Subject: Re: Network virtualization/isolation
Posted by ebiederm on Sun, 26 Nov 2006 20:52:14 GMT
View Forum Message <> Reply to Message

Herbert Poetzl <herbert@13thfloor.at> writes:

> On Sat, Nov 25, 2006 at 01:21:39AM -0700, Eric W. Biederman wrote:

```
>> There are two techniques in real use.
>> - Bind/Accept filtering
>>
>>   Which layer 3 addresses a socket can bind/accept are filtered,
>>   but otherwise the network stack remains unchanged. When your
>>   container/VE only has a single IP address this works great.
>
>>   When you get multiple IPs this technique starts to fall down because
>>   it is not obvious how to make this correctly handle wild card ip
>>   addresses.
>
> not really, you have to check for a (sub)set of IPs
> that's quite simple and not hard to get right, I agree
> that it increases the overhead on those checks, but
> this only occurs at bind/connect/accept time ...
```

The general problem is you get into mental model problems. You think
you are isolated but you don't realize you can route packets over the
loopback interface for example. But with care yes you can solve it.

However while I think there is value in this technique it doesn't
solve any of my problems, nor do I think it can be easily stretched
to solve my problems. My gut feel for implementation still says
this should be a new netfilter table that filters binds and accepts
if we implement this.

For most of us we need more power than we can get with the simple
bind/accept filtering so we I think the network namespace work should
concentrate on the general technique that gives us the entire power of
the current network stack. At least until we have proved the
overheads are unacceptable.

```
>> Given that performance is the primary concern this is something a
>> network stack expert might be able to help with.  My gut feel is
>> the extra pointer indirection for the more general technique is
>> negligible and will not affect the network performance.  The network
>> stack can be very sensitive to additional cache misses so I could be
>> wrong.  Opinions?
>
> well, here we are talking about layer2 _isolation_
```

> if I got that right, i.e. you split the physical
> interfaces up into separate network namespaces, which
> then can make full use of the assigned interfaces

Yes. Layer 2 isolation is a good description.

> this is something which I'm perfectly fine with, as
> I do not think it adds significant overhead (nevertheless
> it needs some testing)

Yes lots of testing and careful implementation.

> but at the same time, this is
> something which isn't very useful in the generic case,
> where folks will have, let's say two network interfaces
> and want to share one of them between 100 guests ...

It is useful in the generic case. It just requires being smart to
keep the overheads down.

> as the checks would require to identify the interface,
> that would immediately result in O(N) overhead for
> each packet received, plus the overhead added by
> disabling the hardware filters ... but maybe that
> changed over the years, I'm definitely no network
> stack/device expert ...

Getting this to O(log(N)) is easy, and you can probably
get the average case to O(1) without trying too hard. This
is no worse than routing tables or multiple IP addresses on
a single interface. Ben Greear has addressed this. His experience
suggest that even O(N) is not likely to be a significant problem.

Now I'm going to go bury my head in the sand for a bit. The hard
problems are not how do we reshape the network stack but how do we
get the appropriate context into all of our user space interfaces.

Eric