

---

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 namespace 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 to desired one. On shutdown socket is moved back to init\_net prior to final put.

Signed-off-by: Stanislav Kinsbursky <[skinsbursky@parallels.com](mailto: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 namespace 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.

---