
Subject: [PATCH 1/6 net-2.6.25] [NETNS] Add netns parameter to fib_lookup.
Posted by [den](#) on Mon, 21 Jan 2008 14:50:41 GMT

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

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

```
include/net/ip_fib.h | 9 +++++----
net/ipv4/fib_frontend.c | 4 +---
net/ipv4/fib_rules.c | 4 +---
net/ipv4/fib_semantics.c | 2 +-
net/ipv4/route.c | 6 +++++
5 files changed, 13 insertions(+), 12 deletions(-)
```

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

```
index 08ebb1e..9daa60b 100644
```

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

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

```
@@ -178,15 +178,16 @@ static inline struct fib_table *fib_new_table(struct net *net, u32 id)
    return fib_get_table(net, id);
}
```

```
-static inline int fib_lookup(const struct flowi *flp, struct fib_result *res)
```

```
+static inline int fib_lookup(struct net *net, const struct flowi *flp,
```

```
+    struct fib_result *res)
```

```
{
    struct fib_table *table;
```

```
- table = fib_get_table(&init_net, RT_TABLE_LOCAL);
```

```
+ table = fib_get_table(net, RT_TABLE_LOCAL);
```

```
    if (!table->tb_lookup(table, flp, res))
```

```
        return 0;
```

```
- table = fib_get_table(&init_net, RT_TABLE_MAIN);
```

```
+ table = fib_get_table(net, RT_TABLE_MAIN);
```

```
    if (!table->tb_lookup(table, flp, res))
```

```
        return 0;
```

```
    return -ENETUNREACH;
```

```
@@ -200,7 +201,7 @@ extern void __net_exit fib4_rules_exit(struct net *net);
```

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

```
#endif
```

```
-extern int fib_lookup(struct flowi *flp, struct fib_result *res);
```

```
+extern int fib_lookup(struct net *n, struct flowi *flp, struct fib_result *res);
```

```
extern struct fib_table *fib_new_table(struct net *net, u32 id);
```

```
extern struct fib_table *fib_get_table(struct net *net, u32 id);
```

```
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
```

```
index 8c0081c..dcd3a28 100644
```

```

--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -256,7 +256,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
    if (in_dev == NULL)
        goto e_inval;

- if (fib_lookup(&fl, &res))
+ if (fib_lookup(&init_net, &fl, &res))
    goto last_resort;
    if (res.type != RTN_UNICAST)
        goto e_inval_res;
@@ -280,7 +280,7 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
    fl.oif = dev->ifindex;

    ret = 0;
- if (fib_lookup(&fl, &res) == 0) {
+ if (fib_lookup(&init_net, &fl, &res) == 0) {
    if (res.type == RTN_UNICAST) {
        *spec_dst = FIB_RES_PREFSRC(res);
        ret = FIB_RES_NH(res).nh_scope >= RT_SCOPE_HOST;
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 2b43002..19274d0 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -54,14 +54,14 @@ u32 fib_rules_tclass(struct fib_result *res)
}
#endif

-int fib_lookup(struct flowi *flp, struct fib_result *res)
+int fib_lookup(struct net *net, struct flowi *flp, struct fib_result *res)
{
    struct fib_lookup_arg arg = {
        .result = res,
    };
    int err;

- err = fib_rules_lookup(init_net.ipv4.rules_ops, flp, 0, &arg);
+ err = fib_rules_lookup(net->ipv4.rules_ops, flp, 0, &arg);
    res->r = arg.rule;

    return err;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 0e08df4..ecd91c6 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -559,7 +559,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
    /* It is not necessary, but requires a bit of thinking */
    if (fl.fl4_scope < RT_SCOPE_LINK)

```

```

    fl.fl4_scope = RT_SCOPE_LINK;
- if ((err = fib_lookup(&fl, &res)) != 0)
+ if ((err = fib_lookup(&init_net, &fl, &res)) != 0)
    return err;
}
err = -EINVAL;
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 162e738..c107bc3 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1559,7 +1559,7 @@ void ip_rt_get_source(u8 *addr, struct rtable *rt)

    if (rt->fl.iif == 0)
        src = rt->rt_src;
- else if (fib_lookup(&rt->fl, &res) == 0) {
+ else if (fib_lookup(&init_net, &rt->fl, &res) == 0) {
    src = FIB_RES_PREFSRC(res);
    fib_res_put(&res);
} else
@@ -1911,7 +1911,7 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
/*
 * Now we are ready to route packet.
 */
- if ((err = fib_lookup(&fl, &res)) != 0) {
+ if ((err = fib_lookup(&init_net, &fl, &res)) != 0) {
    if (!IN_DEV_FORWARD(in_dev))
        goto e_hostunreach;
    goto no_route;
@@ -2363,7 +2363,7 @@ static int ip_route_output_slow(struct rtable **rp, const struct flowi
*oldflp)
    goto make_route;
}

- if (fib_lookup(&fl, &res)) {
+ if (fib_lookup(&init_net, &fl, &res)) {
    res.fi = NULL;
    if (oldflp->oif) {
        /* Apparently, routing tables are wrong. Assume,
--
1.5.3.rc5

```