整合 OPNFV 與 SDN 技術動態配置專屬 OpenStack 叢集

胡仁維　曾惠敏　劉德隆
財團法人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

隨著虛擬化技術發展的越趨成熟，在這幾年網路功能虛擬化(NFV)也受到相當程度的關注，其是運用了伺服器虛擬化技術，將原本在硬體設備上的功能抽離成網路元件，讓此功能在高容量的伺服器上執行，並隨時可以根據需求掛載或卸除。歐洲電信標準協會標準(ESTI)負責與多家網路服務和電信營運商制定NFV標準，而OPNFV是根據ESTI所制訂的標準所發展出一套開放原始碼的框架，滿足供應商、應用開發商和用戶端的需求，以快速地部署所需的NFV測試平台，此框架可以同時建構出多個OpenStack叢集，不過目前此框架在各叢集間，使用Flat的網路作為其連線之架構，但這會產生效能與安全上的問題，因此在此篇論文裡，我們將結合OPNFV與SDN技術分配不同的虛擬網路，讓各個叢集間的網路能區隔開來，彼此相互獨立運作。最後透過實驗情境來驗證此方法之可行性。

關鍵詞：NFV、OPNFV、OpenStack、OpenFlow、軟體定義網路。

1. 前言

為了提升使用效率、簡化管理和節省空間，網路功能虛擬化 (Network Function Virtualization, NFV) [1]，已成為全球網路服務與電信營運商朝向的主要發展趨勢。在有限的機房空間下，因應的業務創新加速，無法提供多餘的空間和電力供應這些設備，加上要在既有網路架構上整合這些龐大的硬體設備，常有不相容的情況下複雜度相對的提高，故利用虛擬化的技術將網路功能以軟體的方式實作，讓網路功能不再仰賴於專屬硬體架構，進而降低網路功能設備的硬體建置與維運成本，使資源運用更加靈活和彈性，借助虛擬化技術和各式軟體，即能以少量的設備來應付大量且動態的應用，例如，防火牆、負載平衡設備、網路快取、交換器等網路功能，從專屬的硬體設備搬移出來並且標準化，讓他們在軟體中執行以提供服務，並且可以支援在各種標準的伺服器硬體上，可按需求遷移或安裝在網路上的不同地方，完全不需要重新安裝新的硬體設備，以提升服務推陳出新的效率和便捷性。

雖然有不少的網路設備商已開始將防火牆、網路入侵偵測/防禦、網域名稱服務等網路功能，從專屬的硬體設備當中脫離，讓它們運行於虛擬機器之中，不過仍屬於較封閉式的開發環境，因此部署網路功能虛擬化解決方案的開放平台 Open Platform for NFV (OPNFV) [2]被概念應運而生，其是符合歐洲電信標準協會標準(ETSI) NFV ISG 的架構框架，它的主要目的是，讓供應商、應用開發商和用戶端的需求，根據自己的使用案例選擇所需要之套件、作業系統和 SDN 控制器，以快速的部署一個測試平台。

在今年年初，OPNFV 發行了第二個版本，不過目前 OPNFV 框架在整合作業系統與套件上已考慮相當完善，但在網路部分，則還是以 Flat 的架構為主，也就是讓所有伺服器都部署在同一個 Broadcast domain 之中，雖然 Flat 的架構簡單，但相對地也隨著設備的增加，會形成一個大的廣播區域，造成封包碰撞的機會也會相對提高，導致效能上的問題；另外，所有伺服器為於同一個 Flat 網路中，各叢集間是彼此通透的，因此也會造成安全上的問題，因此為了解決這些問題，本篇論文將運用我們先前所開發的 L2OVX [3]整合現有的 OPNFV 系統，來達到區隔各個不同 OpenStack [4]叢集，讓不同的叢集間彼此的虛擬網路獨立開來；此外，給每個 OpenStack 叢集也會指定一個獨立的 OpenFlow 控制器來管控，分擔目前 OPNFV 所提的由單一控制器可能會產生的效能瓶頸;最後 L2OVX 是以交換器埠來區分各個不同的叢集，可以讓虛擬網路的建置與應用更為方便與彈性。

在本篇論文的第 2 節中，我們將說明 NFV 的架構，第 3 節將介紹以 OPNFV 所提供的工具和各個版本的差異及配置的網路功能虛擬化環境，第 4 節將介紹整個系統的概括性架構,如何整合 L2OVX 與 OPNFV 系統，提供 OpenStack 叢集間各自獨立的網路虛擬功能，將詳述於第 5 節裡,在第 6 節中，針對我們的整合系統進行驗證，最後於第 7 節中作出本篇論文的結論。

2. NFV 的標準與開發平台

本章節將對 NFV 以及基於此標準所發展出可提供部署網路功能虛擬化的開放平台 OPNFV 進行介紹。

2.1 NFV 介紹

NFV 的架構，是 2012 年由歐洲電信標準協會標準(ESTI) 所提出，參與制定標準的有多家網路服務和電信營運商如：美國 AT&T、英國電信(BT)、義大利電信(Telecom Italia)、德國電信(Deutsche

Telekom)、西班牙電信(Telefonica)，日本電信(NTT)等指標性電信商，至目前為止 ETSI 的 NFV 產業規範小組 (Industry Specification Group, ISG)已超過 290 個組織成員加入推動。ETSI NFV 的架構，如圖 1 所示，包含以下五大功能：

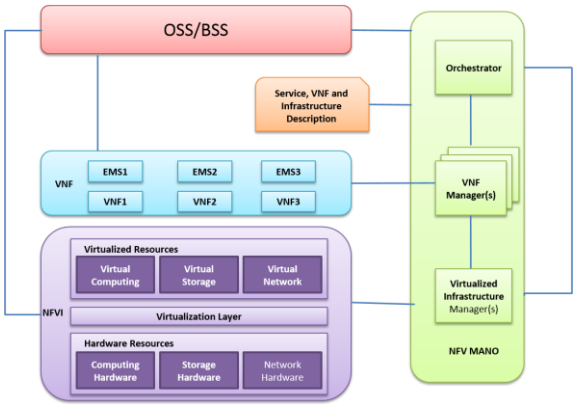

圖 1、ETSI NFV 架構圖

- **虛擬化網路功能 (Virtualized Network Function, VNF)**
  由各種不同的虛擬網路功能和網路元件管理系統 (Element Management System, EMS)所組成，可看作是網路設備功能，例如 Firewall，DHCP Servers 等，這些虛擬網路功能可被建置在一個或多個虛擬機器上，由 EMS 負責一個或是多個 VNF 的操作與管理。
- **網路功能虛擬化基礎建設 (NFV Infrastructure, NFVI)**
  NFV 架構中的基礎建設，包括虛擬資源(Virtualized Resources)、虛擬層(Virtualization Layer)、硬體資源(Hardware Resources)三個區塊，虛擬層介於虛擬資源和硬體資源之間，提供跨軟硬體間的資源存取轉換，包含計算，儲存和網路等用以提供 VFN 所需的虛擬化資源。
- **網路功能虛擬化管理與協調流程 (NFV Management and Orchestration, NFV MANO)**
  負責所有的 NFV 基礎設施和軟體的協調與管理，包含流程協調(Orchestrator)、VNF 管理(VNF Manager)與虛擬化基礎建設管理(Virtualized Infrastructure Manager)三個部分。當有軟硬體資源的需求時，NFV MANO 會協調、驗證、預留、授權相關的資源請求，同時負責管理 VNF 的生命週期(Lifecycle)，例如新增、更新、查詢、終止等，目前 OpenStack 最適合扮演這個重要的角色。
- **營運支援系統及企業支援系統 Operations Support Systems (OSS) and Business Support Systems (BSS)**
  代表營運商各自的營運支援系統和企業支援系統，提供給 NFV MANO 執行資源調配取用任務時參考用。
- **服務、虛擬化網路功能及基礎建設描述 (Service, VNF and Infrastructure Description)**
  因為 NFV 是跨廠商間的產品整合，當 NFV MANO 進行流程協調管理時，需辨別及參考相關資料，這些資訊就被定義在服務、虛擬化網路功能及基礎建設描述的資料檔中，包含 VNF 部署樣板、VNF 轉送圖、NFVI 資訊模型和各式服務相關的資訊。

2.2 OPNFV 平台

OPNFV 是一個部署網路功能虛擬化解決方案的開放平台，採歐洲電信標準協會標準(ETSI) NFV ISG 的架構框架，目前 OPNFV 著重在網路功能虛擬化基礎設施(NFVI)和虛擬化基礎架構管理(VIM)兩個部分的整體解決方案。OPNFV 包含了 OpenStack 和超過 30 個以上的其他套件，且增加了電信級的功能。

Brahmaputra 是 OPNFV 發布的第二個平台，2016 年 3 月 1 日發行，目前是 Brahmaputra 2.0 版，第一個版本是 ARNO，2015 年 6 月 4 日發行，目的是部署 NFV 及發展虛擬化網路功能(VNF)。新的一版 Brahmaputra 擴展初始發行版本的功能測試套件、增加系統測試、多性能測試架構方法及解決一些已知問題，具體來說，包括裸機部署的啟用，套件的升級 (例如：OpenDaylight 支援)和修正一些相關錯誤。如圖 2，展示了 OPNFV 計畫與他關聯的上游套件關係，圖的左邊說明 Brahmaputra 支援的相關上游套件，如：Ceph [5]、KVM [6]、OpenDaylight [7]、OpenStack 和 Open vSwitch [8]等，這些套件將與 API 和其他 NFV 元素一起組成虛擬化網路功能(VNF)以及網路虛擬化管理與協調流程(MANO)組件所需的基礎設施，另外，Pharos 測試實驗室，亦提供一個測試平台可以讓用戶測試不同環境和不同硬體下的 NFV 方案，使得 OPNFV 不依賴於某一種特定的硬體和是廠牌。圖的右邊說明整合、測試和新增功能。這些功能都是通過 OPNFV 計畫的測試核可後被加入到 OPNFV 中。

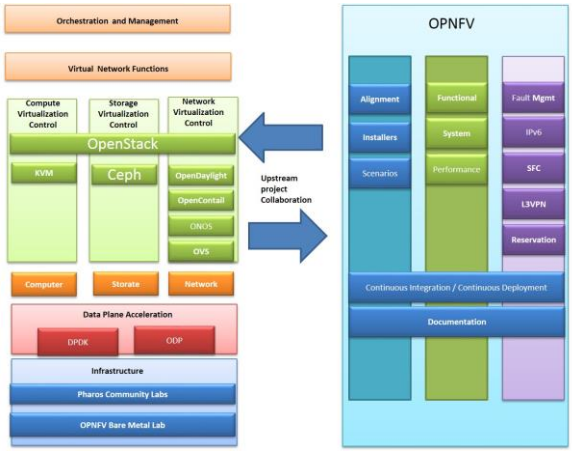

圖 2、Brahmaputra 與上游支援套件

Brahmaputra 支援四種不同 OpenStack 安裝版本，包括 Apex、Compass、Fuel、Joid 等四種，安裝環境可被部署在虛擬機或是裸機上。Apex 是使用 Triple O [9]結合 Puppet [10]的 RDO [11]來部署 OpenStack，RDO 是由 Red Hat 開發的一種部署 OpenStack 工具。Compass 利用 Ansible [12]部署 OpenStack，Ansible 是一個基於 Python 開發的自動化管理開源工具，適用於批次系統配置、程序部署、執行命令等。Fuel 是一個 WEB 化的部署工具，整合 OPNFV 功能後用來部署和管理 OpenStack。Joid 使用 Canonical 推出的部署工具 Maas [13]和 Juju [14]，Maas 用於安裝 Ubuntu，Juju 用於部署應用。表 1 說明這四種版本目前支援的 OpenStack 套件內容，可使用相關的附加套件作為部署的工具和測試框架，而 OpenStack 在 OPNFV 中扮演的角色用於控制計算、儲存和網路資源等，這裡是使用 OpenStack Liberty 版本，這四種 OpenStack 安裝版本的執行機器都是使用 Linux 作為作業系統，如：Fuel、Compass 和 Joid 支援 Ubuntu 14 版本，而 Apex、Compass 則支援 Centos 7 版本。同時也支援不同的 SDN 控制器，如果不使用 SDN 控制器，也可使用 OpenStack 本身提供的 Neutron 的網路功能，這裡支援的 SDN 控制器包含目前發布的 OpenDaylight(Beryllium)、ONOS 1.4(EMU) 版本等，如表 1 所示。

表 1、四種軟體支援之 OpenStack 套件

Services	type	Apex	Compass	Fuel	Joid
aodh	alarming	Available	-	-	-
ceilometer	metering	Available	Available	Available	Available
cinder	volume	Available	Available	Available	Available
cloud	coludformation	-	Available	Available	Available
glance	image	Available	Available	Available	Available
heat	orchestration	Available	Available	Available	Available
keystone	identity	Available	Available	Available	Available
neutron	network	Available	Available	Available	Available
nova	compute	Available	Available	Available	Available
swift	objectstore	Available	-	Available	Available

3. 系統架構

在這個章節中，將描述運用 OPNFV 與 SDN 技術所提出的整合系統之架構，我們參考[15]把系統架構分成三層，如圖 3 所示，由下至上分別為伺服器層、網路層與控制部署層。伺服器層是由多個伺服器群所組成，每台伺服器上都會安裝虛擬網路交換器，目前 OPNFV 的軟體部署工具如 Compass，預設是安裝 Open vSwitch 作為其虛擬交換器，此虛擬交換器的用途，對外是與上層的實體交換器串接；而對內則是做為運行於伺服器上虛擬機器所連接之用。

第二層為網路層，此層的實體交換器必須能夠提供軟體介面讓外部的控制器方便管理，以符合所謂的軟體定義與可程式化兩大需求。

此架構的最上層為控制與部署層，主要是利用這些伺服器搭配著控制管理的軟體工具，來達到自動部署與供裝虛擬網路。我們分成三種不同的功能的伺服器，第一個是負責伺服器的部署，包括作業系統與套件軟體的安裝、網路的設定等等，如前一章的介紹，目前在 OPNFV 中，有不同的整合工具可以使用，如 Compass、Fuel 等；第二個是負責虛擬網路的建立，透過 OPNFV 工具所產生的虛擬機器，雖然有虛擬交換器連接，但跨機器間的連接，還是必須藉由網路層的實體交換器，所以如何能將這些建立的虛擬機器連接，就是透過此功能的伺服器來協助，目前我們所使用的，就是我們所開發的 L2OVX 來負責建立此虛擬網路，其是利用 SDN 技術實作出支援 layer2 的動態虛擬網路供裝系統，並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；最後一個則是 OpenFlow 的控制器群，由於前面有提及，網路層的硬體交換器須支援 OpenFlow 這個控制協定，所以上層必須有一個 OpenFlow 的控制器來連接，每個產生的虛擬網路，皆有一個對應的 OpenFlow 控制器來負責做封包的接收與傳送。

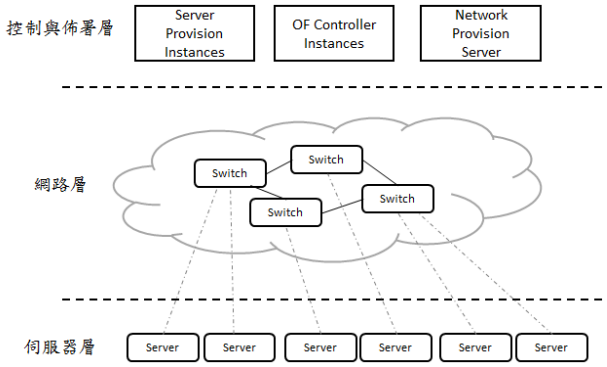

圖 3、系統架構

4. OPNFV 與 L2OVX 整合機制

在前一章節，我們說明了此整合系統的概念性架構後，接下來在此章節中，我們將 OPNFV 與 L2OVX 兩系統的整合機制，進行詳細的介紹。目前 OPNFV 有四種不同的工具可以用來部署、安裝與設定虛擬機器，在本篇論文中，我們將使用其中的 Compass 此工具，做為與 L2OVX 整合之用。

圖 4 為 Compass 與 L2OVX 兩系統的整合核心元件，我們將在下列分別做介紹。首先是 Hardware Discover Engine，這個元件是要搜尋能夠被使用的硬體資源，目前 Compass 是用 SNMP 協定來偵測所管控的交換器，有哪幾台伺服器與其連接，因此伺服器的 MAC 位址與連接的交換器埠等資訊，可以透過此元件取得，不過根據其官方文件所示，目前 Compass 支援的交換器，只有 Pica8、HP、Dell、Arista 與 Huawei 等這幾個廠牌。OS Provisioning Engine 與 Package Deployment Engine 此兩元件，分別是整合 Cobbler [16]與 Chef [17]工具，前者是用來安裝指定的作業系統至伺服器；而後者則是負責

工具程式或系統軟體的安裝與相關設定之用，使用這些工具，目的皆為讓部署機器更自動化並且減少人為上的操作疏失。最後是 Restful API Engine 與 Persistence Engine 是 Compass 用來處理 API 的運作，並將使用者操作與設定的內容，透過 Persistence Engine 儲存於系統之中。

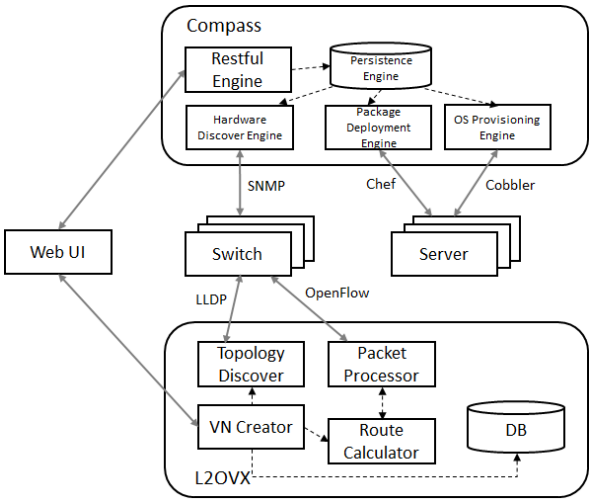

圖 4、系統的整合核心元件

而 L2OVX 系統中的核心元件，首先是 Topology Discover，其是用來建構出目前網路層中實體交換器的連接拓樸，由於 SDN 交換器會定期發送 LLDP 封包，因此此元件會根據所接收的 LLDP 封包進行解析並取得交換器的連線關係與狀態；目前網路層交換器接連接到 L2OVX，因此所有流經交換器的封包，都會經由 Packet Processor 來做分配，並根據封包的檔頭轉送至各個 OpenFlow 控制器，而若此元件收到的是尚未由 L2OVX 所配發的交換器埠進來的封包，會自動寫入 Drop Flow entry 至交換器中，讓其能直接在交換器上將封包丟棄，增加系統之安全與提高 L2OVX 的處理效能。而虛擬網路的建立，是透過 VN Creator 來處理，此元件會指定 OpenFlow 控制器給虛擬網路，並將所有管理的交換器組成一個大交換器，這樣的優勢是可以透過 L2OVX 的另一個元件 Route Calculator 來算出備援的路徑，提供給虛擬網路使用。

系統整合的流程圖，如圖 5 所示，首先透過 WEB 介面建立 OpenStack 叢集，在流程的第二個步驟，藉由發送 SNMP 取得網路層中的 OpenFlow 交換器有哪些連接的伺服器可以提供給 OpenStack 叢集來使用，伺服器與交換器的連接資訊會記錄於系統之中，在之後建立虛擬網路時會被使用，接下來是指定所選擇的伺服器於 OpenStack 叢集所要安裝的角色，目前有 Controller、Compute 與 Storage 等三種可以選擇，當設定完成後，Compass 就會透過其核心元件開始安裝與設定伺服器，此時，我們再藉由所得知的伺服器所連接的交換器埠資訊，將其送至 L2OVX 中的 Route Calculator，透過此元件算出的路徑，並把路徑轉成對應的 Flow 資訊寫入交換器之中，完成虛擬網路的建置，當 Compass 所安裝的伺服器設定完成後，OpenStack 叢集內的伺服器即可正常溝通與運作。

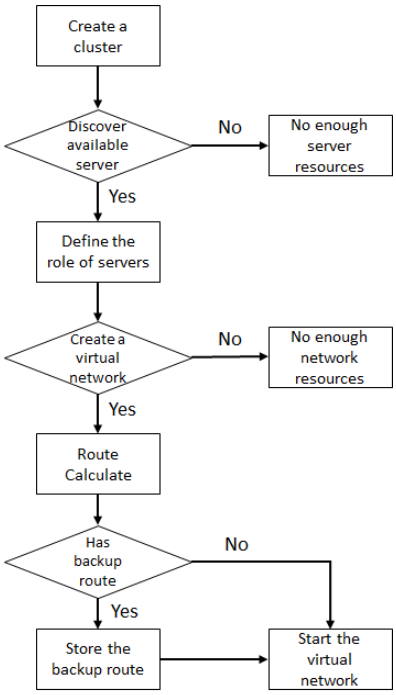

圖 5、整合機制流程圖

5. 實驗環境與情境

為了驗證整合系統的可行性，我們在實驗室裡建立了一個實驗環境，如圖 6 所示，由 6 台伺服器與 4 台交換器所組成，我們選擇其中一台伺服器，作業系統是 Ubuntu 14.04，作為安裝 Compass (OPNFV 第二版 Brahmaputra)與 L2OVX 整合系統的主機；每個虛擬網路，L2OVX 都會指定一個獨立的 OpenFlow Controller，用來控制虛擬網路，因此我們準備了一台伺服器作為此 Controller Instances 的 pool 之用；剩下的 4 台伺服器，則用來讓 Compass 建置 OpenStack 叢集之用，此些伺服器皆有網路管理埠並啟動 IPMI 相關設定，會將此管理埠接上一台 L2 交換器與 Compass 能夠相連接；另外，每台伺服器還有兩個 Ethernet 網路介面，其中一個接到 OpenFlow 交換器；另一個網路介面則是接到 OpenStack 的網路，當 OpenStack 叢集起動時，會建置彼此間的溝通環境。接下來是交換器的配置，除了剛剛作為連接伺服器 IPMI 介面的 L2 交換器外，另外有 3 台支援 OpenFlow 的交換器，用 Full mesh 的方式對接，並設定 Controller 指向我們所開發的 L2OVX 系統，各設備的功能與安裝軟體如表 2 所示。

表 2、實驗環境設備

設備類別	設備對應名稱	安裝套件
伺服器	L2OVX & Compass	1. L2OVX 2. Compass
	OpenFlow Controller	Floodlight

	Host1 – Host4	作為 OpenStack 叢集之用，根據不同角色安裝不同的工具
交換器	L2SW	傳統 L2 交換器
	OF1 – OF3	OpenFlow 交換器

透過 Compass 與 L2OVX 的系統 WEB 介面建置兩個 OpenStack 叢集，第一個叢集(圖 6 中的灰色區塊)是由 Host1 作為叢集的 Controller Node ，Host3 是 Compute Node，分別接上 OF1 與 OF2 的 OpenFlow 交換器，L2OVX 也配置 C1 作為此虛擬網路的控制器；經由類似的步驟，再建立另一個 OpenStack 叢集，是由 Host2 與 Host4 所組成，分別作為第二個叢集中的 Controller 與 Compute 之用。

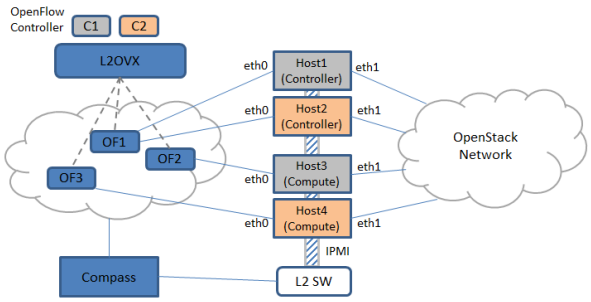

圖 6、實驗環境

由於這兩個 OpenStack 叢集的虛擬網路，是透過整合系統中的 L2OVX 所完成，可以透過 L2OVX 的管理介面，列出目前系統上的虛擬連線相關資訊，如圖 7 所示，L2OVX 會將此兩個叢集視為不同的獨立虛擬網路，所以他們即便使用相同的私有網段 (如：192.168.0.0/24)，在同一個叢集內的伺服器是可以相互連結上 (如：Host1 與 Host3)，但在屬於不同的叢集的伺服器間 (如：Host1 與 Host2)，是無法建立連結的，藉由 L2OVX 所建立的虛擬網路將其分開達到網路間的獨立，除此之外，也提高整體系統網路運作的安全性。

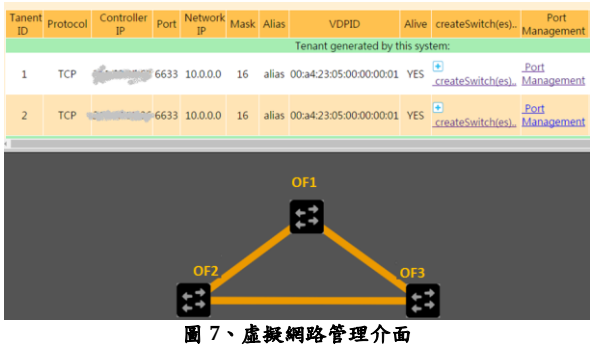

圖 7、虛擬網路管理介面

6. 結論

在本篇論文裡，我們基於 OPNFV 實作 Compass 的 OpenStack 環境整合現有 L2OVX 系統來達到區隔各個不同 OpenStack 叢集的目的，讓不同的叢集間彼此的虛擬網路獨立開來，各個叢集間流量互不干擾，透過 L2OVX 集中管理所有的 Switch 簡化很多在管理上程序，同時也大大的提升網路的安全。未來，我們將藉由開發 L2OVX 的 Neutron plugin，整合至 OpenStack 的網路之中，除了管理叢集間的網路之外，也可以進一步的管理 OpenStack 下虛擬機器間的網路，以達到網路統一集中管理的目的。

參考文獻

[1] NFV, “Network Functions Virtualisation”. Available: http://www.etsi.org/technologies-clusters/technologies/nfv

[2] OPNFV, “Open Platform for NFV”. Available: https://www.opnfv.org/

[3] 胡仁維、曾惠敏，劉德隆「以 OpenFlow 於骨幹上實作動態虛擬網路之供裝」，TANet2015 論文集，南投，2015 年 10 月。

[4] OpenStack, “Open Source Cloud Computing Software”. Available: http://www.openstack.org/

[5] S.A.Weil, S. A. Brandt, E. L. Miller, D. E. Long, and C. Maltzahn, “Ceph: A Scalable, High-Performance Distributed File System,” in OSDI, 2006, pp. 307-320.

[6] KVM, “Kernel-based Virtual Machine”. Available: http://www.linux-kvm.org/page/Main_Page

[7] OpenDaylight, “Open Source SDN Platform”. Available: https://www.opendaylight.org/

[8] Open vSwitch, “Open Virtual Switch”. Available: http://openvswitch.org/

[9] Triple O, “OpenStack on OpenStack”. Available: https://wiki.openstack.org/wiki/TripleO

[10] Puppet, “Open Source Puppet”. Available: https://docs.puppet.com/

[11] RDO. Available: https://www.rdoproject.org/

[12] Ansible. Available: https://www.ansible.com/

[13] Maas, “Metal as a Service”. Available: http://maas.io/

[14] Juju, “open source service orchestration management tool”. Available: https://jujucharms.com/

[15] S. Yang, W. Shao, H. Song, and W. Xu, “Compass: A Data Center Bootloader for Software Defined Infrastructure, ” in ICNS, 2015, pp. 98-102.

[16] Cobbler, “Linux install and update server”. Available: http://cobbler.github.io/

[17] Chef, “Automate Your Infrastructure”. Available: https://www.chef.io/chef/