
Subject: [PATCH net-2.6.26 1/3][TUN][NETNS]: Introduce the tun_net structure.
Posted by [Pavel Emelianov](#) on Wed, 02 Apr 2008 13:38:48 GMT

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

This is the first step in making tuntap devices work in net namespaces. The structure mentioned is pointed by struct net and tun driver will fill this one on load. It will contain only the tun devices list.

So declare this structure and introduce net init and exit hooks.

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

```
---
drivers/net/tun.c      | 43 ++++++++++++++++++++++++++++++++++++++
include/net/net_namespace.h | 4 +++
2 files changed, 46 insertions(+), 1 deletions(-)
```

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

```
index 7b816a0..3d518b1 100644
```

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

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

```
@@ -72,6 +72,9 @@ static int debug;
```

```
#endif
```

```
/* Network device part of the driver */
```

```
+struct tun_net {
+ struct list_head dev_list;
+};
```

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

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

```
@@ -788,6 +791,29 @@ static struct miscdevice tun_miscdev = {
    .fops = &tun_fops,
};
```

```
+/* pernet operations */
```

```
+
```

```
+static int tun_init_net(struct net *net)
```

```
+
```

```
+ net->tun = kmalloc(sizeof(struct tun_net), GFP_KERNEL);
```

```
+ if (net->tun == NULL)
```

```
+ return -ENOMEM;
```

```
+
```

```
+ INIT_LIST_HEAD(&net->tun->dev_list);
```

```
+ return 0;
```

```
+
```

```
+
```

```

+static void tun_exit_net(struct net *net)
+{
+ kfree(net->tun);
+ net->tun = NULL;
+}
+
+static struct pernet_operations tun_net_ops = {
+ .init = tun_init_net,
+ .exit = tun_exit_net,
+};
+
+/* ethtool interface */

static int tun_get_settings(struct net_device *dev, struct ethtool_cmd *cmd)
@@ -880,9 +906,23 @@ static int __init tun_init(void)
    printk(KERN_INFO "tun: %s, %s\n", DRV_DESCRIPTION, DRV_VERSION);
    printk(KERN_INFO "tun: %s\n", DRV_COPYRIGHT);

+ ret = register_pernet_device(&tun_net_ops);
+ if (ret) {
+ printk(KERN_ERR "tun: Can't register pernet ops\n");
+ goto err_pernet;
+ }
+
    ret = misc_register(&tun_miscdev);
- if (ret)
+ if (ret) {
    printk(KERN_ERR "tun: Can't register misc device %d\n", TUN_MINOR);
+ goto err_misc;
+ }
+
+ return 0;
+
+err_misc:
+unregister_pernet_device(&tun_net_ops);
+err_pernet:
    return ret;
}

@@ -899,6 +939,7 @@ static void tun_cleanup(void)
}
rtnl_unlock();

+ unregister_pernet_device(&tun_net_ops);
}

module_init(tun_init);
diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h

```

```
index c01d45f..52616fa 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -19,6 +19,7 @@ struct proc_dir_entry;
 struct net_device;
 struct sock;
 struct ctl_table_header;
+struct tun_net;

struct net {
    atomic_t count; /* To decided when the network
@@ -57,6 +58,9 @@ struct net {
#ifdef CONFIG_NETFILTER
    struct netns_xt xt;
#endif
+
+ /* modules private data */
+ struct tun_net *tun;
};

#ifdef CONFIG_NET
--
1.5.3.4
```
