Subject: Re: [PATCH] Virtual ethernet tunnel
Posted by Stephen Hemminger on Wed, 06 Jun 2007 16:17:17 GMT
View Forum Message <> Reply to Message

On Wed, 06 Jun 2007 19:11:38 +0400
Pavel Emelianov <xemul@openvz.org> wrote:

> Veth stands for Virtual ETHernet. It is a simple tunnel driver
> that works at the link layer and looks like a pair of ethernet
> devices interconnected with each other.
>
> Mainly it allows to communicate between network namespaces but
> it can be used as is as well.
>
> Eric recently sent a similar driver called etun. This
> implementation uses another interface - the RTM_NRELINK
> message introduced by Patric. The patch fits today netdev
> tree with Patrick's patches.
>
> The newlink callback is organized that way to make it easy
> to create the peer device in the separate namespace when we
> have them in kernel.
>
> The patch for an ip utility is also provided.
>
> Eric, since ethtool interface was from your patch, I add
> your Signed-off-by line.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> Signed-off-by: Pavel Emelianov <xemul@openvz.org>
>
> ---
>
> diff --git a/drivers/net/Kconfig b/drivers/net/Kconfig
> index 7d57f4a..7e144be 100644
> --- a/drivers/net/Kconfig
> +++ b/drivers/net/Kconfig
> @@ -119,6 +119,12 @@ config TUN
>
> If you don't know what to use this for, you don't need it.
>
> +config VETH
> + tristate "Virtual ethernet device"
> + ---help---
> + The device is an ethernet tunnel. Devices are created in pairs. When
> + one end receives the packet it appears on its pair and vice versa.
> +
> config NET_SB1000

>   tristate "General Instruments Surfboard 1000"
>   depends on PNP
> diff --git a/drivers/net/Makefile b/drivers/net/Makefile
> index a77affa..4764119 100644
> --- a/drivers/net/Makefile
> +++ b/drivers/net/Makefile
> @@ -185,6 +185,7 @@ obj-$(CONFIG_MACSONIC) += macsonic.o
> obj-$(CONFIG_MACMACE) += macmace.o
> obj-$(CONFIG_MAC89x0) += mac89x0.o
> obj-$(CONFIG_TUN) += tun.o
> +obj-$(CONFIG_VETH) += veth.o
> obj-$(CONFIG_NET_NETX) += netx-eth.o
> obj-$(CONFIG_DL2K) += dl2k.o
> obj-$(CONFIG_R8169) += r8169.o
> diff --git a/drivers/net/veth.c b/drivers/net/veth.c
> new file mode 100644
> index 0000000..6746c91
> --- /dev/null
> +++ b/drivers/net/veth.c
> @@ -0,0 +1,391 @@
> +/*
> + * drivers/net/veth.c
> + *
> + * Copyright (C) 2007 OpenVZ http://openvz.org, SWsoft Inc
> + *
> + */
> +
> +#include <linux/list.h>
> +#include <linux/netdevice.h>
> +#include <linux/ethtool.h>
> +#include <linux/etherdevice.h>
> +
> +#include <net/dst.h>
> +#include <net/xfrm.h>
> +#include <net/veth.h>
> +
> +#define DRV_NAME "veth"
> +#define DRV_VERSION "1.0"
> +
> +struct veth_priv {
> + struct net_device *peer;
> + struct net_device *dev;
> + struct list_head list;
> + struct net_device_stats stats;
> + unsigned ip_summed;
> +};
> +
> +static LIST_HEAD(veth_list);

```
> +
> +/*
> + * ethtool interface
> + */
> +
> +static struct {
> + const char string[ETH_GSTRING_LEN];
> +} ethtool_stats_keys[] = {
> + { "peer_ifindex" },
> +};
```

Seems like a good usage of sysfs attributes, rather than ethtool.

Then you can get rid of all the useless ethtool for what is
basically a virtual device.

```
> +/*
> + * xmit
> + */
> +
> +static int veth_xmit(struct sk_buff *skb, struct net_device *dev)
> +{
> + struct net_device *rcv = NULL;
> + struct veth_priv *priv, *rcv_priv;
> + int length;
> +
> + skb_orphan(skb);
> +
> + priv = netdev_priv(dev);
> + rcv = priv->peer;
> + rcv_priv = netdev_priv(rcv);
> +
> + if (!(rcv->flags & IFF_UP))
> +  goto outf;
> +
> + skb->dev = rcv;
> + skb->pkt_type = PACKET_HOST;
> + skb->protocol = eth_type_trans(skb, rcv);
> + if (dev->features & NETIF_F_NO_CSUM)
> +  skb->ip_summed = rcv_priv->ip_summed;
> +
> + dst_release(skb->dst);
> + skb->dst = NULL;
> +
> + secpath_reset(skb);
> + nf_reset(skb);
> +
> + length = skb->len;
```

> +
> + priv->stats.tx_bytes += length;
> + priv->stats.tx_packets++;
> +
> + rcv_priv->stats.rx_bytes += length;
> + rcv_priv->stats.rx_packets++;

Per-cpu stats? This will cacheline thrash.

> + netif_rx(skb);
> + return 0;
> +
> +outf:
> + kfree_skb(skb);
> + priv->stats.tx_dropped++;
> + return 0;
> +}

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