

軟體定義網路技術應用於OPNFV

SDN Technology Applied in OPNFV

許名宏 顏至寬
Ming-Hung Hsu, Simon Yen

中文摘要

SDN 與 NFV 兩種技術都是近幾年來網路通訊產業的熱門議題，且都引發了重要的開源計畫，對於實現開放式創新服務有顯著的影響。因此在本文中，我們透過整理、回顧 NFV 技術構想的背景、使用案例，以及 ETSI NFV 標準參考架構與其中核心的模組功能，來闡述開放原始碼計畫對於發展 NFV 系統的重要性。接著我們會簡介目前最重要的 NFV 開源計畫 OPNFV 的發展背景、目標與現況，並呈現已實作在 OPNFV 計畫中的 SDN 服務功能鏈結技術。期望透過本文的整理介紹，可吸引更多人員投入相關產業技術研發。

Abstract

In recent years, SDN and NFV are both hot topics in computer networking industry. The two technologies both trigger important open-source projects, significantly influencing the realization of open innovation of network services. Therefore in this study, by reviewing the background context, the use cases, and the reference architectural framework of ETSI NFV, we clarify the importance of open-source projects on developing NFV technology. Then we briefly describe the status of the most important NFV open-source project – OPNFV, and present an SDN-based service function chaining technique implemented in OPNFV. We wish that this survey about SDN/NFV could attract more engineers to join in SDN/NFV technology development.

關鍵詞(Key Words)

軟體定義網路 (Software Defined Networking ; SDN)
網路功能虛擬化 (Network Functions Virtualization ; NFV)
歐洲電信標準化協會 (European Telecommunications Standards Institute ; ETSI)
虛擬網路功能 (Virtual Network Function ; VNF)
服務功能鏈結 (Service Function Chaining ; SFC)
網路功能虛擬化開放平台計畫 (Open Platform for NFV ; OPNFV)

1 · 前言

自從2012年四月，Google 在 ONS (Open Networking Summit) 大會上，發表他們將基於

OpenFlow v1.0 的軟體定義網路 (Software-Defined Networking ; SDN) 技術[1]，實際運用在連接資料中心之間的廣域網路，獲得強大的效能改進及商業效益之後[2][3]，“SDN”

與“OpenFlow”頓時成為網路通訊產業的熱門議題，帶動起一波方興未艾的產業革新，SDN 也被視為動態配置、管控資料中心網路的最佳方案。另一方面，網路功能虛擬化(Network Functions Virtualization ; NFV)技術的發想[4]，與SDN的興起幾乎是在同一時期，其發展的誘因主要來自於傳統電信網路設備昂貴的硬體成本，加上難以動態配置以滿足不斷新出現的服務需求，使得電信服務營運商越來越難以經由傳統營運方式獲利。而雲端運算概念及技術逐漸成熟，使得計算能力、儲存及網路等硬體資源的虛擬化成為可行且有效的。種種因素的堆疊啟發了NFV的技術構想：將傳統網路功能軟體化及虛擬化，執行在高效能的通用伺服器上。相較於傳統分散式控制的網路只能倚賴國際設備大廠(例如Cisco、Juniper)各自的緩步創新模式，NFV 與SDN也成為網通產業實現開放式創新(open innovation)的重點方向，兩者的結合亦被認為將是未來5G通訊技術的重要組成[5]。

歐洲電信標準化協會(European

Telecommunications Standards Institute ;ETSI)的 NFV ISG (Industry Specification Group)小組是推動 NFV 標準技術發展的重要推手，在2012至2014之間的每一年都發表了一篇關於 NFV 技術發展的白皮書[4][6][7]，不僅制訂了 NFV 的標準參考架構，也清楚地說明了NFV 的發展緣由及應用案例。圖1完整地揭示了，運用 NFV技術在不同場域可被虛擬化的網路功能(Virtual Network Function ; VNF)，例如可將電信核心網路(core network)的 MME 功能虛擬化，或是可將位於家用網路與核網之間的RGW (Residential Gateway)功能虛擬化，以達到動態擴增(scale up)或是即時備援等目的。圖1也展示了一個當使用者終端設備經由LTE 無線通訊介面與核網的虛擬應用伺服器(App Server) 連接時，中間可能依序經過多個 VNF 的範例，即vBS→MME→SGW→PGW→App Server；此類將多個實體或虛擬網路功能依特定順序串連

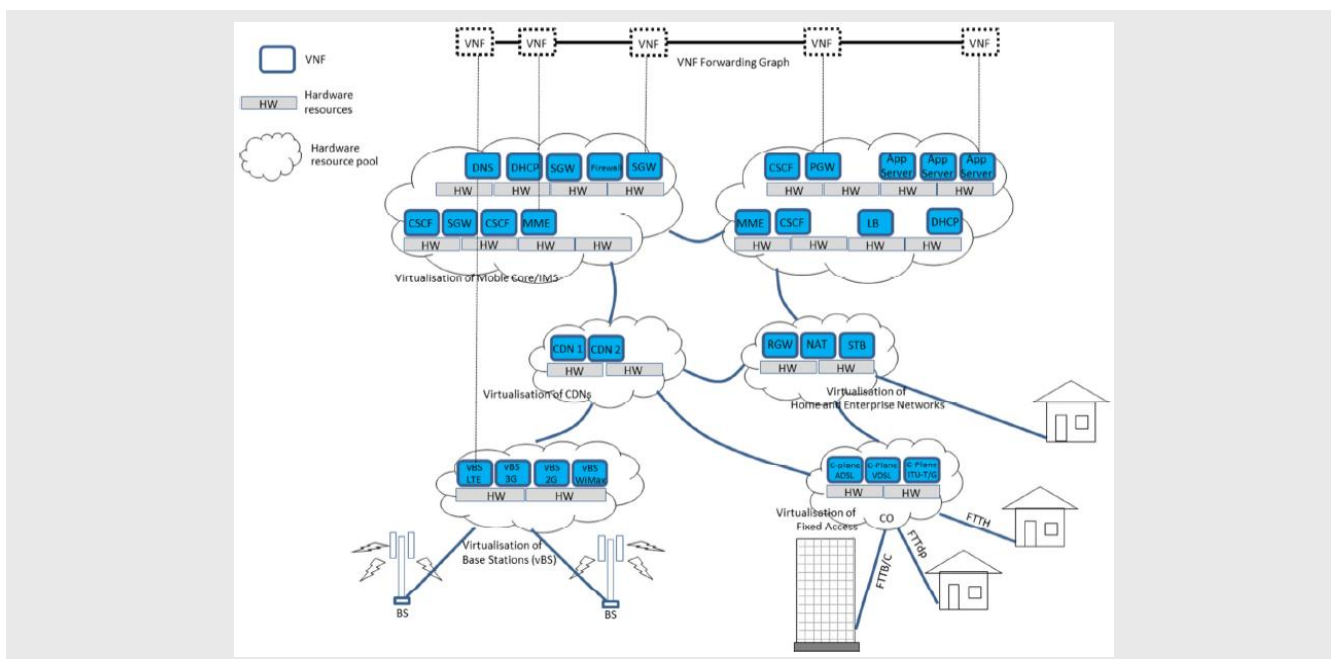


圖 1 NFV 技術在不同通訊場域的應用案例總覽[]

起來的技術特徵，一般稱為服務功能鏈結 (Service Function Chaining ; SFC)。SFC的架構組成及相關定義在近期也被正式定義於 IETF RFC 7665 [8]。

有鑒於 SDN 與 NFV 同樣具備開放創新特性、引發了許多深具產業影響力的開放原始碼(open source, 簡稱開源)計畫，例如 OpenDaylight [9]、ONOS [10]、OPNFV [11] 等等，同時此兩技術彼此密切相關又具有互補效益，因此，本文在第二章會介紹 NFV 系統的標準架構，以及結合 SDN 與 NFV 的效益所在。隨後在第三章會介紹目前最重要的 NFV 開源計畫 OPNFV (Open Platform for NFV)的發展背景、目標與現況，以及在第四章描述 SDN 技術在 OPNFV 計畫中如何被運用於實現服務功能鏈結。最後第五章是本文的結論。

2 · 網路功能虛擬化技術介紹

相對於近十年來雲端運算的技術發展已趨於成熟且廣為人知，NFV 仍是較新穎的概念。本章節簡單地介紹 ETSI 所制訂的 NFV 標準架構，以及 SDN 技術在 NFV 系統中所扮演的角色。

2.1 ETSI NFV 標準架構

下圖2是歐洲電信標準化協會(ETSI)在2013年所發表的 NFV 系統參考架構，以下介紹其中的主要模組及其功能：

- NFV Infrastructure (NFVI)：實現 NFV 技術的基礎設施，可提供虛擬化的計算 (通常為 virtual machine ; VM)、儲存、網路等資源。
- VNF：在 NFVI 上運行的虛擬網路功能，例如 Firewall、NAT、CDN 等等。
- Virtual Infrastructure Manager (VIM)：負責管理 NFVI 及分配虛擬資源的模組。在已知的 NFV 概念驗證案例中，絕大部份是直接擴充原本雲端運算管

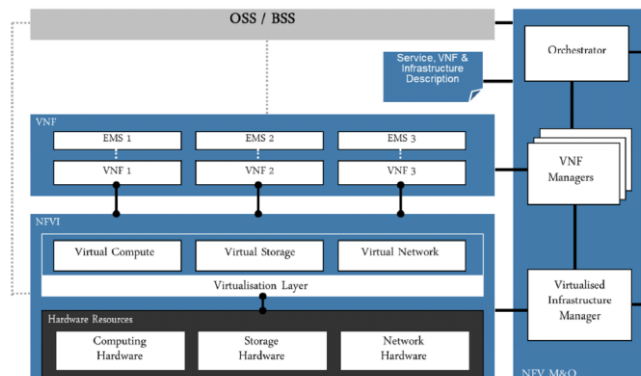


圖 2 由 ETSI 所制訂的 NFV 標準參考架構[]

理平台 (Cloud Management System ; CMS)而成為此模組。目前最被廣泛採用的 VIM 解決方案是 OpenStack [12]。

- VNF Manager (VNFM): 負責管理 VNF 的模組，其任務包括 VNF 的啟動與中止、狀態監控以及組態設定等等。在一個 NFV 系統中可由不同的 VNF 供應商提供各自的 VNF Manager 及 VNF，使得系統可支援更多樣的網路功能。
- OSS/BSS：負責一般電信業務所需的服務開通(service provisioning)、會計帳務、使用者資料維護等功能。
- Orchestrator：負責依據OSS/BSS提出的需求，例如為某個終端設備的流量提供防火牆及入侵偵測等功能，而協調VNF Manager 及 VIM 各自負責的任務，達到編排(orchestrate)一個特定服務，使其具有多個網路功能的目標。

圖2右側的 Orchestrator、VNF Manager 及 VIM 共同形成一個 NFV系統的管理平台，一般稱為 NFV MANO (Management and Orchestration)。

2.2 結合SDN與NFV—服務功能鏈結

在目前已知的 SDN 或 NFV 使用案例當中，動態或是即時的服務功能鏈結，應該

是最能表現 SDN 與 NFV 結合的強大效益所在。換句話說，SDN 使虛擬網路的佈署變成可動態、彈性的實現，而這樣的特點正是 NFV 系統，或說電信營運商，所極度需要的。當各種網路功能可被虛擬化、動態配置及即時監控的時候，如何動態鏈結多個網路功能，以實現特定的服務鏈(Service Chain)來對使用者流量做合適的處置，便是 SDN 技術在 NFV 系統中要發揮的作用。圖3表示了一個邏輯上的 End-to-End 網路服務是由多個 VNF 組成，在 NFV 系統的管理平台中會以 VNF Forwarding Graph 的表示法，來描述一個服務鏈中的 VNF 之間的相對順序。在本文第四章也會介紹如何以 SDN 技術來實作服務功能鏈結。

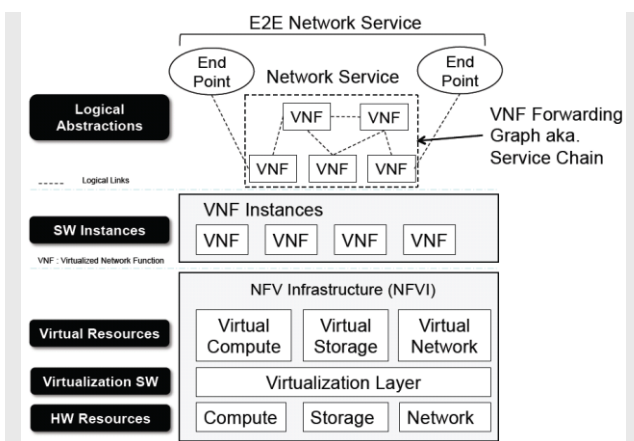


圖 3 NFV 結合 SDN 服務功能鏈結技術

3 · NFV開放原始碼計畫—OPNFV

3.1 OPNFV 的計畫背景、目標與定位

如同本文前面曾經描述的，NFV 技術構想是因應電信服務營運商的需求而誕生。然而，一個不難想像的情況是，對於習慣於購買、佈署、管理傳統網路設備的電信營運商而言，龐大的 NFV 系統軟體是複雜而難以掌握的，更別提要如何持續測試、整合整個系統或是優化部份元件以達到電信等級(carrier-grade)的穩定性及效能。在這樣的背

景下，NFV 開源計畫—OPNFV，被期待成為 NFV 產業界重要的發展基礎。OPNFV 是由 Linux Foundation 在 2014年9月所發起，其主要目標有以下五項：

- (1) 發展一個經過完整整合測試、可運用於建置 NFV 系統功能的開源平台。
- (2) 積極邀請領先的終端使用者(end users)，即一流的電信服務營運商，加入參與計畫，以驗證 OPNFV 符合使用者社群的需求。
- (3) 參與那些會被運用於 OPNFV 平台的上游(upstream)開源計畫，並適時提出貢獻；確保這些開源元件之間的一致性(consistency)、效能(performance)以及互通性(interoperability)。
- (4) 基於開放標準及開源軟體，建立NFV 解決方案的生態系統(ecosystem)，以符合終端用戶(end users)的需求。
- (5) 促使 OPNFV 成為 NFV 開源軟體社群的發展重心。

由這些目標可看出，OPNFV 不僅是個實現 NFV 技術的平台，更是個蒐集、瞭解使用者需求，並將必要資訊回饋給上游開源計畫的溝通平台。

下頁的圖4是由 Heavy Reading 所做的 OPNFV問卷調查¹，對象包括211家各類網路營運商或軟、硬供應商，結果顯示有超過90%的回覆是正面肯定 OPNFV 對於 NFV 產業發展的重要性。

3.2 OPNFV發展現況

至今年3月為止，OPNFV 在成立僅18個月後已成長到相當可觀的規模：已有58個會

¹http://events.linuxfoundation.org/sites/events/files/slides/OPNFV%20survey%20results%20presentation-%20R%20Roseboro_0.pdf

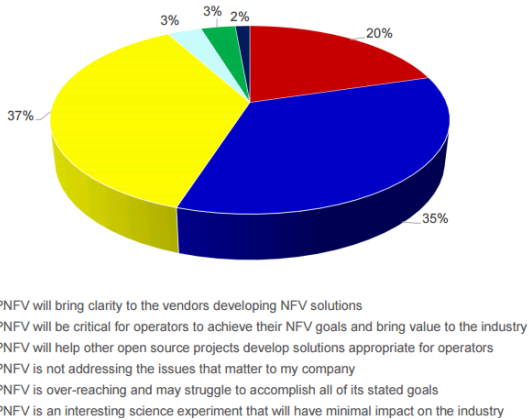


圖 4 Heavy Reading 的 OPNFV 意見調查結果

員廠商加入，其中有41個出資贊助，會員並包括12個國際知名的電信服務營運商（例如 AT&T、China Mobile 及 NTT docomo）；底下已有47個通過審核的子計畫（approved projects），數量還在持續增加中；參與的程式碼開發者已達到168人，也在增加中；更已有11個研究單位的實驗室，分享各自不同的環境與設備，做為 OPNFV 底下各個子計畫的多樣化測試情境；另外 OPNFV 也已在今年2月下旬發布了第二版的軟體釋出，名為 Brahmaputra。附帶一提，OPNFV 的版本皆以著名的河流名稱來命名。

圖5展現了 OPNFV Brahmaputra release 的重點總覽。圖5的左半部中的灰色方塊表示出，在使用 Brahmaputra 佈署 NFV

Infrastructure 時，可以採用的上游開源計畫，其中與網路相關的計畫有：加快作業系統處理封包速度的 DPDK [13] 及 OpenDataPlane (ODP) [14]、被廣泛使用的軟體交換器 Open vSwitch (OVS)[15] 以及三個開源的 SDN 控制器 (controller)—OpenDaylight[9]、OpenContrail [16]及 ONOS[10]。而關於將硬體資源虛擬化的管理配置(意即 VIM 的角色)，則一致透過 OpenStack 這個已被廣泛使用於雲端系統管理的開源平台來達成[12]。雖然在 OpenStack 底下有多種技術可實現計算或儲存資源虛擬化，目前 OPNFV Brahmaputra 只有採用 KVM[17]及 Ceph[18][19]來進行關於佈署工具的整合或是效能強化的子計畫。

由圖5中較大的色塊還可以看出，OPNFV Brahmaputra 底下的子計畫可以分成三大類：Integration (and Deployment)、Testing 及 New Features。關於Integration 類別的子計畫，其目的是讓使用者可實現自動化安裝及佈署，因此有許多整合操作介面的工作。而 Testing 類別的子計畫是要驗證所開發的 NFV 平台，在功能及效能面皆可確實符合營運商的需求。另外，OPNFV Brahmaputra 這版的新 Features 包含了第四章會介紹的服務功能鏈結(SFC)。儘管New features 類別的子計

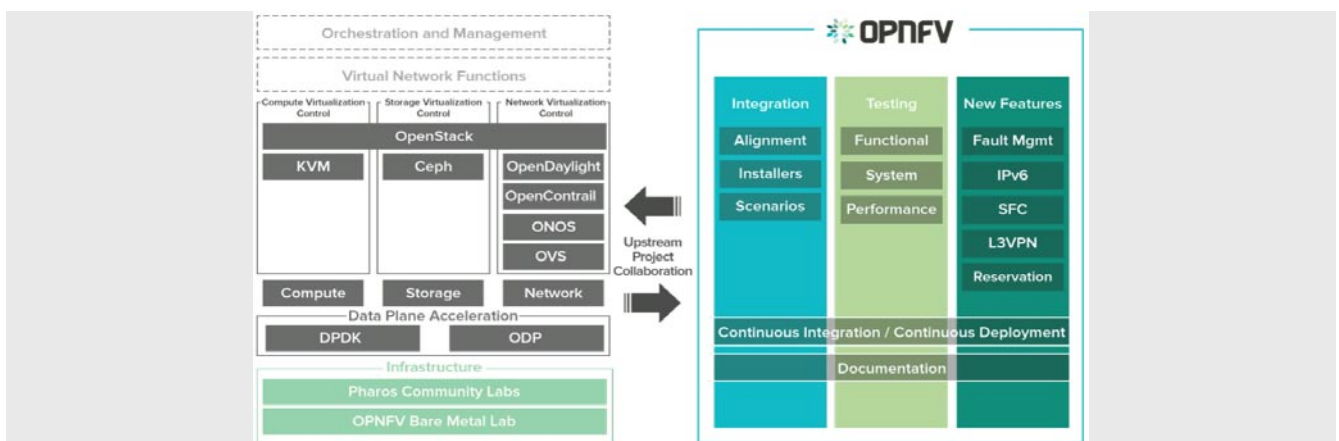


圖 5 OPNFV 的第二版釋出—Brahmaputra 總覽

畫不少，但其實對應的程式開發工作絕大部份都會回饋到上游計畫去執行，這也是 OPNFV 執行委員會所謂的「Upstream First」政策。

4 · SDN服務功能鏈結技術

在 OPNFV 的上游計畫中，OpenDaylight與 OpenStack 都有支援 SFC 的特徵功能。目前 OPNFV 在 Brahma Putra 這版是採取整合 OpenDaylight的方式，但未來仍有選擇整合 OpenStack 的可能性，因此本章節會介紹兩上游計畫各自實作服務功能鏈結的方法。

4.1 OPNFV Brahma Putra實現 SFC 的方法

目前 OPNFV Brahma Putra 版支援服務功能鏈結的方式，是採用 IETF 草擬中的 NSH (Network Service Header) 資料格式 [20]，將封包該走哪條服務鏈、該服務鏈有多少服務功能中繼站(hop)等資訊，分別記錄於 NSH 標頭的 NSP (NSH Path，長度為24 bits) 及 NSI (NSH Index，長度為8 bits) 兩欄位當中，為服務鏈的封包加上 NSH 標頭後送出，再由路徑上受 OpenDaylight 控制的服務功能轉送站 (Service Function Forwarder ;SFF) 決定該封包該如何轉送。

以下是 OPNFV 實現 Service Function Chain 的重要核心元件介紹：

- 流量分類器(Classifier)：根據設定的政策 (policy)，將匹配符合(matched)的封包加上 NSH 標頭，在對應欄位填入服務鏈識別碼(identifier)後，將封包送到服務鏈的傳送路徑上。
- Service Function (SF)：具有特定作用的、實體或是虛擬的網路功能，負責將收到的封包進行處理。例如 Firewall、NAT、DNS、DPI甚至是IDS/IPS等等，皆為特定的服務功能。
- Service Function Forwarder (SFF)：目

前大多是以 OVS 實作，負責轉送服務鏈上的封包至對應的目的地 SF。

- 服務鏈：由多個 SF 串連建構出一條服務鏈。封包通過流量分類器後會依序經過服務鏈中的每個 SF，例如使用者可設定三個功能 Firewall、IPS、DPI 依序為一條服務鏈。
- Rendered Service Path (RSP)：進入服務功能鏈的封包所經過的實際路徑。

下圖 6 為 OPNFV 透過 OpenDaylight 控制器實現服務功能鏈結的範例示意圖，以下的流程描述對應於圖6的藍色箭頭。假設只有 Host1 傳給 Host2 的 HTTP 封包須進入該服務功能鏈，當Host1傳送給 Host2 時，封包會先經過classifier，只有 Host1 傳送給 Host2 的TCP 埠號 80 的封包會被附加上 NSH 標頭，並將對應的服務鏈識別資訊填入 NSP 及 NSI 等欄位，並轉送到該服務功能鏈中，而其餘的封包按照原本預設的方式轉送到目的地。進入服務功能鏈的封包進入 SFF1，SFF1會檢視該封包的 NSH 標頭內容，確認該封包目前尚未處理，因此轉送至 SF1(例如 Firewall) 處理，當封包進入SF1 處理過後更新 NSH 標頭 (即 NSP 不變、NSI 的值減1)，再送到 SFF1。SFF1 接收從 SF1 送來的封包會再檢視該封包的 NSH 標頭，發現該封包已通過第一站，因此必須將封包轉到下一個 SFF。依據上述步驟，SFF2 收到封包會轉送給SF2處理，再送回SFF2轉發給

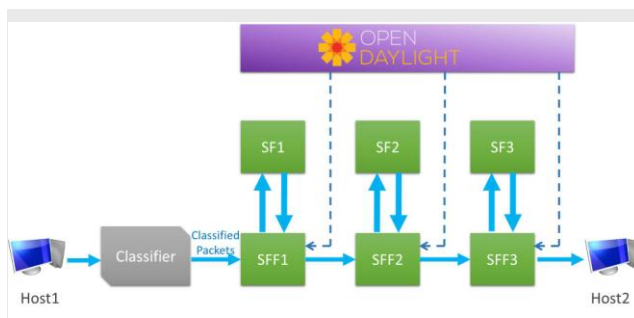


圖 6 以 NSH 標頭實作服務功能鏈結範例

SFF3，SFF3收到後轉送給SF3處理，再送回至SFF3。最後 SFF3 檢視 NSH 標頭發現該封包已經過該服務鏈的最後一站處理，因此會將 NSH 標頭拿掉並轉發給目的地 Host2。由於服務鏈有方向性，如果在圖6的例子中只設定 Host1 至 Host2 單一方向的服务功能鏈，Host2 回給 Host1 的回覆 (reply) 封包將不會進入該服務功能鏈中，而是照預設轉送的方式回到 Host1。

4.2 實作於 OpenStack 的 SFC 功能

與 OPNFV目前的實作方式不同，在 OpenStack 底下與網路相關的 SFC 專案(名為networking-SFC)不使用 NSH 所定義的欄位，而是透過 OVS 修改目的端 MAC 位址的方式來實作 SFC 的功能。假設將環境簡化為由一個 OpenStack 實例(instance)所管理的資料中心，圖7可作為 OpenStack 實作 SFC 的流程說明範例：實體伺服器 Node A 上運行流量來源虛擬機 Source VM (VM1)；而虛擬網路服務功能 Service Function (VM2) 執行在實體伺服器 Node B 上；目的端 VM (VM3)則運行於實體伺服器 Node C 上；實體伺服器之間用 Tunnel 連接，每台伺服器上皆有 OVS 已進行封包目的端 MAC 轉換。SDN 控制器會預先將 SFC 的轉送規則(forwarding rule)設定在每台伺服器的 OVS 上，當封包經過 OVS 時便會匹配對應的規則以執行 SFC 的路徑轉送及必要的封包欄位修改。

以圖7為例，首先 VM1 要傳送一個封包給 VM3，來源端 IP 位址為 IP1，來源端 MAC 設定為 M1，目的端 IP 位址為 IP3，目的端 MAC 位址為 M3，且該封包必須經過服務鏈。當封包進入OVS1時，會將目的端 MAC 位址改為下一站 VM2 的 MAC 位址 M2，並在外面封裝(encapsulate)一層 tunnel 的標頭(header)，送往 VM2 所在的

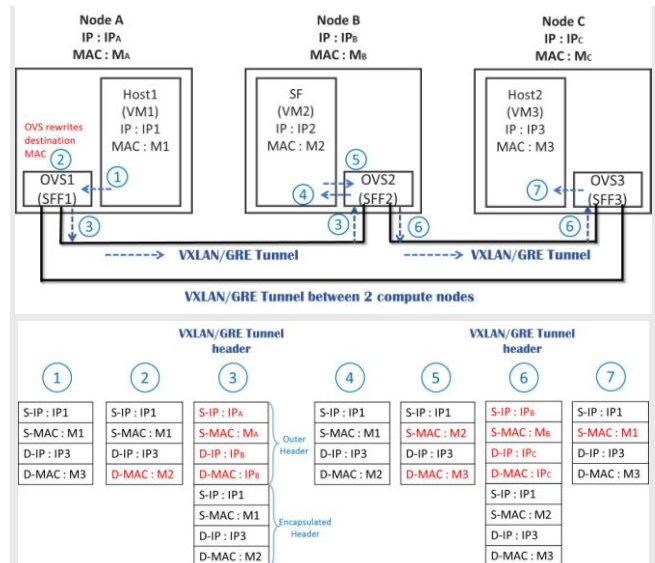


圖7 以修改 MAC 位址搭配 VXLAN 或 GRE tunneling 而實現服務功能鏈結的範例

Node B。Tunneling 後的封包送到 Node B後會將外層的 tunnel 標頭解封裝(decapsulate)，OVS 看到解封裝後的封包目的端 MAC 位址為 VM2，會將該封包送給 VM2 (SF)處理。當 VM2 處理完之後，預設會將封包來源端 MAC 位址設為自身MAC 位址 M2，並送往目的端 IP 位址所在，即 IP3。該封包經過 OVS2 時，會被 OVS2 修改為下一站的 MAC 位址，在此例中即為終點站 VM3，因此目的端 MAC 被修改為 M3 後經過 tunnel 封裝轉發至 Node C。最後封包到達 Node C 並解開 tunnel 標頭，OVS3 將封包的來源端 MAC 位址改為正確的 M1，目的端 MAC 位址改為正確的 M3 後，轉送到最終目的地 VM3，完成此次服務功能鏈結的任務。

5 . 結論

本文回顧了NFV技術發展的背景、過程，並整理、說明相關重要開源計畫 OPNFV 的目標與現況，也呈現了兩種 SDN 實現SFC 的方法，且其中一種已在 OPNFV 平台上被採用及實作。以 SDN為基礎的自動化SFC技

術，搭配各種網路功能的 NFV 虛擬化技術，將使得各式各樣 on-demand 的創新網路服務加速誕生。期望本文能讓更多資通訊從業人員，對於 SDN 與 NFV 的技術差異性、互補性、重要性有更進一步的認識，從而使得有更多人力資源能夠投入相關技術研發，或是進行相關開源計畫的追蹤及貢獻。

參考文獻

- [1] OpenFlow Switch Specification v1.0.0. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf>
- [2] OpenFlow@Google, ONS 2012. <http://www.opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf>
- [3] S. Jain et al., “B4: Experience with a Globally-deployed Software Defined Wan”; SIGCOMM, pp.3-14, 2013.
- [4] Network Functions Virtualization—Introductory White Paper, Oct. 2012. http://portal.etsi.org/NFV/NFV_White_Paper.pdf
- [5] A. Hakiri and P. Berthou, “Leveraging SDN for the 5G Networks: Trends, prospects and challenges”, Mobile and Wireless Communications, pp. 1-23, 2015
- [6] Network Functions Virtualization – Update White Paper, Oct. 2013. http://portal.etsi.org/NFV/NFV_White_Paper2.pdf
- [71] Network Functions Virtualization – Update White Paper#3, Oct. 2014. http://portal.etsi.org/NFV/NFV_White_Paper3.pdf
- [8] IETF RFC 7665, Service Function Chaining (SFC) Architecture. <https://tools.ietf.org/html/rfc7665>
- [9] Introduction to OpenDaylight. <https://wiki.opendaylight.org/view/Open>

Daylight SDN Controller Platform (OS CP):Main

- [10] Open Network Operating System (ONOS) website. <http://onosproject.org/>
- [11] Open Platform for NFV (OPNFV) website. <https://www.opnfv.org/>
- [12] OpenStack overview. <https://www.openstack.org/software/>
- [13] DPDK: Data Plane Development Kit. <http://dpdk.org/>
- [14] OpenDataPlane project website. <http://www.opendataplane.org/>
- [15] Open vSwitch project website. <http://openvswitch.org/>
- [16] OpenContrail website. <http://www.opencontrail.org/>
- [17] KVM: Kernel-based Virtual Machine. http://www.linux-kvm.org/page/Main_Page
- [18] S. Weil et al., “Ceph: A Scalable, High-Performance Distributed File System”; Proceedings of the 7th Conference on Operating Systems Design and Implementation (OSDI '06), 2006.
- [19] Ceph homepage. <http://ceph.com>
- [20] IETF draft, Network Service Header. <https://tools.ietf.org/html/draft-ietf-sfc-nsh-05>

作者簡介

許名宏



現任資料中心架構與雲端應用軟體組資深工程師。專長為軟體定義網路技術、資訊檢索。

[E-mail: mhhsu@itri.org.tw](mailto:mhhsu@itri.org.tw)

顏至寬



現任資料中心架構與雲端應用軟體組副工程師。專長為軟體定義網路技術、雲端運算、Linux系統管理。

[E-mail: simonyen@itri.org.tw](mailto:simonyen@itri.org.tw)