```c
/*
** c_srvudp_echo.c -- Servidor de echo UDP usando connect()
*/

#include <stdio.h>
#include <signal.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MYPORT 4950     // the port users will be connecting to

#define MAXBUFLEN 1000

void alarme();

int sockfd, numbytes, inlines = 0, inchars = 0;
struct sockaddr_in their_addr; // connector’s address information
socklen_t addr_len;

void alarme() {
  /* Passou um segundo sem receber nada, sendo que ja tinha recebido algo antes.
     Vamos imprimir as estatisticas até agora e resetar a contagem. */

  fprintf(stderr, "Linhas recebidas: %d\nCaracteres recebidos: %d\n", inlines, inchars);
  inlines = 0; inchars = 0;
  fflush(stdout);

  // UDP (Dis)Connect
  their_addr.sin_family = AF_UNSPEC;
  connect( sockfd, (struct sockaddr *)&their_addr, addr_len );
}

void quit() {

  // UDP (Dis)Connect
  their_addr.sin_family = AF_UNSPEC;
  connect( sockfd, (struct sockaddr *)&their_addr, addr_len );

  close(sockfd);
  fprintf(stderr, "\rSinal capturado, saindo.\n");
  exit(0);
}

int main(void) {

  struct sockaddr_in my_addr;   // my address information


  char buf[MAXBUFLEN];          /* Buffer para receber msgs */

  if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1)
    {
      perror("socket");
      exit(1);
    }

  my_addr.sin_family = AF_INET;          // host byte order
  my_addr.sin_port = htons(MYPORT);      // short, network byte order
  my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
  memset(&(my_addr.sin_zero), ’\0’, 8); // zero the rest of the struct

  if (bind(sockfd, (struct sockaddr *)&my_addr,
           sizeof(struct sockaddr)) == -1)
    {
      perror("bind");
      exit(1);
    }

  addr_len = (socklen_t)sizeof(struct sockaddr);

  signal(SIGINT, quit);
  signal(SIGALRM, alarme);      /* Associando o sinal ao handler. */

  while(1)
    {                   /* Loop principal */

      //Get first message/IP from client
      if( recvfrom(sockfd, buf, MAXBUFLEN−1 , 0, (struct sockaddr *)&their_addr, &addr_len)!= −1
)
        {
          printf("%s", buf);
          // UDP Connect
          if( ( connect( sockfd, (struct sockaddr *)&their_addr, addr_len ) ) == −1 )
            {
              perror("connect");
              exit(1);
            }
        }
      else
        {
          perror("recvfrom");
          exit(1);
        }

      alarm(1);
        if ((numbytes = send(sockfd, buf, strlen(buf), 0)) == −1)
        {
          perror("send");
          exit(1);
        }


      /* printf("Connected\n"); */

      inlines = 1; inchars = numbytes;
      while((numbytes = recv(sockfd, buf, MAXBUFLEN−1 , 0)) > 0)
        {
          if (numbytes == −1)
            {
              perror("recvfrom");
              exit(1);
            }
          inchars += numbytes;
          inlines++;
          buf[numbytes] = ’\0’;
          printf("%s", buf);
          if ((numbytes = send(sockfd, buf, strlen(buf), 0)) == −1)
            {
              perror("send");
              exit(1);
            }
          alarm(1);                     /* Se nao receber datagrama com 0 bytes em até um segundo
                                           ele desperta, e chama a funcao alarme() */
        }

      fprintf(stderr, "Linhas recebidas: %d\nCaracteres recebidos: %d\n", inlines, inchars);
      alarm(0);                 /* Terminou com um cliente, desliga o alarme. */
      fflush(stdout);           /* Forçar a saída */
    }

  return 0;
}
```