
Subject: [net-2.6.24][patch 2/2] Dynamically allocate the loopback device

Posted by [Daniel Lezcano](#) on Mon, 17 Sep 2007 13:45:11 GMT

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

From: Daniel Lezcano <dlezcano@fr.ibm.com>

Doing this makes loopback.c a better example of how to do a simple network device, and it removes the special case single static allocation of a struct net_device, hopefully making maintenance easier.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

Acked-By: Kirill Korotaev <dev@sw.ru>

Acked-by: Benjamin Thery <benjamin.thery@bull.net>

drivers/net/loopback.c | 69 ++++++-----
1 file changed, 43 insertions(+), 26 deletions(-)

Index: net-2.6.24/drivers/net/loopback.c

=====

--- net-2.6.24.orig/drivers/net/loopback.c

+++ net-2.6.24/drivers/net/loopback.c

@@ -202,44 +202,61 @@ static const struct ethtool_ops loopback

* The loopback device is special. There is only one instance and
* it is statically allocated. Don't do this for other devices.

*/

-struct net_device __loopback_dev = {

- .name = "lo",

- .get_stats = &get_stats,

- .mtu = (16 * 1024) + 20 + 20 + 12,

- .hard_start_xmit = loopback_xmit,

- .hard_header = eth_header,

- .hard_header_cache = eth_header_cache,

- .header_cache_update = eth_header_cache_update,

- .hard_header_len = ETH_HLEN, /* 14 */

- .addr_len = ETH_ALEN, /* 6 */

- .tx_queue_len = 0,

- .type = ARPHRD_LOOPBACK, /* 0x0001*/

- .rebuild_header = eth_rebuild_header,

- .flags = IFF_LOOPBACK,

- .features = NETIF_F_SG | NETIF_F_FRAGLIST

+static void loopback_setup(struct net_device *dev)

+{

+ dev->get_stats = &get_stats;

+ dev->mtu = (16 * 1024) + 20 + 20 + 12;

+ dev->hard_start_xmit = loopback_xmit;

+ dev->hard_header = eth_header;

```

+ dev->hard_header_cache = eth_header_cache;
+ dev->header_cache_update = eth_header_cache_update;
+ dev->hard_header_len = ETH_HLEN; /* 14 */
+ dev->addr_len = ETH_ALEN; /* 6 */
+ dev->tx_queue_len = 0;
+ dev->type = ARPHRD_LOOPBACK; /* 0x0001*/
+ dev->rebuild_header = eth_rebuild_header;
+ dev->flags = IFF_LOOPBACK;
+ dev->features = NETIF_F_SG | NETIF_F_FRAGLIST
#ifdef LOOPBACK_TSO
-   | NETIF_F_TSO
+ | NETIF_F_TSO
#endif
-   | NETIF_F_NO_CSUM | NETIF_F_HIGHDMA
-   | NETIF_F_LLTX
-   | NETIF_F_NETNS_LOCAL,
- .ethtool_ops = &loopback_ethtool_ops,
- .nd_net = &init_net,
-};
-
-struct net_device *loopback_dev = &__loopback_dev;
+ | NETIF_F_NO_CSUM
+ | NETIF_F_HIGHDMA
+ | NETIF_F_LLTX
+ | NETIF_F_NETNS_LOCAL,
+ dev->ethtool_ops = &loopback_ethtool_ops;
+}

/* Setup and register the loopback device. */
static int __init loopback_init(void)
{
- int err = register_netdev(loopback_dev);
+ struct net_device *dev;
+ int err;
+
+ err = -ENOMEM;
+ dev = alloc_netdev(0, "lo", loopback_setup);
+ if (!dev)
+ goto out;

+ err = register_netdev(dev);
+ if (err)
+ goto out_free_netdev;
+
+ err = 0;
+ loopback_dev = dev;
+
+out:

```

```
if (err)
    panic("loopback: Failed to register netdevice: %d\n", err);
+ return err;

+out_free_netdev:
+ free_netdev(dev);
+ goto out;
    return err;
};

-module_init(loopback_init);
+fs_initcall(loopback_init);

+struct net_device *loopback_dev;
EXPORT_SYMBOL(loopback_dev);

--
```

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