Lab4

171830635 俞星凯

|  |  |
| --- | --- |
| 实验目的 | 1. 设计和实现IP包的静态路由转发程序。 2. 加深对链路层和网络层协议衔接及静态路由的理解。 |
| 数据结构说明 | 1. 路由配置相关数据结构   struct route\_item{  char destination[16];  char gateway[16];  char netmask[16];  char interface[16];  }route\_info[MAX\_ROUTE\_INFO];  int route\_item\_index=0;  struct arp\_table\_item{  char ip\_addr[16];  char mac\_addr[18];  }arp\_table[MAX\_ARP\_SIZE];  int arp\_item\_index =0;  struct device\_item{  char interface[14];  char ip\_addr[16];  char mac\_addr[18];  }device[MAX\_DEVICE];  int device\_index=0;  上述数据结构分别定义了静态路由表，ARP 缓存和设备信息。   1. 路由转发相关数据结构   char recv\_buf[256];  char send\_buf[256];  struct sockaddr\_ll src\_ll;  struct sockaddr\_ll dest\_ll;  两个缓冲区用于存储收发的数据。两个sockaddr\_ll 类型变量用于处理收发的源MAC 地址和目的MAC 地址。 |
| 配置文件说明 | IMG_256Router1 的配置文件如左。  可以分为3个部分：  第一部分是3条路由规则，每条有4个表项。  第二部分是2条ARP 缓存，每条有2个表项。  第三部分是2条设备信息，每条有3个表项。 |
| 程序设计的思路以  及运行流程 | IMG_256   1. 运行PC1时创建SOCK\_DGRAM类型套接字，则发送/接收的数据包会自动添加/去除以太网帧头部。从argv[1]获取目的IP 地址，根据静态路由表获取对应网关和接口，在ARP 缓存中查找网关对应的MAC 地址，在设备信息中查找接口对应的设备号，对sockaddr\_ll 类型变量dest\_ll 做好赋值。在发送缓冲区内填入IP，ICMP的必要信息，即可发送ICMP\_ECHO报文。 2. 运行Route1和Router2是一样的。在接收缓冲区中判断目的IP 是否为本地IP，若是，则发送ICMP\_ECHOREPLY 报文，若否，则根据静态路由表转发。转发操作同上。需要注意的是，在本实验中ICMP\_ECHOREPLY 采用了简化实现，由于在recvfrom 时操作系统为sockaddr\_ll 类型变量src\_ll 进行了赋值，故直接将接收缓冲区的内容拷贝到发送缓冲区并利用src\_ll 发送即可。 3. 运行PC2与Router类似，只不过其不具备转发功能，只有ICMP\_ECHOREPLY 功能。 |
| 运行结果截图 | 1. 在每个 PC 和 router 上运行 ifconfig，确保 IP 地址为空。   PC1：IMG_256  PC2：  IMG_256  Router1：  IMG_256  Router2：  IMG_256   1. 用PC1 向PC2 发送ICMP\_ECHO报文，路由器转发报文，PC2 回复ICMP\_ECHOREPLY报文。   PC1：  IMG_256  Router1：  IMG_256  Router2：  IMG_256  PC2：  IMG_256 |
| 相关参考资料 | 仅参考实验教材独立完成。 |
| 代码个人创新以及  思考 | 本实验使用SOCK\_DGRAM类型套接字，编程成功的关键有三：对sockaddr\_ll 类型变量的正确使用，静态路由表的查找以及IP，IPCMP报文的设置。本实验用传统的C语言实现，简洁高效。代码方面创新包括ICMP\_ECHOREPLY 的简化处理以及断言机制的应用。 |