Subject: [PATCH net-2.6.25] [ARP] Remove forward declaration of neigh_changeaddr.
Posted by den on Mon, 14 Jan 2008 13:48:23 GMT
View Forum Message <> Reply to Message

No need for this. It is declared in the neighbour.h

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/core/neighbour.c | 1 -
 1 files changed, 0 insertions(+), 1 deletions(-)

diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 2eab6a5..9b0b773 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -59,7 +59,6 @@ static void neigh_timer_handler(unsigned long arg);
 static void __neigh_notify(struct neighbour *n, int type, int flags);
 static void neigh_update_notify(struct neighbour *neigh);
 static int pneigh_ifdown(struct neigh_table *tbl, struct net_device *dev);
-void neigh_changeaddr(struct neigh_table *tbl, struct net_device *dev);

 static struct neigh_table *neigh_tables;
 #ifdef CONFIG_PROC_FS
--
1.5.3.rc5

Subject: [PATCH net-2.6.25] [ARP] neigh_parms_put(destroy) are essentially local to core/neighbour.c.
Posted by den on Mon, 14 Jan 2008 13:48:24 GMT
View Forum Message <> Reply to Message

Make them static.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/neighbour.h | 7 -------
 net/core/neighbour.c | 11 ++++++++++-
 2 files changed, 10 insertions(+), 8 deletions(-)

diff --git a/include/net/neighbour.h b/include/net/neighbour.h
index a0d42a5..ebbfb50 100644
--- a/include/net/neighbour.h
+++ b/include/net/neighbour.h
@@ -213,7 +213,6 @@ extern struct neighbour *neigh_event_ns(struct neigh_table *tbl,

 extern struct neigh_parms *neigh_parms_alloc(struct net_device *dev, struct neigh_table *tbl);

```
 extern void   neigh_parms_release(struct neigh_table *tbl, struct neigh_parms *parms);
-extern void   neigh_parms_destroy(struct neigh_parms *parms);
 extern unsigned long  neigh_rand_reach_time(unsigned long base);

 extern void   pneigh_enqueue(struct neigh_table *tbl, struct neigh_parms *p,
@@ -254,12 +253,6 @@ static inline void __neigh_parms_put(struct neigh_parms *parms)
  atomic_dec(&parms->refcnt);
 }

-static inline void neigh_parms_put(struct neigh_parms *parms)
-{
- if (atomic_dec_and_test(&parms->refcnt))
-  neigh_parms_destroy(parms);
-}
-
 static inline struct neigh_parms *neigh_parms_clone(struct neigh_parms *parms)
 {
  atomic_inc(&parms->refcnt);
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 9b0b773..41394db 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -55,6 +55,8 @@

 #define PNEIGH_HASHMASK  0xF

+static inline void neigh_parms_put(struct neigh_parms *parms);
+
 static void neigh_timer_handler(unsigned long arg);
 static void __neigh_notify(struct neighbour *n, int type, int flags);
 static void neigh_update_notify(struct neighbour *neigh);
@@ -1348,7 +1350,7 @@ void neigh_parms_release(struct neigh_table *tbl, struct neigh_parms
*parms)
  NEIGH_PRINTK1("neigh_parms_release: not found\n");
 }

-void neigh_parms_destroy(struct neigh_parms *parms)
+static void neigh_parms_destroy(struct neigh_parms *parms)
 {
  release_net(parms->net);
  if (parms->dev)
@@ -1356,6 +1358,13 @@ void neigh_parms_destroy(struct neigh_parms *parms)
  kfree(parms);
 }

+static inline void neigh_parms_put(struct neigh_parms *parms)
+{
+ if (atomic_dec_and_test(&parms->refcnt))
```

```
+  neigh_parms_destroy(parms);
+}
+
+
 static struct lock_class_key neigh_table_proxy_queue_class;

 void neigh_table_init_no_netlink(struct neigh_table *tbl)
--
1.5.3.rc5
```