Subject: [PATCH net-2.6.25 7/7] Switch users of ipv4_devconf(_all) to use the pernet one
Posted by Pavel Emelianov on Tue, 11 Dec 2007 18:02:18 GMT
View Forum Message <> Reply to Message

These are scattered over the code, but almost all the
"critical" places already have the proper struct net
at hand except for snmp proc showing function and routing
rtnl handler.

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

```
diff --git a/include/linux/inetdevice.h b/include/linux/inetdevice.h
index 962a062..b3c5081 100644
--- a/include/linux/inetdevice.h
+++ b/include/linux/inetdevice.h
@@ -44,7 +44,8 @@ struct in_device
 };

 #define IPV4_DEVCONF(cnf, attr) ((cnf).data[NET_IPV4_CONF_ ## attr - 1])
-#define IPV4_DEVCONF_ALL(attr) IPV4_DEVCONF(ipv4_devconf, attr)
+#define IPV4_DEVCONF_ALL(net, attr) \
+ IPV4_DEVCONF((*(net)->ipv4.devconf_all), attr)

 static inline int ipv4_devconf_get(struct in_device *in_dev, int index)
 {
@@ -71,11 +72,14 @@ static inline void ipv4_devconf_setall(struct in_device *in_dev)
  ipv4_devconf_set((in_dev), NET_IPV4_CONF_ ## attr, (val))

 #define IN_DEV_ANDCONF(in_dev, attr) \
- (IPV4_DEVCONF_ALL(attr) && IN_DEV_CONF_GET((in_dev), attr))
+ (IPV4_DEVCONF_ALL(in_dev->dev->nd_net, attr) && \
+  IN_DEV_CONF_GET((in_dev), attr))
 #define IN_DEV_ORCONF(in_dev, attr) \
- (IPV4_DEVCONF_ALL(attr) || IN_DEV_CONF_GET((in_dev), attr))
+ (IPV4_DEVCONF_ALL(in_dev->dev->nd_net, attr) || \
+  IN_DEV_CONF_GET((in_dev), attr))
 #define IN_DEV_MAXCONF(in_dev, attr) \
- (max(IPV4_DEVCONF_ALL(attr), IN_DEV_CONF_GET((in_dev), attr)))
+ (max(IPV4_DEVCONF_ALL(in_dev->dev->nd_net, attr), \
+       IN_DEV_CONF_GET((in_dev), attr)))

 #define IN_DEV_FORWARD(in_dev)  IN_DEV_CONF_GET((in_dev), FORWARDING)
 #define IN_DEV_MFORWARD(in_dev)  IN_DEV_ANDCONF((in_dev), MC_FORWARDING)
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 41a4d73..cd4c95a 100644
```

```
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -861,7 +861,7 @@ static int arp_process(struct sk_buff *skb)

  n = __neigh_lookup(&arp_tbl, &sip, dev, 0);

- if (IPV4_DEVCONF_ALL(ARP_ACCEPT)) {
+ if (IPV4_DEVCONF_ALL(dev->nd_net, ARP_ACCEPT)) {
    /* Unsolicited ARP is not accepted by default.
       It is possible, that this option should be enabled for some
       devices (strip is candidate)
@@ -956,7 +956,7 @@ out_of_mem:
 static int arp_req_set_proxy(struct net *net, struct net_device *dev, int on)
 {
  if (dev == NULL) {
-  IPV4_DEVCONF_ALL(PROXY_ARP) = on;
+  IPV4_DEVCONF_ALL(net, PROXY_ARP) = on;
    return 0;
  }
  if (__in_dev_get_rtnl(dev)) {
diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index cd957f4..e94de54 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -1259,9 +1259,9 @@ static void devinet_copy_dflt_conf(struct net *net, int i)
 static void inet_forward_change(struct net *net)
 {
  struct net_device *dev;
- int on = IPV4_DEVCONF_ALL(FORWARDING);
+ int on = IPV4_DEVCONF_ALL(net, FORWARDING);

- IPV4_DEVCONF_ALL(ACCEPT_REDIRECTS) = !on;
+ IPV4_DEVCONF_ALL(net, ACCEPT_REDIRECTS) = !on;
  IPV4_DEVCONF_DFLT(net, FORWARDING) = on;

  read_lock(&dev_base_lock);
@@ -1361,7 +1361,7 @@ static int devinet_sysctl_forward(ctl_table *ctl, int write,
  if (write && *valp != val) {
   struct net *net = ctl->extra2;

-  if (valp == &IPV4_DEVCONF_ALL(FORWARDING))
+  if (valp == &IPV4_DEVCONF_ALL(net, FORWARDING))
    inet_forward_change(net);
   else if (valp != &IPV4_DEVCONF_DFLT(net, FORWARDING))
    rt_cache_flush(0);
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index c560a93..d3d5906 100644
--- a/net/ipv4/igmp.c
```

```
+++ b/net/ipv4/igmp.c
@@ -130,12 +130,12 @@
  */

 #define IGMP_V1_SEEN(in_dev) \
- (IPV4_DEVCONF_ALL(FORCE_IGMP_VERSION) == 1 || \
+ (IPV4_DEVCONF_ALL(in_dev->dev->nd_net, FORCE_IGMP_VERSION) == 1 || \
   IN_DEV_CONF_GET((in_dev), FORCE_IGMP_VERSION) == 1 || \
   ((in_dev)->mr_v1_seen && \
    time_before(jiffies, (in_dev)->mr_v1_seen)))
 #define IGMP_V2_SEEN(in_dev) \
- (IPV4_DEVCONF_ALL(FORCE_IGMP_VERSION) == 2 || \
+ (IPV4_DEVCONF_ALL(in_dev->dev->nd_net, FORCE_IGMP_VERSION) == 2 || \
   IN_DEV_CONF_GET((in_dev), FORCE_IGMP_VERSION) == 2 || \
   ((in_dev)->mr_v2_seen && \
    time_before(jiffies, (in_dev)->mr_v2_seen)))
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 1187928..9947f52 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -849,7 +849,7 @@ static void mrtsock_destruct(struct sock *sk)
 {
  rtnl_lock();
  if (sk == mroute_socket) {
-  IPV4_DEVCONF_ALL(MC_FORWARDING)--;
+  IPV4_DEVCONF_ALL(sk->sk_net, MC_FORWARDING)--;

   write_lock_bh(&mrt_lock);
   mroute_socket=NULL;
@@ -898,7 +898,7 @@ int ip_mroute_setsockopt(struct sock *sk,int optname,char __user
*optval,int opt
    mroute_socket=sk;
    write_unlock_bh(&mrt_lock);

-   IPV4_DEVCONF_ALL(MC_FORWARDING)++;
+   IPV4_DEVCONF_ALL(sk->sk_net, MC_FORWARDING)++;
   }
   rtnl_unlock();
   return ret;
diff --git a/net/ipv4/proc.c b/net/ipv4/proc.c
index ce34b28..41734db 100644
--- a/net/ipv4/proc.c
+++ b/net/ipv4/proc.c
@@ -309,7 +309,8 @@ static int snmp_seq_show(struct seq_file *seq, void *v)
   seq_printf(seq, " %s", snmp4_ipstats_list[i].name);

  seq_printf(seq, "\nIp: %d %d",
-     IPV4_DEVCONF_ALL(FORWARDING) ? 1 : 2, sysctl_ip_default_ttl);
```

```
+     IPV4_DEVCONF_ALL(&init_net, FORWARDING) ? 1 : 2,
+     sysctl_ip_default_ttl);

  for (i = 0; snmp4_ipstats_list[i].name != NULL; i++)
   seq_printf(seq, " %lu",
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index cc064e6..499e1e3 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2615,7 +2615,7 @@ static int rt_fill_info(struct sk_buff *skb, u32 pid, u32 seq, int event,
   __be32 dst = rt->rt_dst;

   if (MULTICAST(dst) && !LOCAL_MCAST(dst) &&
-      IPV4_DEVCONF_ALL(MC_FORWARDING)) {
+      IPV4_DEVCONF_ALL(&init_net, MC_FORWARDING)) {
    int err = ipmr_get_route(skb, r, nowait);
    if (err <= 0) {
     if (!nowait) {
--
1.5.3.4
```