
Subject: [PATCH net-2.6.25 3/7] Pass the net pointer to the arp_req_set_proxy()
Posted by [Pavel Emelianov](#) on Tue, 11 Dec 2007 17:50:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

This one will need to set the IPV4_DEVCONF_ALL(PROXY_ARP), but there's no ways to get the net right in place, so we have to pull one from the inet_ioctl's struct sock.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

diff --git a/include/net/arp.h b/include/net/arp.h

index f026645..36482bf 100644

--- a/include/net/arp.h

+++ b/include/net/arp.h

@@ -11,7 +11,7 @@ extern struct neigh_table arp_tbl;

extern void arp_init(void);

extern int arp_find(unsigned char *haddr, struct sk_buff *skb);

-extern int arp_ioctl(unsigned int cmd, void __user *arg);

+extern int arp_ioctl(struct net *net, unsigned int cmd, void __user *arg);

extern void arp_send(int type, int ptype, __be32 dest_ip,

struct net_device *dev, __be32 src_ip,

unsigned char *dest_hw, unsigned char *src_hw, unsigned char *th);

diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c

index 7f8b27f..a69b0c6 100644

--- a/net/ipv4/af_inet.c

+++ b/net/ipv4/af_inet.c

@@ -798,7 +798,7 @@ int inet_ioctl(struct socket *sock, unsigned int cmd, unsigned long arg)

case SIOCDDARP:

case SIOCGARP:

case SIOCSARP:

- err = arp_ioctl(cmd, (void __user *)arg);

+ err = arp_ioctl(sk->sk_net, cmd, (void __user *)arg);

break;

case SIOCGIFADDR:

case SIOCSIFADDR:

diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c

index 48e7bf6..41a4d73 100644

--- a/net/ipv4/arp.c

+++ b/net/ipv4/arp.c

@@ -953,7 +953,7 @@ out_of_mem:

* Set (create) an ARP cache entry.

*/

-static int arp_req_set_proxy(struct net_device *dev, int on)

+static int arp_req_set_proxy(struct net *net, struct net_device *dev, int on)

```

{
    if (dev == NULL) {
        IPV4_DEVCONF_ALL(PROXY_ARP) = on;
@@ -966,7 +966,8 @@ static int arp_req_set_proxy(struct net_device *dev, int on)
        return -ENXIO;
    }

-static int arp_req_set_public(struct arpreq *r, struct net_device *dev)
+static int arp_req_set_public(struct net *net, struct arpreq *r,
+ struct net_device *dev)
{
    __be32 ip = ((struct sockaddr_in *)&r->arp_pa)->sin_addr.s_addr;
    __be32 mask = ((struct sockaddr_in *)&r->arp_netmask)->sin_addr.s_addr;
@@ -985,17 +986,18 @@ static int arp_req_set_public(struct arpreq *r, struct net_device *dev)
    return 0;
}

- return arp_req_set_proxy(dev, 1);
+ return arp_req_set_proxy(net, dev, 1);
}

-static int arp_req_set(struct arpreq *r, struct net_device * dev)
+static int arp_req_set(struct net *net, struct arpreq *r,
+ struct net_device * dev)
{
    __be32 ip;
    struct neighbour *neigh;
    int err;

    if (r->arp_flags & ATF_PUBL)
- return arp_req_set_public(r, dev);
+ return arp_req_set_public(net, r, dev);

    ip = ((struct sockaddr_in *)&r->arp_pa)->sin_addr.s_addr;
    if (r->arp_flags & ATF_PERM)
@@ -1081,7 +1083,8 @@ static int arp_req_get(struct arpreq *r, struct net_device *dev)
    return err;
}

-static int arp_req_delete_public(struct arpreq *r, struct net_device *dev)
+static int arp_req_delete_public(struct net *net, struct arpreq *r,
+ struct net_device *dev)
{
    __be32 ip = ((struct sockaddr_in *) &r->arp_pa)->sin_addr.s_addr;
    __be32 mask = ((struct sockaddr_in *)&r->arp_netmask)->sin_addr.s_addr;
@@ -1092,17 +1095,18 @@ static int arp_req_delete_public(struct arpreq *r, struct net_device
*dev)
    if (mask)

```

```

return -EINVAL;

- return arp_req_set_proxy(dev, 0);
+ return arp_req_set_proxy(net, dev, 0);
}

-static int arp_req_delete(struct arpreq *r, struct net_device * dev)
+static int arp_req_delete(struct net *net, struct arpreq *r,
+ struct net_device * dev)
{
    int err;
    __be32 ip;
    struct neighbour *neigh;

    if (r->arp_flags & ATF_PUBL)
- return arp_req_delete_public(r, dev);
+ return arp_req_delete_public(net, r, dev);

    ip = ((struct sockaddr_in *)&r->arp_pa)->sin_addr.s_addr;
    if (dev == NULL) {
@@ -1132,7 +1136,7 @@ static int arp_req_delete(struct arpreq *r, struct net_device * dev)
    * Handle an ARP layer I/O control request.
    */

-int arp_ioctl(unsigned int cmd, void __user *arg)
+int arp_ioctl(struct net *net, unsigned int cmd, void __user *arg)
{
    int err;
    struct arpreq r;
@@ -1180,10 +1184,10 @@ int arp_ioctl(unsigned int cmd, void __user *arg)

    switch (cmd) {
    case SIOCDARP:
- err = arp_req_delete(&r, dev);
+ err = arp_req_delete(net, &r, dev);
        break;
    case SIOCSARP:
- err = arp_req_set(&r, dev);
+ err = arp_req_set(net, &r, dev);
        break;
    case SIOCGARP:
        err = arp_req_get(&r, dev);
--

```

1.5.3.4
