軟體定義網路中邊界閘道協議之實作
SDN-Based BGP Implementation

王韋程　　黃秉鈞
Wei-Cheng Wang, Ping-Chun Huang

中文摘要

軟體定義網路(Software Defined Networking；SDN)發展至今無庸置疑地已是下個世代網路技術的趨勢，但在現實網路環境中，基於距離、地緣與管理者不同等因素，不可能將所有網路設備皆交由統一的SDN控制器(controller)管理，因此傳統網路中的自治系統(Autonomous System; AS)概念將不會被取代，而一個SDN自治系統如何與外界網路連接則成為新的問題，本論文將一個SDN控制器或一個SDN控制器叢集(Cluster)所管理的網路視為一個自治系統，並使用邊界閘道協議(Border Gateway Protocol; BGP)作為與外界溝通的協議，以達到無論是傳統網路或SDN網路都可界接之效果。

本論文說明如何在SDN架構下，布建一個BGP交換中心，使得此SDN網路擁有與網際網路(Internet)中的自治系統溝通的能力；同時此BGP交換中心可透過SDN架構獲得兼容性(Compatibility)、操作彈性(Operational Flexibility)、高可靠性(High Availability)、可擴展性(Scalability)以及廠商獨立性(Vendor Independence)。

Abstract

Software Defined Networking (SDN) is no doubt become a trend of network technology of next generation. Due to distance, geopolitical relations and administrations, it's impossible to replace the concept of the autonomous system entirely with central SDN controller in the real world environment, So the autonomous systems concept in the traditional network will not be replaced. How An SDN autonomous system can connect to external network will become a new problem. In this paper, we viewed a network controlled by an SDN controller or a cluster of SDN controller as an autonomous system, and use Border Gateway Protocol (BGP) to interact with the rest of the Internet, in order to achieve the effect of both the traditional network and SDN network can be connected.

In this paper, we explain how to build an SDN-based BGP peering point that can make SDN network communicate with autonomous systems on the Internet. SDN's concept and technology can make BGP peering point have more compatibility, operational flexibility, high availability, scalability, and vendor independence.

關鍵詞(Key Words)

軟體定義網路 (Software Defined Networking；SDN)
邊界閘道協議 (Border Gateway Protocol；BGP)
自治系統 (Autonomous System；AS)
實驗場域布建 (Field Trial Deployment)
網路路由軟體 (Network Routing Software)

1．前言

近年來，隨著軟體定義網路(Software Defined Networking; SDN)的相關技術日漸成熟，無論是在資料中心(Data Center)、企業(Enterprise)或是電信服務商(Service Provider)中都有實際部署SDN[1]網路的例子，SDN的精神是為了讓網路的管理變得更集中，而且具有可程式化的機制，對電信網路業者或企業的資料中心網路服務來說，可以藉此獲得自動處理與動態因應變化等好處，這樣一來即可節省支出，不需經常購買專用架構網路設備，並且於日常維運時，SDN的網路部署能支援隨成長規模大小付費的模式，杜絕過度提供資源造成浪費。同時，SDN可減少IT服務日常維運的開銷，讓網路透過程式演算法的方式進行控制，而搭配一些可程式化程度提升的網路設備元件，能讓網路的設計、部署、管理、規模延展更為容易。

然而自成一系的SDN網路自治系統，卻缺乏與其他傳統網路或SDN網路溝通的能力，SDN可依據封包目的地決定路由規則，但僅限於目的是在該SDN網路中，也就是無法替別的自治系統轉送封包，此為一嚴重問題。在傳統網路中，經常使用IP(Internet Protocol)層的邊界閘道協議(Border Gateway Protocol; BGP)[2]作為網路與網路之間交換資訊的協議，透過BGP即可學習與附近自治系統間的鄰居關係，並利用維護IP路由表決定封包前往某個子網路的優先路由，甚至規劃冗餘路徑以達到高可靠度的效果，本論文選擇BGP作為與外部網路界接的協定，使用BGP可使傳統網路不需進行任何改變即可與SDN網路互通，而BGP本身的演算法也具有良好的可靠性與擴展性。

本論文貢獻如下所述，提出一種在SDN網路中布建BGP交換中心的方法，使用網路路由軟體(Network Routing Software)[3]模擬路由器(router)接收來自外部的eBGP (external BGP)訊息，路由器學習eBGP (external BGP)訊息並轉換為內部的iBGP (internal BGP)訊息與SDN控制器上的SDN-IP[4]應用程式溝通，本實驗場域選用的網路路由軟體為Quagga[5]。目前SDN的優勢即為在控制器上面開發網路控制軟體，透過OpenFlow[6]協定設定OpenFlow交換機上的繞送規則，來達到軟體定義網路的效果。OpenFlow是控制器與網路設備間溝通的一種協議，OpenFlow的協定讓網路設備可以根據網路封包的第二層至第四層內容進行封包的修改及轉傳，與一般第二層交換機或第三層路由器相比具備更高的彈性，可大幅提升網路的可操控性。本實驗場域透過一個SDN控制器上的網路控制軟體稱為SDN-IP，此軟體可以接收網路路由軟體發送的iBGP (internal BGP)，從中學習如何建立以目的地IP前綴(IP prefix)為匹配欄位的轉送規則，並透過OpenFlow協定將轉送規則設定於OpenFlow 交換機中。本實驗場域將SDN控制器與SDN-IP與BGP speaker結合，打造一個SDN化的BGP交換中心場域，並且與外部網際網路連結，驗證此作法的可行性。

在本論文中，首先在第2章介紹BGP基本運作機制以及SDN控制器的架構，之後在第3章，介紹針對建立BGP交換中心所需的特性而設計的SDN-IP系統架構以及如何透過開放原始碼(Open Source)專案軟體達成自動部署與驗證，並在第4章說明本實驗場域現況與未來規劃，最後在第5章對本論文進行總結。

2．相關技術介紹

2.1 Border Gateway Protocol

2.1.1 BGP簡介

BGP (Border Gateway Protocol)即為邊界閘道協議，是互聯網上一個核心的去中心化自治路由協定。透過維護IP路由表或『首碼』表來實作自治系統(Autonomous System; AS)之間的可達性，屬於向量路由協定。BGP不使用傳統的內部閘道協定(Interior Gateway Protocol; IGP)的指標，而使用基於路徑、網路策略或規則集來決定路由。因此，更適合被稱為向量性

協定，而非路由協定。

2.1.2 eBGP與iBGP控制訊息

在BGP中，各對路由器會透過使用port 179的半永久性TCP連線來交換繞送資訊，如圖1所示，SDN控制器可透過port號碼判斷是否為BGP控制封包。針對每筆TCP連線，連線兩端的路由器稱作BGP對等點(BGP peers)或BGP speaker，而TCP連線加上所有透過該連線傳送BGP訊息，則稱作BGP會談(BGP session)[7]。跨越兩組自治系統的BGP會談則稱作外部BGP會談(eBGP session)，則路由器必須直接連接。而在相同自治系統內的路由器之間的BGP會談則稱作內部BGP會談(iBGP session)，路由器不需要直接連接。

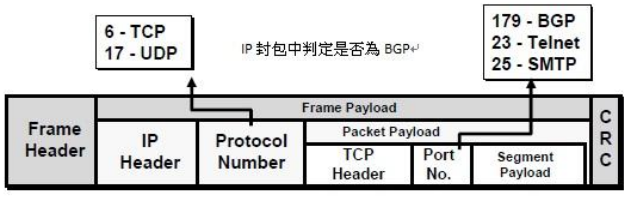

圖 1　IP封包中判定是否為BGP

2.1.3 BGP同步流程

(1) 多條路徑時，BGP Speaker只選擇最優的路徑給自己使用，例如最短路徑。

(2) BGP Speaker只把自己使用的路由通告給鄰居。

(3) BGP Speaker從eBGP (external BGP)獲得的路由會向它所有BGP鄰居通告(包括eBGP (external BGP)和iBGP (internal BGP))。

(4) BGP Speaker從iBGP (internal BGP)獲得的路由不向它的iBGP (internal BGP)鄰居通告。

(5) BGP Speaker從iBGP (internal BGP)獲得的路由是否通告給它的eBGP (external BGP)鄰居要依IGP和BGP同步的情況來決定。

(6) 連接一旦建立，BGP Speaker將把自己的BGP路由通告給新鄰居。

2.2 SDN Controller

2.2.1 ONOS Controller簡介

OpenFlow在2006年由一位史丹佛大學的博士生Martin Casado所提出，經過兩三年的發展後，開始有許多OpenFlow的controller陸續出現，如NOX[8]、POX[9]、Floodlight[10]...等，這些controller皆屬於較不成熟的原型樣板(prototype)控制器，主要用於驗證OpenFlow功能以及證明軟體定義為可行的概念驗證(Proof of Concept)。然而在2014年的12月，ON.Lab(Open Networking Lab)主導推出的開源控制器ONOS(Open Networking Operating System)[11]標榜為適合電信供應商使用的高可擴充性、高可靠性、高性能以及擁有抽象能力的SDN控制器，可以讓開發軟體與服務更便利。

2.2.2 ONOS Controller cluster架構

圖2是ONOS的架構圖，ONOS的特色為它是一個叢集(cluster)的架構，一個ONOS cluster是由多個ONOS實例(instance)[12]所組成，如圖2中為四個ONOS實例組成的ONOS叢集，透過叢集的架構將管理交換機的工作分散在各個ONOS實例中，利用可以隨時增減ONOS實例來達成高可擴充性，同時ONOS實例可以快速容錯轉移(failover)所以具備高可靠性，以及透過將工作負擔分散於各ONOS實例上來達到高性能。

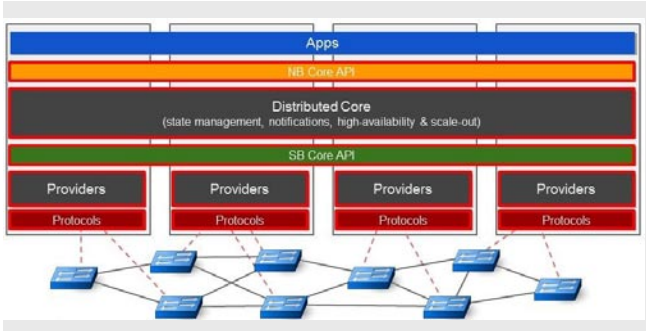

圖 2　ONOS系統架構圖

2.2.3 ONOS Controller南北抽象層

由於ONOS採用叢集架構，所以每個實例間必須要有同步的機制才能維持一致的狀態，ONOS採用Raft[13]作為同步狀態的工具，圖3為ONOS核心模組，紅色與灰色部分為子系統，每個子系統皆會維護自己的同步狀態，進而提供南向或北向使用。南向的部分，ONOS

可以支援各類南向協議，例如OpenFlow、NETCONF、OVSDB...等。北向的部分，ONOS提供網路狀態，其中包含網路拓樸、網路設備狀態以及路徑狀態等等；ONOS也提供意向框架(intent framework)給應用程式使用，意向框架允許應用程式不需考慮網路實際情況，只須制定政策，使用想怎麼做(what)取代該怎麼做(how)。如圖4，應用程式只需定義用戶到用戶之間需要連線，ONOS控制器會自動在交換機上設定相關的規則達到連線效果。此框架使得開發者能專注於應用程式的設計，而不須顧慮底下SDN的行為，使得開發應用程式更便利。

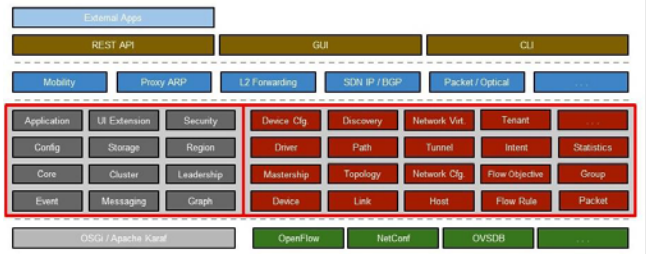

圖 3　ONOS子系統架構圖

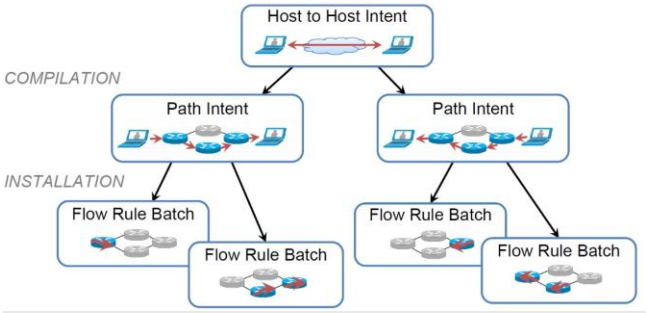

圖 4　ONOS意向框架(intent framework)

3 · SDN-IP系統介紹

3.1 需求、架構目標

此 BGP 交換中心系統主要目的是想要利用 SDN 建構出 Transit Network，其中整合許多知名 Open Source 專案，如 ONOS、Quagga、Open vSwitch 等，透過這些專案的結合，可以讓一個 SDN 網路具備與外界溝通的能力，而當中最重要的功能即為 ONOS 提供的 SDN-IP 應用程式，SDN-IP 負責從 iBGP (internal BGP)訊息中學習如何處理自治系統(AS; Autonomous System)之間的封包轉傳。從 BGP speaker 的角度看，SDN 網路其表現就像是單一的 AS，無論是跨 AS 或者是內部 AS 的路由資訊交換皆需要透過 BGP 協定進行溝通。而 ONOS 則提供一個 SDN-IP 應用程式針對 BGP 資訊交換或者是封包轉送的處理，已達到系統整合的目的，如圖 5 所示，由兩個 BGP speaker 接收來自外部 AS 的 eBGP 訊息學習並轉換成 iBGP 訊息給 SDN-IP 應用程式，SDN-IP 則將需要設定的路由規則透過 ONOS 控制器下達交換機。

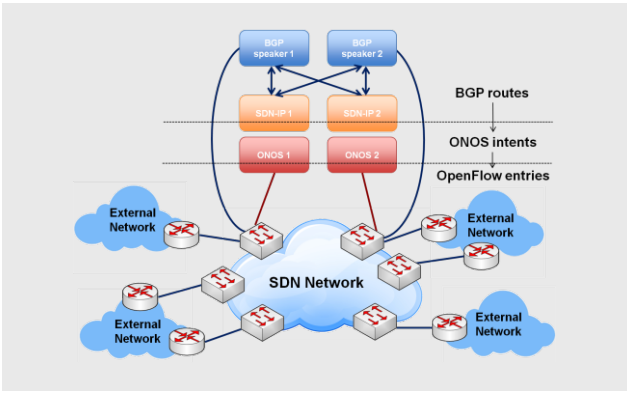

圖 5　與外部研究單位連線示意圖

本 BGP 交換中心的架構目標為利用傳統 BGP 協議達到能與任何網路共存的兼容性(Compatibility)、透過 ONOS 可隨時增減叢集實例的特性以及不同 BGP 布署方式，如 full-mesh、route reflectors 及 confederation，增加操作靈活性(Operational Flexibility)、同時使用多個 Quagga BGP speaker 維持高可用性(High Availability)、可擴展性(Scalability)，最後透過使用標準協議、商用無品牌設備以及開源專案，讓本 BGP 交換中心可達到協定兼容性及廠商獨立性(Protocol Compatibility and Vendor Independence)的目標。

3.2 系統架構

3.2.1 跨國網路連線

為驗證本 BGP 交換中心為實際可行的，因此本實驗場域與美國及韓國研究單位的 SDN BGP 交換中心進行 L2 Connections Peering，故商請國家高速網路與計算中心協助對海外線路及連線至交通大學計算機中心事宜。關於海外

連線部分，如圖 6 所示，連接韓國 KREONET 線路，中間透過位於芝加哥的 StarLight NOC 幫忙跳接至韓國線路；而連接到美國 AMLIGHT 之路線，則是透過位於洛杉磯的 Pacific Wave NOC 跳接至佛羅里達大學的 SDN 維運單位。　　至於國內連線，則是請交通大學計算機中心協助對接至實驗場域，以完成全線接通的目標。

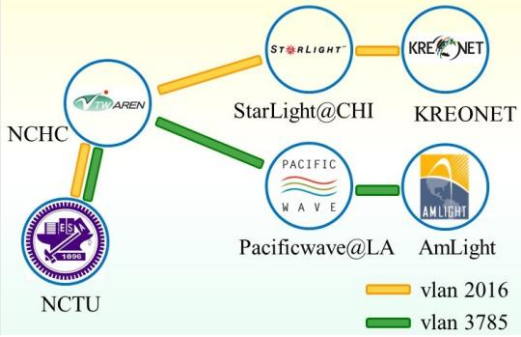

圖 6　與外部研究單位連線示意圖

3.2.2 實體架構

實驗場域內的實體機器，共計五台機架式伺服器(Rackmount Server)，上面預設皆裝載 VMWare 虛擬化技術，一台專門執行具有三個 ONOS 實例的 ONOS 叢集模式，兩台專供 Quagga 擔任 BGP Speaker 處理來自對內外的 BGP Traffic，最後兩台則建立不同 AS 的網路，來進行跨自治系統網路(Autonomous System Network)的封包傳輸行為，如圖 7 所示。另採購四台具備 OpenFlow 功能的交換機，可被 ONOS 控制完整控制，另包含一台 L2 ToR(Top-of-Rack)交換機，透過 Out-of-Band 的方式管理所有的伺服器及交換機，如圖 8 所示。

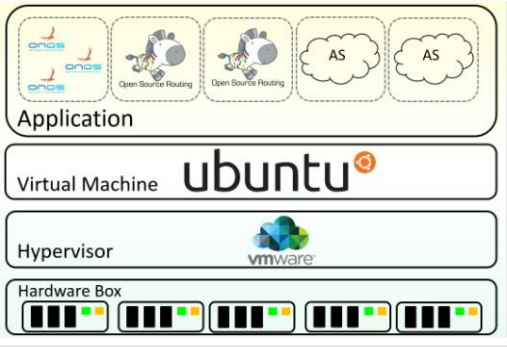

圖 7　Software Stack

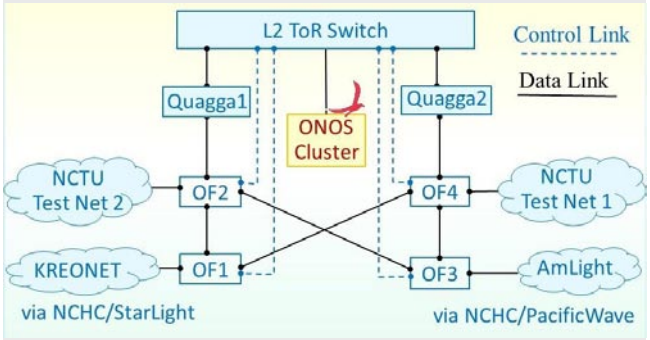

圖 8　實體架構圖

3.2.3 線路規劃

針對 Data Plane 的部分，本實驗場域內的 OpenFlow 交換機之間皆透過 40G DAC(Direct Attach Copper)連線，以達到最高的交換封包效率，而對外的連線則採用單模光纖傳輸。針對 Control Plane 部分，則使用 CAT-6 來確保網路傳輸的可靠度。其中 VMWare 內的網路設計，則大量採用 Virtual Standard 交換機及 Virtual Distributed 交換機，並配合 VLAN 來切開 OpenFlow Control Plane、Management Plane 及 Data Plane 的流向，邏輯上清楚地劃分出以上流量，如圖 9、圖 10 所示，iBGP 與 eBGP 屬於 Management Plane 範疇，而 OpenFlow 的訊息則是屬於 Control Plane，其餘網路封包皆屬於 Data Plan，用特定 VLAN 區分，以確保後續維護人員進行網路除錯會具有比較好的可讀性。

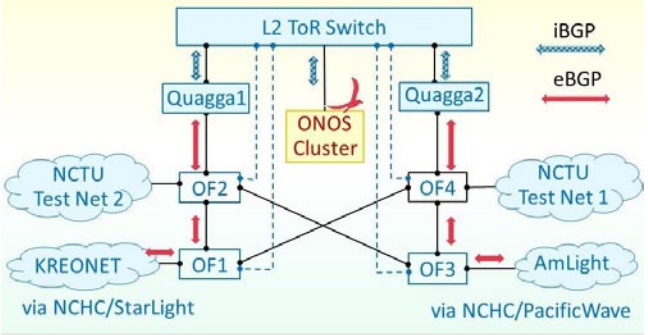

圖 9　內部BGP與外部BGP示意圖

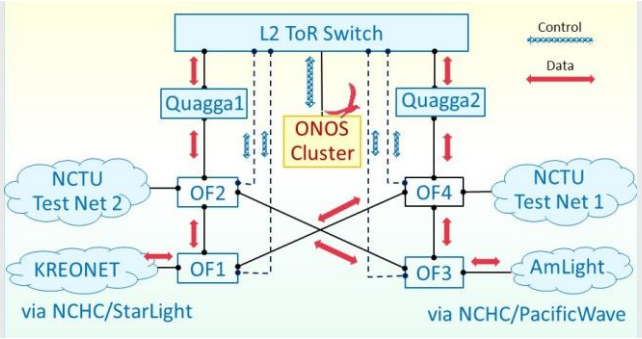

圖 10　控制層與資料層示意圖

3.3 自動部署

因整個 SDN BGP 交換中心實驗場域所使用的虛擬機總數眾多，以作業系統及功能區分，共分為七大類，總數二十二台。為避免操作人員因不熟悉驗證流程及大量減少重複動作，故導入資訊自動化維運工具進行驗證，本場域採用的方案是已被 RedHat 收購的 Ansible[13]專案。

Ansible 是近期 DevOps 議題當中經常出現的專案，主要包含兩種自動化執行方式: Ad-Hoc command 和 Playbook 撰寫，而本專案因僅需確認網路連線功能驗證及一些設定檔顯示的功能，故採用 Shell Script 腳本語言配合 Ansible Ad-Hoc Command 結合進行功能性驗證。

本實驗場域利用此工具大幅改善了反覆操作及耗費大量人力的時間，且可以依據程式碼描述來反覆確認是否功能定義有誤，以達到基礎設施即代碼 (Infrastructure as Code)的目標。

3.4 驗證 BGP 交換中心

本實驗場域在未與其他研究單位介接之前，分別利用兩台伺服器裡面架設好 Quagga、Ubuntu Desktop 及設定好相關的虛擬交換器設定，建立起各一最小單位的 AS 網路架構。於 NCTU Test Net 1 裡放置一台 VLC[14]伺服器，持續地撥放來自台灣觀光局授權的宣傳影片，而 NCTU Test Ne t2 則放置 VLC 撥放器，接受來自 VLC 伺服器的 UDP 封包並且撥放。當中因為兩邊路由是完全不一樣且所屬的 ASN 也不一樣，中間需透過 BGP 交換中心所帶來的功能，進行雙邊路由交換及依據 BGP Speaker 所提供的路由資訊轉送 Data Plane 的封包至正確的連線埠。透過此方式本實驗場域可以在未與國際間連線前，先行確認本實驗場域之預期功能運作正常。

4 · 實驗場域現況與未來規劃

4.1 實驗場域現況

本實驗場域於今年三月初布建完成，並與世界各地的SDN學術研究網(Research&Education Network)連接，如圖11所示。本次參與SDN-IP部署計畫的有美國的Internet2、南美的AMLIGHT、歐洲的GÉANT、澳洲的AARNet以及韓國的KREONET，並於今年三月中的ONS 2016 (Open Networking Summit 2016)中共同展示如圖12，該次展示中，本實驗場域為關鍵核心，首先展示各學術研究網皆為透過BGP方式建立連線，並能順暢撥放即時串流影片，接著為了展示BGP冗餘路徑之特性，將台灣連往美國Internet2之連線切斷，此時BGP會自行將台灣與美國之間的路徑改為藉由韓國為中轉點的方式傳輸，並且即時串流影像保持不中斷，該次展示成果十分顯著，證明SDN網路間可透過BGP方式互相串連。

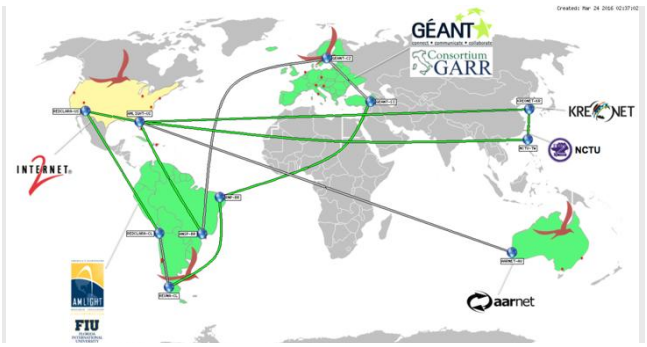

圖 11　SDN-IP全球部署情況

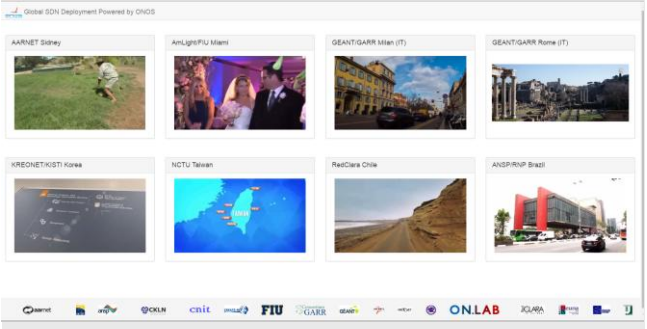

圖 12　SDN-IP ONS DEMO

證實SDN-IP計畫可行後，台灣也有單位陸續加入，如圖13所示，包含使用ONOS控制器及Pica 8交換機的國家高速網路與計算中心及清華大學和使用Ryu控制器及NEC交換機的中華電信研究院，本計畫除了證明SDN BGP交換中心可行外，同時也驗證了不同廠牌的交換機與SDN控制器同樣可以達到互相串連的效果，同時規劃繼續推廣此計畫，讓更多SDN網域互通。

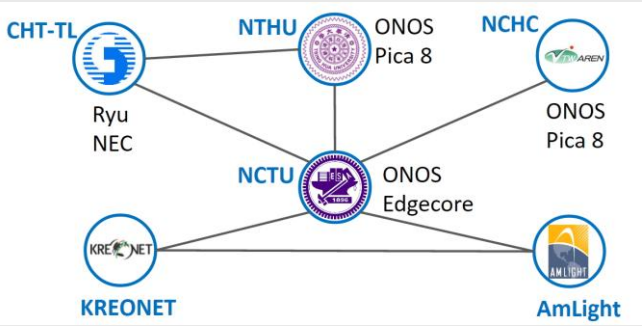

圖 13　ONOS系統架構圖

4.2 未來規劃

目前實驗場域已穩定，未來打算將資料層的網路拓樸改為fabric[15]架構，即為將交換機分為上層脊柱(spine)與下層樹葉(leaf)，增加網路使用率以及縮短傳輸路徑；同時增加流量監測服務，提供ONOS控制器作為Traffic Engineering的依據。

除此之外，本實驗場域預計測試不同控制器以及BGP軟體是否也擁有部建SDN之BGP交換中心的能力，例如NTT的GoBGP以及BIRD。

最後本實驗場域期望繼續使用其他開發與維運工具(DevOps)，例如監測工具Zabbix或ELK(Elasticsearch, Logstash, Kibana)專案；同時擴大應用Ansible進行全面自動部署與驗證，並規劃導入Puppet專案使DevOps效果更全面。

5 · 結論

SDN已是下世代網路技術的發展趨勢，世界各國各產業都投入資源在其中，但各個SDN網路要如何互通仍是一個問題。在本論文中，提出了一套基於SDN的BGP交換中心架構，利用全狀拓樸(Full Mesh Topology)將多個網路路由軟體(Quagga)與多個ONOS控制器實例連接，以達到基於SDN架構之BGP交換中心所需的兼容性、操作靈活性、高可靠性、可擴展性、協定兼容性以及廠商獨立性。如此一來，每個SDN自治系統可以管理內部網路，同時具備與外界網路交換訊息能力，以達到實際將SDN落實到網路世界裡，而非創造一個封閉的網路環境。

本論文除了實現基於SDN架構之BGP交換中心外，同時利用多種開源工具來達到自動化部署實例以及驗證的效果，透過自動化部署可使得管理網路更便捷，自動化驗證也可以使得人為失誤降到最低。

參考文獻

[1] Software Defined Networking (SDN), https://www.opennetworking.org/sdn-resources/sdn-definition

[2] Border Gateway Protocol (BGP), http://www.routeralley.com/guides/bgp.pdf

[3] Network Routing Software (Quagga), https://www.opensourcerouting.org/

[4] SDN-IP, https://wiki.onosproject.org/display/ONOS/SDN-IP+Architecture

[5] Quagga Router Suite, http://www.nongnu.org/quagga/

[6] OpenFlow, https://www.opennetworking.org/sdn-resources/openflow

[7] BGP session, http://sls.weco.net/node/10676

[8] NOX controller, https://github.com/noxrepo/nox-classic/wiki

[9] POX controller, https://openflow.stanford.edu/display/ONL/POX+Wiki

[10] Floodlight controller, http://www.projectfloodlight.org/floodlight/

[11] ONOS white paper, http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf
[12] Distributed ONOS, https://wiki.onosproject.org/display/ONOS/Distributed+ONOS
[13] The Raft Consensus Algorithm, https://raft.github.io/
[14] Ansible – IT Automation, https://www.ansible.com/
[15] VLC Media Player, http://www.videolan.org/vlc/
[16] Fabric Network, https://wiki.onosproject.org/display/ONOS/CORD%3A+Leaf-Spine+Fabric+with+Segment+Routing

作者簡介

王韋程

工研院資通所通路通訊服務技術部副工程師。交通大學資訊工程碩士，網際網路通訊技術、SDN技術等。
E-mail: wilsonwang@itri.org.tw

黃秉鈞

交通大學資訊工程學系碩士班，無線網際網路實驗室，SDNDS-TW Co-Founder，且為ONOS Ambassador Program一員，平時喜愛參與OpenSource專案，近年主要接觸SDN/NFV領域，擅長系統及網路整合。
E-mail: pichuang@cs.nctu.edu.tw