Subject: [patch 1/4] Network namespaces: cleanup of dev_base list use
Posted by Andrey Savochkin on Mon, 26 Jun 2006 09:49:45 GMT
View Forum Message <> Reply to Message

Cleanup of dev_base list use, with the aim to make device list per-namespace.
In almost every occasion, use of dev_base variable and dev->next pointer
could be easily replaced by for_each_netdev loop.
A few most complicated places were converted to using
first_netdev()/next_netdev().

Signed-off-by: Andrey Savochkin <saw@swsoft.com>

```
 arch/s390/appldata/appldata_net_sum.c |    2
 arch/sparc64/solaris/ioctl.c          |    2
 drivers/block/aoe/aoecmd.c            |    8 ++-
 drivers/net/wireless/strip.c          |    4 -
 drivers/parisc/led.c                  |    2
 include/linux/netdevice.h             |   28 ++++++++++--
 net/8021q/vlan.c                      |    4 -
 net/8021q/vlanproc.c                  |   10 ++--
 net/bridge/br_if.c                    |    4 -
 net/bridge/br_ioctl.c                 |    4 +
 net/bridge/br_netlink.c               |    3 -
 net/core/dev.c                        |   70 ++++++++++++++++++----------------
 net/core/dev_mcast.c                  |    4 -
 net/core/rtnetlink.c                  |   18 ++++----
 net/decnet/af_decnet.c                |   11 +++--
 net/decnet/dn_dev.c                   |   17 ++++----
 net/decnet/dn_fib.c                   |    2
 net/decnet/dn_route.c                 |   12 ++---
 net/ipv4/devinet.c                    |   15 ++++---
 net/ipv4/igmp.c                       |   25 +++++++-----
 net/ipv6/addrconf.c                   |   28 ++++++++-----
 net/ipv6/anycast.c                    |   22 ++++++----
 net/ipv6/mcast.c                      |   20 +++++---
 net/llc/llc_core.c                    |    7 ++-
 net/netrom/nr_route.c                 |    4 -
 net/rose/rose_route.c                 |    8 ++-
 net/sched/sch_api.c                   |    8 ++-
 net/sctp/protocol.c                   |    2
 net/tipc/eth_media.c                  |   12 +++--
 29 files changed, 200 insertions, 156 deletions

--- ./arch/s390/appldata/appldata_net_sum.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./arch/s390/appldata/appldata_net_sum.c Thu Jun 22 12:03:07 2006
@@ -108,7 +108,7 @@ static void appldata_get_net_sum_data(vo
  tx_dropped = 0;
  collisions = 0;
```

```
  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if (dev->get_stats == NULL) {
    continue;
   }
--- ./arch/sparc64/solaris/ioctl.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./arch/sparc64/solaris/ioctl.c Thu Jun 22 12:03:07 2006
@@ -686,7 +686,7 @@ static inline int solaris_i(unsigned int
    int i = 0;

    read_lock_bh(&dev_base_lock);
-   for (d = dev_base; d; d = d->next) i++;
+   for_each_netdev(d) i++;
    read_unlock_bh(&dev_base_lock);

    if (put_user (i, (int __user *)A(arg)))
--- ./drivers/block/aoe/aoecmd.c.vedevbase Wed Jun 21 18:50:28 2006
+++ ./drivers/block/aoe/aoecmd.c Thu Jun 22 12:03:07 2006
@@ -204,14 +204,17 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigne
  sl = sl_tail = NULL;

  read_lock(&dev_base_lock);
- for (ifp = dev_base; ifp; dev_put(ifp), ifp = ifp->next) {
+ for_each_netdev(dev) {
   dev_hold(ifp);
-  if (!is_aoe_netif(ifp))
+  if (!is_aoe_netif(ifp)) {
+   dev_put(ifp);
    continue;
+  }

   skb = new_skb(ifp, sizeof *h + sizeof *ch);
   if (skb == NULL) {
    printk(KERN_INFO "aoe: aoecmd_cfg: skb alloc failure\n");
+   dev_put(ifp);
    continue;
   }
   if (sl_tail == NULL)
@@ -229,6 +232,7 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigne

   skb->next = sl;
   sl = skb;
+  dev_put(ifp);
  }
  read_unlock(&dev_base_lock);

--- ./drivers/net/wireless/strip.c.vedevbase Wed Jun 21 18:50:43 2006
```

```
+++ ./drivers/net/wireless/strip.c Thu Jun 22 12:03:07 2006
@@ -1970,8 +1970,7 @@ static struct net_device *get_strip_dev(
         sizeof(zero_address))) {
   struct net_device *dev;
   read_lock_bh(&dev_base_lock);
-  dev = dev_base;
-  while (dev) {
+  for_each_netdev(dev) {
    if (dev->type == strip_info->dev->type &&
        !memcmp(dev->dev_addr,
         &strip_info->true_dev_addr,
@@ -1982,7 +1981,6 @@ static struct net_device *get_strip_dev(
     read_unlock_bh(&dev_base_lock);
     return (dev);
    }
-   dev = dev->next;
   }
   read_unlock_bh(&dev_base_lock);
  }
--- ./drivers/parisc/led.c.vedevbase Wed Jun 21 18:52:58 2006
+++ ./drivers/parisc/led.c Thu Jun 22 12:03:07 2006
@@ -368,7 +368,7 @@ static __inline__ int led_get_net_activi
   * for reading should be OK */
  read_lock(&dev_base_lock);
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
      struct net_device_stats *stats;
      struct in_device *in_dev = __in_dev_get_rcu(dev);
      if (!in_dev || !in_dev->ifa_list)
--- ./include/linux/netdevice.h.vedevbase Wed Jun 21 18:53:17 2006
+++ ./include/linux/netdevice.h Thu Jun 22 18:57:50 2006
@@ -289,8 +289,8 @@ struct net_device

  unsigned long  state;

- struct net_device *next;
-
+ struct list_head dev_list;
+
  /* The device initialization function. Called only once. */
  int   (*init)(struct net_device *dev);

@@ -543,9 +543,27 @@ struct packet_type {
 #include <linux/interrupt.h>
 #include <linux/notifier.h>

-extern struct net_device  loopback_dev;  /* The loopback */
```

```
-extern struct net_device  *dev_base;  /* All devices */
-extern rwlock_t    dev_base_lock;  /* Device list lock */
+extern struct net_device loopback_dev;  /* The loopback */
+extern struct list_head  dev_base_head;  /* All devices */
+extern rwlock_t   dev_base_lock;  /* Device list lock */
+
+#define for_each_netdev(p) list_for_each_entry(p, &dev_base_head, dev_list)
+
+/* DO NOT USE first_netdev/next_netdev, use loop defined above */
+#define first_netdev()  ({ \
+     list_empty(&dev_base_head) ? NULL : \
+      list_entry(dev_base_head.next, \
+       struct net_device, \
+       dev_list); \
+     })
+#define next_netdev(dev) ({ \
+     struct list_head *__next; \
+     __next = (dev)->dev_list.next; \
+     __next == &dev_base_head ? NULL : \
+      list_entry(__next, \
+       struct net_device, \
+       dev_list); \
+     })

 extern int    netdev_boot_setup_check(struct net_device *dev);
 extern unsigned long  netdev_boot_base(const char *prefix, int unit);
--- ./net/8021q/vlan.c.vedevbase Wed Jun 21 18:51:08 2006
+++ ./net/8021q/vlan.c Thu Jun 22 12:03:07 2006
@@ -121,8 +121,8 @@ static void __exit vlan_cleanup_devices(
  struct net_device *dev, *nxt;

  rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
-  nxt = dev->next;
+ for (dev = first_netdev(); dev; dev = nxt) {
+  nxt = next_netdev(dev);
   if (dev->priv_flags & IFF_802_1Q_VLAN) {
    unregister_vlan_dev(VLAN_DEV_INFO(dev)->real_dev,
         VLAN_DEV_INFO(dev)->vlan_id);
--- ./net/8021q/vlanproc.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./net/8021q/vlanproc.c Thu Jun 22 12:03:07 2006
@@ -242,7 +242,7 @@ int vlan_proc_rem_dev(struct net_device
 static struct net_device *vlan_skip(struct net_device *dev)
 {
  while (dev && !(dev->priv_flags & IFF_802_1Q_VLAN))
-  dev = dev->next;
+  dev = next_netdev(dev);
```

```
  return dev;
 }
@@ -258,8 +258,8 @@ static void *vlan_seq_start(struct seq_f
  if (*pos == 0)
   return SEQ_START_TOKEN;

- for (dev = vlan_skip(dev_base); dev && i < *pos;
-      dev = vlan_skip(dev->next), ++i);
+ for (dev = vlan_skip(first_netdev()); dev && i < *pos;
+      dev = vlan_skip(next_netdev(dev)), ++i);

  return  (i == *pos) ? dev : NULL;
 }
@@ -269,8 +269,8 @@ static void *vlan_seq_next(struct seq_fi
  ++*pos;

  return vlan_skip((v == SEQ_START_TOKEN)
-       ? dev_base
-       : ((struct net_device *)v)->next);
+       ? first_netdev()
+       : next_netdev((struct net_device *)v));
 }

 static void vlan_seq_stop(struct seq_file *seq, void *v)
--- ./net/bridge/br_if.c.vedevbase Wed Jun 21 18:53:18 2006
+++ ./net/bridge/br_if.c Thu Jun 22 12:03:07 2006
@@ -468,8 +468,8 @@ void __exit br_cleanup_bridges(void)
  struct net_device *dev, *nxt;

  rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
-  nxt = dev->next;
+ for (dev = first_netdev(); dev; dev = nxt) {
+  nxt = next_netdev(dev);
   if (dev->priv_flags & IFF_EBRIDGE)
    del_br(dev->priv);
  }
--- ./net/bridge/br_ioctl.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./net/bridge/br_ioctl.c Thu Jun 22 12:03:07 2006
@@ -27,7 +27,9 @@ static int get_bridge_ifindices(int *ind
  struct net_device *dev;
  int i = 0;

- for (dev = dev_base; dev && i < num; dev = dev->next) {
+ for_each_netdev(dev) {
+  if (i >= num)
+   break;
   if (dev->priv_flags & IFF_EBRIDGE)
```

```
    indices[i++] = dev->ifindex;
  }
--- ./net/bridge/br_netlink.c.vedevbase Wed Jun 21 18:53:18 2006
+++ ./net/bridge/br_netlink.c Thu Jun 22 12:03:07 2006
@@ -109,7 +109,8 @@ static int br_dump_ifinfo(struct sk_buff
  int err = 0;

  read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next) {
+ idx = 0;
+ for_each_netdev(dev) {
    struct net_bridge_port *p = dev->br_port;

    /* not a bridge port */
--- ./net/core/dev.c.vedevbase Wed Jun 21 18:53:18 2006
+++ ./net/core/dev.c Thu Jun 22 17:40:13 2006
@@ -174,13 +174,12 @@ static spinlock_t net_dma_event_lock;
  * unregister_netdevice(), which must be called with the rtnl
  * semaphore held.
  */
-struct net_device *dev_base;
-static struct net_device **dev_tail = &dev_base;
 DEFINE_RWLOCK(dev_base_lock);
-
-EXPORT_SYMBOL(dev_base);
 EXPORT_SYMBOL(dev_base_lock);

+LIST_HEAD(dev_base_head);
+EXPORT_SYMBOL(dev_base_head);
+
 #define NETDEV_HASHBITS 8
 static struct hlist_head dev_name_head[1<<NETDEV_HASHBITS];
 static struct hlist_head dev_index_head[1<<NETDEV_HASHBITS];
@@ -575,7 +574,7 @@ struct net_device *dev_getbyhwaddr(unsig

  ASSERT_RTNL();

- for (dev = dev_base; dev; dev = dev->next)
+ for_each_netdev(dev)
    if (dev->type == type &&
        !memcmp(dev->dev_addr, ha, dev->addr_len))
      break;
@@ -589,7 +588,7 @@ struct net_device *dev_getfirstbyhwtype(
  struct net_device *dev;

  rtnl_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
```

```
   if (dev->type == type) {
    dev_hold(dev);
    break;
@@ -617,7 +616,7 @@ struct net_device * dev_get_by_flags(uns
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if (((dev->flags ^ if_flags) & mask) == 0) {
    dev_hold(dev);
    break;
@@ -680,7 +679,7 @@ int dev_alloc_name(struct net_device *de
   if (!inuse)
    return -ENOMEM;

-  for (d = dev_base; d; d = d->next) {
+  for_each_netdev(d) {
    if (!sscanf(d->name, name, &i))
     continue;
    if (i < 0 || i >= max_netdevices)
@@ -966,7 +965,7 @@ int register_netdevice_notifier(struct n
  rtnl_lock();
  err = raw_notifier_chain_register(&netdev_chain, nb);
  if (!err) {
-  for (dev = dev_base; dev; dev = dev->next) {
+  for_each_netdev(dev) {
    nb->notifier_call(nb, NETDEV_REGISTER, dev);

    if (dev->flags & IFF_UP)
@@ -1903,7 +1902,7 @@ static int dev_ifconf(char __user *arg)
   */

  total = 0;
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
   for (i = 0; i < NPROTO; i++) {
    if (gifconf_list[i]) {
     int done;
@@ -1935,26 +1934,25 @@ static int dev_ifconf(char __user *arg)
  * This is invoked by the /proc filesystem handler to display a device
  * in detail.
  */
-static __inline__ struct net_device *dev_get_idx(loff_t pos)
-{
- struct net_device *dev;
- loff_t i;
-
```

```
- for (i = 0, dev = dev_base; dev && i < pos; ++i, dev = dev->next);
-
- return i == pos ? dev : NULL;
-}
-
 void *dev_seq_start(struct seq_file *seq, loff_t *pos)
 {
+ struct net_device *dev;
+ loff_t off = 1;
  read_lock(&dev_base_lock);
- return *pos ? dev_get_idx(*pos - 1) : SEQ_START_TOKEN;
+ if (!*pos)
+  return SEQ_START_TOKEN;
+ for_each_netdev(dev) {
+  if (off++ == *pos)
+   return dev;
+ }
+ return NULL;
 }

 void *dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
 {
+ struct net_device *dev = v;
  ++*pos;
- return v == SEQ_START_TOKEN ? dev_base : ((struct net_device *)v)->next;
+ return v == SEQ_START_TOKEN ? first_netdev() : next_netdev(dev);
 }

 void dev_seq_stop(struct seq_file *seq, void *v)
@@ -2837,11 +2835,9 @@ int register_netdevice(struct net_device

  set_bit(__LINK_STATE_PRESENT, &dev->state);

- dev->next = NULL;
  dev_init_scheduler(dev);
  write_lock_bh(&dev_base_lock);
- *dev_tail = dev;
- dev_tail = &dev->next;
+ list_add_tail(&dev->dev_list, &dev_base_head);
  hlist_add_head(&dev->name_hlist, head);
  hlist_add_head(&dev->index_hlist, dev_index_hash(dev->ifindex));
  dev_hold(dev);
@@ -3119,8 +3115,6 @@ void synchronize_net(void)

 int unregister_netdevice(struct net_device *dev)
 {
- struct net_device *d, **dp;
-
```

```
  BUG_ON(dev_boot_phase);
  ASSERT_RTNL();

@@ -3138,23 +3132,11 @@ int unregister_netdevice(struct net_devi
    dev_close(dev);

  /* And unlink it from device chain. */
- for (dp = &dev_base; (d = *dp) != NULL; dp = &d->next) {
-  if (d == dev) {
-   write_lock_bh(&dev_base_lock);
-   hlist_del(&dev->name_hlist);
-   hlist_del(&dev->index_hlist);
-   if (dev_tail == &dev->next)
-    dev_tail = dp;
-   *dp = d->next;
-   write_unlock_bh(&dev_base_lock);
-   break;
-  }
- }
- if (!d) {
-  printk(KERN_ERR "unregister net_device: '%s' not found\n",
-         dev->name);
-  return -ENODEV;
- }
+ write_lock_bh(&dev_base_lock);
+ list_del(&dev->dev_list);
+ hlist_del(&dev->name_hlist);
+ hlist_del(&dev->index_hlist);
+ write_unlock_bh(&dev_base_lock);

  dev->reg_state = NETREG_UNREGISTERING;

--- ./net/core/dev_mcast.c.vedevbase Wed Jun 21 18:53:18 2006
+++ ./net/core/dev_mcast.c Thu Jun 22 12:03:08 2006
@@ -225,7 +225,7 @@ static void *dev_mc_seq_start(struct seq
  loff_t off = 0;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
   if (off++ == *pos)
    return dev;
  }
@@ -236,7 +236,7 @@ static void *dev_mc_seq_next(struct seq_
 {
  struct net_device *dev = v;
  ++*pos;
- return dev->next;
```

```
+ return next_netdev(dev);
 }

 static void dev_mc_seq_stop(struct seq_file *seq, void *v)
--- ./net/core/rtnetlink.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/core/rtnetlink.c Thu Jun 22 12:03:08 2006
@@ -320,14 +320,16 @@ static int rtnetlink_dump_ifinfo(struct
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
-  if (idx < s_idx)
-   continue;
-  if (rtnetlink_fill_ifinfo(skb, dev, RTM_NEWLINK,
-       NETLINK_CB(cb->skb).pid,
-       cb->nlh->nlmsg_seq, 0,
-       NLM_F_MULTI) <= 0)
-   break;
+ idx = 0;
+ for_each_netdev(dev) {
+  if (idx >= s_idx) {
+   if (rtnetlink_fill_ifinfo(skb, dev, RTM_NEWLINK,
+       NETLINK_CB(cb->skb).pid,
+       cb->nlh->nlmsg_seq, 0,
+       NLM_F_MULTI) <= 0)
+    break;
+  }
+  idx++;
  }
  read_unlock(&dev_base_lock);
  cb->args[0] = idx;
--- ./net/decnet/af_decnet.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/decnet/af_decnet.c Thu Jun 22 12:03:08 2006
@@ -721,7 +721,7 @@ static int dn_bind(struct socket *sock,
  struct sock *sk = sock->sk;
  struct dn_scp *scp = DN_SK(sk);
  struct sockaddr_dn *saddr = (struct sockaddr_dn *)uaddr;
- struct net_device *dev;
+ struct net_device *pdev, *dev;
  int rv;

  if (addr_len != sizeof(struct sockaddr_dn))
@@ -745,12 +745,15 @@ static int dn_bind(struct socket *sock,

  if (!(saddr->sdn_flags & SDF_WILD)) {
   if (dn_ntohs(saddr->sdn_nodeaddrl)) {
+   dev = NULL;
    read_lock(&dev_base_lock);
```

```
-   for(dev = dev_base; dev; dev = dev->next) {
-    if (!dev->dn_ptr)
+   for_each_netdev(pdev) {
+    if (!pdev->dn_ptr)
      continue;
-    if (dn_dev_islocal(dev, dn_saddr2dn(saddr)))
+    if (dn_dev_islocal(pdev, dn_saddr2dn(saddr))) {
+     dev = pdev;
      break;
+    }
    }
    read_unlock(&dev_base_lock);
   if (dev == NULL)
--- ./net/decnet/dn_dev.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/decnet/dn_dev.c Thu Jun 22 12:03:08 2006
@@ -776,13 +776,14 @@ static int dn_dev_dump_ifaddr(struct sk_
  s_idx = cb->args[0];
  s_dn_idx = dn_idx = cb->args[1];
  read_lock(&dev_base_lock);
- for(dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ idx = 0;
+ for_each_netdev(dev) {
   if (idx < s_idx)
-   continue;
+   goto cont;
   if (idx > s_idx)
    s_dn_idx = 0;
   if ((dn_db = dev->dn_ptr) == NULL)
-   continue;
+   goto cont;

   for(ifa = dn_db->ifa_list, dn_idx = 0; ifa; ifa = ifa->ifa_next, dn_idx++) {
    if (dn_idx < s_dn_idx)
@@ -795,6 +796,8 @@ static int dn_dev_dump_ifaddr(struct sk_
            NLM_F_MULTI) <= 0)
     goto done;
   }
+cont:
+  idx++;
  }
 done:
  read_unlock(&dev_base_lock);
@@ -1265,7 +1268,7 @@ void dn_dev_devices_off(void)
  struct net_device *dev;

  rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next)
+ for_each_netdev(dev)
```

```
   dn_dev_down(dev);
  rtnl_unlock();

@@ -1276,7 +1279,7 @@ void dn_dev_devices_on(void)
   struct net_device *dev;

   rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
    if (dev->flags & IFF_UP)
     dn_dev_up(dev);
   }
@@ -1297,7 +1300,7 @@ int unregister_dnaddr_notifier(struct no
  static inline struct net_device *dn_dev_get_next(struct seq_file *seq, struct net_device *dev)
  {
   do {
-   dev = dev->next;
+   dev = next_netdev(dev);
   } while(dev && !dev->dn_ptr);

   return dev;
@@ -1307,7 +1310,7 @@ static struct net_device *dn_dev_get_idx
  {
   struct net_device *dev;

- dev = dev_base;
+ dev = first_netdev();
   if (dev && !dev->dn_ptr)
    dev = dn_dev_get_next(seq, dev);
   if (pos) {
--- ./net/decnet/dn_fib.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/decnet/dn_fib.c Thu Jun 22 12:03:08 2006
@@ -631,7 +631,7 @@ static void dn_fib_del_ifaddr(struct dn_

   /* Scan device list */
   read_lock(&dev_base_lock);
- for(dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
    dn_db = dev->dn_ptr;
    if (dn_db == NULL)
     continue;
--- ./net/decnet/dn_route.c.vedevbase Wed Jun 21 18:53:19 2006
+++ ./net/decnet/dn_route.c Thu Jun 22 12:03:08 2006
@@ -923,16 +923,16 @@ static int dn_route_output_slow(struct d
     goto out;
    }
    read_lock(&dev_base_lock);
-   for(dev_out = dev_base; dev_out; dev_out = dev_out->next) {
```

```
+  for_each_netdev(dev_out) {
    if (!dev_out->dn_ptr)
     continue;
-   if (dn_dev_islocal(dev_out, oldflp->fld_src))
-    break;
+   if (dn_dev_islocal(dev_out, oldflp->fld_src)) {
+    dev_hold(dev_out);
+    goto source_ok;
+   }
   }
   read_unlock(&dev_base_lock);
-  if (dev_out == NULL)
-   goto out;
-  dev_hold(dev_out);
+  goto out;
 source_ok:
   ;
  }
--- ./net/ipv4/devinet.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/ipv4/devinet.c Thu Jun 22 12:03:08 2006
@@ -842,7 +842,7 @@ no_in_dev:
   */
  read_lock(&dev_base_lock);
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
    continue;

@@ -921,7 +921,7 @@ u32 inet_confirm_addr(const struct net_d

  read_lock(&dev_base_lock);
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((in_dev = __in_dev_get_rcu(dev))) {
    addr = confirm_addr_indev(in_dev, dst, local, scope);
    if (addr)
@@ -1095,17 +1095,18 @@ static int inet_dump_ifaddr(struct sk_bu
  struct in_ifaddr *ifa;
  int s_ip_idx, s_idx = cb->args[0];

+ idx = 0;
  s_ip_idx = ip_idx = cb->args[1];
  read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ for_each_netdev(dev) {
   if (idx < s_idx)
```

```
-   continue;
+   goto cont;
    if (idx > s_idx)
     s_ip_idx = 0;
    rcu_read_lock();
    if ((in_dev = __in_dev_get_rcu(dev)) == NULL) {
     rcu_read_unlock();
-   continue;
+   goto cont;
    }

    for (ifa = in_dev->ifa_list, ip_idx = 0; ifa;
@@ -1120,6 +1121,8 @@ static int inet_dump_ifaddr(struct sk_bu
     }
    }
    rcu_read_unlock();
+cont:
+  idx++;
   }

 done:
@@ -1171,7 +1174,7 @@ void inet_forward_change(void)
   ipv4_devconf_dflt.forwarding = on;

   read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
    struct in_device *in_dev;
    rcu_read_lock();
    in_dev = __in_dev_get_rcu(dev);
--- ./net/ipv4/igmp.c.vedevbase Wed Jun 21 18:53:19 2006
+++ ./net/ipv4/igmp.c Thu Jun 22 12:03:08 2006
@@ -2255,19 +2255,21 @@ struct igmp_mc_iter_state {

 static inline struct ip_mc_list *igmp_mc_get_first(struct seq_file *seq)
 {
+ struct net_device *dev;
   struct ip_mc_list *im = NULL;
   struct igmp_mc_iter_state *state = igmp_mc_seq_private(seq);

- for (state->dev = dev_base, state->in_dev = NULL;
-      state->dev;
-      state->dev = state->dev->next) {
+ state->dev = NULL;
+ state->in_dev = NULL;
+ for_each_netdev(dev) {
    struct in_device *in_dev;
-   in_dev = in_dev_get(state->dev);
```

```
+  in_dev = in_dev_get(dev);
   if (!in_dev)
    continue;
   read_lock(&in_dev->mc_list_lock);
   im = in_dev->mc_list;
   if (im) {
+   state->dev = dev;
    state->in_dev = in_dev;
    break;
   }
@@ -2286,7 +2288,7 @@ static struct ip_mc_list *igmp_mc_get_ne
    read_unlock(&state->in_dev->mc_list_lock);
    in_dev_put(state->in_dev);
   }
-  state->dev = state->dev->next;
+  state->dev = next_netdev(state->dev);
   if (!state->dev) {
    state->in_dev = NULL;
    break;
@@ -2417,15 +2419,17 @@ struct igmp_mcf_iter_state {

 static inline struct ip_sf_list *igmp_mcf_get_first(struct seq_file *seq)
 {
+ struct net_device *dev;
  struct ip_sf_list *psf = NULL;
  struct ip_mc_list *im = NULL;
  struct igmp_mcf_iter_state *state = igmp_mcf_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL, state->im = NULL;
-      state->dev;
-      state->dev = state->dev->next) {
+ state->dev = NULL;
+ state->im = NULL;
+ state->idev = NULL;
+ for_each_netdev(dev) {
   struct in_device *idev;
-  idev = in_dev_get(state->dev);
+  idev = in_dev_get(dev);
   if (unlikely(idev == NULL))
    continue;
   read_lock(&idev->mc_list_lock);
@@ -2434,6 +2438,7 @@ static inline struct ip_sf_list *igmp_mc
    spin_lock_bh(&im->lock);
    psf = im->sources;
    if (likely(psf != NULL)) {
+    state->dev = dev;
     state->im = im;
     state->idev = idev;
```

```
     break;
@@ -2459,7 +2464,7 @@ static struct ip_sf_list *igmp_mcf_get_n
     read_unlock(&state->idev->mc_list_lock);
     in_dev_put(state->idev);
    }
-   state->dev = state->dev->next;
+   state->dev = next_netdev(state->dev);
    if (!state->dev) {
     state->idev = NULL;
     goto out;
--- ./net/ipv6/addrconf.c.vedevbase Wed Jun 21 18:53:20 2006
+++ ./net/ipv6/addrconf.c Thu Jun 22 12:03:08 2006
@@ -470,7 +470,7 @@ static void addrconf_forward_change(void
  struct inet6_dev *idev;

  read_lock(&dev_base_lock);
- for (dev=dev_base; dev; dev=dev->next) {
+ for_each_netdev(dev) {
   read_lock(&addrconf_lock);
   idev = __in6_dev_get(dev);
   if (idev) {
@@ -889,7 +889,7 @@ int ipv6_dev_get_saddr(struct net_device
  read_lock(&dev_base_lock);
  read_lock(&addrconf_lock);

- for (dev = dev_base; dev; dev=dev->next) {
+ for_each_netdev(dev) {
   struct inet6_dev *idev;
   struct inet6_ifaddr *ifa;

@@ -1971,7 +1971,7 @@ static void sit_add_v4_addrs(struct inet
   return;
  }

-       for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   struct in_device * in_dev = __in_dev_get_rtnl(dev);
   if (in_dev && (dev->flags & IFF_UP)) {
    struct in_ifaddr * ifa;
@@ -2120,7 +2120,7 @@ static void ip6_tnl_add_linklocal(struct
    return;
  }
  /* then try to inherit it from any device */
- for (link_dev = dev_base; link_dev; link_dev = link_dev->next) {
+ for_each_netdev(link_dev) {
   if (!ipv6_inherit_linklocal(idev, link_dev))
    return;
  }
```

```
@@ -3005,18 +3005,19 @@ static int inet6_dump_addr(struct sk_buf
  struct ifmcaddr6 *ifmca;
  struct ifacaddr6 *ifaca;

+ idx = 0;
  s_idx = cb->args[0];
  s_ip_idx = ip_idx = cb->args[1];
  read_lock(&dev_base_lock);

- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ for_each_netdev(dev) {
   if (idx < s_idx)
-   continue;
+   goto cont;
   if (idx > s_idx)
    s_ip_idx = 0;
   ip_idx = 0;
   if ((idev = in6_dev_get(dev)) == NULL)
-   continue;
+   goto cont;
   read_lock_bh(&idev->lock);
   switch (type) {
   case UNICAST_ADDR:
@@ -3063,6 +3064,8 @@ static int inet6_dump_addr(struct sk_buf
   }
   read_unlock_bh(&idev->lock);
   in6_dev_put(idev);
+cont:
+  idx++;
  }
 done:
  if (err <= 0) {
@@ -3230,17 +3233,20 @@ static int inet6_dump_ifinfo(struct sk_b
  struct net_device *dev;
  struct inet6_dev *idev;

+ idx = 0;
  read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ for_each_netdev(dev) {
   if (idx < s_idx)
-   continue;
+   goto cont;
   if ((idev = in6_dev_get(dev)) == NULL)
-   continue;
+   goto cont;
   err = inet6_fill_ifinfo(skb, idev, NETLINK_CB(cb->skb).pid,
     cb->nlh->nlmsg_seq, RTM_NEWLINK, NLM_F_MULTI);
```

```
   in6_dev_put(idev);
   if (err <= 0)
    break;
+cont:
+  idx++;
  }
  read_unlock(&dev_base_lock);
  cb->args[0] = idx;
@@ -3864,7 +3870,7 @@ void __exit addrconf_cleanup(void)
   * clean dev list.
   */

- for (dev=dev_base; dev; dev=dev->next) {
+ for_each_netdev(dev) {
   if ((idev = __in6_dev_get(dev)) == NULL)
    continue;
   addrconf_ifdown(dev, 1);
--- ./net/ipv6/anycast.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/ipv6/anycast.c Thu Jun 22 12:03:08 2006
@@ -428,11 +428,13 @@ int ipv6_chk_acast_addr(struct net_devic
  if (dev)
   return ipv6_chk_acast_dev(dev, addr);
  read_lock(&dev_base_lock);
- for (dev=dev_base; dev; dev=dev->next)
-  if (ipv6_chk_acast_dev(dev, addr))
-   break;
+ for_each_netdev(dev)
+  if (ipv6_chk_acast_dev(dev, addr)) {
+   read_unlock(&dev_base_lock);
+   return 1;
+  }
  read_unlock(&dev_base_lock);
- return dev != 0;
+ return 0;
 }


@@ -446,19 +448,21 @@ struct ac6_iter_state {

 static inline struct ifacaddr6 *ac6_get_first(struct seq_file *seq)
 {
+ struct net_device *dev;
  struct ifacaddr6 *im = NULL;
  struct ac6_iter_state *state = ac6_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL;
-      state->dev;
-      state->dev = state->dev->next) {
```

```
+ state->dev = NULL;
+ state->idev = NULL;
+ for_each_netdev(dev) {
   struct inet6_dev *idev;
-  idev = in6_dev_get(state->dev);
+  idev = in6_dev_get(dev);
   if (!idev)
    continue;
   read_lock_bh(&idev->lock);
   im = idev->ac_list;
   if (im) {
+   state->dev = dev;
    state->idev = idev;
    break;
   }
@@ -477,7 +481,7 @@ static struct ifacaddr6 *ac6_get_next(st
    read_unlock_bh(&state->idev->lock);
    in6_dev_put(state->idev);
   }
-  state->dev = state->dev->next;
+  state->dev = next_netdev(state->dev);
   if (!state->dev) {
    state->idev = NULL;
    break;
--- ./net/ipv6/mcast.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/ipv6/mcast.c Thu Jun 22 12:03:08 2006
@@ -2326,9 +2326,8 @@ static inline struct ifmcaddr6 *igmp6_mc
  struct ifmcaddr6 *im = NULL;
  struct igmp6_mc_iter_state *state = igmp6_mc_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL;
-      state->dev;
-      state->dev = state->dev->next) {
+ state->idev = NULL;
+ for_each_netdev(state->dev) {
   struct inet6_dev *idev;
   idev = in6_dev_get(state->dev);
   if (!idev)
@@ -2355,7 +2354,7 @@ static struct ifmcaddr6 *igmp6_mc_get_ne
    read_unlock_bh(&state->idev->lock);
    in6_dev_put(state->idev);
   }
-  state->dev = state->dev->next;
+  state->dev = next_netdev(state->dev);
   if (!state->dev) {
    state->idev = NULL;
    break;
@@ -2466,15 +2465,17 @@ struct igmp6_mcf_iter_state {
```

```
 static inline struct ip6_sf_list *igmp6_mcf_get_first(struct seq_file *seq)
 {
+ struct net_device *dev;
  struct ip6_sf_list *psf = NULL;
  struct ifmcaddr6 *im = NULL;
  struct igmp6_mcf_iter_state *state = igmp6_mcf_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL, state->im = NULL;
-      state->dev;
-      state->dev = state->dev->next) {
+ state->dev = NULL;
+ state->im = NULL;
+ state->idev = NULL;
+ for_each_netdev(dev) {
   struct inet6_dev *idev;
-  idev = in6_dev_get(state->dev);
+  idev = in6_dev_get(dev);
   if (unlikely(idev == NULL))
    continue;
   read_lock_bh(&idev->lock);
@@ -2483,6 +2484,7 @@ static inline struct ip6_sf_list *igmp6_
    spin_lock_bh(&im->mca_lock);
    psf = im->mca_sources;
    if (likely(psf != NULL)) {
+    state->dev = dev;
     state->im = im;
     state->idev = idev;
     break;
@@ -2508,7 +2510,7 @@ static struct ip6_sf_list *igmp6_mcf_get
     read_unlock_bh(&state->idev->lock);
     in6_dev_put(state->idev);
    }
-   state->dev = state->dev->next;
+   state->dev = next_netdev(state->dev);
    if (!state->dev) {
     state->idev = NULL;
     goto out;
--- ./net/llc/llc_core.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/llc/llc_core.c Thu Jun 22 12:03:08 2006
@@ -161,8 +161,11 @@ static struct packet_type llc_tr_packet_

 static int __init llc_init(void)
 {
- if (dev_base->next)
-  memcpy(llc_station_mac_sa, dev_base->next->dev_addr, ETH_ALEN);
+ struct net_device *dev;
+
```

```
+ dev = next_netdev(first_netdev());
+ if (dev)
+  memcpy(llc_station_mac_sa, dev->dev_addr, ETH_ALEN);
  else
   memset(llc_station_mac_sa, 0, ETH_ALEN);
  dev_add_pack(&llc_packet_type);
--- ./net/netrom/nr_route.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./net/netrom/nr_route.c Thu Jun 22 12:03:08 2006
@@ -595,7 +595,7 @@ struct net_device *nr_dev_first(void)
  struct net_device *dev, *first = NULL;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM)
    if (first == NULL || strncmp(dev->name, first->name, 3) < 0)
     first = dev;
@@ -615,7 +615,7 @@ struct net_device *nr_dev_get(ax25_addre
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM && ax25cmp(addr,
(ax25_address *)dev->dev_addr) == 0) {
    dev_hold(dev);
    goto out;
--- ./net/rose/rose_route.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/rose/rose_route.c Thu Jun 22 12:03:08 2006
@@ -600,7 +600,7 @@ struct net_device *rose_dev_first(void)
  struct net_device *dev, *first = NULL;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE)
    if (first == NULL || strncmp(dev->name, first->name, 3) < 0)
     first = dev;
@@ -618,12 +618,13 @@ struct net_device *rose_dev_get(rose_add
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE && rosecmp(addr, (rose_address
*)dev->dev_addr) == 0) {
    dev_hold(dev);
    goto out;
```

```
   }
  }
+ dev = NULL;
 out:
  read_unlock(&dev_base_lock);
  return dev;
@@ -634,10 +635,11 @@ static int rose_dev_exists(rose_address
  struct net_device *dev;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for_each_netdev(dev) {
   if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE && rosecmp(addr, (rose_address
*)dev->dev_addr) == 0)
    goto out;
  }
+ dev = NULL;
 out:
  read_unlock(&dev_base_lock);
  return dev != NULL;
--- ./net/sched/sch_api.c.vedevbase Mon Mar 20 08:53:29 2006
+++ ./net/sched/sch_api.c Thu Jun 22 12:03:08 2006
@@ -830,12 +830,15 @@ static int tc_dump_qdisc(struct sk_buff
  struct net_device *dev;
  struct Qdisc *q;

+ idx = 0;
  s_idx = cb->args[0];
  s_q_idx = q_idx = cb->args[1];
  read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
-  if (idx < s_idx)
+ for_each_netdev(dev) {
+  if (idx < s_idx) {
+   idx++;
    continue;
+  }
   if (idx > s_idx)
    s_q_idx = 0;
   read_lock_bh(&qdisc_tree_lock);
@@ -853,6 +856,7 @@ static int tc_dump_qdisc(struct sk_buff
    q_idx++;
   }
   read_unlock_bh(&qdisc_tree_lock);
+  idx++;
  }

 done:
```

```
--- ./net/sctp/protocol.c.vedevbase Wed Jun 21 18:53:23 2006
+++ ./net/sctp/protocol.c Thu Jun 22 12:03:08 2006
@@ -177,7 +177,7 @@ static void __sctp_get_local_addr_list(v
  struct sctp_af *af;

  read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
   __list_for_each(pos, &sctp_address_families) {
    af = list_entry(pos, struct sctp_af, list);
    af->copy_addrlist(&sctp_local_addr_list, dev);
--- ./net/tipc/eth_media.c.vedevbase Wed Jun 21 18:51:09 2006
+++ ./net/tipc/eth_media.c Thu Jun 22 12:03:08 2006
@@ -118,17 +118,19 @@ static int recv_msg(struct sk_buff *buf,

 static int enable_bearer(struct tipc_bearer *tb_ptr)
 {
- struct net_device *dev = dev_base;
+ struct net_device *pdev, *dev;
  struct eth_bearer *eb_ptr = &eth_bearers[0];
  struct eth_bearer *stop = &eth_bearers[MAX_ETH_BEARERS];
  char *driver_name = strchr((const char *)tb_ptr->name, ':') + 1;

  /* Find device with specified name */

- while (dev && dev->name &&
-        (memcmp(dev->name, driver_name, strlen(dev->name)))) {
-  dev = dev->next;
- }
+ dev = NULL;
+ for_each_netdev(pdev)
+  if (!memcmp(pdev->name, driver_name, strlen(pdev->name))) {
+   dev = pdev;
+   break;
+  }
  if (!dev)
   return -ENODEV;
```