Subject: [patch 14/38][IPV6] ip6_fib - make fib6_clean_all per namespace
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:16:50 GMT
View Forum Message <> Reply to Message

The function fib6_clean_all takes the network namespace
as parameter. This is useful for example to flush the routes
related to a specific 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 | 3 ++-
 net/ipv6/ip6_fib.c | 7 ++++---
 net/ipv6/route.c | 22 ++++++++++++++++++-----
 3 files changed, 23 insertions(+), 9 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
@@ -207,7 +207,8 @@ struct fib6_node *fib6_locate(struct fi
          struct in6_addr *daddr, int dst_len,
          struct in6_addr *saddr, int src_len);

-extern void   fib6_clean_all(int (*func)(struct rt6_info *, void *arg),
+extern void   fib6_clean_all(struct net *net,
+   int (*func)(struct rt6_info *, void *arg),
           int prune, void *arg);

 extern int   fib6_add(struct fib6_node *root,
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
@@ -1354,7 +1354,7 @@ static void fib6_clean_tree(struct fib6_
  fib6_walk(&c.w);
 }

-void fib6_clean_all(int (*func)(struct rt6_info *, void *arg),
+void fib6_clean_all(struct net *net, int (*func)(struct rt6_info *, void *arg),
       int prune, void *arg)
 {
  struct fib6_table *table;
@@ -1364,7 +1364,7 @@ void fib6_clean_all(int (*func)(struct r

  rcu_read_lock();
  for (h = 0; h < FIB_TABLE_HASHSZ; h++) {
- head = &init_net.fib_table_hash[h];

```
+  head = &net->fib_table_hash[h];
   hlist_for_each_entry_rcu(table, node, head, tb6_hlist) {
    write_lock_bh(&table->tb6_lock);
    fib6_clean_tree(&table->tb6_root, func, prune, arg);
@@ -1453,7 +1453,8 @@ void fib6_run_gc(unsigned long dummy)
  gc_args.more = 0;

  ndisc_dst_gc(&gc_args.more);
- fib6_clean_all(fib6_age, 0, NULL);
+
+ fib6_clean_all(&init_net, fib6_age, 0, NULL);

  if (gc_args.more)
   mod_timer(&ip6_fib_timer, jiffies + ip6_rt_gc_interval);
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
@@ -1876,7 +1876,7 @@ static int fib6_ifdown(struct rt6_info *

 void rt6_ifdown(struct net_device *dev)
 {
- fib6_clean_all(fib6_ifdown, 0, dev);
+ fib6_clean_all(dev->nd_net, fib6_ifdown, 0, dev);
 }

 struct rt6_mtu_change_arg
@@ -1932,7 +1932,7 @@ void rt6_mtu_change(struct net_device *d
   .mtu = mtu,
  };

- fib6_clean_all(rt6_mtu_change_route, 0, &arg);
+ fib6_clean_all(dev->nd_net, rt6_mtu_change_route, 0, &arg);
 }

 static const struct nla_policy rtm_ipv6_policy[RTA_MAX+1] = {
@@ -2318,13 +2318,25 @@ static int rt6_info_route(struct rt6_inf

 static int ipv6_route_show(struct seq_file *m, void *v)
 {
- fib6_clean_all(rt6_info_route, 0, m);
+ struct net *net = (struct net *)m->private;
+ fib6_clean_all(net, rt6_info_route, 0, m);
  return 0;
 }

 static int ipv6_route_open(struct inode *inode, struct file *file)
 {
```

```
- return single_open(file, ipv6_route_show, NULL);
+ struct net *net = get_proc_net(inode);
+ if (!net)
+  return -ENXIO;
+ return single_open(file, ipv6_route_show, net);
+}
+
+static int ipv6_route_release(struct inode *inode, struct file *file)
+{
+ struct seq_file *seq = file->private_data;
+ struct net *net = seq->private;
+ put_net(net);
+ return single_release(inode, file);
 }

 static const struct file_operations ipv6_route_proc_fops = {
@@ -2332,7 +2344,7 @@ static const struct file_operations ipv6
  .open  = ipv6_route_open,
  .read  = seq_read,
  .llseek  = seq_lseek,
- .release = single_release,
+ .release = ipv6_route_release,
 };

 static int rt6_stats_seq_show(struct seq_file *seq, void *v)

--
```

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