

SDN 標準與測試

王辰佑 林盈達

國立交通大學資訊工程系

Email: arthur81542.cs03g@nctu.edu.tw; ymlin@cs.nctu.edu.tw

September 15, 2014

摘要

在現今的網路世界中，充斥著眾多設備與協定，但沒有統一的管理以及設定方式，因而衍生出 SDN 網路，這新型的網路架構將 control plane 分離並集中化，使網路設備只進行封包傳遞，並簡化設備管理、設定流程。本篇針對 SDN 網路標準 OpenFlow 之測試報告進行分析其未通過測試原因，並使用 Open vSwitch 符合性 (Conformance) 測試的結果做為分析目標。該測試結果共進行 112 項功能測試，Open vSwitch 則有 21 項不支援功能，當中有 7 項的實作與測試 controller 有差異，實作差異中有 6 項目為 OpenFlow keyword 不同導致 Open vSwitch 無法辨識，另一項目其運算結果與測試 controller 不相符，該項目中反而是測試方判斷錯誤。依此結果，OpenFlow 產品會因未支援功能、實作不符合規範、實作錯誤等原因，導致測試項目失敗，進而在產品間搭配使用時，部分功能無法正常運作。

關鍵字：SDN、OpenFlow、測試、符合性

1. 簡介

Software-Defined Networking[1] (SDN) 是現今新型的網路型態，其概念是將網路設備中的 control plane 與 data plane 抽離，並將網路設備中控制、邏輯等處理集中化，使網路設備處理封包流程簡化並提供虛擬化的概念。現今網路為解決使用者以及廠商需求，因而衍生許多獨立協定，卻沒有提供統一的管理方式，導致網路組成更加複雜，在此以設定新網路設備舉例：為了配合現有的網路，通常需要調整交換器、路由器和防火牆等設備，並加入 VLANs、quality of services (QoS) 等多項設定，而這些設定必需與現有網路設定一致，設備才可正常運作。此例中，因為設定軟體皆為廠商提供並整合至網路設備中，且廠商間設定方法又有差異性，因此，只能逐一進行設定，使得整個過程複雜且難以控管；而 SDN 針對網路架構的改變，所帶來的是可程式化、自動化等特性，並能以擴充、調整功能來解決網路需求，相對於傳統網路更是具有快速佈署、中央控管及動態配置等優勢，如表 1 所示，不論是管理層面或是應用層面，SDN 網路皆比傳統網路更加適合現今網路需求。

SDN 網路因採取分離 control plane 與 data plane 的作法，使得網路處理過程中增加 control plane 與 data plane 溝通的過程，在實作中勢必需要一個統一且開源的規範，其規範即為 SDN 架構下 controller 與網路設備間的溝通標準。

表 1：SDN 與傳統網路特性比較表

Class	SDN network	Traditional network
Management complexity	Low	High
Consistence policy	Easy	Hard
Service scale	Easy	Hard
Vendor dependence	Low	High

在接下來的章節中，本文將依序介紹 SDN 標準、OpenFlow 測試，並對測試提出驗證測試項目的方法，接著進行未通過測試的項目分析，最後是本文的結論。

2. SDN 標準

SDN 架構下需有 control plane 與 data plane 間的溝通介面，其能處理現有協定資料封包，此外因應網路需求的成長，因此協定的擴充性及增值服務也需納入考量。如表 2 所示，當中 OpenFlow[2]、OpFlex[3] 是專為 SDN 所設計的協定，又以 OpenFlow 完成度最高，且已被廣泛的被應用於設備及 controller 當中，因此本文會以 OpenFlow 為主，而 Extensible Messaging and Presence Protocol[4] (XMPP)、Network Configuration Protocol[5] (NETCONF) 均為現有協定，其中 XMPP 且因其優良的擴展性，符合 SDN 部分需求，NETCONF 則針對 SDN 網路提出網路管理等概念；Network Service Header[6] (NSH)、Network Functions Virtualization[7] (NFV) 則是在 SDN 中是能以增值服務的型式存在，替 SDN 網路加入更多的可能性。

表 2：SDN 相關標準協定

Protocol	Pupose	Method	State	Vendor
OpenFlow	SDN standard	Imperative	Release	ONF
OpFlex	SDN standard	Declarative	Developing	Cisco
XMPP	Alternative SDN standard	Publish/subscribe	Existence	Juniper
NETCONF	Alternative SDN standard	Configure	Existence	IETF
NSH	Service virtualization	Create service plane	Developing	IETF
NFV	Similar to SDN, virtualization approach	Move network app to virtual container	Developing	ETSI

OpenFlow 介紹

OpenFlow[2] 做為 SDN 的標準，提供統一的溝通介面使 control plane 與 data plane 能正常溝通，且允許直接控管 data plane（如：forwarding）。目前 OpenFlow 發展至 1.4 版，其歷代版本功能差異如表 3 所示，其中 Flow table 為 control plane 控管 data plane 的方式，亦是 OpenFlow 核心的功能，該表負責儲存 data plane forwarding 或 routing 的條件、動作等資訊（即 flow entry），data plane 則依此表判斷、執行，進而實現直接控管 data plane 的效果。此外，從表 3 亦可看出 OpenFlow 不斷的新增功能，以增加協定本身的可用性及應用性。

表 3：OpenFlow 版本功能差異表

Version	Flow table	Multi table	Group table	Meter table	Eviction, vacancy
1.0	✓				
1.2	✓	✓	✓		
1.3	✓	✓	✓	✓	
1.4	✓	✓	✓	✓	✓

其中 Multi table 為 OpenFlow 1.2 版新增的功能，該功能加強 OpenFlow 處理封包的流程，如圖 1 所示，OpenFlow 將所要執行的動作儲存在 action set 並判斷是否需要進行下一張 flow table 的比對，整個流程完成後，再執行 action set 裡所有的動作，此舉使 OpenFlow 具有處理更複雜網路需求的能力；另外 Group table、Meter table 增加 OpenFlow 動作的功能性，其中 Group table 為將 action 集合整合至一張表並給予該集合一 ID，而 action 可指定該 ID 即可執行該 action 集合，此表可應用於 load balance、failover 等實際情境，Meter table 則提供流量控制的功能，使 OpenFlow 具有處理 QoS、DiffServ 的能力。最後 eviction、vacancy 是 OpenFlow 在 1.4 版為解決 flow table 空間不足所新增的功能，eviction 是在 flow table 沒有空間的情況下，新的 flow entry 會取代舊有且不重要的 flow entry，使整體架構能繼續以新的設定執行，而 vacancy 是在 flow table 空間接近滿的情況，網路設備需發出通知給予 controller，並由 controller 決定如何處理。

上述的所有功能，在 SDN 架構下皆需要 controller 進行輔助設定，而 controller、網路設備兩方皆需進行訊息交換，在 OpenFlow 規範下，大致分為 symmetric、asynchronous、controller-to-switch 三類訊息，當中 symmetric 為 controller 與網路設備間為建立溝通管道雙方互相交換資訊的訊息，asynchronous 為網路設備需要通知 controller 或是協助時所發出的訊息，controller-to-switch 則是提供 controller 對網路設備進行設定、控制等訊息。如表 4 所示，該表為 OpenFlow 所提供訊息之分類，在此舉例說明訊息用法：

假設一 OpenFlow switch 需與 OpenFlow controller 進行連線，且 switch flow table 內容預先設定為空並設定為若無適合 flow entry 則會請 controller 進行協助，同時假設 controller 已知現在網路拓撲資訊，而在連線後有一流量需經由 switch 至其他主機，其處理流程如下：

- a. 首先建立連線，雙方需互相發送 Hello 訊息，該訊息帶有雙方所支援 OpenFlow 版本訊息。
- b. 連線建立後，雙方會繼續使用 Echo 訊息確認連線是否存在。
- c. 在流量進入 switch 時，switch 會查找 flow table 符合的 flow entry，因為目前 flow table 為空，使 switch 會發出 Packet-in 訊息（該訊息會帶有流量封包資訊）請求 controller 協助處理。
- d. controller 在接收 Packet-in 訊息後，以 Packet-out 告知 switch 將封包導至對應主機，同時以 Flow-mod 建立 flow entry，該 flow entry 即包含如何處理該類型封包的資訊，使 switch 之後不需再發送 Packet-in 請求 controller 協助。

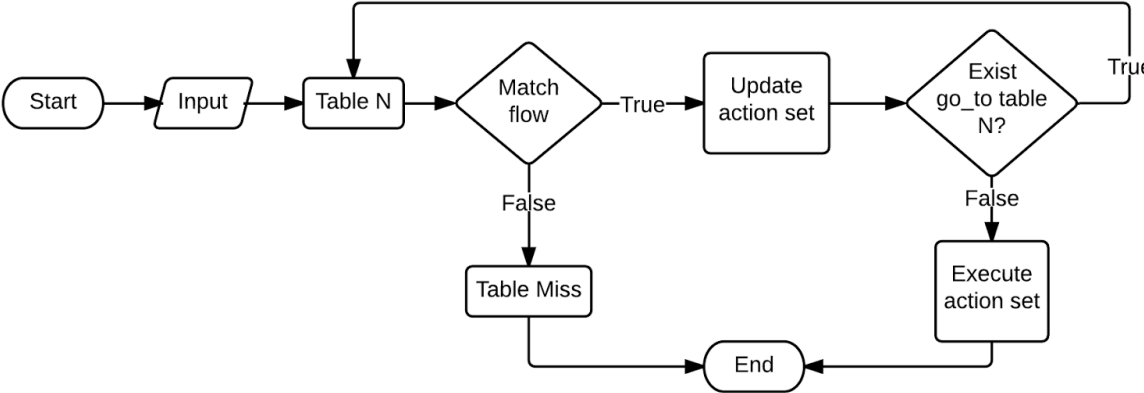


圖 1：OpenFlow pipeline 流程圖

表 4：OpenFlow 訊息分類

Symmetric	Asynchronous	Controller-to-switch
Hello Echo	Packet-in Flow-removed Port-status Error	Features Read-state Flow-mod Packet-out

OpenFlow 產品

如表 5 所示，目前在開源社群上有許多支援 OpenFlow 的虛擬 switch，且市面上已有廠商推出實體 OpenFlow switch，且所列之實體 OpenFlow switch 皆以 Open vSwitch 為基

底，使 switch 具有處理 OpenFlow 的功能（註：表 5 所列之 switch 皆都支援 OpenFlow 1.3 版），而在 controller 方面，如表 6 所示，同樣有著開源專案的 OpenFlow controller，但各自所支援的 OpenFlow 版本狀況不一。

表 5：OpenFlow switch 列表

Switch	Implementation
Open vSwitch	Software
CpQd	Software
Trema Switch	Software
Pica8 P-3290	Hardware
Centec V350	Hardware

表 6：OpenFlow controller 列表

Controller	1.0	1.2	1.3	1.4
Floodlight	✓			
OpenDaylight	✓	✓	✓	
Ryu	✓	✓	✓	✓

3. OpenFlow 測試

在上節所介紹的 OpenFlow 產品，在使用上必然會將產品對 OpenFlow 的實作程度列入考慮，該資訊則必需經過測試得知，而在測試又可分為符合性（Conformance）、互通性（interoperability）、效能（Performance）等面向。

測試規範

而在測試方式中，如表 7 所示，Open Networking Foundation（簡稱 ONF，為 OpenFlow 開發以及推廣的組織）有定義一測試標準，該標準針對符合性測試給予應有的測試網路拓撲、應測項目等資訊，使廠商或使用者可以使用該測試標準進行測試及評估產品實作程度；Systematic OpenFlow Testing (SOFT) 則是使用 symbolic execution 對兩個 OpenFlow switch 進行測試，若兩個裝置有不同的行為或結果時，即將該次輸入列入為測試項目，並以此測試項目繼續相關測試，以此作為互通性的測試；IXIA 是一家提供各式測試服務的公司，其 IxANVL 提供網路協定驗證測試服務，並也提出 OpenFlow 的符合性測試方式，且該測試有將 OpenFlow 1.3 版列入測試中。

表 7：OpenFlow test suite

Test suite	OpenFlow	Target	Test plane
ONF Test spec[8]	1.0	Data plane	Conformance
SOFT[9]	1.0	Data plane	Interoperability
IXIA IxANVL[10]	1.0/ 1.3	Data plane	Conformance

測試服務

除了以上測試規範、方法外，亦有廠商、實驗室提供 OpenFlow 測試服務，如表 8 所示，其中 PlugFest 為 ONF 召集眾多廠商進行 OpenFlow 測試，並提供 OpenFlow 認證服務；InCNTRE 則是 ONF 所認證之實驗室，亦會提供相關 OpenFlow 測試服務；另外，Spirent 亦由提供 OpenFlow 測試，並且有針對效能方面的測試服務。

表 8：OpenFlow 測試服務

Service	OpenFlow	Target	Test plane
ONF PlugFest[11]	1.3	Data plane	Conformance, interoperability
InCNTRE SDN testing services[12]	1.0	Data plane	Conformance, interoperability
Spirent TestCenter[13]	1.0/ 1.3	Data plane	Conformance, performance

測試工具

而 OpenFlow 相關組織以及研究單位亦有推出測試工具可供自行測試，如表 9 所示，其中 Ryu test tool 為 Ryu controller 推出的測試工具，而該組織亦利用此套工具進行 Ryu 與多款 switch 間的符合性測試，並在 Ryu certification[17] 頁面公佈測試報告，其測試項目分為必需、選擇兩類，該分類是以 OpenFlow 協定下是否該項為必需實作功能作為區分標準。

表 9：OpenFlow 測試工具

Tool	OpenFlow	Target	Test plane
OFTest[14]	1.0	Data plane	Conformance
Ryu test tool[15]	1.3/ 1.4	Data plane	Conformance
NICE[16]	1.1	Controller module	Functionality

在眾多的測試中，必然會有未通過測試的項目，其未通過的原因是值得進一步探討的，以 Ryu certification 的測試報告中 Open vSwitch 舉例，其必需實作功能通過比率為 100%，而可選實作功能通過比率僅 63%，此狀況在 SDN 架構下，可能會使 controller 無法將特定

的功能，統一套用至 switch，其中的可能性為 controller 或 switch 單一方面實作錯誤，或兩者對 OpenFlow 實作有不同的實現方式，亦或是對該功能未進行實作等。

4. 驗證測試項目方法

本次實驗選擇使用 Open vSwitch 做為待測品，並以 Mininet 進行模擬 SDN 網路測試環境，再以 Ryu 做為 OpenFlow 的 controller，架構如圖 2 所示。當中 Open vSwitch 版本為 2.3.90 與 Ryu certification 測試報告版本相同，Mininet 版本為 2.1.0，Ryu 版本為 3.12。

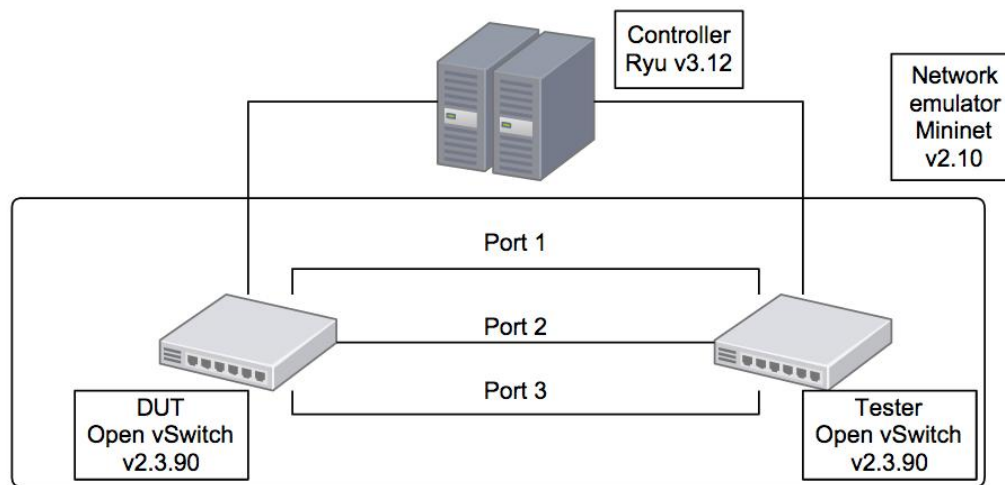


圖 2：測試環境架構圖

本次實驗將重新執行 Ryu test tool 測試，並將結果與測試報告進行比對，確認其環境是否與測試報告相符，再針對未通過測試項目，使用 Wireshark (1.12.0 版本) 進行監聽 OpenFlow 封包，以確認 OpenFlow 封包內容與測試項目相符合，並利用 ovs-ofctl 此款 Open vSwitch 所提供設定 flow entry 的工具，新增與測試項目相同之 flow entry，以此流程進行驗證測試項目。

5. 未通過測試項目分析

在本次實驗所進行的測試結果，共測試 112 項的功能，且與 Ryu certification 中的 Open vSwitch 測試報告大致相同，而其未通過測試項目依測試類型、測試結果報告及 Open vSwitch 所回傳之 OpenFlow 錯誤碼進行分類、統整，其中錯誤碼部份利用 Wireshark 查看對應意義。

如表 10 所示，其中 Meter 類型中，所進行的所有測試皆因為 Open vSwitch 所提供 meter 數量為零，導致所有測試項目皆因無足夠空間而失敗，Group 類型中，select 為自 action set 隨機選取一 action 執行動作，在該測試項目中，使用一流量輸入至 switch，而對應之 action set 為輸出至 port 1、port 2，在 select 的運作機制下，應為 port 1、2 皆需接收

流量，而在 Open vSwitch 僅有 port 1 有流量資料，該情況明顯不符合隨機定義，故為 Open vSwitch 實作錯誤。

表 10：未通過測試項目分類

Type	Action		set_field	Match	Group	Meter	
Case(s)	copy_ttl_out copy_ttl_in set_nw_ttl dec_nw_ttl push_pbb pop_pbb push_vlan	push_mpls	eth_type ip_proto icmpv4_type icmpv4_code icmpv6_type icmpv6_code	ipv6_flabel ipv6_nd_target ipv6_nd_sll ipv6_nd_ttl mpls_bos	pbb_isid ipv6_exthdr	select	All test cases
Reason	Unsupported	Incorrect packet	Bad argument	Bad field	Unexpected throughput	Out of meters	

OpenFlow 實作不符合

針對 Incorrect packet、Bad argument、Bad field 等錯誤原因，因這些項目皆以 flow entry 進行測試，所以本實驗透過 ovs-ofctl 直接略過 controller 直接新增測試項目 flow entry，透過此方式獲取 Open vSwitch 本身所回報的錯誤訊息，而透過該訊息進而得知 Open vSwitch 與 Ryu 的實作有所差異，其結果如表 11 所示，當中 Open vSwitch 在對 push_mpls 處理上，因不符合 Ryu 的實作而導致未通過，其中 eth_type=0x86dd, actions=push_mpls:0x8847 為替 IPv6 封包加上 MPLS header，Open vSwitch 處理該測試項目時，會將封包 MPLS label 欄位設為 2，但在 Ryu 的實作上則是以 0 為正確結果，依據 MPLS[18] 所規範的標準，其 MPLS label 欄位在該狀況下應為 2，因此 Ryu 在 MPLS 實作部分並不正確；另一測試項目 eth_0x8606, actions=push_mpls:0x8847 為替 ARP 封包加上 MPLS header，Open vSwitch 在處理此測試項目是將封包 MPLS ttl 設定 64，而 Ryu 同樣以 0 為正確結果。另在 ICMPv4、IPv6 等部分測試項目，Open vSwitch 與 Ryu 同樣也有實作不符合規範的情況，其中 ICMPv4 中 Open vSwitch 以 icmp_type、icmp_code 表示，而 Ryu 是以 icmpv4_type、icmpv4_code 表示，這會使雙方在溝通時，因參數無法辨識，使測試項目失敗。

OpenFlow 未支援功能

在發現部分測試項目在 Open vSwitch 有其特定 keyword，本實驗修改測試項目內容使其符合 Open vSwitch keyword，並同樣使用 ovs-ofctl 進行驗證，結果如表 12 所示，其中 push_vlan:0x88a8 是依 ovs-ofctl manual page[19] 得知目前 Open vSwitch 僅支援 push_vlan:0x8100，而表 12 中 set_field 類別中的未通過測試項，其錯誤訊息表示不支援修

改的動作，另外 Match 中 pbb_isid、ipv6_exthdr 等屬性 Open vSwitch 無法進行辨識，因此也歸為未支援一類。

表 11：Open vSwitch & Ryu OpenFlow 實作差異

Test case	Open vSwitch	Ryu
eth_type=0x86dd, actions=push_mpls:0x8847	mpls, label=2	mpls, label=0
eth_type=0x8606, actions=push_mpls:0x8847	mpls, ttl=64	mpls, ttl=0
ICMP type	icmp_type	icmpv4_type
ICMP code	icmp_code	icmpv4_code
IPv6 flow label	ipv6_label	ipv6_flabel
IPv6 nd target	nd_target	ipv6_nd_target
IPv6 nd sll	nd_sll	ipv6_nd_sll
IPv6 nd ttl	nd_ttl	ipv6_nd_ttl

表 12：Open vSwitch 未支援功能列表

Type	Action		set_field	Match
Caes(s)	copy_ttl_out copy_ttl_in set_nw_ttl dec_nw_ttl push_pbb pop_pbb	push_vlan	eth_type, ip_proto, icmp_type, icmp_code, icmpv6_type, icmpv6_code, ipv6_label, nd_target, nd_sll, nd_ttl, mpls_bos	pbb_isid, ipv6_exthdr
Reason	Unsupported	Only support 0x8100	Read-only error	Unknown keyword

6. 結論

由以上的介紹，可得知目前 SDN 相關標準，其中是以 ONF 所推廣的 OpenFlow 為主流，此外 Cisco 亦有提出 OpFlex 做為 SDN 標準。當中 OpenFlow 提供許多功能可供 control plane 控制 data plane；而為求產品之品質，便有測試需求產生，目前則有 ONF、廠商提出符合性測試規範、方法或提供測試服務，此外也有開源的測試工具（Ryu、OFTest）可進行符合性測試，以此也可看出目前尚缺少互通性測試、效能等方面 SDN 的測試方法與工具，且因 SDN 架構下功能均集中至 controller，其處理效率也必需列入考量，但在目前也未有測試方法被提出。

本篇報告為探討符合性測試未通過的原因，便針對 Open vSwitch 的符合性測試結果進行分析，該測試結果共測試 112 項的功能，其共有 21 項為不支援功能，而當中有 7 項的實作與測試 controller 有差異，實作差異中有 6 項目為 OpenFlow keyword 不同導致 Open vSwitch 無法辨視，另一項目其運算結果不與測試 controller 相符，該項目中反而是測試方判斷錯誤。依此結果，OpenFlow 產品會因未支援功能、實作不符合標準、實作錯誤等原因，導致測試項目失敗，進而在產品間搭配使用時，部分功能無法正常運作。

參考文獻

- [1] “Software-Defined Networking: The New Norm for Networks”, Open Networking Foundation, Apr 13, 2012
- [2] “OpenFlow Switch Specification Version 1.4.0”, Open Networking Foundation, Oct 14, 2013
- [3] “OpFlex: An Open Policy Protocol”, Cisco, 2014
- [4] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core”, Cisco, Mar, 2011
- [5] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman, “Network Configuration Protocol (NETCONF)”, Internet Engineering Task Force, Jun, 2011
- [6] P. Quinn, J. Guichard, R. Fernando, S. Kumar, M. Smith, N. Yadav, P. Agarwal, R. Manur, A. Chauhan, U. Elzur, P. Garg, B. McConnell, C. Wright, “Network Service Header”, Internet Engineering Task Force, Jul 3, 2014
- [7] Network Functions Virtualization, <http://www.etsi.org/technologies-clusters/technologies/nfv>
- [8] “Conformance Test Specification for OpenFlow Switch Specification 1.0.1”, Open Networking Foundation, Jun 13, 2013
- [9] M Kuzniar, P Peresini, M Canini, D Venzano, “A SOFT Way for OpenFlow Switch Interoperability Testing”, IMDEA Networks, Spain, 2012
- [10] IxANVL OpenFlow Test Suite, IXIA, <http://www.ixiacom.com/products/display?skey=ixanvl-openflow>
- [11] Plug Fest & Open Flow Certification Update, Open Networking Foundation, May, 2013
- [12] SDN Lab Testing Services, InCNTRE SDN Lab, <http://incntre.iu.edu/testing>
- [13] OpenFlow 1.0 / 1.3 Switch Testing Package, Spirent TestCenter, Rev.B 10/13
- [14] OFTest Documentation, <http://www.openflowhub.org/display/OFTest/OFTest+--+Validating+OpenFlow+Switiches>
- [15] OpenFlow Switch Test Tool, Ryubook 1.0 documentation (http://osrg.github.io/ryubook/en/html/switch_test_tool.html), 2014
- [16] M Canini, D Venzano, P Peresini, D Kostic, J Rexford, “A NICE Way to Test OpenFlow Applications, Princeton University”, Apr, 2012
- [17] Ryu Certification, Ryu SDN Framework Community, <http://osrg.github.io/ryu/certification.html>, 2014

- [18] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta, “MPLS Label Stack Encoding”, Internet Engineering Task Force, Jan, 2001
- [19] ovs-ofctl, <http://openvswitch.org/cgi-bin/ovsman.cgi?page=utilities%2Fovs-ofctl.8>