Subject: [PATCH 0/4] Diet struct sk_buff a bit
Posted by Pavel Emelianov on Fri, 19 Oct 2007 08:59:17 GMT
View Forum Message <> Reply to Message

The __u16 queue_mapping field only makes sense in the
CONFIG_NETDEVICES_MULTIQUEUE=y case only.

Despite this field may be set explicitly to some non-zero
value (in net/core/pktgen.c), the exact value affects
nothing in case the config option in question is N (mainly
it is used in netif_subqueue_stopped(), which will always
return 0 in this case).

So cleanup the code a bit and move this field under the
config option.

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

Subject: [PATCH 1/4] Use the skb_set_queue_mapping where appropriate
Posted by Pavel Emelianov on Fri, 19 Oct 2007 09:00:52 GMT
View Forum Message <> Reply to Message

There's already such a helper to initialize this field.
Use it.

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

```
diff --git a/net/core/dev.c b/net/core/dev.c
index 38b03da..8726589 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -1661,7 +1661,7 @@ gso:
   q = dev->qdisc;
   if (q->enqueue) {
    /* reset queue_mapping to zero */
-   skb->queue_mapping = 0;
+   skb_set_queue_mapping(skb, 0);
    rc = q->enqueue(skb, q);
    qdisc_run(dev);
    spin_unlock(&dev->queue_lock);
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 8cae60c..d4d9a36 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -2603,8 +2603,7 @@ static struct sk_buff *fill_packet_ipv4(struct net_device *odev,
```

```
  skb->network_header = skb->tail;
  skb->transport_header = skb->network_header + sizeof(struct iphdr);
  skb_put(skb, sizeof(struct iphdr) + sizeof(struct udphdr));
- skb->queue_mapping = pkt_dev->cur_queue_map;
-
+ skb_set_queue_mapping(skb, pkt_dev->cur_queue_map);
  iph = ip_hdr(skb);
  udph = udp_hdr(skb);

@@ -2941,8 +2940,7 @@ static struct sk_buff *fill_packet_ipv6(struct net_device *odev,
  skb->network_header = skb->tail;
  skb->transport_header = skb->network_header + sizeof(struct ipv6hdr);
  skb_put(skb, sizeof(struct ipv6hdr) + sizeof(struct udphdr));
- skb->queue_mapping = pkt_dev->cur_queue_map;
-
+ skb_set_queue_mapping(skb, pkt_dev->cur_queue_map);
  iph = ipv6_hdr(skb);
  udph = udp_hdr(skb);
```

Subject: [PATCH 2/4] Make and use skb_get_queue_mapping
Posted by Pavel Emelianov on Fri, 19 Oct 2007 09:02:31 GMT
View Forum Message <> Reply to Message

Make the helper for getting the field, symmetrical to
the "set" one. Return 0 if CONFIG_NETDEVICES_MULTIQUEUE=n

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

```
diff --git a/drivers/net/cpmac.c b/drivers/net/cpmac.c
index ed53aaa..ae41973 100644
--- a/drivers/net/cpmac.c
+++ b/drivers/net/cpmac.c
@@ -471,7 +471,7 @@ static int cpmac_start_xmit(struct sk_buff *skb, struct net_device *dev)
  }

  len = max(skb->len, ETH_ZLEN);
- queue = skb->queue_mapping;
+ queue = skb_get_queue_mapping(skb);
 #ifdef CONFIG_NETDEVICES_MULTIQUEUE
  netif_stop_subqueue(dev, queue);
 #else
diff --git a/include/linux/skbuff.h b/include/linux/skbuff.h
index f93f22b..580ed1f 100644
--- a/include/linux/skbuff.h
+++ b/include/linux/skbuff.h
```

```
@@ -1770,6 +1771,15 @@ static inline void skb_set_queue_mapping(struct sk_buff *skb, u16
queue_mapping)
 #endif
 }

+static inline u16 skb_get_queue_mapping(struct sk_buff *skb)
+{
+#ifdef CONFIG_NETDEVICES_MULTIQUEUE
+ return skb->queue_mapping;
+#else
+ return 0;
+#endif
+}
+
 static inline void skb_copy_queue_mapping(struct sk_buff *to, const struct sk_buff *from)
 {
 #ifdef CONFIG_NETDEVICES_MULTIQUEUE
diff --git a/net/sched/sch_teql.c b/net/sched/sch_teql.c
index be57cf3..421281d 100644
--- a/net/sched/sch_teql.c
+++ b/net/sched/sch_teql.c
@@ -266,7 +266,7 @@ static int teql_master_xmit(struct sk_buff *skb, struct net_device *dev)
  int busy;
  int nores;
  int len = skb->len;
- int subq = skb->queue_mapping;
+ int subq = skb_get_queue_mapping(skb);
  struct sk_buff *skb_res = NULL;

  start = master->slaves;
```

Subject: [PATCH 3/4] Hide the queue_mapping field inside netif_subqueue_stopped
Posted by Pavel Emelianov on Fri, 19 Oct 2007 09:04:13 GMT
View Forum Message <> Reply to Message

Many places get the queue_mapping field from skb to pass it
to the netif_subqueue_stopped() which will be 0 in any case.

Make the helper that works with sk_buff

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

```
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 6f85db3..4a3f54e 100644
--- a/include/linux/netdevice.h
```

```
+++ b/include/linux/netdevice.h
@@ -996,7 +996,7 @@ static inline void netif_stop_subqueue(struct net_device *dev, u16
queue_index)
  *
  * Check individual transmit queue of a device with multiple transmit queues.
  */
-static inline int netif_subqueue_stopped(const struct net_device *dev,
+static inline int __netif_subqueue_stopped(const struct net_device *dev,
       u16 queue_index)
 {
 #ifdef CONFIG_NETDEVICES_MULTIQUEUE
@@ -1007,6 +1007,11 @@ static inline int netif_subqueue_stopped(const struct net_device *dev,
 #endif
 }

+static inline int netif_subqueue_stopped(const struct net_device *dev,
+      struct sk_buff *skb)
+{
+ return __netif_subqueue_stopped(dev, skb_get_queue_mapping(skb));
+}

 /**
  * netif_wake_subqueue - allow sending packets on subqueue
diff --git a/net/core/dev.c b/net/core/dev.c
index 38b03da..8726589 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
@@ -1553,7 +1553,7 @@ gso:
    return rc;
   }
   if (unlikely((netif_queue_stopped(dev) ||
-        netif_subqueue_stopped(dev, skb->queue_mapping)) &&
+        netif_subqueue_stopped(dev, skb)) &&
        skb->next))
    return NETDEV_TX_BUSY;
  } while (skb->next);
@@ -1692,7 +1692,7 @@ gso:
    HARD_TX_LOCK(dev, cpu);

    if (!netif_queue_stopped(dev) &&
-       !netif_subqueue_stopped(dev, skb->queue_mapping)) {
+       !netif_subqueue_stopped(dev, skb)) {
     rc = 0;
     if (!dev_hard_start_xmit(skb, dev)) {
      HARD_TX_UNLOCK(dev);
diff --git a/net/core/netpoll.c b/net/core/netpoll.c
index 95daba6..bf8d18f 100644
--- a/net/core/netpoll.c
```

```
+++ b/net/core/netpoll.c
@@ -67,7 +67,7 @@ static void queue_process(struct work_struct *work)
   local_irq_save(flags);
   netif_tx_lock(dev);
   if ((netif_queue_stopped(dev) ||
-       netif_subqueue_stopped(dev, skb->queue_mapping)) ||
+       netif_subqueue_stopped(dev, skb)) ||
        dev->hard_start_xmit(skb, dev) != NETDEV_TX_OK) {
    skb_queue_head(&npinfo->txq, skb);
    netif_tx_unlock(dev);
@@ -269,7 +269,7 @@ static void netpoll_send_skb(struct netpoll *np, struct sk_buff *skb)
        tries > 0; --tries) {
    if (netif_tx_trylock(dev)) {
     if (!netif_queue_stopped(dev) &&
-        !netif_subqueue_stopped(dev, skb->queue_mapping))
+        !netif_subqueue_stopped(dev, skb))
      status = dev->hard_start_xmit(skb, dev);
     netif_tx_unlock(dev);

diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 8cae60c..d4d9a36 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -3385,7 +3383,7 @@ static __inline__ void pktgen_xmit(struct pktgen_dev *pkt_dev)

  if ((netif_queue_stopped(odev) ||
       (pkt_dev->skb &&
-       netif_subqueue_stopped(odev, pkt_dev->skb->queue_mapping))) ||
+       netif_subqueue_stopped(odev, pkt_dev->skb))) ||
      need_resched()) {
   idle_start = getCurUs();

@@ -3402,7 +3400,7 @@ static __inline__ void pktgen_xmit(struct pktgen_dev *pkt_dev)
   pkt_dev->idle_acc += getCurUs() - idle_start;

   if (netif_queue_stopped(odev) ||
-      netif_subqueue_stopped(odev, pkt_dev->skb->queue_mapping)) {
+      netif_subqueue_stopped(odev, pkt_dev->skb)) {
    pkt_dev->next_tx_us = getCurUs(); /* TODO */
    pkt_dev->next_tx_ns = 0;
    goto out; /* Try the next interface */
@@ -3431,7 +3429,7 @@ static __inline__ void pktgen_xmit(struct pktgen_dev *pkt_dev)

  netif_tx_lock_bh(odev);
  if (!netif_queue_stopped(odev) &&
-     !netif_subqueue_stopped(odev, pkt_dev->skb->queue_mapping)) {
+     !netif_subqueue_stopped(odev, pkt_dev->skb)) {
```

```
   atomic_inc(&(pkt_dev->skb->users));
        retry_now:
diff --git a/net/sched/sch_teql.c b/net/sched/sch_teql.c
index be57cf3..421281d 100644
--- a/net/sched/sch_teql.c
+++ b/net/sched/sch_teql.c
@@ -284,7 +284,7 @@ restart:
   if (slave->qdisc_sleeping != q)
    continue;
   if (netif_queue_stopped(slave) ||
-      netif_subqueue_stopped(slave, subq) ||
+       __netif_subqueue_stopped(slave, subq) ||
       !netif_running(slave)) {
    busy = 1;
    continue;
@@ -294,7 +294,7 @@ restart:
   case 0:
    if (netif_tx_trylock(slave)) {
     if (!netif_queue_stopped(slave) &&
-        !netif_subqueue_stopped(slave, subq) &&
+         !__netif_subqueue_stopped(slave, subq) &&
         slave->hard_start_xmit(skb, slave) == 0) {
      netif_tx_unlock(slave);
      master->slaves = NEXT_SLAVE(q);
```

Subject: [PATCH 4/4] Cut off the queue_mapping field from sk_buff
Posted by Pavel Emelianov on Fri, 19 Oct 2007 09:05:40 GMT
View Forum Message <> Reply to Message

Just hide it behind the #ifdef, because nobody wants
it now.

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

```
diff --git a/include/linux/skbuff.h b/include/linux/skbuff.h
index f93f22b..580ed1f 100644
--- a/include/linux/skbuff.h
+++ b/include/linux/skbuff.h
@@ -301,8 +301,9 @@ struct sk_buff {
 #endif

  int   iif;
+#ifdef CONFIG_NETDEVICES_MULTIQUEUE
  __u16   queue_mapping;
-
```

```
+#endif
 #ifdef CONFIG_NET_SCHED
  __u16   tc_index; /* traffic control index */
 #ifdef CONFIG_NET_CLS_ACT
```

Subject: Re: [PATCH 1/4] Use the skb_set_queue_mapping where appropriate
Posted by davem on Mon, 22 Oct 2007 00:01:03 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Fri, 19 Oct 2007 13:00:52 +0400

> There's already such a helper to initialize this field.
> Use it.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Applied.

```
>     /* reset queue_mapping to zero */
> -   skb->queue_mapping = 0;
> +   skb_set_queue_mapping(skb, 0);
```

This right here is a great example why I add next to no comments to
the code I write. They are %99.999 of the time stating the obvious,
and where a comment is needed it's the code's fault and the code in
such cases could be rewritten to make what's going on more clear thus
making any commentary, again, redundant and stating the obvious.

Here the case is the former, it's of the form:

```
 /* Add one to 'a'! */
 a += 1;
```

Gee thanks for letting me know! How insightful! :-))

Subject: Re: [PATCH 2/4] Make and use skb_get_queue_mapping
Posted by davem on Mon, 22 Oct 2007 00:01:37 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Fri, 19 Oct 2007 13:02:31 +0400

> Make the helper for getting the field, symmetrical to
> the "set" one. Return 0 if CONFIG_NETDEVICES_MULTIQUEUE=n

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

Applied.

Subject: Re: [PATCH 3/4] Hide the queue_mapping field inside netif_subqueue_stopped
Posted by davem on Mon, 22 Oct 2007 00:02:05 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Fri, 19 Oct 2007 13:04:13 +0400

> Many places get the queue_mapping field from skb to pass it
> to the netif_subqueue_stopped() which will be 0 in any case.
>
> Make the helper that works with sk_buff
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Applied, thanks!

Subject: Re: [PATCH 4/4] Cut off the queue_mapping field from sk_buff
Posted by davem on Mon, 22 Oct 2007 00:02:40 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Fri, 19 Oct 2007 13:05:40 +0400

> Just hide it behind the #ifdef, because nobody wants
> it now.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Also applied, thanks!