Subject: [PATCH 4/6] [IPV4]: fib_sync_down rework.
Posted by den on Thu, 31 Jan 2008 12:00:48 GMT
View Forum Message <> Reply to Message

fib_sync_down can be called with an address and with a device. In reality
it is called either with address OR with a device. The codepath inside is
completely different, so lets separate it into two calls for these two
cases.

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 include/net/ip_fib.h | 3 +-
 net/ipv4/fib_frontend.c | 4 +-
 net/ipv4/fib_semantics.c | 104 ++++++++++++++++++++++++++----------------
 3 files changed, 57 insertions(+), 54 deletions(-)

diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h
index 9daa60b..1b2f008 100644
--- a/include/net/ip_fib.h
+++ b/include/net/ip_fib.h
@@ -218,7 +218,8 @@ extern void fib_select_default(struct net *net, const struct flowi *flp,

 /* Exported by fib_semantics.c */
 extern int ip_fib_check_default(__be32 gw, struct net_device *dev);
-extern int fib_sync_down(__be32 local, struct net_device *dev, int force);
+extern int fib_sync_down_dev(struct net_device *dev, int force);
+extern int fib_sync_down_addr(__be32 local);
 extern int fib_sync_up(struct net_device *dev);
 extern __be32 __fib_res_prefsrc(struct fib_result *res);
 extern void fib_select_multipath(const struct flowi *flp, struct fib_result *res);
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d0507f4..d69ffa2 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -808,7 +808,7 @@ static void fib_del_ifaddr(struct in_ifaddr *ifa)
       First of all, we scan fib_info list searching
       for stray nexthop entries, then ignite fib_flush.
    */
-   if (fib_sync_down(ifa->ifa_local, NULL, 0))
+   if (fib_sync_down_addr(ifa->ifa_local))
     fib_flush(dev->nd_net);
  }
 }
@@ -898,7 +898,7 @@ static void nl_fib_lookup_exit(struct net *net)

 static void fib_disable_ip(struct net_device *dev, int force)
 {
- if (fib_sync_down(0, dev, force))

```
+ if (fib_sync_down_dev(dev, force))
   fib_flush(dev->nd_net);
  rt_cache_flush(0);
  arp_ifdown(dev);
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index c791286..5beff2e 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -1031,70 +1031,72 @@ nla_put_failure:
       referring to it.
     - device went down -> we must shutdown all nexthops going via it.
   */
-
-int fib_sync_down(__be32 local, struct net_device *dev, int force)
+int fib_sync_down_addr(__be32 local)
 {
   int ret = 0;
- int scope = RT_SCOPE_NOWHERE;
-
- if (force)
-  scope = -1;
+ unsigned int hash = fib_laddr_hashfn(local);
+ struct hlist_head *head = &fib_info_laddrhash[hash];
+ struct hlist_node *node;
+ struct fib_info *fi;

- if (local && fib_info_laddrhash) {
-  unsigned int hash = fib_laddr_hashfn(local);
-  struct hlist_head *head = &fib_info_laddrhash[hash];
-  struct hlist_node *node;
-  struct fib_info *fi;
+ if (fib_info_laddrhash == NULL || local == 0)
+  return 0;

-  hlist_for_each_entry(fi, node, head, fib_lhash) {
-   if (fi->fib_prefsrc == local) {
-    fi->fib_flags |= RTNH_F_DEAD;
-    ret++;
-   }
+ hlist_for_each_entry(fi, node, head, fib_lhash) {
+  if (fi->fib_prefsrc == local) {
+   fi->fib_flags |= RTNH_F_DEAD;
+   ret++;
   }
  }
+ return ret;
+}
```

```
- if (dev) {
-  struct fib_info *prev_fi = NULL;
-  unsigned int hash = fib_devindex_hashfn(dev->ifindex);
-  struct hlist_head *head = &fib_info_devhash[hash];
-  struct hlist_node *node;
-  struct fib_nh *nh;
+int fib_sync_down_dev(struct net_device *dev, int force)
+{
+ int ret = 0;
+ int scope = RT_SCOPE_NOWHERE;
+ struct fib_info *prev_fi = NULL;
+ unsigned int hash = fib_devindex_hashfn(dev->ifindex);
+ struct hlist_head *head = &fib_info_devhash[hash];
+ struct hlist_node *node;
+ struct fib_nh *nh;

-  hlist_for_each_entry(nh, node, head, nh_hash) {
-   struct fib_info *fi = nh->nh_parent;
-   int dead;
+ if (force)
+  scope = -1;

-   BUG_ON(!fi->fib_nhs);
-   if (nh->nh_dev != dev || fi == prev_fi)
-    continue;
-   prev_fi = fi;
-   dead = 0;
-   change_nexthops(fi) {
-    if (nh->nh_flags&RTNH_F_DEAD)
-     dead++;
-    else if (nh->nh_dev == dev &&
-      nh->nh_scope != scope) {
-     nh->nh_flags |= RTNH_F_DEAD;
+ hlist_for_each_entry(nh, node, head, nh_hash) {
+  struct fib_info *fi = nh->nh_parent;
+  int dead;
+
+  BUG_ON(!fi->fib_nhs);
+  if (nh->nh_dev != dev || fi == prev_fi)
+   continue;
+  prev_fi = fi;
+  dead = 0;
+  change_nexthops(fi) {
+   if (nh->nh_flags&RTNH_F_DEAD)
+    dead++;
+   else if (nh->nh_dev == dev &&
+     nh->nh_scope != scope) {
+    nh->nh_flags |= RTNH_F_DEAD;
```

```
 #ifdef CONFIG_IP_ROUTE_MULTIPATH
-     spin_lock_bh(&fib_multipath_lock);
-     fi->fib_power -= nh->nh_power;
-     nh->nh_power = 0;
-     spin_unlock_bh(&fib_multipath_lock);
+     spin_lock_bh(&fib_multipath_lock);
+     fi->fib_power -= nh->nh_power;
+     nh->nh_power = 0;
+     spin_unlock_bh(&fib_multipath_lock);
 #endif
-     dead++;
-    }
+     dead++;
+   }
 #ifdef CONFIG_IP_ROUTE_MULTIPATH
-    if (force > 1 && nh->nh_dev == dev) {
-     dead = fi->fib_nhs;
-     break;
-    }
-#endif
-   } endfor_nexthops(fi)
-   if (dead == fi->fib_nhs) {
-    fi->fib_flags |= RTNH_F_DEAD;
-    ret++;
+   if (force > 1 && nh->nh_dev == dev) {
+    dead = fi->fib_nhs;
+    break;
    }
+#endif
+  } endfor_nexthops(fi)
+  if (dead == fi->fib_nhs) {
+   fi->fib_flags |= RTNH_F_DEAD;
+   ret++;
   }
  }

--
1.5.3.rc5
```