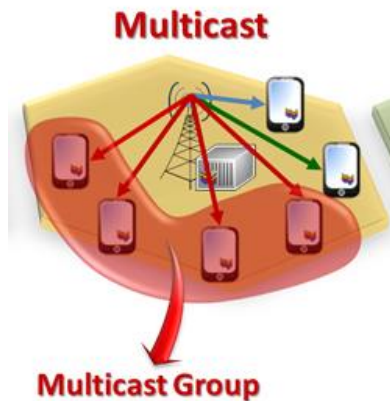


I. Pengenalan Aplikasi Chatting Multicast (Multithread)



Hampir serupa dengan unicast, transmisi jaringan atau aliran data dengan sistem multicast dapat dilakukan dengan lebih dari dua perangkat sistem atau komputer. Multicast adalah transmisi jaringan one-to-many.

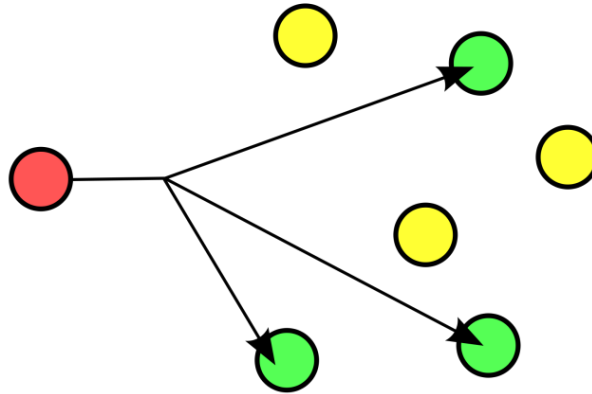
Dengan menggunakan transmisi jaringan multicast dimaksudkan agar dapat mengirim data dan informasi kepada banyak komputer atau sistem tujuan, dengan tidak semua sebagai host. Multicast diperlukan dalam kondisi-kondisi tertentu, seperti ketika beberapa komputer dalam grup ingin menerima transmisi tersebut.

Salah satu contohnya adalah streaming video atau audio. Misalkan banyak komputer ingin terhubung ke komputer host dan menerima transmisi data video atau audio tersebut secara bersamaan. Jika transmisi dilakukan secara individu dan data informasi dikirim secara satu persatu, maka akan banyak aliran data dan waktu yang diperlukan. Sedangkan jika menggunakan transmisi broadcast, maka proses tidak lagi diperlukan sehingga komputer host dan komputer tujuan tidak lagi dapat berkomunikasi melalui jaringan tersebut.

Oleh karena itu, dengan menggunakan transmisi multicast maka video dan audio hanya dikirim sekali akan tetapi diterima oleh banyak sistem dan komputer.

Transmisi multicast analoginya ketika pemberi undangan ingin memberikan informasi kepada teman sekolahnya. Pemberi sebagai host mengirimkan undangannya secara langsung di kelasnya, sehingga informasi tersebut hanya dikirimkan satu kali dan diterima oleh semua teman dikelasnya. Melalui proses transmisi ini pemberi dapat mengetahui respon dari penerima.

II. Cara Kerja Aplikasi Chatting Multicast (Multithread)



Aplikasi chatting multicast dengan multithreading adalah aplikasi yang memungkinkan pengguna untuk mengirim pesan kepada sekelompok orang secara bersamaan, menggunakan konsep multithreading untuk memfasilitasi pengiriman pesan yang efisien.

Dalam konteks aplikasi chatting multicast dengan multithreading, pengguna dapat membuat grup atau kanal dan mengirim pesan kepada semua anggota grup secara bersamaan. Aplikasi ini menggunakan multithreading, yaitu teknik yang memungkinkan pemrosesan pesan secara paralel atau simultan, untuk mengirim pesan kepada semua anggota grup dalam waktu yang singkat.

Aplikasi chatting multicast dengan multithreading bekerja dengan memanfaatkan konsep multicast dan multithreading dalam pengiriman pesan kepada sekelompok pengguna secara bersamaan. Berikut adalah gambaran umum tentang cara kerja aplikasi tersebut:

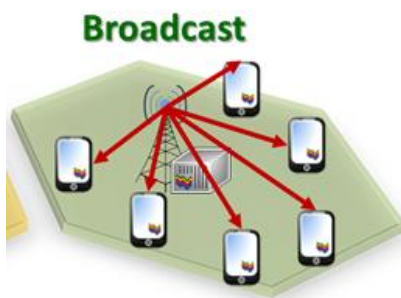
1. Pembuatan grup atau kanal: Pengguna dapat membuat grup atau kanal dan mengundang anggota-anggota lain untuk bergabung. Setiap grup memiliki identitas unik yang membedakan satu grup dari yang lain.
2. Pengiriman pesan: Ketika seorang pengguna mengirim pesan ke dalam grup, pesan tersebut dikirim ke server aplikasi.
3. Multithreading: Di sisi server, aplikasi menggunakan multithreading untuk mengelola pengiriman pesan multicast secara paralel atau simultan. Setiap thread (benang) yang terbentuk akan bertanggung jawab mengirimkan pesan ke sekelompok pengguna dalam grup tersebut.
4. Pengiriman pesan multicast: Setiap thread akan mengirim pesan ke anggota grup yang dituju menggunakan protokol multicast. Protokol ini memungkinkan pengiriman pesan ke beberapa penerima secara bersamaan dalam satu operasi pengiriman.

5. Penerimaan pesan: Setiap anggota grup yang terhubung dengan aplikasi akan menerima pesan tersebut. Pesan tersebut akan ditampilkan dalam antarmuka aplikasi chat mereka secara real-time.
6. Notifikasi: Anggota grup yang terhubung ke aplikasi biasanya akan menerima notifikasi tentang adanya pesan baru dalam grup yang mereka ikuti. Notifikasi ini membantu pengguna tetap up-to-date dengan percakapan dalam grup.

Dengan menggunakan multithreading, aplikasi dapat mengirim pesan multicast dengan efisien dan cepat kepada sekelompok pengguna dalam grup. Teknik multithreading memungkinkan pengiriman pesan yang simultan, yang dapat meningkatkan responsivitas dan mengurangi waktu pengiriman pesan.

Multicast: Pengiriman data dari 1 pengirim ke beberapa penerima. untuk melakukan multicast dalam jaringan komputer, kita menggunakan Multicast Address dengan range 224.0.0.0 – 239.255.255.255

III. Pengenalan Aplikasi Chatting Broadcast (Multithread)



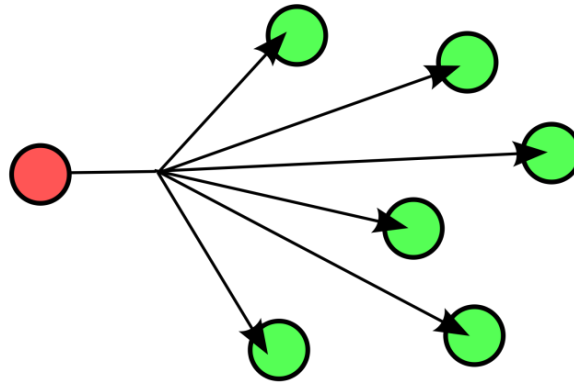
Broadcast merupakan transmisi jaringan one-to-all. Transmisi jaringan dengan sistem broadcast dapat dilakukan dengan banyak sekali penerima atau perangkat sistem tujuan. Akan tetapi sistem transmisi broadcast cenderung membuang resource.

Dalam jaringan LAN, komputer asal sebagai host mengirimkan data informasi kepada sembarang komputer lain yang terhubung di dalam jaringan. Transmisi jaringan broadcast memiliki kekurangan yaitu sistem asal yang menjadi host tidak akan mengetahui bagaimana respon dari sistem tujuan apakah data diterima dengan baik atau data terjadi corrupt. Contoh penggunaan broadcast adalah transmisi saluran data pada televisi.

Transmisi jaringan broadcast memiliki analogi seperti seorang pemberi undangan yang ingin mengirimkan informasi kepada orang lain. Pemberi undangan sebagai host membagikan undangan dengan menyebarnya di jalan-jalan. Oleh karena itu, perangkat host tidak akan mengetahui bahwa informasi yang dikirimkannya telah

sampai dan respon penerima pun tidak akan diketahui oleh perangkat sistem yang menjadi host.

IV. Cara Kerja Aplikasi Chatting Broadcast (Multithread)



Aplikasi chatting broadcast dengan multithreading adalah aplikasi yang memungkinkan pengguna untuk mengirim pesan kepada seluruh pengguna yang terhubung dalam jaringan atau grup secara bersamaan. Istilah "broadcast" merujuk pada pengiriman pesan kepada semua penerima yang ada.

Dalam konteks aplikasi chatting broadcast dengan multithreading, setiap pesan yang dikirim oleh pengguna akan dikirim secara langsung kepada semua pengguna yang terhubung. Aplikasi ini menggunakan multithreading, yaitu teknik yang memungkinkan pemrosesan pesan secara paralel atau simultan, untuk mengirim pesan kepada semua penerima dalam waktu yang singkat.

Aplikasi chatting broadcast dengan multithreading bekerja dengan memanfaatkan konsep broadcast dan multithreading dalam pengiriman pesan kepada seluruh pengguna yang terhubung. Berikut adalah gambaran umum tentang cara kerja aplikasi tersebut:

1. Pengiriman pesan broadcast: Ketika seorang pengguna mengirim pesan broadcast, pesan tersebut dikirim ke server aplikasi.
2. Multithreading: Di sisi server, aplikasi menggunakan multithreading untuk mengelola pengiriman pesan broadcast secara paralel atau simultan. Setiap thread (benang) yang terbentuk akan bertanggung jawab mengirimkan pesan kepada semua pengguna yang terhubung dalam jaringan atau grup.
3. Pengiriman pesan: Setiap thread akan mengirim pesan ke pengguna-pengguna yang terhubung menggunakan protokol broadcast. Protokol ini memungkinkan pengiriman pesan kepada semua penerima secara bersamaan dalam satu operasi pengiriman.

4. Penerimaan pesan: Setiap pengguna yang terhubung ke aplikasi akan menerima pesan tersebut. Pesan akan ditampilkan dalam antarmuka aplikasi chat mereka secara real-time.
5. Notifikasi: Pengguna yang terhubung ke aplikasi biasanya akan menerima notifikasi tentang adanya pesan baru yang dikirim sebagai broadcast. Notifikasi ini membantu pengguna tetap up-to-date dengan pesan-pesan yang masuk.

Dengan menggunakan multithreading, aplikasi dapat mengirim pesan broadcast dengan efisien dan cepat kepada seluruh pengguna yang terhubung dalam jaringan atau grup. Teknik multithreading memungkinkan pengiriman pesan yang simultan, yang dapat meningkatkan responsivitas dan mengurangi waktu pengiriman pesan.

Broadcast: Pengiriman data dari 1 pengirim ke seluruh penerima. untuk melakukan broadcast ke seluruh penerima, kita menggunakan broadcast address yang merupakan IP terakhir dari sebuah network address, atau disebut juga dengan last IP address.

V. Pembuatan Simple Aplikasi Chatting Multicast (Multithread)

Berikut adalah contoh sederhana program Pembuatan Aplikasi Chatting Multicast (Multithread) menggunakan Python:

sender.py :

```
# SENDER

import socket

group = '224.1.1.1'
port = 5004

# 2-hop restriction in network
ttl = 2

sock = socket.socket(socket.AF_INET,
                     socket.SOCK_DGRAM,
                     socket.IPPROTO_UDP)
sock.setsockopt(socket.IPPROTO_IP,
               socket.IP_MULTICAST_TTL,
               ttl)

while True:
    pesan = input("masukkan pesan :")
    sock.sendto(pesan.encode('utf-8'), (group, port))
```

receiver1.py :

```
# RECEIVER

import socket
import struct

MCAST_GRP = '224.1.1.1'
MCAST_PORT = 5004

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind(('', MCAST_PORT))
mreq = struct.pack("4sl", socket.inet_aton(MCAST_GRP), socket.INADDR_ANY)

sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)

while True:
    print(sock.recv(1024))
```

receiver2.py :

```
# RECEIVER

import socket
import struct

MCAST_GRP = '224.1.1.1'
MCAST_PORT = 5004

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind(('', MCAST_PORT))
mreq = struct.pack("4sl", socket.inet_aton(MCAST_GRP), socket.INADDR_ANY)

sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)

while True:
    print(sock.recv(1024))
```

VI. Pembuatan Simple Aplikasi Chatting Broadcast (Multithread)

Berikut adalah contoh sederhana program Pembuatan Aplikasi Chatting Broadcast (Multithread) menggunakan Python:

Server:

```

import socket

s = socket.socket()
host = socket.gethostname()
port = 8080
s.bind((host, port))
s.listen(1)

print("Proses Dua Koneksi")

conn1, addr1 = s.accept()
print("Client_1 terkoneksi")
conn1.send("Selamat datang di Server...".encode())
conn2, addr2 = s.accept()
print("Client_2 terkoneksi")
conn2.send("Selamat datang di Server...".encode())

while 1:
    message = input("Masukan Pesan : ")
    message = str(message).encode()
    conn1.send(message)
    conn2.send(message)
    print("Pesan Terkirim")
    recv_message = conn1.recv(1024)
    print("Client_1 : ", recv_message.decode())
    conn2.send(recv_message)
    recv_message = conn2.recv(1024)
    print("Client_2 : ", recv_message.decode())
    conn1.send(recv_message)

```

Client 1 :

```

import socket

s = socket.socket()
host = socket.gethostname()
port = 8080
s.connect((host, port))

print("Menyambungkan ke Server")

message = s.recv(1024)
message = message.decode()
print("Pesan dari server : ", message)

while 1:
    message = s.recv(1024)
    message = message.decode()

```

```
print("Server : ", message)
new_message = input("Masukkan Pesan : ")
new_message = str(new_message).encode()
s.send(new_message)
print("Pesan Terkirim")
message = s.recv(1024)
message = message.decode()
print("Client_2 : ", message)
```

Client 2 :

```
import socket

s = socket.socket()
host = socket.gethostname()
port = 8080
s.connect((host, port))

print("Menyambungkan ke Server")

message = s.recv(1024)
message = message.decode()
print("Pesan dari server : ", message)

while 1:
    message = s.recv(1024)
    message = message.decode()
    print("Server : ", message)
    message = s.recv(1024)
    message = message.decode()
    print("Client_1 : ", message)
    new_message = input("Masukkan Pesan : ")
    new_message = str(new_message).encode()
    s.send(new_message)
    print("Pesan Terkirim")
```

VII. Tugas dan Latihan

1. Membuat laporan percobaan praktikum dan beri Analisa Hasil Percobaan tadi yang sudah dibuat Aplikasi Chatting Multicast & Broadcast (Multithread)
2. Tugas tambahan, proses 4 koneksi dengan tambahkan client menjadi 4 client