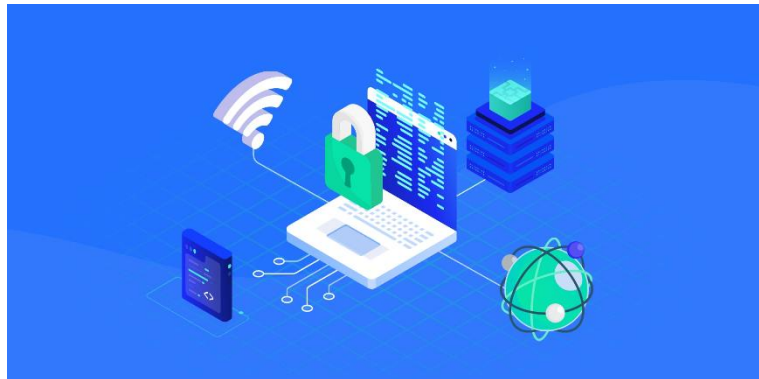


Tecnológico de Costa Rica
Escuela de Ingeniería en Computación



Proyecto I “Protocolos de comunicación”

Redes

Profesor:
Ing. Juan Manuel Sánchez Corrales

Presentado por:
Bonilla Espinoza, Alina
Hernandez Cordoba, Taylor
Rios Montero, Fabricio

Alajuela
Octubre, 2023

Introducción

En el campo de las redes computacionales, la comunicación entre los dispositivos es un ámbito muy importante y se busca que sea lo más estable y seguro posible. Este proyecto consiste en desarrollar una simulación que está compuesta por 6 tipos de protocolos de comunicación, los cuales son: Utopía, Stop-and-wait, PAR, sliding window de 1 bit, go-back-n y selective-repeat.

Por consiguiente, se trabajó con la capa física y la capa de enlace donde se crearon los protocolos vistos en clase, cada uno fue diseñado con el fin de garantizar una comunicación punto a punto confiable entre dos máquinas. Cabe recalcar, que todos los protocolos estudiados tienen sus propios desafíos, como lo es la comunicación unidireccional hasta la bidireccional, pasando por la gestión de ventanas y la detección de errores.

Por otro lado, el simulador presenta una interfaz sencilla que permite visualizar el intercambio de información entre las dos máquinas, además tiene otras opciones como pausar la simulación para un análisis detenido. Además para la ejecución de algunos protocolos se modelaron diferentes tipos de eventos, desde la llegada de frames hasta la detección de errores y la disponibilidad de datos de la capa de red.

A continuación, se detalla el proceso que se realizó para la creación del proyecto:

1. Lenguaje de programación y entorno

El equipo a cargo tomo la decisión de seleccionar el lenguaje de programación Java para la elaboración del simulador y como Entorno de Desarrollo Integrado (IDE) se hizo uso de Visual Studio Code.

2. Creación de paquetes y clases:

Se tomó la decisión de dividir el proyecto en dos paquetes Protocolos y Clases, los cuales están compuestos por las siguientes clases:

- Protocolos:

En este paquete se encuentran las 6 clases correspondientes a cada uno de los protocolos desarrollados (Utopía, Stop-and-wait, PAR, sliding window de 1 bit, go-back-n y selective-repeat).

Además, la clase Interfaz la cual se encarga de crear todos los componentes necesarios para la parte visual del proyecto y la de Protocolo en esta se encuentran todas las funciones relacionadas para el manejo de la capa de enlace.

- Clases:

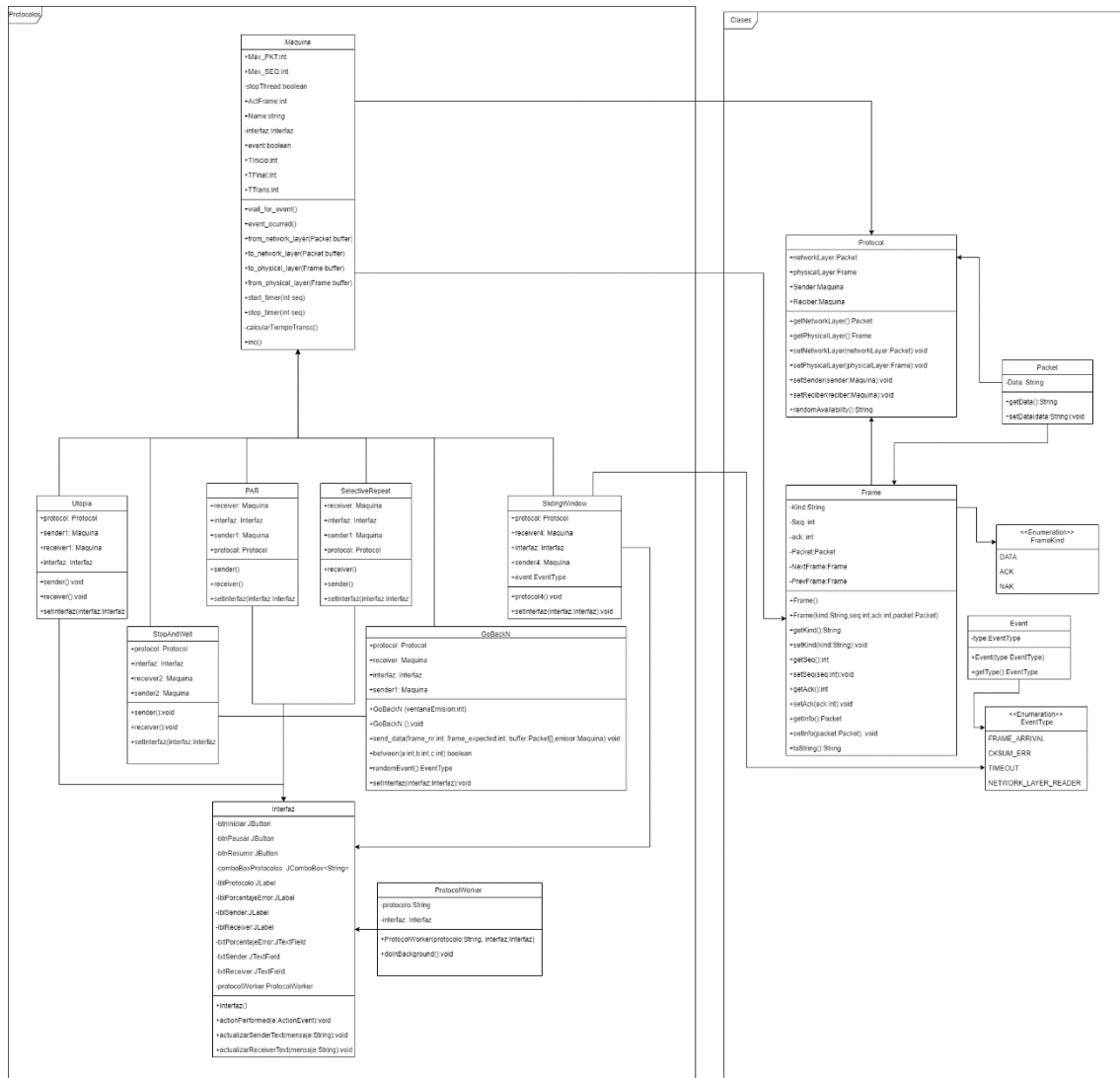
Dicho paquete esta compuesto por 5 clases (Event, EventType, Frame, FrameKind y Packet). Las dos primeras se encargan de manejar todo lo referente a los 4 eventos que existen en el proyecto, estos son detallados a continuación:

Tipos de eventos	Descripción
frame_arrival	Generado por la capa física cuando tiene un frame disponible sin errores para la capa de enlace.
cksum_err	Generado por la capa física cuando recibió un frame con errores.
timeout, ack_timeout	Generado por un timer al terminar su tiempo.
network_layer_ready	Generado por la capa de red cuando tiene un dato disponible para ser enviado.

Seguidamente, tenemos las clases de tipo Packet, la cual corresponde a una cadena de caracteres y la clase Frame esta corresponde a un objeto con los

atributos tipo, número de secuencia, número de confirmación e información de packet.

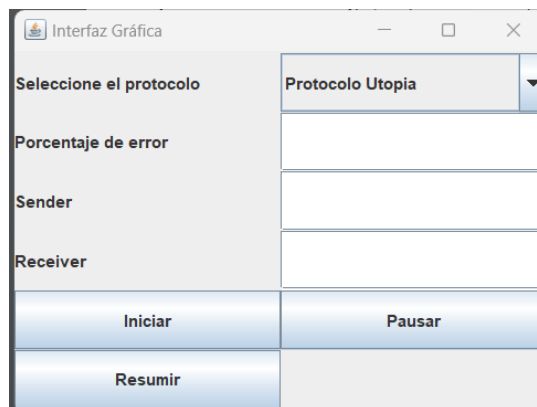
3. Diagrama del proyecto: Se puede acceder al diagrama de la clase por medio del siguiente link <https://app.diagrams.net/#G1SR8xCp42-ZXhVZAuwQECYoJKYwuFQIZ>



Análisis de resultados

A continuación, se detalla los resultados obtenidos durante el desarrollo del proyecto:

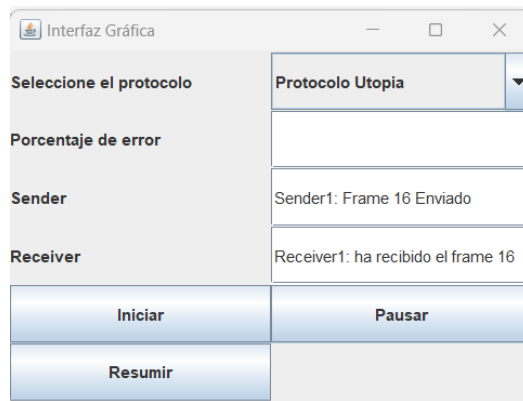
Interfaz del proyecto: Se opto por una interfaz sencilla pero con toda la información necesaria para el uso correcto de la simulación.



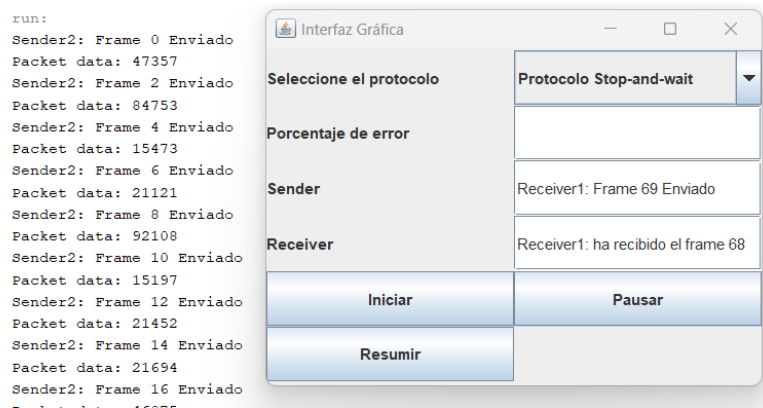
En la primera selección se debe seleccionar el protocolo de comunicación que deseamos trabajar, luego se debe dar click en el botón de iniciar y el sistema procederá a simular dicho protocolo. Se muestra en el campo de Sender los datos relacionados con el envío de paquetes y en el de Receiver la información de recepción de datos.

Protocolo Utopía: Durante la ejecución de este protocolo se puede observar el envío y recepción de los paquetes por medio de los campos Sender y Receiver. Cabe recalcar que por la naturaleza de este protocolo la comunicación se realiza en un solo sentido, es decir, desde el emisor (A) al receptor (B).

Tampoco existen limitaciones en la disponibilidad de la capa de red, no existen errores de transmisión por lo cual no se muestra ningún porcentaje de error y la máquina receptora es capaz de procesar los datos casi de inmediato.

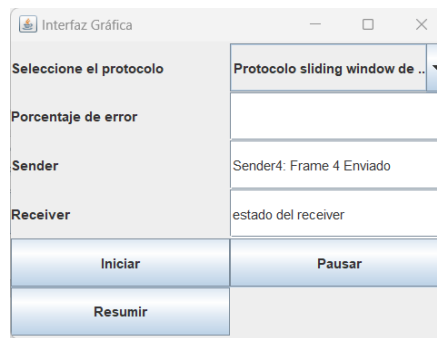


Protocolo Stop-and-wait: De igual manera que en el anterior se pueden observar los datos de envío y recepción de paquetes. Además, podemos ver el dato del Packet en consola. Recordemos que en este protocolo la comunicación es unilateral, sin embargo el emisor espera una confirmación del receptor antes de enviar el siguiente paquete, dicho comportamiento se puede observar en la simulación.



Protocolo PAR: Funciona similar al Stop-and-Wait, solo que en este caso la maquina receptora puede aprovechar la oportunidad para enviar datos junto con la confirmación, reduciendo así la latencia.

Protocolo sliding window: Durante este protocolo se permite la comunicación bidireccional, es decir, ambas maquinas pueden enviar y recibir paquetes. El tamaño de la ventana de emisión y recepción es 1, lo que significa que solo se permite un paquete en tránsito a la vez.



Protocolo go-back-n: Acá se permite enviar varios paquetes antes de recibir confirmaciones. Si se detecta un error en un paquete, todos los paquetes posteriores se descartan y se vuelven a enviar desde el último paquete confirmado.

Protocolo selective-repeat: Funciona similar al Go-Back-N, la diferencia está en la detección de error en el envío de un paquete, ya que si existe un error solo ese paquete se reenvía, mientras que los demás paquetes en tránsito continúan su proceso.

Conclusiones

Para concluir, con la elaboración de este proyecto los miembros del equipo tuvieron la oportunidad de aplicar y fortalecer conocimientos los diferentes protocolos de comunicación estudiados en la clase. Es importante mencionar que los protocolos implementados en este proyecto representan una estrategia para la transmisión confiable de datos en redes de computadoras.

Por otro lado, tomando en cuenta los resultados obtenidos se puede concluir que no existe un protocolo mejor, ya que esto dependerá de los factores específicos de la red y todo protocolo cuenta con desafíos y condiciones variables que pueden surgir en la comunicación punto a punto.

Sin embargo, se puede notar que dependiendo del protocolo existe una pequeña diferencia en el tiempo de ejecución, esto debido al comportamiento propio de los protocolos en comparación, algunos son mas lentos, ya que invierten tiempo esperando confirmaciones o son más sensibles a los errores.

Bibliografía

Difference between stop and wait protocol and Sliding window protocol (2023)

GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/difference-between-stop-and-wait-protocol-and-sliding-window-protocol/>

Kumar, K. (2022) Difference between stop and wait, GoBackN and selective repeat protocols, Online Courses and eBooks Library. Available at:

<https://www.tutorialspoint.com/difference-between-stop-and-wait-gobackn-and-selective-repeat-protocols>

Gaurav, S. (2023) Stop and wait arq, Scaler Topics. Available at:

<https://www.scaler.com/topics/computer-network/stop-and-wait-arq/>