Subject: [patch 1/2][NETNS][RFD] store the network namespace pointer in the dst_entry structure
Posted by Daniel Lezcano on Tue, 11 Dec 2007 13:12:32 GMT
View Forum Message <> Reply to Message

Store the network namespace pointer in the dst_entry structure when it is
allocated.
The different protocols redefine the route object as a derivate object from
dst_entry. So using the dst_entry to store the network namespace pointer will
allow to take into account the ipv4, ipv6, dccp protocols in one shot through
the different route objects, rtable, rt6_info, ...

---
 include/net/dst.h | 3 ++-
 net/core/dst.c | 3 ++-
 net/decnet/dn_route.c | 4 ++--
 net/ipv4/route.c | 14 +++++++-------
 net/ipv6/route.c | 18 ++++++++++--------
 net/xfrm/xfrm_policy.c | 2 +-
 6 files changed, 24 insertions(+), 20 deletions(-)

Index: linux-2.6-netns/include/net/dst.h
===
--- linux-2.6-netns.orig/include/net/dst.h
+++ linux-2.6-netns/include/net/dst.h
@@ -81,6 +81,7 @@ struct dst_entry
   struct dn_route *dn_next;
  };
  char   info[0];
+ struct net *net;
 };

@@ -181,7 +182,7 @@ static inline struct dst_entry *dst_pop(
 }

 extern int dst_discard(struct sk_buff *skb);
-extern void * dst_alloc(struct dst_ops * ops);
+extern void * dst_alloc(struct dst_ops * ops, struct net *net);
 extern void __dst_free(struct dst_entry * dst);
 extern struct dst_entry *dst_destroy(struct dst_entry * dst);

Index: linux-2.6-netns/net/core/dst.c
===
--- linux-2.6-netns.orig/net/core/dst.c
+++ linux-2.6-netns/net/core/dst.c
@@ -160,7 +160,7 @@ int dst_discard(struct sk_buff *skb)
 }

```
 EXPORT_SYMBOL(dst_discard);

-void * dst_alloc(struct dst_ops * ops)
+void * dst_alloc(struct dst_ops * ops, struct net *net)
 {
  struct dst_entry * dst;

@@ -176,6 +176,7 @@ void * dst_alloc(struct dst_ops * ops)
  dst->lastuse = jiffies;
  dst->path = dst;
  dst->input = dst->output = dst_discard;
+ dst->net = net;
 #if RT_CACHE_DEBUG >= 2
  atomic_inc(&dst_total);
 #endif
Index: linux-2.6-netns/net/decnet/dn_route.c
===================================================================
--- linux-2.6-netns.orig/net/decnet/dn_route.c
+++ linux-2.6-netns/net/decnet/dn_route.c
@@ -1086,7 +1086,7 @@ make_route:
  if (dev_out->flags & IFF_LOOPBACK)
   flags |= RTCF_LOCAL;

- rt = dst_alloc(&dn_dst_ops);
+ rt = dst_alloc(&dn_dst_ops, &init_net);
  if (rt == NULL)
   goto e_nobufs;

@@ -1350,7 +1350,7 @@ static int dn_route_input_slow(struct sk
  }

 make_route:
- rt = dst_alloc(&dn_dst_ops);
+ rt = dst_alloc(&dn_dst_ops, &init_net);
  if (rt == NULL)
   goto e_nobufs;

Index: linux-2.6-netns/net/ipv4/route.c
===================================================================
--- linux-2.6-netns.orig/net/ipv4/route.c
+++ linux-2.6-netns/net/ipv4/route.c
@@ -1115,7 +1115,7 @@ void ip_rt_redirect(__be32 old_gw, __be3
     dst_hold(&rth->u.dst);
     rcu_read_unlock();

-    rt = dst_alloc(&ipv4_dst_ops);
+    rt = dst_alloc(&ipv4_dst_ops, dev->nd_net);
     if (rt == NULL) {
```

```
      ip_rt_put(rth);
      in_dev_put(in_dev);
@@ -1565,7 +1565,7 @@ static int ip_route_input_mc(struct sk_b
      dev, &spec_dst, &itag) < 0)
   goto e_inval;

- rth = dst_alloc(&ipv4_dst_ops);
+ rth = dst_alloc(&ipv4_dst_ops, dev->nd_net);
  if (!rth)
   goto e_nobufs;

@@ -1704,7 +1704,7 @@ static inline int __mkroute_input(struct
  }


- rth = dst_alloc(&ipv4_dst_ops);
+ rth = dst_alloc(&ipv4_dst_ops, in_dev->dev->nd_net);
  if (!rth) {
   err = -ENOBUFS;
   goto cleanup;
@@ -1888,7 +1888,7 @@ brd_input:
  RT_CACHE_STAT_INC(in_brd);

 local_input:
- rth = dst_alloc(&ipv4_dst_ops);
+ rth = dst_alloc(&ipv4_dst_ops, net);
  if (!rth)
   goto e_nobufs;

@@ -2079,7 +2079,7 @@ static inline int __mkroute_output(struc
  }


- rth = dst_alloc(&ipv4_dst_ops);
+ rth = dst_alloc(&ipv4_dst_ops, dev_out->nd_net);
  if (!rth) {
   err = -ENOBUFS;
   goto cleanup;
@@ -2413,9 +2413,9 @@ static struct dst_ops ipv4_dst_blackhole
 static int ipv4_dst_blackhole(struct rtable **rp, struct flowi *flp, struct sock *sk)
 {
  struct rtable *ort = *rp;
- struct rtable *rt = (struct rtable *)
-  dst_alloc(&ipv4_dst_blackhole_ops);
+ struct rtable *rt;

+ rt = (struct rtable *) dst_alloc(&ipv4_dst_blackhole_ops, &init_net);
  if (rt) {
```

```
   struct dst_entry *new = &rt->u.dst;

Index: linux-2.6-netns/net/ipv6/route.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/route.c
+++ linux-2.6-netns/net/ipv6/route.c
@@ -195,9 +195,9 @@ struct rt6_info ip6_blk_hole_entry = {
 #endif

 /* allocate dst with ip6_dst_ops */
-static __inline__ struct rt6_info *ip6_dst_alloc(void)
+static __inline__ struct rt6_info *ip6_dst_alloc(struct net *net)
 {
- return (struct rt6_info *)dst_alloc(&ip6_dst_ops);
+ return (struct rt6_info *)dst_alloc(&ip6_dst_ops, net);
 }

 static void ip6_dst_destroy(struct dst_entry *dst)
@@ -790,10 +790,11 @@ EXPORT_SYMBOL(ip6_route_output);
 int ip6_dst_blackhole(struct sock *sk, struct dst_entry **dstp, struct flowi *fl)
 {
  struct rt6_info *ort = (struct rt6_info *) *dstp;
- struct rt6_info *rt = (struct rt6_info *)
-  dst_alloc(&ip6_dst_blackhole_ops);
  struct dst_entry *new = NULL;
+ struct rt6_info *rt;
+ struct net *net = &init_net;

+ rt = (struct rt6_info *) dst_alloc(&ip6_dst_blackhole_ops, net);
  if (rt) {
   new = &rt->u.dst;

@@ -923,7 +924,7 @@ struct dst_entry *ndisc_dst_alloc(struct
  if (unlikely(idev == NULL))
   return NULL;

- rt = ip6_dst_alloc();
+ rt = ip6_dst_alloc(dev->nd_net);
  if (unlikely(rt == NULL)) {
   in6_dev_put(idev);
   goto out;
@@ -1054,6 +1055,7 @@ int ip6_route_add(struct fib6_config *cf
  struct net_device *dev = NULL;
  struct inet6_dev *idev = NULL;
  struct fib6_table *table;
+ struct net *net = &init_net;
  int addr_type;
```

```
  if (cfg->fc_dst_len > 128 || cfg->fc_src_len > 128)
@@ -1081,7 +1083,7 @@ int ip6_route_add(struct fib6_config *cf
   goto out;
  }

- rt = ip6_dst_alloc();
+ rt = ip6_dst_alloc(net);

  if (rt == NULL) {
   err = -ENOMEM;
@@ -1560,7 +1562,7 @@ out:

 static struct rt6_info * ip6_rt_copy(struct rt6_info *ort)
 {
- struct rt6_info *rt = ip6_dst_alloc();
+ struct rt6_info *rt = ip6_dst_alloc(ort->u.dst.net);

  if (rt) {
   rt->u.dst.input = ort->u.dst.input;
@@ -1828,7 +1830,7 @@ struct rt6_info *addrconf_dst_alloc(stru
         const struct in6_addr *addr,
         int anycast)
 {
- struct rt6_info *rt = ip6_dst_alloc();
+ struct rt6_info *rt = ip6_dst_alloc(idev->dev->nd_net);

  if (rt == NULL)
   return ERR_PTR(-ENOMEM);
Index: linux-2.6-netns/net/xfrm/xfrm_policy.c
===================================================================
--- linux-2.6-netns.orig/net/xfrm/xfrm_policy.c
+++ linux-2.6-netns/net/xfrm/xfrm_policy.c
@@ -1259,7 +1259,7 @@ static inline struct xfrm_dst *xfrm_allo
  if (!afinfo)
   return ERR_PTR(-EINVAL);

- xdst = dst_alloc(afinfo->dst_ops) ?: ERR_PTR(-ENOBUFS);
+ xdst = dst_alloc(afinfo->dst_ops, &init_net) ?: ERR_PTR(-ENOBUFS);

  xfrm_policy_put_afinfo(afinfo);


--
```