实现广播，离不开广播地址，同一个子网（[局域网](http://baike.baidu.com/link?url=o-mzKpdjU2rkaEAPcvsUVDBl1Lp-y_TD3IyWox716eADVkOg0oVNNrJRFcuw6JXsgzMaUm5qBBwfUtU1OHSJQK" \t "_blank)）的所有主机网卡都会接收所在网段广播地址的数据包。**广播地址应用于局域网内的所有主机。**

[广播地址(Broadcast Address)](http://blog.csdn.net/tennysonsky/article/details/45564479)是专门用于同时向网络中（通常指同一子网）所有工作站进行发送的一个地址。

**广播地址主要分为两类**

1)受限广播

路由器从来不会转发受限广播的数据包，但同一个子网的所有主机都会接收到受限广播的数据包。

IP 地址的网络字段和主机字段全为 1 就是**受限广播地址255.255.255.255**。

2)直接广播（也叫定向广播）

直接广播可以被路由转发，发送到目标网络的所有主机，如：ip地址为 192.168.2.1的主机也可以发送广播到 192.168.1.0 这个网络。**当然不是所有的路由器，通常路由器是默认阻止直接广播的（可以设置不阻止）。**

IP 地址的网络字段定义这个网络，主机字段通常全为 1，**如192.168.10.0/24 的直接广播（定向广播）地址为：192.168.10.255**。

[广播地址的更详细介绍，请看《广播地址介绍》。](http://blog.csdn.net/tennysonsky/article/details/45564479)

**广播特点**

**对于一个带网卡设备的主机，它能接收到哪些网络数据包呢？**

1）网卡会接收目的 ip 和它的 ip 地址相同的数据包（至于能不能到应用层我们暂时不管，至于MAC地址如何确定我们暂时也不管），这个就是单播传输数据。

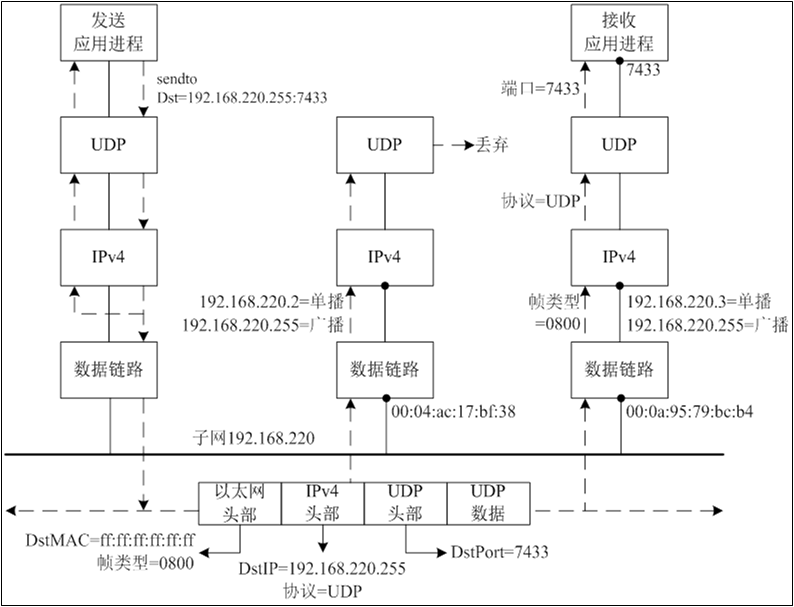
2）网卡会接收到目的 ip 为广播地址数据包，这个广播地址的 MAC 地址为：ff:ff:ff:ff:ff:ff 。

3）如果这个主机加入了多播组，它也会接收该多播组地址的数据包，[详情请看《多播》](http://blog.csdn.net/tennysonsky/article/details/44493407" \t "_blank)。

**广播只能用 UDP 或原始IP实现，不能用 TCP**。

UDP 广播特点如下：

* 处于同一子网的所有主机都必须处理数据。
* UDP 数据包会沿[协议栈](http://baike.baidu.com/link?url=rYY2yF5EAfijwGcqwndasNvWjosaibFLlVWDAOICr_bi486_VMm7mRo-AUxuBzeu_1YWFnQG5-F-iJUnLuDuua" \t "_blank)向上一直到 UDP 层，因为到 UDP 层，端口不匹配的话，数据才会丢弃，如下图，所以，运行音视频等较高速率工作的应用，会带来较大的负担。



**广播示例使用**

默认的情况下，不允许发送广播数据包，需要修改套接口选项：

int setsockopt( int sockfd,

int level,

int optname,

const void \*optval,

socklen\_t optlen);

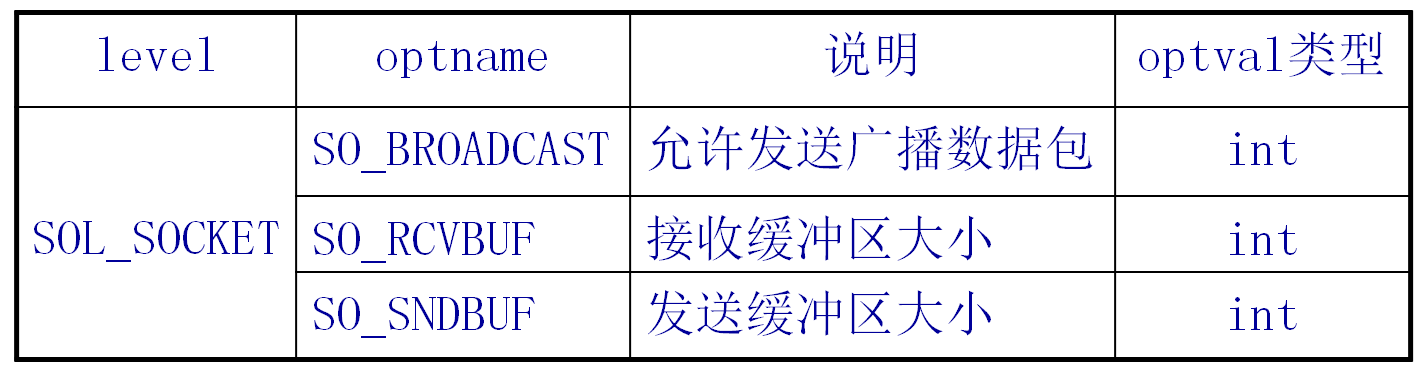
功能：

设置套接字选项

参数：

sockfd：套接字

其它参数说明如下：



注意：optval的值应该这样填，定义一个赋值为 1 的整型变量，int opt=1, 然后把这个变量的地址放这个位置，如：&opt

返回值：

成功执行返回0，否则返回-1

使用例子如下：

**[objc]** [view plaincopy](http://blog.csdn.net/tennysonsky/article/details/45563851)[在CODE上查看代码片](https://code.csdn.net/snippets/662208)

1. //设置该套接字为广播类型
2. **int** opt=1;
3. setsockopt(sockfd, SOL\_SOCKET, SO\_BROADCAST, &opt, **sizeof**(opt));

下面我们写一个 UDP 广播发送数据包例子，目的端口指定为 8000，目的 IP 为受限广播地址：255.255.255.255，那样，同一个局域网的所有主机网卡都会收到这个数据，但是，只有端口为 8000 的网络应用程序才会收到这个数据包：

**[objc]** [view plaincopy](http://blog.csdn.net/tennysonsky/article/details/45563851)[在CODE上查看代码片](https://code.csdn.net/snippets/662208)

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. #include <unistd.h>
5. #include <sys/socket.h>
6. #include <netinet/in.h>
7. #include <arpa/inet.h>
9. **int** main(**int** argc, **charchar** \*argv[])
10. {
11. unsigned **short** port = 8000;             // 端口
12. **charchar** \*server\_ip = "255.255.255.255";    // 受限广播地址
14. **int** sockfd;
15. sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);        //创建UDP套接字
16. **if**(sockfd < 0)
17. {
18. perror("socket");
19. exit(-1);
20. }
22. **struct** sockaddr\_in dest\_addr;
23. bzero(&dest\_addr, **sizeof**(dest\_addr));
24. dest\_addr.sin\_family = AF\_INET;
25. dest\_addr.sin\_port   = htons(port);
26. inet\_pton(AF\_INET, server\_ip, &dest\_addr.sin\_addr);
28. printf("send data to UDP server %s:%d!\n", server\_ip, port);
30. //设置该套接字为广播类型，这个很重要
31. **int** opt=1;
32. setsockopt(sockfd, SOL\_SOCKET, SO\_BROADCAST, &opt, **sizeof**(opt));
34. **char** send\_buf[512] = "this is a broadcast data";
35. sendto(sockfd, send\_buf, strlen(send\_buf), 0, (**struct** sockaddr\*)&dest\_addr, **sizeof**(dest\_addr));//发送数据
37. close(sockfd);
39. **return** 0;
40. }

[http://static.blog.csdn.net/images/save_snippets.png](javascript:;)