
Subject: [PATCH 2/4] net: Initialize every fl_net member in struct flowi

Posted by [den](#) on Fri, 30 Nov 2007 16:08:16 GMT

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

net: Initialize every fl_net member in struct flowi

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

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

```
drivers/infiniband/core/addr.c      |  2 ++
drivers/infiniband/hw/cxgb3/iwch_cm.c |  3 ++-
drivers/net/bonding/bond_main.c      |  1 +
include/net/route.h                 |  4 +++-
net/atm/clip.c                      |  5 ++++-
net/bridge/br_netfilter.c           |  1 +
net/dccp/ipv4.c                     |  3 ++-
net/dccp/ipv6.c                     |  5 +++++
net/decnet/af_decnet.c               |  1 +
net/decnet/dn_fib.c                  |  1 +
net/decnet/dn_nsp_out.c              |  1 +
net/decnet/dn_route.c                |  7 +++++--
net/decnet/dn_rules.c                |  5 ++++-
net/ipv4/af_inet.c                   |  1 +
net/ipv4/arp.c                       | 18 ++++++++-----
net/ipv4/fib_frontend.c              | 16 ++++++++-----
net/ipv4/fib_semantics.c             |  1 +
net/ipv4/icmp.c                      |  4 +++-
net/ipv4/igmp.c                      | 12 ++++++-----
net/ipv4/inet_connection_sock.c      |  3 ++-
net/ipv4/ip_gre.c                    | 10 ++++++---
net/ipv4/ip_output.c                 |  6 ++++-
net/ipv4/ipip.c                      |  7 +++++--
net/ipv4/ipmr.c                      |  6 ++++-
net/ipv4/ipvs/ip_vs_xmit.c           |  3 +++
net/ipv4/netfilter.c                 |  1 +
net/ipv4/netfilter/nf_nat_rule.c     |  5 ++++-
net/ipv4/raw.c                       |  3 ++-
net/ipv4/route.c                     |  7 +++++--
net/ipv4/syncookies.c                |  3 ++-
net/ipv4/udp.c                       |  3 ++-
net/ipv4/xfrm4_policy.c              |  2 ++
net/ipv6/af_inet6.c                  |  1 +
net/ipv6/datagram.c                  |  1 +
net/ipv6/icmp.c                      |  2 ++
net/ipv6/inet6_connection_sock.c     |  1 +
net/ipv6/ip6_flowlabel.c             |  1 +
net/ipv6/ip6_tunnel.c                |  1 +
net/ipv6/ipv6_sockglue.c             |  1 +
```

```

net/ipv6/ndisc.c          | 1 +
net/ipv6/netfilter.c      | 1 +
net/ipv6/netfilter/ip6t_REJECT.c | 1 +
net/ipv6/raw.c           | 1 +
net/ipv6/route.c          | 4 +++++
net/ipv6/sit.c            | 7 ++++++--
net/ipv6/tcp_ipv6.c       | 6 ++++++
net/ipv6/udp.c            | 1 +
net/ipv6/xfrm6_policy.c   | 5 ++++-
net/netfilter/nf_conntrack_h323_main.c | 2 ++
net/rxrpc/ar-peer.c      | 1 +
net/sctp/ipv6.c           | 2 ++
net/sctp/protocol.c       | 1 +
net/xfrm/xfrm_policy.c    | 1 +
53 files changed, 151 insertions(+), 41 deletions(-)

```

```
diff --git a/drivers/infiniband/core/addr.c b/drivers/infiniband/core/addr.c
```

```
index 5381c80..3b22bdb 100644
```

```
--- a/drivers/infiniband/core/addr.c
```

```
+++ b/drivers/infiniband/core/addr.c
```

```
@@ -157,6 +157,7 @@ static void addr_send_arp(struct sockaddr_in *dst_in)
    u32 dst_ip = dst_in->sin_addr.s_addr;
```

```

    memset(&fl, 0, sizeof fl);
+ fl.fl_net = &init_net;
    fl.nl_u.ip4_u.daddr = dst_ip;
    if (ip_route_output_key(&rt, &fl))
        return;
```

```
@@ -177,6 +178,7 @@ static int addr_resolve_remote(struct sockaddr_in *src_in,
    int ret;
```

```

    memset(&fl, 0, sizeof fl);
+ fl.fl_net = &init_net;
    fl.nl_u.ip4_u.daddr = dst_ip;
    fl.nl_u.ip4_u.saddr = src_ip;
    ret = ip_route_output_key(&rt, &fl);
```

```
diff --git a/drivers/infiniband/hw/cxgb3/iwch_cm.c b/drivers/infiniband/hw/cxgb3/iwch_cm.c
```

```
index 20ba372..b4d5c2b 100644
```

```
--- a/drivers/infiniband/hw/cxgb3/iwch_cm.c
```

```
+++ b/drivers/infiniband/hw/cxgb3/iwch_cm.c
```

```

@@ -329,7 +329,8 @@ static struct rtable *find_route(struct t3cdev *dev, __be32 local_ip,
    .ports = {
        .sport = local_port,
        .dport = peer_port}
- }
+ },
+ .fl_net = &init_net,
    };

```

```

    if (ip_route_output_flow(&rt, &fl, NULL, 0))
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 423298c..f59aae1 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -2508,6 +2508,7 @@ static void bond_arp_send_all(struct bonding *bond, struct slave
*slave)
    * can tag the ARP with the proper VLAN tag.
    */
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.fl4_dst = targets[i];
    fl.fl4_tos = RTO_ONLINK;

diff --git a/include/net/route.h b/include/net/route.h
index f7ce625..7079e2a 100644
--- a/include/net/route.h
+++ b/include/net/route.h
@@ -147,7 +147,8 @@ static inline int ip_route_connect(struct rtable **rp, __be32 dst,
    __be16 sport, __be16 dport, struct sock *sk,
    int flags)
{
- struct flowi fl = { .oif = oif,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = oif,
     .nl_u = { .ip4_u = { .daddr = dst,
     .saddr = src,
     .tos = tos } },
@@ -178,6 +179,7 @@ static inline int ip_route_newports(struct rtable **rp, u8 protocol,
    struct flowi fl;

    memcpy(&fl, &(*rp)->fl, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.fl_ip_sport = sport;
    fl.fl_ip_dport = dport;
    fl.proto = protocol;
diff --git a/net/atm/clip.c b/net/atm/clip.c
index 741742f..b0c39cb 100644
--- a/net/atm/clip.c
+++ b/net/atm/clip.c
@@ -517,7 +517,10 @@ static int clip_setentry(struct atm_vcc *vcc, __be32 ip)
    struct atmarp_entry *entry;
    int error;
    struct clip_vcc *clip_vcc;
- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = ip, .tos = 1 } } };
+ struct flowi fl = {
+     .fl_net = &init_net,

```

```

+ .nl_u = { .ip4_u = { .daddr = ip, .tos = 1 }}
+ };
  struct rtable *rt;

  if (vcc->push != clip_push) {
diff --git a/net/bridge/br_netfilter.c b/net/bridge/br_netfilter.c
index 5f5b98a..e90951a 100644
--- a/net/bridge/br_netfilter.c
+++ b/net/bridge/br_netfilter.c
@@ -310,6 +310,7 @@ static int br_nf_pre_routing_finish(struct sk_buff *skb)
  if ((err = ip_route_input(skb, iph->daddr, iph->saddr, iph->tos, dev))) {
    struct rtable *rt;
    struct flowi fl = {
+   .fl_net = &init_net,
    .nl_u = {
      .ip4_u = {
        .daddr = iph->daddr,
diff --git a/net/dccp/ipv4.c b/net/dccp/ipv4.c
index db17b83..9cc0f7b 100644
--- a/net/dccp/ipv4.c
+++ b/net/dccp/ipv4.c
@@ -456,7 +456,8 @@ static struct dst_entry* dccp_v4_route_skb(struct sock *sk,
  struct sk_buff *skb)
  {
    struct rtable *rt;
-   struct flowi fl = { .oif = ((struct rtable *)skb->dst)->rt_iif,
+   struct flowi fl = { .fl_net = &init_net,
+   .oif = ((struct rtable *)skb->dst)->rt_iif,
    .nl_u = { .ip4_u =
      { .daddr = ip_hdr(skb)->saddr,
        .saddr = ip_hdr(skb)->daddr,
diff --git a/net/dccp/ipv6.c b/net/dccp/ipv6.c
index 87c98fb..6abcc35 100644
--- a/net/dccp/ipv6.c
+++ b/net/dccp/ipv6.c
@@ -142,6 +142,7 @@ static void dccp_v6_err(struct sk_buff *skb, struct inet6_skb_parm *opt,
  for now.
  */
  memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.proto = IPPROTO_DCCP;
  ipv6_addr_copy(&fl.fl6_dst, &np->daddr);
  ipv6_addr_copy(&fl.fl6_src, &np->saddr);
@@ -242,6 +243,7 @@ static int dccp_v6_send_response(struct sock *sk, struct request_sock
 *req,
  int err = -1;

  memset(&fl, 0, sizeof(fl));

```

```

+ fl.fl_net = &init_net;
  fl.proto = IPPROTO_DCCP;
  ipv6_addr_copy(&fl.fl6_dst, &ireq6->rmt_addr);
  ipv6_addr_copy(&fl.fl6_src, &ireq6->loc_addr);
@@ -320,6 +322,7 @@ static void dccp_v6_ctl_send_reset(struct sock *sk, struct sk_buff *rxskb)
    &rxip6h->daddr);

  memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  ipv6_addr_copy(&fl.fl6_dst, &rxip6h->saddr);
  ipv6_addr_copy(&fl.fl6_src, &rxip6h->daddr);

@@ -536,6 +539,7 @@ static struct sock *dccp_v6_request_rcv_sock(struct sock *sk,
  struct flowi fl;

  memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.proto = IPPROTO_DCCP;
  ipv6_addr_copy(&fl.fl6_dst, &ireq6->rmt_addr);
  if (opt != NULL && opt->srct != NULL) {
@@ -888,6 +892,7 @@ static int dccp_v6_connect(struct sock *sk, struct sockaddr *uaddr,
  return -EAFNOSUPPORT;

  memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

  if (np->sndflow) {
    fl.fl6_flowlabel = usin->sin6_flowinfo & IPV6_FLOWINFO_MASK;
diff --git a/net/deccnet/af_deccnet.c b/net/deccnet/af_deccnet.c
index 57d5749..55450a1 100644
--- a/net/deccnet/af_deccnet.c
+++ b/net/deccnet/af_deccnet.c
@@ -947,6 +947,7 @@ static int __dn_connect(struct sock *sk, struct sockaddr_dn *addr, int
  addrlen,

  err = -EHOSTUNREACH;
  memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.oif = sk->sk_bound_dev_if;
  fl.fld_dst = dn_saddr2dn(&scp->peer);
  fl.fld_src = dn_saddr2dn(&scp->addr);
diff --git a/net/deccnet/dn_fib.c b/net/deccnet/dn_fib.c
index 4aa9a42..e05acf5 100644
--- a/net/deccnet/dn_fib.c
+++ b/net/deccnet/dn_fib.c
@@ -221,6 +221,7 @@ static int dn_fib_check_nh(const struct rtmsg *r, struct dn_fib_info *fi,
  struct
  }

```

```

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.fld_dst = nh->nh_gw;
  fl.oif = nh->nh_oif;
  fl.fld_scope = r->rtm_scope + 1;
diff --git a/net/decnet/dn_nsp_out.c b/net/decnet/dn_nsp_out.c
index 7404653..8bdb9f6 100644
--- a/net/decnet/dn_nsp_out.c
+++ b/net/decnet/dn_nsp_out.c
@@ -91,6 +91,7 @@ @@ try_again:
 }

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.oif = sk->sk_bound_dev_if;
  fl.fld_src = dn_saddr2dn(&scp->addr);
  fl.fld_dst = dn_saddr2dn(&scp->peer);
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index eb07df9..c353de0 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -866,7 +866,8 @@ @@ static inline __le16 dn_fib_rules_map_destination(__le16 daddr, struct
dn_fib_re

static int dn_route_output_slow(struct dst_entry **pprt, const struct flowi *oldflp, int try_hard)
{
- struct flowi fl = { .nl_u = { .dn_u =
+ struct flowi fl = { .fl_net = &init_net,
+   .nl_u = { .dn_u =
       { .daddr = oldflp->fld_dst,
         .saddr = oldflp->fld_src,
         .scope = RT_SCOPE_UNIVERSE,
@@ -1213,7 +1214,8 @@ @@ static int dn_route_input_slow(struct sk_buff *skb)
int flags = 0;
__le16 gateway = 0;
__le16 local_src = 0;
- struct flowi fl = { .nl_u = { .dn_u =
+ struct flowi fl = { .fl_net = &init_net,
+   .nl_u = { .dn_u =
       { .daddr = cb->dst,
         .saddr = cb->src,
         .scope = RT_SCOPE_UNIVERSE,
@@ -1524,6 +1526,7 @@ @@ static int dn_cache_getroute(struct sk_buff *in_skb, struct nlmsghdr
*nlh, void
return -EINVAL;

    memset(&fl, 0, sizeof(fl));

```

```

+ fl.fl_net = &init_net;
  fl.proto = DNPROTO_NSP;

  skb = alloc_skb(NLMSG_GOODSIZE, GFP_KERNEL);
diff --git a/net/decnet/dn_rules.c b/net/decnet/dn_rules.c
index ffebea0..bce33fa 100644
--- a/net/decnet/dn_rules.c
+++ b/net/decnet/dn_rules.c
@@ -175,7 +175,10 @@ static int dn_fib_rule_compare(struct fib_rule *rule, struct fib_rule_hdr
*frh,

  unsigned dnet_addr_type(__le16 addr)
  {
- struct flowi fl = { .nl_u = { .dn_u = { .daddr = addr } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .dn_u = { .daddr = addr } }
+ };
  struct dn_fib_res res;
  unsigned ret = RTN_UNICAST;
  struct dn_fib_table *tb = dn_fib_get_table(RT_TABLE_LOCAL, 0);
diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index c75f20b..1faa09a 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -1089,6 +1089,7 @@ int inet_sk_rebuild_header(struct sock *sk)
  daddr = inet->opt->faddr;
  {
  struct flowi fl = {
+ .fl_net = &init_net,
  .oif = sk->sk_bound_dev_if,
  .nl_u = {
  .ip4_u = {
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index b3f366a..45c59f5 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -420,8 +420,10 @@ static int arp_ignore(struct in_device *in_dev, struct net_device *dev,

  static int arp_filter(__be32 sip, __be32 tip, struct net_device *dev)
  {
- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = sip,
- .saddr = tip } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .ip4_u = { .daddr = sip, .saddr = tip } }
+ };
  struct rtable *rt;

```

```

int flag = 0;
/*unsigned long now; */
@@ -987,8 +989,10 @@ static int arp_req_set(struct arpreq *r, struct net_device * dev)
if (r->arp_flags & ATF_PERM)
r->arp_flags |= ATF_COM;
if (dev == NULL) {
- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = ip,
- .tos = RTO_ONLINK } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .ip4_u = { .daddr = ip, .tos = RTO_ONLINK } }
+ };
struct rtable * rt;
if ((err = ip_route_output_key(&rt, &fl)) != 0)
return err;
@@ -1094,8 +1098,10 @@ static int arp_req_delete(struct arpreq *r, struct net_device * dev)
}

if (dev == NULL) {
- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = ip,
- .tos = RTO_ONLINK } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .ip4_u = { .daddr = ip, .tos = RTO_ONLINK } }
+ };
struct rtable * rt;
if ((err = ip_route_output_key(&rt, &fl)) != 0)
return err;
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index c211887..242e002 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -132,7 +132,10 @@ static void fib_flush(void)

struct net_device * ip_dev_find(__be32 addr)
{
- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = addr } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .ip4_u = { .daddr = addr } }
+ };
struct fib_result res;
struct net_device *dev = NULL;
struct fib_table *local_table;
@@ -157,7 +160,10 @@ out:

unsigned inet_addr_type(__be32 addr)
{

```

```

- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = addr } } };
+ struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = { .ip4_u = { .daddr = addr } }
+ };
    struct fib_result res;
    unsigned ret = RTN_BROADCAST;
    struct fib_table *local_table;
@@ -194,7 +200,8 @@ int fib_validate_source(__be32 src, __be32 dst, u8 tos, int oif,
    struct net_device *dev, __be32 *spec_dst, u32 *itag)
{
    struct in_device *in_dev;
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+ .nl_u = { .ip4_u =
+     { .daddr = src,
+       .saddr = dst,
+       .tos = tos } },
@@ -780,7 +787,8 @@ static void nl_fib_lookup(struct fib_result_nl *frn, struct fib_table *tb)
{
    struct fib_result res;
- struct flowi fl = { .mark = frn->fl_mark,
+ struct flowi fl = { .fl_net = &init_net,
+ .mark = frn->fl_mark,
+ .nl_u = { .ip4_u = { .daddr = frn->fl_addr,
+ .tos = frn->fl_tos,
+ .scope = frn->fl_scope } } };
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index ec9b0dd..e06cdcb 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -544,6 +544,7 @@ static int fib_check_nh(struct fib_config *cfg, struct fib_info *fi,
}
{
    struct flowi fl = {
+ .fl_net = &init_net,
+ .nl_u = {
+ .ip4_u = {
+ .daddr = nh->nh_gw,
diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index c0898c5..176f29d 100644
--- a/net/ipv4/icmp.c
+++ b/net/ipv4/icmp.c
@@ -397,7 +397,8 @@ static void icmp_reply(struct icmp_bxm *icmp_param, struct sk_buff
*skb)
    daddr = icmp_param->replyopts.faddr;
}

```

```

{
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+   .nl_u = { .ip4_u =
      { .daddr = daddr,
        .saddr = rt->rt_spec_dst,
        .tos = RT_TOS(ip_hdr(skb)->tos) } },
@@ -547,6 +548,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)

{
  struct flowi fl = {
+   .fl_net = &init_net,
   .nl_u = {
     .ip4_u = {
       .daddr = icmp_param.replyopts.srr ?
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index c560a93..97b0831 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -297,7 +297,8 @@ static struct sk_buff *igmpv3_newpack(struct net_device *dev, int size)
  return NULL;

{
- struct flowi fl = { .oif = dev->ifindex,
+ struct flowi fl = { .fl_net = &init_net,
+   .oif = dev->ifindex,
   .nl_u = { .ip4_u = {
     .daddr = IGMPV3_ALL_MCR } },
   .proto = IPPROTO_IGMP };
@@ -642,7 +643,8 @@ static int igmp_send_report(struct in_device *in_dev, struct ip_mc_list
*pmc,
  dst = group;

{
- struct flowi fl = { .oif = dev->ifindex,
+ struct flowi fl = { .fl_net = &init_net,
+   .oif = dev->ifindex,
   .nl_u = { .ip4_u = { .daddr = dst } },
   .proto = IPPROTO_IGMP };
  if (ip_route_output_key(&rt, &fl))
@@ -1382,8 +1384,10 @@ void ip_mc_destroy_dev(struct in_device *in_dev)

static struct in_device * ip_mc_find_dev(struct ip_mreqn *imr)
{
- struct flowi fl = { .nl_u = { .ip4_u =
-   { .daddr = imr->imr_multiaddr.s_addr } } };
+ struct flowi fl = {
+   .fl_net = &init_net,

```

```

+ .nl_u = { .ip4_u = { .daddr = imr->imr_multiaddr.s_addr } }
+ };
    struct rtable *rt;
    struct net_device *dev = NULL;
    struct in_device *idev = NULL;
diff --git a/net/ipv4/inet_connection_sock.c b/net/ipv4/inet_connection_sock.c
index 1c2a32f..7fc31c2 100644
--- a/net/ipv4/inet_connection_sock.c
+++ b/net/ipv4/inet_connection_sock.c
@@ -320,7 +320,8 @@ struct dst_entry* inet_csk_route_req(struct sock *sk,
    struct rtable *rt;
    const struct inet_request_sock *ireq = inet_rsk(req);
    struct ip_options *opt = inet_rsk(req)->opt;
- struct flowi fl = { .oif = sk->sk_bound_dev_if,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = sk->sk_bound_dev_if,
     .nl_u = { .ip4_u =
         { .daddr = ((opt && opt->srr) ?
           opt->faddr :
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 02b02a8..289c160 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -475,6 +475,7 @@ out:

    /* Try to guess incoming interface */
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.fl4_dst = eiph->saddr;
    fl.fl4_tos = RT_TOS(eiph->tos);
    fl.proto = IPPROTO_GRE;
@@ -740,7 +741,8 @@ static int ipgre_tunnel_xmit(struct sk_buff *skb, struct net_device *dev)
    }

    {
- struct flowi fl = { .oif = tunnel->parms.link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = tunnel->parms.link,
     .nl_u = { .ip4_u =
         { .daddr = dst,
           .saddr = tiph->saddr,
@@ -1109,7 +1111,8 @@ static int ipgre_open(struct net_device *dev)
    struct ip_tunnel *t = netdev_priv(dev);

    if (MULTICAST(t->parms.iph.daddr)) {
- struct flowi fl = { .oif = t->parms.link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = t->parms.link,

```

```

        .nl_u = { .ip4_u =
            { .daddr = t->parms.iph.daddr,
              .saddr = t->parms.iph.saddr,
@@ -1181,7 +1184,8 @@ static int ipgre_tunnel_init(struct net_device *dev)
    /* Guess output device to choose reasonable mtu and hard_header_len */

    if (iph->daddr) {
- struct flowi fl = { .oif = tunnel->parms.link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = tunnel->parms.link,
        .nl_u = { .ip4_u =
            { .daddr = iph->daddr,
              .saddr = iph->saddr,
diff --git a/net/ipv4/ip_output.c b/net/ipv4/ip_output.c
index 3dc0c12..0f7a8f5 100644
--- a/net/ipv4/ip_output.c
+++ b/net/ipv4/ip_output.c
@@ -335,7 +335,8 @@ int ip_queue_xmit(struct sk_buff *skb, int ipfragok)
    daddr = opt->faddr;

    {
- struct flowi fl = { .oif = sk->sk_bound_dev_if,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = sk->sk_bound_dev_if,
        .nl_u = { .ip4_u =
            { .daddr = daddr,
              .saddr = inet->saddr,
@@ -1368,7 +1369,8 @@ void ip_send_reply(struct sock *sk, struct sk_buff *skb, struct
ip_reply_arg *ar
    }

    {
- struct flowi fl = { .oif = arg->bound_dev_if,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = arg->bound_dev_if,
        .nl_u = { .ip4_u =
            { .daddr = daddr,
              .saddr = rt->rt_spec_dst,
diff --git a/net/ipv4/ipip.c b/net/ipv4/ipip.c
index 8c2b2b0..b91b1a0 100644
--- a/net/ipv4/ipip.c
+++ b/net/ipv4/ipip.c
@@ -402,6 +402,7 @@ out:

    /* Try to guess incoming interface */
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.fl4_daddr = eiph->saddr;

```

```

    fl.fl4_tos = RT_TOS(eiph->tos);
    fl.proto = IPPROTO_IPIP;
@@ -541,7 +542,8 @@ static int ipip_tunnel_xmit(struct sk_buff *skb, struct net_device *dev)
}

{
- struct flowi fl = { .oif = tunnel->parms.link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = tunnel->parms.link,
+     .nl_u = { .ip4_u =
+         { .daddr = dst,
+           .saddr = tiph->saddr,
@@ -805,7 +807,8 @@ static int ipip_tunnel_init(struct net_device *dev)
    memcpy(dev->broadcast, &tunnel->parms.iph.daddr, 4);

    if (iph->daddr) {
- struct flowi fl = { .oif = tunnel->parms.link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = tunnel->parms.link,
+     .nl_u = { .ip4_u =
+         { .daddr = iph->daddr,
+           .saddr = iph->saddr,
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 1187928..524ec2a 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1177,7 +1177,8 @@ static void ipmr_queue_xmit(struct sk_buff *skb, struct mfc_cache *c,
int vifi)
#endif

    if (vif->flags&VIFF_TUNNEL) {
- struct flowi fl = { .oif = vif->link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = vif->link,
+     .nl_u = { .ip4_u =
+         { .daddr = vif->remote,
+           .saddr = vif->local,
@@ -1187,7 +1188,8 @@ static void ipmr_queue_xmit(struct sk_buff *skb, struct mfc_cache *c,
int vifi)
    goto out_free;
    encap = sizeof(struct iphdr);
} else {
- struct flowi fl = { .oif = vif->link,
+ struct flowi fl = { .fl_net = &init_net,
+     .oif = vif->link,
+     .nl_u = { .ip4_u =
+         { .daddr = iph->daddr,
+           .tos = RT_TOS(iph->tos) } },

```

```

diff --git a/net/ipv4/ipvs/ip_vs_xmit.c b/net/ipv4/ipvs/ip_vs_xmit.c
index 1e96bf8..0bda0fa 100644
--- a/net/ipv4/ipvs/ip_vs_xmit.c
+++ b/net/ipv4/ipvs/ip_vs_xmit.c
@@ -70,6 +70,7 @@ __ip_vs_get_out_rt(struct ip_vs_conn *cp, u32 rtos)
    if (!(rt = (struct rtable *)
        __ip_vs_dst_check(dest, rtos, 0))) {
        struct flowi fl = {
+       .fl_net = &init_net,
        .oif = 0,
        .nl_u = {
            .ip4_u = {
@@ -93,6 +94,7 @@ __ip_vs_get_out_rt(struct ip_vs_conn *cp, u32 rtos)
        spin_unlock(&dest->dst_lock);
    } else {
        struct flowi fl = {
+       .fl_net = &init_net,
        .oif = 0,
        .nl_u = {
            .ip4_u = {
@@ -160,6 +162,7 @@ ip_vs_bypass_xmit(struct sk_buff *skb, struct ip_vs_conn *cp,
    u8    tos = iph->tos;
    int    mtu;
    struct flowi fl = {
+   .fl_net = &init_net,
    .oif = 0,
    .nl_u = {
        .ip4_u = {
diff --git a/net/ipv4/netfilter.c b/net/ipv4/netfilter.c
index d902246..744520d 100644
--- a/net/ipv4/netfilter.c
+++ b/net/ipv4/netfilter.c
@@ -18,6 +18,7 @@ int ip_route_me_harder(struct sk_buff *skb, unsigned addr_type)
    unsigned int hh_len;
    unsigned int type;

+   fl.fl_net = &init_net;
    type = inet_addr_type(iph->saddr);
    if (addr_type == RTN_UNSPEC)
        addr_type = type;
diff --git a/net/ipv4/netfilter/nf_nat_rule.c b/net/ipv4/netfilter/nf_nat_rule.c
index ee39ed8..fac1243 100644
--- a/net/ipv4/netfilter/nf_nat_rule.c
+++ b/net/ipv4/netfilter/nf_nat_rule.c
@@ -94,7 +94,10 @@ static unsigned int ipt_snat_target(struct sk_buff *skb,
    static void warn_if_extra_mangle(__be32 dstip, __be32 srcip)
    {
        static int warned = 0;

```

```

- struct flowi fl = { .nl_u = { .ip4_u = { .daddr = dstip } } };
+ struct flowi fl = {
+   .fl_net = &init_net,
+   .nl_u = { .ip4_u = { .daddr = dstip } }
+ };
    struct rtable *rt;

    if (ip_route_output_key(&rt, &fl) != 0)
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index f99828e..e3dec11 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -536,7 +536,8 @@ static int raw_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr
 *msg,
 }

 {
- struct flowi fl = { .oif = ipc.oif,
+ struct flowi fl = { .fl_net = &init_net,
+   .oif = ipc.oif,
+   .nl_u = { .ip4_u =
+     { .daddr = daddr,
+       .saddr = saddr,
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 134cab5..dae1290 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -1777,7 +1777,8 @@ static int ip_route_input_slow(struct sk_buff *skb, __be32 daddr,
__be32 saddr,
{
    struct fib_result res;
    struct in_device *in_dev = in_dev_get(dev);
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+   .nl_u = { .ip4_u =
+     { .daddr = daddr,
+       .saddr = saddr,
+       .tos = tos,
@@ -2156,7 +2157,8 @@ static inline int ip_mkroute_output(struct rtable **rp,
static int ip_route_output_slow(struct rtable **rp, const struct flowi *oldflp)
{
    u32 tos = RT_FL_TOS(oldflp);
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+   .nl_u = { .ip4_u =
+     { .daddr = oldflp->fl4_dst,
+       .saddr = oldflp->fl4_src,
+       .tos = tos & IPTOS_RT_MASK,

```

```

@@ -2618,6 +2620,7 @@ static int inet_rtm_getroute(struct sk_buff *in_skb, struct nlmsg_hdr*
nlh, void
    err = -rt->u.dst.error;
} else {
    struct flowi fl = {
+ .fl_net = &init_net,
    .nl_u = {
        .ip4_u = {
            .daddr = dst,
diff --git a/net/ipv4/syncookies.c b/net/ipv4/syncookies.c
index 2da1be0..346bb33 100644
--- a/net/ipv4/syncookies.c
+++ b/net/ipv4/syncookies.c
@@ -253,7 +253,8 @@ struct sock *cookie_v4_check(struct sock *sk, struct sk_buff *skb,
    * no easy way to do this.
    */
    {
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+ .nl_u = { .ip4_u =
        { .daddr = ((opt && opt->srr) ?
            opt->faddr :
            ireq->rmt_addr),
diff --git a/net/ipv4/udp.c b/net/ipv4/udp.c
index 03c400c..258d826 100644
--- a/net/ipv4/udp.c
+++ b/net/ipv4/udp.c
@@ -633,7 +633,8 @@ int udp_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr *msg,
    rt = (struct rtable*)sk_dst_check(sk, 0);

    if (rt == NULL) {
- struct flowi fl = { .oif = ipc.oif,
+ struct flowi fl = { .fl_net = &init_net,
+ .oif = ipc.oif,
        .nl_u = { .ip4_u =
            { .daddr = faddr,
              .saddr = saddr,
diff --git a/net/ipv4/xfrm4_policy.c b/net/ipv4/xfrm4_policy.c
index b4948c1..19fdf8a 100644
--- a/net/ipv4/xfrm4_policy.c
+++ b/net/ipv4/xfrm4_policy.c
@@ -22,6 +22,7 @@ static struct dst_entry *xfrm4_dst_lookup(int tos, xfrm_address_t *saddr,
xfrm_address_t *daddr)
{
    struct flowi fl = {
+ .fl_net = &init_net,
    .nl_u = {
        .ip4_u = {

```

```

        .tos = tos,
@@ -121,6 +122,7 @@ _decode_session4(struct sk_buff *skb, struct flowi *fl)
    u8 *xprth = skb_network_header(skb) + iph->ihl * 4;

    memset(fl, 0, sizeof(struct flowi));
+ fl->fl_net = &init_net;
    if (!(iph->frag_off & htons(IP_MF | IP_OFFSET))) {
        switch (iph->protocol) {
            case IPPROTO_UDP:
diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index 64135e2..e258d71 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -653,6 +653,7 @@ int inet6_sk_rebuild_header(struct sock *sk)
    struct flowi fl;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = sk->sk_protocol;
    ipv6_addr_copy(&fl.fl6_dst, &np->daddr);
    ipv6_addr_copy(&fl.fl6_src, &np->saddr);
diff --git a/net/ipv6/datagram.c b/net/ipv6/datagram.c
index 2ed689a..5c1f357 100644
--- a/net/ipv6/datagram.c
+++ b/net/ipv6/datagram.c
@@ -60,6 +60,7 @@ int ip6_datagram_connect(struct sock *sk, struct sockaddr *uaddr, int
    addr_len)
    return -EAFNOSUPPORT;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    if (np->sndflow) {
        fl.fl6_flowlabel = usin->sin6_flowinfo&IPV6_FLOWINFO_MASK;
        if (fl.fl6_flowlabel&IPV6_FLOWLABEL_MASK) {
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c
index 4582bbc..070b67f 100644
--- a/net/ipv6/icmp.c
+++ b/net/ipv6/icmp.c
@@ -379,6 +379,7 @@ void icmpv6_send(struct sk_buff *skb, int type, int code, __u32 info,
    mip6_addr_swap(skb);

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = IPPROTO_ICMPV6;
    ipv6_addr_copy(&fl.fl6_dst, &hdr->saddr);
    if (saddr)
@@ -495,6 +496,7 @@ static void icmpv6_echo_reply(struct sk_buff *skb)
    tmp_hdr.icmp6_type = ICMPV6_ECHO_REPLY;

```

```

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.proto = IPPROTO_ICMPV6;
  ipv6_addr_copy(&fl.fl6_dst, &ipv6_hdr(skb)->saddr);
  if (saddr)
diff --git a/net/ipv6/inet6_connection_sock.c b/net/ipv6/inet6_connection_sock.c
index 78de42a..a82bc8d 100644
--- a/net/ipv6/inet6_connection_sock.c
+++ b/net/ipv6/inet6_connection_sock.c
@@ -184,6 +184,7 @@ int inet6_csk_xmit(struct sk_buff *skb, int ipfragok)
    struct in6_addr *final_p = NULL, final;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.proto = sk->sk_protocol;
  ipv6_addr_copy(&fl.fl6_dst, &np->daddr);
  ipv6_addr_copy(&fl.fl6_src, &np->saddr);
diff --git a/net/ipv6/ip6_flowlabel.c b/net/ipv6/ip6_flowlabel.c
index b12cc22..9e30603 100644
--- a/net/ipv6/ip6_flowlabel.c
+++ b/net/ipv6/ip6_flowlabel.c
@@ -329,6 +329,7 @@ int fl_create(struct in6_flowlabel_req *freq, char __user *optval, int optlen,
int *

    msg.msg_controllen = olen;
    msg.msg_control = (void*)(fl->opt+1);
+ flowi.fl_net = &init_net;
  flowi.oif = 0;

  err = datagram_send_ctl(&msg, &flowi, fl->opt, &junk, &junk);
diff --git a/net/ipv6/ip6_tunnel.c b/net/ipv6/ip6_tunnel.c
index 29b5321..bd7ec7c 100644
--- a/net/ipv6/ip6_tunnel.c
+++ b/net/ipv6/ip6_tunnel.c
@@ -530,6 +530,7 @@ int ip4ip6_err(struct sk_buff *skb, struct inet6_skb_parm *opt,

/* Try to guess incoming interface */
memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
  fl.fl4_dst = eiph->saddr;
  fl.fl4_tos = RT_TOS(eiph->tos);
  fl.proto = IPPROTO_IPIP;
diff --git a/net/ipv6/ipv6_sockglue.c b/net/ipv6/ipv6_sockglue.c
index 1334fc1..e4bb3a0 100644
--- a/net/ipv6/ipv6_sockglue.c
+++ b/net/ipv6/ipv6_sockglue.c
@@ -458,6 +458,7 @@ sticky_done:

```

```

    struct flowi fl;
    int junk;

+ fl.fl_net = &init_net;
  fl.fl6_flowlabel = 0;
  fl.oif = sk->sk_bound_dev_if;

diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index 39ff179..eb20314 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -447,6 +447,7 @@ static inline void ndisc_flow_init(struct flowi *fl, u8 type,
    int oif)
    {
        memset(fl, 0, sizeof(*fl));
+ fl->fl_net = &init_net;
        ipv6_addr_copy(&fl->fl6_src, saddr);
        ipv6_addr_copy(&fl->fl6_dst, daddr);
        fl->proto = IPPROTO_ICMPV6;
diff --git a/net/ipv6/netfilter.c b/net/ipv6/netfilter.c
index 175e19f..4da853c 100644
--- a/net/ipv6/netfilter.c
+++ b/net/ipv6/netfilter.c
@@ -14,6 +14,7 @@ int ip6_route_me_harder(struct sk_buff *skb)
    struct ipv6hdr *iph = ipv6_hdr(skb);
    struct dst_entry *dst;
    struct flowi fl = {
+ .fl_net = &init_net,
    .oif = skb->sk ? skb->sk->sk_bound_dev_if : 0,
    .mark = skb->mark,
    .nl_u =
diff --git a/net/ipv6/netfilter/ip6t_REJECT.c b/net/ipv6/netfilter/ip6t_REJECT.c
index 960ba17..ec5f3b1 100644
--- a/net/ipv6/netfilter/ip6t_REJECT.c
+++ b/net/ipv6/netfilter/ip6t_REJECT.c
@@ -87,6 +87,7 @@ static void send_reset(struct sk_buff *oldskb)
    }

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = IPPROTO_TCP;
    ipv6_addr_copy(&fl.fl6_src, &oip6h->daddr);
    ipv6_addr_copy(&fl.fl6_dst, &oip6h->saddr);
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index b34631e..7fae057 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -757,6 +757,7 @@ static int rawv6_sendmsg(struct kiocb *iocb, struct sock *sk,

```

```

* Get and verify the address.
*/
memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

if (sin6) {
    if (addr_len < SIN6_LEN_RFC2133)
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 0f3bfe7..e36cac9 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -554,6 +554,7 @@ struct rt6_info *rt6_lookup(struct in6_addr *daddr, struct in6_addr *saddr,
    int oif, int strict)
{
    struct flowi fl = {
+ .fl_net = &init_net,
    .oif = oif,
    .nl_u = {
        .ip6_u = {
@@ -738,6 +739,7 @@ void ip6_route_input(struct sk_buff *skb)
    struct ipv6hdr *iph = ipv6_hdr(skb);
    int flags = RT6_LOOKUP_F_HAS_SADDR;
    struct flowi fl = {
+ .fl_net = &init_net,
    .iif = skb->dev->ifindex,
    .nl_u = {
        .ip6_u = {
@@ -1369,6 +1371,7 @@ static struct rt6_info *ip6_route_redirect(struct in6_addr *dest,
    int flags = RT6_LOOKUP_F_HAS_SADDR;
    struct ip6rd_flowi rdfl = {
        .fl = {
+ .fl_net = &init_net,
        .oif = dev->ifindex,
        .nl_u = {
            .ip6_u = {
@@ -2174,6 +2177,7 @@ static int inet6_rtm_getroute(struct sk_buff *in_skb, struct nlmsg_hdr*
    nlh, void

    err = -EINVAL;
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

    if (tb[RTA_SRC]) {
        if (nla_len(tb[RTA_SRC]) < sizeof(struct in6_addr))
diff --git a/net/ipv6/sit.c b/net/ipv6/sit.c
index b3b8513..8653d8e 100644
--- a/net/ipv6/sit.c
+++ b/net/ipv6/sit.c

```

```

@@ -391,6 +391,7 @@ isatap_srcok(struct sk_buff *skb, struct iphdr *iph, struct net_device *dev)

    iph6 = ipv6_hdr(skb);
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = iph6->nexthdr;
    ipv6_addr_copy(&fl.fl6_dst, &iph6->saddr);
    fl.oif = dev->ifindex;
@@ -551,7 +552,8 @@ static int ipip6_tunnel_xmit(struct sk_buff *skb, struct net_device *dev)
}

{
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+     .nl_u = { .ip4_u =
        { .daddr = dst,
          .saddr = tiph->saddr,
          .tos = RT_TOS(tos) } },
@@ -822,7 +824,8 @@ static int ipip6_tunnel_init(struct net_device *dev)
    memcpy(dev->broadcast, &tunnel->parms.iph.daddr, 4);

    if (iph->daddr) {
- struct flowi fl = { .nl_u = { .ip4_u =
+ struct flowi fl = { .fl_net = &init_net,
+     .nl_u = { .ip4_u =
        { .daddr = iph->daddr,
          .saddr = iph->saddr,
          .tos = RT_TOS(iph->tos) } },
diff --git a/net/ipv6/tcp_ipv6.c b/net/ipv6/tcp_ipv6.c
index 93980c3..3d87045 100644
--- a/net/ipv6/tcp_ipv6.c
+++ b/net/ipv6/tcp_ipv6.c
@@ -142,6 +142,7 @@ static int tcp_v6_connect(struct sock *sk, struct sockaddr *uaddr,
    return(-EAFNOSUPPORT);

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

    if (np->sndflow) {
        fl.fl6_flowlabel = usin->sin6_flowinfo&IPV6_FLOWINFO_MASK;
@@ -380,6 +381,7 @@ static void tcp_v6_err(struct sk_buff *skb, struct inet6_skb_parm *opt,
    for now.
    */
    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = IPPROTO_TCP;
    ipv6_addr_copy(&fl.fl6_dst, &np->daddr);
    ipv6_addr_copy(&fl.fl6_src, &np->saddr);

```

```

@@ -473,6 +475,7 @@ static int tcp_v6_send_synack(struct sock *sk, struct request_sock *req,
    int err = -1;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = IPPROTO_TCP;
    ipv6_addr_copy(&fl.fl6_dst, &req->rmt_addr);
    ipv6_addr_copy(&fl.fl6_src, &req->loc_addr);
@@ -1052,6 +1055,7 @@ static void tcp_v6_send_reset(struct sock *sk, struct sk_buff *skb)
    buff->csum = csum_partial((char *)t1, sizeof(*t1), 0);

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    ipv6_addr_copy(&fl.fl6_dst, &ipv6_hdr(skb)->saddr);
    ipv6_addr_copy(&fl.fl6_src, &ipv6_hdr(skb)->daddr);

@@ -1153,6 +1157,7 @@ static void tcp_v6_send_ack(struct tcp_timewait_sock *tw,
    buff->csum = csum_partial((char *)t1, tot_len, 0);

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    ipv6_addr_copy(&fl.fl6_dst, &ipv6_hdr(skb)->saddr);
    ipv6_addr_copy(&fl.fl6_src, &ipv6_hdr(skb)->daddr);

@@ -1391,6 +1396,7 @@ static struct sock * tcp_v6_syn_rcv_sock(struct sock *sk, struct
sk_buff *skb,
    struct flowi fl;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
    fl.proto = IPPROTO_TCP;
    ipv6_addr_copy(&fl.fl6_dst, &req->rmt_addr);
    if (opt && opt->srcrt) {
diff --git a/net/ipv6/udp.c b/net/ipv6/udp.c
index ee1cc3f..32bccaa 100644
--- a/net/ipv6/udp.c
+++ b/net/ipv6/udp.c
@@ -652,6 +652,7 @@ do_udp_sendmsg:
    ulen += sizeof(struct udphdr);

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

    if (sin6) {
        if (sin6->sin6_port == 0)
diff --git a/net/ipv6/xfrm6_policy.c b/net/ipv6/xfrm6_policy.c
index 31456c7..cc0d151 100644
--- a/net/ipv6/xfrm6_policy.c

```

```

+++ b/net/ipv6/xfrm6_policy.c
@@ -30,7 +30,9 @@ static struct xfrm_policy_afinfo xfrm6_policy_afinfo;
static struct dst_entry *xfrm6_dst_lookup(int tos, xfrm_address_t *saddr,
xfrm_address_t *daddr)
{
- struct flowi fl = {};
+ struct flowi fl = {
+ .fl_net = &init_net,
+ };
struct dst_entry *dst;
int err;

@@ -133,6 +135,7 @@ _decode_session6(struct sk_buff *skb, struct flowi *fl)
u8 nexthdr = nh[IP6CB(skb)->nhoff];

memset(fl, 0, sizeof(struct flowi));
+ fl->fl_net = &init_net;
ipv6_addr_copy(&fl->fl6_dst, &hdr->daddr);
ipv6_addr_copy(&fl->fl6_src, &hdr->saddr);

diff --git a/net/netfilter/nf_conntrack_h323_main.c b/net/netfilter/nf_conntrack_h323_main.c
index f23fd95..ead805c 100644
--- a/net/netfilter/nf_conntrack_h323_main.c
+++ b/net/netfilter/nf_conntrack_h323_main.c
@@ -713,6 +713,8 @@ static int callforward_do_filter(union nf_conntrack_address *src,

memset(&fl1, 0, sizeof(fl1));
memset(&fl2, 0, sizeof(fl2));
+ fl1.fl_net = &init_net;
+ fl2.fl_net = &init_net;

switch (family) {
case AF_INET: {
diff --git a/net/rxrpc/ar-peer.c b/net/rxrpc/ar-peer.c
index 90fa107..439a4aa 100644
--- a/net/rxrpc/ar-peer.c
+++ b/net/rxrpc/ar-peer.c
@@ -41,6 +41,7 @@ static void rxrpc_assess_MTU_size(struct rxrpc_peer *peer)
peer->if_mtu = 1500;

memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;

switch (peer->srx.transport.family) {
case AF_INET:
diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index 7f31ff6..f76aa61 100644
--- a/net/sctp/ipv6.c

```

```

+++ b/net/sctp/ipv6.c
@@ -201,6 +201,7 @@ static int sctp_v6_xmit(struct sk_buff *skb, struct sctp_transport
*transport,

    memset(&fl, 0, sizeof(fl));

+ fl.fl_net = &init_net;
+ fl.proto = sk->sk_protocol;

/* Fill in the dest address from the route entry passed with the skb
@@ -242,6 +243,7 @@ static struct dst_entry *sctp_v6_get_dst(struct sctp_association *asoc,
struct flowi fl;

    memset(&fl, 0, sizeof(fl));
+ fl.fl_net = &init_net;
+ ipv6_addr_copy(&fl.fl6_dst, &daddr->v6.sin6_addr);
+ if (ipv6_addr_type(&daddr->v6.sin6_addr) & IPV6_ADDR_LINKLOCAL)
+     fl.oif = daddr->v6.sin6_scope_id;
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index d50f610..bdd8afd 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -439,6 +439,7 @@ static struct dst_entry *sctp_v4_get_dst(struct sctp_association *asoc,
union sctp_addr dst_saddr;

    memset(&fl, 0x0, sizeof(struct flowi));
+ fl.fl_net = &init_net;
+ fl.fl4_dst = daddr->v4.sin_addr.s_addr;
+ fl.proto = IPPROTO_SCTP;
+ if (asoc) {
diff --git a/net/xfrm/xfrm_policy.c b/net/xfrm/xfrm_policy.c
index 1a4d70a..505100c 100644
--- a/net/xfrm/xfrm_policy.c
+++ b/net/xfrm/xfrm_policy.c
@@ -1738,6 +1738,7 @@ xfrm_decode_session(struct sk_buff *skb, struct flowi *fl, unsigned
short family
    if (unlikely(afinfo == NULL))
        return -EAFNOSUPPORT;

+ fl->fl_net = &init_net;
+ afinfo->decode_session(skb, fl);
+ err = security_xfrm_decode_session(skb, &fl->secid);
+ xfrm_policy_put_afinfo(afinfo);
--
1.5.3.rc5

```
