
Subject: [PATCH] Rework dev_base via list_head (v3)

Posted by [xemul](#) on Thu, 03 May 2007 15:32:54 GMT

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

Netdev tree was updated today soon after I sent a patch for it, so this version applies to it and the fs/afs/netdevices.c file is added. Hope I am not late yet again :)

Cleanup of dev_base list use, with the aim to simplify making 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: Pavel Emelianov <xemul@openvz.org>

Acked-by: Kirill Korotaev <dev@openvz.org>

Cc: Patrick McHardy <kaber@trash.net>

```
diff --git a/arch/s390/appldata/appldata_net_sum.c b/arch/s390/appldata/appldata_net_sum.c
index a43f348..2180ac1 100644
```

```
--- a/arch/s390/appldata/appldata_net_sum.c
```

```
+++ b/arch/s390/appldata/appldata_net_sum.c
```

```
@@ -107,7 +107,7 @@ 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) {
```

```
    stats = dev->get_stats(dev);
```

```
    rx_packets += stats->rx_packets;
```

```
    tx_packets += stats->tx_packets;
```

```
diff --git a/arch/sparc64/solaris/ioctl.c b/arch/sparc64/solaris/ioctl.c
```

```
index 330743c..18352a4 100644
```

```
--- a/arch/sparc64/solaris/ioctl.c
```

```
+++ b/arch/sparc64/solaris/ioctl.c
```

```
@@ -686,7 +686,8 @@ 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)))
```

```
diff --git a/drivers/block/aoe/aoecmd.c b/drivers/block/aoe/aoecmd.c
```

```

index 1a6aeac..01fbdd3 100644
--- a/drivers/block/aoe/aoecmd.c
+++ b/drivers/block/aoe/aoecmd.c
@@ -194,15 +194,15 @@ 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(ifp) {
    dev_hold(ifp);
    if (!is_aoe_netif(ifp))
- continue;
+ goto cont;

    skb = new_skb(sizeof *h + sizeof *ch);
    if (skb == NULL) {
        printk(KERN_INFO "aoe: skb alloc failure\n");
- continue;
+ goto cont;
    }
    skb_put(skb, sizeof *h + sizeof *ch);
    skb->dev = ifp;
@@ -221,6 +221,8 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigne

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

diff --git a/drivers/net/wireless/strip.c b/drivers/net/wireless/strip.c
index 2a299a0..ef32a5c 100644
--- a/drivers/net/wireless/strip.c
+++ b/drivers/net/wireless/strip.c
@@ -1971,8 +1971,7 @@ 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,
@@ -1983,7 +1982,6 @@ static struct net_device *get_strip_dev(
    read_unlock_bh(&dev_base_lock);
    return (dev);

```

```

    }
-   dev = dev->next;
    }
    read_unlock_bh(&dev_base_lock);
}
diff --git a/drivers/parisc/led.c b/drivers/parisc/led.c
index 3df82fe..98be288 100644
--- a/drivers/parisc/led.c
+++ b/drivers/parisc/led.c
@@ -365,7 +365,7 @@ 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)
diff --git a/fs/afs/netdevices.c b/fs/afs/netdevices.c
index ce08977..fc27d4b 100644
--- a/fs/afs/netdevices.c
+++ b/fs/afs/netdevices.c
@@ -47,7 +47,7 @@ int afs_get_ipv4_interfaces(struct afs_i
    ASSERT(maxbufs > 0);

    rtnl_lock();
-   for (dev = dev_base; dev; dev = dev->next) {
+   for_each_netdev(dev) {
        if (dev->type == ARPHRD_LOOPBACK && !wantloopback)
            continue;
        idev = __in_dev_get_rtnl(dev);
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 4428f1c..3044622 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -304,7 +304,7 @@ 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);
@@ -575,9 +575,31 @@ struct packet_type {
#include <linux/notifier.h>

extern struct net_device loopback_dev; /* The loopback */

```

```

-extern struct net_device *dev_base; /* All devices */
+extern struct list_head dev_base_head; /* All devices */
extern rwlock_t dev_base_lock; /* Device list lock */

+#define for_each_netdev(d) \
+ list_for_each_entry(d, &dev_base_head, dev_list)
+#define for_each_netdev_safe(d, n) \
+ list_for_each_entry_safe(d, n, &dev_base_head, dev_list)
+#define for_each_netdev_continue(d) \
+ list_for_each_entry_continue(d, &dev_base_head, dev_list)
+#define net_device_entry(lh) list_entry(lh, struct net_device, dev_list)
+
+static inline struct net_device *next_net_device(struct net_device *dev)
+{
+ struct list_head *lh;
+
+ lh = dev->dev_list.next;
+ return lh == &dev_base_head ? NULL : net_device_entry(lh);
+}
+
+static inline struct net_device *first_net_device(void)
+{
+ return list_empty(&dev_base_head) ? NULL :
+ net_device_entry(dev_base_head.next);
+}
+
extern int netdev_boot_setup_check(struct net_device *dev);
extern unsigned long netdev_boot_base(const char *prefix, int unit);
extern struct net_device *dev_getbyhwaddr(unsigned short type, char *hwaddr);
diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
index c0c7bb8..bd93c45 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
@@ -117,8 +117,7 @@ static void __exit vlan_cleanup_devices(
 struct net_device *dev, *nxt;

    rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
-  nxt = dev->next;
+ for_each_netdev_safe(dev, nxt) {
    if (dev->priv_flags & IFF_802_1Q_VLAN) {
      unregister_vlan_dev(VLAN_DEV_INFO(dev)->real_dev,
        VLAN_DEV_INFO(dev)->vlan_id);
diff --git a/net/8021q/vlanproc.c b/net/8021q/vlanproc.c
index 5e24f72..d216a64 100644
--- a/net/8021q/vlanproc.c
+++ b/net/8021q/vlanproc.c
@@ -237,13 +237,9 @@ int vlan_proc_rem_dev(struct net_device

```

```

* The following few functions build the content of /proc/net/vlan/config
*/

-/* starting at dev, find a VLAN device */
-static struct net_device *vlan_skip(struct net_device *dev)
+static inline int is_vlan_dev(struct net_device *dev)
{
- while (dev && !(dev->priv_flags & IFF_802_1Q_VLAN))
- dev = dev->next;
-
- return dev;
+ return dev->priv_flags & IFF_802_1Q_VLAN;
}

/* start read of /proc/net/vlan/config */
@@ -257,19 +253,35 @@ 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_each_netdev(dev) {
+ if (!is_vlan_dev(dev))
+ continue;
+
+ if (i++ == *pos)
+ return dev;
+ }

- return (i == *pos) ? dev : NULL;
+ return NULL;
}

static void *vlan_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
+ struct net_device *dev;
+
+ ++*pos;

- return vlan_skip((v == SEQ_START_TOKEN)
- ? dev_base
- : ((struct net_device *)v)->next);
+ dev = (struct net_device *)v;
+ if (v == SEQ_START_TOKEN)
+ dev = net_device_entry(&dev_base_head);
+
+ for_each_netdev_continue(dev) {
+ if (!is_vlan_dev(dev))

```

```

+ continue;
+
+ return dev;
+ }
+
+ return NULL;
}

static void vlan_seq_stop(struct seq_file *seq, void *v)
diff --git a/net/bridge/br_if.c b/net/bridge/br_if.c
index 690573b..849deaf 100644
--- a/net/bridge/br_if.c
+++ b/net/bridge/br_if.c
@@ -475,11 +475,9 @@ void __exit br_cleanup_bridges(void)
 struct net_device *dev, *nxt;

    rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
-     nxt = dev->next;
+ for_each_netdev_safe(dev, nxt)
+     if (dev->priv_flags & IFF_EBRIDGE)
+         del_br(dev->priv);
- }
    rtnl_unlock();

}
diff --git a/net/bridge/br_ioctl.c b/net/bridge/br_ioctl.c
index eda0fbf..bb15e9e 100644
--- a/net/bridge/br_ioctl.c
+++ b/net/bridge/br_ioctl.c
@@ -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;
+ }
diff --git a/net/bridge/br_netlink.c b/net/bridge/br_netlink.c
index 35facc0..0fcf6f0 100644
--- a/net/bridge/br_netlink.c
+++ b/net/bridge/br_netlink.c
@@ -109,7 +109,8 @@ static int br_dump_ifinfo(struct sk_buff
 struct net_device *dev;
 int idx;

```

```

- for (dev = dev_base, idx = 0; dev; dev = dev->next) {
+ idx = 0;
+ for_each_netdev(dev) {
    /* not a bridge port */
    if (dev->br_port == NULL || idx < cb->args[0])
        goto skip;
diff --git a/net/core/dev.c b/net/core/dev.c
index c305819..f27d4ab 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -156,13 +156,13 @@ static spinlock_t net_dma_event_lock;
#endif

/*
- * The @dev_base list is protected by @dev_base_lock and the rtnl
+ * The @dev_base_head list is protected by @dev_base_lock and the rtnl
 * semaphore.
 *
 * Pure readers hold dev_base_lock for reading.
 *
 * Writers must hold the rtnl semaphore while they loop through the
- * dev_base list, and hold dev_base_lock for writing when they do the
+ * dev_base_head list, and hold dev_base_lock for writing when they do the
 * actual updates. This allows pure readers to access the list even
 * while a writer is preparing to update it.
 */
@@ -174,11 +174,11 @@ 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;
+LIST_HEAD(dev_base_head);
+DEFINE_RWLOCK(dev_base_lock);

-EXPORT_SYMBOL(dev_base);
+EXPORT_SYMBOL(dev_base_head);
EXPORT_SYMBOL(dev_base_lock);

#define NETDEV_HASHBITS 8
@@ -567,11 +566,12 @@ struct net_device *dev_getbyhwaddr(unsigned

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;
- return dev;
+ return dev;
+
+ return NULL;
}

EXPORT_SYMBOL(dev_getbyhwaddr);
@@ -581,11 +581,11 @@ struct net_device *__dev_getfirstbyhwtyp
    struct net_device *dev;

    ASSERT_RTNL();
- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev)
    if (dev->type == type)
- break;
- }
- return dev;
+ return dev;
+
+ return NULL;
}

EXPORT_SYMBOL(__dev_getfirstbyhwtype);
@@ -617,17 +617,19 @@ EXPORT_SYMBOL(dev_getfirstbyhwtype);

struct net_device *dev_get_by_flags(unsigned short if_flags, unsigned short mask)
{
- struct net_device *dev;
+ struct net_device *dev, *ret;

+ ret = NULL;
    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);
+ ret = dev;
        break;
    }
}
    read_unlock(&dev_base_lock);
- return dev;
+ return ret;
}

/**

```



```

@@ -693,7 +695,7 @@ 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)
@@ -975,7 +977,7 @@ 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)
@@ -2049,7 +2051,7 @@ 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;
@@ -2081,26 +2083,28 @@ static int dev_ifconf(char __user *arg)
    * This is invoked by the /proc filesystem handler to display a device
    * in detail.
    */
- static struct net_device *dev_get_idx(loff_t pos)
+ void *dev_seq_start(struct seq_file *seq, loff_t *pos)
    {
+ loff_t off;
        struct net_device *dev;
- loff_t i;

- for (i = 0, dev = dev_base; dev && i < pos; ++i, dev = dev->next);
+ read_lock(&dev_base_lock);
+ if (!*pos)
+ return SEQ_START_TOKEN;

- return i == pos ? dev : NULL;
- }
+ off = 1;
+ for_each_netdev(dev)
+ if (off++ == *pos)

```

```

+ return dev;

-void *dev_seq_start(struct seq_file *seq, loff_t *pos)
-{
- read_lock(&dev_base_lock);
- return *pos ? dev_get_idx(*pos - 1) : SEQ_START_TOKEN;
+ return NULL;
}

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

void dev_seq_stop(struct seq_file *seq, void *v)
@@ -3082,11 +3086,9 @@ 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);
@@ -3360,8 +3362,6 @@ void synchronize_net(void)

void unregister_netdevice(struct net_device *dev)
{
- struct net_device *d, **dp;
-
  BUG_ON(dev_boot_phase);
  ASSERT_RTNL();

@@ -3381,19 +3381,11 @@ void unregister_netdevice(struct net_dev
    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;
- }
- }
- BUG_ON(!d);
+ 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;
```

```

diff --git a/net/core/dev_mcast.c b/net/core/dev_mcast.c
index 7d57bf7..5a54053 100644
--- a/net/core/dev_mcast.c
+++ b/net/core/dev_mcast.c
@@ -223,7 +223,7 @@ 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;
    }
@@ -232,9 +232,8 @@ static void *dev_mc_seq_start(struct seq

static void *dev_mc_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
- struct net_device *dev = v;
- ++*pos;
- return dev->next;
+ return next_net_device((struct net_device *)v);
}

static void dev_mc_seq_stop(struct seq_file *seq, void *v)
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index cec1111..8c971a2 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -539,13 +539,16 @@ static int rtnl_dump_ifinfo(struct sk_buff
    int s_idx = cb->args[0];
    struct net_device *dev;

```

```

- 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 (rtnl_fill_ifinfo(skb, dev, NULL, 0, RTM_NEWLINK,
        NETLINK_CB(cb->skb).pid,
        cb->nlh->nmsg_seq, 0, NLM_F_MULTI) <= 0)
        break;
+cont:
+   idx++;
    }
    cb->args[0] = idx;

```

```

diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index a205eaa..9fbe87c 100644

```

```

--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -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 *dev, *ldev;
    int rv;

    if (addr_len != sizeof(struct sockaddr_dn))
@@ -746,14 +746,17 @@ static int dn_bind(struct socket *sock,
    if (!(saddr->sdn_flags & SDF_WILD)) {
        if (dn_ntohs(saddr->sdn_nodeaddr)) {
            read_lock(&dev_base_lock);
-       for (dev = dev_base; dev; dev = dev->next) {
+       ldev = NULL;
+       for_each_netdev(dev) {
            if (!dev->dn_ptr)
                continue;
-           if (dn_dev_islocal(dev, dn_saddr2dn(saddr)))
+           if (dn_dev_islocal(dev, dn_saddr2dn(saddr))) {
+               ldev = dev;
                break;
            }
        }
        read_unlock(&dev_base_lock);
-       if (dev == NULL)
+       if (ldev == NULL)
            return -EADDRNOTAVAIL;

```

```

    }
}
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index 5c2a995..764a56a 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -799,9 +799,10 @@ static int dn_nl_dump_ifaddr(struct sk_b
    skip_ndevs = cb->args[0];
    skip_naddr = cb->args[1];

- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ idx = 0;
+ for_each_netdev(dev) {
    if (idx < skip_ndevs)
- continue;
+ goto cont;
    else if (idx > skip_ndevs) {
        /* Only skip over addresses for first dev dumped
         * in this iteration (idx == skip_ndevs) */
@@ -809,18 +810,20 @@ static int dn_nl_dump_ifaddr(struct sk_b
    }

    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 < skip_naddr)
- continue;
+ goto cont;

        if (dn_nl_fill_ifaddr(skb, ifa, NETLINK_CB(cb->skb).pid,
                             cb->nlnh->nlnmsg_seq, RTM_NEWADDR,
                             NLM_F_MULTI) < 0)
            goto done;
    }
+cont:
+ idx++;
    }
done:
    cb->args[0] = idx;
@@ -1296,7 +1299,7 @@ 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();

@@ -1307,7 +1310,7 @@ 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);
    }
@@ -1325,62 +1328,56 @@ int unregister_dnaddr_notifier(struct no
}

#ifdef CONFIG_PROC_FS
-static inline struct net_device *dn_dev_get_next(struct seq_file *seq, struct net_device *dev)
+static inline int is_dn_dev(struct net_device *dev)
{
- do {
-     dev = dev->next;
- } while(dev && !dev->dn_ptr);
-
- return dev;
+ return dev->dn_ptr != NULL;
}

-static struct net_device *dn_dev_get_idx(struct seq_file *seq, loff_t pos)
+static void *dn_dev_seq_start(struct seq_file *seq, loff_t *pos)
{
+ int i;
    struct net_device *dev;

- dev = dev_base;
- if (dev && !dev->dn_ptr)
-     dev = dn_dev_get_next(seq, dev);
- if (pos) {
-     while(dev && (dev = dn_dev_get_next(seq, dev)))
-         --pos;
- }
- return dev;
-}
+ read_lock(&dev_base_lock);

-static void *dn_dev_seq_start(struct seq_file *seq, loff_t *pos)
-{
- if (*pos) {
-     struct net_device *dev;

```

```

- read_lock(&dev_base_lock);
- dev = dn_dev_get_idx(seq, *pos - 1);
- if (dev == NULL)
- read_unlock(&dev_base_lock);
- return dev;
+ if (*pos == 0)
+ return SEQ_START_TOKEN;
+
+ i = 1;
+ for_each_netdev(dev) {
+ if (!is_dn_dev(dev))
+ continue;
+
+ if (i++ == *pos)
+ return dev;
+ }
- return SEQ_START_TOKEN;
+
+ return NULL;
}

static void *dn_dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
- struct net_device *dev = v;
- loff_t one = 1;
+ struct net_device *dev;

- if (v == SEQ_START_TOKEN) {
- dev = dn_dev_seq_start(seq, &one);
- } else {
- dev = dn_dev_get_next(seq, dev);
- if (dev == NULL)
- read_unlock(&dev_base_lock);
- }
++*pos;
- return dev;
+
+ dev = (struct net_device *)v;
+ if (v == SEQ_START_TOKEN)
+ dev = net_device_entry(&dev_base_head);
+
+ for_each_netdev_continue(dev) {
+ if (!is_dn_dev(dev))
+ continue;
+
+ return dev;
+ }
+

```

```

+ return NULL;
}

static void dn_dev_seq_stop(struct seq_file *seq, void *v)
{
- if (v && v != SEQ_START_TOKEN)
- read_unlock(&dev_base_lock);
+ read_unlock(&dev_base_lock);
}

static char *dn_type2asc(char type)
diff --git a/net/decnet/dn_fib.c b/net/decnet/dn_fib.c
index 310a862..d2bc19d 100644
--- a/net/decnet/dn_fib.c
+++ b/net/decnet/dn_fib.c
@@ -602,7 +602,7 @@ 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;
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 5d7337b..a8bf106 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -886,7 +886,7 @@ static int dn_route_output_slow(struct d
    .iif = loopback_dev.ifindex,
    .oif = oldflp->oif };
    struct dn_route *rt = NULL;
- struct net_device *dev_out = NULL;
+ struct net_device *dev_out = NULL, *dev;
    struct neighbour *neigh = NULL;
    unsigned hash;
    unsigned flags = 0;
@@ -925,15 +925,17 @@ 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) {
- if (!dev_out->dn_ptr)
+ for_each_netdev(dev) {
+ if (!dev->dn_ptr)
    continue;
- if (!dn_dev_islocal(dev_out, oldflp->fld_src))
+ if (!dn_dev_islocal(dev, oldflp->fld_src))

```



```

        continue;
-   if ((dev_out->flags & IFF_LOOPBACK) &&
+   if ((dev->flags & IFF_LOOPBACK) &&
        oldflp->fld_dst &&
-       !dn_dev_islocal(dev_out, oldflp->fld_dst))
+       !dn_dev_islocal(dev, oldflp->fld_dst))
        continue;
+
+   dev_out = dev;
    break;
}
    read_unlock(&dev_base_lock);
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 088888d..7f95e6e 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -910,7 +910,7 @@ 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;

@@ -989,7 +989,7 @@ __be32 inet_confirm_addr(const struct ne

    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)
@@ -1182,23 +1182,26 @@ static int inet_dump_ifaddr(struct sk_bu
    int s_ip_idx, s_idx = cb->args[0];

    s_ip_idx = ip_idx = cb->args[1];
-   for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+   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_ip_idx = 0;
        if ((in_dev = __in_dev_get_rtnl(dev)) == NULL)
-       continue;

```

```

+ goto cont;

    for (ifa = in_dev->ifa_list, ip_idx = 0; ifa;
        ifa = ifa->ifa_next, ip_idx++) {
        if (ip_idx < s_ip_idx)
- continue;
+ goto cont;
        if (inet_fill_ifaddr(skb, ifa, NETLINK_CB(cb->skb).pid,
            cb->nlh->nlmsg_seq,
            RTM_NEWADDR, NLM_F_MULTI) <= 0)
            goto done;
    }
+cont:
+ idx++;
}

done:
@@ -1243,7 +1246,7 @@ 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);
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 2506021..f4dd474 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -2288,9 +2288,8 @@ static inline struct ip_mc_list *igmp_mc
    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->in_dev = NULL;
+ for_each_netdev(state->dev) {
    struct in_device *in_dev;
    in_dev = in_dev_get(state->dev);
    if (!in_dev)
@@ -2316,7 +2315,7 @@ 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_net_device(state->dev);

```

```

    if (!state->dev) {
        state->in_dev = NULL;
        break;
@@ -2450,9 +2449,9 @@ static inline struct ip_sf_list *igmp_mc
    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->idev = NULL;
+ state->im = NULL;
+ for_each_netdev(state->dev) {
    struct in_device *idev;
    idev = in_dev_get(state->dev);
    if (unlikely(idev == NULL))
@@ -2488,7 +2487,7 @@ 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_net_device(state->dev);
    if (!state->dev) {
        state->idev = NULL;
        goto out;
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index 597c800..342ca8d 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -192,7 +192,7 @@ static int __init ic_open_devs(void)
    if (dev_change_flags(&loopback_dev, loopback_dev.flags | IFF_UP) < 0)
        printk(KERN_ERR "IP-Config: Failed to open %s\n", loopback_dev.name);

- for (dev = dev_base; dev; dev = dev->next) {
+ for_each_netdev(dev) {
    if (dev == &loopback_dev)
        continue;
    if (user_dev_name[0] ? !strcmp(dev->name, user_dev_name) :
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 3452433..d02685c 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -449,7 +449,7 @@ 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) {

```

```

    rcu_read_lock();
    idev = __in6_dev_get(dev);
    if (idev) {
@@ -911,7 +911,7 @@ int ipv6_dev_get_saddr(struct net_device
    read_lock(&dev_base_lock);
    rcu_read_lock();

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

@@ -2064,7 +2064,7 @@ 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;
@@ -2225,7 +2225,7 @@ 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;
}

@@ -3257,14 +3257,15 @@ static int inet6_dump_addr(struct sk_buf
    s_idx = cb->args[0];
    s_ip_idx = ip_idx = cb->args[1];

- 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_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:
@@ -3311,6 +3312,8 @@ static int inet6_dump_addr(struct sk_buf
    }
    read_unlock_bh(&idev->lock);
    in6_dev_put(idev);
+cont:
+ idx++;
    }
done:
    if (err <= 0) {
@@ -3575,16 +3578,19 @@ static int inet6_dump_ifinfo(struct sk_b
    struct inet6_dev *idev;

    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 ((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;
@@ -4247,7 +4253,7 @@ 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);
diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index 09117d6..9b81264 100644
--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -423,14 +423,18 @@ static int ipv6_chk_acast_dev(struct net
    */

```

```

int ipv6_chk_acast_addr(struct net_device *dev, struct in6_addr *addr)
{
+ int found = 0;
+
  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))
+ for_each_netdev(dev)
+ if (ipv6_chk_acast_dev(dev, addr)) {
+   found = 1;
+   break;
+ }
  read_unlock(&dev_base_lock);
- return dev != 0;
+ return found;
}

```

```

@@ -447,9 +451,8 @@ static inline struct ifacaddr6 *ac6_get_
  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->idev = NULL;
+ for_each_netdev(state->dev) {
  struct inet6_dev *idev;
  idev = in6_dev_get(state->dev);
  if (!idev)

```

```

@@ -476,7 +479,7 @@ 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_net_device(state->dev);
  if (!state->dev) {
    state->idev = NULL;
    break;

```

diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c

index 6c27589..3e308fb 100644

--- a/net/ipv6/mcast.c

+++ b/net/ipv6/mcast.c

```

@@ -2331,9 +2331,8 @@ 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)
@@ -2360,7 +2359,7 @@ 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_net_device(state->dev);
    if (!state->dev) {
        state->idev = NULL;
        break;
@@ -2475,9 +2474,9 @@ static inline struct ip6_sf_list *igmp6_
    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->idev = NULL;
+ state->im = NULL;
+ for_each_netdev(state->dev) {
    struct inet6_dev *idev;
    idev = in6_dev_get(state->dev);
    if (unlikely(idev == NULL))
@@ -2513,7 +2512,7 @@ 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_net_device(state->dev);
    if (!state->dev) {
        state->idev = NULL;
        goto out;
diff --git a/net/llc/llc_core.c b/net/llc/llc_core.c
index d12413c..d4b13a0 100644
--- a/net/llc/llc_core.c
+++ b/net/llc/llc_core.c
@@ -160,8 +160,14 @@ 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 = first_net_device();
+ if (dev != NULL)
+ dev = next_net_device(dev);
+
+ if (dev != NULL)
+ 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);
diff --git a/net/netrom/nr_route.c b/net/netrom/nr_route.c
index 8e6bd4e..2f76e06 100644
--- a/net/netrom/nr_route.c
+++ b/net/netrom/nr_route.c
@@ -598,7 +598,7 @@ 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;
@@ -618,12 +618,13 @@ struct net_device *nr_dev_get(ax25_addr
  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;
  }
}
+ dev = NULL;
out:
  read_unlock(&dev_base_lock);
  return dev;
diff --git a/net/rose/rose_route.c b/net/rose/rose_route.c
index 1f9aefd..929a784 100644
--- a/net/rose/rose_route.c
+++ b/net/rose/rose_route.c
@@ -596,7 +596,7 @@ 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;
@@ -614,12 +614,13 @@ 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;
@@ -630,10 +631,11 @@ 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;
diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index 8699e70..bec600a 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -894,9 +894,10 @@ static int tc_dump_qdisc(struct sk_buff
    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++) {
+ idx = 0;
+ for_each_netdev(dev) {

```

```

    if (idx < s_idx)
-   continue;
+   goto cont;
    if (idx > s_idx)
        s_q_idx = 0;
        q_idx = 0;
@@ -910,6 +911,8 @@ static int tc_dump_qdisc(struct sk_buff
    goto done;
    q_idx++;
}
+cont:
+   idx++;
}

done:
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index c361deb..d4afafc 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -170,7 +170,7 @@ static void sctp_get_local_addr_list(voi
    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);
diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index 67bb29b..0ee6ded 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -120,16 +120,18 @@ 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 *dev, *pdev;
    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 && strncmp(dev->name, driver_name, IFNAMSIZ)) {
-       dev = dev->next;
-   }
+   dev = NULL;

```

```
+ for_each_netdev(pdev)
+ if (!strcmp(dev->name, driver_name, IFNAMSIZ)) {
+   dev = pdev;
+   break;
+ }
+ if (!dev)
+   return -ENODEV;
```

Subject: Re: [PATCH] Rework dev_base via list_head (v3)

Posted by [davem](#) on Fri, 04 May 2007 00:17:35 GMT

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

From: Pavel Emelianov <xemul@sw.ru>

Date: Thu, 03 May 2007 19:32:54 +0400

> Netdev tree was updated today soon after I sent a patch for it,
> so this version applies to it and the fs/afs/netdevices.c file
> is added. Hope I am not late yet again :)
>
> Cleanup of dev_base list use, with the aim to simplify making
> 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: Pavel Emelianov <xemul@openvz.org>
> Aacked-by: Kirill Korotaev <dev@openvz.org>
> Cc: Patrick McHardy <kaber@trash.net>

Looks good, I've applied this, thanks!
