
Subject: Re: [PATCH] Virtual ethernet tunnel (v.2)
Posted by [Pavel Emelianov](#) on Thu, 07 Jun 2007 16:04:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Greear wrote:

> Pavel Emelianov wrote:

>> Ben Greear wrote:

>>

>>> Pavel Emelianov wrote:

>>>

>>>> Veth stands for Virtual ETHernet. It is a simple tunnel driver
>>>> that works at the link layer and looks like a pair of ethernet
>>>> devices interconnected with each other.

>>>>

>>> As Dave mentioned, there is already a driver known as 'veth'. Maybe
>>> borrow

>>> the etun name as well?

>>>

>>

>> We have already seen that this driver uses ethXXX names for
>> its devices and Dave agreed with veth one. Moreover Alexey
>> Kuznetsov said that he would prefer the name veth for etun.

>>

> Ok, fine by me. I started reading mail from the wrong direction this
> morning :)

>>

>>> I would also like some way to identify veth from other device types,
>>> preferably

>>> something like a value in sysfs. However, that should not hold up

>>>

>>

>> We can do this with ethtool. It can get and print the driver name of
>> the device.

>>

> I think I'd like something in sysfs that we could query for any
> interface. Possible return

> strings could be:

> VLAN

> VETH

> ETH

> PPP

> BRIDGE

> AP /* wifi access point interface */

> STA /* wifi station */

>

>

> I will cook up a patch for consideration after veth goes in.

OK.

>>> I think you need at least the option to zero out the time-stamp,
>>> otherwise it will
>>> not be re-calculated when received on the peer, and it potentially spent
>>> significant
>>> time since it was last calculated (think netem delay or similar).

>>>

```
>>> + /* Zero out the time-stamp so that receiving code is forced  
>>> + * to recalculate it.  
>>> + */  
>>> + skb->tstamp.off_sec = 0;  
>>> + skb->tstamp.off_usec = 0;
```

>>>

>>>

```
>>>> +  
>>>> + rcv_priv = netdev_priv(rcv);  
>>>> + skb->pkt_type = PACKET_HOST;  
>>>> + skb->protocol = eth_type_trans(skb, rcv);  
>>>> + if (dev->features & NETIF_F_NO_CSUM)  
>>>> +     skb->ip_summed = rcv_priv->ip_summed;  
>>>> +  
>>>> + dst_release(skb->dst);  
>>>> + skb->dst = NULL;  
>>>> + secpath_reset(skb);  
>>>> + nf_reset(skb);  
>>>> + skb->mark = 0;  
>>>> +  
>>>> + length = skb->len;
```

>>>>

>>> This should be done before you do the eth_type_trans, as that pulls the
>>> header and your
>>> byte counters will be off.

>>>

>>

>> This will be ETH_HLEN larger, do you mean this? I think this is
>> normal as this device tries to look like an "iron" ethernet card :)

>>

> For device counters, it should count the number of bytes received,
> including all headers,
> but excluding the ethernet FCS. If an 'iron' card did differently, I'd
> consider it a bug.

Hmm... The loopback must be doing bad things then. It first calls
eth_type_trans and then accounts for the new skb->len.

> Thanks,
> Ben

>

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
