Subject: [PATCH net-2.6.25 1/10][NETNS][FRAGS]: Move ctl tables around.
Posted by Pavel Emelianov on Tue, 22 Jan 2008 13:55:23 GMT
View Forum Message <> Reply to Message

This is a preparation for sysctl netns-ization.
Move the ctl tables to the files, where the tuning
variables reside. Plus make the helpers to register
the tables.

This will simplify the later patches and will keep
similar things closer to each other.

ipv4, ipv6 and conntrack_reasm are patched differently,
bu the result is all the tables are in appropriate files.

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

---
 include/net/ip.h | 5 --
 include/net/ipv6.h | 1 -
 include/net/netfilter/ipv6/nf_conntrack_ipv6.h | 4 +-
 net/ipv4/ip_fragment.c | 74 ++++++++++++++++++++++++-
 net/ipv4/sysctl_net_ipv4.c | 42 ----------
 net/ipv6/af_inet6.c | 5 --
 net/ipv6/netfilter/nf_conntrack_l3proto_ipv6.c | 29 ---------
 net/ipv6/netfilter/nf_conntrack_reasm.c | 31 ++++++++-
 net/ipv6/reassembly.c | 66 ++++++++++++++++++++-
 net/ipv6/sysctl_net_ipv6.c | 40 +----------
 10 files changed, 169 insertions(+), 128 deletions(-)

diff --git a/include/net/ip.h b/include/net/ip.h
index 2ad4d2f..ff14fc8 100644
--- a/include/net/ip.h
+++ b/include/net/ip.h
@@ -179,11 +179,6 @@ extern int sysctl_ip_nonlocal_bind;

 extern struct ctl_path net_ipv4_ctl_path[];

-/* From ip_fragment.c */
-struct inet_frags_ctl;
-extern struct inet_frags_ctl ip4_frags_ctl;
-extern int sysctl_ipfrag_max_dist;
-
 /* From inetpeer.c */
 extern int inet_peer_threshold;
 extern int inet_peer_minttl;
diff --git a/include/net/ipv6.h b/include/net/ipv6.h
index 3712cae..87ca1bf 100644

```
--- a/include/net/ipv6.h
+++ b/include/net/ipv6.h
@@ -587,7 +587,6 @@ extern int ip6_mc_msfget(struct sock *sk, struct group_filter *gsf,

 #ifdef CONFIG_PROC_FS
 extern struct ctl_table *ipv6_icmp_sysctl_init(struct net *net);
-extern void ipv6_frag_sysctl_init(struct net *net);
 extern struct ctl_table *ipv6_route_sysctl_init(struct net *net);

 extern int  ac6_proc_init(void);
diff --git a/include/net/netfilter/ipv6/nf_conntrack_ipv6.h
b/include/net/netfilter/ipv6/nf_conntrack_ipv6.h
index f703533..abc55ad 100644
--- a/include/net/netfilter/ipv6/nf_conntrack_ipv6.h
+++ b/include/net/netfilter/ipv6/nf_conntrack_ipv6.h
@@ -16,6 +16,8 @@ extern void nf_ct_frag6_output(unsigned int hooknum, struct sk_buff *skb,
          int (*okfn)(struct sk_buff *));

 struct inet_frags_ctl;
-extern struct inet_frags_ctl nf_frags_ctl;
+
+#include <linux/sysctl.h>
+extern struct ctl_table nf_ct_ipv6_sysctl_table[];

 #endif /* _NF_CONNTRACK_IPV6_H*/
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 2143bf3..a53463e 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -50,7 +50,7 @@
  * as well. Or notify me, at least. --ANK
  */

-int sysctl_ipfrag_max_dist __read_mostly = 64;
+static int sysctl_ipfrag_max_dist __read_mostly = 64;

 struct ipfrag_skb_cb
 {
@@ -74,7 +74,7 @@ struct ipq {
  struct inet_peer *peer;
 };

-struct inet_frags_ctl ip4_frags_ctl __read_mostly = {
+static struct inet_frags_ctl ip4_frags_ctl __read_mostly = {
  /*
   * Fragment cache limits. We will commit 256K at one time. Should we
   * cross that limit we will prune down to 192K. This should cope with
@@ -607,8 +607,78 @@ int ip_defrag(struct sk_buff *skb, u32 user)
```

```
  return -ENOMEM;
 }

+#ifdef CONFIG_SYSCTL
+static int zero;
+
+static struct ctl_table ip4_frags_ctl_table[] = {
+ {
+  .ctl_name = NET_IPV4_IPFRAG_HIGH_THRESH,
+  .procname = "ipfrag_high_thresh",
+  .data  = &ip4_frags_ctl.high_thresh,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_IPFRAG_LOW_THRESH,
+  .procname = "ipfrag_low_thresh",
+  .data  = &ip4_frags_ctl.low_thresh,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV4_IPFRAG_TIME,
+  .procname = "ipfrag_time",
+  .data  = &ip4_frags_ctl.timeout,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_jiffies,
+  .strategy = &sysctl_jiffies
+ },
+ {
+  .ctl_name = NET_IPV4_IPFRAG_SECRET_INTERVAL,
+  .procname = "ipfrag_secret_interval",
+  .data  = &ip4_frags_ctl.secret_interval,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_jiffies,
+  .strategy = &sysctl_jiffies
+ },
+ {
+  .procname = "ipfrag_max_dist",
+  .data  = &sysctl_ipfrag_max_dist,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_minmax,
+  .extra1  = &zero
```

```
+ },
+ { }
+};
+
+static int ip4_frags_ctl_register(struct net *net)
+{
+ struct ctl_table_header *hdr;
+
+ hdr = register_net_sysctl_table(net, net_ipv4_ctl_path,
+   ip4_frags_ctl_table);
+ return hdr == NULL ? -ENOMEM : 0;
+}
+#else
+static inline int ip4_frags_ctl_register(struct net *net)
+{
+ return 0;
+}
+#endif
+
+static int ipv4_frags_init_net(struct net *net)
+{
+ return ip4_frags_ctl_register(net);
+}
+
 void __init ipfrag_init(void)
 {
+ ipv4_frags_init_net(&init_net);
  ip4_frags.ctl = &ip4_frags_ctl;
  ip4_frags.hashfn = ip4_hashfn;
  ip4_frags.constructor = ip4_frag_init;
diff --git a/net/ipv4/sysctl_net_ipv4.c b/net/ipv4/sysctl_net_ipv4.c
index 45536a9..82cdf23 100644
--- a/net/ipv4/sysctl_net_ipv4.c
+++ b/net/ipv4/sysctl_net_ipv4.c
@@ -284,22 +284,6 @@ static struct ctl_table ipv4_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = NET_IPV4_IPFRAG_HIGH_THRESH,
-  .procname = "ipfrag_high_thresh",
-  .data  = &ip4_frags_ctl.high_thresh,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV4_IPFRAG_LOW_THRESH,
-  .procname = "ipfrag_low_thresh",
```

```
-  .data  = &ip4_frags_ctl.low_thresh,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
   .ctl_name = NET_IPV4_DYNADDR,
   .procname = "ip_dynaddr",
   .data  = &sysctl_ip_dynaddr,
@@ -308,15 +292,6 @@ static struct ctl_table ipv4_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = NET_IPV4_IPFRAG_TIME,
-  .procname = "ipfrag_time",
-  .data  = &ip4_frags_ctl.timeout,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_jiffies,
-  .strategy = &sysctl_jiffies
- },
- {
   .ctl_name = NET_IPV4_TCP_KEEPALIVE_TIME,
   .procname = "tcp_keepalive_time",
   .data  = &sysctl_tcp_keepalive_time,
@@ -659,23 +634,6 @@ static struct ctl_table ipv4_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = NET_IPV4_IPFRAG_SECRET_INTERVAL,
-  .procname = "ipfrag_secret_interval",
-  .data  = &ip4_frags_ctl.secret_interval,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_jiffies,
-  .strategy = &sysctl_jiffies
- },
- {
-  .procname = "ipfrag_max_dist",
-  .data  = &sysctl_ipfrag_max_dist,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_minmax,
-  .extra1  = &zero
- },
- {
   .ctl_name = NET_TCP_NO_METRICS_SAVE,
   .procname = "tcp_no_metrics_save",
```

```
   .data  = &sysctl_tcp_nometrics_save,
diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index 6738a7b..bddac0e 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -721,10 +721,6 @@ static void cleanup_ipv6_mibs(void)
 static int inet6_net_init(struct net *net)
 {
  net->ipv6.sysctl.bindv6only = 0;
- net->ipv6.sysctl.frags.high_thresh = 256 * 1024;
- net->ipv6.sysctl.frags.low_thresh = 192 * 1024;
- net->ipv6.sysctl.frags.timeout = IPV6_FRAG_TIMEOUT;
- net->ipv6.sysctl.frags.secret_interval = 10 * 60 * HZ;
  net->ipv6.sysctl.flush_delay = 0;
  net->ipv6.sysctl.ip6_rt_max_size = 4096;
  net->ipv6.sysctl.ip6_rt_gc_min_interval = HZ / 2;
@@ -734,7 +730,6 @@ static int inet6_net_init(struct net *net)
  net->ipv6.sysctl.ip6_rt_mtu_expires = 10*60*HZ;
  net->ipv6.sysctl.ip6_rt_min_advmss = IPV6_MIN_MTU - 20 - 40;
  net->ipv6.sysctl.icmpv6_time = 1*HZ;
- ipv6_frag_sysctl_init(net);

  return 0;
 }
diff --git a/net/ipv6/netfilter/nf_conntrack_l3proto_ipv6.c
b/net/ipv6/netfilter/nf_conntrack_l3proto_ipv6.c
index cf42f5c..2d7b024 100644
--- a/net/ipv6/netfilter/nf_conntrack_l3proto_ipv6.c
+++ b/net/ipv6/netfilter/nf_conntrack_l3proto_ipv6.c
@@ -297,35 +297,6 @@ static struct nf_hook_ops ipv6_conntrack_ops[] __read_mostly = {
  },
 };

-#ifdef CONFIG_SYSCTL
-static ctl_table nf_ct_ipv6_sysctl_table[] = {
- {
-  .procname = "nf_conntrack_frag6_timeout",
-  .data  = &nf_frags_ctl.timeout,
-  .maxlen  = sizeof(unsigned int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_jiffies,
- },
- {
-  .ctl_name = NET_NF_CONNTRACK_FRAG6_LOW_THRESH,
-  .procname = "nf_conntrack_frag6_low_thresh",
-  .data  = &nf_frags_ctl.low_thresh,
-  .maxlen  = sizeof(unsigned int),
-  .mode  = 0644,
```

```
-  .proc_handler = &proc_dointvec,
- },
- {
-  .ctl_name = NET_NF_CONNTRACK_FRAG6_HIGH_THRESH,
-  .procname = "nf_conntrack_frag6_high_thresh",
-  .data  = &nf_frags_ctl.high_thresh,
-  .maxlen  = sizeof(unsigned int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec,
- },
- { .ctl_name = 0 }
-};
-#endif
-
 #if defined(CONFIG_NF_CT_NETLINK) || defined(CONFIG_NF_CT_NETLINK_MODULE)

 #include <linux/netfilter/nfnetlink.h>
diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index e170c67..d631631 100644
--- a/net/ipv6/netfilter/nf_conntrack_reasm.c
+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c
@@ -70,7 +70,7 @@ struct nf_ct_frag6_queue
  __u16   nhoffset;
 };

-struct inet_frags_ctl nf_frags_ctl __read_mostly = {
+static struct inet_frags_ctl nf_frags_ctl __read_mostly = {
  .high_thresh  = 256 * 1024,
  .low_thresh  = 192 * 1024,
  .timeout  = IPV6_FRAG_TIMEOUT,
@@ -79,6 +79,35 @@ struct inet_frags_ctl nf_frags_ctl __read_mostly = {

 static struct inet_frags nf_frags;

+#ifdef CONFIG_SYSCTL
+struct ctl_table nf_ct_ipv6_sysctl_table[] = {
+ {
+  .procname = "nf_conntrack_frag6_timeout",
+  .data  = &nf_frags_ctl.timeout,
+  .maxlen  = sizeof(unsigned int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_jiffies,
+ },
+ {
+  .ctl_name = NET_NF_CONNTRACK_FRAG6_LOW_THRESH,
+  .procname = "nf_conntrack_frag6_low_thresh",
+  .data  = &nf_frags_ctl.low_thresh,
+  .maxlen  = sizeof(unsigned int),
```

```
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec,
+ },
+ {
+  .ctl_name = NET_NF_CONNTRACK_FRAG6_HIGH_THRESH,
+  .procname = "nf_conntrack_frag6_high_thresh",
+  .data  = &nf_frags_ctl.high_thresh,
+  .maxlen  = sizeof(unsigned int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec,
+ },
+ { .ctl_name = 0 }
+};
+#endif
+
 static unsigned int ip6qhashfn(__be32 id, struct in6_addr *saddr,
          struct in6_addr *daddr)
 {
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 4dfcddc..1815ff0 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -625,12 +625,70 @@ static struct inet6_protocol frag_protocol =
  .flags  = INET6_PROTO_NOPOLICY,
 };

-void ipv6_frag_sysctl_init(struct net *net)
+#ifdef CONFIG_SYSCTL
+static struct ctl_table ip6_frags_ctl_table[] = {
+ {
+  .ctl_name = NET_IPV6_IP6FRAG_HIGH_THRESH,
+  .procname = "ip6frag_high_thresh",
+  .data  = &init_net.ipv6.sysctl.frags.high_thresh,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV6_IP6FRAG_LOW_THRESH,
+  .procname = "ip6frag_low_thresh",
+  .data  = &init_net.ipv6.sysctl.frags.low_thresh,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec
+ },
+ {
+  .ctl_name = NET_IPV6_IP6FRAG_TIME,
+  .procname = "ip6frag_time",
```

```
+  .data  = &init_net.ipv6.sysctl.frags.timeout,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_jiffies,
+  .strategy = &sysctl_jiffies,
+ },
+ {
+  .ctl_name = NET_IPV6_IP6FRAG_SECRET_INTERVAL,
+  .procname = "ip6frag_secret_interval",
+  .data  = &init_net.ipv6.sysctl.frags.secret_interval,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec_jiffies,
+  .strategy = &sysctl_jiffies
+ },
+ { }
+};
+
+static int ip6_frags_sysctl_register(struct net *net)
+{
+ struct ctl_table_header *hdr;
+
+ hdr = register_net_sysctl_table(net, net_ipv6_ctl_path,
+   ip6_frags_ctl_table);
+ return hdr == NULL ? -ENOMEM : 0;
+}
+#else
+static inline int ip6_frags_sysctl_register(struct net *net)
 {
- if (net != &init_net)
-  return;
+ return 0;
+}
+#endif

+static int ipv6_frags_init_net(struct net *net)
+{
  ip6_frags.ctl = &net->ipv6.sysctl.frags;
+
+ net->ipv6.sysctl.frags.high_thresh = 256 * 1024;
+ net->ipv6.sysctl.frags.low_thresh = 192 * 1024;
+ net->ipv6.sysctl.frags.timeout = IPV6_FRAG_TIMEOUT;
+ net->ipv6.sysctl.frags.secret_interval = 10 * 60 * HZ;
+
+ return ip6_frags_sysctl_register(net);
 }

 int __init ipv6_frag_init(void)
```

```
@@ -641,6 +699,8 @@ int __init ipv6_frag_init(void)
  if (ret)
   goto out;

+ ipv6_frags_init_net(&init_net);
+
  ip6_frags.hashfn = ip6_hashfn;
  ip6_frags.constructor = ip6_frag_init;
  ip6_frags.destructor = NULL;
diff --git a/net/ipv6/sysctl_net_ipv6.c b/net/ipv6/sysctl_net_ipv6.c
index 7197eb7..408691b 100644
--- a/net/ipv6/sysctl_net_ipv6.c
+++ b/net/ipv6/sysctl_net_ipv6.c
@@ -38,40 +38,6 @@ static ctl_table ipv6_table_template[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = NET_IPV6_IP6FRAG_HIGH_THRESH,
-  .procname = "ip6frag_high_thresh",
-  .data  = &init_net.ipv6.sysctl.frags.high_thresh,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV6_IP6FRAG_LOW_THRESH,
-  .procname = "ip6frag_low_thresh",
-  .data  = &init_net.ipv6.sysctl.frags.low_thresh,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec
- },
- {
-  .ctl_name = NET_IPV6_IP6FRAG_TIME,
-  .procname = "ip6frag_time",
-  .data  = &init_net.ipv6.sysctl.frags.timeout,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_jiffies,
-  .strategy = &sysctl_jiffies,
- },
- {
-  .ctl_name = NET_IPV6_IP6FRAG_SECRET_INTERVAL,
-  .procname = "ip6frag_secret_interval",
-  .data  = &init_net.ipv6.sysctl.frags.secret_interval,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = &proc_dointvec_jiffies,
```

```
-  .strategy = &sysctl_jiffies
- },
- {
   .ctl_name = NET_IPV6_MLD_MAX_MSF,
   .procname = "mld_max_msf",
   .data  = &sysctl_mld_max_msf,
@@ -126,16 +92,12 @@ static int ipv6_sysctl_net_init(struct net *net)
  ipv6_table[1].child = ipv6_icmp_table;

  ipv6_table[2].data = &net->ipv6.sysctl.bindv6only;
- ipv6_table[3].data = &net->ipv6.sysctl.frags.high_thresh;
- ipv6_table[4].data = &net->ipv6.sysctl.frags.low_thresh;
- ipv6_table[5].data = &net->ipv6.sysctl.frags.timeout;
- ipv6_table[6].data = &net->ipv6.sysctl.frags.secret_interval;

  /* We don't want this value to be per namespace, it should be global
     to all namespaces, so make it read-only when we are not in the
     init network namespace */
  if (net != &init_net)
-  ipv6_table[7].mode = 0444;
+  ipv6_table[3].mode = 0444;

  net->ipv6.sysctl.table = register_net_sysctl_table(net, net_ipv6_ctl_path,
          ipv6_table);
--
1.5.3.4
```