
Subject: [PATCH net-2.6.25 7/19] [IPV4] Unify access to the routing tables.

Posted by [den](#) on Wed, 19 Dec 2007 15:24:31 GMT

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

Replace the direct pointers to local and main tables with calls to fib_get_table() with appropriate argument.

This doesn't introduce additional dereferences, but makes the access to fib tables uniform in any (CONFIG_IP_MULTIPLE_TABLES) case.

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

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

```
include/net/ip_fib.h | 39 ++++++-----
net/ipv4/fib_frontend.c | 29 ++++++-----
2 files changed, 40 insertions(+), 28 deletions(-)
```

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

```
index 338d3ed..06e6a33 100644
```

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

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

```
@@ -120,16 +120,22 @@ struct fib_result_nl {
    int      err;
};
```

```
+extern struct hlist_head fib_table_hash[];
```

```
+
```

```
#ifdef CONFIG_IP_ROUTE_MULTIPATH
```

```
#define FIB_RES_NH(res) ((res).fi->fib_nh[(res).nh_sel])
```

```
#define FIB_RES_RESET(res) ((res).nh_sel = 0)
```

```
+#define FIB_TABLE_HASHSZ 2
```

```
+
```

```
#else /* CONFIG_IP_ROUTE_MULTIPATH */
```

```
#define FIB_RES_NH(res) ((res).fi->fib_nh[0])
```

```
#define FIB_RES_RESET(res)
```

```
+#define FIB_TABLE_HASHSZ 256
```

```
+
```

```
#endif /* CONFIG_IP_ROUTE_MULTIPATH */
```

```
#define FIB_RES_PREFSRC(res) ((res).fi->fib_prefsrc ? : __fib_res_prefsrc(&res))
```

```
@@ -156,14 +162,17 @@ struct fib_table {
```

```
#ifndef CONFIG_IP_MULTIPLE_TABLES
```

```

-extern struct fib_table *ip_fib_local_table;
-extern struct fib_table *ip_fib_main_table;
+#define TABLE_LOCAL_INDEX 0
+#define TABLE_MAIN_INDEX 1

static inline struct fib_table *fib_get_table(u32 id)
{
- if (id != RT_TABLE_LOCAL)
- return ip_fib_main_table;
- return ip_fib_local_table;
+ struct hlist_head *ptr;
+
+ ptr = id == RT_TABLE_LOCAL ?
+ &fib_table_hash[TABLE_LOCAL_INDEX] :
+ &fib_table_hash[TABLE_MAIN_INDEX];
+ return hlist_entry(ptr->first, struct fib_table, tb_hlist);
}

static inline struct fib_table *fib_new_table(u32 id)
@@ -173,16 +182,24 @@ static inline struct fib_table *fib_new_table(u32 id)

static inline int fib_lookup(const struct flowi *flp, struct fib_result *res)
{
- if (ip_fib_local_table->tb_lookup(ip_fib_local_table, flp, res) &&
- ip_fib_main_table->tb_lookup(ip_fib_main_table, flp, res))
- return -ENETUNREACH;
- return 0;
+ struct fib_table *table;
+
+ table = fib_get_table(RT_TABLE_LOCAL);
+ if (!table->tb_lookup(table, flp, res))
+ return 0;
+
+ table = fib_get_table(RT_TABLE_MAIN);
+ if (!table->tb_lookup(table, flp, res))
+ return 0;
+ return -ENETUNREACH;
}

-static inline void fib_select_default(const struct flowi *flp, struct fib_result *res)
+static inline void fib_select_default(const struct flowi *flp,
+ struct fib_result *res)
{
+ struct fib_table *table = fib_get_table(RT_TABLE_MAIN);
+ if (FIB_RES_GW(*res) && FIB_RES_NH(*res).nh_scope == RT_SCOPE_LINK)
- ip_fib_main_table->tb_select_default(ip_fib_main_table, flp, res);
+ table->tb_select_default(table, flp, res);
}

```

```

#else /* CONFIG_IP_MULTIPLE_TABLES */
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 714fff9..df9716f 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -50,39 +50,34 @@
#define FFprint(a...) printk(KERN_DEBUG a)

static struct sock *fibnl;
+struct hlist_head fib_table_hash[FIB_TABLE_HASHSZ];

#ifdef CONFIG_IP_MULTIPLE_TABLES

-struct fib_table *ip_fib_local_table;
-struct fib_table *ip_fib_main_table;
-
-#define FIB_TABLE_HASHSZ 1
-static struct hlist_head fib_table_hash[FIB_TABLE_HASHSZ];
-
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)
+ struct fib_table *local_table, *main_table;
+
+ local_table = fib_hash_init(RT_TABLE_LOCAL);
+ if (local_table == NULL)
    return -ENOMEM;

- ip_fib_main_table = fib_hash_init(RT_TABLE_MAIN);
- if (ip_fib_main_table == NULL)
+ main_table = fib_hash_init(RT_TABLE_MAIN);
+ if (main_table == NULL)
    goto fail;

- hlist_add_head_rcu(&ip_fib_local_table->tb_hlist, &fib_table_hash[0]);
- hlist_add_head_rcu(&ip_fib_main_table->tb_hlist, &fib_table_hash[0]);
+ hlist_add_head_rcu(&local_table->tb_hlist,
+   &fib_table_hash[RT_TABLE_LOCAL_INDEX]);
+ hlist_add_head_rcu(&main_table->tb_hlist,
+   &fib_table_hash[RT_TABLE_MAIN_INDEX]);
    return 0;

fail:
- kfree(ip_fib_local_table);
- ip_fib_local_table = NULL;
+ kfree(local_table);

```

```
    return -ENOMEM;
}
#else

#define FIB_TABLE_HASHSZ 256
-static struct hlist_head fib_table_hash[FIB_TABLE_HASHSZ];
-
struct fib_table *fib_new_table(u32 id)
{
    struct fib_table *tb;
--
1.5.3.rc5
```
