
Subject: [PATCH net-2.6.25 6/19] [NETNS] Refactor fib initialization so it can handle multiple namespaces.

Posted by [den](#) on Wed, 09 Jan 2008 17:59:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

This patch makes the fib to be initialized as a subsystem for the network namespaces. The code does not handle several namespaces yet, so in case of a creation of a network namespace, the creation/initialization will not occur.

Acked-by: Benjamin Thery <benjamin.thery@bull.net>

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

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
include/net/ip_fib.h | 3 +-
net/ipv4/af_inet.c   | 4 --
net/ipv4/fib_frontend.c | 86 ++++++-----
net/ipv4/fib_hash.c   | 6 +---
net/ipv4/fib_rules.c  | 11 ++++--
net/ipv4/fib_trie.c   | 6 +---
6 files changed, 90 insertions(+), 26 deletions(-)
```

diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h

index cbff18d..338d3ed 100644

--- a/include/net/ip_fib.h

+++ b/include/net/ip_fib.h

```
@@ -186,7 +186,8 @@ static inline void fib_select_default(const struct flowi *flp, struct fib_result
}
```

```
#else /* CONFIG_IP_MULTIPLE_TABLES */
```

```
-extern int __init fib4_rules_init(void);
```

```
+extern int __net_init fib4_rules_init(struct net *net);
```

```
+extern void __net_exit fib4_rules_exit(struct net *net);
```

```
#ifdef CONFIG_NET_CLS_ROUTE
```

```
extern u32 fib_rules_tclass(struct fib_result *res);
```

diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c

index ea9fd3d..19b557a 100644

--- a/net/ipv4/af_inet.c

+++ b/net/ipv4/af_inet.c

```
@@ -1470,15 +1470,11 @@ static int __init ipv4_proc_init(void)
```

```
    goto out_tcp;
```

```
    if (udp4_proc_init())
```

```
        goto out_udp;
```

```
- if (fib_proc_init(&init_net))
```

```
- goto out_fib;
```

```
    if (ip_misc_proc_init())
```

```
        goto out_misc;
```

```
out:
```

```

    return rc;
out_misc:
- fib_proc_exit(&init_net);
-out_fib:
    udp4_proc_exit();
out_udp:
    tcp4_proc_exit();
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 82d0e00..714fff9 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -59,7 +59,7 @@ struct fib_table *ip_fib_main_table;
#define FIB_TABLE_HASHSZ 1
static struct hlist_head fib_table_hash[FIB_TABLE_HASHSZ];

-static int __init fib4_rules_init(void)
+static int __net_init fib4_rules_init(struct net *net)
{
    ip_fib_local_table = fib_hash_init(RT_TABLE_LOCAL);
    if (ip_fib_local_table == NULL)
@@ -860,10 +860,18 @@ static void nl_fib_input(struct sk_buff *skb)
    netlink_unicast(fibnl, skb, pid, MSG_DONTWAIT);
}

-static void nl_fib_lookup_init(void)
+static int nl_fib_lookup_init(struct net *net)
{
- fibnl = netlink_kernel_create(&init_net, NETLINK_FIB_LOOKUP, 0,
+ fibnl = netlink_kernel_create(net, NETLINK_FIB_LOOKUP, 0,
    nl_fib_input, NULL, THIS_MODULE);
+ if (fibnl == NULL)
+ return -EAFNOSUPPORT;
+ return 0;
+}
+
+static void nl_fib_lookup_exit(struct net *net)
+{
+ sock_put(fibnl);
}

static void fib_disable_ip(struct net_device *dev, int force)
@@ -946,22 +946,22 @@ static struct notifier_block fib_netdev_notifier = {
    .notifier_call = fib_netdev_event,
};

-void __init ip_fib_init(void)
+static int __net_init ip_fib_net_init(struct net *net)
{

```

```

unsigned int i;

for (i = 0; i < FIB_TABLE_HASHSZ; i++)
    INIT_HLIST_HEAD(&fib_table_hash[i]);

- BUG_ON(fib4_rules_init());
+ return fib4_rules_init(net);
+}
+
+static void __net_exit ip_fib_net_exit(struct net *net)
+{
+ unsigned int i;

- register_netdevice_notifier(&fib_netdev_notifier);
- register_inetaddr_notifier(&fib_inetaddr_notifier);
- nl_fib_lookup_init();
+ #ifdef CONFIG_IP_MULTIPLE_TABLES
+ fib4_rules_exit(net);
+ #endif
+
+ for (i = 0; i < FIB_TABLE_HASHSZ; i++) {
+ struct fib_table *tb;
+ struct hlist_head *head;
+ struct hlist_node *node, *tmp;
+
+ head = &fib_table_hash[i];
+ hlist_for_each_entry_safe(tb, node, tmp, head, tb_hlist) {
+ hlist_del(node);
+ tb->tb_flush(tb);
+ kfree(tb);
+ }
+ }
+}
+
+static int __net_init fib_net_init(struct net *net)
+{
+ int error;

+ error = 0;
+ if (net != &init_net)
+ goto out;
+
+ error = ip_fib_net_init(net);
+ if (error < 0)
+ goto out;
+ error = nl_fib_lookup_init(net);
+ if (error < 0)
+ goto out_nfl;

```

```

+ error = fib_proc_init(net);
+ if (error < 0)
+ goto out_proc;
+out:
+ return error;
+
+out_proc:
+ nl_fib_lookup_exit(net);
+out_nfl:
+ ip_fib_net_exit(net);
+ goto out;
+}
+
+static void __net_exit fib_net_exit(struct net *net)
+{
+ fib_proc_exit(net);
+ nl_fib_lookup_exit(net);
+ ip_fib_net_exit(net);
+}
+
+static struct pernet_operations fib_net_ops = {
+ .init = fib_net_init,
+ .exit = fib_net_exit,
+};
+
+void __init ip_fib_init(void)
+{
+ rtnl_register(PF_INET, RTM_NEWROUTE, inet_rtm_newroute, NULL);
+ rtnl_register(PF_INET, RTM_DELROUTE, inet_rtm_delroute, NULL);
+ rtnl_register(PF_INET, RTM_GETROUTE, NULL, inet_dump_fib);
+
+ register_pernet_subsys(&fib_net_ops);
+ register_netdevice_notifier(&fib_netdev_notifier);
+ register_inetaddr_notifier(&fib_inetaddr_notifier);
+}

EXPORT_SYMBOL(inet_addr_type);
diff --git a/net/ipv4/fib_hash.c b/net/ipv4/fib_hash.c
index f5476e2..11277f6 100644
--- a/net/ipv4/fib_hash.c
+++ b/net/ipv4/fib_hash.c
@@ -745,11 +745,7 @@ static int fn_hash_dump(struct fib_table *tb, struct sk_buff *skb, struct
netlin
    return skb->len;
}

#ifdef CONFIG_IP_MULTIPLE_TABLES
struct fib_table * fib_hash_init(u32 id)

```

```

-#else
-struct fib_table * __init fib_hash_init(u32 id)
-#endif
+struct fib_table *fib_hash_init(u32 id)
{
    struct fib_table *tb;

diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 0751734..1aae61c 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -311,11 +311,11 @@ static int __init fib_default_rules_init(void)
    return 0;
}

-int __init fib4_rules_init()
+int __net_init fib4_rules_init(struct net *net)
{
    int err;

- fib_rules_register(&init_net, &fib4_rules_ops);
+ fib_rules_register(net, &fib4_rules_ops);
    err = fib_default_rules_init();
    if (err < 0)
        goto fail;
@@ -323,6 +323,11 @@ int __init fib4_rules_init()

fail:
    /* also cleans all rules already added */
- fib_rules_unregister(&init_net, &fib4_rules_ops);
+ fib_rules_unregister(net, &fib4_rules_ops);
    return err;
}
+
+void __net_exit fib4_rules_exit(struct net *net)
+{
+ fib_rules_unregister(net, &fib4_rules_ops);
+}
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 3c7f668..43b6e94 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -1953,11 +1953,7 @@ out:

/* Fix more generic FIB names for init later */

-#ifdef CONFIG_IP_MULTIPLE_TABLES
-struct fib_table * fib_hash_init(u32 id)

```

```
-#else
-struct fib_table * __init fib_hash_init(u32 id)
-#endif
+struct fib_table *fib_hash_init(u32 id)
{
    struct fib_table *tb;
    struct trie *t;
}
--
1.5.3.rc5
```
