P4可程式化網路平台預約系統自動化之研討

胡乃元　黃文源　周大源　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
2103081, wunyuan, 1203053, 0303118, tlliu@narlabs.org.tw

摘要

因應軟體定義網路（SDN）的技術演進，開發者除了針對控制層的 OpenFlow 也逐漸往針對 Data Plane 的 P4語言來進行研究，但因基於 Tofino 晶片的實體 P4交換器價格昂貴，所以大多數研究者僅使用基於 V1model 的 BMv2軟體式交換器來進行開發，但因 BMv2與 Tofino 架構上的差異性因素，導致移植上實體機器會有相當多的問題等待克服，所以我們提供了 P4可程式化網路平台做相關實體交換機上之實驗，但因需要在每位不同的用戶使用時提供乾淨的實驗環境，所以我們需要重置交換機以及 VM 之設定，我們將會在本論文上研討預計上線的 P4可程式化網路平台中預約系統後台會使用到的自動化技術，待等 P4可程式化網路平台正式上線後可將 P4實體設備介接在台灣高品質學術研究網路 TWAREN 上做相關應用。

關鍵詞：SDN、P4、Tofino、BMv2、TWAREN

1. 前言

軟體定義網路（SDN）[7]是一種網路架構。SDN 將路由器中的控制平面與數據平面分開，使用 OpenFlow 協議來實現這一效果，利用集中式的 Controller 來控制網路，分為資料層、控制層和應用層，從而降低了管理的複雜性，並且允許管理人員在不改變任何硬體的情況下規劃網路以及控制流量，另外，在2008 年，由 McKeown [10]等人在斯坦福大學提出了 OpenFlow ，OpenFlow 由 Open Networking Foundation (ONF) 維護。OpenFlow 的架構包括 OpenFlow Controller、OpenFlow Switch 和安全通道。 OpenFlow Controller 使用 TCP port 6633 與設備溝通。OpenFlow Controller 可以控制底層設備，以便底層設備可以接收控制消息和更改設置。 支援 OpenFlow 協議的設備可以根據 Flow Table 轉發數據，但因開發者認為 OpenFlow 會受限於封包格式以及部分限制而無法帶出 SDN 真正的效能，所以逐漸往針對 Data Plane 開發的 P4語言研究，P4能夠突破以往 OpenFlow 的限制達到更多的功能，因為 P4實驗資源難以取得,所以使用者可以在本實驗平台中做相關測試。

本次論文在章節**2**講解了 P4相關的背景知識，章節**3**講解了 P4可程式化網路平台預約系統相關之架構，章節**4**講解了實驗平台後端測試網路拓樸，章節**5**講解了自動化執行腳本流程，章節**6**則是結論。

2. 背景知識

本章節將會講解論文所用到之背景知識。

2.1 P4(Programming Protocol-independent Packet Processors)

P4[5]是一種用於可程式化資料層的高階語言，提供比 OpenFlow 更為彈性的功能，透過 P4，開發者可以直接規劃出一個 Switch 能夠處理的封包，P4主要宣傳是可支援任何通訊協議，可支援任何平台，可隨時更改交換規則，支援任何通訊協議代表的是我們可以非常有彈性的去處理所有封包，可支援任何平台則代表可以在 FPGA、DPDK、Tofino、Smart NIC 等等上支援，不過目前仍以 Tofino 晶片為大宗，但是不同晶片上之 Pipeline[8]略有差異，在可隨時更改交換規則這點上是允許我們對各交換器設備去重新定義對封包處理的方式，這也代表我們可以指定網路設備如何去處理網路封包，由於以前網路晶片的限制，推出一種新功能都需要數年的時間來讓晶片支援，透過 P4語言以及強大的 Tofino 晶片，達到為了實現特定的封包行為，開發者可以在幾分鐘之內完成一種網路協議的更動而並非耗費幾年的時間。

2.2 P4 WorkFlow

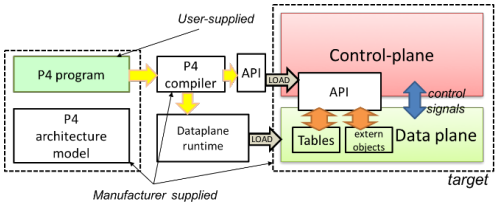

圖1 P4 WorkFlow[5]

P4的運行流程如圖1，將 P4程式透過 P4c 編譯，接下來將 P4讀取到 Data plane 設備，例如各種支援 P4 的軟硬體交換機：BMv2、Tofino Model、實體 Tofino 交換器等等，然後透過控制層利用 P4 runtime 等 API 向 Data plane 下發相關的 Entry。

2.3 Tofino

Tofino[2]是世界上第一款用戶可程式化的乙太網路交換器 ASIC，專門設計給資料中心作應用，能夠即時監視控制軟體內的封包，並使用協議獨立交換器架構(PISA)建構，這代表如需更新協議時能夠直接像是軟體升級一般的佈署，在軟體

內調整網路協議並直接編譯到交換機中。

Tofino 晶片可以同時處理四條流水線並展現優異的6.5Tbps 的吞吐量並可以提供最高100GbE 的頻寬，二代與三代則分別可以處理12.8Tbps 以及25.6Tbps 並提供到400GbE 的頻寬，但是 Intel 在目前為止已經暫停 Tofino 晶片的開發，不過使用 Tofino 晶片的產品仍在正常銷售，各國家仍然利用 P4設備來達成各個網路架構的任務。

2.4 BMv2

BMv2[3]的全名為 Behavioral model version 2，使用 V1model 的流水線，用於測試、開發 P4 的數據層以及控制層的連接，BMv2將由 P4c 從 P4 程式所生成的 json 文件導入來實驗 P4程式所指定的處理模式。但此軟體僅是為了方便測試 P4程式所使用，所以吞吐量以及效能都會比正規的軟體路由器遜色，例如與 OpenvSwitch[9]等產品比較。

2.5 P4可程式化網路平台

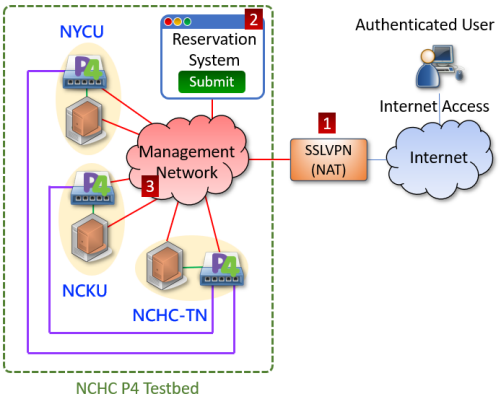

圖2 P4可程式化實驗網路平台總覽圖

圖2為本 P4可程式化實驗網路平台之總覽圖，可程式化實驗網路平台之 P4 Switch 擺放位置橫跨台南國網中心、成功大學以及陽明交通大學，本預約平台之總覽大致分為三點，第一點為登入 SSLVPN，登入 SSLVPN 後將會有相關權限去使用預約平台之相關服務，第二點為進入到預約系統內預定使用者所需要使用的時間以及哪些地點之機器，時間一到將會把相關 IP 資訊以及當次的隨機密碼傳送給使用者作登入使用，第三點則是本論文所提到之內部自動化所需流程，這將在章節**5**提到。

3. P4可程式化網路平台預約系統

本節介紹在 P4可程式化網路平台預約系統上整體之流程。

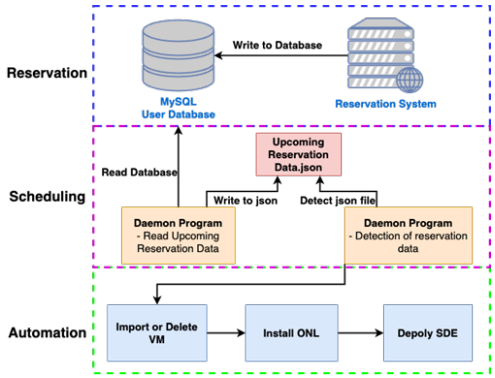

圖3 P4可程式化實驗網路平台後端總覽圖

P4可程式化實驗網路平台後端流程如圖3所示，我們透過網頁預約系統將使用者資料以及想使用 P4交換器的時間點寫入至 MySQL 資料庫中，將透過常駐程式讀取即將使用的使用者資料並寫入至 json 檔案，再透過另一個常駐程式偵測即將執行的時間點，時間一到就創建 VM 給使用者當作客戶端，這個 VM 將用來當通往 P4交換器的跳板客戶端，並且同時在使用者想要的若干台交換機上重置 ONL(Open Network Linux)[4]作業系統，重置 ONL 的步驟將在章節**5**中的**5.1**提到，並且佈署最新版 SDE 供使用者使用，這將在章節**5**中的**5.2**中提到。

4. 實驗平台後端測試網路拓樸

此章節將介紹本次論文所使用之實驗拓樸。

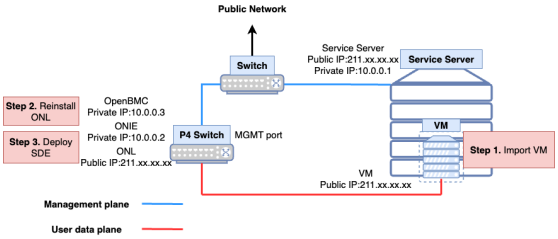

圖4 P4可程式化實驗網路平台實驗拓樸

本實驗網路拓樸如圖4， 由一台 Service Server 配有 Linux 系統，運行各常駐程式，執行指令以及派發 VM，本次實驗的 IP 位址皆為暫時性之 IP 位址。實際的 P4可程式化平台將會配合 SSL VPN 去進行 IP 分配，Service Server 配有一個 Public IP 以及 Private IP，第一步將執行匯入 VM 的動作，本台 VM 將配有一個 Public IP 供使用者連網以及遠端操作 P4交換器，第二步會進行重新安裝 ONL 作業系統的操作，P4交換器上的 OpenBMC 將會預先配置一個 Private IP 供 Service Server 做重新安裝作業系統或是佈署 SDE，安裝完 ONL 作業系統之後將是進行佈署 SDE 的操作，過程中將全自動執行。

5. 自動化執行腳本流程

本章節將會講解論文所用到之自動化執行腳本流程。

5.1 ONL 作業系統自動安裝腳本

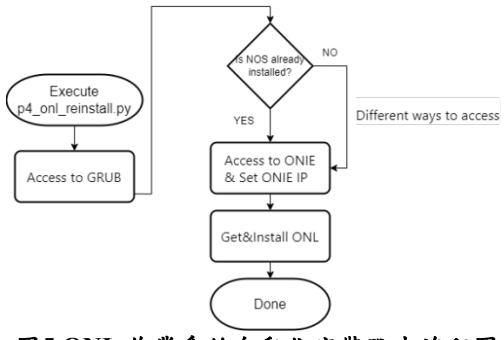

圖5 ONL 作業系統自動化安裝腳本流程圖

ONL 作業系統自動安裝腳本如圖5所示，Service Server 將在使用者使用前的一個小時三十分鐘自動開始運行自動化腳本，首先會透過 P4交換機的 GRUB 進行 OS 的安裝，這時自動化腳本將會判斷這台是已經安裝過或是新上架還沒安裝過作業系統的交換機，因為這會導致 GRUB 頁面選項之不同，所以需加此判斷，然後進入 ONIE 介面，進入後開始設定 IP，使用遠端派送的方式派發 ONL 映像檔供 P4交換機安裝，獲取完成後將進行安裝 ONL 的步驟，安裝完成後本 ONL 作業系統自動化安裝腳本執行完畢。

5.2 佈署 SDE 自動化腳本

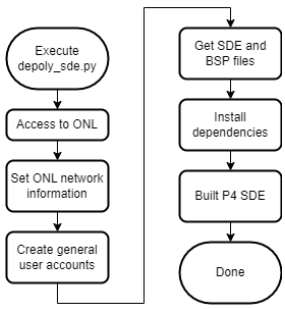

圖6 佈署 SDE 自動化腳本流程圖

佈署 SDE 自動化腳本流程圖如圖6所示，本腳本將在 ONL 作業系統自動化安裝腳本執行完畢後接續執行，若使用者需同時使用各地點之交換器，本腳本將同步執行佈署 SDE 的動作，首先會利用 root 帳號進入 ONL 設置 ONL 的 IP 資訊供作業系統連上網路，接著創建具有 root 權限的一般使用者帳戶供使用者做使用，我們會利用一般使用者的帳戶進行獲取最新版 SDE 檔案以及 BSP 檔案供交換機做使用，獲取完畢後開始安裝依賴庫（Dependencies）所需的各式程式，最後將編譯 SDE，將完成最後的步驟即可讓使用者在預定的時間獲得操作 P4交換機的權限。

5.3 自動化腳本讀取之 json 格式

```
{
    "start_day": "2023/5/1" ,
    "start_time": "6:00",
    "end_day": "2023/5/1",
    "end_time":  "9:00",
    "user_id": "000001",
    "booking_machine":["TN"],
    "already_executed":"0",
    "email":"1234@google.com"
}
```

圖7 自動化資料讀取格式示意圖

我們需要使用者透過預約網站填寫資料以獲得如圖7的資訊，分別有起始日、起始時間、結束日、結束時間、依填寫資料的順序設置 user id、預約機器的地點、是否已經執行(預設為0)以及 e-mail 讓使用者獲得使用機器之密碼。

5.4 自動化腳本執行測試時間

表 1 自動化腳本執行時間測試時間

	ONL reinstall	Deploy SDE
第一次	507秒	2779秒
第二次	508秒	2794秒
第三次	506秒	2783秒

本次測試的 Service Server 為 Dell PowerEdge R640，將會從這台 Server 上執行自動化腳本，該腳本將會操控本預約平台之 P4 Switch，型號為 Wedge100BF-32X，自動化腳本測試時間如表1，OS 重新安裝的時間約為500秒，佈署 SDE 的時間約為2800秒，將會利用安裝所需時間的時間差，來安排預約時間重置使用者環境之空檔。雖然 ONL Reinstall 加上 Deploy SDE 的時間可以在1小時內完成，但我們會初步以3小時重整時間為單位，此重整的時間單位會和實際的運作情形作調整，亦即第1小時允許使用者進行系統備份、第2小時與第3小時保留安裝 ONL 與佈署 SDE。萬一出現異常狀況，將有更多緩衝時間來重新進行。故以3小時為重整單位時間，確實為較妥適之作法。

6. 結論

在透過測試自動化佈署腳本後，可使本 P4實驗預約平台更邁向完整成品，可以建構一個國內

遠距之大型可程式化交換網路，不但能夠增加可程式化網路研究之能量，更能增加與國內外學研單位合作進行可程式化交換網路之合作機會。若實驗平台完成後，國內學研單位若有針對可程式化交換網路有所研究，便可向本中心申請租用本服務，讓各項理論能夠在實際的場域中得到驗證以及測試。未來，若學研單位使用者若能透過使用本中心可程式化交換器而有論文產出，亦能增加本中心相關論文數量。

參考文獻

[1] P4_16 Portable Switch Architecture (PSA). [Online]. Available: https://p4.org/p4-spec/docs/PSA.html

[2] Intel® Tofino™ 系列可程式化乙太網路交換器 ASIC. [Online]. Available: https://www.intel.com.tw/content/www/tw/zh/products/network-io/programmable-ethernet-switch/tofino-series.html

[3] BEHAVIORAL MODEL (bmv2) [Online]. Available: https://github.com/p4lang/behavioral-model

[4] Open Networking Foundation - Next-Gen SDN Tutorial - Session 1: P4 and P4Runtime Basics. [Online]. Available: https://opennetworking.org/wp-content/uploads/2019/10/NG-SDN-Tutorial-Session-1.pdf

[5] P4 – Language Consortium. [Online]. Available: https://p4.org/

[6] Open Networking Foundation. [Online]. Available: https://opennetworking.org/

[7] E. Haleplidis, K. Pentikousis, S. Denazis, H. Salim, D. Meyer, and O. Koufopavlou. Software-Dened Networking (SDN): Layers and Architecture Terminology. RFC 7426, Internet Engineering Task Force (IETF), January 2015.

[8] v1model Architecture Definition. [Online]. Available: https://github.com/p4lang/p4c/blob/main/p4include/v1model.p4

[9] Open vSwitch. [Online]. Available: https://www.openvswitch.org/

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford,

S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks.