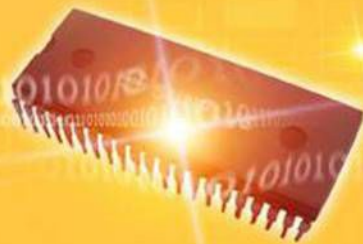


嵌入式系统工程师



UDP编程-TFTP、广播、多播



- TFTP简介、通信过程
- TFTP协议分析
- 练习一-TFTP客户端
- UDP广播
- UDP多播



- TFTP简介、通信过程
- TFTP协议分析
- 练习—TFTP客户端
- UDP广播
- UDP多播



➤ TFTP：简单文件传送协议

- 最初用于引导无盘系统，被设计用来传输小文件

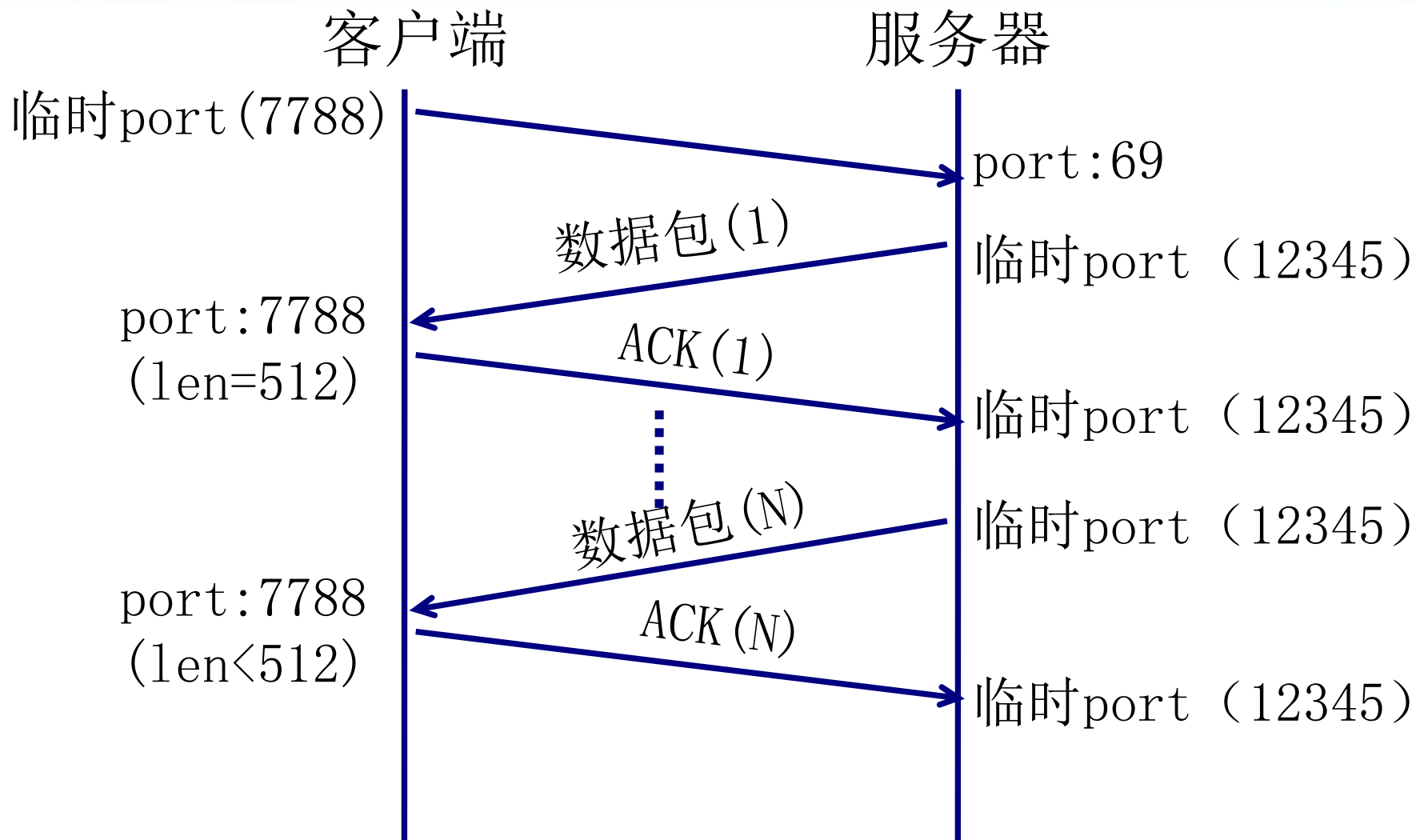
➤ 特点：

- 基于UDP协议实现
- 不进行用户有效性认证

➤ 数据传输模式

- octet：二进制模式
- netascii：文本模式
- mail：已经不再支持

TFTP通信过程（无选项）



➤ TFTP通信过程总结（无选项）

- 服务器在69号端口等待客户端的请求
- 服务器若批准此请求, 则使用临时端口与客户端进行通信
- 每个数据包的编号都有变化（从1开始）
- 每个数据包都要得到ACK的确认如果出现超时, 则需要重新发送最后的包（数据或ACK）
- 数据的长度以512Byte传输
- 小于512Byte的数据意味着传输结束

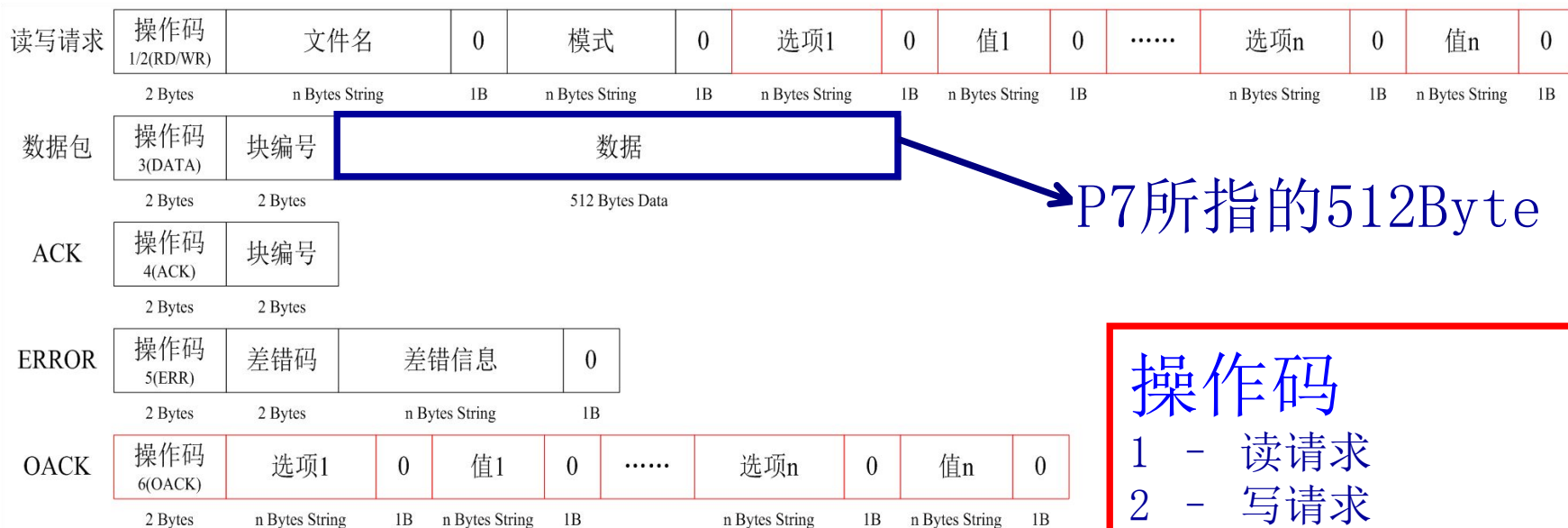
➤ 想一想

- TFTP协议的格式是什么样子？

- TFTP简介、通信过程
- TFTP协议分析
- 练习—TFTP客户端
- UDP广播
- UDP多播



➤ TFTP协议详解



操作码

- 1 - 读请求
- 2 - 写请求
- 3 - 数据包
- 4 - 数据确认
- 5 - 错误信息
- 6 - 选项确认

➤ 注意

- 以上的0代表的是' \0'
- 不同的差错码对应不同的错误信息

➤ TFTP协议详解

ERROR	操作码 5(ERR)	差错码	差错信息	0
-------	---------------	-----	------	---

```
$ ./tftpc.exe 127.0.0.1 file_id
[SEND len = 操作码 文件名 模式
0x0x22c4b0:00 01 66 69 6C 65 5F 69 64 00 6F 63 74 65 74 00 !..file_id.octet.
0x0x22c4c0:62 6C 6B 73 69 7A 65 00 35 31 32 00 74 73 69 7A !blksize.512.tsiz
0x0x22c4d0:65 00 30 00 74 69 6D 65 6F 75 74 00 35 00 ie.0.timeout.5.
[RECU len = 191 错误号 错误信息描述
0x0x2 ERR 00 05 00 01 46 69 6C 65 20 6E 6F 74 20 66 6F 75 !....File not fou
0x0x22c4c0:6E 64 00 ind.
[SEND len = 41
0x0x22c4ac:08 00 00 00 !....
[RECU len = 01
0x0x22c4b0: :
```

错误码

- 0 未定义, 参见错误信息
- 1 File not found.
- 2 Access violation.
- 3 Disk full or allocation exceeded.
- 4 illegal TFTP operation.
- 5 Unknown transfer ID.
- 6 File already exists.
- 7 No such user.
- 8 Unsupported option(s) requested.

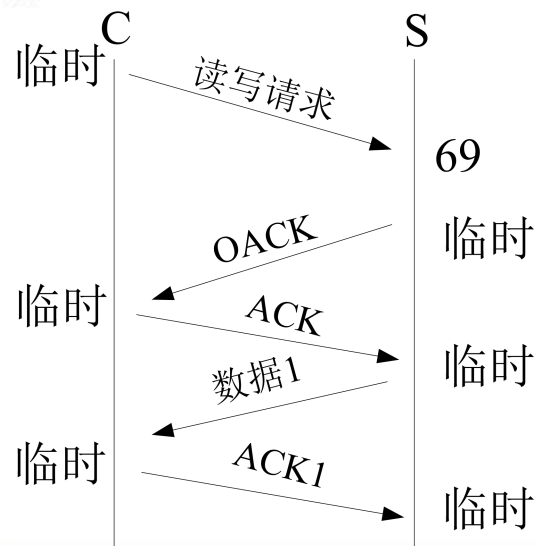
➤ 想一想

- 传输的数据的大小一定是512Byte吗?
- 由于网络的原因, 一方收不到另一方的数据怎么办?

➤ 读写请求中修改了选项



➤ 如果发送带选项的读写请求



➤ tsize选项

- 当读操作时，tsize选项的参数必须为“0”，服务器会返回待读取的文件的大小
- 当写操作时，tsize选项参数应为待写入文件的大小，服务器会回显该选项

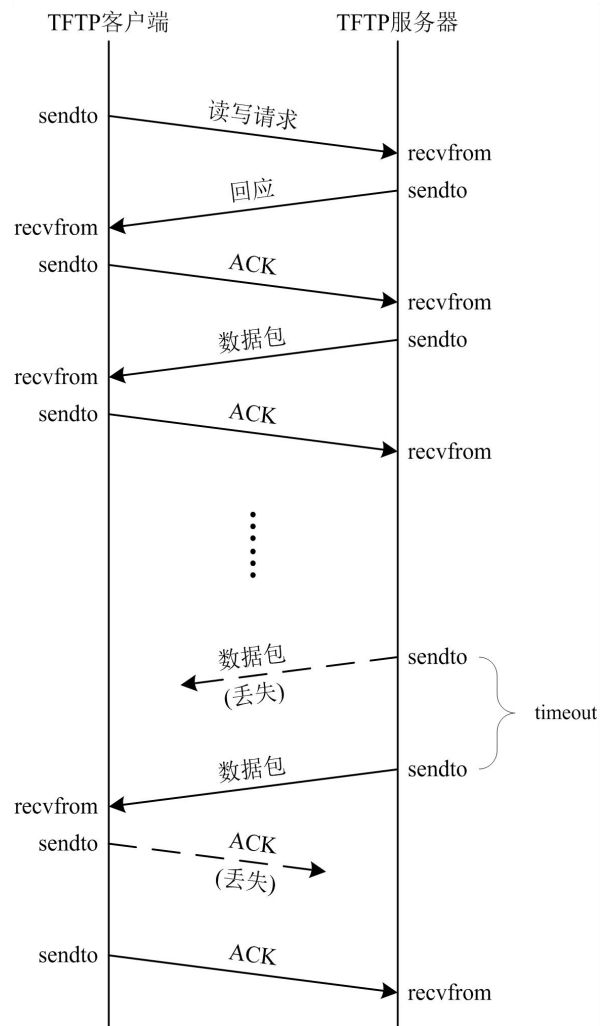
➤ blksize选项

- 修改传输文件时使用的数据块的大小（范围：8～65464）

➤ timeout选项

- 修改默认的数据传输超时时间（单位：秒）

TFTP通信过程（带选项）



➤ TFTP通信过程总结（带选项）

- 可以通过发送带选项的读/写请求发送给server
- 如果server允许修改选项则发送选项修改确认包
- server发送的数据、选项修改确认包都是临时port
- server通过timeout来对丢失数据包的重新发送

- TFTP简介、通信过程
- TFTP协议分析
- 练习一TFTP客户端
- UDP广播
- UDP多播



➤ 练习要求

- 使用TFTP协议，下载server上的文件到本地

➤ 实现思路

- ①构造请求报文，送至服务器(69号端口)
 - ②等待服务器回应
 - ③分析服务器回应
 - ④接收数据, 直到接收到的数据包小于规定数据长度
- 详细参考 《TFTP文件下载流程图.pdf》

- TFTP简介、通信过程
- TFTP协议分析
- 练习—TFTP客户端
- UDP广播
- UDP多播



- 在生活中广播无处不在，像平时在大街上卖物品的商人，在进行推广产品的时候往往都是使用一个喇叭来进行“吆喝”，这就是广播



- 想一想：在网络中的广播是什么样子？

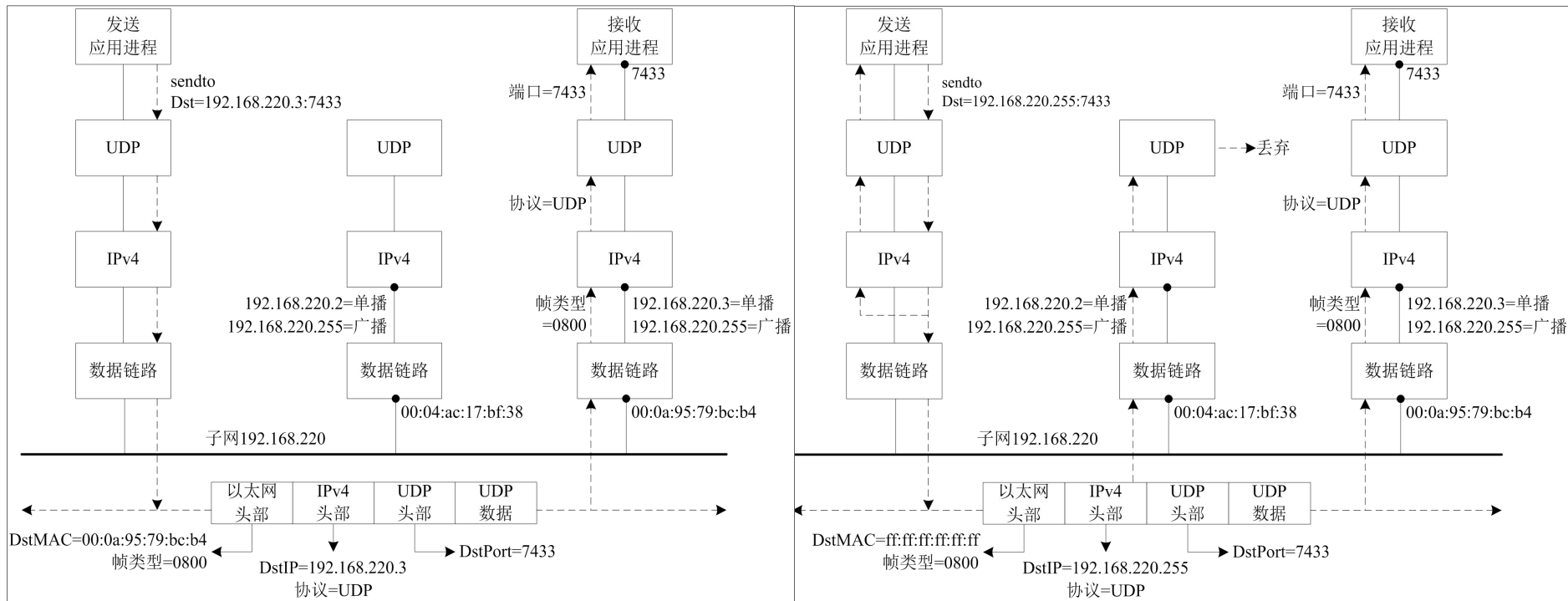
- 广播：由一台主机向该主机所在子网内的所有主机发送数据的方式
- 广播只能用UDP或原始IP实现，不能用TCP
- 广播的用途
 - 单个服务器与多个客户主机通信时减少分组流通
 - 地址解析协议（ARP）
 - 动态主机配置协议（DHCP）
 - 网络时间协议（NTP）

➤ UDP广播的特点

- 处于同一子网的所有主机都必须处理数据
- UDP数据包会 协议栈向上一直到UDP层
- 运行音视频等较高速率工作的应用，会带来大负
- 局限于局域网内使用

- {网络ID, 主机ID}
 - 网络ID表示由子网掩码中1覆盖的连续位
 - 主机ID表示由子网掩码中0覆盖的连续位
- 定向广播地址：主机ID全1
 - 例：对于192.168.220.0/24，其定向广播地址为192.168.220.255
 - 通常路由器不转发该广播
- 受限广播地址：255.255.255.255
 - 路由器从不转发该广播

广播与单播的对比



单播

广播

➤ 套接口选项

- `int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);`
- 成功执行返回0，否则返回-1

level	optname	说明	optval类型
SOL_SOCKET	SO_BROADCAST	允许发送广播数据包	int
	SO_RCVBUF	接收缓冲区大小	int
	SO_SNDBUF	发送缓冲区大小	int

► 广播示例:

```
14  int main(int argc, char *argv[])
15  {
16      int sock_fd = 0;
17      char buff[1024] = "";
18      unsigned short port = 8000;
19      struct sockaddr_in send_addr;
20
21      bzero(&send_addr, sizeof(send_addr));
22      send_addr.sin_family = AF_INET;
23      send_addr.sin_port = htons(port);
24
25      sock_fd = socket(AF_INET, SOCK_DGRAM, 0);
26      if(sock_fd < 0)
27      {
28          perror("socket failed");
29          close(sock_fd);
30          exit(1);
31      }
```


➤ 广播示例

```
33     if(argc > 1)
34     {
35         send_addr.sin_addr.s_addr = inet_addr(argv[1]);
36     }
37     else
38     {
39         printf("not have a server IP");
40         exit(1);
41     }
42     int yes = 1;
43     setsockopt(sock_fd, SOL_SOCKET, SO_BROADCAST, &yes, sizeof(yes));
44     strcpy(buff, "boardcast sucess");
45     int len = sendto(sock_fd, buff, strlen(buff), 0, \
46     (struct sockaddr *)&send_addr, sizeof(send_addr));
47     if(len < 0)
48     {
49         printf("send error\n");
50         close(sock_fd);
51         exit(1);
52     }
53     return 0;
54 }
```

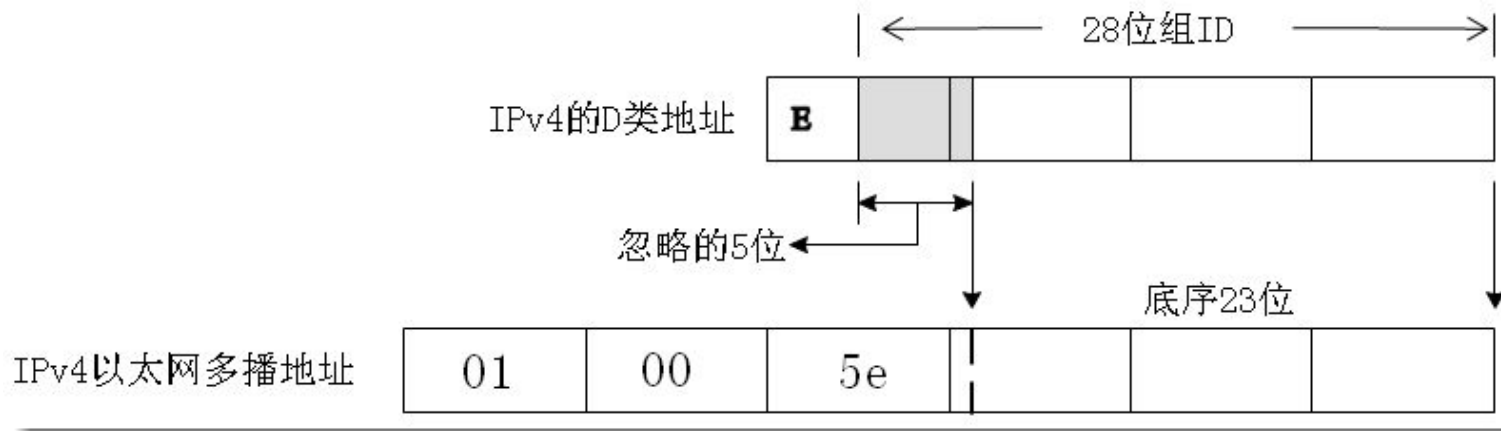
- 概述
- 基本UDP编程
- UDP编程实例——TFTP协议实现
- UDP广播
- UDP多播



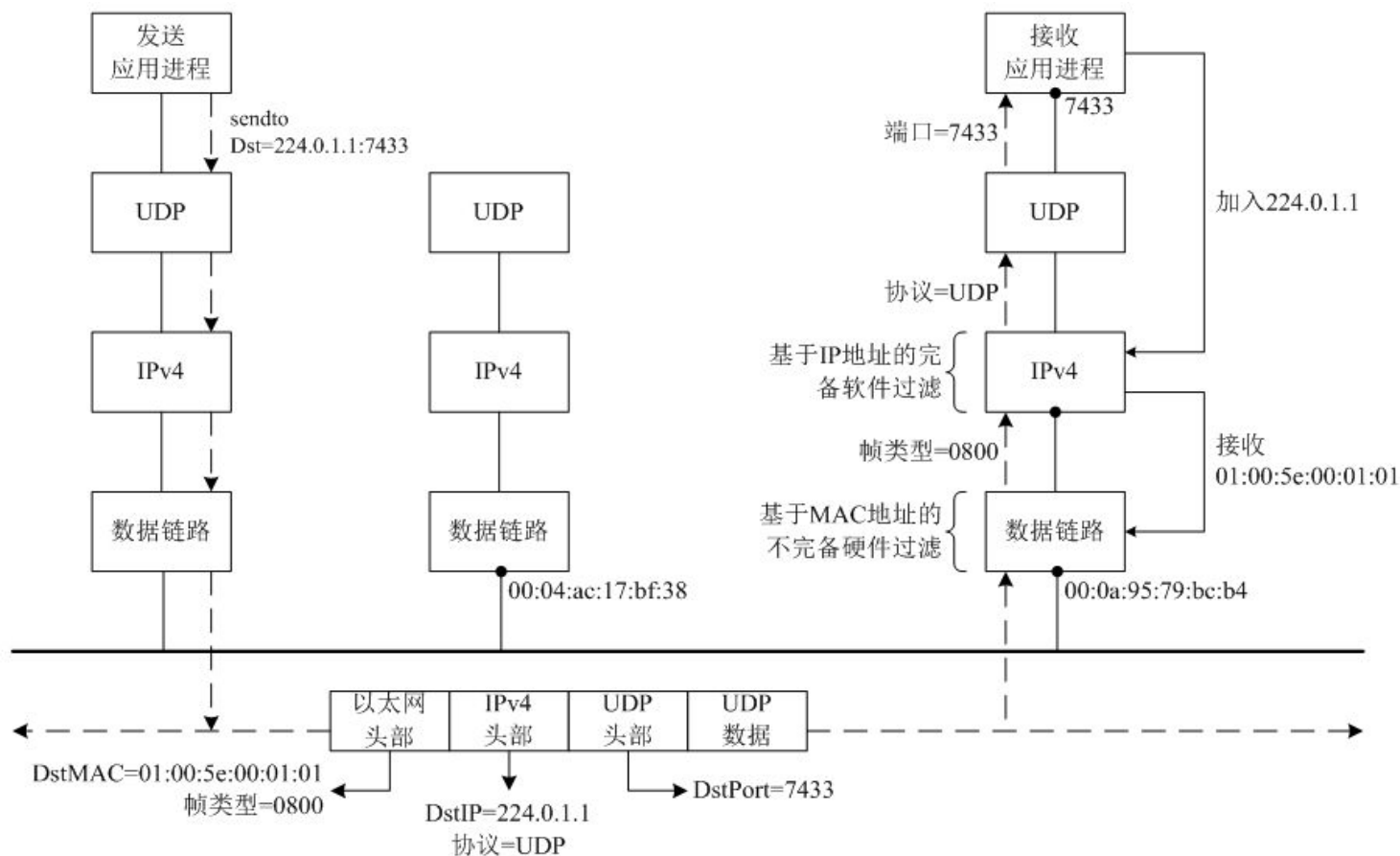
- 多播：数据的收发仅仅在同一分组中进行
- 多播的特点：
 - 多播地址标示一组接口
 - 多播可以用于广域网使用
 - 在IPv4中，多播是可选的

类型	IPv4	IPv6
单播	✓	✓
多播	可选	✓
广播	✓	✗
任播	✓，但未广泛用	✓

- IPv4的D类地址是多播地址
 - 十进制: 224. 0. 0. 1 → 239. 255. 255. 254
 - 十六进制: E0. 00. 00. 01 → EF. FF. FF. FE
- 多播地址向以太网MAC地址的映射



➤ UDP多播工作过程



- 在IPv4因特网域(AF_INET)中，多播地址结构体用如下结构体ip_mreq表示

```
1 struct in_addr
2 {
3     in_addr_t s_addr;
4 };
5 struct ip_mreq
6 {
7     struct in_addr imr_multiaddr; //多播组IP
8     struct in_addr imr_interface; //将要添加到多播组的IP
9 };
```

➤ 套接口选项

- `int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);`
- 成功执行返回0，否则返回-1

level	optname	说明	optval类型
IPPROTO_IP	IP_ADD_MEMBERSHIP	加入多播组	ip_mreq {}
	IP_DROP_MEMBERSHIP	离开多播组	ip_mreq {}

➤ 加入多播组示例

```
1 char group[INET_ADDRSTRLEN] = "224.0.1.1";  
2  
3 //定义一个多播组地址  
4 struct ip_mreq mreq;  
5  
6 //添加一个多播组IP  
7 mreq.imr_multiaddr.s_addr = inet_addr(group);  
8  
9 //添加一个将要添加到多播组的IP  
10 mreq.imr_interface.s_addr = htonl(INADDR_ANY);  
11  
12 setsockopt(sockfd, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(mreq));
```




值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

