CS 490.31: Software Defined Networks

1st Lecture
14/3/2013

Xenofontas Dimitropoulos
ETH Zurich

Learning Objectives

- What is SDN?
- How key SDN technologies work?
- SDN applications
- How to program SDN networks?

Course Schedule

Time	Description
14/03/2013	Introduction to SDN, OpenFlow
21/03/2013	Switches & SDN controllers
28/03/2013	SDN Applications
04/04/2013	Network virtualization
11/04/2013	More on SDN apps (tentative)
18/04/2013	From protocols to abstractions (tentative)

More details on the website: http://www.csd.uoc.gr/~hy490-31

3

Course Project

- Program an SDN application
- Milestones:
  - Select project 29/3/2013
  - Intermediate presentation 18/4/2013
  - Final project report & code 12/5/2013

Course Logistics

- Website: http://www.csd.uoc.gr/~hy490-31
- Mailing list: subscribe to hy490-31-list
- Have questions? hy490-31@csd.uoc.gr
- Teaching assistant: Stelios Frantzeskakis sfrantz@csd.uoc.gr
- Instructor: Xenofontas Dimitropoulos fontas@tik.ee.ethz.ch

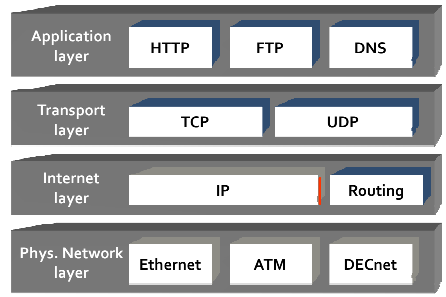
Agenda

Time	Description
9:15 – 9:30	Course Logistics
9:30-10:00	Background on Routing Protocol
10:15 – 11:00	SDN/OpenFlow Introduction
11:15-12:00	Hands on: Learn Development Tools (Part 4 of OpenFlow Tutorial)

6

Quick Recap of Internet Routing Architecture

IP Protocol Stack



8

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.

9

Routing algorithm

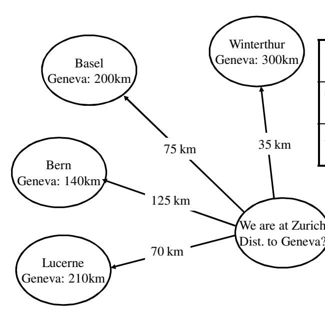
- A *distributed algorithm* executed among the routers which builds the routing tables. Path selection can be based on different metrics:
  - Quantative: #hops, bandwidth, available capacity, delay, delay jitter,…
  - Others: Policy, utilization, revenue maximization, politics,…
- Design and evaluation 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.

10

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)

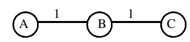
12

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.*

13

Count-to-infinity Problem



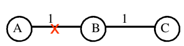
Node A		
Destination	Distance	Next node
B	1	B
C	2	B

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

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

14

Count-to-infinity Problem



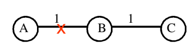
Node A		
Destination	Distance	Next node
B	1	B
C	2	B

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

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

15

Count-to-infinity Problem



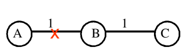
Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

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

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

16

Count-to-infinity Problem



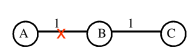
Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

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

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

17

Count-to-infinity Problem



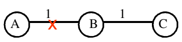
Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

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

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

18

Count-to-infinity Problem



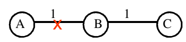
Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

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

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

19

Count-to-infinity Problem



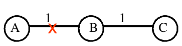
Node A		
Destination	Distance	Next node
B	N.E.	
C	N.E.	

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

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

20

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]

21

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

22

Link State Routing

23

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

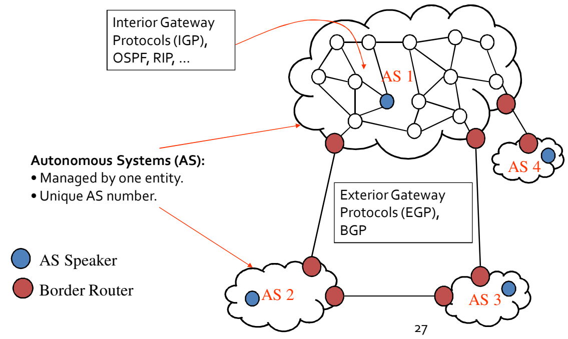
24

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)

26

Hierarchical routing domains, AS



27

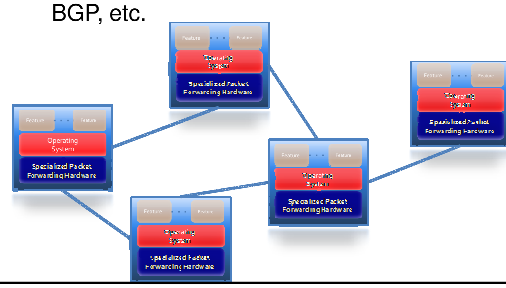
Internet intra-domain routing protocols

- 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

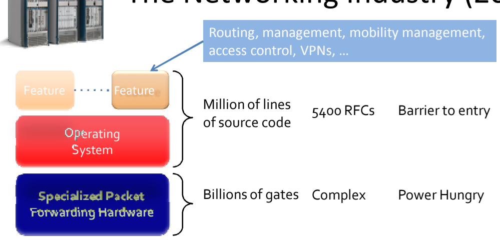
28

Classical network architecture

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



The Networking Industry (2007)



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

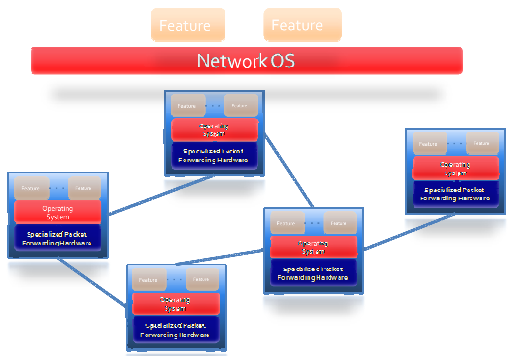
Functionality defined by standards, put in hardware, deployed on nodes

30

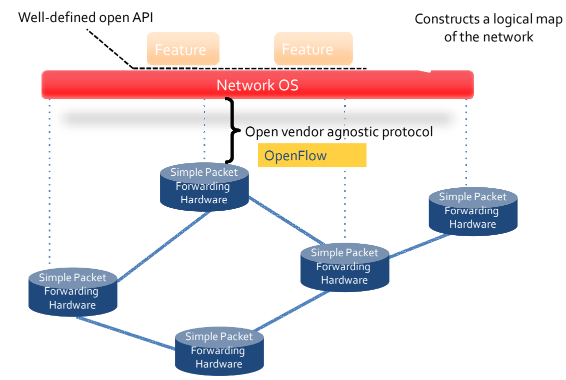
SDN

- 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:

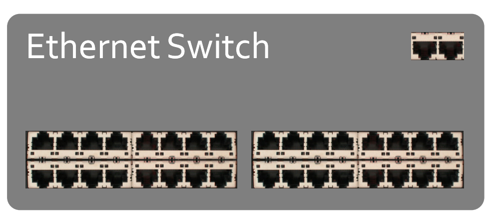
- Get state information **from** forwarding elements
- Give control directives **to** forwarding elements

OpenFlow

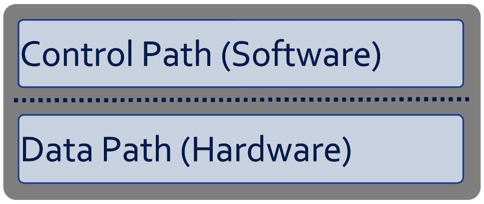
- OpenFlow
  - is a protocol for remotely controlling the forwarding table of a switch or router
  - is one element of SDN

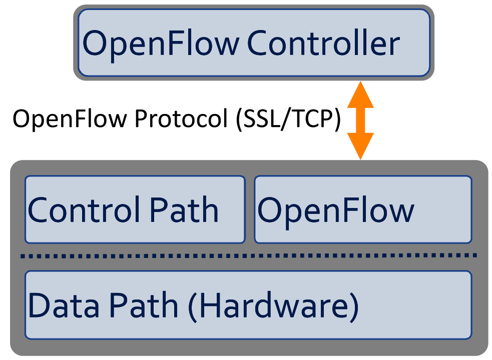
How does OpenFlow work?

36

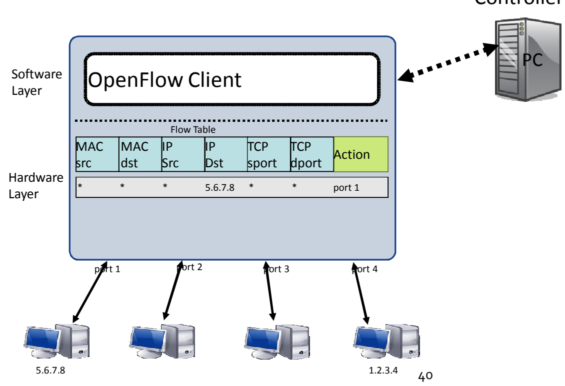


37



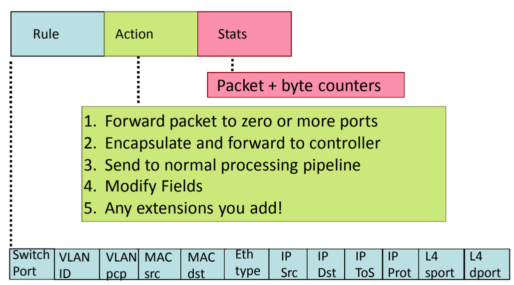


OpenFlow Example



OpenFlow Basics

Flow Table Entries



+ 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

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.

46

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]

47

SDN App example

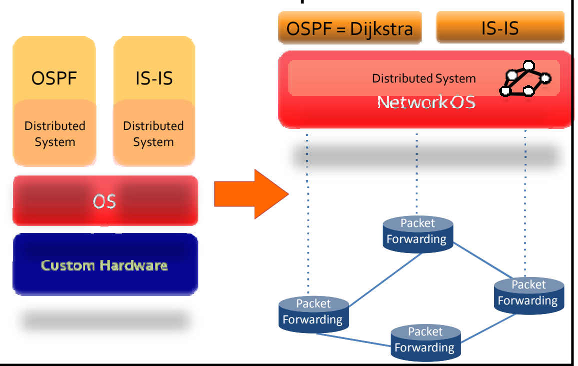
OSPF and Dijkstra

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

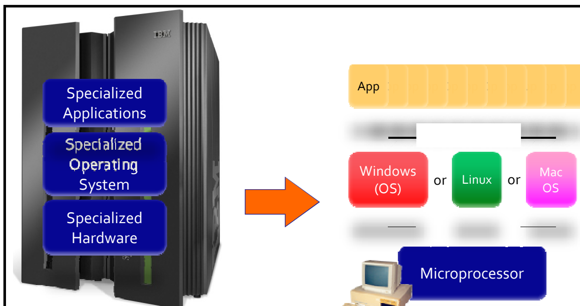


Other SDN Use Cases

- 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 …

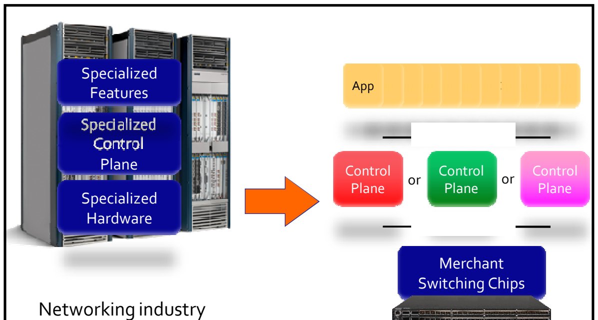
See http://www.openflow.org/videos/

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

[Hands-on Tutorial]

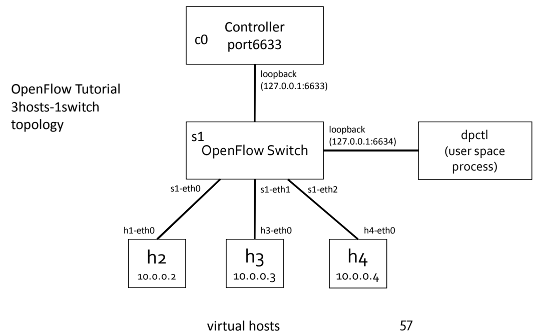
http://www.openflow.org/wk/index.php/OpenFlow_Tutorial

55

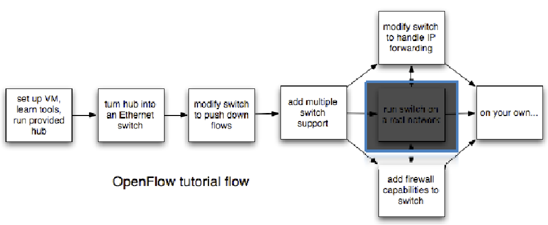
Hands-on Tutorial

- This lecture:
  - Will do part 4 of tutorial
- Next lecture:
  - Bring your laptop
  - Install virtual machine (parts 1-3 of tutorial) before coming to the lecture

Tutorial Setup



TutorialFlow



58

This talk wouldn't be possible without:

- Past slides from:
  - Brandon Heller
  - Nick McKeown
  - Guru Parulkar
  - Scott Shenker

Further reading

- http://www.openflow.org/wk/index.php/OpenFlow_Tutorial
- http://www.openflow.org/videos/
- www.csd.uoc.gr/~hy490-31/links.html