Subject: [PATCH 4/14][TUN]: Actually make the tun_dev_list per-net.
Posted by Pavel Emelianov on Thu, 10 Apr 2008 14:48:10 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.

This shows how to use the generic pointer.

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

```
---
 drivers/net/tun.c |   37 ++++++++++++++++++++++++++-----------
 1 files changed, 19 insertions(+), 18 deletions(-)

diff --git a/drivers/net/tun.c b/drivers/net/tun.c
index 9bfba02..74263a4 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 <net/netns/generic.h>

@@ -79,7 +80,6 @@ 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. */
@@ -443,12 +443,12 @@ 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 tun_net *tn, const char *name)
 {
  struct tun_struct *tun;
```

```
  ASSERT_RTNL();
- list_for_each_entry(tun, &tun_dev_list, list) {
+ list_for_each_entry(tun, &tn->dev_list, list) {
    if (!strncmp(tun->dev->name, name, IFNAMSIZ))
        return tun;
  }
@@ -456,13 +456,15 @@ 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_net *tn;
  struct tun_struct *tun;
  struct net_device *dev;
  int err;

- tun = tun_get_by_name(ifr->ifr_name);
+ tn = net_generic(net, tun_net_id);
+ tun = tun_get_by_name(tn, ifr->ifr_name);
  if (tun) {
    if (tun->attached)
      return -EBUSY;
@@ -475,7 +477,7 @@ 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;
@@ -528,7 +530,7 @@ 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, &tn->dev_list);
  }

  DBG(KERN_INFO "%s: tun_set_iff\n", tun->dev->name);
@@ -573,7 +575,7 @@ 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)
@@ -900,8 +902,17 @@ static int tun_init_net(struct net *net)
 static void tun_exit_net(struct net *net)
 {
  struct tun_net *tn;
+ struct tun_struct *tun, *nxt;

  tn = net_generic(net, tun_net_id);
+
+ rtnl_lock();
+ list_for_each_entry_safe(tun, nxt, &tn->dev_list, list) {
+  DBG(KERN_INFO "%s cleaned up\n", tun->dev->name);
+  unregister_netdevice(tun->dev);
+ }
+ rtnl_unlock();
+
  kfree(tn);
 }

@@ -938,17 +949,7 @@ 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_gen_device(tun_net_id, &tun_net_ops);
 }

--
1.5.3.4
```

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers