
Subject: Re: [PATCH] Virtual ethernet tunnel

Posted by [Patrick McHardy](#) on Mon, 11 Jun 2007 11:39:18 GMT

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

Pavel Emelianov wrote:

> Patrick McHardy wrote:

>

>>>+ 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);

>>

>>

>>Is skb->mark supposed to survive communication between different

>>namespaces?

>

>

> I guess it must not. Thanks.

I guess there are a few others that should be cleared as well,

like the tc related members, secmark, ipvs_property, ...

>>The rtnl_link codes looks fine. I don't like the VETH_INFO_MAC attribute

>>very much though, we already have a generic device attribute for MAC

>>addresses. Of course that only allows you to supply one MAC address, so

>>I'm wondering what you think of allocating only a single device per

>>newlink operation and binding them in a seperate enslave operation?

>

>

> I did this at the very first version, but Alexey showed me that this

> would be wrong. Look. When we create the second device it must be in

> the other namespace as it is useless to have them in one namespace.

> But if we have the device in the other namespace the RTNL_NEWLINK

> message from kernel would come into this namespace thus confusing ip

> utility in the init namespace. Creating the device in the init ns and

> moving it into the new one is rather a complex task.

>

> But with such approach the creation looks really logical. We send a

> packet to the kernel and have a single response about the new device

> appearance. At the same time we have a RTNL_NEWLINK message arrived at

> the destination namespace informing that a new device has appeared
> there as well.

The question is how to proceed. I haven't read all mails yet, but it seems there is some disagreement about whether to create all devices in the same namespace and move them later or create them directly in their target namespace. For now I guess it doesn't matter much, so can everyone agree to adding a IFLA_PARTNER attribute that includes a complete ifinfo msg and the attributes and you later decide how to handle namespaces?

```
>>>+enum {
>>>+ VETH_INFO_UNSPEC,
>>>+ VETH_INFO_MAC,
>>>+ VETH_INFO_PEER,
>>>+ VETH_INFO_PEER_MAC,
>>>+
>>>+ VETH_INFO_MAX
>>>+};
>>
>>Please follow the
>>
>>#define VETH_INFO_MAX (__VETH_INFO_MAX - 1)
>>
>>convention here.
>
>
> Could you please clarify this point. I saw the lines
> enum {
> ...
> RTNL_NEWLINK
> #define RTNL_NEWLINK RTNL_NEWLINK
> ...
> }
> and had my brains exploded imagining what this would mean :(
```

That's just to make the new attributes visible as preprocessor symbols so userspace can use them for #ifdefs. We usually use it when adding new attributes/message types, but it's not necessary for the initial set of attributes if you already have some other preprocessor-visible symbol (like VETH_INFO_MAX) userspace can use.

What I was referring to is this convention:

```
enum {
```

```
...  
__IFLA_MAX  
};
```

```
#define IFLA_MAX (__IFLA_MAX - 1)
```

Which is used to make sure that IFLA_MAX is really the max and not max + 1 and additionally people won't forget to update it.

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