
Subject: Re: [PATCH 6/8 net-2.6.25] [ARP] neigh_parms_put(destroy) are essentially local to core/neighbour.c.

Posted by [davem](#) on Tue, 15 Jan 2008 07:04:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: "Denis V. Lunev" <den@openvz.org>

Date: Mon, 14 Jan 2008 17:59:59 +0300

> Make them static.

>

> Signed-off-by: Denis V. Lunev <den@openvz.org>

It's completely awkward to put the inline after all the call sites. A non-global-optimizing compiler won't be able to even inline it.

I've reworked this patch as follows when applying it.

```
diff --git a/include/net/neighbour.h b/include/net/neighbour.h
```

```
index a0d42a5..ebbf50 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..7cc772a 100644
```

```
--- a/net/core/neighbour.c
```

```
+++ b/net/core/neighbour.c
```

```

@@ -577,6 +577,13 @@ static int neigh_ifdown(struct neigh_table *tbl, struct net_device *dev)
    return -ENOENT;
}

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

/*
 * neighbour must already be out of the table;
@@ -1348,7 +1355,7 @@ 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)

```
