

---

Subject: Re: Network virtualization/isolation  
Posted by [ebiederm](#) on Sat, 25 Nov 2006 19:26:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

"Leonid Grossman" <Leonid.Grossman@neterion.com> writes:

>  
>  
>> -----Original Message-----  
>> From: netdev-owner@vger.kernel.org  
>> [mailto:netdev-owner@vger.kernel.org] On Behalf Of Eric W. Biederman  
>  
>> Then the question is how do we reduce the overhead when we  
>> don't have enough physical network interfaces to go around.  
>> My feeling is that we could push the work to the network  
>> adapters and allow single physical network adapters to  
>> support multiple network interfaces, each with a different  
>> link-layer address. At which point the overhead is nearly  
>> nothing and newer network adapters may start implementing  
>> enough filtering in hardware to do all of the work for us.  
>  
> Correct, to a degree.  
> There will be always a limit on the number of physical "channels" that a  
> NIC  
> can support, while keeping these channels fully independent and  
> protected at the hw level.  
> So, you will probably still need to implement the sw path,  
> with the assumption that some containers (that care about performance)  
> will get a separate  
> NIC interface and avoid the overhead, and other containers will have to  
> use the sw path.  
> There are some multi-channel NICs shipping today so it would be possible  
> to see the overhead between the two options (I suspect it will be quite  
> noticeable), but for a general idea about what work could be pushed down  
> to network adapters in the near future you can look at the pcisig.com  
> I/O Virtualization Workgroup.  
> Once the single root I/O Virtualization spec is completed, it is likely  
> to be supported by several NIC vendors to provide multiple network  
> interfaces on a single NIC that you are looking for.

Pushing it all of the way into the hardware is an optimization, that while great is likely not necessary. Simply doing a table lookup by link-level address and selecting between several network interfaces is enough to ensure we only traverse the network stack once.

To keep overhead down in the container case I don't need the hardware support to be so good you can do kernel bypass and still trust that everything is safe. I simply a fast link-level address to container

mapping. We already look at the link-level address on every packet received so that should not generate any extra cache misses.

In the worst case I might need someone to go as far as the Grand Unified Lookup to remove all of the overheads. Except for distributing the work load more evenly across the machine with separate interrupts and the like I see no need for separate hardware channels to make things go fast for my needs.

Despite the title of this thread there is no virtualization or emulation of the hardware involved. Just enhancements to the existing hardware abstractions.

Eric

---

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

---