
Subject: [patch 0/3][NETNS][IPV6] make anycast and ip6_flowlabels per namespace

Posted by [Daniel Lezcano](#) on Wed, 26 Mar 2008 12:28:32 GMT

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

This patch makes the anycast and ip6_flowlabel per namespace.

--

Containers mailing list

Containers@lists.linux-foundation.org

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

Subject: [patch 1/3][NETNS][IPV6] anycast - handle several network namespace

Posted by [Daniel Lezcano](#) on Wed, 26 Mar 2008 12:28:33 GMT

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

Make use of the network namespace information to have this protocol to handle several network namespace.

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

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

```
include/net/addrconf.h | 3 +-
include/net/ipv6.h      | 17 ++++++-----
net/ipv6/af_inet6.c     | 12 ++++++-----
net/ipv6/anycast.c      | 38 ++++++-----
net/ipv6/ndisc.c        | 2 +-
5 files changed, 37 insertions(+), 35 deletions(-)
```

Index: net-2.6.26/net/ipv6/anycast.c

=====

--- net-2.6.26.orig/net/ipv6/anycast.c

+++ net-2.6.26/net/ipv6/anycast.c

```
@@ -82,6 +82,7 @@ int ipv6_sock_ac_join(struct sock *sk, i
    struct net_device *dev = NULL;
    struct inet6_dev *idev;
    struct ipv6_ac_socklist *pac;
+ struct net *net = sock_net(sk);
    int ishost = !ipv6_devconf.forwarding;
    int err = 0;
```

```
@@ -89,7 +90,7 @@ int ipv6_sock_ac_join(struct sock *sk, i
    return -EPERM;
    if (ipv6_addr_is_multicast(addr))
    return -EINVAL;
- if (ipv6_chk_addr(&init_net, addr, NULL, 0))
```

```

+ if (ipv6_chk_addr(net, addr, NULL, 0))
    return -EINVAL;

    pac = sock_kmalloc(sk, sizeof(struct ipv6_ac_socklist), GFP_KERNEL);
@@ -101,7 +102,7 @@ int ipv6_sock_ac_join(struct sock *sk, i
    if (ifindex == 0) {
        struct rt6_info *rt;

- rt = rt6_lookup(&init_net, addr, NULL, 0, 0);
+ rt = rt6_lookup(net, addr, NULL, 0, 0);
        if (rt) {
            dev = rt->rt6i_dev;
            dev_hold(dev);
@@ -112,10 +113,10 @@ int ipv6_sock_ac_join(struct sock *sk, i
        } else {
            /* router, no matching interface: just pick one */

- dev = dev_get_by_flags(&init_net, IFF_UP, IFF_UP|IFF_LOOPBACK);
+ dev = dev_get_by_flags(net, IFF_UP, IFF_UP|IFF_LOOPBACK);
        }
    } else
- dev = dev_get_by_index(&init_net, ifindex);
+ dev = dev_get_by_index(net, ifindex);

    if (dev == NULL) {
        err = -ENODEV;
@@ -176,6 +177,7 @@ int ipv6_sock_ac_drop(struct sock *sk, i
    struct ipv6_pinfo *np = inet6_sk(sk);
    struct net_device *dev;
    struct ipv6_ac_socklist *pac, *prev_pac;
+ struct net *net = sock_net(sk);

    write_lock_bh(&ipv6_sk_ac_lock);
    prev_pac = NULL;
@@ -196,7 +198,7 @@ int ipv6_sock_ac_drop(struct sock *sk, i

    write_unlock_bh(&ipv6_sk_ac_lock);

- dev = dev_get_by_index(&init_net, pac->acl_ifindex);
+ dev = dev_get_by_index(net, pac->acl_ifindex);
    if (dev) {
        ipv6_dev_ac_dec(dev, &pac->acl_addr);
        dev_put(dev);
@@ -210,6 +212,7 @@ void ipv6_sock_ac_close(struct sock *sk)
    struct ipv6_pinfo *np = inet6_sk(sk);
    struct net_device *dev = NULL;
    struct ipv6_ac_socklist *pac;
+ struct net *net = sock_net(sk);

```

```

int prev_index;

write_lock_bh(&ipv6_sk_ac_lock);
@@ -224,7 +227,7 @@ void ipv6_sock_ac_close(struct sock *sk)
    if (pac->acl_ifindex != prev_index) {
        if (dev)
            dev_put(dev);
-   dev = dev_get_by_index(&init_net, pac->acl_ifindex);
+   dev = dev_get_by_index(net, pac->acl_ifindex);
        prev_index = pac->acl_ifindex;
    }
    if (dev)
@@ -422,14 +425,15 @@ static int ipv6_chk_acast_dev(struct net
/*
 * check if given interface (or any, if dev==0) has this anycast address
 */
-int ipv6_chk_acast_addr(struct net_device *dev, struct in6_addr *addr)
+int ipv6_chk_acast_addr(struct net *net, struct net_device *dev,
+ struct in6_addr *addr)
{
    int found = 0;

    if (dev)
        return ipv6_chk_acast_dev(dev, addr);
    read_lock(&dev_base_lock);
-   for_each_netdev(&init_net, dev)
+   for_each_netdev(net, dev)
        if (ipv6_chk_acast_dev(dev, addr)) {
            found = 1;
            break;
@@ -441,6 +445,7 @@ int ipv6_chk_acast_addr(struct net_devic

#ifdef CONFIG_PROC_FS
struct ac6_iter_state {
+ struct seq_net_private p;
    struct net_device *dev;
    struct inet6_dev *idev;
};
@@ -451,9 +456,10 @@ static inline struct ifacaddr6 *ac6_get_
{
    struct ifacaddr6 *im = NULL;
    struct ac6_iter_state *state = ac6_seq_private(seq);
+ struct net *net = seq_file_net(seq);

    state->idev = NULL;
-   for_each_netdev(&init_net, state->dev) {
+   for_each_netdev(net, state->dev) {
        struct inet6_dev *idev;

```

```

    idev = in6_dev_get(state->dev);
    if (!idev)
@@ -551,8 +557,8 @@ static const struct seq_operations ac6_s

static int ac6_seq_open(struct inode *inode, struct file *file)
{
- return seq_open_private(file, &ac6_seq_ops,
-   sizeof(struct ac6_iter_state));
+ return seq_open_net(inode, file, &ac6_seq_ops,
+   sizeof(struct ac6_iter_state));
}

static const struct file_operations ac6_seq_fops = {
@@ -560,20 +566,20 @@ static const struct file_operations ac6_
    .open = ac6_seq_open,
    .read = seq_read,
    .llseek = seq_lseek,
- .release = seq_release_private,
+ .release = seq_release_net,
};

-int __init ac6_proc_init(void)
+int ac6_proc_init(struct net *net)
{
- if (!proc_net_fops_create(&init_net, "anycast6", S_IRUGO, &ac6_seq_fops))
+ if (!proc_net_fops_create(net, "anycast6", S_IRUGO, &ac6_seq_fops))
    return -ENOMEM;

    return 0;
}

-void ac6_proc_exit(void)
+void ac6_proc_exit(struct net *net)
{
- proc_net_remove(&init_net, "anycast6");
+ proc_net_remove(net, "anycast6");
}
#endif

```

Index: net-2.6.26/include/net/ipv6.h

```

=====
--- net-2.6.26.orig/include/net/ipv6.h
+++ net-2.6.26/include/net/ipv6.h
@@ -591,8 +591,8 @@ extern int ip6_mc_msfgget(struct sock *sk
    int __user *optlen);

#ifdef CONFIG_PROC_FS
-extern int ac6_proc_init(void);

```

```

-extern void ac6_proc_exit(void);
+extern int  ac6_proc_init(struct net *net);
+extern void ac6_proc_exit(struct net *net);
extern int  raw6_proc_init(void);
extern void raw6_proc_exit(void);
extern int  tcp6_proc_init(struct net *net);
@@ -607,15 +607,10 @@ extern int snmp6_register_dev(struct ine
extern int snmp6_unregister_dev(struct inet6_dev *idev);

#else
-static inline int snmp6_register_dev(struct inet6_dev *idev)
-{
- return 0;
-}
-
-static inline int snmp6_unregister_dev(struct inet6_dev *idev)
-{
- return 0;
-}
+static inline int ac6_proc_init(struct net *net) { return 0; }
+static inline void ac6_proc_exit(struct net *net) { }
+static inline int snmp6_register_dev(struct inet6_dev *idev) { return 0; }
+static inline int snmp6_unregister_dev(struct inet6_dev *idev) { return 0; }
#endif

```

```

#ifdef CONFIG_SYSCTL

```

```

Index: net-2.6.26/net/ipv6/af_inet6.c

```

```

=====

```

```

--- net-2.6.26.orig/net/ipv6/af_inet6.c

```

```

+++ net-2.6.26/net/ipv6/af_inet6.c

```

```

@@ -862,11 +862,16 @@ static int inet6_net_init(struct net *ne
    err = tcp6_proc_init(net);
    if (err)
        goto proc_tcp6_fail;
+ err = ac6_proc_init(net);
+ if (err)
+     goto proc_ac6_fail;
out:
#endif
    return err;

```

```

#ifdef CONFIG_PROC_FS

```

```

+proc_ac6_fail:
+ tcp6_proc_exit(net);
proc_tcp6_fail:
    udp6_proc_exit(net);
    goto out;
@@ -878,6 +883,7 @@ static void inet6_net_exit(struct net *n

```

```

#ifdef CONFIG_PROC_FS
    udp6_proc_exit(net);
    tcp6_proc_exit(net);
+ ac6_proc_exit(net);
#endif
}

```

```

@@ -965,9 +971,6 @@ static int __init inet6_init(void)
    goto proc_udplite6_fail;
    if (ipv6_misc_proc_init())
        goto proc_misc6_fail;
-
- if (ac6_proc_init())
- goto proc_anycast6_fail;
    if (if6_proc_init())
        goto proc_if6_fail;
#endif
@@ -1039,8 +1042,6 @@ ip6_route_fail:
#ifdef CONFIG_PROC_FS
    if6_proc_exit();
    proc_if6_fail:
- ac6_proc_exit();
- proc_anycast6_fail:
    ipv6_misc_proc_exit();
    proc_misc6_fail:
    udplite6_proc_exit();
@@ -1101,7 +1102,6 @@ static void __exit inet6_exit(void)

```

```

/* Cleanup code parts. */
if6_proc_exit();
- ac6_proc_exit();
    ipv6_misc_proc_exit();
    udplite6_proc_exit();
    raw6_proc_exit();
Index: net-2.6.26/include/net/addrconf.h
=====
--- net-2.6.26.orig/include/net/addrconf.h
+++ net-2.6.26/include/net/addrconf.h
@@ -134,7 +134,8 @@ extern int inet6_ac_check(struct sock *s

```

```

extern int ipv6_dev_ac_inc(struct net_device *dev, struct in6_addr *addr);
extern int __ipv6_dev_ac_dec(struct inet6_dev *idev, struct in6_addr *addr);
-extern int ipv6_chk_acast_addr(struct net_device *dev, struct in6_addr *addr);
+extern int ipv6_chk_acast_addr(struct net *net, struct net_device *dev,
+    struct in6_addr *addr);

/* Device notifier */

```

Index: net-2.6.26/net/ipv6/ndisc.c

```
=====
--- net-2.6.26.orig/net/ipv6/ndisc.c
+++ net-2.6.26/net/ipv6/ndisc.c
@@ -773,7 +773,7 @@ static void ndisc_rcv_ns(struct sk_buff
    return;
}

- if (ipv6_chk_acast_addr(dev, &msg->target) ||
+ if (ipv6_chk_acast_addr(dev_net(dev), dev, &msg->target) ||
    (idev->cnf.forwarding &&
     (ipv6_devconf.proxy_ndp || idev->cnf.proxy_ndp) &&
     (pneigh = pneigh_lookup(&nd_tbl, dev_net(dev),

--
```

Containers mailing list

Containers@lists.linux-foundation.org

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

Subject: [patch 2/3][NETNS][IPV6] flowlabels - make flowlabels per namespace

Posted by [Daniel Lezcano](#) on Wed, 26 Mar 2008 12:28:34 GMT

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

This patch introduces a new member, fl_net, in struct ip6_flowlabel.

This allows to create labels with the same value in different namespaces.

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

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

```
---
include/net/ipv6.h      |   1
net/ipv6/ip6_flowlabel.c |  72 ++++++-----
2 files changed, 57 insertions(+), 16 deletions(-)
```

Index: net-2.6.26/include/net/ipv6.h

```
=====
--- net-2.6.26.orig/include/net/ipv6.h
+++ net-2.6.26/include/net/ipv6.h
@@ -202,6 +202,7 @@ struct ip6_flowlabel
    u32    owner;
    unsigned long    lastuse;
    unsigned long    expires;
+ struct net    *fl_net;
};

#define IPV6_FLOWINFO_MASK __constant_htonl(0xFFFFFFFF)
Index: net-2.6.26/net/ipv6/ip6_flowlabel.c
```

```

=====
--- net-2.6.26.orig/net/ipv6/ip6_flowlabel.c
+++ net-2.6.26/net/ipv6/ip6_flowlabel.c
@@ -62,23 +62,23 @@ static DEFINE_RWLOCK(ip6_fl_lock);
static DEFINE_RWLOCK(ip6_sk_fl_lock);

-static __inline__ struct ip6_flowlabel * __fl_lookup(__be32 label)
+static inline struct ip6_flowlabel * __fl_lookup(struct net *net, __be32 label)
{
    struct ip6_flowlabel *fl;

    for (fl=fl_ht[FL_HASH(label)]; fl; fl = fl->next) {
- if (fl->label == label)
+ if (fl->label == label && fl->fl_net == net)
        return fl;
    }
    return NULL;
}

-static struct ip6_flowlabel * fl_lookup(__be32 label)
+static struct ip6_flowlabel *fl_lookup(struct net *net, __be32 label)
{
    struct ip6_flowlabel *fl;

    read_lock_bh(&ip6_fl_lock);
- fl = __fl_lookup(label);
+ fl = __fl_lookup(net, label);
    if (fl)
        atomic_inc(&fl->users);
    read_unlock_bh(&ip6_fl_lock);
@@ -88,8 +88,10 @@ static struct ip6_flowlabel * fl_lookup(

static void fl_free(struct ip6_flowlabel *fl)
{
- if (fl)
+ if (fl) {
+ release_net(fl->fl_net);
    kfree(fl->opt);
+ }
    kfree(fl);
}

@@ -112,7 +114,6 @@ static void fl_release(struct ip6_flowla
    time_after(ip6_fl_gc_timer.expires, ttd))
    mod_timer(&ip6_fl_gc_timer, ttd);
}
-

```

```

write_unlock_bh(&ip6_fl_lock);
}

```

```

@@ -148,13 +149,34 @@ static void ip6_fl_gc(unsigned long dumm
if (!sched && atomic_read(&fl_size))
    sched = now + FL_MAX_LINGER;
if (sched) {
- ip6_fl_gc_timer.expires = sched;
- add_timer(&ip6_fl_gc_timer);
+ mod_timer(&ip6_fl_gc_timer, sched);
}
write_unlock(&ip6_fl_lock);
}

```

```

-static struct ip6_flowlabel *fl_intern(struct ip6_flowlabel *fl, __be32 label)
+static void ip6_fl_purge(struct net *net)
+{
+ int i;
+
+ write_lock(&ip6_fl_lock);
+ for (i = 0; i <= FL_HASH_MASK; i++) {
+ struct ip6_flowlabel *fl, **flp;
+ flp = &fl_ht[i];
+ while ((fl = *flp) != NULL) {
+ if (fl->fl_net == net && atomic_read(&fl->users) == 0) {
+ *flp = fl->next;
+ fl_free(fl);
+ atomic_dec(&fl_size);
+ continue;
+ }
+ flp = &fl->next;
+ }
+ }
+ write_unlock(&ip6_fl_lock);
+}
+
+static struct ip6_flowlabel *fl_intern(struct net *net,
+ struct ip6_flowlabel *fl, __be32 label)
+{
+ struct ip6_flowlabel *lfl;

```

```

@@ -165,7 +187,7 @@ static struct ip6_flowlabel *fl_intern(s
for (;;) {
    fl->label = htonl(net_random())&IPV6_FLOWLABEL_MASK;
    if (fl->label) {
- lfl = __fl_lookup(fl->label);
+ lfl = __fl_lookup(net, fl->label);
    if (lfl == NULL)

```

```

        break;
    }
@@ -179,7 +201,7 @@ static struct ip6_flowlabel *fl_intern(s
    * done in ipv6_flowlabel_opt - sock is locked, so new entry
    * with the same label can only appear on another sock
    */
- fl = __fl_lookup(fl->label);
+ fl = __fl_lookup(net, fl->label);
    if (fl != NULL) {
        atomic_inc(&fl->users);
        write_unlock_bh(&ip6_fl_lock);
@@ -298,7 +320,8 @@ static int fl6_renew(struct ip6_flowlabel
    }

static struct ip6_flowlabel *
-fl_create(struct in6_flowlabel_req *freq, char __user *optval, int optlen, int *err_p)
+fl_create(struct net *net, struct in6_flowlabel_req *freq, char __user *optval,
+ int optlen, int *err_p)
{
    struct ip6_flowlabel *fl;
    int olen;
@@ -343,6 +366,7 @@ fl_create(struct in6_flowlabel_req *freq
    }
}

+ fl->fl_net = hold_net(net);
    fl->expires = jiffies;
    err = fl6_renew(fl, freq->flr_linger, freq->flr_expires);
    if (err)
@@ -441,6 +465,7 @@ static inline void fl_link(struct ipv6_p
int ipv6_flowlabel_opt(struct sock *sk, char __user *optval, int optlen)
{
    int err;
+ struct net *net = sock_net(sk);
    struct ipv6_pinfo *np = inet6_sk(sk);
    struct in6_flowlabel_req freq;
    struct ipv6_fl_socklist *sfl1=NULL;
@@ -483,7 +508,7 @@ int ipv6_flowlabel_opt(struct sock *sk,
    read_unlock_bh(&ip6_sk_fl_lock);

    if (freq.flr_share == IPV6_FL_S_NONE && capable(CAP_NET_ADMIN)) {
- fl = fl_lookup(freq.flr_label);
+ fl = fl_lookup(net, freq.flr_label);
    if (fl) {
        err = fl6_renew(fl, freq.flr_linger, freq.flr_expires);
        fl_release(fl);
@@ -496,7 +521,7 @@ int ipv6_flowlabel_opt(struct sock *sk,
    if (freq.flr_label & ~IPV6_FLOWLABEL_MASK)

```

```

return -EINVAL;

- fl = fl_create(&freq, optval, optlen, &err);
+ fl = fl_create(net, &freq, optval, optlen, &err);
  if (fl == NULL)
    return err;
  sfl1 = kmalloc(sizeof(*sfl1), GFP_KERNEL);
@@ -518,7 +543,7 @@ int ipv6_flowlabel_opt(struct sock *sk,
  read_unlock_bh(&ip6_sk_fl_lock);

  if (fl1 == NULL)
-   fl1 = fl_lookup(freq.flr_label);
+   fl1 = fl_lookup(net, freq.flr_label);
  if (fl1) {
recheck:
    err = -EEXIST;
@@ -559,7 +584,7 @@ release:
  if (sfl1 == NULL || (err = mem_check(sk)) != 0)
    goto done;

- fl1 = fl_intern(fl, freq.flr_label);
+ fl1 = fl_intern(net, fl, freq.flr_label);
  if (fl1 != NULL)
    goto recheck;

@@ -717,13 +742,28 @@ static inline void ip6_flowlabel_proc_fi
}
#endif

+static inline void ip6_flowlabel_net_exit(struct net *net)
+{
+ ip6_fl_purge(net);
+}
+
+static struct pernet_operations ip6_flowlabel_net_ops = {
+ .exit = ip6_flowlabel_net_exit,
+};
+
+int ip6_flowlabel_init(void)
+{
+ int err;
+
+ err = register_pernet_subsys(&ip6_flowlabel_net_ops);
+ if (err)
+   return err;
+ return ip6_flowlabel_proc_init(&init_net);
+}

```

```
void ip6_flowlabel_cleanup(void)
{
    del_timer(&ip6_fl_gc_timer);
+ unregister_pernet_subsys(&ip6_flowlabel_net_ops);
    ip6_flowlabel_proc_fini(&init_net);
}

--
```

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

Subject: [patch 3/3][NETNS][IPV6] flowlabels - make proc per namespace
Posted by [Daniel Lezcano](#) on Wed, 26 Mar 2008 12:28:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Make /proc/net/ip6_flowlabel show only flow labels belonging to the current network namespace.

Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/ipv6/ip6_flowlabel.c | 39 ++++++-----
1 file changed, 23 insertions(+), 16 deletions(-)

Index: net-2.6.26/net/ipv6/ip6_flowlabel.c

=====

--- net-2.6.26.orig/net/ipv6/ip6_flowlabel.c
+++ net-2.6.26/net/ipv6/ip6_flowlabel.c
@@ -611,6 +611,7 @@ done:
#ifdef CONFIG_PROC_FS

```
struct ip6fl_iter_state {
+ struct seq_net_private p;
    int bucket;
};
```

```
@@ -620,12 +621,15 @@ static struct ip6_flowlabel *ip6fl_get_f
{
    struct ip6_flowlabel *fl = NULL;
    struct ip6fl_iter_state *state = ip6fl_seq_private(seq);
+ struct net *net = seq_file_net(seq);
```

```
    for (state->bucket = 0; state->bucket <= FL_HASH_MASK; ++state->bucket) {
- if (fl_ht[state->bucket]) {
- fl = fl_ht[state->bucket];
```

```

+ fl = fl_ht[state->bucket];
+
+ while (fl && fl->fl_net != net)
+ fl = fl->next;
+ if (fl)
+     break;
- }
+ }
+ return fl;
+ }
@@ -633,12 +637,18 @@ static struct ip6_flowlabel *ip6fl_get_f
static struct ip6_flowlabel *ip6fl_get_next(struct seq_file *seq, struct ip6_flowlabel *fl)
{
+ struct ip6fl_iter_state *state = ip6fl_seq_private(seq);
+ struct net *net = seq_file_net(seq);

+ fl = fl->next;
+try_again:
+ while (fl && fl->fl_net != net)
+ fl = fl->next;
+
+ while (!fl) {
- if (++state->bucket <= FL_HASH_MASK)
+ if (++state->bucket <= FL_HASH_MASK) {
+     fl = fl_ht[state->bucket];
- else
+ goto try_again;
+ } else
+     break;
+ }
+ return fl;
@@ -708,8 +718,8 @@ static const struct seq_operations ip6fl

static int ip6fl_seq_open(struct inode *inode, struct file *file)
{
- return seq_open_private(file, &ip6fl_seq_ops,
-     sizeof(struct ip6fl_iter_state));
+ return seq_open_net(inode, file, &ip6fl_seq_ops,
+     sizeof(struct ip6fl_iter_state));
+ }

static const struct file_operations ip6fl_seq_fops = {
@@ -717,12 +727,13 @@ static const struct file_operations ip6f
+ .open = ip6fl_seq_open,
+ .read = seq_read,
+ .llseek = seq_llseek,
- .release = seq_release_private,
+ .release = seq_release_net,

```

```

};

static int ip6_flowlabel_proc_init(struct net *net)
{
- if (!proc_net_fops_create(net, "ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops))
+ if (!proc_net_fops_create(net, "ip6_flowlabel",
+   S_IRUGO, &ip6fl_seq_fops))
    return -ENOMEM;
    return 0;
}
@@ -745,25 +756,21 @@ static inline void ip6_flowlabel_proc_fi
static inline void ip6_flowlabel_net_exit(struct net *net)
{
    ip6_fl_purge(net);
+ ip6_flowlabel_proc_fini(net);
}

static struct pernet_operations ip6_flowlabel_net_ops = {
+ .init = ip6_flowlabel_proc_init,
    .exit = ip6_flowlabel_net_exit,
};

int ip6_flowlabel_init(void)
{
- int err;
-
- err = register_pernet_subsys(&ip6_flowlabel_net_ops);
- if (err)
-     return err;
- return ip6_flowlabel_proc_init(&init_net);
+ return register_pernet_subsys(&ip6_flowlabel_net_ops);
}

void ip6_flowlabel_cleanup(void)
{
    del_timer(&ip6_fl_gc_timer);
    unregister_pernet_subsys(&ip6_flowlabel_net_ops);
- ip6_flowlabel_proc_fini(&init_net);
}

--

```

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

Subject: Re: [patch 0/3][NETNS][IPV6] make anycast and ip6_flowlabels per namespace

Posted by [davem](#) on Thu, 27 Mar 2008 00:01:05 GMT

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

From: Daniel Lezcano <dlezcano@fr.ibm.com>

Date: Wed, 26 Mar 2008 13:28:32 +0100

> This patch makes the anycast and ip6_flowlabel per namespace.

Applied and pushed out to net-2.6.26, thanks!

Containers mailing list

Containers@lists.linux-foundation.org

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