A REVIEW OF OPEN FLOW PROTOCOL IN SOFTWARE-DEFINED NETWORK

Mrs. Hemagowri J[1] *Dr.P.TamilSelvan*[2]

ABSTRACT

In recent years network traffic has been increasing rapidly beIn recent years network traffic has been increasing rapidly because of the usage of a lot of online applications and cloud services using mobile devices. Then the collection of data formats, online devices and service types are needed to manage the network. Further, the network operators need to ensure availability of service and its quality and security. All this should be done without any increase in the cost of operation and equipment. Nowadays, Software-defined networking (SDN) has attracted more attention in IT and research fields. And, Open Flow (OF) is one of the first SDN standards which helps the SDN controller communicate directly with the data plane of network devices such as switch and router, which can be either physical or virtual[1]. This paper, first outlines the basic principles of SDN and open flow and also explains the features which are supported by different versions of open flow protocol.

***Keywords*:** Open Flow Protocol, SDN

1. INTRODUCTION

In the past few years, with the usage of internet data communication has been growing steadily necessitating the need of new technologies in Information and Communication through various applications, which is a challenge to the future of internet. In a traditional approach, the configuration of devices is done manually, and there is a possibility of error. To solve this problem network should be maintained and updated dynamically [6]. This can be done by using Software-Defined Network. In a traditional method, the basic functions of router are receiving the packets, checking the routing table, forwarding packets etc. These functions are done through router of the control plane and data plane, and the SDN is used to separates the control plane from forwarding plane (data plane). While the control plane acts as a brain or decision-maker when transferring the packets, the data plane acts a packet forwarder in dedicated hardware. A single control plane controls several forwarding planes. Open Flow is a southbound API that is standardized between data plane and control plane which enables the technology in SDN environment that helps the SDN controller communicate directly with the data plane of network devices such as router and switch.

2. OPEN FLOW OVERVIEW

The three main components of Open Flow architecture are

1) one or more Open Flow switches, which is referred to as data/forwarding plane
2) one or more Open Flow controllers(control plane) and
3) a secure path to connect both planes.

The Open Flow protocol supports messages to be sent and received between network devices ((i.e. switches) and controller. The Open Flow protocol has three different kind of messages, namely **(i) Controller-to-switch, (ii) Asynchronous and (iii) Symmetric.**

Controller-to-Switch Message

This message is initiated by the controller, which can manage or examine the switch state. It may expect switch to respond to the query sent by controller. For example, the controller may send a feature-request query to switch; the switch must send a reply that specifies the switch capability.

[1]Assistant Professor, Dept. of CA, CS & IT
Karpagam Academy of Higher Education, Coimbatore, India.

[2]Associate Professor, Dept. of CA, CS & IT
Karpagam Academy of Higher Education, Coimbatore, India.

Likewise, the controller will send configuration query, modify-state messages, read-state messages, packet-out messages and barrier request/reply messages to switch with the purpose of knowing or managing the switch state [1].

Asynchronous Message

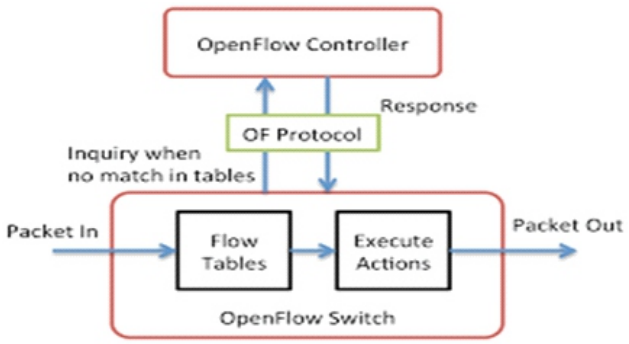
Switch initiates asynchronous messages for the purpose of notifying the network activities and any modification in the switch state. The switch usually sends asynchronous message when packet enters into the switch or when the status of switch change or any fault occurs. It has four sub-types, namely flow-removed message, packet-in message, error message and port-status message.

Symmetric Message

Symmetric messages are sent by either controller or switch for the purpose of knowing the connection state or to measure the latency. The sub-message types are Hello message and Echo message.

3. OPEN FLOW PROTOCOL

Open Flow is an open network standard protocol, which is used to control traffic flow among network devices such as routers and switches. This protocol explains the API between Controller and an Open Flow switch; also, it helps to send messages between the programmable controller and a switch within SDN architecture. Further, this protocol permits the programmable controller tell the Open Flow switch to handle incoming packets of data. The main aim of Open Flow is to connect multiple switches together to create a packet flow data, and then manage the overall infrastructure, setting policies and traffic accordingly.

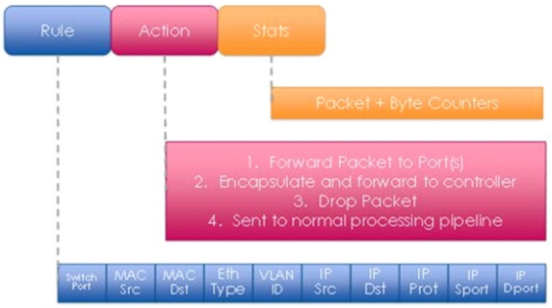


Open Flow switch may be programmed to:

1. identify the packets from an incoming port and categorize them based on the packet header field,
2. the packets are processed in many ways, which include adding, deleting or modifying the header field and,
3. the packets may push or drop to a corresponding outgoing port or to the Open Flow Controller.

The advantage of an open flow environment is that it can communicate with any devices of an SDN Controller that must support the Open Flow protocol. The centralized controller directs switches as to where to forward the incoming packets by inject-flow rules in switch-flow-tables. The Open Flow specification is applied between controller and switch to understand the Open Flow messages.

The Open Flow message contains three parts:



Match Patterns/ Rule: Rule to be constructed from twelve different packet header fields, with various bit widths. The value is computed by the switch based on IP addresses in the packet. In order to predict the same data flow, all the packets make true matching patterns.

Actions: Action part of the corresponding rule tells the switch what types of action must be performed for each packet. The most common actions are: forward, drop, or modify the packets.

Counters: Counter helps to track how many packets match the flow. Matching rule provides a selection criterion. According to the criterion, packets are selected. Actions are

what are to bedone to the selected packets. And counters are the statistic keepers**.**

Open Flow Protocol Specification

The Open Flow Protocol Specifications are available in different versions. A list of updates was added in every version. The first version of the Open Flow protocol (Open Flow 1.0.0) was released in December 2008[1]. This Specification supports 12 header files and payload fields. When a packet arrives at a particular switch, it checks with the flow table, and one or more header fields of the packet may match, and based on priority action will take place.

In the next version Open Flow 1.1.0 specification, a switch has many flow tables and a group table, where Open Flow version 1.0.0 has a single flow table. The table below shows the core components of Open Flow switch and those highlighted are the added attributes of the open flow 1.1.0 version [7]. The packets arrive at the switch that changed as more flow tables are available in the switch. Then the flow tables are connected to one another through the switch, and this process is termed as pipeline processing. When the switch receives the packet, first it will look for the flow table to check whether it matches with the flow entry. If it does, then the packet will be processed, and it may be passed to another flow table if its flow points to another table. This process will continue until the flow entry does not point to other flow tables in the switch. There is a special type of table called group table, which is mainly designed to do operations which are common for multiple flows [1]. In this group table multipath and link aggregation have been introduced. In this specification, instead of action instructions are introduced. These instructions are highly complex and include modifying a data packet, and updating the metadata or action set.

In Dec 2011, The Open Flow specification version 1.2 was released and it included a few major features. Mainly this specification supportedIPv6 addressing, and matching was done using the IPv6 source and destination addresses[1]. This open flow version 1.2 supported the switches to connect simultaneously and maintain the connection state with multi-controller. Therefore, the switches could communicate with one another throughout the connection. The multiple controllers provided reliable, robust service and achieved load balancing.

Ingress Port
Ether src
Ether dst
Ether type
VLAN id
VLAN priority CoS
IP src
IP dst
IP Proto
IP ToS bits
TCP/UDP src port
TCP/UDP dst port

Ingress port
Metadata
Ether src
Ether dst
Ether type
VLAN id
VLAN priority
MPLS label
MPLS EXP traffic class
IPv4 src
IPv4 dst
IPv4 proto / ARP opcode
IPv4 ToS bits
TCP/UDP/SCTP src port. ICMP Type
TCP/UDP/SCTP dst port. ICMP Code

The Open Flow specification 1.3 was released in June 2012[1]. It is possible that the packet flow rate could be measured and controlled by using flow meters. The auxiliary connection state was maintained in switch and controller. The important feature was that cookies and duration field could be added to the message packets sent between switch and controller.

A secure channel is to establish open flow message to be sent from the controller to switch and vice versa. The open flow protocol defines the structure of the open flow

message and it should be understood and created by both switch and controller. The Open Flow protocol is part of the Open Flow specification and it can be used for both Open Flow control plane and Open Flow switch.

4. CONCLUSION

Open Flow is a promising technology to be made use of for high end functionality in programmable networks. Though different versions of Open Flow specification are available, open Flow 1.0.0 is the most widely used specification.

References

[1] [1] A. Lara, A. Kolasani and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 493-512, First Quarter 2014.

[2] Wenfeng Xia, Yonggang Wen, Senior Member, IEEE, Chuan Heng Foh, Senior Member, IEEE, Dusit Niyato, Member, IEEE, and Haiyong Xie, Member, IEEE, "A Survey on Software-Defined Networking" IEEE Communication Surveys & Tutorials, Vol. 17, No. 1, First Quarter 2015 27.

[3] Bruno Astuto A. Nunes ; Marc Mendonca ; Xuan-Nam Nguyen ; Katia Obracz "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks"Publisher: IEEE.

[4] Diego Kreutz, Member, IEEE, Fernando M. V. Ramos, Member, IEEE, Paulo Verissimo, Fellow, IEEE, Christian Esteve Rothenberg, Member, IEEE, Siamak Azodolmolky, Senior Member, IEEE, and Steve Uhlig, Member, IEEE, "Software-Defined Networking: A Comprehensive Survey".

[5] Shailendra Mishra, Mohammed Abdul Rahman AlShehri, "Software Defined Networking: Research Issues, Challenges and Opportunities", Indian Journal of Science and Technology, Vol 10(29), August 2017 ISSN: 0974-6846 ISSN (Online) : 0974-5645.

[6] Yili Gong, Wei Huang, Wenjie Wang, Yingchun Lei, "A survey on software defined networking and its applications",Frontiers of Computer Science, ISSN: 2095-2228 (Print) 2095-2236.

[7] J Hemagowri & Dr B Firdaus Begam, "A Study on Software Defined Networking", Journal of Multidimensional Research and Review (JMRR), Vol.1, Iss.1, pp.35-41, 2019

[8] Braun, Wolfgang & Menth, Michael. (2014). Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices. Future Internet. 6. 302-336. 10.3390/fi6020302.