
Subject: [patch 09/10][NETNS][IP6_FIB] move the gc timer to the namespace
Posted by [Daniel Lezcano](#) on Thu, 15 Nov 2007 14:01:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Move the timer definition inside the network namespace. The gc timer is no longer referenced as a global variable but as to be relative to the init_net.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

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

```
include/net/net_namespace.h | 2 ++
net/ipv6/ip6_fib.c          | 37 ++++++-----
2 files changed, 21 insertions(+), 18 deletions(-)
```

Index: linux-2.6-netns/include/net/net_namespace.h

```
=====
--- linux-2.6-netns.orig/include/net/net_namespace.h
+++ linux-2.6-netns/include/net/net_namespace.h
@@ -38,6 +38,8 @@ struct net {
#ifdef CONFIG_IPV6_MULTIPLE_TABLES
    struct fib6_table    *fib6_local_tbl;
#endif /* CONFIG_IPV6_MULTIPLE_TABLES */
+
+ struct timer_list     *ip6_fib_timer;
#endif /* CONFIG_IPV6 */
```

```
    struct sock *rtnl; /* rtnetlink socket */
```

Index: linux-2.6-netns/net/ipv6/ip6_fib.c

```
=====
--- linux-2.6-netns.orig/net/ipv6/ip6_fib.c
+++ linux-2.6-netns/net/ipv6/ip6_fib.c
@@ -93,8 +93,6 @@ static int fib6_walk_continue(struct fib

static __u32 rt_sernum;

-static struct timer_list *ip6_fib_timer;
-
static struct fib6_walker_t fib6_walker_list = {
    .prev = &fib6_walker_list,
    .next = &fib6_walker_list,
@@ -664,15 +662,15 @@ static int fib6_add_rt2node(struct fib6_

static __inline__ void fib6_start_gc(struct rt6_info *rt)
{
- if (ip6_fib_timer->expires == 0 &&
+ if (init_net.ip6_fib_timer->expires == 0 &&
    (rt->rt6i_flags & (RTF_EXPIRES|RTF_CACHE)))
```

```

- mod_timer(ip6_fib_timer, jiffies + ip6_rt_gc_interval);
+ mod_timer(init_net.ip6_fib_timer, jiffies + ip6_rt_gc_interval);
}

void fib6_force_start_gc(void)
{
- if (ip6_fib_timer->expires == 0)
- mod_timer(ip6_fib_timer, jiffies + ip6_rt_gc_interval);
+ if (init_net.ip6_fib_timer->expires == 0)
+ mod_timer(init_net.ip6_fib_timer, jiffies + ip6_rt_gc_interval);
}

/*
@@ -1439,7 +1437,7 @@ void fib6_run_gc(unsigned long expires,
} else {
    local_bh_disable();
    if (!spin_trylock(&fib6_gc_lock)) {
- mod_timer(ip6_fib_timer, jiffies + HZ);
+ mod_timer(init_net.ip6_fib_timer, jiffies + HZ);
    local_bh_enable();
    return;
}
@@ -1452,10 +1450,10 @@ void fib6_run_gc(unsigned long expires,
    fib6_clean_all(net, fib6_age, 0, NULL);

    if (gc_args.more)
- mod_timer(ip6_fib_timer, jiffies + ip6_rt_gc_interval);
+ mod_timer(init_net.ip6_fib_timer, jiffies + ip6_rt_gc_interval);
    else {
- del_timer(ip6_fib_timer);
- ip6_fib_timer->expires = 0;
+ del_timer(init_net.ip6_fib_timer);
+ init_net.ip6_fib_timer->expires = 0;
    }
    spin_unlock_bh(&fib6_gc_lock);
}
@@ -1520,19 +1518,22 @@ static struct pernet_operations fib6_net

void __init fib6_init(void)
{
+ struct timer_list *timer;
+
    fib6_node_kmem = kmem_cache_create("fib6_nodes",
        sizeof(struct fib6_node),
        0, SLAB_HWCACHE_ALIGN|SLAB_PANIC,
        NULL);

- ip6_fib_timer = kzalloc(sizeof(*ip6_fib_timer), GFP_KERNEL);

```

```

- if (!ip6_fib_timer)
+ timer = kzalloc(sizeof(*timer), GFP_KERNEL);
+ if (!timer)
    panic("IPV6: failed to allocate the gc timer\n");

- ip6_fib_timer->function = fib6_gc_timer_cb;
- ip6_fib_timer->expires = 0;
- ip6_fib_timer->data = (unsigned long)&init_net;
- ip6_fib_timer->base = &boot_tvec_bases;
+ timer->function = fib6_gc_timer_cb;
+ timer->expires = 0;
+ timer->data = (unsigned long)&init_net;
+ timer->base = &boot_tvec_bases;
+ init_net.ip6_fib_timer = timer;

    register_pernet_subsys(&fib6_net_ops);
    __rtnl_register(PF_INET6, RTM_GETROUTE, NULL, inet6_dump_fib);
@@ -1540,8 +1541,8 @@ void __init fib6_init(void)

void fib6_gc_cleanup(void)
{
- del_timer(ip6_fib_timer);
- kfree(ip6_fib_timer);
+ del_timer(timer);
+ kfree(timer);
    unregister_pernet_subsys(&fib6_net_ops);
    kmem_cache_destroy(fib6_node_kmem);
}

--

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
