Subject: Re: strict isolation of net interfaces
Posted by serue on Fri, 30 Jun 2006 02:39:47 GMT
View Forum Message <> Reply to Message

Quoting Cedric Le Goater (clg@fr.ibm.com):
> Sam Vilain wrote:
> > jamal wrote:
> >>> note: personally I'm absolutely not against virtualizing
> >>> the device names so that each guest can have a separate
> >>> name space for devices, but there should be a way to
> >>> 'see' _and_ 'identify' the interfaces from outside
> >>> (i.e. host or spectator context)
> >>>
> >>>
> >> Makes sense for the host side to have naming convention tied
> >> to the guest. Example as a prefix: guest0-eth0. Would it not
> >> be interesting to have the host also manage these interfaces
> >> via standard tools like ip or ifconfig etc? i.e if i admin up
> >> guest0-eth0, then the user in guest0 will see its eth0 going
> >> up.
> >
> > That particular convention only works if you have network namespaces and
> > UTS namespaces tightly bound. We plan to have them separate - so for
> > that to work, each network namespace could have an arbitrary "prefix"
> > that determines what the interface name will look like from the outside
> > when combined. We'd have to be careful about length limits.
> >
> > And guest0-eth0 doesn't necessarily make sense; it's not really an
> > ethernet interface, more like a tun or something.
> >
> > So, an equally good convention might be to use sequential prefixes on
> > the host, like "tun", "dummy", or a new prefix - then a property of that
> > is what the name of the interface is perceived to be to those who are in
> > the corresponding network namespace.
> >
> > Then the pragmatic question becomes how to correlate what you see from
> > `ip addr list' to guests.
>
>
> we could work on virtualizing the net interfaces in the host, map them to
> eth0 or something in the guest and let the guest handle upper network layers ?
>
> lo0 would just be exposed relying on skbuff tagging to discriminate traffic
> between guests.

This seems to me the preferable way. We create a full virtual net
device for each new container, and fully virtualize the device
namespace.

```
> host                 |  guest 0  |  guest 1  |  guest2
>  ---------------------+-----------+-----------+------------ -
>   |                   |           |           |
>   |-> l0      <-------+-> lo0 ... | lo0       | lo0
>   |                   |           |           |
>   |-> bar0    <-------+-> eth0    |           |
>   |                   |           |           |
>   |-> foo0    <-------+-----------+-----------+-> eth0
>   |                   |           |           |
>   `-> foo0:1  <-------+-----------+-> eth0    |
>                       |           |           |
>
>
> is that clear ? stupid ? reinventing the wheel ?
```

The last one in your diagram confuses me - why foo0:1? I would have thought it'd be

```
host                 |  guest 0  |  guest 1  |  guest2
 ---------------------+-----------+-----------+------------ -
  |                   |           |           |
  |-> l0      <-------+-> lo0 ... | lo0       | lo0
  |                   |           |           |
  |-> eth0            |           |           |
  |                   |           |           |
  |-> veth0   <-------+-> eth0    |           |
  |                   |           |           |
  |-> veth1   <-------+-----------+-----------+-> eth0
  |                   |           |           |
  |-> veth2   <-------+-----------+-> eth0    |
```

I think we should avoid using device aliases, as trying to do something like giving eth0:1 to guest1 and eth0:2 to guest2 while hiding eth0:1 from guest2 requires some uglier code (as I recall) than working with full devices. In other words, if a namespace can see eth0, and eth0:2 exists, it should always see eth0:2.

So conceptually using a full virtual net device per container certainly seems cleaner to me, and it seems like it should be simpler by way of statistics gathering etc, but are there actually any real gains? Or is the support for multiple IPs per device actually enough?

Herbert, is this basically how ngnet is supposed to work?

-serge