

SDN 網控與網管技術

黃建穎 林盈達

國立交通大學資訊工程系

Email: lyon0417.cs03g@nctu.edu.tw; ydlin@cs.nctu.edu.tw

September 15, 2014

摘要

本篇報告分析傳統網路與 SDN 在網路控制跟網路管理的差別，網路控制中有別於傳統網路分散式存在於網路設備，SDN 將控制層集中於控制器上，分離的控制層不必綁死於硬體的設備上，降低了硬體成本，並且控制器上可以進程式化，遠端去進行操控。不僅能簡單的了解路由情況，也能更靈活的去對應網路的變化；網路管理上傳統網路與 SDN 皆為集中式的管理，SDN 上 ONF(Open Networking Foundation)所提供的 OF-CONFIG 協定，定義一個開放介面用於遠端配置和控制 OpenFlow 交換器，在 OpenFlow 架構上整合交換器出配置節點進而管理網路設備，它可分割交換器群組，也能將配置點配置到傳統網路管理設備。並結合 NETCONF 協定進行配置行為。而實驗中得知現在實體開發上 OF-CONFIG 尚未發展成熟。與傳統的網路管理協定相比，除了配置上利用 NETCONF 有較好的功能，但是在於監控設備上協定沒有較明確的監控管理內容。通知上 ONF 有 OpenFlow Notifications Framework 規劃，但是目前沒有實作，整體可了解當設備繁多時，與傳統網路架構相比，在於 OF-CONFIG 協定規劃下會有較好的直觀性與較低的複雜度。

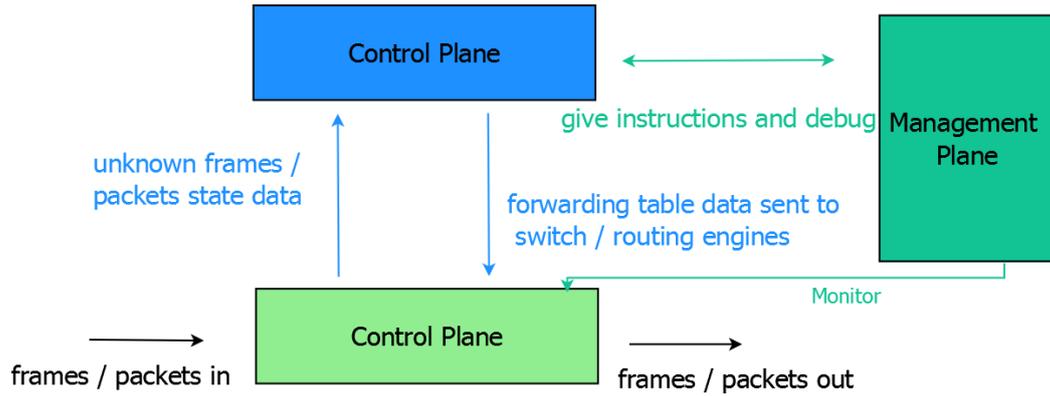
關鍵字: SDN、網路管理、網路控制、OF-CONFIG

1. 介紹

在於網際網路中，網路的管理與控制一直都是很熱門的話題。現今的網路管理已經是集中式的管理，而網路控制則是在於網路設備上，SDN(Software-defined networking)這新型態網路架構的崛起，將網路設備的控制層從資料層中分離並且集中交於控制器管理。使得網路控制與傳統產生了很大的改變。本文依序介紹網路控制與網路管理之間的差異性，在進而比較傳統網路與SDN中網路管理與網路控制的情況。其實驗為簡單的架構出傳統網路與SDN的管理環境，去測試配置情況、監控、通知與指令下達方式。了解目前在 OF-CONFIG(OpenFlow Management and Configuration Protocol)的協定下，對於網路管理的監控、通知功能向未發展完全，但在於擴展性而言，卻讓人看到更遙遠的未來。

網路設備架構介紹

圖一為基本設備架構圖，現今的網路設備架構基本上分為 Data plane(資料層)、Control plane(控制層)與 Management plane(管理層)。在資料層中，擔任承載封包流量傳輸的通道，並進行查表轉發至其他地點；控制層，負責根據封包內容運算出封包的路徑，在進而提供資料層轉發表的訊息；在管理層中則負責進行設備的管理監控，當有需求亦或是有突發的情況發生時進行修改與配置。

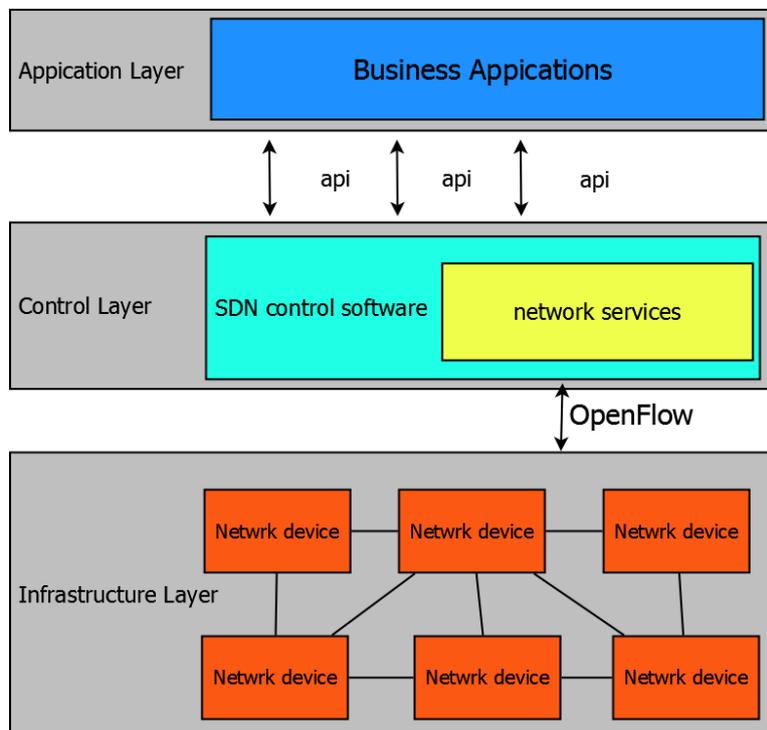


圖一 網路設備架構流程圖[1]

SDN 介紹

Software-defined networking[2]，是一種新的網路概念架構，目的是為了使資料通訊更靈活、易於操作和管理，並且達到節約成本、高相容性、能夠更好地應對應用程式發展和網路條件不斷變化的需求。

圖二其架構為SDN架構，將控制層從資料層中分離，集中於外部的控制器(controller)指揮，基礎網路設備僅需負責處理封包傳送工作。在控制器中以軟體的形式執行所定義的網路行為，藉由協定作為控制層與資料層之間溝通的橋樑(例如:OpenFlow)，以達到不必去更換硬體設備便可以部屬自己的網路環境。因控制器上的集中並程式化所有的控制層管理，使得在這樣的架構下我們可以在避免控制器負擔的情況下，利用各式各樣的 API 提供加值服務。網路設備不再是主角，軟體躍升為各種網管控制的關鍵，甚至未來可能出現網管 App Store，各式各樣的專用網路設備將被一支支網管 APP 所取代。



圖二 SDN 架構[3]

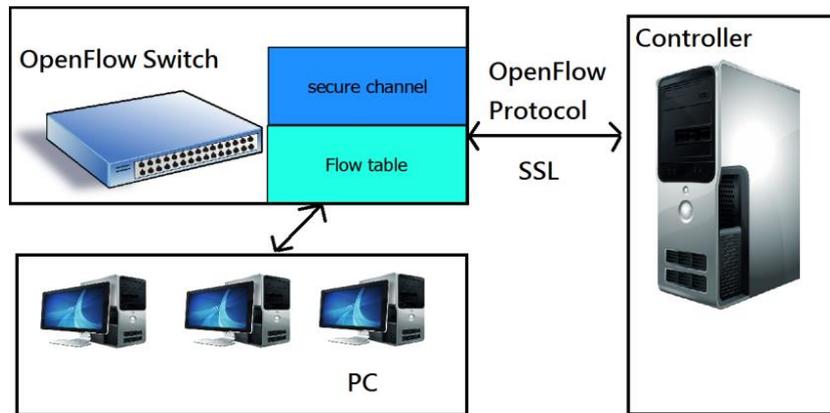
OPENFLOW 介紹

OPENFLOW[4]是開放的標準協定，用於建立控制器與資料層之間的溝通。

主要由三部分組成如圖三所示：

1. 定義網路封包傳輸路徑的 OpenFlow 路由表 (Flow Table)
2. 決定網路封包流向的軟體控制器 (Controller)
3. 作為傳輸溝通用的 OpenFlow 協定 (OpenFlow Protocol)

控制器與 OpenFlow 交換器硬體之間需先建立秘密頻道 (Secure Channel) 連接，通道之間以 SSL 加密技術加密，確保傳送之間的安全。網管人員可藉由 OpenFlow 協定所提供的開放原始碼與交換器硬體溝通，設定 Flow table，由 Flow table 定義封包的傳送路徑。



圖三 OpenFlow 架構[5]

SDN 與傳統網路整體比較

表一為整體上 SDN 與傳統網路整體比較，傳統網路的管理協定以 SNMP 與 NETCONF 作為範例，在於網路管理上基本都為集中式的管理，其差異在於 SDN 控制層的集中並且可程式化，網路設備剝離控制層，降低了硬體的成本。SDN 讓整體網路架構更具有較低的複雜度與更好的擴展性，更能適應網路的變化。

表一 SDN 與傳統網路整體比較

Problem	SDN	Traditional Network
Control plane	Centralized	distributed
Management plane	Centralized	Centralized
Management protocol	OF-CONFIG	SNMP, NETCONF
Programmable	✓	
Openness	✓	
Automation	✓	✓
cost	low	high
Agility	high	low
Structural complexity	low	high
Extensibility	high	low

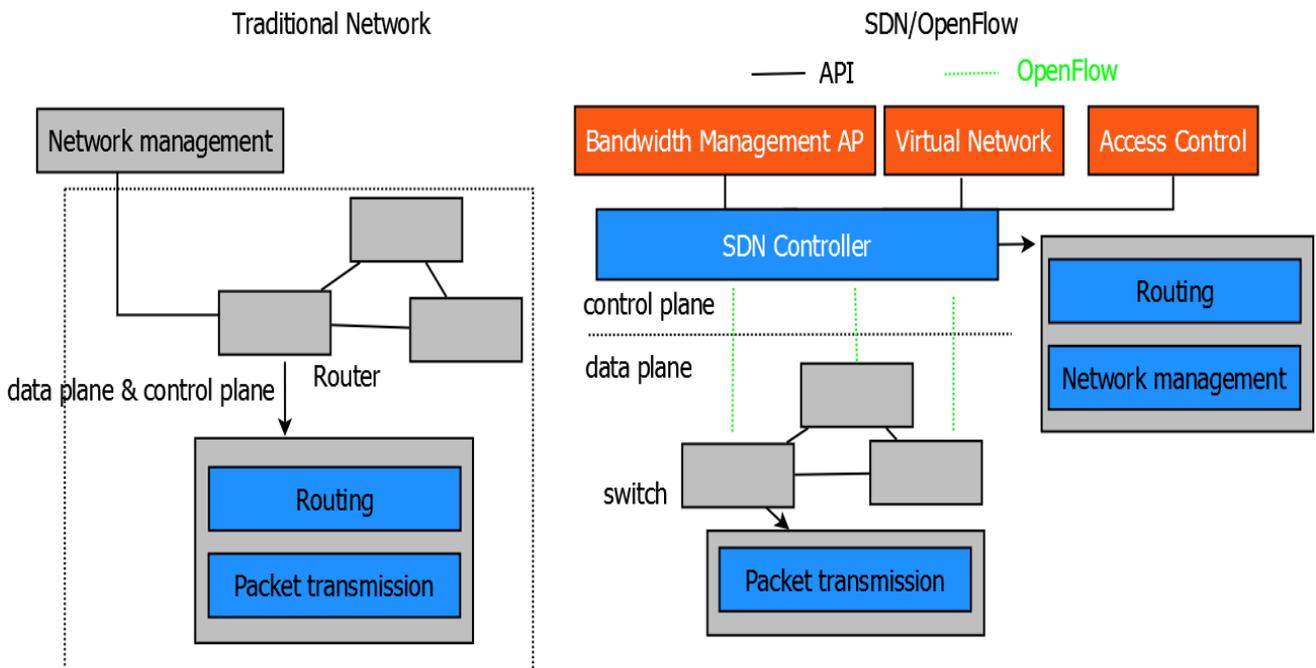
2. SDN 與傳統網路之網路控制比較

在控制層中最重要便是路由功能。傳統上，控制層與資料層向未分開，封包進入時會先進入資料層，再由控制層計算封包的路徑提供資料層轉發表，資料層再進行查表的動作將封包轉至其他地點。而管理層負責監控設備與流量，當某設備發生問題時，可以進行除錯或指派其他設備接替工作，避免網路停滯。控制層主要目的就是進行查表匹配的動作，藉由路由協定(BGP、OSPF)或者交換協定(STP、TRILL)進行計算後生成轉發表提供資料層進行查表轉發。

而在 SDN 上控制層集中於控制器上，利用 OpenFlow 做為與資料層連結的橋樑，利用 Flow 作為控制來源與目的地的連結。每一台交換器中包含多個 Flow table 與一組 Group table 進行查表匹配，控制器一樣負責計算，但分離的控制層不必綁死於硬體的設備上，而可在於控制器上進程式化，並且可以遠端操控。不僅可以簡單明確的了解路由情況，這樣的架構大幅增加硬體設備的可變性。

如圖四所示，在傳統網路中，控制層與資料層都存在於網路設備上，管理設備是藉由分散式下達指令，必須針對每種不一樣型態交換器或路由器設定，因資料層與控制層同時在網路設備中，常使得控制層宛如黑箱一般不透明，是藉由各自的廠商定義其規範。各式各樣的網路管理技術或是網路作業系統軟體，網管軟體彼此之間難以相容，一旦企業購買某一廠牌的設備，未來更新設備時就必須遷就於該廠牌的網管功能。

在 SDN 上網路管理與網路控制的功能交由控制層的控制器負責，開發者可以開發應用軟體部署於控制器內，亦或是透過 API 架設其他加值服務。控制器藉由標準規範(OpenFlow)傳達訊息，並下達指令給資料層的設備，故網路設備則負責查表與封包的傳送。



圖四 SDN 與傳統網路架構比較圖[5]

3. SDN 與傳統網路之網路管理比較

傳統網路管理方法

在傳統網路管理上，一般而言是建設在一些網路管理的協定上，最常見的有 SNMP[6]與 NETCONF[7]等協定。

SNMP(Simple Network Management Protocol)主要是為管理員提供了一個網管平台(NMS)，又稱為管理站，負責網管命令的發出、資料存儲、及資料分析。被監管的設備上運行一個 SNMP 代理(Agent)，代理實現設備與管理站的 SNMP 通信管理站與代理端通過 MIB 進行介面統一，MIB 定義了設備中的被管理物件。管理站向代理申請 MIB 中定義的資料，代理識別後，將管理設備提供的相關狀態或參數等資料轉換為 MIB 定義的格式，應答給管理站，完成一次管理操作。

NETCONF(Network Configuration Protocol)是由 IETF 開發和標準化的網路管理協議。NETCONF 的一個獨特優勢是它涉及大量的設備強大的配置更改事務的支持。利用 XML 進行統一的設備配置。

表二為根據基於協定上對於 SNMP 與 NETCONF 的比較，簡單分為配置、通知、監控、建模能力做比較，我們可以得知出 NETCONF 對於配置方面有較完善的功能，而 SNMP 在監控設備情況下表現較為突出。

表二 NETCONF 與 SNMP 的比較

	SNMP	NETCONF
Configuration	High	Low
Notification	Low	High
Monitoring	Low	High
Modeling capabilities	High	Low
Language	YANG	SMI(SMIV2)

SDN 的網路管理方法

在於 SDN 的網路管理上，主要區分為

1. 利用在 ONF(Open Networking Foundation)所提供的協定 OF-CONFIG[8]管理交換器與控制器的情况。
2. 不利用 OF-CONFIG 的協定，利用建立在控制器基礎上的 Module 或 Application 達到網路管理的效果。

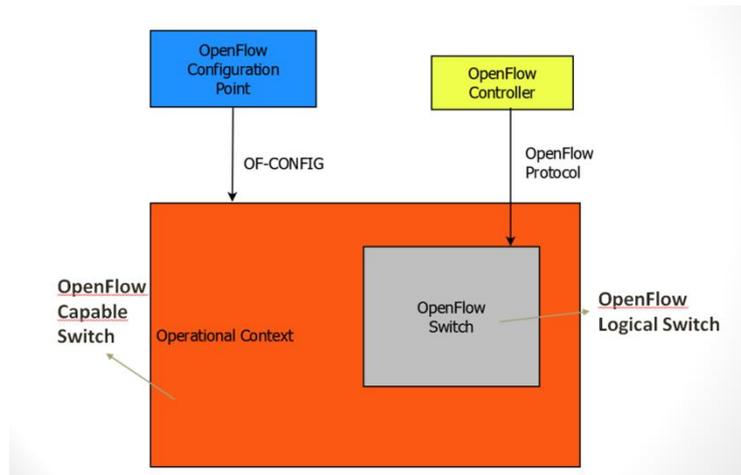
以下介紹這兩項的差異。

OF-CONFIG

OF-CONFIG(OpenFlow Management and Configuration Protocol)的本質是提供一個開放介面用於遠端配置和控制 OpenFlow 交換器，但是它並不會影響到 Flow table 的內容和資料轉發行為，對即時性也沒有太高的要求。具體地說，諸如構建 Flow table 和確定資料流程走向等事項將由 OpenFlow 規範進行規定，而諸如如何在 OpenFlow 交換器上配置控制器 IP 位址、如何

對交換器的各個埠(port)進行 enable/disable 操作則由 OF-CONFIG 協議完成[8]。

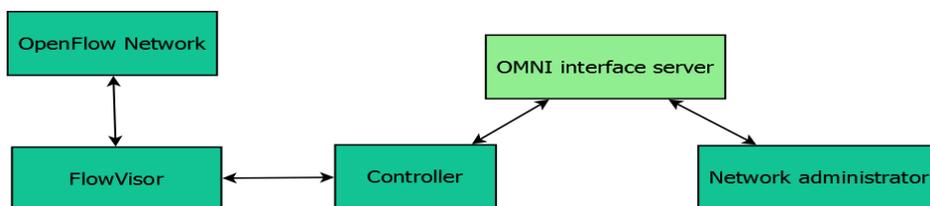
OpenFlow 交換器上所有參與資料轉發的軟硬體(例如埠、佇列等)都可被視為網路資源，而 OF-CONFIG 的作用就是對這些資源進行管理。如圖五所示，OF-CONFIG 在 OpenFlow 架構上利用 OpenFlow capable switch 管理 OpenFlow logical switch 整合出被稱作 OpenFlow Configuration Point 的配置節點。這個節點既可以是控制器上的一個軟體進程，也可以是傳統的網管設備，它通過 OF-CONFIG 協定對 OpenFlow 交換器進行管理，並結合 NETCONF 協定進行配置行為，但是在協定規範下缺乏了關於通知方面的規範整理。而 OpenFlow Notifications Framework[9]在規劃通知者(交換器)與接受通知者(控制器)的角色。



圖五 OF-CONFIG 架構圖[8]

不使用 OF-CONFIG 的網管方法

在此以 Openflow Managemet Infrastructure(OMNI)[10]作為範例，OMNI 是由 GTA/UFRJ 開發出的一套整合工具。如圖六所示，它建立在 NOX 控制器上，OMNI 提供一個遠程管理 OpenFlow 網路的介面，並允許監測和 Flow 的動態配置等，這些行為皆可在於它所提供的網頁 GUI 上執行。此外也 OMNI 提供基於開發介面管理的多代理平台，此工具呈現低響應時間回覆系統管理員或由代理請求的動作。架構上基於利用 FlowVisor 進行切割多個邏輯網路切片(slice)，每一個網路切片擁有不同的轉發邏輯策略。在切片模式下，多個控制器能夠同時管理一台交換器，多個網路實驗能夠同時運行在同一個真實網路中，網路管理者能夠並行地控制網路，因此網路正常流量可以運行在獨立的切片模式下,從而保證正常流量不受干擾。



圖六 OMNI 簡單架構圖[10]

OF-CONFIG 與 OpenFlow 的差異

表三比較 OF-CONFIG 與 OpenFlow 的差異，OpenFlow 主要是提出了由控制器向 OpenFlow 交換器發送 Flow table 以控制資料流通過網路所經過的路徑的方式，但是並沒有規定怎樣管理和配置這些網路設備，而 OF-CONFIG 就是為解決這一問題而提出的。他們之間主要的差距在

於一個負責定義如何執行如表三前四項，一個負責配置管理這些功能，並且利用 OpenFlow Configuration Point 去切割配置交換器管理群組，並且可將配置點配置於傳統網路管理設備上做結合。

表三 OF-CONFIG 與 OpenFlow 的差異

	OF-CONFIG	OpenFlow
Routing		✓
Modify table message		✓
Switch connect with controller(s)		✓
Modify ports and queues		✓
Modify the configuration	✓	
Cutting Groups	✓	
Combine with the traditional network management?	✓(OpenFlow configuration point)	
TCP/TLS	✓	✓
BEEP	✓	

OF-CONFIG 與 SNMP 的差異

表四為傳統網路管理協定與 SDN 網路管理協定的比較，OF-CONFIG 與 SNMP 會類似於 NETCONF 的比較，因他在配置支持 NETCONF 的協定，但目前版本(OF-CONFIG 1.2)中只支援到關於配置的動作。除了語法的差異，主要的差異性主要在於下達指令的方式。OF-CONFIG 是利用 XML 集中下達指令，而 SNMP 對設備下達指令是個別分散式下達。

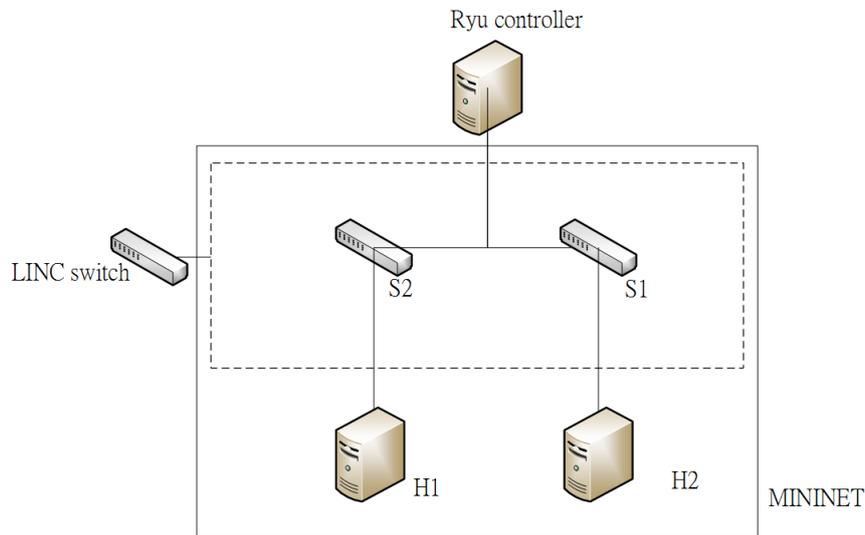
表四 OF-CONFIG 與 SNMP 的差異

	OF-CONFIG	SNMP
Data Unit	XML	Object
Data acquisition	Centralized	Distributed
Support	NETCONF	
Issued commands	OpenFlow Configuration Point(controller)→Switch	Manager → Agents
Language	YANG	SMI(SMIv2)

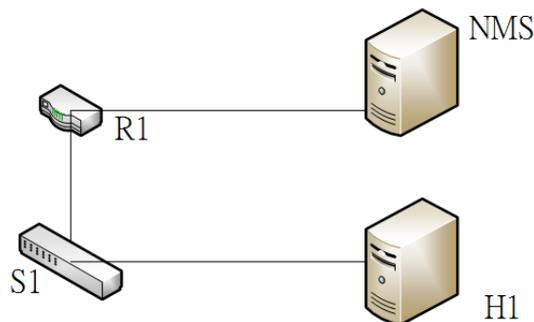
4. 網管實驗比較

此實驗是測試目前 OF-CONFIG 網路管理功能情況，並簡單的建立傳統網路管理環境做為比較組。測試配置、監控設備、通知、指令下達方式，此實驗設定為二：

1. 如圖七所示，利用 LINC switch[11]作為 OF-CONFIG 結構中的 OpenFlow capable switch 結合 MININET[12]模擬 OpenvSwitch(S1、S2)與 Host(H1、H2)與 Ryu[13]控制器架構出利用 OpenFlow Configuration Point 進行配置等功能。
2. 如圖八所示，利用 GNS3[14]套件模擬 c3600 Router(R1)、基本 Switch(S1)與 Host(H1)，NET-SNMP[15]與 MRTG 建立基本 SNMP 管理環境。其中 NMS 為 ubuntu13.04 的環境安裝 NET-SNMP5.7.2.1 作為監控端，。



圖七 實驗設定一



圖八 實驗設定二

表五為這次實驗項目比較的結論，OF-CONFIG 的架構向未成熟。除了在配置上利用 NETCONF 比起 SNMPSET 有較為優秀的表現，但在於 LINC switch 的操作必須用到 Erlang[16] 語法進行操作是相對比較麻煩的部分，在指令的下達方面，雖然都是集中式的網路管理，但在於下達配置時基本上 SNMPSET 還是依循個別下達指令，而 OF-CONFIG 利用 XML 統一配置設備情況。在監控方面，現在目前沒有去支持 NETCONF 相關方面的動作，故無法藉由 OF-CONFIG 去進行像 SNMP 的 MRTG 顯示獲得設備 CPU 情況、硬碟空間、記憶體資訊。通知方面其實 OF-COFIG 有 OpenFlow Notifications Framework 作為支援，但尚未實施。OF-CONFIG 發展尚未成熟，但結構上可了解當設備繁多時，在 LINC switch 的管理之下，可以確實分割設備管理的配置點，讓管理設備上會有較好的直觀性與較低的結構複雜度。

表五 實驗情況比較表

	實驗設定一	實驗設定二
Configuration	High	Low
Notification	Low	High
Monitoring	Low	High
Data Unit	XML	Object
Data acquisition	Centralized	Distributed
Issued commands	OpenFlow Configuration Point(controller)→Switch	Manager → Agents

5. 結論

網路控制方面，SDN 將控制層集中於控制器上，分離的控制層不必綁死於硬體的設備上，並且控制器上可以進程式化，遠端去進行操控。在成本、結構複雜度、靈活性與擴展性上優於傳統網路架構；網路管理方面，OF-CONFIG 的本質是提供一個開放介面用於遠端配置和控制 OpenFlow 交換器，整合出的配置點可以進行網路設備切割與結合傳統網路架構。實驗比較證實表四協定 OF-CONFIG 與 SNMP 的差異，但依目前 OF-CONFIG 的發展，尚未將 NETCONF 的協定完全落實。再通知方面 OpenFlow Notifications Framework 也尚未實體成形。導致整體上目前的 OF-CONFIG 比不上傳統發展多年的協定，也比不上目前相關 SDN 的管理方法。但是在於擴展性與架構上很明顯可以見到 OF-CONFIG 更宏大的未來。

參考資料

- [1] G. Ferro, “OpenFlow and Software Defined Networking: Is it Routing or Switching?” Etherealmind, December 21, 2011.
- [2] ONF, Open Networking Foundation, <https://www.opennetworking.org/sdn-resources/sdn-definition/>
- [3] Xinguard, <http://www.xinguard.com/content.aspx?id=34>
- [4] OpenFlow Switch Specification 1.3.0, Open Networking Foundation, June 25, 2012.
- [5] ithome, <http://www.ithome.com.tw/node/7735>
- [6] SNMP, http://www.snmp.com/protocol/snmp_rfcs.shtml
- [7] IETF, “Network Configuration Protocol.” RFC6241, June, 2011.
- [8] OpenFlow Management and Configuration Protocol 1.1.1, Open Networking Foundation, March 23, 2013.
- [9] OpenFlow Notifications Framework 1.0, Open Networking Foundation, Dec 14, 2013.
- [10] Diogo M. F. Mattos, Natalia C. Fernandes, Victor T. da Costa, Leonardo P. Cardoso, Miguel Elias M. Campista, Lu´is Henrique M. K. Costa, Otto Carlos M. B. Duarte, “OMNI: OpenFlow MaNagement Infrastructure.” Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brazil, Nov

28-30, 2011.

[11] LINC-Switch, <https://github.com/FlowForwarding/LINC-Switch>

[12] Mininet, <http://mininet.org/>

[13] Ryu SDN Framework, <http://osrg.github.io/ryu/>

[14] GNS3, <http://www.gns3.net/>

[15] Net-SNMP, <http://www.net-snmp.org/download.html>

[16] Erlang, <http://zh.wikipedia.org/wiki/Erlang>

[17] T. Slattery, “Will SDN Be the Future of Network Change Management?” NOJITTER, September 04, 2013.

[17] C. Alaettinoglu, “Sticking Point : SDN Management Challenges.” Data center knowledge, February 11, 2014.

[18] B. Carroll, “Software Defined Networking vs. Traditional Networking.” Globalconfig, September 3, 2013.