Subject: [patch 31/38][IPV6] rt6_info - make rt6_info accessed as a pointer
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:17:07 GMT
View Forum Message <> Reply to Message

This patch make mindless changes and prepares the code to use dynamic
allocation for rt6_info structure. The code accesses the rt6_info
structure as a pointer instead of a global static variable.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/ip6_route.h | 6 +++---
 net/ipv6/addrconf.c | 12 ++++++------
 net/ipv6/fib6_rules.c | 12 ++++++------
 net/ipv6/ip6_fib.c | 20 ++++++++++----------
 net/ipv6/route.c | 36 ++++++++++++++++++++++--------------
 5 files changed, 46 insertions(+), 40 deletions(-)

Index: linux-2.6-netns/include/net/ip6_route.h
===
--- linux-2.6-netns.orig/include/net/ip6_route.h
+++ linux-2.6-netns/include/net/ip6_route.h
@@ -36,11 +36,11 @@ struct route_info {
 #define RT6_LOOKUP_F_REACHABLE 0x2
 #define RT6_LOOKUP_F_HAS_SADDR 0x4

-extern struct rt6_info ip6_null_entry;
+extern struct rt6_info *ip6_null_entry;

 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
-extern struct rt6_info ip6_prohibit_entry;
-extern struct rt6_info ip6_blk_hole_entry;
+extern struct rt6_info *ip6_prohibit_entry;
+extern struct rt6_info *ip6_blk_hole_entry;
 #endif

 extern int ip6_rt_gc_interval;
Index: linux-2.6-netns/net/ipv6/addrconf.c
===
--- linux-2.6-netns.orig/net/ipv6/addrconf.c
+++ linux-2.6-netns/net/ipv6/addrconf.c
@@ -4265,13 +4265,13 @@ int __init addrconf_init(void)
  if (err)
   return err;

- ip6_null_entry.u.dst.dev = init_net.loopback_dev;
- ip6_null_entry.rt6i_idev = in6_dev_get(init_net.loopback_dev);
+ ip6_null_entry->u.dst.dev = init_net.loopback_dev;

```
+ ip6_null_entry->rt6i_idev = in6_dev_get(init_net.loopback_dev);
 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
- ip6_prohibit_entry.u.dst.dev = init_net.loopback_dev;
- ip6_prohibit_entry.rt6i_idev = in6_dev_get(init_net.loopback_dev);
- ip6_blk_hole_entry.u.dst.dev = init_net.loopback_dev;
- ip6_blk_hole_entry.rt6i_idev = in6_dev_get(init_net.loopback_dev);
+ ip6_prohibit_entry->u.dst.dev = init_net.loopback_dev;
+ ip6_prohibit_entry->rt6i_idev = in6_dev_get(init_net.loopback_dev);
+ ip6_blk_hole_entry->u.dst.dev = init_net.loopback_dev;
+ ip6_blk_hole_entry->rt6i_idev = in6_dev_get(init_net.loopback_dev);
 #endif

  register_netdevice_notifier(&ipv6_dev_notf);
Index: linux-2.6-netns/net/ipv6/fib6_rules.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/fib6_rules.c
+++ linux-2.6-netns/net/ipv6/fib6_rules.c
@@ -45,8 +45,8 @@ struct dst_entry *fib6_rule_lookup(struc
  if (arg.result)
   return arg.result;

- dst_hold(&ip6_null_entry.u.dst);
- return &ip6_null_entry.u.dst;
+ dst_hold(&ip6_null_entry->u.dst);
+ return &ip6_null_entry->u.dst;
 }

 static int fib6_rule_action(struct fib_rule *rule, struct flowi *flp,
@@ -61,14 +61,14 @@ static int fib6_rule_action(struct fib_r
  case FR_ACT_TO_TBL:
   break;
  case FR_ACT_UNREACHABLE:
-  rt = &ip6_null_entry;
+  rt = ip6_null_entry;
   goto discard_pkt;
  default:
  case FR_ACT_BLACKHOLE:
-  rt = &ip6_blk_hole_entry;
+  rt = ip6_blk_hole_entry;
   goto discard_pkt;
  case FR_ACT_PROHIBIT:
-  rt = &ip6_prohibit_entry;
+  rt = ip6_prohibit_entry;
   goto discard_pkt;
  }

@@ -76,7 +76,7 @@ static int fib6_rule_action(struct fib_r
  if (table)
```

```
   rt = lookup(table, flp, flags);

- if (rt != &ip6_null_entry) {
+ if (rt != ip6_null_entry) {
   struct fib6_rule *r = (struct fib6_rule *)rule;

   /*
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
@@ -196,7 +196,7 @@ static struct fib6_table *fib6_alloc_tab
  table = kzalloc(sizeof(*table), GFP_ATOMIC);
  if (table != NULL) {
   table->tb6_id = id;
-  table->tb6_root.leaf = &ip6_null_entry;
+  table->tb6_root.leaf = ip6_null_entry;
   table->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;
  }

@@ -713,8 +713,8 @@ int fib6_add(struct fib6_node *root, str
    if (sfn == NULL)
     goto st_failure;

-   sfn->leaf = &ip6_null_entry;
-   atomic_inc(&ip6_null_entry.rt6i_ref);
+   sfn->leaf = ip6_null_entry;
+   atomic_inc(&ip6_null_entry->rt6i_ref);
    sfn->fn_flags = RTN_ROOT;
    sfn->fn_sernum = fib6_new_sernum();

@@ -773,7 +773,7 @@ out:
 #if RT6_DEBUG >= 2
    if (!pn->leaf) {
     BUG_TRAP(pn->leaf != NULL);
-    pn->leaf = &ip6_null_entry;
+    pn->leaf = ip6_null_entry;
    }
 #endif
    atomic_inc(&pn->leaf->rt6i_ref);
@@ -958,7 +958,7 @@ struct fib6_node * fib6_locate(struct fi
 static struct rt6_info * fib6_find_prefix(struct fib6_node *fn)
 {
  if (fn->fn_flags&RTN_ROOT)
-  return &ip6_null_entry;
+  return ip6_null_entry;

  while(fn) {
```

```
   if(fn->left)
@@ -1008,7 +1008,7 @@ static struct fib6_node * fib6_repair_tr
 #if RT6_DEBUG >= 2
    if (fn->leaf==NULL) {
     BUG_TRAP(fn->leaf);
-    fn->leaf = &ip6_null_entry;
+    fn->leaf = ip6_null_entry;
    }
 #endif
    atomic_inc(&fn->leaf->rt6i_ref);
@@ -1110,7 +1110,7 @@ static void fib6_del_route(struct fib6_n
  rt->u.dst.rt6_next = NULL;

  if (fn->leaf == NULL && fn->fn_flags&RTN_TL_ROOT)
-  fn->leaf = &ip6_null_entry;
+  fn->leaf = ip6_null_entry;

  /* If it was last route, expunge its radix tree node */
  if (fn->leaf == NULL) {
@@ -1153,7 +1153,7 @@ int fib6_del(struct rt6_info *rt, struct
   return -ENOENT;
  }
 #endif
- if (fn == NULL || rt == &ip6_null_entry)
+ if (fn == NULL || rt == ip6_null_entry)
   return -ENOENT;

  BUG_TRAP(fn->fn_flags&RTN_RTINFO);
@@ -1498,7 +1498,7 @@ static int fib6_net_init(struct net *net
   goto out_fib6_main_tbl;

  net->fib6_main_tbl->tb6_id = RT6_TABLE_MAIN;
- net->fib6_main_tbl->tb6_root.leaf = &ip6_null_entry;
+ net->fib6_main_tbl->tb6_root.leaf = ip6_null_entry;
  net->fib6_main_tbl->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;

 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
@@ -1508,7 +1508,7 @@ static int fib6_net_init(struct net *net
   goto out_fib6_main_tbl;
  }
  net->fib6_local_tbl->tb6_id = RT6_TABLE_LOCAL;
- net->fib6_local_tbl->tb6_root.leaf = &ip6_null_entry;
+ net->fib6_local_tbl->tb6_root.leaf = ip6_null_entry;
  net->fib6_local_tbl->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;
 #endif

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
@@ -132,7 +132,7 @@ static struct dst_ops ip6_dst_blackhole_
  .entry_size  = sizeof(struct rt6_info),
 };

-struct rt6_info ip6_null_entry = {
+static struct rt6_info __ip6_null_entry = {
  .u = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
@@ -143,7 +143,7 @@ struct rt6_info ip6_null_entry = {
    .input  = ip6_pkt_discard,
    .output  = ip6_pkt_discard_out,
    .ops  = &ip6_dst_ops,
-   .path  = (struct dst_entry*)&ip6_null_entry,
+   .path  = (struct dst_entry*)&__ip6_null_entry,
   }
  },
  .rt6i_flags = (RTF_REJECT | RTF_NONEXTHOP),
@@ -151,12 +151,14 @@ struct rt6_info ip6_null_entry = {
  .rt6i_ref = ATOMIC_INIT(1),
 };

+struct rt6_info *ip6_null_entry = &__ip6_null_entry;
+
 #ifdef CONFIG_IPV6_MULTIPLE_TABLES

 static int ip6_pkt_prohibit(struct sk_buff *skb);
 static int ip6_pkt_prohibit_out(struct sk_buff *skb);

-struct rt6_info ip6_prohibit_entry = {
+struct rt6_info __ip6_prohibit_entry = {
  .u = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
@@ -167,7 +169,7 @@ struct rt6_info ip6_prohibit_entry = {
    .input  = ip6_pkt_prohibit,
    .output  = ip6_pkt_prohibit_out,
    .ops  = &ip6_dst_ops,
-   .path  = (struct dst_entry*)&ip6_prohibit_entry,
+   .path  = (struct dst_entry*)&__ip6_prohibit_entry,
   }
  },
  .rt6i_flags = (RTF_REJECT | RTF_NONEXTHOP),
@@ -175,7 +177,9 @@ struct rt6_info ip6_prohibit_entry = {
  .rt6i_ref = ATOMIC_INIT(1),
 };
```

```
-struct rt6_info ip6_blk_hole_entry = {
+struct rt6_info *ip6_prohibit_entry = &__ip6_prohibit_entry;
+
+static struct rt6_info __ip6_blk_hole_entry = {
  .u = {
   .dst = {
    .__refcnt = ATOMIC_INIT(1),
@@ -186,7 +190,7 @@ struct rt6_info ip6_blk_hole_entry = {
    .input  = dst_discard,
    .output  = dst_discard,
    .ops  = &ip6_dst_ops,
-   .path  = (struct dst_entry*)&ip6_blk_hole_entry,
+   .path  = (struct dst_entry*)&__ip6_blk_hole_entry,
   }
  },
  .rt6i_flags = (RTF_REJECT | RTF_NONEXTHOP),
@@ -194,6 +198,8 @@ struct rt6_info ip6_blk_hole_entry = {
  .rt6i_ref = ATOMIC_INIT(1),
 };

+struct rt6_info *ip6_blk_hole_entry = &__ip6_blk_hole_entry;
+
 #endif

 /* allocate dst with ip6_dst_ops */
@@ -276,7 +282,7 @@ static __inline__ struct rt6_info *rt6_d
    return local;

   if (strict)
-   return &ip6_null_entry;
+   return ip6_null_entry;
  }
  return rt;
 }
@@ -437,7 +443,7 @@ static struct rt6_info *rt6_select(struc
  RT6_TRACE("%s() => %p\n",
    __FUNCTION__, match);

- return (match ? match : &ip6_null_entry);
+ return (match ? match : ip6_null_entry);
 }

 #ifdef CONFIG_IPV6_ROUTE_INFO
@@ -522,7 +528,7 @@ int rt6_route_rcv(struct net_device *dev

 #define BACKTRACK(saddr) \
 do { \
```

```
- if (rt == &ip6_null_entry) { \
+ if (rt == ip6_null_entry) { \
   struct fib6_node *pn; \
   while (1) { \
    if (fn->fn_flags & RTN_TL_ROOT) \
@@ -691,7 +697,7 @@ restart_2:
 restart:
  rt = rt6_select(fn, oif, strict | reachable);
  BACKTRACK(&fl->fl6_src);
- if (rt == &ip6_null_entry ||
+ if (rt == ip6_null_entry ||
      rt->rt6i_flags & RTF_CACHE)
   goto out;

@@ -709,7 +715,7 @@ restart:
  }

  dst_release(&rt->u.dst);
- rt = nrt ? : &ip6_null_entry;
+ rt = nrt ? : ip6_null_entry;

  dst_hold(&rt->u.dst);
  if (nrt) {
@@ -1259,7 +1265,7 @@ static int __ip6_del_rt(struct rt6_info
  int err;
  struct fib6_table *table;

- if (rt == &ip6_null_entry)
+ if (rt == ip6_null_entry)
   return -ENOENT;

  table = rt->rt6i_table;
@@ -1374,7 +1380,7 @@ restart:
  }

  if (!rt)
-  rt = &ip6_null_entry;
+  rt = ip6_null_entry;
  BACKTRACK(&fl->fl6_src);
 out:
  dst_hold(&rt->u.dst);
@@ -1419,7 +1425,7 @@ void rt6_redirect(struct in6_addr *dest,

  rt = ip6_route_redirect(dest, src, saddr, neigh->dev);

- if (rt == &ip6_null_entry) {
+ if (rt == ip6_null_entry) {
   if (net_ratelimit())
```

```
    printk(KERN_DEBUG "rt6_redirect: source isn't a valid nexthop "
           "for redirect target\n");
@@ -1889,7 +1895,7 @@ struct rt6_info *addrconf_dst_alloc(stru
 static int fib6_ifdown(struct rt6_info *rt, void *arg)
 {
  if (((void*)rt->rt6i_dev == arg || arg == NULL) &&
-     rt != &ip6_null_entry) {
+     rt != ip6_null_entry) {
   RT6_TRACE("deleted by ifdown %p\n", rt);
   return -1;
  }

--
```

_____

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