Subject: Re: [PATCH] Virtual ethernet tunnel
Posted by Daniel Lezcano on Thu, 07 Jun 2007 14:42:09 GMT
View Forum Message <> Reply to Message

Kirill Korotaev wrote:
> Deniel,
>
> Daniel Lezcano wrote:
>
>> Pavel Emelianov wrote:
>>
>>
>>>>>> 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.
>>>>>
>>>>>
>>>>>
>>>> Pavel,
>>>>
>>>> moving the netdevice to another namespace is not a complex task. Eric
>>>> Biederman did it in its patchset (cf. http://lxc.sf.net/network)
>>>>
>>>>
>>> By saying complex I didn't mean that this is difficult to implement,
>>> but that it consists (must consist) of many stages. I.e. composite.
>>> Making the device right in the namespace is liter.
>>>
>>>
>>>
>>>
>> When the pair device is created, both extremeties are into the init
>> namespace and you can choose to which namespace to move one extremity.
>>
>>
> I do not mind that.
>
>
>
>> When the network namespace dies, the netdev is moved back to the init
>> namespace.
>> That facilitate network device management.
>>
>> Concerning netlink events, this is automatically generated when the

>>>> network device is moved through namespaces.
>>>>
>>>> IMHO, we should have the network device movement between namespaces in
>>>> order to be able to move a physical network device too (eg. you have 4
>>>> NIC and you want to create 3 containers and assign 3 NIC to each of them)
>>>>
>>>>
>>> Agree. Moving the devices is a must-have functionality.
>>>
>>> I do not mind making the pair in the init namespace and move the second
>>> one into the desired namespace. But if we *always* will have two ends in
>>> different namespaces what to complicate things for?
>>>
>>>
>> Just to provide a netdev sufficiently generic to be used by people who
>> don't want namespaces but just want to do some network testing, like Ben
>> Greear does. He mentioned in a previous email, he will be happy to stop
>> redirecting people to out of tree patch.
>>
>> https://lists.linux-foundation.org/pipermail/containers/2007-April/004420.html
>>
>
> no one is against generic code and ability to create 2 interfaces in *one* namespace.
> (Like we currently allow to do so in OpenVZ)
>
> However, believe me, moving an interface is a *hard* operation. Much harder then netdev
> register from the scratch.
>
> Because it requires to take into account many things like:
> - packets in flight which requires synchronize and is slow on big machines
> - asynchronous sysfs entries registration/deregistration from
>   rtln_unlock -> netdev_run_todo
> - name/ifindex collisions
> - shutdown/cleanup of addresses/routes/qdisc and other similar stuff
>
>
All of what you are describing is already implemented in the Eric's
patchset.

You can have a look at :

http://lxc.sourceforge.net/patches/2.6.20/2.6.20-netns1/broken_out/

And more precisly:

for sysfs issues:
http://lxc.sourceforge.net/patches/2.6.20/2.6.20-netns1/broken_out/0065-sysfs-Shadow-directory-support.patch

for network device movement:
http://lxc.sourceforge.net/patches/2.6.20/2.6.20-netns1/broken_out/0096-net-Implment-network-device-movement-between-namesp.patch

Thanks,
Daniel

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