Subject: [patch 23/38][IPV6] rt6_stats - make the rt6_stats relative to the namespace
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:16:59 GMT
View Forum Message <> Reply to Message

This patch moves the rt6_stats structure inside the network
namespace structure. That allows to reference the rt6_stats
relatively from a network namespace.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/ipv6.h | 1 -
 include/net/net_namespace.h | 1 +
 net/ipv6/ip6_fib.c | 18 ++++++++----------
 net/ipv6/route.c | 12 +++++-------
 4 files changed, 16 insertions(+), 16 deletions(-)

Index: linux-2.6-netns/include/net/ipv6.h
===
--- linux-2.6-netns.orig/include/net/ipv6.h
+++ linux-2.6-netns/include/net/ipv6.h
@@ -620,7 +620,6 @@ extern void ipv6_misc_proc_exit(void);
 extern int snmp6_register_dev(struct inet6_dev *idev);
 extern int snmp6_unregister_dev(struct inet6_dev *idev);

-extern struct rt6_statistics *rt6_stats;
 #else
 static inline int snmp6_register_dev(struct inet6_dev *idev)
 {
Index: linux-2.6-netns/include/net/net_namespace.h
===
--- linux-2.6-netns.orig/include/net/net_namespace.h
+++ linux-2.6-netns/include/net/net_namespace.h
@@ -46,6 +46,7 @@ struct net {

  /* ipv6 routing table */
 #if defined(CONFIG_IPV6) || defined(CONFIG_IPV6_MODULE)
+ struct rt6_statistics   *rt6_stats;
   struct hlist_head   *fib_table_hash;
   struct fib6_table   *fib6_main_tbl;
 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
Index: linux-2.6-netns/net/ipv6/ip6_fib.c
===
--- linux-2.6-netns.orig/net/ipv6/ip6_fib.c
+++ linux-2.6-netns/net/ipv6/ip6_fib.c
@@ -48,8 +48,6 @@
 #define RT6_TRACE(x...) do { ; } while (0)

```
 #endif

-struct rt6_statistics *rt6_stats;
-
 static struct kmem_cache * fib6_node_kmem __read_mostly;

 enum fib_walk_state_t
@@ -650,10 +648,10 @@ static int fib6_add_rt2node(struct fib6_
  rt->rt6i_node = fn;
  atomic_inc(&rt->rt6i_ref);
  inet6_rt_notify(RTM_NEWROUTE, rt, info);
- rt6_stats->fib_rt_entries++;
+ init_net.rt6_stats->fib_rt_entries++;

  if ((fn->fn_flags & RTN_RTINFO) == 0) {
-  rt6_stats->fib_route_nodes++;
+  init_net.rt6_stats->fib_route_nodes++;
   fn->fn_flags |= RTN_RTINFO;
  }

@@ -1088,8 +1086,8 @@ static void fib6_del_route(struct fib6_n
  /* Unlink it */
  *rtp = rt->u.dst.rt6_next;
  rt->rt6i_node = NULL;
- rt6_stats->fib_rt_entries--;
- rt6_stats->fib_discarded_routes++;
+ init_net.rt6_stats->fib_rt_entries--;
+ init_net.rt6_stats->fib_discarded_routes++;

  /* Reset round-robin state, if necessary */
  if (fn->rr_ptr == rt)
@@ -1115,7 +1113,7 @@ static void fib6_del_route(struct fib6_n
  /* If it was last route, expunge its radix tree node */
  if (fn->leaf == NULL) {
   fn->fn_flags &= ~RTN_RTINFO;
-  rt6_stats->fib_route_nodes--;
+  init_net.rt6_stats->fib_route_nodes--;
   fn = fib6_repair_tree(fn);
  }

@@ -1544,8 +1542,8 @@ void __init fib6_init(void)
         0, SLAB_HWCACHE_ALIGN|SLAB_PANIC,
         NULL);

-   rt6_stats = kzalloc(sizeof(*rt6_stats), GFP_KERNEL);
-   if (!rt6_stats)
+   init_net.rt6_stats = kzalloc(sizeof(*init_net.rt6_stats), GFP_KERNEL);
+   if (!init_net.rt6_stats)
```

```
     panic("IPV6: failed to allocate rt6_stats.\n");

  register_pernet_subsys(&fib6_net_ops);
@@ -1555,6 +1553,6 @@ void __init fib6_init(void)
 void fib6_gc_cleanup(void)
 {
  unregister_pernet_subsys(&fib6_net_ops);
- kfree(rt6_stats);
+ kfree(init_net.rt6_stats);
  kmem_cache_destroy(fib6_node_kmem);
 }
Index: linux-2.6-netns/net/ipv6/route.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/route.c
+++ linux-2.6-netns/net/ipv6/route.c
@@ -2350,11 +2350,13 @@ static const struct file_operations ipv6
 static int rt6_stats_seq_show(struct seq_file *seq, void *v)
 {
  seq_printf(seq, "%04x %04x %04x %04x %04x %04x %04x\n",
-        rt6_stats->fib_nodes, rt6_stats->fib_route_nodes,
-        rt6_stats->fib_rt_alloc, rt6_stats->fib_rt_entries,
-        rt6_stats->fib_rt_cache,
-        atomic_read(&ip6_dst_ops.entries),
-        rt6_stats->fib_discarded_routes);
+     init_net.rt6_stats->fib_nodes,
+     init_net.rt6_stats->fib_route_nodes,
+     init_net.rt6_stats->fib_rt_alloc,
+     init_net.rt6_stats->fib_rt_entries,
+     init_net.rt6_stats->fib_rt_cache,
+     atomic_read(&ip6_dst_ops.entries),
+     init_net.rt6_stats->fib_discarded_routes);

  return 0;
 }

--
```