Subject: [PATCH 1/12] L2 network namespace (v3): current network namespace operations
Posted by Mishin Dmitry on Wed, 17 Jan 2007 15:57:40 GMT
View Forum Message <> Reply to Message

Added functions and macros required to operate with network namespaces.
They are required in order to switch network namespace for incoming packets and
to not extend current network interface by additional network namespace argue.

Signed-off-by: Dmitry Mishin <dim@openvz.org>

---
 include/linux/net_namespace.h | 49 ++--
 kernel/fork.c | 1
 kernel/sched.c | 5 ++++
 net/core/net_namespace.c | 3 ++
 4 files changed, 56 insertions(+), 2 deletions(-)

--- linux-2.6.20-rc4-mm1.net_ns.orig/include/linux/net_namespace.h
+++ linux-2.6.20-rc4-mm1.net_ns/include/linux/net_namespace.h
@@ -33,7 +33,34 @@ static inline void put_net_ns(struct net
  kref_put(&ns->kref, free_net_ns);
 }

-#else
+DECLARE_PER_CPU(struct net_namespace *, exec_net_ns);
+#define current_net_ns (__get_cpu_var(exec_net_ns))
+
+static inline void init_current_net_ns(int cpu)
+{
+ get_net_ns(&init_net_ns);
+ per_cpu(exec_net_ns, cpu) = &init_net_ns;
+}
+
+static inline struct net_namespace *push_net_ns(struct net_namespace *to)
+{
+ struct net_namespace *orig;
+
+ orig = current_net_ns;
+ get_net_ns(to);
+ current_net_ns = to;
+ put_net_ns(orig);
+ return orig;
+}
+
+static inline void pop_net_ns(struct net_namespace *to)
+{
+ (void)push_net_ns(to);

```
+}
+
+#define net_ns_match(target, ns) ((target) == (ns))
+
+#else /* CONFIG_NET_NS */

 #define INIT_NET_NS(net_ns)

@@ -58,6 +85,24 @@ static inline int copy_net_ns(int flags,
 static inline void put_net_ns(struct net_namespace *ns)
 {
 }
-#endif
+
+#define current_net_ns   NULL
+
+static inline void init_current_net_ns(int cpu)
+{
+}
+
+static inline struct net_namespace *push_net_ns(struct net_namespace *ns)
+{
+ return NULL;
+}
+
+static inline void pop_net_ns(struct net_namespace *ns)
+{
+}
+
+#define net_ns_match(target, ns) ((void)(ns), 1)
+
+#endif /* !CONFIG_NET_NS */

 #endif /* _LINUX_NET_NAMESPACE_H */
--- linux-2.6.20-rc4-mm1.net_ns.orig/kernel/fork.c
+++ linux-2.6.20-rc4-mm1.net_ns/kernel/fork.c
@@ -1719,6 +1719,7 @@ asmlinkage long sys_unshare(unsigned lon
   if (new_net) {
    net = current->nsproxy->net_ns;
    current->nsproxy->net_ns = new_net;
+   pop_net_ns(new_net);
    new_net = net;
   }

--- linux-2.6.20-rc4-mm1.net_ns.orig/kernel/sched.c
+++ linux-2.6.20-rc4-mm1.net_ns/kernel/sched.c
@@ -53,6 +53,7 @@
 #include <linux/tsacct_kern.h>
```

```
 #include <linux/kprobes.h>
 #include <linux/delayacct.h>
+#include <linux/net_namespace.h>
 #include <asm/tlb.h>

 #include <asm/unistd.h>
@@ -1824,6 +1825,9 @@ static inline void finish_task_switch(st
   kprobe_flush_task(prev);
   put_task_struct(prev);
  }
+
+ (void)push_net_ns(current->nsproxy->net_ns);
+
 }

 /**
@@ -7066,6 +7070,7 @@ void __init sched_init(void)
    // delimiter for bitsearch
    __set_bit(MAX_PRIO, array->bitmap);
   }
+  init_current_net_ns(i);
  }

  set_load_weight(&init_task);
--- linux-2.6.20-rc4-mm1.net_ns.orig/net/core/net_namespace.c
+++ linux-2.6.20-rc4-mm1.net_ns/net/core/net_namespace.c
@@ -18,6 +18,9 @@ struct net_namespace init_net_ns = {

 #ifdef CONFIG_NET_NS

+DEFINE_PER_CPU(struct net_namespace *, exec_net_ns);
+EXPORT_PER_CPU_SYMBOL_GPL(exec_net_ns);
+
 /*
  * Clone a new ns copying an original net ns, setting refcount to 1
  * @old_ns: namespace to clone
```