
Subject: [PATCH net-2.6.25 10/19] [NETNS] Add netns to nl_info structure.

Posted by [den](#) on Wed, 19 Dec 2007 15:24:40 GMT

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

nl_info is used to track the end-user destination of routing change notification. This is a natural object to hold a namespace on. Place it there and utilize the context in the appropriate places.

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

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

```
include/net/netlink.h | 1 +
net/ipv4/fib_frontend.c | 5 +++++
net/ipv4/fib_semantics.c | 16 ++++++++-----
net/ipv6/ip6_fib.c | 4 +++-
net/ipv6/route.c | 8 ++++++--
5 files changed, 25 insertions(+), 9 deletions(-)
```

diff --git a/include/net/netlink.h b/include/net/netlink.h

index db4b935..87cf0bc 100644

--- a/include/net/netlink.h

+++ b/include/net/netlink.h

@@ -217,6 +217,7 @@ struct nla_policy {

*/

```
struct nl_info {
    struct nlmsghdr *nlh;
+ struct net *nl_net;
    u32 pid;
};
```

diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c

index 312a728..08414eb 100644

--- a/net/ipv4/fib_frontend.c

+++ b/net/ipv4/fib_frontend.c

@@ -310,6 +310,7 @@ static int rentry_to_fib_config(int cmd, struct rentry *rt,
 int plen;

```
    memset(cfg, 0, sizeof(*cfg));
+ cfg->fc_nlinfo.nl_net = &init_net;
```

```
    if (rt->rt_dst.sa_family != AF_INET)
        return -EAFNOSUPPORT;
```

@@ -516,6 +517,7 @@ static int rtm_to_fib_config(struct sk_buff *skb, struct nlmsghdr *nlh,

```
    cfg->fc_nlinfo.pid = NETLINK_CB(skb).pid;
    cfg->fc_nlinfo.nlh = nlh;
+ cfg->fc_nlinfo.nl_net = &init_net;
```

```

if (cfg->fc_type > RTN_MAX) {
    err = -EINVAL;
@@ -670,6 +672,9 @@ static void fib_magic(int cmd, int type, __be32 dst, int dst_len, struct
in_ifad
    .fc_prefsrc = ifa->ifa_local,
    .fc_oif = ifa->ifa_dev->dev->ifindex,
    .fc_nflflags = NLM_F_CREATE | NLM_F_APPEND,
+ .fc_nlinfo = {
+ .nl_net = &init_net,
+ },
};

if (type == RTN_UNICAST)
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index c1263e2..0de6102 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -320,11 +320,11 @@ void rtmsg_fib(int event, __be32 key, struct fib_alias *fa,
    kfree_skb(skb);
    goto errout;
}
- err = rtnl_notify(skb, &init_net, info->pid, RTNLGRP_IPV4_ROUTE,
+ err = rtnl_notify(skb, info->nl_net, info->pid, RTNLGRP_IPV4_ROUTE,
    info->nlh, GFP_KERNEL);
errout:
    if (err < 0)
- rtnl_set_sk_err(&init_net, RTNLGRP_IPV4_ROUTE, err);
+ rtnl_set_sk_err(info->nl_net, RTNLGRP_IPV4_ROUTE, err);
}

/* Return the first fib alias matching TOS with
@@ -531,9 +531,11 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,

    if (cfg->fc_scope >= RT_SCOPE_LINK)
        return -EINVAL;
- if (inet_addr_type(&init_net, nh->nh_gw) != RTN_UNICAST)
+ if (inet_addr_type(cfg->fc_nlinfo.nl_net,
+     nh->nh_gw) != RTN_UNICAST)
        return -EINVAL;
- if ((dev = __dev_get_by_index(&init_net, nh->nh_oif)) == NULL)
+ if ((dev = __dev_get_by_index(cfg->fc_nlinfo.nl_net,
+     nh->nh_oif)) == NULL)
        return -ENODEV;
    if (!(dev->flags&IFF_UP))
        return -ENETDOWN;
@@ -795,7 +797,8 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    if (nhs != 1 || nh->nh_gw)
        goto err_inval;

```

```

    nh->nh_scope = RT_SCOPE_NOWHERE;
- nh->nh_dev = dev_get_by_index(&init_net, fi->fib_nh->nh_oif);
+ nh->nh_dev = dev_get_by_index(cfg->fc_nlnfo.nl_net,
+     fi->fib_nh->nh_oif);
    err = -ENODEV;
    if (nh->nh_dev == NULL)
        goto failure;
@@ -809,7 +812,8 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    if (fi->fib_prefsrc) {
        if (cfg->fc_type != RTN_LOCAL || !cfg->fc_dst ||
            fi->fib_prefsrc != cfg->fc_dst)
-     if (inet_addr_type(&init_net, fi->fib_prefsrc) != RTN_LOCAL)
+     if (inet_addr_type(cfg->fc_nlnfo.nl_net,
+         fi->fib_prefsrc) != RTN_LOCAL)
        goto err_inval;
    }

diff --git a/net/ipv6/ip6_fib.c b/net/ipv6/ip6_fib.c
index df05c6f..242a8ca 100644
--- a/net/ipv6/ip6_fib.c
+++ b/net/ipv6/ip6_fib.c
@@ -1315,7 +1315,9 @@ static int fib6_walk(struct fib6_walker_t *w)

static int fib6_clean_node(struct fib6_walker_t *w)
{
- struct nl_info info = {};
+ struct nl_info info = {
+     .nl_net = &init_net,
+ };
    int res;
    struct rt6_info *rt;
    struct fib6_cleaner_t *c = container_of(w, struct fib6_cleaner_t, w);
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 02354a7..7f03ae9 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -604,7 +604,9 @@ static int __ip6_ins_rt(struct rt6_info *rt, struct nl_info *info)

int ip6_ins_rt(struct rt6_info *rt)
{
- struct nl_info info = {};
+ struct nl_info info = {
+     .nl_net = &init_net,
+ };
    return __ip6_ins_rt(rt, &info);
}

@@ -1262,7 +1264,9 @@ static int __ip6_del_rt(struct rt6_info *rt, struct nl_info *info)

```

```
int ip6_del_rt(struct rt6_info *rt)
{
- struct nl_info info = {};
+ struct nl_info info = {
+ .nl_net = &init_net,
+ };
    return __ip6_del_rt(rt, &info);
}
```

--

1.5.3.rc5
