Analysis of the New Features of OpenFlow 1.4

Tiantian Ren
East China Normal University
National Engineering Research Center for Broadband Networks & Applications
Shanghai, China
rentian0619@126.com

Yanwei Xu
National Engineering Research Center for Broadband Networks & Applications
Shanghai, China
ywxu@bnc.org.cn

***Abstract*—Software Defined Networking (SDN) separates the control plane from the data plane within the network, allowing the intelligence and state of the network to be managed centrally while abstracting the complexity of the underlying physical network. Among the three layers architecture of SDN, the southbound interface is very important since it is used for connecting the control layer and the infrastructure layer. The OpenFlow protocol is the first standard communication interface defined as a southbound interface. This article introduces the new features of the latest version of OpenFlow released on Oct. 15, 2013, and discusses the possible applications that can benefit from these new features.**

Keywords-SDN; OpenFlow; bundles; multi-switch; multi-controller; optical port properties

I. INTRODUCTION

Software Defined Network (SDN), an emerging paradigm in computer networking that allows a logically centralized software program to control the behavior of an entire network, provides a new architecture opportunity for network security [1]. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The SDN architecture is directly programmable, agile to meet changing needs, centrally managed programmatically configured, open standards-based and vendor-neutral, which make it as a revolutionary progress to the network technology and can support many innovative applications [2].

As shown in Fig.1, there are three layers defined in the SDN framework: application layer, control layer and infrastructure layer [3]. The application layer running upon the control layer includes various business applications, such as, firewall, VLAN and many others. The control layer, which works as a centralized Control Center of the whole network, determines the activities of all the network devices in the infrastructure layer. The Infrastructure Layer, which can be also called forwarding layer including a large number of OpenFlow switches mainly responsible for data processing, forwarding and collecting the status. The Southbound interface is used for connecting control layer and infrastructure layer, which reports the status and changes of the infrastructure layer to the control layer and sends the instructions of the control layer to the forwarding layer.

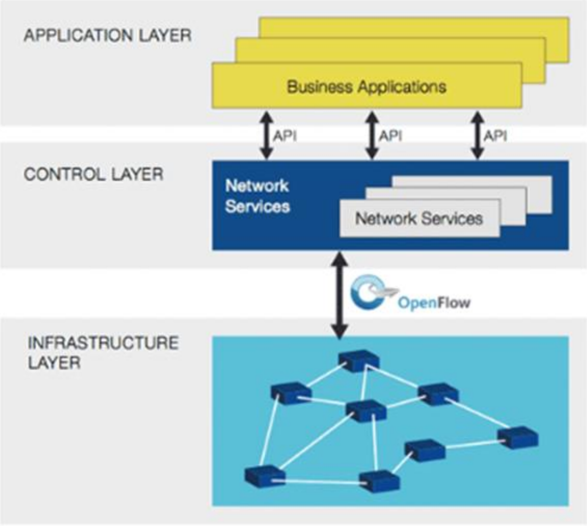

Figure 1: Framework of SDN

The OpenFlow protocol is the first standard communication interface defined as a southbound interface, which is the first SDN standard and a vital element of an open software-defined network architecture. In OpenFlow, there are three types of messages defined for transmitting information between the switches and the control layer. OpenFlow was firstly proposed by Nick McKeown in April 2008[4]. After the Open Network Foundation (ONF) (a non-profit, user-driven organization dedicated to the promotion and adoption of SDN through open standards development) was founded in March 2011, OpenFlow has been brought into a rapid growth stage. In recent years, ONF has released several versions of OpenFlow, which introduce many new features and improvements, such as multi flow tables, group and meter tables. Figure 2 describes the history of the growing of OpenFlow. On Oct. 15, 2013, ONF released the latest OpenFlow: version 1.4.0, which improved the extensibility of the protocol and added some new features. This paper introduces the new features of OpenFlow 1.4.0, and discusses the possible applications that can benefit from them. [5]

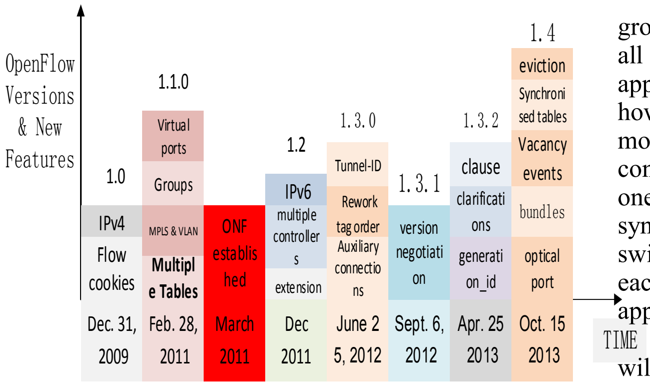

Figure 2: The history of OpenFlow protocol

II. THE NEW FEATURES OF OPENFLOW 1.4.0

As shown in table 1, the new features of OpenFlow 1.4.0 can be categorized into three types: additions, improvements and change. The additions are the completely new features to the OpenFlow protocol; the improvements are the revisions introduced to the existing properties for improving the completeness of the protocol; and the third type is the change of the default TCP port. In the following, we will introduce the new features of the three categories in detail and discusses the potential applications that can benefit from each of them, if possible.

Type	**Features**
Additions	Bundles Optical ports Synchronized tables
Improvements	Flow-removed reason for meter delete New mechanisms support for multi-controller Eviction and vacancy events More descriptive reasons for packet-in PBB UCA header field New error codes
Change	Change default TCP port to 6653

Table 1: The New Features of OpenFlow 1.4.0

A. Additions

There are three new additional features in OpenFlow 1.4.0: bundles, synchronized table, and optical port properties. Bundles and synchronized table can highly facilitate the data synchronization of multi switches, which is very important in, such as switch backuping and switch clusters. And the addition of supporting the optical ports enables OpenFlow to be adopted in situations involving fiber channels.

1) Bundles

The Bundles mechanism is added by OpenFlow 1.4.0 to "bundle" a group of OpenFlow messages as a single operation, which enables the quasi-atomic application of related changes, and to better synchronizes changes across a series of switches [6].

The bundles can realize two functions. The first one is grouping the related state modifications on a switch so that all modifications are applied together or that none of them is applied, that is to say, if any one fails, all of them fail, however, only if all of them are applied successfully, the modifications can be applied successfully. It equals to converting several operations into an atomic operation on one OpenFlow switch. The second function is better synchronizing the changes across a set of OpenFlow switches. The bundles can be prepared and pre-validated on each switch, then will be applied at the same time. It can be applied on the multi-switch.

When a controller wants to add bundles to a switch, it will send an OFPBCT_OPEN_REQUEST message to open a bundle in a specified switch, then adds messages to the bundle one by one. If the Nth message fails, the switch will inform the controller with an error reply message, then the controller will control the switch to discard the bundle including all the messages in it. Under this mechanism, it is more convenient for the messages to rollback, while a problem which cannot be neglected is that once an error appears, all the transmitted messages become invalid. If the amount is large, it creates so much system overhead that the utilization ratio of the channel is reduced. Based on this case, considering that the bundles can be applied in various scopes and environments, it is more reasonable to propose a maximum quantity of the messages in a bundle as a reference value without making a strict standard. What's more, if a fault tolerance mechanism for a bundle can be added to the switch, the efficiency of the bundle will be improved substantially. At last, version 1.4 mentioned that the type of the messages can be added to a bundle is limited, but this protocol did not say the details.

2) Synchronized Tables

Flows tables can be synchronized bidirectionally or unidirectionally. For any synchronized table, a table property describes the source flow table it synchronizes from. If flow table synchronization is bidirectional, the changes done by the controller on the synchronized flow entry need to be reflected on the source flow entry [6].

Synchronized table is expressed using a new property in the table feature, OFPTFPT_TABLE_SYNC_FROM. It defines the synchronization abstraction between the two flow tables; however, it does not define and express the flow entry transformation between the flow tables [6]. The translation between those synchronized flow entries is not specified by the OpenFlow protocol and depends on the switch implementation and configuration.

Many switches can perform multiple lookups on the same lookup data. For example, a standard Ethernet learning table performs a learning lookup and a forwarding lookup on the same set of MAC addresses. The synchronized table feature enables to represent those constructs as a set of two tables which data is synchronized.

3) Optical Port Properties

In version 1.4, many parts of the protocol have been retrofitted with TLV structures for improved extensibility. This TLV work affected many areas of the protocol. New TLVs have been added in previously fixed structures in the form of properties at the end of the structure. For example, in port structures, add port description properties, add port mod properties and add port stats properties. In some areas, the existing TLVs have been changed to use the common property TLV format. For example, Set-async structures convert set-async-config to TLVs and add set-async experimenter property. Thanks to the additional extensibility, the protocol is able to provide a much easier way to add new features in the future, and also greatly extend the Experimenter Extension API.

Among port structures, a new set of port properties add support for Optical ports, they include fields to configure and monitor the transmission and receive frequency of a laser, as well as its power. Those new properties can be used to configure and monitor either Ethernet optical port or optical ports on circuit switches. With the optical port, Fiber Channel over Ethernet (FCoE) can be applied in SDN, which is a computer network technology that encapsulates Fiber Channel frames over Ethernet networks. This allows Fiber Channel to use 10 Gigabit Ethernet networks (or higher speeds) while preserving the Fiber Channel protocol.

The addition of supporting the optical ports enables SDN to be adopted in technologies involving fiber channels, such as Fiber Channel over Ethernet (FCoE). FCoE is a computer network technology that encapsulates Fiber Channel frames over Ethernet networks, which allows Fiber Channel to use 10 Gigabit Ethernet networks (or higher speeds) while preserving the Fiber Channel protocol [7]. However, today's FCoE technologies all are proprietary technologies [8], which prevents it's widely adoption and the inter-communication of systems of different vendors. SDN provides a great possibility of developing FCoE systems based on ordinary network devices, with low spending and well interoperability.

B. Improvements

1）Flow-Removed Reason for Meter Delete

Flow entries are removed from flow tables in three ways, either at the request of the controller, via the optional switch eviction mechanism, or via the switch flow expiry or invalid mechanism. When a meter is deleted on the switch, all the flow entries that use that meter are removed because of third way.

A meter table consists of meter entries, defining per flow meters. Per flow meters enable OpenFlow to implement various simple QoS operations, such as rate-limiting, and can be combined with per-port queues to implement complex QoS frameworks. The meter is an "Optional Instruction" type in a flow entry. The meter measures and controls the rate of the aggregate of all flow entries to which it is attached. Multiple meters can be used in the same table, but in an exclusive way (disjoint set of flow entries). Multiple meters can be used on the same set of packets by using them in successive flow tables.

OpenFlow 1.4.0 adds a new reason value OFPRR_METER_DELETE for the ofp_flow_removed message to denote that the flow entry was removed as a result of a meter deletion. When a meter is deleted on the switch, all the flow entries using that meter are removed, then the switch will inform the corresponding controller of it.

2）Improvements for multi-controller

Since OpenFlow 1.2.0, a multi-controller scheme[9] is introduced for a switch to be controlled by multiple controllers, which can extremely increase the throughput of the control layer compared to single controller since different controllers can be in different roles and responsible for processing different messages of the switch. Therefore, state synchronization mechanism is crucial [10]. OpenFlow 1.4.0 brings in following three new features to improve the synchronization among controllers.

a) Flow Monitoring

The OFPMP_FLOW_MONITOR multipart type allows a controller to manage several flow monitors so that it can keep track of changes to the flow tables. Each flow monitor includes a table id and a match pattern that defines the subset monitored. When any flow entry is added, modified or removed in one of the subset defined by a flow monitor, an event is sent to the controller to inform it of the change. This way, when a switch controlled by several controllers, only if a controller has set a flow monitor on the switch, any operation done on the flow table included in the monitor by any controller will be learned by the controller. In a multi-controller deployment, flow monitoring enables a controller to be aware of changes made to the flow tables by other controllers in real time and then realizes the synchronization of the controllers.

A controller can send a multipart request ofp_flow_monitor_request to set flow monitors on the switch. When any of the table entry was updated, the switch will send an update event with full details using ofp_flow update_full or abbreviated using ofp_flow_update_abbrev to the controller.

b) The notifications of group and meter change

Group and Meter change notifications allow a controller to monitor in real time the changes to the group table or meter table done by other controllers. This mechanism provide a support for the synchronization among multi-controller. The "group-mod" and "meter-mod" requests are simple encapsulated in an OFPT_REQUESTFORWARD asyn-chronous message sent to other controllers. Those notifications are enabled and disabled via the "set-async-config" message. [6]

c) Role status events

Version 1.2 of the specification added the ability for a controller to set its role in a multi-controller environment. When the controller wants to change its role, it uses the OFPT_ROLE_REQUEST message. If the role value is OFPCR_ROLE_MASTER, all other controllers whose role was OFPCR_ROLE_MASTER are changed to OFPCR_ROLE_SLAVE. In version 1.4, if the switch must change the role of another controller from

OFPCR_ROLE_MASTER to OFPCR_ROLE_SLAVE, it must send an OFPT_ROLE_STATUS message to the controller so that it can be informed about the change on its role [6]. The event supports the messages' synchronization between multiple controllers.

3）Eviction and vacancy events

Most flow tables have a finite capacity. In previous versions of the specification, when a flow table is full, new flow entries are not inserted in the flow table and an error is returned to the controller. However, reaching that point is pretty problematic, as the controller need time to operate on the flow table and this may cause a disruption of service. In this case, OpenFlow 1.4.0 provides two effective ways to solve the problem.

Eviction adds a mechanism enabling the switch to automatically eliminate entries of lower importance to make space for newer entries. This enables to smoother degradation of behavior when the table is full. Flow entry eviction is an optional feature, and the mechanism used to select which flow entry to be evicted is defined by the switch or depends on flow entry parameters, resource mappings in the switch and other internal switch constraints.

Vacancy events add a mechanism enabling the controller to get an early warning based on a capacity threshold chosen by the controller. This allows the controller to react in advance and avoid getting the table full.

4）More descriptive reasons for packet-in

The OpenFlow pipeline saw extensive changes since 1.0, however, the reason values in the ofp_packet_in messages did not change. As a result, many distinct parts of the pipeline are using the same reason value. Version 1.4 introduces more descriptive reasons, so that the controller can properly distinguish which part of the pipeline redirected the packet to the controller.

The main change is that the "output action" reason OFPR_ACTION is effectively split into four reasons, "apply-action", "action-set", "group bucket" and "packet-out", representing the four distinct contexts where this action is used. The "no match" reason OFPR_NO_MATCH is renamed to properly reflect the fact that it is generated by the table miss flow entry.

5）PBB UCA header field

A new OXM field OFPXMT_OFB_PBB_UCA which supports IEEE 802.1ah UCA has been added to match the "use customer address" header field from the PBB header. For a packet with a PBB header, this is the UCA (Use Customer Address) from the outermost service tag. The UCA field is in the first PBB service instance tag. This is a better application of PBB (Provider Backbone Bridge) technology.

6）New error codes

The OpenFlow 1.4.0 adds five new kinds of error codes: OFPBIC_DUP_INST, OFPET_ASYNC_CONFIG_FAILE, OFPFMFC_BAD_PRIORITY, OFPBRC_MULTIPART_REQUEST_TIMEOUT, and OFPBRC_MULTIPART_REPLY_TIMEOUT.

The OpenFlow specification defines the instructions included in a flow entry as a set, and that an instruction cannot be duplicated in that set. The new error code, OFPBIC_DUP_INST, enables the switch to properly inform the controller when flow entries contain duplicate instructions.

The OFPT_GET_ASYNC_REQUEST feature was introduced in version 1.3.0. There was no error messages defined for that features, however it is possible for this request to fail. The new error type, OFPET_ASYNC_CONFIG_FAILED, avoids the problem effectively.

Some switches may have restrictions on the priorities that can be used in a table. If a switch enforces some "longest prefix match" rules in a table, the priority is required to be related to the mask. The new error code, OFPFMFC_BAD_PRIORITY, enables the switch to properly inform the controller when it happens.

Multipart request and replies are encoded as a sequence of messages. This version of the specification defines minimum timeout (100ms) and error code (ORPBRC_MULTIPART_REQUEST_TIMEOUT) for unterminated multipart request sequences and minimum timeout (1s) and error code (OFPBRC _MULTIPART_REPLY_TIMEOUT) for unterminated multipart reply sequences.

C. Change Default TCP port to 6653

IANA allocated to ONF the TCP port number 6653 to be used by the OpenFlow switch protocol. All uses of the previous port numbers, 6633 and 976, should be discontinued. The OpenFlow switches and OpenFlow controllers must use 6653 by default (when not using a user specified port number) [6].

III. CONCLUSION

The new features in version 1.4 focus on the control layer so that the controller can be informed of the switch status and flow table status more accurately based on flow monitoring, flow-removed reason for meter delete message, group and meter change notifications. Moreover, in a multi-controller scheme, controllers can be synchronized based on flow monitoring, group and meter change notifications, role status events. The addition of the optical port provides more support for new technologies.

In this article we give a comprehensive survey of SDN and the OpenFlow, and then discuss the relationship between them. The article focuses on the new features of the OpenFlow 1.4.0, which aredescribed and discussed in details in the second section. The OpenFlow protocol 1.4 optimizes the previous one and can support more technologies so that SDN will be applied in a wider scope.

ACKNOWLEDGMENT

This work is supported by the project Radio and TV Network-based Integration of Innovative Demonstration Network (2011AA01A109) of the National High

Technology Research and Development Program of China (863 plan).

REFERENCES

[1] Software-Defined Networking, http://en.wikipedia.org/wiki/Software-defined_networking

[2] Software-Defined Networking: The New Norm for Networks. White paper. Open Networking Foundation, 2013.

[3] Software-Defined Networking Definition, https://www.opennetworking.org/sdn-resources/sdn-definition

[4] Nick McKeown, Tom Anderson, Hari Balakrishnan, et al., “OpenFlow: enabling innovation in campus networks”, Computer Communication Review (CCR), vol. 38, pp. 69-74, 2008

[5] ONF Overview, Open Networking Foundation, 2013

[6] openflow-spec-v1.4.0, Open Networking Foundation (ONF), Oct. 15, 2013.

[7] Fibre Channel: Backbone - 5 revision 2.00. American National Standard for Information Technology International Committee for Information Technology Standards Technical Group T11. June 4, 2009

[8] Fibre Channel over Ethernet,

http://en.wikipedia.org/wiki/Fibre_Channel_over_Ethernet, 2013.

[9] Amin Tootoonchian et al. On controller performance in software-defined networks. In HotICE, 2012.

[10] Andreas Voellmy et al. Scalable software defined network controllers. In SIGCOMM, 2012.