
Subject: [PATCH] Explicitly call fib_get_table() in fib_frontend.c
Posted by [Pavel Emelianov](#) on Mon, 22 Oct 2007 15:55:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

In case the "multiple tables" config option is y, the ip_fib_local_table is not a variable, but a macro, that calls fib_get_table(RT_TABLE_LOCAL).

Some code uses this "variable" *3* times in one place, thus implicitly making 3 calls. Fix it.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 78b514b..6012390 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -128,13 +128,14 @@ struct net_device * ip_dev_find(__be32 addr)
    struct flowi fl = { .nl_u = { .ip4_u = { .daddr = addr } } };
    struct fib_result res;
    struct net_device *dev = NULL;
+ struct fib_table *local_table;

#ifdef CONFIG_IP_MULTIPLE_TABLES
    res.r = NULL;
#endif

- if (!ip_fib_local_table ||
-     ip_fib_local_table->tb_lookup(ip_fib_local_table, &fl, &res))
+ local_table = fib_get_table(RT_TABLE_LOCAL);
+ if (!local_table || local_table->tb_lookup(local_table, &fl, &res))
    return NULL;
    if (res.type != RTN_LOCAL)
        goto out;
@@ -152,6 +153,7 @@ unsigned inet_addr_type(__be32 addr)
    struct flowi fl = { .nl_u = { .ip4_u = { .daddr = addr } } };
    struct fib_result res;
    unsigned ret = RTN_BROADCAST;
+ struct fib_table *local_table;

    if (ZERONET(addr) || BADCLASS(addr))
        return RTN_BROADCAST;
@@ -162,10 +164,10 @@ unsigned inet_addr_type(__be32 addr)
    res.r = NULL;
#endif

- if (ip_fib_local_table) {
```

```
+ local_table = fib_get_table(RT_TABLE_LOCAL);
+ if (local_table) {
    ret = RTN_UNICAST;
- if (!ip_fib_local_table->tb_lookup(ip_fib_local_table,
-     &fl, &res)) {
+ if (!local_table->tb_lookup(local_table, &fl, &res)) {
    ret = res.type;
    fib_res_put(&res);
}
```
