Software Defined Networking and OpenFlow

Jeffrey Dalla Tezza and Nate Schloss

Agenda

- What is SDN
- SDN Today
- What is OpenFlow
- Why OpenFlow
- What’s next for SDN
- Our OpenFlow Demonstration

Software Defined Networking

- Wikipedia defines it with three characteristics:
  - An approach to building computer networks that **separates and abstracts elements of these systems**.
  - Allows system administrators to **quickly provision network connections on the fly** instead of manually configuring policies.
  - Allows network administrators to have **programmable central control of network traffic**.

Software Defined Networking

- Those are the goals for SDN, but it is really just programmable control of networking devices
- Current models
  - JunOS by Juniper
  - IOS by Cisco
  - Application Fluent Network by Alcatel-Lucent
  - OpenFlow
- All of those give some degree of programmatic control, but they all have trade offs

Current SDN Technologies

- IOS by Cisco and JunOS by Juniper

✗ **Separates and abstracts elements of networking systems**

✗ **Allows system administrators to quickly provision network connections on the fly**

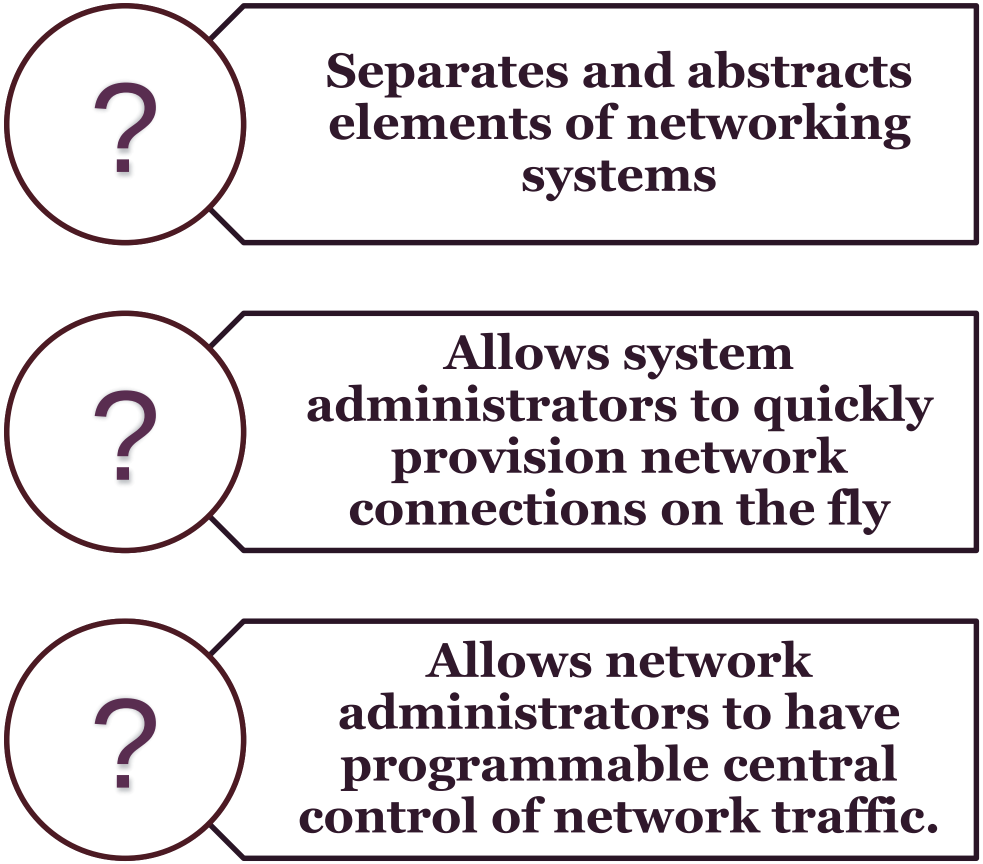
✓ **Allows network administrators to have programmable ~~central~~ control of network traffic.**

Current SDN Technologies

- Other companies also have proprietary solutions
  - Application Fluent Network by Alcatel-Lucent
  - Linerate systems
- Since these all require specific hardware large scale adoption is unlikely

Current SDN Technologies

- OpenFlow by the Open Networking Alliance



Separates and abstracts elements of networking systems

Allows system administrators to quickly provision network connections on the fly

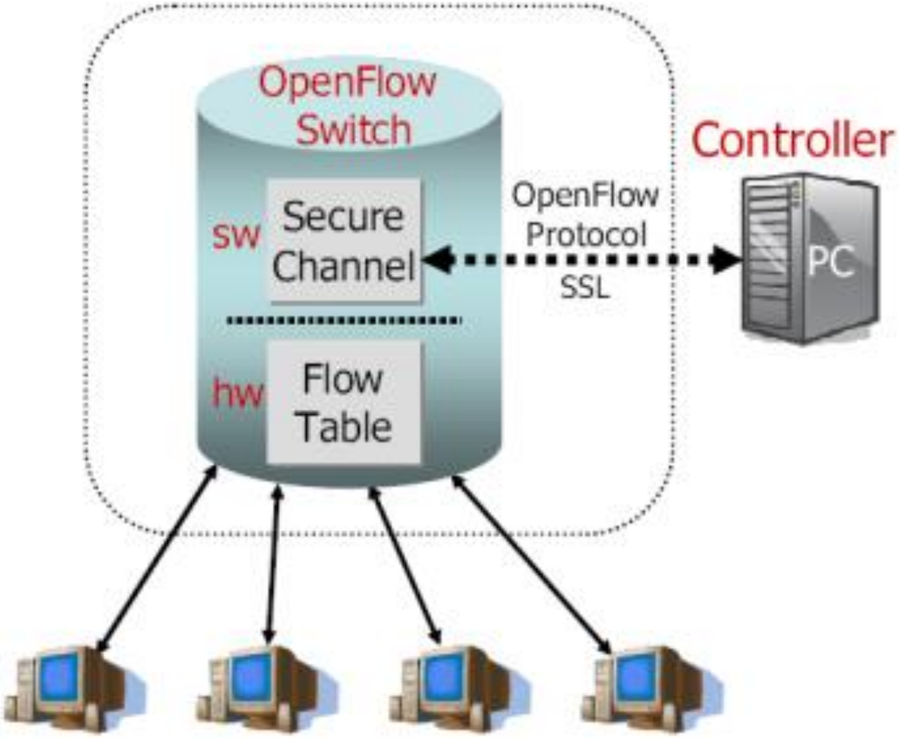
Allows network administrators to have programmable central control of network traffic.

What is OpenFlow

- OpenFlow is an open specification by the Open Networking Foundation for connecting to and controlling routers and switches
- Basic Capabilities:
  - Define and query the routing table
  - Intercept and modify packets
  - Query routers and switches for statistics about the network

OpenFlow

- The basic architecture



Flow Table

- Each switch maintains a **Flow Table**[3]
- Flow tables contain entries (flows) of the form: <Header Fields | Counters | Actions>
  - Packets are **matched** against header fields
  - Counters are then **updated** based on the matching packet
  - Actions are then **applied** to packets

Matching

- Matching Fields
  - Ingress Port
  - Ethernet source/destination address
  - Ethernet type
  - VLAN id/priority
  - IP source/destination address
  - IP protocol/ToS
  - Transport source/destination port
- Fields can be partially matched (e.g. IP subnets) or wild carded

Counters

- If a packet matches a flow entry it can update the relevant counters.
- Counters can be maintained:
  - Per table
  - Per flow
  - Per queue
- Counters can track:
  - Received packets
  - Received bytes
  - Duration
  - Transmitted Packets
  - Transmit/Receive errors
  - Etc..

Actions

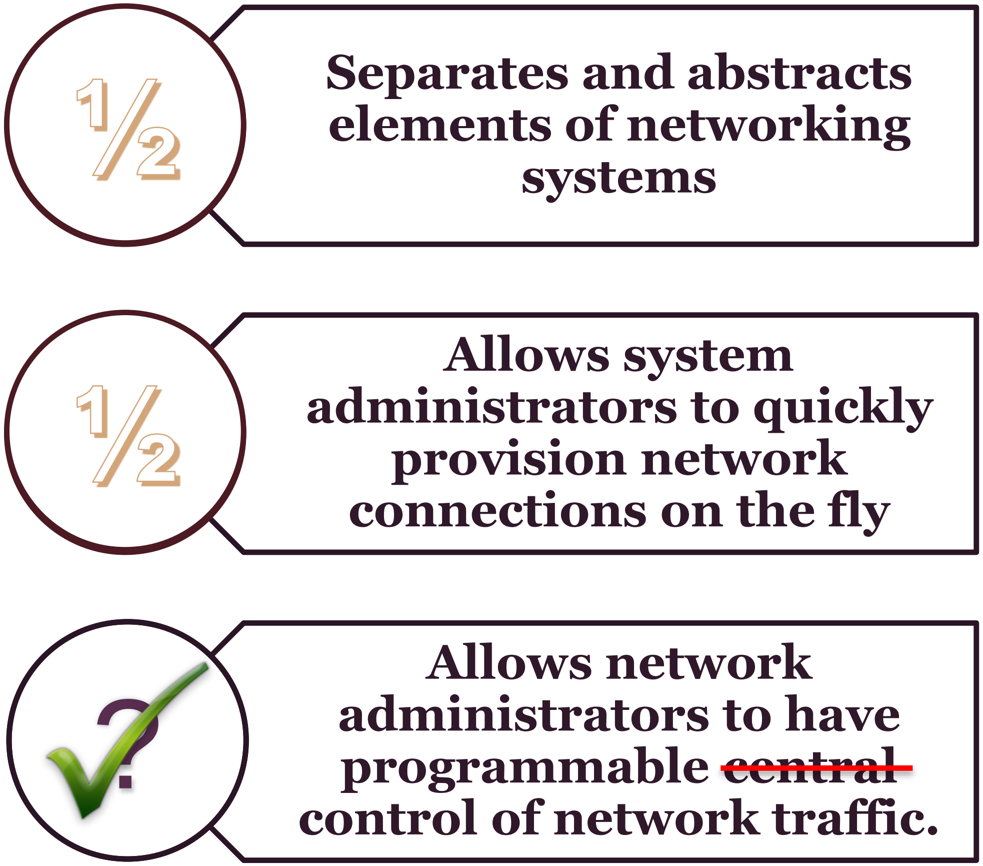
- After matching a packet the switch can apply the following actions:
  - Forward out of a port(s)
  - Encapsulate and send to controller
  - Drop packets
  - Modify packet headers

Controller

- The controller is connected to the switch, through the OpenFlow communication protocol it can query and modify counters and the flow table
- It can also receive packets from the data plane
- The controller can be any arbitrary program that uses the OpenFlow protocol
  - NOX/POX
  - Beacon
  - Floodlight
  - Maestro
  - Ryu
  - and others

OpenFlow

- Does OpenFlow by itself give us SDN?



½ **Separates and abstracts elements of networking systems**

½ **Allows system administrators to quickly provision network connections on the fly**

✓? **Allows network administrators to have programmable ~~central~~ control of network traffic.**

- But why would it?
- You don’t expect x86 to have merge sort right out of the box

So, what's the point?

- OpenFlow gives unified specifications
  - Any hardware vendor can support it
  - Any 3rd party software vendor can write software for it
- IOS, JunOS and Application Fluent Network provide hardware specific solutions
- As OpenFlow adoption increases the incentive for vendors to support OpenFlow increases

People using OpenFlow

- OpenFlow is currently used in all of Google's data centers[1]
- Almost all the big names are members of the Open Networking Foundation[2]
  - Google
  - Facebook
  - Verizon
  - Cisco
  - Samsung
  - Broadcom
  - etc...

How do we Realize SDN Using OpenFlow?

- OpenFlow provides network control, but at a low level
- We still need proper abstractions and centralized control

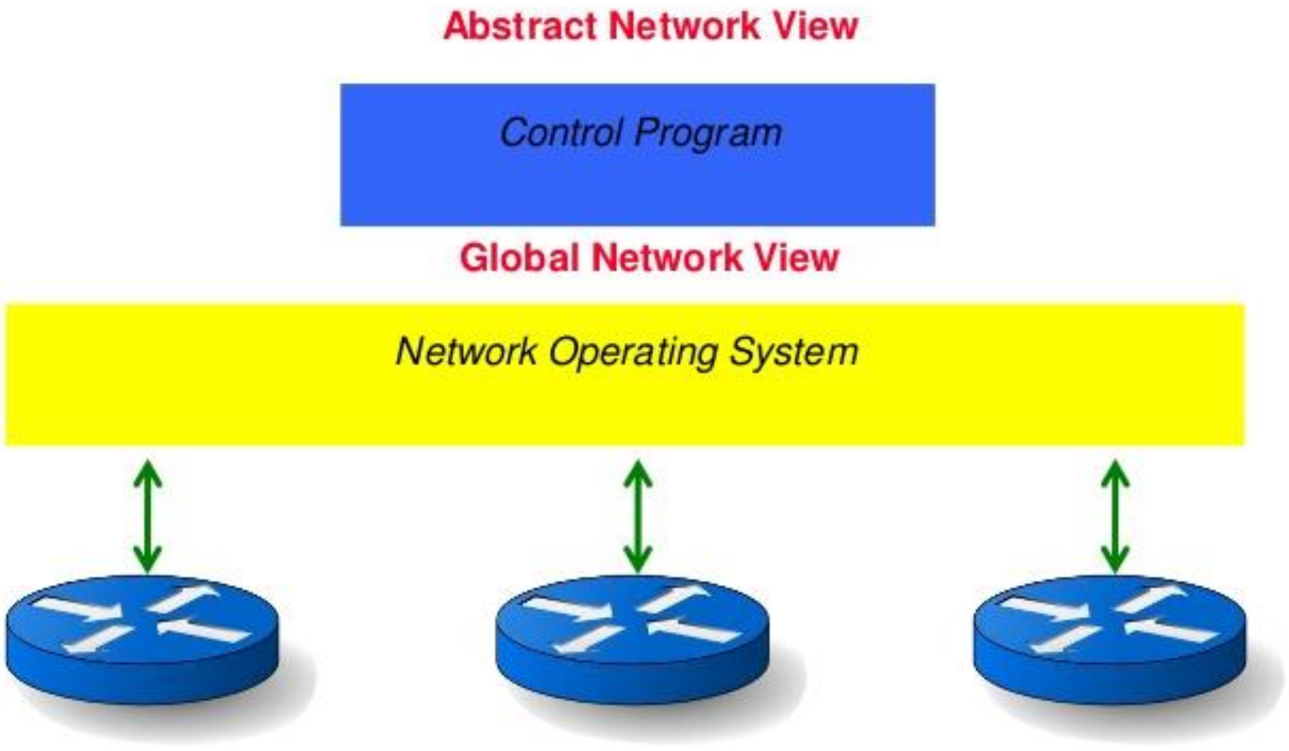
Centralized Control

- Real systems are globally distributed
- Large systems have to account for failure
- We need one logical controller with a global view of the network
- This requires coordination between physical servers distributed geographically

Proper Abstractions

- Once we have a distributed controller we need the ability to install the configuration on the network
- Control policies should be specified at a high level, they should not be dependent on the state of the network

Proper Abstractions



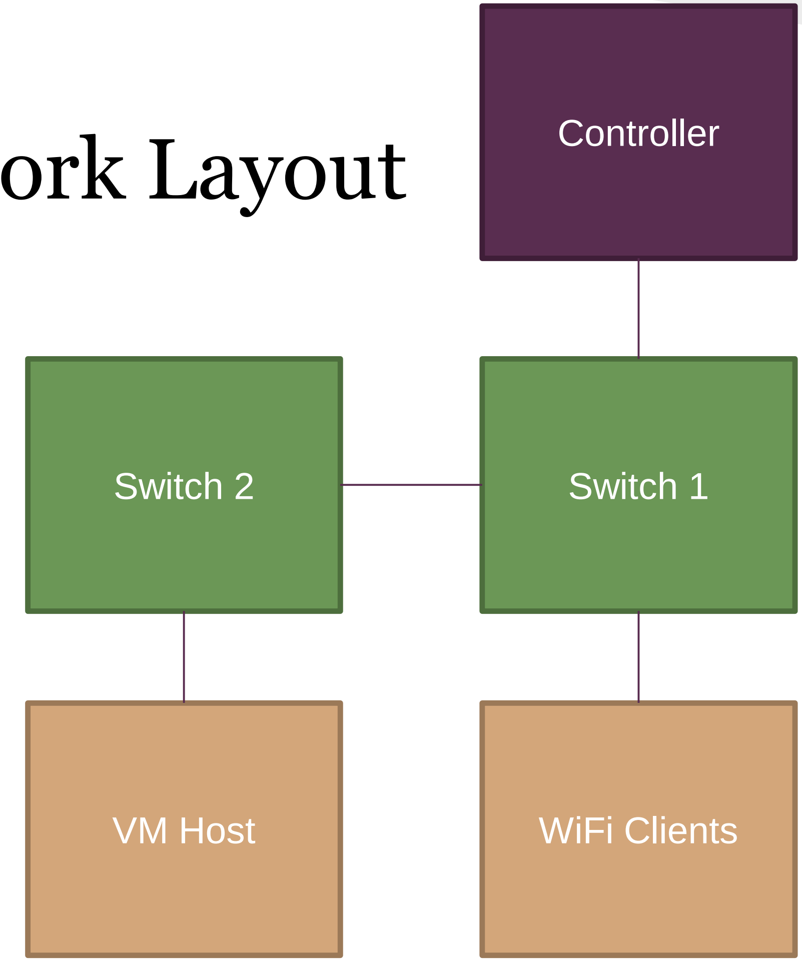
Source: Scott Shenker

Demonstration

- Load balancing using OpenFlow
- Clients are directed to different webservers by the controller
- A program connected to the controller can specify access control for specific IPs

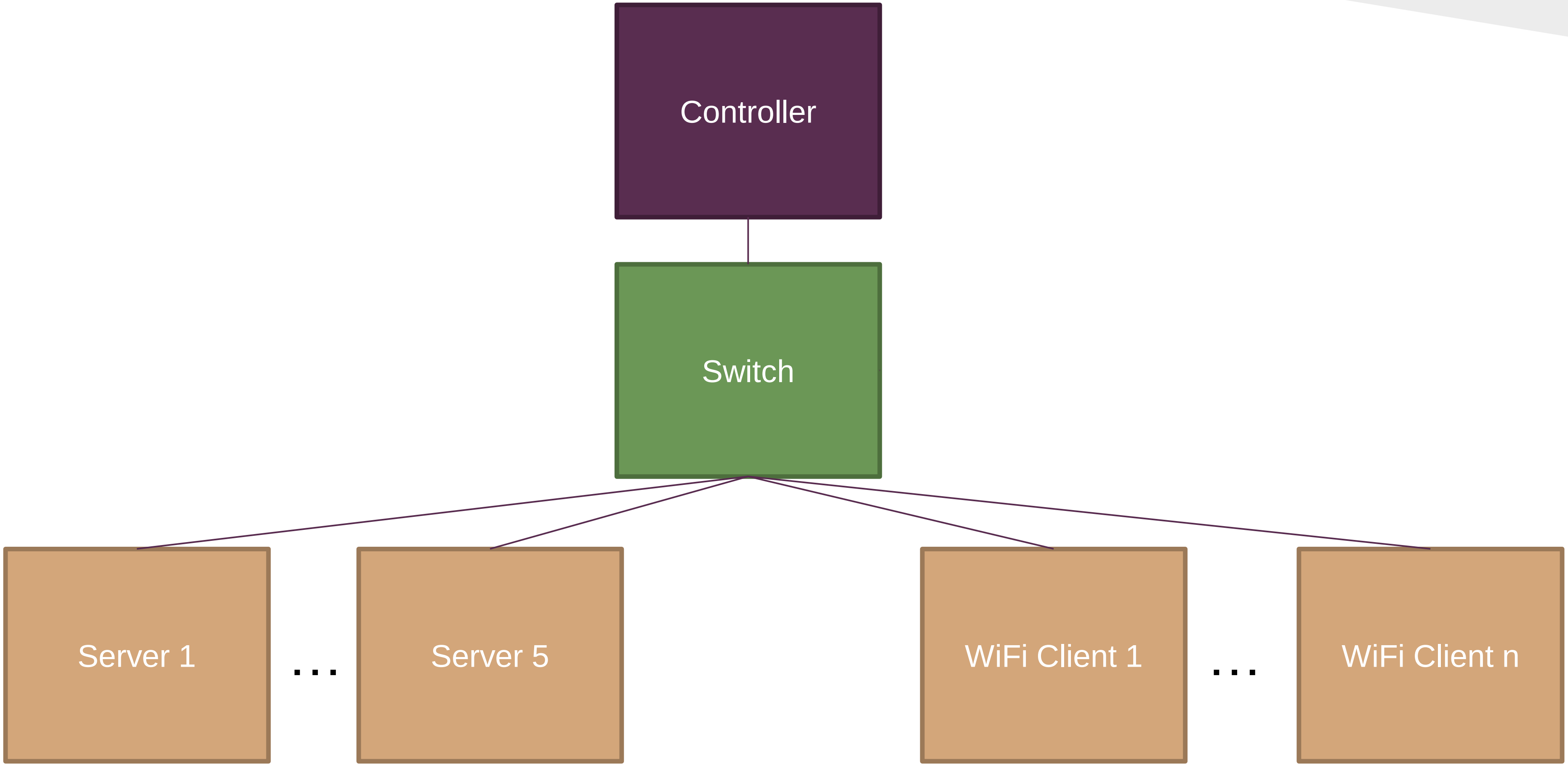
Demonstration

Physical Network Layout



Demonstration

Abstract Network Layout



Demonstration

- Get out your laptop/phone and connect to the open network named ‘**OpenFlow**’
- In your web browser go to **http://192.168.0.1/**
- To be reassigned to a new server, wait 15 seconds and refresh

Bibliography

- 1: http://searchsdn.techtarget.com/news/2240181909/Vint-Cerf-At-Google-OpenFlow-now-runs-in-all-data-center-networks
- 2: https://www.opennetworking.org/membership/member-listing
- 3: http://www.openflow.org