Laboratorio 1: Redes de Computadores

Profesor: Jorge Díaz
Ayudantes: Iñaki Oyarzun M. & Javiera Cárdenas

Marzo - Abril 2024

1 Objetivos del laboratorio

- Aprender a utilizar sockets UDP y TCP en Go.
- Conocer y aplicar la estructura cliente-servidor.
- Aprender a realizar el análisis del tráfico de red a partir de la herramienta Wireshark.
- Familiarizarse con protocolos altamente usados en Internet.

2 Introducción

Un socket corresponde a la interfaz existente entre las capas de aplicación y transporte, estos permiten establecer una conexión entre aplicaciones, para que, de esta forma, pueda ser llevado a cabo el intercambio de mensajes entre las mismas. Estas aplicaciones pueden estar ejecutándose tanto en una misma maquina o en dos, en donde esta interacción permite dar lugar a la estructura cliente-servidor. Es en este sentido donde, tanto Python como Go entregan la posibilidad de facilitar el trabajo de establecer una conexión a partir del uso de sockets (de dominio Unix en este caso). Por un lado, Python posee una libreria llamada **socket** y por el otro, Go con su libreria **net**. Como fue adelantado previamente, este laboratorio pretende evaluar una estructura clásica en el ámbito de redes: la **cliente-servidor**, donde la carga de trabajo se encuentra distribuida entre los servidores (proveedores del servicio) y los clientes (consumidores del servicio provisto). Ademas de aquello, se agrega la utilización de la herramienta Wireshark. Este software es el encargado de realizar el análisis del trafico de red en tiempo real, teniendo como agregado, la posibilidad de facilitar la identificación de los protocolos antes mencionados **TCP** y **UDP**.

3 Laboratorio 1

3.1 Enunciado

Es el año 16XX en los turbulentos mares del Caribe, usted es un intrepido pirata legendario que ha alcanzado proporciones míticas, pero su reinado como el más temido pirata se ve desafiado por la sombría figura del Capitán Blackbeard. Este último, con su ferocidad y astucia, ha socavado su reputación, robando su botín y desafiando su autoridad en cada paso. Un día, una espesa niebla envuelve repentinamente el oceano en un manto de incertidumbre, y tus instrumentos de navegación se encuentran destrozados por la tormenta reciente.

Ante esta situación, entre la niebla logras divisar el barco del Capitán Blackbeard. Es tu momento de acabar con el, pero te ves obligado a confiar en su instinto y experiencia marinera para ubicar la posición del navío de su rival debido a la niebla. La tensión en el aire es palpable mientras tu tripulación se apresura a preparar los cañones para el enfrentamiento inminente. En medio de la niebla, todos esperan su decisión para lograr vencer a Blackbeard.....

3.2 Explicación

Deberá realizar 2 programas, un cliente y un servidor, en el cual usted pueda competir con el ultimo en el conocido juego de mesa "Battleship" con tableros de **tamaño 2x2** en donde cada ubicación va a ser representada por una letra: A, B, C y D.

3.2.1 Inicio:

- **Cliente** debe abrir una **conexión UDP** enviar un mensaje al servidor solicitando iniciar la partida y recibir su tablero correctamente para iniciar mediante este mismo tipo de conexión.
- **Servidor** deberá generar ambos tableros y asignar los barcos en lugares aleatorios, posteriormente este espera el mensaje del cliente para saber que este se encuentra disponible para jugar e enviar uno de los tableros a este para que usted pueda conocer la ubicación de su barco por **conexión UDP** junto con la IP y puerto a utilizar para la **conexión TCP** en el desarrollo del juego.

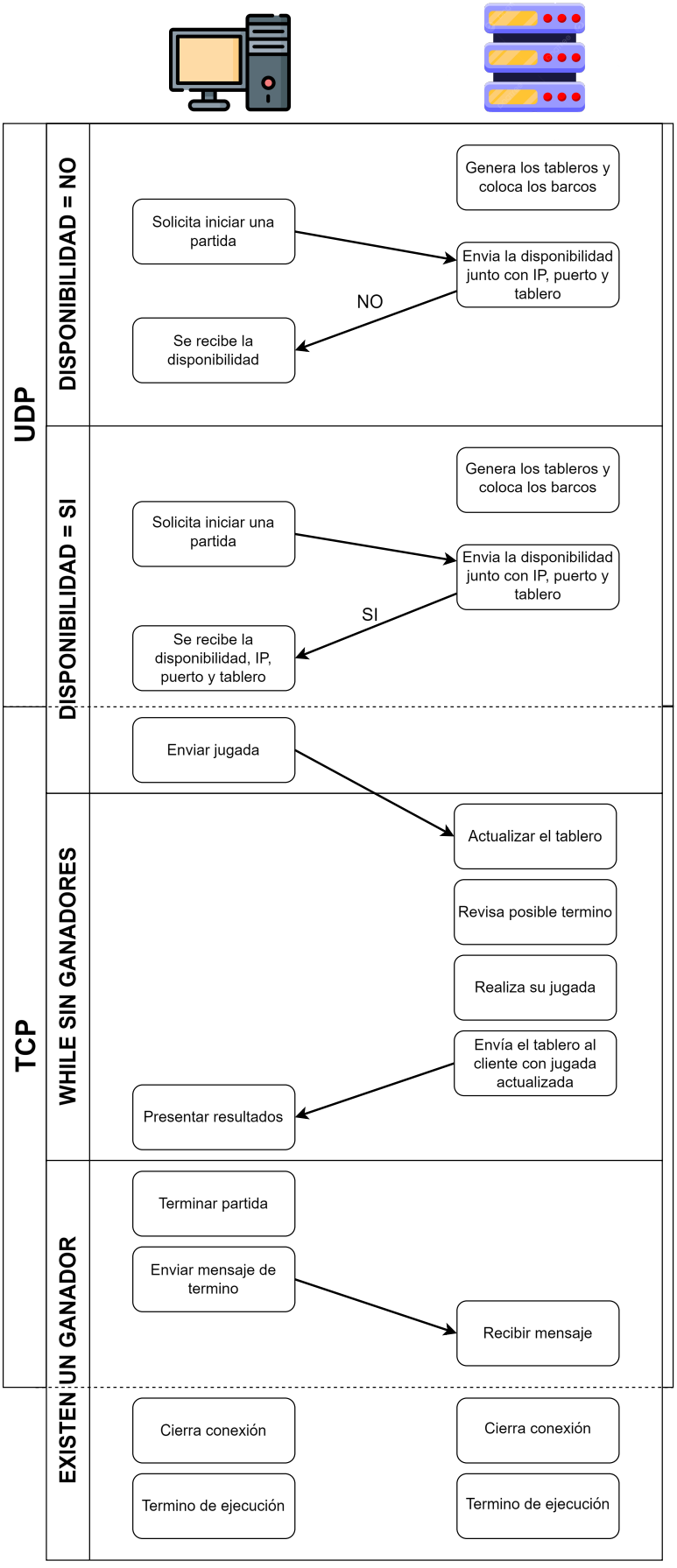
3.2.2 Desarrollo:

- **Cliente** abre una **conexión TCP** utilizando el puerto e IP previamente recibidos y envía su jugada indicado la ubicación en donde desea lanzar un ataque.
- **Servidor** actualiza los tableros en relación a las jugadas realizadas y revisa si el cliente o el servidor es el ganador, en caso de que esto no ocurra, le devuelve al cliente su tablero actualizado mediante **conexión TCP** y continua el juego, pero si existe un ganador se procede con el final del juego.

3.2.3 Final:

- **Cliente** Recibe el mensaje mediante la **conexión TCP** indicando si el servidor o cliente fue el ganador, responde indicando que el mensaje fue recibido, cierra la **conexión TCP** y procede con el termino de su ejecución.
- **Servidor** Envía un mensaje mediante la **conexión TCP** indicando quien es el ganador, espera la respuesta del cliente, cierra la conexión y procede con el termino de su ejecución.

Para mayor claridad se adjunta el siguiente diagrama:


UDP
DISPONIBILIDAD = NO
Genera los tableros y coloca los barcos
Solicita iniciar una partida
Envia la disponibilidad junto con IP, puerto y tablero
NO
Se recibe la disponibilidad
DISPONIBILIDAD = SI
Genera los tableros y coloca los barcos
Solicita iniciar una partida
Envia la disponibilidad junto con IP, puerto y tablero
SI
Se recibe la disponibilidad, IP, puerto y tablero
TCP
Enviar jugada
WHILE SIN GANADORES
Actualizar el tablero
Revisa posible termino
Realiza su jugada
Envía el tablero al cliente con jugada actualizada
Presentar resultados
EXISTEN UN GANADOR
Terminar partida
Enviar mensaje de termino
Recibir mensaje
Cierra conexión
Cierra conexión
Termino de ejecución
Termino de ejecución

4 Reglas de entrega

- El laboratorio se realiza en parejas durante el horario seleccionado en el laboratorio de redes .
- La fecha de entrega depende en el horario que este realizando el laboratorio.
- Para las conexiones del código en Go, se debe utilizar la librería net.
- La entrega debe realizarse a través de Aula, en un archivo comprimido `.zip`, indicando el número de Laboratorio y grupo en el siguiente formato: `L1-Grupo[Nº Grupo].zip`, Ejemplo: `L1-Grupo01.zip`.
- Debe entregar todos los archivos fuente necesarios para la correcta ejecución de la entrega. Teniendo al menos un archivo para el Cliente y un Servidor. Con el código bien identado, comentado, sin warnings ni errores.
- Debe entregar un **README** con nombre y rol de cada integrante del grupo, además de las instrucciones necesarias para ejecutar correctamente el laboratorio.(**ADVERTENCIA:** Si no se entrega dicha información, se colocará un cero a la entrega y posteriormente se tendrá que coordinar una sesión de apelación.)
- Cada hora de retraso penalizará el laboratorio, descontando 30 ptos.
- Cualquier sospecha de copia será notificada debidamente al profesor y evaluada con nota 0. **Siendo tomado en cuenta también cualquier copia directa de algún sitio web o foro**. Se tendrá un software a mano para realizar dichas comparaciones.