
Subject: [PATCH] Consolidate the ip6_pol_route_ (input|output) pair

Posted by [Pavel Emelianov](#) on Mon, 15 Oct 2007 11:55:21 GMT

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

The difference in both functions is in the "id" passed to the rt6_select, so just pass it as an extra argument from two outer helpers.

This is minus 60 lines of code and 360 bytes of .text

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

net/ipv6/route.c | 77 ++++++-----
1 files changed, 9 insertions(+), 68 deletions(-)

diff --git a/net/ipv6/route.c b/net/ipv6/route.c

index 6ff19f9..cce9941 100644

--- a/net/ipv6/route.c

+++ b/net/ipv6/route.c

@@ -663,7 +663,7 @@ static struct rt6_info *rt6_alloc_clone(struct rt6_info *ort, struct in6_addr *d

return rt;
}

-static struct rt6_info *ip6_pol_route_input(struct fib6_table *table,
+static struct rt6_info *ip6_pol_route(struct fib6_table *table, int oif,
struct flowi *fl, int flags)

{
struct fib6_node *fn;

@@ -682,7 +682,7 @@ restart_2:

fn = fib6_lookup(&table->tb6_root, &fl->fl6_dst, &fl->fl6_src);

restart:

- rt = rt6_select(fn, fl->iif, strict | reachable);

+ rt = rt6_select(fn, oif, strict | reachable);

BACKTRACK(&fl->fl6_src);

if (rt == &ip6_null_entry ||
rt->rt6i_flags & RTF_CACHE)

@@ -735,6 +735,12 @@ out2:

return rt;
}

+static struct rt6_info *ip6_pol_route_input(struct fib6_table *table,

+ struct flowi *fl, int flags)

+{

+ return ip6_pol_route(table, fl->iif, fl, flags);

```

+}
+
void ip6_route_input(struct sk_buff *skb)
{
    struct ipv6hdr *iph = ipv6_hdr(skb);
@@ -761,72 +767,7 @@ void ip6_route_input(struct sk_buff *skb)
    static struct rt6_info *ip6_pol_route_output(struct fib6_table *table,
        struct flowi *fl, int flags)
    {
- struct fib6_node *fn;
- struct rt6_info *rt, *nrt;
- int strict = 0;
- int attempts = 3;
- int err;
- int reachable = ipv6_devconf.forwarding ? 0 : RT6_LOOKUP_F_REACHABLE;
-
- strict |= flags & RT6_LOOKUP_F_IFACE;
-
-relookup:
- read_lock_bh(&table->tb6_lock);
-
-restart_2:
- fn = fib6_lookup(&table->tb6_root, &fl->fl6_dst, &fl->fl6_src);
-
-restart:
- rt = rt6_select(fn, fl->oif, strict | reachable);
- BACKTRACK(&fl->fl6_src);
- if (rt == &ip6_null_entry ||
-     rt->rt6i_flags & RTF_CACHE)
-     goto out;
-
- dst_hold(&rt->u.dst);
- read_unlock_bh(&table->tb6_lock);
-
- if (!rt->rt6i_nexthop && !(rt->rt6i_flags & RTF_NONEXTHOP))
-     nrt = rt6_alloc_cow(rt, &fl->fl6_dst, &fl->fl6_src);
- else {
-#if CLONE_OFFLINK_ROUTE
-     nrt = rt6_alloc_clone(rt, &fl->fl6_dst);
-#else
-     goto out2;
-#endif
- }
-
- dst_release(&rt->u.dst);
- rt = nrt ? : &ip6_null_entry;
-
- dst_hold(&rt->u.dst);

```

```

- if (nrt) {
-   err = ip6_ins_rt(nrt);
-   if (!err)
-     goto out2;
- }
-
- if (--attempts <= 0)
-   goto out2;
-
- /*
-  * Race condition! In the gap, when table->tb6_lock was
-  * released someone could insert this route. Relookup.
-  */
- dst_release(&rt->u.dst);
- goto relookup;
-
-out:
- if (reachable) {
-   reachable = 0;
-   goto restart_2;
- }
- dst_hold(&rt->u.dst);
- read_unlock_bh(&table->tb6_lock);
-out2:
- rt->u.dst.lastuse = jiffies;
- rt->u.dst.__use++;
- return rt;
+ return ip6_pol_route(table, fl->oif, fl, flags);
}

struct dst_entry * ip6_route_output(struct sock *sk, struct flowi *fl)

```
