
Subject: [PATCH net-2.6.25 9/11][INET] Merge sys.net.ipv4.ip_forward and sys.net.ipv4.conf.all.forwarding

Posted by [Pavel Emelianov](#) on Tue, 04 Dec 2007 10:15:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

AFAIS these two entries should do the same thing - change the forwarding state on ipv4_devconf and on all the devices.

I propose to merge the handlers together using ctl paths.

The inet_forward_change() is static after this and I move it higher to be closer to other "propagation" helpers and to avoid diff making patches based on { and } matching :) i.e. - make them easier to read.

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

```
diff --git a/include/linux/inetdevice.h b/include/linux/inetdevice.h
index d83fee2..dd093ea 100644
--- a/include/linux/inetdevice.h
+++ b/include/linux/inetdevice.h
@@ -135,7 +135,6 @@ extern struct in_device *inetdev_by_index(int);
extern __be32 inet_select_addr(const struct net_device *dev, __be32 dst, int scope);
extern __be32 inet_confirm_addr(const struct net_device *dev, __be32 dst, __be32 local, int scope);
extern struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix, __be32 mask);
-extern void inet_forward_change(void);
```

```
static __inline__ int inet_ifa_match(__be32 addr, struct in_ifaddr *ifa)
```

```
{
```

```
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
```

```
index c19c8db..0b5f042 100644
```

```
--- a/net/ipv4/devinet.c
```

```
+++ b/net/ipv4/devinet.c
```

```
@@ -1264,6 +1264,28 @@ static void devinet_copy_dflt_conf(int i)
```

```
    read_unlock(&dev_base_lock);
```

```
}
```

```
+static void inet_forward_change(void)
```

```
+{
```

```
+ struct net_device *dev;
```

```
+ int on = IPV4_DEVCONF_ALL(FORWARDING);
```

```
+
```

```
+ IPV4_DEVCONF_ALL(ACCEPT_REDIRECTS) = !on;
```

```
+ IPV4_DEVCONF_DFLT(FORWARDING) = on;
```

```
+
```

```

+ read_lock(&dev_base_lock);
+ for_each_netdev(&init_net, dev) {
+ struct in_device *in_dev;
+ rcu_read_lock();
+ in_dev = __in_dev_get_rcu(dev);
+ if (in_dev)
+ IN_DEV_CONF_SET(in_dev, FORWARDING, on);
+ rcu_read_unlock();
+ }
+ read_unlock(&dev_base_lock);
+
+ rt_cache_flush(0);
+}
+
static int devinet_conf_proc(ctl_table *ctl, int write,
                             struct file* filp, void __user *buffer,
                             size_t *lenp, loff_t *ppos)
@@ -1333,28 +1355,6 @@ static int devinet_conf_sysctl(ctl_table *table, int __user *name, int
nlen,
    return 1;
}

-void inet_forward_change(void)
-{
- struct net_device *dev;
- int on = IPV4_DEVCONF_ALL(FORWARDING);
-
- IPV4_DEVCONF_ALL(ACCEPT_REDIRECTS) = !on;
- IPV4_DEVCONF_DFLT(FORWARDING) = on;
-
- read_lock(&dev_base_lock);
- for_each_netdev(&init_net, dev) {
- struct in_device *in_dev;
- rcu_read_lock();
- in_dev = __in_dev_get_rcu(dev);
- if (in_dev)
- IN_DEV_CONF_SET(in_dev, FORWARDING, on);
- rcu_read_unlock();
- }
- read_unlock(&dev_base_lock);
-
- rt_cache_flush(0);
-}
-
static int devinet_sysctl_forward(ctl_table *ctl, int write,
                                  struct file* filp, void __user *buffer,
                                  size_t *lenp, loff_t *ppos)
@@ -1537,6 +1537,27 @@ static void devinet_sysctl_unregister(struct ipv4_devconf *p)

```

```

}
#endif

+static struct ctl_table ctl_forward_entry[] = {
+ {
+ .ctl_name = NET_IPV4_FORWARD,
+ .procname = "ip_forward",
+ .data = &ipv4_devconf.data[
+   NET_IPV4_CONF_FORWARDING - 1],
+ .maxlen = sizeof(int),
+ .mode = 0644,
+ .proc_handler = devinet_sysctl_forward,
+ .strategy = devinet_conf_sysctl,
+ .extra1 = &ipv4_devconf,
+ },
+ { },
+};
+
+static __initdata struct ctl_path net_ipv4_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { .procname = "ipv4", .ctl_name = NET_IPV4, },
+ { },
+};
+
+void __init devinet_init(void)
+{
+   register_gifconf(PF_INET, inet_gifconf);
@@ -1550,6 +1571,7 @@ void __init devinet_init(void)
+   &ipv4_devconf);
+   __devinet_sysctl_register("default", NET_PROTO_CONF_DEFAULT,
+   &ipv4_devconf_dflt);
+ register_sysctl_paths(net_ipv4_path, ctl_forward_entry);
#endif
}

```

diff --git a/net/ipv4/sysctl_net_ipv4.c b/net/ipv4/sysctl_net_ipv4.c

index bfd0dec..844f26f 100644

--- a/net/ipv4/sysctl_net_ipv4.c

+++ b/net/ipv4/sysctl_net_ipv4.c

@@ -27,62 +27,6 @@ static int tcp_retr1_max = 255;

static int ip_local_port_range_min[] = { 1, 1 };

static int ip_local_port_range_max[] = { 65535, 65535 };

-static

-int ipv4_sysctl_forward(ctl_table *ctl, int write, struct file * filp,

- void __user *buffer, size_t *lenp, loff_t *ppos)

-{

- int val = IPV4_DEVCONF_ALL(FORWARDING);

```

- int ret;
-
- ret = proc_dointvec(ctl, write, filp, buffer, lenp, ppos);
-
- if (write && IPV4_DEVCONF_ALL(FORWARDING) != val)
-   inet_forward_change();
-
- return ret;
-}
-
-static int ipv4_sysctl_forward_strategy(ctl_table *table,
-   int __user *name, int nlen,
-   void __user *oldval, size_t __user *oldlenp,
-   void __user *newval, size_t newlen)
-{
- int *valp = table->data;
- int new;
-
- if (!newval || !newlen)
-   return 0;
-
- if (newlen != sizeof(int))
-   return -EINVAL;
-
- if (get_user(new, (int __user *)newval))
-   return -EFAULT;
-
- if (new == *valp)
-   return 0;
-
- if (oldval && oldlenp) {
-   size_t len;
-
-   if (get_user(len, oldlenp))
-     return -EFAULT;
-
-   if (len) {
-     if (len > table->maxlen)
-       len = table->maxlen;
-     if (copy_to_user(oldval, valp, len))
-       return -EFAULT;
-     if (put_user(len, oldlenp))
-       return -EFAULT;
-   }
- }
-
- *valp = new;
- inet_forward_change();

```

```

- return 1;
-}
-
extern seqlock_t sysctl_port_range_lock;
extern int sysctl_local_port_range[2];

@@ -282,15 +226,6 @@ static struct ctl_table ipv4_table[] = {
    .proc_handler = &proc_dointvec
},
{
- .ctl_name = NET_IPV4_FORWARD,
- .procname = "ip_forward",
- .data = &IPV4_DEVCONF_ALL(FORWARDING),
- .maxlen = sizeof(int),
- .mode = 0644,
- .proc_handler = &ipv4_sysctl_forward,
- .strategy = &ipv4_sysctl_forward_strategy
- },
- {
    .ctl_name = NET_IPV4_DEFAULT_TTL,
    .procname = "ip_default_ttl",
    .data = &sysctl_ip_default_ttl,
--

```

1.5.3.4
