Lab2

171830635 俞星凯

|  |  |
| --- | --- |
| 实验目的 | 1. 学习socket相关知识 2. 通过学习ip，icmp，arp的报文格式来理解这些协议的功能和作用 3. 掌握初步raw sockt编程基础 |
| 数据结构说明 | 1. 在Linux中，IP报头格式数据结构（<netinet/ip.h>）定义如下：   struct ip { #if \_\_BYTE\_ORDER == \_\_LITTLE\_ENDIAN unsigned int ip\_hl:4; /\* header length \*/ unsigned int ip\_v:4; /\* version \*/ #endif #if \_\_BYTE\_ORDER == \_\_BIG\_ENDIAN unsigned int ip\_v:4; /\* version \*/ unsigned int ip\_hl:4; /\* header length \*/ #endif u\_int8\_t ip\_tos; /\* type of service \*/ u\_short ip\_len; /\* total length \*/ u\_short ip\_id; /\* identification \*/ u\_short ip\_off; /\* fragment offset field \*/ #define IP\_RF 0x8000 /\* reserved fragment flag \*/ #define IP\_DF 0x4000 /\* dont fragment flag \*/ #define IP\_MF 0x2000 /\* more fragments flag \*/ #define IP\_OFFMASK 0x1fff /\* mask for fragmenting bits \*/ u\_int8\_t ip\_ttl; /\* time to live \*/ u\_int8\_t ip\_p; /\* protocol \*/ u\_short ip\_sum; /\* checksum \*/ struct in\_addr ip\_src, ip\_dst; /\* source and dest address \*/ };  IMG_256   1. 在Linux中，ICMP\_ECHO和ICMP\_ECHOREPLY数据结构（<netinet/ip\_icmp.h>）定义如下：   struct icmp { u\_int8\_t icmp\_type; /\* type of message, see below \*/ u\_int8\_t icmp\_code; /\* type sub code \*/ u\_int16\_t icmp\_cksum; /\* ones complement checksum of struct \*/  struct ih\_idseq /\* echo datagram \*/ { u\_int16\_t icd\_id; u\_int16\_t icd\_seq; } ih\_idseq;  }  IMG_256 |
| 程序设计的思路以  及运行流程 | 1. raw\_socket   创建一个raw socket，不断收包，显示包的MAC和IP地址，并根据包头类型域的值来显示包的类型。   1. raw\_socket\_ping 注：括号内为本次程序的函数名 2. 根据参数是主机名还是IP地址进行不同设置 3. 创建一个raw socket 4. 发送一个包(send\_packet)，需要先对包头进行封装(pack)，其中检验和字段采用检验和算法(cal\_cksum)。 5. 接收一个包(recv\_packet)，然后再对包头进行解封(unpack)，计算rtt(cal\_internal)，输出相关信息。 6. 循环(3)(4) 7. 进行数据统计。 |
| 运行结果截图 | 1. raw\_socket   IMG_256   1. raw\_socket\_ping   IMG_256 |
| 相关参考资料 | 计算机网络实验教材2.02(修订)  https://blog.csdn.net/xtank\_nie/article/details/39215225 |
| 对比样例程序 | 参考了教材上的raw\_socket样例程序，并对其略做修改使之支持ARP的识别。 |
| 代码个人创新以及思考 | 1. raw\_socket   根据MAC包的第13字节判断IP数据包还是ARP请求/应答。   1. raw\_socket\_ping 2. 修改了计算rtt的算法，使之能够正确显示到小数点后三位。 3. 由于只需实现简易的ping程序，故简化了代码中的大量冗余语句。 4. 采用系统ping程序的格式进行输出。 |
| 该程序的应用场景创新（非必须） | 在多跳网络中每一跳都显示一次，即可判断断路位置。 |