Subject: [patch 13/38][IPV6] ip6_fib - make the fib table per network namespace
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:16:49 GMT
View Forum Message <> Reply to Message

The patch makes the ip6 fib being accessed relatively
to the network namespace.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/ip6_fib.h | 4 ++--
 net/ipv6/fib6_rules.c | 4 ++--
 net/ipv6/ip6_fib.c | 46 ++++++++++++++++++++++++++----------------
 net/ipv6/route.c | 12 ++++++------
 4 files changed, 31 insertions(+), 35 deletions(-)

Index: linux-2.6-netns/include/net/ip6_fib.h
===
--- linux-2.6-netns.orig/include/net/ip6_fib.h
+++ linux-2.6-netns/include/net/ip6_fib.h
@@ -194,8 +194,8 @@ typedef struct rt6_info *(*pol_lookup_t)
  * exported functions
  */

-extern struct fib6_table * fib6_get_table(u32 id);
-extern struct fib6_table * fib6_new_table(u32 id);
+extern struct fib6_table * fib6_get_table(struct net *net, u32 id);
+extern struct fib6_table * fib6_new_table(struct net *net, u32 id);
 extern struct dst_entry * fib6_rule_lookup(struct flowi *fl, int flags,
        pol_lookup_t lookup);

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
@@ -71,7 +71,7 @@ static int fib6_rule_action(struct fib_r
   goto discard_pkt;
  }

- table = fib6_get_table(rule->table);
+ table = fib6_get_table(&init_net, rule->table);
  if (table)
   rt = lookup(table, flp, flags);

@@ -151,7 +151,7 @@ static int fib6_rule_configure(struct fi
   if (rule->table == RT6_TABLE_UNSPEC)
    goto errout;

```
-  if (fib6_new_table(rule->table) == NULL) {
+  if (fib6_new_table(&init_net, rule->table) == NULL) {
     err = -ENOBUFS;
     goto errout;
   }
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
@@ -172,7 +172,7 @@ static __inline__ void rt6_release(struc
 #define FIB_TABLE_HASHSZ 1
 #endif

-static void fib6_link_table(struct fib6_table *tb)
+static void fib6_link_table(struct net *net, struct fib6_table *tb)
 {
  unsigned int h;

@@ -188,7 +188,7 @@ static void fib6_link_table(struct fib6_
    * No protection necessary, this is the only list mutatation
    * operation, tables never disappear once they exist.
    */
- hlist_add_head_rcu(&tb->tb6_hlist, &init_net.fib_table_hash[h]);
+ hlist_add_head_rcu(&tb->tb6_hlist, &net->fib_table_hash[h]);
 }

 #ifdef CONFIG_IPV6_MULTIPLE_TABLES
@@ -207,24 +207,24 @@ static struct fib6_table *fib6_alloc_tab
  return table;
 }

-struct fib6_table *fib6_new_table(u32 id)
+struct fib6_table *fib6_new_table(struct net *net, u32 id)
 {
  struct fib6_table *tb;

  if (id == 0)
   id = RT6_TABLE_MAIN;
- tb = fib6_get_table(id);
+ tb = fib6_get_table(net, id);
  if (tb)
   return tb;

  tb = fib6_alloc_table(id);
  if (tb != NULL)
-  fib6_link_table(tb);
+  fib6_link_table(net, tb);
```

```
  return tb;
 }

-struct fib6_table *fib6_get_table(u32 id)
+struct fib6_table *fib6_get_table(struct net *net, u32 id)
 {
  struct fib6_table *tb;
  struct hlist_head *head;
@@ -235,7 +235,7 @@ struct fib6_table *fib6_get_table(u32 id
   id = RT6_TABLE_MAIN;
  h = id & (FIB_TABLE_HASHSZ - 1);
  rcu_read_lock();
- head = &init_net.fib_table_hash[h];
+ head = &net->fib_table_hash[h];
  hlist_for_each_entry_rcu(tb, node, head, tb6_hlist) {
   if (tb->tb6_id == id) {
    rcu_read_unlock();
@@ -247,33 +247,33 @@ struct fib6_table *fib6_get_table(u32 id
  return NULL;
 }

-static void __init fib6_tables_init(void)
+static void __init fib6_tables_init(struct net *net)
 {
- fib6_link_table(init_net.fib6_main_tbl);
- fib6_link_table(init_net.fib6_local_tbl);
+ fib6_link_table(net, net->fib6_main_tbl);
+ fib6_link_table(net, net->fib6_local_tbl);
 }

 #else

-struct fib6_table *fib6_new_table(u32 id)
+struct fib6_table *fib6_new_table(struct net *net, u32 id)
 {
- return fib6_get_table(id);
+ return fib6_get_table(net, id);
 }

-struct fib6_table *fib6_get_table(u32 id)
+struct fib6_table *fib6_get_table(struct net *net, u32 id)
 {
- return init_net.fib6_main_tbl;
+ return net->fib6_main_tbl;
 }

 struct dst_entry *fib6_rule_lookup(struct flowi *fl, int flags,
       pol_lookup_t lookup)
```

```
 {
- return (struct dst_entry *) lookup(init_net.fib6_main_tbl, fl, flags);
+ return (struct dst_entry *) lookup(fl->fl_net->fib6_main_tbl, fl, flags);
 }

-static void __init fib6_tables_init(void)
+static void __init fib6_tables_init(struct net *net)
 {
- fib6_link_table(init_net.fib6_main_tbl);
+ fib6_link_table(net, net->fib6_main_tbl);
 }

 #endif
@@ -357,9 +357,6 @@ static int inet6_dump_fib(struct sk_buff
  struct hlist_head *head;
  int res = 0;

- if (net != &init_net)
-  return 0;
-
  s_h = cb->args[0];
  s_e = cb->args[1];

@@ -388,7 +385,7 @@ static int inet6_dump_fib(struct sk_buff

  for (h = s_h; h < FIB_TABLE_HASHSZ; h++, s_e = 0) {
   e = 0;
-  head = &init_net.fib_table_hash[h];
+  head = &net->fib_table_hash[h];
   hlist_for_each_entry(tb, node, head, tb6_hlist) {
    if (e < s_e)
     goto next;
@@ -1471,9 +1468,6 @@ static int fib6_net_init(struct net *net
 {
  int ret;

- if (net != &init_net)
-  return -EPERM;
-
  ret = -ENOMEM;
  net->fib_table_hash = kzalloc(sizeof(*net->fib_table_hash)*FIB_TABLE_HASHSZ,
         GFP_KERNEL);
@@ -1499,7 +1493,9 @@ static int fib6_net_init(struct net *net
  net->fib6_local_tbl->tb6_root.fn_flags = RTN_ROOT | RTN_TL_ROOT | RTN_RTINFO;
 #endif

- fib6_tables_init();
+ fib6_tables_init(net);
```

```
+
+ return 0;

 out_fib6_main_tbl:
  kfree(net->fib_table_hash);
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
@@ -1075,7 +1075,7 @@ int ip6_route_add(struct fib6_config *cf
  if (cfg->fc_metric == 0)
   cfg->fc_metric = IP6_RT_PRIO_USER;

- table = fib6_new_table(cfg->fc_table);
+ table = fib6_new_table(&init_net, cfg->fc_table);
  if (table == NULL) {
   err = -ENOBUFS;
   goto out;
@@ -1283,7 +1283,7 @@ static int ip6_route_del(struct fib6_con
  struct rt6_info *rt;
  int err = -ESRCH;

- table = fib6_get_table(cfg->fc_table);
+ table = fib6_get_table(&init_net, cfg->fc_table);
  if (table == NULL)
   return err;

@@ -1598,7 +1598,7 @@ static struct rt6_info *rt6_get_route_in
  struct rt6_info *rt = NULL;
  struct fib6_table *table;

- table = fib6_get_table(RT6_TABLE_INFO);
+ table = fib6_get_table(&init_net, RT6_TABLE_INFO);
  if (table == NULL)
   return NULL;

@@ -1653,7 +1653,7 @@ struct rt6_info *rt6_get_dflt_router(str
  struct rt6_info *rt;
  struct fib6_table *table;

- table = fib6_get_table(RT6_TABLE_DFLT);
+ table = fib6_get_table(&init_net, RT6_TABLE_DFLT);
  if (table == NULL)
   return NULL;

@@ -1695,7 +1695,7 @@ void rt6_purge_dflt_routers(void)
  struct fib6_table *table;
```

```
  /* NOTE: Keep consistent with rt6_get_dflt_router */
- table = fib6_get_table(RT6_TABLE_DFLT);
+ table = fib6_get_table(&init_net, RT6_TABLE_DFLT);
  if (table == NULL)
   return;

@@ -1857,7 +1857,7 @@ struct rt6_info *addrconf_dst_alloc(stru

  ipv6_addr_copy(&rt->rt6i_dst.addr, addr);
  rt->rt6i_dst.plen = 128;
- rt->rt6i_table = fib6_get_table(RT6_TABLE_LOCAL);
+ rt->rt6i_table = fib6_get_table(&init_net, RT6_TABLE_LOCAL);

  atomic_set(&rt->u.dst.__refcnt, 1);


--
```

_____

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