Subject: [PATCH net-2.6.25 5/7] Move the devinet pointers on the struct net
Posted by Pavel Emelianov on Tue, 11 Dec 2007 17:57:18 GMT
View Forum Message <> Reply to Message

This is the core.

Add all and default pointers on the netns_ipv4 and register
a new pernet subsys to initialize them.

Also add the ctl_table_header to register the
net.ipv4.ip_forward ctl.

I don't allocate additional memory for init_net, but use
global devinets.

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

---

```
diff --git a/include/net/netns/ipv4.h b/include/net/netns/ipv4.h
index ce830d5..e06d7cf 100644
--- a/include/net/netns/ipv4.h
+++ b/include/net/netns/ipv4.h
@@ -4,6 +4,12 @@

 #ifndef __NETNS_IPV4_H__
 #define __NETNS_IPV4_H__
+struct ctl_table_header;
+struct ipv4_devconf;
+
 struct netns_ipv4 {
+ struct ctl_table_header *forw_hdr;
+ struct ipv4_devconf *devconf_all;
+ struct ipv4_devconf *devconf_dflt;
 };
 #endif
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index bfb0fb0..996f07e 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -62,6 +62,7 @@
 #include <net/route.h>
 #include <net/ip_fib.h>
 #include <net/rtnetlink.h>
+#include <net/net_namespace.h>

 struct ipv4_devconf ipv4_devconf = {
   .data = {
```

```
@@ -1498,7 +1499,7 @@ static int __devinet_sysctl_register(struct net *net, char *dev_name,
  devinet_ctl_path[DEVINET_CTL_PATH_DEV].procname = t->dev_name;
  devinet_ctl_path[DEVINET_CTL_PATH_DEV].ctl_name = ctl_name;

- t->sysctl_header = register_sysctl_paths(devinet_ctl_path,
+ t->sysctl_header = register_net_sysctl_table(net, devinet_ctl_path,
    t->devinet_vars);
  if (!t->sysctl_header)
   goto free_procname;
@@ -1558,27 +1559,113 @@ static struct ctl_table ctl_forward_entry[] = {
  { },
 };

-static __initdata struct ctl_path net_ipv4_path[] = {
+static __net_initdata struct ctl_path net_ipv4_path[] = {
  { .procname = "net", .ctl_name = CTL_NET, },
  { .procname = "ipv4", .ctl_name = NET_IPV4, },
  { },
 };

+static __net_init int devinet_init_net(struct net *net)
+{
+ int err;
+ struct ctl_table *tbl;
+ struct ipv4_devconf *all, *dflt;
+ struct ctl_table_header *forw_hdr;
+
+ err = -ENOMEM;
+ all = &ipv4_devconf;
+ dflt = &ipv4_devconf_dflt;
+ tbl = ctl_forward_entry;
+
+ if (net != &init_net) {
+  all = kmemdup(all, sizeof(ipv4_devconf), GFP_KERNEL);
+  if (all == NULL)
+   goto err_alloc_all;
+
+  dflt = kmemdup(dflt, sizeof(ipv4_devconf_dflt), GFP_KERNEL);
+  if (dflt == NULL)
+   goto err_alloc_dflt;
+
+  tbl = kmemdup(tbl, sizeof(ctl_forward_entry), GFP_KERNEL);
+  if (tbl == NULL)
+   goto err_alloc_ctl;
+
+  tbl[0].data = &all->data[NET_IPV4_CONF_FORWARDING - 1];
+  tbl[0].extra1 = all;
+  tbl[0].extra2 = net;
```

```
+ }
+
+#ifdef CONFIG_SYSCTL
+ err = __devinet_sysctl_register(net, "all",
+   NET_PROTO_CONF_ALL, all);
+ if (err < 0)
+  goto err_reg_all;
+
+ err = __devinet_sysctl_register(net, "default",
+   NET_PROTO_CONF_DEFAULT, dflt);
+ if (err < 0)
+  goto err_reg_dflt;
+
+ err = -ENOMEM;
+ forw_hdr = register_net_sysctl_table(net, net_ipv4_path, tbl);
+ if (forw_hdr == NULL)
+  goto err_reg_ctl;
+#endif
+
+ net->ipv4.forw_hdr = forw_hdr;
+ net->ipv4.devconf_all = all;
+ net->ipv4.devconf_dflt = dflt;
+ return 0;
+
+#ifdef CONFIG_SYSCTL
+err_reg_ctl:
+ __devinet_sysctl_unregister(dflt);
+err_reg_dflt:
+ __devinet_sysctl_unregister(all);
+err_reg_all:
+ if (tbl != ctl_forward_entry)
+  kfree(tbl);
+#endif
+err_alloc_ctl:
+ if (dflt != &ipv4_devconf_dflt)
+  kfree(dflt);
+err_alloc_dflt:
+ if (all != &ipv4_devconf)
+  kfree(all);
+err_alloc_all:
+ return err;
+}
+
+static __net_exit void devinet_exit_net(struct net *net)
+{
+ struct ctl_table *tbl;
+
+ tbl = net->ipv4.forw_hdr->ctl_table_arg;
```

```
+#ifdef CONFIG_SYSCTL
+ unregister_net_sysctl_table(net->ipv4.forw_hdr);
+ __devinet_sysctl_unregister(net->ipv4.devconf_dflt);
+ __devinet_sysctl_unregister(net->ipv4.devconf_all);
+#endif
+ kfree(tbl);
+ kfree(net->ipv4.devconf_dflt);
+ kfree(net->ipv4.devconf_all);
+}
+
+static __net_initdata struct pernet_operations devinet_ops = {
+ .init = devinet_init_net,
+ .exit = devinet_exit_net,
+};
+
 void __init devinet_init(void)
 {
+ register_pernet_subsys(&devinet_ops);
+
  register_gifconf(PF_INET, inet_gifconf);
  register_netdevice_notifier(&ip_netdev_notifier);

  rtnl_register(PF_INET, RTM_NEWADDR, inet_rtm_newaddr, NULL);
  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(&init_net, "all", NET_PROTO_CONF_ALL,
-   &ipv4_devconf);
- __devinet_sysctl_register(&init_net, "default", NET_PROTO_CONF_DEFAULT,
-   &ipv4_devconf_dflt);
- register_sysctl_paths(net_ipv4_path, ctl_forward_entry);
-#endif
 }

 EXPORT_SYMBOL(in_dev_finish_destroy);
--
1.5.3.4
```