
Subject: [PATCH 4/12 net-2.6.25] [NETNS]: Add namespace parameter to ip_dev_find.

Posted by [den](#) on Tue, 22 Jan 2008 15:59:21 GMT

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

in_dev_find() need a namespace to pass it to fib_get_table(), so add an argument.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
drivers/infiniband/core/addr.c | 4 +---
drivers/infiniband/core/cma.c | 2 +-
include/linux/inetdevice.h    | 2 +-
net/ipv4/fib_frontend.c       | 4 +---
net/ipv4/igmp.c               | 2 +-
net/ipv4/ip_sockglue.c        | 2 +-
net/ipv4/ipmr.c               | 2 +-
net/ipv4/route.c              | 6 +++---
8 files changed, 12 insertions(+), 12 deletions(-)
```

diff --git a/drivers/infiniband/core/addr.c b/drivers/infiniband/core/addr.c

index 0802b79..963177e 100644

--- a/drivers/infiniband/core/addr.c

+++ b/drivers/infiniband/core/addr.c

@@ -110,7 +110,7 @@ int rdma_translate_ip(struct sockaddr *addr, struct rdma_dev_addr *dev_addr)

```
__be32 ip = ((struct sockaddr_in *) addr)->sin_addr.s_addr;
int ret;
```

```
- dev = ip_dev_find(ip);
+ dev = ip_dev_find(&init_net, ip);
if (!dev)
    return -EADDRNOTAVAIL;
```

@@ -261,7 +261,7 @@ static int addr_resolve_local(struct sockaddr_in *src_in, __be32 dst_ip = dst_in->sin_addr.s_addr; int ret;

```
- dev = ip_dev_find(dst_ip);
+ dev = ip_dev_find(&init_net, dst_ip);
if (!dev)
    return -EADDRNOTAVAIL;
```

diff --git a/drivers/infiniband/core/cma.c b/drivers/infiniband/core/cma.c

index b37045c..ef9efb3 100644

--- a/drivers/infiniband/core/cma.c

+++ b/drivers/infiniband/core/cma.c

@@ -1280,7 +1280,7 @@ static int iw_conn_req_handler(struct iw_cm_id *cm_id,

```

atomic_inc(&conn_id->dev_remove);
conn_id->state = CMA_CONNECT;

- dev = ip_dev_find(iw_event->local_addr.sin_addr.s_addr);
+ dev = ip_dev_find(&init_net, iw_event->local_addr.sin_addr.s_addr);
  if (!dev) {
    ret = -EADDRNOTAVAIL;
    cma_enable_remove(conn_id);
diff --git a/include/linux/inetdevice.h b/include/linux/inetdevice.h
index e74a2ee..8d9eaae 100644
--- a/include/linux/inetdevice.h
+++ b/include/linux/inetdevice.h
@@ -129,7 +129,7 @@ struct in_ifaddr
extern int register_inetaddr_notifier(struct notifier_block *nb);
extern int unregister_inetaddr_notifier(struct notifier_block *nb);

-extern struct net_device *ip_dev_find(__be32 addr);
+extern struct net_device *ip_dev_find(struct net *net, __be32 addr);
extern int inet_addr_onlink(struct in_device *in_dev, __be32 a, __be32 b);
extern int devinet_ioctl(unsigned int cmd, void __user *);
extern void devinet_init(void);
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 7e3e732..d282618 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -153,7 +153,7 @@ static void fib_flush(struct net *net)
 * Find the first device with a given source address.
 */

-struct net_device *ip_dev_find(__be32 addr)
+struct net_device *ip_dev_find(struct net *net, __be32 addr)
{
  struct flowi fl = { .nl_u = { .ip4_u = { .daddr = addr } } };
  struct fib_result res;
@@ -164,7 +164,7 @@ struct net_device *ip_dev_find(__be32 addr)
  res.r = NULL;
#endif

- local_table = fib_get_table(&init_net, RT_TABLE_LOCAL);
+ local_table = fib_get_table(net, RT_TABLE_LOCAL);
  if (!local_table || local_table->tb_lookup(local_table, &fl, &res))
    return NULL;
  if (res.type != RTN_LOCAL)
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 928bc32..1f5314c 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -1395,7 +1395,7 @@ static struct in_device *ip_mc_find_dev(struct ip_mreqn *imr)

```

```

    return idev;
}
if (imr->imr_address.s_addr) {
- dev = ip_dev_find(imr->imr_address.s_addr);
+ dev = ip_dev_find(&init_net, imr->imr_address.s_addr);
    if (!dev)
        return NULL;
    dev_put(dev);
diff --git a/net/ipv4/ip_sockglue.c b/net/ipv4/ip_sockglue.c
index 82817e5..754b0a5 100644
--- a/net/ipv4/ip_sockglue.c
+++ b/net/ipv4/ip_sockglue.c
@@ -594,7 +594,7 @@ static int do_ip_setsockopt(struct sock *sk, int level,
    err = 0;
    break;
}
- dev = ip_dev_find(mreq.imr_address.s_addr);
+ dev = ip_dev_find(&init_net, mreq.imr_address.s_addr);
    if (dev) {
        mreq.imr_ifindex = dev->ifindex;
        dev_put(dev);
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 4198615..2212717 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -423,7 +423,7 @@ static int vif_add(struct vifctl *vifc, int mrtsock)
    return -ENOBUFFS;
    break;
case 0:
- dev = ip_dev_find(vifc->vifc_lcl_addr.s_addr);
+ dev = ip_dev_find(&init_net, vifc->vifc_lcl_addr.s_addr);
    if (!dev)
        return -EADDRNOTAVAIL;
    dev_put(dev);
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 4313255..674575b 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2282,14 +2282,14 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
    goto out;

    /* It is equivalent to inet_addr_type(saddr) == RTN_LOCAL */
- dev_out = ip_dev_find(oldflp->fl4_src);
+ dev_out = ip_dev_find(&init_net, oldflp->fl4_src);
    if (dev_out == NULL)
        goto out;

```

/* I removed check for oif == dev_out->oif here.

It was wrong for two reasons:

- 1. ip_dev_find(saddr) can return wrong iface, if saddr is
- assigned to multiple interfaces.
- + 1. ip_dev_find(net, saddr) can return wrong iface, if saddr
- + is assigned to multiple interfaces.
- 2. Moreover, we are allowed to send packets with saddr
- of another iface. --ANK

*/

--

1.5.3.rc5
