

---

Subject: Re: Network virtualization/isolation

Posted by [Daniel Lezcano](#) on Tue, 28 Nov 2006 14:15:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Eric W. Biederman wrote:

[ snip ]

>>

>> The packets arrive to the real device and go through the routes  
>> engine. From this point, the used route is enough to know to which  
>> container the traffic can go and the sockets subset assigned to the  
>> container.

>

> Note this has potentially the highest overhead of them all because  
> this is the only approach in which it is mandatory to inspect the  
> network packets to see which container they are in.

If the container is in the route information, when you use the route,  
you have the container destination with it. I don't see the overhead here.

>

> My real problem with this approach besides seriously complicating  
> the administration by not delegating it is that you loose enormous  
> amounts of power.

I don't understand why you say administration is more complicated.  
unshare -> ifconfig

1 container = 1 IP

[ snip ]

> So you have two columns that you rate these things that I disagree  
> with, and you left out what the implications are for code maintenance.

>

> 1) Network setup.

> Past a certainly point both bind filtering and Daniel's L3 use a new  
> paradigm for managing the network code and become nearly impossible for  
> system administrators to understand. The classic one is routing packets  
> between machines over the loopback interface by accident. Huh?

What is this new paradigm you are talking about ?

>

> The L2. Network setup iss simply the cost of setting up a multiple  
> machine network. This is more complicated but it is well understood  
> and well documented today. Plus for the common cases it is easy to

- > get a tool to automate this for you. When you get a complicated
- > network this wins hands down because the existing tools work and
- > you don't have to retrain your sysadmins to understand what is
- > happening.

unshare -> (guest) add mac address  
          (host) add mac address  
          (guest) set ip address  
          (host) set ip address  
          (host) setup bridge

1 container = 2 net devices (root + guest), 2 IPs, 2 mac addresses, 1 bridge.

100 containers = 200 net devices, 200 IPs, 200 mac addresses, 1 bridge.

- >
- > 2) Runtime Overhead.
- >
- > Your analysis is confused. Bind/Accept filter is much cheaper than
- > doing a per packet evaluation in the route cache of which container
- > it belongs to. Among other things Bind/Accept filtering allows all
- > of the global variables in the network stack to remain global and
- > only touches a slow path. So it is both very simple and very cheap.
- >
- > Next in line comes L2 using real network devices, and Daniel's
- > L3 thing. Because there are multiple instances of the networking data
- > structures we have an extra pointer indirection.

There is not extra networking data structure instantiation in the Daniel's L3.

- >
- > Finally we get L2 with an extra network stack traversal, because
- > we either need the full power of netfilter and traffic shaping
- > gating access to what a node is doing or we simply don't have
- > enough real network interfaces. I assert that we can optimize
- > the lack of network interfaces away by optimizing the drivers
- > once this becomes an interesting case.

- >
- > 3) Long Term Code Maintenance Overhead.

- >
- > - A pure L2 implementation. There is a big one time cost of
- > changing all of the variable accesses. Once that transition
- > is complete things just work. All code is shared so there
- > is no real overhead.

- >
- > - Bind/Connect/Accept filtering. There are so few places in
- > the code this is easy to maintain without sharing code with
- > everyone else.

For isolation too ? Can we build network migration on top of that ?

>  
> - Daniel's L3. A big mass of special purpose code with peculiar  
> semantics that no one else in the network stack cares about  
> but is right in the middle of the code.

Thanks Eric for all your comments.

-- Daniel

---

Containers mailing list  
Containers@lists.osdl.org  
<https://lists.osdl.org/mailman/listinfo/containers>

---