

P4 網路遙測數據整合系統的設計

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
{wunyuanyuan, 1203053, 0303118, tlliu}@narlabs.org.tw

摘要

本論文提出一個針對 P4 INT 網路數據整合的系統設計，其目的在於獲取並整合不同組織間 P4 INT 所記錄的網路效能數據，並且展示於 Web UI 之上。P4 INT 的技術能夠於封包之中嵌入網路效能數據，然後必須實作或是尋找蒐集 INT 數據的應用程式來取得網路效能數據，並存入資料庫之中，然而眾多建置 P4 INT 網路的組織單位，其所獲取的 INT 數據的方法不盡相同，因此我們設計一套系統，利用集中式的做法向不同組織的系統獲取 INT 效能數據並整合分析，最後再以圖表和圖形化的方式呈現於 Web UI 之上，如此一來，便能有益於跨各單位組織間的 P4 網路管理。

關鍵詞：ONOS、P4、INT、In-band Network Telemetry

1. 前言

近十年網際網路的蓬勃發展，許多地網路技術也相繼被提出，像是大數據傳輸與雲端計算，而這些技術也與其他電腦領域的技術結合，使原本的技術大幅提升，例如：AI 結合大數據傳輸與雲端計算，讓研究者能利用更多的頻寬傳送資料以及更好的運算來減少運算與傳輸時間。

當然新網路技術的產生也讓許多便利的應用相繼的出現，而這些應用也讓全世界的網路使用者遽增，故應用服務供應商們也開始建置新的資料中心 (Data center) 來服務遽增的使用者，避免因為設備不足流失許多的客戶。

當建置資料中心時，往往會遇到 QoS、IP 缺少與虛擬化等的傳統網路限制，而為了解決這些問題，廠商們勢必要添購更多的硬體設備，如此一來便大幅的增加成本，故廠商為了降低成本便會尋求新型態網路架構的解決方案，其中最為人熟知的就是軟體定義網路 (Software Defined Network; SDN)[1] 此項技術也被大型公司 Google 與 Amazon 所採用。

SDN 的出現帶來新的網路變革與思維，其修改了傳統網路的控制模式，將網路設備的資料層和控制層分開。SDN 的架構圖如圖 1 所示，控制層包含著控制平台，其負責決定封包如何處理的工作，常見的控制平台有 ONOS (Open Network Operating System) [2][3]、OpenDaylight (ODL)[4] 與 Ryu[5]。資料層包含著相容 SDN 的交換器，其工作是根據

控制層的決策來把封包送到目的地。控制平台跟資料平台之間的聯繫通常採用相容 SDN 的協定例如：OpenFlow，透過這個協定，資料平台就可以針對控制平台的決策結果來設定其內部的 Flow Table，之後資料平台處理封包就根據 Flow Table 的紀錄來處理。藉由 SDN 的可程式化開放架構的特性，使用者就可以於獨立的控制平台上新增與開發所需功能以滿足客製化網路需求、提高頻寬使用率和減少成本[6]等。

雖然 SDN 可以讓使用者客制化許多的應用程式，但是資料平台上對於封包的處理仍然尚未完全開放，因此 ONF 提出了 P4 language[7][8]，一種可以讓使用者於資料平台上自己撰寫如何處理封包的架構。P4.org 除了制訂 P4 的架構外，也制定了 P4Runtime 的協定[9]，讓使用者可以實作或使用 P4Runtime 的控制平台來管理 P4 交換器。P4 和 ONOS 均為 ONF 的專案，因此 ONF 也於 ONOS 平台上增加 P4 的支援，其中也包含了 P4Runtime，所以我們的研究採用 ONOS 為主。

P4.org 除了 P4 架構與協定的制定外，也發展了名為 In-band Network Telemetry (INT)[10] 的應用技術，此技術可以利用 P4 交換器將網路效能的資訊嵌入至封包標頭中並傳送至 next hop，等送至終點時便可擷取封包標頭檔的 INT 欄位來分析網路效能，因此使用者能夠利用支援 INT 封包的 P4 交換器，並搭配 P4Runtime 的控制平台，例如：ONOS，便能構建一個私人的並擁有封包遙測技術的網路監控環境。

因為 INT 能夠提供網路封包的各種資訊，因此也有很多研究人員在 P4 上實作了 INT 技術與監控的應用程式，然而實作出來的應用程式彼此之間的資訊是不相通，因此當眾多組織單位間，他們的 P4 INT 的網路平台要合作組成一個大型網路平台時，可能無法取得彼此間的 INT 資訊，例如本中心預計在 TWAREN 上使用 P4 INT 測試平台與國外西北大學 iCAIR 的 P4 INT 網路連線，然而兩邊的 P4 INT 擷取的資訊可能會有差異，故本論文將以我們的環境來設計一個系統以解決這個問題。

本篇論文中，首先會介紹 P4 與 INT 背景知識，然後會分析幾個針對 INT 技術實作出來的應用程式，最後再提出我們所規劃的 P4 INT 可視化監控系統。

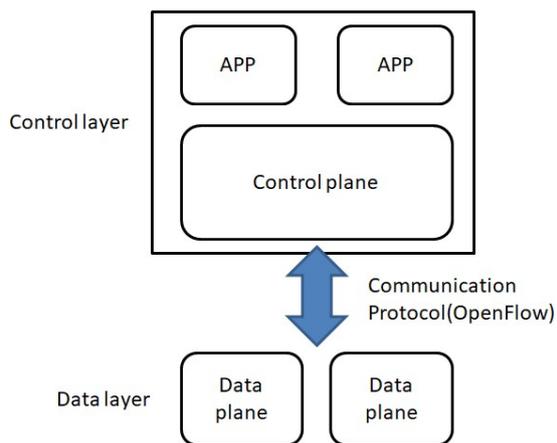


圖 1 SDN 網路架構

2. 相關知識

這一章中將會描述 P4 的基本概念以及什麼是 INT，藉由這些基本概念，便能夠知道 INT 技術是如何傳遞網路效能數據。

2.1 P4

P4 交換器 (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。

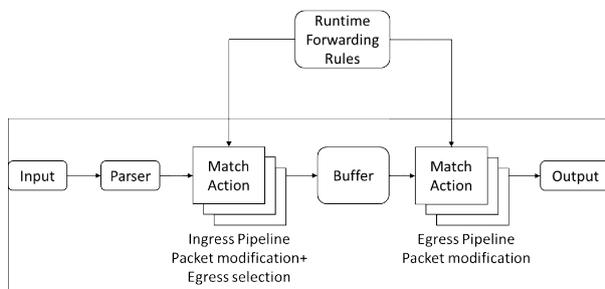


圖 1 P4 Pipeline

圖 1 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。

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。

由於 Whitebox 交換器盛行，針對實體網路，

目前有其他的可程式化交換器平台，例如 ONF Stratum[11]與 Microsoft Sonic[12]，皆為開放原始碼軟體，並支援提供 P4 相關支援，提供業界實作硬體交換器的核心元件。

2.2 In-band Network Telemetry

INT 是一種以資料平台為主，收集與回報資料平台網路狀態的框架。這種架構不需要控制平台介入。相較傳統的網管技術需要額外指令來進行網路狀態監控，INT 是將偵測到的效能數據加入資料封包標頭的 metadata 欄位中。這樣不會額外造成網路的負擔。這樣的好處就是：由於網路狀態資訊是附在資料封包內，因此當網路封包量愈大、網路狀態更新的頻率愈高。理論上，INT 可依最高達到 line rate 的速率來進行網路狀態偵測 [13]。

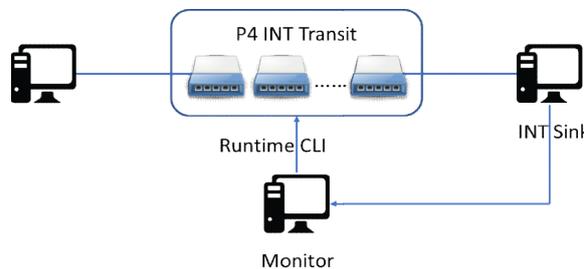


圖 2 典型的 P4 INT 框架

在 P4 交換器中，也有針對 INT 進行實作，詳細規格如 [10]。網路裝置在收到這些封包後，就會把偵測指令所指定的資訊寫入資料封包中。圖 2 為 P4 INT 框架示意圖。INT 框架包含 INT Source、INT Sink，以及 INT Transit。一般在 INT Source 中會建構一個包含 INT 偵測指令的標頭。而 INT Transit 即為在資料傳輸路徑中每一部支援 INT 的裝置。當 INT Transit 收到資料封包時，會將 INT 偵測指令所對應的狀態資訊寫入資料封包中。而前述的資訊會收集到 INT Sink，並會傳給 Monitor。Monitor 所收集到的網路效能資訊可以直接送至管理人員指定的地方進行分析。

在使用 P4 INT 時可以自行定義並蒐集任何交換機內部的資訊，目前 P4 INT 的官方規範中提供了幾種可以使用的 Metadata，其中大多數可以直接透過 P4 定義的 Standard Metadata 直接從設備中取得：

- Switch identifier: 交換機的唯一 ID。
- Ingress port ID: 接收 INT 封包的 port ID。
- Ingress timestamp: 設備接收到 INT 封包時的本地時間戳記。
- Egress port ID: INT 送出封包的 port ID。
- Hop latency: INT 封包在設備中傳輸的延遲。
- Egress port TX Link utilization: 送出 INT 封包的 port 當前的使用率。

- Queue occupancy：INT 封包在設備中傳送時觀察到 Queue 中已儲存的流量。
- Queue congestion status：當前 Queue 的壅塞狀態。

3. 問題與系統設計

在第一節中，我們將解釋傳統網路監控技術上的問題，第二節中，將說明我們於 INT 應用程式的研究，並比較各自的優缺點，而最後兩節將說明我們所設計的系統。

3.1 傳統網路監控技術的問題

在傳統上的網路管理技術以 SNMP/RMON、NetFlow/sFlow、protocol analyzers，以及 network traffic probes 等等技術為主[14]。SNMP/RMON 相當簡便，但能夠監控的範圍太少，且無法針對歷史性的 flow 資訊進行監控。而 NetFlow/sFlow 工具則會消耗大量網路裝置的資源，進而降低網路裝置於轉送資料的效能。而 traffic probe 工具則需要部署大量裝置在待監測的網路之中。Protocol analyzers 僅能擷取小部份資料，卻無法進行長期的觀測與歷史資料分析。

另外於軟體定義網路中，控制平台與資料平台是區隔開的。因此，網路管理者可以透過控制平台來監控網路的狀態。然而，因為軟體定義網路與傳統網路的依存度相當高，所以在軟體定義網路上的監控管理技術往往也是傾向採用傳統的網管監控技術，因此在今日大規模的實務應用上仍然不甚理想。

綜上所述，傳統的網路監控技術幾乎都需要安裝獨立的硬體，或者進行額外的軟體設定。這樣會使得網路監控的成本增加、門檻變高。

3.2 INT 應用程式研究

在支援 INT 的 P4 資料平台上，封包會嵌入 INT 標頭，而這些 INT 標頭會包含著 INT Metadata，使用者可蒐集 INT Metadata 再結合第三方或自己實作的視覺化網路監控軟體來分析與觀察網路情況。以下是 INT 搭配可視化監控軟體的相關文獻與解決方案的析。

[15]是 P4.org 提出的一個 INT 實作，此實作提供了 INT over Vxlan 程式碼、INT 分析工具、監控系統與使用者介面，因此使用者可以參考或是利用這些工具來打造自己的 INT 網路環境，然而實作的架構並未搭配任何資料庫，因此其可視化界面只能顯示當前的網路狀態。

[16]是 ONOS 之上實作的 INT service 以及與其搭配的 INTCollector 工具，INTCollector 能夠蒐集與分析 INT 封包，且也會將結果儲存至資料庫之中

(支援 Influxdb 和 Prometheus)。INT 與 ONOS 結合，除了減少 ONOS 使用者跨足 INT 難度外，也提供了一個想學習與使用 INT 的入門者一個較簡易的管道。然而 INTCollector 無提供可視化界面，故使用者必須額外使用第三方可視化軟體。

一般情況下，INT 封包會持續的轉送至 INT 監控系統之上，但是這種情況往往會讓 INT 監控系統蒐集許多不必要或是重複的 INT 資料，導致效能上的浪費，因此[13]的作者結合 ONOS 的 INT service 提出了一個 INT event detection 的框架以及一些 INT event detection 的演算法，讓滿足特定條件的 INT 發送至 INT 監控系統，以減少 INT 監控系統的負荷。第二，開發出了一套比 INTCollector 效能高，並且支援 kafka 的 INT 監控系統，並結合 ELK，使系統能擁有可視化的網路監控介面。然而從文獻中來看，此系統僅將數據圖表化，並未有圖形化 Flow Path 或拓樸檢視等相關功能。

[17]是由 Liberouter、Flowmon 與 NETCOPE 共同合作的 P4 INT demo，他們在此 demo 中，開發了一個用來蒐集 INT 資料的元件，此元件除了蒐集 INT 資料外，也會將資料轉成 NetFlow 格式，因此成功將 INT 與現有的 Flowmon 視覺化網路監控軟體產品整合，可作為為 INT 打造一個可視化監控軟體的參考案例，然而此軟體為收費產品，故需得額外考慮取得的成本。

[18]中設計了一個 ML-INT 的系統，此系統中，包含著他們開發的 Optical performance monitor、特殊設計的 P4 Pipeline 以及高效能的 data analyzer，透過這些功能，此系統成功地將 INT 與 Optical 網路整合，使 Optical 網路相關數據能夠被解析與保存。然而此文獻同樣也著重於數據的圖表化，並未發現圖形化 topology 與 flow 檢視等相關功能，於了解整體網路 topology 分布與 flow 狀況上較為薄弱。

[19]為 Barefoot 所開發的支援 INT 監控的產品，由於它們有開發 P4 的晶片，在 P4 的涉略上較為深入，他們開發的 Deep Insight 產品在 P4.org 所制定的 INT report 格式上有較佳的相容性。此產品除了基本的圖表分析外，也包含著 Topology 檢視與流量相關聯的功能，透過這些功能，使用者更加能夠了解整體的網路狀況。

表1是上述文獻與解決方案的比較表格，大部分的方法都採用資料庫來儲存數據，然後以圖表的方式呈現 INT 資料的分析結果，僅有少部分支援圖形化 topology 與 flow 的檢視。如果能夠完整的支援這三種功能，對於大型網路的管理人員而言，能透過讀取多個網域的資料庫，與搭配各個網域的分析資料和圖形化的 topology 來了解與管理網路，例如：管理人員可以從監控系統看到大型網路的圖形化 topology、flow 資訊或是各個網域的健康度等資訊，從而快速的發現網路問題與解決問題。

表1: 於資料庫、分析圖表與 topology 檢視功能支援上的比較

	Database	資料分析與圖表	圖形化 Flow Path 檢視
P4.org INT Implementation	X	X	O
ONOS INT Service	O	X	X
Event Detection INT Monitor	O	O	X
Flowmon Solution	O	O	X
ML-INT Data Analyzer	O	O	X
Barefoot Deep Insight	-	O	O

3.3 P4 INT 可視化監控系統架構

在前一章中描述了各個應用程式的優缺點，由於大部分文獻與解決方案通常採用不同的資料庫，且大部分視覺化監控軟體也不支援圖形化 topology 與 flow 檢視，故我們設計一個能存取不同資料庫與支援圖形化 topology 與 flow 檢視的應用程式，圖3是我們設計的軟體架構，為了因應資料庫不同的問題，因此我們軟體的最底層為 DB Driver Layer，裡面包含了一些需要實作的 DB Driver，透過這些 Driver，使我們設計的軟體能夠存取不同的資料庫，然後把資料送往 Analyze Layer。Analyze Layer 負責處理與分析從資料庫得到的資訊，並根據 UI Layer 提供的功能整理成特定的資料格式，然後於 UI Layer 將資料視覺化，例如: nodes 與 links 的資料可供 Web UI 產生圖形化 topology，以及像 delay time 等一些數據資料，供 Web UI 產生圖表。

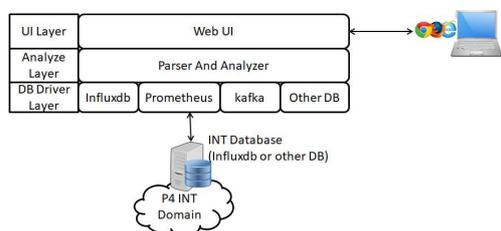


圖3 可視化監控系統架構

3.4 系統前端介面與後端資料處理的設計

在此小節中，將說明我們針對系統的前端後端的設計，我們的前端會設計一個能查詢資料、圖表顯示與 topology display 的介面，以滿足我們的 P4

在 TWAREN 上監控的需求。

圖4為簡單的設計畫面，因為我們需要圖表的資訊顯示，所以設計的介面將會結合 Grafana[20]，讓 Grafana 做為圖表展示的介面，而主要的網頁介面會分為三大區塊，分別是資訊輸入框架1、2與資訊顯示框架，資訊輸入框架1主要是取得資料庫資料以及在 Grafana 新增資料來源，因此可能的輸入欄位有 DB 的選擇、DB IP 與 Port、Grafana IP、Port 與 Grafana 資料來源名稱等。資訊輸入框架2是以 Grafana 資料查詢與 Grafana 表格新增為主，可能的欄位有 Grafana IP、Port、資料來源名稱與 Dashboard Title 等。資訊顯示框架則是會顯示資料庫回傳的資料和 Grafana 回傳的資料與結果。



圖4 前端 Web UI 設計

後端部分為 Analyze Layer 與 DB Driver Layer，圖5為這兩層的架構設計，後端的資料發送與接收均採用 JSON 的格式，運作流程如下：

- Step 1: Web Server 收到 request，將資料送至 Data Parser。
- Step 2: Data Parser 根據接收的資料建構所屬的資料結構，接著往下至 DB Selector。
- Step 3: DB Selector 根據這些資料結構來確認呼叫所需的 DB Driver。
- Step 4: 從資料庫得到的資料會送至 Data Classification，將資料分成一般資料與拓樸資料，然後將分類好的資料送至 Topology Creator 和 Normal Data Parser。
- Step 5: Topology Creator 會根據 topology 資料建構 Node 與 Link 等相關的資料結構。Normal Data Parser 則是直接將接收到的資料取出並根據資料型態建構適合的資料結構。
- Step 6: Data Integrator 會把 topology 的資料結構和一般資料的資料結構整合，並送至 JSON Formatter 轉成 JSON 格式的資料回傳給前端。

上述是一般資料庫查詢的流程，而我們設計的介面有搭配 Grafana 來滿足圖表和 topology display 的功能，故我們後端也設計了自動建構 Grafana 圖

表與 topology display 的功能，其流程如下：

- Step 1~2: 同一般的資料庫查詢流程。
- Step 3: DB Selector 根據接收到的所有資料結構呼叫 Grafana Creator。
- Step 4: Grafana Creator，將所有的資料結構轉成 Grafana Server 能識別的內容，並將這些內容以 JSON 格式送至 Grafana Server 建立各種圖表與 topology graph。
- Step 5: Grafana Server 回傳後的資料會送至 Data Classification，然後將資料轉送至 Normal Data Parser。
- Step 6: Normal Data Parser 則是直接將接收到的資料取出並根據資料型態建構適合的資料結構，並交給最後的 Data Integrator 與 JSON Formatter 做後續處理。

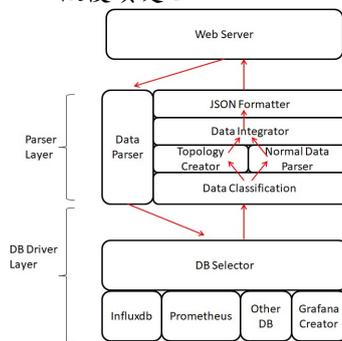


圖5 Analyze Layer 與 DB Driver Layer 的設計

根據上述的設計構想，我們就能確保此系統能滿足在 TWAREN 上監控 P4 網路的需求，使負責的管理人員能夠藉由此設計的系統獲取所需之資訊順利的維護 P4 網路。

4. 系統的實驗測試

根據前述的設計，我們實作了一個簡單的系統，因此這一章節將利用此系統來驗證設計的可行性。這個系統僅實作 Influxdb [21] 的讀取，而實驗的 INT P4 環境方面，將使用 ONOS、bmv2[22] 與 BPFCollector 建置，建置的教學可以參考 ONOS 的 wiki 頁面[16]。

此實驗環境的 topology 如圖6所示，圖中的設備有4個 P4 交換器，分別是 s11、s12、s21 與 s22，2 個 host，它們是 h1 與 h2，以及一個 ONOS 伺服器，各設備的連接方式是 ONOS 和每個 P4 交換器連線，並採用 P4 Runtime 協定來通訊，h1 接至 s11，h2 與 s12 連接，而交換器之間的連接就如圖來互連。當佈置好環境後，就讓 h1 將流量送至 h2，使 INT 的資料被 BPFCollector 蒐集並儲存至 Influxdb 之中，接下來必須安裝 Grafana 並啟動伺服器以便讓 Grafana 伺服器能夠與我們實作的簡易網頁通訊。

上述環境佈置完成後，就啟動瀏覽器進入實作好的網頁介面，圖7(a)是一開始的網頁介面，圖中為我們前面設計的輸入框架1，將資料送出後會進

入圖7(b)的畫面，資訊框架顯示的內容為資料庫回傳的資料。圖7(c)則是選擇 query dashboard 功能後將輸入框架2的資訊送出後的結果畫面，最後圖7(d)就是選擇 create or update dashboard 功能時，將資訊送出後的結果，顯示的結果中會出現連結點，利用此連結點就可以開啟建立好的 Grafana dashboard，以便檢視圖表與圖形化的 flow path，如圖7(e)所示。

從這個實驗中可以知道前述的設計概念的確可以順利的取得 INT 資料，並且也能以圖表和圖形化的方式來展示，如此一來只需將我們的設計概念完全的實作出來將可以滿足於 TWAREN 上管理與監控 P4 的基本需求。

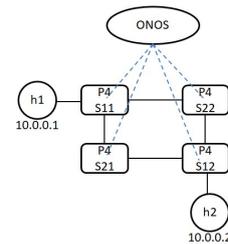


圖6 Analyze Layer 與 DB Driver Layer 的設計

(a)

(b)

(c)

NAR Labs
National Center for High-performance Computing

Database IP: 192.168.56.101 Port: 8086 Query Measurement

Grafana IP: localhost Grafana Port: 3000 Grafana DB Name: influxDB

CreateDataSources Create a DB sources Don't Create a DB sources

show Database information query dashboard create or update dashboard

Grafana Information: Grafana IP: localhost Grafana Port: 3000 Dashboard Title: wunyuans DB Source Name: influxDB

Great DL_PanelJoin

Dashboard Id	uid	title	version	hyperlink
9	9d56AnHGK	wunyuans	3	link

(d)



(e)

圖7 簡易 INT 系統實驗結果

5. 結論

因為我們計畫於 TWAREN 上建置 P4 網路平台並與其他組織單位連線合作，因此需要考量建置後的維運，而 P4 上有著 INT 的技術，讓 P4 的使用者可以利用此技術取得網路效能數據來管理維護網路，然而每個組織單位間雖然使用著 INT 技術，但是取得 INT 網路效能數據的技術可能不盡相同，而為了能順利的與各個單位合作來獲得所有的 INT 網路效能數據，因此便提出本篇的設計以滿足我們的需求。

未來方面，我們將再強化我們的設計，增加更多對管理維運更友善的功能，並且以前述用來實驗的系統為基底，徹底的實現我們設計的所有功能，而未來實作出來的系統便能有益於我們以及合作單位於 P4 網路上的管理。

參考文獻

[1] Software Defined Network, <https://www.opennetworking.org/sdn-definition/>

[2] P. Berde, M. Gerola, J.Hart, et al. ONOS: Towards An Open, Distributed SDN OS. Proceedings of the 3rd Workshop on Hot topics in Software Defined Networking. ACM, 2014: 1-6.

[3] ONOS white paper, <http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf>

[4] OpenDaylight, <https://www.opendaylight.org/>

[5] Ryu, <https://osrg.github.io/ryu/>

[6] S. Ortiz, "Software-defined networking: On the verge of a breakthrough?" Computer, vol. 46, no. 7, pp. 10–12, Jul. 2013.

[7] P4 Language Consortium, <https://p4.org/>

[8] 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.

[9] P4 Runtime, <https://p4.org/p4runtime/spec/v1.2.0/P4Runtime-Spec.html>

[10] In-band Network Telemetry, https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_1.pdf

[11] ONF Stratum, https://www.opennetworking.org/stratum/?utm_referrer=https://www.sdnlab.com/23020.html

[12] Microsoft Sonic, <https://github.com/Azure/SONiC>

[13] J. Vestin, A. Kassler, D. Bhamare, K. Grinnemo, J. Andersson, and G. Pongracz, "Programmable Event Detection for In-Band Network Telemetry," Proceedings of the IEEE CloudNet 2019, 2019.

[14] J. Geng, J. Yan, Y. Ren, Y. Zhang, "Design and Implementation of Network Monitoring and Scheduling Architecture Based on P4," Proceedings of the 2nd International Conference on Computer Science and Application Engineering, No. 182, Oct. 2018.

[15] P4 INT Implementation, <https://github.com/p4lang/p4factory/tree/master/apps/int>

[16] In-band Network Telemetry (INT) with ONOS and P4, <https://wiki.onosproject.org/display/ONOS/In-band+Network+Telemetry+%28INT%29+with+ONOS+and+P4>

[17] InBand Network Telemetry with P4 and FPGA at 100 Gbps, <https://www.liberouter.org/wp-content/uploads/2013/02/2017-06-07-DXDD.pdf>

[18] B. Niu, J. Kong, S. Tang, Y. Li, and Z. Zhu, "Visualize Your IP-Over-Optical Network in Realtime: A P4-Based Flexible Multilayer In-Band Network Telemetry (ML-INT) System," IEEE Access, pp. 82413 - 82423, 2019.

[19] Barefoot Deep Insight, <https://www.barefootnetworks.com/products/brief-deep-insight/>

[20] Grafana, <https://grafana.com/grafana/>

[21] Influxdb, <https://www.influxdata.com/>

[22] behavioral-model(bmv2), <https://github.com/p4lang/behavioral-model>