编制组播程序

一、实验目的

理解组播通信的概念及原理，比较组播通信方式和通常的单播通信方式有何不同。理解组播通信的实现，通过编制简单的利用组播方式通信的应用程序来加深对组播的理解。

二、实验原理

1 组播地址

组播不同于通常的单播，要实现组播需要使用 D 类地址，成为组播 IP 地址。在 IPv4 中，从 224.0.0.1 到 239.255.255.255 间的所有 IP 地址都属于 D 类地址。编制组播发送或接受程序时，必须使用以上指定范围的 IP 地址。

2 组播的工作过程

在局域网内，主机的网络接口将到目的主机的数据包发送到高层，这些数据包中的目的地址是物理接口地址或广播地址。

如果主机已经加入到一个组播组中，主机的网络接口就会识别出发送到该组成员的数据包。

因此，如果主机接口的物理地址为 80:C0:F6:A0:4A:B1，其加入的组播组为 224.0.1.10，则发送给主机的数据包中的目的地址必是下面三种类型之一：

接口地址：80:C0:F6:A0:4A:B1

广播地址：FF:FF:FF:FF:FF:FF:FF:FF

组播地址：01:00:5E:00:01:0A（组播地址如何映射到 MAC 层地址可以参见 RFC1112。

广域网中，路由器必须支持组播路由。当主机中运行的进程加入到某个组播组中时，主机向子网中的所有组播路由器发送 IGMP（Internet 分组管理协议）报文，告诉路由器凡是发送到这个组播组的组播报文都必须发送到本地的子网中，这样主机的进程就可以接收到报文了。子网中的路由器再通知其它的路由器，这些路由器就知道该将组播报文转发到哪些子网中去。

子网中的路由器也向 224.0.0.1 发送一个 IGMP 报文（224.0.0.1 代表组中的全部主机），要求组中的主机提供组的相关信息。组中的主机收到这个报文后，都各将计数器的值设为随机值，当计数器递减为 0 时再向路由器发送应答。这样就防止了组中所有的主机同时向路由器发送应答，造成网络拥塞。主机向组播地址发送一个报文做为对路由器的应答，组中的其它主机一旦看到这个应答报文，就不再发送应答报文了，因为组中的主机向路由器提供的都是相同的信息，所以子网路由器只需得到组中一个主机提供的信息就可以了。

如果组中的主机都退出了，路由器就收不到应答，因此路由器认为该组目前没有主机加入，遂停止到该子网报文的路由。IGMPv2 的解决方案是：组中的主机在退出时向 224.0.0.2 发送报文通知组播路由器。

3 应用编程接口（API）

如果你有套接字编程的经验，就会发现，对组播选项所进行的操作只需五个新的套接字操作。函数 setsockopt()及 getsockopt()用来建立和读取这五个选项的值。下表中列出了组

播的可选项，并列出其数据类型和描述：

IPv4 选项 数据类型 描 述

IP_ADD_MEMBERSHIP struct ip_mreq 加入到组播组中

IP_ROP_MEMBERSHIP struct ip_mreq 从组播组中退出

IP_MULTICAST_IF struct ip_mreq 指定提交组播报文的接口

IP_MULTICAST_TTL u_char 指定提交组播报文的 TTL

IP_MULTICAST_LOOP u_char 使组播报文环路有效或无效

在<linux/in.h>头文件中定义了 ip_mreq 结构：

```
struct ip_mreq {
  struct in_addr imr_multiaddr; /* IP multicast address of group */
  struct in_addr imr_interface; /* local IP address of interface */
  };
```

在头文件中组播选项的值为：

```
#define IP_MULTICAST_IF 32
#define IP_MULTICAST_TTL 33
#define IP_MULTICAST_LOOP 34
#define IP_ADD_MEMBERSHIP 35
#define IP_DROP_MEMBERSHIP 36
```

IP_ADD_MEMBERSHIP

若进程要加入到一个组播组中，用 soket 的 setsockopt()函数发送该选项。该选项类型是 ip_mreq 结构，它的第一个字段 imr_multiaddr 指定了组播组的地址，第二个字段 imr_interface 指定了接口的 IPv4 地址。

IP_DROP_MEMBERSHIP

该选项用来从某个组播组中退出。数据结构 ip_mreq 的使用方法与上面相同。

IP_MULTICAST_IF

该选项可以修改网络接口，在结构 ip_mreq 中定义新的接口。

IP_MULTICAST_TTL

设置组播报文的数据包的 TTL（生存时间）。默认值是 1，表示数据包只能在本地的子网中传送。

IP_MULTICAST_LOOP

组播组中的成员自己也会收到它向本组发送的报文。这个选项用于选择是否激活这种状态。

4 一个简单的组播通信例子

下面给出 Linux 下的一个简单的例子实现简单的组播通信：由一个进程向一个组播组发送报文，组播组中的相关进程接收报文，并将报文显示到屏幕上。

下面的代码实现了一个服务进程，它将标准输入接口输入的信息全部发送到组播组 224.0.1.1。你会发现，将信息发送到组播组不需要特别的操作，只要设置好组播组的目的地址就足够了。若在实验过程中，Loopback 和 TTL 这两个选项的默认值不适合应用程序，可以加以调整。

服务程序

将标准输入端口的输入发送到组播组 224.0.1.1。

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <arpa/inet.h>
#include <string.h>
#include <stdio.h>

#define MAXBUF 256
#define PUERTO 5000
#define GRUPO "224.0.1.1"

int main(void) {
   int s;
   struct sockaddr_in srv;
   char buf[MAXBUF];

   bzero(&srv, sizeof(srv));
   srv.sin_family = AF_INET;
   srv.sin_port = htons(PUERTO);
   if (inet_aton(GRUPO, &srv.sin_addr) < 0) {
    perror("inet_aton");
    return 1;
   }
   if ((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
    perror("socket");
    return 1;
   }

   while (fgets(buf, MAXBUF, stdin)) {
      if (sendto(s, buf, strlen(buf), 0,
                  (struct sockaddr *)&srv, sizeof(srv)) < 0) {
         perror("recvfrom");
      } else {
         fprintf(stdout, "Enviado a %s: %s
", GRUPO, buf);
      }
   }
}
```

客户端程序

下面的代码是客户端程序，它负责接收由服务程序发送到组播组中的信息，将收到的报文在标准输出设备中显示。程序中唯一与接收 UDP 报文过程不同是它设置了 IP_ADD_MEMBERSHIP 选项。

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <arpa/inet.h>
#include <stdio.h>

#define MAXBUF 256
#define PUERTO 5000
#define GRUPO "224.0.1.1"

int main(void) {
   int s, n, r;
   struct sockaddr_in srv, cli;
   struct ip_mreq mreq;
   char buf[MAXBUF];

   bzero(&srv, sizeof(srv));
   srv.sin_family = AF_INET;
   srv.sin_port = htons(PUERTO);
   if (inet_aton(GRUPO, &srv.sin_addr) < 0) {
      perror("inet_aton");
      return 1;
   }

   if ((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
      perror("socket");
      return 1;
   }

   if (bind(s, (struct sockaddr *)&srv, sizeof(srv)) < 0) {
      perror("bind");
      return 1;
   }

   if (inet_aton(GRUPO, &mreq.imr_multiaddr) < 0) {
      perror("inet_aton");
      return 1;
   }
   mreq.imr_interface.s_addr = htonl(INADDR_ANY);

   if (setsockopt(s,IPPROTO_IP,IP_ADD_MEMBERSHIP,&mreq,sizeof(mreq))
         < 0) {
      perror("setsockopt");
      return 1;
   }

   n = sizeof(cli);
```

```
    while (1) {
      if ((r = recvfrom(s, buf, MAXBUF, 0, (struct sockaddr *)
            &cli, &n)) < 0) {
        perror("recvfrom");
      } else {
        buf[r] = 0;
        fprintf(stdout, "Mensaje desde %s: %s
",
                  inet_ntoa(cli.sin_addr), buf);
      }
    }
}
```

三、实验要求

编写一个使用组播的应用程序。（2 人合作，分工要明确）（包括 1 个组播发送程序，和组播接收程序）。（新闻广播中心软件编写，实现服务器端输入最新的新闻信息，通过组播发送给接听的客户。）（此程序可以进一步扩展成为视频播放系统）。最后要给出程序详细的流程图和对程序的详细介绍、说明；指出自己设计的得意之处或有想法但没有完成的地方（若独到之处可以加分）。

四、思考题

1. 思考组播通信较之单播通信的优缺点，考虑可以使用组播通信的场合。
2. 考虑组播通信在发送和接受数据的安全性方面的问题。（如，任意用户可以向指定的组播地址发送干挠数据，没有授权的用户可以毫无限制的接受组播数据等）