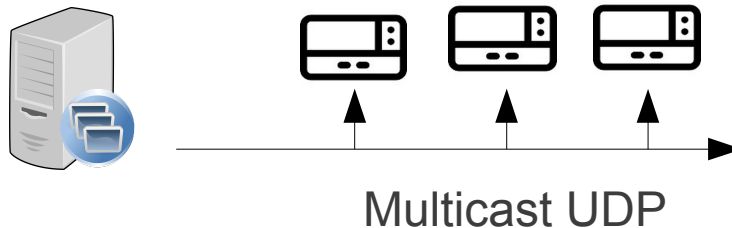


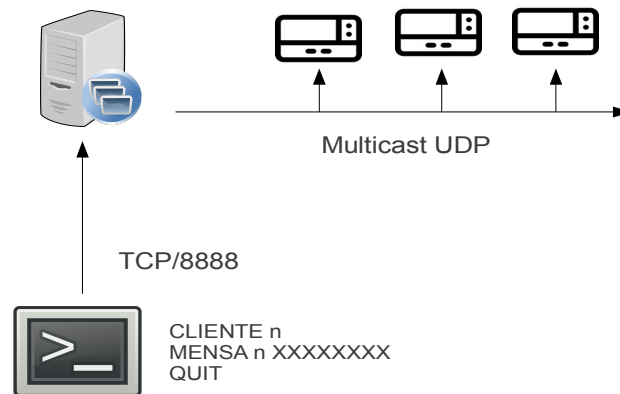
Ejercicio de valoración de conocimientos. Segunda Evaluación 2018
2º DAMM. Programación de Servicios y Procesos

Se desea implementar en una empresa una **aplicación de mensajería** que permita la distribución de mensajes a aplicaciones cliente mediante un protocolo de **multicast UDP**. Para ello, un servidor será el encargado de enviar los mensajes a todas las estaciones que estén a la escucha mediante este protocolo. De esta forma el servidor se despreocupa de las estaciones receptoras.



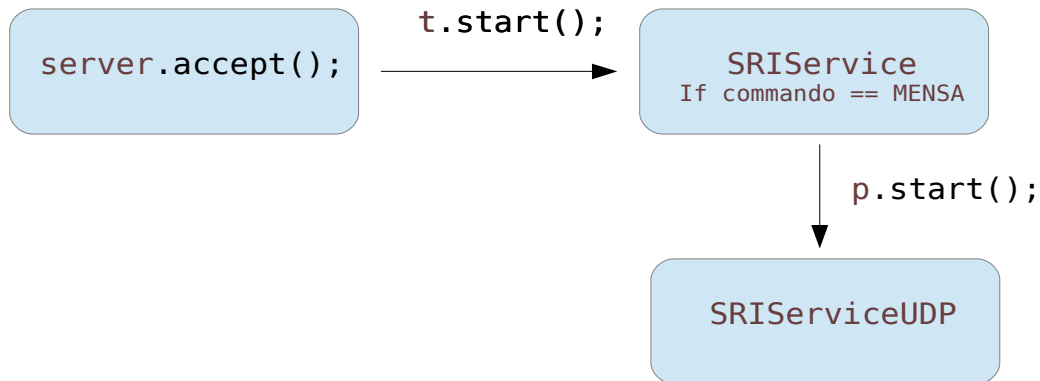
Para **enviar los mensajes** a las estaciones receptoras un **cliente-emisor** podrá **conectarse** al servidor mediante el protocolo TCP en el **puerto 8899**. El emisor dispondrá de un conjunto de comandos para comunicar con el servidor. Estos comandos son los siguientes:

- **CLIENTE**. Este comando nos devolverá un número de cliente que podremos utilizar para comunicaciones posteriores.
- **MENSA <numero cliente> <MENSAJE>**. Este comando es el que se utiliza para enviar desde el cliente-emisor al servidor el mensaje que queremos hacer llegar a las estaciones receptoras. Por ejemplo si enviamos: MENSA 1 HOLA las estaciones receptoras recibirán el cliente que envía y el mensaje. Este **mensaje se enviará dos veces ya que UDP no garantiza la recepción**. Un ejemplo "Cliente 1: Esta tarde reunión a las 18:00h".
- **QUIT**. Este comando desconecta el cliente-emisor del servidor de mensajería.



El funcionamiento básico del servidor será similar al visto en el aula o desarrollado en las prácticas (BankServer o SRI). El servidor estará a la espera de las conexiones entrantes. Una vez que un cliente-emisor se conecte, se iniciará un hilo encargado de gestionar el servicio de comunicaciones con el cliente-emisor [Clase **SRIService**] ya que la gestión de las comunicaciones se realizará mediante programación multihilo. Cuando esta clase **SRIService** procese el comando MENSA iniciará un nuevo servicio (clase **SRIServiceUDP**), otro hilo. Este servicio será el encargado de la emisión multicast (esto es el mismo servicio que se ha realizado en prácticas, pero en vez de emitir la hora se enviará el mensaje recibido y deberemos cambiar la clase para que funcione como un hilo).

Básicamente el funcionamiento del servidor es como se describe en la siguiente imagen:



NO ES NECESARIO IMPLEMENTAR LA APLICACIÓN DESDE CERO. Para ello vamos a contar con parte del código de la práctica del servidor SRI propuesta para su realización por los alumnos(o con el código de vuestras propias prácticas). Es posible que este código no esté totalmente optimizado y nos sobren clases y/o métodos pero esto no es relevante ya que no será necesario tocarlas (Clases **SRI** y **SRIRespuesta**). También necesitaremos utilizar el código del servidor y cliente multicast creado como servidor horario o el suministrado en el apartado 1.6 Comunicación Cliente/Servidor con multicast del documento **UT2.7 Programación de comunicaciones**.

Por ello vamos a dividir nuestro trabajo en tareas:

- 1ª **Tarea.** Terminar el servidor **SRIserver** para que funcione en multihilo, iniciando una instancia de la clase **SRIService** cada vez que reciba una conexión de entrada.
- 2ª **Tarea.** Implementar los comandos de control del servidor CLIENTE y QUIT.
- 3ª **Tarea.** Implementar el comando MENSA iniciando la clase **SRIServiceUDP** en forma de hilo.
- 4ª **Tarea.** Crear una estación receptora multicast. Modificación básica del cliente UDP multicast.

Las clases que debemos tener en el **servidor** son las siguientes:

- **SRIServer** Clase principal del servidor que debemos terminar para que inicie la clase **SRIService** en multihilo.
- **SRIService** Clase encargada de gestionar la comunicaciones con el cliente-emisor, debe gestionar los comandos e iniciar la clase **SRIServiceUDP** como otro hilo.
- **SRIServiceUDP** Clase encargada de emitir los mensajes por UDP multicast. Esta clase debe implementarla el alumno (usar como base el servidor multicast).
- **SRI** y **SRIRespuesta** Clases del anterior proyecto que creaban un array para las respuestas de los clientes. **No es necesario tocarlas si optas por reutilizar el código suministrado.**

Además se deberá crear otro proyecto con el **código de las estación receptora** que escucha por UDP multicast. Esta aplicación mostrará los mensajes del servidor. La dirección de multicast es a criterio del alumno.

Criterios de evaluación:

1. Las clase SRIService se ha implementado adecuadamente, realizando la función sugerida y los parámetros de la clase son coherentes con su funcionalidad. (2 puntos)
2. Las clase SRIServiceUDP se ha implementado adecuadamente, realizando correctamente la función sugerida y los parámetros de la clase son coherentes con su funcionalidad. (2 puntos)
3. El funcionamiento de la aplicación mediante el uso de multihilo que gestiona las comunicaciones con el cliente-emisor se ha realizado adecuadamente. (2 puntos)
4. El cliente UDP Multicast se ha realizado correctamente (2 puntos)
5. La aplicación presenta una lógica coherente, funciona correctamente y está comentada adecuadamente. (2 puntos)

Entrega del ejercicio

El ejercicio deberá ser entregado en un archivo comprimido con el nombre y apellidos. El archivo contendrá dos directorios, uno con los fuentes del servidos y otro con los del cliente multicast. Para el cliente-emisor podemos utilizar netcat.