
Subject: [patch 12/20] [Network namespace] When no source address is specified, search from the dev list the

Posted by Daniel Lezcano on Sun, 10 Dec 2006 21:58:29 GMT

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

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

```
include/linux/net_namespace.h | 14 ++++++++
net/core/net_namespace.c    | 55 ++++++++++++++++++++++++++++++++
net/ipv4/route.c           | 27 ++++++++
3 files changed, 86 insertions(+), 10 deletions(-)
```

Index: 2.6.19-rc6-mm2/net/ipv4/route.c

=====

```
--- 2.6.19-rc6-mm2.orig/net/ipv4/route.c
+++ 2.6.19-rc6-mm2/net/ipv4/route.c
@@ -2472,17 +2472,17 @@ static int ip_route_output_slow(struct r
```

```
if (LOCAL_MCAST(olddfip->fl4_dst) || oldfip->fl4_dst == htonl(0xFFFFFFFF)) {
    if (!fl.fl4_src)
-     fl.fl4_src = inet_select_addr(dev_out, 0,
-                                   RT_SCOPE_LINK);
+     fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+                                   RT_SCOPE_LINK);
    goto make_route;
}
if (!fl.fl4_src) {
    if (MULTICAST(olddfip->fl4_dst))
-     fl.fl4_src = inet_select_addr(dev_out, 0,
-                                   fl.fl4_scope);
+     fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+                                   fl.fl4_scope);
    else if (!olddfip->fl4_dst)
-     fl.fl4_src = inet_select_addr(dev_out, 0,
-                                   RT_SCOPE_HOST);
+     fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+                                   RT_SCOPE_HOST);
}
}
```

```
@@ -2522,8 +2522,8 @@ static int ip_route_output_slow(struct r
 */
```

```
if (fl.fl4_src == 0)
- fl.fl4_src = inet_select_addr(dev_out, 0,
-                               RT_SCOPE_LINK);
```

```

+ fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+     RT_SCOPE_LINK);
    res.type = RTN_UNICAST;
    goto make_route;
}
@@ -2536,7 +2536,12 @@ static int ip_route_output_slow(struct r

if (res.type == RTN_LOCAL) {
    if (!fl.fl4_src)
+ifdef CONFIG_NET
+    fl.fl4_src = net_ns_select_source_address(dev_out, 0,
+        RT_SCOPE_LINK);
+else
    fl.fl4_src = fl.fl4_dst;
+endif
    if (dev_out)
        dev_put(dev_out);
    dev_out = &loopback_dev;
@@ -2558,8 +2563,10 @@ static int ip_route_output_slow(struct r
    fib_select_default(&fl, &res);

    if (!fl.fl4_src)
- fl.fl4_src = FIB_RES_PREFSRC(res);
-
+ fl.fl4_src = res.fi->fib_prefsrc ? :
+    SELECT_SRC_ADDR(FIB_RES_DEV(res),
+        FIB_RES_GW(res),
+        res.scope);
    if (dev_out)
        dev_put(dev_out);
    dev_out = FIB_RES_DEV(res);
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h
=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -4,6 +4,7 @@ @
#include <linux/kref.h>
#include <linux/nsproxy.h>
#include <linux/errno.h>
+#include <linux/types.h>

struct net_ns_net_device {
    struct hlist_head *name_head;
@@ -90,6 +91,11 @@ extern int net_ns_ioctl(unsigned int cmd

extern int net_ns_check_bind(int addr_type, u32 addr);

+extern __be32 net_ns_select_source_address(const struct net_device *dev,

```

```

+     u32 dst, int scope);
+
+">#define SELECT_SRC_ADDR net_ns_select_source_address
+
+#else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)
@@ -146,6 +152,14 @@ static inline int net_ns_check_bind(int
    return 0;
}

+static inline __be32 net_ns_select_source_address(struct net_device *dev,
+     u32 dst, int scope)
+{
+    return 0;
+}
+
+/#define SELECT_SRC_ADDR inet_select_addr
+
#endif /* !CONFIG_NET_NS */

#endif /* _LINUX_NET_NAMESPACE_H */
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -320,4 +320,59 @@ out:
    return ret;
}

+_be32 net_ns_select_source_address(const struct net_device *dev,
+     u32 dst, int scope)
+{
+    __be32 addr = 0;
+    struct in_device *in_dev;
+    struct net_namespace *net_ns = current_net_ns;
+
+    if (!dev)
+        goto no_dev;
+
+    rCU_read_lock();
+    in_dev = __in_dev_get_rcu(dev);
+    if (!in_dev)
+        goto no_in_dev;
+
+    for_ifa(in_dev) {
+        if (ifa->ifa_scope > scope)
+            continue;
+
+        if (ifa->ifa_scope == scope)
+            addr = ifa->ifa_local;
+
+        if (ifa->ifa_scope < scope)
+            break;
+
+    }
+
+    if (addr == 0)
+        goto no_src;
+
+    if (addr > 0xffffffff)
+        goto no_src;
+
+    return addr;
+
+no_src:
+    return 0;
+
+no_in_dev:
+    rCU_read_unlock();
+    goto out;
+
+no_dev:
+    rCU_read_unlock();
+    goto out;
+
+no_src:
+    return 0;
}

```

```

+ if (ifa->ifa_net_ns != net_ns)
+ continue;
+ if (!dst || inet_ifa_match(dst, ifa)) {
+   addr = ifa->ifa_local;
+   break;
+ }
+ if (!addr)
+   addr = ifa->ifa_local;
+ } endfor_ifa(in_dev);
+no_in_dev:
+rcu_read_unlock();
+
+ if (addr)
+   goto out;
+
+no_dev:
+read_lock(&dev_base_lock);
+rcu_read_lock();
+for (dev = dev_base; dev; dev = dev->next) {
+ if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
+   continue;
+
+ for_ifa(in_dev) {
+   if (ifa->ifa_scope != RT_SCOPE_LINK &&
+       ifa->ifa_scope <= scope &&
+       ifa->ifa_net_ns == net_ns) {
+     addr = ifa->ifa_local;
+     goto out_unlock_both;
+   }
+ } endfor_ifa(in_dev);
+
+out_unlock_both:
+read_unlock(&dev_base_lock);
+rcu_read_unlock();
+out:
+return addr;
+}
#endif /* CONFIG_NET_NS */

```

Containers mailing list
 Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
