
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/dechnet/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_dump rule(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->nlgrou p, 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
```
