Lycée La Fayette
Champagne-sur-Seine • Fontaineroux

Informatique et Réseaux

Rapport de Travaux Pratiques
BTS SN-IR

PNET - TP SOCK MultiCast

Flavian LAXENAIRE
22 octobre 2021

Forme / 20 [coef. 1] →		Fond / 20 [coef. 2] →		Note / 20	
Qualité du rapport		Méthodologie			
Expression écrite		Respect du Cahier des Charges			
Pertinence de la rédaction		Qualité technique			
Respect des standards de codage		État d'avancement		Malus	

Sommaire

1. Codage du serveur

```
#define MAXBUF 256
#define PORT 5000
#define GROUP "224.0.0.224" //Associer ici une adresse IP autorisée pour du multicast
int main(void)
{
  int s,r ;
  struct sockaddr_in srv; //structure serveur
  char buf[MAXBUF];

  bzero(&srv, sizeof(srv));

  /* TODO creation du serveur sur port 5000 */
  srv.sin_family = PF_INET;
  srv.sin_port = htons(PORT);

  /* TODO renseigne l'adresse du groupe de multicast */
  if (inet_aton(GROUP, &srv.sin_addr) < 0)
  {
    perror("inet_aton");
    return 1;
  }

  /* TODO creation de la socket en DGRAM */
  if ((s = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0)
  {
    perror("socket");
    return 1;
  }
  printf("N socket serveur %d\n", s);

  /* TO DO boucle de lecture clavier et envoi sur la socket de multicast */
  for (;;)
  {
    int taille = sizeof(srv);
    for (int i = 0; i < sizeof(buf); *(buf + i++) = '\0')  //Mise à 0 du buf
      ;
    int recu = read(0, buf, sizeof(buf)); //Lecture sur le clavier
    buf[recu] = 0; //Mise en place du terminateur
    if (recu) //si quelque chose recu
    {
      r = sendto(s, buf, strlen(buf), 0, (struct sockaddr *)&srv, taille); //Envoi sur le client
    }
    printf("envoyé : %d\n", r);
  }
}
```

2. Codage du client

```
#define MAXBUF 256
#define PORT 5000
int main(int argc, char *argv[])
{
  bzero(&srv, sizeof(srv));
  srv.sin_family = PF_INET;
  srv.sin_port = htons(PORT); (port 5000)
  if (argc == 2)
  {
    sprintf(GROUP, "%s", argv[1]); // l'ip multicast est prise en argument sur la ligne de commande
  }
  else
  {
    fprintf(stderr, "Usage:  %s <IP Multicast>\n", argv[0]); //erreur si l'adresse n'est pas sur la ligne de commande
    exit(1);
  }

  if ((inet_aton(GROUP, &srv.sin_addr)) < 0)
  {
    perror("inet_aton");
    return 1;
  }

  /* TO DO creation socket MODE DGRAM */
  if ((s = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0) //creation de la socket en UDP
  {
    perror("socket");
    return 1;
  }
  printf("N socket cliente %d\n", s);

  /* Attachement socket à la structure locale */
  if (bind(s, (struct sockaddr *)&srv, sizeof(srv)) < 0)
  {
    perror("bind");
    return 1;
  }
  mreq.imr_multiaddr.s_addr = inet_addr(GROUP);
  mreq.imr_interface.s_addr = htonl(INADDR_ANY); /* toute adresse possible */
  if (setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(mreq)) < 0)
  {
    perror("setsockopt");
    return 1;
  }
  socklen_t n = sizeof(remote); // taille du distant => taille du "serveur multicast "
                                /* TO DO boucle de reception et affichage !! */
  while (1)
  {
    r = recvfrom(s, buf, sizeof(buf), 0, (struct sockaddr *)&srv, &n); //Client reçoit du serveur
    buf[r] = 0; //Mise en place du terminateur
    printf("reçu : %s", buf);
  }
}
```

Des tests ont été réalisé sur deux ordinateur lié au même réseau. Voici une capture d'écran d'un test sur le même ordinateur.