专题19-移植TFTP客户端

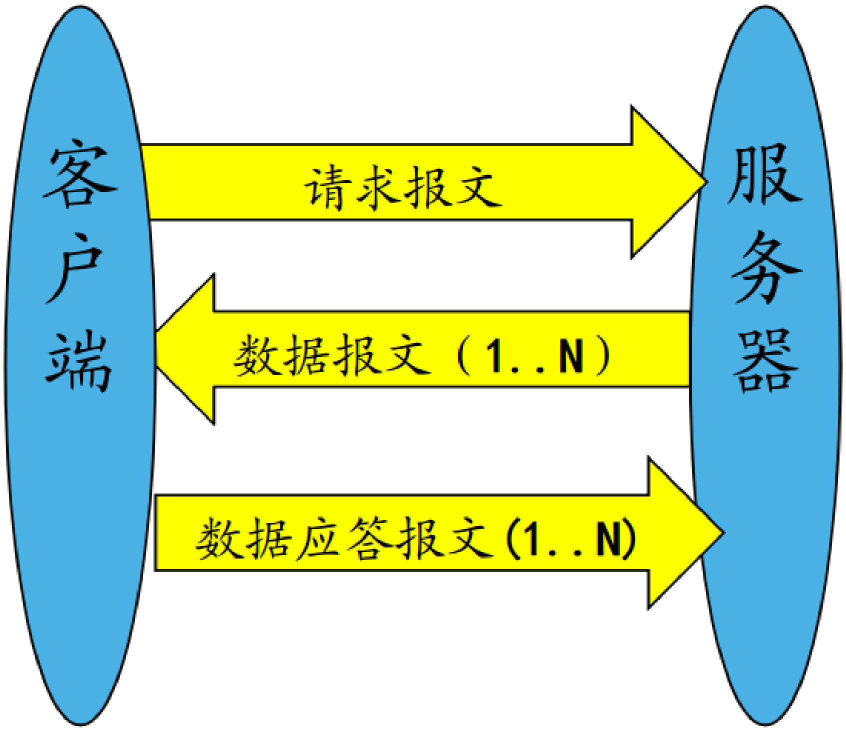
一、TFTP网络协议分析

1.1、TFTP作用

用于交叉开发的文件下载。

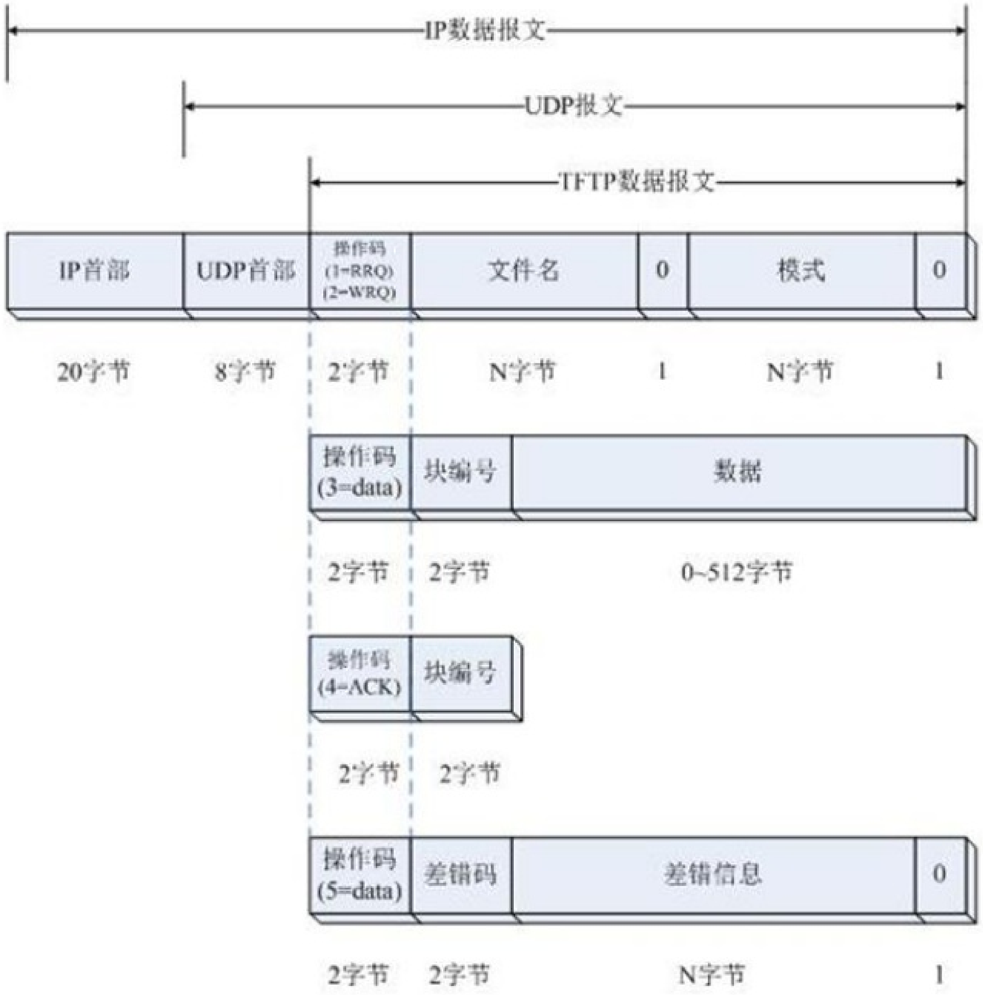
1.2、TFTP协议分析

1.2.1、TFTP交互过程



操作流程	TFTP客户端	TFTP服务器
1	请求包（包含文件名，传输类型，文件大小=0）	
2		配置应答包0（包含文件大小）
3	应答包0	
4		数据包1（内部包含512bytes数据
5	应答包1	
6		数据包2（内部包含512bytes数据
7	应答包2	
8		...
9	...	
10		最后一个数据包last
11	应答last	

1.2.2、TFTP报文格式



二、TFTP网络协议实现

构造TFTP网络协议包

```
#include "string.h"
#include "arp.h"

unsigned char sendbuf[1024];

unsigned short checksum(unsigned char * ptr, int len)
{
    unsigned long sum = 0;
    unsigned short *p = (unsigned short *)ptr;

    while (len > 1)
    {
        sum += *p++;
        len -= 2;
    }

    if(len == 1)
        sum += *(unsigned char *)p;

    while(sum>>16)
        sum = (sum&0xffff) + (sum>>16);

    return (unsigned short)((~sum)&0xffff);
}

void tftp_send_request(const char *filename)
{
 unsigned char *ptftp = sendbuf[200];
 unsigned long tftp_len = 0;
 UDP_HDR * udphdr;
 unsigned char * iphdr;
 /*Constructs a TFTP packet*/

 /*option code: 1=RRQ;2=WRQ*/
 ptftp[0] = 0x00;
 ptftp[1] = 0x01;
 tftp_len += 2;

 /*file name*/
 sprintf(&ptftp[tftp_len], "%s", filename);
 tftp_len += strlen(filename);
 ptftp[tftp_len] = "0";
 tftp_len += 1;

 /*mode*/
 sprintf(&ptftp[tftp_len], "%s", "octect");
 tftp_len += strlen("octect");
 ptftp[tftp_len] = "0";
 tftp_len += 1;

 /*Constructs a UDP packet*/
 udphdr = ptftp - sizeof(UDP_HDR);
 iphdr = udphdr + sizeof(ETH_HDR);

 udphdr->sport = HON(48915);
 udphdr->dport = HON(69);
 udphdr->len =  HON(tftp_len + sizeof(UDP_HDR) - sizeof(IP_HDR));
 udphdr->udpchecksum = 0x00;

 /*Constructs a IP packet*/
 udphdr->iphdr.vhl = 0x45;
 udphdr->iphdr.tos = 0x00;
 udphdr->iphdr.len = HON(tftp_len + sizeof(UDP_HDR) - sizeof(ETH_HDR));
 udphdr->iphdr.ipid = HON(0x00);
 udphdr->iphdr.ipoffset = HON(0x4000);
 udphdr->iphdr.ttl = 0xff;
 udphdr->iphdr.proto = 0x11;
 memcpy(udphdr->iphdr.srcipaddr, ip_addr, 4);
 memcpy(udphdr->iphdr.destipaddr, host_ip_addr, 4);
 udphdr->iphdr.ipchecksum = 0;
 udphdr->iphdr.ipchecksum = checksum(iphdr, 20);

 /*Constructs a ETH packet*/
 memcpy(udphdr->iphdr.ethhdr.s_mac, mac_addr, 6);
 memcpy(udphdr->iphdr.ethhdr.d_mac, host_mac_addr, 6);
 udphdr->iphdr.ethhdr.type = HON(PROTO_IP);

  /*call eth_send function */
 eth_send((unsigned long *)udphdr, (sizeof(UDP_HDR) + tftp_len));
}
```

因前面ARP或者DM9000的程序有未知问题，导致存在故障，无法完美的实现之后的功能，所以这两章节在后续有时间处理之后再完成。