TWAREN 可程式化交換器實驗場域佈建工具

周大源　黃文源　曾惠敏　劉德隆
財團法人國家實驗研究院國家高速網路與計算中心
E-mail: {1203053, wunyuan, 0303118, tlliu}@narlabs.org.tw

摘要

本論文擬提出一套 TWAREN 可程式化交換器實驗網路的整合式的視覺化實驗場域佈建工具。透過 Web UI 的方式，可以依據程式開發者所選定的 model 建構基礎骨架 (skeleton) 系統，並可針對基礎骨架部份的邏輯運作方式進一步地撰寫相對應的程式碼。而在程式碼佈建階段，則可以藉由本系統提供之佈建工具選擇擬更新之遠端可程式化交換器來進行同步更新。透過這套系統，可以協助網路管理人員，提升管控效率。在這樣的基礎上，我們也將提供國內學研界更多可程式化交換網路之實驗環境。

關鍵詞：可程式化交換器, 軟體定義網路

Abstract

This paper proposes an integrated visualized deployment system for the experimental environment of programmable switches TWAREN. Via the Web-UI system, the skeleton procedures can be initially constructed. Then, the programmers can further construct the corresponding codes for each procedure. The proposed system can help network administrators manage the network more efficiently. Based upon the proposed system, we can provide more experimental environments of programmable switches for the organizations of research and education.

Keywords: programmable switches, P4, software-defined network, SDN

1. 前言

近年來，新一代的資訊技術蓬勃發展，包含人工智慧 (artificial intelligence)、區塊鍊 (block chain)、雲端運算 (cloud computing) ，以及大數據 (big data) 等等相關領域與應用與日遽增。在這些先進資訊技術，至關重要的是高速而穩定的網路建設。為了要能夠強化內部資訊傳遞能量與速率，並與國際介接，各國紛紛建置高速骨幹網路以串接各大學研界與業界組織。然而，為了要能夠管理規模日漸複雜的各類網路架構，網路的通訊協定與管理技術是相當重要的。

現行網路多數以 TCP/IP 通訊協定為基礎，將協定實作在硬體晶片中，並嵌入各類網路裝置（路由器、交換器、...等等）。為了要能夠與現存網路裝置溝通，並達到相容效果，許多功能都要遵循現有的通訊協定，能夠客製化的部份相當有限。對於網路維運或管理人員而言，一旦網路發生問題，要除錯將會是非常困難又耗時的程序。

為了要讓網路架構能夠更加開放、讓工程師能夠高度參與網路裝置之運作，達成客製化的目的，軟體定義網路 (Software Defined Network, SDN) 的技術應運而生。這項技術主要源自於 Stanford 大學的一項計畫[1]。該項計畫提出新的網路設備溝通協定，稱為 OpenFlow。如圖1所示，SDN 的技術主要是將網路中的控制平面(control plane) 與資料平面(data plane)分開。前者與後者分別為 SDN 網路控制器與 SDN 交換器。

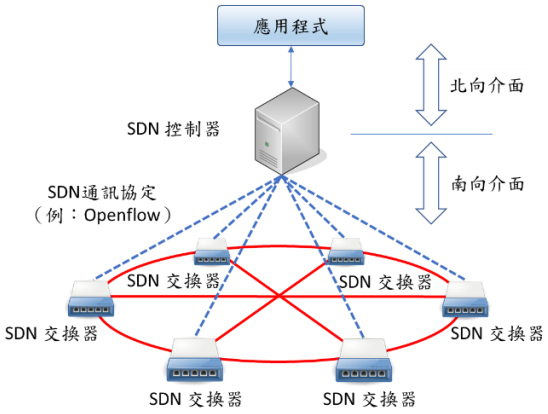

圖 1 軟體定義網路架構圖

在 SDN 網路中，SDN 控制器通常是在另一部伺服器上，以軟體方式實作應用，透過獨立通道（一般為 TCP 或 SSL 連線）來控制多部 SDN 交換器，又稱為南向介面。在 SDN 控制器的部份，可以讓網路管理者透過 policy 或是 rule 設定，來控制資料傳輸的方式。而 SDN 交換器則是專注於資料的轉送(forwarding)。在每部 SDN 交換器會透過許多資料路徑互相連接，當收到用戶 host 傳來的資料封包時，會先查閱交換器本身內部的 flow table ，檢測是否有 match 的部份。若有 match 的 flow，則會直接套用該 flow 進行轉送。相反地，若是沒有 match 的 flow，就會透過獨立通道向控制器詢問資料處理的方法。因此，在 SDN 的網路中，只要控制器與交換器均遵循同樣的 SDN 通訊協定(例如 Openflow)，便可以讓控制器順利對交換器進行管控。如此一來，SDN 控制器與 SDN 交換器並不限制一定要同一種廠牌。再者，使用者可以透過北向介面的各種應用來針對 SDN 進行控制，達成控制面的客製化功能。

雖然網路管理者能夠針對控制平面的部份進行客製化管控，並能夠實作各類型的第三方應用 (3rd -party Application)與控制器溝通，但對於資料平面，亦即 SDN 交換器的客製化程度卻不足。為了要進一步針對資料傳輸的部份進行客製化與可程式化，可程式化交換器 (簡稱 P4) 的技術便應運而生。

P4[2][3]交換器 (Programmable Protocol- Independent Packet Processors, P4) 是一種與協定無關的可程式化交換器。藉由撰寫 P4程式語言，使用者可以進一步針對 P4的行為定義相對應的處理程序，讓 P4交換器達成更高度客製化的特性。換句

話說，程式開發者可以藉由程式語言的撰寫，打造更加符合需求的交換器。有關於 P4程式語言相關工具，可以參考 P4 Language 的網站 [2]，在網站上有詳細的安裝過程。一般使用者可以依照該資訊逐步安裝。另外，如果想要省略這個部份，也可以下載 P4官方網站所提供的虛擬機器版本 tutorial 來進行測試。

台灣先進學術研究網路 (TaiWan Advanced Research and Education Network) 是串接國內外多個學研單位的寬頻網路。TWAREN 本身是由國網中心維運管理，提供多種7x24小時服務。不但肩負起各大學研單位間的大資料高速傳輸，本身亦能夠架設許多網路相關的實驗測試平台。

本中心於105年度進行 TWAREN 100G 骨幹升級作業[4]。圖 2 為 TWAREN 100G 線路架構，包含5個骨幹(Core)節點與12個區域網路(GigaPop)節點。在5個主節點之間線路頻寬均為100G，而區域網路節點之間則為50G。更多 TWAREN 相關資訊可以參閱 TWAREN 網站[5]。

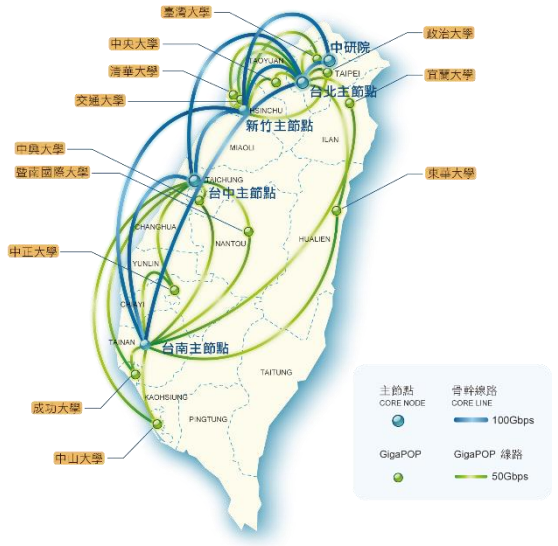

圖2 TWAREN 100G 網路架構圖

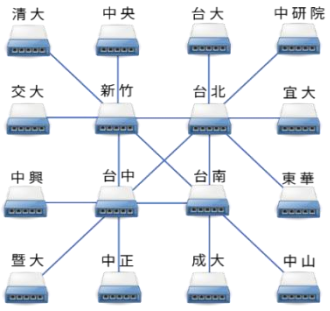

圖 3 TWAREN 虛擬專用連線 topology

為了更進一步地研究軟體定義網路技術，本團隊持續在 TWAREN 上建置國內大型的 SDN 實驗場域，亦即在4個骨幹節點與12個區網節點佈建 SDN 交換器。其間之資料路徑由 TWAREN 骨幹的 VPLS 線路達成。TWAREN SDN 網路架構如圖3所示。透過開放使用實驗性質服務方式，以實際使用案例的方式驗證 SDN 應用，例如：

- 本中心台中分部與美國西北大學 iCAIR 實驗室之間的 DTN 實驗線路。
- 與國立交通大學共同進行 SDN-IP 網路介接，並與日本、韓國等多個學研單位進行連線。
- 106年8月，本中心提供 TWAREN SDN 虛擬專用連線，與使用與中華電信研究院、工研院、資策會等單位一同進行世大運影音轉播[6][7]。

本團隊亦開發一套 Web UI 版本的虛擬網路供裝系統[8]，並進行高速傳輸測試[9]。利用 Web GUI 介面的方式取代指令輸入，無需記憶大量指令。再者，利用視覺選單管理者點選方式來取代資料輸入，避免資料輸入錯誤。另外，以圖形界面顯示網路 topology，取代大量 JSON 資料，讓管理者更能在短時間內掌握網路狀態。而眾多歷程記錄則是結合資料庫功能，不但可以將各類資訊以額外登載，亦可以藉由系統自動參照，方便管理者識別用戶資訊、節點資訊、port 資訊。

為了要提供國內學研界更多可程式化交換器實驗服務，本單位未來將進一步地於 TWAREN 之骨幹節點與區網節點擇取數個節點來進行硬體可程式化交換器佈建工作，提供國內大型遠距線路實驗場域。考慮到未來網路管理方面的實務應用，針對管理人員而言，倘若需要針對多部可程式化交換器進行程式碼更新作業，將會非常耗時又沒有效率。目前在 P4交換器當中尚無顯示可程式化交換器的工具，也沒有供裝與管控的工具。因此，如果能夠有一套整合式的視覺化程式撰寫、更新部署與管理工具，將能夠有效地協助管理人員針對可程式化交換網路進行管控，提昇效率。

因此，本論文擬提出一套針對可程式化交換器實驗網路的整合式的視覺化實驗場域佈建工具。透過 Web UI 的方式，可以依據程式開發者所選定的 model 建構基礎骨架 (skeleton) 系統，並可針對基礎骨架部份的邏輯運作方式進一步地撰寫相對應的程式碼。而在程式碼佈建階段，則可以藉由本系統提供之佈建工具選擇擬更新之遠端可程式化交換器來進行同步更新。

本論文的主要內容如下。第2.節會針對P4交換器進行簡介。第3.節會針對我們解決方案進行解說。實作的系統在第4.節中展示。而最後一節則是結論與未來展望。

2. P4交換器

可程式化交換器[2][3] (Programming protocol-independent packet processors, P4) 是一種與協定無關的可程式化交換器。藉由撰寫 P4程式語言，使用者可以進一步針對 P4的行為定義相對應的處理程序，讓 P4交換器達成更高度客製化的特性。

圖4是 P4程式語言的 Pipeline。程式設計者可以使用 P4 語言定義處理資料封包的方式，並編譯產生一個 JSON 檔案以針對交換器晶片進行組態設定。程式設計者也可以使用 P4語言定義各種交換器、防火牆，或者負載平衡器、...等等裝置。

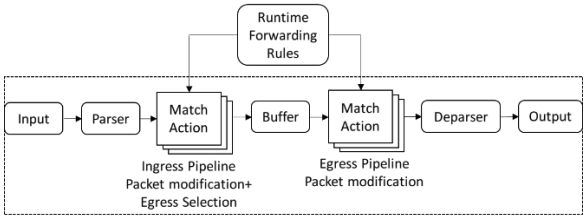

圖4 P4 Pipeline

交換器在收到資料封包後，會經由 Parser 做剖析，得出偵測指令與需偵測的對象。資料封包稍後則進入 match + action 階段進行處理。此階段會進行 Ingress Pipeline 處理，得出相對應的 Egress，並修改資料封包。基於 Runtime Forwarding Rules，封包轉送至 Egress 進行 match + action，並再修改資料封包後輸出。

基礎的 P4交換器 Simple switch 的基礎程序如下：

```
V1Switch(
   MyParser(),
   MyVerifyChecksum(),
   MyIngress(),
   MyEgress(),
   MyComputeChecksum(),
   MyDeparser()
) main;
```

如上所示，基礎的 V1 model switch 就是經過上述的標準運作程序。在 MyParser()中主要是要針對封包進行剖析。而 MyVerifyChecksum()是用來驗證檢查碼是否正確。接下來 MyIngress()用以處理封包進入的 port 的相關程序。而 MyEgress()用以處理封包輸出 port 的相關程序。藉由 MyComputeChecksum()程序，可以針對封包的檢查碼進行更新。在所有對應的處置動作完成後，MyDeparser()會將最後結果包裝成對應的封包並傳到輸出的 port 中。

在 P4 可程式化交換器中，需要注意的是以下幾部份。

- Behavior Model(BMv2)：這個部份是用來描述硬體架構。
- P4 compiler：用來編譯 P4程式碼的工具。
- P4 runtime：這個是 P4程式碼的執行環境。

對於語法部份，P4官方網站也有提供 P4 cheat sheet [10] 文件，讓程式開發者能夠針對重要的關鍵語法進行參考。

3. 實作方式

3.1 佈建系統架構

由於 P4交換器運行於 Linux 作業系統中，一般而言，若要撰寫 P4交換器的程式，都是以 SSH 連線方式登入遠端的 P4 switch 中進行程式碼撰寫、編譯並執行。

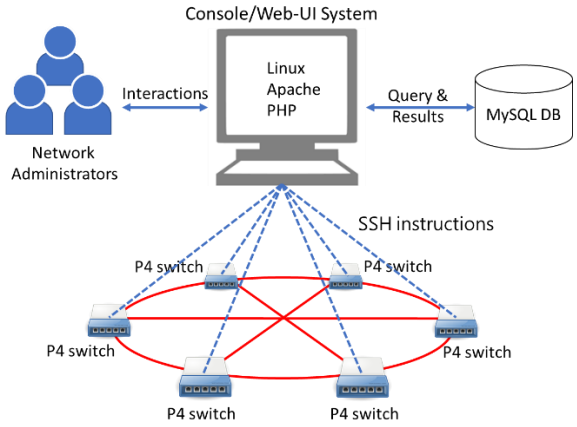

圖5 P4佈建工具架構圖

如圖5所示，本論文所開發之 P4程式部署系統透過可以執行 SSH 連線之安全通道與多部互相連接之 P4交換器連線。其中，佈建系統是以 LAMP 系統，包含 Linux 系統、Apache 伺服器 MySQL 資料庫系統，以及 PHP 程式語言所構成。另外，系統中也採用 HTML5網頁與 Javascript 程式輔助系統功能運作。

由於 P4交換器是安裝於 Linux 作業中，因此在 Console 中採用 LAMP 架構之系統在相容性與互通性上均為相當適用的選擇。再者，上述 LAMP 方案屬於 Open Source 方案，客製化程度較高。

在本系統中，網路管理者或程式開發者可以用瀏覽器連線至 Console 中的佈建系統。透過瀏覽器與網頁的互動，便可以連線至各大 P4交換器進行程式碼修改與佈建。而 MySQL 資料庫則是用於 P4程式碼的版本管控與備份。

3.2 HTML5 網頁

現行的網頁語法格式為超文字標記語言(Hyper Text Markup Language, HTML)，目前的新版本為 HTML5。在 HTML5 中最大的特色就是對於多媒體的支援，包含圖形、聲音、影片等等，都有較前幾個版本有更完整的支援。

另外，HTML 網頁中原本就有的表單(Web form)是與使用者互動瀏覽的工具。表單介面中有多種輸入工具，可供使用者透過表單的介面輸入相關資訊，並以 GET 或 POST 方式傳至 server 端。

3.3 JavaScript

JavaScript 程式語言屬於 client 端動態網頁程式語言。在使用者瀏覽網頁時，JavaScript 使用 client 端瀏覽器的解譯功能，得到相對應的結果。

早期 JavaScript 僅僅只是網頁設計者用來增加一些網頁特效的小工具，或者是協助進行資料檢驗用的工具。近年來，在非同步技術 JavaScript (Asynchronous JavaScript and XML, AJAX) 與 Web

技術興起，JavaScript 的地位日趨重要。藉由 JavaScript 的效果，以往在桌上型應用程式的與網頁程式之間的差距變得愈來愈小。許多企業更傾向以 Web 網頁方式來開發各種管理系統。

然而，因為瀏覽器種類繁多，對於 JavaScript 的支援程度並不一致，致使 JavaScript 使用者開發程式時必須留意各種瀏覽器版本的相容性。為了能夠適應各種瀏覽器間的差異，並且降低開發時之成本，開發者可以套用許多以 JavaScript 程式語言為基礎的公用函式庫或者框架（framework）來進行擴充應用。

3.4 PHP 動態網頁

在 WebUI 系統中，主要是以 PHP (Hypertext Preprocessor) 程式語言進行開發。其特性如下所述。PHP 程式語言是 server 端的動態網頁，因此會在 server 端先進行轉譯，並收集相關資訊後，可以形成相對應的回應訊息，並傳至 client 端的瀏覽器上。因此，client 端不需要額外安裝特殊的軟體、亦不需要進行額外環境設定。只需要使用網頁瀏覽器，便可以瀏覽與操作這套系統。

PHP 程式語言可以與 MySQL 資料庫進行互動存取。在 P4 交換器中有許多資料都是在 shell 中的 CLI 上顯示。在一般情況下，若是使用者要將 shell 回應內容寫至 MySQL 資料庫中，只能以手動方式輸入、pipeline，或者撰寫 shell script 來達成。而 PHP 程式語言對於與 shell 及 MySQL 的互動兩方面，皆有相當完整的支援。因此，使用 PHP 程式語言作為 shell 與 MySQL 之間溝通的渠道，是相當便利的解決方案。

PHP 程式語言可以進行系統呼叫，亦即可以與本身所在的主機之作業系統直接進行互動，不但可以傳送指令給作業系統，也可以接收作業系統傳回的訊息。因此，若欲整合互動的應用程式是 CLI 版本的應用程式，如本研究中所要整合的 P4 交換器系統，在整合上的困難度將會大大降低。

然而，由於本論文所開發的 Web-UI 系統必須連線至遠端主機，而非單純地呼叫本身所在的系統，故無法使用一般的系統呼叫。因此，我們必須額外安裝 PHP 本身所提供的 SSH 連線函式庫[11]。透過這套函式庫，系統有兩種模式可以呼叫遠端主機：第一種是一次函式（即 cmdExec），而第二種是開啟一個 Shell（即 openShell），直到關閉此 Shell 為止。如果要執行單一指令，且每個指令皆為獨立事件，則採用前者。反之，如果要執行多道指令，而且每道指令皆有執行順序的關係，則採用後者。

3.5 建立信任連線

由於 P4交換器是安裝在 Linux 系統中，使用者可以透過 SSH 工具進行遠端登入。因此，假設使用者在 client 與 server 兩端皆有帳號，若要登入遠端 P4交換器並下相關指令時，只要在對端主機顯示要輸入密碼的提示字元時，輸入相對應的密碼即可。

上述步驟在 CLI 介面操作並不會發生問題。然而，若要以 PHP 實作為 Web 介面時，就會無法正常運行。由於 PHP 程式語言僅會執行一次 SSH 的指令，無法達成即時互動機制。因此，本研究以 P4交換器與主控台間設定 SSH 互相信任的免密碼機制。

如圖6所示，若要個別建立 Console 與 P4 switch 之間的信任連線，可以將雙方存取點伺服器的 RSA public key 複製到對方主機，即可讓兩部主機成為互相信任主機，免去輸入密碼的步驟。如此店可克服無法及時互動輸入密碼之問題。

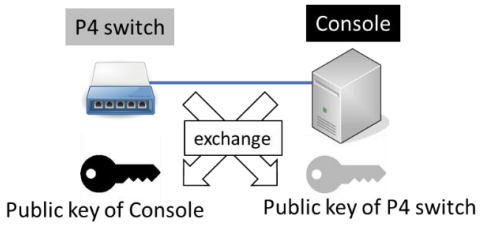

圖6 建立信任連線方式

3.6 MySQL 資料庫規劃

為了要能夠將許多額外資訊記載下來，本系統採用 MySQL 資料庫作為資料記載的儲存體。在資料庫中主要的輔助資料表主要有記載每一次修改 P4 程式碼的記錄。表格中的欄位記載程式碼流水號、P4 檔案名稱、修改時間，以及佈建時間。針對該版程式碼佈建的目標也會記載在表格中。

表1 程式碼佈建資料表

欄位名稱	資料型態	意義
CodeID	Integer	程式流水號
FileName	String	P4 程式名稱
Switches	String	目標交換器之代碼（以逗號分隔）
CreateDate	DateTime	程式碼建立時間
ModDate	DateTime	程式碼修改時間
DepDate	DateTime	程式碼佈建時間
Remarks	String	其他註解

透過這個資料表，就可以將所有 P4 程式碼的資訊記載下來。一旦有異常發生時，便可以瞭解發生問題的部份，並可以退回舊版本修復，以便進行故障排除。

4. 系統展示

4.1 實驗環境

目前若要架設可程式化交換網路的實驗環境，可以藉由以下幾種方案來進行：

- 硬體交換器方案：以多部硬體交換器與實體電腦主機進行，並透過實體線路進行介接。

這種方式最為實際，但成本也最高。對於尚未有 P4硬體交換器之單位入門門檻較高。

- 虛擬機器內方案：在同一部虛擬機器內安裝多部 P4交換器，並由 Mininet 等軟體達成 topology 設置。
- 虛擬機器間方案：這種方式是上述兩種方式的折衷方式。使用多部虛擬機器，每部虛擬機器代表一部 P4交換器或一部 client host。在虛擬機器間可以使用內部網路的方式進行介接，亦即將各大虛擬機器之網路介面卡指向同一個虛擬網路，便可以模擬實體線路的介接。這種方式可以節省硬體投資成本，也可以貼近實體網路卡介面之狀況。

在本實驗中，我們採用虛擬機器內與虛擬機器間兩種方案交替使用，達成功能性測試。

4.2 程式碼撰寫階段

如圖7所示，在程式碼撰寫階段中，系統首先會提示程式開發者先輸入 P4程式的檔案名稱。接下來，在 Including File(s) 欄位中，程式開發者可以勾選需要引入哪些 P4程式碼的檔案。一般而言，P4程式都會引入 core.p4。而程式開發者若要套用 v1model，則會引入 v1model.p4。在按下 Generate 按鈕後，系統便會根據目前所選擇的 p4程式碼檔案產生目標 P4程式的骨架(skeleton)程式碼。

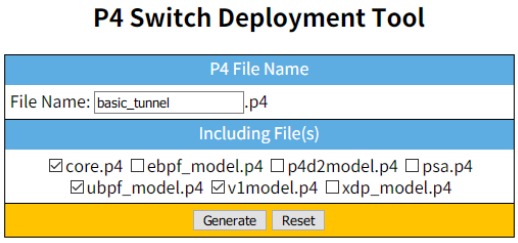

圖7 檔案命名與引入檔案作業

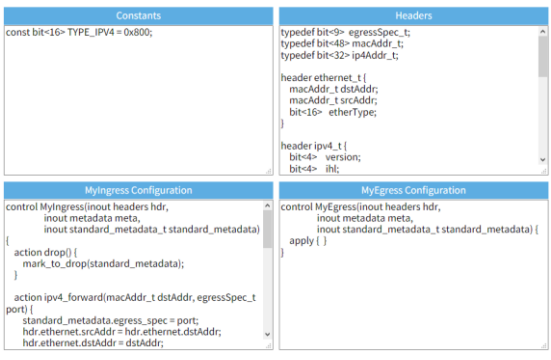

圖8 程式碼編輯畫面1

圖8與圖9是程式碼編輯畫面。由於目前所套用的模型為 v1model，因此系統會比照 v1model 的標準程序產生出對應數量的區塊。程式開發者也可以分別檢視各大程式碼區塊的程式碼是否已經填入。同時，程式開發者也可以交叉比對各畫面之間的變數宣告、呼叫等等作業，便於識別。

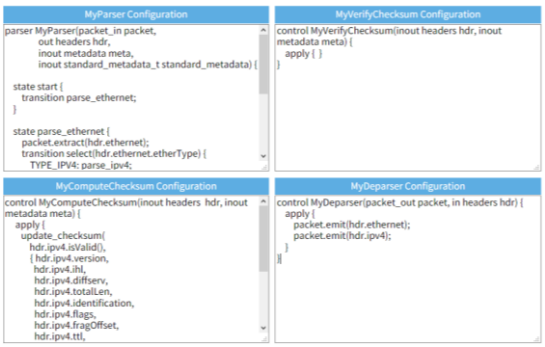

圖9 程式碼編輯功能畫面2

4.3 程式碼佈建階段

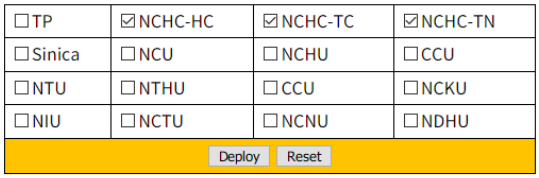

圖10 P4程式碼佈建功能畫面

在程式碼佈建階段中，程式開發者可以透過網頁介面點選方式選擇，希望將目前修改完畢的 P4程式碼佈署至哪些 P4交換器中。在此我們以模擬 TWAREN 的三個節點，分別為 NCHC-HC、NCHC-TC，以及 NCHC-TN。

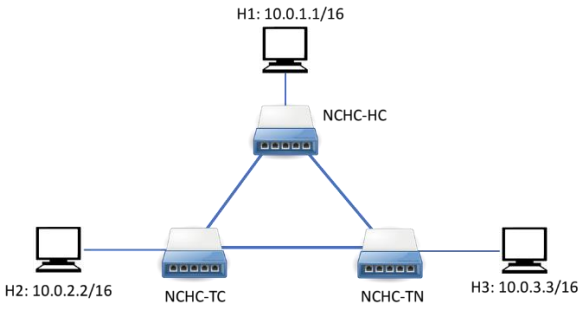

圖11 NCHC-HC、NCHC-TC，與 NCHC-TN 連線 topology

如圖11所示，我們指定介接在三部 P4交換器 NCHC-HC、NCHC-TC，以及 NCHC-TN 上之 hosts H1、H2、H3 之 IP 位址分別為 10.0.1.1、10.0.2.2，以及10.0.3.3。

透過本系統之佈建，在三部交換器的程式碼佈建完畢後，可以針對每一部交換器對其他兩部交換器進行 ping 通測試。連通測試結果分別為圖12、圖13，以及圖14所示。

經由本佈建系統，程式開發者可以將程式碼佈建至所選擇之 P4交換器，包含 NCHC-HC、NCHC-TC，以及 NCHC-TN 上的交換器進行更新，達成交換器之連通性。

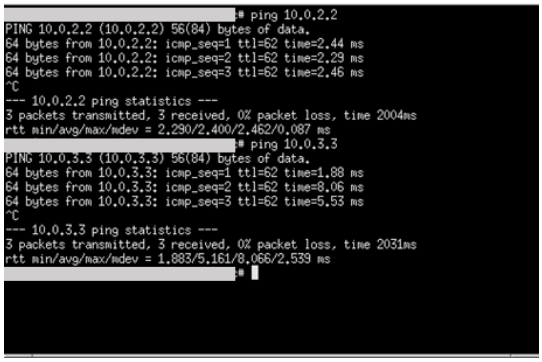

圖12 NCHC-HC 對其他兩點發送封包情況

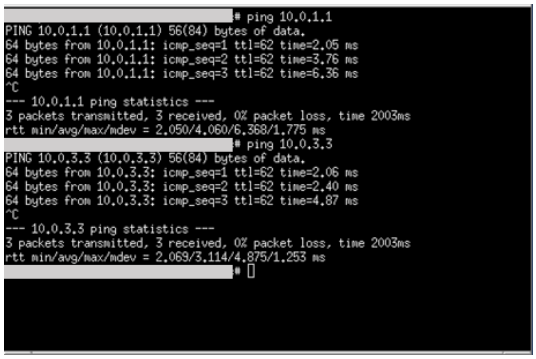

圖13 NCHC-TC 對其他兩點發送封包情況

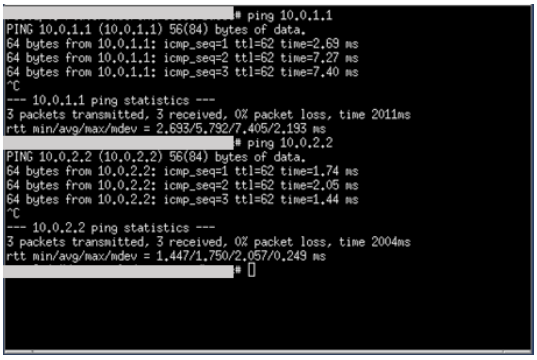

圖14 NCHC-TN 對其他兩點發送封包情況

5. 結論

本論文展示一個視覺化可程式化交換器系統。透過這套系統，可以協助網路管理人員，提升管控效率。未來我們將在台灣先進學術研究網路 TWAREN 上的骨幹節點與區域網路節點中擇取部份節點進行硬體式可程式化交換網路佈建，期望能夠提供給學研界更多實驗用資源，協助學研單位導入並驗證可程式化交換網路技術，增進國內可程式化網路技術研究能量。

參考文獻

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, April 2008.

[2] P4 Language Consortium, https://p4.org/

[3] A. Sivaraman, C. Kim, R. Krishnamoorthy, A.t Dixit, and M. Budiu, "DC.p4: Programming the Forwarding Plane of a Data-Center Switch," Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, Article No. 2, 2015.

[4] 台灣 100G 教育學術研究網路正式啟用 http://technews.tw/2016/10/06/taiwan-100g-education-network/

[5] 台灣先進學術研究網路(TaiWan Advanced Research and Education Network, TWAREN), https://www.twaren.net/

[6] 無線轉播無限精彩 世大運賽事試驗轉播 精彩賽事不累格打造未來直播新境界 https://www.cna.com.tw/postwrite/Detail/219194.aspx

[7] 周大源, 胡仁維, 黃文源, 劉德隆,TWAREN SDN 虛擬專用連線管理系統," TANet2017集，台中，2017年10月

[8] 周大源, 胡仁維, 黃文源, 劉德隆, "WebGUI 版本虛擬網路供裝系統," 2016 年雲端與大數據研討會, 2016

[9] 周大源, 楊哲男, 古立其, 劉德隆, "TWAREN SDN 虛擬專用連線之高速傳輸應用," TANet2016論文集，花蓮，2016年10月

[10] P4 Language Cheat Sheet, https://p4.org/assets/P4WS_2018/p4-cheat-sheet.pdf

[11] Secure Shell2 of PHP, http://php.net/manual/en/book.ssh2.php