Computer Networks 102 (2016) 1–19

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A technique for full flow virtualization of multi-tenant OpenFlow networks

Hiroaki Yamanaka*, Eiji Kawai, Shinji Shimojo

National Institute of Information and Communications Technology, Japan

ARTICLE INFO

Article history:
Received 8 May 2015
Revised 24 February 2016
Accepted 26 February 2016
Available online 11 March 2016

Keywords:
Software-defined networking
OpenFlow
Multi-tenancy
Flow space virtualization
Prefix and wildcard matching

ABSTRACT

OpenFlow provides new capabilities for wide-area network applications such as traffic engineering and security applications with many implementations having been proposed. OpenFlow allows flows to be described using arbitrary values or wildcards in the header fields. This functionality enables a controller to manage flow entries using only a limited number of the entries. In particular, in the case of wide-area networks, the sharing of the infrastructure among tenants (e.g., application service providers and their customers) is necessary, owing to the high capital outlay and operational costs. A technique for multi-tenancy creates logically isolated virtual OpenFlow networks so that multiple tenants can handle flows independently on a single physical OpenFlow network. Of the proposed techniques, OpenVirteX virtualizes the header address space (called the flow space) for tenants. This encourages tenants to participate because, differing from some other implementations, negotiations by tenants to divide the flow space are not required. However, OpenVirteX forces tenant controllers to set either exact Media Access Control (MAC) or Internet Protocol (IP) address in the matching fields because matching fields in the physical OpenFlow switches must have either exact MAC or IP addresses to isolate flows of the different tenants. Although prefix or wildcard matching is widely used to handle a large number of flows with a small number of flow setting by an OpenFlow controller, there is no available virtualization technique that enables prefix or wildcard matching with flow space virtualization. In this paper, we propose an OpenFlow network virtualization technique that allows prefix or wildcard matching with flow space virtualization. The proposed technique translates flow entries set by tenant controllers using prefix or wildcard matching to exact-match flow entries through lazy evaluation of the actual data packets. Evaluation results obtained using a prototype of the proposed technique show that the overhead of virtualization has an acceptable impact.

1. Introduction

Software-defined networking (SDN) provides new norms for a variety of application areas such as traffic engineering and security in wide-area networks (WANs). OpenFlow [1] is the most popular SDN implementation with many applications using OpenFlow in WAN. For example, B4 [2] and SWAN [3] implement centralized traffic engineering mechanisms with OpenFlow, using a multi-path forwarding technique to balance the traffic according to available network capacity and application demands. The main features of OpenFlow are a global view of the network and programmability of flow control. In OpenFlow, flows can be defined by a set of values, prefixes, or wildcard header fields in layers 2–4 and the

* Corresponding author. Tel.: +81 335106271.
E-mail addresses: hyamanaka@nict.go.jp (H. Yamanaka), eiji-ka@nict.go.jp (E. Kawai), sshinji@nict.go.jp (S. Shimojo).

http://dx.doi.org/10.1016/j.comnet.2016.02.029
1389-1286/

port number in layer 1, with all layers comprising the so-called “flow space”.

Especially for a wide-area OpenFlow infrastructure, sharing the infrastructure with multiple application service providers is necessary owing to the substantial capital and operational costs of such infrastructure. Multi-tenancy improves the cost effectiveness of the wide-area OpenFlow network infrastructure. In this paper, a tenant refers to a group consisting of an OpenFlow controller and some end-hosts, while multi-tenancy implies that multiple tenants are accommodated concurrently on a single physical OpenFlow network. For each tenant, a virtual OpenFlow network that is logically isolated from other virtual OpenFlow networks is created. Each tenant can use its virtual OpenFlow network as if it were an exclusive OpenFlow network. Thus, multiple tenants can run their application services concurrently with independent flow controls on a single physical OpenFlow network.

The original architecture of OpenFlow assumes that a single controller handles all OpenFlow switches in the network, and thus, OpenFlow itself does not support multi-tenancy. FlowVisor [4] realizes multi-tenancy by dividing the flow space among individual tenants. The FlowVisor proxy relays all OpenFlow protocol messages between the tenant controllers and the OpenFlow switches and manages the division of the flow space for the tenants. When relaying a message from the controller, the proxy rejects the message if the flow definition in the message is not in the divided flow space for the tenant. Through this access control, flows of virtual OpenFlow networks are isolated in physical OpenFlow networks. Because the FlowVisor proxy simply manages the division of the flow space and filters messages, it can realize multi-tenancy in a lightweight implementation. However, with FlowVisor, different tenants cannot use an overlapping flow space and hence need to negotiate with each other to divide the flow space. This limitation becomes a barrier to accepting many tenants although accepting many tenants improves the cost effectiveness of wide-area OpenFlow networks.

OpenVirteX [5] realizes multi-tenancy with flow space virtualization. The OpenVirteX proxy interprets OpenFlow protocol messages and data packets sent between the OpenFlow switches, and the tenant controllers and end-hosts. Flow space virtualization enables tenants to use overlapping header values in data packet headers and matching fields of flow entries. No negotiations by tenants are required to divide the flow space. This feature contributes to improve the cost-effectiveness of wide-area OpenFlow networks because it makes participation easy for tenants with virtual OpenFlow networks. To virtualize the flow space, the proxy interprets Media Access Control (MAC) and Internet Protocol (IP) addresses in OpenFlow protocol messages and data packets. We refer to the MAC or IP address seen by the tenant controllers or end-hosts as the virtual MAC or IP address, and that seen by the physical OpenFlow switches as the physical MAC or IP address. Even when the virtual MAC or IP addresses for different tenants overlap, different values are allocated as the corresponding physical MAC or IP addresses to isolate the flows of different tenants. When relaying a message that sets a flow entry of the L2 rule (i.e., the matching field of the MAC address), the proxy translates the virtual MAC address into the physical one. If a relayed message is to set a flow entry of the L3 rule (i.e., the matching field of the IP address), the proxy translates the virtual IP address into the physical one. To translate data packet headers, the OpenVirteX proxy modifies the OpenFlow protocol messages from the tenant controllers handling data packets at the switch ports connected to end-hosts. For data packets from the end-host, the proxy modifies the message by replacing the virtual MAC and IP addresses with the physical ones. For data packets sent to the end-host, the proxy modifies the message by replacing the physical MAC and IP addresses with the virtual ones.

While virtualizing the flow space, OpenVirteX prevents tenant controllers from setting a flow entry whose MAC and IP addresses in the matching fields are both wildcards or prefixes. The exact values of physical MAC or IP addresses in the matching fields are needed to isolate flows of different tenants in the physical OpenFlow network and to correspond to the unique virtual MAC and IP addresses. Hence, the tenant controllers are forced to set flow entries with either the exact MAC or IP address in the matching fields in the OpenVirteX mechanism. An “L3 rule” of a wildcard or prefix IP address (i.e., wildcard MAC address with a wildcard or prefix IP address in the matching fields) is not allowed for the tenant controller. An “L2 rule” of the wildcard or prefix MAC address (i.e., wildcard IP address with a wildcard or prefix MAC address in the matching fields) is also not allowed.

This limitation prevents tenant controllers from offloading (i.e., reducing the setting of flow entries by a tenant controller) when handling a large number of flows. A prefix or wildcard of both MAC and IP addresses in matching fields is widely used for this purpose. Wang et al. [6] proposed the flow aggregation method using IP prefix matching to control server load-balancing without setting a large number of flow entries by the controller. In PortLand [7], flows are defined using prefix MAC addresses to reduce the number of flow entries in large-scale datacenters. Pallete [8] compresses the flow entries by aggressively using wildcard matching while not requiring either exact MAC or IP addresses in the matching fields. The controllers in these examples are not applicable to the OpenVirteX mechanism. An OpenFlow network virtualization technique should enable the functionality of prefix and wildcard matching without any restriction for tenant controllers.

In this paper, we propose an OpenFlow network virtualization technique that enables both the flow space and prefix and wildcard matching to be fully virtualized. In the proposed technique, the MAC address is used to isolate flows of different tenants in the physical OpenFlow network. Although the flow entry must have the exact MAC address in the physical OpenFlow network, our technique enables the tenant controllers to use prefix or wildcard matching fields of MAC addresses in the matching fields. For such matching fields, the proxy determines the exact MAC addresses using the actual data packet headers. The proxy stores the matching fields of the prefix or wildcard MAC addresses set by the tenant controllers. When a data packet arrives at a physical OpenFlow switch, it is