
Subject: [PATCH] Set a separate lockdep class for neighbour table's proxy_queue
Posted by [xemul](#) on Mon, 16 Apr 2007 12:03:52 GMT

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

Otherwise the following calltrace will lead to a wrong lockdep warning:

```
neigh_proxy_process()
  `- lock(neigh_table->proxy_queue.lock);
arp_redo /* via tbl->proxy_redo */
arp_process
neigh_event_ns
neigh_update
skb_queue_purge
  `- lock(neighbor->arp_queue.lock);
```

This is not a deadlock actually, as neighbor table's proxy_queue and the neighbor's arp_queue are different queues.

Lockdep thinks there is a deadlock as both queues are initialized with skb_queue_head_init() and thus have a common class.

```
--- a/include/linux/skbuff.h 2007-04-09 22:52:27.000000000 +0400
+++ b/include/linux/skbuff.h 2007-04-09 22:52:32.000000000 +0400
@@ -628,6 +628,13 @@ static inline void skb_queue_head_init(s
    list->qlen = 0;
}

+static inline void skb_queue_head_init_class(struct sk_buff_head *list,
+ struct lock_class_key *class)
+{
+ skb_queue_head_init(list);
+ lockdep_set_class(&list->lock, class);
+}
+
+/*
+ * Insert an sk_buff at the start of a list.
+ */
--- a/net/core/neighbour.c 2007-04-09 22:52:28.000000000 +0400
+++ b/net/core/neighbour.c 2007-04-09 22:53:01.000000000 +0400
@@ -1327,6 +1327,8 @@ void neigh_parms_destroy(struct neigh_pa
    kfree(parms);
}

+static struct lock_class_key neigh_table_proxy_queue_class;
+
void neigh_table_init_no_netlink(struct neigh_table *tbl)
{
```

```
    unsigned long now = jiffies;
@@ -1379,7 +1381,8 @@ void neigh_table_init_no_netlink(struct
    init_timer(&tbl->proxy_timer);
    tbl->proxy_timer.data = (unsigned long)tbl;
    tbl->proxy_timer.function = neigh_proxy_process;
- skb_queue_head_init(&tbl->proxy_queue);
+ skb_queue_head_init_class(&tbl->proxy_queue,
+ &neigh_table_proxy_queue_class);

    tbl->last_flush = now;
    tbl->last_rand = now + tbl->parms.reachable_time * 20;
```
