
Subject: [PATCH net-2.6.25 4/6][NETNS]: Create ipv6 devconf-s for namespaces
Posted by [Pavel Emelianov](#) on Thu, 10 Jan 2008 14:06:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is the core. Declare and register the pernet subsys for
addrconf. The init callback the will create the devconf-s.

The init_net will reuse the existing statically declared confs,
so that accessing them from inside the ipv6 code will still
work.

The register_pernet_subsys() is moved above the ipv6_add_dev()
call for loopback, because this function will need the
net->devconf_dflt pointer to be already set.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
---
include/net/netns/ipv6.h | 2 +
net/ipv6/addrconf.c      | 82 ++++++-----
2 files changed, 72 insertions(+), 12 deletions(-)

diff --git a/include/net/netns/ipv6.h b/include/net/netns/ipv6.h
index 10733a6..06b4dc0 100644
--- a/include/net/netns/ipv6.h
+++ b/include/net/netns/ipv6.h
@@ -28,5 +28,7 @@ struct netns_sysctl_ipv6 {

struct netns_ipv6 {
    struct netns_sysctl_ipv6 sysctl;
+ struct ipv6_devconf *devconf_all;
+ struct ipv6_devconf *devconf_dflt;
};
#endif

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index bde50c6..3ad081e 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -4135,6 +4135,70 @@ static void addrconf_sysctl_unregister(struct inet6_dev *idev)

#endif

+static int addrconf_init_net(struct net *net)
+{
+    int err;
+    struct ipv6_devconf *all, *dflt;
+
+    err = -ENOMEM;
```

```

+ all = &ipv6_devconf;
+ dflt = &ipv6_devconf_dflt;
+
+ if (net != &init_net) {
+   all = kmemdup(all, sizeof(ipv6_devconf), GFP_KERNEL);
+   if (all == NULL)
+     goto err_alloc_all;
+
+   dflt = kmemdup(dflt, sizeof(ipv6_devconf_dflt), GFP_KERNEL);
+   if (dflt == NULL)
+     goto err_alloc_dflt;
+ }
+
+ net->ipv6.devconf_all = all;
+ net->ipv6.devconf_dflt = dflt;
+
+#ifdef CONFIG_SYSCTL
+ err = __addrconf_sysctl_register(net, "all", NET_PROTO_CONF_ALL,
+   NULL, all);
+ if (err < 0)
+   goto err_reg_all;
+
+ err = __addrconf_sysctl_register(net, "default", NET_PROTO_CONF_DEFAULT,
+   NULL, dflt);
+ if (err < 0)
+   goto err_reg_dflt;
+#endif
+ return 0;
+
+#ifdef CONFIG_SYSCTL
+err_reg_dflt:
+ __addrconf_sysctl_unregister(all);
+err_reg_all:
+ kfree(dflt);
+#endif
+err_alloc_dflt:
+ kfree(all);
+err_alloc_all:
+ return err;
+}
+
+static void addrconf_exit_net(struct net *net)
+{
+#ifdef CONFIG_SYSCTL
+ __addrconf_sysctl_unregister(net->ipv6.devconf_dflt);
+ __addrconf_sysctl_unregister(net->ipv6.devconf_all);
+#endif
+ if (net != &init_net) {

```

```

+ kfree(net->ipv6.devconf_dflt);
+ kfree(net->ipv6.devconf_all);
+ }
+}
+
+static struct pernet_operations addrconf_ops = {
+ .init = addrconf_init_net,
+ .exit = addrconf_exit_net,
+};
+
+/*
+ * Device notifier
+ */
@@ -4167,6 +4231,8 @@ int __init addrconf_init(void)
    return err;
}

+ register_pernet_subsys(&addrconf_ops);
+
+/* The addrconf netdev notifier requires that loopback_dev
+ * has it's ipv6 private information allocated and setup
+ * before it can bring up and give link-local addresses
@@ -4190,7 +4256,7 @@ int __init addrconf_init(void)
    err = -ENOMEM;
    rtnl_unlock();
    if (err)
- return err;
+ goto errlo;

    ip6_null_entry.u.dst.dev = init_net.loopback_dev;
    ip6_null_entry.rtnl_dev = in6_dev_get(init_net.loopback_dev);
@@ -4218,16 +4284,16 @@ int __init addrconf_init(void)

    ipv6_addr_label_rtnl_register();

-#ifdef CONFIG_SYSCTL
- __addrconf_sysctl_register(&init_net, "all", NET_PROTO_CONF_ALL,
- NULL, &ipv6_devconf);
- __addrconf_sysctl_register(&init_net, "default", NET_PROTO_CONF_DEFAULT,
- NULL, &ipv6_devconf_dflt);
-#endif
-
    return 0;
errout:
    unregister_netdevice_notifier(&ipv6_dev_notifier);
+errlo:
+ unregister_pernet_subsys(&addrconf_ops);

```

```
    return err;
}
@@ -4240,10 +4301,7 @@ void addrconf_cleanup(void)

    unregister_netdevice_notifier(&ipv6_dev_notf);

-#ifdef CONFIG_SYSCTL
- __addrconf_sysctl_unregister(&ipv6_devconf_dflt);
- __addrconf_sysctl_unregister(&ipv6_devconf);
-#endif
+ unregister_pernet_subsys(&addrconf_ops);

    rtnl_lock();

--
1.5.3.4
```
