UNIVERSITÄT BERN

Spring Semester 2016 Seminar

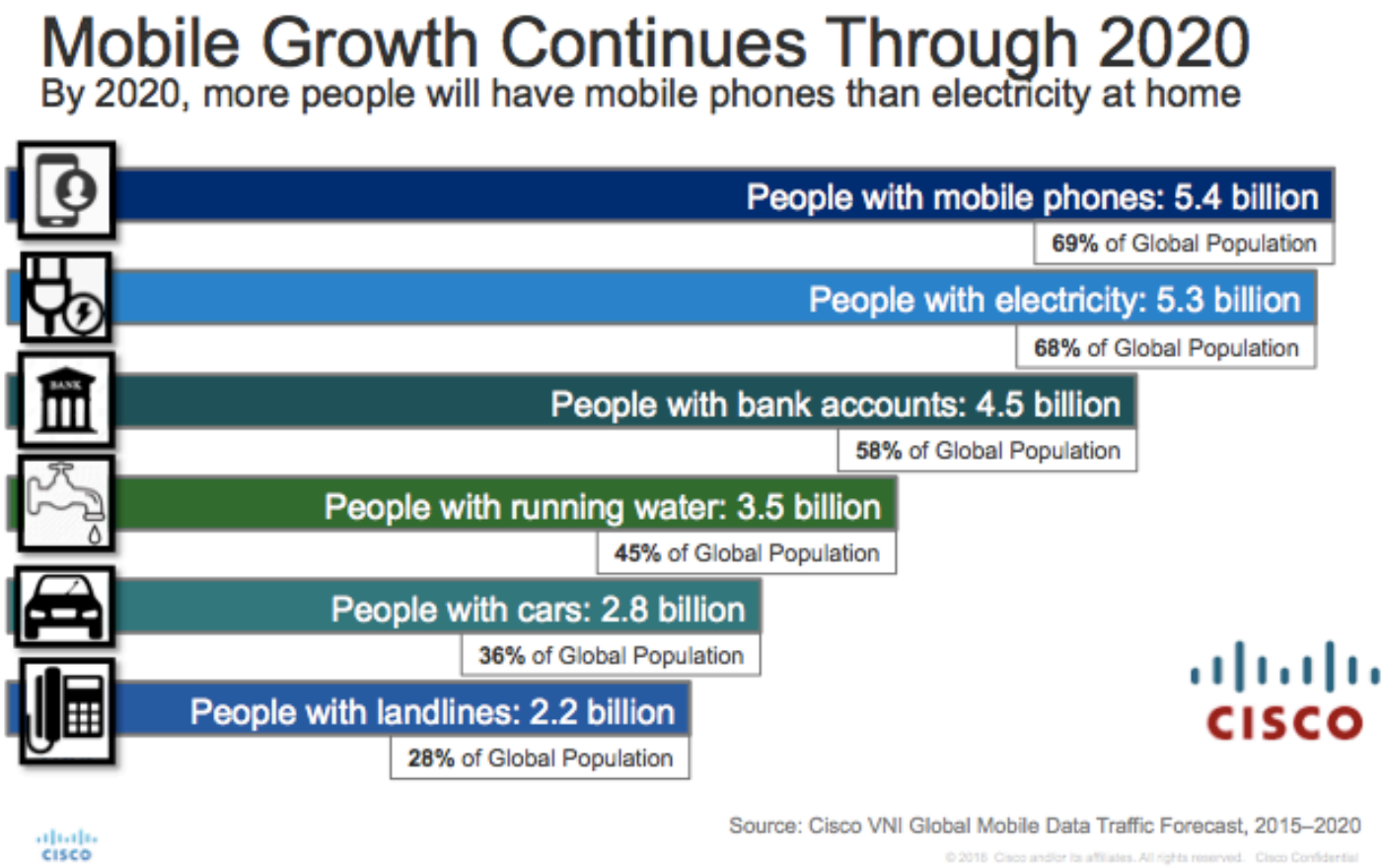
Software-Defined Wireless Networking

Zhongliang Zhao
Universität Bern
07.03.2016

Outline

- Motivation
- Software-Defined Networking & OpenFlow
- Software-Defined Wireless Networking & OpenFlow Extension
- SwissSenseSynergy Project Use Cases
- Conclusions

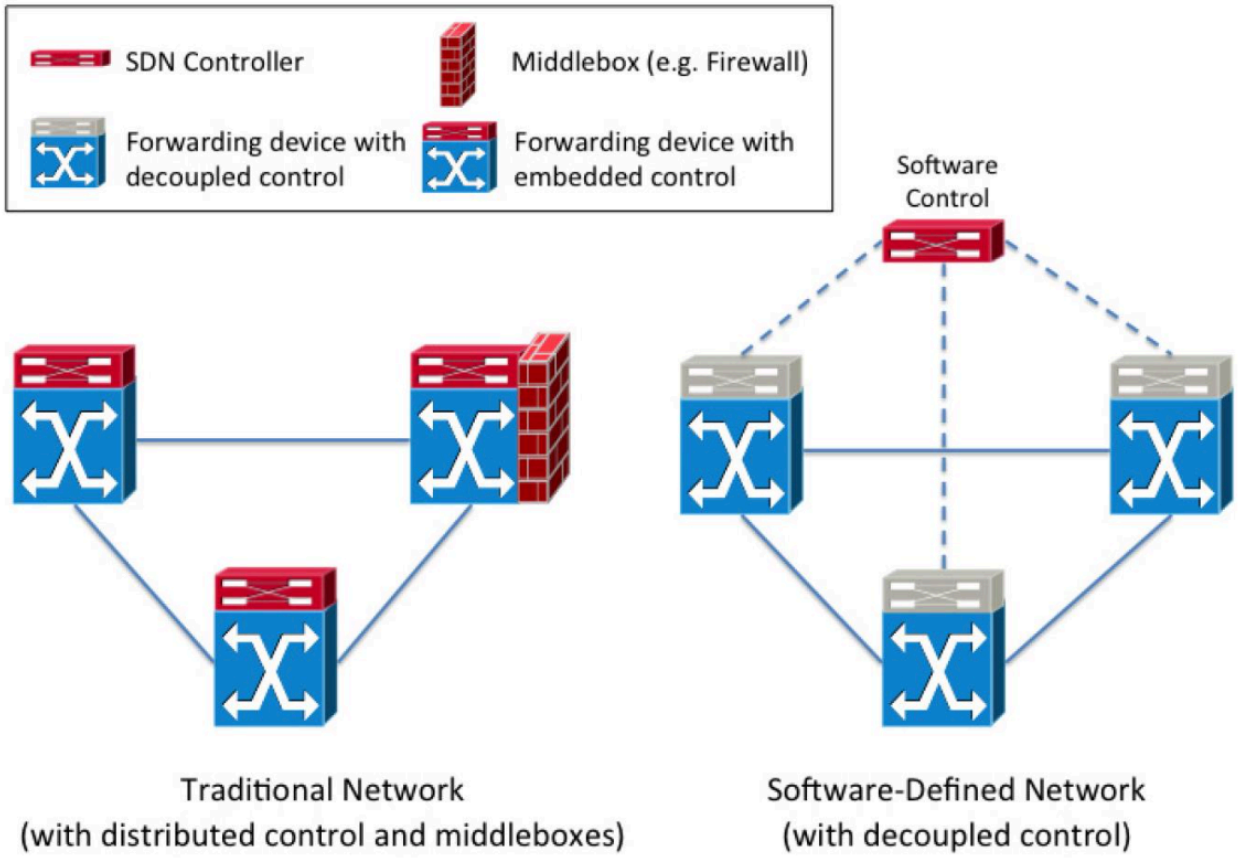
Wireless Mobile Data Growth



Software-Defined Networking (SDN)

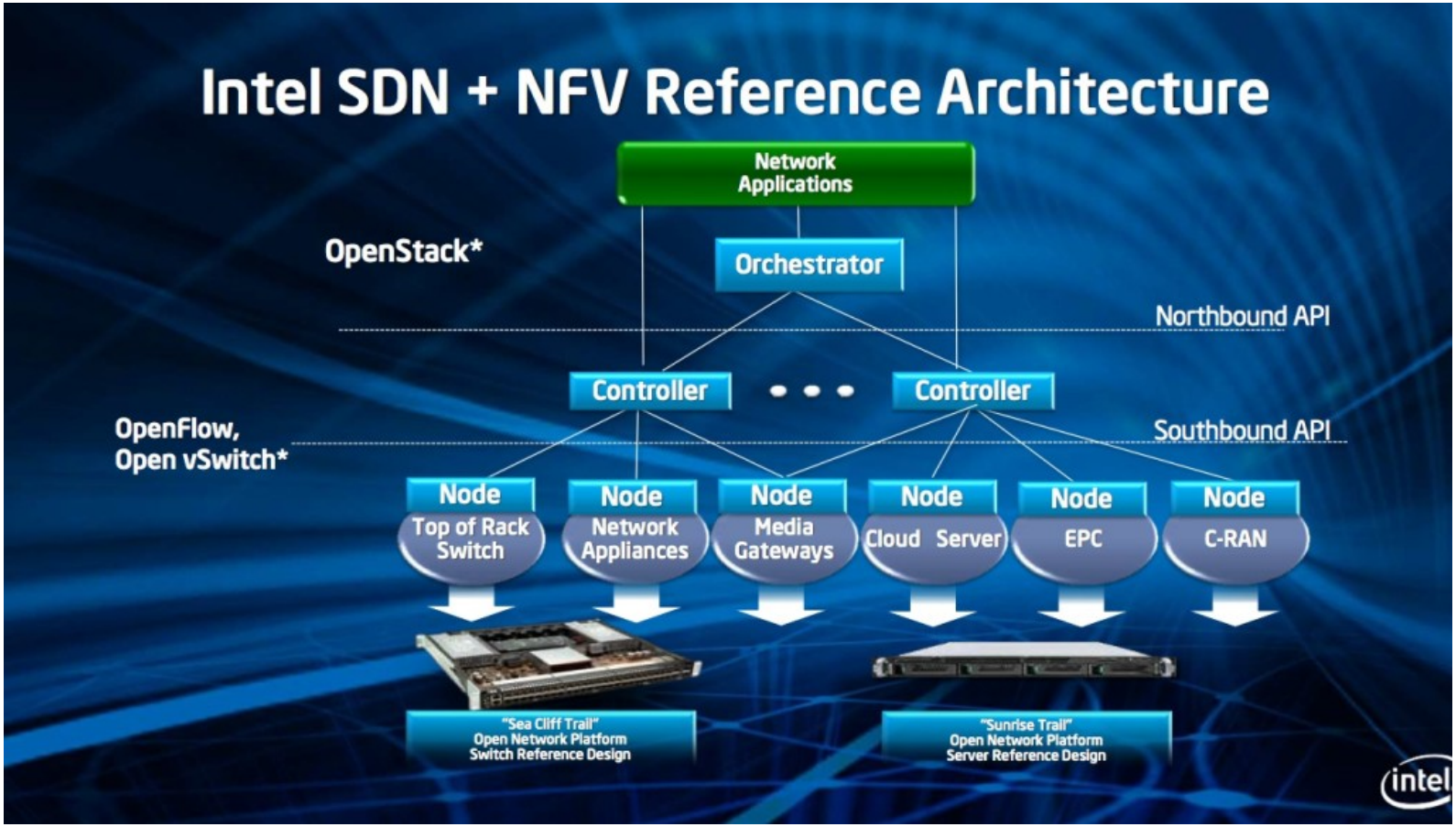
- Decouple forwarding hardware / control decisions
- Network devices are functionally broken up into
  - Software-based controllers (control plane)
  - Packet forwarding devices (data plane)
- Attracting attention from academia and industry
  - Open Networking Foundation (ONF)
  - Network function virtualized on commodity hardware
  - Reducing CapEx (e.g., S/P-GW millions of $ per box)
- Standardization efforts on SDN

SDN: Decoupled Control



"A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks (Fig 1)"

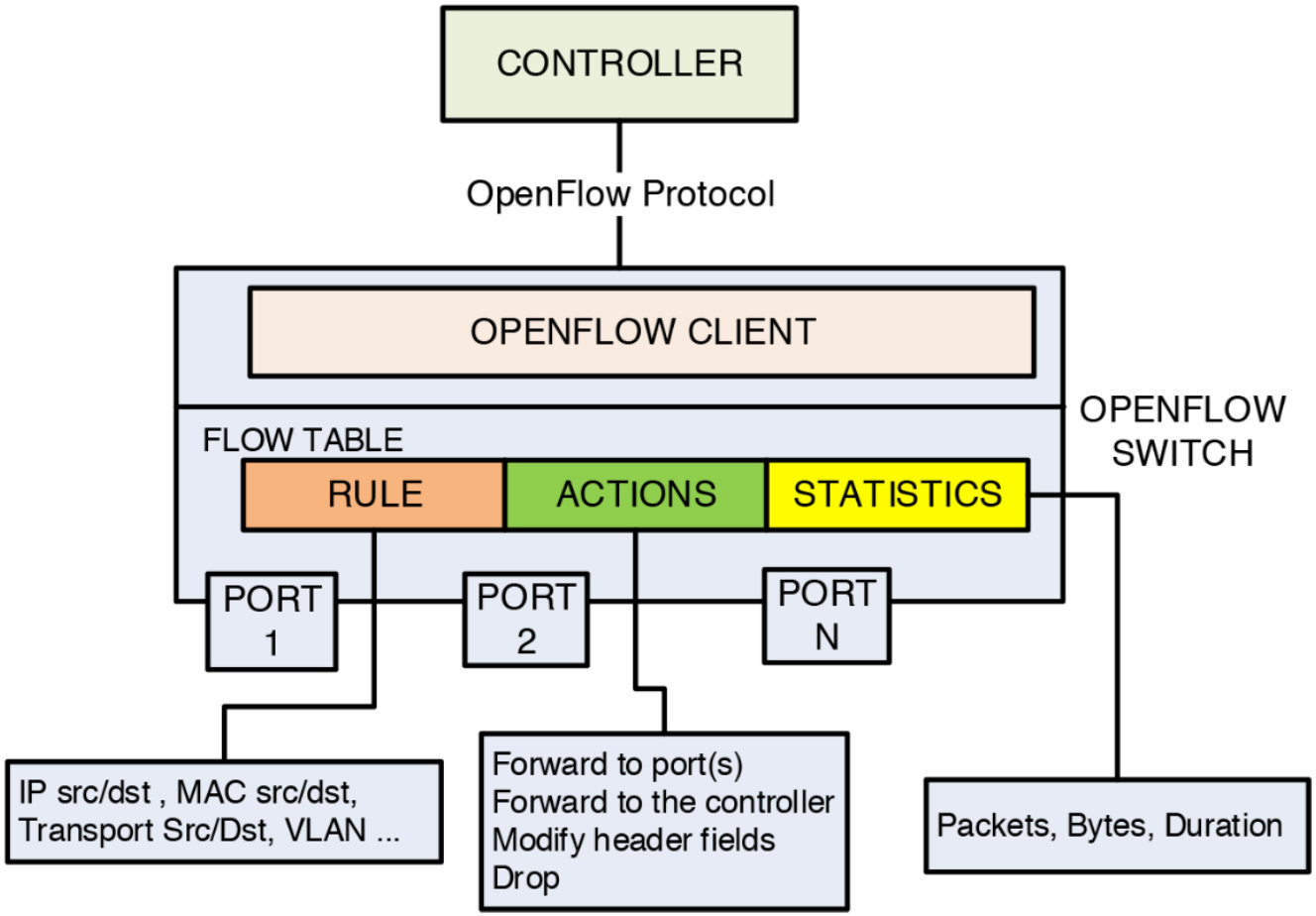
SDN Reference Architecture



SDN Application Environments

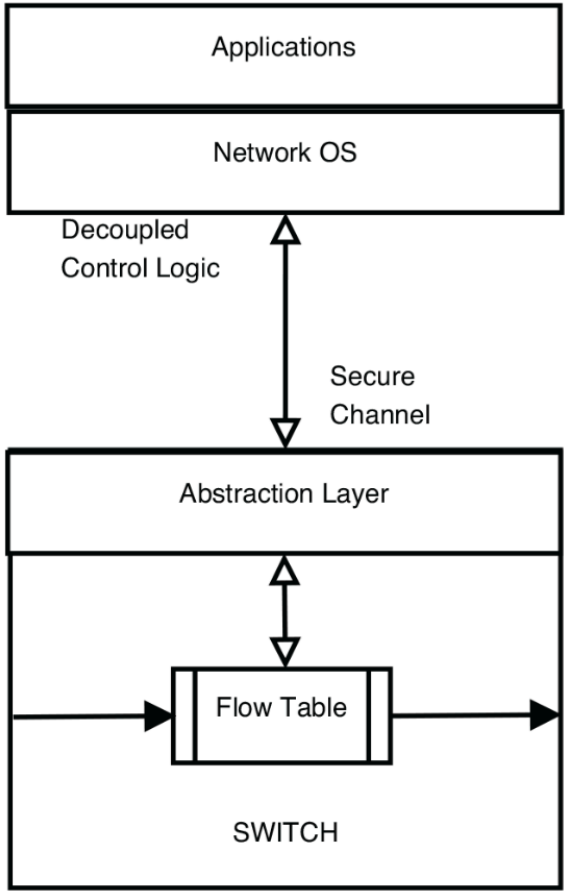
Scenarios	Use cases
Datacenter	Virtualization, multi-tenancy, failure recovery, traffic engineering, load-balancing
Backbone	Resiliency, reliability, determinism, traffic engineering and load-balancing
Campus network	Network access control, guest access, monitoring malicious behavior
Security	Firewalls, intrusion detection and prevention, blacklists, enforced quarantine
Wireless	Mobile wireless backhaul, heterogeneous wireless access, service chaining, load balanced packet core.

OpenFlow



"A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks (Fig 2)"

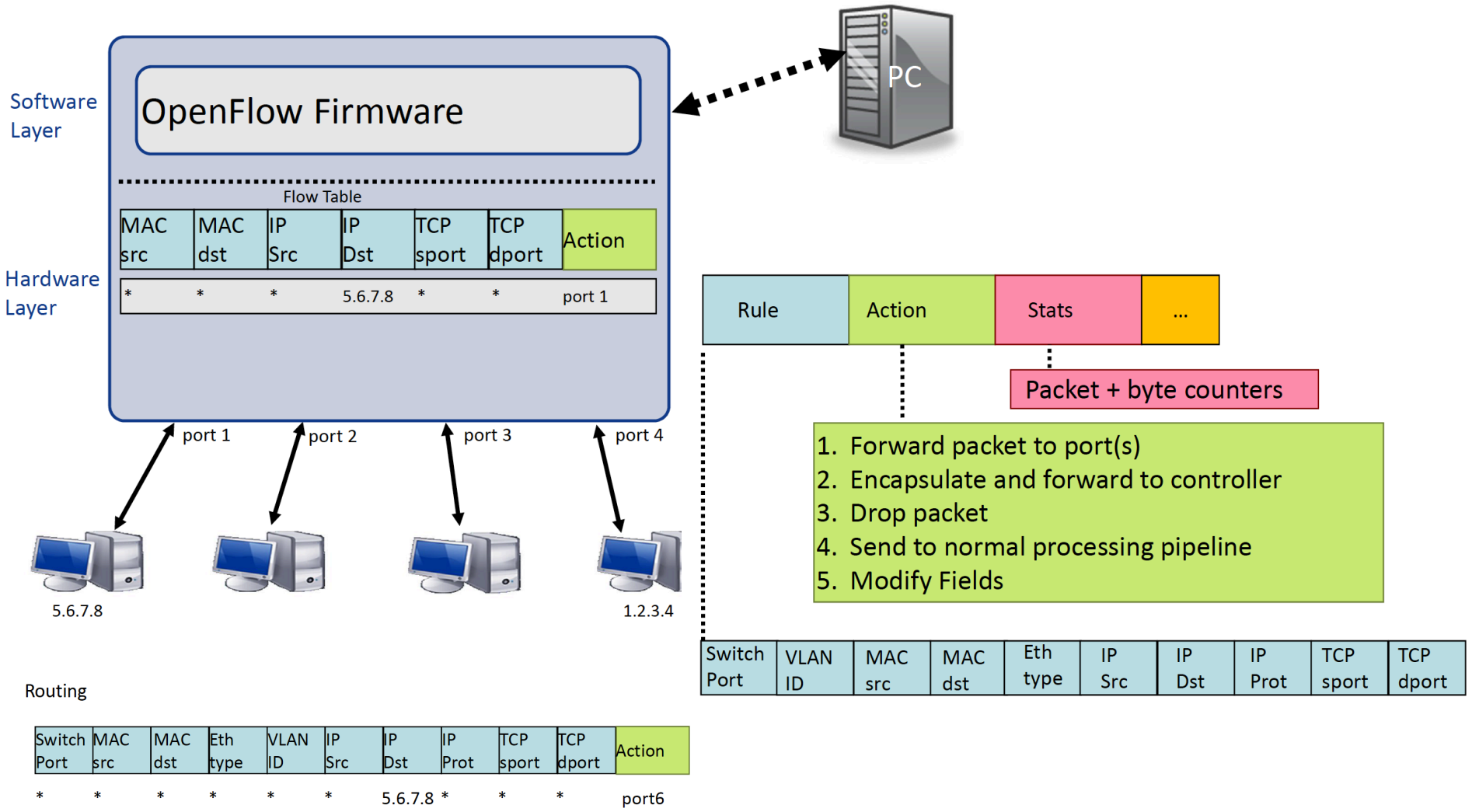
OpenFlow Controller



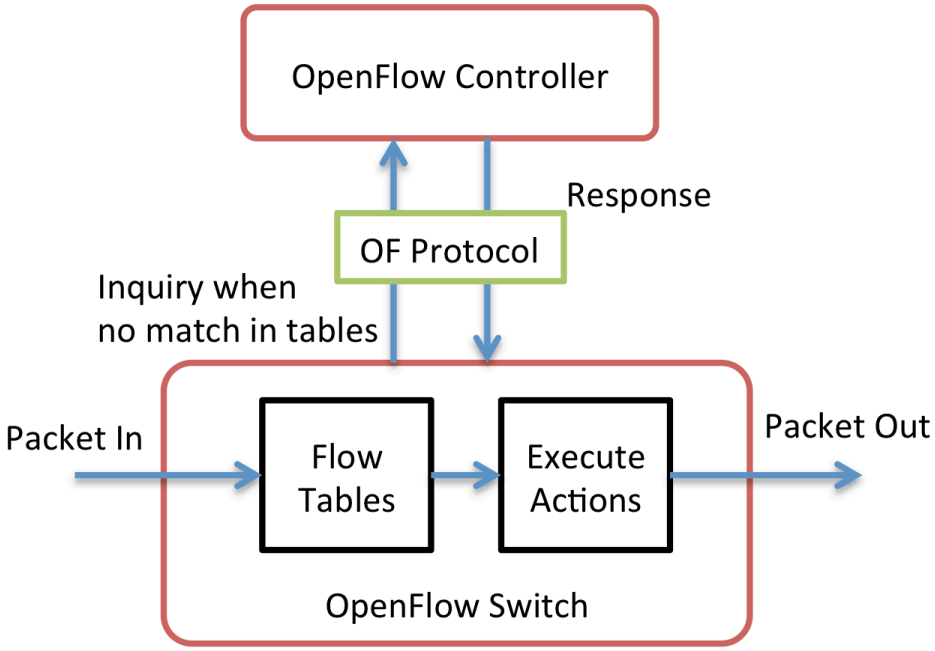
- A network "operating system"
- Programmatic interface to the network
- Southbound communication standardized through the OpenFlow protocol
- Northbound communication: no standard defined between controllers and applications (software implementation)

"A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks (Fig 3)"

Flow Table


Software Layer
OpenFlow Firmware
PC
Flow Table
MAC src
MAC dst
IP Src
IP Dst
TCP sport
TCP dport
Action
Hardware Layer
* * * 5.6.7.8 * * port 1
port 1
port 2
port 3
port 4
5.6.7.8
1.2.3.4
Rule
Action
Stats
...
Packet + byte counters
1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline
5. Modify Fields
Switch Port
VLAN ID
MAC src
MAC dst
Eth type
IP Src
IP Dst
IP Prot
TCP sport
TCP dport
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

OpenFlow Switch Packet Handling: <Match, Action>



- Match
  - Match on any header
- Action
  - Forward to ports, drop packet, send to controller
  - Overwrite header with mask, push or pop into waiting queue
  - Forward at specific bit-rate
  - Allows multiple actions

Flow Table Entries

- Exact rules
  - All fields are specified
  - Higher priority than wildcard rules

Ingress port	Eth dst	Eth src	Eth type	…	Statistic	action
5	00:12…	00:07…	0x0800	…	counters	Act 0

- Wildcard rules
  - At least one field contains a wildcard or a prefix
  - Multiple rules can match a packet → priorities

Ingress port	Eth dst	Eth src	Eth type	…	Statistic	action
*	00:23…	*	*	…	counters	Act 0

OpenFlow Key Message and Types

Message	Direction	Description
Packet-in	Switch → Controller	Transfer the control of a packet to the Controller that ***does not match***. Packet-in events can be configured to buffer packets
Packet-out	Controller → Switch	Instruct switch to send a packet out of a specified port. Sent in response to Packet-in messages to ***manage flow entries***
Modify-state	Controller → Switch	Add, delete and modify flow/group entries in the flow tables (aka Flow-mod); set switch port properties
Flow-removed	Switch → Controller	Inform the controller about the removal of a flow entry from a flow table

Open Networking Foundation (ONF) - Wireless Mobile WG (WMWG)

UNIVERSITÄT BERN

- Goal: identify OpenFlow enhancement to improve operations of wireless networks
- WMWG use-cases and use-case project teams

1. Flexible scalable packet core
2. Dynamic resource management for wireless backhaul
3. Mobile Traffic Management
4. Connection-Oriented SDN for Wireless SCB
5. Management of secured flows in LTE
6. Media-Independent Handover
7. SDN Enhanced Distributed P/S-GW
8. Network-Aware UE Multiple Radio Interface Management
9. S-GW virtualization
10. Service Chaining in Mobile Service Domain
11. Energy Efficiency in Mobile Backhaul Network
12. Security and Backhaul Optimization
13. Unified Equipment Management and Control
14. Network Based Mobility Management
15. SDN-Based Mobility Management in LTE
16. IEEE OmniRAN
17. Unified Access Network for Enterprise and Large Campus

Use Case Project Teams

1. **Mobile Packet Core**
2. **Wireless Transport**
3. **Unified Access**

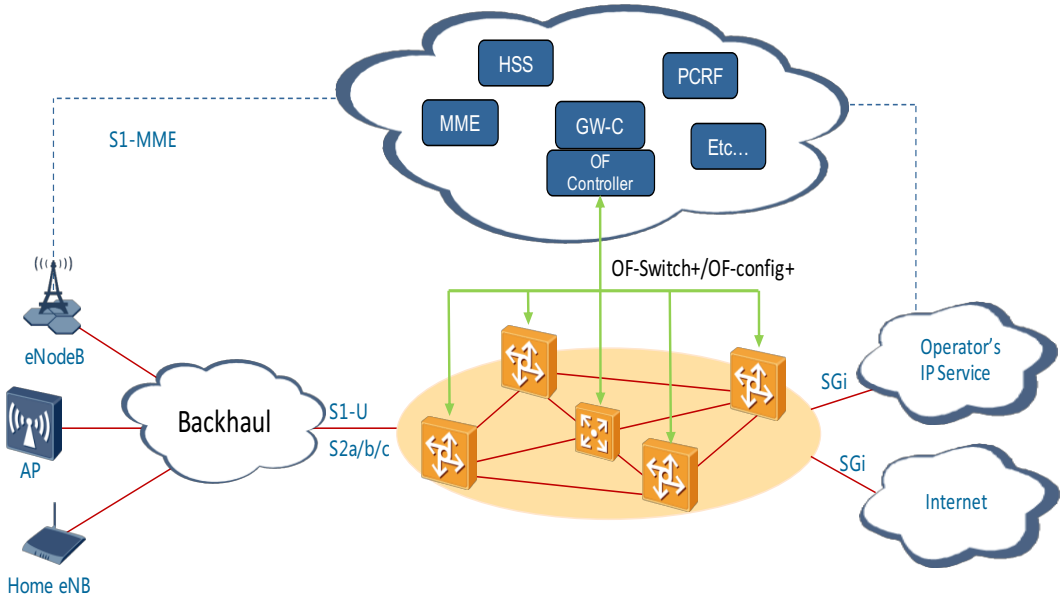
WMWG Projects

UNIVERSITÄT BERN

Mobile Packet Core	Wireless Backhaul	Unified Access
—Apply OpenFlow to 3GPP Evolved Packet Core (EPC) —Many uses such as user/data plane separation in GW, mobility management and mobile flow steering for offload.	—Backhaul links are wireless —Central SDN controller optimizes radio parameters in data plane using enhanced OpenFlow	—Develop a unified access network that uses a common controller to manage both wireless access points (AP) and wired switches

Mobile Packet Core Project

UNIVERSITÄT BERN



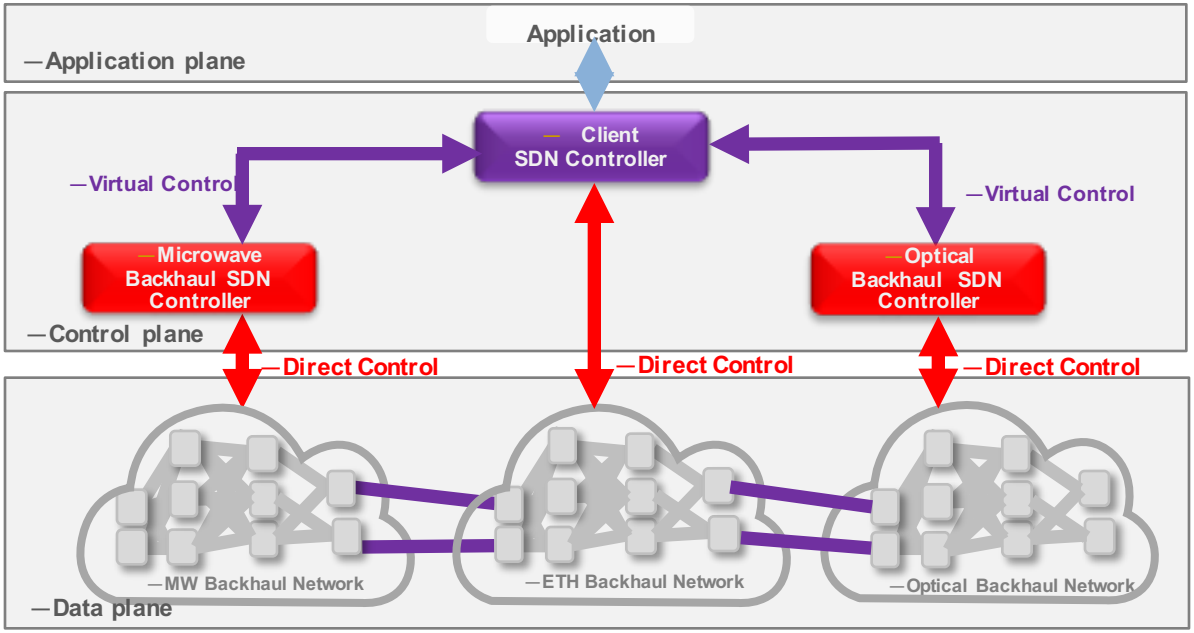
Address Three Use Cases

1. **SDN based Evolved Packet Core**
2. **SDN based Mobility Management**
3. **Service Chaining in Mobile Service Domain**

- EPC control plane and SDN controller separated from data plane implemented by enhanced OpenFlow switches
- Place and move the routing of GTP and non-GTP tunnel flows through EPC data plane using OpenFlow while supporting the needs of the wireless network
- ***OpenFlow extensions needed*** to support:
  - GTP/non-GTP tunneling, Policy Control, and Lawful Interception

Wireless Backhaul Project

UNIVERSITÄT BERN

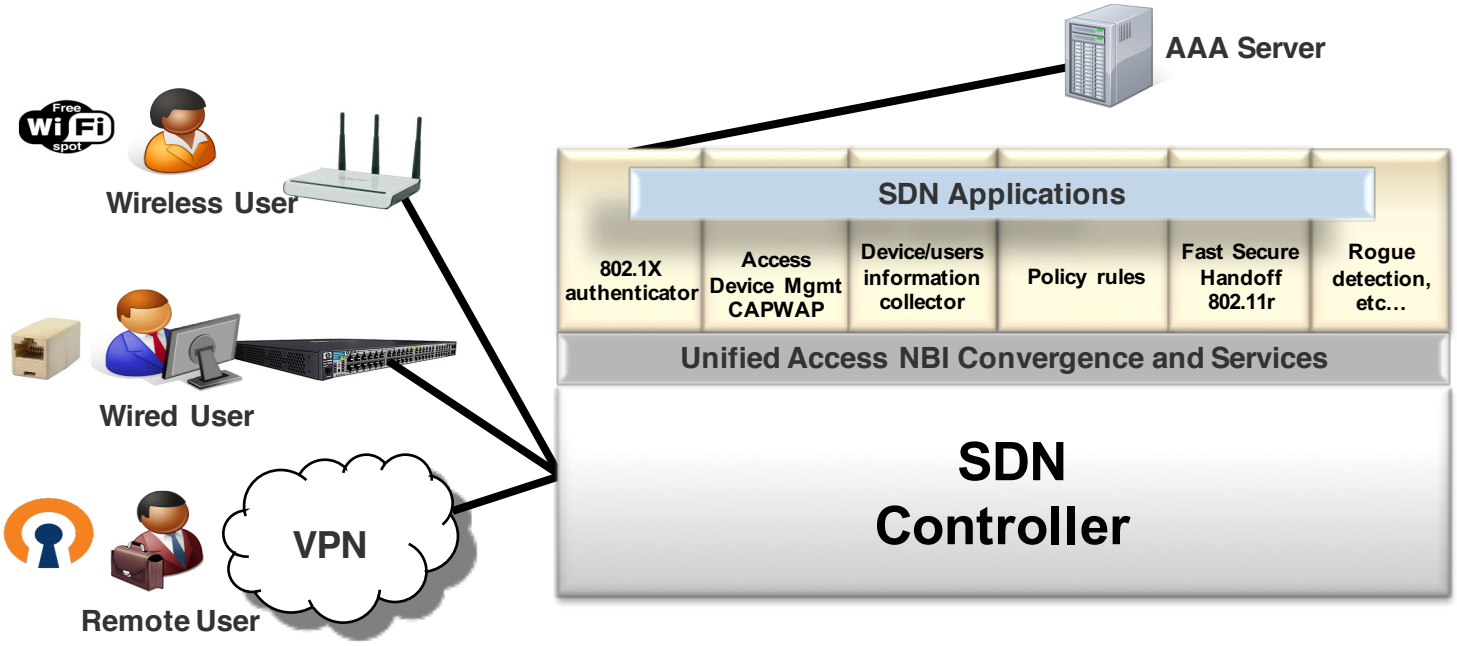


Combining 4 Use Cases

1. Backhaul resource management
2. Energy Efficiency
3. Unified Equipment Management
4. Common Public Radio Interface (CPRI) and Ethernet support

- Central SDN controller calculates the path and assigns the backhaul resources considering:
  - SLA parameters (e.g., guaranteed vs. non-guaranteed)
  - Link availability, capacity, e.g., adjusting modulation.
  - Collection of traffic statistics to estimate the actual throughput
- Define new OpenFlow port types for wireless backhaul links (e.g. microwave)

Unified Access Project



- Controller leverages enhanced OpenFlow to manage both wired and wireless AP
- An unified architecture and consistent means of managing
  - User role/location based policy
  - Real-time monitoring
- Need to support strong authentication of endpoints, fast roaming

Software-Defined Wireless Networking

- Wireless networks are more complex then Ethernet
  - Increasing amount of UE
  - Mobility
  - Unicast, multicast, broadcast
  - Various QoS policy requirements
  - Wireless radio interfaces have huge variety of properties
    - Radio properties (channel, tx power, RSSI levels)
    - TX characteristics (antenna features)
    - MAC layer issues (group-cast, broadcast)

- SDN-based wireless networking requirements:
  - Provide control interface to query wireless network about availability, quality, speed, user location …
  - Control granularly how individual user or application traffic is handed by the network

OpenFlow Extension for Wireless

- OF was originally designed for wired networks
- It assumes that underlying network is composed of high-speed Ethernet switch/IP routers
- Currently no support for the matching of 802.11 specific fields
- Latest OpenFlow specification (v1.5) defines three types of port properties
  - Ethernet
  - Optical
  - Experimental
- OF wireless extension is essential
  - Support for wireless port properties
  - Specific counters for wireless devices

```
/* Port description property types.
 */
enum ofp_port_desc_prop_type {
    OFPPDPT_ETHERNET          = 0,
    OFPPDPT_OPTICAL           = 1,
    OFPPDPT_PIPELINE_INPUT    = 2,
    OFPPDPT_PIPELINE_OUTPUT   = 3,
    OFPPDPT_RECIRCULATE       = 4,
    OFPPDPT_EXPERIMENTER      = 0xFFFF,
};
```

Match/Action: Wired OpenFlow

- "*If* a packet comes from port X, *then* apply VLAN Y, and forward through port Z"
- ***Match fields*** defined for fields of Ethernet frame, IPv4/IPv6 packet, TCP segment, MPLS…
- Not consider IEEE 802.11 or eNB specific frame format
- ***Action fields*** defined
  - Forward packet to physical port
  - Enqueue packet to queue in the port
  - Drop packet
  - Modify field
- ***Statistic fields*** defined per table, flow, port, queue

Match/Action: Wireless OpenFlow

- "*If* a packet is for user *M,* with QoS requirement *N*, *then* apply encoding mechanism *X*, and transmit through port Y in rate Z and power-level *B*"
- ***Match fields*** identify flows (wireless_flow_id) of individual users (ue_id) and/or application requirements
- ***Action fields*** control how packets should be routed
  - in which rate and power level
  - using which encoding mechanism
  - via which access point
  - how they are scheduled at access points

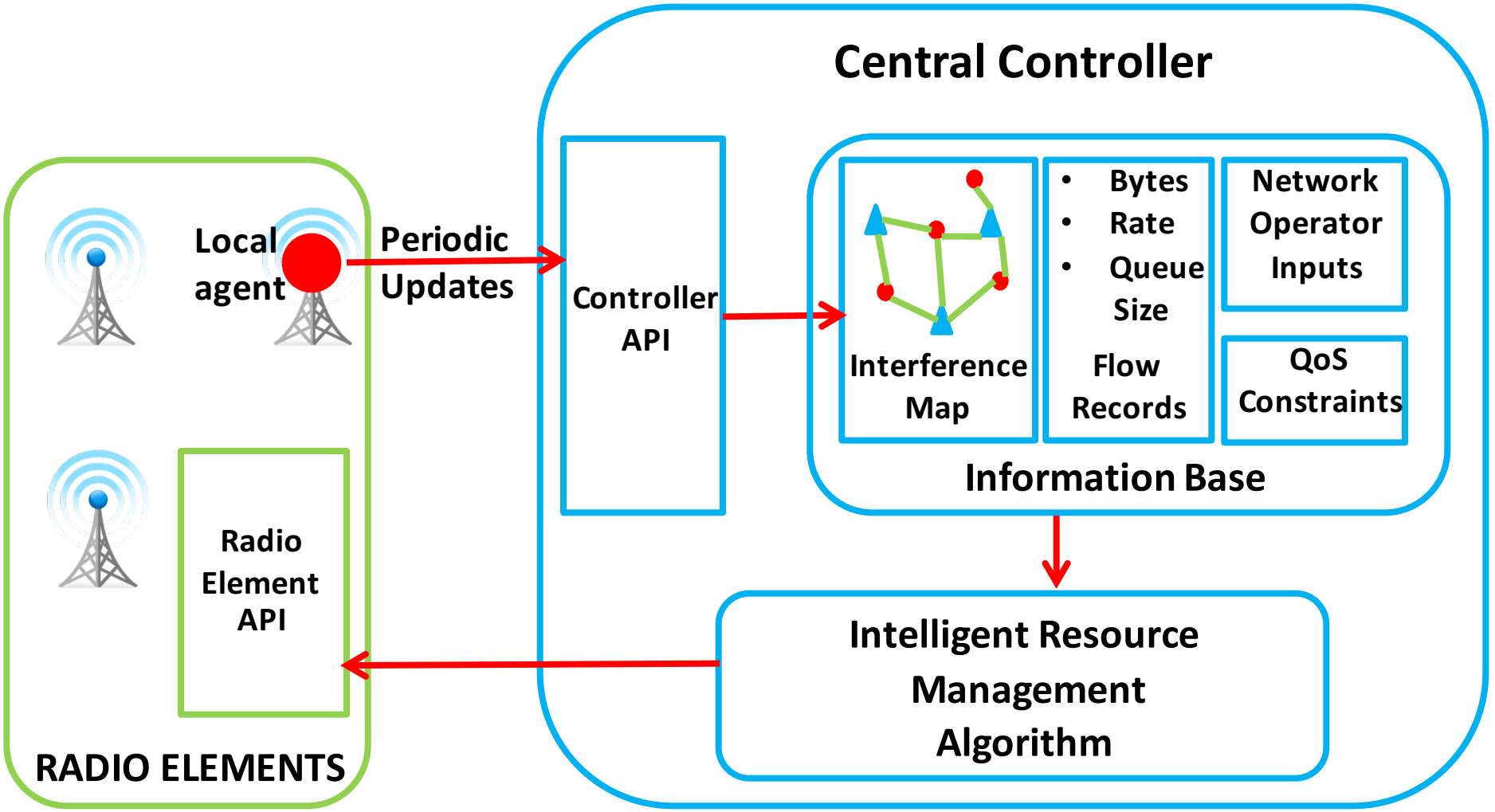
- How to define/label a wireless flow?
  - Radio technique dependent

Hierarchical Controller

- **Local agent (LA)** at each base station or access point
  - Measure local network traffic, subscriber usage statistics, assess QoS policies, etc.
  - Perform local management tasks under the supervision of the central controller
  - Notify central controller if the traffic exceeds a certain threshold, tag some packets to be redirected to a transcoder, etc.

- **Central controller (CC)**
  - Delegate some local controller function to LA
  - Management overall network status
  - Intelligent resource management

Hierarchical Controller (cont.)



Hierarchical Controller: Message Updates

- Local agent → Controller
  - Flow information (downlink/uplink)
  - Channel states (reported by clients to agent)
  - Link statistics (monitored by agent)
- Network operator → Controller
  - QoS requirements
  - Flow preferences

- Reactive: delay
- Proactive: periodic update, which frequency?

Controller and Switch Support

- OpenFlow Controller
  - NOX (C++/Python)
  - Floodlight (Java)
  - OpenDaylight (Java)
- OpenFlow Switch
  - Commercial Switch
  - Software Switch
    - Open vSwitch
    - OpenWRT
    - Mininet
    - CPqD

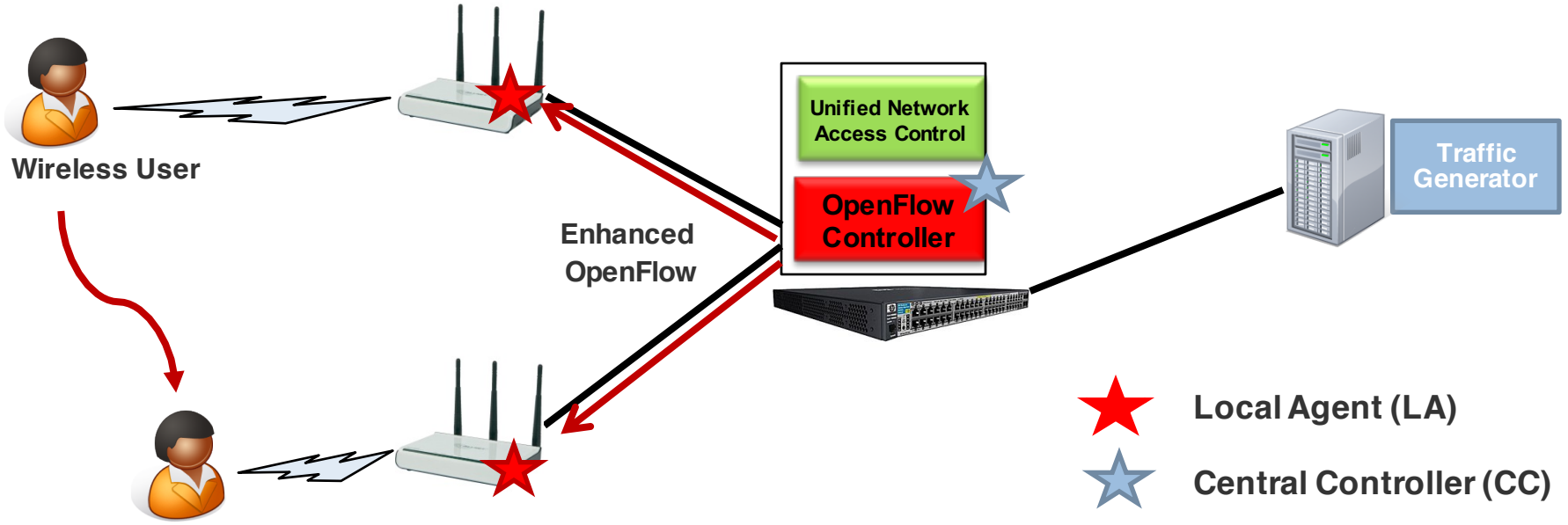
	POX	Ryu	Trema	FloodLight	OpenDaylight
Interfaces	SB (OpenFlow)	SB (OpenFlow) +SB Management (OVSDB JSON)	SB (OpenFlow)	SB (OpenFlow) NB (Java & REST)	SB (OpenFlow & Others SB Protocols) NB (REST & Java RPC)
Virtualization	Mininet & Open vSwitch	Mininet & Open vSwitch	Built-in Emulation Virtual Tool	Mininet & Open vSwitch	Mininet & Open vSwitch
GUI	Yes	Yes (Initial Phase)	No	Web UI (Using REST)	Yes
REST API	No	Yes (For SB Interface only)	No	Yes	Yes
Productivity	Medium	Medium	High	Medium	Medium
Open Source	Yes	Yes	Yes	Yes	Yes
Documentation	Poor	Medium	Medium	Good	Medium
Language Support	Python	Python-Specific + Message Passing Reference	C/Ruby	Java + Any language that uses REST	Java
Modularity	Medium	Medium	Medium	High	High
Platform Support	Linux, Mac OS, and Windows	Most Supported on Linux	Linux Only	Linux, Mac & Windows	Linux
TLS Support	Yes	Yes	Yes	Yes	Yes
Age	1 year	1 year	2 years	2 years	2 Month
OpenFlow Support	OF v1.0	OF v1.0 v2.0 v3.0 & Nicira Extensions	OF v1.0	OF v1.0	OF v1.0
OpenStack Networking (Quantum)	NO	Strong	Weak	Medium	Medium

SwissSenseSynergy Interests using SDN in Wireless Access

- Traffic steering / radio planning
  - OpenFlow extension for WiFi port properties
  - Steering traffic based on statistics

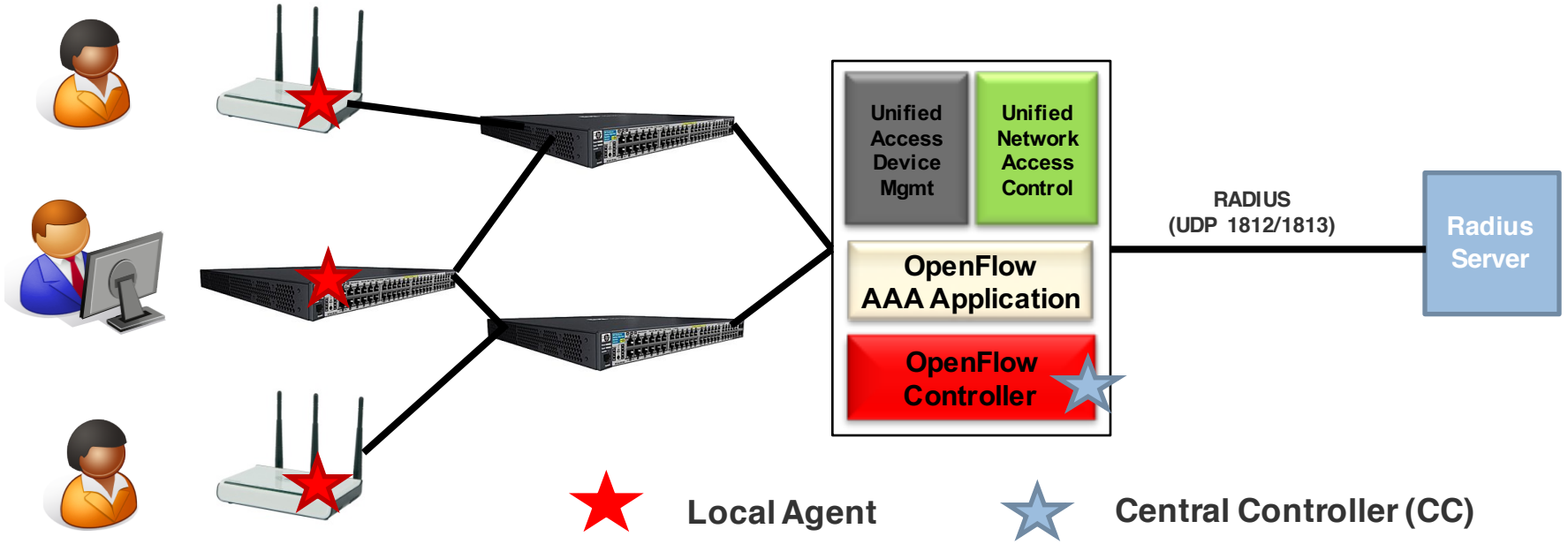
- Privacy-preserving location-based access control
  - OpenFlow message extension for privacy-related content

Traffic Steering / Radio Planning



- OF ***Action*** extension of real-time functions (rate-adaption)
- OF ***Match*** support (message extension) for WiFi port properties and 802.11 frames
- OF-Config support for deploying WiFi configurations
- Traffic steering based on monitored statistics and resource utilization
- Handover

Privacy-preserving Location-based Access Control



- OF ***Match*** support (message extension) for location-awareness
- OF ***Match*** support (message extension) for key-distribution
- OF captures/intercepts 802.1X messages
- AAA application on OF controller performs 802.1X AAA functions

Conclusions

- SDN & OpenFlow
- SDO activities
- OpenFlow extension for wireless networks
- SSS use-cases
  - Traffic steering
    - OpenFlow extension for WiFi ports
    - Traffic steering using collected statistics
  - Access control
    - OpenFlow extension for location-awareness and privacy mechanisms
  - Mininet + Floodlight + Open vSwitch (or OpenWRT)