
Subject: [PATCH 11/12] [NETNS]: Make icmpv6_sk per namespace.

Posted by [den](#) on Fri, 29 Feb 2008 13:40:57 GMT

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

All preparations are done. Now just add a hook to perform an initialization on namespace startup and replace icmpv6_sk macro with proper inline call. Actual namespace the packet belongs too will be passed later along with the one for the routing.

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

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

```
include/net/netns/ipv6.h | 1 +
net/ipv6/icmp.c          | 68 ++++++
2 files changed, 48 insertions(+), 21 deletions(-)
```

```
diff --git a/include/net/netns/ipv6.h b/include/net/netns/ipv6.h
```

```
index 1dd7de4..82623d3 100644
```

```
--- a/include/net/netns/ipv6.h
```

```
+++ b/include/net/netns/ipv6.h
```

```
@@ -36,5 +36,6 @@ struct netns_ipv6 {
```

```
    struct xt_table *ip6table_mangle;
```

```
    struct xt_table *ip6table_raw;
```

```
#endif
```

```
+ struct sock **icmp_sk;
```

```
};
```

```
#endif
```

```
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c
```

```
index 7341d79..9f55a96 100644
```

```
--- a/net/ipv6/icmp.c
```

```
+++ b/net/ipv6/icmp.c
```

```
@@ -80,8 +80,10 @@ EXPORT_SYMBOL(icmpv6msg_statistics);
```

```
*
```

```
* On SMP we have one ICMP socket per-cpu.
```

```
*/
```

```
-static struct sock *__icmpv6_sk = NULL;
```

```
-#define icmpv6_sk (__icmpv6_sk[smp_processor_id()])
```

```
+static inline struct sock *icmpv6_sk(struct net *net)
```

```
+{
```

```
+ return net->ipv6.icmp_sk[smp_processor_id()];
```

```
+}
```

```
static int icmpv6_rcv(struct sk_buff *skb);
```

```
@@ -389,7 +391,7 @@ void icmpv6_send(struct sk_buff *skb, int type, int code, __u32 info,
```

```
    fl.fl_icmp_code = code;
```

```
    security_skb_classify_flow(skb, &fl);
```

```

- sk = icmpv6_sk;
+ sk = icmpv6_sk(&init_net);
  np = inet6_sk(sk);

  if (icmpv6_xmit_lock(sk))
@@ -535,7 +537,7 @@ static void icmpv6_echo_reply(struct sk_buff *skb)
  fl.fl_icmp_type = ICMPV6_ECHO_REPLY;
  security_skb_classify_flow(skb, &fl);

- sk = icmpv6_sk;
+ sk = icmpv6_sk(&init_net);
  np = inet6_sk(sk);

  if (icmpv6_xmit_lock(sk))
@@ -780,13 +782,14 @@ drop_no_count:
  */
static struct lock_class_key icmpv6_socket_sk_dst_lock_key;

-int __init icmpv6_init(void)
+static int __net_init icmpv6_sk_init(struct net *net)
{
  struct sock *sk;
  int err, i, j;

- __icmpv6_sk = kzalloc(nr_cpu_ids * sizeof(struct sock *), GFP_KERNEL);
- if (__icmpv6_sk == NULL)
+ net->ipv6.icmp_sk =
+ kzalloc(nr_cpu_ids * sizeof(struct sock *), GFP_KERNEL);
+ if (net->ipv6.icmp_sk == NULL)
  return -ENOMEM;

  for_each_possible_cpu(i) {
@@ -801,8 +804,8 @@ int __init icmpv6_init(void)
    goto fail;
  }

- __icmpv6_sk[i] = sk = sock->sk;
- sk_change_net(sk, &init_net);
+ net->ipv6.icmp_sk[i] = sk = sock->sk;
+ sk_change_net(sk, net);

  sk->sk_allocation = GFP_ATOMIC;
  /*
@@ -822,33 +825,56 @@ int __init icmpv6_init(void)

  sk->sk_prot->unhash(sk);
}
-

```

```

-
- if (inet6_add_protocol(&icmpv6_protocol, IPPROTO_ICMPV6) < 0) {
- printk(KERN_ERR "Failed to register ICMP6 protocol\n");
- err = -EAGAIN;
- goto fail;
- }
-
return 0;

fail:
for (j = 0; j < i; j++)
- sk_release_kernel(__icmpv6_sk[j]);
-
+ sk_release_kernel(net->ipv6.icmp_sk[j]);
+ kfree(net->ipv6.icmp_sk);
return err;
}

-void icmpv6_cleanup(void)
+static void __net_exit icmpv6_sk_exit(struct net *net)
{
int i;

for_each_possible_cpu(i) {
- sk_release_kernel(__icmpv6_sk[i]);
+ sk_release_kernel(net->ipv6.icmp_sk[i]);
}
+ kfree(net->ipv6.icmp_sk);
+}
+
+static struct pernet_operations __net_initdata icmpv6_sk_ops = {
+ .init = icmpv6_sk_init,
+ .exit = icmpv6_sk_exit,
+};
+
+int __init icmpv6_init(void)
+{
+ int err;
+
+ err = register_pernet_subsys(&icmpv6_sk_ops);
+ if (err < 0)
+ return err;
+
+ err = -EAGAIN;
+ if (inet6_add_protocol(&icmpv6_protocol, IPPROTO_ICMPV6) < 0)
+ goto fail;
+ return 0;
+
+

```

```
+fail:
+ printk(KERN_ERR "Failed to register ICMP6 protocol\n");
+ unregister_pernet_subsys(&icmpv6_sk_ops);
+ return err;
+}
+
+void __exit icmpv6_cleanup(void)
+{
+ unregister_pernet_subsys(&icmpv6_sk_ops);
+ inet6_del_protocol(&icmpv6_protocol, IPPROTO_ICMPV6);
+}

+
+static const struct icmp6_err {
+ int err;
+ int fatal;
+}
--
1.5.3.rc5
```

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