仮想化環境でflowの検証

Playing flow in virtualized environment

荒井則之(BBIX)
maoke@bbix.net

JANOG36 Kita-Kyushyu
July 17 2005

フロー運用検証のハードル
Hurdle in verifying flow operation

- フロー運用検証の複雑性 Verification of flow operation is complex
  - 実運用のネットワーク構造 … due to the complexity of real network itself
    - Multi-router, multi-AS, multi-homing, etc.
  - フロー関連の仕組み … and the organization of flow components
    - Probe/exporter, collector, viewer, etc.
- 検証という目的とコストの不対称 … Costly verification is not desired
  - 試験専用環境の構築にお金がかかるし、時間も費やす
    It costs much time and money to build a lab with real routers, switches and flow equipment.
  - HW環境の変更や再構築はしにくい
    … and the hardware-based lab is hard to change or to reconfigure.

↓

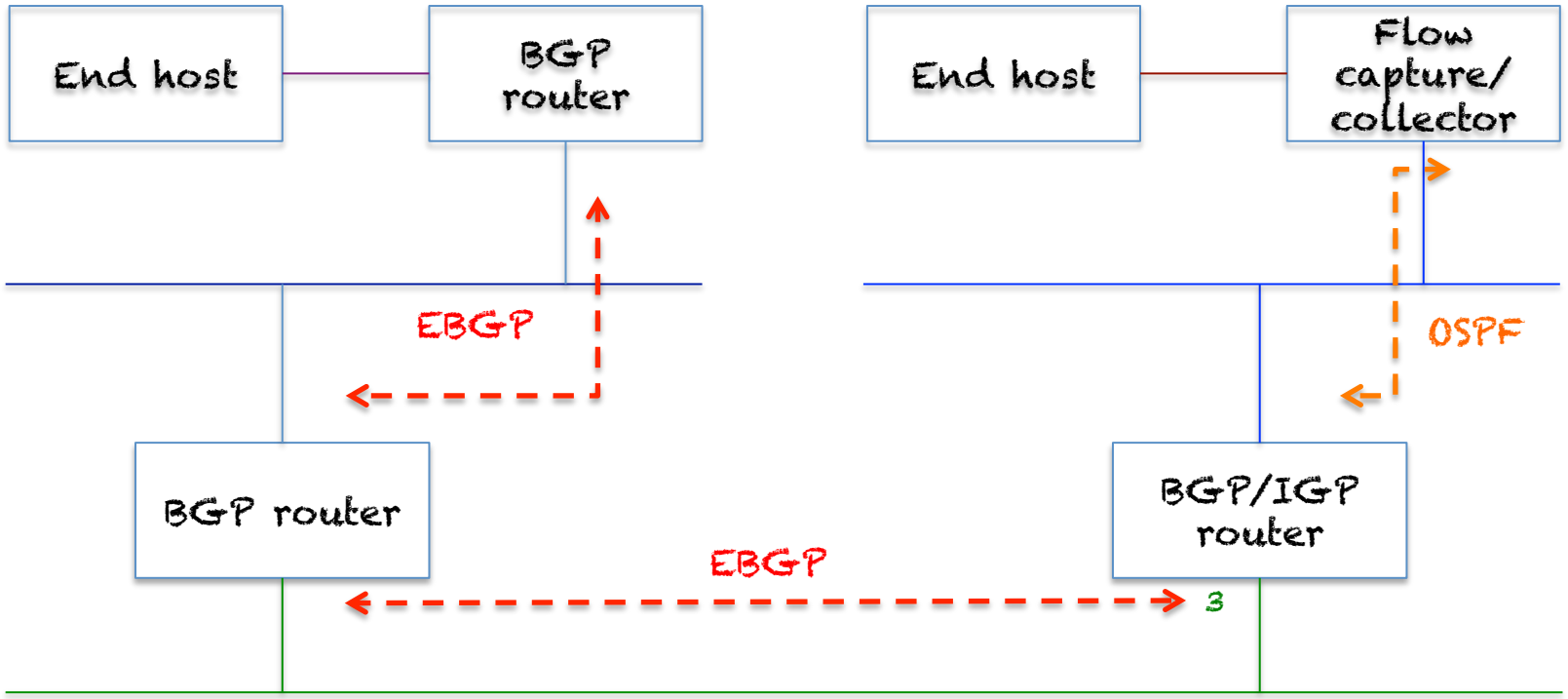
- 欲しいフロー運用の検証環境 Criteria for the flow verification environment
  - コストが低い Low cost
  - 構築しやすい Easy deployment
  - カスタマイズ Customizable, able to hack
  - 速い構築、速い再構築 Quick prototyping, fast reconstruction

フロー検証の要件
Necessary features for flow experiment

- トラフィックの発生 Traffic generation
  - End hosts
- フローの採取 Flow data collection
  - Probe
  - Collector
- ルーティング Routing facility
  - BGP Router
  - AS paths with with at least 2 AS-hops
  - IGP too

フロー検証の最小環境例

A minimum verification platform



材料 Materials

- 仮想化環境！Virtualization
  - VMware, VirtualBox
  - KVM, Xen, …
  - OpenVZ, docker, …
    - そもそもdockerは仮想ホスティングの目的ではないが
      … even though the docker was not originally designed for virtual hosting but isolation
- オープンソース！Open source
  - ソフトルータsoftware router：quagga, bird
  - フローツールflow tool：pmacct
  - トラフィック発生traffic generator：D-ITG

豆知識 Some tricks

- Virtualboxについて about virtualbox
  - Default NICは一つ、NATタイプ
    NIC is only one in default and working for NAT
    - 触らないほうがいい let it be
      - インターネットへの入り口となり、ファイルやソフトの取得の際に利用がある we need to install or transfer something from the Internet, when that default NAT helps
  - 追加NIC、別VMと繋げるように
    Make another NIC in order to connect with other VM siblings
    - 「host-only interface」という
    - VirtualBox Managerでしか追加できず host-only interface should be added through the VirtualBox Manager
      - VM停止状態で作業することが必要 … therefore it could be done only when the VM is stopped

豆知識 Some tricks

- Dockerについて about docker
  - Ubuntu 14.x or 15.x にインストールするなら
    Installing in Ubuntu 14+
    - 「apt-get install docker」も「apt-get install docker.io」もやらなくて
      … don’t rely on the apt-get install
    - Docker online documentsに参照すべき
      Follow the instruction in the docker online doc!
  - Default以外のネットワーキング手段について
    about non-default networking settings
      - cf. https://docs.docker.com/articles/networking/
    - IPv6
      - --ipv6 --fixed-cidr-v6=... for DOCKER_OPT of docker daemon
    - コンテナ内にはforwardingを0から1に変更できず
      - --privileged for container creation
    - コンテナ内tcpdumpは変なエラーが出ることがある
      - --storage-driver=devicemapper for DOCKER_OPT of docker daemon
      ✳ tsharkなら、devicemapperが必須ではない

コンテナ間のネットワーキング
Inter-container networking

- 直接にホストのNAT機能を利用すること
  docker host NAT is applicable for regular use
  - --net=bridge
  - Default gateway: 172.17.42.1/16

- ホストに関わらず、コンテナ自分のインターフェイスを立ち上げること
  host-independent interfaces are available for containers
  - --net=none
  - どう利用できるの？ How could we use the interfaces

Network Namespace

- dockerツールでnamespaceを触り、コンテナを繋げよう Apply docker tools to manipulate the namespaces of containers, int[illegible]necting them
  - namespace情報を取得 Obtain the namespace i[illegible]
    - docker ps
    - docker inspect -f '{{.State.Pid}}' [illegible]
    - ln -s /proc/<pid>/ns/net /var/[illegible]pid>
  - peer-to-peerリンクを作り、[illegible]aceに配布する Make peer-to peer link ar[illegible]namespaces
    - ip link add <ifa[illegible]l peer name <iface2>
    - ip link set [illegible]s <pid>
  - namesp[illegible]ーフェイスを設定する Set [illegible]erfaces through manipulating the namespaces directly
    - [illegible] exec <pid> ip addr <address>/<preflen> dev <iface>
    - [illegible]etns exec <pid> ip link <iface> up
    - ip netns exec <pid> ip route <address>/<preflen> dev <iface>

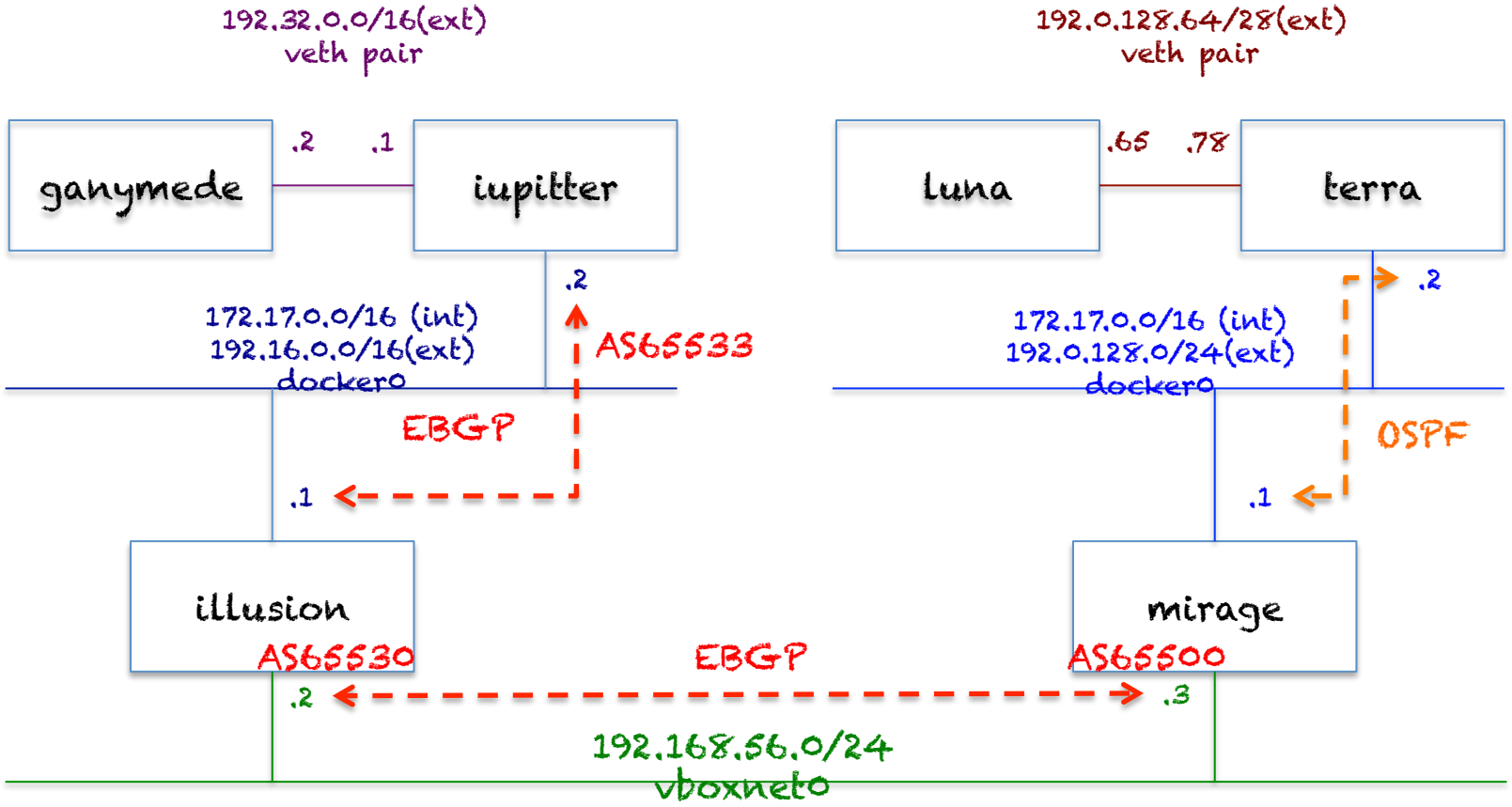
Woops, with restarting the container, its process id has changed. ☹

mkpeer

```
#!/bin/sh

CID_L=          # node1's container id
CID_P=          # node2's container id

PID_L=`docker inspect -f '{{.State.Pid}}'
$CID_L`
PID_P=`docker inspect -f '{{.State.Pid}}'
$CID_P`
IFACEL=         # node1's peer interface name
IFACEP=         # node2's peer interface name
ADDRL=          # node1's address
ADDRP=          # node2's address

if [ ! -d /var/run/netns ]; then
        mkdir -p /var/run/netns
fi

ln -s /proc/${PID_L}/ns/net /var/run/netns/$
{PID_L}
ln -s /proc/${PID_P}/ns/net /var/run/netns/$
{PID_P}
```

```
ip link add $IFACEL type veth peer name $IFACEP

ip link set $IFACEL netns $PID_L
ip link set $IFACEP netns $PID_P

ip netns exec $PID_L ip addr add $ADDRL/28 dev
$IFACEL
ip netns exec $PID_L ip link set $IFACEL up
ip netns exec $PID_L ip route add $ADDRL/28 dev
$IFACEL

ip netns exec $PID_P ip addr add $ADDRP/28 dev
$IFACEP
ip netns exec $PID_P ip link set $IFACEP up
ip netns exec $PID_P ip route add $ADDRP/28 dev
$IFACEP

ip netns exec $PID_P ip route add 0.0.0.0/0 via
$ADDRL  dev $IFACEP
```

例のサンプルネットワーク
The sample network



検証の実現 & LIVE SHOW

THINGS IN THE BOX

ありがとうございました

pmacctd as nf probe

```
!
daemonize: true
imt_path[inbound]:     /tmp/collect.pipe-eth0-in
imt_path[outbound]:    /tmp/collect.pipe-eth0-
out
imt_path[debug]: /tmp/collect.pipe-debug
pidfile:    /var/run/pmacctd.pid
logfile:    /var/log/pmacctd.log
interface:  eth0
!
pmacctd_net:      bgp
bgp_peer_src_as_type:  bgp
bgp_src_as_path_type:  bgp
aggregate[inbound]:    src_host, dst_host,
src_as, peer_src_as, peer_src_ip, src_as_path
aggregate[outbound]:   src_host, dst_host,
dst_as, peer_dst_as, peer_dst_ip, as_path
aggregate_filter[inbound]:   dst net
192.0.128.0/24
aggregate_filter[outbound]:  src net
192.0.128.0/24
aggregate[collect]:    src_host, dst_host,
src_port, dst_port, proto, tos
aggregate[debug]:src_host, dst_host, src_port,
dst_port, proto, tos
!
```

```
!
plugins:    memory[inbound], memory[outbound],
memory[debug], nfprobe[collect]
!
nfprobe_receiver:172.17.0.2:2100
nfprobe_source_ip:     172.17.0.2
nfprobe_version: 9
!
pmacctd_as: bgp
bgp_daemon: true
bgp_daemon_ip:    192.0.128.2
bgp_daemon_id:    192.0.128.2
bgp_agent_map:    /home/maoke/pmacct_work/maps/
agent_to_peer.map-v4-eth0
!
plugin_pipe_size:2000000
plugin_buffer_size:    10000
imt_mem_pools_number:  0
!
bgp_table_dump_file:   /tmp/bgp-$peer_src_ip.txt
bgp_table_dump_refresh_time: 300
!
! for debug only
nfprobe_timeouts[collect]:
      maxlife=60:expint=60
```

pmacct nfprobeのoutput例

```
~/pmacct_work$ pmacct -s -p /tmp/collect.pipe-eth0-in
SRC_AS      SRC_AS_PATH              PEER_SRC_AS PEER_SRC_IP                          SRC_IP
DST_IP                                           PACKETS             BYTES
0           ^$                       0             0                                192.168.56.3
192.0.128.2                                      9563                1898350
0           ^$                       0             0                                192.16.0.2
192.0.128.65                                     8                   672
0           ^$                       0             0                                192.168.56.2
192.0.128.2                                      4                   208
65530       65530                    65530         0                                192.16.0.2
192.0.128.65                                     38202               3740227
0           ^$                       0             0                                192.0.128.1
192.0.128.65                                     248                 12896
0           ^$                       0             0                                192.0.128.2
192.0.128.1                                      878                 120683
0           ^$                       0             0                                192.0.128.1
192.0.128.2                                      1497                98380
0           ^$                       0             0                                192.0.128.65
192.0.128.1                                      248                 25856
0           ^$                       0             0                                192.32.0.2
192.0.128.65                                     8                   672
65533       65530_65533              65530         0                                192.32.0.2
192.0.128.65                                     20854               2458290

For a total of: 10 entries
```

pmacct nfprobeのoutput例

```
~/pmacct_work$ pmacct -s -p /tmp/collect.pipe-eth0-out
DST_AS      AS_PATH                 PEER_DST_AS PEER_DST_IP                         SRC_IP
      DST_IP                                        PACKETS             BYTES
0           ^$                      0           0                                   192.0.128.65
      192.32.0.2                                    30                  4841
0           ^$                      0           0                                   192.0.128.2
      224.0.0.5                                     690                 47000
0           ^$                      0           0                                   192.0.128.1
      192.0.128.65                                  72                  5044
0           ^$                      0           0                                   192.0.128.2
      192.0.128.1                                   8                   526
0           ^$                      0           0                                   192.0.128.1
      192.0.128.2                                   10                  823
0           ^$                      0           192.0.128.1                         192.0.128.1
      192.0.128.2                                   842                 54216
0           ^$                      0           192.0.128.1                         192.0.128.65
      192.0.128.1                                   202                 21268
0           ^$                      0           0                                   192.0.128.65
      192.0.128.1                                   43                  4264
0           ^$                      0           0                                   192.0.128.2
      192.168.56.2                                  9                   828
0           ^$                      0           192.0.128.1                         192.0.128.2
      192.0.128.1                                   536                 88264
65533       65530_65533             65530       192.168.56.2                        192.0.128.65
      192.32.0.2                                    323                 22124
0           ^$                      0           192.0.128.1                         192.0.128.1
      192.0.128.65                                  303                 19964
0           ^$                      0           0                                   192.0.128.1
      224.0.0.5                                     690                 47064
0           ^$                      0           0                                   192.0.128.65
      192.168.56.2                                  12                  828

For a total of: 14 entries
```

nfacctd as collector

```
!
daemonize:     true
logfile:  /var/log/nfacctd.log
nfacctd_ip:    172.17.0.2
nfacctd_port:  2100
plugins:  memory[display]
!
nfacctd_net:    bgp
bgp_peer_src_as_type:   bgp
bgp_src_as_path_type:   bgp
!
aggregate[display]:      src_host,
dst_host, src_as, dst_as, peer_src_as,
peer_dst_as, as_path, src_as_path,
peer_src_ip, peer_dst_ip
!
!classifiers:  /home/maoke/pmacct_work/
maps/pretag.map-eth0
!
```

```
!
nfacctd_as_new:     bgp
bgp_daemon:    true
bgp_daemon_ip: 172.17.0.2
bgp_daemon_id: 172.17.0.2
bgp_agent_map: /home/maoke/pmacct_work/
maps/agent_to_peer.map-v4-eth0
!
!plugin_pipe_size:  2000000
!plugin_buffer_size:     10000
!imt_mem_pools_number:   0
!
bgp_table_dump_file:     /tmp/bgp-nfacctd-
$peer_src_ip.txt
bgp_table_dump_refresh_time:  300
!
```

nfacctd collectorのoutput例

```
~/pmacct_work$ pmacct -s -p /tmp/collect.pipe
SRC_AS      DST_AS      AS_PATH                  SRC_AS_PATH              PEER_SRC_AS PEER_DST_AS PEER_SRC_IP                    PEER_DST_IP
SRC_IP                                           DST_IP                                          PACKETS     BYTES
0           65533       65530_65533                ^$                     0           65530       172.17.0.2                 192.168.56.2
172.17.0.2                                       192.32.0.2                                      12          1008
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
fe80::5484:7aff:fefe:9799                        ff02::5                                         209         15884
0           0           ^$                         ^$                     0           0           172.17.0.2                 172.17.42.1
192.0.128.65                                     192.0.128.1                                     34          3536
0           0           ^$                         ^$                     0           0           172.17.0.2                 172.17.42.1
192.0.128.2                                      192.0.128.1                                     330         59865
0           65530       65530                      ^$                     0           65530       172.17.0.2                 192.168.56.2
192.0.128.65                                     192.16.0.2                                      2980        158870
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.2                                      192.0.128.1                                     225         22520
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.65                                     192.32.0.2                                      5           420
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.1                                      192.0.128.2                                     424         29740
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.2                                      192.168.56.3                                    2004        2231292
65530       0           ^$                         65530                  65530       0           172.17.0.2                 172.17.42.1
192.16.0.2                                       192.0.128.65                                    11          924
65533       0           ^$                         65530_65533            65530       0           172.17.0.2                 0
192.32.0.2                                       172.17.0.2                                      4           336
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.1                                      224.0.0.5                                       197         13432
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
172.17.42.1                                      172.17.0.2                                      64          4134
0           65533       65530_65533                ^$                     0           65530       172.17.0.2                 192.168.56.2
192.0.128.65                                     192.32.0.2                                      149         11622
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.2                                      224.0.0.5                                       196         13348
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.32.0.2                                       192.0.128.65                                    6           504
0           0           ^$                         ^$                     0           0           172.17.0.2                 172.17.42.1
192.0.128.1                                      192.0.128.65                                    34          1768
65533       0           ^$                         65530_65533            65530       0           172.17.0.2                 172.17.42.1
192.32.0.2                                       192.0.128.65                                    964         114610
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.0.128.65                                     192.16.0.2                                      6           504
0           0           ^$                         ^$                     0           0           172.17.0.2                 172.17.42.1
192.168.56.3                                     192.0.128.2                                     1494        314102
0           0           ^$                         ^$                     0           0           172.17.0.2                 172.17.42.1
192.0.128.1                                      192.0.128.2                                     494         32077
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.168.56.3                                     192.0.128.2                                     8           648
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
192.16.0.2                                       192.0.128.65                                    6           504
0           0           ^$                         ^$                     0           0           172.17.0.2                 0
172.17.0.2                                       172.17.42.1                                     35          2462

For a total of: 24 entries
```

グラフの作り

- pmacctでpipeから呼び出す
- RRDtoolで処理
- MySQLでリソース定義を記録され

- PNGグラフィック
- csv出力可能