
Subject: Complex routing and bridging with OpenVZ - further explanation

Posted by [Rob Wilson](#) on Fri, 22 Feb 2008 14:36:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

After my first email

(<http://openvz.org/pipermail/users/2008-February/001766.html>)

and Denis V. Luniv's corresponding reply, I thought a visual representation along with machine configuration output may help to explain things further.

For a more in depth explanation of why this configuration is desired see my previous email.

A diagram of the network setup:

<http://dust.cx/~rw/ovz-br.png>

To restate, the behavior I'm seeing is as follows:

- * HOST receives ping reply from VZ2
- * VZ2 receives ping reply from VZ3, VZ4 and HOST
- * VZ3 receives ping reply from VZ2 and VZ4
- * VZ4 receives ping reply from VZ2 and VZ3

Up until this point, all my tests succeed, the following behavior is what's odd:

HOST pings to VZ3 or VZ4 can be seen using tcpdump on the following interfaces:

- * HOST:br0 - echo REQUEST
- * VZ2:eth0 - echo REQUEST
- * VZ2:eth1 - echo REQUEST
- * ICMP echo REQUEST is NOT seen on 'br1'

VZ4 pings to HOST be seen on the following interfaces:

- * VZ4:eth0 - echo REQUEST
- * VZ3:eth1 - echo REQUEST
- * VZ3:eth0 - echo REQUEST
- * VZ2:eth1 - echo REQUEST
- * VZ2:eth0 - echo REQUEST
- * HOST:br0 - echo REQUEST
- * HOST:br0 - echo REPLY
- * VZ2:eth0 - echo REPLY
- * VZ2:eth1 - echo REPLY
- * ICMP echo REPLY is NOT seen on 'br1'

At the bottom of this email I've pasted the script that generates my configuration - make sure all interfaces are in state 'UP' with ``ip link show``.

I've also included output from 'ip address show' and 'ip route show' for the HOST, VZ2, VZ3 and VZ4.

I can also provide a LiveCD which this script can be run on to duplicate my exact configuration if necessary.

Thanks,

Rob

This script generates my configuration - make sure all interfaces are in state 'UP' with `ip link show`.

```
#!/bin/bash
#
# Test script for OpenVZ bridging configuration
#

# Add a bridge for each pair of machines
brctl addbr br0
brctl addbr br1
brctl addbr br2

# disable icmp redirects
echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/eth0/send_redirects

# recommended in bridge FAQ
echo 0 > /proc/sys/net/bridge/bridge-nf-call-arptables
echo 0 > /proc/sys/net/bridge/bridge-nf-call-iptables
echo 0 > /proc/sys/net/bridge/bridge-nf-filter-vlan-tagged

# Create OpenVZ instances and assign virtual ethernet devices
sh /OpenVZ/bin/vz.sh create 4
vzctl set 2 --netif_add eth0,00:60:00:00:01:01,veth2.0,00:60:00:00:02:01 --save
vzctl set 2 --netif_add eth1,00:60:00:00:01:02,veth2.1,00:60:00:00:02:02 --save
vzctl set 3 --netif_add eth0,00:60:00:00:01:03,veth3.0,00:60:00:00:02:03 --save
vzctl set 3 --netif_add eth1,00:60:00:00:01:04,veth3.1,00:60:00:00:02:04 --save
vzctl set 4 --netif_add eth0,00:60:00:00:01:05,veth4.0,00:60:00:00:02:05 --save
vzctl set 4 --netif_add eth1,00:60:00:00:01:06,veth4.1,00:60:00:00:02:06 --save
sh /OpenVZ/bin/vz.sh start

# Create the bridges
# Host and VZ2
# for [host:br0 vz2:eth0] bridge
brctl addif br0 br0 veth2.0

# VZ2 and VZ3
```

```
brctl addif br1 veth2.1 veth3.0
```

```
# VZ3 and VZ4
```

```
brctl addif br2 veth3.1 veth4.0
```

```
# Bring the bridges up
```

```
ip link set br0 up
```

```
ip link set br1 up
```

```
ip link set br2 up
```

```
# Make sure all virtual ethernet devices are up
```

```
ip link set veth2.0 up
```

```
ip link set veth2.1 up
```

```
ip link set veth3.0 up
```

```
ip link set veth3.1 up
```

```
ip link set veth4.0 up
```

```
ip link set veth4.1 up
```

```
# Give the Host an address on the same subnet as A and configure routing
```

```
# for [host:br0 vz2:eth0] bridge
```

```
ip a a 192.168.1.1/32 dev br0
```

```
ip r a 192.168.1.0/24 dev br0
```

```
ip r a 192.168.2.0/24 via 192.168.1.2 dev br0
```

```
ip r a 192.168.3.0/24 via 192.168.1.2 dev br0
```

```
# Configure VZ2
```

```
vzctl exec 2 ip a a 192.168.1.2/32 dev eth0
```

```
vzctl exec 2 ip a a 192.168.2.2/32 dev eth1
```

```
vzctl exec 2 ip link set eth0 up
```

```
vzctl exec 2 ip link set eth1 up
```

```
vzctl exec 2 ip r a 192.168.1.0/24 dev eth0
```

```
vzctl exec 2 ip r a 192.168.2.0/24 dev eth1
```

```
vzctl exec 2 ip r a 192.168.3.0/24 via 192.168.2.3 dev eth1
```

```
vzctl exec 2 ip r a 192.168.4.0/24 via 192.168.2.3 dev eth1
```

```
# Configure VZ3
```

```
vzctl exec 3 ip a a 192.168.2.3/32 dev eth0
```

```
vzctl exec 3 ip a a 192.168.3.2/32 dev eth1
```

```
vzctl exec 3 ip link set eth0 up
```

```
vzctl exec 3 ip link set eth1 up
```

```
vzctl exec 3 ip r a 192.168.2.0/24 dev eth0
```

```
vzctl exec 3 ip r a 192.168.3.0/24 dev eth1
```

```
vzctl exec 3 ip r a 192.168.1.0/24 via 192.168.2.2 dev eth0
```

```
vzctl exec 3 ip r a 192.168.4.0/24 via 192.168.3.3 dev eth1
```

```
# Configure VZ4
```

```
vzctl exec 4 ip a a 192.168.3.3 dev eth0
```

```
vzctl exec 4 ip a a 192.168.4.2 dev eth1
```

```
vzctl exec 4 ip link set eth0 up
vzctl exec 4 ip link set eth1 up
vzctl exec 4 ip r a 192.168.3.0/24 dev eth0
vzctl exec 4 ip r a 192.168.4.0/24 dev eth1
vzctl exec 4 ip r a 192.168.1.0/24 via 192.168.3.2 dev eth0
vzctl exec 4 ip r a 192.168.2.0/24 via 192.168.3.2 dev eth0
```

```
#
# HOST OUTPUT
#
```

```
HOST# ip a s
2: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
4: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP,10000> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:1a:92:05:2d:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.69/24 brd 192.168.0.255 scope global eth0
    inet6 fe80::21a:92ff:fe05:2d99/64 scope link
        valid_lft forever preferred_lft forever
6: br0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/32 brd 192.168.1.255 scope global br0
    inet6 fe80::200:ff:fe00:0/64 scope link
        valid_lft forever preferred_lft forever
8: br1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:02 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:202/64 scope link
        valid_lft forever preferred_lft forever
10: br2: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:04 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:204/64 scope link
        valid_lft forever preferred_lft forever
1: veth2.0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:01 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:201/64 scope link
        valid_lft forever preferred_lft forever
3: veth2.1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:02 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:202/64 scope link
        valid_lft forever preferred_lft forever
5: veth3.0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:03 brd ff:ff:ff:ff:ff:ff
```

```
inet6 fe80::260:ff:fe00:203/64 scope link
    valid_lft forever preferred_lft forever
7: veth3.1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:04 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:204/64 scope link
        valid_lft forever preferred_lft forever
9: veth4.0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:05 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:205/64 scope link
        valid_lft forever preferred_lft forever
11: veth4.1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:02:06 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::260:ff:fe00:206/64 scope link
        valid_lft forever preferred_lft forever
```

```
HOST# ip r s
192.168.3.0/24 via 192.168.1.2 dev br0
192.168.2.0/24 via 192.168.1.2 dev br0
192.168.1.0/24 dev br0  scope link
192.168.0.0/24 dev eth0  scope link
default via 192.168.0.1 dev eth0
```

```
#
# VZ2 OUTPUT
#
```

```
VZ2# ip a s
1: lo: <LOOPBACK> mtu 16436 qdisc noop
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/32 brd 192.168.1.255 scope global eth0
    inet6 fe80::260:ff:fe00:101/64 scope link
        valid_lft forever preferred_lft forever
5: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
    link/ether 00:60:00:00:01:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.2/32 brd 192.168.2.255 scope global eth1
    inet6 fe80::260:ff:fe00:102/64 scope link
        valid_lft forever preferred_lft forever
```

```
VZ2# ip r s
192.168.4.0/24 via 192.168.2.3 dev eth1
192.168.3.0/24 via 192.168.2.3 dev eth1
192.168.2.0/24 dev eth1  scope link
192.168.1.0/24 dev eth0  scope link
```

```
#
# VZ3 OUTPUT
```

#

VZ3# ip a s

```
1: lo: <LOOPBACK> mtu 16436 qdisc noop
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
   link/ether 00:60:00:00:01:03 brd ff:ff:ff:ff:ff:ff
   inet 192.168.2.3/32 brd 192.168.2.255 scope global eth0
   inet6 fe80::260:ff:fe00:103/64 scope link
       valid_lft forever preferred_lft forever
5: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
   link/ether 00:60:00:00:01:04 brd ff:ff:ff:ff:ff:ff
   inet 192.168.3.2/32 brd 192.168.3.255 scope global eth1
   inet6 fe80::260:ff:fe00:104/64 scope link
       valid_lft forever preferred_lft forever
```

VZ3# ip r s

```
192.168.4.0/24 via 192.168.3.3 dev eth1
192.168.3.0/24 dev eth1  scope link
192.168.2.0/24 dev eth0  scope link
192.168.1.0/24 via 192.168.2.2 dev eth0
```

#

VZ4 OUTPUT

#

VZ4# ip a s

```
1: lo: <LOOPBACK> mtu 16436 qdisc noop
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
   link/ether 00:60:00:00:01:05 brd ff:ff:ff:ff:ff:ff
   inet 192.168.3.3/32 brd 192.168.3.255 scope global eth0
   inet6 fe80::260:ff:fe00:105/64 scope link
       valid_lft forever preferred_lft forever
5: eth1: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc noqueue
   link/ether 00:60:00:00:01:06 brd ff:ff:ff:ff:ff:ff
   inet 192.168.4.2/32 brd 192.168.4.255 scope global eth1
   inet6 fe80::260:ff:fe00:106/64 scope link
       valid_lft forever preferred_lft forever
```

VZ4# ip r s

```
192.168.4.0/24 dev eth1  scope link
192.168.3.0/24 dev eth0  scope link
192.168.2.0/24 via 192.168.3.2 dev eth0
192.168.1.0/24 via 192.168.3.2 dev eth0
```
