Subject: [patch 30/38][IPV6] route6 - make route6 per namespace
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:17:06 GMT
View Forum Message <> Reply to Message

This patch makes the routing engine use the network namespaces
to access routing informations:
Add a network namespace parameter to ipv6_route_ioctl and
propagate the network namespace value to all the routing code
that have not yet been changed.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/ip6_route.h | 4 ++
 net/ipv6/addrconf.c | 9 ++++++
 net/ipv6/af_inet6.c | 3 +-
 net/ipv6/route.c | 68 ++++++++++++++++++++++++++++++-----------------
 4 files changed, 49 insertions(+), 35 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
@@ -53,7 +53,9 @@ extern struct dst_entry * ip6_route_outp
 extern void   ip6_route_init(void);
 extern void   ip6_route_cleanup(void);

-extern int   ipv6_route_ioctl(unsigned int cmd, void __user *arg);
+extern int   ipv6_route_ioctl(struct net *net,
+   unsigned int cmd,
+   void __user *arg);

 extern int   ip6_route_add(struct fib6_config *cfg);
 extern int   ip6_ins_rt(struct rt6_info *);
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
@@ -1530,6 +1530,9 @@ addrconf_prefix_route(struct in6_addr *p
   .fc_expires = expires,
   .fc_dst_len = plen,
   .fc_flags = RTF_UP | flags,
+ .fc_nlinfo.pid = 0,
+ .fc_nlinfo.nlh = NULL,
+ .fc_nlinfo.net = &init_net,
  };

  ipv6_addr_copy(&cfg.fc_dst, pfx);

```
@@ -1556,6 +1559,9 @@ static void addrconf_add_mroute(struct n
   .fc_ifindex = dev->ifindex,
   .fc_dst_len = 8,
   .fc_flags = RTF_UP,
+  .fc_nlinfo.pid = 0,
+  .fc_nlinfo.nlh = NULL,
+  .fc_nlinfo.net = &init_net,
  };

  ipv6_addr_set(&cfg.fc_dst, htonl(0xFF000000), 0, 0, 0);
@@ -1572,6 +1578,9 @@ static void sit_route_add(struct net_dev
   .fc_ifindex = dev->ifindex,
   .fc_dst_len = 96,
   .fc_flags = RTF_UP | RTF_NONEXTHOP,
+  .fc_nlinfo.pid = 0,
+  .fc_nlinfo.nlh = NULL,
+  .fc_nlinfo.net = &init_net,
  };

  /* prefix length - 96 bits "::d.d.d.d" */
Index: linux-2.6-netns/net/ipv6/af_inet6.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/af_inet6.c
+++ linux-2.6-netns/net/ipv6/af_inet6.c
@@ -441,6 +441,7 @@ EXPORT_SYMBOL(inet6_getname);
 int inet6_ioctl(struct socket *sock, unsigned int cmd, unsigned long arg)
 {
  struct sock *sk = sock->sk;
+ struct net *net = sk->sk_net;

  switch(cmd)
  {
@@ -453,7 +454,7 @@ int inet6_ioctl(struct socket *sock, uns
  case SIOCADDRT:
  case SIOCDELRT:

-  return(ipv6_route_ioctl(cmd,(void __user *)arg));
+  return(ipv6_route_ioctl(net, cmd,(void __user *)arg));

  case SIOCSIFADDR:
   return addrconf_add_ifaddr((void __user *) arg);
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
@@ -610,10 +610,11 @@ static int __ip6_ins_rt(struct rt6_info

 int ip6_ins_rt(struct rt6_info *rt)
```

```
 {
+ struct net *net = rt->rt6i_dev->nd_net;
  struct nl_info info = {
   .nlh = NULL,
   .pid = 0,
-  .net = &init_net,
+  .net = net,
  };
  return __ip6_ins_rt(rt, &info);
 }
@@ -752,7 +753,7 @@ void ip6_route_input(struct sk_buff *skb
  struct ipv6hdr *iph = ipv6_hdr(skb);
  int flags = RT6_LOOKUP_F_HAS_SADDR;
  struct flowi fl = {
-  .fl_net = &init_net,
+  .fl_net = skb->dev->nd_net,
   .iif = skb->dev->ifindex,
   .nl_u = {
    .ip6_u = {
@@ -1055,6 +1056,7 @@ int ipv6_get_hoplimit(struct net_device
 int ip6_route_add(struct fib6_config *cfg)
 {
  int err;
+ struct net *net = cfg->fc_nlinfo.net;
  struct rt6_info *rt = NULL;
  struct net_device *dev = NULL;
  struct inet6_dev *idev = NULL;
@@ -1069,7 +1071,7 @@ int ip6_route_add(struct fib6_config *cf
 #endif
  if (cfg->fc_ifindex) {
   err = -ENODEV;
-  dev = dev_get_by_index(&init_net, cfg->fc_ifindex);
+  dev = dev_get_by_index(net, cfg->fc_ifindex);
   if (!dev)
    goto out;
   idev = in6_dev_get(dev);
@@ -1080,7 +1082,7 @@ int ip6_route_add(struct fib6_config *cf
  if (cfg->fc_metric == 0)
   cfg->fc_metric = IP6_RT_PRIO_USER;

- table = fib6_new_table(&init_net, cfg->fc_table);
+ table = fib6_new_table(net, cfg->fc_table);
  if (table == NULL) {
   err = -ENOBUFS;
   goto out;
@@ -1127,12 +1129,12 @@ int ip6_route_add(struct fib6_config *cf
  if ((cfg->fc_flags & RTF_REJECT) ||
      (dev && (dev->flags&IFF_LOOPBACK) && !(addr_type&IPV6_ADDR_LOOPBACK))) {
```

```
   /* hold loopback dev/idev if we haven't done so. */
-  if (dev != init_net.loopback_dev) {
+  if (dev != net->loopback_dev) {
    if (dev) {
     dev_put(dev);
     in6_dev_put(idev);
    }
-   dev = init_net.loopback_dev;
+   dev = net->loopback_dev;
    dev_hold(dev);
    idev = in6_dev_get(dev);
    if (!idev) {
@@ -1169,7 +1171,7 @@ int ip6_route_add(struct fib6_config *cf
    if (!(gwa_type&IPV6_ADDR_UNICAST))
     goto out;

-   grt = rt6_lookup(&init_net, gw_addr, NULL, cfg->fc_ifindex, 1);
+   grt = rt6_lookup(net, gw_addr, NULL, cfg->fc_ifindex, 1);

    err = -EHOSTUNREACH;
    if (grt == NULL)
@@ -1273,10 +1275,11 @@ static int __ip6_del_rt(struct rt6_info

 int ip6_del_rt(struct rt6_info *rt)
 {
+ struct net *net = rt->rt6i_dev->nd_net;
  struct nl_info info = {
   .nlh = NULL,
   .pid = 0,
-  .net = &init_net,
+  .net = net,
  };
  return __ip6_del_rt(rt, &info);
 }
@@ -1288,7 +1291,7 @@ static int ip6_route_del(struct fib6_con
  struct rt6_info *rt;
  int err = -ESRCH;

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

@@ -1389,7 +1392,7 @@ static struct rt6_info *ip6_route_redire
  int flags = RT6_LOOKUP_F_HAS_SADDR;
  struct ip6rd_flowi rdfl = {
   .fl = {
-   .fl_net = &init_net,
```

```
+    .fl_net = dev->nd_net,
     .oif = dev->ifindex,
     .nl_u = {
      .ip6_u = {
@@ -1663,7 +1666,7 @@ struct rt6_info *rt6_get_dflt_router(str
  struct rt6_info *rt;
  struct fib6_table *table;

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

@@ -1690,6 +1693,9 @@ struct rt6_info *rt6_add_dflt_router(str
    .fc_ifindex = dev->ifindex,
    .fc_flags = RTF_GATEWAY | RTF_ADDRCONF | RTF_DEFAULT |
        RTF_UP | RTF_EXPIRES | RTF_PREF(pref),
+   .fc_nlinfo.pid = 0,
+   .fc_nlinfo.nlh = NULL,
+   .fc_nlinfo.net = dev->nd_net,
  };

  ipv6_addr_copy(&cfg.fc_gateway, gwaddr);
@@ -1722,7 +1728,8 @@ restart:
  read_unlock_bh(&table->tb6_lock);
 }

-static void rtmsg_to_fib6_config(struct in6_rtmsg *rtmsg,
+static void rtmsg_to_fib6_config(struct net *net,
+      struct in6_rtmsg *rtmsg,
       struct fib6_config *cfg)
 {
  memset(cfg, 0, sizeof(*cfg));
@@ -1735,12 +1742,16 @@ static void rtmsg_to_fib6_config(struct
  cfg->fc_src_len = rtmsg->rtmsg_src_len;
  cfg->fc_flags = rtmsg->rtmsg_flags;

+ cfg->fc_nlinfo.pid = 0;
+ cfg->fc_nlinfo.nlh = NULL;
+ cfg->fc_nlinfo.net = net;
+
  ipv6_addr_copy(&cfg->fc_dst, &rtmsg->rtmsg_dst);
  ipv6_addr_copy(&cfg->fc_src, &rtmsg->rtmsg_src);
  ipv6_addr_copy(&cfg->fc_gateway, &rtmsg->rtmsg_gateway);
 }

-int ipv6_route_ioctl(unsigned int cmd, void __user *arg)
+int ipv6_route_ioctl(struct net *net, unsigned int cmd, void __user *arg)
```

```
 {
  struct fib6_config cfg;
  struct in6_rtmsg rtmsg;
@@ -1756,7 +1767,7 @@ int ipv6_route_ioctl(unsigned int cmd, v
    if (err)
     return -EFAULT;

-  rtmsg_to_fib6_config(&rtmsg, &cfg);
+   rtmsg_to_fib6_config(net, &rtmsg, &cfg);

    rtnl_lock();
    switch (cmd) {
@@ -1836,18 +1847,19 @@ struct rt6_info *addrconf_dst_alloc(stru
         const struct in6_addr *addr,
         int anycast)
 {
+ struct net *net = idev->dev->nd_net;
  struct rt6_info *rt = ip6_dst_alloc();

  if (rt == NULL)
   return ERR_PTR(-ENOMEM);

- dev_hold(init_net.loopback_dev);
+ dev_hold(net->loopback_dev);
  in6_dev_hold(idev);

  rt->u.dst.flags = DST_HOST;
  rt->u.dst.input = ip6_input;
  rt->u.dst.output = ip6_output;
- rt->rt6i_dev = init_net.loopback_dev;
+ rt->rt6i_dev = net->loopback_dev;
  rt->rt6i_idev = idev;
  rt->u.dst.metrics[RTAX_MTU-1] = ipv6_get_mtu(rt->rt6i_dev);
  rt->u.dst.metrics[RTAX_ADVMSS-1] = ipv6_advmss(dst_mtu(&rt->u.dst));
@@ -1867,7 +1879,7 @@ 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(&init_net, RT6_TABLE_LOCAL);
+ rt->rt6i_table = fib6_get_table(net, RT6_TABLE_LOCAL);

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

@@ -2025,13 +2037,9 @@ errout:

 static int inet6_rtm_delroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
- struct net *net = skb->sk->sk_net;
```

```
  struct fib6_config cfg;
  int err;

- if (net != &init_net)
-  return -EINVAL;
-
  err = rtm_to_fib6_config(skb, nlh, &cfg);
  if (err < 0)
   return err;
@@ -2041,13 +2049,9 @@ static int inet6_rtm_delroute(struct sk_

 static int inet6_rtm_newroute(struct sk_buff *skb, struct nlmsghdr* nlh, void *arg)
 {
- struct net *net = skb->sk->sk_net;
  struct fib6_config cfg;
  int err;

- if (net != &init_net)
-  return -EINVAL;
-
  err = rtm_to_fib6_config(skb, nlh, &cfg);
  if (err < 0)
   return err;
@@ -2190,16 +2194,13 @@ static int inet6_rtm_getroute(struct sk_
  struct flowi fl;
  int err, iif = 0;

- if (net != &init_net)
-  return -EINVAL;
-
  err = nlmsg_parse(nlh, sizeof(*rtm), tb, RTA_MAX, rtm_ipv6_policy);
  if (err < 0)
   goto errout;

  err = -EINVAL;
  memset(&fl, 0, sizeof(fl));
- fl.fl_net = &init_net;
+ fl.fl_net = net;

  if (tb[RTA_SRC]) {
   if (nla_len(tb[RTA_SRC]) < sizeof(struct in6_addr))
@@ -2223,7 +2224,7 @@ static int inet6_rtm_getroute(struct sk_

  if (iif) {
   struct net_device *dev;
-  dev = __dev_get_by_index(&init_net, iif);
+  dev = __dev_get_by_index(net, iif);
   if (!dev) {
```

```
    err = -ENODEV;
    goto errout;
@@ -2253,7 +2254,7 @@ static int inet6_rtm_getroute(struct sk_
   goto errout;
  }

- err = rtnl_unicast(skb, &init_net, NETLINK_CB(in_skb).pid);
+ err = rtnl_unicast(skb, net, NETLINK_CB(in_skb).pid);
 errout:
  return err;
 }
@@ -2263,6 +2264,7 @@ void inet6_rt_notify(int event, struct r
  struct sk_buff *skb;
  u32 pid = info->pid, seq = info->nlh ? info->nlh->nlmsg_seq : 0;
  struct nlmsghdr *nlh = info->nlh;
+ struct net *net = info->net;
  int err = -ENOBUFS;

  skb = nlmsg_new(rt6_nlmsg_size(), gfp_any());
@@ -2276,10 +2278,10 @@ void inet6_rt_notify(int event, struct r
   kfree_skb(skb);
   goto errout;
  }
- err = rtnl_notify(skb, &init_net, pid, RTNLGRP_IPV6_ROUTE, nlh, gfp_any());
+ err = rtnl_notify(skb, net, pid, RTNLGRP_IPV6_ROUTE, nlh, gfp_any());
 errout:
  if (err < 0)
-  rtnl_set_sk_err(&init_net, RTNLGRP_IPV6_ROUTE, err);
+  rtnl_set_sk_err(net, RTNLGRP_IPV6_ROUTE, err);
 }

 /*

--
```