

國立臺灣科技大學資訊工程系

一百零六學年度第一學期專題研究

總報告

SDN 下 Switch 尋找路徑的模擬

研究組員

_____	_____
_____	_____
_____	_____

指導教授：_____

中 華 民 國 106 年 月 日

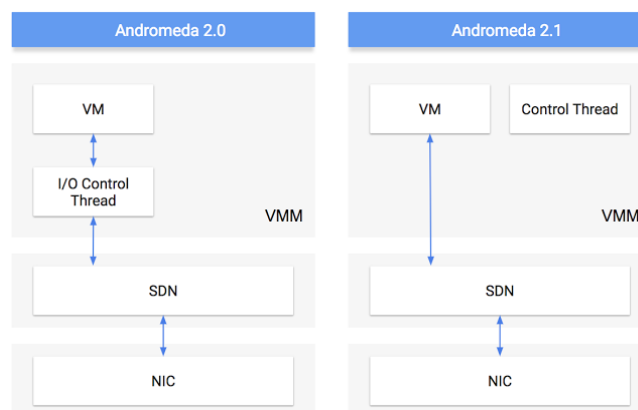
SDN 下 Switch 尋找路徑的模擬

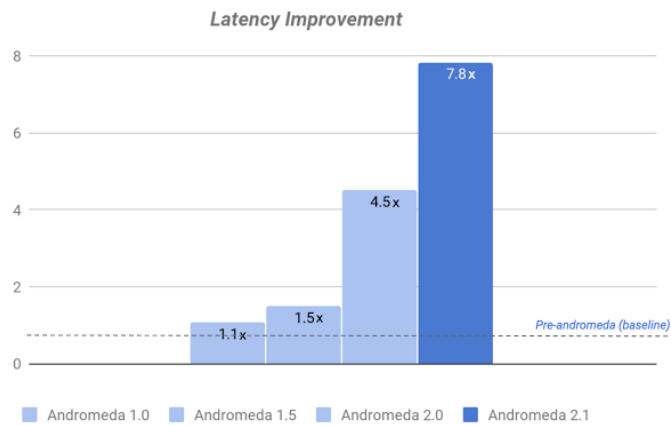
一、 摘要

專題目的是從 Host A 發送資料給 Host B，中間經由多條路徑，如果當前的路徑壞掉，可以及時找到新的路徑繼續發送資料。通過修改 SDN 環境下的 Controller 的程式碼，使得 Controller 可以不通過外部 API，直接控制 Switch 的行為，以使 Switch 的路徑發生改變，可以運作新的轉發規則和尋找策略，從而解決路徑選擇的問題。

二、 專題介紹

SDN 可以說是近年來最火熱的新型網絡架構，像是 Google 及 Facebook 這些大公司早在幾年前就開始部署 SDN 的數據中心。Google 所使用的 SDN 架構 Andromeda 為 Google 帶來降低 40% 網絡延遲的好處¹。不光是在數據中心，在工廠環境中 SDN 也可以帶來新的優勢，通過 SDN 架構可以靈活改變網絡需求，能夠降低成本提升性能。由此可以看出 SDN 將是未來云時代和物聯網時代的重要技術，所以對於 SDN 的研究將會是非常必要的。





Google 使用的 SDN 架構以及性能提升（出處參考資料 1）

本專題研究的場景是，在一個複數路徑的 SDN 架構的拓撲下，某 Switch 會向另一端發送訊息，此時路徑發生 failure 無法連通，Switch 能夠及時尋找新的路徑繼續發送訊息。情景非常簡單，但是問題卻很突出，在大型網絡架構中，節點 failure 的情況非常普遍，在傳統網絡環境下也是非常棘手的問題。如何能夠在 SDN 的環境下，實作一種新的解決方案是本專題的重點。

本專題經歷過多個階段。

1．早期基礎學習階段

- a) 本階段主要進行 SDN 的基礎學習，瞭解架構的運作機制，熟悉不同 Controller 的特點，學習使用 mininet 建立拓撲及使用 python 編寫拓撲建置腳本²³⁴⁵。
- b) 熟悉 controller 與 Switch 和 Router 的溝通機制，明白在複雜網絡拓撲情況下的 SDN 運行方式⁶。

2．專題實作階段一

- a) 實作 SDN 下 host 之間通訊的 Demo。
- b) 實作在網絡拓撲中進行流量測試，編寫 python 自動化脚本，設立不同的 switch 參數，觀察在不同參數下流量的變化。
- c) 實作 Miniedit 工具，圖形化繪製網絡拓撲。

3．專題實作階段二

- a) 研讀追蹤 Open vSwitch⁷的程式碼，借由修改程式碼以達到實作出 Switch 的 Virtual Port 功能，可以提供轉發路徑的功能。

在暑期追蹤程式碼的過程中，發現由於 OVS 的代碼量非常龐大，今時不同往日，嘗試修改程式碼去實作新功能相對來說複雜度比較大，而且對於代碼熟悉度要求很高。並且在通讀文檔和更新日志的過程中，發現 Virtual Port 功能已經實作在當前的版本中，使用上和預想有一定出入，因 OVS 下已有一個 Open Source 的項目 OVN⁹解決了 Virtual Network 的問題，本階段至此告一段落。
- b) 實作 POX Controller⁸下通過修改 Controller 程式碼，使 Switch 可以改變轉發規則和尋找策略。

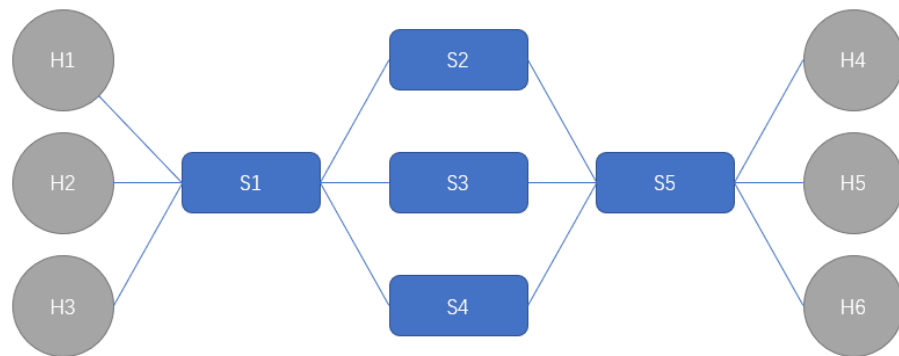
三、 理論方法

POX Controller 是一個使用 Python 編寫的 OpenFlow Controller，也可以作為 OpenFlow 的 Switch 來使用，這裏使用 POX 來作為場景中的 Controller。

主要方法是使用 POX 在固定時間周期內通過輪詢的方法，動態改變 Switch 的轉發規則，在 SDN 環境中有多種方法可以達成此目的，這次使用修改 POX 腳本的方法是為了可以自動化運行，循環執行並打印出日志記錄以供觀察。

通過 Mininet 建立一個五個 Switch 相互連接的拓撲。如圖所示。

Simulation Topo



專題設定的模擬情景拓撲

其中編號為 1 的 Switch S1 連接了三個 Host，H1、H2、H3。編號為 5 的 Switch S5 連接了另外三個 Host，H4、H5、H6。

H1 發往 H4 的 packet，默認路徑會經過 S1，S2，S5。根據專題討論的場景，這裏假設此路徑無法使用，我們需要 S1 使用新路徑繼續發送數據。此處路徑從拓撲中直接斷開連接或者只是不去使用他，對於實驗模擬不會有影響，在直接斷開的情況下，Controller 會發覺路徑無法連接，日志中可以觀察到，因為 Controller 部分是采用輪詢的方法，在下一個時間周期后，會下達新的轉發規則到 Switch，這時 Switch 會使用新的路徑繼續發送訊息。在本次實驗中我們保留了初始路徑的連通性，使得拓撲可以重複使用。在 Controller

的脚本中，指定了 S1 會有三個不同的 port 指向不同的 Switch，S1 的 Port4 指向了 S2，Port5 指向了 S3，Port6 指向了 S4。

在這張拓撲中，S1 到達 S5，實際上有三條路綫，這裏通過不同的 port 定義了路徑，初始路徑之後是 S1，S3，S5，在之後是 S1，S4，S5。至此達到在 SDN 環境下模擬 Switch 的路徑尋找。

四、 實驗

a) 實驗環境：

作業系統：Linux Mint 18.2 Cinnamon 64-bit on vmware

Mininet VM：mininet 2.2.2 on Ubuntu 14.04.4 server

Pox version：Branch carp

OpenDayLight version：Nitrogen SR1

Python version：python 2.7

b) 實驗步驟：

1. 將 mininet VM 開啓，以備之後可以 SSH 進去進行操作。
2. 在系統中 POX 的目錄裏面運行修改過的脚本文件 testcon.py 并且另加參數來顯示 web 界面，設定 openflow 功能以及展示拓撲。指令

為：
`sudo ./pox.py testcon samples.pretty_log web messenger messenger.log_service messenger.ajax_transport`

3. 通過 SSI `openflow.of_service poxdesk openflow.discovery poxdesk.tinytopo poxdesk.terminal`
mininet@

4. 在 mininet 中執行：`sudo python mini2.py` 這條命令會建立上圖所示的拓撲。
5. 打開瀏覽器，輸入 `http://127.0.0.1:8000/poxdesk` 進入 POX 的 web 界面 Poxdesk。打開后在右下角菜單欄中開啓 logviewer，topoviewer，tableviewer，其中 tableviewer 切換到 S1。
6. 執行 Pingall 測試拓撲是否成功建立并且能夠連通。
7. 在 mininet 中執行：`h1 ping -i 0.1 h4` 這條命令會每秒從 H1 傳送十個封包到 H4。
8. 觀察 Poxdesk 中 logviewer 是否打印出正確的日志，是否存在錯誤訊息。可以查看 topoviewer 中拓撲是否和情景設定好的拓撲一致。
下圖為正常情況下拓撲的顯示。在 OpenDayLight 中可以顯示 Host，在 Poxdesk 中則沒有顯示。



Poxdesk 中拓撲的顯示

9. 在 tableviewer 中，觀察 S1 中發往 H4 的目的地的條目，nw_dst

10.0.0.4/32。這裏會發現在 actions 中，他的 Output 會隨著時間進行變化，Output 最開始是 Port4，隨即變成 Port5，然後是 Port6。通過這個現象可以發現 POX 下的確可以使用輪詢的方法動態改變 Switch 的路徑和尋找策略。

10. 錯誤排除：在步驟 8 中，如果發現拓撲顯示和預期不一樣，此時可以在 mininet 中清空拓撲重新建置，重新進行步驟 4 到 6，此時會顯示出應有的拓撲。原因可能是在之前建立過的拓撲還遺留在 mininet 中被 POX 追蹤到，導致顯示錯誤，錯誤的拓撲無法順利連通，在第六步 Pingall 中可以提前知道，如果無法 Pingall 就可以執行清除命令 `mn -c`。

五、 結論

在本專題中，實作出一個在 SDN 環境中簡單但有效的網絡程式，通過修改 POX 程式的方法去控制 Switch 的轉發行爲，並且可以成功觀察到實驗結果，可以成功模擬專題所要求的測試情景。從 Host A 發送資料給 Host B，中間經由多條路徑，如果當前的路徑壞掉，可以及時找到新的路徑繼續發送資料。

六、 參考資料

1. <https://cloudplatform.googleblog.com/2017/11/Andromeda-2-1-reduces-G>

CPs-intra-zone-latency-by-40-percent.html

2. OpenFlow MPLS and the open source label switched router Teletraffic Congress (ITC), 2011 23rd International
3. An SDN-based solution for increasing flexibility and reliability of dedicated network environments 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)
4. Chaos Monkey Increasing SDN Reliability through Systematic Network Destruction SIGCOMM ' 15 August 17-21, 2015
5. <http://mininet.org/>
6. <https://www.opendaylight.org/>
7. <http://openvswitch.org/> <https://github.com/openvswitch/ovs>
8. <https://openflow.stanford.edu/display/ONL/POX+Wiki>
9. <http://openvswitch.org/support/slides/OVN-Vancouver.pdf>