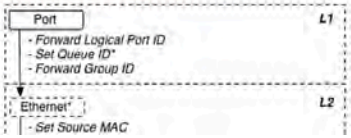
Instructions	Description
⚠ Apply	Apply action list immediately, bypass action set.
⚠ Clear	Clear the action set.
Write Action	Apply action list to action set.
⚠ Write Metadata	Update the metadata.
Goto	Continue processing at the indicated table.
Experimenter	Custom instruction extensions.

Actions Stack



This version of OpenFlow can perform actions against port IDs, queue IDs, and up to 4 protocols. It is important to note that support for actions against Ethernet, IPv4, TCP, and UDP, are optional. A controller will only know if a switch supports theses optional actions based on a table StatsRes. The action_capabilities will indicate which actions are supported by a switch.

Actions Dependencies



Actions Types

Protocol	Behavior	Target
Empty	Drop	Discards packet
Group	Forward	Groud ID: apply group processing

Provides Fine Grain References

Vendor/Experimenter Extension

Custom instructions can be carried in the Experimenter Instruction structure, cu
structure, or an Experimenter message can be used to carry custom extensions

References

Instructions: OpenFlow Switch Specification 1.3.2: pages 21, 55-57

Actions: OpenFlow Switch Specification 1.3.2: pages 21-25, 57-62

Flowgrammable
Driving the Next SDN Generation