P4 程式碼基於 Tofino model 與 V1model 編寫之差異性探討

胡乃元 財團法人國家實驗研究院國家高速網路與計算中心 2103081@narlabs.org.tw

黃文源 財團法人國家實驗研究院國家高速網路與計算中心 wunyuan@narlabs.org.tw

周大源 財團法人國家實驗研究院國家高速網路與計算中心 1203053@narlabs.org.tw

曾惠敏 財團法人國家實驗研究院國家高速網路與計算中心 0303118@narlabs.org.tw

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 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，所以移植上實體機器會有相當多的問題等待克服，在本論文將會簡單研討 V1model 與 Tofino 在開發過程中所遇到的問題以及差異點，相關分析可提供 P4程式碼於跨架構撰寫上之參考，並將 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

I. 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，利用集中的方式來控制網絡，分為資料層、控制層和應用層，並且 允許管理人員在不改變任何硬體的情況下規劃網絡以及控制流量，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 端口 6633 與設備溝通。 設備可以控制底層設備，以便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。

本次論文在章節 II 講解了 P4相關的背景知識，章節 III 講解了在 V1model 以及 Tofino 流水線架構上的差異性，章節 IV 整理了部份不同流水線之間程式碼的開發差異性，章節 V 則是結論，主要目的是為了能了解 V1model 與 Tofino 架構上的開發差異並記錄下來，方便在需要移植 V1model 至 Tofino 的設備時修改相關程式碼使能夠執行。

II. 背景知識

A. 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 等等上支援，不過目前仍以 Tofino 晶片為大宗，可隨時更改交換規則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，透過 P4以及強大的 Tofino 晶片，為了實現特定的網路封包行為，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。

B. P4 WorkFlow

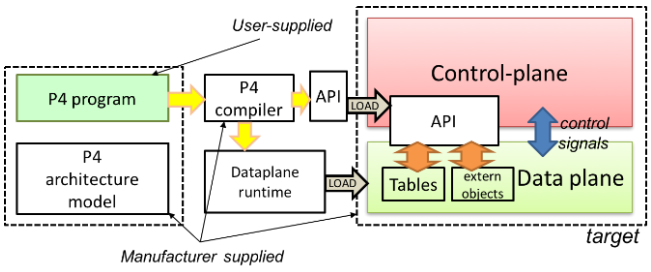

圖1. P4 workflow[5]

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。

C. 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部屬，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。

D. 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。

III. 架構差異性

本節介紹在不同 pipeline 架構上的差異性。

A. V1model 及 TNA 的 Pipeline 差異

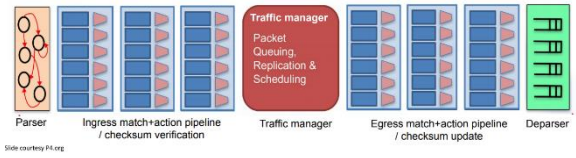

圖 2. V1model pipeline[4]

圖2說明了 V1model 的 pipeline 運作方式，Ingress 以及 Egress 代表了封包的進入以及離開，Ingress 會先利用 Parser 解析封包，將解析出來的資料進行儲存，Match-Action Pipeline 的部分則根據儲存的資料派往對應的 Action table 進行修改或增加資料，最後則利用 Deparser 逆解析數據到封包中。

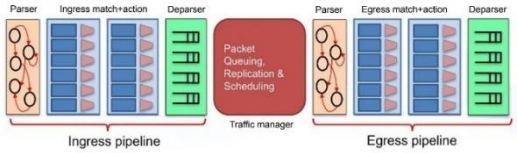

圖 3. Tofino Native Architecture (TNA) pipeline[4]

Tofino 的流水線被稱為 Tofino Native Architec-ture (TNA) ，是基於 PSA（Portable Switch Architecture）[1]所定義出來的，所謂 PSA 則是像 C 語言的 library，是專門給 P4語言的 library，定義出這些格式是為了能夠支援更多的 Target，例如：FPGA，ASIC 等等，則圖3說明了 TNA 的流水線，這時可發現 TNA 的 Ingress 和 Egress 流水線與 V1model 最大的不同為各自有他們獨有的 Parser 以及 Deparser，另外與原本 V1model 所使用的 P4Runtime 不同的是，Barefoot 利用 P4_16語言將 gRPC 接口對應到 Tofino 平台，這被稱為 BfRuntime。

B. Parser 及 Deparser 宣告上之差異

Bmv2的 Parser 及 Deparser 是全局共用的，這代表在整個程式編寫過程中只需要宣告一次。Tofino 的 Parser 及 Deparser 為 Ingress Egress 分開，這代表 Parser 到 Egress 階段需要再宣告一次，Deparser 到 Ingress 階段的尾端需要再宣告一次。

C. Metadata 宣告上之差異

在編寫程式上，BMv2的 Metadata 是全局共用，這代表透過 V1model 的 pipeline 編寫程式，過程中只需要宣告一次 Metadata。Tofino 的 Metadata 則為 Ingress 以及 Egress 是分開的，這代表到 Egress 階段需要再宣告一次 Metadata。

IV. 程式碼差異性

在介紹完架構差異後，此章節將介紹本次論文所整理到在不同 pipeline 架構之間，程式碼的差異性。

A. Include

編寫過程中使用的 library 不一樣，在撰寫 BMv2所使用的版本會使用#include <v1model.p4> ，但在 Tofino 中會替換成 SDE 中的#include <tna.p4>。

B. Metadata

在 V1model 中呼叫一般的全局性基本參數是利用 standard_metadata_t，在程式碼的寫法為：

```
inout standard_metadata_t standard_metadata
```

則在 Tofino 裡面是利用 intrinsic_metadata_t，在程式碼的寫法為：

```
out ingress_intrinsic_metadata_t ig_intr_md
```

intrinsic_metadata 在 Tofino 則有分成 ingress 以及 egress 的部分，例如：ingress_intrinsic_metadata 以及 egress_intrinsic_metadata，然後會有部分的

intrinsic_metadata 種類會在離開 Parser 的階段時候產生 intrinsic_metadata_from_parser 和 intrinsic_metadata_for_deparser 以及 intrinsic_metadata_for_tm，這些 metadata 資料也一樣會有分成 ingress 以及 egress 的部分，因為這三種 Metadata 是自動產生的，所以在脫離 Parser 之後可以不用經過 out，可以直接拿 in 或 inout 做資料上的應用。

C. Parser

BMv2 沒有 Egress Parser 的部分，所以開發 Tofino 程式時需要補上 Egress Parser 的部分。

在自訂義的 metadata 之中，進入 parser 的資料走向在兩種 pipeline 中有些許不一樣，在 BMv2中是使用 inout 走向，但在 Tofino 之中需要利用 out 進行資料的初始化。

BMv2範例例如：

```
parser prs_main(packet_in packet,
        out headers hdr,
        inout metadata meta,
        inout standard_metadata_t standard_metadata)
```

Tofino 範例例如：

```
parser prs_main(packet_in pkt,
        out headers hdr,
        out metadata_t md,
        out ingress_intrinsic_metadata_t ig_intr_md)
```

如果直接使用 inout 則會讓 p4c 無法編譯，另外在 BMv2之中的 standard_metadata_t 以及 Tofino 所使用的 ingress_intrinsic_metadata_t 也是有相同情況。

D. Match action

V1model 中 standard_metadata 的 egress_spec 通常在會被使用在指定封包要轉發至甚麼 port，但是在 Tofino 是直接調用 egress_port 參數，在 V1model 裡面的 egress_port 並不是用來讓 control flow 控制封包要轉發至指定 port 的，而只是被寫入真實要出去的 port 資訊。

在進行 muticast 的開發需要注意，V1model 的 muticast group 函數是使用 mcast_grp，則在 Tofino 是使用在 Ingress Intrinsic Metadata for TM 中的 mcast_grp_a 以及 mcast_grp_b。

E. Deparser

Deparser 階段是 P4用來將解析後的封包打包的步驟，在章節 III-*A* 中有提到在 BMv2之中並沒有 Ingress Deparser 的區塊，故在撰寫 Tofino 程式碼或是 BMv2程式碼的時候必須針對相關的區塊作不同的設定。

F. Checksum

由 Pipeline 可得知，BMv2在 Parser 與 Ingress 之間以及 Egress 與 Deparser 之間各自有獨立的 Checksum 區塊，然後在會有另一個獨立的區塊，然而在 TNA 架構裡面並不是獨立的，Tofino 的 Checksum 功能整合至 Ingress parser 以及在 Ingress deparser 進行 checksum 的更新。

在 Ingress parser 是將宣告函數 Checksum() ipv4_checksum;放在 state start 前，checksum 更新則是放在 Ingress deparser 的區塊，位於進行 pkt.emit 前的部分。

G. Ipv4 Forwarding

在進行 BMv2的開發時，以下範例常常會使用到：

```
action act_ipv4_send(mac_addr_t dst_mac_addr,
egress_spec_t egress_port) {

hdr.ethernet.src_mac_addr = hdr.ethernet.dst_mac_addr;

hdr.ethernet.dst_mac_addr = dst_mac_addr;

standard_metadata.egress_spec = egress_port;

hdr.ipv4.ttl = hdr.ipv4.ttl -1;

}
```

但我們發現在實際的 Tofino 設備上除了有提到 egress_spec 的函數不同之外，在傳輸封包時，

```
hdr.ethernet.src_mac_addr = hdr.ethernet.dst_mac_addr;
```

會與以下程式碼

```
hdr.ethernet.dst_mac_addr = dst_mac_addr;
```

造成衝突，在 Tofino 設備上不允許封包同時擁有相同的 src_mac_addr 以及 dst_mac_addr，所以開發時設置成 src_mac_addr 利用 data 的方式寫入進去以避免這類情形發生。

V. 結論

在開發 Tofino 專用的 P4程式會需要比開發 bmv2更多的知識，他們擁有各自不同的開發參數，Tofino 擁有更嚴苛的程式碼限制以及對系統資源上的分配，讓 ASIC 晶片有更大的效能可以處理經過 p4程式的封包，基於開發 Tofino 專屬的 p4程式需要簽訂 NDA 條款才能利用 SDE 以及虛擬 Tofino model 又讓開發門檻更高，透過記錄比較 BMv2以及開發 Tofino 上的差異可以使同儕間開發更加順利，更加順手應用 Tofino 設備以應用在台灣高品質學術研究網路 TWAREN 上。

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.

ACM SIGCOMM Computer Communication Review, 38(2):69{74, April 2008