
Subject: [PATCH 9/11 net-2.6.26] [IPV6]: Simplify IPv6 control socket creation.

Posted by [den](#) on Fri, 28 Mar 2008 09:08:56 GMT

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

Replace sock_create_kern with inet_csk_ctl_sock_create.

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

```
net/ipv6/icmp.c | 10 ++++++--  
net/ipv6/mcast.c |  7 +----  
net/ipv6/ndisc.c |  7 +----  
3 files changed, 7 insertions(+), 17 deletions(-)
```

```
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c  
index 63309d1..fe8ddab 100644  
--- a/net/ipv6/icmp.c  
+++ b/net/ipv6/icmp.c  
@@ -808,9 +808,8 @@ static int __net_init icmpv6_sk_init(struct net *net)  
    return -ENOMEM;  
  
    for_each_possible_cpu(i) {  
-     struct socket *sock;  
-     err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6,  
-                           &sock);  
+     err = inet_csk_ctl_sock_create(&sk, PF_INET6,  
+                                   SOCK_RAW, IPPROTO_ICMPV6);  
        if (err < 0) {  
            printk(KERN_ERR  
                  "Failed to initialize the ICMP6 control socket "  
@@ -819,10 +818,9 @@ static int __net_init icmpv6_sk_init(struct net *net)  
            goto fail;  
        }  
  
-     net->ipv6.icmp_sk[i] = sk = sock->sk;  
+     net->ipv6.icmp_sk[i] = sk;  
        sk_change_net(sk, net);  
  
-     sk->sk_allocation = GFP_ATOMIC;  
/*  
 * Split off their lock-class, because sk->sk_dst_lock  
 * gets used from softirqs, which is safe for  
@@ -837,8 +835,6 @@ static int __net_init icmpv6_sk_init(struct net *net)  
 */  
        sk->sk_sndbuf =  
            (2 * ((64 * 1024) + sizeof(struct sk_buff)));  
-  
-     sk->sk_prot->unhash(sk);  
    }
```

```
return 0;
```

```
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index d810cff..4092d2e 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -2673,11 +2673,10 @@ static void igmp6_proc_exit(struct net *net)
static int igmp6_net_init(struct net *net)
{
    struct ipv6_pinfo *np;
-   struct socket *sock;
    struct sock *sk;
    int err;

-   err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6, &sock);
+   err = inet_csk_ctl_sock_create(&sk, PF_INET6, SOCK_RAW, IPPROTO_ICMPV6);
    if (err < 0) {
        printk(KERN_ERR
              "Failed to initialize the IGMP6 control socket (err %d).\n",
@@ -2685,10 +2684,8 @@ static int igmp6_net_init(struct net *net)
        goto out;
    }

-   net->ipv6.igmp_sk = sk = sock->sk;
+   net->ipv6.igmp_sk = sk;
    sk_change_net(sk, net);
-   sk->sk_allocation = GFP_ATOMIC;
-   sk->sk_prot->unhash(sk);

    np = inet6_sk(sk);
    np->hop_limit = 1;
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index e77b74e..9d8c552 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1731,12 +1731,11 @@ static int ndisc_ifinfo_sysctl_strategy(ctl_table *ctl, int __user
 *name,
```



```
static int ndisc_net_init(struct net *net)
{
-   struct socket *sock;
    struct ipv6_pinfo *np;
    struct sock *sk;
    int err;

-   err = sock_create_kern(PF_INET6, SOCK_RAW, IPPROTO_ICMPV6, &sock);
+   err = inet_csk_ctl_sock_create(&sk, PF_INET6, SOCK_RAW, IPPROTO_ICMPV6);
    if (err < 0) {
```

```
ND_PRINTK0(KERN_ERR
    "ICMPv6 NDISC: Failed to initialize the control socket (err %d).\n",
@@ -1744,15 +1743,13 @@ static int ndisc_net_init(struct net *net)
    return err;
}

- net->ipv6.ndisc_sk = sk = sock->sk;
+ net->ipv6.ndisc_sk = sk;
    sk_change_net(sk, net);

    np = inet6_sk(sk);
- sk->sk_allocation = GFP_ATOMIC;
    np->hop_limit = 255;
/* Do not loopback ndisc messages */
    np->mc_loop = 0;
- sk->sk_prot->unhash(sk);

    return 0;
}
--
```

1.5.3.rc5

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