

---

Subject: IPv6 native strangeness

Posted by [broquea](#) on Wed, 12 Nov 2008 23:59:30 GMT

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

---

I'm BACK!

So lets start with what the HN is etc:

Ubuntu 8.04 64bit server

Kernel = 2.6.24-21-openvz

{EDIT} Forgot to mention I use VENET not VETH interfaces on all OpenVZ machines involved

Symptoms:

Using native IPv6, I cannot allocate addresses out of the same /64 I use on the HN. I find that I have to route a second /64 allocation through the first in order for the VEs to get native connectivity. This smells like IPv6 forwarding is not working correctly, except it does work if there is a static route in place on the router. This is not the case with my CentOS 5.2 based OpenVZ machines, only Ubuntu.

Meaning:

2001:470:0:c8::1/64 is on the router's interface

2001:470:0:c8::2/64 is configured on eth0 on the server

sysctl is configured correctly to allow IPv6 packet forwarding

2001:470:0:c8::3/64 (through the end of the range) cannot be reached when configured on a VE on the Ubuntu system. It can on CentOS.

However when I statically route a second allocation (in this case 2001:470:0:c9::/64) through 2001:470:0:c8::2/64 on my router and I assign 2001:470:0:c9::1/64 (through the end of the range) to a VE, IPv6 forwarding seems to work without issue.

Notes:

Funny enough, and by no means am I comparing one virtualization platform to another, but another Ubuntu 8.04 64bit machine running Xen and a Xen kernel (2.6.24-21-xen) this bridging for IPv6 seems to work without the need for any static routes.

For reference the CentOS 5.2 OpenVZ machine I have running uses kernel

2.6.18-92.1.13.el5.028stab059.3. On THIS OpenVZ machine, I can allocate addresses out of a single /64 for the VEs, and not resort to statically routing allocations through each other.

THE QUESTION:

I'm looking for any insight into why the current Ubuntu kernel for OpenVZ behaves differently from either an older OpenVZ kernel, or a same rev Xen kernel with regards to IPv6 packet forwarding and bridging.

---