Subject: [PATCH 04/10] sysctl: Fix neighbour table sysctls.
Posted by ebiederm on Fri, 10 Aug 2007 00:56:09 GMT
View Forum Message <> Reply to Message

- In ipv6 ndisc_ifinfo_syctl_change so it doesn't depend on binary
  sysctl names for a function that works with proc.

- In neighbour.c reorder the table to put the possibly unused entries
  at the end so we can remove them by terminating the table early.

- In neighbour.c kill the entries with questionable binary sysctl
  handling behavior.

- In neighbour.c if we don't have a strategy routine remove the
  binary path. So we don't the default sysctl strategy routine
  on data that is not ready for it.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 net/core/neighbour.c | 75 ++++++++++++++++++++++++++++----------------------
 net/ipv6/ndisc.c | 24 ++++++---------
 2 files changed, 49 insertions(+), 50 deletions(-)

diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index ca2a153..27c3f4e 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -2498,7 +2498,6 @@ static struct neigh_sysctl_table {
     .proc_handler = &proc_dointvec,
    },
    {
-    .ctl_name = NET_NEIGH_RETRANS_TIME,
     .procname = "retrans_time",
     .maxlen = sizeof(int),
     .mode = 0644,
@@ -2543,27 +2542,40 @@ static struct neigh_sysctl_table {
     .proc_handler = &proc_dointvec,
    },
    {
-    .ctl_name = NET_NEIGH_ANYCAST_DELAY,
     .procname = "anycast_delay",
     .maxlen = sizeof(int),
     .mode = 0644,
     .proc_handler = &proc_dointvec_userhz_jiffies,
    },
    {
-    .ctl_name = NET_NEIGH_PROXY_DELAY,
     .procname = "proxy_delay",

```
    .maxlen  = sizeof(int),
    .mode  = 0644,
    .proc_handler = &proc_dointvec_userhz_jiffies,
   },
   {
-   .ctl_name = NET_NEIGH_LOCKTIME,
    .procname = "locktime",
    .maxlen  = sizeof(int),
    .mode  = 0644,
    .proc_handler = &proc_dointvec_userhz_jiffies,
   },
   {
+   .ctl_name = NET_NEIGH_RETRANS_TIME_MS,
+   .procname = "retrans_time_ms",
+   .maxlen  = sizeof(int),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec_ms_jiffies,
+   .strategy = &sysctl_ms_jiffies,
+  },
+  {
+   .ctl_name = NET_NEIGH_REACHABLE_TIME_MS,
+   .procname = "base_reachable_time_ms",
+   .maxlen  = sizeof(int),
+   .mode  = 0644,
+   .proc_handler = &proc_dointvec_ms_jiffies,
+   .strategy = &sysctl_ms_jiffies,
+  },
+  {
    .ctl_name = NET_NEIGH_GC_INTERVAL,
    .procname = "gc_interval",
    .maxlen  = sizeof(int),
@@ -2592,22 +2604,7 @@ static struct neigh_sysctl_table {
    .mode  = 0644,
    .proc_handler = &proc_dointvec,
   },
-  {
-   .ctl_name = NET_NEIGH_RETRANS_TIME_MS,
-   .procname = "retrans_time_ms",
-   .maxlen  = sizeof(int),
-   .mode  = 0644,
-   .proc_handler = &proc_dointvec_ms_jiffies,
-   .strategy = &sysctl_ms_jiffies,
-  },
-  {
-   .ctl_name = NET_NEIGH_REACHABLE_TIME_MS,
-   .procname = "base_reachable_time_ms",
-   .maxlen  = sizeof(int),
-   .mode  = 0644,
```

```
-   .proc_handler = &proc_dointvec_ms_jiffies,
-   .strategy = &sysctl_ms_jiffies,
-  },
+  {}
  },
  .neigh_dev = {
   {
@@ -2660,42 +2657,48 @@ int neigh_sysctl_register(struct net_device *dev, struct neigh_parms
*p,
  t->neigh_vars[9].data  = &p->anycast_delay;
  t->neigh_vars[10].data = &p->proxy_delay;
  t->neigh_vars[11].data = &p->locktime;
+ t->neigh_vars[12].data  = &p->retrans_time;
+ t->neigh_vars[13].data  = &p->base_reachable_time;

  if (dev) {
   dev_name_source = dev->name;
   t->neigh_dev[0].ctl_name = dev->ifindex;
-  t->neigh_vars[12].procname = NULL;
-  t->neigh_vars[13].procname = NULL;
-  t->neigh_vars[14].procname = NULL;
-  t->neigh_vars[15].procname = NULL;
+  /* Terminate the table early */
+  memset(&t->neigh_vars[14], 0, sizeof(t->neigh_vars[14]));
  } else {
   dev_name_source = t->neigh_dev[0].procname;
-  t->neigh_vars[12].data = (int *)(p + 1);
-  t->neigh_vars[13].data = (int *)(p + 1) + 1;
-  t->neigh_vars[14].data = (int *)(p + 1) + 2;
-  t->neigh_vars[15].data = (int *)(p + 1) + 3;
+  t->neigh_vars[14].data = (int *)(p + 1);
+  t->neigh_vars[15].data = (int *)(p + 1) + 1;
+  t->neigh_vars[16].data = (int *)(p + 1) + 2;
+  t->neigh_vars[17].data = (int *)(p + 1) + 3;
  }

- t->neigh_vars[16].data  = &p->retrans_time;
- t->neigh_vars[17].data  = &p->base_reachable_time;

  if (handler || strategy) {
   /* RetransTime */
   t->neigh_vars[3].proc_handler = handler;
   t->neigh_vars[3].strategy = strategy;
   t->neigh_vars[3].extra1 = dev;
+  if (!strategy)
+   t->neigh_vars[3].ctl_name = CTL_UNNUMBERED;
   /* ReachableTime */
   t->neigh_vars[4].proc_handler = handler;
```

```
   t->neigh_vars[4].strategy = strategy;
   t->neigh_vars[4].extra1 = dev;
+  if (!strategy)
+   t->neigh_vars[4].ctl_name = CTL_UNNUMBERED;
   /* RetransTime (in milliseconds)*/
-  t->neigh_vars[16].proc_handler = handler;
-  t->neigh_vars[16].strategy = strategy;
-  t->neigh_vars[16].extra1 = dev;
+  t->neigh_vars[12].proc_handler = handler;
+  t->neigh_vars[12].strategy = strategy;
+  t->neigh_vars[12].extra1 = dev;
+  if (!strategy)
+   t->neigh_vars[12].ctl_name = CTL_UNNUMBERED;
   /* ReachableTime (in milliseconds) */
-  t->neigh_vars[17].proc_handler = handler;
-  t->neigh_vars[17].strategy = strategy;
-  t->neigh_vars[17].extra1 = dev;
+  t->neigh_vars[13].proc_handler = handler;
+  t->neigh_vars[13].strategy = strategy;
+  t->neigh_vars[13].extra1 = dev;
+  if (!strategy)
+   t->neigh_vars[13].ctl_name = CTL_UNNUMBERED;
  }

  dev_name = kstrdup(dev_name_source, GFP_KERNEL);
diff --git a/net/ipv6/ndisc.c b/net/ipv6/ndisc.c
index 0358e60..d388429 100644
--- a/net/ipv6/ndisc.c
+++ b/net/ipv6/ndisc.c
@@ -1570,30 +1570,26 @@ int ndisc_ifinfo_sysctl_change(struct ctl_table *ctl, int write, struct
file * f
  struct inet6_dev *idev;
  int ret;

- if (ctl->ctl_name == NET_NEIGH_RETRANS_TIME ||
-     ctl->ctl_name == NET_NEIGH_REACHABLE_TIME)
+ if ((strcmp(ctl->procname, "retrans_time") == 0) ||
+     (strcmp(ctl->procname, "base_reachable_time") == 0))
   ndisc_warn_deprecated_sysctl(ctl, "syscall", dev ? dev->name : "default");

- switch (ctl->ctl_name) {
- case NET_NEIGH_RETRANS_TIME:
+ if (strcmp(ctl->procname, "retrans_time") == 0)
   ret = proc_dointvec(ctl, write, filp, buffer, lenp, ppos);
-  break;
- case NET_NEIGH_REACHABLE_TIME:
+
+ else if (strcmp(ctl->procname, "base_reachable_time") == 0)
```

```
   ret = proc_dointvec_jiffies(ctl, write,
         filp, buffer, lenp, ppos);
-  break;
- case NET_NEIGH_RETRANS_TIME_MS:
- case NET_NEIGH_REACHABLE_TIME_MS:
+
+ else if ((strcmp(ctl->procname, "retrans_time_ms") == 0) ||
+    (strcmp(ctl->procname, "base_reacable_time_ms") == 0))
   ret = proc_dointvec_ms_jiffies(ctl, write,
           filp, buffer, lenp, ppos);
-  break;
- default:
+ else
   ret = -1;
- }

  if (write && ret == 0 && dev && (idev = in6_dev_get(dev)) != NULL) {
-  if (ctl->ctl_name == NET_NEIGH_REACHABLE_TIME ||
-      ctl->ctl_name == NET_NEIGH_REACHABLE_TIME_MS)
+  if (ctl->data == &idev->nd_parms->base_reachable_time)
    idev->nd_parms->reachable_time =
neigh_rand_reach_time(idev->nd_parms->base_reachable_time);
   idev->tstamp = jiffies;
   inet6_ifinfo_notify(RTM_NEWLINK, idev);
--
1.5.1.1.181.g2de0
```