Subject: [patch 05/12] net namespace : ioctl to push ifa to net namespace l3
Posted by Daniel Lezcano on Fri, 19 Jan 2007 15:47:19 GMT
View Forum Message <> Reply to Message

From: Daniel Lezcano <dlezcano@fr.ibm.com>

New ioctl to "push" ifaddr to a container. Actually, the push is done
from the current namespace, so the right word is "pull". That will be
changed to move ifaddr from l2 network namespace to l3.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

---
 include/linux/net_namespace.h | 7 ++
 include/linux/sockios.h | 4 +
 net/core/net_namespace.c | 118 ++-
 net/ipv4/af_inet.c | 4 +
 4 files changed, 132 insertions(+), 1 deletion(-)

Index: 2.6.20-rc4-mm1/include/linux/sockios.h
===
--- 2.6.20-rc4-mm1.orig/include/linux/sockios.h
+++ 2.6.20-rc4-mm1/include/linux/sockios.h
@@ -122,6 +122,10 @@
 #define SIOCBRADDIF 0x89a2 /* add interface to bridge   */
 #define SIOCBRDELIF 0x89a3 /* remove interface from bridge */

+/* Container calls */
+#define SIOCNETNSPUSHIF 0x89b0   /* add ifaddr to namespace   */
+#define SIOCNETNSPULLIF 0x89b1   /* remove ifaddr to namespace   */
+
 /* Device private ioctl calls */

 /*
Index: 2.6.20-rc4-mm1/net/ipv4/af_inet.c
===
--- 2.6.20-rc4-mm1.orig/net/ipv4/af_inet.c
+++ 2.6.20-rc4-mm1/net/ipv4/af_inet.c
@@ -789,6 +789,10 @@
   case SIOCSIFFLAGS:
    err = devinet_ioctl(cmd, (void __user *)arg);
    break;
+   case SIOCNETNSPUSHIF:
+   case SIOCNETNSPULLIF:
+   err = net_ns_ioctl(cmd, (void __user *)arg);
+   break;
   default:
    if (sk->sk_prot->ioctl)

```
     err = sk->sk_prot->ioctl(sk, cmd, arg);
Index: 2.6.20-rc4-mm1/include/linux/net_namespace.h
===================================================================
--- 2.6.20-rc4-mm1.orig/include/linux/net_namespace.h
+++ 2.6.20-rc4-mm1/include/linux/net_namespace.h
@@ -91,6 +91,8 @@

 #define net_ns_hash(ns)   ((ns)->hash)

+extern int net_ns_ioctl(unsigned int cmd, void __user *arg);
+
 #else /* CONFIG_NET_NS */

 #define INIT_NET_NS(net_ns)
@@ -141,6 +143,11 @@

 #define net_ns_hash(ns)   (0)

+static inline int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+ return -ENOSYS;
+}
+
 #endif /* !CONFIG_NET_NS */

 #endif /* _LINUX_NET_NAMESPACE_H */
Index: 2.6.20-rc4-mm1/net/core/net_namespace.c
===================================================================
--- 2.6.20-rc4-mm1.orig/net/core/net_namespace.c
+++ 2.6.20-rc4-mm1/net/core/net_namespace.c
@@ -10,7 +10,9 @@
 #include <linux/nsproxy.h>
 #include <linux/net_namespace.h>
 #include <linux/net.h>
+#include <linux/in.h>
 #include <linux/netdevice.h>
+#include <linux/inetdevice.h>
 #include <net/ip_fib.h>

 struct net_namespace init_net_ns = {
@@ -123,6 +125,33 @@
  return err;
 }

+/*
+ * The function will move the ifaddr to the l2 network namespace
+ * parent.
+ * @net_ns: the related network namespace
```

```
+ */
+static void release_ifa_to_parent(const struct net_namespace* net_ns)
+{
+ struct net_device *dev;
+ struct in_device *in_dev;
+
+ read_lock(&dev_base_lock);
+ rcu_read_lock();
+ for (dev = dev_base; dev; dev = dev->next) {
+  in_dev = __in_dev_get_rcu(dev);
+  if (!in_dev)
+   continue;
+
+  for_ifa(in_dev) {
+   if (ifa->ifa_net_ns != net_ns)
+    continue;
+   ifa->ifa_net_ns = net_ns->parent;
+  } endfor_ifa(in_dev);
+ }
+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+}
+
 void free_net_ns(struct kref *kref)
 {
  struct net_namespace *ns;
@@ -139,12 +168,99 @@
   }
  }

- if (ns->level == NET_NS_LEVEL3)
+ if (ns->level == NET_NS_LEVEL3) {
+  release_ifa_to_parent(ns);
   put_net_ns(ns->parent);
+ }

  printk(KERN_DEBUG "NET_NS: net namespace %p destroyed\n", ns);
  kfree(ns);
 }
 EXPORT_SYMBOL_GPL(free_net_ns);

+/*
+ * This function allows to assign an IP address from a l2 network
+ * namespace to one of his l3 child or to release from an l3 network
+ * namespace to his l2 network namespace parent.
+ * @cmd: a "push" / "pull" command
+ * @arg: an userspace buffer containing an ifreq structure
+ * Returns:
```

```
+ * - EPERM : if caller has no CAP_NET_ADMIN capabilities or the
+ *           current level of network namespace is not layer 2
+ * - EFAULT : if arg is an invalid buffer
+ * - EADDRNOTAVAIL : if the specified ifaddr does not exists
+ * - EINVAL : if cmd is unknown
+ * - zero on success
+ */
+int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+ struct ifreq ifr;
+ struct sockaddr_in *sin = (struct sockaddr_in *)&ifr.ifr_addr;
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_device *dev;
+ struct in_device *in_dev;
+ struct in_ifaddr **ifap = NULL;
+ struct in_ifaddr *ifa = NULL;
+ char *colon;
+ int err;
+
+ if (!capable(CAP_NET_ADMIN))
+  return -EPERM;
+
+ if (net_ns->level != NET_NS_LEVEL3)
+  return -EPERM;
+
+ if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
+  return -EFAULT;
+
+ ifr.ifr_name[IFNAMSIZ - 1] = 0;
+
+
+ colon = strchr(ifr.ifr_name, ':');
+ if (colon)
+  *colon = 0;
+
+ rtnl_lock();
+
+ err = -ENODEV;
+ dev = __dev_get_by_name(ifr.ifr_name);
+ if (!dev)
+  goto out;
+
+ if (colon)
+  *colon = ':';
+
+ err = -EADDRNOTAVAIL;
+ in_dev = __in_dev_get_rtnl(dev);
+ if (!in_dev)
```

```
+  goto out;
+
+ for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
+       ifap = &ifa->ifa_next)
+  if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+       sin->sin_addr.s_addr == ifa->ifa_local)
+   break;
+ if (!ifa)
+  goto out;
+
+ err = -EINVAL;
+ switch(cmd) {
+
+ case SIOCNETNSPUSHIF:
+  ifa->ifa_net_ns = net_ns;
+  break;
+
+ case SIOCNETNSPULLIF:
+  ifa->ifa_net_ns = net_ns->parent;
+  break;
+ default:
+  goto out;
+ }
+
+ err = 0;
+out:
+ rtnl_unlock();
+ return err;
+}
+
 #endif /* CONFIG_NET_NS */

--
```