

以 OpenFlow 於骨幹上實作動態虛擬網路之供裝

胡仁維 曾惠敏 劉德隆

國家實驗研究院國家高速網路與計算中心
{hujw, n00hmt00, tlliu}@narlabs.org.tw

摘要

網路網路的應用越來越多樣化，但要實現所有的應用，必須仰賴點對點間的連線建立，layer1 層光通道雖然可配置專屬的頻寬通道，但成本高與其頻寬無法有效被運用成了此技術的缺點，因此使用 layer2 層技術來共用光通道是目前可行的技術與趨勢，然而不同設備的設定方式差異，使得系統整合變得複雜，SDN 中被廣為運用的 OpenFlow 協定，讓網路設備的管理有了一致性且對於上層系統的整合也變得更加容易。本論文是基於 OpenVirteX 之架構所實作的一個可以建立跨地理區域專屬網路之管理系統，其支援大部分的硬體式 OpenFlow 交換器，此外，透過實驗證明能控制網路交換器中 flow 數的成長速度，且符合對上層系統的透明性。

關鍵詞：SDN；虛擬網路；OpenFlow；OpenVirteX。

1. 前言

網際網路 Internet 已經成為我們生活上不可或缺的技术，創新的應用與迅速可靠的資料傳輸，帶給我們極大的助益，而要實現這些應用皆由點對點的連線建立開始。以網路實體層為例，首先要有實體的線路連線，接下來透過光通道(light path)的設定與建立，才能讓資料利用此連線來進行傳輸。雖然光通道的優勢，是保證配置一個專屬通道，不過也因為如此，失去此傳輸通道的彈性，成本價格的昂貴也是另一個因素，因此大多數的網路服務提供者，會運用上層的網路技術來分享光通道，使其可讓多個使用者或者應用共享頻寬資源，目前在區域網路內，通常利用設定 VLAN 來達到此需求；若跨地理區域則由 VPLS (Virtual Private LAN Service) [1]等技術來實現。不過不管是運用哪種技術，都必須針對影響的網路設備逐台設定，而且有可能每台網路設備的設定方式不同，對網路管理人員造成不便利性，亦可能因為設定的錯誤導致網路服務不正常等。

SDN (Software Defined Network) 可以提供解決此問題的技術之一，網路設備是由 data plane 與 control plane 兩大元件所構成，前者會根據設備內部的 forwarding table 來對所接收到的 flow 進行對應的處理動作，如：往指定的埠傳送或丟棄封包等；而後者，則如同大腦一樣，會依照所定義的邏輯對 flow 進行特定地處理。在過去，網路設備就如

同一個黑箱，我們無法對設備有任何控制的能力，只能仰賴設備商所能提供的功能來操作使用，不過在 Stanford 大學的一個計畫打破了此限制，此計畫提出了新的網路設備溝通協定(即 OpenFlow [2])，並將 control plane 由原本存在於網路硬體設備中抽離出來，使用者可以藉由裝有控制器的伺服器，透過 TCP 或 SSL 連線至網路設備，利用 OpenFlow 協定與網路設備進行溝通與控制，讓所有網路設備有統一的協定來管理，此外，更可容易地與上層應用程式進行整合，讓許多應用得以實現。

OpenVirteX [3]是建立於 OpenFlow 協定的一個虛擬網路系統，本論文是基於此 OpenVirteX 的架構實作了一個可以建立跨地理區域專屬網路之管理系統，此系統會使用 OpenVirteX 是因為其可給定不同使用者一個專屬的虛擬網路，此特性剛好符合我們跨地理區域的專屬網路需求，不過由於 OpenVirteX 目前是整合與應用於軟體式的網路交換器，因此對於硬體式網路交換器支援並不完善，因此在此篇論文的貢獻是讓我們所提出的虛擬網路動態供裝系統，能支援大部分的硬體式 OpenFlow 交換器；此外，此系統也能讓網路交換器控制儲存 flow 的空間，不會隨著連接的設備增多而快速成長。

2. 文獻探討

此章節將針對現有可以達成網路虛擬化的系統進行探討，並在接下來的內容簡單地介紹他們實作之概念以及優缺點。

2.1 FlowVisor

FlowVisor [4]是一個特殊的 OpenFlow 控制器，與大部分的 OpenFlow 控制器最大不同之處，在於其允許多個控制器同時共用底層的網路設備資源。FlowVisor 定義了一個稱為 slice 的資料結構，此 slice 是由 OpenFlow 所定義的欄位組合而成(如：網路埠、MAC 位址、IP 位址與 TCP/UDP 埠等)，其能把不同的 slice 指定給不同的使用者，因此利用 FlowVisor 即能支援多個使用者共用相同的底層網路資源，達到所謂的網路虛擬化。而 FlowVisor 唯一的限制就是每個 slice 的定義不能夠有任何一個欄位資訊重疊。

雖然 FlowVisor 可以保證每個使用者都有其自己的網路資源，且不會被其他人所影響，但

FlowVisor 限制了使用 slice 的應用程式必須與其所定義的欄位資訊穩合，這會對開發應用程式造成不小的限制；此外，隨著 FlowVisor 所管理的網路設備越來越多與複雜，網路管理人員要避免 slice 間的重疊，也會變得越來越困難，必須有額外的條件檢查。

2.2 OpenVirteX

FlowVisor 除了上述的問題外，也隨著計劃的結束，目前已沒有開發人員對程式碼進行維護，取而代之的，是由同一個開發團隊所發展的另一個開放原始碼工具 OpenVirteX，此軟體被認為是下一個版本的 FlowVisor，其運用位址的轉換實現了允許不同使用者，定義相同的實驗 IP 位址，對於建立虛擬網路更有其彈性，也擺脫 FlowVisor 對使用者端應用程式的不便性。

OpenVirteX 有兩個主要的轉換程序，一個稱為 virtualize；另一個則是 devirtualize，前者是用來當底層的網路，有封包要送至所連接的 controller 時，位於中間的 OpenVirteX 會透過此 virtualize 的程序，根據條件送至指定的使用者 controller，達成所謂的 isolation。而相反地，若從上層的使用者程式，要寫入 flow entry 至底層的網路設備，此時 OpenVirteX 就會使用 devirtualize 的程序來處理，保證每個使用者的虛擬網路環境不會被其他人所影響。

不過目前的 OpenVirteX 版本，只支援能夠修改 layer3 欄位的 OpenFlow 交換器，由於 OpenVirteX 主要是著重在雲端的虛擬網路環境，以軟體式的網路交換器應都有提供此功能，但若佈署於硬體式的網路交換器，將有其困難性；此外，OpenVirteX 是透過 MAC 位址與使用者資訊來作為轉換條件，因此會根據連接的設備產生 flow entry 資訊，當在其上傳送的設備增多，會造成 flow 數的增加。

3. 系統架構

在這個章節，將介紹我們所開發的網路虛擬化系統，此系統是基於 OpenVirteX 所發展而來，其為一個三層式的架構，如圖 1 所示，由下而上將系統分成網路資源層、虛擬化層與系統管理層。首先，位於最底層的網路資源層是用來提供給使用者運用的網路資源，也就是我們所管理的網路設備，如交換器、路由器等；第二層為虛擬化層，也是整個系統最核心的部份，其負責在虛擬網路與實體網路間作轉換，讓每個虛擬網路間不會相互影響，達到所謂使用者的 isolation；而位於最上層的系統管理層則是用來提供給系統管理人員操作與維護之用，包含了提供給使用者的虛擬主機群、負責定期執行服務的排程模組等，且也定義介接的程式介面能方便與上層的應用程式進行整合之用。

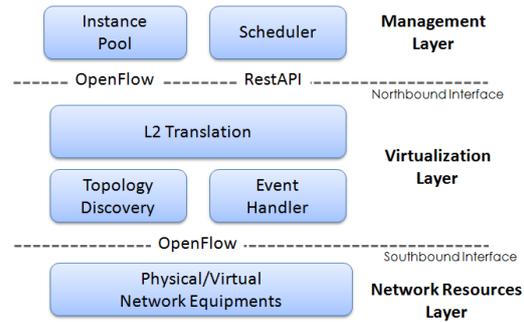


圖 1 系統架構

3.1 網路資源層

此層是負責提供建立點對點連線時，所需要被使用到的網路資源，目前，除了可使用硬體式的網路交換器外，軟體式的網路交換器也包含在其中，不管是硬體式或軟體式的交換器，唯一的限制條件就是必須要支援 OpenFlow 協定。隨著雲端的應用與發展，軟體式的 OpenFlow 交換器也逐漸被廣泛地與 hypervisor 進行整合，目前較普遍被使用有 Open vSwitch [5]、Lagopus [6] 等，此軟體式的 OpenFlow 交換器，優點為支援功能完整，但缺乏的就是可以與其介接的埠數較硬體式交換器來的不足，且必須仰賴所安裝的機器 CPU 來處理進出的封包與控制之邏輯，也就是說有可能會有效能上的問題產生。

相對於軟體式的 OpenFlow 交換器，即是所位的硬體式網路交換器，隨著 SDN 的發展，目前大部分的網路設備製造商幾乎都已支援 OpenFlow 協定，但由於 OpenFlow 協定定義了不少可選擇性實作的功能，這也造成每家設備商所支援的程度有所差別，也間接的影響上層應用程式整合與開發的困難，現階段支援 OpenFlow 協定的設備，大多是網路交換器，因此對 layer2 的 OpenFlow 功能支援程度也較為完善。

3.2 虛擬化層

此層主要是實現網路虛擬化，也就是讓底層的網路資源如交換器、路由器等能讓多人同時共用。在上一章節有提到，我們的系統是基於 OpenVirteX 所開發出來，雖然 OpenVirteX 已經實現了網路虛擬化，也就是其可以讓每個使用者擁有屬於自己的虛擬網路，但 OpenVirteX 仍有許多問題需要被克服，其中最主要的兩個就是(1)與現有 OpenFlow 網路設備的相容性以及(2)能擁有透明性(Transparency)，完整地支援上層各種網路服務。

為了達成能夠廣泛地支援現有 OpenFlow 網路設備，經過不同廠牌測試的結果(比較表列於第四章)，我們改用 layer2 取代原本 OpenVirteX 利用 layer3 來做位址轉換機制。如圖 2 中，兩個圓點代表分屬兩地的網路設備或 host 端，透過實體線路連接至我們系統所控制的端點交換器，所連接的埠藉

由系統指令與使用者資訊綁定，最後註冊至系統之中並產生使用者的虛擬網路。

當使用者端的封包從指定的埠送入端點交換器，此封包會因為交換器中沒有符合的 Flow 而把它轉送至我們系統中，根據虛擬網路設定時所產生的資訊交由指定的控制器進行封包的處理，此過程稱為 virtualization；而當最上層的控制器決定好封包的處理方式後，會透過 OpenFlow 訊息 FlowMod 或者 PacketOut 往交換器傳遞，此時我們系統再次參考使用者虛擬網路的資訊，將此訊息加上 layer2 的資訊(我們目前是使用 VLAN)，透過 OpenFlow 協定寫入 Flow 至建立此專屬通道所會經過的所有網路交換器中，此過程我們稱為 devirtualization。

若此專屬通道持續有傳輸的流量，在第一筆封包透過系統處理後，接下來將透過網路交換器的硬體來轉送封包，會加快封包處理的速度。為了能讓交換器做更有效地運用，預設系統寫入的 Flow 都會使用 timeout 欄位，當然，若為一個長期性會使用的專屬通道，也可以讓此 Flow 保留在交換器中，增加其傳輸效能。

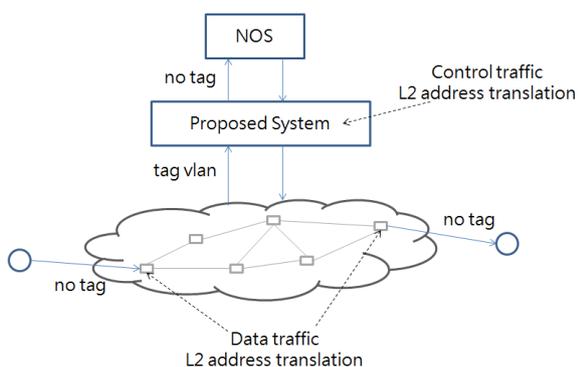


圖 2 Layer2 位址轉換機制示意圖

此小節的第二個部份，是讓系統能達到透明性 (Transparency)，在上述內容有提及，所有由使用者端發出的封包，若沒有符合交換器中任何的 Flow，則會往使用者的控制器傳送，由此機制可看出，不管是 OpenVirteX 或者是我們系統皆不會限制任何型態的封包，但實驗結果發現，唯一的例外就是 LLDP 封包，此 LLDP 是在 OpenFlow 網路裡，用來找出整個網路拓樸之用，由 3.1 網路資源層中的網路設備，皆是符合 OpenFlow 協定，因此若能夠解析整個網路拓樸，也是必須仰賴 LLDP。此外，由於使用者端也有可能是一個 OpenFlow 的網路環境，所以也會有使用者端的 LLDP 的封包在其上傳遞，OpenVirteX 雖有實作解析 LLDP 的訊息，但卻會將使用者端的 LLDP 丟棄，此會造成在使用者端其無法正確地解析出自己的 OpenFlow 網路拓樸，也因此會失去所謂的透明性。

基於上述問題，我們提出的系統在建立虛擬網路時，每個使用者在通道兩端的端點交換器有屬於自己的埠口，因此當我們收到來自特定使用者的

LLDP 封包，就將其轉送至其他端點交換器中屬於相同使用者的埠，藉由此機制，可以讓使用者端的 OpenFlow 網路拓樸能在控制器被解析。

3.3 系統管理層

在系統管理層中，我們使用 OpenStack [7] 的環境來配置虛擬機器群 (Instance Pool)，此是用來建立虛擬網路的控制器 (Controller Instance)。OpenStack 是一個在 Linux 環境下運作的開放源碼，2007年由美國太空總署 (NASA) 及 Rackspace 公司合作研發，做為打造基本的雲端環境，其提供了許多 API 方便讓使用者能自行客制化專屬的服務。

OpenStack 架構上定義了九大基本模組，包括 Horizon、Keystone、OpenStack Networking、Cinder、Nova Compute、Glance、Swift、Heat、Ceilometer 等九大模組，在我們的虛擬機器群環境中，使用到以下五個模組，分別是：

- **Horizon (Dashboard)**：儀表板模組，作為統一管理 OpenStack 上所有模組的一 Web 圖型化管理介面，如新增虛擬主機、配置網路、儲存容量和安全存取控制設定等。
- **Keystone**：身份認證模組，用來作 OpenStack 各種服務的身份認證和存取控管，同時也提供多種驗證方式。
- **OpenStack Networking (Neutron)**：網路模組，提供虛擬網路連接環境，除了基本的靜態或動態 IP 分配外，也支援 SDN 的 OpenFlow 協定等。
- **Nova Compute**：運算模組，提供虛擬化技術，負責虛擬機器的環境佈署、管理和排程。
- **Glance**：映像檔管理模組，建立虛擬機器時使用的映像檔範本。

圖 3 為配置虛擬機器群之架構圖，在我們的虛擬主機群架構中，包含佈署 3 台實體主機，其中 Controller Node 用來控制與分配 Compute Node 上的所有資源，Network Node 為配置虛擬主機的網路位址，Compute Node 是作為新增虛擬主機的角色。在佈署的 Neutron 網路環境中，三台實體主機分別透過管理網路 (Management Network) 建立內部的溝通管道和進行訊息交換。

在圖 3 中，可看到 Network Node 和 Compute Node 之間建立一條 Instance Tunnel，此是作為網路互通的橋梁，該隧道技術是採用 generic routing encapsulation (GRE) 封裝格式，我們在 Network Node 上設定 10.0.1.2 為此通道的網路介面 IP 位址，在 Compute Node 上設定 10.0.1.3 為此通道的另一個網路介面 IP 位址，當有多台 computer node 的資源可提供虛擬機器供裝時，會同時建立多條的 Instance Tunnel，為了能夠存取這些虛擬機器群，我們指定了一段 192.168.1.0/24 的網段作為虛擬機器網路位址，並以 DHCP 的方式於新增一台虛擬機器時隨機

的分配。

我們透過 Glance 建置了映像檔，此映像檔裝有 OpenFlow 控制器，當使用者透過我們的系統提出建立虛擬網路申請時，會利用 OpenStack 的虛擬化技術動態地佈建虛擬機器作為其虛擬網路的控制器。

在此系統管理層，還有一個模組是 Scheduler，有一些定期檢查的排程工作是透過它來處理，例如：檢查使用者所申請的虛擬網路使用期限、在虛擬機器群裡的控制器 VM 是否足夠等工作。

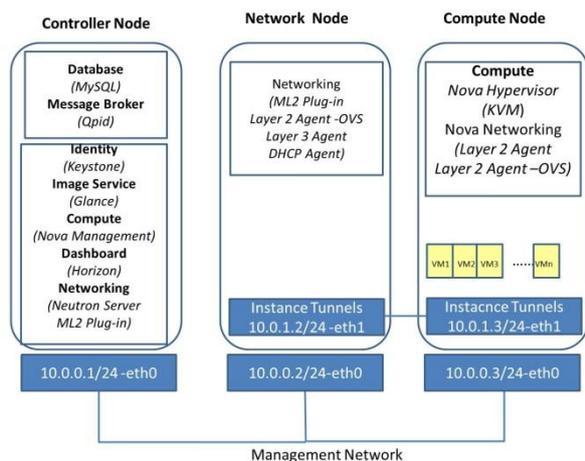


圖 3 虛擬機器群架構圖

4. 實驗與比較

在此章節的開始，首先我們先針對目前所實際測試過的硬體交換器進行相容性驗證。表格 1 列出不同廠牌間對 OpenFlow 欄位修改功能的支援狀況，我們分別比較兩組不同支援的功能，一個是 layer 2 欄位修改(也就是對 MAC 位址、VLAN)；而另一個是針對 layer 3 的欄位修改(也就是 IP 位址)。由表格 1 可看出，目前硬體的交換器中，支援 layer 2 與 layer 3 欄位修改的只有 Pica8，而其他的交換器，在設備出廠的韌體皆無法正確支援 layer3 的欄位修改，但較特別的是 Edge-core，雖然其原生的韌體是無法修改 layer3 欄位，不過此款交換器可以改由 Open vSwitch 作為其韌體，在此情況下，其功能就與 Pica8 相同，也可支援 layer3 的欄位修改。

表 1 各廠牌 OpenFlow 交換器比較表

	Edge core AS4600	HP 3800	Pica8 3297	Brocade 6610	Open vSwitch
支援 layer2	V	▲	V	V	V
支援 layer3	X	X	V	X	V

▲: 功能為軟體實作

對支援修改 layer2 欄位而言，除了 HP 3800 是使用軟體的方式來實作外，其餘的硬體式交換器皆能運用硬體的晶片來達成如 VLAN 與 MAC 位址的修改，這兩者的差別在於效能的好壞。因此整體而言，目前市場上的 OpenFlow 交換器對 layer2 欄位支援程度較佳，所以在設計或開發系統時，應以 layer2 的功能為主，能達到比較完整的相容性。

接下來我們將此篇論文所提出的系統與原生的 OpenVirteX 之間進行比較。我們的測試實驗環境如圖 4 所示，網路交換器部分是由 4 台硬體式的 OpenFlow 交換器與 2 台由伺服器安裝 Open vSwitch 做為軟體式的網路交換器所構成，每台交換器的控制器指向我們的系統，就如同代理伺服器一樣，而我們系統會利用 OpenStack 所產生的虛擬機器，在其上安裝 Floodlight 做為每個虛擬網路的控制器之用。

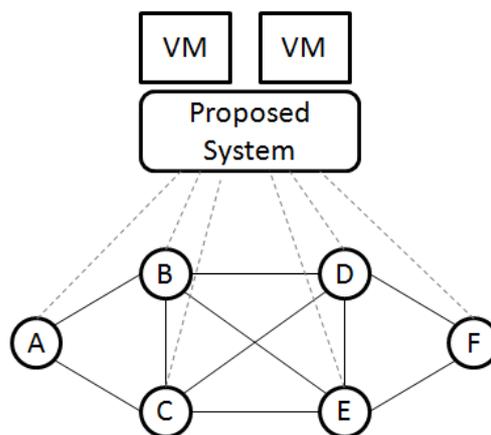


圖 4 實驗拓樸

此測試架構下的第一個實驗，我們假設有 20 位使用者透過此系統申請專屬通道的服務，透過增加每個使用者在專屬通道上所擁有的設備來觀察交換器上 Flow 的成長速度。從圖 5 可以看出來，我們所提出的系統，Flow 的數量是呈現直線(Flow 數量皆維持在固定的 20 條)，此代表我們的系統與每個使用者在專屬通道內的使用設備數量無關；而 OpenVirteX，其在交換器裡 Flow 的數量，會隨著使用者所連接的設備而有指數性地增加，這是因為 OpenVirteX 會針對每個設備進行位址的轉換所造成。

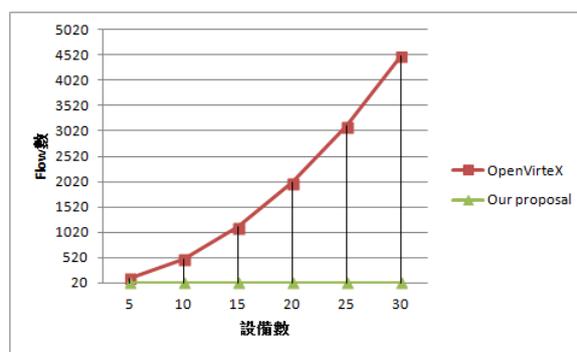


圖 5 兩系統在交換器 Flow 數的比較

第二個部份是模擬使用者端可能會有的情境，首先我們在實驗的環境中選擇兩台交換器(端點 A 與端點 E)，做為某個服務使用者分屬兩地的端點交換器，透過我們的系統於此兩台交換器間建立一個專屬通道，由於使用者端除了串接傳統的網路交換器外，也可能會佈署 OpenFlow 網路，在此實驗裡，我們在端點 A 與端點 E 的交換器上再各接上一台 OpenFlow 交換器，並建立一個 VM 做為此兩台 OpenFlow 交換器的控制器，以此情境模擬使用者端的 OpenFlow 環境，圖 6 顯示利用使用者端控制器所提供的 Web 介面呈現其所實際連線的拓樸，此結果也表示我們所提出的系統，可以讓使用者端的 OpenFlow 環境拓樸正確地顯示；接下來相同的實驗環境條件下，將我們的系統換回原本的 OpenVirteX，則原本圖 6 中兩台交換器間的連將會無法顯現出來，這是由於原本的 OpenVirteX 並未考慮上層有可能也是一個 OpenFlow 的網路環境所造成。

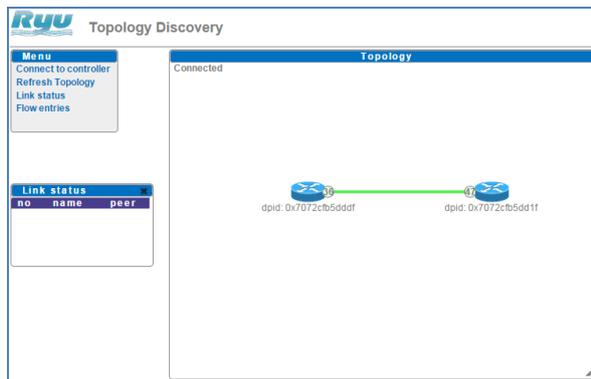


圖 6 使用者端於 GUI 顯示的連線拓樸

5. 結論

此篇論文，我們基於 OpenVirteX 實作出支援 layer2 的動態虛擬網路供裝系統，其可廣泛地相容於目前大多的 OpenFlow 交換器，此外，所提出的系統也能大量減少在 OpenFlow 交換器上 Flow 所需的數量，此對於系統實際佈署的延展性有相當大的助益；而在系統的上層，使用 OpenStack 來產生虛擬機器服務使用者，有效地運用 OpenStack 的管理介面來提供使用者所需的控制器。

在未來工作上，目前已規畫將在 TWAREN 的 4 個主節點與 12 個 GigaPOP 上各佈建支援 OpenFlow 的 SDN 交換器，做為網路虛擬化的節點，並透過我們所開發的系統與這些交換器介接，期能提供 TWAREN 在 SDN 上另一種的 VPLS 的服務。

參考文獻

- [1] M. Lasserre, V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling," IETF RCF 4762, 2007.
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, April 2008.
- [3] A. Al-Shabibi, M. Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, B. Snow, "OpenVirteX: make your virtual SDNs programmable," In Proceedings of the 3rd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), pp. 25-30, 2014.
- [4] R. Sherwood, G. Gibb, K-K, Yap, G. Appenzeller, M. Casado, N. McKeown, G. Parulkar, "FlowVisor: A Network Virtualization Layer," Technology Report 2009.
- [5] Open vSwitch. Available: <http://openvswitch.org/>
- [6] Lagopus switch: a high-performance software OpenFlow 1.3 switch. Available: <https://lagopus.github.io/>
- [7] OpenStack. Available: <https://www.openstack.org/>