

---

Subject: [PATCH net-2.6.26 2/3][TUN][NETNS]: Actually make the tun\_dev\_list per-net.

Posted by [Pavel Emelianov](#) on Wed, 02 Apr 2008 13:44:21 GMT

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

---

Remove the static tun\_dev\_list and replace its occurrences in driver with per-net one.

It is used in two places - in tun\_set\_iff and tun\_cleanup. In the first case it's legal to use current net\_ns. In the cleanup call - move the loop, that unregisters all devices in net exit hook.

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

---

```
drivers/net/tun.c | 35 ++++++-----
1 files changed, 17 insertions(+), 18 deletions(-)
```

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

```
index 3d518b1..e3210bf 100644
```

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

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

```
@@ -62,6 +62,7 @@
```

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

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

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

```
+#include <linux/nsproxy.h>
```

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

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

```
@@ -76,7 +77,6 @@ struct tun_net {
```

```
    struct list_head dev_list;
```

```
};
```

```
-static LIST_HEAD(tun_dev_list);
```

```
static const struct ethtool_ops tun_ethtool_ops;
```

```
/* Net device open. */
```

```
@@ -440,12 +440,12 @@ static void tun_setup(struct net_device *dev)
```

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

```
}
```

```
-static struct tun_struct *tun_get_by_name(const char *name)
```

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

```
{
```

```
    struct tun_struct *tun;
```

```

ASSERT_RTNL();
- list_for_each_entry(tun, &tun_dev_list, list) {
+ list_for_each_entry(tun, &net->tun->dev_list, list) {
    if (!strncmp(tun->dev->name, name, IFNAMSIZ))
        return tun;
}
@@ -453,13 +453,13 @@ static struct tun_struct *tun_get_by_name(const char *name)
    return NULL;
}

-static int tun_set_iff(struct file *file, struct ifreq *ifr)
+static int tun_set_iff(struct net *net, struct file *file, struct ifreq *ifr)
{
    struct tun_struct *tun;
    struct net_device *dev;
    int err;

- tun = tun_get_by_name(ifr->ifr_name);
+ tun = tun_get_by_name(net, ifr->ifr_name);
    if (tun) {
        if (tun->attached)
            return -EBUSY;
@@ -472,7 +472,7 @@ static int tun_set_iff(struct file *file, struct ifreq *ifr)
        !capable(CAP_NET_ADMIN))
            return -EPERM;
    }
- else if (__dev_get_by_name(&init_net, ifr->ifr_name))
+ else if (__dev_get_by_name(net, ifr->ifr_name))
    return -EINVAL;
    else {
        char *name;
@@ -525,7 +525,7 @@ static int tun_set_iff(struct file *file, struct ifreq *ifr)
        if (err < 0)
            goto err_free_dev;

- list_add(&tun->list, &tun_dev_list);
+ list_add(&tun->list, &net->tun->dev_list);
    }

    DBG(KERN_INFO "%s: tun_set_iff\n", tun->dev->name);
@@ -570,7 +570,7 @@ static int tun_chr_ioctl(struct inode *inode, struct file *file,
    ifr.ifr_name[IFNAMSIZ-1] = '\0';

    rtnl_lock();
- err = tun_set_iff(file, &ifr);
+ err = tun_set_iff(current->nsproxy->net_ns, file, &ifr);
    rtnl_unlock();

```

```

if (err)
@@ -805,6 +805,15 @@ static int tun_init_net(struct net *net)

static void tun_exit_net(struct net *net)
{
+ struct tun_struct *tun, *nxt;
+
+ rtnl_lock();
+ list_for_each_entry_safe(tun, nxt, &net->tun->dev_list, list) {
+ DBG(KERN_INFO "%s cleaned up\n", tun->dev->name);
+ unregister_netdevice(tun->dev);
+ }
+ rtnl_unlock();
+
+ kfree(net->tun);
+ net->tun = NULL;
+ }
@@ -928,17 +937,7 @@ err_pernet:

```

```

static void tun_cleanup(void)
{
- struct tun_struct *tun, *nxt;
-
- misc_deregister(&tun_miscdev);
-
- rtnl_lock();
- list_for_each_entry_safe(tun, nxt, &tun_dev_list, list) {
- DBG(KERN_INFO "%s cleaned up\n", tun->dev->name);
- unregister_netdevice(tun->dev);
- }
- rtnl_unlock();
-
- unregister_pernet_device(&tun_net_ops);
}

```

```

--
1.5.3.4

```