
Subject: [PATCH][VLAN] Merge tree equal tails in vlan_skb_rcv

Posted by [Pavel Emelianov](#) on Fri, 07 Dec 2007 10:22:07 GMT

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

There are tree paths in it, that set the skb->proto and then perform common receive manipulations (basically call netif_rx()).

I think, that we can make this code flow easier to understand by introducing the vlan_set_encap_proto() function (I hope the name is good) to setup the skb proto and merge the paths calling netif_rx() together.

Surprisingly, but gcc detects this thing and merges these paths by itself, so this patch doesn't make the vlan module smaller.

Fits both net-2.6 and net-2.6.25.

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

```
diff --git a/net/8021q/vlan_dev.c b/net/8021q/vlan_dev.c
```

```
index 4f99bb8..11198c1 100644
```

```
--- a/net/8021q/vlan_dev.c
```

```
+++ b/net/8021q/vlan_dev.c
```

```
@@ -90,6 +90,40 @@ static inline struct sk_buff *vlan_check_reorder_header(struct sk_buff *skb)
    return skb;
}
```

```
+static inline void vlan_set_encap_proto(struct sk_buff *skb,
+ struct vlan_hdr *vhdr)
+{
+ __be16 proto;
+ unsigned char *rawp;
+
+ /*
+ * Was a VLAN packet, grab the encapsulated protocol, which the layer
+ * three protocols care about.
+ */
+
+ proto = vhdr->h_vlan_encapsulated_proto;
+ if (ntohs(proto) >= 1536) {
+   skb->protocol = proto;
+   return;
+ }
+
+ rawp = skb->data;
```

```

+ if (*(unsigned short *)rawp == 0xFFFF)
+ /*
+  * This is a magic hack to spot IPX packets. Older Novell
+  * breaks the protocol design and runs IPX over 802.3 without
+  * an 802.2 LLC layer. We look for FFFF which isn't a used
+  * 802.2 SSAP/DSAP. This won't work for fault tolerant network
+  * but does for the rest.
+  */
+ skb->protocol = htons(ETH_P_802_3);
+ else
+ /*
+  * Real 802.2 LLC
+  */
+ skb->protocol = htons(ETH_P_802_2);
+}
+
+/*
+ * Determine the packet's protocol ID. The rule here is that we
+ * assume 802.3 if the type field is short enough to be a length.
@@ -115,12 +149,10 @@ static inline struct sk_buff *vlan_check_reorder_header(struct sk_buff
*skb)
int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
struct packet_type* ptype, struct net_device *orig_dev)
{
- unsigned char *rawp = NULL;
- struct vlan_hdr *vhdr;
- unsigned short vid;
- struct net_device_stats *stats;
- unsigned short vlan_TCI;
- __be16 proto;

if (dev->nd_net != &init_net) {
kfree_skb(skb);
@@ -236,70 +268,11 @@ int vlan_skb_recv(struct sk_buff *skb, struct net_device *dev,
break;
}

- /* Was a VLAN packet, grab the encapsulated protocol, which the layer
- * three protocols care about.
- */
- /* proto = get_unaligned(&vhdr->h_vlan_encapsulated_proto); */
- proto = vhdr->h_vlan_encapsulated_proto;
-
- skb->protocol = proto;
- if (ntohs(proto) >= 1536) {
- /* place it back on the queue to be handled by
- * true layer 3 protocols.
- */

```

```

-
- /* See if we are configured to re-write the VLAN header
- * to make it look like ethernet...
- */
- skb = vlan_check_reorder_header(skb);
-
- /* Can be null if skb-clone fails when re-ordering */
- if (skb) {
-     netif_rx(skb);
- } else {
-     /* TODO: Add a more specific counter here. */
-     stats->rx_errors++;
- }
- rcu_read_unlock();
- return 0;
- }
-
- rawp = skb->data;
-
- /*
- * This is a magic hack to spot IPX packets. Older Novell breaks
- * the protocol design and runs IPX over 802.3 without an 802.2 LLC
- * layer. We look for FFFF which isn't a used 802.2 SSAP/DSAP. This
- * won't work for fault tolerant netware but does for the rest.
- */
- if (*(unsigned short *)rawp == 0xFFFF) {
-     skb->protocol = htons(ETH_P_802_3);
-     /* place it back on the queue to be handled by true layer 3 protocols.
-     */
- }
-
- /* See if we are configured to re-write the VLAN header
- * to make it look like ethernet...
- */
- skb = vlan_check_reorder_header(skb);
-
- /* Can be null if skb-clone fails when re-ordering */
- if (skb) {
-     netif_rx(skb);
- } else {
-     /* TODO: Add a more specific counter here. */
-     stats->rx_errors++;
- }
- rcu_read_unlock();
- return 0;
- }
-
- /*
- * Real 802.2 LLC

```

```
- */
- skb->protocol = htons(ETH_P_802_2);
  /* place it back on the queue to be handled by upper layer protocols.
  */

+ vlan_set_encap_proto(skb, vhdr);
+
  /* See if we are configured to re-write the VLAN header
  * to make it look like ethernet...
  */
--
1.5.3.4
```
