



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Departamento de Ciencias e Ingeniería
de la Computación



Aplicaciones para Comunicaciones en Red

Practica 03 “Aplicación Chat”

Profesor: Axel Ernesto Moreno Cervantes

Grupo: 6CM1

Integrantes:

Ruvalcaba Flores Martha Catalina

Sandoval Hernández Eduardo

Fecha de entrega: 09/12/2022

Objetivo

El estudiante implementará un servicio de chat haciendo uso de sockets de datagrama, así como sockets de multidifusión.

Planteamiento del problema

Muchas empresas hacen uso de Internet para ofrecer sus productos y servicios, ya que a través de este medio pueden estar en contacto directo con los clientes potenciales. Algunas de las principales ventajas de Internet como medio para realizar negocios son la cobertura a nivel global, así como la disponibilidad del servicio los 365 días, las 24 hrs. del día. Sin embargo, estas ventajas se convierten también en un reto para las empresas, pues se debe proporcionar a los clientes una vía de comunicación directa para resolver las dudas de los clientes potenciales, atender sus peticiones, así como proporcionarles información adicional a la expuesta en sus portales. Existen diversos recursos que pueden ser utilizados, como los foros o el correo electrónico, pero ninguno de ellos permite la comunicación en tiempo real. El chat es un excelente recurso para este tipo de propósito, ya que además de ser una vía de comunicación en tiempo real, permite comunicar a dos o más usuarios entre sí. Típicamente los chats se han implementado haciendo uso de sockets de flujo, pero existen otras alternativas que también merece la pena probar, tales como los sockets de datagrama o los de multidifusión.

Introducción Teórica

Hoy día, en muchos tipos de aplicaciones se cuenta con un servicio de chat, ya sea con fines de diversión o de negocio. Este tipo de servicio es muy socorrido cuando se trata de brindar una comunicación más personalizada que la brindada por medios tales como correo electrónico, o foros. En el chat la comunicación puede ser fluida, es decir, en tiempo real y además pueden interactuar dos o más personas a la vez.

Una red peer-to-peer, red de pares, red entre iguales o red entre pares (P2P, por sus siglas en inglés) es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es más, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados. Cuando un cliente entra a este sistema hace una conexión directa a uno de estos últimos, en donde recolecta y almacena toda la información y contenido disponible para compartir. Se trata entonces de un programa cuya función es la de conectar a los usuarios a través de una red sin servidores que facilita la descarga de música, películas, libros, fotos y software entre todos los otros usuarios, de manera gratuita. Estos archivos son compartidos “de computador a computador” por el solo hecho de tener acceso al sistema [1].

Una red peer-to-peer también se puede usar para implementar un chat donde los usuarios puedan comunicarse sin ningún costo asociado y dándoles más libertad puesto que los mensajes son almacenados entre los usuarios de la red y no en un servidor central.

Desarrollo de la práctica

En esta práctica se implementó una aplicación chat la cual permite a los usuarios comunicarse entre sí a través de una sala en común, además de permitirles el envío de mensajes privados haciendo uso de sockets de multidifusión.

A partir del programa “Principal” contenido en el paquete “Multicast2.zip” se realizaron las modificaciones correspondientes de manera que el programa funciona de la siguiente manera:

1. Cuando el usuario ingresa a la aplicación, el usuario proporciona un nombre de usuario y selecciona una interfaz, entonces se envía un mensaje al grupo indicando que el usuario ha entrado al chat.

2. Una vez enviado el mensaje de inicio, la aplicación cliente deberá iniciar la lectura del socket para recibir posibles mensajes de otros clientes.
3. El cliente que recién se ha unido recibirá un mensaje por cada uno de los clientes conectados de manera que así tendrá conocimiento de quienes se encuentran actualmente en el grupo.
4. Cuando llegue un mensaje, este deberá ser interpretado con base al tipo de mensaje y en su caso, deberán realizarse las acciones correspondientes.
5. En caso de que el mensaje recibido sea un mensaje privado, solamente la aplicación del usuario que tenga asociado el nombre de usuario del destinatario indicado en el mensaje será quien muestre dicho mensaje en pantalla. Las demás aplicaciones ignorarán esos mensajes.

El formato que tienen los mensajes para esta práctica es el siguiente:

- Para indicar que un usuario se ha unido al chat: `<msj><[INICIO:user_id]>[Ha entrado al chat]`
- Cuando un usuario ha enviado un mensaje a todos en el chat, para el resto de los usuarios que lo reciben se mostrará de la siguiente manera: `<msj><user_id> Este es un mensaje`. Mientras que para el que envía se verá de la siguiente manera: `Este es un mensaje`.
- Cuando un usuario desea enviar un mensaje privado a un usuario en específico la nomenclatura debe ser la siguiente: `<user_id_destino> Este es un mensaje`. Cuidando el espacio entre “>” y el mensaje a enviar.
- El usuario que reciba el mensaje privado lo verá de la siguiente manera: `<privado><user_id_remitente><user_id_destino> Este es un mensaje`.
- Cuando un usuario se une al chat, si se encuentran más usuarios en la sala previo a su ingreso, estos enviarán un mensaje presentándose y por tanto el usuario que acaba de ingresar recibirá el siguiente mensaje por cada usuario en la sala: `<privado><user_id_remitente><user_id_ingreso>[Bienvenido, me encuentro actualmente en el chat]`.

Estructura del programa

El programa contiene solo una clase “ChatP2P_Console.java” la cual además de la clase principal contiene las clases “Send.java” y “Receive.java” las cuales heredan la clase “Thread” para la ejecución de hilos, esto debido a que la clase “Send” será la encargada de enviar los mensajes y la clase “Receive” de recibirlos. Desde la clase “ChatP2P_Console” se elige el nombre de usuario y la interfaz a utilizar de la tarjeta de red y esos datos se envían a las clases “Send” y “Receive” los cuales en sus métodos “run” realizan las acciones que les corresponden a lo descrito anteriormente.

Simulación

Para la simulación se ejecutaron 3 instancias del programa, cada una con los siguientes nombres de usuario: “Lalo”, “Martha” y “Oscar”, siendo “Lalo” el primero en ingresar y por tanto recibirá las notificaciones de ingreso de los demás usuarios en cuanto estos se unan. “Martha” será el segundo cliente en unirse y por tanto después de enviar el mensaje de inicio recibirá un mensaje de bienvenida por parte de “Lalo”, por último, “Oscar” se unirá al chat notificando a “Lalo” y “Martha” y recibiendo de igual manera un mensaje de bienvenida por parte de cada uno de los usuarios que ya se encontraban en el chat. Posteriormente se enviarán varios mensajes desde los distintos clientes para mostrar el funcionamiento de la aplicación. La conversación ejemplo será la siguiente:

- Lalo: Hola, buenas noches.
- Martha: Buenas buenas.
- Oscar: Buenas noches joven.
- Mensaje privado de Lalo a Martha: Oscar llega tarde, como siempre.

- Oscar: Perdonen la tardanza.

```
12
Elige la interfaz multicast:
Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Lalo]>[Ha entrado al chat]
<msj><[INICIO:Martha]>[Ha entrado al chat]
<msj><[INICIO:Oscar]>[Ha entrado al chat]
|
```

Figura 1. Inicio de chat para el cliente “Lalo”.

```
12
Elige la interfaz multicast:
Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Martha]>[Ha entrado al chat]
<privado><Lalo><Martha>[Bienvenido, me encuentro actualmente en el chat]
<msj><[INICIO:Oscar]>[Ha entrado al chat]
|
```

Figura 2. Inicio de chat para cliente “Martha”.

```
12
Elige la interfaz multicast:
Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Oscar]>[Ha entrado al chat]
<privado><Lalo><Oscar>[Bienvenido, me encuentro actualmente en el chat]
<privado><Martha><Oscar>[Bienvenido, me encuentro actualmente en el chat]
|
```

Figura 3. Inicio de chat para cliente “Oscar”.

```
Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Lalo]>[Ha entrado al chat]
<msj><[INICIO:Martha]>[Ha entrado al chat]
<msj><[INICIO:Oscar]>[Ha entrado al chat]
Hola, buenas noches
<msj><Martha>Buenas buenas
<msj><Oscar>Buenas noches joven
<Martha> Oscar llega tarde, como siempre
<msj><Oscar>Perdonen la tardanza
|
```

Figura 4. Conversación desde la perspectiva del usuario “Lalo”.

```

Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Martha]>[Ha entrado al chat]
<privado><Lalo><Martha>[Bienvenido, me encuentro actualmente en el chat]
<msj><[INICIO:Oscar]>[Ha entrado al chat]
<msj><Lalo>Hola, buenas noches
Buenas buenas
<msj><Oscar>Buenas noches joven
<privado><Lalo><Martha> Oscar llega tarde, como siempre
<msj><Oscar>Perdonen la tardanza

```

Figura 5. Conversación desde la perspectiva del usuario “Martha”.

```

Elige la interfaz multicast:
Elegiste Realtek RTL8822BE 802.11ac PCIe Adapter
Socket unido al grupo /ff3e:0:0:0:0:1234:1
<msj><[INICIO:Oscar]>[Ha entrado al chat]
<privado><Lalo><Oscar>[Bienvenido, me encuentro actualmente en el chat]
<privado><Martha><Oscar>[Bienvenido, me encuentro actualmente en el chat]
<msj><Lalo>Hola, buenas noches
<msj><Martha>Buenas buenas
Buenas noches joven
Perdonen la tardanza

```

Figura 6. Conversación desde la perspectiva del usuario “Oscar”.

Preguntas sobre el resultado final de la práctica

¿Qué ventajas presenta el uso de sockets de multidifusión contra unidifusión?

La mayor ventaja es que permite tener una comunicación multicast la cual a su vez permite el envío de datos desde un emisor hacia muchos receptores o desde muchos emisores hacia muchos receptores, siempre y cuando la gestión de estos se haga de manera adecuada [2].

¿Qué modificaciones a nivel del campo Tiempo de vida (TTL) es necesario considerar?

Si la aplicación se deseara implementar a una mayor escala sería bueno considerar incrementar la cantidad de saltos que debe seguir el paquete, siempre considerando un límite para que los paquetes no se mantengan dando saltos entre los enrutadores de manera indefinida.

¿Qué consecuencia tendrá en el desempeño de la aplicación habilitar el algoritmo de Nagle?

En el caso de una ejecución local no tendría un impacto alto debido a que solo mejora la eficiencia cuando existen retardos grandes como sucede en redes WAN.

Conclusiones Individuales

Ruvalcaba Flores Martha Catalina

Para resolver la práctica, la aplicación de sockets de datagrama y sockets multidifusión fueron útiles para cumplir con los requisitos de la práctica, puesto que se utilizaron los programas proporcionados por el profesor, fue mas sencillo realizar las modificaciones necesarias para que cumpliera con su objetivo. Con esta practica se logró comprender de mejor forma como funcionan los sockets y los hilos. De igual forma, el que no existiera un servidor y varios clientes en una sala común, se puede observar como funciona que los mensajes lleguen a un grupo o a un solo cliente de forma privada, de acuerdo con un formato es especifico para el envío de mensajes.

Sandoval Hernández Eduardo

Está práctica resultó ser más sencilla que las anteriores en cuanto a la implementación de la lógica necesaria para el programa, esto gracias a que se contaba con los programas provistos por el profesor lo cual facilitó su modificación para adaptarlo a los requerimientos de la práctica, sin embargo, si hubo algunos problemas como es el caso de al ingresar el nombre de usuario y la interfaz de la tarjeta de red a utilizar pues los mensajes que indican al usuario que debe ingresar dichos datos ocurren después de que se ingresan, otro problema que se pudo identificar fue que en ocasiones cuando se unía un cliente al grupo este envía el mensaje de inicio pero no recibe los mensajes de bienvenida por parte de los demás usuarios o no de todos. A pesar de estos detalles, la aplicación es funcional y puede mejorarse más adelante, además de que se cumplió el objetivo de la práctica el cual era comprender el uso de los sockets de multidifusión y sockets de datagrama.

Referencias

[1] Wikipedia (s.f). Peer-to-peer. [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Peer-to-peer>

[2] R. Hernández Palacios, G. Hernández Hernández (s.f.). Comunicaciones Multicast. [En línea]. Disponible en: <https://www.uaeh.edu.mx/scige/boletin/huejutla/n9/r1.html>