Subject: [PATCH net-2.6.25 (resend) 2/3][IPV6] Unify and cleanup calls to addrconf_sysctl_register
Posted by Pavel Emelianov on Sat, 01 Dec 2007 13:45:09 GMT
View Forum Message <> Reply to Message

Currently this call is (ab)used similar to devinet one - it
registers sysctls for devices and for the "default" confs, while
the "all" sysctls are registered separately. But unlike its
devinet brother, the passed inet6_device is needed.

The fix is to make a __addrconf_sysctl_register(), which registers
sysctls for all "devices" we need, including "default" and "all" :)

The original addrconf_sysctl_register() calls the introduced
function, passing the inet6_device, device name and ifindex (to
be used as procname and ctl_name) into it.

Thanks to Herbert again for pointing out, that we can shrink the
argument list to 1 :)

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

```
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 2d2886a..ea1673d 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -101,7 +101,7 @@
 #define TIME_DELTA(a,b) ((unsigned long)((long)(a) - (long)(b)))

 #ifdef CONFIG_SYSCTL
-static void addrconf_sysctl_register(struct inet6_dev *idev, struct ipv6_devconf *p);
+static void addrconf_sysctl_register(struct inet6_dev *idev);
 static void addrconf_sysctl_unregister(struct ipv6_devconf *p);
 #endif

@@ -400,7 +400,7 @@ static struct inet6_dev * ipv6_add_dev(struct net_device *dev)
         NET_IPV6_NEIGH, "ipv6",
         &ndisc_ifinfo_sysctl_change,
         NULL);
- addrconf_sysctl_register(ndev, &ndev->cnf);
+ addrconf_sysctl_register(ndev);
 #endif
  /* protected by rtnl_lock */
  rcu_assign_pointer(dev->ip6_ptr, ndev);
@@ -2386,7 +2386,7 @@ static int addrconf_notify(struct notifier_block *this, unsigned long
event,
```

```
           NET_IPV6, NET_IPV6_NEIGH, "ipv6",
           &ndisc_ifinfo_sysctl_change,
           NULL);
-   addrconf_sysctl_register(idev, &idev->cnf);
+   addrconf_sysctl_register(idev);
 #endif
    err = snmp6_register_dev(idev);
    if (err)
@@ -4118,12 +4118,11 @@ static struct addrconf_sysctl_table
  },
 };

-static void addrconf_sysctl_register(struct inet6_dev *idev, struct ipv6_devconf *p)
+static void __addrconf_sysctl_register(char *dev_name, int ctl_name,
+  struct inet6_dev *idev, struct ipv6_devconf *p)
 {
  int i;
- struct net_device *dev = idev ? idev->dev : NULL;
  struct addrconf_sysctl_table *t;
- char *dev_name = NULL;

  t = kmemdup(&addrconf_sysctl, sizeof(*t), GFP_KERNEL);
  if (t == NULL)
@@ -4133,13 +4132,6 @@ static void addrconf_sysctl_register(struct inet6_dev *idev, struct
ipv6_devconf
   t->addrconf_vars[i].data += (char*)p - (char*)&ipv6_devconf;
   t->addrconf_vars[i].extra1 = idev; /* embedded; no ref */
  }
- if (dev) {
-  dev_name = dev->name;
-  t->addrconf_dev[0].ctl_name = dev->ifindex;
- } else {
-  dev_name = "default";
-  t->addrconf_dev[0].ctl_name = NET_PROTO_CONF_DEFAULT;
- }

  /*
   * Make a copy of dev_name, because '.procname' is regarded as const
@@ -4150,6 +4142,7 @@ static void addrconf_sysctl_register(struct inet6_dev *idev, struct
ipv6_devconf
  if (!dev_name)
   goto free;

+ t->addrconf_dev[0].ctl_name = ctl_name;
  t->addrconf_dev[0].procname = dev_name;

  t->addrconf_dev[0].child = t->addrconf_vars;
@@ -4172,6 +4165,12 @@ out:
```

```
  return;
 }

+static void addrconf_sysctl_register(struct inet6_dev *idev)
+{
+ __addrconf_sysctl_register(idev->dev->name, idev->dev->ifindex,
+    idev, &idev->cnf);
+}
+
 static void addrconf_sysctl_unregister(struct ipv6_devconf *p)
 {
  if (p->sysctl) {
@@ -4270,9 +4269,10 @@ int __init addrconf_init(void)
  ipv6_addr_label_rtnl_register();

 #ifdef CONFIG_SYSCTL
- addrconf_sysctl.sysctl_header =
-  register_sysctl_table(addrconf_sysctl.addrconf_root_dir);
- addrconf_sysctl_register(NULL, &ipv6_devconf_dflt);
+ __addrconf_sysctl_register("all", NET_PROTO_CONF_ALL,
+    NULL, &ipv6_devconf);
+ __addrconf_sysctl_register("default", NET_PROTO_CONF_DEFAULT,
+    NULL, &ipv6_devconf_dflt);
 #endif

  return 0;
--
1.5.3.4
```