Subject: [PATCH 1/2 net-2.6.25] [IPV4] Remove extra argument from arp_ignore.
Posted by den on Mon, 14 Jan 2008 14:04:40 GMT
View Forum Message <> Reply to Message

arp_ignore has two arguments: dev & in_dev. dev is used for inet_confirm_addr
calling only.

inet_confirm_addr, in turn, either gets in_dev from the device passed or
iterates over all network devices if the device passed is NULL. It seems
logical to directly pass in_dev into inet_confirm_addr.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/linux/inetdevice.h | 2 +-
 net/ipv4/arp.c | 11 +++++------
 net/ipv4/devinet.c | 17 ++++++++-----------
 3 files changed, 12 insertions(+), 18 deletions(-)

diff --git a/include/linux/inetdevice.h b/include/linux/inetdevice.h
index b3c5081..45f3731 100644
--- a/include/linux/inetdevice.h
+++ b/include/linux/inetdevice.h
@@ -135,7 +135,7 @@ extern int devinet_ioctl(unsigned int cmd, void __user *);
 extern void devinet_init(void);
 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 __be32 inet_confirm_addr(struct in_device *in_dev, __be32 dst, __be32 local, int scope);
 extern struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix, __be32 mask);

 static __inline__ int inet_ifa_match(__be32 addr, struct in_ifaddr *ifa)
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index e944d98..f38e1a9 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -382,8 +382,7 @@ static void arp_solicit(struct neighbour *neigh, struct sk_buff *skb)
   read_unlock_bh(&neigh->lock);
 }

-static int arp_ignore(struct in_device *in_dev, struct net_device *dev,
-       __be32 sip, __be32 tip)
+static int arp_ignore(struct in_device *in_dev, __be32 sip, __be32 tip)
 {
  int scope;

@@ -403,7 +402,7 @@ static int arp_ignore(struct in_device *in_dev, struct net_device *dev,
  case 3: /* Do not reply for scope host addresses */

```
   sip = 0;
   scope = RT_SCOPE_LINK;
-  dev = NULL;
+  in_dev = NULL;
   break;
  case 4: /* Reserved */
  case 5:
@@ -415,7 +414,7 @@ static int arp_ignore(struct in_device *in_dev, struct net_device *dev,
  default:
   return 0;
  }
- return !inet_confirm_addr(dev, sip, tip, scope);
+ return !inet_confirm_addr(in_dev, sip, tip, scope);
 }

 static int arp_filter(__be32 sip, __be32 tip, struct net_device *dev)
@@ -807,7 +806,7 @@ static int arp_process(struct sk_buff *skb)
  if (sip == 0) {
   if (arp->ar_op == htons(ARPOP_REQUEST) &&
       inet_addr_type(&init_net, tip) == RTN_LOCAL &&
-      !arp_ignore(in_dev,dev,sip,tip))
+      !arp_ignore(in_dev, sip, tip))
    arp_send(ARPOP_REPLY, ETH_P_ARP, sip, dev, tip, sha,
      dev->dev_addr, sha);
   goto out;
@@ -825,7 +824,7 @@ static int arp_process(struct sk_buff *skb)
     int dont_send = 0;

     if (!dont_send)
-     dont_send |= arp_ignore(in_dev,dev,sip,tip);
+     dont_send |= arp_ignore(in_dev,sip,tip);
     if (!dont_send && IN_DEV_ARPFILTER(in_dev))
      dont_send |= arp_filter(sip,tip,dev);
     if (!dont_send)
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 03db15b..dc1665a 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -968,24 +968,19 @@ static __be32 confirm_addr_indev(struct in_device *in_dev, __be32
dst,

 /*
  * Confirm that local IP address exists using wildcards:
- * - dev: only on this interface, 0=any interface
+ * - in_dev: only on this interface, 0=any interface
  * - dst: only in the same subnet as dst, 0=any dst
  * - local: address, 0=autoselect the local address
  * - scope: maximum allowed scope value for the local address
```

```
 */
-__be32 inet_confirm_addr(const struct net_device *dev, __be32 dst, __be32 local, int scope)
+__be32 inet_confirm_addr(struct in_device *in_dev,
+    __be32 dst, __be32 local, int scope)
 {
  __be32 addr = 0;
- struct in_device *in_dev;
-
- if (dev) {
-  rcu_read_lock();
-  if ((in_dev = __in_dev_get_rcu(dev)))
-   addr = confirm_addr_indev(in_dev, dst, local, scope);
-  rcu_read_unlock();
+ struct net_device *dev;

-  return addr;
- }
+ if (in_dev != NULL)
+  return confirm_addr_indev(in_dev, dst, local, scope);

  read_lock(&dev_base_lock);
  rcu_read_lock();
--
1.5.3.rc5
```