

Linux操作系统内核ARP的设计实现概述 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E6_93_8D_E4_BD_c103_144947.htm ARP (Address Resolution Protocol)

是用来将IP地址转化成机器的网卡物理地址（硬件地址）。当一台机器要向另外一台物理上相连的机器发送IP包的时候，它要先检查一下自己的ARP缓存，试图找到对方的硬件地址，如果找不到的话，将要发送的IP包放入等待队列中，接着发出一个ARP请求。等到收到ARP应答的时候，构造好原来正在等待的IP包的ethernet头部（目的硬件地址，源硬件地址），再将这个IP包发送出去。Linux的ARP实现相当复杂，部分原因是Linux不止要支持ethernet，还要支持其它类型的网络，另外一部分原因是ARP的实现实际上是和路由处理（routing）相关，所以理解起来不太容易。首先我要讲一下ARP函数的调用过程：(1) 当系统初始化时，调用arp_init来初始化ARP缓存（arp_tbl），并且注册ARP协议的接收函数。(2) 当网卡驱动程序收到一个网络包（packet）的时候，会分配一个sk_buff(skb)，将数据拷贝进这个缓冲区，然后调用netif_rx把skb放入等待队列（input_pkt_queue）中，并且产生一个软中断。当系统处理这个软中断的时候，会调用net_rx_action，它根据网络包的类型，调用相应的接收函数来处理。如果是ARP包，则调用arp_rcv。(3) arp_rcv判断这个arp请求是不是询问本机或者本机代理的硬件地址，如果是的话，调用arp_send发回arp应答。另外arp_rcv还尽量保留对方机器的mac address。(4) arp_send分配一个sk_buff(skb)，填好arp包的类型，源硬件地址，源IP地址，目的硬件地址，目

的IP地址，然后调用dev_queue_xmit这个arp包发送出去。其次，描述一下ARP主要的数据结构：(1) neigh_table

neigh_table是一个用来描述物理上互相连接的机器的信息的哈希表，ARP缓存arp_tbl就是这样的一个neigh_table。系统中所有的neigh_table都连在一起。下面是一些主要的域：

```
struct neighbour *hash_buckets[NEIGH_HASHMASK 1]. hash_buckets  
存放着所有邻居（物理上相连的机器）的信息，共有32  
个bucket，每一个bucket存放着一条neighbor链表。 struct  
pneigh_entry *phash_buckets[PNEIGH_HASHMASK  
1].phash_buckets存放着所有proxy arp的entry，每一个entry由  
网卡设备和ip地址组成，指明由哪个网卡设备代理哪个ip  
的mac地址。共有16个bucket. int family. 网络类型，  
为AF_INET int entry_size. 大小为sizeof(struct neighbour) 4 int  
key_len. 键的长度，为4 __u32 (*hash)(const void *pkey, const  
struct net_device *).int (*constructor)(struct neighbour *).int  
(*pconstructor)(struct pneigh_entry *).这几个分别是ARP的哈希  
函数，neighbour和pneigh_entry的构造函数， struct  
neigh_parms parms. 100Test 下载频道开通，各类考试题目直接  
下载。详细请访问 www.100test.com
```