

---

Subject: [PATCH RFC 26/31] net: Make the netlink methods in rtinetlink handle multiple network namespaces

Posted by [ebiederm](#) on Thu, 25 Jan 2007 19:00:28 GMT

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

---

From: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)> - unquoted

It turns out after a quick audit that except for removing the checks there is really nothing to do here.

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

---

net/core/rtinetlink.c | 21 +++++-----  
1 files changed, 3 insertions(+), 18 deletions(-)

```
diff --git a/net/core/rtinetlink.c b/net/core/rtinetlink.c
index 29a81bf..0a42258 100644
--- a/net/core/rtinetlink.c
+++ b/net/core/rtinetlink.c
@@ -409,9 +409,6 @@ static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
    int s_idx = cb->args[0];
    struct net_device *dev;
 
-   if (!net_eq(net, init_net()))
-   return 0;
-
-   read_lock(&per_net(dev_base_lock, net));
+   for (dev=per_net(dev_base, net), idx=0; dev; dev = dev->next, idx++) {
+       if (idx < s_idx)
@@ -446,9 +443,6 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsghdr *nlh, void *arg)
    struct nlattr *tb[IFLA_MAX+1];
    char ifname[IFNAMSIZ];
 
-   if (!net_eq(net, init_net()))
-   return -EINVAL;
-
-   err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
+   if (err < 0)
     goto errout;
@@ -622,9 +616,6 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
    int iw_buf_len = 0;
    int err;
 
-   if (!net_eq(net, init_net()))
-   return -EINVAL;
-
-   err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFLA_MAX, ifla_policy);
+   if (err < 0)
```

```

    return err;
@@ -673,13 +664,9 @@ errout:

static int rtnl_dump_all(struct sk_buff *skb, struct netlink_callback *cb)
{
- net_t net = skb->sk->sk_net;
int idx;
int s_idx = cb->family;

- if (!net_eq(net, init_net()))
- return 0;
-
if (s_idx == 0)
s_idx = 1;
for (idx=1; idx<NPROTO; idx++) {
@@ -701,6 +688,7 @@ static int rtnl_dump_all(struct sk_buff *skb, struct netlink_callback *cb)

void rtmsg_ifinfo(int type, struct net_device *dev, unsigned change)
{
+ net_t net = dev->nd_net;
struct sk_buff *skb;
int err = -ENOBUFS;

@@ -712,10 +700,10 @@ void rtmsg_ifinfo(int type, struct net_device *dev, unsigned change)
/* failure implies BUG in if_nlmsg_size() */
BUG_ON(err < 0);

- err = rtnl_notify(skb, init_net(), 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
+ err = rtnl_notify(skb, net, 0, RTNLGRP_LINK, NULL, GFP_KERNEL);
errout:
if (err < 0)
- rtnl_set_sk_err(init_net(), RTNLGRP_LINK, err);
+ rtnl_set_sk_err(net, RTNLGRP_LINK, err);
}

/* Protected by RTNL sempahore. */
@@ -862,9 +850,6 @@ static int rtnetlink_event(struct notifier_block *this, unsigned long event,
voi
{
struct net_device *dev = ptr;

- if (!net_eq(dev->nd_net, init_net()))
- return NOTIFY_DONE;
-
switch (event) {
case NETDEV_UNREGISTER:
rtmsg_ifinfo(RTM_DELLINK, dev, ~0U);
--
```

1.4.4.1.g278f

---

Containers mailing list  
Containers@lists.osdl.org  
<https://lists.osdl.org/mailman/listinfo/containers>

---