Subject: [patch 08/20] [Network namespace] Move the dev name hash relative to the namespace.
Posted by Daniel Lezcano on Sun, 10 Dec 2006 21:58:25 GMT
View Forum Message <> Reply to Message

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

---

```
 include/linux/net_namespace.h |   14 +++++++++++++
 include/linux/netdevice.h     |    1 +
 net/core/dev.c                |   17 ++++++++++-----
 net/core/net_namespace.c      |   32 ++++++++++++++++++++++++++++++++-
 4 files changed, 58 insertions(+), 6 deletions(-)

Index: 2.6.19-rc6-mm2/net/core/dev.c
===================================================================
--- 2.6.19-rc6-mm2.orig/net/core/dev.c
+++ 2.6.19-rc6-mm2/net/core/dev.c
@@ -188,9 +188,13 @@ EXPORT_SYMBOL(dev_base);
 DEFINE_RWLOCK(dev_base_lock);
 EXPORT_SYMBOL(dev_base_lock);

-#define NETDEV_HASHBITS 8
+#ifdef CONFIG_NET_NS
+#define dev_name_head (current_net_ns->net_device.name_head)
+#define dev_index_head (current_net_ns->net_device.index_head)
+#else
 static struct hlist_head dev_name_head[1<<NETDEV_HASHBITS];
 static struct hlist_head dev_index_head[1<<NETDEV_HASHBITS];
+#endif

 static inline struct hlist_head *dev_name_hash(const char *name,
       struct net_namespace *ns)
@@ -486,13 +490,11 @@ __setup("netdev=", netdev_boot_setup);
 struct net_device *__dev_get_by_name(const char *name)
 {
  struct hlist_node *p;
- struct net_namespace *ns = current_net_ns;
+ struct net_namespace *net_ns = current_net_ns;

- hlist_for_each(p, dev_name_hash(name, ns)) {
+ hlist_for_each(p, dev_name_hash(name, net_ns)) {
    struct net_device *dev
     = hlist_entry(p, struct net_device, name_hlist);
-  if (!net_ns_match(dev->net_ns, ns))
-   continue;
    if (!strncmp(dev->name, name, IFNAMSIZ))
```

```
    return dev;
  }
@@ -3636,6 +3638,11 @@ static int __init net_dev_init(void)
  if (netdev_sysfs_init())
   goto out;

+#ifdef CONFIG_NET_NS
+ if (hlist_dev_name_init(current_net_ns))
+  goto out;
+#endif
+
  INIT_LIST_HEAD(&ptype_all);
  for (i = 0; i < 16; i++)
   INIT_LIST_HEAD(&ptype_base[i]);
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
===================================================================
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -12,6 +12,8 @@
 #include <linux/net.h>
 #include <linux/netdevice.h>
 #include <net/ip_fib.h>
+#include <linux/inetdevice.h>
+#include <linux/in.h>

 static spinlock_t net_ns_list_lock = SPIN_LOCK_UNLOCKED;

@@ -31,6 +33,33 @@ struct net_namespace init_net_ns = {

 #ifdef CONFIG_NET_NS

+int hlist_dev_name_init(struct net_namespace *net_ns)
+{
+ struct hlist_head *hlist_index, *hlist_name;
+ const int size = sizeof(struct hlist_head)*(1<<NETDEV_HASHBITS);
+
+ hlist_name = kmalloc(size, GFP_KERNEL);
+ if (!hlist_name)
+  return -ENOMEM;
+
+ hlist_index = kmalloc(size, GFP_KERNEL);
+ if (!hlist_index) {
+  kfree(hlist_name);
+  return -ENOMEM;
+ }
+
+ net_ns->net_device.name_head = hlist_name;
+ net_ns->net_device.index_head = hlist_index;
```

```
+
+ return 0;
+}
+
+static inline void hlist_dev_name_cleanup(struct net_namespace *net_ns)
+{
+ kfree(net_ns->net_device.name_head);
+ kfree(net_ns->net_device.index_head);
+}
+
 /*
  * Clone a new ns copying an original net ns, setting refcount to 1
  * @level: level of namespace to create
@@ -52,7 +81,6 @@ static struct net_namespace *clone_net_n

  kref_init(&ns->kref);
  ns->ns = old_ns->ns;
- ns->hash = net_random();
  INIT_LIST_HEAD(&ns->child_list);
  spin_lock_irq(&net_ns_list_lock);
  get_net_ns(old_ns);
@@ -62,6 +90,7 @@ static struct net_namespace *clone_net_n

  if (level ==  NET_NS_LEVEL2) {

+  ns->hash = net_random();
   ns->dev_base_p = NULL;
   ns->dev_tail_p = &ns->dev_base_p;

@@ -153,4 +182,5 @@ void free_net_ns(struct kref *kref)
 }
 /* because of put_net_ns() */
 EXPORT_SYMBOL(free_net_ns);
+
 #endif /* CONFIG_NET_NS */
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h
===================================================================
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -5,11 +5,18 @@
 #include <linux/nsproxy.h>
 #include <linux/errno.h>

+struct net_ns_net_device {
+       struct hlist_head *name_head;
+       struct hlist_head *index_head;
+};
+
```

```
 struct net_namespace {
  struct kref  kref;
  struct nsproxy  *ns;
  struct net_device *dev_base_p, **dev_tail_p;
  struct net_device *loopback_dev_p;
+ struct net_ns_net_device net_device;
+
  struct pcpu_lstats *pcpu_lstats_p;
 #ifndef CONFIG_IP_MULTIPLE_TABLES
  struct fib_table *fib4_local_table, *fib4_main_table;
@@ -77,6 +84,8 @@ static inline void pop_net_ns(struct net

 extern struct net_namespace *find_net_ns(unsigned int id);

+extern int hlist_dev_name_init(struct net_namespace *net_ns);
+
 #else /* CONFIG_NET_NS */

 #define INIT_NET_NS(net_ns)
@@ -123,6 +132,11 @@ static inline struct net_namespace *find
  return NULL;
 }

+static inline int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+ return 0;
+}
+
 #endif /* !CONFIG_NET_NS */

 #endif /* _LINUX_NET_NAMESPACE_H */
Index: 2.6.19-rc6-mm2/include/linux/netdevice.h
===================================================================
--- 2.6.19-rc6-mm2.orig/include/linux/netdevice.h
+++ 2.6.19-rc6-mm2/include/linux/netdevice.h
@@ -81,6 +81,7 @@ struct netpoll_info;
 #define NETDEV_TX_BUSY 1 /* driver tx path was busy*/
 #define NETDEV_TX_LOCKED -1 /* driver tx lock was already taken */

+#define NETDEV_HASHBITS 8       /* hash bit size for netdev hash table */
 /*
  * Compute the worst case header length according to the protocols
  * used.

--
_____
Containers mailing list
Containers@lists.osdl.org
```