Subject: [PATCH 7/17] [NETNS]: Disable multicaststing configuration inside namespace.
Posted by den on Wed, 06 Feb 2008 10:53:27 GMT
View Forum Message <> Reply to Message

Do not calls hooks from device notifiers and disallow configuration from
ioctl/netlink layer.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv4/igmp.c | 39 ++
 1 files changed, 39 insertions(+), 0 deletions(-)

diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 994648b..fe2e6cd 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -1199,6 +1199,9 @@ void ip_mc_inc_group(struct in_device *in_dev, __be32 addr)

  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+ return;
+
  for (im=in_dev->mc_list; im; im=im->next) {
   if (im->multiaddr == addr) {
    im->users++;
@@ -1278,6 +1281,9 @@ void ip_mc_dec_group(struct in_device *in_dev, __be32 addr)

  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+ return;
+
  for (ip=&in_dev->mc_list; (i=*ip)!=NULL; ip=&i->next) {
   if (i->multiaddr==addr) {
    if (--i->users == 0) {
@@ -1305,6 +1311,9 @@ void ip_mc_down(struct in_device *in_dev)

  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+ return;
+
  for (i=in_dev->mc_list; i; i=i->next)
   igmp_group_dropped(i);

@@ -1325,6 +1334,9 @@ void ip_mc_init_dev(struct in_device *in_dev)

```
 {
  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+  return;
+
  in_dev->mc_tomb = NULL;
 #ifdef CONFIG_IP_MULTICAST
  in_dev->mr_gq_running = 0;
@@ -1348,6 +1360,9 @@ void ip_mc_up(struct in_device *in_dev)

  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+  return;
+
  ip_mc_inc_group(in_dev, IGMP_ALL_HOSTS);

  for (i=in_dev->mc_list; i; i=i->next)
@@ -1364,6 +1379,9 @@ void ip_mc_destroy_dev(struct in_device *in_dev)

  ASSERT_RTNL();

+ if (in_dev->dev->nd_net != &init_net)
+  return;
+
  /* Deactivate timers */
  ip_mc_down(in_dev);

@@ -1745,6 +1763,9 @@ int ip_mc_join_group(struct sock *sk , struct ip_mreqn *imr)
  if (!ipv4_is_multicast(addr))
   return -EINVAL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();

  in_dev = ip_mc_find_dev(imr);
@@ -1813,6 +1834,9 @@ int ip_mc_leave_group(struct sock *sk, struct ip_mreqn *imr)
  u32 ifindex;
  int ret = -EADDRNOTAVAIL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();
  in_dev = ip_mc_find_dev(imr);
```

```
  ifindex = imr->imr_ifindex;
@@ -1858,6 +1882,9 @@ int ip_mc_source(int add, int omode, struct sock *sk, struct
  if (!ipv4_is_multicast(addr))
   return -EINVAL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();

  imr.imr_multiaddr.s_addr = mreqs->imr_multiaddr;
@@ -1991,6 +2018,9 @@ int ip_mc_msfilter(struct sock *sk, struct ip_msfilter *msf, int ifindex)
      msf->imsf_fmode != MCAST_EXCLUDE)
   return -EINVAL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();

  imr.imr_multiaddr.s_addr = msf->imsf_multiaddr;
@@ -2071,6 +2101,9 @@ int ip_mc_msfget(struct sock *sk, struct ip_msfilter *msf,
  if (!ipv4_is_multicast(addr))
   return -EINVAL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();

  imr.imr_multiaddr.s_addr = msf->imsf_multiaddr;
@@ -2133,6 +2166,9 @@ int ip_mc_gsfget(struct sock *sk, struct group_filter *gsf,
  if (!ipv4_is_multicast(addr))
   return -EINVAL;

+ if (sk->sk_net != &init_net)
+  return -EPROTONOSUPPORT;
+
  rtnl_lock();

  err = -EADDRNOTAVAIL;
@@ -2217,6 +2253,9 @@ void ip_mc_drop_socket(struct sock *sk)
  if (inet->mc_list == NULL)
   return;

+ if (sk->sk_net != &init_net)
+  return;
+
```

```
  rtnl_lock();
  while ((iml = inet->mc_list) != NULL) {
   struct in_device *in_dev;
--
1.5.3.rc5
```