

Subject: [patch 01/38][IPv6] net-2.6.25 - make netns cleanup to run in a separate queue

Posted by [Daniel Lezcano](#) on Mon, 03 Dec 2007 16:16:37 GMT

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

This patch adds a separate workqueue for cleaning up a network namespace. If we use the keventd workqueue to execute cleanup_net(), there is a problem to unregister devices in IPv6. Indeed the code that cleans up also schedule work in keventd: as long as cleanup_net() hasn't return, dst_gc_task() cannot run and as long as dst_gc_task() has not run, there are still some references pending on the net devices and cleanup_net() can not unregister and exit the keventd workqueue.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>

Acked-by: Denis V. Lunev <den@openvz.org>

Acked-By: Kirill Korotaev <dev@sw.ru>

net/core/net_namespace.c | 9 +++++++-
1 file changed, 8 insertions(+), 1 deletion(-)

Index: linux-2.6-netns/net/core/net_namespace.c

=====

--- linux-2.6-netns.orig/net/core/net_namespace.c

+++ linux-2.6-netns/net/core/net_namespace.c

@@ -58,6 +58,7 @@ out_undo:

```
#ifdef CONFIG_NET_NS
static struct kmem_cache *net_cachep;
+static struct workqueue_struct *netns_wq;

static struct net *net_alloc(void)
{
@@ -149,7 +150,7 @@ void __put_net(struct net *net)
{
/* Cleanup the network namespace in process context */
INIT_WORK(&net->work, cleanup_net);
- schedule_work(&net->work);
+ queue_work(netns_wq, &net->work);
}
EXPORT_SYMBOL_GPL(__put_net);

@@ -171,7 +172,13 @@ static int __init net_ns_init(void)
net_cachep = kmem_cache_create("net_namespace", sizeof(struct net),
SMP_CACHE_BYTES,
SLAB_PANIC, NULL);
+
+ /* Create workqueue for cleanup */
```

```
+ netns_wq = create_singlethread_workqueue("netns");
+ if (!netns_wq)
+   panic("Could not create netns workq");
+ #endif
+
+   mutex_lock(&net_mutex);
+   err = setup_net(&init_net);
```

--

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
