**Sending process:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <string.h>

#define err\_log(log) do{perror(log); exit(1);}while(0)

#define N 128

int main(int argc, const char \*argv[])

{

int sockfd;

struct sockaddr\_in broadcastaddr;

char buf[N] = {0};

if((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0)

{

err\_log("fail to socket");

}

broadcastaddr.sin\_family = AF\_INET;

broadcastaddr.sin\_addr.s\_addr = inet\_addr("192.168.1.255"); //Broadcast address

broadcastaddr.sin\_port = htons(10000);

int optval = 1;

if(setsockopt(sockfd, SOL\_SOCKET, SO\_BROADCAST, &optval, sizeof(int)) < 0)

{

err\_log("fail to setsockopt");

}

while(1)

{

printf("Input > ");

fgets(buf, N, stdin);

if(sendto(sockfd,buf, N, 0, (struct sockaddr \*)&broadcastaddr, sizeof(broadcastaddr)) < 0)

{

err\_log("fail to sendto");

}

}

return 0;

}

**Receiving process:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <string.h>

#define err\_log(log) do{perror(log); exit(1);}while(0)

#define N 128

int main(int argc, const char \*argv[])

{

int sockfd;

char buf[N];

struct sockaddr\_in broadcastaddr, srcaddr;

if((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0)

{

err\_log("fail to socket");

}

broadcastaddr.sin\_family = AF\_INET;

broadcastaddr.sin\_addr.s\_addr = inet\_addr("I**NADDR\_ANY** or 192.168.1.255");

//Broadcast address/ I**NADDR\_ANY**

broadcastaddr.sin\_port = htons(10000);

if(bind(sockfd, (struct sockaddr\*)&broadcastaddr, sizeof(broadcastaddr)) < 0)

{

err\_log("fail to bind");

}

socklen\_t addrlen = sizeof(struct sockaddr);

while(1)

{

if(recvfrom(sockfd,buf, N, 0, (struct sockaddr \*)&srcaddr, &addrlen) < 0)

{

err\_log("fail to sendto");

}

printf("buf:%s ---> %s %d\n", buf, inet\_ntoa(srcaddr.sin\_addr), ntohs(srcaddr.sin\_port));

}

return 0;

}