NAME

arp, ARP – Address Resolution Protocol

SYNOPSIS

#include <sys/fcntl.h>
#include <sys/socket.h>
#include <net/if_arp.h>
#include <netinet/in.h>

s = socket(AF_INET, SOCK_DGRAM, 0);

d = open ("/dev/arp", *oflag***);**

DESCRIPTION

ARP is a protocol used to map dynamically between Internet Protocol (IP) and 10Mb/s Ethernet addresses. It is used by all the 10Mb/s Ethernet datalink providers (interface drivers) and it can be used by other datalink providers that support broadcast (such as FDDI and Token Ring). ARP is not specific to the Internet Protocol but this implementation supports only that network layer protocol.

ARP caches IP-to-Ethernet address mappings. When an interface requests a mapping for an address not in the cache, ARP queues the message that requires the mapping and broadcasts a message on the associated network requesting the address mapping. If a response is provided, the new mapping is cached and any pending message is transmitted. ARP will queue at most four packets while waiting for a mapping request to be responded to; only the four most recently transmitted packets are kept.

APPLICATION PROGRAMMING INTERFACE

The STREAMS device **/dev/arp** is not a Transport Level Interface (TLI) transport provider and may not be used with the TLI interface.

To facilitate communications with systems which do not use ARP, **ioctl( )** requests are provided to enter and delete entries in the IP-to-Ethernet tables.

```
#include <sys/sockio.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/if_arp.h>
struct arpreq arpreq;
ioctl(s, SIOCSARP, (caddr_t)&arpreq);
ioctl(s, SIOCGARP, (caddr_t)&arpreq);
ioctl(s, SIOCDARP, (caddr_t)&arpreq);
```

Each **ioctl( )** request takes the same structure as an argument. **SIOCSARP** sets an ARP entry, **SIOCGARP** gets an ARP entry, and **SIOCDARP** deletes an ARP entry. These **ioctl( )** requests may be applied to any Internet family socket descriptor *s*, or to a descriptor for the ARP device, but only by the privileged user.

The **arpreq** structure contains:

```
/*
 * ARP ioctl request
 */
struct arpreq {
        struct sockaddr    arp_pa;          /* protocol address */
        struct sockaddr    arp_ha;          /* hardware address */
        int                arp_flags;       /* flags */
};

/*  arp_flags field values */
#define ATF_COM            0x2  /* completed entry (arp_ha valid) */
#define ATF_PERM           0x4  /* permanent entry */
#define ATF_PUBL           0x8  /* publish (respond for other host) */
#define ATF_USETRAILERS 0x10  /* send trailer packets to host */
```

The address family for the **arp_pa sockaddr** must be **AF_INET**; for the **arp_ha sockaddr** it must be **AF_UNSPEC**. The only flag bits that may be written are **ATF_PUBL** and **ATF_USETRAILERS. ATF_PERM** makes the entry permanent if the **ioctl( )** request succeeds. The peculiar nature of the ARP tables may cause the **ioctl( )** request to fail if too many permanent IP addresses hash to the same slot. **ATF_PUBL** specifies that the ARP code should respond to ARP requests for the indicated host coming from other machines. This allows a host to act as an "ARP server", which may be useful in convincing an ARP-only machine to talk to a non-ARP machine.

ARP is also used to negotiate the use of trailer IP encapsulations; trailers are an alternate encapsulation used to allow efficient packet alignment for large packets despite variable-sized headers. Hosts that wish to receive trailer encapsulations so indicate by sending gratuitous ARP translation replies along with replies to IP requests; they are also sent in reply to IP translation replies. The negotiation is thus fully symmetrical, in that either or both hosts may request trailers. The **ATF_USETRAILERS** flag is used to record the receipt of such a reply, and enables the transmission of trailer packets to that host.

ARP watches passively for hosts impersonating the local host (that is, a host which responds to an ARP mapping request for the local host's address).

SEE ALSO

arp(1M), **ifconfig**(1M), **if_tcp**(7P), **inet**(7P)

Plummer, Dave, ''*An Ethernet Address Resolution Protocol -or- Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware*,'' RFC 826, Network Information Center, SRI International, Menlo Park, Calif., November 1982.

Leffler, Sam, and Michael Karels, ''*Trailer Encapsulations*,'' RFC 893, Network Information Center, SRI International, Menlo Park, Calif., April 1984.

DIAGNOSTICS

IP: Hardware address '%x:%x:%x:%x:%x:%x' trying to be our address '%d.%d.%d.%d'!
Duplicate IP address. ARP has discovered another host on the local network which responds to mapping requests for the Internet address of this system.

IP: Proxy ARP problem? Hardware address ′%x:%x:%x:%x:%x:%x′ thinks it is ′%d.%d.%d.%d′

This message will appear if **arp**(1M) has been used to create a published entry and some other host on the local network responds to mapping requests for the published arp entry.