

Subject: [PATCH net-next 1/9] netns: add namespace parameter to rt_cache_flush
Posted by [den](#) on Fri, 04 Jul 2008 13:17:04 GMT

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

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

```
---
include/net/route.h      | 2 +-
net/ipv4/arp.c           | 2 +-
net/ipv4/devinet.c       | 8 +++++---
net/ipv4/fib_frontend.c  | 17 ++++++++-----
net/ipv4/fib_hash.c      | 6 +++---
net/ipv4/fib_rules.c     | 2 +-
net/ipv4/fib_trie.c      | 6 +++---
net/ipv4/route.c         | 8 +++++---
8 files changed, 27 insertions(+), 24 deletions(-)
```

diff --git a/include/net/route.h b/include/net/route.h

index fc836ff..3140cc5 100644

--- a/include/net/route.h

+++ b/include/net/route.h

@@ -111,7 +111,7 @@ struct in_device;

extern int ip_rt_init(void);

extern void ip_rt_redirect(__be32 old_gw, __be32 dst, __be32 new_gw,
 __be32 src, struct net_device *dev);

-extern void rt_cache_flush(int how);

+extern void rt_cache_flush(struct net *net, int how);

extern int __ip_route_output_key(struct net *, struct rtable **, const struct flowi *flp);

extern int ip_route_output_key(struct net *, struct rtable **, struct flowi *flp);

extern int ip_route_output_flow(struct net *, struct rtable **rp, struct flowi *flp, struct sock *sk, int
flags);

diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c

index 20c515a..29df75a 100644

--- a/net/ipv4/arp.c

+++ b/net/ipv4/arp.c

@@ -1197,7 +1197,7 @@ static int arp_netdev_event(struct notifier_block *this, unsigned long
event, void

switch (event) {

case NETDEV_CHANGEADDR:

neigh_changeaddr(&arp_tbl, dev);

- rt_cache_flush(0);

+ rt_cache_flush(dev_net(dev), 0);

break;

default:

break;

diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c

index 9de2514..2e667e2 100644

--- a/net/ipv4/devinet.c

+++ b/net/ipv4/devinet.c

```

@@ -1348,7 +1348,7 @@ static int devinet_sysctl_forward(ctl_table *ctl, int write,
    dev_disable_lro(iddev->dev);
    }
    rtnl_unlock();
-   rt_cache_flush(0);
+   rt_cache_flush(net, 0);
    }
}

@@ -1362,9 +1362,10 @@ int ipv4_doint_and_flush(ctl_table *ctl, int write,
    int *valp = ctl->data;
    int val = *valp;
    int ret = proc_dointvec(ctl, write, filp, buffer, lenp, ppos);
+   struct net *net = ctl->extra2;

    if (write && *valp != val)
-   rt_cache_flush(0);
+   rt_cache_flush(net, 0);

    return ret;
}
@@ -1375,9 +1376,10 @@ int ipv4_doint_and_flush_strategy(ctl_table *table, int __user *name,
int nlen,
{
    int ret = devinet_conf_sysctl(table, name, nlen, oldval, oldlenp,
        newval, newlen);
+   struct net *net = table->extra2;

    if (ret == 1)
-   rt_cache_flush(0);
+   rt_cache_flush(net, 0);

    return ret;
}
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index 5ad01d6..65c1503 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -144,7 +144,7 @@ static void fib_flush(struct net *net)
    }

    if (flushed)
-   rt_cache_flush(-1);
+   rt_cache_flush(net, -1);
    }

    /*
@@ -897,21 +897,22 @@ static void fib_disable_ip(struct net_device *dev, int force)

```

```

{
    if (fib_sync_down_dev(dev, force))
        fib_flush(dev_net(dev));
- rt_cache_flush(0);
+ rt_cache_flush(dev_net(dev), 0);
    arp_ifdown(dev);
}

static int fib_inetaddr_event(struct notifier_block *this, unsigned long event, void *ptr)
{
    struct in_ifaddr *ifa = (struct in_ifaddr*)ptr;
+ struct net_device *dev = ifa->ifa_dev->dev;

    switch (event) {
    case NETDEV_UP:
        fib_add_ifaddr(ifa);
#ifdef CONFIG_IP_ROUTE_MULTIPATH
- fib_sync_up(ifa->ifa_dev->dev);
+ fib_sync_up(dev);
#endif
- rt_cache_flush(-1);
+ rt_cache_flush(dev_net(dev), -1);
        break;
    case NETDEV_DOWN:
        fib_del_ifaddr(ifa);
@@ -919,9 +920,9 @@ static int fib_inetaddr_event(struct notifier_block *this, unsigned long
event,
    /* Last address was deleted from this interface.
       Disable IP.
    */
- fib_disable_ip(ifa->ifa_dev->dev, 1);
+ fib_disable_ip(dev, 1);
    } else {
- rt_cache_flush(-1);
+ rt_cache_flush(dev_net(dev), -1);
    }
    break;
}
@@ -949,14 +950,14 @@ static int fib_netdev_event(struct notifier_block *this, unsigned long
event, void
#ifdef CONFIG_IP_ROUTE_MULTIPATH
    fib_sync_up(dev);
#endif
- rt_cache_flush(-1);
+ rt_cache_flush(dev_net(dev), -1);
    break;
    case NETDEV_DOWN:
        fib_disable_ip(dev, 0);

```

```

    break;
case NETDEV_CHANGEMTU:
case NETDEV_CHANGE:
- rt_cache_flush(0);
+ rt_cache_flush(dev_net(dev), 0);
    break;
}
return NOTIFY_DONE;
diff --git a/net/ipv4/fib_hash.c b/net/ipv4/fib_hash.c
index eec4bf..c8cac6c 100644
--- a/net/ipv4/fib_hash.c
+++ b/net/ipv4/fib_hash.c
@@ -472,7 +472,7 @@ static int fn_hash_insert(struct fib_table *tb, struct fib_config *cfg)

    fib_release_info(fi_drop);
    if (state & FA_S_ACCESSED)
-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlnfo.nl_net, -1);
    rtmsg_fib(RTM_NEWROUTE, key, fa, cfg->fc_dst_len, tb->tb_id,
              &cfg->fc_nlnfo, NLM_F_REPLACE);
    return 0;
@@ -532,7 +532,7 @@ static int fn_hash_insert(struct fib_table *tb, struct fib_config *cfg)

    if (new_f)
        fz->fz_nent++;
-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlnfo.nl_net, -1);

    rtmsg_fib(RTM_NEWROUTE, key, new_fa, cfg->fc_dst_len, tb->tb_id,
              &cfg->fc_nlnfo, 0);
@@ -614,7 +614,7 @@ static int fn_hash_delete(struct fib_table *tb, struct fib_config *cfg)
    write_unlock_bh(&fib_hash_lock);

    if (fa->fa_state & FA_S_ACCESSED)
-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlnfo.nl_net, -1);
    fn_free_alias(fa, f);
    if (kill_fn) {
        fn_free_node(f);
diff --git a/net/ipv4/fib_rules.c b/net/ipv4/fib_rules.c
index 1fb5687..bc05de4 100644
--- a/net/ipv4/fib_rules.c
+++ b/net/ipv4/fib_rules.c
@@ -260,7 +260,7 @@ static size_t fib4_rule_nlmsg_payload(struct fib_rule *rule)

static void fib4_rule_flush_cache(void)
{
-   rt_cache_flush(-1);

```

```

+ rt_cache_flush(&init_net, -1);
}

static struct fib_rules_ops fib4_rules_ops_template = {
diff --git a/net/ipv4/fib_rie.c b/net/ipv4/fib_rie.c
index 394db9c..d16ae46 100644
--- a/net/ipv4/fib_rie.c
+++ b/net/ipv4/fib_rie.c
@@ -1271,7 +1271,7 @@ static int fn_rie_insert(struct fib_table *tb, struct fib_config *cfg)

    fib_release_info(fi_drop);
    if (state & FA_S_ACCESSED)
-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlinf0.nl_net, -1);
    rtmsg_fib(RTM_NEWROUTE, htonl(key), new_fa, plen,
              tb->tb_id, &cfg->fc_nlinf0, NLM_F_REPLACE);

@@ -1316,7 +1316,7 @@ static int fn_rie_insert(struct fib_table *tb, struct fib_config *cfg)
    list_add_tail_rcu(&new_fa->fa_list,
                     (fa ? &fa->fa_list : fa_head));

-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlinf0.nl_net, -1);
    rtmsg_fib(RTM_NEWROUTE, htonl(key), new_fa, plen, tb->tb_id,
              &cfg->fc_nlinf0, 0);
    succeeded:
@@ -1664,7 +1664,7 @@ static int fn_rie_delete(struct fib_table *tb, struct fib_config *cfg)
    trie_leaf_remove(t, l);

    if (fa->fa_state & FA_S_ACCESSED)
-   rt_cache_flush(-1);
+   rt_cache_flush(cfg->fc_nlinf0.nl_net, -1);

    fib_release_info(fa->fa_info);
    alias_free_mem_rcu(fa);
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index fe3a022..cedc366 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -791,7 +791,7 @@ static void rt_cache_invalidate(void)
    * delay < 0 : invalidate cache (fast : entries will be deleted later)
    * delay >= 0 : invalidate & flush cache (can be long)
    */
-void rt_cache_flush(int delay)
+void rt_cache_flush(struct net *net, int delay)
{
    rt_cache_invalidate();
    if (delay >= 0)

```

@@ -2825,7 +2825,7 @@ done:

```
void ip_rt_multicast_event(struct in_device *in_dev)
{
- rt_cache_flush(0);
+ rt_cache_flush(dev_net(in_dev->dev), 0);
}
```

#ifdef CONFIG_SYSCTL

```
@@ -2837,7 +2837,7 @@ static int ipv4_sysctl_rtcache_flush(ctl_table *ctl, int write,
{
    if (write) {
        proc_dointvec(ctl, write, filp, buffer, lenp, ppos);
-    rt_cache_flush(flush_delay);
+    rt_cache_flush(&init_net, flush_delay);
        return 0;
    }
}
```

```
@@ -2857,7 +2857,7 @@ static int ipv4_sysctl_rtcache_flush_strategy(ctl_table *table,
    return -EINVAL;
    if (get_user(delay, (int __user *)newval))
        return -EFAULT;
-    rt_cache_flush(delay);
+    rt_cache_flush(&init_net, delay);
    return 0;
}
```

--

1.5.3.rc5

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
