Subject: [PATCH net-2.6.25] Make macro to specify the ptype_base size
Posted by Pavel Emelianov on Fri, 23 Nov 2007 13:47:36 GMT
View Forum Message <> Reply to Message

Currently this size is 16, but as the comment says this
is so only because all the chains (except one) has the
length 1. I think, that some day this may change, so
growing this hash will be much easier.

Besides, symbolic names are read better than magic constants.

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

```
diff --git a/net/core/dev.c b/net/core/dev.c
index 043e2f8..d88d832 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -150,8 +150,11 @@
  *  86DD IPv6
  */

+#define PTYPE_HASH_SIZE (16)
+#define PTYPE_HASH_MASK (PTYPE_HASH_SIZE - 1)
+
 static DEFINE_SPINLOCK(ptype_lock);
-static struct list_head ptype_base[16] __read_mostly; /* 16 way hashed list */
+static struct list_head ptype_base[PTYPE_HASH_SIZE] __read_mostly;
 static struct list_head ptype_all __read_mostly; /* Taps */

 #ifdef CONFIG_NET_DMA
@@ -362,7 +365,7 @@ void dev_add_pack(struct packet_type *pt)
  if (pt->type == htons(ETH_P_ALL))
   list_add_rcu(&pt->list, &ptype_all);
  else {
-  hash = ntohs(pt->type) & 15;
+  hash = ntohs(pt->type) & PTYPE_HASH_MASK;
   list_add_rcu(&pt->list, &ptype_base[hash]);
  }
  spin_unlock_bh(&ptype_lock);
@@ -391,7 +394,7 @@ void __dev_remove_pack(struct packet_type *pt)
  if (pt->type == htons(ETH_P_ALL))
   head = &ptype_all;
  else
-  head = &ptype_base[ntohs(pt->type) & 15];
+  head = &ptype_base[ntohs(pt->type) & PTYPE_HASH_MASK];
```

```
  list_for_each_entry(pt1, head, list) {
   if (pt == pt1) {
@@ -1420,7 +1423,8 @@ struct sk_buff *skb_gso_segment(struct sk_buff *skb, int features)
  }

  rcu_read_lock();
- list_for_each_entry_rcu(ptype, &ptype_base[ntohs(type) & 15], list) {
+ list_for_each_entry_rcu(ptype,
+    &ptype_base[ntohs(type) & PTYPE_HASH_MASK], list) {
   if (ptype->type == type && !ptype->dev && ptype->gso_segment) {
    if (unlikely(skb->ip_summed != CHECKSUM_PARTIAL)) {
     err = ptype->gso_send_check(skb);
@@ -2077,7 +2081,8 @@ ncls:
   goto out;

  type = skb->protocol;
- list_for_each_entry_rcu(ptype, &ptype_base[ntohs(type)&15], list) {
+ list_for_each_entry_rcu(ptype,
+    &ptype_base[ntohs(type) & PTYPE_HASH_MASK], list) {
   if (ptype->type == type &&
       (!ptype->dev || ptype->dev == skb->dev)) {
    if (pt_prev)
@@ -2521,7 +2526,7 @@ static void *ptype_get_idx(loff_t pos)
   ++i;
  }

- for (t = 0; t < 16; t++) {
+ for (t = 0; t < PTYPE_HASH_SIZE; t++) {
   list_for_each_entry_rcu(pt, &ptype_base[t], list) {
    if (i == pos)
     return pt;
@@ -2555,10 +2560,10 @@ static void *ptype_seq_next(struct seq_file *seq, void *v, loff_t *pos)
   hash = 0;
   nxt = ptype_base[0].next;
  } else
-  hash = ntohs(pt->type) & 15;
+  hash = ntohs(pt->type) & PTYPE_HASH_MASK;

  while (nxt == &ptype_base[hash]) {
-  if (++hash >= 16)
+  if (++hash >= PTYPE_HASH_SIZE)
    return NULL;
   nxt = ptype_base[hash].next;
  }
@@ -4396,7 +4401,7 @@ static int __init net_dev_init(void)
   goto out;

  INIT_LIST_HEAD(&ptype_all);
```

```
- for (i = 0; i < 16; i++)
+ for (i = 0; i < PTYPE_HASH_SIZE; i++)
    INIT_LIST_HEAD(&ptype_base[i]);

  if (register_pernet_subsys(&netdev_net_ops))
--
1.5.3.4
```
