

Trabajo Práctico

Segundo semestre 2024

Introducción

En este trabajo práctico se explorarán varios aspectos del protocolo TCP y la capa de transporte. El objetivo es implementar una conexión segura entre un servidor y un cliente sobre un canal de envío de paquetes inseguro usando la librería de Python Scapy¹. Luego de implementarla deberán realizar una experimentación utilizando la solución propuesta para descubrir aspectos de la conexión subyacente.

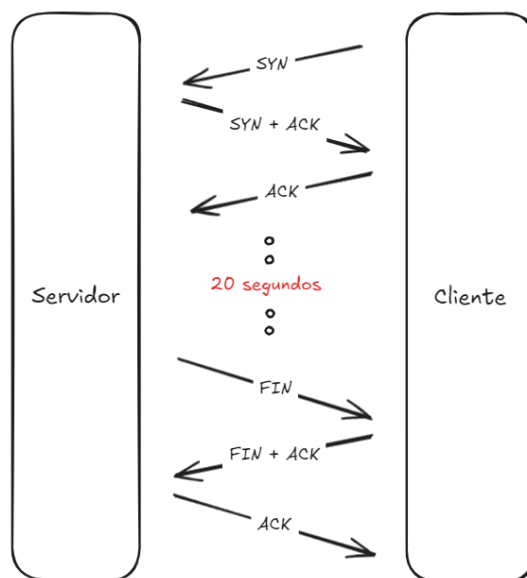
Contexto/Canal inseguro

En la librería `canalruidoso` se tiene la función `envio_paquetes_inseguro` la cual simula un canal de envío de paquetes en el cual puede ocurrir delay, corrupción de paquetes y pérdida de paquetes. Qué un paquete llegue con delay lo definimos como que tarde más de 3 segundos en llegar. Estos eventos son independientes, siempre ocurre uno solo, por ejemplo si el paquete llegó con delay no puede estar corrupto. Además los paquetes pueden llegar sin ningún tipo de problema.

El servidor y el cliente tendrán que comunicarse a través de ese canal y se deberá implementar un servicio confiable sobre ese canal. Quien debe iniciar la conexión es el cliente, se debe realizar el 3-way handshake entre este y el servidor para establecer la conexión. Luego de 20 segundos de establecida la conexión, el servidor va enviar un mensaje de cierre al cliente y la conexión se debe cerrar de la manera correcta. El tiempo de espera de un ACK (SYN ACK incluido) luego de enviar un mensaje es de 3 segundos, si después de ese tiempo no llega el ACK que confirma el mensaje enviado, se deberá retransmitir el paquete.

A continuación se muestra un intercambio de mensajes entre cliente y servidor, en un caso donde todos los paquetes llegan bien.

¹<https://scapy.readthedocs.io/en/latest/> - Documentación oficial de Scapy
<https://0xbharath.github.io/art-of-packet-crafting-with-scapy/index.html>



La función `envio_paquetes_inseguro` utiliza la función `send()` de `scapy` para enviar la información. Cada vez que se necesite enviar un mensaje del cliente al servidor o del servidor al cliente se utilizará la función `envio_paquetes_inseguro`. No está permitido la comunicación con funciones que no sean esta.

Consignas:

1. Implementar una conexión confiable entre el cliente y el servidor teniendo en cuenta todos los problemas mencionados anteriormente. El programa debe estar implementado en Python 3, utilizando Scapy. Se debe realizar el 3-way handshake entre cliente y servidor, siendo el cliente quien inicia la conexión, y el cierre de la conexión, donde el servidor será quien cierre.
2. Calcular todos los porcentajes de los errores que ocurren en el canal inseguro y averiguar el delay promedio cuando un paquete se retrasa.

Experimentación e Informe

Además, deberán realizar un informe sobre el trabajo realizado. En el informe se deberán explicar cómo es que sus códigos logran hacer un canal seguro ante cualquiera de los fallos. Además deberán mostrar los cálculos realizados para los porcentajes de errores utilizando

gráficos para justificarlos. Tienen la libertad de agregar más experimentos si los consideran relevantes. El informe debe tener un mínimo de 3 páginas (sin carátula).

Modalidad de entrega

El trabajo debe realizarse en grupos de 3 alumnos.

Deberán entregar un solo archivo en formato .zip, el cual deberá contener los siguientes archivos:

- Informe en formato PDF
- Archivos de Python con la implementación (Uno para el servidor y uno para el cliente)

El TP debe ser entregado a través del campus. La fecha límite de entrega es el 3/11 hasta las 23:59 hs. No se aceptarán TPs entregados después de este horario.