

# SDN效能測試系統之架構設計

## A System Architecture for SDN Performance Testing

王姿琳 方湘婷 林育勳  
Tzu-Lin Wang, Claudia Fang, Chester Lin

### 中文摘要

雲端應用的興起使得軟體定義網路(Software-Defined Networking ; SDN)獲得注目，而SDN控制器(SDN Controller)藉由集中控制網路設備的方式，讓網管人員能快速且無誤的下達指令或規則(Rule)至指定的網路設備。為了參與SDN技術的發展，各家廠商或組織紛紛提出SDN解決方案，使用者安裝所需的網路專案至SDN控制器中，藉此改善網路效能。為了減少開發成本，許多使用者會利用模擬器整合SDN控制器進行網路專案開發。然而目前僅有SDN模擬器提供建置SDN環境，卻沒有一套評估套用特定網路專案後的效能評估系統，導致使用者無法真正了解各套網路專案之間的效能差異。為此，本論文首先提出一SDN效能評估系統之架構設計，當使用者上傳簡單的自定義網路拓樸與流量設定檔後，本系統能自動產生欲測試專案所需的網路拓樸，以供該網路專案使用，並能產生網路流量傳送至該網路中，藉此評估該網路專案的傳輸效能。於架構設計中，本系統根據蒐集的網路封包進行效能評估並提供評估報告給使用者。本論文未來擬完成整體SDN效能測試系統之實作與增加更多效能評估項目，以利使用者了解各網路專案之特性。

### Abstract

Software-defined networking (SDN) has obtained much more attention due to the rapid development of cloud applications. Users can quickly add rules to the designated network devices with high reliability by using centralized SDN controller. To participate in the SDN marketing, many companies and foundations propose their SDN solutions, and users can install the solution to controller for improving the network performance. To reduce the development cost of SDN solution, users often integrate SDN simulator and controller to simulate and develop their own SDN solutions. However, the existing well-known network simulators only support the construction of SDN environment, lacking the performance evaluation of SDN solution which the user is concerned. Therefore, users cannot understand the differences between their SDN solutions. To overcome the problem, this paper firstly proposes an architecture of the SDN performance testing system. After uploading simple configurations of the network topology and traffic pattern, it automatically sets up the network topology and generates network traffic flow for network performance analysis of that SDN solution. In the proposed architecture, the traffic information will be collected and analyzed for providing evaluation reports to users. In future works, we will implement the whole SDN performance testing system and add more evaluation factors so that users would know more about the characteristic of each SDN solution.

### 關鍵詞(Key Words)

軟體定義網路(Software-defined Networking ; SDN)  
效能測量(Performance Measurement)

## 1 · 前言

軟體定義網路 (Software-defined Networking ; SDN)[1] 讓 SDN 控制器 (SDN Controller)能集中控制各式網路設備。網管人員透過控制器便能下達指令或規則(Rule)至指定的網路設備，除了能滿足客戶需求外，亦能大幅降低人工配置成本與錯誤率。隨著SDN的蓬勃發展，各式各樣的組織或廠商紛紛提出自家的 SDN 解決方案，如 Ryu [2]、NOX[3]/POX[4]、OpenDaylight[5] 或 OpenContrail[6] 等 SDN 控制器。以 OpenDaylight Beryllium[7]為例，其包含各式網路相關專案，例如認證與授權相關專案、控制器專案、專案整合或測試專案、網路設定相關專案、網路虛擬化相關專案、專案開發貢獻度與相關開發工具等專案。

而對於SDN技術而言，如何使網路傳輸更有效率則顯得非常重要。為此，眾多網路效能改善專案紛紛被提出。但在眾多網路專案當中，該如何確切得知哪種網路專案真正適合自身開發的系統，亦是重要議題。目前雖有網路仿真器[8]或模擬器[9]能幫助開發者模擬SDN網路，然而上述模擬器與仿真器並沒有提供網路專案效能測試方面的評估。

為了提供一個能進行網路專案效能評估的系統，本論文首先提出一個對SDN效能測試系統之架構設計，整合了Mininet[9]與自行開發的效能測試相關模組。當使用者輸入效能測試所需的參數後，本系統能自動產生各式網路拓樸，並藉由產生的拓樸測試各網路專案對於封包傳輸效能的影響。此外，本系統也提出效能評估項目，可針對負載平衡、傳輸量、傳輸延遲時間與線路使用率(link utilization)等項目進行測量。未來除了完成整個SDN效能測試系統之實作外，亦將加入更多的評估項目，以利使用者了解各網路專案的特性及提供準確的使用參考。

本文章架構如下，第2章介紹相關背景知識；第3章介紹系統運作流程與效能評估矩陣；第

4章顯示使用本系統之部份實作畫面；第5章總結本文所提出的SDN效能測試系統架構設計與其未來發展方向。

## 2 · 相關背景知識

本章節分別介紹Mininet與EstiNet等知名網路仿真器與模擬器。Mininet為基於OpenFlow協定[10]的網路仿真器，其採用輕量虛擬化技術模擬整個網路。使用者可透過Mininet模擬虛擬主機、交換機、線路及SDN控制器等網路設備。除了可大幅減少硬體成本外，亦避免使用者在真實網路環境測試時，可能導致的網路癱瘓。Mininet利用使用者命令列介面(Command-line Interface ; CLI)進行指令操作，但使用者需較複雜的網路運作時，Mininet的操作則略顯繁雜且不易使用。

交通大學團隊所開發之EstiNet可模擬網路設備之間的通訊協定，使用者透過圖型化介面可了解各個通訊協定之間的封包傳輸情況，其目前支援IEEE802.3、IEEE802.11、VANET(IEEE802.11/1609)與OpenFlow協定。於SDN方面，EstiNet可模擬OpenFlow控制器並自動指定IP至OpenFlow交換機，以模擬SDN環境。

雖然Mininet與EstiNet能模擬SDN環境，但當使用者進行模擬實驗時，需逐一建立網路架構與各式元件，容易花費大量時間且操作不便。為了減少使用者負擔，本論文提出一基於Mininet的SDN效能測試系統架構，除了能提供數種較常見的網路型態，如Three-tier、Fat-tree及Leaf-spine之外，亦加強各網路型態的架構彈性。使用者透過較少的操作，便能建立欲進行模擬的網路架構，且基於該架構進行不同網路專案的效能評估，最後本系統將產出一份效能評估報告給使用者，以利使用者了解各網路專案的特性。

## 3 · 系統運作流程與效能評估矩陣

除了介紹本系統擬開發之相關模組外，本章亦詳盡介紹其架構設計、運作流程與效能評

估矩陣。

### 3.1 系統運作流程

本節首先介紹系統架構設計的各项模組，如圖1方框內所示：

- **Topology Configuration and Traffic Model**：使用者透過Topology Configuration能設定建立網路拓樸所需的參數，例如交換機數量、網路型態與虛擬機器數量等參數，以產生網路拓樸；而Traffic Model則允許使用者自訂網路流量相關參數。
- **Topology Generator and Traffic Generator**：接收使用者上傳的網路拓樸與流量設定檔後，Topology Generator與Traffic Generator分別根據設定檔產生效能測試所需的網路拓樸與流量。
- **Format Translator**：由於待測試的網路專案其接收的網路拓樸格式或參數不盡相同，因此當本模組接收來自Topology Generator產生的網路拓樸後，可自動轉換上述拓樸為欲測試網路專案可接受的拓樸格式，以利後續專案效能評估。
- **Testing Project and Rule Deployment**：Testing Project為欲測試的網路專案，當其接收來自Format Translator轉換的網路拓樸後，便可利用該拓樸進行封包路徑規劃或其他網路配置。封包路徑或其他網路配置則透過該網路專案進行Rule Deployment，意即將規則設置到相關網路設備中。
- **Traffic Info. Collector**：當規則佈建至指定網路設備後，Traffic Info. Collector便連線至各台網路設備擷取封包，爾後將封包統計結果傳送至效能評估模組進行評估。
- **Performance Evaluation Matrix**：本系統所開發之效能評估矩陣可針對負載平衡、傳輸延遲時間與線路使用率等項目進行評估。當此模組接收來自Traffic Info. Collector的資料後便開始進行效能評估，最後將評估結果傳送給使用者。

接著介紹系統運作流程，如圖1所示，1)使用者分別上傳拓樸設定參數與流量參數至Topology Generator與Traffic Generator後，2)Topology Generator便將產生的網路拓樸傳送至Format Translator。3)轉換後的網路拓樸接著傳送至欲測試的網路專案中，該專案則根據網路拓樸進行封包傳輸路徑規劃。4-5)而規劃好的傳輸路徑可透過Rule Deployment設定規則至相關網路設備。6)Traffic Generator依照使用者設定的參數產生網路流量，並將流量導入Testing Project。7)當網路流量產生後，Traffic Info. Collector可連線至各網路設備蒐集並統計網路封包，8)並將結果送至Performance Evaluation Matrix進行效能評估。

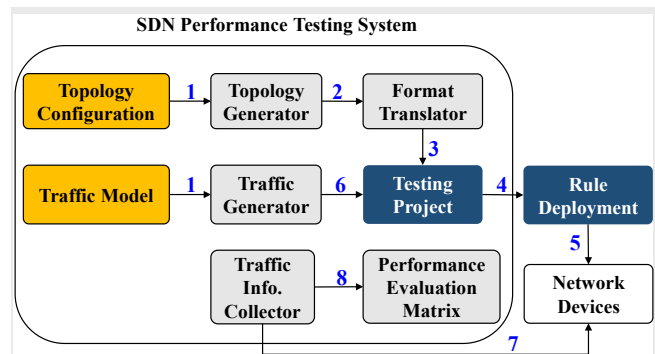


圖 1 SDN效能測試系統流程圖

### 3.2 效能評估矩陣

本節說明SDN效能測試系統所使用的評估矩陣，此矩陣目前提供三種基本的評估項目，分別為平均網路頻寬使用率(average bandwidth utilization)、線路使用標準差(standard deviation of link utilization)、虛擬主機之間的封包傳輸延遲時間(packet transmission delay)，以下說明這三個評估項目。

#### 3.2.1 平均網路頻寬使用率

由於傳統網路採用擴展樹協定(Spanning Tree Protocol; STP)協定，故所有封包均在同

一個封包傳輸路徑上傳輸，造成網路其他實體線路的使用率不高，間接導致主機傳送封包至乙太網路交換機的數量受到限制。本項目評估當採用欲測試的網路專案時，主機連接乙太網路交換機傳送封包的平均網路頻寬使用率。如公式(1)所示，令符號 $U$ 為平均網路頻寬使用率、 $n$ 為主機總數、 $L$ 為所有實體線路 $l$ 的集合、 $c_l$ 為一實體線路 $l$ 的最大傳輸量(Maximum capacity)、 $o_h$ 為一主機 $h$ 的對外流量而 $H$ 為所有主機的集合。 $U$ 值越高代表在使用此網路專案的情況下，主機能傳輸的封包較多，該網路專案能提供良好的封包傳輸效能。

$$U = \frac{\sum_{h \in H} \frac{o_h}{c_l}}{n} \quad (1)$$

### 3.2.2 線路使用標準差

除了平均網路頻寬使用率之外，另一個評分項目為線路使用標準差。令符號 $\xi$ 為網路流量平均值、實體線路集合 $L$ 有 $m$ 條線路 $l \in L$ ，而符號 $t_l$ 為實體線路 $l$ 上的網路流量，網路平均流量計算如公式(2)所示。

$$\xi = \frac{\sum_{l \in L} \frac{t_l}{c_l}}{m} \quad (2)$$

而線路使用率標準差 $\varphi$ 則透過公式(3)計算，

$$\varphi = \sqrt{\frac{1}{m} \sum_{l \in L} \left( \frac{t_l}{c_l} - \xi \right)^2} \quad (3)$$

若 $\varphi$ 值越低則表示各條實體線路之間的流量較平均，顯示所套用的網路專案擁有較好的網路負載平衡能力。

### 3.2.3 封包傳輸延遲時間

本系統目前最後一個使用的評估項目為封包傳輸延遲時間，評估公式的大意為每一個主機均傳送封包至網路中所有主機。當本系統獲得每個主機收到封包的傳輸時間後，便能計算封包傳輸延遲時間。令VLAN集合 $V$ 中含有 $k$ 個VLAN，VLAN  $v \in V$ 中含有 $x$ 個主機，而該VLAN  $v$ 的主機集合為 $M_v$ ，符號 $d(m,n)$ 表示主機 $m$ 與 $n$

之間的封包傳輸延遲時間，封包傳輸延遲時間的計算方式如公式(4)所示。

$$\rho = \frac{\sum_{v \in V} \frac{2 \sum_{m,n \in M_v} d(m,n)}{x(x-1)}}{k} \quad (4)$$

當 $\rho$ 值越低時，表示當使用該網路專案的傳輸路徑傳送封包，其傳輸對之間的傳輸延遲較低，該網路專案能獲得較好的分數。

## 4 · 系統實作畫面

本章將於4.1節說明待測試的SDN解決方案，並在4.2節介紹本系統之部份實作畫面。

### 4.1 The SDN Solution-Peregrine

本系統欲測試的網路專案為工研院雲端中心提出的Peregrine[11]SDN解決方案，其採用L2 Fabric架構，提供網路虛擬化、動態路徑規劃與快速故障轉移等功能。Peregrine主要特色之一是能即時收集網路流量進行動態封包路徑規劃。為了評估Peregrine的效能，本系統介接Peregrine進行測試，Peregrine目前提供兩種路徑規劃演算法，分別為Simple與Greedy Load-Balanced (GLB)演算法。由於本效能測試系統針對GLB演算法進行評估，以下將介紹GLB演算法：Peregrine計算封包傳輸路徑時，會考慮每條實體線路(physical link)上目前容納的VLAN線路數量，意即Peregrine計算VLAN樹(VLAN Tree)時，若該實體線路上已存在某一VLAN樹所包含的線路，其將換一個實體線路佈建新VLAN樹的線路，如此一來每個VLAN樹的線路便均勻分散在所有實體線路上，充分利用網路資源以提升實體線路使用率。

### 4.2 系統實作畫面

本節將展示SDN效能測試系統產生網路拓樸的過程及演算法評估結果。為了產生網路拓樸，使用者須提供兩個配置檔，分別為topology.json (圖2)及host.csv (圖3)。其中topology.json內含四個參數，參數" type" 可讓使用者決定拓樸架構，參數" core"、" edge" 及" edge-to-core"

則允許使用者自行決定網路拓樸的大小。



圖 2 使用者上傳的自定義topology.json檔

而host.csv(圖3)則讓使用者自行定義虛擬主機使用的VLAN ID(A欄位)、虛擬主機個數(B欄位)及其可使用的網段(C欄位)。透過使用者上傳的兩個檔案，本系統便能利用Topology Generator產生網路拓樸。

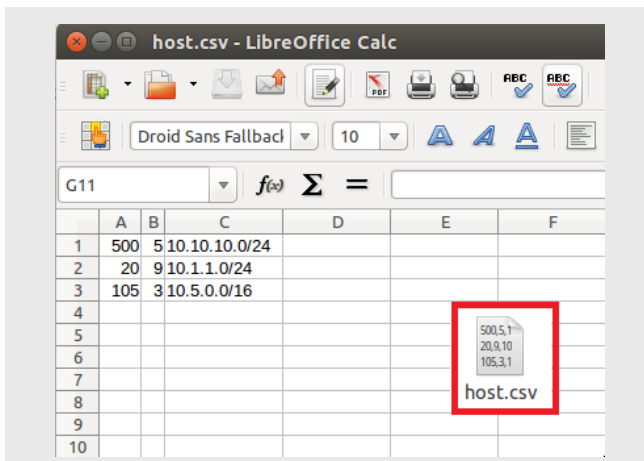


圖 3 使用者上傳的自定義host.csv檔

如圖4所示，使用者接著透過終端機輸入建立網路拓樸腳本指令，並在指令最後提供topology.json與host.csv的檔案位置，如input1與input2。

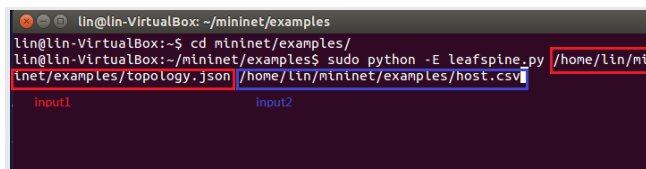


圖 4 使用者於終端機執行網路拓樸建立腳本指令

輸入指令後，系統將呈現使用者建立的網路各節點，包含虛擬主機、乙太交換機與線路等相關資訊，並可讓使用者利用CLI進行網路架構的檢視動作，如圖5所示。

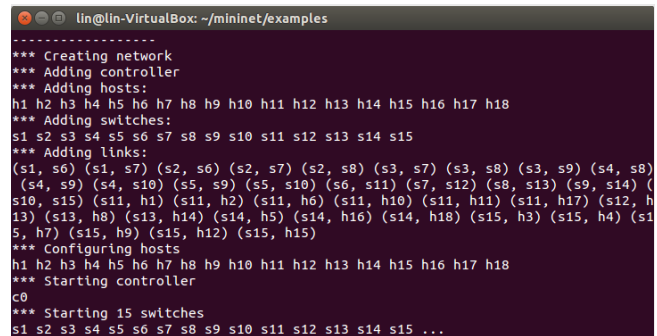


圖 5 本系統透過Mininet讓使用者建立網路拓樸

本系統除了提供使用者進行網路訊息檢視外，拓樸建立腳本亦記錄各節點的資訊，而節點資訊則透過Format Translator轉換為Peregrine能讀取的格式，如圖6所示。

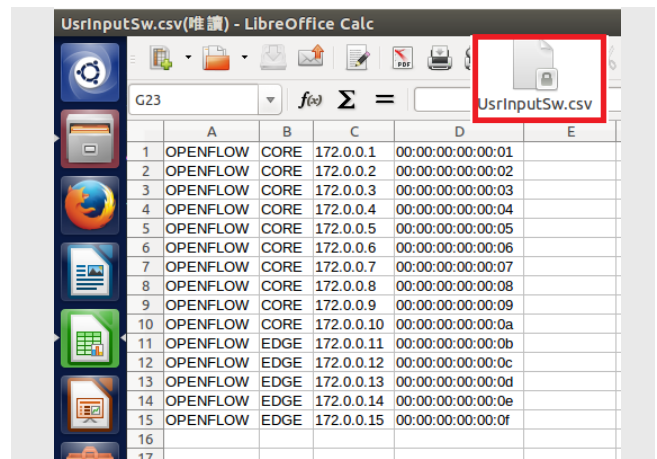


圖 6 網路拓樸資訊透過Format Translator轉換格式

接下來展示當本系統套用Peregrine進行效能評估的情況。由於本系統目前僅完成平均網路頻寬使用率與線路使用標準差等評估項目實作，故以下說明本系統針對Peregrine GLB演算法的評估結果。圖7為Peregrine GLB與STP的平均網路頻寬使用率比較，其中x軸為平均網路頻寬使用率而y軸為乙太網路交換機之間的實體線路數量。如圖7所示，當使用GLB演算法後，乙太網路交換機之間的實體線路其網路流量都能均勻分散，以避免超過每條線路的最大傳輸能力，而STP則讓所有VLAN的流量走同一個傳輸樹，間接導致各主機的平均網路頻寬使用率較低，透過公式(1)可知GLB演算法的

效能明顯優於傳統STP的做法。

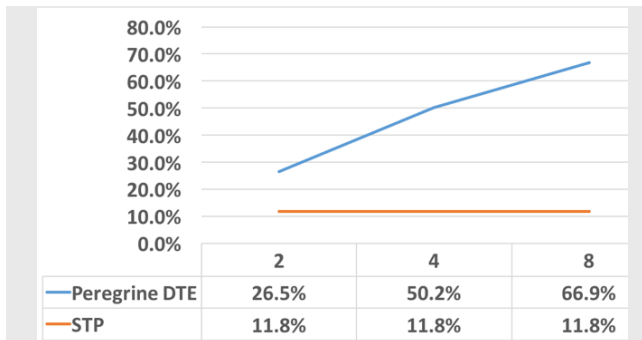


圖 7 Peregrine GLB與STP的平均網路頻寬使用率比較

圖 8 為 Peregrine GLB 與 STP 的線路使用標準差比較，其中 x 軸為線路使用標準差而 y 軸為乙太網路交換機之間的實體線路數量。如圖 8 所示，當使用 GLB 演算法後，由於每個 VLAN 樹的傳輸路徑都均勻分散在整個實體網路的線路上，透過公式 (3) 可知 GLB 演算法的效能明顯優於傳統 STP 的做法。

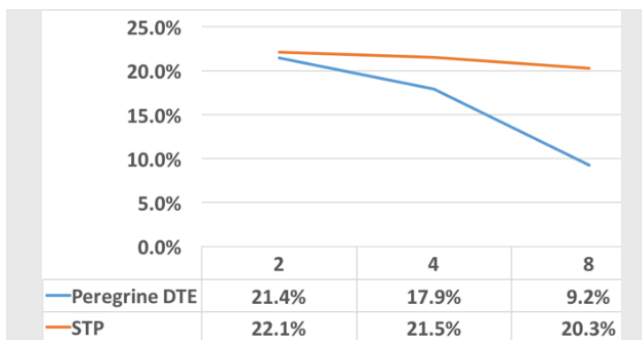


圖 8 Peregrine GLB與STP的線路使用標準差比較

## 5 · 結論

由於 SDN 技術能讓網管人員快速且無誤的下達指令或規則至指定的網路設備，此特性讓各家廠商或組織紛紛提出 SDN 解決方案，使得 SDN 技術發展更加迅速。但由於目前並沒有一套評估 SDN 測試評估系統，導致使用者無法了解各套網路專案之間的效能差異。本論文提出一 SDN 效能評估系統之架構設計，能測試與分析採用某一特定網路專案後的效能，並提供評估報告給使用者。而未來本系統將持續完成

整個 SDN 效能測試系統之實作，且預計增加更多效能評估項目，讓使用者了解各網路專案的特性。

## 參考文獻

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, USA, Aug. 2008.
- [2] Ryu, SDN Controller [Online]. Available: <https://osrg.github.io/ryu/>
- [3] NOX, SDN Controller [Online]. Available: <http://www.noxrepo.org/>
- [4] POX, SDN Controller [Online]. Available: <https://github.com/noxrepo/pox>
- [5] OpenDaylight, OpenDaylight Lithium-SR2 [Online]. Available: <https://www.opendaylight.org/software/downloads/lithium-sr2>
- [6] OpenContrail, SDN Controller [Online]. Available: <http://www.opencontrail.org/>
- [7] OpenDaylight Beryllium, [Online]. Available: <https://www.opendaylight.org/software/downloads/beryllium>
- [8] Mininet, [Online]. Available: <http://mininet.org/>
- [9] EstiNet, [Online]. Available: <http://www.estinet.com/>
- [10] OpenFlow, [Online]. Available: <https://www.opennetworking.org/index.php>
- [11] IEEE 802.11, [Online]. Available: <http://www.estinet.com/>
- [12] C. Y. Lin, "A VLAN-Based Peregrine SDN Solution," *J. Informat. Commun. Technol.*, vol. 161, pp.21-28, 2015.

## 作者簡介

王姿琳



於2014年取得淡江大學資訊工程學系博士學位，現任工研院資通所資料中心架構與雲端應用軟體組工程師。專長為雲端運算、軟體定義網路及無線感測網路技術。目前從事軟體定義網路技術開發。

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

方湘婷



於2015年取得國立暨南國際大學資訊工程學系碩士學位，現任工研院資通所資料中心架構與雲端應用軟體組副工程師。專長為雲端運算、軟體定義網路、自動化佈建平台。目前從事軟體定義網路技術程式開發與測試佈建平台建置。

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

林育勳



於2015年取得中興大學資訊工程學士學位，2016年就讀於中央大學資訊工程學系碩士一年級。現任工研院資通所資料中心架構與雲端應用軟體組實習生，目前從事軟體定義網路技術研究。

E-mail: [jacky26822865@gmail.com](mailto:jacky26822865@gmail.com)