

K-014

OpenFlow をベースとしたネットワーク制御の実験について(2) On experiment of network control based on OpenFlow(2)

中野 陽太[†] 神田 未来[‡] 荒木 智行[‡]
Akihiro Nakano Mirai Kanda Tomoyuki Araki

1. はじめに

ネットワーク機器は通常ブラックボックス化されており、学生が自身でルータ等のネットワーク機器の通信制御プログラムに手を加えることは困難であった。OpenFlow ネットワーク[1][2]は仕様がオープンであり、それを制御するコントローラやスイッチがオープンソースで提供されているため、カスタマイズ可能でかつ低コストで実験可能となった。そのため、学生がネットワークの勉強をするにあたって非常に有用である。図1のような一般的なPCや、安価な家庭向けルータをLinux化することでもOpenFlow対応機器を用意することができる。また、通常ネットワーク機器はベンダごとに設定方法が異なるものであるが、OpenFlowではコントローラと呼ばれる機器がOpenFlowプロトコルによってネットワークを一元的に管理するため、ネットワークの管理が容易で、ベンダの統一も行わなくてよい。OpenFlowでは制御部分と転送部分が分離されており、制御をコントローラが行い、転送処理をスイッチが行うからである。

2. 実験環境

図3のようなトポロジでコントローラとスイッチを接続し、経路の制御を行う。コントローラおよび各スイッチは図2のような標準的なデスクトップPCにNICを増設したものを使用する。コントローラおよびスイッチの仕様は表1の通りである。各PCはオープンソースのソフトウェアをインストールすることでコントローラおよびスイッチとして動作する。コントローラにはRyuを使用し、



図2 OpenFlow機器として使用したPC

表1 コントローラおよびスイッチの仕様

OS	Ubuntu 14.04LTS x64
CPU	Intel Core2Duo E6600 2.40GHz
メモリ	DDR2 4GB
ネットワークカード	I-O DATA ETG3-PCI

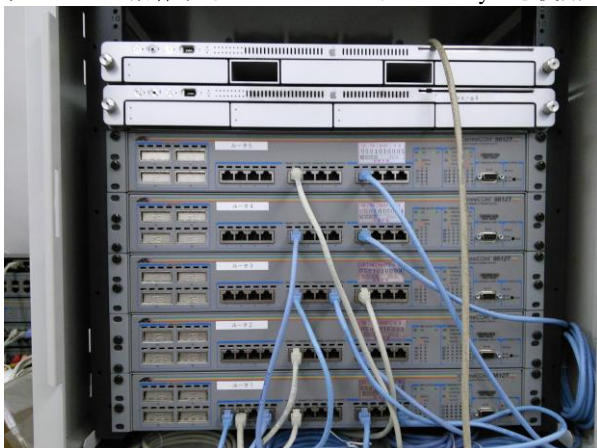


図1 通常のルーター

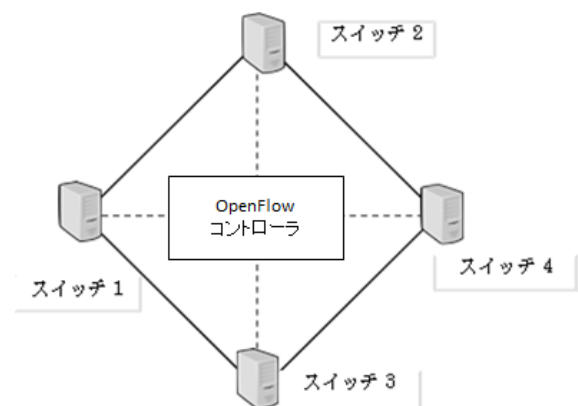


図3 実験に使用するネットワーク図

スイッチとしてOpenvSwitchを使用した。NICに搭載されているネットワークコントローラはRealtek社製RTL8169SCである。

3. ループのある経路制御

同一ネットワーク内に閉路が存在するネットワークでは、ブロードキャストパケット等のパケットが同じ経路を延々とめぐり続けるストームが発生する。これによって帯域の占有や、ネットワーク機器の故障を招くことがある。そこで、そのストームを抑制するために一時的に一部の経路を切断することで閉路の一部を切断し、閉路をなくす制御を行う。スイッチとコントローラを図 3 のように接続したネットワーク内で、ストームを抑制するための経路制御を行う。ここでは単に経路を切断するだけではなく、ネットワーク内に流れるトラフィック量に応じて切断経路を変更する。トラフィック量は、各スイッチから取得できる。

3.1 切断する経路の選択

最初にどれかの経路を無効経路とし、単位時間各スイッチに通信を行わせたとする。この時間内に各スイッチにより表 2 のような端点での通信量が観測されたとする。端点での通信量とは、あるパケットを最初に生成・送信した送信元と、最終的な送信先のペア(Source, Destination)の間で行われる通信である。例えば、無効経路を経路 D(スイッチ 4, スイッチ 3 間)とした時、(スイッチ 4, スイッチ 3 間)の通信量は、実際には迂回して経路 B(スイッチ 3, スイッチ 1 間), 経路 A(スイッチ 1, スイッチ 2 間), 経路 C(スイッチ 2, スイッチ 4 間)を通るが、通信(スイッチ 4, スイッチ 3 間)の通信量はそれぞれの迂回経路の端点での通信量には含まれていない。

まず無効経路を経路 D とした場合を考える。各経路に流れたトラフィック量は次のように算出できる。

最初に、表 2 の Source と Destination のスイッチ(端点)の組み合わせ(Combination)について、それらのトラフィック量を足し合わせてその組み合わせのトラフィック量とする。例えば(スイッチ 3, スイッチ 4 間)のトラフィック量 150MB と(スイッチ 4, スイッチ 3 間)のトラフィック量 300MB を足しあわせ、表におけるスイッチの組み合わせ {スイッチ 3, スイッチ 4 間} のトラフィック量 450MB となる。こうして各端点の組み合わせの通信が通る経路と通信量の表が完成する。

次に、各経路に流れたトラフィック量とそれらの合計トラフィック量の表を作る。例えば経路 A は、(スイッチ 1, スイッチ 2 間), (スイッチ 1, スイッチ 4 間), (スイッチ 2, スイッチ 3 間), (スイッチ 3, スイッチ 3 間)の 4 つの端点の組み合わせの通信が利用しているから、これらのトラフィック量を足しあわせ、

$$55 + 205 + 35 + 450 = 745 \text{ MB}$$

の通信が経路 A に流れるということになる。同様に他の経路 B, C それぞれについても求め、それらの合計を合計トラフィック量とする。無効経路を D としたときの各経路のトラフィック量と合計トラフィック量を表 3 に示す。この計算を、無効経路を A とした場合、B とした場合、C とした場合にも同様に行い各トラフィック量と合計トラフィック量を比較し、合計トラフィック量が最少で各経路における最大トラフィック量が最少のものを、通信効率の良い無効経路として選定した。今回の例の場合、経

路 B を無効経路とするのが最も効率が良いことが確認された。

3.2 結果

無効経路を動的に変更することにより、ping の送信時間を短くすることができた。その時の結果を図 4 に示す。

4. むすび

今回の実験により、OpenFlow でルータ以上の制御を独自に実装することができることがわかった。

今後は、大学の実験実習の中で、

- Linux のインストール
- コントローラおよびスイッチのインストール
- 制御プログラムの実装
- 動作確認

をどのように行っていくか考えていく予定である。

表 2 端点間の通信量

Source	Destination	Traffic
スイッチ1	スイッチ2	30MB
スイッチ1	スイッチ3	3MB
スイッチ1	スイッチ4	100MB
スイッチ2	スイッチ1	25MB
スイッチ2	スイッチ3	15MB
スイッチ2	スイッチ4	5MB
スイッチ3	スイッチ1	10MB
スイッチ3	スイッチ2	20MB
スイッチ3	スイッチ4	150MB
スイッチ4	スイッチ1	105MB
スイッチ4	スイッチ2	8MB
スイッチ4	スイッチ3	300MB

表 3 D を無効経路としたときの各経路の通信量

Route	Traffic
経路A	745MB
経路B	498MB
経路C	668MB
合計	1911MB

```
arapc10@arapc10:~$ ping 192.168.0.93
PING 192.168.0.93 (192.168.0.93) 56(84) bytes of data:
64 bytes from 192.168.0.93: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 192.168.0.93: icmp_seq=6 ttl=64 time=0.495 ms
64 bytes from 192.168.0.93: icmp_seq=7 ttl=64 time=0.178 ms
```

図 4 無効経路の変更による Ping 通信時間の変化

参考文献

- [1] あきみち, 宮永, 岩田; マスタリング TCP/IP OpenFlow 編
- [2] <https://www.opennetworking.org/>