Subject: [PATCH] Cleanup calling netdev notifiers
Posted by Pavel Emelianov on Fri, 14 Sep 2007 10:34:22 GMT
View Forum Message <> Reply to Message

The call_netdev_notifiers routine can successfully be used in
the net/core_dev.c itself.

This will save 6 lines of code and 62 ;) bytes of .text section.

62 is rather small, but I have one more patch saving ~30 bytes
from netns code (sent to Eric), so altogether they can save
some more noticeable amount.

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

```
---

 net/core/dev.c |   32 +++++++++++++-------------------
 1 files changed, 13 insertions(+), 19 deletions(-)

diff --git a/net/core/dev.c b/net/core/dev.c
index a76021c..c1c292f 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -850,7 +850,7 @@ rollback:
  hlist_add_head(&dev->name_hlist, dev_name_hash(dev->name));
  write_unlock_bh(&dev_base_lock);

- ret = raw_notifier_call_chain(&netdev_chain, NETDEV_CHANGENAME, dev);
+ ret = call_netdevice_notifiers(NETDEV_CHANGENAME, dev);
  ret = notifier_to_errno(ret);

  if (ret) {
@@ -876,7 +876,7 @@ rollback:
  */
 void netdev_features_change(struct net_device *dev)
 {
- raw_notifier_call_chain(&netdev_chain, NETDEV_FEAT_CHANGE, dev);
+ call_netdevice_notifiers(NETDEV_FEAT_CHANGE, dev);
 }
 EXPORT_SYMBOL(netdev_features_change);

@@ -891,8 +891,7 @@ EXPORT_SYMBOL(netdev_features_change);
 void netdev_state_change(struct net_device *dev)
 {
  if (dev->flags & IFF_UP) {
-  raw_notifier_call_chain(&netdev_chain,
-    NETDEV_CHANGE, dev);
```

```
+  call_netdevice_notifiers(NETDEV_CHANGE, dev);
   rtmsg_ifinfo(RTM_NEWLINK, dev, 0);
  }
 }
@@ -988,7 +987,7 @@ int dev_open(struct net_device *dev)
   /*
    * ... and announce new interface.
    */
-  raw_notifier_call_chain(&netdev_chain, NETDEV_UP, dev);
+  call_netdevice_notifiers(NETDEV_UP, dev);
  }
  return ret;
 }
@@ -1011,7 +1010,7 @@ int dev_close(struct net_device *dev)
   * Tell people we are going down, so that they can
   * prepare to death, when device is still operating.
   */
- raw_notifier_call_chain(&netdev_chain, NETDEV_GOING_DOWN, dev);
+ call_netdevice_notifiers(NETDEV_GOING_DOWN, dev);

  dev_deactivate(dev);

@@ -1048,7 +1047,7 @@ int dev_close(struct net_device *dev)
  /*
   * Tell people we are down
   */
- raw_notifier_call_chain(&netdev_chain, NETDEV_DOWN, dev);
+ call_netdevice_notifiers(NETDEV_DOWN, dev);

  return 0;
 }
@@ -2906,8 +2905,7 @@ int dev_change_flags(struct net_device *
  if (dev->flags & IFF_UP &&
      ((old_flags ^ dev->flags) &~ (IFF_UP | IFF_PROMISC | IFF_ALLMULTI |
        IFF_VOLATILE)))
-  raw_notifier_call_chain(&netdev_chain,
-    NETDEV_CHANGE, dev);
+  call_netdevice_notifiers(NETDEV_CHANGE, dev);

  if ((flags ^ dev->gflags) & IFF_PROMISC) {
   int inc = (flags & IFF_PROMISC) ? +1 : -1;
@@ -2953,8 +2951,7 @@ int dev_set_mtu(struct net_device *dev,
  else
   dev->mtu = new_mtu;
  if (!err && dev->flags & IFF_UP)
-  raw_notifier_call_chain(&netdev_chain,
-    NETDEV_CHANGEMTU, dev);
+  call_netdevice_notifiers(NETDEV_CHANGEMTU, dev);
```

```
  return err;
 }

@@ -2970,8 +2967,7 @@ int dev_set_mac_address(struct net_devic
    return -ENODEV;
  err = dev->set_mac_address(dev, sa);
  if (!err)
-  raw_notifier_call_chain(&netdev_chain,
-    NETDEV_CHANGEADDR, dev);
+  call_netdevice_notifiers(NETDEV_CHANGEADDR, dev);
  return err;
 }

@@ -3027,8 +3023,7 @@ static int dev_ifsioc(struct ifreq *ifr,
     return -EINVAL;
    memcpy(dev->broadcast, ifr->ifr_hwaddr.sa_data,
           min(sizeof ifr->ifr_hwaddr.sa_data, (size_t) dev->addr_len));
-   raw_notifier_call_chain(&netdev_chain,
-         NETDEV_CHANGEADDR, dev);
+   call_netdevice_notifiers(NETDEV_CHANGEADDR, dev);
    return 0;

   case SIOCGIFMAP:
@@ -3475,7 +3470,7 @@ int register_netdevice(struct net_device
  write_unlock_bh(&dev_base_lock);

  /* Notify protocols, that a new device appeared. */
- ret = raw_notifier_call_chain(&netdev_chain, NETDEV_REGISTER, dev);
+ ret = call_netdevice_notifiers(NETDEV_REGISTER, dev);
  ret = notifier_to_errno(ret);
  if (ret)
   unregister_netdevice(dev);
@@ -3546,8 +3541,7 @@ static void netdev_wait_allrefs(struct n
    rtnl_lock();

    /* Rebroadcast unregister notification */
-   raw_notifier_call_chain(&netdev_chain,
-         NETDEV_UNREGISTER, dev);
+   call_netdevice_notifiers(NETDEV_UNREGISTER, dev);

    if (test_bit(__LINK_STATE_LINKWATCH_PENDING,
         &dev->state)) {
@@ -3794,7 +3788,7 @@ void unregister_netdevice(struct net_dev
  /* Notify protocols, that we are about to destroy
     this device. They should clean all the things.
  */
- raw_notifier_call_chain(&netdev_chain, NETDEV_UNREGISTER, dev);
+ call_netdevice_notifiers(NETDEV_UNREGISTER, dev);
```

```
/*
 * Flush the unicast and multicast chains
```