

---

Subject: [PATCH 2/12 net-2.6.26] [ICMP]: Add return code to icmp\_init.

Posted by [den](#) on Fri, 29 Feb 2008 13:40:48 GMT

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

---

icmp\_init could fail and this is normal for namespace other than initial.  
So, the panic should be triggered only on init\_net initialization path.

Additionally create rollback path for icmp\_init as a separate function.  
It will also be used later during namespace destruction.

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

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

---  
include/net/icmp.h | 2 +-  
net/ipv4/af\_inet.c | 3 +-  
net/ipv4/icmp.c | 26 ++++++-----  
3 files changed, 25 insertions(+), 6 deletions(-)

```
diff --git a/include/net/icmp.h b/include/net/icmp.h
index 7bf714d..faba64d 100644
--- a/include/net/icmp.h
+++ b/include/net/icmp.h
@@ -48,7 +48,7 @@ struct sk_buff;
extern void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info);
extern int icmp_rcv(struct sk_buff *skb);
extern int icmp_ioctl(struct sock *sk, int cmd, unsigned long arg);
-extern void icmp_init(void);
+extern int icmp_init(void);
extern void icmp_out_count(unsigned char type);
```

```
/* Move into dst.h ? */
diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index a7a99ac..4f539bd 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -1430,7 +1430,8 @@ static int __init inet_init(void)
 * Set the ICMP layer up
 */
```

```
- icmp_init();
+ if (icmp_init() < 0)
+ panic("Failed to create the ICMP control socket.\n");
```

```
/*
 * Initialise the multicast router
diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index 98372db..b345b3d 100644
--- a/net/ipv4/icmp.c
```

```

+++ b/net/ipv4/icmp.c
@@ -1139,19 +1139,32 @@ static const struct icmp_control icmp_pointers[NR_ICMP_TYPES +
1] = {
},
};

-void __init icmp_init(void)
+static void __exit icmp_exit(void)
{
- struct inet_sock *inet;
  int i;

  for_each_possible_cpu(i) {
- int err;
+ struct socket *sock;
+
+ sock = per_cpu(__icmp_socket, i);
+ if (sock == NULL)
+ continue;
+ per_cpu(__icmp_socket, i) = NULL;
+ sock_release(sock);
+ }
+}

+int __init icmp_init(void)
+{
+ struct inet_sock *inet;
+ int i, err;
+
+ for_each_possible_cpu(i) {
  err = sock_create_kern(PF_INET, SOCK_RAW, IPPROTO_ICMP,
    &per_cpu(__icmp_socket, i));

  if (err < 0)
- panic("Failed to create the ICMP control socket.\n");
+ goto fail;

  per_cpu(__icmp_socket, i)->sk->sk_allocation = GFP_ATOMIC;

@@ -1171,6 +1184,11 @@ void __init icmp_init(void)
  */
  per_cpu(__icmp_socket, i)->sk->sk_prot->unhash(per_cpu(__icmp_socket, i)->sk);
}
+ return 0;
+
+fail:
+ icmp_exit();
+ return err;

```

```
}
```

```
EXPORT_SYMBOL(icmp_err_convert);
```

```
--
```

```
1.5.3.rc5
```

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---