

---

Subject: [PATCH 9/9] network namespaces: playing with pass-through device  
Posted by [Andrey Savochkin](#) on Tue, 15 Aug 2006 14:48:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Temporary code to debug and play with pass-through device.

Create device pair by

modprobe veth

echo 'add veth1 0:1:2:3:4:1 eth0 0:1:2:3:4:2' >/proc/net/veth\_ctl

and your shell will appear into a new namespace with `eth0' device.

Configure device in this namespace

ip l s eth0 up

ip a a 1.2.3.4/24 dev eth0

and in the root namespace

ip l s veth1 up

ip a a 1.2.3.1/24 dev veth1

to establish a communication channel between root namespace and the newly created one.

Signed-off-by: Andrey Savochkin <[saw@swsoft.com](mailto:saw@swsoft.com)>

---

veth.c | 113 ++++++-----+-----+-----+-----+-----+-----+-----+-----+

+++++

1 files changed, 113 insertions(+)

--- ./drivers/net/veth.c.veveth-dbg Tue Aug 15 13:47:48 2006

+++ ./drivers/net/veth.c Tue Aug 15 14:08:04 2006

@@ -251,6 +251,116 @@ void veth\_entry\_del\_all(void)

```
/* ----- *  
*  
+ * Temporary interface to create veth devices  
+ *  
+ * ----- */  
+  
+#ifdef CONFIG_PROC_FS  
+  
+static int veth_debug_open(struct inode *inode, struct file *file)  
{  
+ return 0;  
}  
+  
+static char *parse_addr(char *s, char *addr)  
{  
+ int i, v;  
+  
+ for (i = 0; i < ETH_ALEN; i++) {  
+ if (!isxdigit(*s))  
+ return NULL;
```

```

+ *addr = 0;
+ v = isdigit(*s) ? *s - '0' : toupper(*s) - 'A' + 10;
+ s++;
+ if (isxdigit(*s)) {
+   *addr += v << 16;
+   v = isdigit(*s) ? *s - '0' : toupper(*s) - 'A' + 10;
+   s++;
+ }
+ *addr++ += v;
+ if (i < ETH_ALEN - 1 && ispunct(*s))
+   s++;
+
+ return s;
+}
+
+extern int net_ns_start(void);
+static ssize_t veth_debug_write(struct file *file, const char __user *user_buf,
+ size_t size, loff_t *ppos)
+{
+ char buf[128], *s, *parent_name, *child_name;
+ char parent_addr[ETH_ALEN], child_addr[ETH_ALEN];
+ struct net_namespace *parent_ns, *child_ns;
+ int err;
+
+ s = buf;
+ err = -EINVAL;
+ if (size >= sizeof(buf))
+   goto out;
+ err = -EFAULT;
+ if (copy_from_user(buf, user_buf, size))
+   goto out;
+ buf[size] = 0;
+
+ err = -EBADRQC;
+ if (!strncmp(buf, "add ", 4)) {
+   parent_name = buf + 4;
+   if ((s = strchr(parent_name, ' ')) == NULL)
+     goto out;
+   *s = 0;
+   if ((s = parse_addr(s + 1, parent_addr)) == NULL)
+     goto out;
+   if (!*s)
+     goto out;
+   child_name = s + 1;
+   if ((s = strchr(child_name, ' ')) == NULL)
+     goto out;
+   *s = 0;
+   if ((s = parse_addr(s + 1, child_addr)) == NULL)

```

```

+ goto out;
+
+ parent_ns = get_net_ns(current_net_ns);
+ err = net_ns_start();
+ if (err)
+ goto out;
+ /* return to parent context */
+ push_net_ns(parent_ns, child_ns);
+ err = veth_entry_add(parent_name, parent_addr,
+ child_name, child_addr, child_ns);
+ pop_net_ns(child_ns);
+ put_net_ns(parent_ns);
+ if (!err)
+ err = size;
+
+out:
+ return err;
+}
+
+static struct file_operations veth_debug_ops = {
+ .open = &veth_debug_open,
+ .write = &veth_debug_write,
+};
+
+static int veth_debug_create(void)
+{
+ proc_net_fops_create("veth_ctl", 0200, &veth_debug_ops);
+ return 0;
+}
+
+static void veth_debug_remove(void)
+{
+ proc_net_remove("veth_ctl");
+}
+
+#else
+
+static int veth_debug_create(void) { return -1; }
+static void veth_debug_remove(void) { }
+
+#endif
+
+/* -----
+ * Information in proc
+ *
+ * ----- */
@@ @ -310,12 +420,15 @@ static inline void veth_proc_remove(void)

```

```
int __init veth_init(void)
{
+ if (veth_debug_create())
+ return -EINVAL;
veth_proc_create();
return 0;
}

void __exit veth_exit(void)
{
+ veth_debug_remove();
veth_proc_remove();
veth_entry_del_all();
}
```

---