Subject: Connecting VM eth0 to guest tap device
Posted by HRogge on Thu, 05 Apr 2007 11:39:33 GMT
View Forum Message <> Reply to Message

Hello,

I'm looking for a way to connect a simulated eth0 device inside the virtual machine with a tap device on the guest computer.

I'm using NS2 to emulate a WLAN connection and NS2 was connected with tap devices to UML (User Mode Linux) VMs. Unfortunately UML has performance problems so I would like to try the same with OpenVZ.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by Vasily Tarasov on Fri, 06 Apr 2007 09:00:57 GMT
View Forum Message <> Reply to Message

May be this can help you?

Thanks,
Vasily.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Fri, 06 Apr 2007 19:56:01 GMT
View Forum Message <> Reply to Message

Vasily Tarasov wrote on Fri, 06 April 2007 11:00May be this can help you?

Thanks,
Vasily.

????

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Sat, 07 Apr 2007 09:54:10 GMT
View Forum Message <> Reply to Message

Not sure what Vasiliy has meant
but, OVZ supports tun/tap inside and outside VE.
veth is just a bridge between VE and VE0.
So there should be no problem is your configuration.
Probably, if you expect more detailed answer you have to provide much more details on your configuration.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 10 Apr 2007 07:08:54 GMT
View Forum Message <> Reply to Message

Okay, I will try to be more specific.

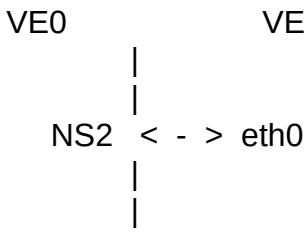
I'm working on a project with a simulated multihop WLAN network (http://en.wikipedia.org/wiki/Mobile_ad-hoc_network).

I'm using NS-2 to emulate the OSI-Layers 1 and 2 of a WLAN network. NS-2 opens a number of tap devices to communicate with real TCP/IP network stacks.

I need a way to start a OpenVZ instance that has a simulated network device inside (eth0 for example) that just connects to the tap device on the real computer so that NS-2 gets any traffic which is send to the eth0 device inside the OpenVZ instance.

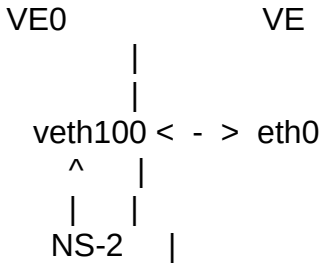
Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 10 Apr 2007 07:37:02 GMT
View Forum Message <> Reply to Message

Am I correct that you want the following configuration:




i.e. you want VE eth0 device to work via NS2 in VE0?

If so, then you need to setup veth device in VE with veth pair in VE0 and connect the pair to NS2 like this:



http://wiki.openvz.org/Virtual_Ethernet_device

---

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 10 Apr 2007 08:39:53 GMT
View Forum Message <> Reply to Message

Exactly... unfortunately I cannot connect NS2 to anything, NS2 creates a tap device and I have to use it...

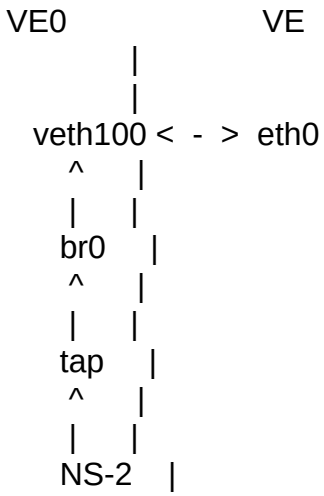
so I would need a way to put anything (including broadcasts) from the eth0 device in VE0 into the tap device and anything out of the tap device into the eth0. This traffic must not be limited by the mac/ip number of the eth0 device.

maybe I can do it the other way... if I allow VE to access a tap device on VE0 (found some doc about it in the wiki), can I tell VE to use this tap as a network device similar to eth0 ?

---

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 10 Apr 2007 08:48:55 GMT
View Forum Message <> Reply to Message

So you can simply bridge veth and tap devices like this:




No, tap cannnot be granted to VE (moved), since it's another pair should be at the same VE.
I suppose bridging is what you need, i.e. you need to add both veth100 and tap to bridge br0.

---

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 10 Apr 2007 09:10:45 GMT
View Forum Message <> Reply to Message

Can I use the virtual bridge in a "hub" modus ? So that anyone connected to the bridge gets all traffic (not limited by IP adress) ?

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 10 Apr 2007 09:37:30 GMT
View Forum Message <> Reply to Message

bridge works on MAC level, not IP.
if it doesn't know where to send packets to then it does broadcast.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Mon, 16 Apr 2007 13:10:27 GMT
View Forum Message <> Reply to Message

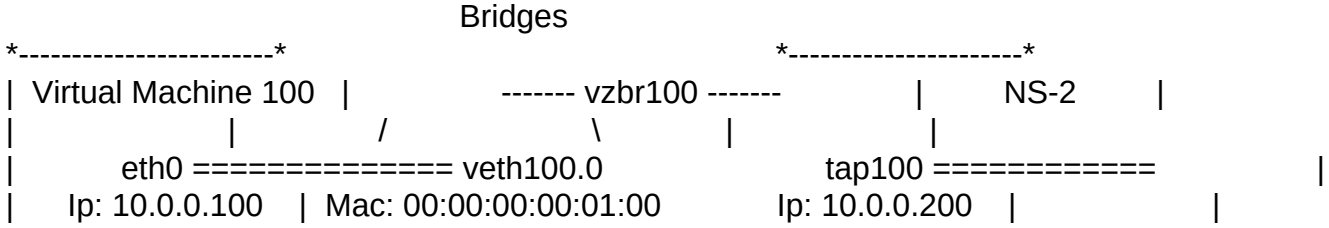
I just did a few experiments with a pair of bridges each connecting a OpenVZ instance with a tap device. By using tcpdump I dicovered a problem:

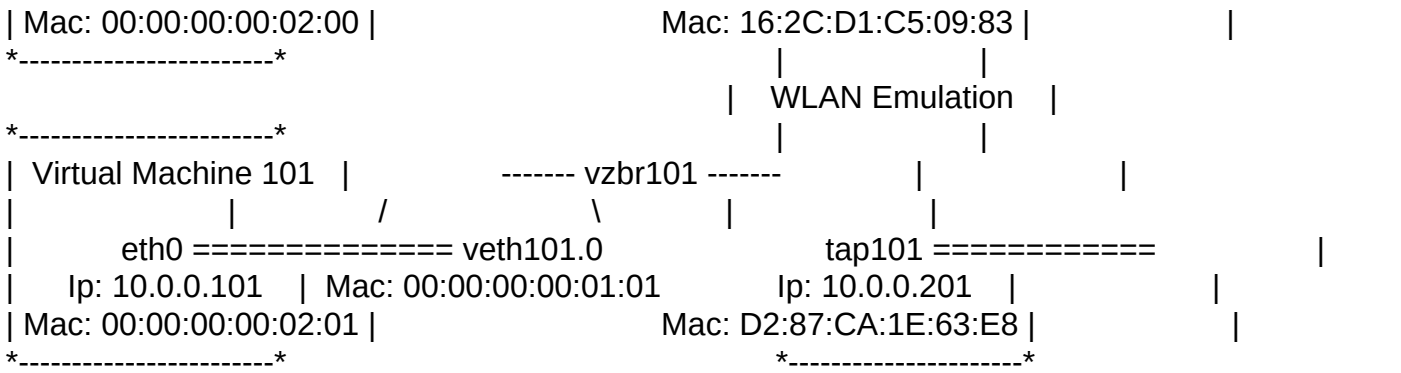
- the bridges receive broadcasts from the VMs and transmit them into the tap devices.
- the packets are received on the tap devices, processed and retransmitted through the other tap device as a broadcast (I can see both VMs on each of the tap devices with tcp dump)

- but the retransmitted packages are not send back through the bridge to the VMs ! (I tried tcpdump on the bridges, on the veth devices on the host and on the eth devices on the VM, I only see on VM in the output !)

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 06:32:03 GMT
View Forum Message <> Reply to Message

This is the network I'm working on:






- VM 100 is sending UDP broadcasts:

vn100:/# tcpdump -e -i eth0
06:23:05.415945 00:00:00:00:02:00 (oui Ethernet) > Broadcast, ethertype IPv4 (0x0800), length 62: 10.0.0.100.698 > 10.0.0.255.698: UDP, length 20

- I can see this packages on veth100.0 on the host-machine:
host:/# tcpdump -e -i veth100.0
08:25:32.584301 00:00:00:00:02:00 (oui Ethernet) > Broadcast, ethertype IPv4 (0x0800), length 62: 10.0.0.100.698 > 10.0.0.255.698: UDP, length 20

- I can see them on the bridge vzbr100:
host:/# tcpdump -e -i vzbr100
08:26:48.403584 00:00:00:00:02:00 (oui Ethernet) > Broadcast, ethertype IPv4 (0x0800), length 62: 10.0.0.100.698 > 10.0.0.255.698: UDP, length 20

- The packages enter the NS2 emulation through tap100:
host:/# tcpdump -e -i tap100
08:27:43.959279 00:00:00:00:02:00 (oui Ethernet) > Broadcast, ethertype IPv4 (0x0800), length 62: 10.0.0.100.698 > 10.0.0.255.698: UDP, length 20

- They are leaving the emulation through tap101:
host:/# tcpdump -e -i tap101
08:28:37.466638 00:00:00:00:02:00 (oui Ethernet) > Broadcast, ethertype IPv4 (0x0800), length 62: 10.0.0.100.698 > 10.0.0.255.698: UDP, length 20

- But they do NOT appear on the second bridge vzbr101
- They don't reach veth101.0
- And they don't reach the eth device of VM 101

Maybe this informations will help you to understand my configuration and my problem.

The same happens when I start the broadcasting on VM 101. They packages go through NS2 and don't appear on bridge vzbr100.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 17 Apr 2007 08:53:53 GMT
View Forum Message <> Reply to Message

do these packets go out vzrb101 interface in this situation:
host# tcpdump -e -i vzbr101
?

you can also install a printk() in br_flood() to diagnose whether bridge flooding is called at all in this case and if not then need to debug callers.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 09:22:04 GMT
View Forum Message <> Reply to Message

dev wrote on Tue, 17 April 2007 10:53do these packets go out vzrb101 interface in this situation:
host# tcpdump -e -i vzbr101
?
No, tcpdump on vzbr101, veth101.0 or eth (inside VM101) don't show any packets.

Quote:you can also install a printk() in br_flood() to diagnose whether bridge flooding is called at all in this case and if not then need to debug callers.

Just a "printk("Flooding active.\n");" ?

(sorry, I'm no linux kernel hacker )

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 17 Apr 2007 09:33:49 GMT
View Forum Message <> Reply to Message

which means that bridge dropped the packets at all.
this should not happen normally...

how to handle it?
1. try
# brctl setfd <bridge> 0

2. try to disable STP:
# brctl stp <bridge> off

3. yes, you can try to install printk() in appropriate function in kernel. it is quite easy

check http://wiki.openvz.org/Kernel_build for details.

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 10:35:16 GMT
View Forum Message <> Reply to Message

dev wrote on Tue, 17 April 2007 11:33which means that bridge dropped the packets at all.
this should not happen normally...

how to handle it?
1. try
# brctl setfd  <bridge> 0

No effect.

[quote]2. try to disable STP:
# brctl stp <bridge> off[/qote]
STP is off... (and it does not work with STP enabled)

Quote:3. yes, you can try to install printk() in appropriate function in kernel. it is quite easy

check http://wiki.openvz.org/Kernel_build for details.

I will try... (and will come back with results in an hour).

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 10:59:01 GMT
View Forum Message <> Reply to Message

I tried to add

```
* called under bridge lock */
static void br_flood(struct net_bridge *br, struct sk_buff *skb, int clone,
        void (*__packet_hook)(const struct net_bridge_port *p,
                              struct sk_buff *skb))
{
        struct net_bridge_port *p;
        struct net_bridge_port *prev;

+         printk("Flood !\n");
```

to net/bridge/br_forward.c but got nothing in the system log (dmesg).

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 17 Apr 2007 11:05:20 GMT

ok, then try to move it to br_handle_frame()
in case it gets triggered - don't forget to print bridge port name and device where skb arrived
(p->br->dev->name and skb->dev->name)

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 12:04:04 GMT


Okay, please hit me with something... I activated the wrong kernel during the last test.

So here is the dmesg output with BOTH printk.

NS2 and VM100/101 is starting:
Quote:device tap100 entered promiscuous mode
device tap101 entered promiscuous mode
VE: 100: started
device veth100.0 entered promiscuous mode
vzbr100: port 2(veth100.0) entering learning state
vzbr100: port 1(tap100) entering learning state
br_flood: vzbr100 vzbr100
VE: 101: started
br_flood: vzbr100 vzbr100
device veth101.0 entered promiscuous mode
vzbr101: port 2(veth101.0) entering learning state
vzbr101: port 1(tap101) entering learning state
br_flood: vzbr101 vzbr101
br_flood: vzbr100 vzbr100
br_flood: vzbr101 vzbr101
br_handle_frame: vzbr100 veth100.0
br_flood: vzbr101 vzbr101
br_handle_frame: vzbr100 veth100.0
br_handle_frame: vzbr101 veth101.0
br_handle_frame: vzbr100 veth100.0
br_handle_frame: vzbr101 veth101.0
br_handle_frame: vzbr100 veth100.0
br_handle_frame: vzbr101 veth101.0
br_flood: vzbr100 vzbr100
br_flood: vzbr100 vzbr100
br_flood: vzbr101 vzbr101
br_flood: vzbr101 vzbr101
br_handle_frame: vzbr100 veth100.0
br_handle_frame: vzbr101 veth101.0
br_handle_frame: vzbr101 veth101.0

br_flood: vzbr100 vzbr100
vzbr100: no IPv6 routers present
br_flood: vzbr101 vzbr101
veth100.0: no IPv6 routers present
veth101.0: no IPv6 routers present
vzbr101: no IPv6 routers present
br_handle_frame: vzbr100 veth100.0
br_handle_frame: vzbr101 veth101.0
eth0: no IPv6 routers present
eth0: no IPv6 routers present
vzbr100: topology change detected, propagating
vzbr100: port 2(veth100.0) entering forwarding state
vzbr100: topology change detected, propagating
vzbr100: port 1(tap100) entering forwarding state
vzbr101: topology change detected, propagating
vzbr101: port 2(veth101.0) entering forwarding state
vzbr101: topology change detected, propagating
vzbr101: port 1(tap101) entering forwarding state

UDP broadcast traffic is activated on VM 100:
Quote:br_handle_frame: vzbr100 veth100.0
br_flood: vzbr100 veth100.0
br_handle_frame: vzbr100 veth100.0
br_flood: vzbr100 veth100.0
br_handle_frame: vzbr100 veth100.0
br_flood: vzbr100 veth100.0
br_handle_frame: vzbr100 veth100.0
br_flood: vzbr100 veth100.0
br_handle_frame: vzbr100 veth100.0
br_flood: vzbr100 veth100.0
....

---

Subject: Re: Connecting VM eth0 to guest tap device
Posted by dev on Tue, 17 Apr 2007 12:24:34 GMT
View Forum Message <> Reply to Message

You also need to insert printk() in veth_xmit() and add more output like UDP src/dst addresses.
So you will be able to track how packet goes in kernel and where it is lost.

---

Subject: Re: Connecting VM eth0 to guest tap device
Posted by HRogge on Tue, 17 Apr 2007 13:05:25 GMT

I tried something like this but got a kernel panic:

```
static int veth_xmit(struct sk_buff *skb, struct net_device *dev)
{
	struct net_device_stats *stats;
	struct net_device *rcv = NULL;
	struct veth_struct *entry;
	int length;

	const unsigned char *dest = eth_hdr(skb)->h_dest;
	printk("veth_xmit dst: %02x:%02x:%02x:%02x:%02x:%02x\n",
		(int)(dest[0]),(int)(dest[1]),(int)(dest[2]),(int)(dest[3]),(int)(dest[4]),(int)(dest[5]));
```