Subject: Re: [PATCH 1/5] net: Modify all rtnetlink methods to only work in the initial namespace
Posted by ebiederm on Thu, 11 Oct 2007 17:08:51 GMT
View Forum Message <> Reply to Message

"Denis V. Lunev" <den@sw.ru> writes:

>> Grr. That last sentence should have been I do not see a need for the more
>> fundamental change you seem to be advocating.
>
> why?

Because I do not see the need. There are certainly details that can be improved, but you seem to be talking about ripping everything out, ignoring the reviews and the acceptance that has happened and starting from scratch all over again.

I don't see the need to start again from scratch.

> veth is not mainly affected, if it is connected to a bridge there will
> be no problem. You are talking about decision to switch/choose VE on
> level 2, I am told about layer 3, i.e. at the moment of routing/based on
> IP. So,
>
> I do not understand how this will help me in my usecase. The macvlan
> device transmit the packet into real device. OK. But it does not help if
> I want to setup router in one namespace for another.
>
> Basically, queue stop will not help, as routing namespace can have two
> interfaces, one fast and one slow. In your case we should stop accepting
> the message based on the slowest one?

You are proposing a solution that only works for namespaces, while the problem continues to exist in all other routing cases. Solving this in a namespace specific way seems to reduce the value of using namespaces for testing weird aspects of the networking stack. And except for OpenVZ this is not a common case for anyone yet, so it seems a seriously premature optimization. Given that it could only be one path having something like ECN for UDP to slow the transmitter down would be nice.

The point of the macvlan example is that I believe that will be our common case when things network namespaces are deployed in the real world. Both because macvlans are easier to setup then routing or bridging, and because they perform better.

Further this problem feels like an application bug to me. Going from a fast to a slow network when you are saturating the fast network

with UDP packets will cause packets to be dropped. If this is a
problem the application should slow down it's packet transmission rate
or at least wait for an ACK periodically.

> how much registers do you have on i386 for this? the call found in my
> original letter already has 6. Additionally, we can be shot by embedded
> people arguing about 1 more argument and additional not needed for them
> code.... And namespace code becomes unremovable one by usual ifdef
> way.

Mucking with data on the skb has all kinds of potential for slowing down the fast path of the networking stack, and if the networking stack is slow the embedded people won't even use it. So as much as I am sympathetic to preventing code size growth, maintenance and performance are higher priorities. Especially when you are advocating growth in the fast path, while I am only advocating growth in the slow path.

As for i386 stack accesses are optimized almost as well as registers, so function parameters are still not a bad deal.

>From the first couple of rounds of review the clear message was:
Maintainable and comprehensible code and don't touch the fast path.

If to achieve the above I have to use a solution that distresses the embedded people my apologies.

Eric