## MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY,

## Santosh, Tangail -1902

| | | |
|---|---|---|
| **Assignment No** | **:** | 02 |
| **Assignment Name** | **:** | OpenFlow Protocol |
| **Course Name** | **:** | Introduction to Telecommunication Systems |

**Submitted by,**

**Name :** Sabrin Afroz

**ID :** IT-17007

**Session :** 2016-17

Dept. of ICT, MBSTU.

**Submitted to,**

Nazrul Islam

Assistant Professor

Dept. of ICT, MBSTU.

An OpenFlow Switch consists of one or more flow tables and a group table, which perform packet lookups and forwarding, and an OpenFlow channel to an external controller (Figure 2-1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol. Using the OpenFlow protocol, the controller can add, update, and delete flow entries in flow tables, both reactively (in response to packets) and proactively. Each flow table in the switch contains a set of flow entries; each flow entry consists of match fields, counters, and a set of instructions to apply to matching packets. Matching starts at the first flow table and may continue to additional flow tables. Flow entries match packets in priority order, with the first matching entry in each table being used. If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry: for example, the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table. Page | 51 SDN-Labs Instructions associated with each flow entry either contain actions or modify pipeline processing. Actions included in instructions describe packet forwarding, packet modification and group table processing. Pipeline processing instructions allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, to be communicated between tables. Table pipeline processing stops when the instruction set associated with a matching flow entry does not specify a next table; at this point the packet is usually modified and forwarded. Flow entries may forward to a port. This is usually a physical port, but it may also be a logical port defined by the switch or a reserved port defined by this specification. Reserved ports may specify generic forwarding actions such as sending to the controller, flooding, or forwarding using non-OpenFlow methods, such as normal switch processing while switch-defined logical ports may specify link aggregation groups, tunnels or loopback interfaces. Actions associated with flow entries may also direct packets to a group, which specifies additional processing. Groups represent sets of actions for flooding, as well as more complex forwarding semantics (e.g. multipath, fast reroute, and link aggregation). As a general layer of indirection, groups also enable multiple flow entries to forward to a single identifier (e.g. IP forwarding to a common next hop). This abstraction allows common output actions across flow entries to be changed efficiently. The group table contains group entries; each group entry contains a list of action buckets with specific semantics dependent on group type. The actions in one or more action buckets are applied to packets sent to the group. Switch designers are free to implement the internals in any way convenient, provided that correct match and instruction semantics are preserved. For example, while a flow entry may use an all group to forward to multiple ports, a switch designer may choose to implement this as a single bitmask within the hardware forwarding table. Another example is matching; the pipeline exposed by an OpenFlow switch may be physically implemented with a different number of hardware tables.

**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 wildcarded (match any value) and in some cases bitmasked. Page | 52 SDN-Labs

- 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.

**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. Page | 53 SDN-Labs 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.

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.

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.

**Q . Explain at least two the advantage and disadvantage of using mininet .**

Ans :

**Mininet Advantages:**

Mininet combines many of the best features of emulators, hardware testbeds, and simulators.

**Mininet Disadvantages:**

Mininet-based networks cannot (currently) exceed the CPU or bandwidth available on a single server.

Mininet cannot (currently) run non-Linux-compatible OpenFlow switches or applications; this has not been a major issue in practice.

**Q. Explain how the open flow tables are created?**

Ans : OpenFlow switch atchitecture An OpenFlow Switch consists of one or more flow tables and a group table, which perform packet lookups and forwarding, and a secure OpenFlow channel that provide communication between controller and openflow switches. Flow table are the fundamental data structure in an openflow device. Each flow table in the switch contains a set of flow entries. These flow tables allow the devices to inspect incoming packet based on certain filed and take proper action according on the contents of the packet information that has been received. Matching starts at the first flow table and may continue to additional flow tables. Flow entries match packets in priority order, with the first matching entry in each table being used. If a matching entry is found, the instructions associated with the specific flow entry are executed. These actions may include forwarding the packet to a specific port, dropping the packet, or flooding the packet on all port. If no match is found in a flow table, the outcome depends on switch configuration: the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table.

**Conclusion :** The OpenFlow protocol defines the communication between an OpenFlow controller and an OpenFlow switch. This protocol is what most uniquely identifies OpenFlow technology. At its essence, the protocol consists of a set of messages that are sent from the controller to the switch and a corresponding set of messages that are sent in the opposite direction.

For this experiment, we have to need USB and Ethernet connection between PC and Zodiac FX.