Subject: [PATCH 7/14][VLAN]: Tag vlan_group with device, not ifindex.
Posted by Pavel Emelianov on Thu, 10 Apr 2008 14:54:00 GMT
View Forum Message <> Reply to Message

We'll have to lookup the vlan_group by two keys - ifindex and
net. Turning the vlan_group lookup key to struct net_device
pointer will make this process easier.

Besides, this will eliminate one more place in the networking,
that assumes that indexes are unique in the kernel.

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

---
 include/linux/if_vlan.h | 4 +++-
 net/8021q/vlan.c | 22 ++++++++++++----------
 2 files changed, 14 insertions(+), 12 deletions(-)

```
diff --git a/include/linux/if_vlan.h b/include/linux/if_vlan.h
index edd55af..15ace02 100644
--- a/include/linux/if_vlan.h
+++ b/include/linux/if_vlan.h
@@ -81,7 +81,9 @@ extern void vlan_ioctl_set(int (*hook)(struct net *, void __user *));
 #define VLAN_GROUP_ARRAY_PART_LEN
(VLAN_GROUP_ARRAY_LEN/VLAN_GROUP_ARRAY_SPLIT_PARTS)

 struct vlan_group {
- int real_dev_ifindex; /* The ifindex of the ethernet(like) device the vlan is attached to. */
+ struct net_device *real_dev; /* The ethernet(like) device
+         * the vlan is attached to.
+         */
   unsigned int  nr_vlans;
   struct hlist_node hlist; /* linked list */
   struct net_device **vlan_devices_arrays[VLAN_GROUP_ARRAY_SPLIT_PARTS];
diff --git a/net/8021q/vlan.c b/net/8021q/vlan.c
index 5975ec3..cf8d810 100644
--- a/net/8021q/vlan.c
+++ b/net/8021q/vlan.c
@@ -65,14 +65,14 @@ static inline unsigned int vlan_grp_hashfn(unsigned int idx)
 }

 /* Must be invoked with RCU read lock (no preempt) */
-static struct vlan_group *__vlan_find_group(int real_dev_ifindex)
+static struct vlan_group *__vlan_find_group(struct net_device *real_dev)
 {
   struct vlan_group *grp;
   struct hlist_node *n;
- int hash = vlan_grp_hashfn(real_dev_ifindex);
```

```
+ int hash = vlan_grp_hashfn(real_dev->ifindex);

  hlist_for_each_entry_rcu(grp, n, &vlan_group_hash[hash], hlist) {
-  if (grp->real_dev_ifindex == real_dev_ifindex)
+  if (grp->real_dev == real_dev)
     return grp;
  }

@@ -86,7 +86,7 @@ static struct vlan_group *__vlan_find_group(int real_dev_ifindex)
 struct net_device *__find_vlan_dev(struct net_device *real_dev,
        unsigned short VID)
 {
- struct vlan_group *grp = __vlan_find_group(real_dev->ifindex);
+ struct vlan_group *grp = __vlan_find_group(real_dev);

  if (grp)
   return vlan_group_get_device(grp, VID);
@@ -103,7 +103,7 @@ static void vlan_group_free(struct vlan_group *grp)
  kfree(grp);
 }

-static struct vlan_group *vlan_group_alloc(int ifindex)
+static struct vlan_group *vlan_group_alloc(struct net_device *real_dev)
 {
  struct vlan_group *grp;

@@ -111,9 +111,9 @@ static struct vlan_group *vlan_group_alloc(int ifindex)
  if (!grp)
   return NULL;

- grp->real_dev_ifindex = ifindex;
+ grp->real_dev = real_dev;
  hlist_add_head_rcu(&grp->hlist,
-      &vlan_group_hash[vlan_grp_hashfn(ifindex)]);
+   &vlan_group_hash[vlan_grp_hashfn(real_dev->ifindex)]);
  return grp;
 }

@@ -151,7 +151,7 @@ void unregister_vlan_dev(struct net_device *dev)

  ASSERT_RTNL();

- grp = __vlan_find_group(real_dev->ifindex);
+ grp = __vlan_find_group(real_dev);
  BUG_ON(!grp);

  vlan_proc_rem_dev(dev);
@@ -246,9 +246,9 @@ int register_vlan_dev(struct net_device *dev)
```

```
  struct vlan_group *grp, *ngrp = NULL;
  int err;

- grp = __vlan_find_group(real_dev->ifindex);
+ grp = __vlan_find_group(real_dev);
  if (!grp) {
-  ngrp = grp = vlan_group_alloc(real_dev->ifindex);
+  ngrp = grp = vlan_group_alloc(real_dev);
   if (!grp)
    return -ENOBUFS;
  }
@@ -412,7 +412,7 @@ static int vlan_device_event(struct notifier_block *unused, unsigned long
event,
   goto out;
  }

- grp = __vlan_find_group(dev->ifindex);
+ grp = __vlan_find_group(dev);
  if (!grp)
   goto out;

--
1.5.3.4
```

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