
Subject: [PATCH RFC 16/31] net: Make the device list and device lookups per namespace.

Posted by [ebiederm](#) on Thu, 25 Jan 2007 19:00:18 GMT

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

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch makes most of the generic device layer network namespace safe. This patch makes `dev_base`, `dev_base_lock` per network namespace variables, and then it picks up a few associated variables. The funnctions:

`dev_getbyhwaddr`
`dev_getfirsthwbytype`
`dev_get_by_flags`
`dev_get_by_name`
`__dev_get_by_name`
`dev_get_by_index`
`__dev_get_by_index`
`dev_ioctl`
`dev_ethtool`
`dev_load`
`wireless_process_ioctl`

were modified to take a network namespace argument, and deal with it.

`vlan_ioctl_set` and `briocli_set` were modified so their hooks will receive a network namespace argument.

So basically anything in the core of the network stack that was affected to by the change of `dev_base` and `dev_base_lock` was modified to handle multiple network namespaces. The rest of the network stack was simply modified to explicitly use `init_net()` the initial network namespace. This can be fixed when those components of the network stack are modified to handle multiple network namespaces.

For now the ifindex generator is left global.

Fundametally ifindex numbers are per namespace, or else we will have corner case problems with migration when we get that far.

At the same time there are assumptions in the network stack that the ifindex of a network device won't change. Making the ifindex number global seems a good compromise until the network stack can cope with ifindex changes when you change namespaces, and the like.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
arch/s390/appldata/appldata_net_sum.c | 6 +-
arch/sparc64/solaris/ioctl.c           | 6 +-
drivers/atm/idt77252.c                 | 2 +-
drivers/block/aoe/aoecmd.c             | 7 +-
drivers/net/bonding/bond_main.c         | 6 +-
drivers/net/bonding/bond_sysfs.c        | 3 +-
drivers/net/eql.c                      | 9 +-
drivers/net/pppoe.c                    | 2 +-
drivers/net/shaper.c                   | 3 +-
drivers/net/tun.c                      | 3 +-
drivers/net/wan/dlci.c                 | 4 +-
drivers/net/wan/sbni.c                 | 2 +-
drivers/net/wireless/strip.c           | 8 +-
drivers/parisc/led.c                   | 6 +-
include/linux/if_bridge.h              | 2 +-
include/linux/if_vlan.h                | 2 +-
include/linux/netdevice.h              | 24 +++-
include/net/iw_handler.h               | 2 +-
net/802/tr.c                           | 2 +-
net/8021q/vlan.c                       | 10 +-
net/8021q/vlan_dev.c                   | 12 +-
net/8021q/vlanproc.c                   | 8 +-
net/appletalk/ddp.c                    | 6 +-
net/atm/mpc.c                          | 2 +-
net/ax25/af_ax25.c                     | 2 +-
net/bridge/br_if.c                     | 6 +-
net/bridge/br_ioctl.c                  | 7 +-
net/bridge/br_netlink.c                 | 9 +-
net/bridge/br_private.h                 | 2 +-
net/core/dev.c                         | 282 ++++++-----
net/core/dev_mcast.c                   | 46 +++++-
net/core/ethtool.c                     | 4 +-
net/core/fib_rules.c                   | 4 +-
net/core/link_watch.c                  | 5 +-
net/core/neighbour.c                   | 6 +-
net/core/net-sysfs.c                   | 27 +++-
net/core/netpoll.c                     | 2 +-
net/core/pktgen.c                       | 2 +-
net/core/rtnetlink.c                   | 24 +++-
net/core/sock.c                         | 3 +-
net/core/wireless.c                    | 43 +++++-
net/decnet/af_decnet.c                  | 6 +-
net/decnet/dn_dev.c                    | 32 +++-
net/decnet/dn_fib.c                     | 12 +-
net/decnet/dn_route.c                  | 14 +-

```

```

net/dechnet/sysctl_net_dechnet.c      | 4 +-
net/econet/af_econet.c                 | 2 +-
net/ipv4/arp.c                         | 4 +-
net/ipv4/devinet.c                     | 36 +++-
net/ipv4/fib_frontend.c                | 2 +-
net/ipv4/fib_semantics.c               | 4 +-
net/ipv4/igmp.c                        | 12 +-
net/ipv4/ip_fragment.c                 | 2 +-
net/ipv4/ip_gre.c                      | 4 +-
net/ipv4/ip_sockglue.c                 | 2 +-
net/ipv4/ipconfig.c                    | 2 +-
net/ipv4/ipip.c                        | 4 +-
net/ipv4/ipmr.c                        | 4 +-
net/ipv4/ipvs/ip_vs_sync.c             | 10 +-
net/ipv4/netfilter/ipt_CLUSTERIP.c     | 2 +-
net/ipv4/route.c                       | 4 +-
net/ipv6/addrconf.c                    | 44 +++--
net/ipv6/af_inet6.c                   | 2 +-
net/ipv6/anycast.c                     | 20 +++-
net/ipv6/datagram.c                    | 2 +-
net/ipv6/ip6_tunnel.c                  | 6 +-
net/ipv6/ipv6_sockglue.c               | 2 +-
net/ipv6/mcast.c                       | 20 +++-
net/ipv6/raw.c                         | 2 +-
net/ipv6/reassembly.c                  | 2 +-
net/ipv6/route.c                       | 4 +-
net/ipv6/sit.c                         | 4 +-
net/ipv6/af_ipx.c                      | 6 +-
net/llc/af_llc.c                       | 4 +-
net/llc/llc_core.c                     | 5 +-
net/netrom/nr_route.c                  | 14 +-
net/packet/af_packet.c                  | 18 +-
net/rose/rose_route.c                  | 20 +++-
net/sched/act_mirred.c                  | 2 +-
net/sched/cls_api.c                     | 4 +-
net/sched/em_meta.c                     | 2 +-
net/sched/sch_api.c                     | 14 +-
net/sctp/ipv6.c                         | 4 +-
net/sctp/protocol.c                     | 6 +-
net/socket.c                           | 22 +-
net/tipc/eth_media.c                   | 2 +-
net/wanrouter/af_wanpipe.c              | 24 +++-
net/x25/x25_route.c                    | 2 +-
88 files changed, 597 insertions(+), 433 deletions(-)

```

```

diff --git a/arch/s390/appldata/appldata_net_sum.c b/arch/s390/appldata/appldata_net_sum.c
index 075e619..4a32370 100644
--- a/arch/s390/appldata/appldata_net_sum.c

```

```

+++ b/arch/s390/appldata/appldata_net_sum.c
@@ -106,8 +106,8 @@ static void appldata_get_net_sum_data(void *data)
    rx_dropped = 0;
    tx_dropped = 0;
    collisions = 0;
- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    if (dev->get_stats == NULL) {
        continue;
    }
@@ -123,7 +123,7 @@ static void appldata_get_net_sum_data(void *data)
    collisions += stats->collisions;
    i++;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
    net_data->nr_interfaces = i;
    net_data->rx_packets = rx_packets;
    net_data->tx_packets = tx_packets;
diff --git a/arch/sparc64/solaris/ioctl.c b/arch/sparc64/solaris/ioctl.c
index 330743c..1ecf4ab 100644
--- a/arch/sparc64/solaris/ioctl.c
+++ b/arch/sparc64/solaris/ioctl.c
@@ -685,9 +685,9 @@ static inline int solaris_i(unsigned int fd, unsigned int cmd, u32 arg)
    struct net_device *d;
    int i = 0;

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

    if (put_user(i, (int __user *)A(arg)))
        return -EFAULT;
diff --git a/drivers/atm/idt77252.c b/drivers/atm/idt77252.c
index f407861..3e75e0e 100644
--- a/drivers/atm/idt77252.c
+++ b/drivers/atm/idt77252.c
@@ -3569,7 +3569,7 @@ init_card(struct atm_dev *dev)
    * XXX: <hack>
    */
    sprintf(tname, "eth%d", card->index);
- tmp = dev_get_by_name(tname); /* jhs: was "tmp = dev_get(tname);" */
+ tmp = dev_get_by_name(init_net(), tname); /* jhs: was "tmp = dev_get(tname);" */

```

```

if (tmp) {
    memcpy(card->atmdev->esi, tmp->dev_addr, 6);

diff --git a/drivers/block/aoe/aoecmd.c b/drivers/block/aoe/aoecmd.c
index bb022ed..9678169 100644
--- a/drivers/block/aoe/aoecmd.c
+++ b/drivers/block/aoe/aoecmd.c
@@ -9,6 +9,7 @@
#include <linux/skbuff.h>
#include <linux/netdevice.h>
#include <linux/genhd.h>
+#include <net/net_namespace.h>
#include <asm/unaligned.h>
#include "aoe.h"

@@ -192,8 +193,8 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigned char aoeminor, struct
sk_buff **tail)

    sl = sl_tail = NULL;

- read_lock(&dev_base_lock);
- for (ifp = dev_base; ifp; dev_put(ifp), ifp = ifp->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (ifp = per_net(dev_base, init_net()); ifp; dev_put(ifp), ifp = ifp->next) {
    dev_hold(ifp);
    if (!is_aoe_netif(ifp))
        continue;
@@ -221,7 +222,7 @@ aoecmd_cfg_pkts(ushort aoemajor, unsigned char aoeminor, struct
sk_buff **tail)
    skb->next = sl;
    sl = skb;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

if (tail != NULL)
    *tail = sl_tail;
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 3e04f58..2963004 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -2932,7 +2932,7 @@ static void *bond_info_seq_start(struct seq_file *seq, loff_t *pos)
int i;

/* make sure the bond won't be taken away */
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    read_lock_bh(&bond->lock);

```

```

if (*pos == 0) {
@@ -2968,7 +2968,7 @@ static void bond_info_seq_stop(struct seq_file *seq, void *v)
    struct bonding *bond = seq->private;

    read_unlock_bh(&bond->lock);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static void bond_info_show_master(struct seq_file *seq)
@@ -3742,7 +3742,7 @@ static int bond_do_ioctl(struct net_device *bond_dev, struct ifreq *ifr,
int cmd
}

    down_write(&(bonding_rwsem));
- slave_dev = dev_get_by_name(ifr->ifr_slave);
+ slave_dev = dev_get_by_name(init_net(), ifr->ifr_slave);

    dprintk("slave_dev=%p: \n", slave_dev);

diff --git a/drivers/net/bonding/bond_sysfs.c b/drivers/net/bonding/bond_sysfs.c
index ced9ed8..561707c 100644
--- a/drivers/net/bonding/bond_sysfs.c
+++ b/drivers/net/bonding/bond_sysfs.c
@@ -36,6 +36,7 @@
#include <linux/ctype.h>
#include <linux/inet.h>
#include <linux/rtnetlink.h>
+#include <net/net_namespace.h>

/* #define BONDING_DEBUG 1 */
#include "bonding.h"
@@ -298,7 +299,7 @@ static ssize_t bonding_store_slaves(struct class_device *cd, const char
*buffer,
    read_unlock_bh(&bond->lock);
    printk(KERN_INFO DRV_NAME ": %s: Adding slave %s.\n",
        bond->dev->name, ifname);
- dev = dev_get_by_name(ifname);
+ dev = dev_get_by_name(init_net(), ifname);
    if (!dev) {
        printk(KERN_INFO DRV_NAME
            ": %s: Interface %s does not exist!\n",
diff --git a/drivers/net/eql.c b/drivers/net/eql.c
index a93700e..ceae8a0 100644
--- a/drivers/net/eql.c
+++ b/drivers/net/eql.c
@@ -116,6 +116,7 @@

```

```

#include <linux/init.h>
#include <linux/timer.h>
#include <linux/netdevice.h>
+#include <net/net_namespace.h>

#include <linux/if.h>
#include <linux/if_arp.h>
@@ -412,7 +413,7 @@ static int eql_enslave(struct net_device *master_dev, slaving_request_t
__user *
    if (copy_from_user(&srq, srqp, sizeof (slaving_request_t)))
        return -EFAULT;

- slave_dev = dev_get_by_name(srq.slave_name);
+ slave_dev = dev_get_by_name(init_net(), srq.slave_name);
    if (slave_dev) {
        if ((master_dev->flags & IFF_UP) == IFF_UP) {
            /* slave is not a master & not already a slave: */
@@ -460,7 +461,7 @@ static int eql_emancipate(struct net_device *master_dev,
slaving_request_t __use
    if (copy_from_user(&srq, srqp, sizeof (slaving_request_t)))
        return -EFAULT;

- slave_dev = dev_get_by_name(srq.slave_name);
+ slave_dev = dev_get_by_name(init_net(), srq.slave_name);
    ret = -EINVAL;
    if (slave_dev) {
        spin_lock_bh(&eql->queue.lock);
@@ -493,7 +494,7 @@ static int eql_g_slave_cfg(struct net_device *dev, slave_config_t __user
*scp)
    if (copy_from_user(&sc, scp, sizeof (slave_config_t)))
        return -EFAULT;

- slave_dev = dev_get_by_name(sc.slave_name);
+ slave_dev = dev_get_by_name(init_net(), sc.slave_name);
    if (!slave_dev)
        return -ENODEV;

@@ -528,7 +529,7 @@ static int eql_s_slave_cfg(struct net_device *dev, slave_config_t __user
*scp)
    if (copy_from_user(&sc, scp, sizeof (slave_config_t)))
        return -EFAULT;

- slave_dev = dev_get_by_name(sc.slave_name);
+ slave_dev = dev_get_by_name(init_net(), sc.slave_name);
    if (!slave_dev)
        return -ENODEV;

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c

```

index 3618862..3c8b0a7 100644

--- a/drivers/net/pppoe.c

+++ b/drivers/net/pppoe.c

@@ -586,7 +586,7 @@ static int pppoe_connect(struct socket *sock, struct sockaddr *uservaddr,

```
/* Don't re-bind if sid==0 */
if (sp->sa_addr.pppoe.sid != 0) {
- dev = dev_get_by_name(sp->sa_addr.pppoe.dev);
+ dev = dev_get_by_name(init_net(), sp->sa_addr.pppoe.dev);
```

```
error = -ENODEV;
```

```
if (!dev)
```

diff --git a/drivers/net/shaper.c b/drivers/net/shaper.c

index e886e8d..b852055 100644

--- a/drivers/net/shaper.c

+++ b/drivers/net/shaper.c

@@ -86,6 +86,7 @@

```
#include <net/dst.h>
```

```
#include <net/arp.h>
```

```
+#include <net/net_namespace.h>
```

```
struct shaper_cb {
```

```
unsigned long shapeclock; /* Time it should go out */
```

@@ -488,7 +489,7 @@ static int shaper_ioctl(struct net_device *dev, struct ifreq *ifr, int cmd)

```
{
case SHAPER_SET_DEV:
{
- struct net_device *them=__dev_get_by_name(ss->ss_name);
+ struct net_device *them=__dev_get_by_name(init_net(), ss->ss_name);
if(them==NULL)
return -ENODEV;
if(sh->dev)
```

diff --git a/drivers/net/tun.c b/drivers/net/tun.c

index 151a2e1..efa1db8 100644

--- a/drivers/net/tun.c

+++ b/drivers/net/tun.c

@@ -58,6 +58,7 @@

```
#include <linux/if_ether.h>
```

```
#include <linux/if_tun.h>
```

```
#include <linux/crc32.h>
```

```
+#include <net/net_namespace.h>
```

```
#include <asm/system.h>
```

```
#include <asm/uaccess.h>
```

```
@@ -464,7 +465,7 @@ static int tun_set_iff(struct file *file, struct ifreq *ifr)
current->euid != tun->owner && !capable(CAP_NET_ADMIN))
return -EPERM;
```



```

}
- else if (__dev_get_by_name(ifr->ifr_name))
+ else if (__dev_get_by_name(init_net(), ifr->ifr_name))
    return -EINVAL;
    else {
        char *name;
diff --git a/drivers/net/wan/dlci.c b/drivers/net/wan/dlci.c
index f826494..1e3b73b 100644
--- a/drivers/net/wan/dlci.c
+++ b/drivers/net/wan/dlci.c
@@ -361,7 +361,7 @@ static int dlci_add(struct dlci_add *dlci)

    /* validate slave device */
- slave = dev_get_by_name(dlci->devname);
+ slave = dev_get_by_name(init_net(), dlci->devname);
    if (!slave)
        return -ENODEV;

@@ -427,7 +427,7 @@ static int dlci_del(struct dlci_add *dlci)
    int err;

    /* validate slave device */
- master = __dev_get_by_name(dlci->devname);
+ master = __dev_get_by_name(init_net(), dlci->devname);
    if (!master)
        return(-ENODEV);

diff --git a/drivers/net/wan/sbni.c b/drivers/net/wan/sbni.c
index fc5c0c6..ae8f941 100644
--- a/drivers/net/wan/sbni.c
+++ b/drivers/net/wan/sbni.c
@@ -1367,7 +1367,7 @@ sbni_ioctl( struct net_device *dev, struct ifreq *ifr, int cmd )

    if (copy_from_user( slave_name, ifr->ifr_data, sizeof slave_name ))
        return -EFAULT;
- slave_dev = dev_get_by_name( slave_name );
+ slave_dev = dev_get_by_name(init_net(), slave_name );
    if ( !slave_dev || !(slave_dev->flags & IFF_UP) ) {
        printk( KERN_ERR "%s: trying to enslave non-active "
            "device %s\n", dev->name, slave_name );
diff --git a/drivers/net/wireless/strip.c b/drivers/net/wireless/strip.c
index 6c27ff2..6c2ada3 100644
--- a/drivers/net/wireless/strip.c
+++ b/drivers/net/wireless/strip.c
@@ -1971,8 +1971,8 @@ static struct net_device *get_strip_dev(struct strip *strip_info)
    && memcmp(&strip_info->>true_dev_addr, zero_address.c,
        sizeof(zero_address))) {

```

```

    struct net_device *dev;
-   read_lock_bh(&dev_base_lock);
-   dev = dev_base;
+   read_lock_bh(&per_net(dev_base_lock, init_net()));
+   dev = per_net(dev_base, init_net());
    while (dev) {
        if (dev->type == strip_info->dev->type &&
            !memcmp(dev->dev_addr,
@@ -1981,12 +1981,12 @@ static struct net_device *get_strip_dev(struct strip *strip_info)
            printk(KERN_INFO
                "%s: Transferred packet ownership to %s.\n",
                strip_info->dev->name, dev->name);
-   read_unlock_bh(&dev_base_lock);
+   read_unlock_bh(&per_net(dev_base_lock, init_net()));
    return (dev);
    }
    dev = dev->next;
    }
-   read_unlock_bh(&dev_base_lock);
+   read_unlock_bh(&per_net(dev_base_lock, init_net()));
    }
    return (strip_info->dev);
    }
diff --git a/drivers/parisc/led.c b/drivers/parisc/led.c
index 8dac2ba..62662f3 100644
--- a/drivers/parisc/led.c
+++ b/drivers/parisc/led.c
@@ -365,9 +365,9 @@ static __inline__ int led_get_net_activity(void)

    /* we are running as a workqueue task, so locking dev_base
     * for reading should be OK */
-   read_lock(&dev_base_lock);
+   read_lock(&per_net(dev_base_lock, init_net()));
    rcu_read_lock();
-   for (dev = dev_base; dev; dev = dev->next) {
+   for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
        struct net_device_stats *stats;
        struct in_device *in_dev = __in_dev_get_rcu(dev);
        if (!in_dev || !in_dev->ifa_list)
@@ -381,7 +381,7 @@ static __inline__ int led_get_net_activity(void)
        tx_total += stats->tx_packets;
    }
    rcu_read_unlock();
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));

    retval = 0;

```

```
diff --git a/include/linux/if_bridge.h b/include/linux/if_bridge.h
index fd1b6eb..3b74be1 100644
--- a/include/linux/if_bridge.h
+++ b/include/linux/if_bridge.h
@@ -104,7 +104,7 @@ struct __fdb_entry
```

```
#include <linux/netdevice.h>
```

```
-extern void briocntl_set(int (*iocntl_hook)(unsigned int, void __user *));
+extern void briocntl_set(int (*iocntl_hook)(net_t, unsigned int, void __user *));
extern int (*br_handle_frame_hook)(struct net_bridge_port *p, struct sk_buff **pskb);
extern int (*br_should_route_hook)(struct sk_buff **pskb);
```

```
diff --git a/include/linux/if_vlan.h b/include/linux/if_vlan.h
index 35cb385..8ee195a 100644
--- a/include/linux/if_vlan.h
+++ b/include/linux/if_vlan.h
@@ -62,7 +62,7 @@ struct vlan_hdr {
#define VLAN_VID_MASK 0xffff
```

```
/* found in socket.c */
```

```
-extern void vlan_iocntl_set(int (*hook)(void __user *));
+extern void vlan_iocntl_set(int (*hook)(net_t, void __user *));
```

```
#define VLAN_NAME "vlan"
```

```
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 73931a0..0b4a4dc 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -571,21 +571,21 @@ struct packet_type {
#include <linux/notifier.h>
```

```
DECLARE_PER_NET(struct net_device, loopback_dev); /* The loopback */
-extern struct net_device *dev_base; /* All devices */
-extern rwlock_t dev_base_lock; /* Device list lock */
+DECLARE_PER_NET(struct net_device *, dev_base); /* All devices */
+DECLARE_PER_NET(rwlock_t, dev_base_lock); /* Device list lock */
```

```
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);
-extern struct net_device *dev_getfirstbyhwtype(unsigned short type);
+extern struct net_device *dev_getbyhwaddr(net_t net, unsigned short type, char *hwaddr);
+extern struct net_device *dev_getfirstbyhwtype(net_t net, unsigned short type);
extern void dev_add_pack(struct packet_type *pt);
extern void dev_remove_pack(struct packet_type *pt);
extern void __dev_remove_pack(struct packet_type *pt);
```

```

-extern struct net_device *dev_get_by_flags(unsigned short flags,
+extern struct net_device *dev_get_by_flags(net_t net, unsigned short flags,
    unsigned short mask);
-extern struct net_device *dev_get_by_name(const char *name);
-extern struct net_device * __dev_get_by_name(const char *name);
+extern struct net_device *dev_get_by_name(net_t net, const char *name);
+extern struct net_device * __dev_get_by_name(net_t net, const char *name);
extern int dev_alloc_name(struct net_device *dev, const char *name);
extern int dev_open(struct net_device *dev);
extern int dev_close(struct net_device *dev);
@@ -597,8 +597,8 @@ extern void synchronize_net(void);
extern int register_netdevice_notifier(struct notifier_block *nb);
extern int unregister_netdevice_notifier(struct notifier_block *nb);
extern int call_netdevice_notifiers(unsigned long val, struct net_device *dev);
-extern struct net_device *dev_get_by_index(int ifindex);
-extern struct net_device * __dev_get_by_index(int ifindex);
+extern struct net_device *dev_get_by_index(net_t net, int ifindex);
+extern struct net_device * __dev_get_by_index(net_t net, int ifindex);
extern int dev_restart(struct net_device *dev);
#ifdef CONFIG_NETPOLL_TRAP
extern int netpoll_trap(void);
@@ -705,8 +705,8 @@ extern int netif_rx_ni(struct sk_buff *skb);
#define HAVE_NETIF_RECEIVE_SKB 1
extern int netif_receive_skb(struct sk_buff *skb);
extern int dev_valid_name(const char *name);
-extern int dev_ioctl(unsigned int cmd, void __user *);
-extern int dev_ethtool(struct ifreq *);
+extern int dev_ioctl(net_t net, unsigned int cmd, void __user *);
+extern int dev_ethtool(net_t net, struct ifreq *);
extern unsigned dev_get_flags(const struct net_device *);
extern int dev_change_flags(struct net_device *, unsigned);
extern int dev_change_name(struct net_device *, char *);
@@ -982,7 +982,7 @@ extern void dev_set_allmulti(struct net_device *dev, int inc);
extern void netdev_state_change(struct net_device *dev);
extern void netdev_features_change(struct net_device *dev);
/* Load a device via the kmod */
-extern void dev_load(const char *name);
+extern void dev_load(net_t net, const char *name);
extern void dev_mcast_init(void);
extern int netdev_max_backlog;
extern int weight_p;
diff --git a/include/net/iw_handler.h b/include/net/iw_handler.h
index 10559e9..f274eca 100644
--- a/include/net/iw_handler.h
+++ b/include/net/iw_handler.h
@@ -434,7 +434,7 @@ extern int dev_get_wireless_info(char * buffer, char **start, off_t offset,
    int length);

```

```

/* Handle IOCTLS, called in net/core/dev.c */
-extern int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd);
+extern int wireless_process_ioctl(net_t net, struct ifreq *ifr, unsigned int cmd);

/* Handle RtNetlink requests, called in net/core/rtnetlink.c */
extern int wireless_rtnetlink_set(struct net_device * dev,
diff --git a/net/802/tr.c b/net/802/tr.c
index 3324fa6..7a8cfbe 100644
--- a/net/802/tr.c
+++ b/net/802/tr.c
@@ -531,7 +531,7 @@ static int rif_seq_show(struct seq_file *seq, void *v)
    seq_puts(seq,
        "if    TR address    TTL    rcf    routing segments\n");
    else {
- struct net_device *dev = dev_get_by_index(entry->iface);
+ struct net_device *dev = dev_get_by_index(init_net(), entry->iface);
    long ttl = (long) (entry->last_used + sysctl_tr_rif_timeout)
        - (long) jiffies;

diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
index f80cfdd..e03d7de 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
@@ -51,7 +51,7 @@ static char vlan_copyright[] = "Ben Greear <greearb@candelatech.com>";
static char vlan_buggyright[] = "David S. Miller <davem@redhat.com>";

static int vlan_device_event(struct notifier_block *, unsigned long, void *);
-static int vlan_ioctl_handler(void __user *);
+static int vlan_ioctl_handler(net_t net, void __user *);
static int unregister_vlan_dev(struct net_device *, unsigned short );

static struct notifier_block vlan_notifier_block = {
@@ -118,7 +118,7 @@ static void __exit vlan_cleanup_devices(void)
    struct net_device *dev, *nxt;

    rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = nxt) {
    nxt = dev->next;
    if (dev->priv_flags & IFF_802_1Q_VLAN) {
        unregister_vlan_dev(VLAN_DEV_INFO(dev)->real_dev,
@@ -279,7 +279,7 @@ static int unregister_vlan_device(const char *vlan_IF_name)
    int ret;

- dev = dev_get_by_name(vlan_IF_name);
+ dev = dev_get_by_name(init_net(), vlan_IF_name);

```

```

ret = -EINVAL;
if (dev) {
    if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -390,7 +390,7 @@ static struct net_device *register_vlan_device(const char *eth_IF_name,
    goto out_ret_null;

    /* find the device relating to eth_IF_name. */
- real_dev = dev_get_by_name(eth_IF_name);
+ real_dev = dev_get_by_name(init_net(), eth_IF_name);
    if (!real_dev)
        goto out_ret_null;

@@ -678,7 +678,7 @@ out:
    * o execute requested action or pass command to the device driver
    * arg is really a struct vlan_ioctl_args __user *.
    */
-static int vlan_ioctl_handler(void __user *arg)
+static int vlan_ioctl_handler(net_t net, void __user *arg)
{
    int err = 0;
    unsigned short vid = 0;
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
index 9fce3a8..fa2186d 100644
--- a/net/8021q/vlan_dev.c
+++ b/net/8021q/vlan_dev.c
@@ -539,7 +539,7 @@ int vlan_dev_change_mtu(struct net_device *dev, int new_mtu)

int vlan_dev_set_ingress_priority(char *dev_name, __u32 skb_prio, short vlan_prio)
{
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);

    if (dev) {
        if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -556,7 +556,7 @@ int vlan_dev_set_ingress_priority(char *dev_name, __u32 skb_prio, short
vlan_prio

int vlan_dev_set_egress_priority(char *dev_name, __u32 skb_prio, short vlan_prio)
{
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
    struct vlan_priority_tci_mapping *mp = NULL;
    struct vlan_priority_tci_mapping *np;

@@ -596,7 +596,7 @@ int vlan_dev_set_egress_priority(char *dev_name, __u32 skb_prio, short
vlan_prio
/* Flags are defined in the vlan_dev_info class in include/linux/if_vlan.h file. */
int vlan_dev_set_vlan_flag(char *dev_name, __u32 flag, short flag_val)

```

```

{
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);

    if (dev) {
        if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -632,7 +632,7 @@ int vlan_dev_set_vlan_flag(char *dev_name, __u32 flag, short flag_val)

int vlan_dev_get_realdev_name(const char *dev_name, char* result)
{
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
    int rv = 0;
    if (dev) {
        if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -650,7 +650,7 @@ int vlan_dev_get_realdev_name(const char *dev_name, char* result)

int vlan_dev_get_vid(const char *dev_name, unsigned short* result)
{
- struct net_device *dev = dev_get_by_name(dev_name);
+ struct net_device *dev = dev_get_by_name(init_net(), dev_name);
    int rv = 0;
    if (dev) {
        if (dev->priv_flags & IFF_802_1Q_VLAN) {
@@ -821,7 +821,7 @@ int vlan_dev_ioctl(struct net_device *dev, struct ifreq *ifr, int cmd)
    break;

case SIOCETHHTOOL:
- err = dev_ethhtool(&ifrr);
+ err = dev_ethhtool(real_dev->nd_net, &ifrr);
}

if (!err)
diff --git a/net/8021q/vlanproc.c b/net/8021q/vlanproc.c
index abcf58c..0e93991 100644
--- a/net/8021q/vlanproc.c
+++ b/net/8021q/vlanproc.c
@@ -253,12 +253,12 @@ static void *vlan_seq_start(struct seq_file *seq, loff_t *pos)
    struct net_device *dev;
    loff_t i = 1;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));

    if (*pos == 0)
        return SEQ_START_TOKEN;

- for (dev = vlan_skip(dev_base); dev && i < *pos;

```

```

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

    return (i == *pos) ? dev : NULL;
@@ -269,13 +269,13 @@ static void *vlan_seq_next(struct seq_file *seq, void *v, loff_t *pos)
    ++*pos;

    return vlan_skip((v == SEQ_START_TOKEN)
-        ? dev_base
+        ? per_net(dev_base, init_net())
        : ((struct net_device *)v)->next);
}

static void vlan_seq_stop(struct seq_file *seq, void *v)
{
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int vlan_seq_show(struct seq_file *seq, void *v)
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index 61f36b1..4cdcae3 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -677,7 +677,7 @@ static int atif_ioctl(int cmd, void __user *arg)
    if (copy_from_user(&atreq, arg, sizeof(atreq)))
        return -EFAULT;

- dev = __dev_get_by_name(atreq.ifr_name);
+ dev = __dev_get_by_name(init_net(), atreq.ifr_name);
    if (!dev)
        return -ENODEV;

@@ -901,7 +901,7 @@ static int atrtr_ioctl(unsigned int cmd, void __user *arg)
    if (copy_from_user(name, rt.rt_dev, IFNAMSIZ-1))
        return -EFAULT;
    name[IFNAMSIZ-1] = '\0';
- dev = __dev_get_by_name(name);
+ dev = __dev_get_by_name(init_net(), name);
    if (!dev)
        return -ENODEV;
}

@@ -1273,7 +1273,7 @@ static __inline__ int is_ip_over_ddp(struct sk_buff *skb)

static int handle_ip_over_ddp(struct sk_buff *skb)
{
- struct net_device *dev = __dev_get_by_name("ipddp0");
+ struct net_device *dev = __dev_get_by_name(init_net(), "ipddp0");

```



```

struct net_device_stats *stats;

/* This needs to be able to handle ipddp"N" devices */
diff --git a/net/atm/mpc.c b/net/atm/mpc.c
index 4fdb1af..e17c10b 100644
--- a/net/atm/mpc.c
+++ b/net/atm/mpc.c
@@ -244,7 +244,7 @@ static struct net_device *find_lec_by_itfnum(int itf)
    char name[IFNAMSIZ];

    sprintf(name, "lec%d", itf);
- dev = dev_get_by_name(name);
+ dev = dev_get_by_name(init_net(), name);

    return dev;
}
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 8c187a6..e2f6fed 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -632,7 +632,7 @@ static int ax25_setsockopt(struct socket *sock, int level, int optname,
    break;
}

- dev = dev_get_by_name(devname);
+ dev = dev_get_by_name(init_net(), devname);
    if (dev == NULL) {
        res = -ENODEV;
        break;
    }
diff --git a/net/bridge/br_if.c b/net/bridge/br_if.c
index 55bb263..22509f1 100644
--- a/net/bridge/br_if.c
+++ b/net/bridge/br_if.c
@@ -45,7 +45,7 @@ static int port_cost(struct net_device *dev)

    old_fs = get_fs();
    set_fs(KERNEL_DS);
- err = dev_ethtool(&ifr);
+ err = dev_ethtool(dev->nd_net, &ifr);
    set_fs(old_fs);

    if (!err) {
@@ -328,7 +328,7 @@ int br_del_bridge(const char *name)
    int ret = 0;

    rtnl_lock();
- dev = __dev_get_by_name(name);
+ dev = __dev_get_by_name(init_net(), name);

```

```

if (dev == NULL)
    ret = -ENXIO; /* Could not find device */

@@ -483,7 +483,7 @@ void __exit br_cleanup_bridges(void)
    struct net_device *dev, *nxt;

    rtnl_lock();
- for (dev = dev_base; dev; dev = nxt) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = nxt) {
    nxt = dev->next;
    if (dev->priv_flags & IFF_EBRIDGE)
        del_br(dev->priv);
diff --git a/net/bridge/br_ioctl.c b/net/bridge/br_ioctl.c
index 4c61a7e..2be1c2d 100644
--- a/net/bridge/br_ioctl.c
+++ b/net/bridge/br_ioctl.c
@@ -18,6 +18,7 @@
#include <linux/if_bridge.h>
#include <linux/netdevice.h>
#include <linux/times.h>
+#include <net/net_namespace.h>
#include <asm/uaccess.h>
#include "br_private.h"

@@ -27,7 +28,7 @@ static int get_bridge_ifindices(int *indices, int num)
    struct net_device *dev;
    int i = 0;

- for (dev = dev_base; dev && i < num; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev && i < num; dev = dev->next) {
    if (dev->priv_flags & IFF_EBRIDGE)
        indices[i++] = dev->ifindex;
    }
@@ -88,7 +89,7 @@ static int add_del_if(struct net_bridge *br, int ifindex, int isadd)
    if (!capable(CAP_NET_ADMIN))
        return -EPERM;

- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
    if (dev == NULL)
        return -EINVAL;

@@ -362,7 +363,7 @@ static int old_deviceless(void __user *uarg)
    return -EOPNOTSUPP;
}

-int br_ioctl_deviceless_stub(unsigned int cmd, void __user *uarg)
+int br_ioctl_deviceless_stub(net_t net, unsigned int cmd, void __user *uarg)

```

```

{
    switch (cmd) {
    case SIOCGIFBR:
diff --git a/net/bridge/br_netlink.c b/net/bridge/br_netlink.c
index a913968..119b97d 100644
--- a/net/bridge/br_netlink.c
+++ b/net/bridge/br_netlink.c
@@ -13,6 +13,7 @@
#include <linux/kernel.h>
#include <linux/rtnetlink.h>
#include <net/netlink.h>
+#include <net/net_namespace.h>
#include "br_private.h"

static inline size_t br_nlmsg_size(void)
@@ -106,8 +107,8 @@ static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
    struct net_device *dev;
    int idx;

- read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next) {
    /* not a bridge port */
    if (dev->br_port == NULL || idx < cb->args[0])
        goto skip;
@@ -119,7 +120,7 @@ static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
skip:
    ++idx;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    cb->args[0] = idx;

@@ -153,7 +154,7 @@ static int br_rtm_setlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void
*arg)
    if (new_state > BR_STATE_BLOCKING)
        return -EINVAL;

- dev = __dev_get_by_index(ifm->ifi_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifi_index);
    if (!dev)
        return -ENODEV;

diff --git a/net/bridge/br_private.h b/net/bridge/br_private.h
index f1712b9..1d60ee3 100644
--- a/net/bridge/br_private.h

```

```

+++ b/net/bridge/br_private.h
@@ -189,7 +189,7 @@ extern int br_handle_frame(struct net_bridge_port *p, struct sk_buff
**pskb);

/* br_ioctl.c */
extern int br_dev_ioctl(struct net_device *dev, struct ifreq *rq, int cmd);
-extern int br_ioctl_deviceless_stub(unsigned int cmd, void __user *arg);
+extern int br_ioctl_deviceless_stub(net_t net, unsigned int cmd, void __user *arg);

/* br_netfilter.c */
#ifdef CONFIG_BRIDGE_NETFILTER
diff --git a/net/core/dev.c b/net/core/dev.c
index d8aa534..32fe905 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -175,26 +175,27 @@ 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);
+DEFINE_PER_NET(struct net_device *, dev_base);
+static DEFINE_PER_NET(struct net_device **, dev_tail);
+DEFINE_PER_NET(rwlock_t, dev_base_lock);

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

#define NETDEV_HASHBITS 8
-static struct hlist_head dev_name_head[1<<NETDEV_HASHBITS];
-static struct hlist_head dev_index_head[1<<NETDEV_HASHBITS];
+typedef struct hlist_head dev_hlists_t[1<<NETDEV_HASHBITS];
+static DEFINE_PER_NET(dev_hlists_t, dev_name_head);
+static DEFINE_PER_NET(dev_hlists_t, dev_index_head);

-static inline struct hlist_head *dev_name_hash(const char *name)
+static inline struct hlist_head *dev_name_hash(net_t net, const char *name)
{
    unsigned hash = full_name_hash(name, strlen(name), IFNAMSIZ);
-    return &dev_name_head[hash & ((1<<NETDEV_HASHBITS)-1)];
+    return &per_net(dev_name_head, net)[hash & ((1<<NETDEV_HASHBITS)-1)];
}

-static inline struct hlist_head *dev_index_hash(int ifindex)
+static inline struct hlist_head *dev_index_hash(net_t net, int ifindex)

```

```

{
- return &dev_index_head[ifindex & ((1<<NETDEV_HASHBITS)-1)];
+ return &per_net(dev_index_head, net)[ifindex & ((1<<NETDEV_HASHBITS)-1)];
}

/*
@@ -418,7 +419,7 @@ unsigned long netdev_boot_base(const char *prefix, int unit)
    * If device already registered then return base of 1
    * to indicate not to probe for this interface
    */
- if (__dev_get_by_name(name))
+ if (__dev_get_by_name(init_net(), name))
    return 1;

    for (i = 0; i < NETDEV_BOOT_SETUP_MAX; i++)
@@ -473,11 +474,11 @@ __setup("netdev=", netdev_boot_setup);
    * careful with locks.
    */

-struct net_device *__dev_get_by_name(const char *name)
+struct net_device *__dev_get_by_name(net_t net, const char *name)
{
    struct hlist_node *p;

- hlist_for_each(p, dev_name_hash(name)) {
+ hlist_for_each(p, dev_name_hash(net, name)) {
    struct net_device *dev
        = hlist_entry(p, struct net_device, name_hlist);
    if (!strcmp(dev->name, name, IFNAMSIZ))
@@ -497,15 +498,15 @@ struct net_device *__dev_get_by_name(const char *name)
    * matching device is found.
    */

-struct net_device *dev_get_by_name(const char *name)
+struct net_device *dev_get_by_name(net_t net, const char *name)
{
    struct net_device *dev;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_name(name);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_name(net, name);
    if (dev)
        dev_hold(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
    return dev;
}

```

```
@@ -520,11 +521,11 @@ struct net_device *dev_get_by_name(const char *name)
 * or @dev_base_lock.
 */
```

```
-struct net_device * __dev_get_by_index(int ifindex)
+struct net_device * __dev_get_by_index(net_t net, int ifindex)
{
    struct hlist_node *p;
```

```
- hlist_for_each(p, dev_index_hash(ifindex)) {
+ hlist_for_each(p, dev_index_hash(net, ifindex)) {
    struct net_device *dev
        = hlist_entry(p, struct net_device, index_hlist);
    if (dev->ifindex == ifindex)
```

```
@@ -544,15 +545,15 @@ struct net_device * __dev_get_by_index(int ifindex)
 * dev_put to indicate they have finished with it.
 */
```

```
-struct net_device *dev_get_by_index(int ifindex)
+struct net_device *dev_get_by_index(net_t net, int ifindex)
{
    struct net_device *dev;
```

```
- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifindex);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_index(net, ifindex);
    if (dev)
        dev_hold(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
    return dev;
}
```

```
@@ -570,13 +571,13 @@ struct net_device *dev_get_by_index(int ifindex)
 * If the API was consistent this would be __dev_get_by_hwaddr
 */
```

```
-struct net_device *dev_getbyhwaddr(unsigned short type, char *ha)
+struct net_device *dev_getbyhwaddr(net_t net, unsigned short type, char *ha)
{
    struct net_device *dev;
```

```
    ASSERT_RTNL();
```

```
- for (dev = dev_base; dev; dev = dev->next)
+ for (dev = per_net(dev_base, net); dev; dev = dev->next)
```

```

    if (dev->type == type &&
        !memcmp(dev->dev_addr, ha, dev->addr_len))
        break;
@@ -585,12 +586,12 @@ struct net_device *dev_getbyhwaddr(unsigned short type, char *ha)

EXPORT_SYMBOL(dev_getbyhwaddr);

-struct net_device *dev_getfirstbyhwtype(unsigned short type)
+struct net_device *dev_getfirstbyhwtype(net_t net, unsigned short type)
{
    struct net_device *dev;

    rtnl_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
    if (dev->type == type) {
        dev_hold(dev);
        break;
@@ -613,18 +614,18 @@ EXPORT_SYMBOL(dev_getfirstbyhwtype);
    * dev_put to indicate they have finished with it.
    */

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

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev = per_net(dev_base, net); dev != NULL; dev = dev->next) {
    if (((dev->flags ^ if_flags) & mask) == 0) {
        dev_hold(dev);
        break;
    }
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
    return dev;
}

@@ -675,6 +676,10 @@ int dev_alloc_name(struct net_device *dev, const char *name)
    const int max_netdevices = 8*PAGE_SIZE;
    long *inuse;
    struct net_device *d;
+ net_t net;
+
+ BUG_ON(null_net(dev->nd_net));

```

```

+ net = dev->nd_net;

p = strnchr(name, IFNAMSIZ-1, '%');
if (p) {
@@ -691,7 +696,7 @@ int dev_alloc_name(struct net_device *dev, const char *name)
    if (!inuse)
        return -ENOMEM;

- for (d = dev_base; d; d = d->next) {
+ for (d = per_net(dev_base, net); d; d = d->next) {
    if (!sscanf(d->name, name, &i))
        continue;
    if (i < 0 || i >= max_netdevices)
@@ -708,7 +713,7 @@ int dev_alloc_name(struct net_device *dev, const char *name)
}

    snprintf(buf, sizeof(buf), name, i);
- if (!__dev_get_by_name(buf)) {
+ if (!__dev_get_by_name(net, buf)) {
    strcpy(dev->name, buf, IFNAMSIZ);
    return i;
}
@@ -732,9 +737,12 @@ int dev_alloc_name(struct net_device *dev, const char *name)
int dev_change_name(struct net_device *dev, char *newname)
{
    int err = 0;
+ net_t net;

    ASSERT_RTNL();
+ BUG_ON(null_net(dev->nd_net));

+ net = dev->nd_net;
    if (dev->flags & IFF_UP)
        return -EBUSY;

@@ -747,7 +755,7 @@ int dev_change_name(struct net_device *dev, char *newname)
    return err;
    strcpy(newname, dev->name);
}
- else if (__dev_get_by_name(newname))
+ else if (__dev_get_by_name(net, newname))
    return -EEXIST;
else
    strcpy(dev->name, newname, IFNAMSIZ);
@@ -755,7 +763,7 @@ int dev_change_name(struct net_device *dev, char *newname)
err = class_device_rename(&dev->class_dev, dev->name);
if (!err) {
    hlist_del(&dev->name_hlist);

```



```

- hlist_add_head(&dev->name_hlist, dev_name_hash(dev->name));
+ hlist_add_head(&dev->name_hlist, dev_name_hash(net, dev->name));
  raw_notifier_call_chain(&netdev_chain,
    NETDEV_CHANGENAME, dev);
}
@@ -801,13 +809,13 @@ void netdev_state_change(struct net_device *dev)
  * available in this kernel then it becomes a nop.
  */

-void dev_load(const char *name)
+void dev_load(net_t net, const char *name)
{
  struct net_device *dev;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_name(name);
- read_unlock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_name(net, name);
+ read_unlock(&per_net(dev_base_lock, net));

  if (!dev && capable(CAP_SYS_MODULE))
    request_module("%s", name);
@@ -977,11 +985,14 @@ int register_netdevice_notifier(struct notifier_block *nb)
  rtnl_lock();
  err = raw_notifier_chain_register(&netdev_chain, nb);
  if (!err) {
- for (dev = dev_base; dev; dev = dev->next) {
-   nb->notifier_call(nb, NETDEV_REGISTER, dev);
+ net_t net;
+ for_each_net(net) {
+   for (dev = per_net(dev_base, net); dev; dev = dev->next) {
+     nb->notifier_call(nb, NETDEV_REGISTER, dev);

-   if (dev->flags & IFF_UP)
-     nb->notifier_call(nb, NETDEV_UP, dev);
+   if (dev->flags & IFF_UP)
+     nb->notifier_call(nb, NETDEV_UP, dev);
+   }
+ }
  }
  rtnl_unlock();
@@ -1991,7 +2002,7 @@ int register_gifconf(unsigned int family, gifconf_func_t * gifconf)
  * match. --pb
  */

-static int dev_ifname(struct ifreq __user *arg)
+static int dev_ifname(net_t net, struct ifreq __user *arg)

```

```

{
    struct net_device *dev;
    struct ifreq ifr;
@@ -2003,15 +2014,15 @@ static int dev_ifname(struct ifreq __user *arg)
    if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
        return -EFAULT;

- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifr.ifr_ifindex);
+ read_lock(&per_net(dev_base_lock, net));
+ dev = __dev_get_by_index(net, ifr.ifr_ifindex);
    if (!dev) {
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
        return -ENODEV;
    }

    strcpy(ifr.ifr_name, dev->name);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

    if (copy_to_user(arg, &ifr, sizeof(struct ifreq)))
        return -EFAULT;
@@ -2024,7 +2035,7 @@ static int dev_ifname(struct ifreq __user *arg)
    * Thus we will need a 'compatibility mode'.
    */

-static int dev_ifconf(char __user *arg)
+static int dev_ifconf(net_t net, char __user *arg)
{
    struct ifconf ifc;
    struct net_device *dev;
@@ -2048,7 +2059,7 @@ static int dev_ifconf(char __user *arg)
    */

    total = 0;
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
    for (i = 0; i < NPROTO; i++) {
        if (gifconf_list[i]) {
            int done;
@@ -2080,31 +2091,35 @@ 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)
+static __inline__ struct net_device *dev_get_idx(net_t net, loff_t pos)
{

```

```

struct net_device *dev;
loff_t i;

- for (i = 0, dev = dev_base; dev && i < pos; ++i, dev = dev->next);
+ for (i = 0, dev = per_net(dev_base, net); dev && i < pos; ++i, dev = dev->next);

return i == pos ? dev : NULL;
}

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;
+ net_t net = net_from_voidp(seq->private);
+
+ read_lock(&per_net(dev_base_lock, net));
+ return *pos ? dev_get_idx(net, *pos - 1) : SEQ_START_TOKEN;
}

void *dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
+ net_t net = net_from_voidp(seq->private);
++*pos;
- return v == SEQ_START_TOKEN ? dev_base : ((struct net_device *)v)->next;
+ return v == SEQ_START_TOKEN ? per_net(dev_base, net) : ((struct net_device *)v)->next;
}

void dev_seq_stop(struct seq_file *seq, void *v)
{
- read_unlock(&dev_base_lock);
+ net_t net = net_from_voidp(seq->private);
+ read_unlock(&per_net(dev_base_lock, net));
}

static void dev_seq_printf_stats(struct seq_file *seq, struct net_device *dev)
@@ -2198,7 +2213,22 @@ static struct seq_operations dev_seq_ops = {

static int dev_seq_open(struct inode *inode, struct file *file)
{
- return seq_open(file, &dev_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &dev_seq_ops);
+ if (!res) {
+ seq = file->private_data;
+ seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
}

```

```

+}
+
+static int dev_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
+}

static struct file_operations dev_seq_fops = {
@@ -2206,7 +2236,7 @@ static struct file_operations dev_seq_fops = {
    .open    = dev_seq_open,
    .read    = seq_read,
    .llseek  = seq_lseek,
- .release  = seq_release,
+ .release  = dev_seq_release,
};

static struct seq_operations softnet_seq_ops = {
@@ -2235,23 +2265,44 @@ extern int wireless_proc_init(void);
#define wireless_proc_init() 0
#endif

-static int __init dev_proc_init(void)
+static int dev_proc_net_init(net_t net)
+{
+ int rc = -ENOMEM;

- if (!proc_net_fops_create(init_net(), "dev", S_IRUGO, &dev_seq_fops))
+ if (!proc_net_fops_create(net, "dev", S_IRUGO, &dev_seq_fops))
+ goto out;
- if (!proc_net_fops_create(init_net(), "softnet_stat", S_IRUGO, &softnet_seq_fops))
+ if (!proc_net_fops_create(net, "softnet_stat", S_IRUGO, &softnet_seq_fops))
+ goto out_dev;
- if (wireless_proc_init())
- goto out_softnet;
- rc = 0;
out:
return rc;
-out_softnet:
- proc_net_remove(init_net(), "softnet_stat");
out_dev:
- proc_net_remove(init_net(), "dev");
+ proc_net_remove(net, "dev");
+ goto out;
+}
+

```

```

+static void dev_proc_net_exit(net_t net)
+{
+ proc_net_remove(net, "softnet_stat");
+ proc_net_remove(net, "dev");
+}
+
+static struct pernet_operations dev_proc_ops = {
+ .init = dev_proc_net_init,
+ .exit = dev_proc_net_exit,
+};
+
+static int __init dev_proc_init(void)
+{
+ int rc;
+ if ((rc = register_pernet_subsys(&dev_proc_ops)))
+ goto out;
+ if ((rc = wireless_proc_init()))
+ goto out_softnet;
+out:
+ return rc;
+out_softnet:
+ unregister_pernet_subsys(&dev_proc_ops);
+ goto out;
+}
+
+@@ -2485,10 +2536,10 @@ int dev_set_mac_address(struct net_device *dev, struct sockaddr
+*sa)
+/*
+ * Perform the SIOCxIFxxx calls.
+ */
+
+-static int dev_ifsioc(struct ifreq *ifr, unsigned int cmd)
+static int dev_ifsioc(net_t net, struct ifreq *ifr, unsigned int cmd)
+{
+ int err;
+
+ struct net_device *dev = __dev_get_by_name(ifr->ifr_name);
+ struct net_device *dev = __dev_get_by_name(net, ifr->ifr_name);
+
+ if (!dev)
+ return -ENODEV;
+@@ -2641,7 +2692,7 @@ static int dev_ifsioc(struct ifreq *ifr, unsigned int cmd)
+ * positive or a negative errno code on error.
+ */
+
+-int dev_ioctl(unsigned int cmd, void __user *arg)
+int dev_ioctl(net_t net, unsigned int cmd, void __user *arg)
+{
+ struct ifreq ifr;
+ int ret;

```

```
@@ -2654,12 +2705,12 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
```

```
    if (cmd == SIOCGIFCONF) {  
        rtnl_lock();  
- ret = dev_ifconf((char __user *) arg);  
+ ret = dev_ifconf(net, (char __user *) arg);  
        rtnl_unlock();  
        return ret;  
    }  
    if (cmd == SIOCGIFNAME)  
- return dev_ifname((struct ifreq __user *)arg);  
+ return dev_ifname(net, (struct ifreq __user *)arg);
```

```
    if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))  
        return -EFAULT;
```

```
@@ -2689,10 +2740,10 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
```

```
    case SIOCGIFMAP:  
    case SIOCGIFINDEX:  
    case SIOCGIFTXQLEN:  
- dev_load(ifr.ifr_name);  
- read_lock(&dev_base_lock);  
- ret = dev_ifsioc(&ifr, cmd);  
- read_unlock(&dev_base_lock);  
+ dev_load(net, ifr.ifr_name);  
+ read_lock(&per_net(dev_base_lock, net));  
+ ret = dev_ifsioc(net, &ifr, cmd);  
+ read_unlock(&per_net(dev_base_lock, net));  
    if (!ret) {  
        if (colon)  
            *colon = ':';
```

```
@@ -2703,9 +2754,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)  
    return ret;
```

```
    case SIOCETHTOOL:  
- dev_load(ifr.ifr_name);  
+ dev_load(net, ifr.ifr_name);  
    rtnl_lock();  
- ret = dev_ethtool(&ifr);  
+ ret = dev_ethtool(net, &ifr);  
    rtnl_unlock();  
    if (!ret) {  
        if (colon)
```

```
@@ -2727,9 +2778,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
```

```
    case SIOCSIFNAME:  
        if (!capable(CAP_NET_ADMIN))  
            return -EPERM;  
- dev_load(ifr.ifr_name);  
+ dev_load(net, ifr.ifr_name);
```

```

    rtnl_lock();
-   ret = dev_ifsioc(&ifr, cmd);
+   ret = dev_ifsioc(net, &ifr, cmd);
    rtnl_unlock();
    if (!ret) {
        if (colon)
@@ -2768,9 +2819,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    /* fall through */
    case SIOCBONDSLAVEINFOQUERY:
    case SIOCBONDINFOQUERY:
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
-   ret = dev_ifsioc(&ifr, cmd);
+   ret = dev_ifsioc(net, &ifr, cmd);
    rtnl_unlock();
    return ret;

@@ -2790,9 +2841,9 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    if (cmd == SIOCWANDEV ||
        (cmd >= SIOCDEVPRIVATE &&
         cmd <= SIOCDEVPRIVATE + 15)) {
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
-   ret = dev_ifsioc(&ifr, cmd);
+   ret = dev_ifsioc(net, &ifr, cmd);
    rtnl_unlock();
    if (!ret && copy_to_user(arg, &ifr,
        sizeof(struct ifreq)))
@@ -2810,10 +2861,10 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    if (!capable(CAP_NET_ADMIN))
        return -EPERM;
    }
-   dev_load(ifr.ifr_name);
+   dev_load(net, ifr.ifr_name);
    rtnl_lock();
    /* Follow me in net/core/wireless.c */
-   ret = wireless_process_ioctl(&ifr, cmd);
+   ret = wireless_process_ioctl(net, &ifr, cmd);
    rtnl_unlock();
    if (IW_IS_GET(cmd) &&
        copy_to_user(arg, &ifr,
@@ -2834,13 +2885,13 @@ int dev_ioctl(unsigned int cmd, void __user *arg)
    * number. The caller must hold the rtnl semaphore or the
    * dev_base_lock to be sure it remains unique.
    */
-static int dev_new_index(void)

```

```

+static int dev_new_index(net_t net)
{
    static int ifindex;
    for (;;) {
        if (++ifindex <= 0)
            ifindex = 1;
- if (!__dev_get_by_index(ifindex))
+ if (!__dev_get_by_index(net, ifindex))
        return ifindex;
    }
}

@@ -2880,6 +2931,7 @@ int register_netdevice(struct net_device *dev)
    struct hlist_head *head;
    struct hlist_node *p;
    int ret;
+ net_t net;

    BUG_ON(dev_boot_phase);
    ASSERT_RTNL();
@@ -2888,6 +2940,8 @@ int register_netdevice(struct net_device *dev)

    /* When net_device's are persistent, this will be fatal. */
    BUG_ON(dev->reg_state != NETREG_UNINITIALIZED);
+ BUG_ON(null_net(dev->nd_net));
+ net = dev->nd_net;

    spin_lock_init(&dev->queue_lock);
    spin_lock_init(&dev->xmit_lock);
@@ -2913,12 +2967,12 @@ int register_netdevice(struct net_device *dev)
    goto out;
}

- dev->ifindex = dev_new_index();
+ dev->ifindex = dev_new_index(net);
if (dev->iflink == -1)
    dev->iflink = dev->ifindex;

/* Check for existence of name */
- head = dev_name_hash(dev->name);
+ head = dev_name_hash(net, dev->name);
hlist_for_each(p, head) {
    struct net_device *d
        = hlist_entry(p, struct net_device, name_hlist);
@@ -2980,13 +3034,13 @@ int register_netdevice(struct net_device *dev)

    dev->next = NULL;
    dev_init_scheduler(dev);
- write_lock_bh(&dev_base_lock);

```



```

- *dev_tail = dev;
- dev_tail = &dev->next;
+ write_lock_bh(&per_net(dev_base_lock, net));
+ *per_net(dev_tail, net) = dev;
+ per_net(dev_tail, net) = &dev->next;
  hlist_add_head(&dev->name_hlist, head);
- hlist_add_head(&dev->index_hlist, dev_index_hash(dev->ifindex));
+ hlist_add_head(&dev->index_hlist, dev_index_hash(net, dev->ifindex));
  dev_hold(dev);
- write_unlock_bh(&dev_base_lock);
+ write_unlock_bh(&per_net(dev_base_lock, net));

  /* Notify protocols, that a new device appeared. */
  raw_notifier_call_chain(&netdev_chain, NETDEV_REGISTER, dev);
@@ -3252,6 +3306,7 @@ void synchronize_net(void)
int unregister_netdevice(struct net_device *dev)
{
  struct net_device *d, **dp;
+ net_t net = dev->nd_net;

  BUG_ON(dev_boot_phase);
  ASSERT_RTNL();
@@ -3270,15 +3325,15 @@ int unregister_netdevice(struct net_device *dev)
  dev_close(dev);

  /* And unlink it from device chain. */
- for (dp = &dev_base; (d = *dp) != NULL; dp = &d->next) {
+ for (dp = &per_net(dev_base, net); (d = *dp) != NULL; dp = &d->next) {
  if (d == dev) {
-   write_lock_bh(&dev_base_lock);
+   write_lock_bh(&per_net(dev_base_lock, net));
    hlist_del(&dev->name_hlist);
    hlist_del(&dev->index_hlist);
-   if (dev_tail == &dev->next)
-     dev_tail = dp;
+   if (per_net(dev_tail, net) == &dev->next)
+     per_net(dev_tail, net) = dp;
    *dp = d->next;
-   write_unlock_bh(&dev_base_lock);
+   write_unlock_bh(&per_net(dev_base_lock, net));
    break;
  }
}

@@ -3464,6 +3519,26 @@ static int __init netdev_dma_register(void)
static int __init netdev_dma_register(void) { return -ENODEV; }
#endif /* CONFIG_NET_DMA */

+/* Initialize per network namespace state */

```

```

+static int netdev_init(net_t net)
+{
+ int i;
+ per_net(dev_tail, net) = &per_net(dev_base, net);
+ rwlock_init(&per_net(dev_base_lock, net));
+
+ for (i = 0; i < ARRAY_SIZE(per_net(dev_name_head, net)); i++)
+ INIT_HLIST_HEAD(&per_net(dev_name_head, net)[i]);
+
+ for (i = 0; i < ARRAY_SIZE(per_net(dev_index_head, net)); i++)
+ INIT_HLIST_HEAD(&per_net(dev_index_head, net)[i]);
+
+ return 0;
+}
+
+static struct pernet_operations netdev_net_ops = {
+ .init = netdev_init,
+};
+
+/*
+ * Initialize the DEV module. At boot time this walks the device list and
+ * unhooks any devices that fail to initialise (normally hardware not
@@ -3491,11 +3566,8 @@ static int __init net_dev_init(void)
+ for (i = 0; i < 16; i++)
+ INIT_LIST_HEAD(&ptype_base[i]);
+
- for (i = 0; i < ARRAY_SIZE(dev_name_head); i++)
- INIT_HLIST_HEAD(&dev_name_head[i]);
-
- for (i = 0; i < ARRAY_SIZE(dev_index_head); i++)
- INIT_HLIST_HEAD(&dev_index_head[i]);
+ if (register_pernet_subsys(&netdev_net_ops))
+ goto out;

+/*
+ * Initialise the packet receive queues.
diff --git a/net/core/dev_mcast.c b/net/core/dev_mcast.c
index 623e606..131746b 100644
--- a/net/core/dev_mcast.c
+++ b/net/core/dev_mcast.c
@@ -221,11 +221,12 @@ void dev_mc_discard(struct net_device *dev)
#ifdef CONFIG_PROC_FS
static void *dev_mc_seq_start(struct seq_file *seq, loff_t *pos)
{
+ net_t net = net_from_voidp(seq->private);
+ struct net_device *dev;
+ loff_t off = 0;

```

```

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev = per_net(dev_base, net); dev; dev = dev->next) {
    if (off++ == *pos)
        return dev;
}
@@ -241,7 +242,8 @@ static void *dev_mc_seq_next(struct seq_file *seq, void *v, loff_t *pos)

static void dev_mc_seq_stop(struct seq_file *seq, void *v)
{
- read_unlock(&dev_base_lock);
+ net_t net = net_from_voidp(seq->private);
+ read_unlock(&per_net(dev_base_lock, net));
}

@@ -275,7 +277,22 @@ static struct seq_operations dev_mc_seq_ops = {

static int dev_mc_seq_open(struct inode *inode, struct file *file)
{
- return seq_open(file, &dev_mc_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &dev_mc_seq_ops);
+ if (!res) {
+     seq = file->private_data;
+     seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
+}
+
+static int dev_mc_seq_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
+}

static struct file_operations dev_mc_seq_fops = {
@@ -283,14 +300,31 @@ static struct file_operations dev_mc_seq_fops = {
    .open    = dev_mc_seq_open,
    .read    = seq_read,
    .llseek  = seq_lseek,
-   .release = seq_release,
+   .release = dev_mc_seq_release,
};

```

```
#endif
```

```
+static int dev_mc_net_init(net_t net)
+{
+ if (!proc_net_fops_create(net, "dev_mcast", 0, &dev_mc_seq_fops))
+ return -ENOMEM;
+ return 0;
+}
+
+static void dev_mc_net_exit(net_t net)
+{
+ proc_net_remove(net, "dev_mcast");
+}
+
+static struct pernet_operations dev_mc_net_ops = {
+ .init = dev_mc_net_init,
+ .exit = dev_mc_net_exit,
+};
+
+void __init dev_mcast_init(void)
+{
+ - proc_net_fops_create(init_net(), "dev_mcast", 0, &dev_mc_seq_fops);
+ register_pernet_subsys(&dev_mc_net_ops);
+}
```

```
EXPORT_SYMBOL(dev_mc_add);
```

```
diff --git a/net/core/ethtool.c b/net/core/ethtool.c
```

```
index 87dc556..d142377 100644
```

```
--- a/net/core/ethtool.c
```

```
+++ b/net/core/ethtool.c
```

```
@@ -798,9 +798,9 @@ static int ethtool_get_perm_addr(struct net_device *dev, void __user
*useraddr)
```

```
/* The main entry point in this file. Called from net/core/dev.c */
```

```
-int dev_ethtool(struct ifreq *ifr)
+int dev_ethtool(net_t net, struct ifreq *ifr)
+{
+ struct net_device *dev = __dev_get_by_name(ifr->ifr_name);
+ struct net_device *dev = __dev_get_by_name(net, ifr->ifr_name);
+ void __user *useraddr = ifr->ifr_data;
+ u32 ethcmd;
+ int rc;
```

```
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
```

```
index ffc31c1..2fa2708 100644
```

```
--- a/net/core/fib_rules.c
```

```
+++ b/net/core/fib_rules.c
```

```

@@ -12,6 +12,7 @@
#include <linux/kernel.h>
#include <linux/list.h>
#include <net/net_namespace.h>
+#include <net/sock.h>
#include <net/fib_rules.h>

static LIST_HEAD(rules_ops);
@@ -155,6 +156,7 @@ EXPORT_SYMBOL_GPL(fib_rules_lookup);

int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void *arg)
{
+ net_t net = skb->sk->sk_net;
  struct fib_rule_hdr *frh = nlmsg_data(nlh);
  struct fib_rules_ops *ops = NULL;
  struct fib_rule *rule, *r, *last = NULL;
@@ -188,7 +190,7 @@ int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void *arg)

  rule->ifindex = -1;
  nla_strncpy(rule->ifname, tb[FRA_IFNAME], IFNAMSIZ);
- dev = __dev_get_by_name(rule->ifname);
+ dev = __dev_get_by_name(net, rule->ifname);
  if (dev)
    rule->ifindex = dev->ifindex;
}
diff --git a/net/core/link_watch.c b/net/core/link_watch.c
index 549a2ce..8e317cf 100644
--- a/net/core/link_watch.c
+++ b/net/core/link_watch.c
@@ -63,12 +63,13 @@ static unsigned char default_operstate(const struct net_device *dev)

static void rfc2863_policy(struct net_device *dev)
{
+ net_t net = dev->nd_net;
  unsigned char operstate = default_operstate(dev);

  if (operstate == dev->operstate)
    return;

- write_lock_bh(&dev_base_lock);
+ write_lock_bh(&per_net(dev_base_lock, net));

  switch(dev->link_mode) {
  case IF_LINK_MODE_DORMANT:
@@ -83,7 +84,7 @@ static void rfc2863_policy(struct net_device *dev)

  dev->operstate = operstate;

```

```
- write_unlock_bh(&dev_base_lock);
+ write_unlock_bh(&per_net(dev_base_lock, net));
}
```

```
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 90e1d2e..f5d4f92 100644
```

```
--- a/net/core/neighbour.c
```

```
+++ b/net/core/neighbour.c
```

```
@@ -1438,6 +1438,7 @@ int neigh_table_clear(struct neigh_table *tbl)
```

```
int neigh_delete(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
{
```

```
+ net_t net = skb->sk->sk_net;
```

```
struct ndmsg *ndm;
```

```
struct nlattr *dst_attr;
```

```
struct neigh_table *tbl;
```

```
@@ -1453,7 +1454,7 @@ int neigh_delete(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
```

```
ndm = nlmsg_data(nlh);
```

```
if (ndm->ndm_ifindex) {
```

```
- dev = dev_get_by_index(ndm->ndm_ifindex);
```

```
+ dev = dev_get_by_index(net, ndm->ndm_ifindex);
```

```
if (dev == NULL) {
```

```
err = -ENODEV;
```

```
goto out;
```

```
@@ -1503,6 +1504,7 @@ out:
```

```
int neigh_add(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
{
```

```
+ net_t net = skb->sk->sk_net;
```

```
struct ndmsg *ndm;
```

```
struct nlattr *tb[NDA_MAX+1];
```

```
struct neigh_table *tbl;
```

```
@@ -1519,7 +1521,7 @@ int neigh_add(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
```

```
ndm = nlmsg_data(nlh);
```

```
if (ndm->ndm_ifindex) {
```

```
- dev = dev_get_by_index(ndm->ndm_ifindex);
```

```
+ dev = dev_get_by_index(net, ndm->ndm_ifindex);
```

```
if (dev == NULL) {
```

```
err = -ENODEV;
```

```
goto out;
```

```
diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
```

```
index b08c1be..1be6f94 100644
```

```
--- a/net/core/net-sysfs.c
```

```
+++ b/net/core/net-sysfs.c
```

```
@@ -38,12 +38,13 @@ static ssize_t netdev_show(const struct class_device *cd, char *buf,
```

```

    ssize_t (*format)(const struct net_device *, char *))
{
    struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
    ssize_t ret = -EINVAL;
-
- read_lock(&dev_base_lock);
+
+ read_lock(&per_net(dev_base_lock, net));
    if (dev_isalive(dev))
        ret = (*format)(dev, buf);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

    return ret;
}
@@ -109,12 +110,13 @@ static ssize_t format_addr(char *buf, const unsigned char *addr, int
len)
static ssize_t show_address(struct class_device *cd, char *buf)
{
    struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
    ssize_t ret = -EINVAL;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
    if (dev_isalive(dev))
        ret = format_addr(buf, dev->dev_addr, dev->addr_len);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
    return ret;
}

@@ -158,13 +160,14 @@ static const char *operstates[] = {
static ssize_t show_operstate(struct class_device *cd, char *buf)
{
    const struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
    unsigned char operstate;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
    operstate = dev->operstate;
    if (!netif_running(dev))
        operstate = IF_OPER_DOWN;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

```

```

if (operstate >= ARRAY_SIZE(operstates))
    return -EINVAL; /* should not happen */
@@ -248,6 +251,7 @@ static ssize_t netstat_show(const struct class_device *cd, char *buf,
    unsigned long offset)
{
    struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
    struct net_device_stats *stats;
    ssize_t ret = -EINVAL;

@@ -255,13 +259,13 @@ static ssize_t netstat_show(const struct class_device *cd, char *buf,
    offset % sizeof(unsigned long) != 0)
    WARN_ON(1);

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
    if (dev_isalive(dev) && dev->get_stats &&
        (stats = (*dev->get_stats)(dev)))
        ret = sprintf(buf, fmt_ulong,
            *(unsigned long *)(((u8 *) stats) + offset));

- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
    return ret;
}

@@ -338,10 +342,11 @@ static ssize_t wireless_show(struct class_device *cd, char *buf,
    char *)
{
    struct net_device *dev = to_net_dev(cd);
+ net_t net = dev->nd_net;
    const struct iw_statistics *iw = NULL;
    ssize_t ret = -EINVAL;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, net));
    if (dev_isalive(dev)) {
        if(dev->wireless_handlers &&
            dev->wireless_handlers->get_wireless_stats)
@@ -349,7 +354,7 @@ static ssize_t wireless_show(struct class_device *cd, char *buf,
        if (iw != NULL)
            ret = (*format)(iw, buf);
    }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));

    return ret;
}

```



```

diff --git a/net/core/netpoll.c b/net/core/netpoll.c
index 823215d..f2c7c07 100644
--- a/net/core/netpoll.c
+++ b/net/core/netpoll.c
@@ -621,7 +621,7 @@ int netpoll_setup(struct netpoll *np)
    int err;

    if (np->dev_name)
-   ndev = dev_get_by_name(np->dev_name);
+   ndev = dev_get_by_name(init_net(), np->dev_name);
    if (!ndev) {
        printk(KERN_ERR "%s: %s doesn't exist, aborting.\n",
            np->name, np->dev_name);
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 7796b39..a415efb 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -1928,7 +1928,7 @@ static struct net_device *pktgen_setup_dev(struct pktgen_dev
 *pkt_dev)
    pkt_dev->odev = NULL;
    }

-   odev = dev_get_by_name(pkt_dev->ifname);
+   odev = dev_get_by_name(init_net(), pkt_dev->ifname);

    if (!odev) {
        printk("pktgen: no such netdevice: \"%s\"\n", pkt_dev->ifname);
diff --git a/net/core/rtnetlink.c b/net/core/rtnetlink.c
index 8f3dda8..5ac07a0 100644
--- a/net/core/rtnetlink.c
+++ b/net/core/rtnetlink.c
@@ -235,6 +235,7 @@ EXPORT_SYMBOL_GPL(rtnl_put_cacheinfo);

static void set_operstate(struct net_device *dev, unsigned char transition)
{
+   net_t net = dev->nd_net;
    unsigned char operstate = dev->operstate;

    switch(transition) {
@@ -253,9 +254,9 @@ static void set_operstate(struct net_device *dev, unsigned char
transition)
    };

    if (dev->operstate != operstate) {
-   write_lock_bh(&dev_base_lock);
+   write_lock_bh(&per_net(dev_base_lock, net));
        dev->operstate = operstate;
-   write_unlock_bh(&dev_base_lock);

```

```

+ write_unlock_bh(&per_net(dev_base_lock, net));
  netdev_state_change(dev);
}
}
@@ -389,12 +390,13 @@ nla_put_failure:

static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
{
+ net_t net = skb->sk->sk_net;
  int idx;
  int s_idx = cb->args[0];
  struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, net));
+ for (dev=per_net(dev_base, net), idx=0; dev; dev = dev->next, idx++) {
  if (idx < s_idx)
    continue;
  if (rtnl_fill_ifinfo(skb, dev, NULL, 0, RTM_NEWLINK,
@@ -402,7 +404,7 @@ static int rtnl_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
    cb->nlh->nmsg_seq, 0, NLM_F_MULTI) <= 0)
    break;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, net));
  cb->args[0] = idx;

  return skb->len;
@@ -420,6 +422,7 @@ static struct nla_policy ifla_policy[IFLA_MAX+1] __read_mostly = {

static int rtnl_setlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
{
+ net_t net = skb->sk->sk_net;
  struct ifinfomsg *ifm;
  struct net_device *dev;
  int err, send_addr_notify = 0, modified = 0;
@@ -438,9 +441,9 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
  err = -EINVAL;
  ifm = nlmsg_data(nlh);
  if (ifm->if_index >= 0)
- dev = dev_get_by_index(ifm->if_index);
+ dev = dev_get_by_index(net, ifm->if_index);
  else if (tb[IFLA_IFNAME])
- dev = dev_get_by_name(ifname);
+ dev = dev_get_by_name(net, ifname);
  else
    goto errout;

```

```
@@ -566,9 +569,9 @@ static int rtnl_setlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
    set_operstate(dev, nla_get_u8(tb[IFLA_OPERSTATE]));
```

```
    if (tb[IFLA_LINKMODE]) {
-   write_lock_bh(&dev_base_lock);
+   write_lock_bh(&per_net(dev_base_lock, net));
        dev->link_mode = nla_get_u8(tb[IFLA_LINKMODE]);
-   write_unlock_bh(&dev_base_lock);
+   write_unlock_bh(&per_net(dev_base_lock, net));
    }
```

```
    err = 0;
```

```
@@ -590,6 +593,7 @@ errout:
```

```
static int rtnl_getlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
{
```

```
+ net_t net = skb->sk->sk_net;
    struct ifinfomsg *ifm;
    struct nlattr *tb[IFLA_MAX+1];
    struct net_device *dev = NULL;
```

```
@@ -604,7 +608,7 @@ static int rtnl_getlink(struct sk_buff *skb, struct nlmsg_hdr *nlh, void *arg)
```

```
    ifm = nlmsg_data(nlh);
    if (ifm->ifi_index >= 0) {
-   dev = dev_get_by_index(ifm->ifi_index);
+   dev = dev_get_by_index(net, ifm->ifi_index);
        if (dev == NULL)
            return -ENODEV;
    } else
```

```
diff --git a/net/core/sock.c b/net/core/sock.c
```

```
index e42f7df..737838c 100644
```

```
--- a/net/core/sock.c
```

```
+++ b/net/core/sock.c
```

```
@@ -343,6 +343,7 @@ int sock_setsockopt(struct socket *sock, int level, int optname,
    char __user *optval, int optlen)
```

```
{
    struct sock *sk=sock->sk;
```

```
+ net_t net = sk->sk_net;
    struct sk_filter *filter;
    int val;
    int valbool;
```

```
@@ -579,7 +580,7 @@ set_rcvbuf:
```

```
    if (devname[0] == '\0') {
        sk->sk_bound_dev_if = 0;
    } else {
-   struct net_device *dev = dev_get_by_name(devname);
+   struct net_device *dev = dev_get_by_name(net, devname);
```

```

    if (!dev) {
        ret = -ENODEV;
        break;
    }
diff --git a/net/core/wireless.c b/net/core/wireless.c
index faa242f..d1418bf 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -672,7 +672,22 @@ static struct seq_operations wireless_seq_ops = {

    static int wireless_seq_open(struct inode *inode, struct file *file)
    {
- return seq_open(file, &wireless_seq_ops);
+ struct seq_file *seq;
+ int res;
+ res = seq_open(file, &wireless_seq_ops);
+ if (!res) {
+     seq = file->private_data;
+     seq->private = net_to_voidp(get_net(PROC_NET(inode)));
+ }
+ return res;
+ }
+
+ static int wireless_seq_release(struct inode *inode, struct file *file)
+ {
+ struct seq_file *seq = file->private_data;
+ net_t net = net_from_voidp(seq->private);
+ put_net(net);
+ return seq_release(inode, file);
+ }

    static struct file_operations wireless_seq_fops = {
@@ -680,17 +695,33 @@ static struct file_operations wireless_seq_fops = {
        .open    = wireless_seq_open,
        .read    = seq_read,
        .llseek  = seq_lseek,
-       .release = seq_release,
+       .release = wireless_seq_release,
    };

- int __init wireless_proc_init(void)
+ static int wireless_proc_net_init(net_t net)
    {
        /* Create /proc/net/wireless entry */
- if (!proc_net_fops_create(init_net(), "wireless", S_IRUGO, &wireless_seq_fops))
+ if (!proc_net_fops_create(net, "wireless", S_IRUGO, &wireless_seq_fops))
        return -ENOMEM;

        return 0;
    }

```

```

}
+
+static void wireless_proc_net_exit(net_t net)
+{
+ proc_net_remove(net, "wireless");
+}
+
+static struct pernet_operations wireless_proc_ops = {
+ .init = wireless_proc_net_init,
+ .exit = wireless_proc_net_exit,
+};
+
+int __init wireless_proc_init(void)
+{
+ return register_pernet_subsys(&wireless_proc_ops);
+}
+
#endif /* CONFIG_PROC_FS */

/***** IOCTL SUPPORT *****/
@@ -1066,7 +1097,7 @@ static inline int ioctl_private_call(struct net_device * dev,
 * (dev_ioctl() in net/core/dev.c).
 * Check the type of IOCTL and call the appropriate wrapper...
 */
-int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd)
+int wireless_process_ioctl(net_t net, struct ifreq *ifr, unsigned int cmd)
{
    struct net_device *dev;
    iw_handler handler;
@@ -1075,7 +1106,7 @@ int wireless_process_ioctl(struct ifreq *ifr, unsigned int cmd)
 * The copy_to/from_user() of ifr is also dealt with in there */

/* Make sure the device exist */
- if ((dev = __dev_get_by_name(ifr->ifr_name)) == NULL)
+ if ((dev = __dev_get_by_name(net, ifr->ifr_name)) == NULL)
    return -ENODEV;

/* A bunch of special cases, then the generic case...
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index b27b2ac..1cc502a 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -749,14 +749,14 @@ static int dn_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len)

    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) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
    if (!dev->dn_ptr)
        continue;
    if (dn_dev_islocal(dev, dn_saddr2dn(saddr)))
        break;
    }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
    if (dev == NULL)
        return -EADDRNOTAVAIL;
    }
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index dbaf001..c83c8d1 100644
--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -514,7 +514,7 @@ int dn_dev_ioctl(unsigned int cmd, void __user *arg)
    ifr->ifr_name[IFNAMSIZ-1] = 0;

#ifdef CONFIG_KMOD
- dev_load(ifr->ifr_name);
+ dev_load(init_net(), ifr->ifr_name);
#endif

    switch(cmd) {
@@ -532,7 +532,7 @@ int dn_dev_ioctl(unsigned int cmd, void __user *arg)

        rtnl_lock();

- if ((dev = __dev_get_by_name(ifr->ifr_name)) == NULL) {
+ if ((dev = __dev_get_by_name(init_net(), ifr->ifr_name)) == NULL) {
    ret = -ENODEV;
    goto done;
    }
@@ -630,7 +630,7 @@ static struct dn_dev *dn_dev_by_index(int ifindex)
{
    struct net_device *dev;
    struct dn_dev *dn_dev = NULL;
- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
    if (dev) {
        dn_dev = dev->dn_ptr;
        dev_put(dev);
@@ -695,7 +695,7 @@ static int dn_nl_newaddr(struct sk_buff *skb, struct nlmsg_hdr *nlh, void
*arg)
    return -EINVAL;

```

```

    ifm = nlmsg_data(nlh);
- if ((dev = __dev_get_by_index(ifm->ifa_index)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifm->ifa_index)) == NULL)
    return -ENODEV;

    if ((dn_db = dev->dn_ptr) == NULL) {
@@ -796,8 +796,8 @@ static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
    skip_ndevs = cb->args[0];
    skip_naddr = cb->args[1];

- read_lock(&dev_base_lock);
- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
    if (idx < skip_ndevs)
        continue;
    else if (idx > skip_ndevs) {
@@ -821,7 +821,7 @@ static int dn_nl_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
    }
}
done:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    cb->args[0] = idx;
    cb->args[1] = dn_idx;
@@ -862,9 +862,9 @@ int dn_dev_bind_default(__le16 *addr)
    dev = dn_dev_get_default();
last_chance:
    if (dev) {
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
        rv = dn_dev_get_first(dev, addr);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
        dev_put(dev);
        if (rv == 0 || dev == &per_net(loopback_dev, init_net()))
            return rv;
@@ -1296,7 +1296,7 @@ void dn_dev_devices_off(void)
    struct net_device *dev;

    rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next)
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next)
        dn_dev_down(dev);
    rtnl_unlock();

```

```

@@ -1307,7 +1307,7 @@ void dn_dev_devices_on(void)
    struct net_device *dev;

    rtnl_lock();
- for(dev = dev_base; dev; dev = dev->next) {
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
    if (dev->flags & IFF_UP)
        dn_dev_up(dev);
    }
@@ -1338,7 +1338,7 @@ static struct net_device *dn_dev_get_idx(struct seq_file *seq, loff_t
pos)
{
    struct net_device *dev;

- dev = dev_base;
+ dev = per_net(dev_base, init_net());
    if (dev && !dev->dn_ptr)
        dev = dn_dev_get_next(seq, dev);
    if (pos) {
@@ -1352,10 +1352,10 @@ static void *dn_dev_seq_start(struct seq_file *seq, loff_t *pos)
{
    if (*pos) {
        struct net_device *dev;
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
        dev = dn_dev_get_idx(seq, *pos - 1);
        if (dev == NULL)
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
        return dev;
    }
    return SEQ_START_TOKEN;
@@ -1371,7 +1371,7 @@ static void *dn_dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
} else {
    dev = dn_dev_get_next(seq, dev);
    if (dev == NULL)
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}
++*pos;
return dev;
@@ -1380,7 +1380,7 @@ static void *dn_dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
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(&per_net(dev_base_lock, init_net()));
}

```



```

}

static char *dn_type2asc(char type)
diff --git a/net/decnet/dn_fib.c b/net/decnet/dn_fib.c
index 1cf0101..cc2ab1f 100644
--- a/net/decnet/dn_fib.c
+++ b/net/decnet/dn_fib.c
@@ -212,7 +212,7 @@ static int dn_fib_check_nh(const struct rtmsg *r, struct dn_fib_info *fi,
struct
    return -EINVAL;
    if (dnet_addr_type(nh->nh_gw) != RTN_UNICAST)
    return -EINVAL;
- if ((dev = __dev_get_by_index(nh->nh_oif)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), nh->nh_oif)) == NULL)
    return -ENODEV;
    if (!(dev->flags&IFF_UP))
    return -ENETDOWN;
@@ -255,7 +255,7 @@ out:
    if (nh->nh_flags&(RTNH_F_PERVASIVE|RTNH_F_ONLINK))
    return -EINVAL;

- dev = __dev_get_by_index(nh->nh_oif);
+ dev = __dev_get_by_index(init_net(), nh->nh_oif);
    if (dev == NULL || dev->dn_ptr == NULL)
    return -ENODEV;
    if (!(dev->flags&IFF_UP))
@@ -352,7 +352,7 @@ struct dn_fib_info *dn_fib_create_info(const struct rtmsg *r, struct
dn_kern_rta
    if (nhs != 1 || nh->nh_gw)
    goto err_inval;
    nh->nh_scope = RT_SCOPE_NOWHERE;
- nh->nh_dev = dev_get_by_index(fi->fib_nh->nh_oif);
+ nh->nh_dev = dev_get_by_index(init_net(), fi->fib_nh->nh_oif);
    err = -ENODEV;
    if (nh->nh_dev == NULL)
    goto failure;
@@ -598,8 +598,8 @@ static void dn_fib_del_ifaddr(struct dn_ifaddr *ifa)
    ASSERT_RTNL();

    /* Scan device list */
- read_lock(&dev_base_lock);
- for(dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for(dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
    dn_db = dev->dn_ptr;
    if (dn_db == NULL)
    continue;
@@ -610,7 +610,7 @@ static void dn_fib_del_ifaddr(struct dn_ifaddr *ifa)

```

```

    }
    }
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    if (found_it == 0) {
        fib_magic(RTM_DELROUTE, RTN_LOCAL, ifa->ifa_local, 16, ifa);
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index b553cd4..9669e50 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -849,7 +849,7 @@ static __le16 dnet_select_source(const struct net_device *dev, __le16
daddr, int
    int best_match = 0;
    int ret;

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    for(ifa = dn_db->ifa_list; ifa; ifa = ifa->ifa_next) {
        if (ifa->ifa_scope > scope)
            continue;
@@ -863,7 +863,7 @@ static __le16 dnet_select_source(const struct net_device *dev, __le16
daddr, int
        if (best_match == 0)
            saddr = ifa->ifa_local;
    }
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    return saddr;
}
@@ -908,7 +908,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old

    /* If we have an output interface, verify its a DECnet device */
    if (oldflp->oif) {
- dev_out = dev_get_by_index(oldflp->oif);
+ dev_out = dev_get_by_index(init_net(), oldflp->oif);
        err = -ENODEV;
        if (dev_out && dev_out->dn_ptr == NULL) {
            dev_put(dev_out);
@@ -928,8 +928,8 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
            dev_put(dev_out);
            goto out;
        }
- read_lock(&dev_base_lock);

```

```

- for(dev_out = dev_base; dev_out; dev_out = dev_out->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for(dev_out = per_net(dev_base, init_net()); dev_out; dev_out = dev_out->next) {
    if (!dev_out->dn_ptr)
        continue;
    if (!dn_dev_islocal(dev_out, oldflp->fld_src))
@@ -940,7 +940,7 @@ static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi
*old
    continue;
    break;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
    if (dev_out == NULL)
        goto out;
    dev_hold(dev_out);
@@ -1554,7 +1554,7 @@ int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsg_hdr *nlh,
void *arg)

    if (fl.iif) {
        struct net_device *dev;
- if ((dev = dev_get_by_index(fl.iif)) == NULL) {
+ if ((dev = dev_get_by_index(init_net(), fl.iif)) == NULL) {
            kfree_skb(skb);
            return -ENODEV;
        }
diff --git a/net/decnet/sysctl_net_decnet.c b/net/decnet/sysctl_net_decnet.c
index 37fff9a..778b8a5 100644
--- a/net/decnet/sysctl_net_decnet.c
+++ b/net/decnet/sysctl_net_decnet.c
@@ -259,7 +259,7 @@ static int dn_def_dev_strategy(ctl_table *table, int __user *name, int
nlen,

    devname[newlen] = 0;

- dev = dev_get_by_name(devname);
+ dev = dev_get_by_name(init_net(), devname);
    if (dev == NULL)
        return -ENODEV;

@@ -299,7 +299,7 @@ static int dn_def_dev_handler(ctl_table *table, int write,
    devname[*lenp] = 0;
    strip_it(devname);

- dev = dev_get_by_name(devname);
+ dev = dev_get_by_name(init_net(), devname);
    if (dev == NULL)
        return -ENODEV;

```

```

diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index cbf87f4..cd5336b 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -663,7 +663,7 @@ static int ec_dev_ioctl(struct socket *sock, unsigned int cmd, void __user
*arg)
    if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
        return -EFAULT;

- if ((dev = dev_get_by_name(ifr.ifr_name)) == NULL)
+ if ((dev = dev_get_by_name(init_net(), ifr.ifr_name)) == NULL)
    return -ENODEV;

    sec = (struct sockaddr_ec *)&ifr.ifr_addr;
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 0d23fb2..39d2ac4 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -983,7 +983,7 @@ static int arp_req_set(struct arpreq *r, struct net_device * dev)
    if (mask && mask != htonl(0xFFFFFFFF))
        return -EINVAL;
    if (!dev && (r->arp_flags & ATF_COM)) {
- dev = dev_getbyhwaddr(r->arp_ha.sa_family, r->arp_ha.sa_data);
+ dev = dev_getbyhwaddr(init_net(), r->arp_ha.sa_family, r->arp_ha.sa_data);
    if (!dev)
        return -ENODEV;
    }
@@ -1170,7 +1170,7 @@ int arp_ioctl(unsigned int cmd, void __user *arg)
    rtnl_lock();
    if (r.arp_dev[0]) {
        err = -ENODEV;
- if ((dev = __dev_get_by_name(r.arp_dev)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), r.arp_dev)) == NULL)
        goto out;

    /* Mmmm... It is wrong... ARPHRD_NETROM==0 */
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 201442c..b0d12ec 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -419,11 +419,11 @@ struct in_device *inetdev_by_index(int ifindex)
{
    struct net_device *dev;
    struct in_device *in_dev = NULL;
- read_lock(&dev_base_lock);
- dev = __dev_get_by_index(ifindex);
+ read_lock(&per_net(dev_base_lock, init_net()));

```

```

+ dev = __dev_get_by_index(init_net(), ifindex);
  if (dev)
    in_dev = in_dev_get(dev);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  return in_dev;
}

@@ -504,7 +504,7 @@ static struct in_ifaddr *rtm_to_ifaddr(struct nlmsghdr *nlh)
  if (ifm->ifa_prefixlen > 32 || tb[IFA_LOCAL] == NULL)
    goto errout;

- dev = __dev_get_by_index(ifm->ifa_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifa_index);
  if (dev == NULL) {
    err = -ENODEV;
    goto errout;
@@ -627,7 +627,7 @@ int devinet_ioctl(unsigned int cmd, void __user *arg)
  *colon = 0;

#ifdef CONFIG_KMOD
- dev_load(ifr.ifr_name);
+ dev_load(init_net(), ifr.ifr_name);
#endif

  switch(cmd) {
@@ -668,7 +668,7 @@ int devinet_ioctl(unsigned int cmd, void __user *arg)
  rtnl_lock();

  ret = -ENODEV;
- if ((dev = __dev_get_by_name(ifr.ifr_name)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifr.ifr_name)) == NULL)
    goto done;

  if (colon)
@@ -906,9 +906,9 @@ no_in_dev:
  in this case. It is importnat that lo is the first interface
  in dev_base list.
  */
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
  if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
    continue;

@@ -921,7 +921,7 @@ no_in_dev:

```

```

    } endfor_ifa(in_dev);
}
out_unlock_both:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  rcu_read_unlock();
out:
  return addr;
@@ -985,9 +985,9 @@ __be32 inet_confirm_addr(const struct net_device *dev, __be32 dst,
__be32 local,
  return addr;
}

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
  rcu_read_lock();
- for (dev = dev_base; dev; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
  if ((in_dev = __in_dev_get_rcu(dev))) {
    addr = confirm_addr_indev(in_dev, dst, local, scope);
    if (addr)
@@ -995,7 +995,7 @@ __be32 inet_confirm_addr(const struct net_device *dev, __be32 dst,
__be32 local,
  }
}
  rcu_read_unlock();
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  return addr;
}
@@ -1180,8 +1180,8 @@ static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
  int s_ip_idx, 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++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
  if (idx < s_idx)
    continue;
  if (idx > s_idx)
@@ -1207,7 +1207,7 @@ static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)
}

done:

```

```

- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
  cb->args[0] = idx;
  cb->args[1] = ip_idx;

@@ -1258,8 +1258,8 @@ void inet_forward_change(void)
  ipv4_devconf.accept_redirects = !on;
  ipv4_devconf_dflt.forwarding = on;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
  struct in_device *in_dev;
  rcu_read_lock();
  in_dev = __in_dev_get_rcu(dev);
@@ -1267,7 +1267,7 @@ void inet_forward_change(void)
  in_dev->cnf.forwarding = on;
  rcu_read_unlock();
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

  rt_cache_flush(0);
}
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d1859ff..449f42d 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -337,7 +337,7 @@ static int rentry_to_fib_config(int cmd, struct rentry *rt,
  colon = strchr(devname, ':');
  if (colon)
    *colon = 0;
- dev = __dev_get_by_name(devname);
+ dev = __dev_get_by_name(init_net(), devname);
  if (!dev)
    return -ENODEV;
  cfg->fc_oif = dev->oif;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index e63b8a9..76218e5 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -530,7 +530,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
  return -EINVAL;
  if (inet_addr_type(nh->nh_gw) != RTN_UNICAST)
    return -EINVAL;
- if ((dev = __dev_get_by_index(nh->nh_oif)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), nh->nh_oif)) == NULL)

```

```

    return -ENODEV;
    if (!(dev->flags&IFF_UP))
        return -ENETDOWN;
@@ -807,7 +807,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    if (nhs != 1 || nh->nh_gw)
        goto err_inval;
    nh->nh_scope = RT_SCOPE_NOWHERE;
- nh->nh_dev = dev_get_by_index(fi->fib_nh->nh_oif);
+ nh->nh_dev = dev_get_by_index(init_net(), fi->fib_nh->nh_oif);
    err = -ENODEV;
    if (nh->nh_dev == NULL)
        goto failure;
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 92624cc..0455935 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -2262,7 +2262,7 @@ static inline struct ip_mc_list *igmp_mc_get_first(struct seq_file *seq)
    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;
+ for (state->dev = per_net(dev_base, init_net()), state->in_dev = NULL;
        state->dev;
        state->dev = state->dev->next) {
    struct in_device *in_dev;
@@ -2315,7 +2315,7 @@ static struct ip_mc_list *igmp_mc_get_idx(struct seq_file *seq, loff_t
pos)

static void *igmp_mc_seq_start(struct seq_file *seq, loff_t *pos)
{
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    return *pos ? igmp_mc_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
}

@@ -2339,7 +2339,7 @@ static void igmp_mc_seq_stop(struct seq_file *seq, void *v)
    state->in_dev = NULL;
}
state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int igmp_mc_seq_show(struct seq_file *seq, void *v)
@@ -2424,7 +2424,7 @@ static inline struct ip_sf_list *igmp_mcf_get_first(struct seq_file *seq)
    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;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL, state->im = NULL;
    state->dev;
    state->dev = state->dev->next) {
    struct in_device *idev;
@@ -2493,7 +2493,7 @@ static struct ip_sf_list *igmp_mcf_get_idx(struct seq_file *seq, loff_t
pos)

static void *igmp_mcf_seq_start(struct seq_file *seq, loff_t *pos)
{
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    return *pos ? igmp_mcf_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
}

@@ -2521,7 +2521,7 @@ static void igmp_mcf_seq_stop(struct seq_file *seq, void *v)
    state->idev = NULL;
}
    state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int igmp_mcf_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 8ce00d3..078eed3 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -292,7 +292,7 @@ static void ip_expire(unsigned long arg)
    if ((qp->last_in&FIRST_IN) && qp->fragments != NULL) {
        struct sk_buff *head = qp->fragments;
        /* Send an ICMP "Fragment Reassembly Timeout" message. */
-    if ((head->dev = dev_get_by_index(qp->iif)) != NULL) {
+    if ((head->dev = dev_get_by_index(init_net(), qp->iif)) != NULL) {
        icmp_send(head, ICMP_TIME_EXCEEDED, ICMP_EXC_FRAGTIME, 0);
        dev_put(head->dev);
    }
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 476cb60..a21688c 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -266,7 +266,7 @@ static struct ip_tunnel *ipgre_tunnel_locate(struct ip_tunnel_parm
*parms, int
    int i;
    for (i=1; i<100; i++) {
        sprintf(name, "gre%d", i);
-    if (__dev_get_by_name(name) == NULL)
+    if (__dev_get_by_name(init_net(), name) == NULL)

```

```

        break;
    }
    if (i==100)
@@ -1196,7 +1196,7 @@ static int ipgre_tunnel_init(struct net_device *dev)
    }

    if (!tdev && tunnel->parms.link)
-   tdev = __dev_get_by_index(tunnel->parms.link);
+   tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

    if (tdev) {
        hlen = tdev->hard_header_len;
diff --git a/net/ipv4/ip_sockglue.c b/net/ipv4/ip_sockglue.c
index 57d4bae..95094c5 100644
--- a/net/ipv4/ip_sockglue.c
+++ b/net/ipv4/ip_sockglue.c
@@ -597,7 +597,7 @@ static int do_ip_setsockopt(struct sock *sk, int level,
    dev_put(dev);
    }
    } else
-   dev = __dev_get_by_index(mreq.imr_ifindex);
+   dev = __dev_get_by_index(init_net(), mreq.imr_ifindex);

    err = -EADDRNOTAVAIL;
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index ee77938..2606b8c 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -195,7 +195,7 @@ static int __init ic_open_devs(void)
    if (dev_change_flags(lo, lo->flags | IFF_UP) < 0)
        printk(KERN_ERR "IP-Config: Failed to open %s\n", lo->name);

-   for (dev = dev_base; dev; dev = dev->next) {
+   for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
        if (dev == lo)
            continue;
        if (user_dev_name[0] ? !strcmp(dev->name, user_dev_name) :
diff --git a/net/ipv4/ipip.c b/net/ipv4/ipip.c
index 9d719d6..4e75691 100644
--- a/net/ipv4/ipip.c
+++ b/net/ipv4/ipip.c
@@ -232,7 +232,7 @@ static struct ip_tunnel * ipip_tunnel_locate(struct ip_tunnel_parm *parms,
int c
    int i;
    for (i=1; i<100; i++) {
        sprintf(name, "tun%d", i);
-   if (__dev_get_by_name(name) == NULL)

```

```

+ if (__dev_get_by_name(init_net(), name) == NULL)
    break;
}
if (i==100)
@@ -827,7 +827,7 @@ static int ipip_tunnel_init(struct net_device *dev)
}

    if (!tdev && tunnel->parms.link)
- tdev = __dev_get_by_index(tunnel->parms.link);
+ tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

    if (tdev) {
        dev->hard_header_len = tdev->hard_header_len + sizeof(struct iphdr);
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 9afaa13..d2e7e55 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -125,7 +125,7 @@ struct net_device *ipmr_new_tunnel(struct vifctl *v)
{
    struct net_device *dev;

- dev = __dev_get_by_name("tunl0");
+ dev = __dev_get_by_name(init_net(), "tunl0");

    if (dev) {
        int err;
@@ -149,7 +149,7 @@ struct net_device *ipmr_new_tunnel(struct vifctl *v)

    dev = NULL;

- if (err == 0 && (dev = __dev_get_by_name(p.name)) != NULL) {
+ if (err == 0 && (dev = __dev_get_by_name(init_net(), p.name)) != NULL) {
    dev->flags |= IFF_MULTICAST;

    in_dev = __in_dev_get_rtnl(dev);
diff --git a/net/ipv4/ipvs/ip_vs_sync.c b/net/ipv4/ipvs/ip_vs_sync.c
index 7ea2d98..fd6d1ca 100644
--- a/net/ipv4/ipvs/ip_vs_sync.c
+++ b/net/ipv4/ipvs/ip_vs_sync.c
@@ -382,7 +382,7 @@ static int set_mcast_if(struct sock *sk, char *ifname)
    struct net_device *dev;
    struct inet_sock *inet = inet_sk(sk);

- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
    return -ENODEV;

    if (sk->sk_bound_dev_if && dev->ifindex != sk->sk_bound_dev_if)

```

```

@@ -407,7 +407,7 @@ static int set_sync_mesg_maxlen(int sync_state)
    int num;

    if (sync_state == IP_VS_STATE_MASTER) {
- if ((dev = __dev_get_by_name(ip_vs_master_mcast_ifn)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ip_vs_master_mcast_ifn)) == NULL)
        return -ENODEV;

    num = (dev->mtu - sizeof(struct iphdr) -
@@ -418,7 +418,7 @@ static int set_sync_mesg_maxlen(int sync_state)
    IP_VS_DBG(7, "setting the maximum length of sync sending "
        "message %d.\n", sync_send_mesg_maxlen);
    } else if (sync_state == IP_VS_STATE_BACKUP) {
- if ((dev = __dev_get_by_name(ip_vs_backup_mcast_ifn)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ip_vs_backup_mcast_ifn)) == NULL)
        return -ENODEV;

    sync_rcv_mesg_maxlen = dev->mtu -
@@ -446,7 +446,7 @@ join_mcast_group(struct sock *sk, struct in_addr *addr, char *ifname)
    memset(&mreq, 0, sizeof(mreq));
    memcpy(&mreq.imr_multiaddr, addr, sizeof(struct in_addr));

- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
        return -ENODEV;
    if (sk->sk_bound_dev_if && dev->ifindex != sk->sk_bound_dev_if)
        return -EINVAL;
@@ -467,7 +467,7 @@ static int bind_mcastif_addr(struct socket *sock, char *ifname)
    __be32 addr;
    struct sockaddr_in sin;

- if ((dev = __dev_get_by_name(ifname)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), ifname)) == NULL)
        return -ENODEV;

    addr = inet_select_addr(dev, 0, RT_SCOPE_UNIVERSE);
diff --git a/net/ipv4/netfilter/ipt_CLUSTERIP.c b/net/ipv4/netfilter/ipt_CLUSTERIP.c
index 779e2c6..02003ff 100644
--- a/net/ipv4/netfilter/ipt_CLUSTERIP.c
+++ b/net/ipv4/netfilter/ipt_CLUSTERIP.c
@@ -430,7 +430,7 @@ checkentry(const char *tablename,
    return 0;
    }

- dev = dev_get_by_name(e->ip.iniface);
+ dev = dev_get_by_name(init_net(), e->ip.iniface);
    if (!dev) {
        printk(KERN_WARNING "CLUSTERIP: no such interface %s\n", e->ip.iniface);

```

```

    return 0;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index d23a0d7..509bfb1 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2436,7 +2436,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)

    if (oldflp->oif) {
-   dev_out = dev_get_by_index(oldflp->oif);
+   dev_out = dev_get_by_index(init_net(), oldflp->oif);
    err = -ENODEV;
    if (dev_out == NULL)
        goto out;
@@ -2761,7 +2761,7 @@ int inet_rtm_getroute(struct sk_buff *in_skb, struct nlmsg_hdr* nlh, void
*arg)
    if (iif) {
        struct net_device *dev;

-   dev = __dev_get_by_index(iif);
+   dev = __dev_get_by_index(init_net(), iif);
    if (dev == NULL) {
        err = -ENODEV;
        goto errout_free;
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index c9fa27a..7afe698 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -477,8 +477,8 @@ static void addrconf_forward_change(void)
    struct net_device *dev;
    struct inet6_dev *idev;

-   read_lock(&dev_base_lock);
-   for (dev=dev_base; dev; dev=dev->next) {
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for (dev=per_net(dev_base, init_net()); dev; dev=dev->next) {
        rcu_read_lock();
        idev = __in6_dev_get(dev);
        if (idev) {
@@ -489,7 +489,7 @@ static void addrconf_forward_change(void)
    }
    rcu_read_unlock();
    }
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
    }
#endif

```

```

@@ -919,10 +919,10 @@ int ipv6_dev_get_saddr(struct net_device *daddr_dev,

    memset(&hiscore, 0, sizeof(hiscore));

- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    rcu_read_lock();

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

@@ -1151,7 +1151,7 @@ record_it:
    read_unlock_bh(&idev->lock);
}
rcu_read_unlock();
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    if (!ifa_result)
        return -EADDRNOTAVAIL;
@@ -1853,7 +1853,7 @@ int addrconf_set_dstaddr(void __user *arg)
    if (copy_from_user(&ireq, arg, sizeof(struct in6_ifreq)))
        goto err_exit;

- dev = __dev_get_by_index(ireq.ifr6_ifindex);
+ dev = __dev_get_by_index(init_net(), ireq.ifr6_ifindex);

    err = -ENODEV;
    if (dev == NULL)
@@ -1884,7 +1884,7 @@ int addrconf_set_dstaddr(void __user *arg)

    if (err == 0) {
        err = -ENOBUFS;
- if ((dev = __dev_get_by_name(p.name)) == NULL)
+ if ((dev = __dev_get_by_name(init_net(), p.name)) == NULL)
        goto err_exit;
        err = dev_open(dev);
    }
@@ -1913,7 +1913,7 @@ static int inet6_addr_add(int ifindex, struct in6_addr *pfx, int plen,
    if (!valid_lft || preferred_lft > valid_lft)
        return -EINVAL;

- if ((dev = __dev_get_by_index(ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifindex)) == NULL)
    return -ENODEV;

```

```

    if ((idev = addrconf_add_dev(dev)) == NULL)
@@ -1956,7 +1956,7 @@ static int inet6_addr_del(int ifindex, struct in6_addr *pfx, int plen)
    struct inet6_dev *idev;
    struct net_device *dev;

- if ((dev = __dev_get_by_index(ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), ifindex)) == NULL)
    return -ENODEV;

    if ((idev = __in6_dev_get(dev)) == NULL)
@@ -2051,7 +2051,7 @@ static void sit_add_v4_addrs(struct inet6_dev *idev)
    return;
}

- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    struct in_device *in_dev = __in_dev_get_rtnl(dev);
    if (in_dev && (dev->flags & IFF_UP)) {
        struct in_ifaddr *ifa;
@@ -2198,12 +2198,12 @@ static void ip6_tnl_add_linklocal(struct inet6_dev *idev)

    /* first try to inherit the link-local address from the link device */
    if (idev->dev->iflink &&
- (link_dev = __dev_get_by_index(idev->dev->iflink))) {
+ (link_dev = __dev_get_by_index(init_net(), idev->dev->iflink))) {
        if (!ipv6_inherit_linklocal(idev, link_dev))
            return;
    }
    /* then try to inherit it from any device */
- for (link_dev = dev_base; link_dev; link_dev = link_dev->next) {
+ for (link_dev = per_net(dev_base, init_net()); link_dev; link_dev = link_dev->next) {
        if (!ipv6_inherit_linklocal(idev, link_dev))
            return;
    }
@@ -3032,7 +3032,7 @@ inet6_rtm_newaddr(struct sk_buff *skb, struct nlmsg_hdr *nlh, void
*arg)
    valid_lft = INFINITY_LIFE_TIME;
}

- dev = __dev_get_by_index(ifm->ifa_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifa_index);
    if (dev == NULL)
        return -ENODEV;

@@ -3208,9 +3208,9 @@ static int inet6_dump_addr(struct sk_buff *skb, struct netlink_callback
*cb,

```

```

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

- for (dev = dev_base, idx = 0; dev; dev = dev->next, idx++) {
+ for (dev = per_net(dev_base, init_net()), idx = 0; dev; dev = dev->next, idx++) {
    if (idx < s_idx)
        continue;
    if (idx > s_idx)
@@ -3270,7 +3270,7 @@ done:
    read_unlock_bh(&idev->lock);
    in6_dev_put(idev);
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
cb->args[0] = idx;
cb->args[1] = ip_idx;
return skb->len;
@@ -3318,7 +3318,7 @@ static int inet6_rtm_getaddr(struct sk_buff *in_skb, struct nlmsg_hdr*
nlh,

    ifm = nlmsg_data(nlh);
    if (ifm->ifa_index)
- dev = __dev_get_by_index(ifm->ifa_index);
+ dev = __dev_get_by_index(init_net(), ifm->ifa_index);

    if ((ifa = ipv6_get_ifaddr(addr, dev, 1)) == NULL) {
        err = -EADDRNOTAVAIL;
@@ -3477,8 +3477,8 @@ static int inet6_dump_ifinfo(struct sk_buff *skb, struct netlink_callback
*cb)
    struct net_device *dev;
    struct inet6_dev *idev;

- read_lock(&dev_base_lock);
- for (dev=dev_base, idx=0; dev; dev = dev->next, idx++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev=per_net(dev_base, init_net()), idx=0; dev; dev = dev->next, idx++) {
    if (idx < s_idx)
        continue;
    if ((idev = in6_dev_get(dev)) == NULL)
@@ -3489,7 +3489,7 @@ static int inet6_dump_ifinfo(struct sk_buff *skb, struct netlink_callback
*cb)
    if (err <= 0)
        break;
}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

```



```

cb->args[0] = idx;

return skb->len;
@@ -4116,7 +4116,7 @@ void __exit addrconf_cleanup(void)
 * clean dev list.
 */

- for (dev=dev_base; dev; dev=dev->next) {
+ for (dev=per_net(dev_base, init_net()); dev; dev=dev->next) {
    if ((idev = __in6_dev_get(dev)) == NULL)
        continue;
    addrconf_ifdown(dev, 1);
diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index 00bd55a..84f0623 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -302,7 +302,7 @@ int inet6_bind(struct socket *sock, struct sockaddr *uaddr, int addr_len)
    err = -EINVAL;
    goto out;
}
- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
if (!dev) {
    err = -ENODEV;
    goto out;
diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index c42bad9..906ef0f 100644
--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -113,10 +113,10 @@ int ipv6_sock_ac_join(struct sock *sk, int ifindex, struct in6_addr *addr)
} else {
    /* router, no matching interface: just pick one */

- dev = dev_get_by_flags(IFF_UP, IFF_UP|IFF_LOOPBACK);
+ dev = dev_get_by_flags(init_net(), IFF_UP, IFF_UP|IFF_LOOPBACK);
}
} else
- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);

if (dev == NULL) {
    err = -ENODEV;
@@ -197,7 +197,7 @@ int ipv6_sock_ac_drop(struct sock *sk, int ifindex, struct in6_addr *addr)

write_unlock_bh(&ipv6_sk_ac_lock);

- dev = dev_get_by_index(pac->acl_ifindex);
+ dev = dev_get_by_index(init_net(), pac->acl_ifindex);

```

```

if (dev) {
    ipv6_dev_ac_dec(dev, &pac->acl_addr);
    dev_put(dev);
@@ -225,7 +225,7 @@ void ipv6_sock_ac_close(struct sock *sk)
    if (pac->acl_ifindex != prev_index) {
        if (dev)
            dev_put(dev);
-   dev = dev_get_by_index(pac->acl_ifindex);
+   dev = dev_get_by_index(init_net(), pac->acl_ifindex);
        prev_index = pac->acl_ifindex;
    }
    if (dev)
@@ -427,11 +427,11 @@ int ipv6_chk_acast_addr(struct net_device *dev, struct in6_addr *addr)
{
    if (dev)
        return ipv6_chk_acast_dev(dev, addr);
-   read_lock(&dev_base_lock);
-   for (dev=dev_base; dev; dev=dev->next)
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for (dev=per_net(dev_base, init_net()); dev; dev=dev->next)
        if (ipv6_chk_acast_dev(dev, addr))
            break;
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));
    return dev != 0;
}

@@ -449,7 +449,7 @@ static inline struct ifacaddr6 *ac6_get_first(struct seq_file *seq)
    struct ifacaddr6 *im = NULL;
    struct ac6_iter_state *state = ac6_seq_private(seq);

-   for (state->dev = dev_base, state->idev = NULL;
+   for (state->dev = per_net(dev_base, init_net()), state->idev = NULL;
        state->dev;
        state->dev = state->dev->next) {
        struct inet6_dev *idev;
@@ -502,7 +502,7 @@ static struct ifacaddr6 *ac6_get_idx(struct seq_file *seq, loff_t pos)

static void *ac6_seq_start(struct seq_file *seq, loff_t *pos)
{
-   read_lock(&dev_base_lock);
+   read_lock(&per_net(dev_base_lock, init_net()));
    return ac6_get_idx(seq, *pos);
}

@@ -521,7 +521,7 @@ static void ac6_seq_stop(struct seq_file *seq, void *v)
    read_unlock_bh(&state->idev->lock);
    in6_dev_put(state->idev);

```

```

}
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int ac6_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv6/datagram.c b/net/ipv6/datagram.c
index 5c94fea..c5dfb4e 100644
--- a/net/ipv6/datagram.c
+++ b/net/ipv6/datagram.c
@@ -536,7 +536,7 @@ int datagram_send_ctl(struct msghdr *msg, struct flowi *fl,
    if (!src_info->ip6_ifindex)
        return -EINVAL;
    else {
-   dev = dev_get_by_index(src_info->ip6_ifindex);
+   dev = dev_get_by_index(init_net(), src_info->ip6_ifindex);
        if (!dev)
            return -ENODEV;
    }
}
diff --git a/net/ipv6/ip6_tunnel.c b/net/ipv6/ip6_tunnel.c
index 8d91834..9006cbf 100644
--- a/net/ipv6/ip6_tunnel.c
+++ b/net/ipv6/ip6_tunnel.c
@@ -231,7 +231,7 @@ static struct ip6_tnl *ip6_tnl_create(struct ip6_tnl_parm *p)
    int i;
    for (i = 1; i < IP6_TNL_MAX; i++) {
        sprintf(name, "ip6tnl%d", i);
-   if (__dev_get_by_name(name) == NULL)
+   if (__dev_get_by_name(init_net(), name) == NULL)
        break;
    }
    if (i == IP6_TNL_MAX)
@@ -505,7 +505,7 @@ static inline int ip6_tnl_rcv_ctl(struct ip6_tnl *t)
    struct net_device *ldev = NULL;

    if (p->link)
-   ldev = dev_get_by_index(p->link);
+   ldev = dev_get_by_index(init_net(), p->link);

    if ((ipv6_addr_is_multicast(&p->laddr) ||
        likely(ipv6_chk_addr(&p->laddr, ldev, 0))) &&
@@ -621,7 +621,7 @@ static inline int ip6_tnl_xmit_ctl(struct ip6_tnl *t)
    struct net_device *ldev = NULL;

    if (p->link)
-   ldev = dev_get_by_index(p->link);
+   ldev = dev_get_by_index(init_net(), p->link);

```

```

    if (unlikely(!ipv6_chk_addr(&p->laddr, ldev, 0)))
        printk(KERN_WARNING
diff --git a/net/ipv6/ipv6_sockglue.c b/net/ipv6/ipv6_sockglue.c
index 352690e..65d9b82 100644
--- a/net/ipv6/ipv6_sockglue.c
+++ b/net/ipv6/ipv6_sockglue.c
@@ -548,7 +548,7 @@ done:
     if (sk->sk_bound_dev_if && sk->sk_bound_dev_if != val)
        goto e_inval;

-   if (__dev_get_by_index(val) == NULL) {
+   if (__dev_get_by_index(init_net(), val) == NULL) {
        retv = -ENODEV;
        break;
    }
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index 2759571..da45f46 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -215,7 +215,7 @@ int ipv6_sock_mc_join(struct sock *sk, int ifindex, struct in6_addr *addr)
    dst_release(&rt->u.dst);
    }
    } else
-   dev = dev_get_by_index(ifindex);
+   dev = dev_get_by_index(init_net(), ifindex);

    if (dev == NULL) {
        sock_kfree_s(sk, mc_lst, sizeof(*mc_lst));
@@ -266,7 +266,7 @@ int ipv6_sock_mc_drop(struct sock *sk, int ifindex, struct in6_addr *addr)
    *lnc = mc_lst->next;
    write_unlock_bh(&ipv6_sk_mc_lock);

-   if ((dev = dev_get_by_index(mc_lst->ifindex)) != NULL) {
+   if ((dev = dev_get_by_index(init_net(), mc_lst->ifindex)) != NULL) {
        struct inet6_dev *idev = in6_dev_get(dev);

        (void) ip6_mc_leave_src(sk, mc_lst, idev);
@@ -301,7 +301,7 @@ static struct inet6_dev *ip6_mc_find_dev(struct in6_addr *group, int
ifindex)
    dst_release(&rt->u.dst);
    }
    } else
-   dev = dev_get_by_index(ifindex);
+   dev = dev_get_by_index(init_net(), ifindex);

    if (!dev)
        return NULL;
@@ -332,7 +332,7 @@ void ipv6_sock_mc_close(struct sock *sk)

```

```

np->ipv6_mc_list = mc_lst->next;
write_unlock_bh(&ipv6_sk_mc_lock);

- dev = dev_get_by_index(mc_lst->ifindex);
+ dev = dev_get_by_index(init_net(), mc_lst->ifindex);
  if (dev) {
    struct inet6_dev *idev = in6_dev_get(dev);

@@ -2334,7 +2334,7 @@ static inline struct ifmcaddr6 *igmp6_mc_get_first(struct seq_file *seq)
    struct ifmcaddr6 *im = NULL;
    struct igmp6_mc_iter_state *state = igmp6_mc_seq_private(seq);

- for (state->dev = dev_base, state->idev = NULL;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL;
    state->dev;
    state->dev = state->dev->next) {
    struct inet6_dev *idev;
@@ -2388,7 +2388,7 @@ static struct ifmcaddr6 *igmp6_mc_get_idx(struct seq_file *seq, loff_t
pos)

static void *igmp6_mc_seq_start(struct seq_file *seq, loff_t *pos)
{
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    return igmp6_mc_get_idx(seq, *pos);
}

@@ -2409,7 +2409,7 @@ static void igmp6_mc_seq_stop(struct seq_file *seq, void *v)
    state->idev = NULL;
}
state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int igmp6_mc_seq_show(struct seq_file *seq, void *v)
@@ -2478,7 +2478,7 @@ static inline struct ip6_sf_list *igmp6_mcf_get_first(struct seq_file *seq)
    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;
+ for (state->dev = per_net(dev_base, init_net()), state->idev = NULL, state->im = NULL;
    state->dev;
    state->dev = state->dev->next) {
    struct inet6_dev *idev;
@@ -2547,7 +2547,7 @@ static struct ip6_sf_list *igmp6_mcf_get_idx(struct seq_file *seq, loff_t
pos)

```

```

static void *igmp6_mcf_seq_start(struct seq_file *seq, loff_t *pos)
{
- read_lock(&dev_base_lock);
+ read_lock(&per_net(dev_base_lock, init_net()));
    return *pos ? igmp6_mcf_get_idx(seq, *pos - 1) : SEQ_START_TOKEN;
}

@@ -2575,7 +2575,7 @@ static void igmp6_mcf_seq_stop(struct seq_file *seq, void *v)
    state->idev = NULL;
}
    state->dev = NULL;
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
}

static int igmp6_mcf_seq_show(struct seq_file *seq, void *v)
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index 2e1825c..5a68e2d 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -256,7 +256,7 @@ static int rawv6_bind(struct sock *sk, struct sockaddr *uaddr, int
    addr_len)
    if (!sk->sk_bound_dev_if)
        goto out;

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    if (!dev) {
        err = -ENODEV;
        goto out;
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 6f9a904..0441380 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -301,7 +301,7 @@ static void ip6_frag_expire(unsigned long data)

    fq_kill(fq);

- dev = dev_get_by_index(fq->iif);
+ dev = dev_get_by_index(init_net(), fq->iif);
    if (!dev)
        goto out;

diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 6805c39..4519006 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -1045,7 +1045,7 @@ int ip6_route_add(struct fib6_config *cfg)

```

```

#endif
if (cfg->fc_ifindex) {
    err = -ENODEV;
- dev = dev_get_by_index(cfg->fc_ifindex);
+ dev = dev_get_by_index(init_net(), cfg->fc_ifindex);
    if (!dev)
        goto out;
    idev = in6_dev_get(dev);
@@ -2168,7 +2168,7 @@ int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsg_hdr* nlh,
void *arg)

    if (iif) {
        struct net_device *dev;
- dev = __dev_get_by_index(iif);
+ dev = __dev_get_by_index(init_net(), iif);
        if (!dev) {
            err = -ENODEV;
            goto errout;
diff --git a/net/ipv6/sit.c b/net/ipv6/sit.c
index 77b7b09..8f97692 100644
--- a/net/ipv6/sit.c
+++ b/net/ipv6/sit.c
@@ -173,7 +173,7 @@ static struct ip_tunnel * ipip6_tunnel_locate(struct ip_tunnel_parm
*parms, int
    int i;
    for (i=1; i<100; i++) {
        sprintf(name, "sit%d", i);
- if (__dev_get_by_name(name) == NULL)
+ if (__dev_get_by_name(init_net(), name) == NULL)
            break;
    }
    if (i==100)
@@ -759,7 +759,7 @@ static int ipip6_tunnel_init(struct net_device *dev)
}

    if (!tdev && tunnel->parms.link)
- tdev = __dev_get_by_index(tunnel->parms.link);
+ tdev = __dev_get_by_index(init_net(), tunnel->parms.link);

    if (tdev) {
        dev->hard_header_len = tdev->hard_header_len + sizeof(struct iphdr);
diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index f2674fe..0e63fd2 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
@@ -987,7 +987,7 @@ static int ipxif_create(struct ipx_interface_definition *idef)
    if (intrfc)
        ipxif_put(intrfc);

```

```

- dev = dev_get_by_name(idef->ipx_device);
+ dev = dev_get_by_name(init_net(), idef->ipx_device);
  rc = -ENODEV;
  if (!dev)
    goto out;
@@ -1095,7 +1095,7 @@ static int ipxif_delete(struct ipx_interface_definition *idef)
  if (!dlink_type)
    goto out;

- dev = __dev_get_by_name(idef->ipx_device);
+ dev = __dev_get_by_name(init_net(), idef->ipx_device);
  rc = -ENODEV;
  if (!dev)
    goto out;
@@ -1190,7 +1190,7 @@ static int ipxif_ioctl(unsigned int cmd, void __user *arg)
  if (copy_from_user(&ifr, arg, sizeof(ifr)))
    break;
  sipx = (struct sockaddr_ipx *)&ifr.ifr_addr;
- dev = __dev_get_by_name(ifr.ifr_name);
+ dev = __dev_get_by_name(init_net(), ifr.ifr_name);
  rc = -ENODEV;
  if (!dev)
    break;
diff --git a/net/llc/af_llc.c b/net/llc/af_llc.c
index 6bc0fff..ac380ac 100644
--- a/net/llc/af_llc.c
+++ b/net/llc/af_llc.c
@@ -252,7 +252,7 @@ static int llc_ui_autobind(struct socket *sock, struct sockaddr_llc *addr)
  if (!sock_flag(sk, SOCK_ZAPPED))
    goto out;
  rc = -ENODEV;
- llc->dev = dev_getfirstbyhwtype(addr->sllc_arphrd);
+ llc->dev = dev_getfirstbyhwtype(init_net(), addr->sllc_arphrd);
  if (!llc->dev)
    goto out;
  rc = -EUSERS;
@@ -303,7 +303,7 @@ static int llc_ui_bind(struct socket *sock, struct sockaddr *uaddr, int
  addrlen)
  goto out;
  rc = -ENODEV;
  rtnl_lock();
- llc->dev = dev_getbyhwaddr(addr->sllc_arphrd, addr->sllc_mac);
+ llc->dev = dev_getbyhwaddr(init_net(), addr->sllc_arphrd, addr->sllc_mac);
  rtnl_unlock();
  if (!llc->dev)
    goto out;
diff --git a/net/llc/llc_core.c b/net/llc/llc_core.c

```


index f438c38..24a5739 100644

--- a/net/llc/llc_core.c

+++ b/net/llc/llc_core.c

@@ -19,6 +19,7 @@

#include <linux/slab.h>

#include <linux/string.h>

#include <linux/init.h>

+#include <net/net_namespace.h>

#include <net/llc.h>

LIST_HEAD(llc_sap_list);

@@ -159,8 +160,8 @@ static struct packet_type llc_tr_packet_type = {

static int __init llc_init(void)

{

- if (dev_base->next)

- memcpy(llc_station_mac_sa, dev_base->next->dev_addr, ETH_ALEN);

+ if (per_net(dev_base, init_net())->next)

+ memcpy(llc_station_mac_sa, per_net(dev_base, init_net())->next->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 8f88964..5bfd12e 100644

--- a/net/netrom/nr_route.c

+++ b/net/netrom/nr_route.c

@@ -581,7 +581,7 @@ static struct net_device *nr_ax25_dev_get(char *devname)

{

struct net_device *dev;

- if ((dev = dev_get_by_name(devname)) == NULL)

+ if ((dev = dev_get_by_name(init_net(), devname)) == NULL)

return NULL;

if ((dev->flags & IFF_UP) && dev->type == ARPHRD_AX25)

@@ -598,15 +598,15 @@ 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) {

+ read_lock(&per_net(dev_base_lock, init_net()));

+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {

if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM)

if (first == NULL || strncmp(dev->name, first->name, 3) < 0)

first = dev;

}

if (first)

```

    dev_hold(first);
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    return first;
}
@@ -618,15 +618,15 @@ struct net_device *nr_dev_get(ax25_address *addr)
{
    struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    if ((dev->flags & IFF_UP) && dev->type == ARPHRD_NETROM && ax25cmp(addr,
(ax25_address *)dev->dev_addr) == 0) {
        dev_hold(dev);
        goto out;
    }
}
out:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
    return dev;
}

```

diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c

index 6e3b947..4ac9f9f 100644

--- a/net/packet/af_packet.c

+++ b/net/packet/af_packet.c

```

@@ -359,7 +359,7 @@ static int packet_sendmsg_spkt(struct kiocb *iocb, struct socket *sock,
 */

```

```

    saddr->spkt_device[13] = 0;
- dev = dev_get_by_name(saddr->spkt_device);
+ dev = dev_get_by_name(init_net(), saddr->spkt_device);
    err = -ENODEV;
    if (dev == NULL)
        goto out_unlock;
@@ -744,7 +744,7 @@ static int packet_sendmsg(struct kiocb *iocb, struct socket *sock,
}

```

```

- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
    err = -ENXIO;
    if (dev == NULL)
        goto out_unlock;

```

```

@@ -943,7 +943,7 @@ static int packet_bind_spkt(struct socket *sock, struct sockaddr *uaddr,
int add
    return -EINVAL;
    strcpy(name,uaddr->sa_data,sizeof(name));

- dev = dev_get_by_name(name);
+ dev = dev_get_by_name(init_net(), name);
    if (dev) {
        err = packet_do_bind(sk, dev, pkt_sk(sk)->num);
        dev_put(dev);
@@ -971,7 +971,7 @@ static int packet_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len

    if (sll->sll_ifindex) {
        err = -ENODEV;
- dev = dev_get_by_index(sll->sll_ifindex);
+ dev = dev_get_by_index(init_net(), sll->sll_ifindex);
        if (dev == NULL)
            goto out;
    }
@@ -1158,7 +1158,7 @@ static int packet_getname_spkt(struct socket *sock, struct sockaddr
*uaddr,
    return -EOPNOTSUPP;

    uaddr->sa_family = AF_PACKET;
- dev = dev_get_by_index(pkt_sk(sk)->ifindex);
+ dev = dev_get_by_index(init_net(), pkt_sk(sk)->ifindex);
    if (dev) {
        strcpy(uaddr->sa_data, dev->name, 15);
        dev_put(dev);
@@ -1184,7 +1184,7 @@ static int packet_getname(struct socket *sock, struct sockaddr *uaddr,
    sll->sll_family = AF_PACKET;
    sll->sll_ifindex = po->ifindex;
    sll->sll_protocol = po->num;
- dev = dev_get_by_index(po->ifindex);
+ dev = dev_get_by_index(init_net(), po->ifindex);
    if (dev) {
        sll->sll_hatype = dev->type;
        sll->sll_halen = dev->addr_len;
@@ -1237,7 +1237,7 @@ static int packet_mc_add(struct sock *sk, struct packet_mreq_max
*mreq)
    rtnl_lock();

    err = -ENODEV;
- dev = __dev_get_by_index(mreq->mr_ifindex);
+ dev = __dev_get_by_index(init_net(), mreq->mr_ifindex);
    if (!dev)
        goto done;

```

```

@@ -1291,7 +1291,7 @@ static int packet_mc_drop(struct sock *sk, struct packet_mreq_max
*mreq)
    if (--ml->count == 0) {
        struct net_device *dev;
        *mlp = ml->next;
-   dev = dev_get_by_index(ml->ifindex);
+   dev = dev_get_by_index(init_net(), ml->ifindex);
    if (dev) {
        packet_dev_mc(dev, ml, -1);
        dev_put(dev);
@@ -1319,7 +1319,7 @@ static void packet_flush_mclist(struct sock *sk)
    struct net_device *dev;

    po->mclist = ml->next;
-   if ((dev = dev_get_by_index(ml->ifindex)) != NULL) {
+   if ((dev = dev_get_by_index(init_net(), ml->ifindex)) != NULL) {
        packet_dev_mc(dev, ml, -1);
        dev_put(dev);
    }
diff --git a/net/rose/rose_route.c b/net/rose/rose_route.c
index 8028c0d..92343be 100644
--- a/net/rose/rose_route.c
+++ b/net/rose/rose_route.c
@@ -579,7 +579,7 @@ static struct net_device *rose_ax25_dev_get(char *devname)
{
    struct net_device *dev;

-   if ((dev = dev_get_by_name(devname)) == NULL)
+   if ((dev = dev_get_by_name(init_net(), devname)) == NULL)
        return NULL;

    if ((dev->flags & IFF_UP) && dev->type == ARPHRD_AX25)
@@ -596,13 +596,13 @@ static 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) {
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
        if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE)
            if (first == NULL || strcmp(dev->name, first->name, 3) < 0)
                first = dev;
    }
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));

```

```

    return first;
}
@@ -614,15 +614,15 @@ struct net_device *rose_dev_get(rose_address *addr)
{
    struct net_device *dev;

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

```

```

@@ -630,13 +630,13 @@ static int rose_dev_exists(rose_address *addr)
{
    struct net_device *dev;

- read_lock(&dev_base_lock);
- for (dev = dev_base; dev != NULL; dev = dev->next) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev = per_net(dev_base, init_net()); dev != NULL; dev = dev->next) {
    if ((dev->flags & IFF_UP) && dev->type == ARPHRD_ROSE && rosecmp(addr, (rose_address
*)dev->dev_addr) == 0)
        goto out;
    }
out:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));
    return dev != NULL;
}

```

```

diff --git a/net/sched/act_mirred.c b/net/sched/act_mirred.c
index 4838972..2c3e4af 100644

```

```

--- a/net/sched/act_mirred.c

```

```

+++ b/net/sched/act_mirred.c

```

```

@@ -85,7 +85,7 @@ static int tcf_mirred_init(struct rtattr *rta, struct rtattr *est,
    parm = RTA_DATA(tb[TCA_MIRRED_PARMS-1]);

```

```

    if (parm->ifindex) {

```

```

- dev = __dev_get_by_index(parm->ifindex);
+ dev = __dev_get_by_index(init_net(), parm->ifindex);
  if (dev == NULL)
    return -ENODEV;
  switch (dev->type) {
diff --git a/net/sched/cls_api.c b/net/sched/cls_api.c
index edb8fc9..19935f9 100644
--- a/net/sched/cls_api.c
+++ b/net/sched/cls_api.c
@@ -164,7 +164,7 @@ replay:
  /* Find head of filter chain. */

  /* Find link */
- if ((dev = __dev_get_by_index(t->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), t->tcm_ifindex)) == NULL)
    return -ENODEV;

  /* Find qdisc */
@@ -397,7 +397,7 @@ static int tc_dump_tfilter(struct sk_buff *skb, struct netlink_callback *cb)

  if (cb->nlh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
    return skb->len;
- if ((dev = dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
    return skb->len;

  read_lock(&qdisc_tree_lock);
diff --git a/net/sched/em_meta.c b/net/sched/em_meta.c
index 45d47d3..5df7cdf 100644
--- a/net/sched/em_meta.c
+++ b/net/sched/em_meta.c
@@ -291,7 +291,7 @@ META_COLLECTOR(var_sk_bound_if)
  } else {
    struct net_device *dev;

- dev = dev_get_by_index(skb->sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), skb->sk->sk_bound_dev_if);
    *err = var_dev(dev, dst);
    if (dev)
      dev_put(dev);
diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index da7e1eb..912e8e1 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -586,7 +586,7 @@ static int tc_get_qdisc(struct sk_buff *skb, struct nlmsghdr *n, void *arg)
  struct Qdisc *p = NULL;
  int err;

```

```

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
    return -ENODEV;

    if (clid) {
@@ -653,7 +653,7 @@ replay:
    clid = tcm->tcm_parent;
    q = p = NULL;

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
    return -ENODEV;

    if (clid) {
@@ -858,8 +858,8 @@ static int tc_dump_qdisc(struct sk_buff *skb, struct netlink_callback *cb)

    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++) {
+ read_lock(&per_net(dev_base_lock, init_net()));
+ for (dev=per_net(dev_base, init_net()), idx=0; dev; dev = dev->next, idx++) {
    if (idx < s_idx)
        continue;
    if (idx > s_idx)
@@ -882,7 +882,7 @@ static int tc_dump_qdisc(struct sk_buff *skb, struct netlink_callback *cb)
    }

done:
- read_unlock(&dev_base_lock);
+ read_unlock(&per_net(dev_base_lock, init_net()));

    cb->args[0] = idx;
    cb->args[1] = q_idx;
@@ -912,7 +912,7 @@ static int tc_ctl_tclass(struct sk_buff *skb, struct nlmsg_hdr *n, void *arg)
    u32 qid = TC_H_MAJ(clid);
    int err;

- if ((dev = __dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = __dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
    return -ENODEV;

    /*
@@ -1095,7 +1095,7 @@ static int tc_dump_tclass(struct sk_buff *skb, struct netlink_callback
    *cb)

    if (cb->nlnh->nlmsg_len < NLMSG_LENGTH(sizeof(*tcm)))
        return 0;

```

```

- if ((dev = dev_get_by_index(tcm->tcm_ifindex)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), tcm->tcm_ifindex)) == NULL)
    return 0;

    s_t = cb->args[0];
diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index 0217546..10b748c 100644
--- a/net/sctp/ipv6.c
+++ b/net/sctp/ipv6.c
@@ -844,7 +844,7 @@ static int sctp_inet6_bind_verify(struct sctp_sock *opt, union sctp_addr
*addr)
    if (type & IPV6_ADDR_LINKLOCAL) {
        if (!addr->v6.sin6_scope_id)
            return 0;
-   dev = dev_get_by_index(addr->v6.sin6_scope_id);
+   dev = dev_get_by_index(init_net(), addr->v6.sin6_scope_id);
        if (!dev)
            return 0;
        dev_put(dev);
@@ -871,7 +871,7 @@ static int sctp_inet6_send_verify(struct sctp_sock *opt, union sctp_addr
*addr)
    if (type & IPV6_ADDR_LINKLOCAL) {
        if (!addr->v6.sin6_scope_id)
            return 0;
-   dev = dev_get_by_index(addr->v6.sin6_scope_id);
+   dev = dev_get_by_index(init_net(), addr->v6.sin6_scope_id);
        if (!dev)
            return 0;
        dev_put(dev);
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 9461a10..05e2bb5 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -170,14 +170,14 @@ static void sctp_get_local_addr_list(void)
    struct list_head *pos;
    struct sctp_af *af;

-   read_lock(&dev_base_lock);
-   for (dev = dev_base; dev; dev = dev->next) {
+   read_lock(&per_net(dev_base_lock, init_net()));
+   for (dev = per_net(dev_base, init_net()); dev; dev = dev->next) {
        __list_for_each(pos, &sctp_address_families) {
            af = list_entry(pos, struct sctp_af, list);
            af->copy_addrlist(&sctp_local_addr_list, dev);
        }
    }
-   read_unlock(&dev_base_lock);
+   read_unlock(&per_net(dev_base_lock, init_net()));

```



```

}

/* Free the existing local addresses. */
diff --git a/net/socket.c b/net/socket.c
index 0d0c92b..7371654 100644
--- a/net/socket.c
+++ b/net/socket.c
@@ -772,9 +772,9 @@ static ssize_t sock_aio_write(struct kiocb *iocb, const struct iovec *iov,
 */

static DEFINE_MUTEX(br_ioctl_mutex);
-static int (*br_ioctl_hook) (unsigned int cmd, void __user *arg) = NULL;
+static int (*br_ioctl_hook) (net_t, unsigned int cmd, void __user *arg) = NULL;

-void briocctl_set(int (*hook) (unsigned int, void __user *))
+void briocctl_set(int (*hook) (net_t, unsigned int, void __user *))
{
    mutex_lock(&br_ioctl_mutex);
    br_ioctl_hook = hook;
@@ -784,9 +784,9 @@ void briocctl_set(int (*hook) (unsigned int, void __user *))
EXPORT_SYMBOL(briocctl_set);

static DEFINE_MUTEX(vlan_ioctl_mutex);
-static int (*vlan_ioctl_hook) (void __user *arg);
+static int (*vlan_ioctl_hook) (net_t, void __user *arg);

-void vlan_ioctl_set(int (*hook) (void __user *))
+void vlan_ioctl_set(int (*hook) (net_t, void __user *))
{
    mutex_lock(&vlan_ioctl_mutex);
    vlan_ioctl_hook = hook;
@@ -815,16 +815,20 @@ EXPORT_SYMBOL(dlci_ioctl_set);
static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)
{
    struct socket *sock;
+ struct sock *sk;
    void __user *argp = (void __user *)arg;
    int pid, err;
+ net_t net;

    sock = file->private_data;
+ sk = sock->sk;
+ net = sk->sk_net;
    if (cmd >= SIOCDEVPRIVATE && cmd <= (SIOCDEVPRIVATE + 15)) {
- err = dev_ioctl(cmd, argp);
+ err = dev_ioctl(net, cmd, argp);
    } else
#ifdef CONFIG_WIRELESS_EXT

```

```

    if (cmd >= SIOCIWFIRST && cmd <= SIOCIWLAST) {
- err = dev_ioctl(cmd, argp);
+ err = dev_ioctl(net, cmd, argp);
    } else
#endif /* CONFIG_WIRELESS_EXT */
    switch (cmd) {
@@ -850,7 +854,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)

    mutex_lock(&br_ioctl_mutex);
    if (br_ioctl_hook)
- err = br_ioctl_hook(cmd, argp);
+ err = br_ioctl_hook(net, cmd, argp);
    mutex_unlock(&br_ioctl_mutex);
    break;
    case SIOCGIFVLAN:
@@ -861,7 +865,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)

    mutex_lock(&vlan_ioctl_mutex);
    if (vlan_ioctl_hook)
- err = vlan_ioctl_hook(argp);
+ err = vlan_ioctl_hook(net, argp);
    mutex_unlock(&vlan_ioctl_mutex);
    break;
    case SIOCADDLCI:
@@ -884,7 +888,7 @@ static long sock_ioctl(struct file *file, unsigned cmd, unsigned long arg)
    * to the NIC driver.
    */
    if (err == -ENOIOCTLCMD)
- err = dev_ioctl(cmd, argp);
+ err = dev_ioctl(net, cmd, argp);
    break;
}
return err;
diff --git a/net/tipc/eth_media.c b/net/tipc/eth_media.c
index c6f64de..ba207ba 100644
--- a/net/tipc/eth_media.c
+++ b/net/tipc/eth_media.c
@@ -127,7 +127,7 @@ 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 = per_net(dev_base, init_net());
    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;
diff --git a/net/wanrouter/af_wanpipe.c b/net/wanrouter/af_wanpipe.c
index f9b896c..397e876 100644

```

```

--- a/net/wanrouter/af_wanpipe.c
+++ b/net/wanrouter/af_wanpipe.c
@@ -586,7 +586,7 @@ static int wanpipe_sendmsg(struct kiocb *iocb, struct socket *sock,
    addr = saddr->sll_addr;
}

- dev = dev_get_by_index(ifindex);
+ dev = dev_get_by_index(init_net(), ifindex);
if (dev == NULL){
    printk(KERN_INFO "wansock: Send failed, dev index: %i\n",ifindex);
    return -ENXIO;
@@ -769,7 +769,7 @@ static int execute_command(struct sock *sk, unsigned char cmd,
unsigned int fla
    int err=0;
    DECLARE_WAITQUEUE(wait, current);

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
if (dev == NULL){
    printk(KERN_INFO "wansock: Exec failed no dev %i\n",
        sk->sk_bound_dev_if);
@@ -878,7 +878,7 @@ static void wanpipe_unlink_driver (struct sock *sk)
    sk->sk_state = WANSOCK_DISCONNECTED;
    wp_sk(sk)->dev = NULL;

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
if (!dev){
    printk(KERN_INFO "wansock: No dev on release\n");
    return;
@@ -974,7 +974,7 @@ static int wanpipe_release(struct socket *sock)

if (wp->num == htons(X25_PROT) &&
    sk->sk_state != WANSOCK_DISCONNECTED && sock_flag(sk, SOCK_ZAPPED)) {
- struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+ struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    wanpipe_common_t *chan;
    if (dev){
        chan=dev->priv;
@@ -1153,7 +1153,7 @@ static void wanpipe_kill_sock_timer (unsigned long data)

if (wp_sk(sk)->num == htons(X25_PROT) &&
    sk->sk_state != WANSOCK_DISCONNECTED) {
- struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+ struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    wanpipe_common_t *chan;
    if (dev){
        chan=dev->priv;

```

```

@@ -1396,7 +1396,7 @@ static int wanpipe_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_le
    * This is used by PVC mostly
    */
    strcpy(name,sll->sll_device,sizeof(name));
- dev = dev_get_by_name(name);
+ dev = dev_get_by_name(init_net(), name);
    if (dev == NULL){
        printk(KERN_INFO "wansock: Failed to get Dev from name: %s,\n",
            name);
@@ -1641,7 +1641,7 @@ static void wanpipe_wakeup_driver(struct sock *sk)
    struct net_device *dev = NULL;
    wanpipe_common_t *chan=NULL;

- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    if (!dev)
        return;

@@ -1680,7 +1680,7 @@ static int wanpipe_getname(struct socket *sock, struct sockaddr
*uaddr,
    sll->sll_family = AF_WANPIPE;
    sll->sll_ifindex = sk->sk_bound_dev_if;
    sll->sll_protocol = wp_sk(sk)->num;
- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
    if (dev) {
        sll->sll_hatype = dev->type;
        sll->sll_halen = dev->addr_len;
@@ -1898,7 +1898,7 @@ static int wanpipe_debug (struct sock *origsk, void *arg)
    return err;

    if (sk->sk_bound_dev_if) {
- dev = dev_get_by_index(sk->sk_bound_dev_if);
+ dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
        if (!dev)
            continue;

@@ -2009,7 +2009,7 @@ static int set_ioctl_cmd (struct sock *sk, void *arg)

    if (!wp_sk(sk)->mbox) {
        void *mbox_ptr;
- struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+ struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
        if (!dev)
            return -ENODEV;

@@ -2352,7 +2352,7 @@ static int wanpipe_exec_cmd(struct sock *sk, int cmd, unsigned int

```

flags)

```
static int check_driver_busy (struct sock *sk)
{
- struct net_device *dev = dev_get_by_index(sk->sk_bound_dev_if);
+ struct net_device *dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if);
  wanpipe_common_t *chan;

  if (!dev)
@@ -2507,7 +2507,7 @@ static int wanpipe_connect(struct socket *sock, struct sockaddr *uaddr,
int addr
  if (addr->sll_family != AF_WANPIPE)
    return -EINVAL;

- if ((dev = dev_get_by_index(sk->sk_bound_dev_if)) == NULL)
+ if ((dev = dev_get_by_index(init_net(), sk->sk_bound_dev_if)) == NULL)
    return -ENETUNREACH;

  dev_put(dev);
diff --git a/net/x25/x25_route.c b/net/x25/x25_route.c
index 2a3fe98..091b96e 100644
--- a/net/x25/x25_route.c
+++ b/net/x25/x25_route.c
@@ -126,7 +126,7 @@ void x25_route_device_down(struct net_device *dev)
 */
struct net_device *x25_dev_get(char *devname)
{
- struct net_device *dev = dev_get_by_name(devname);
+ struct net_device *dev = dev_get_by_name(init_net(), devname);

  if (dev &&
      (!(dev->flags & IFF_UP) || (dev->type != ARPHRD_X25
--
1.4.4.1.g278f
```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
