

---

Subject: [patch 3/3][NETNS45][V2] remove timewait sockets at cleanup  
Posted by [Daniel Lezcano](#) on Thu, 27 Sep 2007 11:04:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Daniel Lezcano <[dlezcano@fr.ibm.com](mailto:dlezcano@fr.ibm.com)>

Denis Lunev spotted that if we take a reference to the network namespace with the timewait sockets, we will need to wait for their expiration to have the network namespace freed. This is a waste of time, the timewait sockets are for avoiding to receive a duplicate packet from the network, if the network namespace is freed, the network stack is removed, so no chance to receive any packets from the outside world.

This patchset remove/destroy the timewait sockets when the network namespace is freed.

Signed-off-by: Daniel Lezcano <[dlezcano@fr.ibm.com](mailto:dlezcano@fr.ibm.com)>

---

[net/ipv4/tcp.c](#) | 53 ++++++  
1 file changed, 53 insertions(+)

Index: linux-2.6-netns/net/ipv4/tcp.c

=====

--- linux-2.6-netns.orig/net/ipv4/tcp.c

+++ linux-2.6-netns/net/ipv4/tcp.c

@@ -2432,8 +2432,61 @@ static int tcp\_net\_init(struct net \*net)

    return 0;

}

+/\*

+ \* Wipeout tcp timewait sockets, they are no longer needed

+ \* because we destroy the network namespace, so no risk to

+ \* have duplicate packet coming from the network

+ \*/

+static void tcp\_net\_exit(struct net \*net)

+

+ struct inet\_timewait\_sock \*tw;

+ struct sock \*sk;

+ struct hlist\_node \*node, \*tmp;

+ int h;

+ int nbsock = 0;

+

+ /\* Browse the the established hash table \*/

+ for (h = 0; h < (tcp\_hashinfo.ehash\_size); h++) {

+     struct inet\_ehash\_bucket \*head =

+         inet\_ehash\_bucket(&tcp\_hashinfo, h);

+

+ /\* Take the look and disable bh \*/

```

+ write_lock_bh(&head->lock);
+
+ sk_for_each_safe(sk, node, tmp, &head->twchain) {
+
+ tw = inet_twsk(sk);
+ if (tw->tw_net != net)
+ continue;
+
+ /* deschedule the timewait socket */
+ spin_lock(&tcp_death_row.death_lock);
+ if (inet_twsk_del_dead_node(tw)) {
+ inet_twsk_put(tw);
+ if (--tcp_death_row.tw_count == 0)
+ del_timer(&tcp_death_row.tw_timer);
+ }
+ spin_unlock(&tcp_death_row.death_lock);
+
+ /* remove it from the established hash table */
+ __inet_twsk_unehash(tw);
+
+ /* remove it from the bind hash table */
+ inet_twsk_unbhash(tw, tcp_death_row.hashinfo);
+
+ /* last put */
+ inet_twsk_put(tw);
+
+ nbsock++;
+
+ write_unlock_bh(&head->lock);
+
+}
+
+
static struct pernet_operations tcp_net_ops = {
    .init = tcp_net_init,
+ .exit = tcp_net_exit,
};

void __init tcp_init(void)

--
```

---

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

---