Subject: Re: MDNS service discovery with VETH interface
Posted by dev on Mon, 04 Jun 2007 14:31:25 GMT
View Forum Message <> Reply to Message

Daniel Pittman wrote:
> G'day. I have a VE that I set up using a VETH interface rather than a
> VENET interface so I could test ZeroConf service discovery and other
> broadcast/multicast protocols.
>
> I have the VETH device bridged with the physical Ethernet device on the
> hardware node and can see standard broadcast packets without any
> problem.
>
> I don't see the multicast packets cross the interface boundary though;
> it is my belief that the VETH device doesn't correctly handle the
> packets being sent to the Ethernet MAC '01:00:5e:00:00:fb'
>
> This is running kernel '2.6.18-028stab033.1-ovz'
>
>
> Have I missed some critical step in getting multicast services working
> with the VETH device or so?
>
>
> I think, reading the veth.c source, that the problem is in the veth_xmit
> function on line 305 in the current HEAD in get -- the code tests:
>
> if (!is_broadcast_ether_addr(((struct ethhdr *)skb->data)->h_dest)) {
>
> This should probably also pass a multicast packet through to the VE, not
> just the broadcast packets. That way multicast traffic is seen inside
> the system.
>
> The appropriate test would be 'is_multicast_ether_addr(...)' in addition
> to the broadcast.
>
> I have not supplied a patch here because I don't know what, if any,
> extra implications permitting multicast packets in have -- or if there
> is some other mechanism that I have missed that would cause them to pass
> through as expected.

It should be ok to add this check. Thanks for noticing.
patch attached and commited.

Thanks,
Kirill

```
--- ./drivers/net/veth.c.ve2378 2007-06-04 18:26:11.000000000 +0400
+++ ./drivers/net/veth.c 2007-06-04 18:26:16.000000000 +0400
@@ -302,7 +302,7 @@ static int veth_xmit(struct sk_buff *skb
  if (unlikely(rcv->owner_env->disable_net))
   goto outf;
  /* Filtering */
- if (!is_broadcast_ether_addr(((struct ethhdr *)skb->data)->h_dest)) {
+ if (!is_multicast_ether_addr(((struct ethhdr *)skb->data)->h_dest)) {
   if (ve_is_super(dev->owner_env) &&
     !ve_is_super(rcv->owner_env) &&
     !veth_from_netdev(rcv)->allow_mac_change) {
```