

基於P4 Hash演算法的網路流量平衡

Network Traffic Balance Based on P4's Hash Algorithm

第十組

r09921072 劉哲瑞

r09921088 李昕耘

r09942138 馮景泰

Outline

1. **Introduction**
2. **Problem Definition**
3. **Methodology**
4. **Implementation**
5. **Experiment Setup**
6. **Experiment Result**
7. **Related Work**

Introduction

- 人們在上網常是因為是傳輸量太大導致 link failure, 對於網路流量的控管是非常重要的, 畢竟在有些地方對於網路延遲或網路不順會給使用者造成很大的困擾, 我們希望能從中找到解決的方法。
- SDN能有效且方便的對網路做控制, 如網路做拓樸分析, 並且運用一些控制工具, ex: POX, NOX, Floodlight, OpenDaylight等等, 再從獲取到的網路拓樸並對其做 path Algorithm由於SDN相比傳統網路更具有便利性, 所以我們想使用 SDN這項技術去試著解決改善 Traffic balance這個問題。

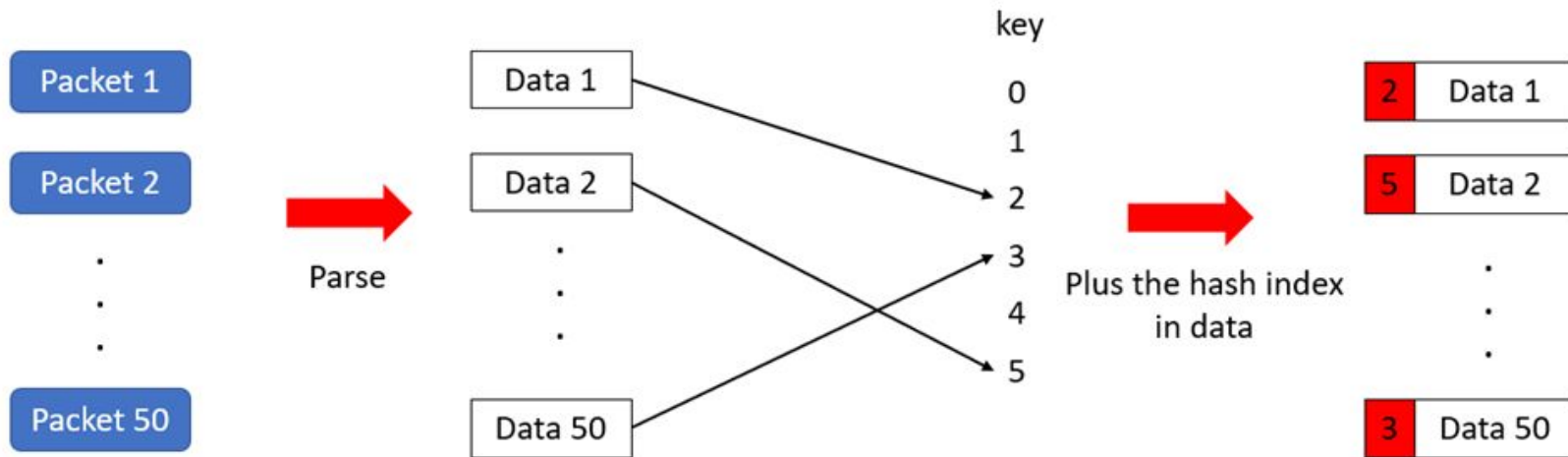
Problem Definition

- 在傳統網路中只能走 shortest path, 並沒有辦法控制 packet 走其他路徑, 如此一來可能導致嚴重的網路壅塞, 再者, 許多單位都有設置流量限制
- problem 1: 如何讓封包走不同路徑以避免網路壅塞
- problem 2: 如何設定 load balance 比例, 即走不同路徑的比例

Methodology

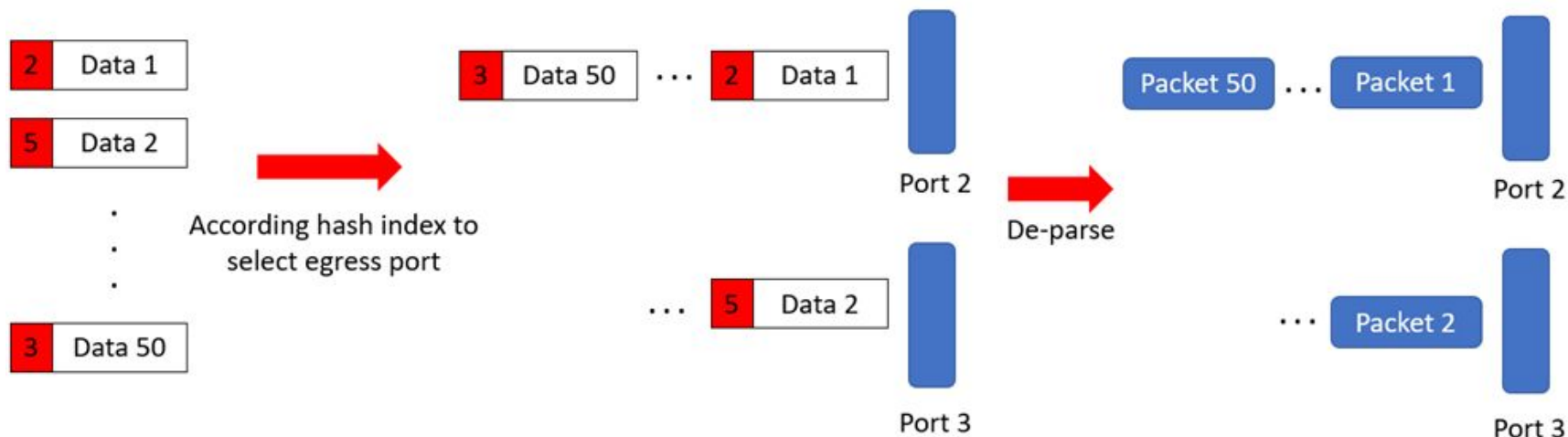
Openflow可以做到把流量平分給 output port, 利用P4中甚至能做到自定義 output到各port的流量比例, 我們的解決方法則是決定使用 P4的hash function來控制流量以相對的比例走第二短(hop數第二少的)的路徑, 來減少全部封包都走最短路徑所發生的網路壅塞情況。

Implementation on ingress processing



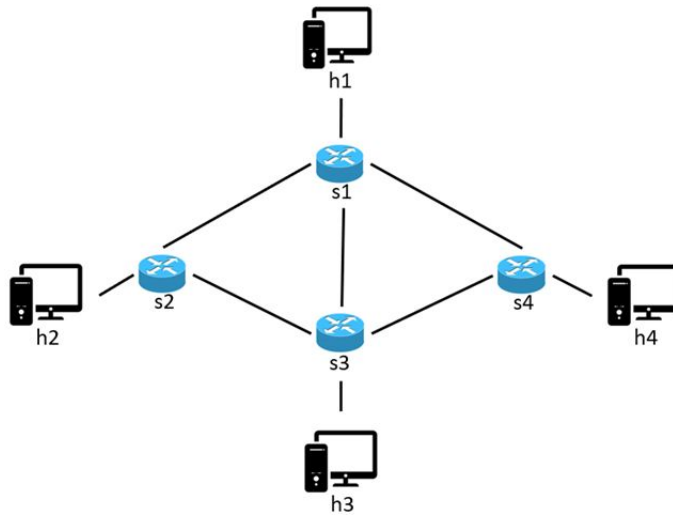
1. 交換機(switch)接收到封包後，會解析其標頭 (header)並提出資料(data)。
2. P4內建的Hash演算法會根據標頭裡的內容在資料前加上一個 hash index $\in [0,1,2,3,4,5]$ 。

Implementation on egress processing



1. 交換機(switch)根據在ingress加上的hash index, 來決定要從哪個 output port發送。
2. P4將標頭加入準備發送的資料 (data), 並將其包裝成封包發送。

Experiment Setup



```
s1 -> gRPC port: 50051
s2 -> gRPC port: 50052
s3 -> gRPC port: 50053
s4 -> gRPC port: 50054
```

```
*****
```

```
h1
```

```
default interface: eth0 10.0.1.1
```

```
08:00:00:00:01:01
```

```
*****
```

```
*****
```

```
h2
```

```
default interface: eth0 10.0.2.2
```

```
08:00:00:00:02:02
```

```
*****
```

```
*****
```

```
h3
```

```
default interface: eth0 10.0.3.3
```

```
08:00:00:00:03:03
```

```
*****
```

```
*****
```

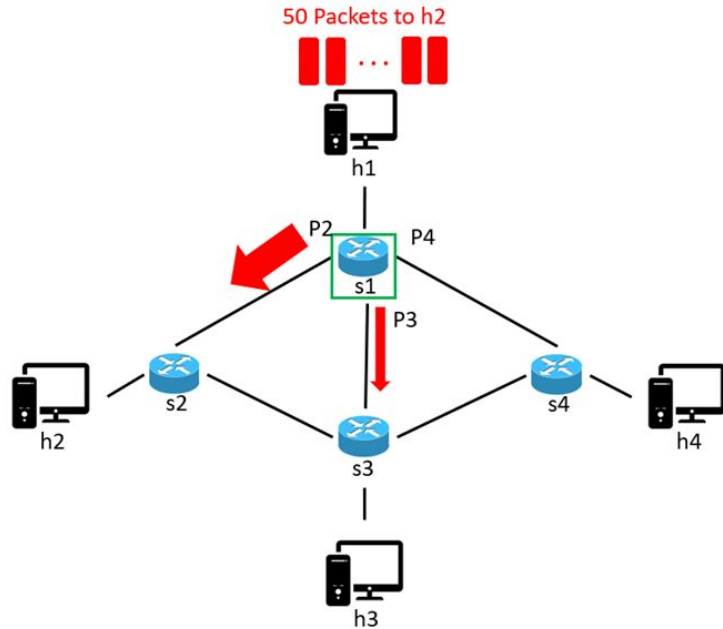
```
h4
```

```
default interface: eth0 10.0.4.4
```

```
08:00:00:00:04:04
```

```
*****
```


Experiment Method

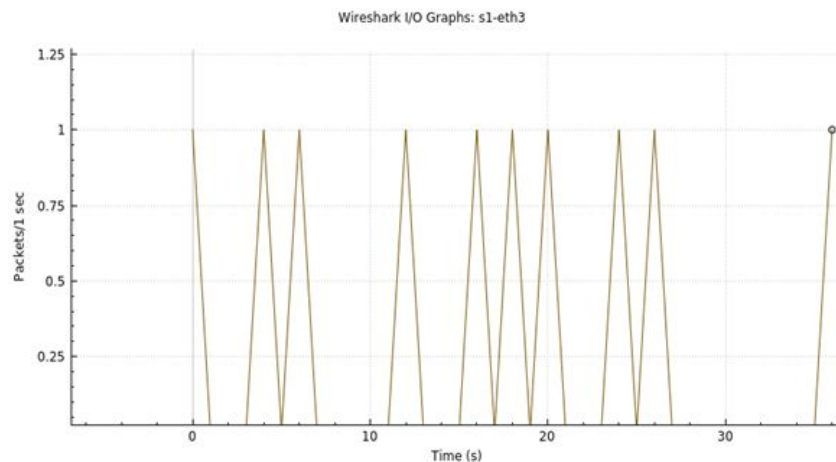
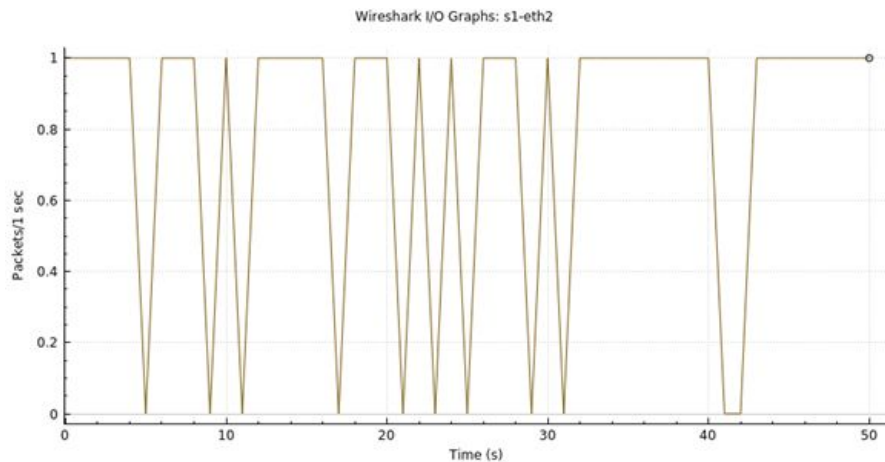


h1將會發送50個封包給h2，而我們的機制應用在s1。

s1將會根據資料上對應的 hash index的分配封包給P2和P3兩個出口。

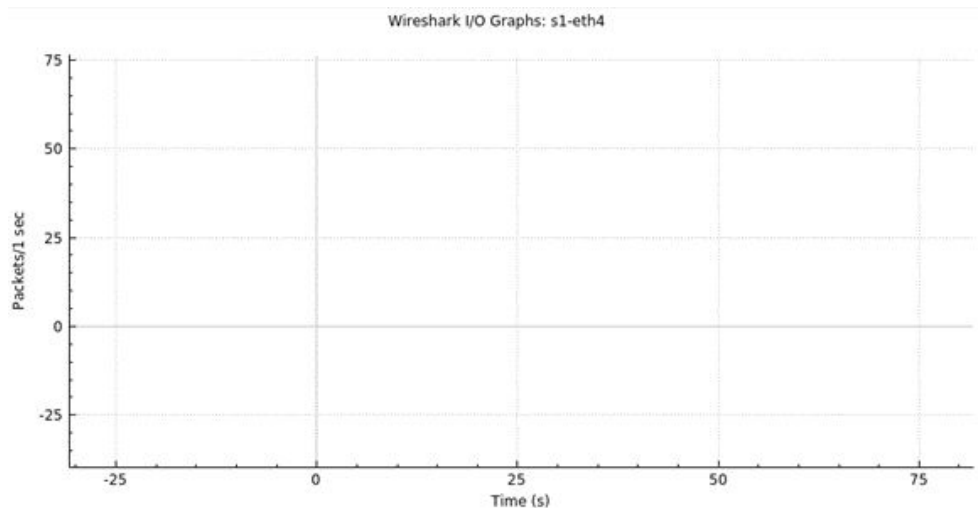
我們使用wireshark來攔截P2, P3和P4這三個出口的封包數，來評估我們的機制。

Experiment Result(1)



大部分的封包都由 port2 出口發出，其中有 10 個封包由 port3 出口發出，跟我們預計要分配的比例 5:1 非常相近。

Experiment Result(2)



沒有封包由port4發出。

結果顯示我們的機制可以實現封包依照使用者分配比例由不同出口發送封包，且走的路徑為最短及次短的路徑。

Reference

[1] Jun Li, Xiangqing Chang, Yongmao Ren, Zexin Zhang, Guodong Wang, “An Effective Path Load Balancing Mechanism Based on SDN ”, 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications.

[2] Mao Qilin, Shen WeiKang, “A Load Balancing Method Based on SDN”, 2015 IEEE 7th International Conference on Measuring Technology and Mechatronics Automation