Mawlana Bhashani Science and Technology University

Department of Information and Communication Technology

Course Code	:	*ICT - 4101*
Course Name	:	*Telecommunication Engineering*
Name of the Assignment	:	*Open Flow Protocol*
Assignment No.	:	*02*

Submitted to:
Mr. Nazrul Islam
Assistant Professor,
Dept. of ICT, MBSTU

Submitted by:
Md. Nazmul Hasan
ID: IT-17005
Session: 2016-17
4th Year 1st Semester

Date of Submission: 20-October-2020

Open Flow Protocol

Objective: The objective of the Assignment is to:

- 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.

Q: 01. What is OpenFlow Protocol?

Answer: OpenFlow (OF) is considered one of the first software-defined networking (SDN) standards. It originally defined the communication protocol in SDN environments that enables the SDN Controller to directly interact with the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based), so it can better adapt to changing business requirements.

Q: 02. Describe the main components of an OpenFlow switch?

Answer: The first standard for the SDN was the OpenFlow. Because of having features such as flexibility, the OpenFlow has more popularity in academic and industrial environments than other protocols such as IP and Ethernet. All in all, using the OpenFlow has a lot of advantages in network virtualization and routing distribution [10]. The OpenFlow protocol makes the link between the OpenFlow Switch and controller via the Secure Socket Layer (SSL). The OpenFlow Switch basically consists of tables called flow tables. The

controller's commands to those tables are performed via the OpenFlow protocol [11]. A profile of the OpenFlow Switch is observed in figure 2.1. This figure, the controller is connected to the OpenFlow Switch via the SSL. It modifies and updates flow tables of the switch via the OpenFlow protocol [11, 12]. The controller is basically the main intelligence of the network and acts as a network operating system. In fact, the OpenFlow controller takes the responsibility of controlling the network hardware and facilitates network management [14]. More information about the OpenFlow controller can be searched in the OpenFlow standard and other resources such as [15-17].

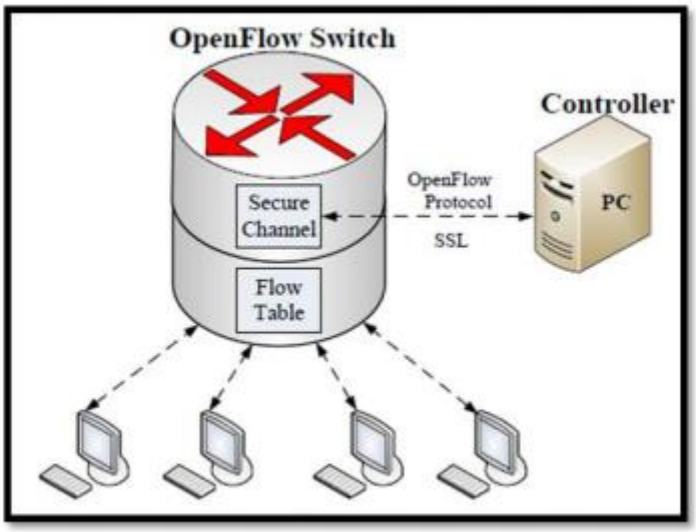


Figure 1: An OpenFlow Switch [13]

The OpenFlow Switch is one of the key components of the SDN architecture. After entering the OpenFlow Switch, packets are investigated in order that necessary actions can be applied on them. As observed in figure 2, the switch consists of three basic components of the OpenFlow Table, OpenFlow Protocol, and Source Channel [11, 12]. The control plane in the switch is responsible for decision making, policy making, while the data plane has the duty of traffic transfer.

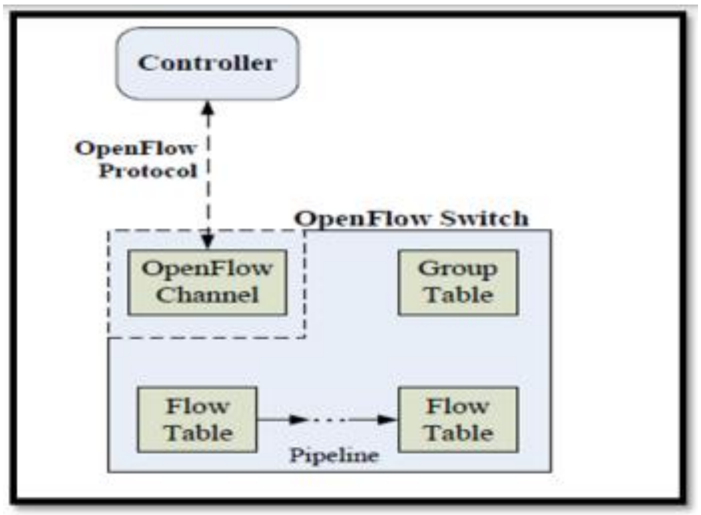


Figure 2: Basic components of the OpenFlow switch [18]

Q: 03. Describe the OpenFlow specification terms?

<u>**Answer:**</u> 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 wildcarded (match any value) and in some cases bitmasked.
- Metadata: a maskable 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.

Q: 04. How to switching hub by OpenFlow protocol?

Answer: 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.

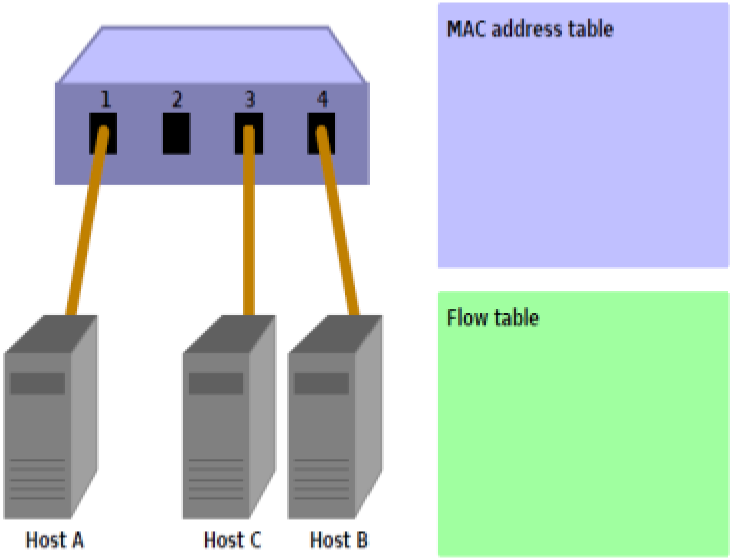
First of all, you need to use the Packet-In function to learn MAC addresses. The controller can use the Packet-In function to receive packets from the switch. The switch analyzes the received packets to learn the MAC address of the host and information about the connected port. After learning, the switch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

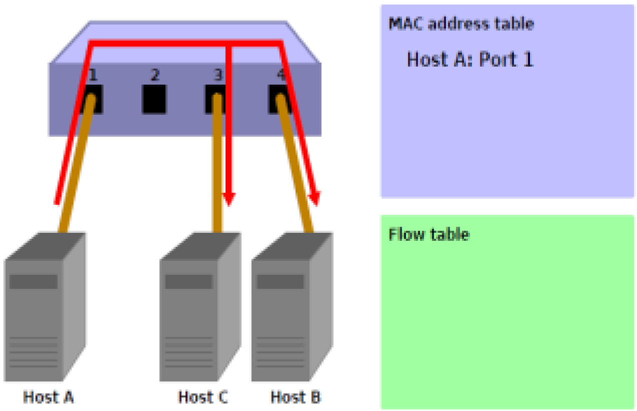
1. Initial status

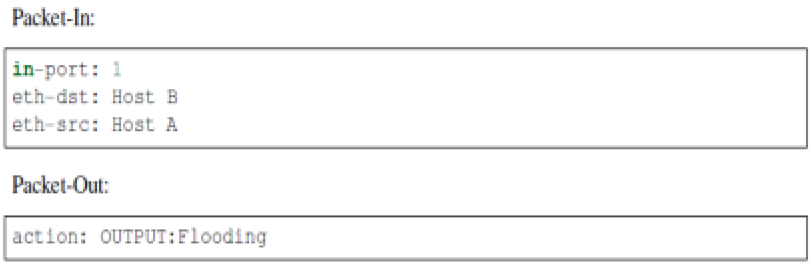
- This is the initial status where the flow table is empty.
- Assuming host A is connected to port 1, host B to part 4, and host C to port 3.



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:

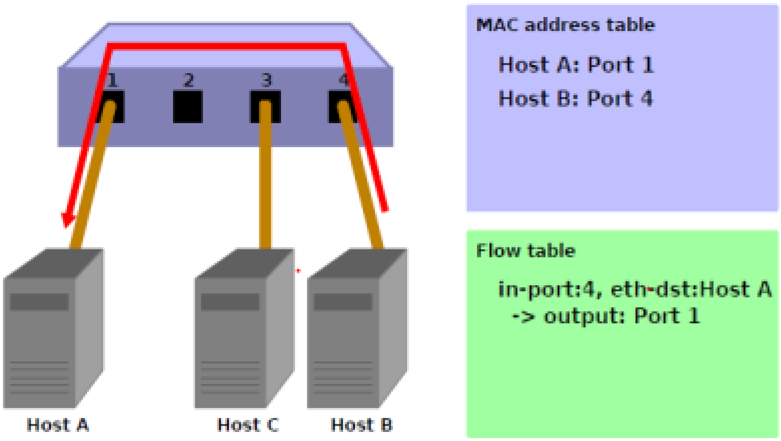
```
in-port: 1
eth-dst: Host B
eth-src: Host A
```

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.



MAC address table

Host A: Port 1
Host B: Port 4

Flow table

in-port:4, eth-dst:Host A
-> output: Port 1

Packet-In:

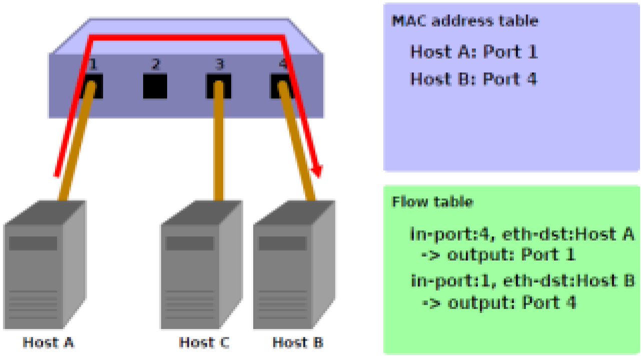
```
in-port: 4
eth-dst: Host A
eth-src: Host B
```

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.



MAC address table

Host A: Port 1
Host B: Port 4

Flow table

in-port:4, eth-dst:Host A
-> output: Port 1
in-port:1, eth-dst:Host B
-> output: Port 4

Packet-In:

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

Packet-Out:

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

Conclusion: This assignment helps me to know the OpenFlow Protocol. I can learn the components of switch of an OpenFlow protocol. The methodology and switching hub of OpenFlow protocol help me to get good information about the functionality of it.