
Subject: Re: cleanup in workq and dst_destroy

Posted by [Daniel Lezcano](#) on Mon, 19 Nov 2007 08:46:28 GMT

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

Denis V. Lunev wrote:

> Daniel Lezcano wrote:

>> Hi all,

>>

>> while doing ipv6 namespace, we were faced to a problem with the loopback

>> and the dst_destroy function.

>>

>> When the network namespace exits, the cleanup function is called by

>> schedule_work and this function will browse the net ops list to call the

>> different exit methods for the registered subsystems.

>>

>> The different subsystems will shutdown their resources and in particular

>> addrconf subsystem will ifdown the loopback. This function will call

>> rt6_ifdown

>> -> fib6_clean_all

>> -> fib6_clean_node

>> -> fib6_clean_tree

>> -> fib6_clean_node

>> -> fib6_del

>> -> fib6_del_route

>> -> rt6_release

>> ->dst_free

>> -> __dst_free

>>

>> The __dst_free function will schedule_delayed_work the dst_gc_work

>> function.

>>

>> The dst_gc_work will call dst_destroy and finally this one will call

>> dst->ops->destroy ops function which is ip6_dst_destroy.

>>

>> The problem here is we have the workq blocked because we are running

>> inside the netns cleanup function. So the delayed work will not run

>> until we exits the cleanup function. But the loopback is still

>> referenced by the ip6 routes, the netdev_unregister will loop

>> indefinitely => dead lock.

>>

>> By the way, this bug appears with ipv6 but it is perhaps pending with ipv4.

>>

>> Benjamin as proposed to create a separate workq for the network

>> namespace, so in the worst case we have the unregister looping until the

>> ip6 route are shut downed. Is it an acceptable solution ?

>>

>

> we are doing this stuff in the special thread. There are a lot of

> difficult things to perform like synchronize_net & netdev_run_todo inside

The special thread ? do you mean keventd_wq ?

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
