

---

Subject: Re: Network namespaces a path to mergable code.

Posted by [ebiederm](#) on Wed, 28 Jun 2006 18:11:24 GMT

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

---

Andrey Savochkin <[saw@swsoft.com](mailto:saw@swsoft.com)> writes:

>> In a slightly different vein your second patch introduced a lot  
>> of `#ifdef CONFIG_NET_NS` in C files. That is something we need to look closely  
>> at.

>>

>> So I think the abstraction that we use to access per network namespace  
>> variables needs some work if we are going to allow the ability to compile  
>> out all of the namespace code. The explicit versus implicit lookup is just  
>> one dimension of that problem.

>

> This is a good comment.

>

> Those `ifdef`'s mostly correspond to places where we walk over lists  
> and need to filter-out entities not belonging to a specific namespace.  
> Those places about the same in your and my implementation.  
> We can think what we can do with them.

> One trick that I used on several occasions is `net_ns_same` macro  
> which doesn't evaluate its arguments if `CONFIG_NET_NS` not defined,  
> and thus can be used without `ifdef`'s.

>

> Returning to implicit vs explicit function arguments, I believe that implicit  
> arguments are more promising in having zero impact on the code when  
> `CONFIG_NET_NS` is disabled.  
> Functions like `inet_addr_type` will translate into exactly the same code as  
> they did without net namespace patches.

Which brings us to a basic question. Does it make sense to have  
a define that completely disables namespace support.

I know all of the simple namespaces have been implemented like that,  
and it was relatively easy there. I'm not at all certain in the long  
term we want a configuration option. Especially if simply enabling  
the code doesn't have an impact on performance. Which I think is  
a merge requirement anyway.

As for `inet_addr_type` and friends I do agree that implicit arguments  
make for an easier implementation of `CONFIG_NET_NS`. My gut feel  
is though that the code with explicit arguments is probably more  
comprehensible in the long term. Especially as we find more weird  
exceptions where the process we are running in does not have the correct  
network namespace.

In general unnecessary `CONFIG` options are a problem because they make

the entire testing process much harder and make the code harder to write (so that both cases work and work cleanly).

So my feeling is that we actually want to kill all of those CONFIG\_XXX\_NS options.

Which simply leaves us with the problem of implementing the code cleanly.

Eric

---