Subject: Re: [PATCH 2/5] net: Make rtnetlink infrastructure network namespace aware
Posted by Patrick McHardy on Sat, 29 Sep 2007 15:44:35 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> void rtnl_unlock(void)
> {
> - mutex_unlock(&rtnl_mutex);
> - if (rtnl && rtnl->sk_receive_queue.qlen)
> + struct net *net;
> +
> + /*
> + * Loop through all of the rtnl sockets until none of them (in
> + * a live network namespace) have queue packets.
> + *
> + * We have to be careful with the locking here as
> + * sk_data_ready aka rtnetlink_rcv takes the rtnl_mutex.
> + *
> + * To ensure the network namespace does not exit while
> + * we are processing packets on it's rtnl socket we
> + * grab a reference to the network namespace, ignoring
> + * it if the network namespace has already exited.
> + */
> +retry:
> + for_each_net(net) {
> + struct sock *rtnl = net->rtnl;
> +
> + if (!rtnl || !rtnl->sk_receive_queue.qlen)
> + continue;
> +
> + if (!maybe_get_net(net))
> + continue;
> +
> + mutex_unlock(&rtnl_mutex);
>   rtnl->sk_data_ready(rtnl, 0);
> + mutex_lock(&rtnl_mutex);
> + put_net(net);
> + goto retry;
> + }
> + mutex_unlock(&rtnl_mutex);
> +
>   netdev_run_todo();
> }

I'm wondering why this receive queue processing on unlock is still
necessary today, we don't do trylock in rtnetlink_rcv anymore, so

all senders will simply wait until the lock is released and then process the queue.

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