Mawlana Bhashani Science and Technology University

Assignment

Assignment No:02

Assignment name: OpenFlow Protocol

Course code: ICT-4101

Course title: Telecommunication Engineering

Date of Performance:10-10-2020

Date of Submission:19-10-2020

Submitted by

Name: Md. Rafatul Haque

ID:IT-17037

4th year 1st semester

Session: 2016-2017

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Assignment name: OpenFlow Protocol

Objective:

- ❖ Understand the working principles of OpenFlow protocol.
- ❖ Configure a basic Software Defined Network for end-to-end communications.
- ❖ Understand the difference between interacting with real and virtual networks.

Theory:

OpenFlow

OpenFlow is a protocol that allows a server to tell network switches where to send packets. In a conventional network, each switch has proprietary software that tells it what to do. With OpenFlow, the packet-moving decisions are centralized, so that the network can be programmed independently of the individual switches and data center gear.

In a conventional switch, packet forwarding (the data path) and high-level routing (the control path) occur on the same device. An OpenFlow switch separates the data path from the control path. The data path portion resides on the switch itself; a separate controller makes high-level routing decisions. The switch and controller communicate by means of the OpenFlow protocol. This methodology, known as software-defined networking (SDN), allows for more effective use of network resources than is possible with traditional networks. OpenFlow has gained favor in applications such as VM (virtual machine) mobility, mission-critical networks, and next generation IP-based mobile networks.

OpenFlow specification terms

This section describes key OpenFlow specification terms:

- **Byte:** an 8-bit octet.
- **Packet:** an Ethernet frame, including header and payload.

- **Port:** where packets enter and exit the OpenFlow pipeline. May be a physical port, a logical port defined by the switch, or a reserved port defined by the OpenFlow protocol.
- **Pipeline:** the set of linked flow tables that provide matching, forwarding, and packet modifications in an OpenFlow switch.
- **Flow Table:** A stage of the pipeline, contains flow entries.
- **Flow Entry:** an element in a flow table used to match and process packets. It contains a set of match fields for matching packets, a priority for matching precedence, a set of counters to track packets, and a set of instructions to apply.
- **Match Field:** a field against which a packet is matched, including packet headers, the ingress port, and the metadata value. A match field may be wild carded (match any value) and in some cases bit masked.

- **Metadata:** a mask able register value that is used to carry information from one table to the next.
- **Instruction:** Instructions are attached to a flow entry and describe the OpenFlow processing that happen when a packet matches the flow entry. An instruction either modifies pipeline processing, such as direct the packet to another flow table, or contains a set of actions to add to the action set, or contains a list of actions to apply immediately to the packet.
- **Action:** an operation that forwards the packet to a port or modifies the packet, such as decrementing the TTL field. Actions may be specified as part of the instruction set associated with a flow entry or in an action bucket associated with a group entry. Actions may be accumulated in the Action Set of the packet or applied immediately to the packet.
- **Action Set:** a set of actions associated with the packet that are accumulated while the packet is processed by each table and that are executed when the instruction set instructs the packet to exit the processing pipeline.
- **Group:** a list of action buckets and some means of choosing one or more of those buckets to apply on a per-packet basis.
- **Action Bucket:** a set of actions and associated parameters, defined for groups.

- **Tag:** a header that can be inserted or removed from a packet via push and pop actions.
- **Outermost Tag:** the tag that appears closest to the beginning of a packet.
- Controller: An entity interacting with the OpenFlow switches using the OpenFlow protocol.
- **Meter:** a switch element that can measure and control the rate of packets. The meter trigger a meter band if the packet rate or byte rate passing through the meter exceed a predefined threshold. If the meter band drops the packet, it is called a Rate Limiter.

Methodology

This lab is designed for working on group of three students. In this activity students will learn how a switch hub works and how it is implemented using OpenFlow.

Traditional Switching Hub

Switching hubs have a variety of functions. Here, we take a look at a switching hub having the following simple functions.

- Learns the MAC address of the host connected to a port and retains it in the MAC address table.
- When receiving packets addressed to a host already learned, transfers them to the port connected to the host.
- When receiving packets addressed to an unknown host, performs flooding.

Switching Hub by OpenFlow

OpenFlow switches can perform the following by receiving instructions from OpenFlow controllers such as Ryu:

• Rewrites the address of received packets or transfers the packets from the specified port.

• Transfers the received packets to the controller (Packet-In).

• Transfers the packets forwarded by the controller from the specified port (Packet-Out).

It is possible to achieve a switching hub having those functions combined.

First of all, you need to use the Packet-In function to learn MAC addresses. The controller can use the Packet-In function to receive packets from the switch. The switch analyzes the received packets to learn the MAC address of the host and information about the connected port.

After learning, the switch transfers the received packets. The switch investigates whether the destination MAC address of the packets belong to the learned host. Depending on the investigation results, the switch performs the following processing.

• If the host is already a learned host ... Uses the Packet-Out function to transfer the packets from the connected port.

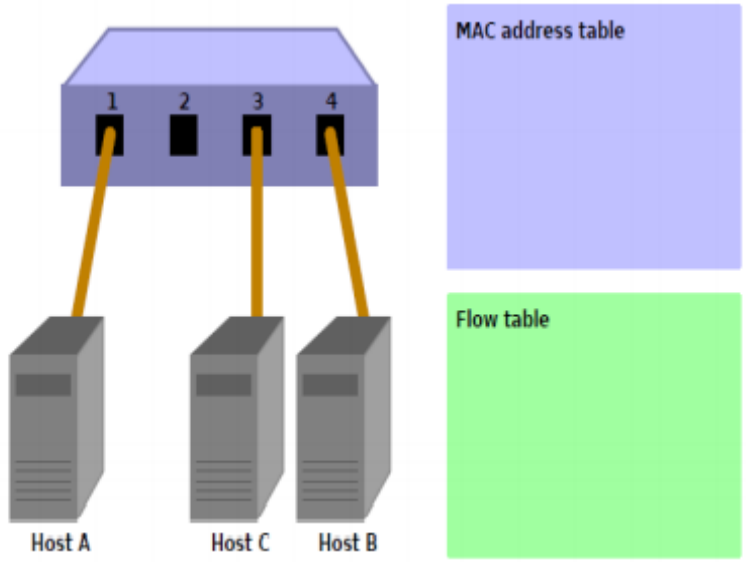
• If the host is unknown host ... Use the Packet-Out function to perform flooding.

The following explains the above operation in a step-by-step way using figures.

1. Initial status

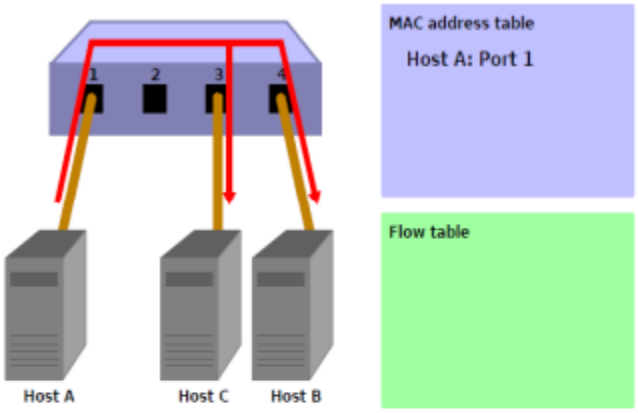
• This is the initial status where the flow table is empty.

• Assuming host A is connected to port 1, host B to part 4, and host C to port 3.



2. Host A -> Host B

• When packets are sent from host A to host B, a Packet-In message is sent and the MAC address of host A is learned by port 1. Because the port for host B has not been found, the packets are flooded and are received by host B and host C.



Packet-In:

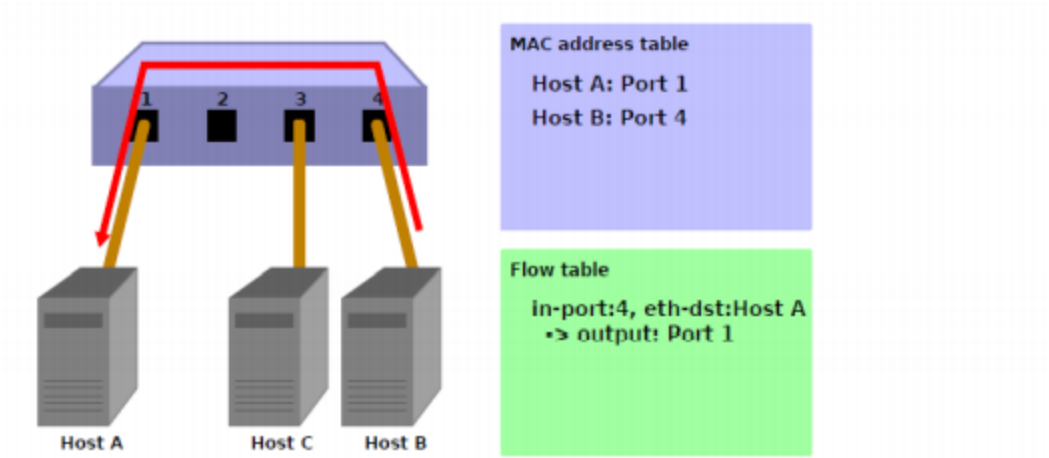
```
in-port: 1
eth-dst: Host B
eth-src: Host A
```

Packet-Out:

```
action: OUTPUT:Flooding
```

3. Host B -> Host A

• When the packets are returned from host B to host A, an entry is added to the flow table and also the packets are transferred to port 1. For that reason, the packets are not received by host C.



Packet-In:

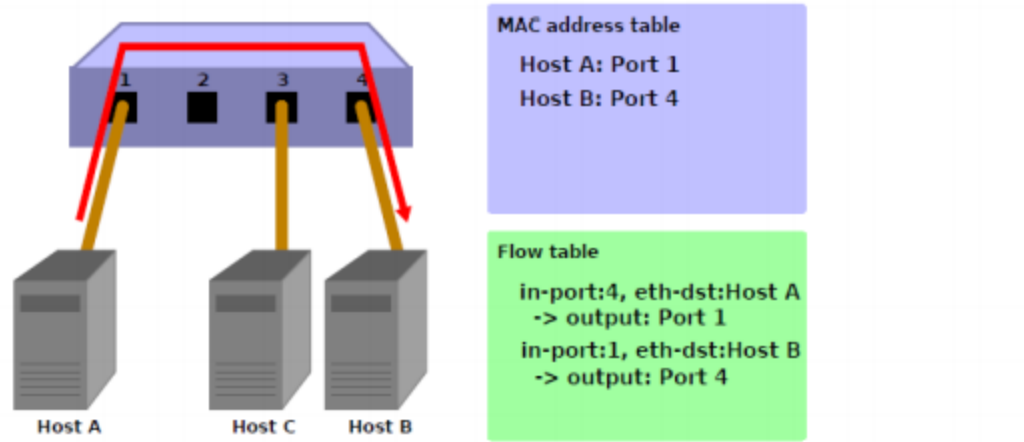
```
in-port: 4
eth-dst: Host A
eth-src: Host B
```

Packet-Out:

```
action: OUTPUT:Port 1
```

4. Host A -> Host B

• Again, when packets are sent from host A to host B, an entry is added to the flow table and also the packets are transferred to port 4.



Packet-In:

```
in-port: 1
eth-dst: Host B
eth-src: Host A
```

Packet-Out:

```
action: OUTPUT:Port 4
```

Conclusion: The realization of splitting the data and control plane from the network equipment and using OpenFlow Protocol to exchange control information between both planes afford an integration of a new network management. The availability to execute instructions to specified indicated packets by the flow table makes the network management very powerful and allows integrating new network protocols to improve the network services. The specialty of OpenFlow is that the manufactures don't need to open their operation systems of their network equipment. They just have to add the OpenFlow feature to their operation system without changing the basic software. This future network management will realize a better scalability, efficiency, stability and integration and offer an optimal routing on a global network. It will replace many network patches and provide an easy integration of new configurable services.