Name: Md.Abdullah

ID : IT-17015

Assignment no: 02

Name of the assignment: OpenFlow Control

Objectives:

- ✔ 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 Protocol

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.

OpenFlow switch

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 MV (virtual machine) mobility, mission-critical networks, and next generation IP-based mobile networks.This speciation covers the components and the basic functions of the switch, and the OpenFlow protocol to manage an OpenFlow switch from a remote controller.

Switch Components :

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.

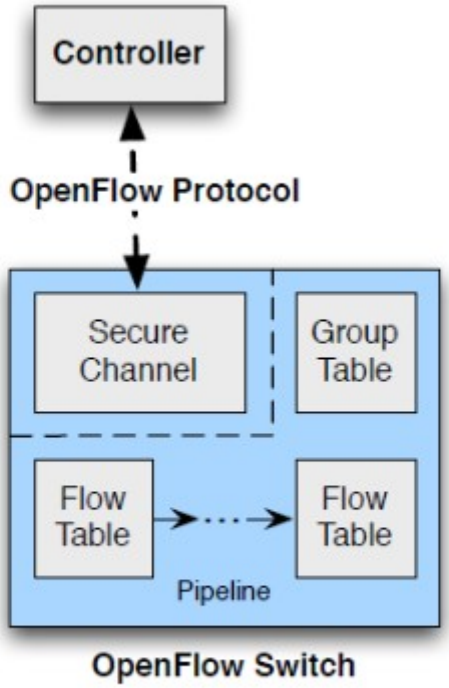


Figure 2-1. Main components of an OpenFlow switch.

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.

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

- ✔ Byte: an 8-bit octet.
- ✔ Packet: an Ethernet frame, including header and payload.
- ✔ Port: where packets enter and exit the OpenFlow pipeline.
- ✔ 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.

- ✔ Match Field: a field against which a packet is matched, including packet headers, the ingress port, and the metadata value.
- ✔ 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.
- ✔ Action: an operation that forwards the packet to a port or modifies the packet, such as decrementing the TTL field
- ✔ 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.

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.

Question 5.1: How the RYU GUI interface can be improved? Provide some ideas.

Ans:

- ✔ Provide an interface to add or remove flow.
- ✔ Provide the configuration of the switches, ports, IP address Vlans.

Question 5.2: Explain at least two the advantage and disadvantage of using mininet or Zodiac FX?

Ans:

Advantages:

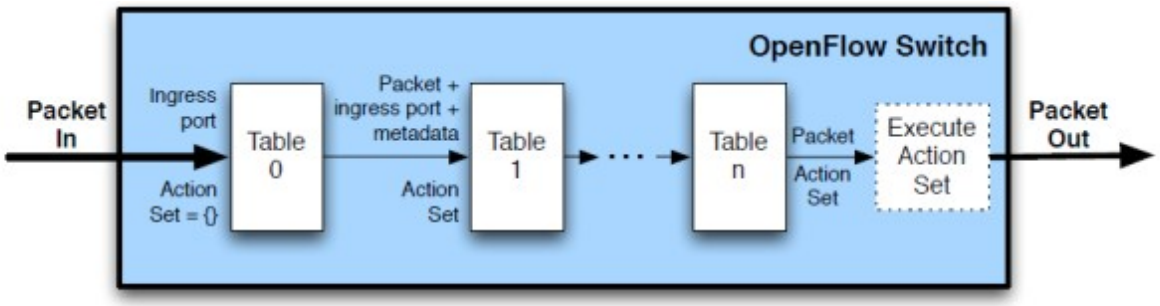
- ✔ Scalability and flexibility in terms of topology
- ✔ Real hardware provide a better learning experience

Disadvantages:

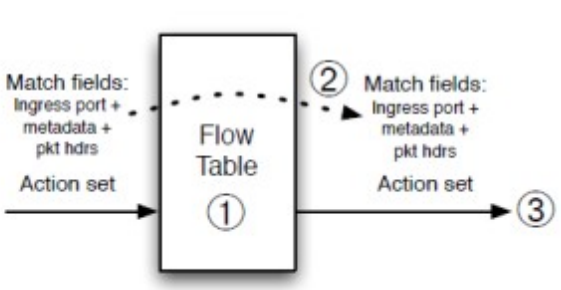
- ➢ Difficult to visualize conflicts
- ➢ Limited number or physical ports

Question 5.3: Explain how the open flow tables are created?

Ans:



(a) Packets are matched against multiple tables in the pipeline



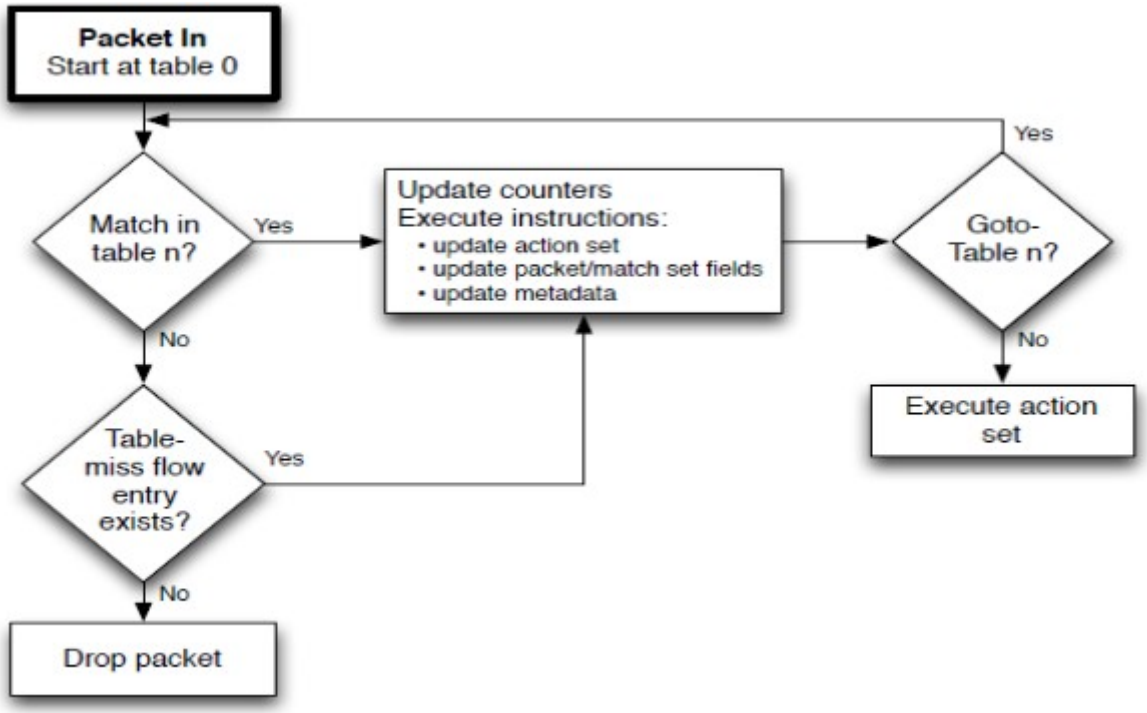
① Find highest-priority matching flow entry

② Apply instructions:
- i. Modify packet & update match fields (apply actions instruction)
- ii. Update action set (clear actions and/or write actions instructions)
- iii. Update metadata

③ Send match data and action set to next table

Question 5.4: When Zodiac FX receive a packet, which functions an OpenFlow Switch performs?

Ans:



Conclusion:

I couldn't be able to install graphical interface of ryu controller. I tried my several times but I couldn't find anything about that help to download ryu graphical interface controller. In addition , we don't have any zodiac FX which will connect to a computer through a cable. For these reason I couldn't be able to complete the exercises. But I learn about what is open flow protocol, open flow switches and how these are works.