

# Software-Defined Networking 技術應用於教室網路管理

## Software-Defined Networking Applications for Computerized Classroom Network Management

陳日揚 陳育聖 曾立宇 林易泉

國立虎尾科技大學資訊工程系

JIH-YANG CHEN & YU-SHENG CHEN & LI-YU ZENG & Yih-Chuan Lin

Department of Information Engineering, National Formosa University

lyc@nfu.edu.tw

### 摘 要

因應電腦教室不同上課情境之需求，往往需要由人工進行設定或是將實體網路線拔除，導致花費大量的人力及時間成本。本論文採用新型態的軟體定義網路 SDN(Software-Defined Networking)架構建置「教室網路管理系統」，提供有效的解決方案。利用常見的無線分享器 (Wireless 路由器)搭配 OpenWrt 韌體、Open vSwitch 套件、及 Floodlight 控制器(Controller)來實現系統。透過簡易的圖形化界面，老師便能輕鬆地切換自由模式、上課模式及考試模式。本文實驗結果證明，將軟體定義網路的技術應用於電腦教室，確實能大幅提升網路管理的效率。

**關鍵詞：**網路管理；OpenFlow；SDN；Open vSwitch；Floodlight

### 1. 前言

大專院校中，許多課程都是在電腦教室中進行教學，當老師讓學生自由練習時，學生可能會利用電腦瀏覽與課程非相關之網頁，而無法專心聽講。如果將網路切斷，又可能會造成其他想利用電腦查詢與課程相關資料同學的不便，在這種情況下，要針對不同模式需求的網路模式做更動、集中化管理，於傳統網路架構來看顯得非常的複雜、不夠靈活且難以進行擴充。

為了改善諸如此類的問題，利用近年來網路的新形態趨勢—軟體定義網路SDN架構來開發網路管理平台，使授課老師能夠更方便、快速的去管理學生使用電腦的規範，讓教學品質得以提升，學習成效也將更加地顯著。

表1為利用傳統網路架構與SDN網路架構實現教室網路管理方法之比較。

表 1 實現教室網路管理方法比較

	傳統網路架構	SDN 網路架構
變換上課情境方式	透過手動插拔網路線或網管設定存取控制清單	操作簡易的網頁介面便可實現
效率	低	高
設備故障率	高，需經常插拔網路線	低

### 2. SDN與OpenFlow介紹

#### 2.1. SDN軟體定義網路的概念

現今網路架構中，各種網路設備是遵循多種不同的傳輸協定來傳送封包。隨著網路的需求日益漸增，網際網路中的路由協定也越來越多並且越來越複雜。

當網路設備(如：交換器、路由器)接收到封包後，必須根據繁雜的傳輸協定規則一一比對，待比對完成後，更要執行封包轉送的程序。如此一來，網路設備的工作繁複，直接地影響到封包傳送的品質，最終導致整體網路傳輸效能不佳。

又舉例，當路由器需要增加路由規則時，網路管理人員必須手動透過命令執行介面(Command-Line Interface, CLI)下達指令才能達成，而各個廠牌的設備所要下達的命令格式又不盡相同，造成龐大的人力負擔。

如圖 1 所示，傳統網路設備需要同時負責路由規則的判斷及封包傳送，相當繁雜。SDN 的主要精神是將網路架構分為控制層 (Control Plane) 與資料層 (Data Plane)。

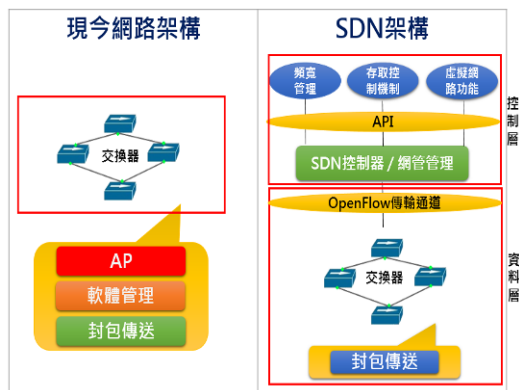


圖 1 現今網路架構與 SDN 架構的差異[1]

控制層中的控制器（以下稱為Controller）具有網路管理的功能，而實現的方式則是透過軟體。若欲下達路由規則，統一由Controller 對資料層的網路設備進行下達，方便集中控管。資料層設備則是單純地負責封包的轉發，而其亦遵循控制層所下達的轉發規則進行傳送。

第三方軟體可透過 Controller 所提供的 API(Application Programming Interface) 對 Controller 下達指令，不須要了解其實做的方法及複雜的指令，方便軟體開發者進行使用者界面的開發。而Controller 在接收到命令後會透過OpenFlow 協定將路由規則下達給資料層的設備，資料層的交換器必須不斷地學習來尋找封包傳送的路徑，如此一來便能大幅地縮減封包傳送的延遲。

## 2.2. OpenFlow[2]

OpenFlow為SDN架構中控制層與資料層之間的通訊協定。OpenFlow 目前是由 ONF(Open Networking Foundation)[3, 5]負責其定義與發展。該協定屬於 OSI 模型(Open System Interconnection Reference Model)中的第四層，主要目的為控制封包的轉送，可匹配(match)封包中網路層相關協定的細節。

在OpenFlow 協定中，傳送規則存放於Flow Table，其中一筆傳送規則稱為一個Flow Entry。透過Flow Entry 的比對，便能決定出封包的傳輸路徑。

資料層中，能夠運行OpenFlow 協定的交換器稱為OpenFlow 交換器。

### (1) Flow Table :

所有Flow Entry 皆存放於Flow Table 內，OpenFlow 1.0 以上版本可利用Multi-Table 搭配Flow Entry 達成類似管線功能，使封包轉送更為彈性。

### (2) Flow Entry :

如圖 2 所示，Flow Entry 包含 Header Fields、Actions 及 Counters 等三部分。

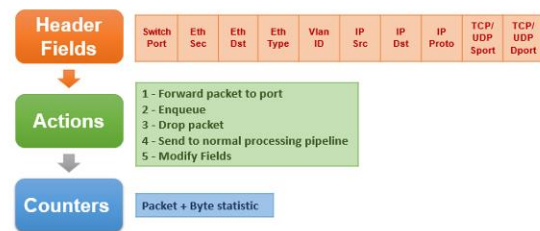


圖 2 Flow Entry[4]

Header Fields用於設定該規則的各種條件，如封包所使用的通訊協定、埠口、來源IP、目的IP等。在封包轉送時，OpenFlow交換器便遵循這些規則進行轉發。

Actions則是掌握整體網路運作的行為，其定義了封包符合規則後封包要如何轉送，而一個Flow Entry可以設定一個或多個Actions。

下述將說明Action的詳細內容：

- **Forward**：將封包由交換器 Port 或 OpenFlow 定義的 Visual Port 送出，如 ALL（將封包傳送到除 in Port 之外的所有 Port）、CONTROLLER（將封包以 OpenFlow 協定傳給 Controller）等。
- **Enqueue**：把封包加入到某個 Port 的 Queue 中，再根據 Queue 的排程來傳送封包。
- **Drop**：丟棄封包。該動作為預設動作。
- **Modify-Field**：修改封包的內容後再傳送出去。Counters 用來統計符合規則的封包數量及大小，可用於觀察網路運作的行為。[4]

## 2.3. OpenFlow 協定的運作：

OpenFlow交換器接收到封包時，會將封包中各欄位的值與Flow Table 中各個Flow Entry 的Header Fields進行比對，若比對成功的話，就會執行該Flow Entry中設定的Actions，並將Counters值累加。當找不到任何相符的Flow Entry 時，OpenFlow交換器會將封包傳送給Controller，讓Controller來決定該封包要如何處理。

## 3. 系統實作方法

### 3.1. Controller：Floodlight[6, 7]

Controller位於控制層，在SDN的網路架構中可將Controller視為一個控管整個網路結構的大腦，它匯集了各個網路節點的裝置資訊（如：OpenFlow交換器資訊、Host資訊等）。

Controller 對上層則提供各種北向(Northbound)API給使用者。透過Controller所提供的北向API，使用者便能獲取各項節點資訊或是對網路架構中的設備進行控制。對下層Controller則是利用南向(Southbound)API對下層交換器中的Flow Entry進行管理以及組態的設定，

一般常見的Controller有如：Floodlight、OpenDaylight等，本文的實作採用前者。Floodlight Controller提供的API如下：

### (1) 北向API (Northbound API)：

針對使用者或是程式提供與交換器或Controller內部設定的溝通管道，其中包含了Controller Web UI(Web User Interface)、REST(Representational State Transfer) API 等。

圖3即為Floodlight所提供的Web UI，圖中顯示了和該Controller所連接的OpenFlow交換器及其底下的Hosts資訊。

Switches (2)						
DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
00:00:00:00:00:00:01	160.244.133.17:45344	Stanford University	0	0	0	2016/3/23 下午2:24:47
00:00:00:00:00:00:02	160.244.133.17:32975	Stanford University	0	0	0	2016/3/23 下午2:40:00

Hosts (1)			
MAC Address	IP Address	Switch Port	Last Seen
e8:5a:9f:b3:9c:5e	152.168.2.140	00:00:00:00:00:00:02:1	2016/3/23 下午2:40:02

圖3 Controller Web UI 示意圖

### (2) 南向API (Southbound API)：

Controller透過南向API與資料層的設備進行溝通，OpenFlow在規格上雖被視為一項通訊協定，但OpenFlow亦為一南向API的實現。Controller可透過南向API控管各個交換器及其底下的Host。本文所採用之Floodlight Controller則提供許多功能，包含設備管理(Device Manager)、模組管理(Modules Manager)、防火牆(Firewall)、負載平衡(Load Balance)等，這些模組化的功能(Controller Modules)也使得系統的使用上更加地便利。

## 3.2. Floodlight REST API

Floodlight Controller提供北向的REST API，第三方程式或使用者透過REST API便能與Controller模組進行溝通。Floodlight支援：存取控制清單(ACL)、防火牆(Firewall)、靜態流插入(Static Flow Pusher)、虛擬網路過濾(Virtual Network Filter)這四種REST API。

本文系統功能的實現便是透過下達HTTP REST指令來完成。REST API亦能查詢網路情境目前的各種狀態資訊。

表2至表6為Floodlight的常用REST API功

能及說明：

表2 常用指令[8]

URI	描述	參數
/wm/staticflowpusher/clear/<switch>/json	清除Flow	Switch:Valid Switch DPID or "all"
/wm/staticflowpusher/json	新增／刪除Flow	HTTP POST data (add flow)，HTTP DELETE(for deletion)
/wm/staticflowpusher/list/<switch>/json	列出Flow	switch:Valid Switch DPID or "all"

表3 每條Flow Entry所需參數[8]

參數	值	描述
name	<string>	Flow的名稱，而且名稱必須是獨特沒有重複的
switch	< DPID>	DPID是用來辨識交換器的值，Flow可以透過這個值來選擇要加到哪一台交換器

表4 每條Flow Entry可選擇增加參數[8]

參數	值	描述
priority	<number>	默認值：32767.
Active	<boolean>	"true" or "false"
Table	<number>	用默認的 flow table

表5 每條Flow Entry可選擇新增比對項目[8]

參數	值	of 版本	描述
in_port	<number>	All	指定封包從某個交換器 Port 進入
eth_type	<number>	All	可以是十六進制或十進制
eth_src	<MAC address>	All	xx:xx:xx:xx:xx:xx
eth_dst	<MAC address>	All	xx:xx:xx:xx:xx:xx

表 6 Action 欄位可以新增的 Action[8]

參數	值	of 版本	描述
output	<number> all controller local ingress-port normal flood	All	如果沒有設定輸出的值，則為”drop”。
table	<number>	OpenFlow 1.1+	可以是十六進制或十進制
group	<number>	OpenFlow 1.1+	可以是十六進制或十進制

圖 4 為透過 REST API 進行新增一條 Flow、列出某台交換器的 Flow、刪除一條 Flow 的實作指令。

#### 新增

```
curl -X POST -d '{"switch": "00:00:00:00:00:01", "name": "flow-mod-1", "cookie": "0", "priority": "32768", "in_port": "1", "active": "true", "actions": {"output": "2"}}' http://<controller_ip>:8080/wm/staticflowpusher/json
```

#### 列出

```
curl http://<controller_ip>:8080/wm/staticflowpusher/list/00:00:00:00:00:01/json
curl http://<controller_ip>:8080/wm/staticflowpusher/list/all/json
```

#### 刪除

```
curl -X DELETE -d '{"name": "flow-mod-1"}' http://<controller_ip>:8080/wm/staticflowpusher/json
```

圖 4 REST API 新增、列出、刪除範例[8]

### 3.3. OpenWrt[9]

OpenWrt 是用於嵌入式系統的 Linux 發行版，屬第三方韌體系統。OpenWrt 可自行添加使用者開發的軟體功能包，客製化屬於自己的功能，自由度高。

本文的實作利用 OpenWrt 高自由度之優點，選擇於無線分享器上安裝該作業系統。OpenWrt 有提供多種擴充性高之軟體套件，本系統藉由安裝 OpenvSwitch 套件來實現實體機器運行 OpenFlow 之功能。

### 3.4. Open vSwitch[10]

Open vSwitch 為一具有產品級質量的多層虛擬交換機，其遵循 Apache2.0 協議，由 Nicira Networks[11] 主導其開源項目。Open vSwitch 充分考慮了在不同虛擬化平台間的移植性，以 C 語言開發。同時透過可編程擴展，以實現網路自動化。

Open vSwitch 支持現有多種標準的管理介面及網路協定（如：netFlow，sFlow，SPAN，RSPAN，CLI，LACP，802.1ag 等），管理員可以透過 Open vSwitch 的虛擬化網路特性更靈活且高效率地實現管理功能。

## 4. 系統實作成果

### 4.1. 系統架構

本文之系統可分成三個層面，如圖 5 所示：

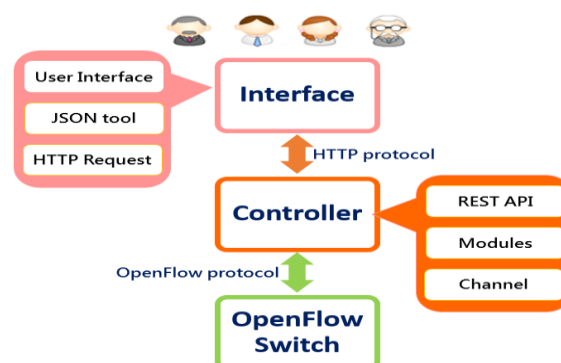


圖 5 系統架構示意圖

最上層為使用者介面（User Interface），由 JSP(JavaServer Pages) 進行撰寫。此介面提供使用者目前教室網路環境的即時資訊，並且使用者能夠透過此介面切換各種上課的模式。使用者介面的實作則是利用北向的 REST API 與 Controller 定時進行溝通。

Controller 接收到 REST API 指令後將規則轉譯為 OpenFlow 協定格式的封包，接著再透過南向 API 傳送到 OpenFlow 交換器。

在最底層的 OpenFlow 交換器則僅負責封包的轉發及 OpenFlow 封包的解析。同時，它也必須依循著 Controller 所下達的傳輸規則運行。

### 4.2. 系統功能

本系統主要針對的使用者即為授課的老師。系統功能如圖 6：



圖 6 系統功能架構示意圖

使用者可以將網路設定成自由、上課、考試這三種模式，也可以設定網路組態，而這些功能都必須要取得 Switch 和 Host 的資料，其中上課及考試模式可依使用者需求來自行新增黑白名單，圖 7 為系統的使用案例圖。



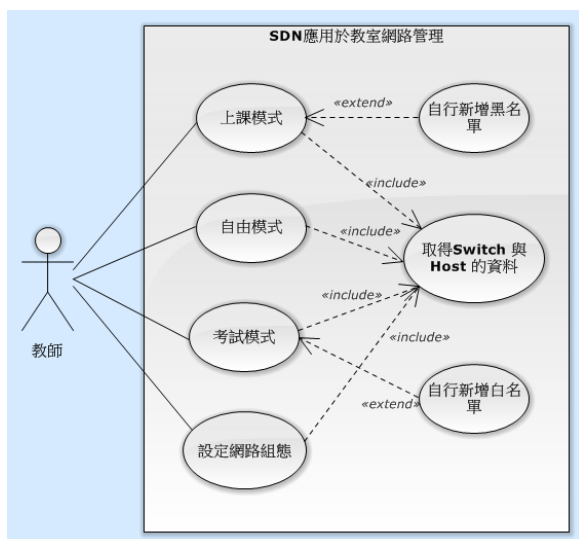


圖 7 系統使用案例圖

使用者可以根據使用者介面進行模式的切換、網站清單的管理...等功能，如圖 8 所示。



圖 8 使用者介面示意圖

老師可依照課程情況選擇以下三種模式：

### (1) 自由模式：

下課時，學生自由使用網路，沒有任何限制。

### (2) 上課模式(黑名單)：

上課時，學生不能瀏覽與教-學無關之網站，故對上課黑名單中的網站進行限制。

在選擇欲封鎖的黑名單網站後，Controller 的 Web UI 便會顯示已經加入的 Flow Entry，如圖 9 所示。

Flows (2)

Cookie	Table	Priority	Match	Apply Actions	Write Actions	Clear Actions	Goto Group	Goto Meter	Write Metadata	Experimenter	Packets	Bytes	Age [s]
453595945453772	0x0	32768	eth_type=0x0800 ip_dst=140.130.0.0/255.255.192.0	actions: output=normal	---	---	---	---	---	---	130	8978	252
0	0x0	0	---	actions: output=controller	---	---	---	---	---	---	1545	143459	4114

圖 9 黑名單 Flow Entry 示意圖

### (3) 考試模式(白名單)：

考試時，規定只能連上特定網站(如：學校的教學平台)，且不允許學生互相傳送資料。

在選擇欲通過的白名單網站後，Controller 的 Web UI 亦會顯示已經加入的 Flow Entry，如圖 10 所示。

Flows (6)

Cookie	Table	Priority	Match	Apply Actions	Write Actions	Clear Actions	Goto Group	Goto Meter	Write Metadata	Experimenter	Packets	Bytes
453595945453773	0x0	32768	eth_type=0x0800 ip_dst=140.130.0.0/255.255.192.0	actions: output=normal	---	---	---	---	---	---	2270	10612
453595951128090	0x0	32768	eth_type=0x0800 ip_dst=140.130.0.0/255.255.192.0	actions: output=normal	---	---	---	---	---	---	21	3336
453595951128090	0x0	32768	eth_type=0x0800 ip_dst=140.130.0.0/255.255.192.0	actions: output=normal	---	---	---	---	---	---	21	1669
453595945453774	0x0	32768	eth_type=0x0800 ip_dst=140.130.0.0/255.255.192.0	actions: output=normal	---	---	---	---	---	---	4180	58919
453595945453772	0x0	100	eth_type=0x0800	---	---	---	---	---	---	---	324	29010
0	0x0	0	---	actions: output=controller	---	---	---	---	---	---	570	53607

圖 10 白名單 Flow Entry 示意圖

## 4.3. 教室網路環境模擬

圖 11 為教室網路環境的模擬拓模圖。一般教室的網路來源是從機房的路由器拉線到教室中的一般交換器，教室中的電腦在再連接到一般交換器。在不改變教室外部來源網路的配置情況下，僅需將一般交換器置換成支援 OpenFlow 之交換器即可。在設定好 OpenFlow 交換器各種組態後，使之能與 Controller 進行連線，便能夠達成本系統之各項功能。

其中，Controller 的架設環境可以是在虛擬機或是實體機皆可，本文是將 Controller 安裝於虛擬機上。而 Controller 的架設位置可以在與電腦教室同一個區域網路之中，亦可在外部的廣域網路，只要確保 OpenFlow 交換器與 Controller 之間的路由設定完整，雙方能互相通訊即可。

Controller 與 Web Server 運行於同一部電腦上，教師電腦僅需開啟瀏覽器並且進入系統的使用者介面便能直接對網路進行控管，無須再另行安裝相關的軟體。

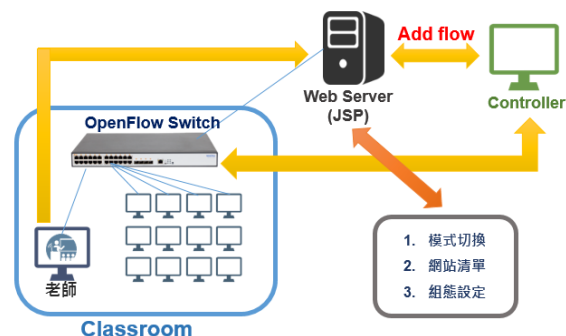


圖 11 教室網路拓模示意圖

## 5. 結論

軟體定義網路可說是未來網際網路的新形態趨勢，Google、Facebook、Yahoo、微軟等多家指標型的大企業皆紛紛投入了 SDN 架構與 OpenFlow

技術的發展。本文藉由校園的環境，簡單地實作出一套教室網路管理系統。在傳統的教室網路環境中，授課教師若想要允許或限制學生瀏覽某些特定的網頁，老師必須向網管人員提出申請需求，再由網管人員進行設定。該流程往往需耗費冗長的時間及人力。透過軟體定義網路的教室網路管理系統，授課教師僅需在電腦上的 UI 介面中選取或自行輸入控制存取清單，便能輕鬆地達到上述之效果。

### 5.1. SDN 架構的好處及應用成果

在中央控管的網路模式下，整體網路環境有很高的即時性及靈活性。對於網路系統的開發者，透過 Controller 北向 API 能夠方便地開發出各種不同情境的應用系統。對於網管人員及使用者，在不了解太多網路設備的指令情況下，能夠用最簡單且效率的方法達成網路管理的需求。

### 5.2. SDN 架構的限制

SDN 架構及本文所實作的教室網路管理系統目前最大的限制在於設備的支援，與傳統的網路交換器相比，支援 OpenFlow 的交換器價格昂貴許多。若校園中欲引入此系統，全校的網路設備必須全數更新為支持 OpenFlow 協定的設備才可達成。對於校方來說，會是一筆不小的負擔。

### 5.3. 未來研究

在 Controller 所提供的 Flow Entry Action 本文僅利用到封包的丟棄而已，在未來計畫能夠加入流量控制的功能，使系統能夠根據不同的單位、實驗室、教室配置出最適當的網路頻寬，達到網路資源的有效利用。再者，我們也希望能夠與學校的課務系統連線，達成匯入課表的功能，系統將能夠根據當時不同的課程限制該時段配置的網路流量。

## 參考文獻

- [1] 張景皓 (2012 年 11 月 19 日) ”新一代網路架構 SDN 顛覆傳統網路的控制模式”。(2016 年 6 月 14 日) <http://www.ithome.com.tw/node/77353>
- [2] OpenFlow 官網 (無日期)。 (2016 年 6 月 14 日)，<http://archive.openflow.org/videos/>
- [3] Open Networking Foundation (無日期)。 (2016 年 6 月 14 日) <https://www.opennetworking.org/about/onf-overview>
- [4] 何哲勳/楊慧卿/許名宏 (2014 年 3 月) ”新電子，OpenFlow 通訊協定穿針引線 SDN 打造高效能/彈性網路”。 (2016 年 6 月 14 日) [http://www.mem.com.tw/article\\_content.asp?sn=1402270012](http://www.mem.com.tw/article_content.asp?sn=1402270012)
- [5] ONF(Open Networking Foundation) (2013 年 10 月 14 日) ”OpenFlow Switch Specification”。 (2016 年 6 月 14 日) <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>
- [6] Floodlight (無日期)。 (2016 年 6 月 14 日) <http://www.projectfloodlight.org/floodlight/>
- [7] Michael Bredel ”OpenFlow Controller”(無日期)， (2016 年 6 月 14 日) <http://www.admin-magazine.com/Articles/Floodlight-Welcome-to-the-World-of-Software-Defined-Networking>
- [8] Ryan Izard ”Floodlight Static Flow Pusher API (New)” (2016 年 5 月 15 日)， (2016 年 6 月 14 日) <https://floodlight.atlassian.net/wiki/pages/viewpage.action?pageId=1343518>
- [9] Openwrt 官網 (無日期)。 (2016 年 6 月 14 日) <https://openwrt.org>
- [10] Open vSwitch (無日期)。 (2016 年 6 月 14 日) <http://openvswitch.org/>
- [11] Nicira Networks Wikipedia, the free encyclopedia (2015 年 7 月 9 日)， (2016 年 6 月 14 日) <https://en.wikipedia.org/wiki/Nicira>