Subject: [PATCH][NEIGH]: Fix race between neighbor lookup and table's hash_rnd update.
Posted by Pavel Emelianov on Fri, 22 Feb 2008 09:37:03 GMT
View Forum Message <> Reply to Message

The neigh_hash_grow() may update the tbl->hash_rnd value, which
is used in all tbl->hash callbacks to calculate the hashval.

Two lookup routines may race with this, since they call the
->hash callback without the tbl->lock held. Since the hash_rnd
is changed with this lock write-locked moving the calls to ->hash
under this lock read-locked closes this gap.

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

```
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 4062b88..2328acb 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -358,11 +358,12 @@ struct neighbour *neigh_lookup(struct neigh_table *tbl, const void
*pkey,
 {
  struct neighbour *n;
  int key_len = tbl->key_len;
- u32 hash_val = tbl->hash(pkey, dev);
+ u32 hash_val;

  NEIGH_CACHE_STAT_INC(tbl, lookups);

  read_lock_bh(&tbl->lock);
+ hash_val = tbl->hash(pkey, dev);
  for (n = tbl->hash_buckets[hash_val & tbl->hash_mask]; n; n = n->next) {
   if (dev == n->dev && !memcmp(n->primary_key, pkey, key_len)) {
    neigh_hold(n);
@@ -379,11 +380,12 @@ struct neighbour *neigh_lookup_nodev(struct neigh_table *tbl, struct
net *net,
 {
  struct neighbour *n;
  int key_len = tbl->key_len;
- u32 hash_val = tbl->hash(pkey, NULL);
+ u32 hash_val;

  NEIGH_CACHE_STAT_INC(tbl, lookups);

  read_lock_bh(&tbl->lock);
+ hash_val = tbl->hash(pkey, NULL);
```

```
for (n = tbl->hash_buckets[hash_val & tbl->hash_mask]; n; n = n->next) {
 if (!memcmp(n->primary_key, pkey, key_len) &&
     (net == n->dev->nd_net)) {
```