Subject: [PATCH][IPX] Use existing sock refcnt debugging infrastructure
Posted by Pavel Emelianov on Fri, 09 Nov 2007 13:42:05 GMT
View Forum Message <> Reply to Message

Just like in the af_packet.c, the ipx_sock_nr variable is used
for debugging purposes.

Switch to using existing infrastructure. Thanks to Arnaldo for
pointing this out.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index a195a66..c76a952 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -92,11 +92,6 @@ extern int ipxrtr_route_skb(struct sk_buff *skb);
 extern struct ipx_route *ipxrtr_lookup(__be32 net);
 extern int ipxrtr_ioctl(unsigned int cmd, void __user *arg);

-#undef IPX_REFCNT_DEBUG
-#ifdef IPX_REFCNT_DEBUG
-atomic_t ipx_sock_nr;
-#endif
-
 struct ipx_interface *ipx_interfaces_head(void)
 {
  struct ipx_interface *rc = NULL;
@@ -151,14 +146,7 @@ static void ipx_destroy_socket(struct sock *sk)
 {
  ipx_remove_socket(sk);
  skb_queue_purge(&sk->sk_receive_queue);
-#ifdef IPX_REFCNT_DEBUG
- atomic_dec(&ipx_sock_nr);
- printk(KERN_DEBUG "IPX socket %p released, %d are still alive\n", sk,
-   atomic_read(&ipx_sock_nr));
- if (atomic_read(&sk->sk_refcnt) != 1)
-  printk(KERN_DEBUG "Destruction sock ipx %p delayed, cnt=%d\n",
-    sk, atomic_read(&sk->sk_refcnt));
-#endif
+ sk_refcnt_debug_dec(sk);
  sock_put(sk);
 }

@@ -1384,11 +1372,8 @@ static int ipx_create(struct net *net, struct socket *sock, int protocol)
  sk = sk_alloc(net, PF_IPX, GFP_KERNEL, &ipx_proto);
```

```
  if (!sk)
   goto out;
-#ifdef IPX_REFCNT_DEBUG
- atomic_inc(&ipx_sock_nr);
- printk(KERN_DEBUG "IPX socket %p created, now we have %d alive\n", sk,
-   atomic_read(&ipx_sock_nr));
-#endif
+
+ sk_refcnt_debug_inc(sk);
  sock_init_data(sock, sk);
  sk->sk_no_check = 1;  /* Checksum off by default */
  sock->ops = &ipx_dgram_ops;
@@ -1409,6 +1394,7 @@ static int ipx_release(struct socket *sock)

  sock_set_flag(sk, SOCK_DEAD);
  sock->sk = NULL;
+ sk_refcnt_debug_release(sk);
  ipx_destroy_socket(sk);
 out:
  return 0;
--
1.5.3.4
```