實作 SDN 架構之網路監控系統

林子傑
台中教育大學資工系
jakevbn@gmail.com

施語筑
台中教育大學資工系
uvia820701@gmail.com

張林煌*
台中教育大學資工系
albertchang04@gmail.com

李宗翰
台中教育大學資工系
thlee@mail.ntcu.edu.tw

劉德隆
國家高速網路中心
tlliu@narlabs.org.tw

摘要

軟體定義網路(Software Define Network, SDN)是一種新型的網路架構，將傳統分散式的網路控制架構轉為集中式的控制架構並可藉由此架構來提供網路更多的創新與靈活性。本論文提出基於 SDN 的網路虛擬化應用及其實現方式，針對運用 SDN 架構於網路管理之議題，研發 SDN 網路監控系統於網頁端與行動裝置應用程式，提供 SDN 網路監控功能、更改、管理與維護機制。我們結合網頁資料與行動裝置應用程式的數據處理，同時為了讓使用 SDN 架構的網管人員能隨時監控流量，相較於網頁版的管理工具，行動裝置應用程式(Application, APP)具有大眾廣泛使用度以及更好的便利性，因此本論文也開發基於 SDN 架構的行動管理 APP，能夠在不使用電腦的情況下還能達到監測 SDN 環境中封包的資訊並下達命令。

關鍵詞：SDN、OpenFlow、APP、Android

Abstract

Software-defined networking (SDN) is a new networking architecture which decouples the control function and forwarding function so that the control and management functions are shifted from network devices to a centralized controller. Consequently, the SDN provides more innovation, flexibility and abstraction of the network resources and improves the network programmability with application programmable interfaces (APIs). In this paper, we design and implement an SDN network monitoring system for web Clients and Mobile Devices Applications (Apps) based on SDN network virtualization applications. The SDN network monitoring system provides network monitoring, modification, management and maintenance mechanisms over SDN networks.

Keywords：SDN、OpenFlow、APP、Android

1. 前言

隨著網路發展需求迫切，傳輸協定的增加，雲端服務以及巨量資料的傳輸日益增加，網際網路的路由表越來越複雜，讓目前的網路架構無法負荷如此大量的資料流動，進而產生了許多問題。傳統的網路架構是基於 STP 來保持封包在網際網路上傳輸的穩定性，為了實現各種網路的協定，交換機或者路由器需要不斷持續的拆解與重新組合封包，而導致傳輸的效率相當不佳，網路的頻寬也無法完全有效的發揮。網路的管理人員針對各種網路設定時，是需要客製化的。當管理人員在操作各種網路裝置時，必須針對每一台路由器與交換機進行設定，造成管理人員相當大的不便，也不容易快速的來更改網路架構以符合企業的需求。透過人工來逐一管理網路也有相當高的風險。

除此之外，不同的廠商雖然能使用共同的網路協定來傳輸，但是每個廠商所使用的網路管理技術不盡相同，造成各個網路的管理軟體彼此間難相容，一旦使用了同個企業所開發的設備，未來就需要遷就著相同企業所開發的新設備進行更新，無法與其他的廠商開發的設備共同使用，這是傳統網路存在的隱憂，同時為了可以持續承載龐大的網路流量，SDN 這項技術誕生了，SDN 的網路架構解決了這些問題。

本論文的目的是建置一個透過以 OpenFlow 為基礎的網路虛擬化網頁操作介面以及行動裝置應用程式的開發，讓交換機與控制器溝通，採用 OpenvSwitch 環境，提供 QoS 控制機制、封包分析、流量監控等功能，讓網路管理者能夠透過控制器有效地管理資料中心內部網路中的資料流。

2. 相關研究

2.1 SDN

SDN 是一種網路架構，其主要的特性是將原本交換機(Switch)裡面負責決定封包路徑的控制層(Control plane)與負責轉送封包的資料層(Data plane)獨立分開，並將控制功能

集中到 SDN 控制器(Controller)上管理，使其管理所有的網路設備與控制封包轉送的需求，而不需要在各個網路設備中安裝控制層軟體。

控制器裡面的控制軟體包含許多網路的管理以及服務，向下連接的是內含許多網路設備的基礎設備層(Infrastructure layer)，如 Ethernet、網路交換器、路由器等，向上連接的應用層(Application layer)提供客戶網路需求以及應用服務，透過這種方式更能即時監控網路設備的狀態及調度情況，網管人員只需在控制器下達指令就可以進行自動化的設定，使得維護資源成本降低，達成有效利用網路資源的目的，SDN 整體架構如圖 1。

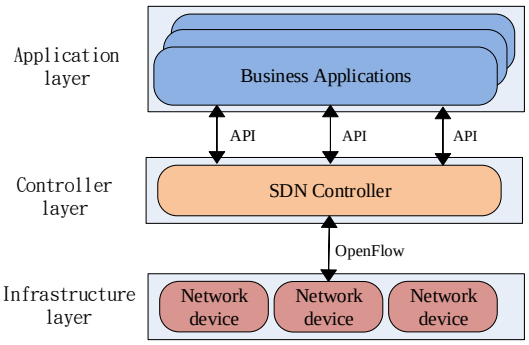

圖 1. SDN 架構圖

實作SDN的方法有很多，目前以OpenFlow實作的最為完整。

2.2 OpenFlow

OpenFlow是一個開放的通訊協定，使得控制器得以對交換機做有效的控制。最早在2008年由美國史丹佛大學（Stanford University）所推動，期望用以提昇網路效能及彈性應用的需求，並協助網路服務供應商(ISP)更精確有效的管理網路。OpenFlow實作SDN的方式如下，我們在交換機中寫入封包的flow，封包進入交換機後依照flow table所定義的flow來傳送封包，使封包能採取正確的動作。當處理flow table沒定義的封包時，交換機會將封包送往控制器，由控制器 計算封包接下來的流向並且建立flow entry，彈性地因應未來此類封包的處理。如此一來，可以由控制器定義不同的網路行為模式，達成使用者的需求。圖2為以OpenFlow 實作SDN的概念圖。

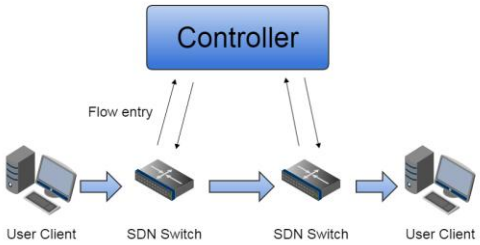

圖 2. OpenFlow 實作 SDN 概念圖

3. 系統設計

3.1 Web 網路監控系統

開發基於SDN架構的Web網路監控系統，如圖3所示，在系統架構上，我們使用mininet模擬一個SDN網路環境，會選用mininet的原因是因為下列兩點：1. 可以輕鬆建立各種topology，不用考慮硬體或虛擬機的實現，而且可以確定在各種topology上，我們的功能都達到實現。2. 有些指令可以方便我們做測試。所以我們先以mininet模擬一個SDN網路環境。選用的controller及資料庫如2.1及2.2所提，使用的controller為ryu，伺服器為apache。

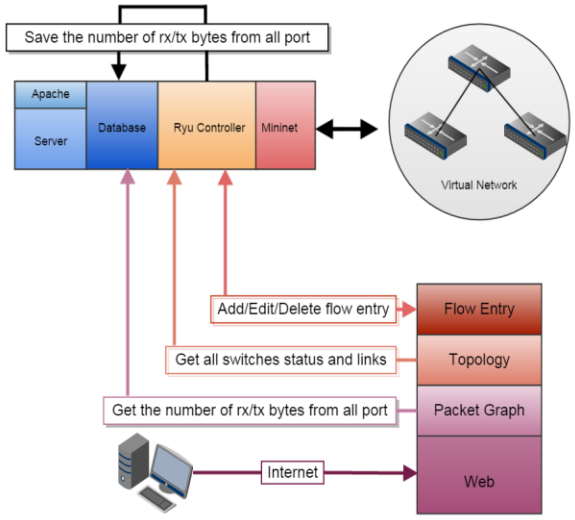

圖 3. Web 網路監控系統架構示意圖

此 Web 網路監控系統有三個功能，分別為 Packet Graph、Topology 及 Flow Entry，將以網頁端呈現。各功能詳細的介紹見表 1。

表 1.Web 監控系統功能介紹

Packet Graph	將流經各節點的流量的資訊，以Graph呈現在網站上，方便使用者清楚看到網路流量的趨勢與變化。
Topology	將switch的連線狀況，呈現在網站上，點擊switch會呈現該switch的個別資料。
Flow Entry	顯示每個switches的flow資料，使用者可利用add、edit、delete的功能來更改flow entry設定。

3.2 行動裝置 APP 系統

同時也開發基於 SDN 架構的行動管理 APP，相較於網頁版的管理工具，行動裝置應用程式方便攜帶也具有大眾廣泛使用度以及更好的便利性，能夠在不使用電腦的情況下監測 SDN 環境中封包的資訊並下達命令。如圖 4 為行動裝置 APP 系統架構圖。

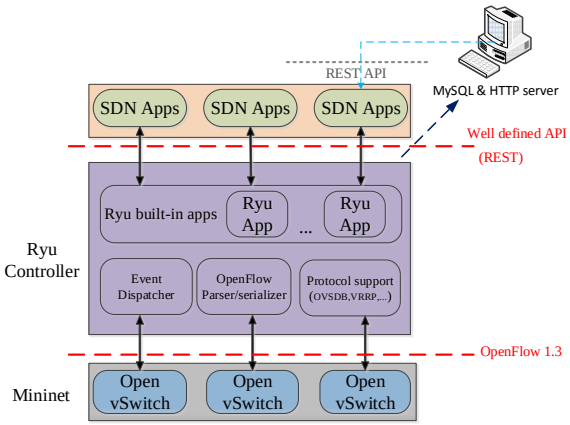

圖 4. 行動裝置 APP 系統架構圖

此行動裝置基於 SDN 架構所開發的應用程式使用的工具為 Kivy、Buildozer 和 Slim PHP，應用程式將包裝成 APK 格式檔案，以手機的 APP 客戶端呈現出來。各工具詳細的介紹見表 2。

表 2. 行動 APP 工具介紹

Kivy	用於開發 APP UI 的跨平台開放式 Python library。
Buildozer	一種將 Python 檔打包成 APP 工具，會自動下載所需套件，例如下載 SL4A、Android SDK、Android NDK 等。
Slim PHP	PHP 的一種微型架構，有效率的協助撰寫簡單但功能強大的網頁 APP 或是 Restful APIs。

4. 系統實作

4.1 Web 端開發

在一般的情況下，透過終端機對 Ryu Controller 進行控制與管理，使用者應用此方式必須對於 Ryu 的指令與架構相當熟悉，所以並沒辦法讓一般大眾使用 SDN 網路架構。

開發了使用者介面(User Interface,UI)可以用簡單且快速的方式進行管理與監控，此系統主要提供了三大功能：Flow Entry、Topology、與 Packet Graph，讓使用者可以更輕鬆的進行 SDN 網路監控與應用。以下分別介紹 Web 網路監控系統之功能如下：

4.1.1 Flow Entry

圖 5 為 Flow Entry 示意圖，此介面可以讓使用者選擇 switches 並顯示其 flow entry，接著可以利用 Web 網路監控系統進行新增、修改與刪除 flow entry 的動作，當使用者修改了 switch1 的 flow entry 如圖 6，透過 Packet Graph 我們可得知修改 flow entry 後的差異如圖 7。

圖 5. Flow Entry 示意圖

圖 6. 修改 Flow Entry 示意圖

圖 7. 修改後 Packet Graph 示意圖

4.1.2 Topology

圖 8 為三層樹網路拓樸，在 WEB 網路監控系統上，使用者可以拖曳 Topology 上所顯示的每個 Switch 位置，當使用者改變了拓樸後，WEB 系統上 Topology 也會隨之更新，當滑鼠移至 Switch 上並點選，則可看到 Switch 相關之資訊，圖 9 為 Switch 相關資訊示意圖。

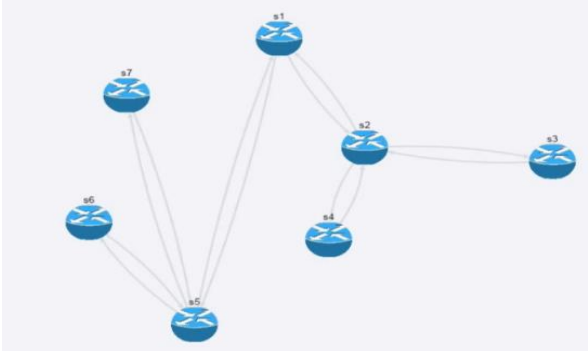

圖 8 Topology 示意圖

圖 9. Switch 相關資訊示意圖

4.1.3 Packet Graph

圖 10 為 Packet Graph 示意圖，使用者可透過此功能選擇觀察的 switch 與 port，並可選擇 Packet Graph 所顯示觀察封包流量時間長度，圖 11 顯示可透過滑鼠框一個範圍後，範圍內的圖示將放大使讓使用者可以更清楚觀察其節點之流量資訊。

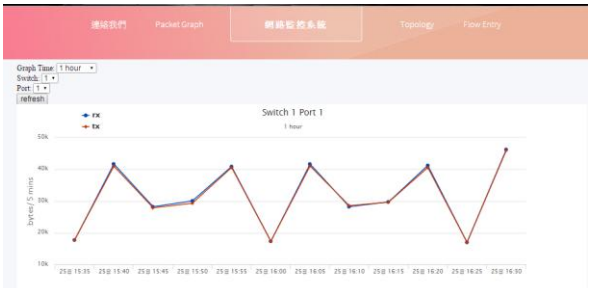

圖 10. Packet Graph 示意圖

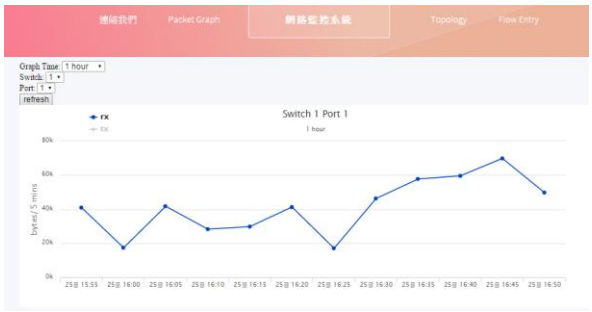

圖 11. rx 流量 Packet Graph 示意圖

4.2 結合 SDN 行動 APP 開發

4.2.1 Topology Viewer

此功能為顯示各個交換機(Switch)之間的連接狀態、Switch 的 DPID 及控制器的 IP，並且點選交換機後可查看其資訊(軟硬體版本)並執行其他功能。Switch 之間連接狀態是由 Ryu 內建的 rest_topology.py 所提供的 API，如 GET /v1.0/topology/switches /<dpid>、GET/v1.0 /topology/links 等來抓取連線到控制器的交換機以及其連接的狀態（LLDP)。如圖 12 所示深度為 2 的樹狀 Topology 的執行畫面。

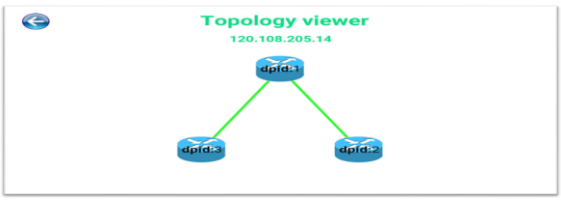

圖 12. Topology Viewer 執行畫面

4.2.2 Traffic Monitor

提供管理者監控各個交換機之間的封包流量資訊。使用者可以選擇交換機各個 port 來做觀察，並可選擇不同的時間範圍(hour、day、week、month)及傳送(Tx)或接收(Rx)的封包流量，如圖 13 所示，為 App 執行時實際操作畫面。

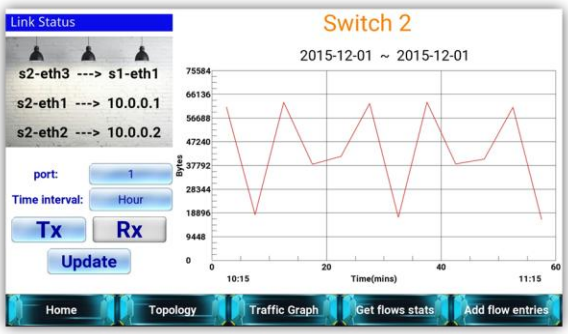

圖 13. Traffic Monitor 執行畫面

流量來源是使用 PHP 來向控制器 Ryu 中的 ofctl_rest.py 程式抓取資訊可以存放於資料庫中，並藉由 Linux 的排程作業 Crontab 來定時執行 PHP。流量存取因考量到安全性需求，由 Slim PHP 架構寫成 Rest API 來做為 APP 端與資料庫（Data Base）之間的溝通介面，而不是由 APP 直接向資料庫存取流量的資訊。

4.2.3 Flow Entry

提供使用者觀/查看、編輯、刪除 Flow Table 的內容，或是新增一個 Flow Table 以對交換機做適當的管理，執行畫面如圖 14 所示，為查詢 Flow Table。

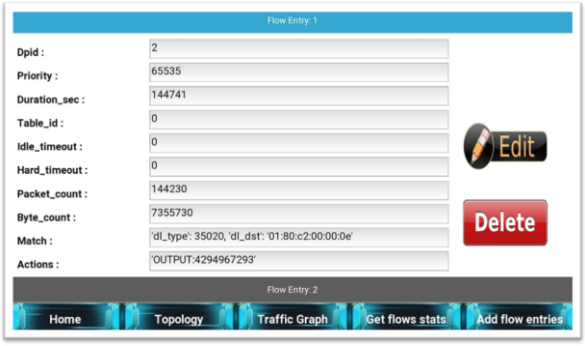

圖 14. Flow Table 的查詢畫面

相對的方便於管理基於 SDN 網路架構的 App，使用者也可以利用此行動程式來新增一個 Flow Table 如圖 15 所示，需以相同的格式撰寫並依表格來填入適當的位置中，成功編寫完成且送出後會出現成功的符號如圖 16 所示，來告訴使用者已經成功新增 Flow Table。

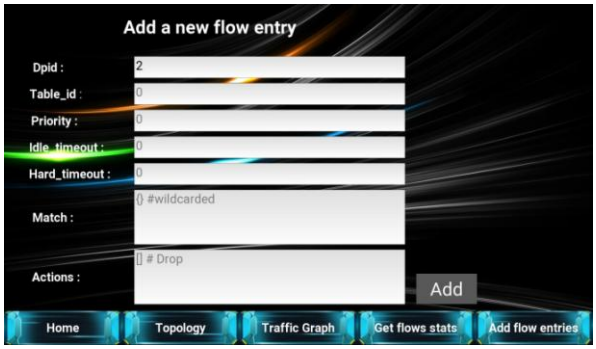

圖 15. 新增 Flow Table 的執行畫面

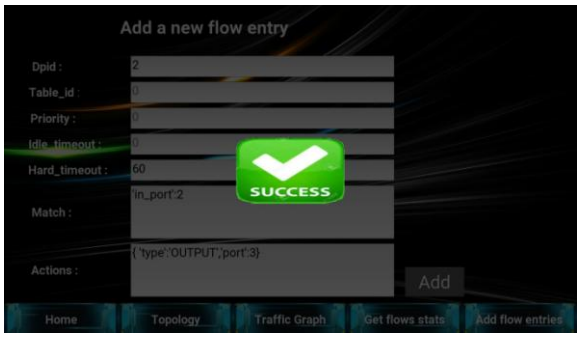

圖 16. Flow Table 新增成功畫面

5. 結論

傳統網路有無法完整利用網路頻寬以及不易快速變動架構來因應企業建置新系統的需求等侷限，所以選擇研究SDN作為我們的改善方法。已完成了具以下特色的具SDN之網路監控系統為圖形化介面連線方便、操作容易、即時動態呈現網路拓樸與流量分析圖表、 可針對網路架構變動快速反應，並可以自行定義 flow entry，以利於我們在不同的情況中設置不同的權限來提升整體的傳輸效能。

現今網路架構逐漸以軟體定義網路來取代傳統分散式的網路架構並可藉由此架構來提供網路更多的靈活性。以 SDN 作為主體架構，實作開發出網站與行動裝置網路監控 APP，為了增加便利性並結合行動裝置讓一般民眾皆可使用。

6. 致謝

本研究感謝專題生王俞文、呂昆霖、陳奕廷、江恆安對系統雛形開發的協助。同時本研究之執行承蒙科技部計畫(MOST 104-2221-E-142-001)、國立臺中教育大學以及教育部特殊優秀人才獎的支持，特此致謝。

參考文獻

[1]Hakiri, Akram, et al. "Software-defined networking: Challenges and research opportunities for future internet." *Computer Networks* 75 (2014): 453-471.

[2]Braun, Wolfgang, and Michael Menth. "Software-Defined Networking using OpenFlow: Protocols, applications and architectural design choices." *Future Internet* 6.2 (2014): 302-336.

[3]Han, Zhi-jie, and Wanli Ren. "A novel

wireless sensor networks structure based on the sdn." *International Journal of Distributed Sensor Networks* 2014 (2014).

[4] “Welcome to RYU the Network Operating System(NOS)”, http://ryu.readthedocs.org/latest/index.html/.

[5] “Mininet - An Instant Virtual Network on PC”, http://mininet.org/.

[6] Bao, Jinzhen, et al. "Towards Software Defined SpaceWire Networks."*SpaceWire Conference (SpaceWire), 2014 International*. IEEE, 2014.

[7] Chang, Hua-Ting, and Shie-Yuan Wang. "Using SDN technology to mitigate congestion in the OpenStack data center network." *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015.