

---

Subject: [PATCH] Module for ip utility to support veth device  
Posted by [Pavel Emelianov](#) on Wed, 06 Jun 2007 15:17:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The usage is

```
# ip link add [name] type veth [peer <name>] [mac <mac>] [peer_mac <mac>]
```

The Makefile is maybe not as beautiful as it could be. It is to be discussed.

One thing I noticed during testing is the following. When launching this with link\_veth.so module and not specifying any module specific parameters, the kernel refuses to accept the packet when parsing the IFLA\_LINKINFO. So the hunk for ip/iplink.c doesn't add an empty extra header if no extra data expected.

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

---

```
diff --git a/ip/Makefile b/ip/Makefile
index 9a5bfe3..b46bce3 100644
--- a/ip/Makefile
+++ b/ip/Makefile
@@ -8,8 +8,9 @@ RTMONOBJ=rtmon.o
ALLOBJ=$(IPOBJ) $(RTMONOBJ)
SCRIPTS=ifcfg rtptr routel routef
TARGETS=ip rtmon
+LIBS=link_veth.so
```

```
-all: $(TARGETS) $(SCRIPTS)
+all: $(TARGETS) $(SCRIPTS) $(LIBS)
```

```
ip: $(IPOBJ) $(LIBNETLINK) $(LIBUTIL)
```

```
@@ -24,3 +25,6 @@ clean:
```

```
LDLIBS += -ldl
LDFLAGS += -Wl,-export-dynamic
```

```
+
```

```
+% .so: %.c
```

```
+ $(CC) $(CFLAGS) -shared $< -o $@
```

```
diff --git a/ip/iplink.c b/ip/iplink.c
```

```
index 5170419..6975990 100644
```

```
--- a/ip/iplink.c
```

```
+++ b/ip/iplink.c
```

```
@@ -287,7 +287,7 @@ static int iplink_modify(int cmd, unsigned
    strlen(type));
```

```

    lu = get_link_type(type);
- if (lu) {
+ if (lu && argc) {
    struct rtattr * data = NLMSG_TAIL(&req.n);
    addattr_l(&req.n, sizeof(req), IFLA_INFO_DATA, NULL, 0);

diff --git a/ip/link_veth.c b/ip/link_veth.c
new file mode 100644
index 0000000..adfdef6
--- /dev/null
+++ b/ip/link_veth.c
@@ -0,0 +1,77 @@
#include <string.h>
+
#include "utils.h"
#include "ip_common.h"
#include "veth.h"
+
#define ETH_ALEN 6
+
static void usage(void)
+{
+ printf("Usage: ip link add ... "
+ "[peer <peer-name>] [mac <mac>] [peer_mac <mac>]\n");
+}
+
static int veth_parse_opt(struct link_util *lu, int argc, char **argv,
+ struct nlmsg_hdr *hdr)
+{
+ __u8 mac[ETH_ALEN];
+
+ for (; argc != 0; argv++, argc--) {
+ if (strcmp(*argv, "peer") == 0) {
+ argv++;
+ argc--;
+ if (argc == 0) {
+ usage();
+ return -1;
+ }
+ addattr_l(hdr, 1024, VETH_INFO_PEER,
+ *argv, strlen(*argv));
+
+ continue;
+ }
+
+ if (strcmp(*argv, "mac") == 0) {

```

```

+ argv++;
+ argc--;
+ if (argc == 0) {
+     usage();
+     return -1;
+ }
+
+ if (hexstring_a2n(*argv, mac, sizeof(mac)) == NULL)
+     return -1;
+
+ addattr_l(hdr, 1024, VETH_INFO_MAC,
+     mac, ETH_ALEN);
+ continue;
+ }
+
+ if (strcmp(*argv, "peer_mac") == 0) {
+     argv++;
+     argc--;
+     if (argc == 0) {
+         usage();
+         return -1;
+     }
+
+     if (hexstring_a2n(*argv, mac, sizeof(mac)) == NULL)
+         return -1;
+
+     addattr_l(hdr, 1024, VETH_INFO_PEER_MAC,
+         mac, ETH_ALEN);
+     continue;
+ }
+
+ usage();
+ return -1;
+ }
+
+ return 0;
+}
+
+struct link_util veth_link_util = {
+ .id = "veth",
+ .parse_opt = veth_parse_opt,
+};
diff --git a/ip/veth.h b/ip/veth.h
new file mode 100644
index 0000000..74c8e1e
--- /dev/null
+++ b/ip/veth.h
@@ -0,0 +1,13 @@

```

```
+ #ifndef __NET_VETH_H__
+ #define __NET_VETH_H__
+
+ enum {
+ VETH_INFO_UNSPEC,
+ VETH_INFO_MAC,
+ VETH_INFO_PEER,
+ VETH_INFO_PEER_MAC,
+
+ VETH_INFO_MAX
+ };
+
+ #endif
```

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---