

Vyatta での PPPoE と Network emulator

Vyatta ユーザ会 in 広島 2011/10/05

津森 洋伸

自己紹介

- ・ **居所**



@tumochan



hiro at tumori.nu

- ・ **お仕事**

- **某通信事業者勤務**

- ・ **伝送設備の構築**



- ・ **インターネットサービスの立ち上げ・運用**

- BGPオペレーション, IPv6で遊ぶ



- ・ **企画系？**

おうちでVyattaを使ってみる

- ・ かなり高機能なブロードバンドルータ (w
- ・ まずは使えるかどうか試してみよう。

#でも普段はFreeBSD使いなの。。

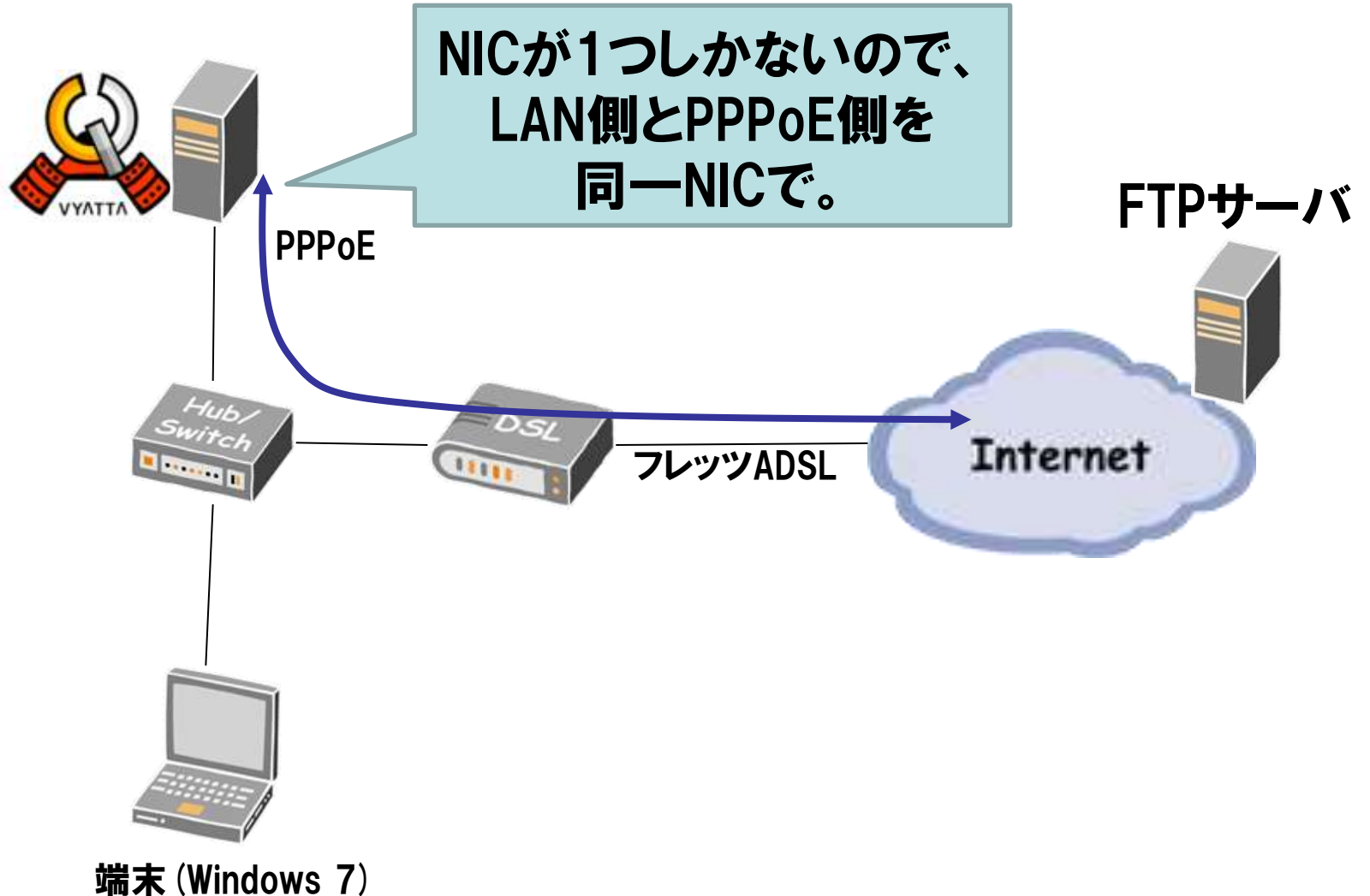
WAN環境

- ・ **回線**
 - NTT西日本 フレッツADSL モアスペシャル
 - 広島市内某局収容
- ・ **ISP**
 - 朝日ネット

準備したVyattaルータの諸元

- ・ **マザーボード**
 - Intel D510MO
 - ・ Atom D510 1.66GHz
- ・ **RAM**
 - 2GB
- ・ **Vyatta Core 6.3**

ネットワーク図



コンフィグレーション(抜粋)

```
interfaces {
  ethernet eth0 {
    address 192.168.1.1/24
    duplex auto
    hw-id xx:xx:xx:xx:xx:xx
    pppoe 0 {
      default-route auto
      mtu 1492
      name-server auto
      password *****
      user-id xxxx-xxx@atson.net
    }
    smp_affinity auto
    speed auto
  }
  loopback lo {
  }
}
```

コンフィギュレーション(抜粋)

```
service {  
  nat {  
    rule 1 {  
      outbound-interface pppoe0  
      type masquerade  
    }  
  }  
}
```


結果 (1)

- ・ 普通に通りました。(当たり前)
- ・ 確認

```
vyatta@vyatta:~$ show interfaces pppoe pppoe0

pppoe0: <POINTOPOINT, MULTICAST, NOARP, UP, LOWER_UP> mtu 1454
qdisc pfifo_fast stat e UNKNOWN qlen 100
  link/ppp
  inet xxx.xxx.216.156 peer xxx.xxx.67.2/32 scope global pppoe0

RX:  bytes      packets      errors      dropped      overrun      mcast
     7162         41           0            0            0            0
TX:  bytes      packets      errors      dropped      carrier collisions
     4623         46           0            0            0            0
```

結果 (2)

- ・ スループット

- 端末: Windows 7
- 某FTPサーバから gcc のソースを落とすのにかかった時間を計測

100% [=====>] 71,688,404 2.78M/s 時間 28s

- そもそもこのテストがどれだけ当てになるのかという議論は割愛。

traceroute

```
1  <1 ms  <1 ms  <1 ms  192.168.1.1
2  25 ms  29 ms  29 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.2]
3  24 ms  29 ms  29 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.30]
4  33 ms  39 ms  39 ms  tkyxxx.asahi-net.or.jp [xxx.xxx.38.1]
5  36 ms  39 ms  39 ms  kddxxx.asahi-net.or.jp [xxx.xxx.32.51]
6  30 ms  39 ms  39 ms  xxx.xxx.107.169
...
(snip)
...
15 43 ms  49 ms  49 ms  ftp.xxxxx.xx.jp [xxx.xxx.xxx.130]
```

ところで。

- ・ **これだけでは面白くないので。**

ネットワークのエミュレーション

- ・ いろいろな面白いエミュレーションが出来るらしい！

```
vyatta@vyatta# set traffic-policy network-emulator AAA
bandwidth          description          packet-corruption  packet-reordering
burst              network-delay       packet-loss         queue-limit
[edit]
```

ネットワークのエミュレーション

```
vyatta@vyatta# set traffic-policy network-emulator AAA network-delay
```

Possible completions:

<number> Time in milliseconds

<number><suffix>

Time with suffix (secs, ms, us)

[edit]

```
vyatta@vyatta# set traffic-policy network-emulator AAA bandwidth
```

Possible completions:

<number> Rate in k (1000) bytes per second

<number><suffix>

Rate with scaling suffix (mbit, mbps, ...)

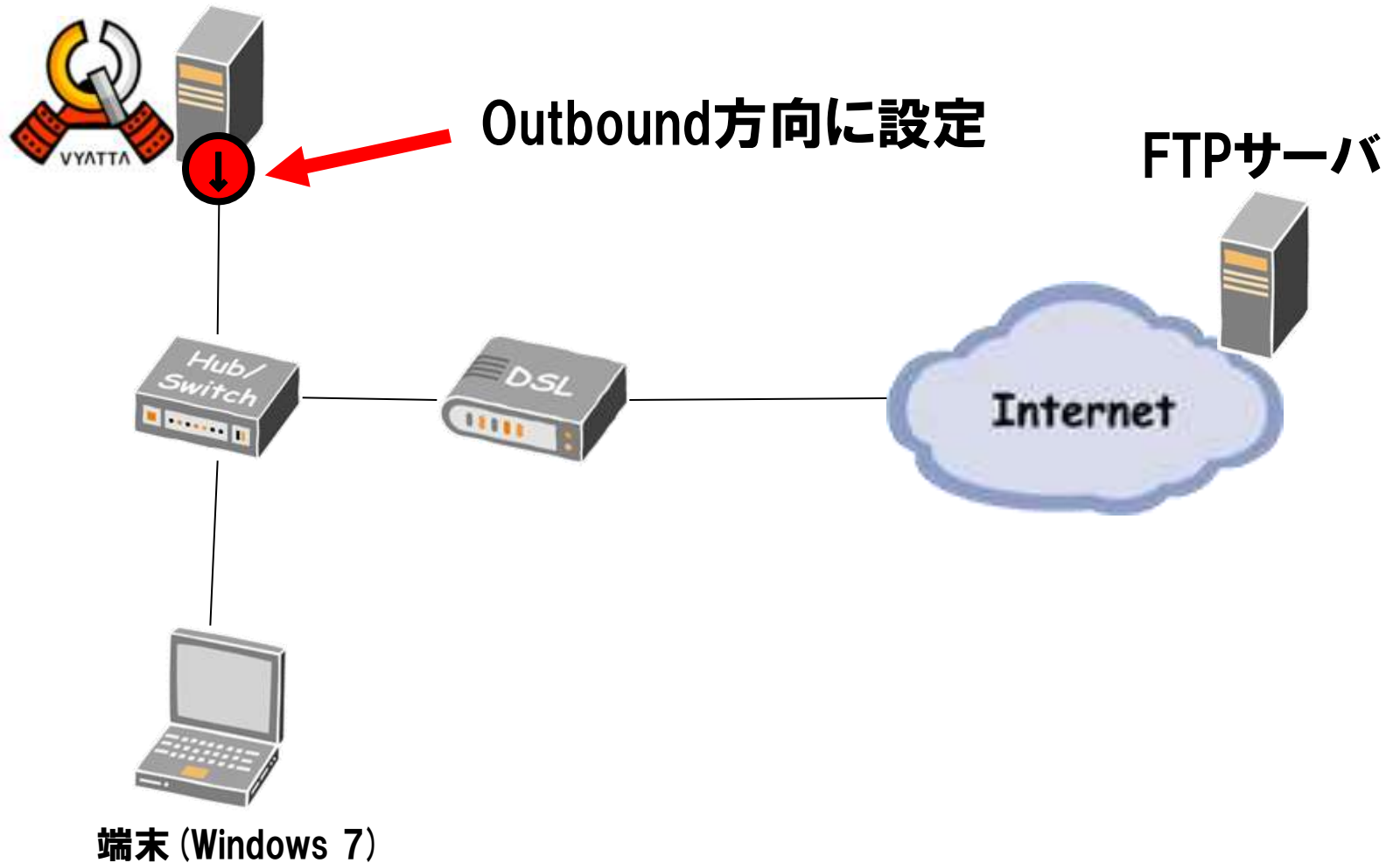
[edit]

```
vyatta@vyatta# set traffic-policy network-emulator AAA packet-loss
```

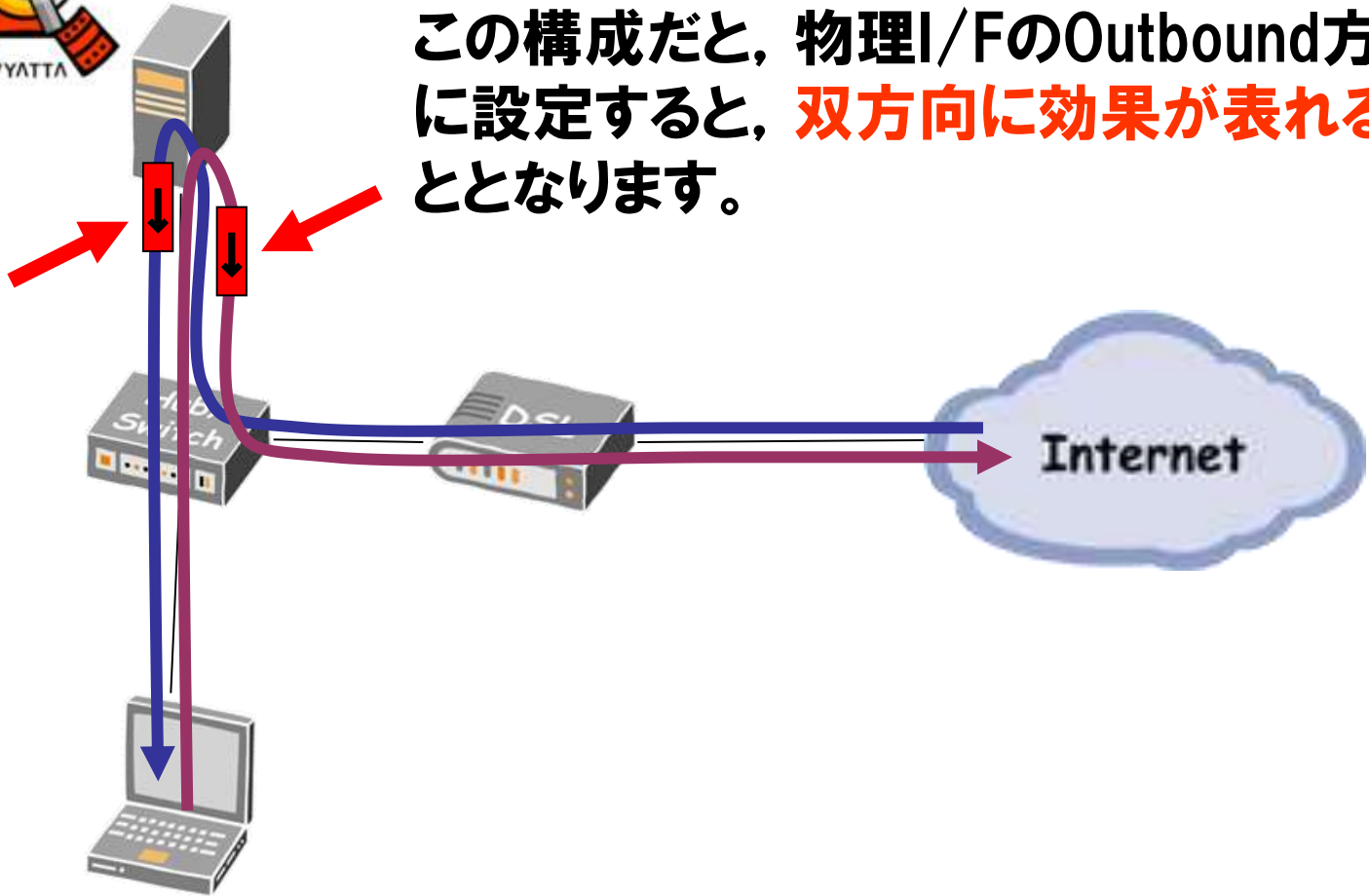
Possible completions:

<number>% Percentage of packets affected

試してみる



このI/Fに設定をかけると



この構成だと、物理I/FのOutbound方向に設定すると、**双方向に効果が表れること**となります。

端末 (Windows 7)

コンフィギュレーション例

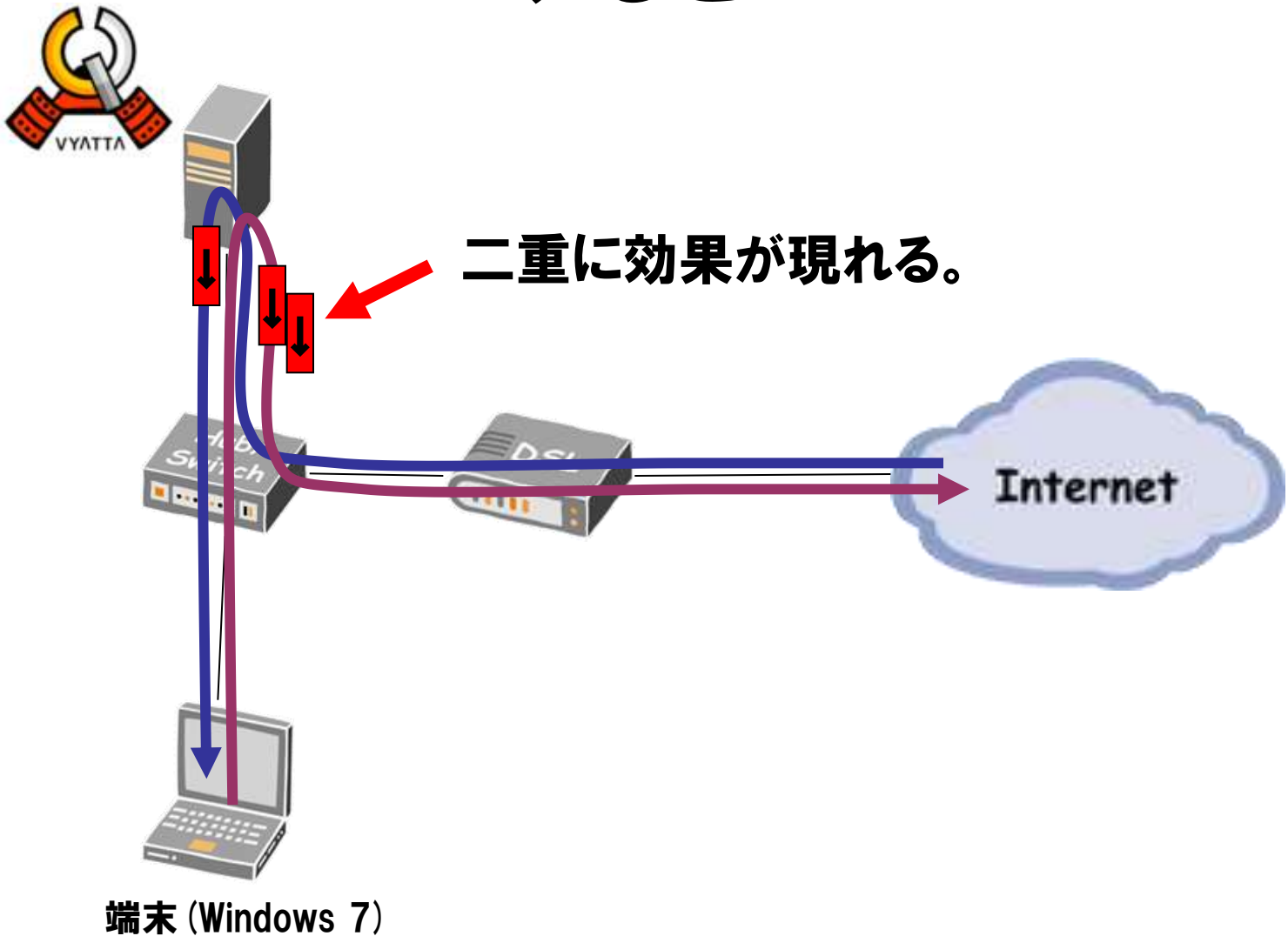
```
interfaces {
  ethernet eth0 {
    .....
    pppoe 0 {
      .....
    }
    traffic-policy {
      out POL
    }
  }
  loopback lo {
  }
}
traffic-policy {
  network-emulator POL {
    burst 15k
    network-delay 100ms
  }
}
```

最初にハマった例

```
interfaces {  
  ethernet eth0 {  
    .....  
    pppoe 0 {  
      .....  
      traffic-policy {  
        out POL  
      }  
    }  
    traffic-policy {  
      out POL  
    }  
  }  
  loopback lo {  
  }  
}
```

PPPoE内にも設定できる！

すると



結果(1) 遅延

- ・ 遅延量を20msにしてみる
- ・ 設定

```
traffic-policy {  
  network-emulator POL {  
    burst 15k  
    network-delay 20ms  
  }  
}
```

traceroute (遅延20ms)

```
1  20 ms  20 ms  20 ms  192.168.1.1
2  65 ms  69 ms  69 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.2]
3  66 ms  69 ms  69 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.30]
4  77 ms  79 ms  79 ms  tkyxxx.asahi-net.or.jp [xxx.xxx.38.1]
5  77 ms  79 ms  79 ms  kddxxx.asahi-net.or.jp [xxx.xxx.32.51]
6  77 ms  79 ms  79 ms  xxx.xxx.107.169
...
(snip)
...
15 87 ms  89 ms  89 ms  ftp.xxxxx.xx.jp [xxx.xxx.xxx.130]
```

traceroute (遅延150ms)

```
1  150 ms  150 ms  150 ms  192.168.1.1
2  322 ms  329 ms  329 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.9]
3  326 ms  329 ms  329 ms  hirxxx.asahi-net.or.jp [xxx.xxx.67.30]
4  337 ms  339 ms  339 ms  tkyxxx.asahi-net.or.jp [xxx.xxx.38.1]
5  332 ms  339 ms  339 ms  kddxxx.asahi-net.or.jp [xxx.xxx.32.51]
6  338 ms  339 ms  339 ms  xxx.xxx.107.169
...
(snip)
...
15 348 ms  349 ms  349 ms  ftp.xxxxx.xx.jp [xxx.xxx.xxx.130]
```

確認

```
vyatta@vyatta:~$ show queueing
```

Interface	Policy	Sent	Dropped	Overlimit
eth0	network-emulator	62037	0	0
pppoe0	default	5514	0	0

```
vyatta@vyatta:~$ show queueing ethernet eth0
```

```
eth0 Queueing:
```

Class	Policy	Sent	Rate	Dropped	Overlimit	Backlog
root	network-emulator	63881		0	0	0

ダウンロード時間

- ・ **バラツキがありすぎたので結果は割愛**
- ・ **意外と遅延があっても速い**
 - Windows 7 のTCPスタックが頑張っている感じもする
 - 本当に頑張ってるのかは、もう少しきちんと見てみる必要がある
- ・ **遅延が大きいと、FTPのコマンド発行→レスポンスが明らかに遅くなるので、そちらの方が影響が大きい。**

結果(2) 帯域制限

- 1Mb/sに制限してみる
- 設定

```
traffic-policy {  
  network-emulator POL {  
    burst 15k  
    bandwidth 1mbit  
  }  
}
```

“1mbps”と書くと
1MByte/s になるので注意

- 結果

```
100% [=====>] 71,688,404 114K/s 時間 10m 14s
```

確認

```
vyatta@vyatta:~$ show queueing ethernet eth0
```

```
eth0 Queueing:
```

Class	Policy	Sent	Rate	Dropped	Overlimit	Backlog
root	rate-control	2045719		0	3963	31

結果(3) パケットロス

- ・ ロス率1%にしてみる
- ・ 設定

```
traffic-policy {  
  network-emulator POL {  
    burst 15k  
    packet-loss 1%  
  }  
}
```

- ・ 結果

100% [=====>] 71,688,404 187K/s 時間 4m 42s

- やっぱり(ロス率という意味で)回線品質は大切。

確認

```
vyatta@vyatta:~$ show queueing ethernet eth0
```

```
eth0 Queueing:
```

Class	Policy	Sent	Rate	Dropped	Overlimit	Backlog
root	network-emulator	17342		3	0	0

まとめ

- ・ PPPoEをしゃべるNATルータとして普通に使いそう。
- ・ ネットワークエミュレータはいろいろな検証に使いそう。
 - きちんと動いているように見えたが、エミュレータの動き方は事前確認した方が良さそう。
- ・ もう少し定量的評価が出来れば良かった。

やっつけ仕事でごめんなさい・・・。

ありがとうございました。

Any Questions?

