Subject: Re: [RFC PATCH 0/31] An introduction and A path for merging network namespace work

Posted by ebiederm on Wed, 07 Mar 2007 04:53:27 GMT

View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Eric W. Biederman wrote:
>
> [cut]
>> Dmitry? Daniel? What do you think.
>>
>
> Hi Eric,
>
> I agree with all the points you presented but I am still 50/50 for both
> approaches.
>
> The major argument in favor of the explicit parameter is that it allows
> to keep track of the network namespace. But the argument against is it
> touchs a lot of files and that can makes the patch less attractive.
> Furthermore, everybody should not be aware of what a network namespace
> is and should not know how to handle the parameter function.

Daniel please look at the patches and see how this interacts. What
you describe is how sight unseen one would expect the situation to be
however that doesn't seem to match the reality of the code.

Besides which a one time big impact is not a problem, for merging
code. It is more of a problem for maintaining out of tree patches.

> The implicit network namespace has the advantage to reduce considerably
> the impact on the code and to have network developer to be unware of the
> network virtualization. But in the same way the network developer should
> "forget" in which network namespace he is running. Another point is the
> race condition we have while doing network namespace switching and that
> can make a contention point.

Actually this is largely false. The implicit parameter does not do
more than remove a few patches. The bulk of the changes are the
fundamental changes like the arp cache, the routing tables etc. In
general all of the basic data structures.

> Concerning the network namespace compile out, that can be done by both
> approaches.

Agreed. My innovation was finding a way to compile out an explicit parameter.

> In the [1/31] patch description, you mention you tryed zero sized
> structure on x86_64, and the optimization works for all architectures.
> Does it mean, you tested it with s390, PowerPC, ia64, etc ... ?

I think my meaning was this: Every where I tested it (i386 (many
compiler versions) and x86_64) the parameter was completely optimized
out. And even if it isn't the code should still work. Further since
passing a void parameter is explicitly allowed in C++ in the right
circumstances I expect the code to work on all architectures.

> IMHO, both approaches are equivalent in terms of pros/cons. Perhaps we
> should ask netdev@ ...

Perhaps we should see if we can resolve it ourselves?

Anyway as soon as I get the stupid sysfs support fixed I'm going to
look at a veth/etun driver and see if we can get that merged.

Eric

_____