Subject: [PATCH net-2.6.25 2/3][IPV4] Unify and cleanup calls to devinet_sysctl_register
Posted by Pavel Emelianov on Fri, 30 Nov 2007 18:26:58 GMT
View Forum Message <> Reply to Message

Currently this call is used to register sysctls for devices
and for the "default" confs. The "all" sysctls are registered
separately.

Besides, the inet_device is passed to this function, but
it is not needed there at all - just the device name and
ifindex are required.

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

The original devinet_sysctl_register() works with struct net_device,
not the inet_device, and calls the introduced function, passing
the device name and ifindex (to be used as procname and ctl_name)
into it.

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

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 34c34c6..f65c350 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -98,7 +98,7 @@ static BLOCKING_NOTIFIER_HEAD(inetaddr_chain);
 static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
     int destroy);
 #ifdef CONFIG_SYSCTL
-static void devinet_sysctl_register(struct in_device *in_dev,
+static void devinet_sysctl_register(struct net_device *dev,
         struct ipv4_devconf *p);
 static void devinet_sysctl_unregister(struct ipv4_devconf *p);
 #endif
@@ -173,7 +173,7 @@ static struct in_device *inetdev_init(struct net_device *dev)
  in_dev_hold(in_dev);

 #ifdef CONFIG_SYSCTL
- devinet_sysctl_register(in_dev, &in_dev->cnf);
+ devinet_sysctl_register(dev, &in_dev->cnf);
 #endif
  ip_mc_init_dev(in_dev);
  if (dev->flags & IFF_UP)
@@ -1120,7 +1120,7 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
```

```
   neigh_sysctl_unregister(in_dev->arp_parms);
   neigh_sysctl_register(dev, in_dev->arp_parms, NET_IPV4,
           NET_IPV4_NEIGH, "ipv4", NULL, NULL);
-  devinet_sysctl_register(in_dev, &in_dev->cnf);
+  devinet_sysctl_register(dev, &in_dev->cnf);
 #endif
   break;
  }
@@ -1502,13 +1502,11 @@ static struct devinet_sysctl_table {
  },
 };

-static void devinet_sysctl_register(struct in_device *in_dev,
-        struct ipv4_devconf *p)
+static void __devinet_sysctl_register(char *dev_name, int ctl_name,
+  struct ipv4_devconf *p)
 {
  int i;
- struct net_device *dev = in_dev ? in_dev->dev : NULL;
  struct devinet_sysctl_table *t;
- char *dev_name = NULL;

  t = kmemdup(&devinet_sysctl, sizeof(*t), GFP_KERNEL);
  if (!t)
@@ -1519,13 +1517,7 @@ static void devinet_sysctl_register(struct in_device *in_dev,
   t->devinet_vars[i].extra1 = p;
  }

- if (dev) {
-  dev_name = dev->name;
-  t->devinet_dev[0].ctl_name = dev->ifindex;
- } else {
-  dev_name = "default";
-  t->devinet_dev[0].ctl_name = NET_PROTO_CONF_DEFAULT;
- }
+ t->devinet_dev[0].ctl_name = ctl_name;

  /*
   * Make a copy of dev_name, because '.procname' is regarded as const
@@ -1557,6 +1549,12 @@ out:
  return;
 }

+static void devinet_sysctl_register(struct net_device *dev,
+  struct ipv4_devconf *p)
+{
+ return __devinet_sysctl_register(dev->name, dev->ifindex, p);
+}
```

```
+
 static void devinet_sysctl_unregister(struct ipv4_devconf *p)
 {
  if (p->sysctl) {
@@ -1578,9 +1576,10 @@ 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.sysctl_header =
-  register_sysctl_table(devinet_sysctl.devinet_root_dir);
- devinet_sysctl_register(NULL, &ipv4_devconf_dflt);
+ __devinet_sysctl_register("all", NET_PROTO_CONF_ALL,
+   &ipv4_devconf);
+ __devinet_sysctl_register("default", NET_PROTO_CONF_DEFAULT,
+   &ipv4_devconf_dflt);
 #endif
 }

--
1.5.3.4
```

Subject: Re: [PATCH net-2.6.25 2/3][IPV4] Unify and cleanup calls to devinet_sysctl_register
Posted by Herbert Xu on Sat, 01 Dec 2007 13:20:00 GMT
View Forum Message <> Reply to Message

On Fri, Nov 30, 2007 at 09:26:58PM +0300, Pavel Emelyanov wrote:
>
> Besides, the inet_device is passed to this function, but
> it is not needed there at all - just the device name and
> ifindex are required.

But it is called devinet_* so an in_dev kind of makes sense :)

> #ifdef CONFIG_SYSCTL
> - devinet_sysctl_register(in_dev, &in_dev->cnf);
> + devinet_sysctl_register(dev, &in_dev->cnf);

How about just giving it in_dev instead?

Thanks,
--
Visit Openswan at http://www.openswan.org/
Email: Herbert Xu ~{PmV>HI~} <herbert@gondor.apana.org.au>
Home Page: http://gondor.apana.org.au/~herbert/
PGP Key: http://gondor.apana.org.au/~herbert/pubkey.txt

Subject: Re: [PATCH net-2.6.25 2/3][IPV4] Unify and cleanup calls to devinet_sysctl_register
Posted by Pavel Emelianov on Sat, 01 Dec 2007 13:25:21 GMT
View Forum Message <> Reply to Message

Herbert Xu wrote:
> On Fri, Nov 30, 2007 at 09:26:58PM +0300, Pavel Emelyanov wrote:
>> Besides, the inet_device is passed to this function, but
>> it is not needed there at all - just the device name and
>> ifindex are required.
>
> But it is called devinet_* so an in_dev kind of makes sense :)
>
>> #ifdef CONFIG_SYSCTL
>> - devinet_sysctl_register(in_dev, &in_dev->cnf);
>> + devinet_sysctl_register(dev, &in_dev->cnf);
>
> How about just giving it in_dev instead?

Hmm... Makes sense. Should I recreate the while set or
just make the incremental one?

> Thanks,

Thanks,
Pavel

Subject: Re: [PATCH net-2.6.25 2/3][IPV4] Unify and cleanup calls to devinet_sysctl_register
Posted by Herbert Xu on Sat, 01 Dec 2007 13:28:59 GMT
View Forum Message <> Reply to Message

On Sat, Dec 01, 2007 at 04:25:21PM +0300, Pavel Emelyanov wrote:
>
> > How about just giving it in_dev instead?
>
> Hmm... Makes sense. Should I recreate the while set or
> just make the incremental one?

I've applied 1/3 for both cases so please just resend 2/3 and 3/3.

Thanks,
--
Visit Openswan at http://www.openswan.org/
Email: Herbert Xu ~{PmV>HI~} <herbert@gondor.apana.org.au>
Home Page: http://gondor.apana.org.au/~herbert/
PGP Key: http://gondor.apana.org.au/~herbert/pubkey.txt

Subject: [PATCH net-2.6.25 (resend) 2/3][IPV4] Unify and cleanup calls to devinet_sysctl_register
Posted by Pavel Emelianov on Sat, 01 Dec 2007 13:38:20 GMT
View Forum Message <> Reply to Message

Currently this call is used to register sysctls for devices
and for the "default" confs. The "all" sysctls are registered
separately.

Besides, the inet_device is passed to this function, but it is
not needed there at all - just the device name and ifindex are
required.

Thanks to Herbert, who noticed, that this call doesn't even
require the devconf pointer (the last argument) - all we need
we can take from the in_device itself.

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

The original devinet_sysctl_register() works with struct net_device,
not the inet_device, and calls the introduced function, passing
the device name and ifindex (to be used as procname and ctl_name)
into it.

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

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 34c34c6..385896f 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -98,8 +98,7 @@ static BLOCKING_NOTIFIER_HEAD(inetaddr_chain);
 static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
     int destroy);
 #ifdef CONFIG_SYSCTL
-static void devinet_sysctl_register(struct in_device *in_dev,
-        struct ipv4_devconf *p);
+static void devinet_sysctl_register(struct in_device *idev);
 static void devinet_sysctl_unregister(struct ipv4_devconf *p);
 #endif

@@ -173,7 +172,7 @@ static struct in_device *inetdev_init(struct net_device *dev)
  in_dev_hold(in_dev);

 #ifdef CONFIG_SYSCTL
- devinet_sysctl_register(in_dev, &in_dev->cnf);
+ devinet_sysctl_register(in_dev);
```

```
 #endif
  ip_mc_init_dev(in_dev);
  if (dev->flags & IFF_UP)
@@ -1120,7 +1119,7 @@ static int inetdev_event(struct notifier_block *this, unsigned long event,
    neigh_sysctl_unregister(in_dev->arp_parms);
    neigh_sysctl_register(dev, in_dev->arp_parms, NET_IPV4,
           NET_IPV4_NEIGH, "ipv4", NULL, NULL);
-   devinet_sysctl_register(in_dev, &in_dev->cnf);
+   devinet_sysctl_register(in_dev);
 #endif
    break;
  }
@@ -1502,13 +1501,11 @@ static struct devinet_sysctl_table {
  },
 };

-static void devinet_sysctl_register(struct in_device *in_dev,
-         struct ipv4_devconf *p)
+static void __devinet_sysctl_register(char *dev_name, int ctl_name,
+   struct ipv4_devconf *p)
 {
  int i;
- struct net_device *dev = in_dev ? in_dev->dev : NULL;
  struct devinet_sysctl_table *t;
- char *dev_name = NULL;

  t = kmemdup(&devinet_sysctl, sizeof(*t), GFP_KERNEL);
  if (!t)
@@ -1519,13 +1516,7 @@ static void devinet_sysctl_register(struct in_device *in_dev,
    t->devinet_vars[i].extra1 = p;
  }

- if (dev) {
-   dev_name = dev->name;
-   t->devinet_dev[0].ctl_name = dev->ifindex;
- } else {
-   dev_name = "default";
-   t->devinet_dev[0].ctl_name = NET_PROTO_CONF_DEFAULT;
- }
+ t->devinet_dev[0].ctl_name = ctl_name;

  /*
   * Make a copy of dev_name, because '.procname' is regarded as const
@@ -1557,6 +1548,12 @@ out:
  return;
 }

+static void devinet_sysctl_register(struct in_device *idev)
```

```
+{
+ return __devinet_sysctl_register(idev->dev->name, idev->dev->ifindex,
+    &idev->cnf);
+}
+
 static void devinet_sysctl_unregister(struct ipv4_devconf *p)
 {
  if (p->sysctl) {
@@ -1578,9 +1575,10 @@ 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.sysctl_header =
-  register_sysctl_table(devinet_sysctl.devinet_root_dir);
- devinet_sysctl_register(NULL, &ipv4_devconf_dflt);
+ __devinet_sysctl_register("all", NET_PROTO_CONF_ALL,
+    &ipv4_devconf);
+ __devinet_sysctl_register("default", NET_PROTO_CONF_DEFAULT,
+    &ipv4_devconf_dflt);
 #endif
 }
```

Subject: [PATCH net-2.6.25 (resend) 3/3][IPV4] Use ctl paths to register devinet sysctls
Posted by Pavel Emelianov on Sat, 01 Dec 2007 13:39:58 GMT
View Forum Message <> Reply to Message

This looks very much like the patch for neighbors.

The path is also located on the stack and is prepared
inside the function. This time, the call to the registering
function is guarded with the RTNL lock, but I decided
to keep it on the stack not to litter the devinet.c file
with unneeded names and to make it look similar to the
neighbors code.

This is also intended to help us with the net namespaces
and saves the vmlinux size as well - this time by more
than 670 bytes.

The difference from the first version is just the patch
offsets, that changed due to changes in the patch #2.

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

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 385896f..c19c8db 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1431,11 +1431,8 @@ int ipv4_doint_and_flush_strategy(ctl_table *table, int __user *name,
int nlen,

 static struct devinet_sysctl_table {
  struct ctl_table_header *sysctl_header;
- ctl_table  devinet_vars[__NET_IPV4_CONF_MAX];
- ctl_table  devinet_dev[2];
- ctl_table  devinet_conf_dir[2];
- ctl_table  devinet_proto_dir[2];
- ctl_table  devinet_root_dir[2];
+ struct ctl_table devinet_vars[__NET_IPV4_CONF_MAX];
+ char *dev_name;
 } devinet_sysctl = {
  .devinet_vars = {
   DEVINET_SYSCTL_COMPLEX_ENTRY(FORWARDING, "forwarding",
@@ -1467,38 +1464,6 @@ static struct devinet_sysctl_table {
   DEVINET_SYSCTL_FLUSHING_ENTRY(PROMOTE_SECONDARIES,
           "promote_secondaries"),
  },
- .devinet_dev = {
-  {
-   .ctl_name = NET_PROTO_CONF_ALL,
-   .procname = "all",
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_vars,
-  },
- },
- .devinet_conf_dir = {
-  {
-   .ctl_name = NET_IPV4_CONF,
-   .procname = "conf",
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_dev,
-  },
- },
- .devinet_proto_dir = {
-  {
-   .ctl_name = NET_IPV4,
-   .procname = "ipv4",
-   .mode  = 0555,
-   .child   = devinet_sysctl.devinet_conf_dir,
-  },
- },
- .devinet_root_dir = {
```

```
-  {
-   .ctl_name = CTL_NET,
-   .procname  = "net",
-   .mode  = 0555,
-   .child  = devinet_sysctl.devinet_proto_dir,
-  },
- },
 };

 static void __devinet_sysctl_register(char *dev_name, int ctl_name,
@@ -1507,6 +1472,16 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
  int i;
  struct devinet_sysctl_table *t;

+#define DEVINET_CTL_PATH_DEV 3
+
+ struct ctl_path devinet_ctl_path[] = {
+  { .procname = "net", .ctl_name = CTL_NET, },
+  { .procname = "ipv4", .ctl_name = NET_IPV4, },
+  { .procname = "conf", .ctl_name = NET_IPV4_CONF, },
+  { /* to be set */ },
+  { },
+ };
+
  t = kmemdup(&devinet_sysctl, sizeof(*t), GFP_KERNEL);
  if (!t)
   goto out;
@@ -1516,24 +1491,20 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
   t->devinet_vars[i].extra1 = p;
  }

- t->devinet_dev[0].ctl_name = ctl_name;
-
  /*
   * Make a copy of dev_name, because '.procname' is regarded as const
   * by sysctl and we wouldn't want anyone to change it under our feet
   * (see SIOCSIFNAME).
   */
- dev_name = kstrdup(dev_name, GFP_KERNEL);
- if (!dev_name)
+ t->dev_name = kstrdup(dev_name, GFP_KERNEL);
+ if (!t->dev_name)
   goto free;

- t->devinet_dev[0].procname    = dev_name;
- t->devinet_dev[0].child       = t->devinet_vars;
- t->devinet_conf_dir[0].child  = t->devinet_dev;
- t->devinet_proto_dir[0].child = t->devinet_conf_dir;
```

```
- t->devinet_root_dir[0].child  = t->devinet_proto_dir;
+ 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_table(t->devinet_root_dir);
+ t->sysctl_header = register_sysctl_paths(devinet_ctl_path,
+    t->devinet_vars);
  if (!t->sysctl_header)
   goto free_procname;

@@ -1541,7 +1512,7 @@ static void __devinet_sysctl_register(char *dev_name, int ctl_name,
  return;

 free_procname:
- kfree(dev_name);
+ kfree(t->dev_name);
 free:
  kfree(t);
 out:
@@ -1560,7 +1531,7 @@ static void devinet_sysctl_unregister(struct ipv4_devconf *p)
   struct devinet_sysctl_table *t = p->sysctl;
   p->sysctl = NULL;
   unregister_sysctl_table(t->sysctl_header);
-  kfree(t->devinet_dev[0].procname);
+  kfree(t->dev_name);
   kfree(t);
  }
 }
--
1.5.3.4
```

Subject: Re: [PATCH net-2.6.25 (resend) 3/3][IPV4] Use ctl paths to register devinet sysctls
Posted by Herbert Xu on Sat, 01 Dec 2007 13:57:43 GMT
View Forum Message <> Reply to Message

On Sat, Dec 01, 2007 at 04:39:58PM +0300, Pavel Emelyanov wrote:
>
> The difference from the first version is just the patch
> offsets, that changed due to changes in the patch #2.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

All applied to net-2.6.25. Thanks Pavel.
--
Visit Openswan at http://www.openswan.org/
Email: Herbert Xu ~{PmV>HI~} <herbert@gondor.apana.org.au>

Home Page: http://gondor.apana.org.au/~herbert/
PGP Key: http://gondor.apana.org.au/~herbert/pubkey.txt