

---

Subject: [patch 34/38][IPv6] addrconf - Pass the proper network namespace parameters to addrconf

Posted by [Daniel Lezcano](#) on Mon, 03 Dec 2007 16:17:10 GMT

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

---

This patch propagates the network namespace pointer to the functions which need it, which means adding a new parameter to these functions, and make them use it instead of using the initial network namespace.

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

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

---

```
include/net/addrconf.h | 9 ++++---
net/ipv6/addrconf.c    | 63 ++++++-----
net/ipv6/af_inet6.c    | 6 ++--
3 files changed, 54 insertions(+), 24 deletions(-)
```

Index: linux-2.6-netns/include/net/addrconf.h

=====

--- linux-2.6-netns.orig/include/net/addrconf.h

+++ linux-2.6-netns/include/net/addrconf.h

@@ -54,9 +54,12 @@ struct prefix\_info {

extern int addrconf\_init(void);

extern void addrconf\_cleanup(void);

-extern int addrconf\_add\_ifaddr(void \_\_user \*arg);

-extern int addrconf\_del\_ifaddr(void \_\_user \*arg);

-extern int addrconf\_set\_dstaddr(void \_\_user \*arg);

+extern int addrconf\_add\_ifaddr(struct net \*net,

+ void \_\_user \*arg);

+extern int addrconf\_del\_ifaddr(struct net \*net,

+ void \_\_user \*arg);

+extern int addrconf\_set\_dstaddr(struct net \*net,

+ void \_\_user \*arg);

extern int ipv6\_chk\_addr(struct net \*net,

struct in6\_addr \*addr,

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

=====

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

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

@@ -1847,7 +1847,7 @@ ok:

\* Special case for SIT interfaces where we create a new "virtual"

\* device.

\*/

-int addrconf\_set\_dstaddr(void \_\_user \*arg)

+int addrconf\_set\_dstaddr(struct net \*net, void \_\_user \*arg)

{

```

struct in6_ifreq ireq;
struct net_device *dev;
@@ -1859,7 +1859,7 @@ int addrconf_set_dstaddr(void __user *ar
    if (copy_from_user(&ireq, arg, sizeof(struct in6_ifreq)))
        goto err_exit;

- dev = __dev_get_by_index(&init_net, ireq.ifr6_ifindex);
+ dev = __dev_get_by_index(net, ireq.ifr6_ifindex);

    err = -ENODEV;
    if (dev == NULL)
@@ -1890,7 +1890,7 @@ int addrconf_set_dstaddr(void __user *ar

    if (err == 0) {
        err = -ENOBUFS;
-    if ((dev = __dev_get_by_name(&init_net, p.name)) == NULL)
+    if ((dev = __dev_get_by_name(net, p.name)) == NULL)
        goto err_exit;
        err = dev_open(dev);
    }
@@ -1905,8 +1905,9 @@ err_exit:
/*
 * Manual configuration of address on an interface
 */
-static int inet6_addr_add(int ifindex, struct in6_addr *pfx, int plen,
-    __u8 ifa_flags, __u32 preferred_lft, __u32 valid_lft)
+static int inet6_addr_add(struct net *net, int ifindex, struct in6_addr *pfx,
+    int plen, __u8 ifa_flags, __u32 preferred_lft,
+    __u32 valid_lft)
{
    struct inet6_ifaddr *ifp;
    struct inet6_dev *idev;
@@ -1920,7 +1921,7 @@ static int inet6_addr_add(int ifindex, s
    if (!valid_lft || preferred_lft > valid_lft)
        return -EINVAL;

- if ((dev = __dev_get_by_index(&init_net, ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(net, ifindex)) == NULL)
    return -ENODEV;

    if ((idev = addrconf_add_dev(dev)) == NULL)
@@ -1965,13 +1966,14 @@ static int inet6_addr_add(int ifindex, s
    return PTR_ERR(ifp);
}

-static int inet6_addr_del(int ifindex, struct in6_addr *pfx, int plen)
+static int inet6_addr_del(struct net *net, int ifindex, struct in6_addr *pfx,
+    int plen)

```

```

{
    struct inet6_ifaddr *ifp;
    struct inet6_dev *idev;
    struct net_device *dev;

- if ((dev = __dev_get_by_index(&init_net, ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(net, ifindex)) == NULL)
    return -ENODEV;

    if ((idev = __in6_dev_get(dev)) == NULL)
@@ -1999,7 +2001,7 @@ static int inet6_addr_del(int ifindex, s
}

-int addrconf_add_ifaddr(void __user *arg)
+int addrconf_add_ifaddr(struct net *net, void __user *arg)
{
    struct in6_ifreq ireq;
    int err;
@@ -2011,13 +2013,14 @@ int addrconf_add_ifaddr(void __user *arg
    return -EFAULT;

    rtnl_lock();
- err = inet6_addr_add(ireq.ifr6_ifindex, &ireq.ifr6_addr, ireq.ifr6_prefixlen,
-     IFA_F_PERMANENT, INFINITY_LIFE_TIME, INFINITY_LIFE_TIME);
+ err = inet6_addr_add(net, ireq.ifr6_ifindex, &ireq.ifr6_addr,
+     ireq.ifr6_prefixlen, IFA_F_PERMANENT,
+     INFINITY_LIFE_TIME, INFINITY_LIFE_TIME);
    rtnl_unlock();
    return err;
}

-int addrconf_del_ifaddr(void __user *arg)
+int addrconf_del_ifaddr(struct net *net, void __user *arg)
{
    struct in6_ifreq ireq;
    int err;
@@ -2029,7 +2032,8 @@ int addrconf_del_ifaddr(void __user *arg
    return -EFAULT;

    rtnl_lock();
- err = inet6_addr_del(ireq.ifr6_ifindex, &ireq.ifr6_addr, ireq.ifr6_prefixlen);
+ err = inet6_addr_del(net, ireq.ifr6_ifindex, &ireq.ifr6_addr,
+     ireq.ifr6_prefixlen);
    rtnl_unlock();
    return err;
}
@@ -3031,7 +3035,7 @@ inet6_rtm_deladdr(struct sk_buff *skb, s

```

```

if (pfx == NULL)
    return -EINVAL;

- return inet6_addr_del(ifm->ifa_index, pfx, ifm->ifa_prefixlen);
+ return inet6_addr_del(net, ifm->ifa_index, pfx, ifm->ifa_prefixlen);
}

static int inet6_addr_modify(struct inet6_ifaddr *ifp, u8 ifa_flags,
@@ -3107,7 +3111,7 @@ inet6_rtm_newaddr(struct sk_buff *skb, s
    valid_lft = INFINITY_LIFE_TIME;
}

- dev = __dev_get_by_index(&init_net, ifm->ifa_index);
+ dev = __dev_get_by_index(net, ifm->ifa_index);
if (dev == NULL)
    return -ENODEV;

@@ -3120,8 +3124,9 @@ inet6_rtm_newaddr(struct sk_buff *skb, s
    * It would be best to check for !NLM_F_CREATE here but
    * userspace already relies on not having to provide this.
    */
- return inet6_addr_add(ifm->ifa_index, pfx, ifm->ifa_prefixlen,
-     ifa_flags, preferred_lft, valid_lft);
+ return inet6_addr_add(net, ifm->ifa_index, pfx,
+     ifm->ifa_prefixlen, ifa_flags, preferred_lft,
+     valid_lft);
}

if (nlh->nmsg_flags & NLM_F_EXCL ||
@@ -4226,6 +4231,22 @@ int unregister_inet6addr_notifier(struct

EXPORT_SYMBOL(unregister_inet6addr_notifier);

+
+static int addrconf_net_init(struct net *net)
+{
+    return 0;
+}
+
+static void addrconf_net_exit(struct net *net)
+{
+    ;
+}
+
+static struct pernet_operations addrconf_net_ops = {
+    .init = addrconf_net_init,
+    .exit = addrconf_net_exit,
+};

```

```

+
/*
 * Init / cleanup code
 */
@@ -4274,6 +4295,10 @@ int __init addrconf_init(void)
    init_net.ipv6_blk_hole_entry->rt6i_idev = in6_dev_get(init_net.loopback_dev);
#endif

+ err = register_pernet_device(&addrconf_net_ops);
+ if (err)
+ return err;
+
    register_netdevice_notifier(&ipv6_dev_notf);

    addrconf_verify(0);
@@ -4300,6 +4325,7 @@ int __init addrconf_init(void)
    return 0;
errout:
    unregister_netdevice_notifier(&ipv6_dev_notf);
+ unregister_pernet_device(&addrconf_net_ops);

    return err;
}
@@ -4350,6 +4376,7 @@ void __exit addrconf_cleanup(void)
    write_unlock_bh(&addrconf_hash_lock);

    del_timer(&addr_chk_timer);
-
    rtnl_unlock();
+
+ unregister_pernet_subsys(&addrconf_net_ops);
}
Index: linux-2.6-netns/net/ipv6/af_inet6.c
=====
--- linux-2.6-netns.orig/net/ipv6/af_inet6.c
+++ linux-2.6-netns/net/ipv6/af_inet6.c
@@ -457,11 +457,11 @@ int inet6_ioctl(struct socket *sock, uns
    return(ipv6_route_ioctl(net, cmd,(void __user *)arg));

    case SIOCSIFADDR:
- return addrconf_add_ifaddr((void __user *) arg);
+ return addrconf_add_ifaddr(net, (void __user *) arg);
    case SIOCDELIFADDR:
- return addrconf_del_ifaddr((void __user *) arg);
+ return addrconf_del_ifaddr(net, (void __user *) arg);
    case SIOCSIFDSTADDR:
- return addrconf_set_dstaddr((void __user *) arg);
+ return addrconf_set_dstaddr(net, (void __user *) arg);

```

```
default:
if (!sk->sk_prot->ioctl)
return -ENOIOCTLCMD;
```

--

---

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

---