

---

Subject: Re: Re: [PATCH 2/5] net: Make rtnetlink infrastructure network namespace aware

Posted by [dlunev](#) on Sun, 30 Sep 2007 13:13:15 GMT

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

---

Hmm, so it looks like we do not need this queue processing at all...

Regards,  
Den

Eric W. Biederman wrote:

> Patrick McHardy <kaber@trash.net> writes:

>

>> Maybe I can save you some time: we used to do down\_trylock()  
>> for the rtnl mutex, so senders would simply return if someone  
>> else was already processing the queue \*or\* the rtnl was locked  
>> for some other reason. In the first case the process already  
>> processing the queue would also process the new messages, but  
>> if it the rtnl was locked for some other reason (for example  
>> during module registration) the message would sit in the  
>> queue until the next rtnetlink sendmsg call, which is why  
>> rtnl\_unlock does queue processing. Commit 6756ae4b changed  
>> the down\_trylock to mutex\_lock, so senders will now simply wait  
>> until the mutex is released and then call netlink\_run\_queue  
>> themselves. This means its not needed anymore.

>

> Sounds reasonable.

>

> I started looking through the code paths and I currently cannot  
> see anything that would leave a message on a kernel rtnl socket.

>

> However I did a quick test adding a WARN\_ON if there were any messages  
> found in the queue during rtnl\_unlock and I found this code path  
> getting invoked from linkwatch\_event. So there is clearly something I  
> don't understand, and it sounds at odds just a bit from your  
> description.

>

> If we can remove the extra queue processing that would be great,  
> as it looks like a nice way to simplify the locking and the special  
> cases in the code.

>

> Eric

---

Containers mailing list

[Containers@lists.linux-foundation.org](mailto:Containers@lists.linux-foundation.org)

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---