

# REDES DE COMUNICACIONES II

## Práctica 3 - Multimedia

[Volver a: Prácticas ➡](#)

### 2 Arquitectura y servidor de descubrimiento

A grandes rasgos, la aplicación P2P de teleconferencia deberá permitir que el usuario realice las siguientes acciones:

- **Gestionar sus nicks (creación y cambio).** Para ello deberá solicitar un nick al usuario, y registrarlo a continuación en el **servidor de descubrimiento**. Este nick tendrá una validez de 24 horas, periodo durante el que estará "reservado", a través del uso de una contraseña, a dicho usuario. Pasado ese tiempo, el servidor eliminará el nick, y otro usuario podrá reclamarlo y registrarlo para sí.
- **Realizar una videollamada con otro usuario.** Para ello la aplicación deberá permitir que el usuario escriba (o elija de una lista) el nick del usuario con el que desea conectar. Tras ello, abrir una conexión directa con este usuario (tal y como se explica en las siguientes secciones), y empezar a capturar, enviar y recibir el video. También deberá permitir pausar y finalizar la llamada.

#### Servidor de descubrimiento (DS)

Como sabemos, los servicios de videoconferencia, como Skype o el que desarrollamos en esta práctica, suelen seguir el esquema P2P, con la conexión directa entre los nodos participantes en la comunicación. Sin embargo, este sistema necesita de un nodo especial, fijo, que funcione a modo de tablón de anuncios donde cada nuevo participante en la red pueda "colgar" su dirección IP y puerto de escucha, de forma que el resto de participantes puedan conectarse a él.

En nuestra práctica, este papel lo realiza el **servidor de descubrimiento**, que consiste en un pequeño servicio que escucha en el puerto 8000 del host *vega.ii.uam.es*, y recibe y envía los mensajes como simples cadenas de texto ASCII.

Su función esencial es mantener una pequeña base de datos con todos los usuarios registrados en la plataforma, guardando para cada uno de ellos, *nick*, *dirección IP*, *puerto de escucha*, *un timestamp del momento en el que se registró o actualizó la validez del nick por último vez y protocolos soportados*.

Para ello, soporta los siguientes mensajes:

##### Función REGISTER

*Objetivo:* Registrar a un usuario en el sistema. También sirve para cambiar el nick de un usuario existente.

*Comando:* REGISTER nick ip\_address port password protocol

*Posibles respuestas/errores:*

- OK WELCOME nick ts : El registro o actualización del nick se ha realizado correctamente. ts contiene el nuevo instante de registro (en formato UNIX)
- NOK WRONG\_PASS : El nick es válido, pero la contraseña proporcionada es errónea

Ejemplo

```
C->S: REGISTER mi_nick 138.100.10.100 8080 secret V1#V2
```

```
S->C: OK WELCOME mi_nick 1521798459.6474934
```

Con este mensaje, el cliente registra el nick 'mi\_nick', con la dirección IP 138.100.10.100 y puerto 8080 en el servidor. También le asigna la contraseña 'secret' e indica que soporta los protocolos V1 y V2.

Como se ha comentado, una vez registrado por primera vez un nick, el servidor lo mantendrá reservado durante 24h, y los intentos de registrar ese nick por parte de otros usuarios serán rechazados (con una respuesta NOK WRONG\_PASS). Para seguir utilizando su nick, el usuario legítimo debe volver a llamar a REGISTER con la contraseña

adecuada, momento en el que se le extenderá la validez del mismo otras 24h. Típicamente, la aplicación debería llamar a REGISTER cada vez que sea abierta por el usuario.

### **Función QUERY**

*Objetivo:* Permite obtener la dirección IP y puerto de un usuario conociendo su nick

*Comando:* QUERY name

*Posibles respuestas/errores:*

- OK USER\_FOUND nick ip\_address port protocols : Devuelve información sobre el usuario. protocols contiene todos los protocolos soportados por ese cliente, separados por '#'.
- NOK USER\_UNKNOWN

### **Función LIST\_USERS**

*Objetivo:* Listar todos los usuarios registrados en el sistema

*Comando:* LIST\_USERS

*Posibles respuestas/errores:*

- OK USERS\_LIST user1#user2#...#usern# : Devuelve una lista de los usuarios registrados en el sistema, mostrando para cada uno la información devuelta por el comando QUERY, separados por el símbolo '#'
- NOK USER\_UNKNOWN

### **Función QUIT**

*Objetivo:* Señalizar el cierre de la conexión con el servidor

*Comando:* QUIT

*Posibles respuestas/errores:*

- BYE

Volver a: Prácticas ➡