

---

Subject: Re: [PATCH 1/12] L2 network namespace: current network namespace operations

Posted by [ebiederm](#) on Fri, 08 Dec 2006 20:50:51 GMT

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

---

Dmitry Mishin <dim@openvz.org> writes:

> Added functions and macros required to operate with current network namespaces.  
> They are required in order to switch network namespace for incoming packets and  
> to not extend current network interface by additional network namespace argue.

>  
> Signed-off-by: Dmitry Mishin <dim@openvz.org>

>  
> -#else  
> +#define current\_net\_ns (current->nsproxy->net\_ns)  
> +#else /\* CONFIG\_NET\_NS \*/

>  
> #define INIT\_NET\_NS(net\_ns)

>  
> @@ -57,6 +78,22 @@ static inline int copy\_net\_ns(int flags,  
> static inline void put\_net\_ns(struct net\_namespace \*ns)

> {  
> }

> -#endif

> +

> +#define current\_net\_ns NULL

> +#endif /\* !CONFIG\_NET\_NS \*/

Ouch! NULL is not a good default.

Can we please pick an idiom for referencing global network stack variables that works if we are compiled in or not. At least if we are going to offer the option.

That way we can merge the changes for looking up all of the globals before merging the network namespace support.

Doing it this way seems to imply we will need context support to implement this.

My initial suggestion is to base the work on the per cpu variable support.

Using `__get_net_var(variable)`. To reference the global variable. And the variables marked as `__per_net` in their declaration so we know the variables are per network namespace.

This allows us to handle ipv6 and other modules that only have their variables present when they are loaded the same way per cpu variables are treated. And it ensures that the form used when everything is

Eric

---

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

---