Subject: Re: [patch 2/6] [Network namespace] Network device sharing by view
Posted by Herbert Poetzl on Mon, 26 Jun 2006 21:26:16 GMT
View Forum Message <> Reply to Message

On Mon, Jun 26, 2006 at 02:37:15PM -0600, Eric W. Biederman wrote:
> Herbert Poetzl <herbert@13thfloor.at> writes:
>
> > On Mon, Jun 26, 2006 at 01:35:15PM -0600, Eric W. Biederman wrote:
> >> Herbert Poetzl <herbert@13thfloor.at> writes:
> >
> > yes, but you will not be able to apply policy on
> > the parent, restricting the child networking in a
> > proper way without jumping through hoops ...
>
> ? I don't understand where you are coming from.
> There is no restriction on where you can apply policy.

in a fully hierarchical system (not that I really think
this is required here) you would be able to 'delegate'
certain addresses (ranges) to a namespace (the child)
below the current one (the parent) with the ability to
limit/control the input/output (which is required for
security)

> >> I really do not believe we have a hotpath issue, if this
> >> is implemented properly. Benchmarks of course need to be taken,
> >> to prove this.
> >
> > I'm fine with proper testing and good numbers here
> > but until then, I can only consider it a prototype
>
> We are taking the first steps to get this all sorted out.
> I think what we have is more than a prototype but less then
> the final implementation. Call it the very first draft version.

what we are desperately missing here is a proper way
to testing this, maybe the network folks can come up
with some test equipment/ideas here ...

> >> There are only two places a sane implementation should show issues.
> >> - When the access to a pointer goes through a pointer to find
> >>   that global variable.
> >> - When doing a lookup in a hash table we need to look at an additional
> >>   field to verify a hash match. Because having a completely separate
> >>   hash table is likely too expensive.
> >>
> >> If that can be shown to really slow down packets on the hot path I am
> >> willing to consider other possibilities. Until then I think we are on

> >> path to the simplest and most powerful version of building a network
> >> namespace usable by containers.
> >
> > keep in mind that you actually have three kinds
> > of network traffic on a typical host/guest system:
> >
> > - traffic between unit and outside
> >   - host traffic should be quite minimal
> >   - guest traffic will be quite high
> >
> > - traffic between host and guest
> >   probably minimal too (only for shared services)
> >
> > - traffic between guests
> >   can be as high (or even higher) than the
> >   outbound traffic, just think web guest and
> >   database guest
>
> Interesting.
>
> >> The routing between network namespaces does have the potential to be
> >> more expensive than just a packet trivially coming off the wire into a
> >> socket.
> >
> > IMHO the routing between network namespaces should
> > not require more than the current local traffic
> > does (i.e. you should be able to achieve loopback
> > speed within an insignificant tolerance) and not
> > nearly the time required for on-wire stuff ...
>
> That assumes on the wire stuff is noticeably slower.
> You can achieve over 1GB/s on some networks.

well, have you ever tried how much you can achieve
over loopback :)

> But I agree that the cost should resemble the current
> loopback device. I have seen nothing that suggests
> it is not.
>
> >> However that is fundamentally from a lack of hardware. If the
> >> rest works smarter filters in the drivers should enable to remove the
> >> cost.
> >>
> >> Basically it is just a matter of:
> >> if (dest_mac == my_mac1) it is for device 1.
> >> If (dest_mac == my_mac2) it is for device 2.
> >> etc.

> >
> > hmm, so you plan on providing a different MAC for
> > each guest? how should that be handled from the
> > user PoV? you cannot simply make up MACs as you
> > go, and, depending on the network card, operation
> > in promisc mode might be slower than for a given
> > set (maybe only one) MAC, no?
>
> The speed is a factor certainly. As for making up
> macs. There is a local assignment bit that you can set.

well, local is fine, but you cannot utilize that
on-wire which basically means that you would have
either to 'map' the MAC on transmission (to the
real one) which would basically make the approach
useless, or to use addresses which are fine within
a certain range of routers ...

> With that set it is just a matter of using a decent random
> number generator. The kernel already does this is some places.

sure you can make up MACs, but you will never
be able to use them 'outside' the box

> >> At a small count of macs it is trivial to understand it will go
> >> fast for a larger count of macs it only works with a good data
> >> structure. We don't hit any extra cache lines of the packet,
> >> and the above test can be collapsed with other routing lookup tests.
> >
> > well, I'm absolutely not against flexibility or
> > full virtualization, but the proposed 'routing'
> > on the host effectively doubles the time the
> > packet spends in the network stack(s), so I can
> > not believe that this approach would not add
> > (significant) overhead to the hot path ...
>
> It might, but I am pretty certain it won't double
> the cost, as you don't do 2 full network stack traversals.

> And even at a full doubling I doubt it will affect bandwith
> or latency very much.

well, for loopback that would mean half the bandwidth
and twice the latency, no?

> If it does we have a lot more to optimize in the network stack than
> just this code.

why? duplicate stack traversal takes roughly twice
the time, or am I wrong here? if so, please enlighten
me ...

best,
Herbert

> Eric