NAME

routing – system support for packet network routing

DESCRIPTION

The network facilities provide general packet routing. Routing table maintenance may be implemented in applications processes.

A simple set of data structures compose a “routing table” used in selecting the appropriate network interface when transmitting packets. This table contains a single entry for each route to a specific network or host. The routing table was designed to support routing for the Internet Protocol (IP), but its implementation is protocol independent and thus it may serve other protocols as well. User programs may manipulate this data base with the aid of two **ioctl**(2) commands, **SIOCADDRT** and **SIOCDELRT**. These commands allow the addition and deletion of a single routing table entry, respectively. Routing table manipulations may only be carried out by privileged user.

A routing table entry has the following form, as defined in **/usr/include/net/route.h**:

```
struct rtentry {
        u_long              rt_hash;     /* to speed lookups */
        struct sockaddr     rt_dst;      /* key */
        struct sockaddr     rt_gateway;  /* value */
        short               rt_flags;    /* up/down?, host/net */
        short               rt_refcnt;   /* # held references */
        u_long              rt_use;      /* raw # packets forwarded */
#ifdef STRNET
        struct ip_provider  *rt_prov;    /* the answer: provider to use */
#else
  struct  ifnet             *rt_ifp;     /* the answer: interface to use */
#endif /* STRNET*/
};
```

with *rt_flags* defined from:

```
#define  RTF_UP         0x1     /* route usable */
#define  RTF_GATEWAY    0x2     /* destination is a gateway */
#define  RTF_HOST       0x4     /* host entry (net otherwise) */
```

Routing table entries come in three flavors: for a specific host, for all hosts on a specific network, for any destination not matched by entries of the first two types (a wildcard route). Each network interface installs a routing table entry when it it is initialized. Normally the interface specifies the route through it is a “direct” connection to the destination host or network. If the route is direct, the transport layer of a protocol family usually requests the packet be sent to the same host specified in the packet. Otherwise, the interface may be requested to address the packet to an entity different from the eventual recipient (that is, the packet is forwarded).

Routing table entries installed by a user process may not specify the hash, reference count, use, or interface fields; these are filled in by the routing routines. If a route is in use when it is deleted (**rt_refcnt** is non-zero), the resources associated with it will not be reclaimed until all references to it are removed.

User processes read the routing tables through the **/dev/ip** device.

The *rt_use* field contains the number of packets sent along the route. This value is used to select among multiple routes to the same destination. When multiple routes to the same destination exist, the least used route is selected.

A wildcard routing entry is specified with a zero destination address value. Wildcard routes are used only when the system fails to find a route to the destination host and network. The combination of wildcard routes and routing redirects can provide an economical mechanism for routing traffic.

ERRORS

EEXIST A request was made to duplicate an existing entry.

ESRCH A request was made to delete a non-existent entry.

ENOBUFS Insufficient resources were available to install a new route.

ENOMEM Insufficient resources were available to install a new route.

ENETUNREACH The gateway is not directly reachable i.e. it does not match the destination/subnet on any of the network interfaces.

FILES

/dev/ip IP device driver

SEE ALSO

route(1M), **ioctl**(2)