
Subject: [patch 38/38][IPV6] ndisc - make ndisc handle multiple network namespaces

Posted by [Daniel Lezcano](#) on Mon, 03 Dec 2007 16:17:14 GMT

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

Make ndisc handle multiple network namespaces:

Remove references to init_net, add network namespace parameters and add pernet_operations for ndisc

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

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>

net/ipv6/ndisc.c | 90 ++++++

1 file changed, 55 insertions(+), 35 deletions(-)

Index: linux-2.6-netns/net/ipv6/ndisc.c

=====

--- linux-2.6-netns.orig/net/ipv6/ndisc.c

+++ linux-2.6-netns/net/ipv6/ndisc.c

@ @ -442,12 +442,12 @ @ static void pndisc_destructor(struct pne

* Send a Neighbour Advertisement

*/

-static inline void ndisc_flow_init(struct flowi *fl, u8 type,

+static inline void ndisc_flow_init(struct net *net, struct flowi *fl, u8 type,
struct in6_addr *saddr, struct in6_addr *daddr,
int oif)

{

memset(fl, 0, sizeof(*fl));

- fl->fl_net = &init_net;

+ fl->fl_net = net;

ipv6_addr_copy(&fl->fl6_src, saddr);

ipv6_addr_copy(&fl->fl6_dst, daddr);

fl->proto = IPPROTO_ICMPV6;

@ @ -475,7 +475,7 @ @ static void __ndisc_send(struct net_devi

type = icmp6h->icmp6_type;

- ndisc_flow_init(&fl, type, saddr, daddr,

+ ndisc_flow_init(dev->nd_net, &fl, type, saddr, daddr,
dev->ifindex);

dst = ndisc_dst_alloc(dev, neigh, daddr, ip6_output);

@ @ -791,7 +791,7 @ @ static void ndisc_recv_ns(struct sk_buff

if (ipv6_chk_acast_addr(dev, &msg->target) ||

(idev->cnf.forwarding &&

(ipv6_devconf.proxy_ndp || idev->cnf.proxy_ndp) &&

- (pneigh = pneigh_lookup(&nd_tbl, &init_net,

```

+   (pneigh = pneigh_lookup(&nd_tbl, dev->nd_net,
    &msg->target, dev, 0)) != NULL)) {
    if (!(NEIGH_CB(skb)->flags & LOCALLY_ENQUEUED) &&
        skb->pkt_type != PACKET_HOST &&
@@ -932,7 +932,7 @@ static void ndisc_rcv_na(struct sk_buff
    */
    if (lladdr && !memcmp(lladdr, dev->dev_addr, dev->addr_len) &&
        ipv6_devconf.forwarding && ipv6_devconf.proxy_ndp &&
-   pneigh_lookup(&nd_tbl, &init_net, &msg->target, dev, 0)) {
+   pneigh_lookup(&nd_tbl, dev->nd_net, &msg->target, dev, 0)) {
    /* XXX: idev->cnf.proxy_ndp */
    goto out;
}
@@ -1440,8 +1440,8 @@ void ndisc_send_redirect(struct sk_buff
return;
}

- ndisc_flow_init(&fl, NDISC_REDIRECT, &saddr_buf, &ipv6_hdr(skb)->saddr,
- dev->ifindex);
+ ndisc_flow_init(dev->nd_net, &fl, NDISC_REDIRECT, &saddr_buf,
+ &ipv6_hdr(skb)->saddr, dev->ifindex);

dst = ip6_route_output(NULL, &fl);
if (dst == NULL)
@@ -1616,9 +1616,6 @@ static int ndisc_netdev_event(struct not
struct net_device *dev = ptr;
struct net *net = dev->nd_net;

- if (dev->nd_net != &init_net)
- return NOTIFY_DONE;
-
switch (event) {
case NETDEV_CHANGEADDR:
    neigh_changeaddr(&nd_tbl, dev);
@@ -1735,6 +1732,52 @@ static int ndisc_ifinfo_sysctl_strategy(

#endif

+static int ndisc_net_init(struct net *net)
+{
+ int err = 0;
+
+
+ net->ndisc_neigh_parms_default =
+ neigh_parms_alloc_default(&nd_tbl, net);
+ if (!net->ndisc_neigh_parms_default) {
+ err = -ENOMEM;
+ goto out;
+ }

```

```

+
+ #ifdef CONFIG_SYSCTL
+ if ((err = neigh_sysctl_register(NULL,
+   net->ndisc_neigh_parms_default,
+   NET_IPV6, NET_IPV6_NEIGH,
+   "ipv6",
+   &ndisc_ifinfo_sysctl_change,
+   &ndisc_ifinfo_sysctl_strategy)))
+   goto out_sysctl;
+ #endif
+ out:
+ return err;
+ #ifdef CONFIG_SYSCTL
+ out_sysctl:
+ neigh_parms_release(&nd_tbl, net->ndisc_neigh_parms_default);
+ goto out;
+ #endif
+ }
+
+ static void ndisc_net_exit(struct net *net)
+ {
+   struct neigh_parms *parms = net->ndisc_neigh_parms_default;
+   if (parms) {
+     #ifdef CONFIG_SYSCTL
+     neigh_sysctl_unregister(parms);
+     #endif
+     neigh_parms_release(&nd_tbl, parms);
+     net->ndisc_neigh_parms_default = NULL;
+   }
+ }
+
+ static struct pernet_operations ndisc_net_ops = {
+   .init = ndisc_net_init,
+   .exit = ndisc_net_exit,
+ };
+
+ int __init ndisc_init(struct net_proto_family *ops)
+ {
+   struct ipv6_pinfo *np;
+   @@ -1764,32 +1807,9 @@ int __init ndisc_init(struct net_proto_f
+
+   neigh_table_init(&nd_tbl);
+
+ - init_net.ndisc_neigh_parms_default =
+ - neigh_parms_alloc_default(&nd_tbl, &init_net);
+ - if (!init_net.ndisc_neigh_parms_default) {
+ -   err = -ENOMEM;
+ -   goto out_neigh_parms;

```

```

- }
-
-#ifdef CONFIG_SYSCTL
- if ((err = neigh_sysctl_register(NULL,
-   init_net.ndisc_neigh_parms_default,
-   NET_IPV6, NET_IPV6_NEIGH,
-   "ipv6",
-   &ndisc_ifinfo_sysctl_change,
-   &ndisc_ifinfo_sysctl_strategy)))
-   goto out_sysctl;
-#endif
+ register_pernet_subsys(&ndisc_net_ops);
  register_netdevice_notifier(&ndisc_netdev_notifier);
-out:
- return err;
-#ifdef CONFIG_SYSCTL
-out_sysctl:
- neigh_parms_release(&nd_tbl, init_net.ndisc_neigh_parms_default);
-#endif
-out_neigh_parms:
- sock_release(ndisc_socket);
- goto out;
+ return 0;
}

void ndisc_cleanup(void)

--

```

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