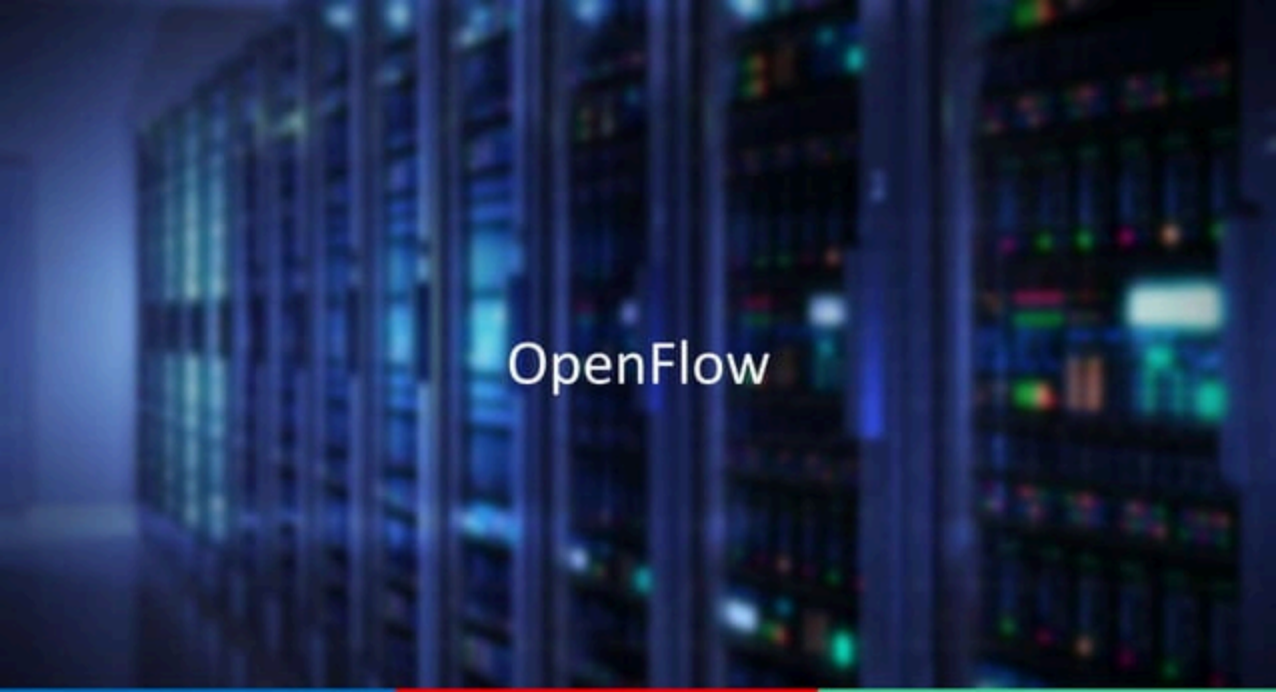
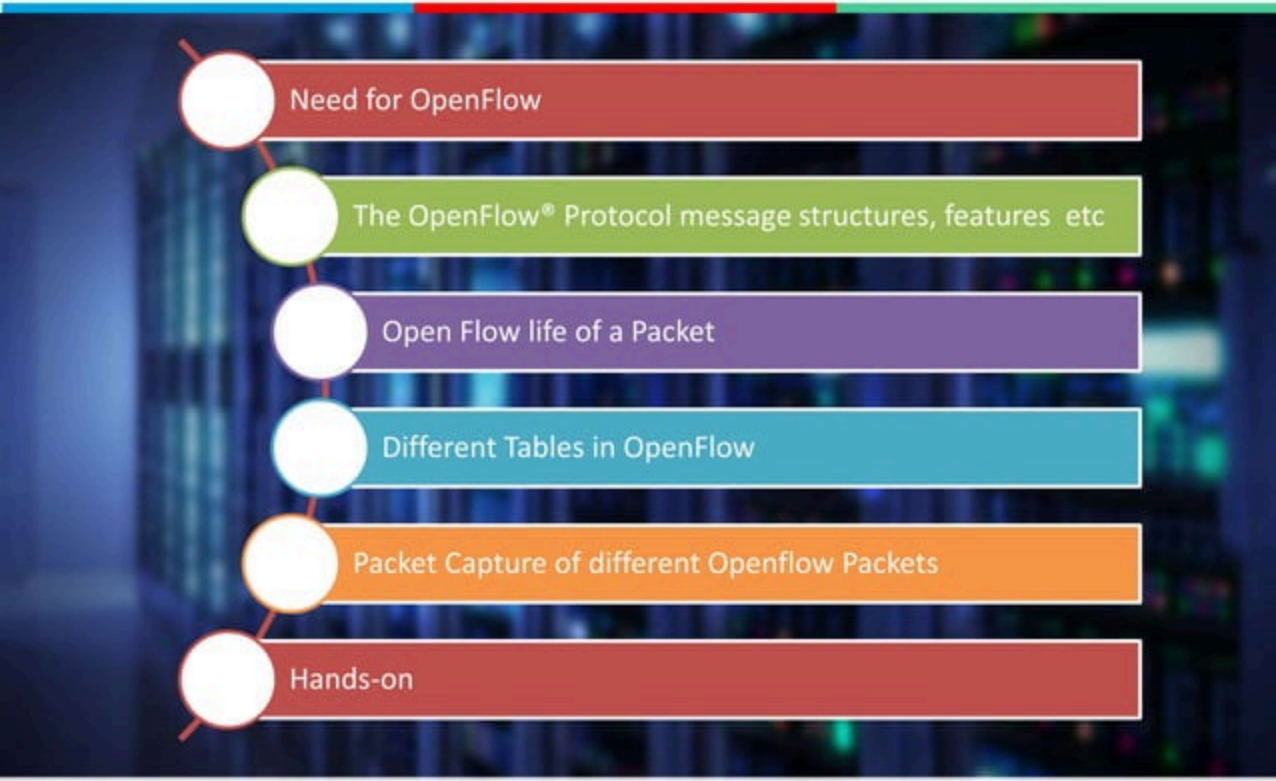


S. Kingston Smiler
kingstonsmiler@gmail.com

Course Objective



Problems

- Closed Systems with no or very minimal abstractions in the network design.
- Hardware centric – usage of custom ASICs with Vendor Specific Software.
- Difficult to perform real world experiments on large scale production networks.
- No standard abstractions towards north bound and south bound interfaces, even though we have standard abstractions in the east / west bound interface with peer routers / switches.

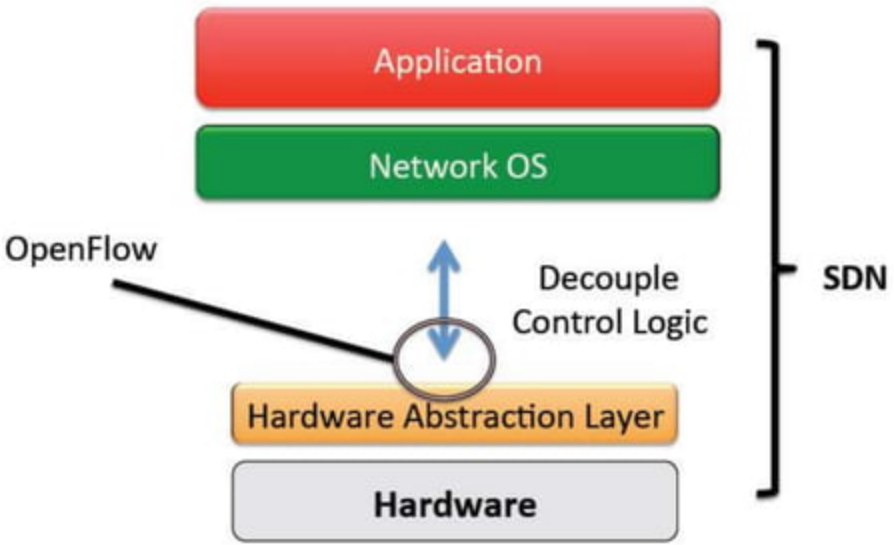
Need for OpenFlow

- Facilitate Innovation in Network
- Layered architecture with Standard Open Interfaces
- Independent innovation at each layer
- More accessibility since software can be easily developed by more vendors
- Speed-to-market – no hardware fabrication cycles
- More flexibility with programmability and ease of customization and integration with other software applications
- Fast upgrades
- Program a network vs Configure a network

OpenFlow

- OpenFlow is a protocol which enables programmability of the forwarding plane
- OpenFlow is like an x86 instruction set for the network nodes.
- Provides open interface to “black box” networking node (ie. Routers, L2/L3 switch) to enable visibility and openness in network
- Separation of control plane and data plane.
  - The datapath of an OpenFlow Switch consists of a Flow Table, and an action associated with each flow entry
  - The control path consists of a controller which programs the flow entry in the flow table

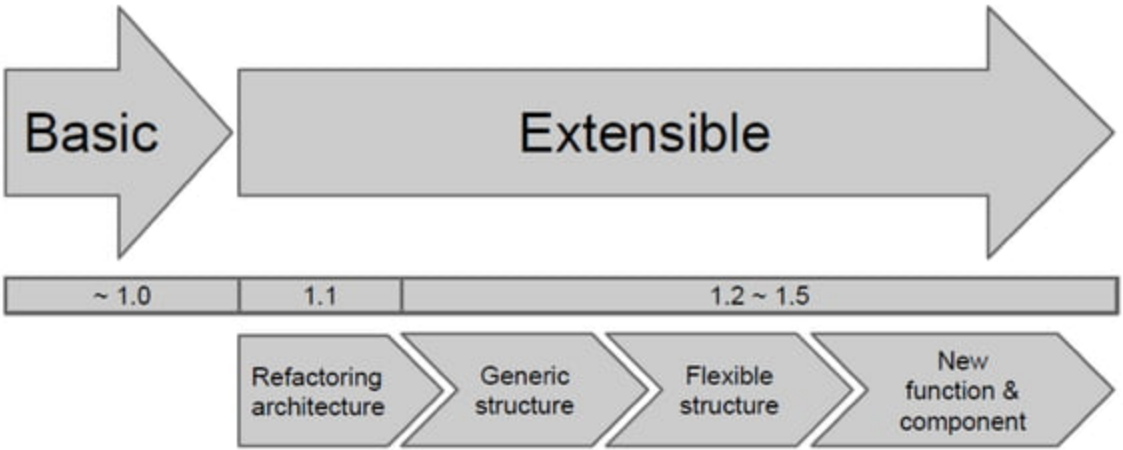
SDN and Open Flow



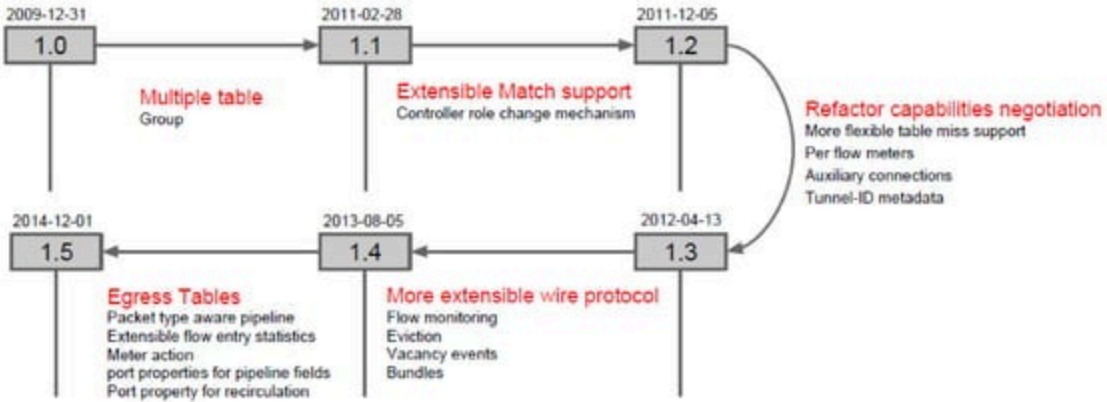
OpenFlow

- General Myth
  - SDN is Open Flow
- Reality
  - OpenFlow is an open API that provides a standard interface for programming the data plane switches

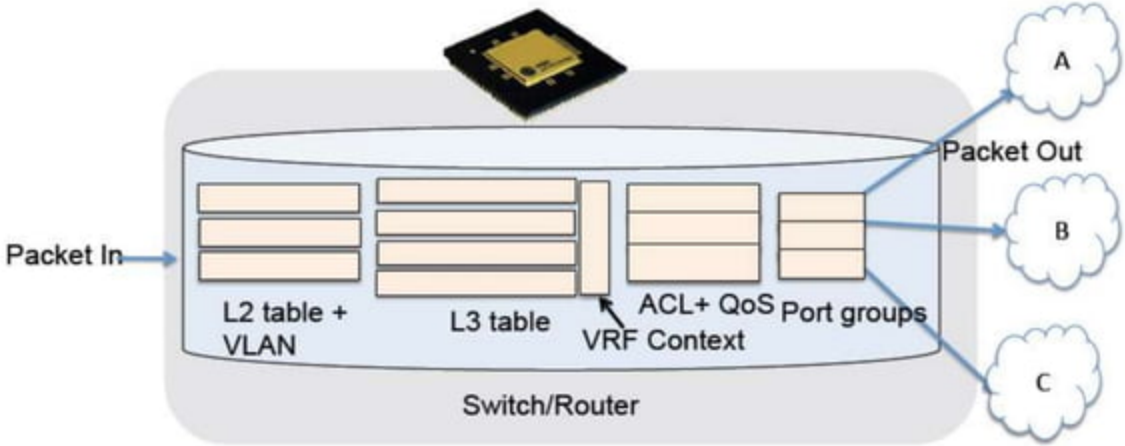
OpenFlow Specification History



OpenFlow Specification History



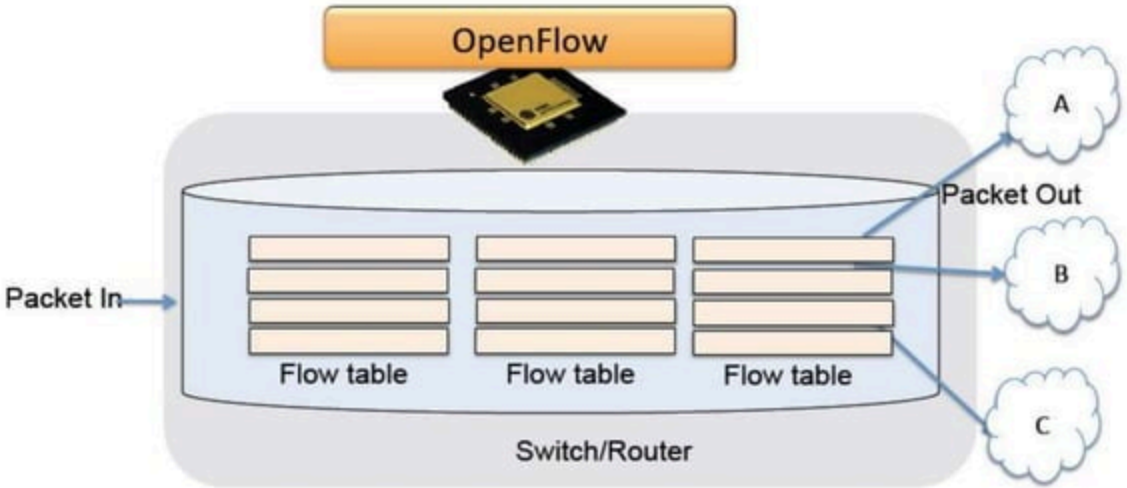
Traditional Switch Forwarding



- Fixed function
- Often expose implementation details
- Non-standard/non-existent state management APIs

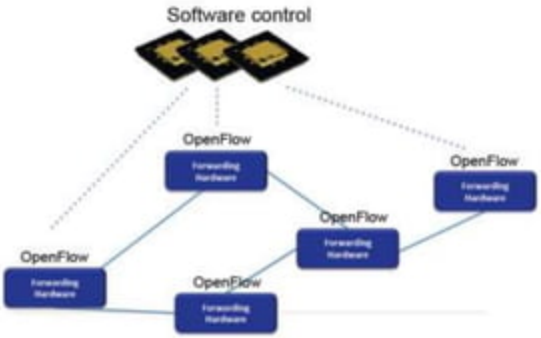
Source: Adopted from ONF11 presentation by Martin Casado

OpenFlow Switch Forwarding



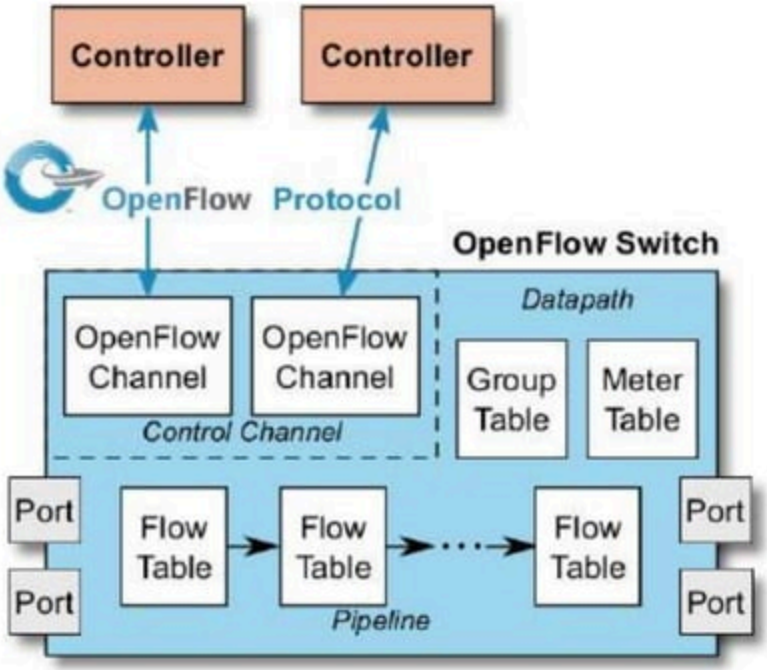
Source: Adopted from ONF11 presentation by Martin Casado

Open Flow Illustration



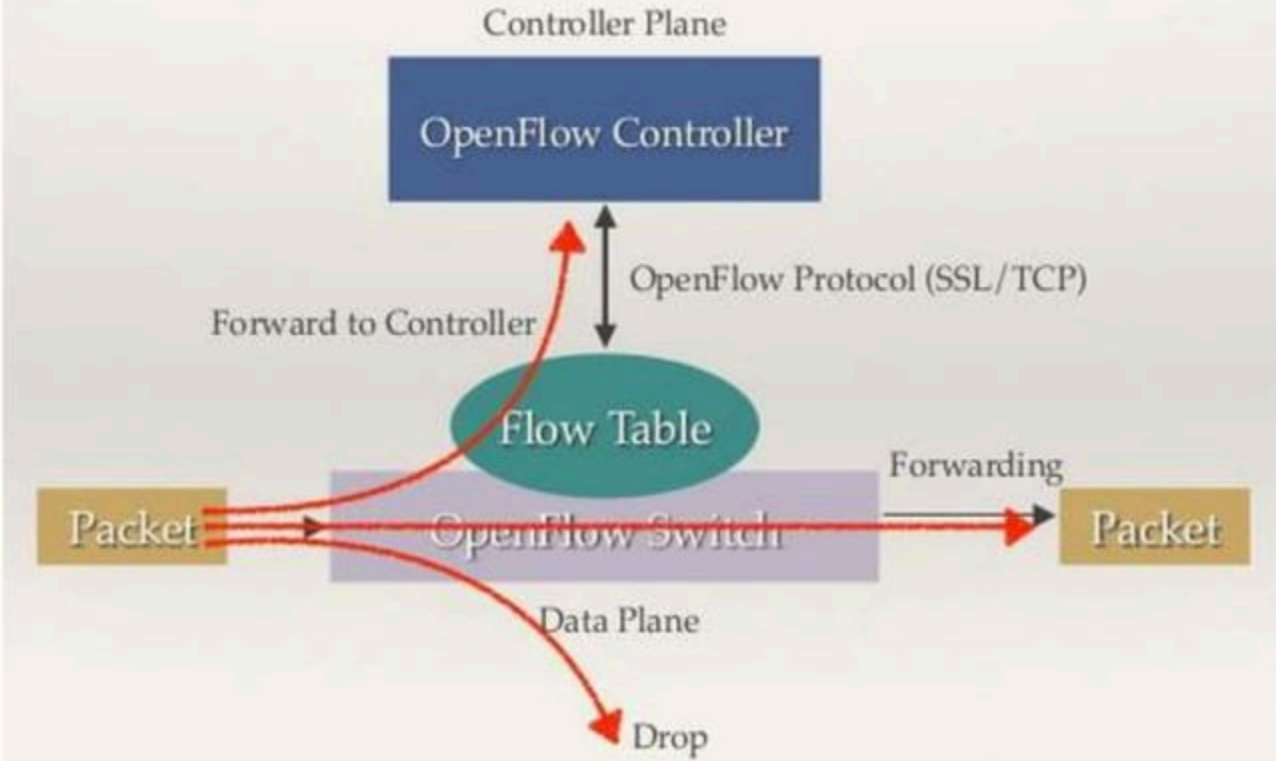
Source: Adopted from ONF11 presentation by Martin Casado

Components of OpenFlow

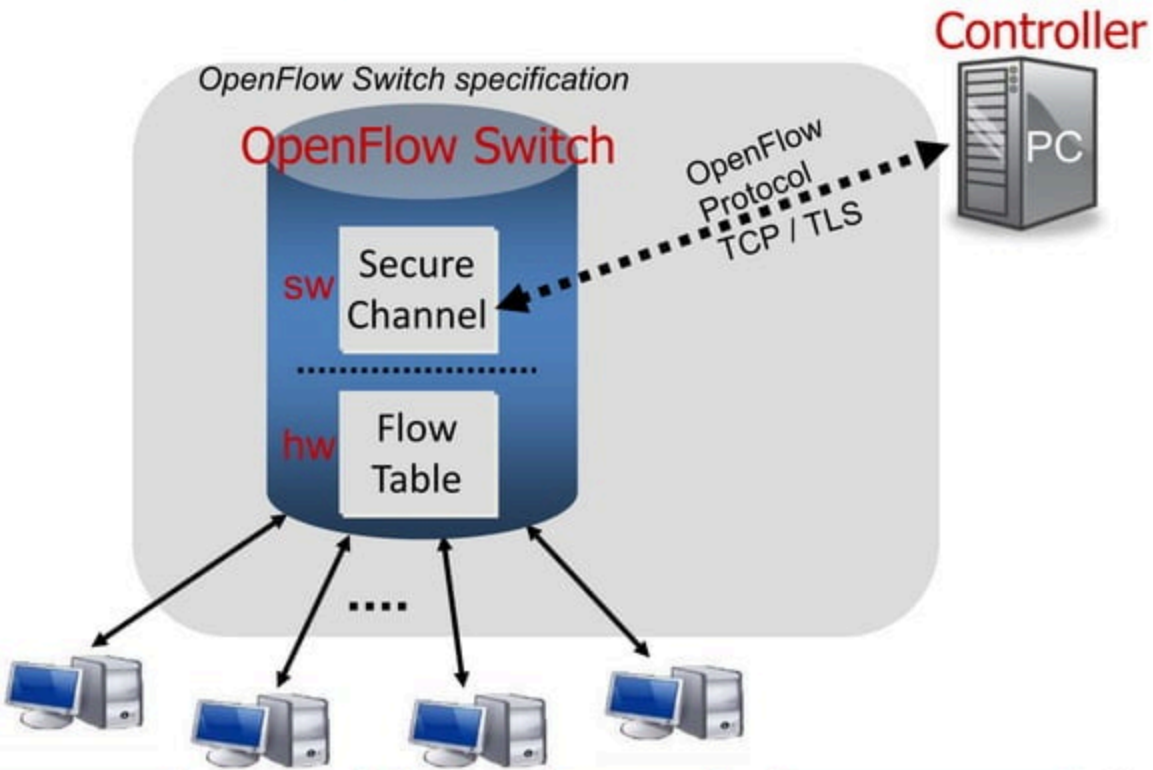


* Figure From OpenFlow Switch Specification

OpenFlow Packet Flow



Components of OpenFlow Network



* Figure From OpenFlow Switch Specification

OpenFlow Controller

- Manages one or more switch via OpenFlow channels.
- Uses OpenFlow protocol to communicate with a OpenFlow aware switch.
- Acts similar to control plane of traditional switch.
- Provides a network wide abstraction for the applications on north bound.
- Responsible for programming various tables in the OpenFlow Switch.
- Single switch can be managed by more than one controller for load balancing or redundancy purpose. In this case the controller can take any one of the following roles.
  - Master.
  - Slave.
  - Equal.

OpenFlow Controller

- OpenSource
  - OpenDayLight
  - Floodlight
  - RYU
  - NOX/POX
  - ONOS

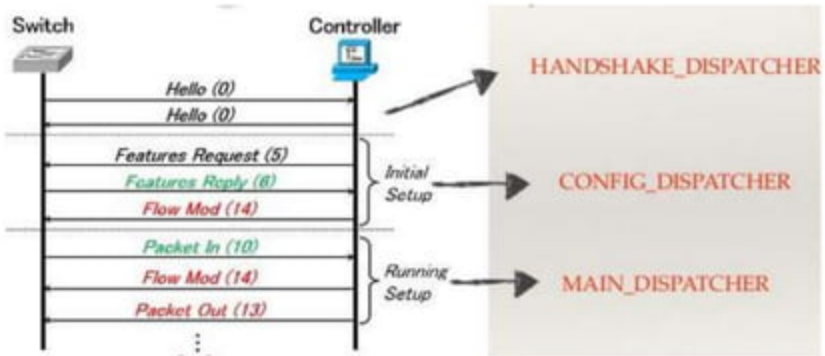
- Commercial Controllers
  - Cisco APIC
  - VMware NSX Controller
  - HP VAN SDN Controller
  - NEC ProgrammableFlow PF6800 Controller
  - Nuage Networks Virtualized Services Controller (VSC)

OpenFlow Channel

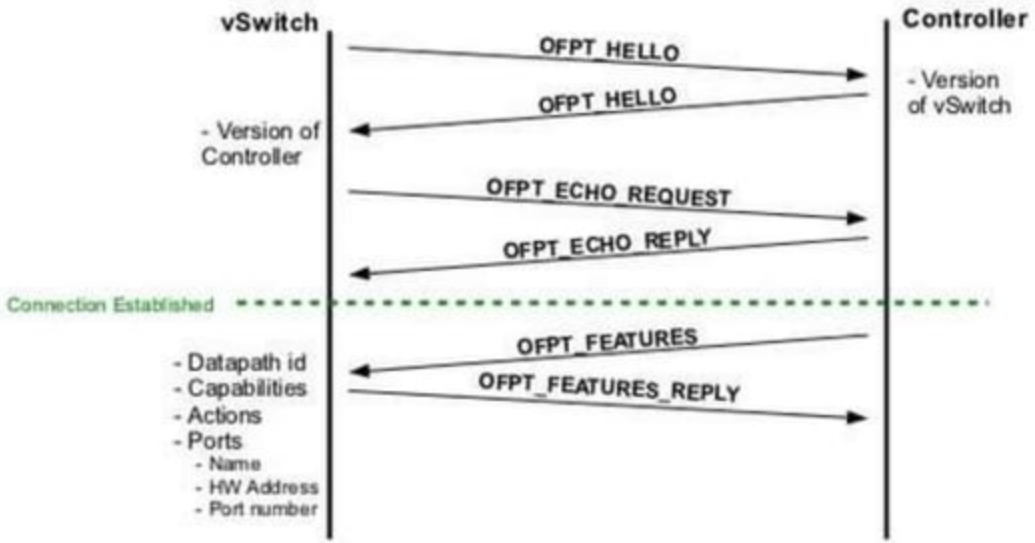
- Used to exchange OpenFlow message between switch and controller.
- Switch can establish single or multiple connections to same or different controllers (auxiliary connections).
- A controller configures and manages the switch, receives events from the switch, and send packets out the switch via this interface
- The SC connection is a TLS/TCP connection. Switch and controller mutually authenticate by exchanging certificates signed by a site-specific private key

OpenFlow Messages

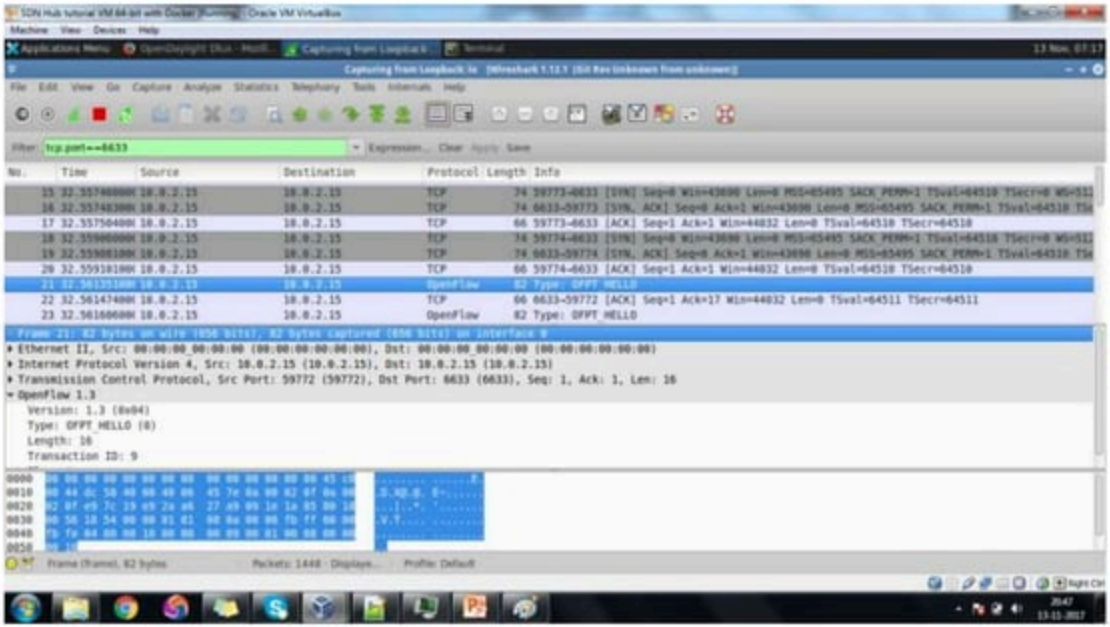
- Three Message Types
  - Controller to Switch
  - Asynchronous Message
  - Asymmetric Message
- Controller to Switch
  - Feature Request
  - Configuration
  - Modify State
  - Packet Out etc.
- Asynchronous
  - Packet-in
  - Flow Removed
  - Port Status
  - Error etc.
- ASymmetric
  - Hello
  - Echo
  - Experimenter etc



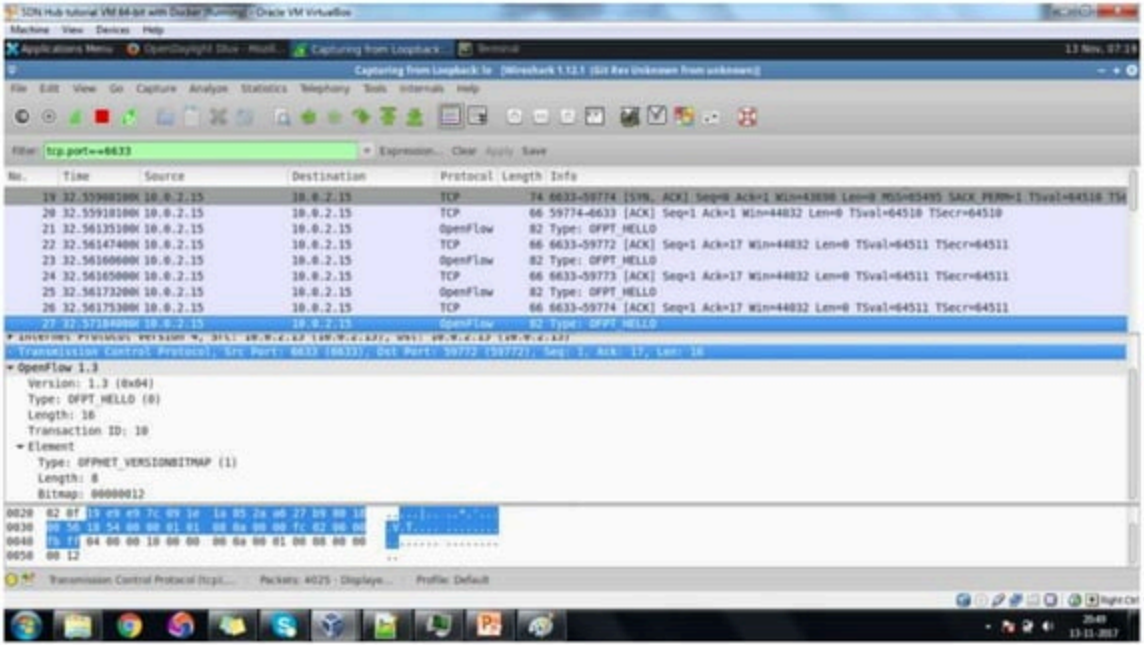
Initial Connection Setup



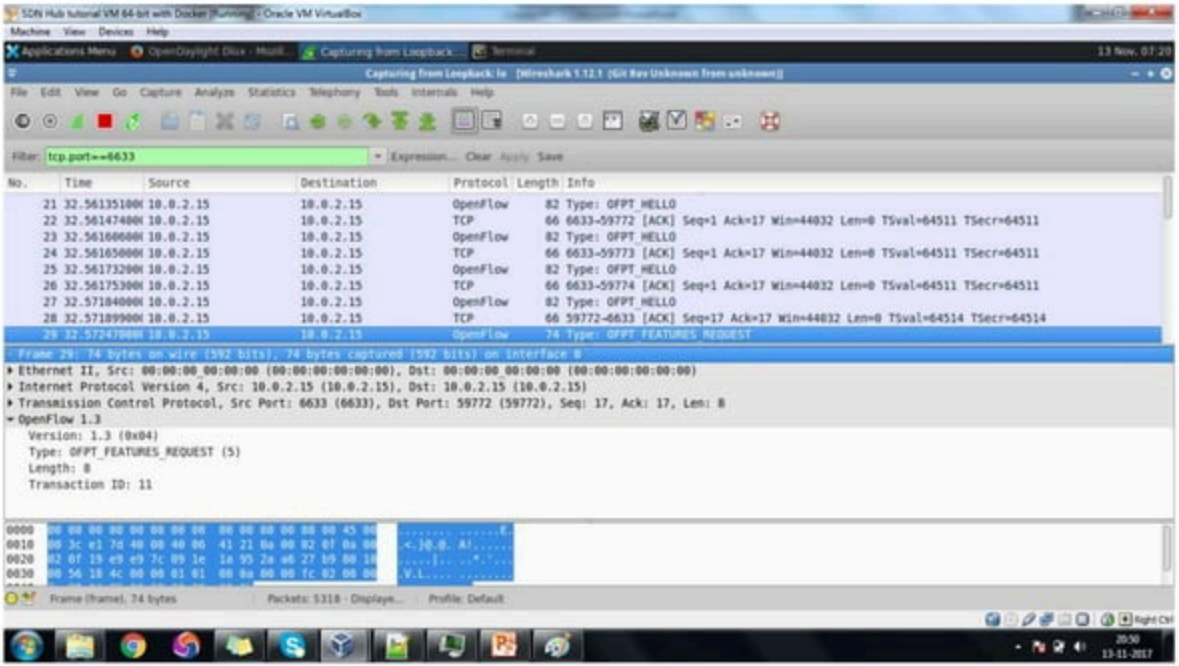
OpenFlow – Hello (From Switch to Controller)



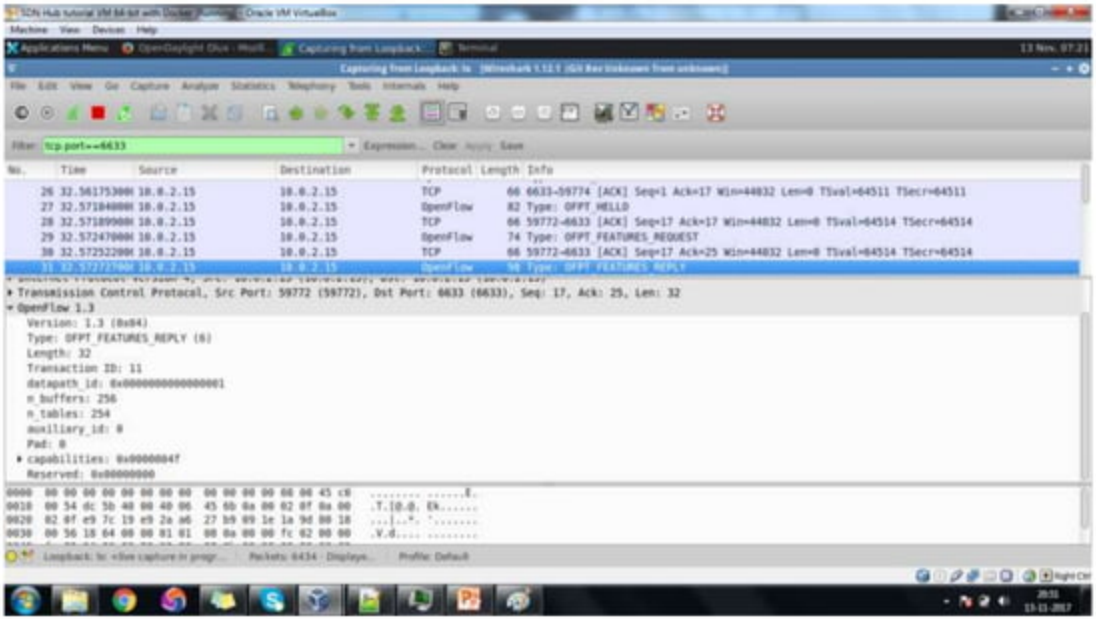
OpenFlow – Hello (From Controller to Switch)



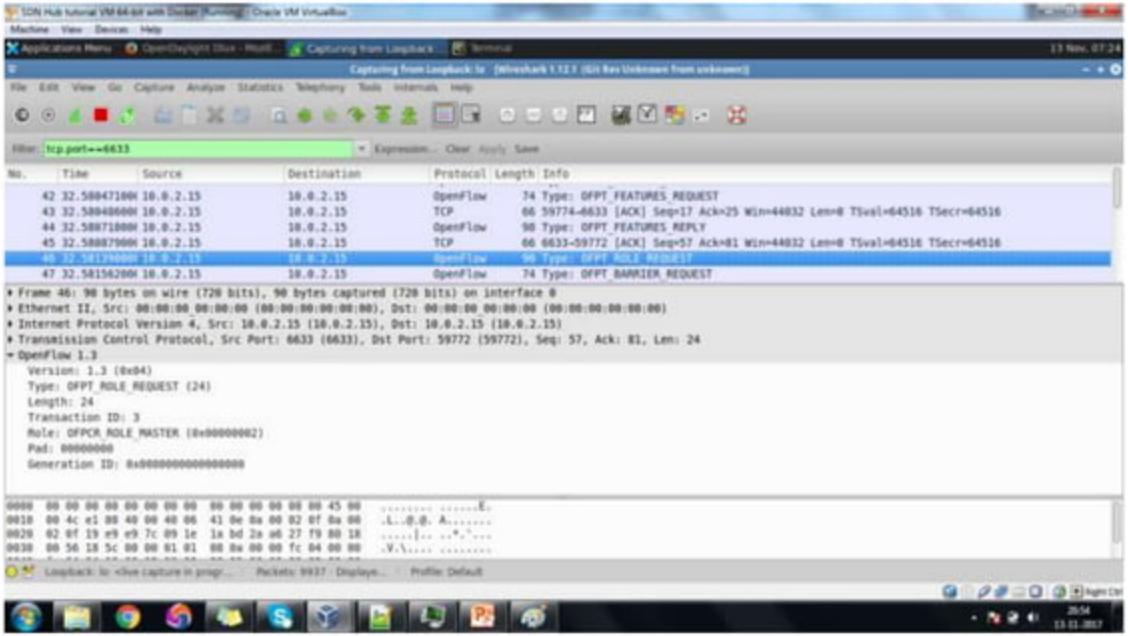
OpenFlow – Feature Request (From Controller to Switch)



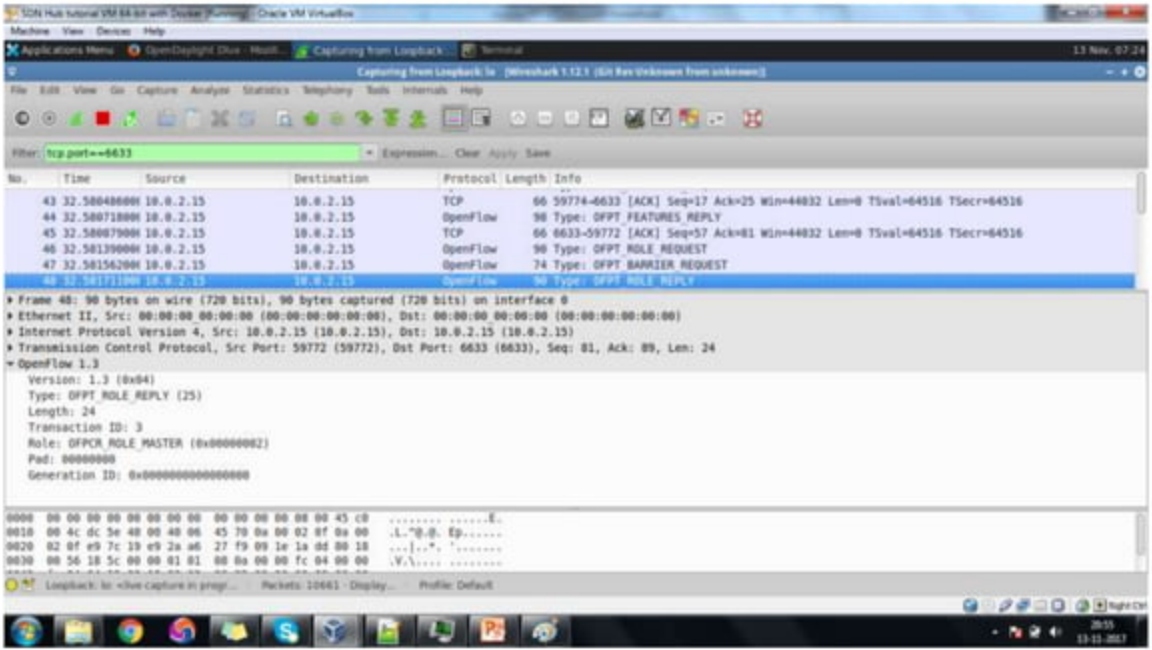
OpenFlow – Feature Reply (From Switch to Controller)



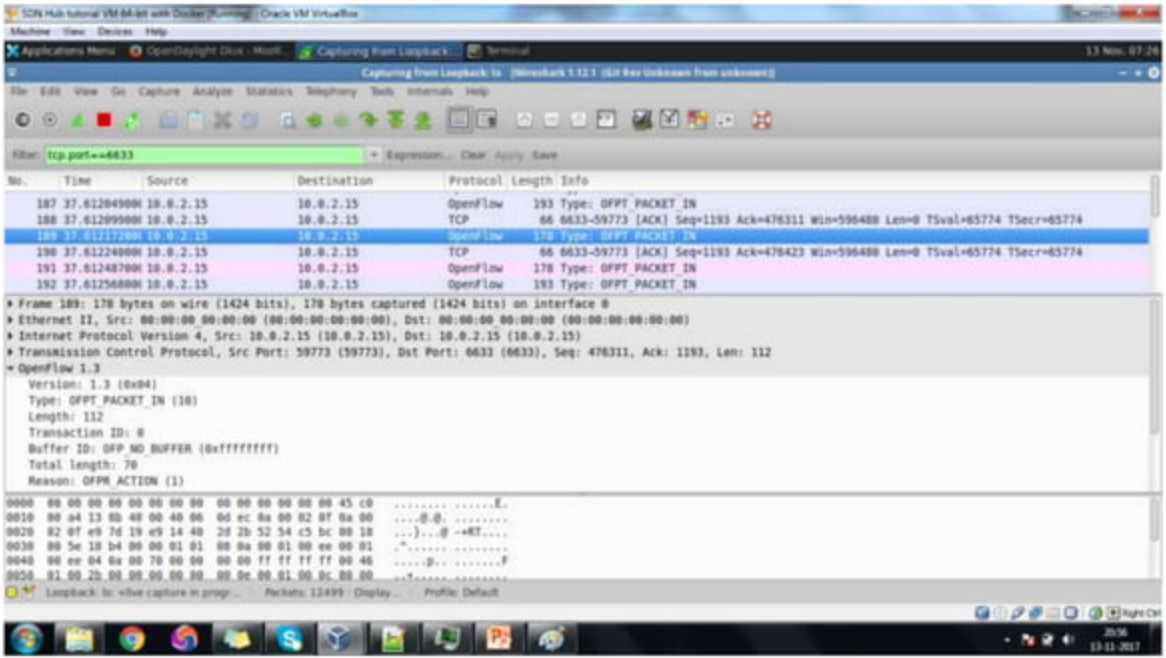
OpenFlow – Role Request (From Controller to Switch)



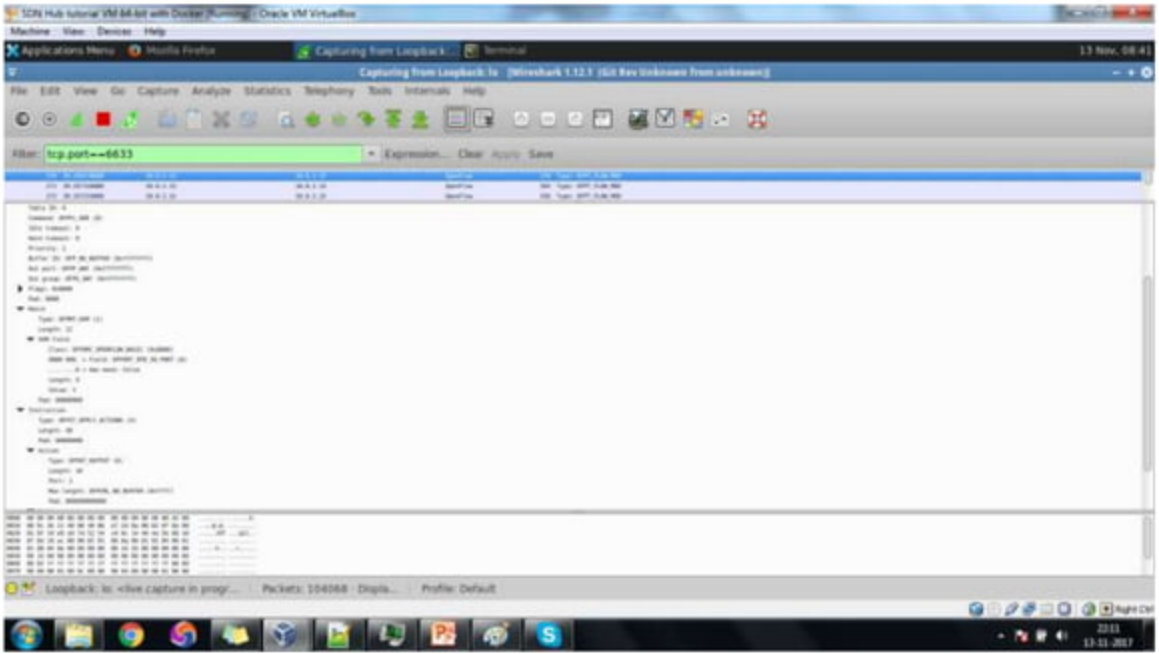
OpenFlow – Role Reply (From Switch to Controller)



OpenFlow – Packet IN (From Switch to Controller)



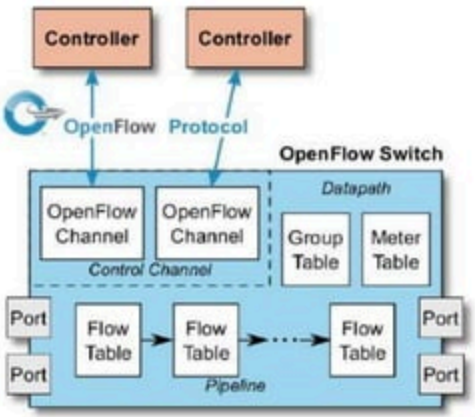
OpenFlow – Flow Config (From Switch to Controller)



OpenFlow Switch

- Types of the switches:
  - Open-flow only
  - Open-flow hybrid

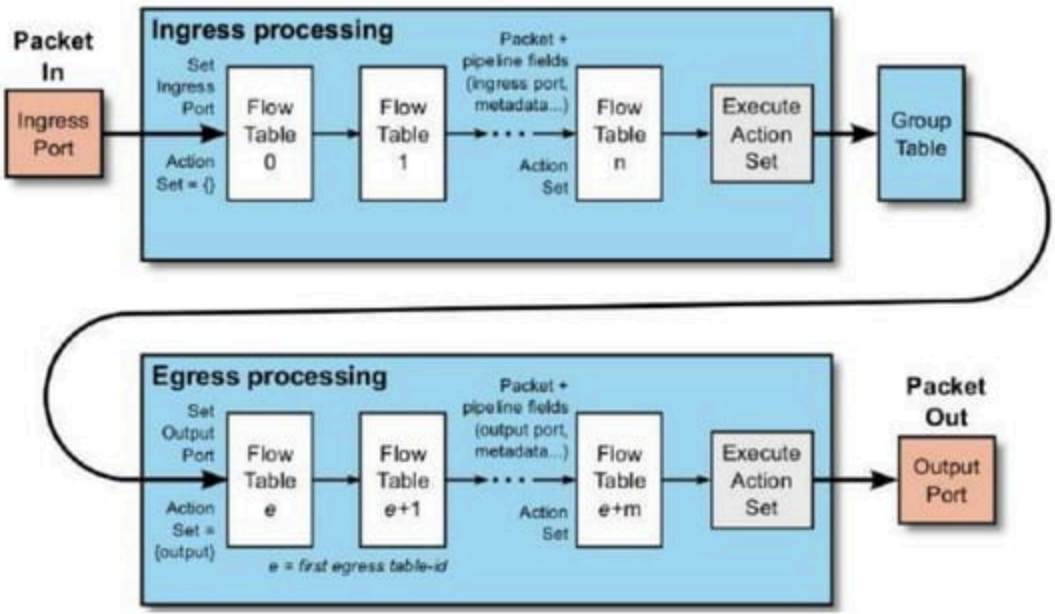
- Major Components
  - Ports
  - Flow Table
  - Group Table
  - Meter Table



OpenFlow Ports

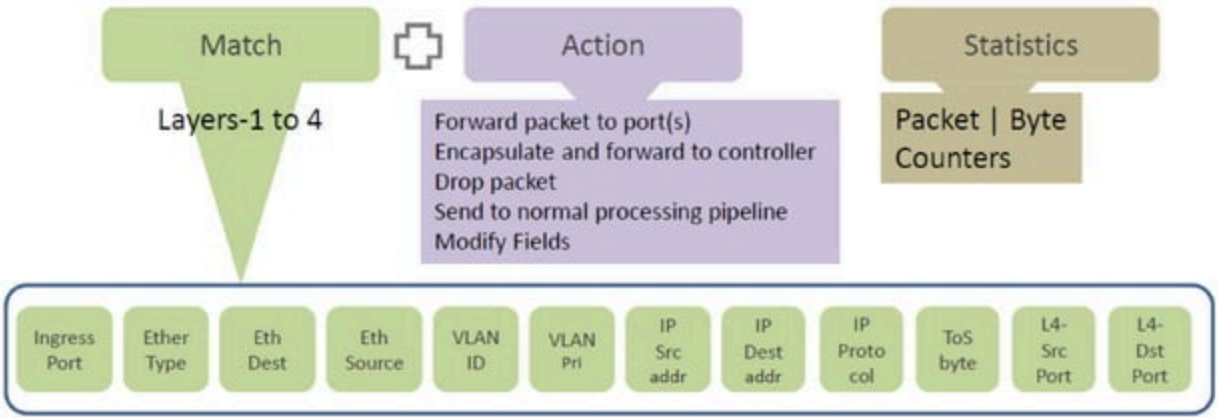
- Network interfaces for passing packets between Openflow Processing and the rest of the Network.
- Types of Ports:
  - Physical Ports
    - Switch defined ports
    - Eg. Physical ports map one to one Ethernet interfaces
  - Logical Ports
    - Switch defined ports that don't correspond to a hardware interface of switch
    - Logical ports include "Tunnel-ID".
  - Reserved Ports
    - Defined by OpenFlow Spec
    - Specifies generic forwarding actions such as sending to the controller, flooding and forwarding using non-openflow methods

Pipeline Processing



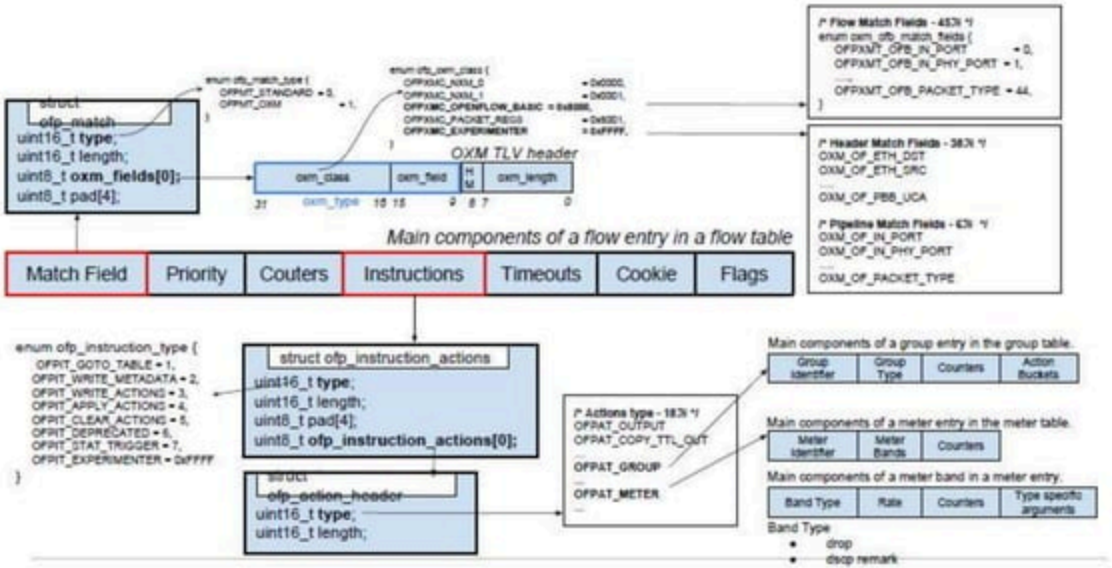
* Figure From OpenFlow Switch Specification

Flow Table Components



* Figure From OpenFlow Switch Specification

Flow Entry



* Figure From OpenFlow Switch Specification

Openflow Table Instructions

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

- Instructions are executed when a packet matches entry.
- Instruction result can
  - Change the packet
  - Action set
  - Pipeline processing etc.
- Sample Instruction Types
  - Meter ID - Direct a packet to the meter id.
  - Apply-Actions - Apply a specific action immediately
  - Clear-Actions - Clear all the actions in the action set
  - Write-Actions - Add a new action into the existing action set
  - Write-Metadata - Write the masked meta data value
  - Goto-Table - Indicate the next table in the processing pipeline

Action Set & Action List

- Action Set
  - Action set is associated with each packet. FE modify the action set using write-action/ clear-action
  - Actions in the action-set will be executed when pipeline is stopped
  - Action set contains maximum of one action of each type. If multiple actions of the same type need to be added then use "Apply-Actions"

- Action List
  - "Apply-action" , "packet-out" messages include action list
  - Execute an action immediately
  - Actions are executed sequentially in the order they have been specified
  - If action list contains an output action, a copy of the packet is forwarded in its current state to the desired port
  - Action-set shouldn't be changed because of action-list

Action

- Action
  - What to do with the packet when match criteria matches with the packet
  - Some of the Action Type
    - Output - Fwd a pkt to the specified open flow port (physical/ logical/reserved)
    - Set Queue - Determines which queue should be used for scheduling and forwarding packet
    - Drop - Packets which doesn't have output action should be dropped
    - Group - Process the packet through specified group
    - Push-Tag/ Pop-Tag - Insert VLAN, MPLS, PBB tage
    - Set-Field - Rewriting a field in the packet header
    - Change TTL - Decrement TTL
    - Copy TTL inwards – apply copy inward actions to the packet
    - Push MPLS – apply MPLS tag push action to the packet
    - Push PBB – apply PBB tag push action to the packet

Flow Entry (Contd...)

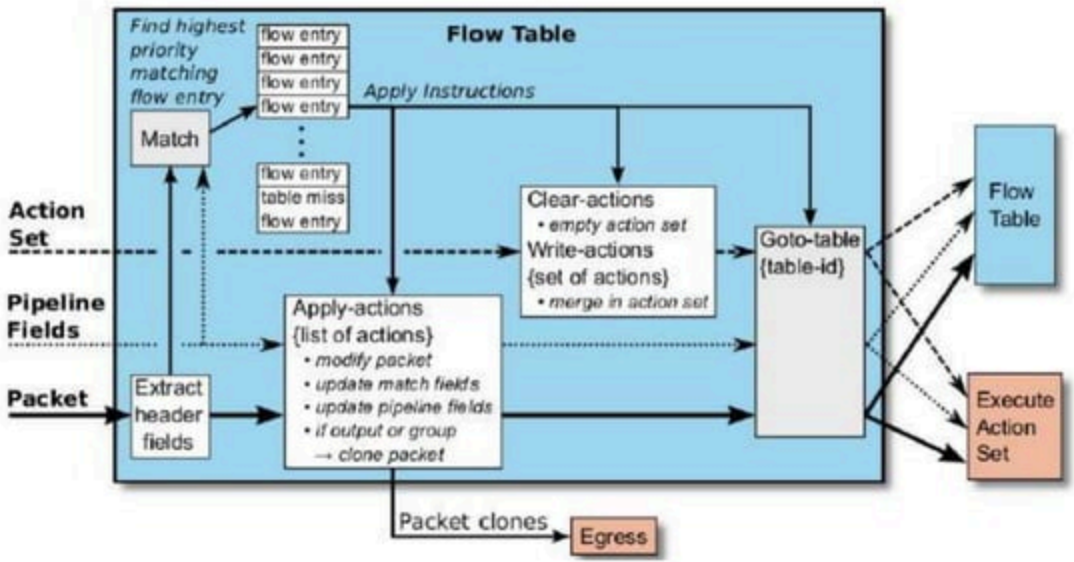
Flow Entry Timeouts

- Each Flow Entry contains an optional Idle and Hard Timeout field.
- Switch removes entry and sends flow removed message to controller when,
  - No packets have matched within the Idle Timeout
  - The flow was inserted more than the Hard Timeout.
- The timeouts are optional

Flow Entry (Contd…)

Reactive Flows	Proactive Flows
First packet of flow is sent to controller for processing	Controller pre-populates flow table in switch.
Controller evaluates packet and sends Flow message to switch insert a flow table entry for the packet	Zero set-up time for new flows
Subsequent packets (in flow) match the entry until idle or hard timeout	Switch in Proactive forwarding 'only' would default to action=drop
Loss of connectivity between switch and controller limit utility of switch	Loss of connection between switch and controller does not disrupt network traffic

Instruction Execution in Flow Table



* Figure From OpenFlow Switch Specification

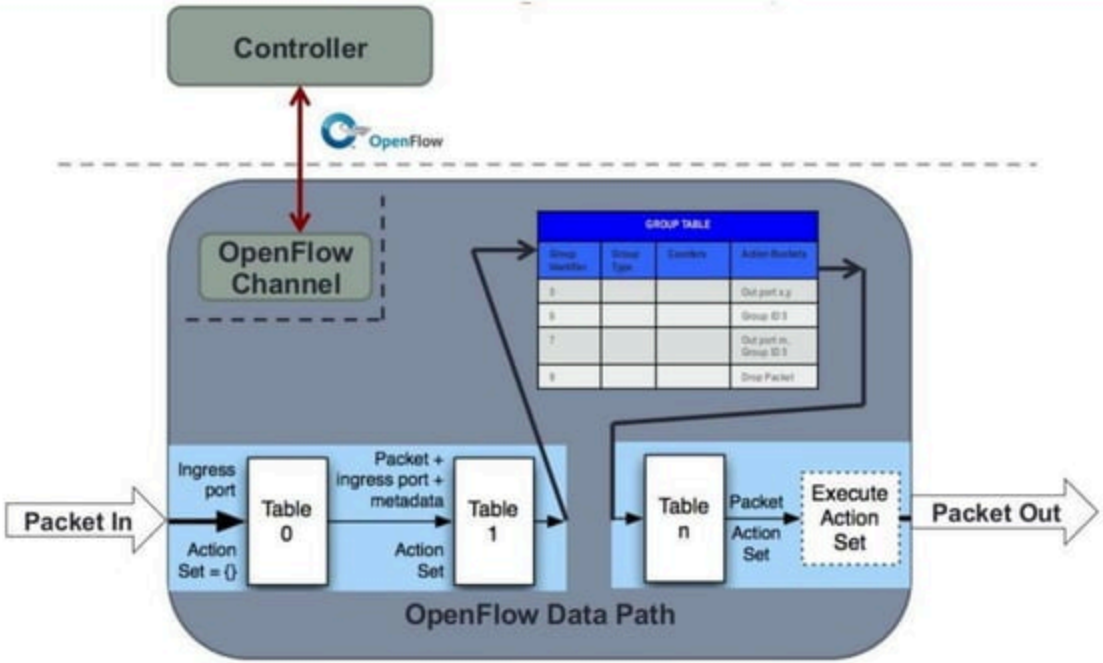
Group Table

- Additional method for forwarding to a group of entries.
- Comprises of Group ID, Group Type, Counters, Action buckets (each action bucket contains a set of actions to be executed)
- Group Types:
  - All
    - Execute all buckets in a group
    - Used mainly for multicast and broadcast – fwd a pkt on all the ports
  - Select
    - Execute one bucket in a Group (Eg. ECMP packets)
    - Implemented for load sharing and redundancy
  - Indirect
    - Execute one defined bucket in this Group
    - Supports only a single bucket (Eg. 40K routes are pointing to same next hop)
  - Fast failover
    - Execute the first live bucket
    - Eg. There is a primary path and secondary path – pass the traffic on primary path and if it fails use the secondary one

Group Table

Group Identifier	Group Type	Counters	Action Buckets
5			Out port x,y
15			Out port a
6			Group ID 5
7			Out port m, Group ID 15
9			Drop Packet

Instruction Execution in Group Table

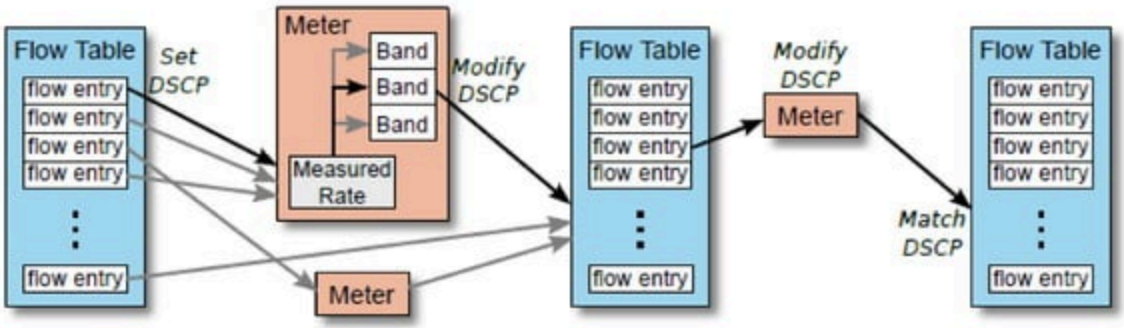


* Figure From OpenFlow Switch Specification

Meter Table

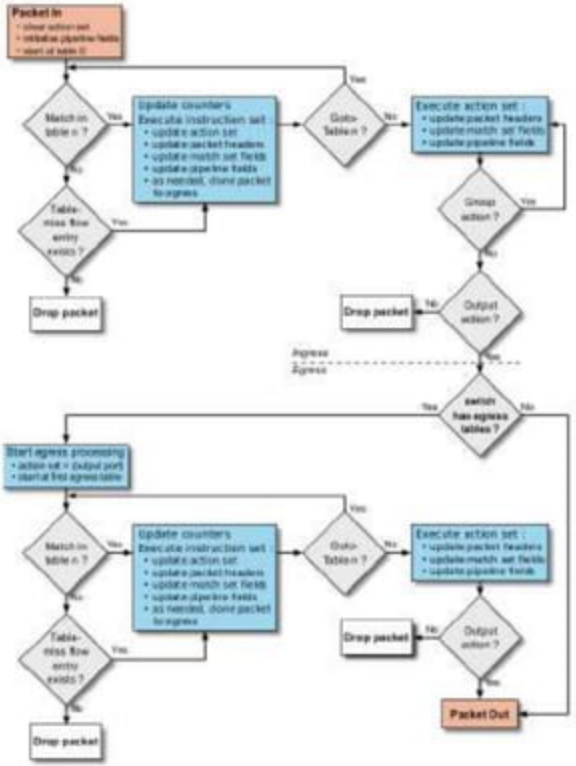
- Consists of meter entries and defining per-flow meters.
- Per-flow meters enable OF to implement QoS operations (rate-limiting)
- Components of Meter table:
  - Meter ID, Meter Band, Counters
- Meters measures the rate of packets assigned to it and enable controlling the rate of those packets
- Meters are attached directly to flow entries
- Meter band: unordered list of meter bands, where each meter band specifies the rate of the band and the way to process packet
- Components of Meter band:
  - Band Type, Rate, Counters, Type specific arguments
  - Band Type : defined how to process a packet (drop/ dscp remark)

Meter Table



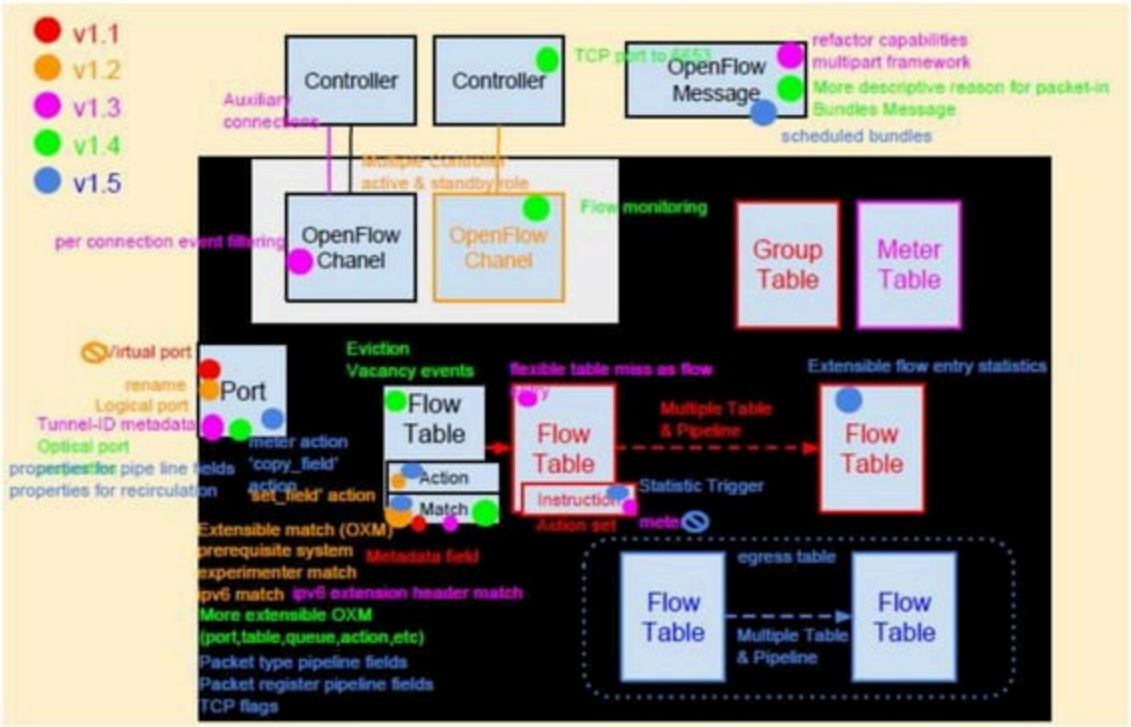
* Figure From OpenFlow Switch Specification

Packet Flow in OpenFlow Switch



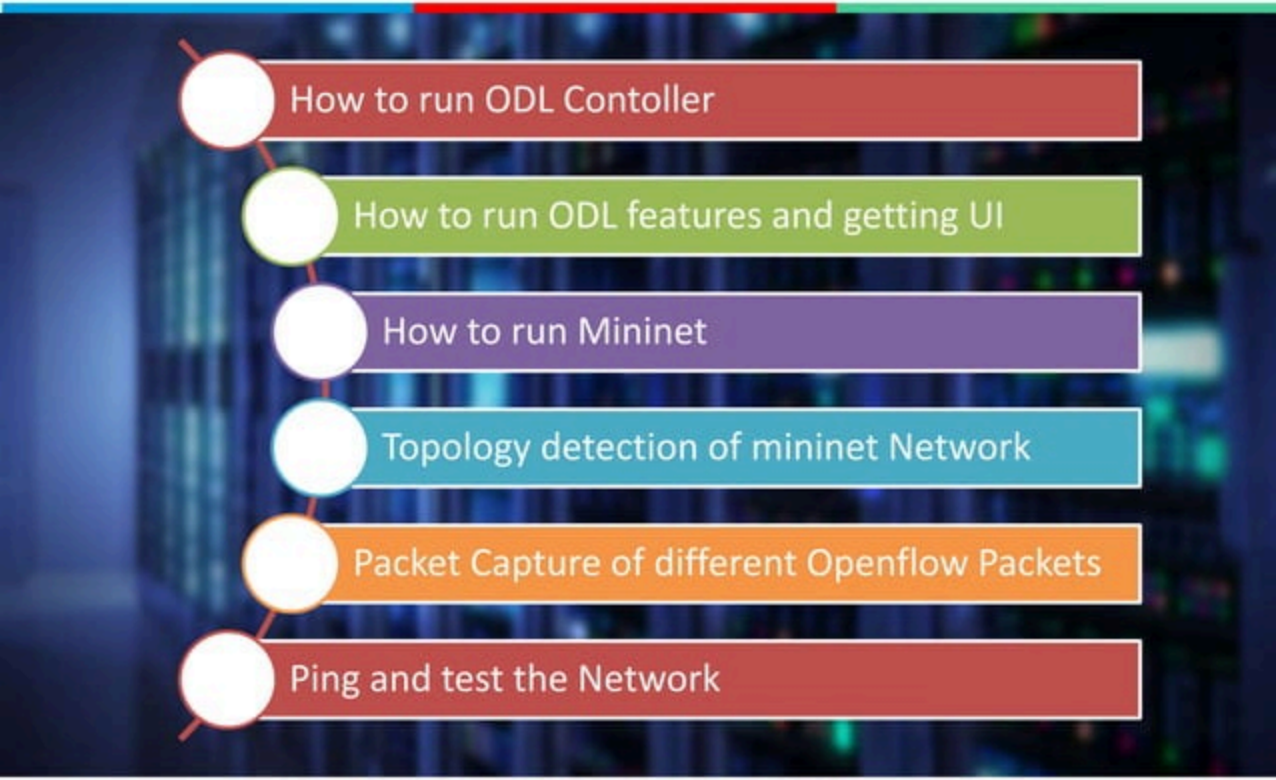
* Figure From OpenFlow Switch Specification

OpenFlow Features Vs Specification




LAB Exercise

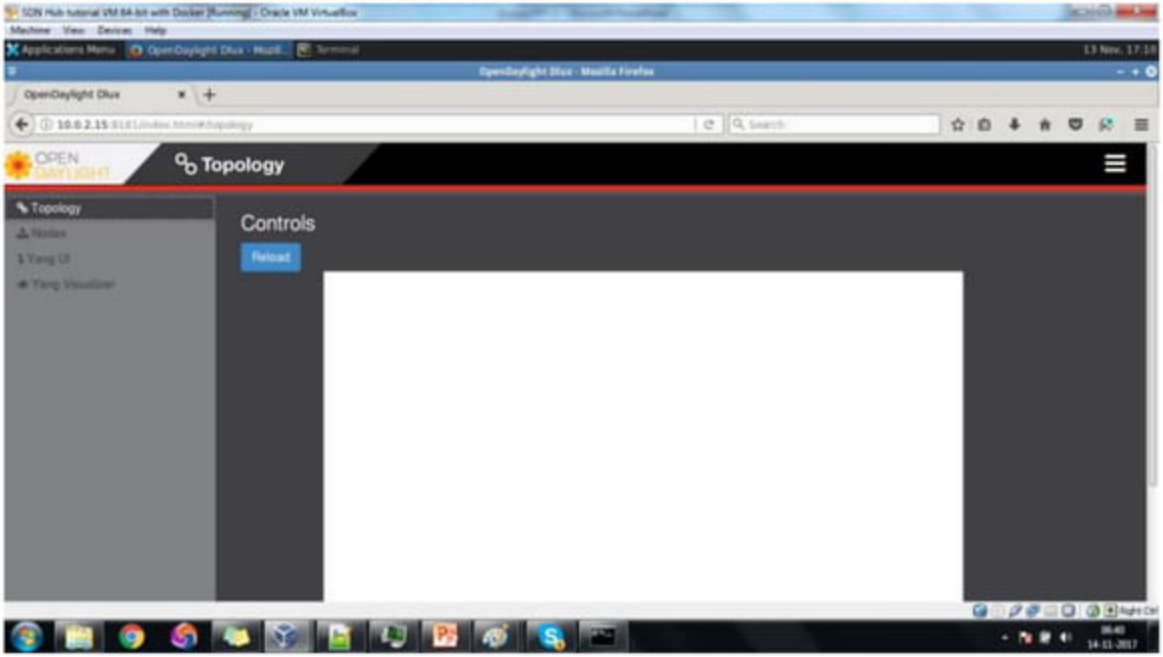
Course Objective



How to start OpenDayLight Controller

- Run the given VM in Virtual Box
- Goto distribution-karaf-0.4.4-Beryllium-SR4/
  - cd distribution-karaf-0.4.4-Beryllium-SR4/
- Run ./bin/karaf
- Install the basic required karaf features/bundles for the lab exercise
  - feature:install odl-restconf-all odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all
  - feature:install odl-restconf-noauth odl-netconf-connector-all
- Go to browser and open the page http://10.0.2.15:8181/index.html
- Login with admin, admin

OpenDaylight Controller UI




MININET & OpenVSwitch (OVS)

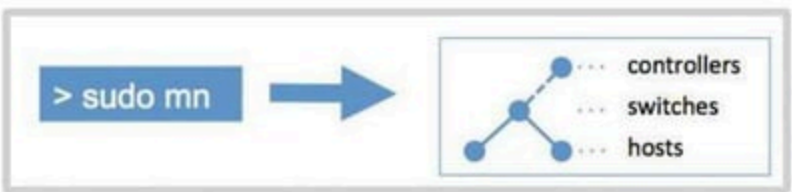
Mininet

- Mininet creates a realistic OpenFlow network, running real kernel, switch and application code, on a single machine (VM, cloud or native), in seconds, with a single command

- sudo mn --topo linear,3 --mac --controller=remote,ip=10.0.2.15,port=6633 --switch ovs,protocols=OpenFlow13

- sudo ovs-ofctl -O OpenFlow13 dump-flows s2

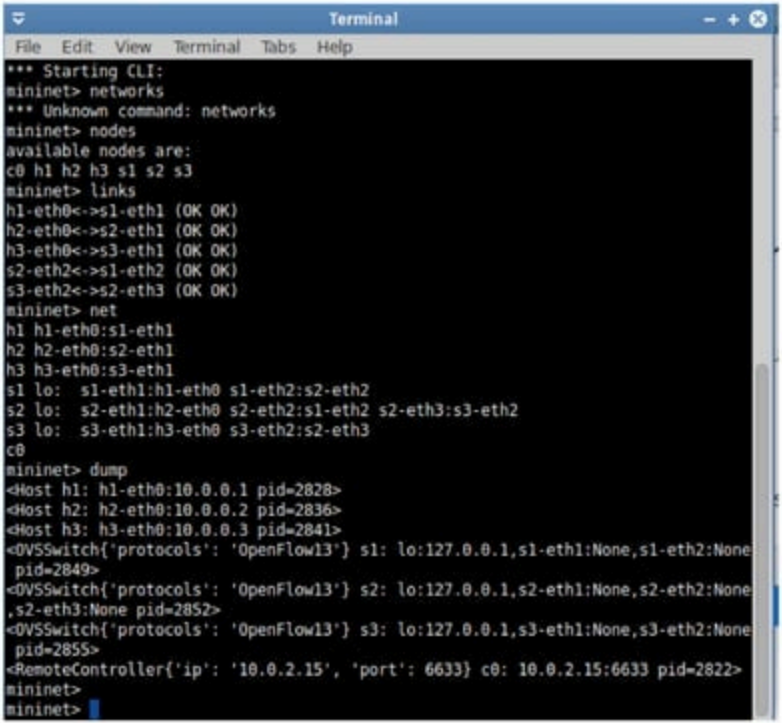
- sudo ovs-vsctl show



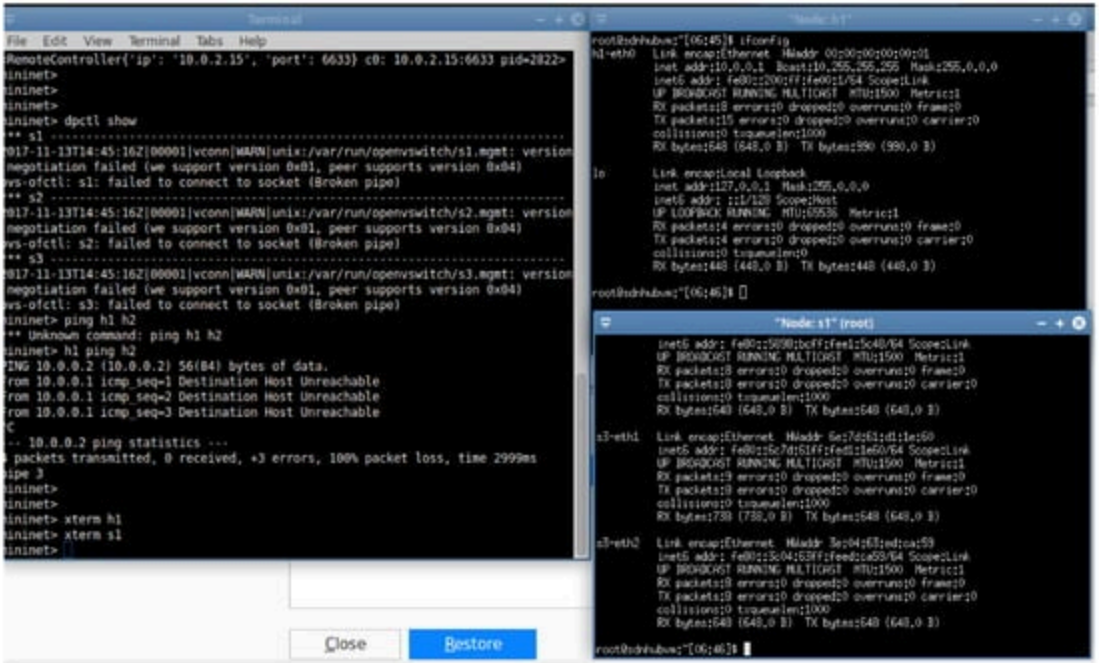
Mininet - Installation

- Option 1: Mininet VM Installation - http://mininet.org/download/#option-1-mininet-vm-installation-easy-recommended
- Option 2: Native Installation from Source - http://mininet.org/download/#option-2-native-installation-from-source
- Option 3: Installation from Packages - http://mininet.org/download/#option-3-installation-from-packages
- Option 4: Upgrading an existing Mininet Installation - http://mininet.org/download/#option-4-upgrading-an-existing-mininet-installation

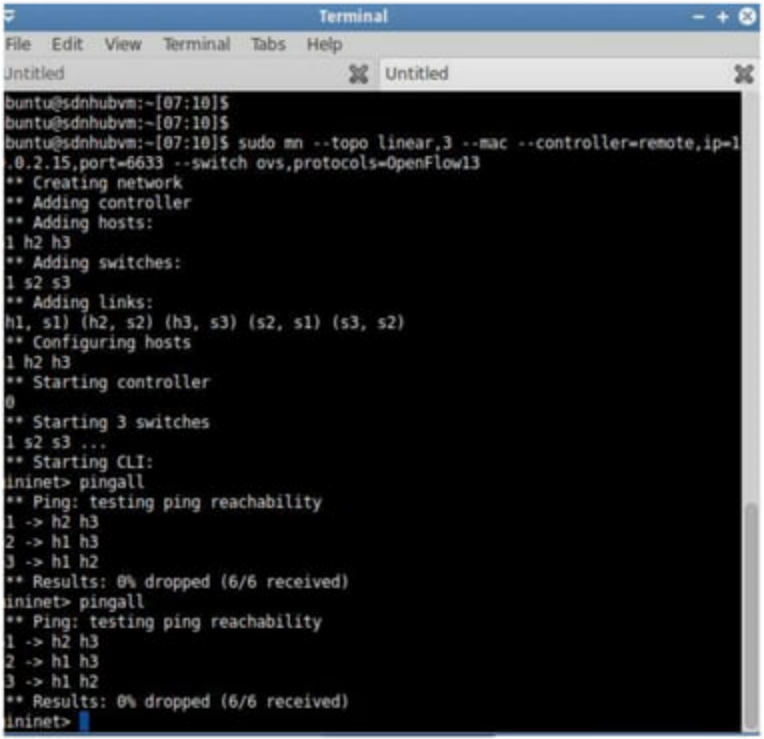
Mininet – Some show commands



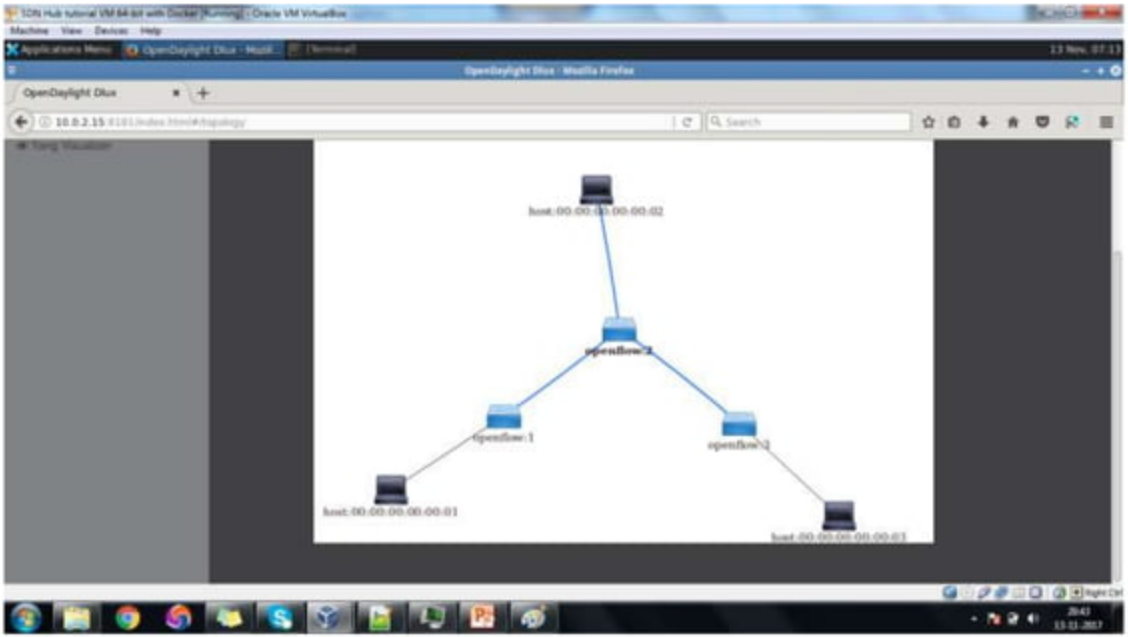
Mininet – Some show commands



Mininet – Some show commands



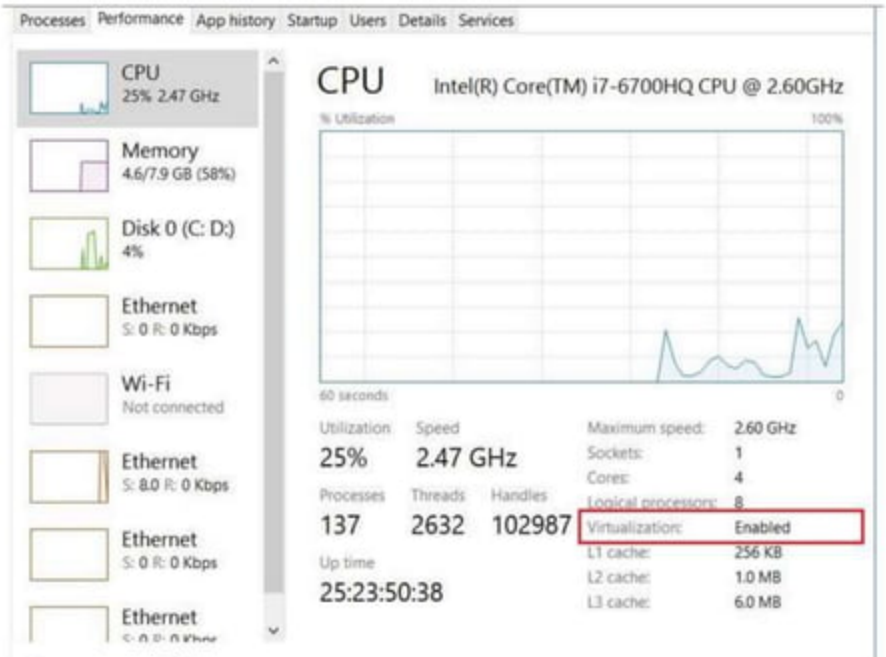
Mininet – Some show commands



Thank you

Backup Slides

How to check VT-x enabled in Windows 8?



How to check VT-x enabled in Windows 7?

- Install the Microsoft® Hardware-Assisted Virtualization Detection Tool from https://www.microsoft.com/en-us/download/details.aspx?id=592
- Run it and see whether you get this output