```
#include <sys/types.h>
#include <sys/time.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <netdb.h>
#include <string.h>
#include <netinet/ip.h>
#include <arpa/inet.h>
#include <unistd.h>

void imprimeTrama(unsigned char *paq, int len){
      for(int i = 0; i < len; i++){
            if((i%16) == 0){
                  printf("\n");
            }
            printf("%.2x ", paq[i]);
      }
      printf("\n");
}

int main(int argc, char const *argv[]){
   unsigned char dhcp_discover[343], dhcp_offer[274], dhcp_ack[280], opc_ack[1] = {0x03};
   int ds, ds2, tamrecv, tamsend, on = 1;
   socklen_t clilen, serverlen;
   struct sockaddr_in server_addr, client_addr;

   ds  = socket(AF_INET, SOCK_DGRAM, 0);

   if((setsockopt(ds, SOL_SOCKET, SO_BROADCAST, &on, sizeof(on))) < 0){
     perror("\nsetsockopt");
   }

     server_addr.sin_family = AF_INET;
   server_addr.sin_addr.s_addr = INADDR_BROADCAST;
   server_addr.sin_port = htons(68);

   client_addr.sin_family = AF_INET;
   client_addr.sin_addr.s_addr = INADDR_ANY;
   client_addr.sin_port = htons(67);
   if((bind(ds, (struct sockaddr *)&client_addr,sizeof(client_addr))) < 0){
     perror("\nError en Bind Cliente ");
   }else{
     printf("Exito en Bind Cliente\n");
           clilen = sizeof(client_addr);
         while(1){
           tamrecv = recvfrom(ds, dhcp_discover, sizeof(dhcp_discover), 0, (struct sockaddr
*)&client_addr, &clilen);
              if(tamrecv < 0){
```

```
        perror("No envia DHCPDISCOVER");
    }else{
        printf("Exito al recibir DHCPDISCOVER\n");
        memcpy(dhcp_offer, dhcp_discover, 235);
        //Offer
        dhcp_offer[0] = 2;
    dhcp_offer[8] = 0;
        dhcp_offer[9] = 0;

        dhcp_offer[16] = 192;
        dhcp_offer[17] = 168;
        dhcp_offer[18] = 1;
        dhcp_offer[19] = 10;

        dhcp_offer[20] = 192;
        dhcp_offer[21] = 168;
        dhcp_offer[22] = 1;
        dhcp_offer[23] = 100;

        dhcp_offer[24] = 192;
        dhcp_offer[25] = 168;
        dhcp_offer[26] = 1;
        dhcp_offer[27] = 1;

        dhcp_offer[236] = 99;
        dhcp_offer[237] = 130;
        dhcp_offer[238] = 83;
        dhcp_offer[239] = 99;
        //OPTION DHCPOFFER
        dhcp_offer[240] = 53;
        dhcp_offer[241] = 1;
        dhcp_offer[242] = 2;
        //OPCION SUBNET MASK
        dhcp_offer[243] = 1;
        dhcp_offer[244] = 4;
        dhcp_offer[245] = 0xff;
        dhcp_offer[246] = 0xff;
        dhcp_offer[247] = 0xff;
        dhcp_offer[248] = 0;
        //OPTION ROUTER
        dhcp_offer[249] = 3;
        dhcp_offer[250] = 4;
        dhcp_offer[251] = 192;
        dhcp_offer[252] = 168;
        dhcp_offer[253] = 1;
        dhcp_offer[254] = 1;
        //OPTION DNS
        dhcp_offer[255] = 6;
        dhcp_offer[256] = 4;
```

```
                dhcp_offer[257] = 148;
                dhcp_offer[258] = 204;
                dhcp_offer[259] = 103;
                dhcp_offer[260] = 2;
                //OPT BROADCAST
                dhcp_offer[261] = 28;
                dhcp_offer[262] = 4;
                dhcp_offer[263] = 192;
                dhcp_offer[264] = 168;
                dhcp_offer[265] = 1;
                dhcp_offer[266] = 255;
                //OPTION ID SERVIDOR
                dhcp_offer[267] = 0x36;
                dhcp_offer[268] = 4;
                dhcp_offer[269] = 192;
                dhcp_offer[270] = 168;
                dhcp_offer[271] = 1;
                dhcp_offer[272] = 100;

                dhcp_offer[273] = 0xff;

                tamsend = sendto(ds, dhcp_offer, 274, 0, (struct sockaddr *)&server_addr, clilen);
                if(tamsend < 0){
                perror("\nError en sendto");
            }else{
                printf("Exito al enviar DHCPOFFER\n");
                //Request
                tamrecv = recvfrom(ds, dhcp_discover, sizeof(dhcp_discover), 0, (struct sockaddr
*)&client_addr, &clilen);
                if(tamrecv < 0){
                    perror("\nError al recibir DHCPREQUEST");
                }else{
                    printf("Exito al recibir DHCPREQUEST\n");
                    memcpy(dhcp_ack, dhcp_discover, 235);
                    //ACK
                    dhcp_ack[0] = 2;
                    dhcp_ack[8] = 0;
                        dhcp_ack[9] = 0;

                        dhcp_ack[16] = 192;
                        dhcp_ack[17] = 168;
                        dhcp_ack[18] = 1;
                        dhcp_ack[19] = 10;

                        dhcp_ack[20] = 192;
                        dhcp_ack[21] = 168;
                        dhcp_ack[22] = 1;
                        dhcp_ack[23] = 100;
```

```
dhcp_ack[24] = 192;
dhcp_ack[25] = 168;
dhcp_ack[26] = 1;
dhcp_ack[27] = 1;

dhcp_ack[236] = 99;
dhcp_ack[237] = 130;
dhcp_ack[238] = 83;
dhcp_ack[239] = 99;

dhcp_ack[240] = 53;
dhcp_ack[241] = 1;
dhcp_ack[242] = 5;
//OPTION TIME
dhcp_ack[243] = 51;
dhcp_ack[244] = 4;
dhcp_ack[245] = 0;
dhcp_ack[246] = 0;
dhcp_ack[247] = 0;
dhcp_ack[248] = 60;
//OPCION SUBNET MASK
dhcp_ack[249] = 1;
dhcp_ack[250] = 4;
dhcp_ack[251] = 0xff;
dhcp_ack[252] = 0xff;
dhcp_ack[253] = 0xff;
dhcp_ack[254] = 0;
//OPTION ROUTER
dhcp_ack[255] = 3;
dhcp_ack[256] = 4;
dhcp_ack[257] = 192;
dhcp_ack[258] = 168;
dhcp_ack[259] = 1;
dhcp_ack[260] = 1;
//OPTION DNS
dhcp_ack[261] = 6;
dhcp_ack[262] = 4;
dhcp_ack[263] = 148;
dhcp_ack[264] = 204;
dhcp_ack[265] = 103;
dhcp_ack[266] = 2;
//OPT BROADCAST
dhcp_ack[267] = 28;
dhcp_ack[268] = 4;
dhcp_ack[269] = 192;
dhcp_ack[270] = 168;
dhcp_ack[271] = 1;
dhcp_ack[272] = 255;
//OPT BROADCAST
```

```
                                dhcp_ack[273] = 0x36;
                                dhcp_ack[274] = 4;
                                dhcp_ack[275] = 192;
                                dhcp_ack[276] = 168;
                                dhcp_ack[277] = 1;
                                dhcp_ack[278] = 100;

                                dhcp_ack[279] = 0xff;
                                if(!memcmp(dhcp_discover+242, opc_ack, 1)){
                                    tamsend = sendto(ds, dhcp_ack, 280, 0, (struct sockaddr *)&server_addr,
clilen);
                                    if(tamsend < 0){
                                        perror("\nError al enviar DHCPACK");
                                    }else{
                                        printf("Exito al enviar DHCPACK\n");
                                    }
                                }
                            }
                        }
                        }
                    }
                }
                close(ds);
            return 0;
}
```