## 一、概述

单播用于两个主机之间的端对端通信

广播用于一个主机对整个局域网上所有主机上的数据通信。

多播对一组特定的主机进行通信，而不是整个局域网上的所有主机

只有加入该多播组的主机才能接收到数据包。

## 二、多播地址

IP 多播地址在 IPv4 中它是一个 D 类 IP 地址，范围从 224.0.0.0 到 239.255.255.255，并

多播被划分为**局部链接多播地址、预留多播地址和管理权限多播地址**三类：

1）局部链接多播地址范围在 224.0.0.0~224.0.0.255，这是为路由协议和其它用途保留的地址，路由器并不转发属于此范围的IP包；

2）预留多播地址为 224.0.1.0~238.255.255.255，可用于全球范围（如Internet）或网络协议；

3）管理权限多播地址为 239.0.0.0~239.255.255.255，可供组织内部使用，类似于私有 IP 地址，不能用于 Internet，可限制多播范围。

一些多播组地址被 IANA 确定为知名地址，它们也被当作永久主机组，这和 TCP 及 UDP 中的知名端口相似。这些地址如下：

224.0.0.1    所有组播主机

224.0.0.2    所有组播路由器

224.0.0.4    DRMRP 路由器

224.0.0.5    所有 OSPF 的路由器

224.0.0.6    OSPF 指派路由器

224.0.0.9    RPIv2 路由器

224.0.0.10  EIGRP 路由器

224.0.0.13  PIM 路由器

224.0.0.22  IGMPv3

224.0.0.25  RGMP

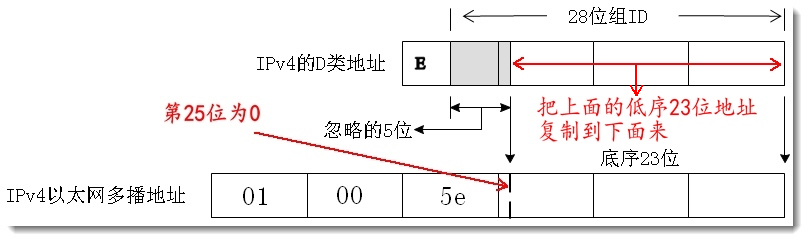
224.0.1.1    NTP 网络时间

## 三、以太网多播地址

多播地址属于网络层，链路层不知道多播地址，所有链路层需要以太网多播地址（其实就是MAC地址）

以太网多播地址组成由：01:00:5e+0bit（1位）+多播地址低23位

IEEE将01:00:5e+0bit（100000000010111100，25位）开头的MAC地址用作多播地址



不同的多播组号可以被映射为同一个以太网地址。例如，多播地址 224.128.64.32（十六进制 e0.80.40.20）和 224.0.64.32（十六进制 e0.00.40.20）都映射为同一以太网地址 01:00:5e:00:40:20。

## ****四、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

1）选项 IP\_MULTICAST\_LOOP

IP\_MULTICAST\_LOOP 用于控制数据是否回送到本地的回环接口。

例如：

unsigned int loop = 1;

// 参数 loop 设置为 0 禁止回送，设置为 1 允许回送。

setsockopt(sockfd, IPPROTO\_IP, IP\_MULTICAST\_LOOP, &loop, sizeof(loop));

2）选项 IP\_ADD\_MEMBERSHIP 和 IP\_DROP\_MEMBERSHIP

加入或者退出一个多播组，其需要结构 struct ip\_mreq 类型的变量进行控制，struct ip\_mreq 原型如下：

struct in\_addr

{

in\_addr\_t s\_addr;

}

struct ip\_mreq

{

struct in\_addr imn\_multiaddr; // 多播组 IP，类似于 QQ 群号

struct in\_addr imr\_interface;   // 将要添加到多播组的 IP，类似于QQ 成员号

};

**多播只能用 UDP 或原始 IP 实现，不能用 TCP。**

**加入多播实例：**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <string.h>

#include <errno.h>

#include <sys/types.h>

int main()

{

    int sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);  //建立套接字

    if (sockfd == -1)

    {

        perror("socket()");

        return -1;

    }

    // 初始化地址

    struct sockaddr\_in local\_addr;      // 本地地址

    memset(&local\_addr, 0, sizeof(local\_addr));

    local\_addr.sin\_family = AF\_INET;

    local\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

    local\_addr.sin\_port = htons(8000);

    // 绑定socket

    int err = bind(sockfd,(struct sockaddr\*)&local\_addr, sizeof(local\_addr));

    if(err < 0)

    {

        perror("bind()");

        return -2;

    }

    // 设置回环许可，控制数据允许回送到本地的回环接口

    int loop = 1;

    err = setsockopt(sockfd,IPPROTO\_IP, IP\_MULTICAST\_LOOP,&loop, sizeof(loop));

    if(err < 0)

    {

        perror("setsockopt():IP\_MULTICAST\_LOOP");

        return -3;

    }

    struct ip\_mreq mreq; // 多播地址结构体

    char group[16] = "224.0.0.88"; // 多播组 IP

    // 加入多播组，相当于创建一个QQ群，某人加入此群

    mreq.imr\_multiaddr.s\_addr = inet\_addr(group); // 多播地址，类似于 QQ 群号

    mreq.imr\_interface.s\_addr = htonl(INADDR\_ANY);// 将本机加入多播组，类似于某人加入此群

    // 加入多播组

    err = setsockopt(sockfd, IPPROTO\_IP, IP\_ADD\_MEMBERSHIP,&mreq, sizeof(mreq));

    if (err < 0)

    {

        perror("setsockopt():IP\_ADD\_MEMBERSHIP");

        return -4;

    }

    int times = 0;

    unsigned int addr\_len = 0;

    char buff[256] = {0};

    int n = 0;

    // 循环接收广播组的消息，1次后退出

    for(times = 0; times<1; times++)

    {

        addr\_len = sizeof(local\_addr);

        memset(buff, 0, sizeof(buff));

        // 接收数据

        n = recvfrom(sockfd, buff, sizeof(buff), 0,(struct sockaddr\*)&local\_addr, &addr\_len);

        if( n== -1)

        {

            perror("recvfrom()");

        }

        printf("Recv %dst message from server:%s\n", times, buff);

        sleep(2);

    }

    // 退出广播组

    err = setsockopt(sockfd, IPPROTO\_IP, IP\_DROP\_MEMBERSHIP,&mreq, sizeof(mreq));

    close(sockfd);

    return 0;

}

**向多播组发送信息的测试示例：**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <string.h>

#include <errno.h>

#include <sys/types.h>

int main()

{

    int sockfd; // 套接字文件描述符

    sockfd = socket(AF\_INET, SOCK\_DGRAM, 0); // 建立套接字

    if (sockfd == -1)

    {

        perror("socket()");

        return -1;

    }

    // 初始化目标 ip 信息

    struct sockaddr\_in dest\_addr; // 目标ip

    memset(&dest\_addr, 0, sizeof(dest\_addr));

    dest\_addr.sin\_family = AF\_INET;

    dest\_addr.sin\_addr.s\_addr = inet\_addr("224.0.0.88"); // 目的地址，为多播地址

    dest\_addr.sin\_port = htons(8000);   // 多播服务器的端口也是 8000

    // 向多播地址发送数据

    char buf[] = "BROADCAST TEST DATA";

    int n = sendto(sockfd, buf, strlen(buf), 0,(struct sockaddr\*)&dest\_addr, sizeof(dest\_addr));

    if( n < 0)

    {

        perror("sendto()");

        return -2;

    }

    return 0;

}