

traceroute to janog

JPNIC

岡田 雅之

traceroute6 traceroute.janog.gr.jp.

```
1 janog1.tokyo-opera-city-shinjuku.tokyo (2001:470:fa0a:101::2) 1.317 ms 0.36 ms 0.245 ms
2 janog2.kdd-at-shinjuku.tokyo (2001:470:fa0a:102::2) 0.352 ms 0.562 ms 0.198 ms
3 janog3.internet-week-98-in-kyoto (2001:470:fa0a:103::2) 0.656 ms 0.28 ms 0.224 ms
4 janog4.nec-at-shiba.tokyo (2001:470:fa0a:104::2) 0.695 ms 0.335 ms 0.273 ms
5 janog5.internet-week-99-in-yokohama (2001:470:fa0a:105::2) 0.545 ms 0.36 ms 0.358 ms
6 janog6.canon-at-shimomaruko.kanagawa (2001:470:fa0a:106::2) 1.351 ms 0.372 ms 0.354 ms
7 janog7.internet-week-00-in-osaka (2001:470:fa0a:107::2) 0.834 ms 0.478 ms 0.388 ms
8 janog8.shibakoen.tokyo (2001:470:fa0a:108::2) 2.119 ms 0.514 ms 0.405 ms
9 janog9.toyama.toyama (2001:470:fa0a:109::2) 1.048 ms 2.303 ms 0.542 ms
10 janog10.meiji-kinenkan.tokyo (2001:470:fa0a:110::2) 0.987 ms 0.488 ms 0.504 ms
11 janog11.akita.akita (2001:470:fa0a:111::2) 2.034 ms 1.429 ms 0.592 ms
12 janog12.sapporo.hokkaido (2001:470:fa0a:112::2) 1.414 ms 1.174 ms 0.988 ms
13 janog13.akasaka.tokyo (2001:470:fa0a:113::2) 0.947 ms 0.644 ms 2.944 ms
14 janog14.miyazaki.miyazaki (2001:470:fa0a:114::2) 1.752 ms 0.728 ms 0.642 ms
15 janog15.kakegawa.shizuoka (2001:470:fa0a:115::2) 1.399 ms 1.68 ms 0.997 ms
16 janog16.fukuoka.fukuoka (2001:470:fa0a:116::2) 0.903 ms 0.773 ms 0.722 ms
17 janog17.sendai.miyagi (2001:470:fa0a:117::2) 1.566 ms 0.802 ms 0.891 ms
18 janog18.ariake.tokyo (2001:470:fa0a:118::2) 1.309 ms 1.271 ms 1.146 ms
19 janog19.naha.okinawa (2001:470:fa0a:119::2) 2.651 ms 1.967 ms 0.971 ms
20 janog20.obihiro.hokkaido (2001:470:fa0a:120::2) 1.695 ms 0.947 ms 1.312 ms
21 janog21.kumamoto.kumamoto (2001:470:fa0a:121::2) 1.102 ms 1.602 ms 0.875 ms
22 janog22.otemachi.tokyo (2001:470:fa0a:122::2) 2.769 ms 1.767 ms 0.948 ms
23 janog23.kochi.kochi (2001:470:fa0a:123::2) 1.383 ms 1.688 ms 1.302 ms
24 janog24.shinagawa.tokyo (2001:470:fa0a:124::2) 1.427 ms 2.268 ms 0.983 ms
25 janog25.niigata.niigata (2001:470:fa0a:125::2) 1.229 ms 1.858 ms 1.163 ms
26 janog26.ebisu.tokyo (2001:470:fa0a:126::2) 1.7 ms 1.737 ms 1.099 ms
27 janog27.kanazawa.ishikawa (2001:470:fa0a:127::2) 2.276 ms 1.964 ms 1.668 ms
28 janog28.nihonbashi.tokyo (2001:470:fa0a:128::2) 1.597 ms 2.884 ms 1.59 ms
29 janog29.wakayama.wakayama (2001:470:fa0a:129::2) 1.46 ms 2.408 ms 1.273 ms
30 janog30.kurashiki.okayama (2001:470:fa0a:130::2) 1.392 ms 1.287 ms 1.519 ms
31 janog31.roppongi.tokyo (2001:470:fa0a:131::2) 2.786 ms 1.367 ms 1.334 ms
32 janog32.osaka.osaka (2001:470:fa0a:132::2) 2.123 ms 2.016 ms 1.366 ms
33 janog33.beppu.oita (2001:470:fa0a:133::2) 1.915 ms 2.739 ms 1.856 ms
34 janog34.takamatsu.kagawa (2001:470:fa0a:134::2) 1.662 ms 2.71 ms 1.881 ms
35 janog35.shizuoka.shizuoka (2001:470:fa0a:135::2) 2.156 ms 1.761 ms 2.487 ms
36 janog36.kitakyushu.fukuoka (2001:470:fa0a:136::2) 2.252 ms 2.441 ms 2.268 ms
37 janog37.nagoya.aichi (2001:470:fa0a:137::2) 2.293 ms 3.528 ms 1.509 ms
38 janog38.naha.okinawa (2001:470:fa0a:138::2) 2.43 ms 1.744 ms 1.542 ms
39 janog39.kanazawa.ishikawa (2001:470:fa0a:139::2) 3.592 ms 2.436 ms 1.772 ms
40 traceroute.janog.gr.jp. (2001:470:fa0a:140::2) 2.973 ms 2.067 ms 2.449 ms
```

JANOG40への軌跡を表現

- 要件

- 40 hop 以上のtracerouteが可能
(=ネットワークセグメント)であること
- 逆引きDNSのptrが登録できること

検討のポイント

- IPアドレス
 - このIPv4在庫枯渇時代にIPv4アドレスが・・・
 - → IPv6を利用すればできるはず
- 逆引きDNS設定できるのか？
 - → 無料のIPv6トンネル接続でも逆引き委任を利用すればできるはず
- 大量のネットワークセグメントはどうするのか？
 - エミュレータ使っちゃえば？
 - なんか負けた気がする
 - 時代は仮想化じゃ
 - やれるだけやってみよう
- “traceroute”という皆知ってるけど実は差異が激しい部分はどうすんだ？
 - UDP 33434～、いや、ICMPじゃろ、などは今回のLTの対象外

大量のNWセグメント: 設計1 VRF

Virtual Routing and Forwarding (VRF)

=====

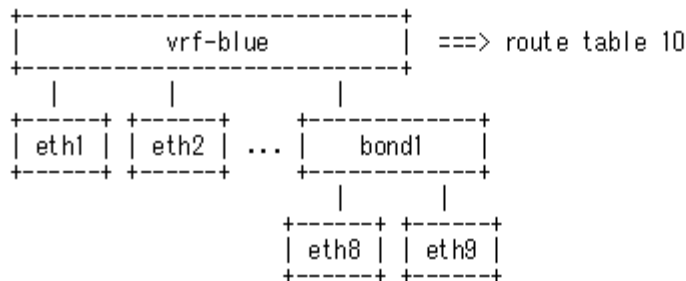
The VRF device combined with ip rules provides the ability to create virtual routing and forwarding domains (aka VRFs, VRF-lite to be specific) in the Linux network stack. One use case is the multi-tenancy problem where each tenant has their own unique routing tables and in the very least need different default gateways.

Processes can be "VRF aware" by binding a socket to the VRF device. Packets through the socket then use the routing table associated with the VRF device. An important feature of the VRF device implementation is that it impacts only Layer 3 and above so L2 tools (e.g., LLDP) are not affected (ie., they do not need to be run in each VRF). The design also allows the use of higher priority ip rules (Policy Based Routing, PBR) to take precedence over the VRF device rules directing specific traffic as desired.

In addition, VRF devices allow VRFs to be nested within namespaces. For example network namespaces provide separation of network interfaces at the device layer, VLANs on the interfaces within a namespace provide L2 separation and then VRF devices provide L3 separation.

Design

A VRF device is created with an associated route table. Network interfaces are then enslaved to a VRF device:



- 大量のVRFを切っても途中の経路がなぜかスルー

大量のネットワークセグメント：設計2多段VM

- Vynos メモリ128Mでも起動する！
 - 40個動かすぜ！
- メモリ8Gでも20VMで死亡・・・



大量のネットワークセグメント：設計3 netns

- namespaceを使えばいいじゃないか
 - 時代はコンテナじゃ！
- 様々なNWの検証にも使える
 - 作ってすぐ壊す
 - 壊してもすぐ作れる
 - 一般のご家庭でも検証にも最適！

RT01~02以下繰り返し設定

```
masa@home01:~/JANOG40$ cat create.sh
#!/bin/sh

SRC=`printf %02d $1`
DST=`printf %02d $2`

ip netns add RT$DST

ip link add RT$SRC-RT$DST type veth peer name RT$DST-RT$SRC

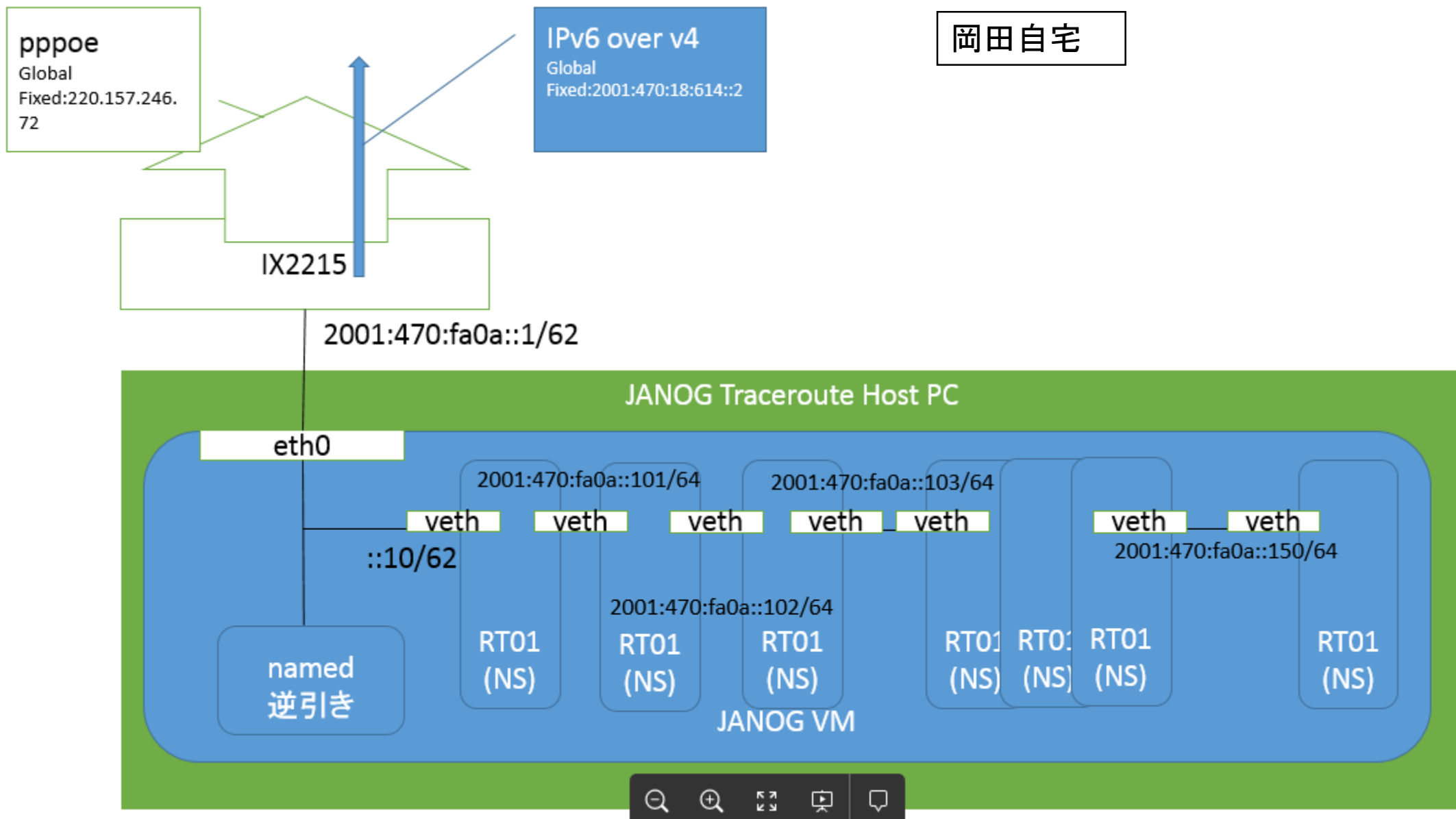
ip link set RT$SRC-RT$DST netns RT$SRC
ip link set RT$DST-RT$SRC netns RT$DST

ip netns exec RT$SRC ifconfig RT$SRC-RT$DST add 2001:470:fa0a:1$DST::1/64
ip netns exec RT$SRC ifconfig RT$SRC-RT$DST up
ip netns exec RT$DST ifconfig RT$DST-RT$SRC add 2001:470:fa0a:1$DST::2/64
ip netns exec RT$DST ifconfig RT$DST-RT$SRC up
ip netns exec RT$DST sysctl net.ipv6.conf.all.forwarding=1

ip netns exec RT$DST route -A inet6 add default gw 2001:470:fa0a:1$DST::1

for i in `seq -w `expr $2 + 1` 50`;
do
ip netns exec RT$SRC route -A inet6 add 2001:470:fa0a:1$i::/64 gw 2001:470:fa0a:1$DST::2
done

masa@home01:~/JANOG40$ for i in `seq -w 1 50`;do ./create.sh $i `expr $i + 1`;done
```



さあ、Traceroute to Janog!

- ここにグラフとかICMPの数をリアルタイムで載せます