Software Defined Networking and Network Function Virtualization

Carleton University

Canada's Capital University

Key: Understand ICT Industry

- Service Providers/Network Operators (i.e., China Telecom, Bell Canada, etc.)
  - Make money by providing communication services to end users (residential, enterprise)
  - Compete with each other on service offerings and price
    - Also compete with Over-The-Top service providers (SMS vs. WeChat)
  - Pay equipment manufacturers (Ericsson, Cisco, Huawei, ZTE, etc.) for networking equipment
- End users
  - Pay for services
- Equipment manufacturers
  - Sell equipment that meets service provider needs
  - Very few, if any, directly sell to private end user (smartphones, for example)
  - May sell equipment to enterprise users (Cisco, Juniper)
  - Compete ferociously (my experience with Ericsson, Alcatel-Lucent)

Motivations of Key Players

- End users
  - Get services they require at best possible price (i.e., cheap)
- Equipment manufacturers
  - Develop equipment that meets customer needs while selling at as high a price as possible
- Service providers
  - Stuck in the middle: provide cheap services with expensive equipment
    - Strategies: avoid being held hostage by single vendor, always source from multiple manufacturers
    - Key enabler: standardization (ITU, IETF)
  - Problem: to compete on services, need to be able to introduce new services fast
    - Standardization takes time (and tells competitors about planned services)
    - If changes to equipment are required, need buy-in from equipment manufacturers
      - Long product development cycles

Case Study: Load Balancing/Content Distribution (Simplified)

- Most Internet services are offered via dedicated servers
  - Google, Skype, WeChat, Alibaba, Amazon, Twitter, Youtube
- When new service is offered, it is easy and enough to have one server and a backup maybe
- As popularity grows, need to balance load across multiple physical devices (but transparent to users)
- Companies explored load-balancing strategies
  - Initially simple, using DNS capabilities
  - Grows more complex to accommodate different application response times, geographic distribution, etc.
- Load balancing could become a service (and therefore revenue source) for ISPs
  - But in reality it is often not
    - Done in-house for large service providers
    - Using over-the-top services such as Akami and Inktomi (not in business anymore)
    - Use cloud service providers: Amazon, Google, Ali…. etc.

What is the Problem?

- Innovation Process slow for Service Providers
  - Their vendors have long product development cycles
  - Their networks are not inherently designed to easily accommodate new services
    - Not trivial when also having to assure that everything else will still work
      - Example: Telephone feature interaction, well over 100 features
  - Mentality issue
    - Service Providers: SMS app in South Korea
    - Equipment Manufacturers: Nortel's Javaphone

- SDN and NFV won't solve ALL these problems but
  - Helps service providers to gain more independence from equipment manufacturers
  - Make it easier to introduce new services as more things are done in software

Take It Not Just From Me

- Ciena’s CTO: The Future of Networks

Some Themes from Video

- Single network to carry a range of different applications
- Applications on top of the network create services
  - Network isolation, highly reliable VPN, manage data centre traffic
- Reduce cost of running the network, in particular as bandwidth/capacity scale up
- Optimize packet handling
- More is done in software (virtualize network elements)
- Distribute the control plane
- Working below the IP layer is cheaper
  - Ciena sells fibre equipment

Some Food for Thought

- Why would equipment manufacturer join the bandwagon?
  - Ciena is NOT a service provider

- Making it easier to add new services is NOT a new idea
  - 1990s: Intelligent Networks (IN) and Advanced Intelligent Networks (AIN) were large efforts within the ITU to provide a flexible platform to introduce new services into the telecommunications networks
  - 2000s: Cisco was dominant switch/router company in the Internet, had its own proprietary OS that was closed to everyone ➔ hard to innovate
    - Active Networks: packets carry code as well as data to allow for a wide range of network behaviours/services
    - XORP: eXtensible Open Routing Platform, a project out of Stanford (?), provides open-source platform to enable innovations in routing
  - None of these you probably heard off and none of them were (in my view) successful
  - Is there anything different now that would indicate more success?

Overview of Presentation

- SDN: separate data and forwarding plane, allow separate controller to manage network resources in an optimal way
  - Data forwarding equipment COTS (Commercial Off-The-Shelf): cheap
  - Smarts are in centralized controller (but Ciena CTO: distribute control)
- NFV: create virtual instances of services
  - Freedom to place them anywhere in the network
  - Freedom to combine things in new ways to offer new services
- Are SDN and NFV the same? Complementary? Unrelated

SDN: Software Defined Networking

Carleton
UNIVERSITY
Canada's Capital University

SDN

- Two ways to look at SDN
  - Design paradigm that is based on the idea of separating data forwarding and network control
  - Reference to a specific (dominant) protocol: OpenFlow
- Similar to TCP/IP

Initial Motivation: Open Systems for Networking Research

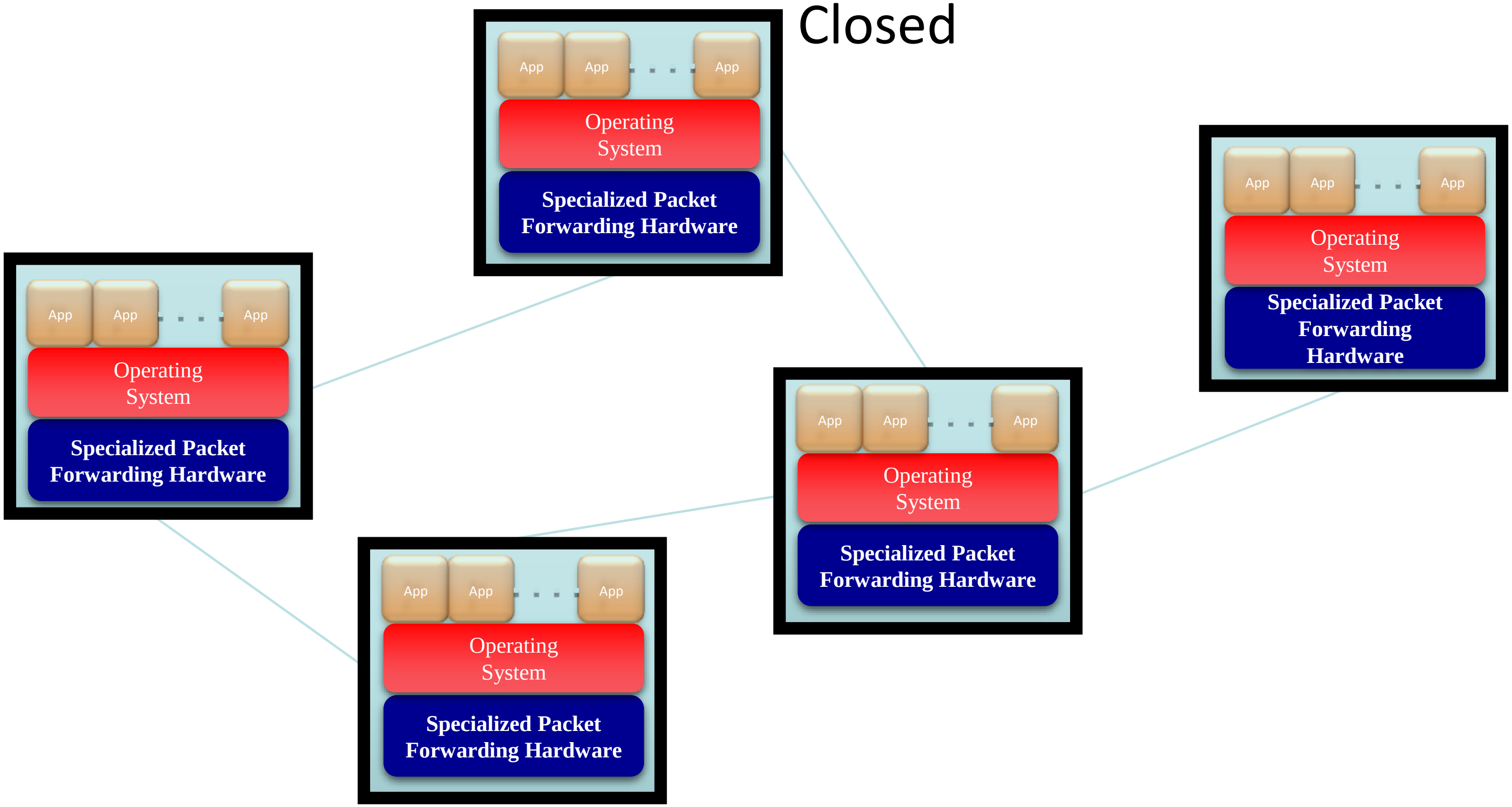
	Performance Fidelity	Scale	Real User Traffic?	Complexity	Open
Simulation	medium	medium	**no**	medium	yes
Emulation	medium	**low**	**no**	medium	yes
Software Switches	**poor**	**low**	yes	medium	yes
NetFPGA	high	**low**	yes	**high**	yes
Network Processors	high	medium	yes	**high**	yes
Vendor Switches	high	high	yes	low	**no**

gap in the tool space

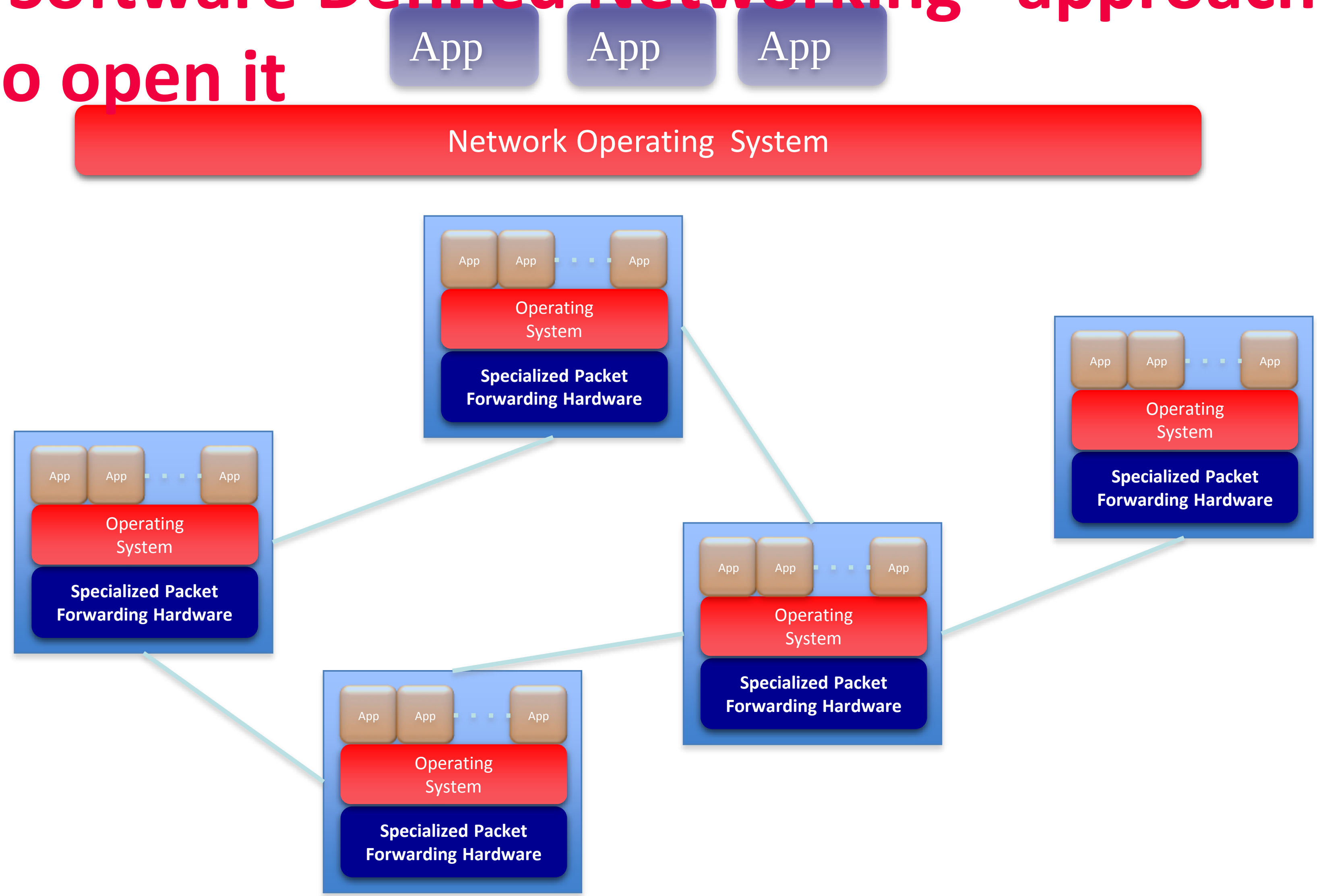
none have all the desired attributes!

Current Internet

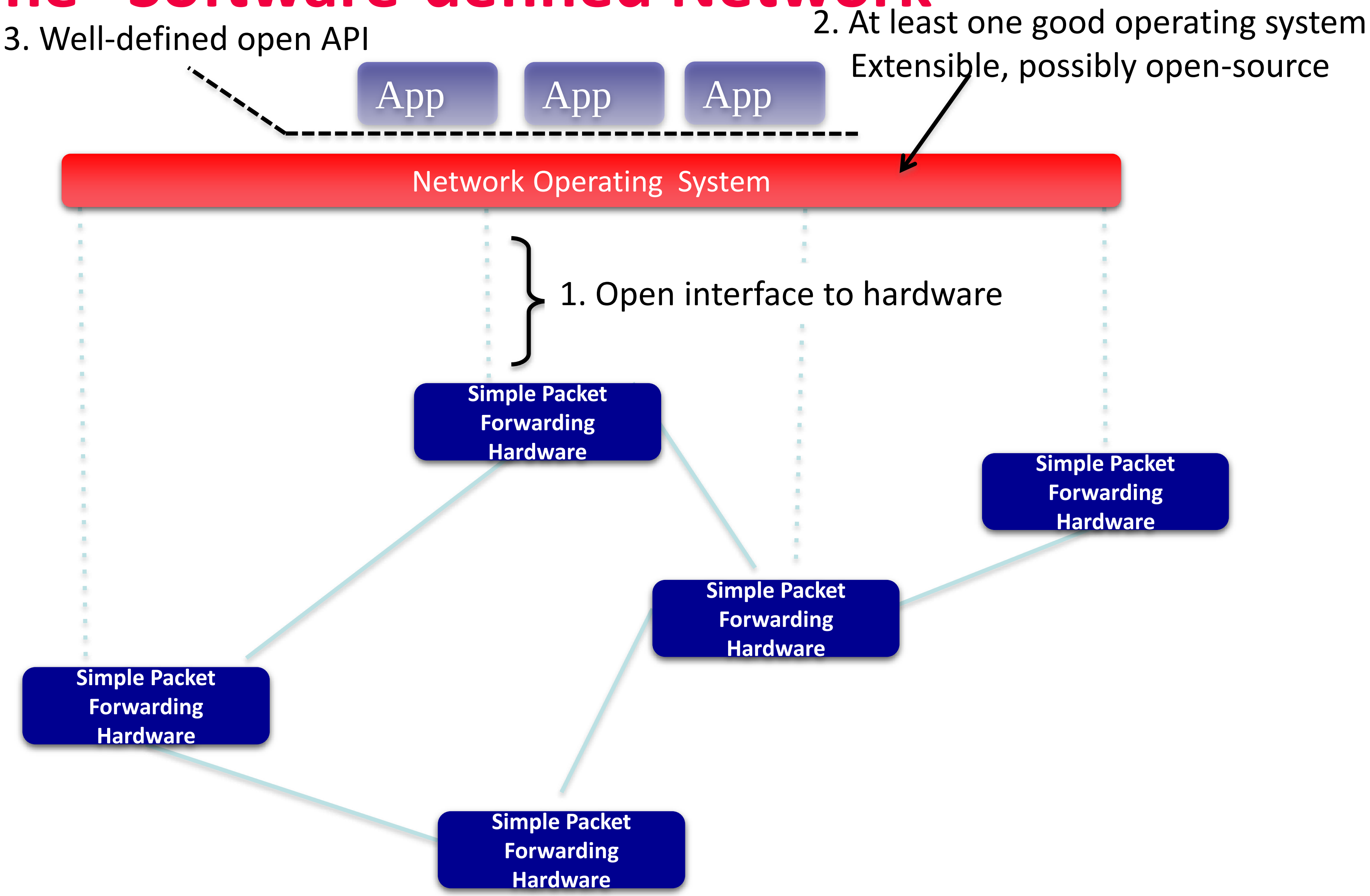
Closed to Innovations in the Infrastructure


Closed
App
App
. . . .
App
Operating System
Specialized Packet Forwarding Hardware
App
App
. . . .
App
Operating System
Specialized Packet Forwarding Hardware
App
App
. . . .
App
Operating System
Specialized Packet Forwarding Hardware
App
App
. . . .
App
Operating System
Specialized Packet Forwarding Hardware
App
App
. . . .
App
Operating System
Specialized Packet Forwarding Hardware

"Software Defined Networking" approach to open it


App
App
App
Network Operating System
App
App
App
Operating System
Specialized Packet Forwarding Hardware
App
App
App
Operating System
Specialized Packet Forwarding Hardware
App
App
App
Operating System
Specialized Packet Forwarding Hardware
App
App
App
Operating System
Specialized Packet Forwarding Hardware
App
App
App
Operating System
Specialized Packet Forwarding Hardware

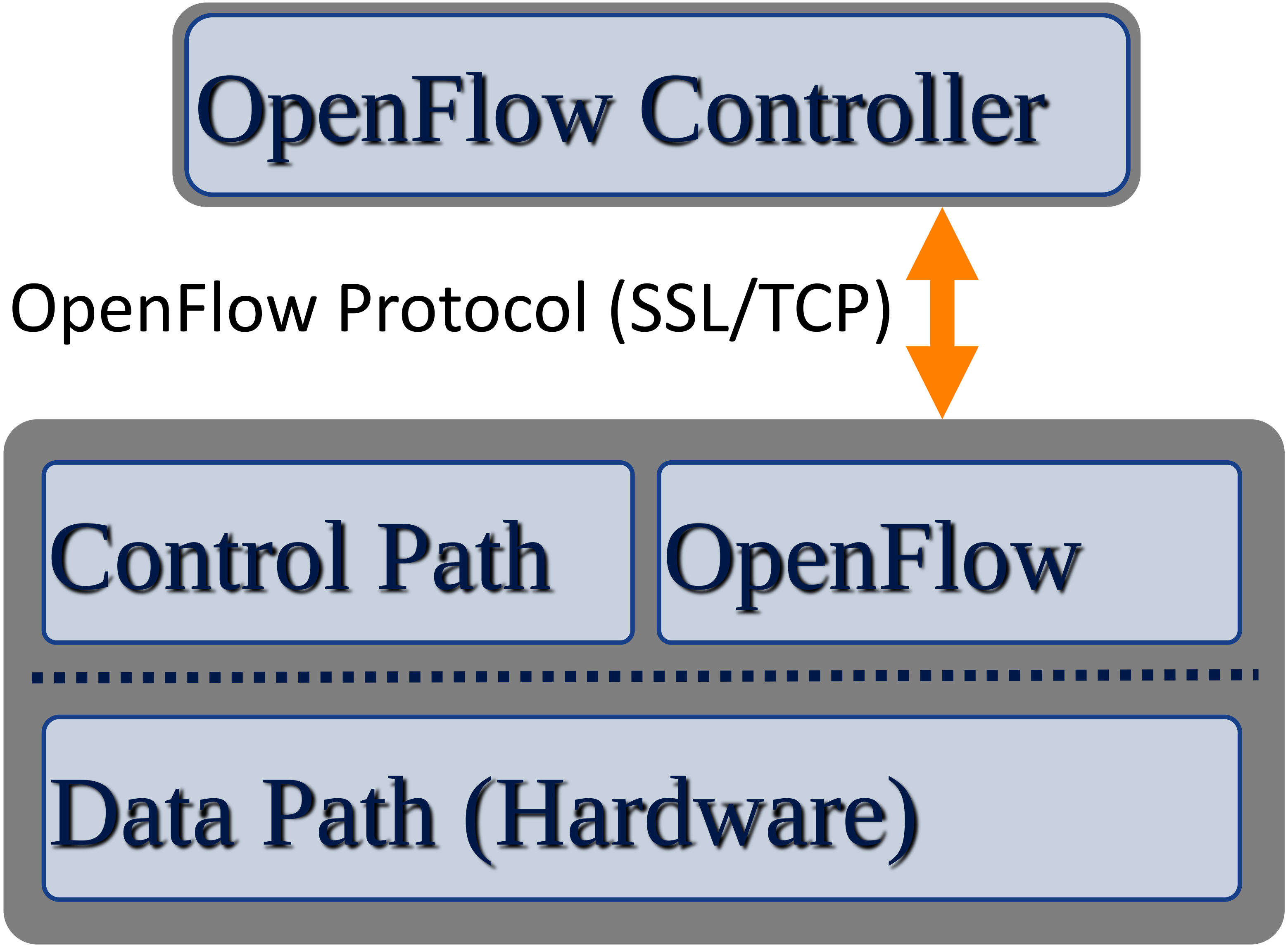
The "Software-defined Network"


3. Well-defined open API
2. At least one good operating system
Extensible, possibly open-source
App
App
App
Network Operating System
1. Open interface to hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware
Simple Packet Forwarding Hardware

Intro to SDN as a Design Paradigm

OpenFlow

- One specific protocol that implements communication between controller and forwarding elements (open API)
- Also referred to as “southbound interface”
- Still needed:
  - Controllers
  - Standard open APIs for Applications: the northbound interface
  - Also, for multi-controller setups: east- and west-bound interfaces



OpenFlow Controller

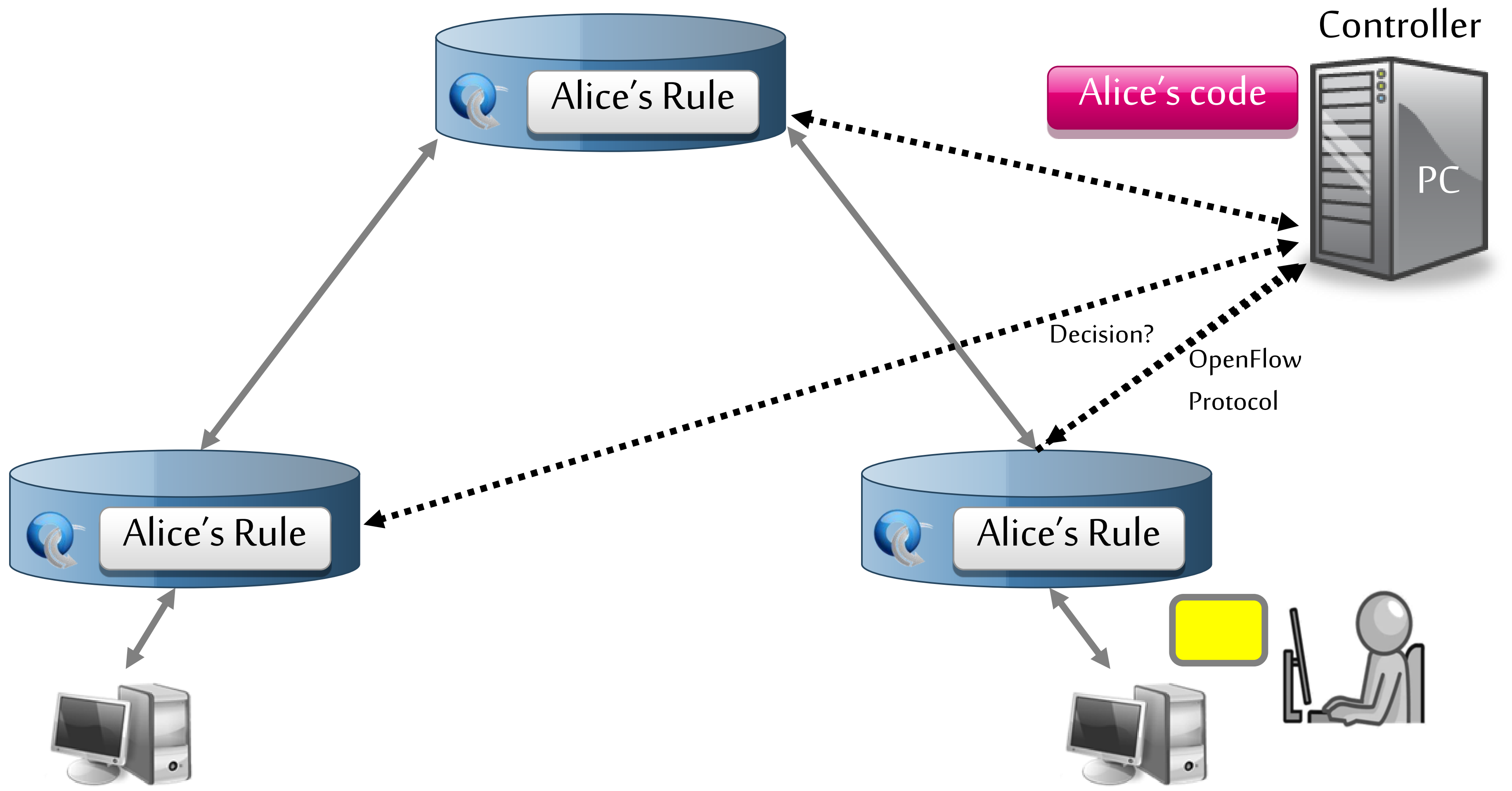
OpenFlow Protocol (SSL/TCP)

Control Path

OpenFlow

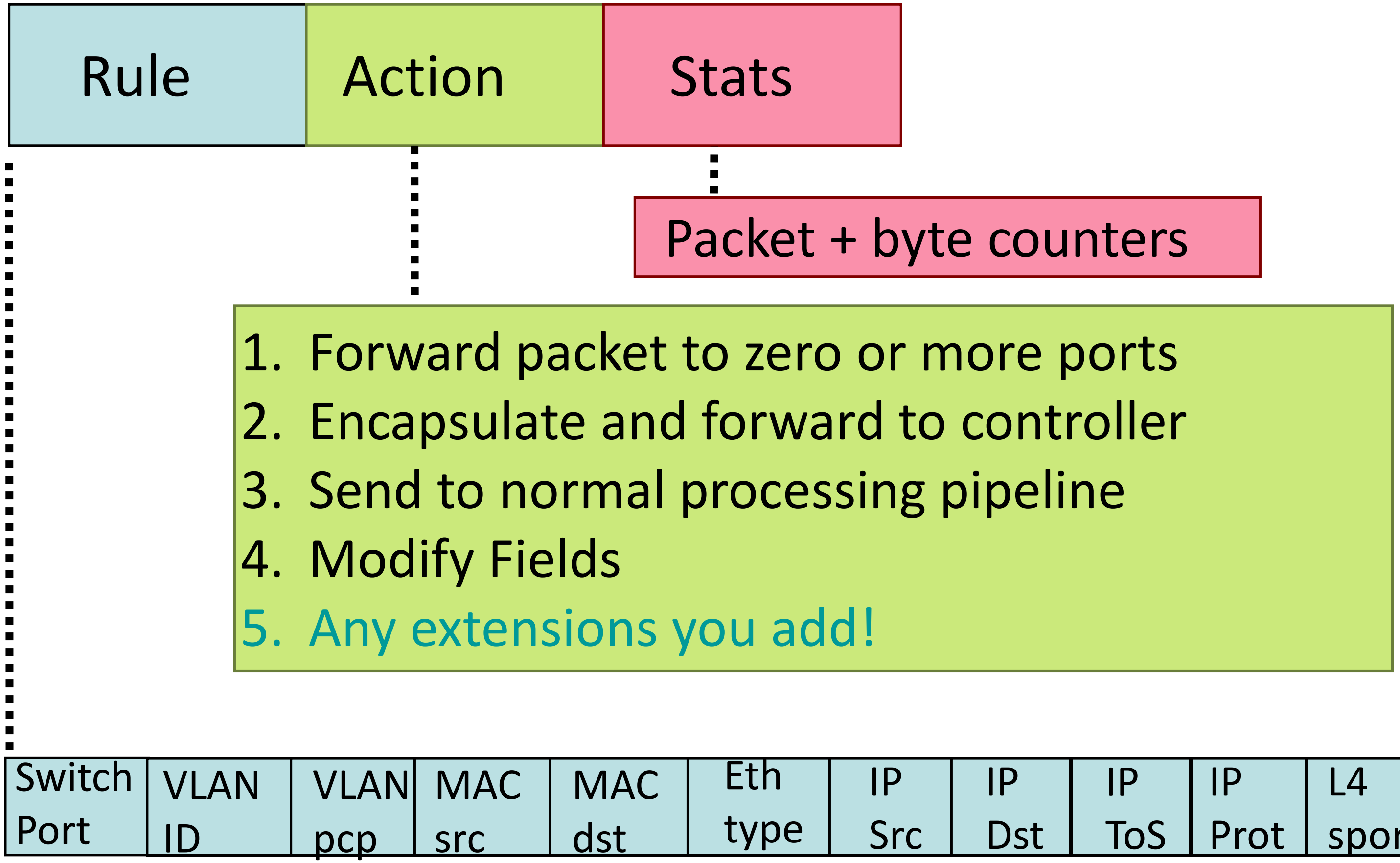
Data Path (Hardware)

OpenFlow Usage



OpenFlow offloads control intelligence to a remote software

OpenFlow Basics: Flow Table Entries



Rule | Action | Stats

Packet + byte counters

1. Forward packet to zero or more ports
2. Encapsulate and forward to controller
3. Send to normal processing pipeline
4. Modify Fields
5. Any extensions you add!

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

+ mask what fields to match

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

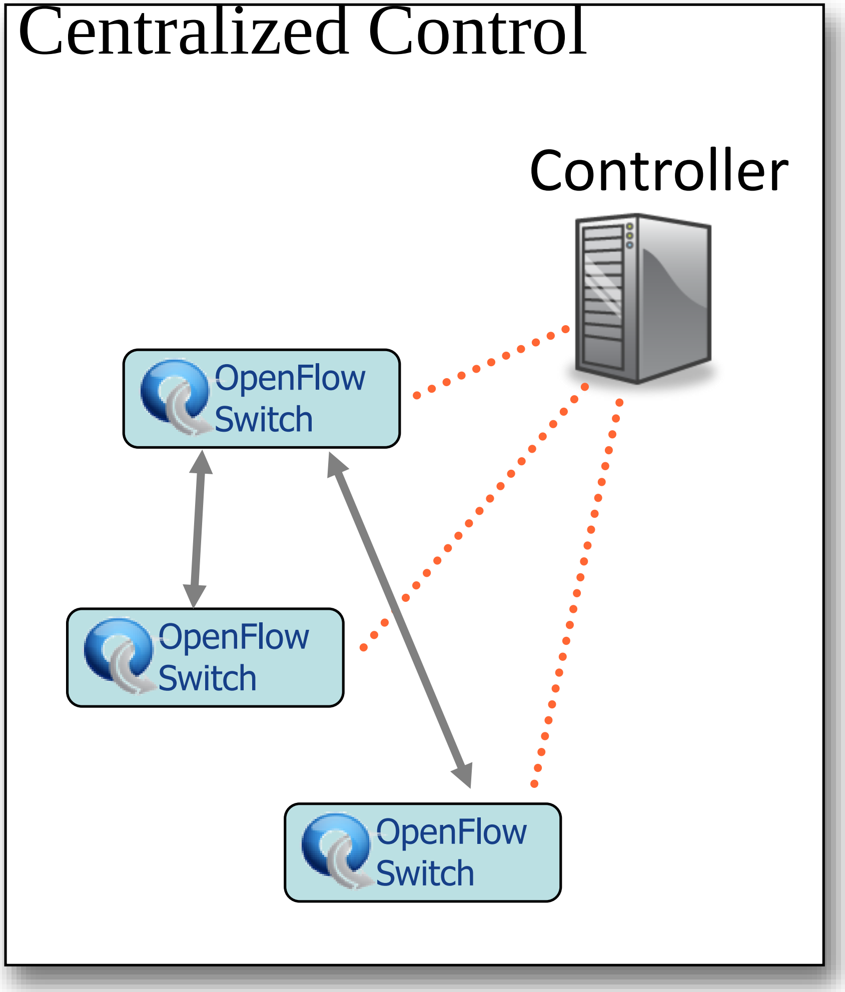
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

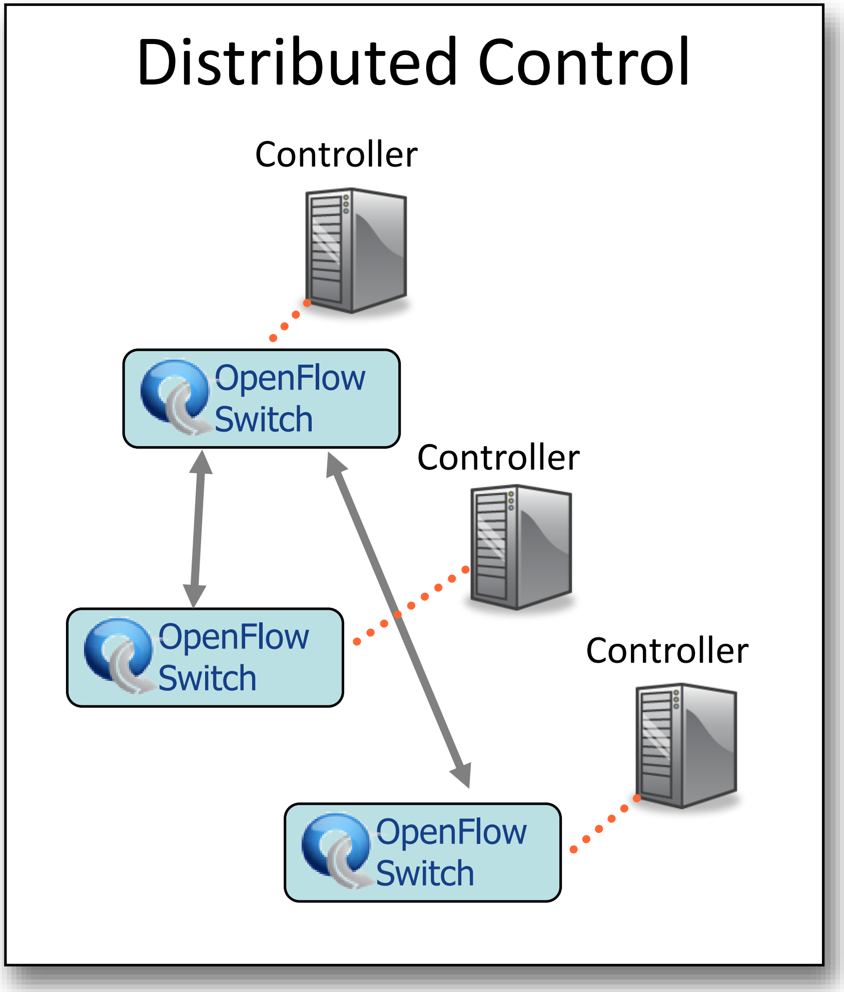
VLAN Switching

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

Centralized vs Distributed Control

Both models are possible with OpenFlow





Flow Routing vs. Aggregation

Both models are possible with OpenFlow

Flow-Based

- Every flow is individually set up by controller
- Exact-match flow entries
- Flow table contains one entry per flow
- Good for fine grain control, e.g. campus networks

Aggregated

- One flow entry covers large groups of flows
- Wildcard flow entries
- Flow table contains one entry per category of flows
- Good for large number of flows, e.g. backbone

Reactive vs. Proactive (pre-populated)

Both models are possible with OpenFlow

Reactive

- First packet of flow triggers controller to insert flow entries
- Efficient use of flow table
- Every flow incurs small additional flow setup time
- If control connection lost, switch has limited utility

Proactive

- Controller pre-populates flow table in switch
- Zero additional flow setup time
- Loss of control connection does not disrupt traffic
- Essentially requires aggregated (wildcard) rules

Usage Examples

- 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

openflow.org/videos

- VM migration
- Server Load balancing
- Mobility manager
- Power management
- Network monitoring and visualization
- Network debugging
- Network slicing

… and much more you can create!

Where is it going?
The Open Networking Foundation:

The founding Consortium

T

Google

facebook

YAHOO!

verizon

NTT Communications

Microsoft

Promoter Members:

- Operators and service providers
- Make up the board of directors
- Have voting rights
- Representative of DTAG is Bruno Orth (GTN S&A)

Adopter Members (as of Feb 2012)

List of Members:

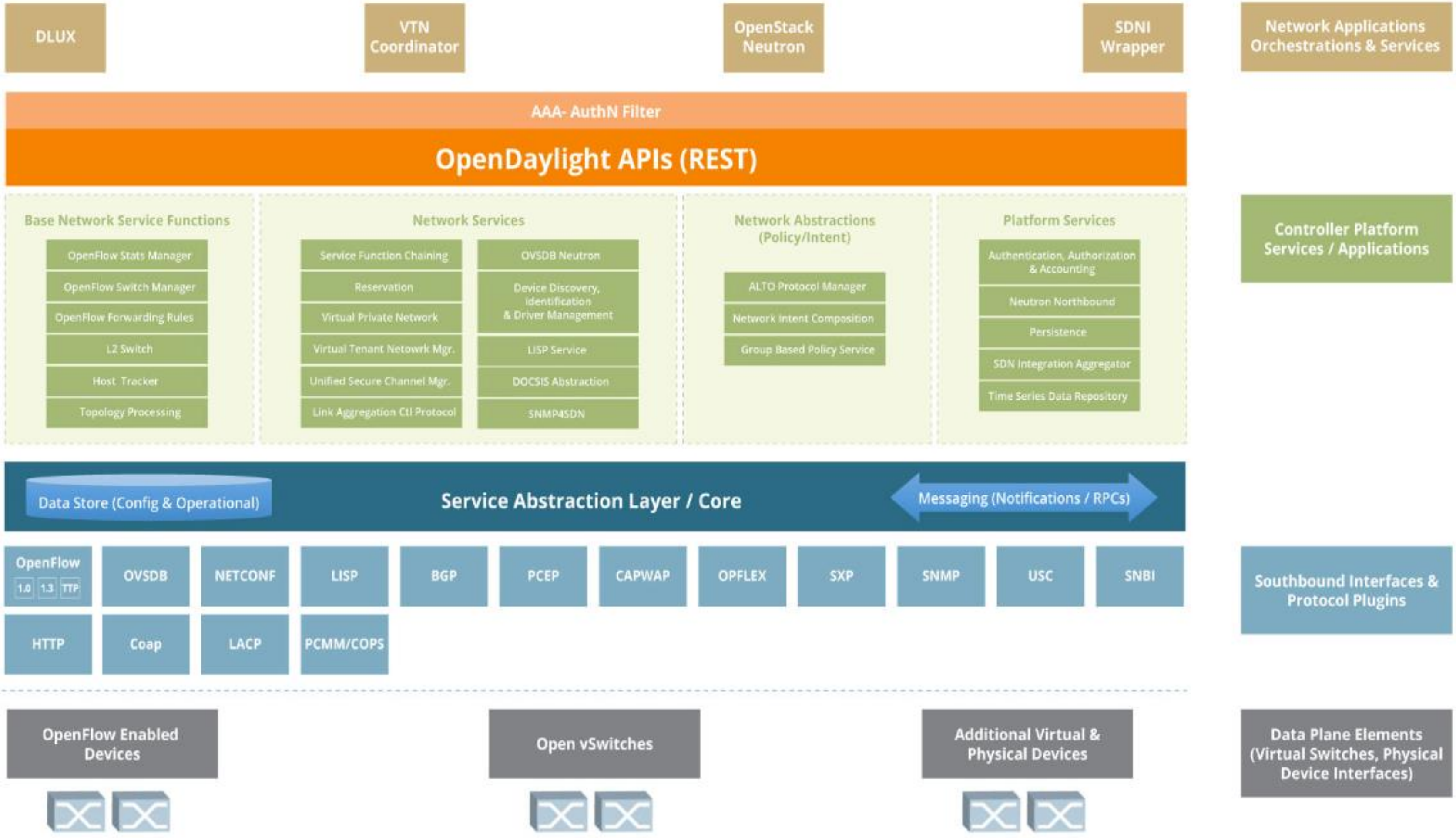
- Big Switch Networks
- Broadcom
- Brocade
- Ciena
- Cisco
- Citrix
- Comcast
- CompTIA
- Cyan
- Dell
- Elbrys
- Ericsson
- ETRI
- Extreme Networks
- EZchip
- Force10Networks
- Fujitsu
- Hitachi
- HP
- Huawei
- IBM
- Infoblox
- Intel
- IP Infusion
- Ixia
- Juniper Networks
- Korea Telecom
- LineRate Systems
- LSI
- Marvell
- Mellanox
- Metaswitch Networks
- Midokura
- NEC
- Netgear
- Netronome
- Nicira Networks
- Nokia Siemens Networks
- Plexxi Inc.
- Pronto Systems
- Radware
- Riverbed Technology
- Samsung
- Spirent
- Tencent
- Texas Instruments
- Vello Systems
- VMware
- ZTE Corporation

Where it's going

- OF v2+
  - generalized matching and actions: an "instruction set" for networking

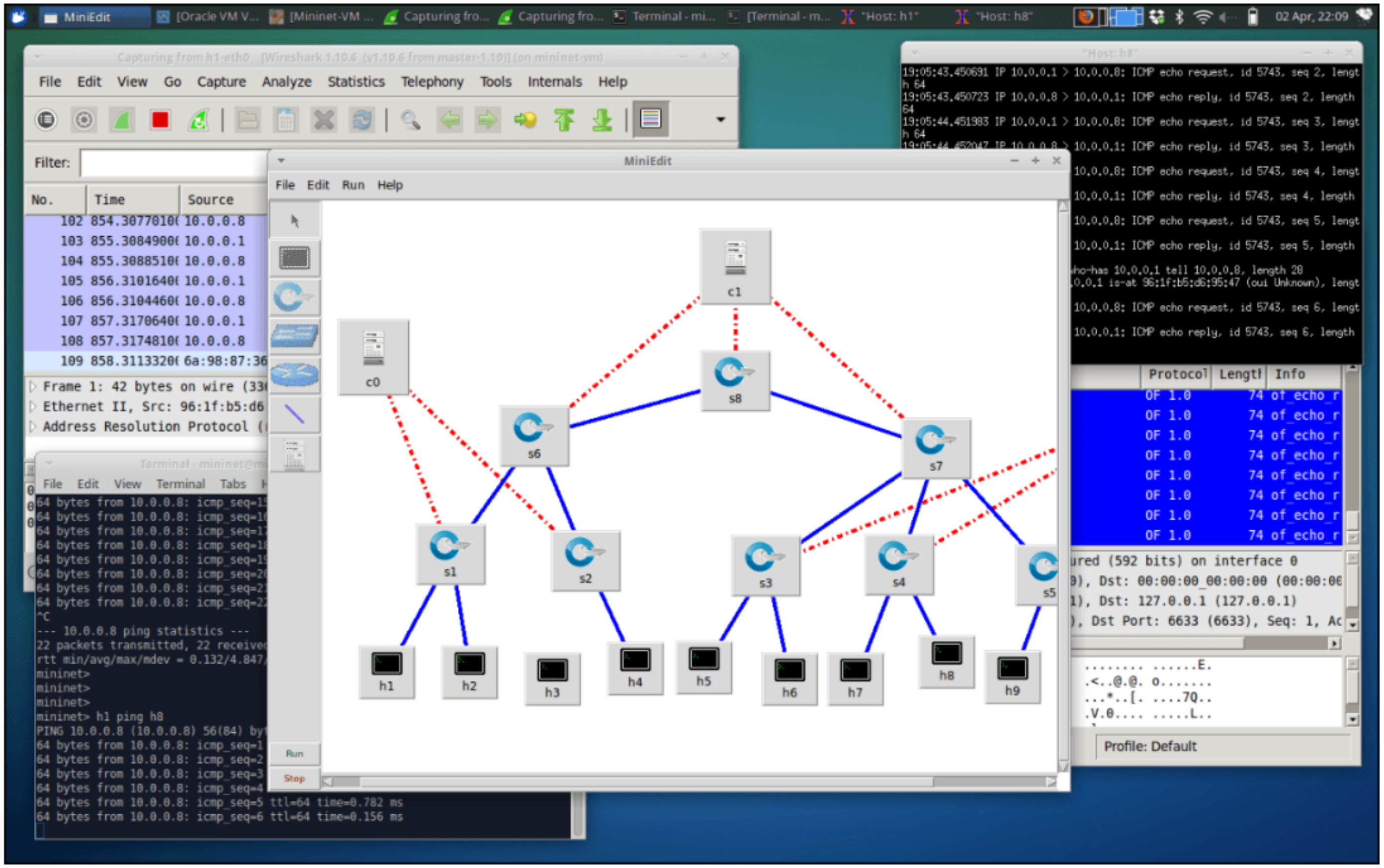
- Several other working groups have been created:
  - **Hybrid group**: Specifies how OpenFlow can be included into legacy switches without assuming clean-slate
  - **Config group**: Will specify an independent protocol that will help configure OpenFlow parameters out-of-band
  - .... And more

Key Tools: OpenDayLight – Open Source SDN Controller (https://www.opendaylight.org/)


DLUX
VTN Coordinator
OpenStack Neutron
SDNI Wrapper
Network Applications Orchestrations & Services
AAA- AuthN Filter
OpenDaylight APIs (REST)
Base Network Service Functions
OpenFlow Stats Manager
OpenFlow Switch Manager
OpenFlow Forwarding Rules
L2 Switch
Host Tracker
Topology Processing
Network Services
Service Function Chaining
Reservation
Virtual Private Network
Virtual Tenant Netowrk Mgr.
Unified Secure Channel Mgr.
Link Aggregation Ctl Protocol
OVSDB Neutron
Device Discovery, identification & Driver Management
LISP Service
DOCSIS Abstraction
SNMP4SDN
Network Abstractions (Policy/Intent)
ALTO Protocol Manager
Network Intent Composition
Group Based Policy Service
Platform Services
Authentication, Authorization & Accounting
Neutron Northbound
Persistence
SDN Integration Aggregator
Time Series Data Repository
Controller Platform Services / Applications
Data Store (Config & Operational)
Service Abstraction Layer / Core
Messaging (Notifications / RPCs)
OpenFlow
1.0
1.3
TTP
OVSDB
NETCONF
LISP
BGP
PCEP
CAPWAP
OPFLEX
SXP
SNMP
USC
SNBI
HTTP
Coap
LACP
PCMM/COPS
Southbound Interfaces & Protocol Plugins
OpenFlow Enabled Devices
Open vSwitches
Additional Virtual & Physical Devices
Data Plane Elements (Virtual Switches, Physical Device Interfaces)

Key Tools: MiniNet (http://mininet.org/)

- Set up emulated network in PC or laptop
  - Supports a range of SDN controllers and switches
  - Supports OF 1.3
  - Very efficient in running networks with 100s of emulated SDN-enabled controllers



```
No.    Time          Source
  102  854.3077010(  10.0.0.8
  103  855.3084900(  10.0.0.1
  104  855.3088510(  10.0.0.8
  105  856.3101640(  10.0.0.1
  106  856.3104460(  10.0.0.8
  107  857.3170640(  10.0.0.1
  108  857.3174810(  10.0.0.8
  109  858.3113320(  6a:98:87:36
▷ Frame 1: 42 bytes on wire (33
▷ Ethernet II, Src: 96:1f:b5:d6
▷ Address Resolution Protocol (
```

```
64 bytes from 10.0.0.8: icmp_seq=15
64 bytes from 10.0.0.8: icmp_seq=16
64 bytes from 10.0.0.8: icmp_seq=17
64 bytes from 10.0.0.8: icmp_seq=18
64 bytes from 10.0.0.8: icmp_seq=19
64 bytes from 10.0.0.8: icmp_seq=20
64 bytes from 10.0.0.8: icmp_seq=21
64 bytes from 10.0.0.8: icmp_seq=22
^C
--- 10.0.0.8 ping statistics ---
22 packets transmitted, 22 received
rtt min/avg/max/mdev = 0.132/4.847/
mininet>
mininet>
mininet>
mininet> h1 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) by
64 bytes from 10.0.0.8: icmp_seq=1
64 bytes from 10.0.0.8: icmp_seq=2
64 bytes from 10.0.0.8: icmp_seq=3
64 bytes from 10.0.0.8: icmp_seq=4
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.782 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.156 ms
```

```
19:05:43.450691 IP 10.0.0.1 > 10.0.0.8: ICMP echo request, id 5743, seq 2, lengt
h 64
19:05:43.450723 IP 10.0.0.8 > 10.0.0.1: ICMP echo reply, id 5743, seq 2, length
64
19:05:44.451983 IP 10.0.0.1 > 10.0.0.8: ICMP echo request, id 5743, seq 3, lengt
h 64
19:05:44.452047 IP 10.0.0.8 > 10.0.0.1: ICMP echo reply, id 5743, seq 3, length
10.0.0.8: ICMP echo request, id 5743, seq 4, lengt
10.0.0.1: ICMP echo reply, id 5743, seq 4, length
10.0.0.8: ICMP echo request, id 5743, seq 5, lengt
10.0.0.1: ICMP echo reply, id 5743, seq 5, length
ho-has 10.0.0.1 tell 10.0.0.8, length 28
.0.0.1 is-at 96:1f:b5:d6:95:47 (oui Unknown), lengt
10.0.0.8: ICMP echo request, id 5743, seq 6, lengt
10.0.0.1: ICMP echo reply, id 5743, seq 6, length
```

```
Protocol  Lengtl  Info
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
OF 1.0        74  of_echo_r
ured (592 bits) on interface 0
0), Dst: 00:00:00_00:00:00 (00:00:00
1), Dst: 127.0.0.1 (127.0.0.1)
), Dst Port: 6633 (6633), Seq: 1, Ac
........ ......E.
.<..@.@. o.......
...*..[. ....7Q..
.V.0.... .....L..
```

NFV: Network Function Virtualization

Carleton
UNIVERSITY
Canada's Capital University

Brief Intro to NFV

Alan Talks Tech on Network Functions Virtualization (NFV)

www.Spirent.com
YouTube "alantalkstech"
http://alantestwiki.pbworks.com

SPIRENT PROPRIETARY

1

NFV Example: Relocate Functions to/from Customer Premises

- NFV quite new, raises many interesting challenges
  - How to define new services?
  - Where to place them in the network?
  - Performance of resulting network?
  - Combining existing services into new services?
    - Personally: seems quite similar to lot of discussion/research we had when orchestrating Web Services

- Discuss some of these issues in the context of offering virtual network functions to end-users (residential customers)
  - Three choices:
    - No virtualization
    - Have cheap CPE (provide basic connectivity) with services offered in data centre
    - Have more powerful CPE that allows to install one or multiple functions

Justifications for Virtualization

Dedicated hardware almost always cheap to produce (hardware) and has high performance

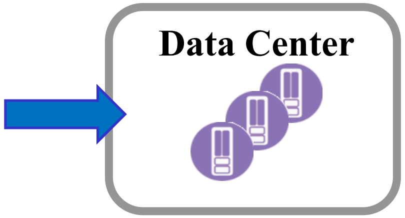
- Mass production leads to economies of scale
- Hardware has higher throughput, lower power consumption, less heat dissipation/lower energy cost

The justifications for virtualization are initially hard(er) to grasp

- lower development efforts and cost (no PCBs)
- flexibility and ability to *upgrade* functionality
- *chaining multiple function*s on a single platform
- facilitating *function relocation*

Function Relocation:

moving the network function from its conventional place
to some other place (e.g., to a **D**ata **C**enter)



Virtualization vs. Function Relocation

Relocation has received much attention in the networking community
since moving networking functions to **D**ata **C**enters
often enables benefiting from economies of scale

This emphasis on this single reason for virtualization has been so strong
that it has led many to completely confuse *virtualization* and *relocation*

when in fact

- Non-virtualized functions can be relocated (at the expense of CAPEX and truck rolls)
- virtualized functions can remain in situ

Function Placement

Telecomm functionalities tend to be placed in *conventional* locations

- Customer Premises
- Aggregation Point
- Point of Presence
- Core Network Edge
- Data Center

Some telecomm functionalities really ***must*** reside at their locations

- LoopBack testing (what would it mean to move LB to a data center?)
- End-to-End security (why encrypt packets after they traverse the network)

Some ***should*** be left in the conventional locations

- End-to-End performance monitoring (it wouldn't be end-to-end – would it ?)
- DDoS attack blocking (best to block as close to source as possible)

Some **may** be placed almost anywhere

- Path Computation
- Charging/billing functionality

Distributed NFV

With **V**irtualized **N**etwork **F**unctions (***not*** virtualized network resources)
placement is no longer dictated by convention or equipment
placement can be optimally determined anywhere in the network

The idea of optimally placing virtualized network functions in the network
is called **Distributed NFV**

Placement decisions can be based on

- resource availability (computational power, storage, bandwidth)
- *real-estate* availability and costs
- energy and cooling
- management and maintenance
- other economies of scale
- function chaining order
- policy
- security and privacy
- regulatory issues
- …

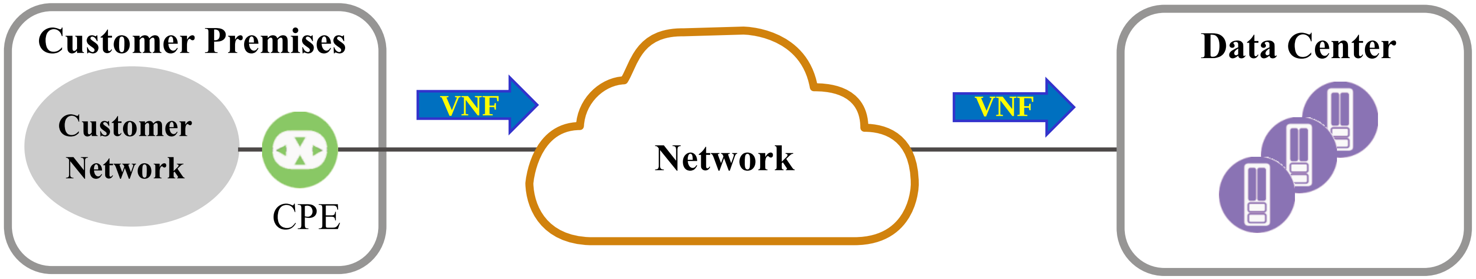
Consider moving a DPI engine from where it is needed
and sending the packets to be inspected to a remote DPI engine
If bandwidth is unavailable or expensive or excessive delay is added
then DPI **must not** be relocated
even if computational resources are less expensive elsewhere!

Some D-NFV criteria

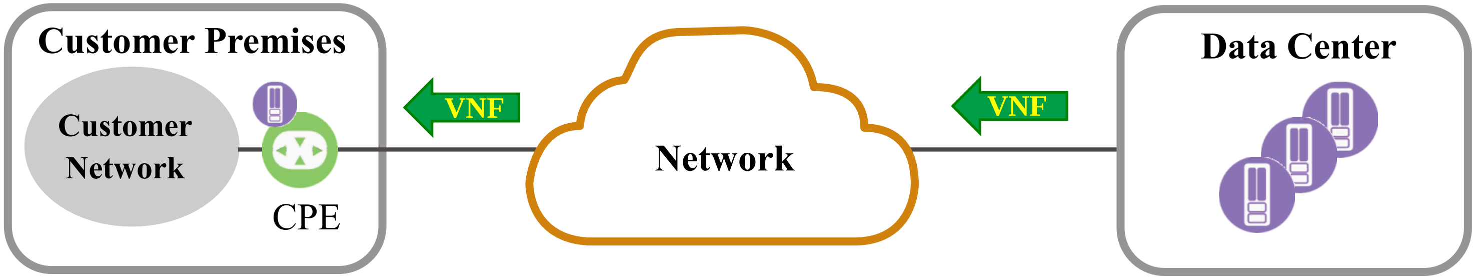
Criterion	Description
Feasibility	• Some functions can't be relocated from customer site, e.g., loopback testing, end-to-end security, traffic conditioning, encryption, WAN optimization
Performance	• Some functions perform better at the customer premises, e.g., end-to-end QoS, application QoE monitoring • Some functions may degrade due to network constraints (bandwidth, delay, availability)
Cost	• Needs for higher network performance and resiliency may lead to cost increases, even with Data Center economies of scale
Policy	• Some functions need to be left near the customer due to corporate privacy, security, and access policies • Regulatory restrictions (e.g., on moving data across jurisdictions) may also apply

Relocation and CPEs

One relocation that has been actively discussed recently
is being called *virtualization of the CPE* (vCPE) (*virtualization* means *relocation*)
Here CPE functionality is virtualized and moved from the customer premises
leaving behind only minimal functionality (OAM, traffic conditioning)



Equally interesting *is virtualization in the CPE*
Here functionalities are moved **to** the customer premises



VM-enhanced NID

Virtualization in the CPE
requires a customer premises device capable of hosting VNFs

A reasonable device would be the **N**etwork **I**nterface **D**emarcation device

For example, RAD has integrated an x86 module into its ETX2 L2/L3 NID

This device retains all its NID functionality (OAM, traffic conditioning)
and acquires the capability of hosting arbitrary software functions

The combined ETX/VM device is
located at the customer premises
under the control of the **S**ervice **P**rovider

Thus the SP can rapidly download arbitrary functionalities to the NID
for its own purposes (diagnostics, visibility, blocking traffic, etc.)
as a Value Added Service for the customer (firewall, NAT, IDS, etc.)
without the need for installing any new network equipment

Advantages of VM-enhanced NID

The NID needs to be deployed in any case
and the additional cost of the computational power is minimal

On-site installation, maintenance, and energy costs
are much lower than for multiple dedicated devices

The marginal cost of a VNF is that of a software license plus OPEX

VNFs can be downloaded on-demand and very rapidly
and can be activated/deactivated/removed as required

Multiple VNFs can be chained on a single device
the only limitation being the module’s computational power and memory

The CPU connects to the internal NID switch ports
and so can operate on packets at various stages (ingress, in-process, egress)

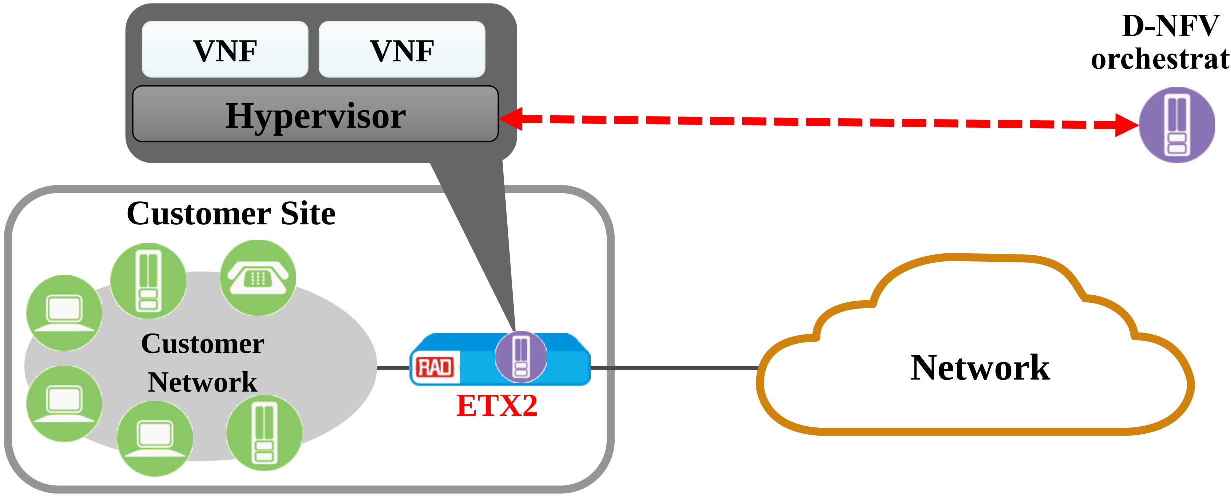
VAS VNFs can be offered on a trial basis
enabling a “try and buy” approach

ETX/VM architecture

The ETX/VM houses three virtual entities

1. standard ETX NID (OAM, policing, shaping, etc.)
2. VM infrastructure (hypervisor)
3. VNFs that run on the VM infrastructure

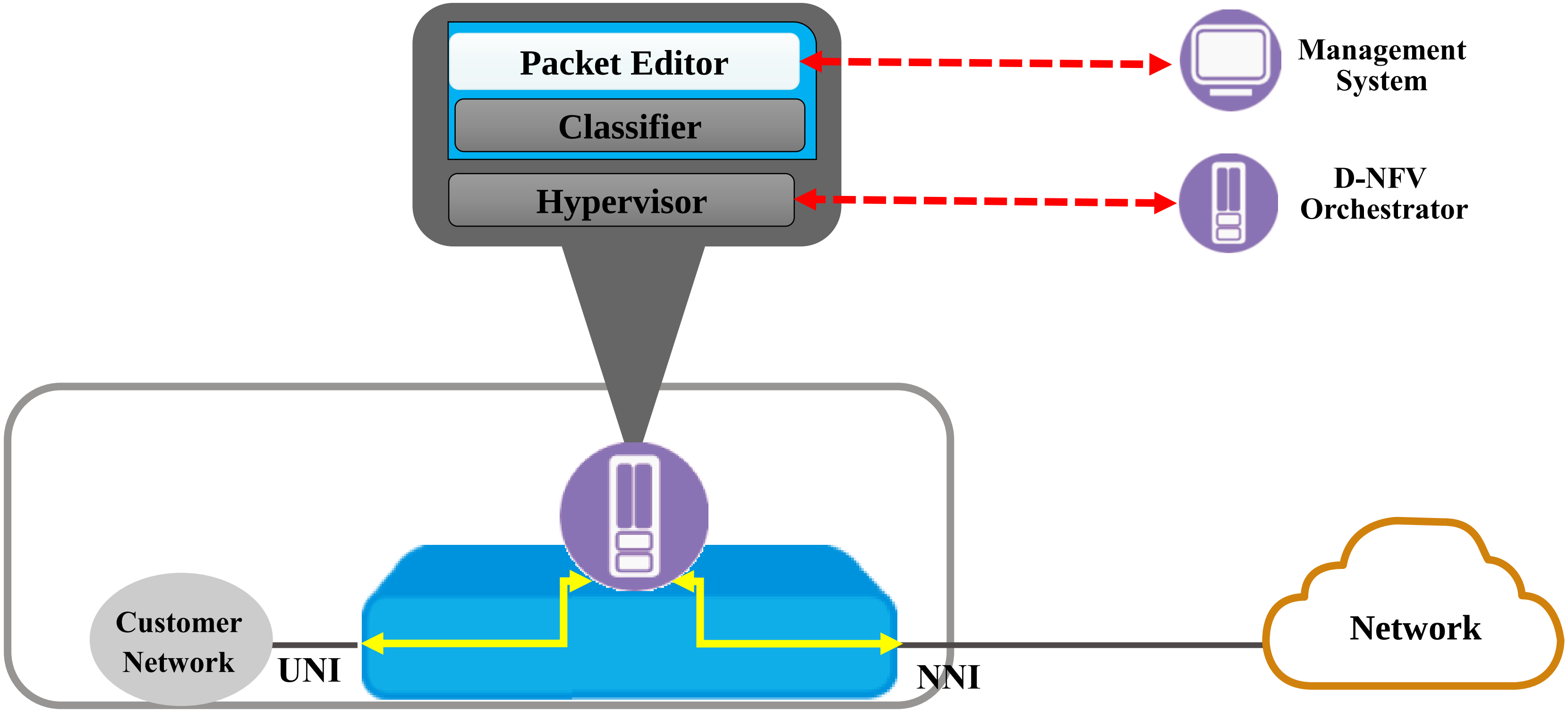
The VNFs are managed by an NFV orchestrator
and are written by compliant vendors or by the Service Providers themselves



Example: Packet Editing

This simple VNF edits particular packet headers, e.g., to

- swap/add/remove VLAN tags or MPLS labels
- tunnel certain packets across another network
- remark packet priorities

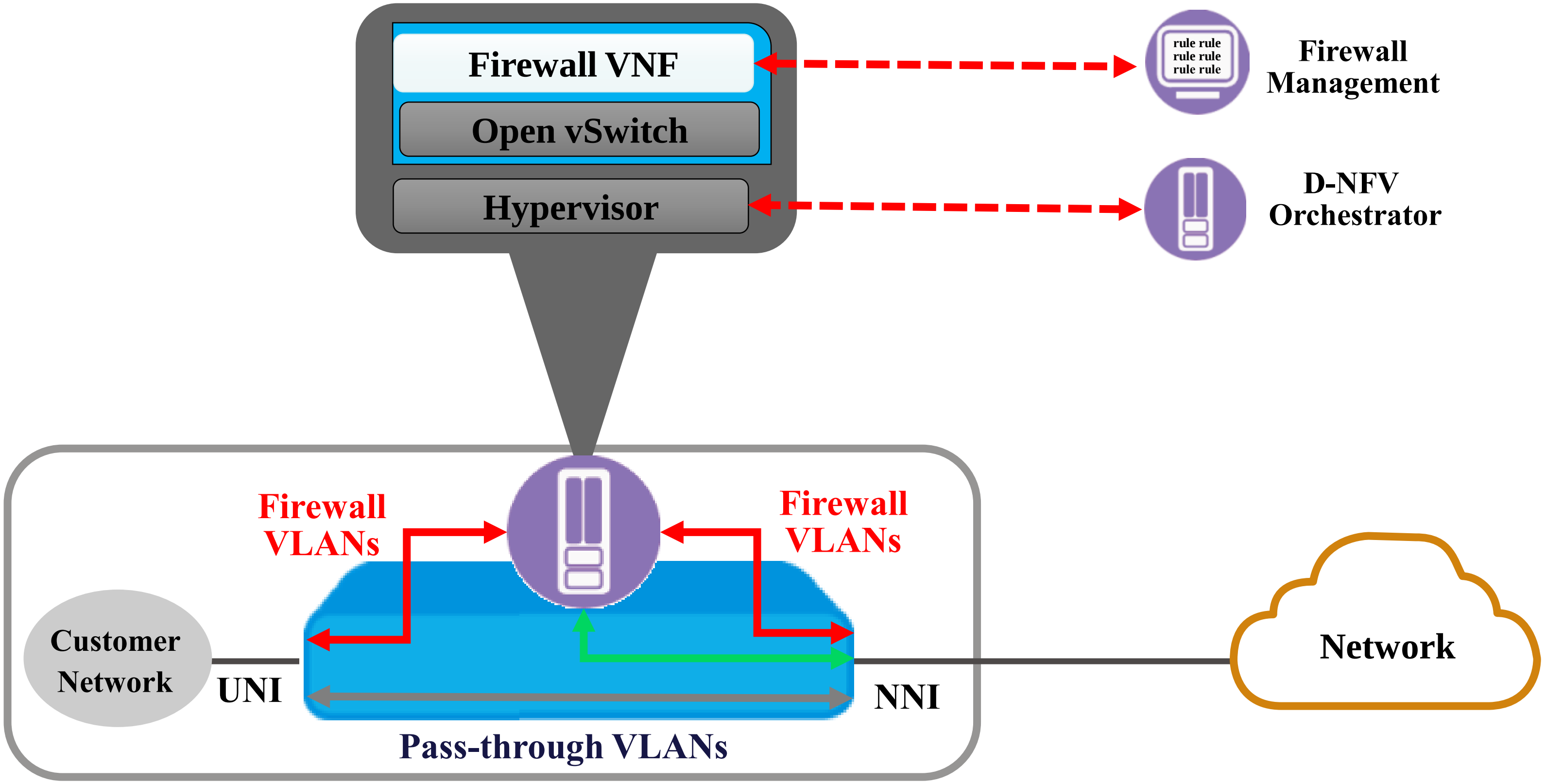


Example: Firewall

As another example, consider a firewall VAS

The hypervisor and vSwitch are Open Source software

The firewall VNF is a third-party application



SDN and NFV

Carleton
UNIVERSITY
Canada's Capital University

Relationship between SDN and NFV

- On the one hand, they are quite separate
  - SDN: separating data and control plane
  - NFV: virtualizing network resources and services
- But they are also similar
  - SDN: separate data and control plane to make it easier to offer NEW SERVICES
  - NFV: virtualize network equipment and make it easier to deploy NEW SERVICES
- To wrap it up: interview at Huawei SoftCom - Ren Xudong, Huawei Technologies in Russia on June 30, 2015
  - talks about Huawei’s perspective on SDN, NFV, and what they are doing

Huawei's View