Universidad Nacional Autónoma de México

Facultad de Ciencias

Redes de computadoras

Creación e implementación de un protocolo de la Capa de Aplicación

Profesor: Paulo Contreras Flores
Ayudante: Virgilio Castro Rendón
Ayudante Lab: Daniel Campuzano Barajas

Objetivo

- El alumno diseñará un protocolo de la capa de aplicación, con una utilidad específica.
- El alumno implementará este protocolo tanto para el cliente como para el servidor.

Desarrollo

Se diseñará un protocolo de la capa de aplicación, el cliente solicitará un Pokemon a capturar, el servidor lo ofrecerá y aleatoriamente se indicará si se capturó o no.

I. Diseño del protocolo

En la *FSM* de la figura 1 se presenta el comportamiento básico de la aplicación.
y en la tabla 1 la descripción de los estados,

Estado	Descripción
s0	Estado inicial, desde aquí iniciara la conexión del protocolo de la capa de aplicación.
s1	Recibe solicitud del cliente, ofrece aleatoriamente un Pokemon para capturar.
s2	Indica si quiere capturar o no el Pokemon ofrecido.
s3	Inicia un contador con {n} como el máximo número de intentos. Aleatoriamente indica si se capturó al Pokemon o no.
s4	Da respuesta para reintentar captura de Pokemon.
s5	Se recibe Pokemon capturado (imagen).
s6	Terminando la sesión.
s7	Cierre de conexión.

Tabla 1: Descripción de los estados

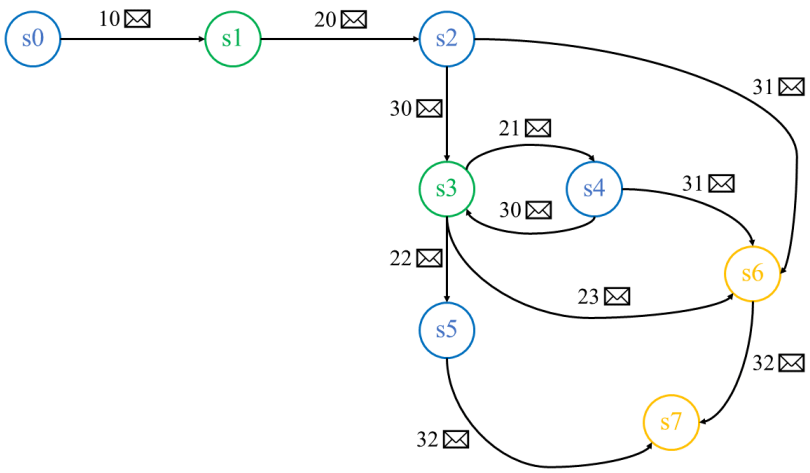

Figura 1: Máquina de estado finita (*FSM*) para el protocolo de la capa de aplicación.

A continuación se definen los tipos de mensajes a utilizar,

Código	Descripción
10	Solicitar al servidor por parte del cliente, un Pokemon para capturar.

Tabla 2: Códigos del cliente

Código	Descripción
20	¿Capturar al Pokemon x?.
21	¿Intentar captura de nuevo? Quedan k intentos.
22	Envía Pokemon (imagen) capturado.
23	Número de intentos de captura agotados.

Tabla 3: Códigos del Servidor

Código	Descripción
30	Sí.
31	No.
32	Terminando sesión.

Tabla 4: Códigos comunes del cliente y del servidor

Los mensajes para los códigos 10, 30, 31, 32, y 23 estarán conformados únicamente por el código, y serán de 1 byte.

code
1 byte

El mensaje para el código 20 es

code	idPokemon
1 byte	1 byte

El mensaje para el código 21 es

code	idPokemon	numAttemps
1 byte	1 byte	1 byte

El mensaje para el código 22 es

code	idPokemon	imageSize	image
1 byte	1 byte	4 bytes	k bytes

II. Implementación del protocolo.

Se implementará tanto el servidor como el cliente siguiendo las características del protocolo, se recomienda el uso del lenguaje de programación C y los ejemplos vistos en clase, sin embargo se deja a elección del alumno la decisión.

II.1 El servidor aceptará conexiones en el puerto 9999. Usará hilos para manejar varias conexiones de los clientes a la vez.

II.2 El cliente guardará y/o mostrará la imagen del Pokemon capturado. Y recibirá como parámetros desde la línea de comandos la dirección IP del servidor y el puerto al cual se conectará.

II.3 Se recomienda el uso de Wireshark durante la programación para facilitar la tarea de encontrar posibles errores.

II.4 Agregar estados y mensajes (códigos 40) para indicar condiciones de error.

II.5 Se deberán de programar tiempos de espera (*timeouts*) para la recepción de los mensajes ya sea por parte del cliente, o por parte del servidor, es decir, si no se obtuviera una respuesta por parte del cliente o del servidor se deberá de cortar la conexión, enviando previamente un mensaje de término.

II.6 Extender la funcionalidad para que del lado del servidor exista una base de datos que contenga el registro de Pokemon capturados por el usuario (un Pokedex), la base de datos ya deberá de contener registros de usuarios (no se tendrá que implementar el alta de usuarios). Cada vez que el usuario de la aplicación atrape un Pokemon, el servidor deberá de registrar en dicha base de datos al nuevo Pokemon capturado; además se deberá de implementar la consulta a la base de datos por parte de los usuarios. Para estos dos últimos puntos se recomienda agregar nuevos tipos de mensajes y al menos un campo de id de usuario (id de entrenador Pokemon) en los mensajes que se requiera.

II.7 No es requisito usar un Manejador de base de datos para almacenar la información, aunque sí se recomienda, por que ya tiene funciones de búsqueda, inserción, etc, implementadas.

II.8 Se alienta al alumno a que mejore la propuesta inicial de la aplicación, mejorando los estados, transiciones y mensajes.

Condiciones de entrega

- El proyecto lo podrán realizar en equipos de tres alumnos como máximo.
- Entregar un reporte en PDF que contenga,
  - Objetivo, redactado para una venta de la aplicación,
  - Diseño del protocolo, hacer el diagrama de la *FSM* en donde se indiquen las transiciones entre estados con base en las adecuaciones solicitadas, incluir en una tabla el significado de los estados, tipo de mensajes, y cómo es que se conforman los mismos. Se recomienda hacers la tabla de transición de estados para facilitar la lógica de la implementación del protocolo.
  - Uso, indicar cómo compilar o interpretar, y ejecutar tanto el cliente como el servidor, deberá incluir imágenes de apoyo en las que se vea el correcto funcionamiento de su aplicación. Además incluir capturas de pantalla de Wireshark en donde se muestre los mensajes tanto del cliente como del servidor, indicando los campos de cada mensaje en el área de datos de Wireshark en donde se muestra el mensaje en hexadecimal.
  - Lista de funciones usadas en la programación, deberán explicar para qué se usa y qué parámetros recibe y cuáles regresa. Esto lo pueden realizar documentando su código y después usando funcionalidades como javadoc, doxygen étc, generar la documentación en HTML.
- Realizar un archivo Makefile para compilar, instalar y borrar la aplicación, uno para el cliente, y otro para el servidor. También deberán de instalar la documentación en el manual de Linux (man), así como en HTML. En este manual se deberán de indicar los parámetros para ejecutar tanto el cliente como el servidor.
- Subir a un repositorio Git el programa y demás archivos que sean necesarios para la ejecución.
- Subir a Moodle el documento en formato PDF, indicando el enlace al repositorio Git.
- La fecha de entrega será a más tardar el día viernes 14 de diciembre.