

---

Subject: [PATCH 2.6.25 3/3] ipv4: last default route is a fib table property

Posted by [den](#) on Fri, 07 Dec 2007 16:11:13 GMT

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

---

ipv4: last default route is a fib table property

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

Acked-by: Alexey Kuznetsov <[kuznet@ms2.inr.ac.ru](mailto:kuznet@ms2.inr.ac.ru)>

---

```
include/net/ip_fib.h | 1 +
net/ipv4/fib_hash.c | 16 ++++++-----
net/ipv4/fib_trie.c | 18 ++++++-----
3 files changed, 18 insertions(+), 17 deletions(-)
```

diff --git a/include/net/ip\_fib.h b/include/net/ip\_fib.h

index 690fb4d..d70b9b4 100644

--- a/include/net/ip\_fib.h

+++ b/include/net/ip\_fib.h

@@ -141,6 +141,7 @@ struct fib\_table {

struct hlist\_node tb\_hlist;

u32 tb\_id;

unsigned tb\_stamp;

+ int tb\_default;

int (\*tb\_lookup)(struct fib\_table \*tb, const struct flowi \*flp, struct fib\_result \*res);

int (\*tb\_insert)(struct fib\_table \*, struct fib\_config \*);

int (\*tb\_delete)(struct fib\_table \*, struct fib\_config \*);

diff --git a/net/ipv4/fib\_hash.c b/net/ipv4/fib\_hash.c

index a52b570..481de47 100644

--- a/net/ipv4/fib\_hash.c

+++ b/net/ipv4/fib\_hash.c

@@ -272,8 +272,6 @@ out:

return err;

}

-static int fn\_hash\_last\_dflt=-1;

-

static void

fn\_hash\_select\_default(struct fib\_table \*tb, const struct flowi \*flp, struct fib\_result \*res)

{

@@ -314,9 +312,9 @@ fn\_hash\_select\_default(struct fib\_table \*tb, const struct flowi \*flp, struct fib

if (next\_fi != res->fi)

break;

} else if (!fib\_detect\_death(fi, order, &last\_resort,

- &last\_idx, fn\_hash\_last\_dflt)) {

+ &last\_idx, tb->tb\_default)) {

fib\_result\_assign(res, fi);

- fn\_hash\_last\_dflt = order;

```

+   tb->tb_default = order;
    goto out;
}
    fi = next_fi;
@@ -325,19 +323,20 @@ fn_hash_select_default(struct fib_table *tb, const struct flowi *flp,
struct fib
{

    if (order <= 0 || fi == NULL) {
-   fn_hash_last_dflt = -1;
+   tb->tb_default = -1;
    goto out;
}

- if (!fib_detect_death(fi, order, &last_resort, &last_idx, fn_hash_last_dflt)) {
+ if (!fib_detect_death(fi, order, &last_resort, &last_idx,
+   tb->tb_default)) {
    fib_result_assign(res, fi);
-   fn_hash_last_dflt = order;
+   tb->tb_default = order;
    goto out;
}

    if (last_idx >= 0)
        fib_result_assign(res, last_resort);
-   fn_hash_last_dflt = last_idx;
+   tb->tb_default = last_idx;
out:
    read_unlock(&fib_hash_lock);
}
@@ -772,6 +771,7 @@ struct fib_table * __init fib_hash_init(u32 id)
    return NULL;

    tb->tb_id = id;
+   tb->tb_default = -1;
    tb->tb_lookup = fn_hash_lookup;
    tb->tb_insert = fn_hash_insert;
    tb->tb_delete = fn_hash_delete;
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 29a06af..850165a 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -1779,8 +1779,6 @@ static int fn_trie_flush(struct fib_table *tb)
    return found;
}

-static int trie_last_dflt = -1;
-

```

```

static void
fn_trie_select_default(struct fib_table *tb, const struct flowi *flp, struct fib_result *res)
{
@@ -1827,28 +1825,29 @@ fn_trie_select_default(struct fib_table *tb, const struct flowi *flp,
struct fib
    if (next_fi != res->fi)
        break;
    } else if (!fib_detect_death(fi, order, &last_resort,
-        &last_idx, trie_last_dflt)) {
+        &last_idx, tb->tb_default)) {
        fib_result_assign(res, fi);
-    trie_last_dflt = order;
+    tb->tb_default = order;
        goto out;
    }
    fi = next_fi;
    order++;
}
if (order <= 0 || fi == NULL) {
-    trie_last_dflt = -1;
+    tb->tb_default = -1;
    goto out;
}

- if (!fib_detect_death(fi, order, &last_resort, &last_idx, trie_last_dflt)) {
+ if (!fib_detect_death(fi, order, &last_resort, &last_idx,
+     tb->tb_default)) {
    fib_result_assign(res, fi);
-    trie_last_dflt = order;
+    tb->tb_default = order;
    goto out;
}
if (last_idx >= 0)
    fib_result_assign(res, last_resort);
-    trie_last_dflt = last_idx;
-    out:;
+    tb->tb_default = last_idx;
+out:
    rcu_read_unlock();
}

@@ -1975,6 +1974,7 @@ struct fib_table * __init fib_hash_init(u32 id)
    return NULL;

    tb->tb_id = id;
+    tb->tb_default = -1;
    tb->tb_lookup = fn_trie_lookup;
    tb->tb_insert = fn_trie_insert;

```

```
tb->tb_delete = fn_trie_delete;
```

```
--
```

1.5.3.rc5

---