Subject: [PATCH net-2.6.25 4/7] Store the net pointer on devinet's ctl tables
Posted by Pavel Emelianov on Tue, 11 Dec 2007 17:53:55 GMT
View Forum Message <> Reply to Message

Some handers and strategies of devinet sysctl tables need
to know the net to propagate the ctl change to all the
net devices.

I use the (currently unused) extra2 pointer on the tables
to get it.

Holding the reference on the struct net is not possible,
because otherwise we'll get a net->ctl_table->net circular
dependency. But since the ctl tables are unregistered during
the net destruction, this is safe to get it w/o additional
protection.

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

---

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 872883e..bfb0fb0 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1237,12 +1237,12 @@ errout:

 #ifdef CONFIG_SYSCTL

-static void devinet_copy_dflt_conf(int i)
+static void devinet_copy_dflt_conf(struct net *net, int i)
 {
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for_each_netdev(&init_net, dev) {
+ for_each_netdev(net, dev) {
   struct in_device *in_dev;
   rcu_read_lock();
   in_dev = __in_dev_get_rcu(dev);
@@ -1253,7 +1253,7 @@ static void devinet_copy_dflt_conf(int i)
  read_unlock(&dev_base_lock);
 }

-static void inet_forward_change(void)
+static void inet_forward_change(struct net *net)
 {
  struct net_device *dev;
```

```
  int on = IPV4_DEVCONF_ALL(FORWARDING);
@@ -1262,7 +1262,7 @@ static void inet_forward_change(void)
  IPV4_DEVCONF_DFLT(FORWARDING) = on;

  read_lock(&dev_base_lock);
- for_each_netdev(&init_net, dev) {
+ for_each_netdev(net, dev) {
   struct in_device *in_dev;
   rcu_read_lock();
   in_dev = __in_dev_get_rcu(dev);
@@ -1283,12 +1283,13 @@ static int devinet_conf_proc(ctl_table *ctl, int write,

  if (write) {
   struct ipv4_devconf *cnf = ctl->extra1;
+  struct net *net = ctl->extra2;
   int i = (int *)ctl->data - cnf->data;

   set_bit(i, cnf->state);

   if (cnf == &ipv4_devconf_dflt)
-   devinet_copy_dflt_conf(i);
+   devinet_copy_dflt_conf(net, i);
  }

  return ret;
@@ -1299,6 +1300,7 @@ static int devinet_conf_sysctl(ctl_table *table, int __user *name, int
nlen,
          void __user *newval, size_t newlen)
 {
  struct ipv4_devconf *cnf;
+ struct net *net;
  int *valp = table->data;
  int new;
  int i;
@@ -1334,12 +1336,13 @@ static int devinet_conf_sysctl(ctl_table *table, int __user *name, int
nlen,
  *valp = new;

  cnf = table->extra1;
+ net = table->extra2;
  i = (int *)table->data - cnf->data;

  set_bit(i, cnf->state);

  if (cnf == &ipv4_devconf_dflt)
-  devinet_copy_dflt_conf(i);
+  devinet_copy_dflt_conf(net, i);
```

```
  return 1;
 }
@@ -1353,8 +1356,10 @@ static int devinet_sysctl_forward(ctl_table *ctl, int write,
  int ret = proc_dointvec(ctl, write, filp, buffer, lenp, ppos);

  if (write && *valp != val) {
+  struct net *net = ctl->extra2;
+
   if (valp == &IPV4_DEVCONF_ALL(FORWARDING))
-   inet_forward_change();
+   inet_forward_change(net);
   else if (valp != &IPV4_DEVCONF_DFLT(FORWARDING))
    rt_cache_flush(0);
  }
@@ -1478,6 +1483,7 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
  for (i = 0; i < ARRAY_SIZE(t->devinet_vars) - 1; i++) {
   t->devinet_vars[i].data += (char *)p - (char *)&ipv4_devconf;
   t->devinet_vars[i].extra1 = p;
+  t->devinet_vars[i].extra2 = net;
  }

  /*
@@ -1525,8 +1531,8 @@ static void devinet_sysctl_register(struct in_device *idev)
 {
  neigh_sysctl_register(idev->dev, idev->arp_parms, NET_IPV4,
    NET_IPV4_NEIGH, "ipv4", NULL, NULL);
- __devinet_sysctl_register(idev->dev->name, idev->dev->ifindex,
-   &idev->cnf);
+ __devinet_sysctl_register(idev->dev->nd_net, idev->dev->name,
+   idev->dev->ifindex, &idev->cnf);
 }

 static void devinet_sysctl_unregister(struct in_device *idev)
@@ -1547,6 +1553,7 @@ static struct ctl_table ctl_forward_entry[] = {
   .proc_handler = devinet_sysctl_forward,
   .strategy = devinet_conf_sysctl,
   .extra1  = &ipv4_devconf,
+  .extra2  = &init_net,
  },
  { },
 };
@@ -1566,9 +1573,9 @@ void __init devinet_init(void)
  rtnl_register(PF_INET, RTM_DELADDR, inet_rtm_deladdr, NULL);
  rtnl_register(PF_INET, RTM_GETADDR, NULL, inet_dump_ifaddr);
 #ifdef CONFIG_SYSCTL
- __devinet_sysctl_register("all", NET_PROTO_CONF_ALL,
+ __devinet_sysctl_register(&init_net, "all", NET_PROTO_CONF_ALL,
    &ipv4_devconf);
```

```
- __devinet_sysctl_register("default", NET_PROTO_CONF_DEFAULT,
+ __devinet_sysctl_register(&init_net, "default", NET_PROTO_CONF_DEFAULT,
    &ipv4_devconf_dflt);
  register_sysctl_paths(net_ipv4_path, ctl_forward_entry);
 #endif
--
1.5.3.4
```

---