Subject: [patch 05/38][IPV6] inet6_addr - make use of the new ipv6_chk_addr function
Posted by Daniel Lezcano on Mon, 03 Dec 2007 16:16:41 GMT
View Forum Message <> Reply to Message

The previous patch changed the ipv6_chk_addr API and tried
to be focused on the network namespace check. A wrapper avoided
to have the differents caller to be modified.

This patch removes the wrapper and propagate the function API
change to all callers.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
---
 include/net/addrconf.h | 9 +--------
 net/ipv6/addrconf.c | 4 ++--
 net/ipv6/af_inet6.c | 2 +-
 net/ipv6/anycast.c | 2 +-
 net/ipv6/datagram.c | 2 +-
 net/ipv6/icmp.c | 2 +-
 net/ipv6/ip6_tunnel.c | 8 ++++++----
 net/ipv6/ndisc.c | 2 +-
 net/ipv6/raw.c | 2 +-
 net/sctp/ipv6.c | 5 +++--
 10 files changed, 16 insertions(+), 22 deletions(-)

Index: linux-2.6-netns/include/net/addrconf.h
===
--- linux-2.6-netns.orig/include/net/addrconf.h
+++ linux-2.6-netns/include/net/addrconf.h
@@ -58,18 +58,11 @@ extern int   addrconf_add_ifaddr(void __
 extern int   addrconf_del_ifaddr(void __user *arg);
 extern int   addrconf_set_dstaddr(void __user *arg);

-extern int   __ipv6_chk_addr(struct net *net,
+extern int   ipv6_chk_addr(struct net *net,
           struct in6_addr *addr,
           struct net_device *dev,
           int strict);

-static inline int ipv6_chk_addr(struct in6_addr *addr,
-    struct net_device *dev,
-    int strict)
-{
- return __ipv6_chk_addr(&init_net, addr, dev, strict);
-}
-

```
 #if defined(CONFIG_IPV6_MIP6) || defined(CONFIG_IPV6_MIP6_MODULE)
 extern int   ipv6_chk_home_addr(struct in6_addr *addr);
 #endif
Index: linux-2.6-netns/net/ipv6/addrconf.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/addrconf.c
+++ linux-2.6-netns/net/ipv6/addrconf.c
@@ -1181,7 +1181,7 @@ static int ipv6_count_addresses(struct i
  return cnt;
 }

-int __ipv6_chk_addr(struct net *net, struct in6_addr *addr,
+int ipv6_chk_addr(struct net *net, struct in6_addr *addr,
     struct net_device *dev, int strict)
 {
  struct inet6_ifaddr * ifp;
@@ -1202,7 +1202,7 @@ int __ipv6_chk_addr(struct net *net, str
  return ifp != NULL;
 }

-EXPORT_SYMBOL(__ipv6_chk_addr);
+EXPORT_SYMBOL(ipv6_chk_addr);

 static
 int ipv6_chk_same_addr(const struct in6_addr *addr, struct net_device *dev)
Index: linux-2.6-netns/net/ipv6/af_inet6.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/af_inet6.c
+++ linux-2.6-netns/net/ipv6/af_inet6.c
@@ -314,7 +314,7 @@ int inet6_bind(struct socket *sock, stru
     */
    v4addr = LOOPBACK4_IPV6;
    if (!(addr_type & IPV6_ADDR_MULTICAST)) {
-    if (!ipv6_chk_addr(&addr->sin6_addr, dev, 0)) {
+    if (!ipv6_chk_addr(&init_net, &addr->sin6_addr, dev, 0)) {
      if (dev)
       dev_put(dev);
      err = -EADDRNOTAVAIL;
Index: linux-2.6-netns/net/ipv6/anycast.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/anycast.c
+++ linux-2.6-netns/net/ipv6/anycast.c
@@ -89,7 +89,7 @@ int ipv6_sock_ac_join(struct sock *sk, i
   return -EPERM;
  if (ipv6_addr_is_multicast(addr))
   return -EINVAL;
- if (ipv6_chk_addr(addr, NULL, 0))
+ if (ipv6_chk_addr(&init_net, addr, NULL, 0))
```

```
   return -EINVAL;

  pac = sock_kmalloc(sk, sizeof(struct ipv6_ac_socklist), GFP_KERNEL);
Index: linux-2.6-netns/net/ipv6/datagram.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/datagram.c
+++ linux-2.6-netns/net/ipv6/datagram.c
@@ -550,7 +550,7 @@ int datagram_send_ctl(struct msghdr *msg
       return -ENODEV;
     }
    }
-   if (!ipv6_chk_addr(&src_info->ipi6_addr, dev, 0)) {
+   if (!ipv6_chk_addr(&init_net, &src_info->ipi6_addr, dev, 0)) {
     if (dev)
      dev_put(dev);
     err = -EINVAL;
Index: linux-2.6-netns/net/ipv6/icmp.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/icmp.c
+++ linux-2.6-netns/net/ipv6/icmp.c
@@ -331,7 +331,7 @@ void icmpv6_send(struct sk_buff *skb, in
   */
  addr_type = ipv6_addr_type(&hdr->daddr);

- if (ipv6_chk_addr(&hdr->daddr, skb->dev, 0))
+ if (ipv6_chk_addr(&init_net, &hdr->daddr, skb->dev, 0))
   saddr = &hdr->daddr;

  /*
Index: linux-2.6-netns/net/ipv6/ip6_tunnel.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/ip6_tunnel.c
+++ linux-2.6-netns/net/ipv6/ip6_tunnel.c
@@ -654,8 +654,8 @@ static inline int ip6_tnl_rcv_ctl(struct
    ldev = dev_get_by_index(&init_net, p->link);

   if ((ipv6_addr_is_multicast(&p->laddr) ||
-       likely(ipv6_chk_addr(&p->laddr, ldev, 0))) &&
-      likely(!ipv6_chk_addr(&p->raddr, NULL, 0)))
+       likely(ipv6_chk_addr(&init_net, &p->laddr, ldev, 0))) &&
+      likely(!ipv6_chk_addr(&init_net, &p->raddr, NULL, 0)))
    ret = 1;

   if (ldev)
@@ -789,12 +789,12 @@ static inline int ip6_tnl_xmit_ctl(struc
   if (p->link)
    ldev = dev_get_by_index(&init_net, p->link);
```

```
-  if (unlikely(!ipv6_chk_addr(&p->laddr, ldev, 0)))
+  if (unlikely(!ipv6_chk_addr(&init_net, &p->laddr, ldev, 0)))
     printk(KERN_WARNING
            "%s xmit: Local address not yet configured!\n",
            p->name);
   else if (!ipv6_addr_is_multicast(&p->raddr) &&
-    unlikely(ipv6_chk_addr(&p->raddr, NULL, 0)))
+    unlikely(ipv6_chk_addr(&init_net, &p->raddr, NULL, 0)))
     printk(KERN_WARNING
            "%s xmit: Routing loop! "
            "Remote address found on this node!\n",
Index: linux-2.6-netns/net/ipv6/ndisc.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/ndisc.c
+++ linux-2.6-netns/net/ipv6/ndisc.c
@@ -654,7 +654,7 @@ static void ndisc_solicit(struct neighbo
  struct in6_addr *target = (struct in6_addr *)&neigh->primary_key;
  int probes = atomic_read(&neigh->probes);

- if (skb && ipv6_chk_addr(&ipv6_hdr(skb)->saddr, dev, 1))
+ if (skb && ipv6_chk_addr(&init_net, &ipv6_hdr(skb)->saddr, dev, 1))
   saddr = &ipv6_hdr(skb)->saddr;

  if ((probes -= neigh->parms->ucast_probes) < 0) {
Index: linux-2.6-netns/net/ipv6/raw.c
===================================================================
--- linux-2.6-netns.orig/net/ipv6/raw.c
+++ linux-2.6-netns/net/ipv6/raw.c
@@ -298,7 +298,7 @@ static int rawv6_bind(struct sock *sk, s
   v4addr = LOOPBACK4_IPV6;
   if (!(addr_type & IPV6_ADDR_MULTICAST)) {
    err = -EADDRNOTAVAIL;
-   if (!ipv6_chk_addr(&addr->sin6_addr, dev, 0)) {
+   if (!ipv6_chk_addr(&init_net, &addr->sin6_addr, dev, 0)) {
     if (dev)
      dev_put(dev);
     goto out;
Index: linux-2.6-netns/net/sctp/ipv6.c
===================================================================
--- linux-2.6-netns.orig/net/sctp/ipv6.c
+++ linux-2.6-netns/net/sctp/ipv6.c
@@ -558,7 +558,7 @@ static int sctp_v6_available(union sctp_
  if (!(type & IPV6_ADDR_UNICAST))
   return 0;

- return ipv6_chk_addr(in6, NULL, 0);
+ return ipv6_chk_addr(&init_net, in6, NULL, 0);
 }
```

```
 /* This function checks if the address is a valid address to be used for
@@ -860,7 +860,8 @@ static int sctp_inet6_bind_verify(struct
    dev = dev_get_by_index(&init_net, addr->v6.sin6_scope_id);
    if (!dev)
     return 0;
-   if (!ipv6_chk_addr(&addr->v6.sin6_addr, dev, 0)) {
+   if (!ipv6_chk_addr(&init_net, &addr->v6.sin6_addr,
+         dev, 0)) {
     dev_put(dev);
     return 0;
    }

--
```