

---

Subject: [PATCH net-2.6.26 3/3][TUN][NETNS]: Allow to register tun devices in namespace.

Posted by [Pavel Emelianov](#) on Wed, 02 Apr 2008 13:49:58 GMT

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

---

This is basically means that a net is set for a new device, but actually this involves two more steps:

1. mark the tun device as "local", i.e. do not allow for it to move across namespaces.

This is done so, since tun device is most often associated to some file (and thus to some process) and moving the device alone is not valid while keeping the file and the process outside.

2. get the tun device's net when tun becomes attached and put one when it becomes detached.

This is needed to handle the case when a task owning the tun dies, but a files lives for some more time - in this case we must not allow for net to be freed, since its exit hook will spoil that file's private data by unregistering the tun from under tun\_chr\_close.

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

---

```
drivers/net/tun.c | 4 +++++
1 files changed, 4 insertions(+), 0 deletions(-)
```

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

```
index e3210bf..bcc4ced 100644
```

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

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

```
@@ -438,6 +438,7 @@ static void tun_setup(struct net_device *dev)
```

```
    dev->stop = tun_net_close;
```

```
    dev->ethtool_ops = &tun_ethtool_ops;
```

```
    dev->destructor = free_netdev;
```

```
+ dev->features |= NETIF_F_NETNS_LOCAL;
```

```
}
```

```
static struct tun_struct *tun_get_by_name(struct net *net, const char *name)
```

```
@@ -503,6 +504,7 @@ static int tun_set_iff(struct net *net, struct file *file, struct ifreq *ifr)
```

```
    if (!dev)
```

```
        return -ENOMEM;
```

```
+ dev_net_set(dev, net);
```

```
    tun = netdev_priv(dev);
```

```
    tun->dev = dev;
```

```
tun->flags = flags;
@@ -542,6 +544,7 @@ static int tun_set_iff(struct net *net, struct file *file, struct ifreq *ifr)

file->private_data = tun;
tun->attached = 1;
+ get_net(dev_net(tun->dev));

strcpy(ifr->ifr_name, tun->dev->name);
return 0;
@@ -757,6 +760,7 @@ static int tun_chr_close(struct inode *inode, struct file *file)
/* Detach from net device */
file->private_data = NULL;
tun->attached = 0;
+ put_net(dev_net(tun->dev));

/* Drop read queue */
skb_queue_purge(&tun->readq);
--
```

1.5.3.4

---