

---

Subject: [PATCH 0/3 net-2.6.25] call FIB rule->action in the correct namespace  
Posted by [den](#) on Thu, 17 Jan 2008 10:08:51 GMT

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

---

FIB rule->action should operate in the same namespace as fib\_lookup.  
This is definitely missed right now.

There are two ways to implement this: pass struct net into another rules  
API call (2 levels) or place netns into rule struct directly. The second  
approach seems better as the code will grow less.

Additionally, the patchset cleans up struct net from  
fib\_rules\_register/unregister to have network namespace context at the  
time of default rules creation.

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

---

---

Subject: [PATCH 1/3 net-2.6.25] Add netns to fib\_rules\_ops.  
Posted by [den](#) on Thu, 17 Jan 2008 10:09:16 GMT

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

---

The backward link from FIB rules operations to the network namespace will  
allow to simplify the API a bit.

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

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

---

```
include/net/fib_rules.h | 1 +
net/decnet/dn_rules.c   | 1 +
net/ipv4/fib_rules.c    | 2 ++
net/ipv6/fib6_rules.c   | 1 +
4 files changed, 5 insertions(+), 0 deletions(-)
```

```
diff --git a/include/net/fib_rules.h b/include/net/fib_rules.h
```

```
index 4f47250..6910e01 100644
```

```
--- a/include/net/fib_rules.h
```

```
+++ b/include/net/fib_rules.h
```

```
@@ -67,6 +67,7 @@ struct fib_rules_ops
```

```
    const struct nla_policy *policy;
```

```
    struct list_head rules_list;
```

```
    struct module *owner;
```

```
+ struct net *fro_net;
```

```
};
```

```
#define FRA_GENERIC_POLICY \
```

```
diff --git a/net/decnet/dn_rules.c b/net/decnet/dn_rules.c
```

```
index c1fae23..964e658 100644
```

```

--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -249,6 +249,7 @@ static struct fib_rules_ops dn_fib_rules_ops = {
    .policy = dn_fib_rule_policy,
    .rules_list = LIST_HEAD_INIT(dn_fib_rules_ops.rules_list),
    .owner = THIS_MODULE,
+   .fro_net = &init_net,
};

void __init dn_fib_rules_init(void)
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 72232ab..8d0ebe7 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -315,6 +315,8 @@ int __net_init fib4_rules_init(struct net *net)
    if (ops == NULL)
        return -ENOMEM;
    INIT_LIST_HEAD(&ops->rules_list);
+   ops->fro_net = net;
+
    fib_rules_register(net, ops);

    err = fib_default_rules_init(ops);
diff --git a/net/ipv6/fib6_rules.c b/net/ipv6/fib6_rules.c
index 76437a1..ead5ab2 100644
--- a/net/ipv6/fib6_rules.c
+++ b/net/ipv6/fib6_rules.c
@@ -249,6 +249,7 @@ static struct fib_rules_ops fib6_rules_ops = {
    .policy = fib6_rule_policy,
    .rules_list = LIST_HEAD_INIT(fib6_rules_ops.rules_list),
    .owner = THIS_MODULE,
+   .fro_net = &init_net,
};

static int __init fib6_default_rules_init(void)
--
1.5.3.rc5

```

---

Subject: [PATCH 2/3 net-2.6.25] [NETNS] FIB rules API cleanup.  
 Posted by [den](#) on Thu, 17 Jan 2008 10:09:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Remove struct net from fib\_rules\_register(unregister)/notify\_change paths  
 and diet code size a bit.

add/remove: 0/0 grow/shrink: 10/12 up/down: 35/-100 (-65)  
 function                      old    new   delta

|                           |      |      |     |
|---------------------------|------|------|-----|
| notify_rule_change        | 273  | 280  | +7  |
| trie_show_stats           | 471  | 475  | +4  |
| fn_trie_delete            | 473  | 477  | +4  |
| fib_rules_unregister      | 144  | 148  | +4  |
| fib4_rule_compare         | 119  | 123  | +4  |
| resize                    | 2842 | 2845 | +3  |
| fn_trie_select_default    | 515  | 518  | +3  |
| inet_sk_rebuild_header    | 836  | 838  | +2  |
| fib_trie_seq_show         | 764  | 766  | +2  |
| __devinet_sysctl_register | 276  | 278  | +2  |
| fn_trie_lookup            | 1124 | 1123 | -1  |
| ip_fib_check_default      | 133  | 131  | -2  |
| devinet_conf_sysctl       | 223  | 221  | -2  |
| snmp_fold_field           | 126  | 123  | -3  |
| fn_trie_insert            | 2091 | 2086 | -5  |
| inet_create               | 876  | 870  | -6  |
| fib4_rules_init           | 197  | 191  | -6  |
| fib_sync_down             | 452  | 444  | -8  |
| inet_gso_send_check       | 334  | 325  | -9  |
| fib_create_info           | 3003 | 2991 | -12 |
| fib_nl_delrule            | 568  | 553  | -15 |
| fib_nl_newrule            | 883  | 852  | -31 |

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

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

---

```
include/net/fib_rules.h | 4 +++
net/core/fib_rules.c   | 20 ++++++++-----
net/decnet/dn_rules.c  | 4 ++--
net/ipv4/fib_rules.c   | 6 +++---
net/ipv6/fib6_rules.c  | 4 ++--
5 files changed, 22 insertions(+), 16 deletions(-)
```

diff --git a/include/net/fib\_rules.h b/include/net/fib\_rules.h

index 6910e01..7f9f4ae 100644

--- a/include/net/fib\_rules.h

+++ b/include/net/fib\_rules.h

```
@@ -102,8 +102,8 @@ static inline u32 frh_get_table(struct fib_rule_hdr *frh, struct nlattr **nla)
    return frh->table;
}
```

```
-extern int fib_rules_register(struct net *, struct fib_rules_ops *);
```

```
-extern void fib_rules_unregister(struct net *, struct fib_rules_ops *);
```

```
+extern int fib_rules_register(struct fib_rules_ops *);
```

```
+extern void fib_rules_unregister(struct fib_rules_ops *);
```

```
extern void fib_rules_cleanup_ops(struct fib_rules_ops *);
```

```
extern int fib_rules_lookup(struct fib_rules_ops *,
```

```

diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
index 541728a..3cd4f13 100644
--- a/net/core/fib_rules.c
+++ b/net/core/fib_rules.c
@@ -37,8 +37,7 @@ int fib_default_rule_add(struct fib_rules_ops *ops,
 }
EXPORT_SYMBOL(fib_default_rule_add);

-static void notify_rule_change(struct net *net, int event,
-    struct fib_rule *rule,
+static void notify_rule_change(int event, struct fib_rule *rule,
    struct fib_rules_ops *ops, struct nlmsg_hdr *nlh,
    u32 pid);

@@ -72,10 +71,13 @@ static void flush_route_cache(struct fib_rules_ops *ops)
    ops->flush_cache();
}

-int fib_rules_register(struct net *net, struct fib_rules_ops *ops)
+int fib_rules_register(struct fib_rules_ops *ops)
{
    int err = -EEXIST;
    struct fib_rules_ops *o;
+ struct net *net;
+
+ net = ops->fro_net;

    if (ops->rule_size < sizeof(struct fib_rule))
        return -EINVAL;
@@ -112,8 +114,9 @@ void fib_rules_cleanup_ops(struct fib_rules_ops *ops)
}
EXPORT_SYMBOL_GPL(fib_rules_cleanup_ops);

-void fib_rules_unregister(struct net *net, struct fib_rules_ops *ops)
+void fib_rules_unregister(struct fib_rules_ops *ops)
{
+ struct net *net = ops->fro_net;

    spin_lock(&net->rules_mod_lock);
    list_del_rcu(&ops->list);
@@ -333,7 +336,7 @@ static int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr *nlh, void
*arg)
    else
        list_add_rcu(&rule->list, &ops->rules_list);

- notify_rule_change(net, RTM_NEWRULE, rule, ops, nlh, NETLINK_CB(skb).pid);
+ notify_rule_change(RTM_NEWRULE, rule, ops, nlh, NETLINK_CB(skb).pid);
    flush_route_cache(ops);

```

```

rules_ops_put(ops);
return 0;
@@ -423,7 +426,7 @@ static int fib_nl_delrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void
*arg)
}

synchronize_rcu();
- notify_rule_change(net, RTM_DELRULE, rule, ops, nlh,
+ notify_rule_change(RTM_DELRULE, rule, ops, nlh,
    NETLINK_CB(skb).pid);
fib_rule_put(rule);
flush_route_cache(ops);
@@ -561,13 +564,15 @@ static int fib_nl_dumprule(struct sk_buff *skb, struct netlink_callback
*cb)
return skb->len;
}

-static void notify_rule_change(struct net *net, int event, struct fib_rule *rule,
+static void notify_rule_change(int event, struct fib_rule *rule,
    struct fib_rules_ops *ops, struct nlmsg_hdr *nlh,
    u32 pid)
{
+ struct net *net;
    struct sk_buff *skb;
    int err = -ENOBUFS;

+ net = ops->fro_net;
    skb = nlmsg_new(fib_rule_nlmsg_size(ops, rule), GFP_KERNEL);
    if (skb == NULL)
        goto errout;
@@ -579,6 +584,7 @@ static void notify_rule_change(struct net *net, int event, struct fib_rule
*rule
    kfree_skb(skb);
    goto errout;
}
+
err = rtnl_notify(skb, net, pid, ops->nlgrouper, nlh, GFP_KERNEL);
errout:
if (err < 0)
diff --git a/net/decnet/dn_rules.c b/net/decnet/dn_rules.c
index 964e658..5b7539b 100644
--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -256,12 +256,12 @@ void __init dn_fib_rules_init(void)
{
    BUG_ON(fib_default_rule_add(&dn_fib_rules_ops, 0x7fff,
        RT_TABLE_MAIN, 0));
- fib_rules_register(&init_net, &dn_fib_rules_ops);

```

```

+ fib_rules_register(&dn_fib_rules_ops);
}

void __exit dn_fib_rules_cleanup(void)
{
- fib_rules_unregister(&init_net, &dn_fib_rules_ops);
+ fib_rules_unregister(&dn_fib_rules_ops);
}

diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 8d0ebe7..3b7affd 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -317,7 +317,7 @@ int __net_init fib4_rules_init(struct net *net)
    INIT_LIST_HEAD(&ops->rules_list);
    ops->fro_net = net;

- fib_rules_register(net, ops);
+ fib_rules_register(ops);

    err = fib_default_rules_init(ops);
    if (err < 0)
@@ -327,13 +327,13 @@ int __net_init fib4_rules_init(struct net *net)

fail:
/* also cleans all rules already added */
- fib_rules_unregister(net, ops);
+ fib_rules_unregister(ops);
    kfree(ops);
    return err;
}

void __net_exit fib4_rules_exit(struct net *net)
{
- fib_rules_unregister(net, net->ipv4.rules_ops);
+ fib_rules_unregister(net->ipv4.rules_ops);
    kfree(net->ipv4.rules_ops);
}
diff --git a/net/ipv6/fib6_rules.c b/net/ipv6/fib6_rules.c
index ead5ab2..695c0ca 100644
--- a/net/ipv6/fib6_rules.c
+++ b/net/ipv6/fib6_rules.c
@@ -274,7 +274,7 @@ int __init fib6_rules_init(void)
    if (ret)
        goto out;

- ret = fib_rules_register(&init_net, &fib6_rules_ops);

```

```

+ ret = fib_rules_register(&fib6_rules_ops);
  if (ret)
    goto out_default_rules_init;
out:
@@ -287,5 +287,5 @@ out_default_rules_init:

void fib6_rules_cleanup(void)
{
- fib_rules_unregister(&init_net, &fib6_rules_ops);
+ fib_rules_unregister(&fib6_rules_ops);
}
--
1.5.3.rc5

```

---

Subject: [PATCH 3/3 net-2.6.25] Process FIB rule action in the context of the namespace.

Posted by [den](#) on Thu, 17 Jan 2008 10:09:18 GMT

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

---

Save namespace context on the fib rule at the rule creation time and call routing lookup in the correct namespace.

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

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

---

```

include/net/fib_rules.h | 1 +
net/core/fib_rules.c   | 2 ++
net/ipv4/fib_rules.c   | 2 +-
3 files changed, 4 insertions(+), 1 deletions(-)

```

```
diff --git a/include/net/fib_rules.h b/include/net/fib_rules.h
```

```
index 7f9f4ae..34349f9 100644
```

```
--- a/include/net/fib_rules.h
```

```
+++ b/include/net/fib_rules.h
```

```
@@ -22,6 +22,7 @@ struct fib_rule
```

```

    u32 target;
    struct fib_rule * ctarget;
    struct rcu_head rcu;
+ struct net * fr_net;
};

```

```
struct fib_lookup_arg
```

```
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
```

```
index 3cd4f13..42ccaf5 100644
```

```
--- a/net/core/fib_rules.c
```

```
+++ b/net/core/fib_rules.c
```

```
@@ -29,6 +29,7 @@ int fib_default_rule_add(struct fib_rules_ops *ops,
```

```

r->pref = pref;
r->table = table;
r->flags = flags;
+ r->fr_net = ops->fro_net;

/* The lock is not required here, the list is unreachable
 * at the moment this function is called */
@@ -242,6 +243,7 @@ static int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void
*arg)
    err = -ENOMEM;
    goto errout;
}
+ rule->fr_net = net;

if (tb[FRA_PRIORITY])
    rule->pref = nla_get_u32(tb[FRA_PRIORITY]);
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 3b7affd..d2001f1 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -91,7 +91,7 @@ static int fib4_rule_action(struct fib_rule *rule, struct flowi *flp,
    goto errout;
}

- if ((tbl = fib_get_table(&init_net, rule->table)) == NULL)
+ if ((tbl = fib_get_table(rule->fr_net, rule->table)) == NULL)
    goto errout;

err = tbl->tb_lookup(tbl, flp, (struct fib_result *) arg->result);
--
1.5.3.rc5

```

---

Subject: Re: [PATCH 0/3 net-2.6.25] call FIB rule->action in the correct namespace  
Posted by [Daniel Lezcano](#) on Thu, 17 Jan 2008 10:41:36 GMT

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

---

Denis V. Lunev wrote:

- > FIB rule->action should operate in the same namespace as fib\_lookup.
- > This is definitely missed right now.
- >
- > There are two ways to implement this: pass struct net into another rules
- > API call (2 levels) or place netns into rule struct directly. The second
- > approach seems better as the code will grow less.
- >
- > Additionally, the patchset cleans up struct net from
- > fib\_rules\_register/unregister to have network namespace context at the
- > time of default rules creation.

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

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

--

Sauf indication contraire ci-dessus:  
Compagnie IBM France  
Sie`ge Social : Tour Descartes, 2, avenue Gambetta, La De'fense 5, 92400  
Courbevoie  
RCS Nanterre 552 118 465  
Forme Sociale : S.A.S.  
Capital Social : 542.737.118 ?  
SIREN/SIRET : 552 118 465 02430

---

---

Subject: Re: [PATCH 0/3 net-2.6.25] call FIB rule->action in the correct namespace  
Posted by [davem](#) on Mon, 21 Jan 2008 00:47:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: "Denis V. Lunev" <den@openvz.org>  
Date: Thu, 17 Jan 2008 13:08:51 +0300

> FIB rule->action should operate in the same namespace as fib\_lookup.  
> This is definitely missed right now.  
>  
> There are two ways to implement this: pass struct net into another rules  
> API call (2 levels) or place netns into rule struct directly. The second  
> approach seems better as the code will grow less.  
>  
> Additionally, the patchset cleans up struct net from  
> fib\_rules\_register/unregister to have network namespace context at the  
> time of default rules creation.  
>  
> Signed-off-by: Denis V. Lunev <den@openvz.org>

All 3 patches applied, thanks.

---