Subject: [PATCH] tun: don't hold network namespace by tun sockets
Posted by Stanislav Kinsbursky on Sun, 11 Mar 2012 15:21:03 GMT
View Forum Message <> Reply to Message

TUN was designed to destroy it's socket on network namesapce shutdown. But this
will never happen for persistent device, because it's socket holds network
namespace.
This patch removes of holding network namespace by TUN socket and replaces it
by creating socket in init_net and then changing it's net it to desired one. On
shutdown socket is moved back to init_net prior to final put.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

---
 drivers/net/tun.c |   10 +++++++---
 1 files changed, 7 insertions(+), 3 deletions(-)

```
diff --git a/drivers/net/tun.c b/drivers/net/tun.c
index 2c5d349..92ef539 100644
--- a/drivers/net/tun.c
+++ b/drivers/net/tun.c
@@ -358,8 +358,11 @@ static void tun_net_uninit(struct net_device *dev)
 static void tun_free_netdev(struct net_device *dev)
 {
  struct tun_struct *tun = netdev_priv(dev);
+ struct sock *sk = tun->socket.sk;

- sock_put(tun->socket.sk);
+ release_net(sock_net(sk));
+ sock_net_set(sk, get_net(&init_net));
+ sock_put(sk);
 }

 /* Net device open. */
@@ -1110,10 +1113,11 @@ static int tun_set_iff(struct net *net, struct file *file, struct ifreq *ifr)
   tun->vnet_hdr_sz = sizeof(struct virtio_net_hdr);

   err = -ENOMEM;
-  sk = sk_alloc(net, AF_UNSPEC, GFP_KERNEL, &tun_proto);
+  sk = sk_alloc(&init_net, AF_UNSPEC, GFP_KERNEL, &tun_proto);
   if (!sk)
    goto err_free_dev;

+  sk_change_net(sk, net);
   tun->socket.wq = &tun->wq;
   init_waitqueue_head(&tun->wq.wait);
   tun->socket.ops = &tun_socket_ops;
@@ -1174,7 +1178,7 @@ static int tun_set_iff(struct net *net, struct file *file, struct ifreq *ifr)
```

```
  return 0;

  err_free_sk:
- sock_put(sk);
+ tun_free_netdev(dev);
  err_free_dev:
  free_netdev(dev);
  failed:
```

---

Subject: Re: [PATCH] tun: don't hold network namespace by tun sockets
Posted by Eric Dumazet on Sun, 11 Mar 2012 17:29:29 GMT
View Forum Message <> Reply to Message

Le dimanche 11 mars 2012 à 19:21 +0400, Stanislav Kinsbursky a écrit :
> TUN was designed to destroy it's socket on network namesapce shutdown. But this
> will never happen for persistent device, because it's socket holds network
> namespace.
> This patch removes of holding network namespace by TUN socket and replaces it
> by creating socket in init_net and then changing it's net it to desired one. On
> shutdown socket is moved back to init_net prior to final put.
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
>  drivers/net/tun.c |   10 +++++++---
>  1 files changed, 7 insertions(+), 3 deletions(-)
>
> diff --git a/drivers/net/tun.c b/drivers/net/tun.c
> index 2c5d349..92ef539 100644
> --- a/drivers/net/tun.c
> +++ b/drivers/net/tun.c
> @@ -358,8 +358,11 @@ static void tun_net_uninit(struct net_device *dev)
>  static void tun_free_netdev(struct net_device *dev)
>  {
>   struct tun_struct *tun = netdev_priv(dev);
> + struct sock *sk = tun->socket.sk;
>
> - sock_put(tun->socket.sk);
> + release_net(sock_net(sk));
> + sock_net_set(sk, get_net(&init_net));
> + sock_put(sk);


Hmm, maybe use sk_release_kernel(), as its should be the thing
associated with sk_change_net().

Or at least make intent clear, since its not obvious.

---