

---

Subject: Re: [PATCH] Virtual ethernet tunnel  
Posted by [Pavel Emelianov](#) on Wed, 13 Jun 2007 09:24:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Patrick McHardy wrote:

> 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, ...

It seems like we are about to have some `skb_reset_all()` routine to  
make the skb look like newborn.

>>> 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 separate `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

The agreement was that we can make any of the above. We can create  
both devices in the init namespace and then move one of them into the  
desired namespace, or we can explicitly specify which namespace to create  
the pair 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 ifinfomsg 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 :(

>  
>

> Thats 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 its 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 refering 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.

OK thanks. This is already done in the v2.

Thanks,  
Pavel

---

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

---