

Simulación de Errores en la Transmisión de Datos en un Entorno Cliente-Servidor

Leonardo Macías

Departamento de Ingeniería en Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
leodamac@espol.edu.ec

Génesis López

Departamento de Ingeniería en Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
gennalop@espol.edu.ec

Kevin Salazar

Departamento de Ingeniería en Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
kejosala@espol.edu.ec

Jorge Herrera

Departamento de Ingeniería en Computación
Escuela Superior Politécnica del Litoral
Guayaquil, Ecuador
joheniet@espol.edu.ec

Abstract

Este trabajo presenta el desarrollo y la simulación de una aplicación en Java que replica el modelo TCP/IP, con el objetivo de estudiar cómo los errores durante la transmisión de datos pueden comprometer la integridad de la información en una red. La simulación procesa el envío de un archivo de texto desde una computadora a otra, a través de un router, siguiendo las capas del modelo TCP/IP y la capa física. Se introducen errores aleatorios, tales como la pérdida de paquetes, el envío fuera de orden y la corrupción de bits, para evaluar su impacto. Si bien la simulación permite identificar estos errores, no incluye mecanismos completos para su corrección.

Keywords

Modelo TCP/IP, errores de transmisión, paquetes, simulación, encapsulación, segmentación, integridad de los datos, hilos

ACM Reference Format:

Leonardo Macías, Kevin Salazar, Génesis López, and Jorge Herrera. 2024. Simulación de Errores en la Transmisión de Datos en un Entorno Cliente-Servidor. Trabajo en clase, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador, 5 pages.

1 Introducción

En el ámbito de las redes de comunicación, la integridad y fiabilidad de la transmisión de datos son aspectos cruciales para el correcto funcionamiento de cualquier sistema. Sin embargo, durante la transmisión, los datos pueden estar sujetos a diversos errores debido a imperfecciones del medio físico o en los dispositivos intermediarios.

La motivación de este proyecto surge de la necesidad de explorar cómo estos errores pueden impactar la integridad de los datos y la entrega final. Con este fin, se desarrolló una simulación utilizando el modelo TCP/IP como marco conceptual, que permite visualizar y analizar errores en un entorno controlado. La utilidad de esta simulación radica en su capacidad de servir como herramienta tanto para la educación como para la investigación, al ofrecer una visión clara de los desafíos que enfrentan las redes al gestionar errores.

2 Marco Teórico

Las redes de computadoras permiten la comunicación entre dispositivos a través de distintos medios, facilitando el intercambio de

información y recursos. En la actualidad, la estructura de redes está basada en diversos protocolos y modelos que estandarizan cómo se deben transmitir los datos entre sistemas.

2.1 Modelo TCP/IP

Uno de los modelos más fundamentales para entender las redes de computadoras es el modelo TCP/IP (Transmission Control Protocol/Internet Protocol). Este modelo define cómo los datos se segmentan, transmiten, enrutan y reciben en una red [3], por tanto, es la base de las comunicaciones en Internet y está compuesto por cuatro capas principales:

- **Aplicación:** Esta es la capa superior del modelo TCP/IP y es la más cercana al usuario, definiendo los protocolos que usarán las aplicaciones que les proporcionan servicios [3].
- **Transporte:** Proporciona una comunicación de extremo a extremo entre aplicaciones. Aquí es donde se manejan la segmentación y el reensamblaje de los datos, así como el control de flujo y la corrección de errores[1]. Sin embargo, en condiciones de alta congestión, la pérdida de paquetes puede situarse entre el 2% y el 5% [5].
- **Red:** Se encarga del direccionamiento de los paquetes y la selección de rutas para entregarlos entre nodos en diferentes redes. La pérdida de paquetes en esta capa puede ser significativa, generalmente entre el 1% y el 2%, debido a problemas de congestión o errores en el enrutamiento [4].
- **Enlace de Datos:** Se ocupa de la transmisión de datos entre dos dispositivos en la misma red física. Maneja la detección y corrección de errores a nivel de enlace y el direccionamiento físico. La pérdida de paquetes es generalmente baja, menor al 1%, gracias a los mecanismos de corrección de errores que implementan los dispositivos de red [9].

2.2 Segmentación y Encapsulación

La segmentación y la encapsulación son procesos clave dentro del protocolo TCP/IP que facilitan la transmisión de datos. La segmentación divide grandes volúmenes de datos en fragmentos más pequeños, o segmentos, para evitar la congestión de la red y mejorar la gestión de recursos. La encapsulación, por su parte, consiste en

envolver estos segmentos con encabezados que contienen información esencial para la transmisión, como direcciones IP y números de puerto. Este proceso asegura que los datos se entreguen correctamente y en el orden adecuado [7].

2.3 Errores en la Transmisión de Datos

Los errores durante la transmisión de datos son comunes en las redes, y pueden ocurrir por diversas razones, como interferencias electromagnéticas, colisiones de paquetes, o problemas en los medios de transmisión [6].

- **Pérdida de Paquetes:** Ocurre cuando uno o más paquetes de datos no llegan a su destino, lo que puede deberse a congestión en la red o a fallos en los dispositivos de red [2].
- **Envío Fuera de Orden:** En este caso, los paquetes llegan al destino en un orden diferente al que fueron enviados. Esto puede causar problemas en la reconstrucción del mensaje original [2].
- **Corrupción de Datos:** Los datos pueden corromperse durante la transmisión, lo que lleva a la recepción de información incorrecta en el destino. Los protocolos de la capa de transporte, como TCP, utilizan checksums para detectar este tipo de errores [8].

3 Método

El presente proyecto simula el proceso de transmisión de datos entre dos computadoras, modelando el envío de un archivo de texto (.txt) utilizando Java. La simulación está diseñada para replicar el modelo TCP/IP junto a la capa física, siendo así un modelo híbrido. La visualización del proceso se realiza mediante una interfaz gráfica creada con JavaFX y una versión modificada de una librería externa [10] que permite implementar la topografía de la red.

3.1 Estructura de la Simulación

Para comprender el funcionamiento de la simulación, es esencial analizar su estructura interna. A continuación, se describen los elementos clave que constituyen su base:

- **Interfaz Gráfica de Usuario (GUI):** Utiliza JavaFX para mostrar una representación visual del entorno de red, incluyendo dos computadoras y un router interconectados. La interfaz permite al usuario cargar un archivo de texto desde una computadora, enviarlo a través de la red simulada y ver el archivo antes y después del envío. Además, muestra información sobre la transferencia de datos, como el número de paquetes enviados, recibidos y los errores inyectados.
- **Dispositivos:** Se simulan dispositivos de red como PCs y un router. Cada dispositivo implementa dos capas de la red (Capa de Enlace y Capa de Red) y la Capa Física para la transmisión real de datos. También se incluyó el elemento cable para simular la conexión física entre estos dispositivos.
- **Capas del Modelo:** Cada capa tiene un papel específico en la preparación y envío de los datos. Estas son las funciones clave asignadas a cada capa dentro de la simulación:
 - **Capa de Aplicación:** Responsable de segmentar los datos en fragmentos manejables, lo que facilita su procesamiento, y de ordenar los datos cuando los recibe de las capas inferiores.

- **Capa de Transporte:** Toma los segmentos generados por la Capa de Aplicación y los encapsula añadiendo información crucial, como los puertos de origen y destino, así como un checksum para detectar datos corruptos.
- **Capa de Red:** Añade las direcciones IP de origen y destino en el encabezado de los paquetes de datos. Además, añade un segundo checksum en el trailer.
- **Capa de Enlace de Datos:** Añade las direcciones MAC de origen y destino a los paquetes encapsulados.
- **Capa Física:** Simula la transmisión real de los bits a través de un cable virtual hacia el router.
- **Errores Simulados:** Se incluyen la pérdida de paquetes, la corrupción de datos y la entrega fuera de orden, que se introducen de manera aleatoria durante la transmisión a través de un parámetro de probabilidad que puede ser modificado, y el uso de hilos.

3.2 Flujo de la Transmisión de Datos

La simulación comienza cuando el usuario activa la aplicación mediante un botón en la interfaz gráfica, lo que configura el entorno de red. En esta etapa, se crean dos computadoras y un router, conectándolos con cables virtuales para simular una red real.

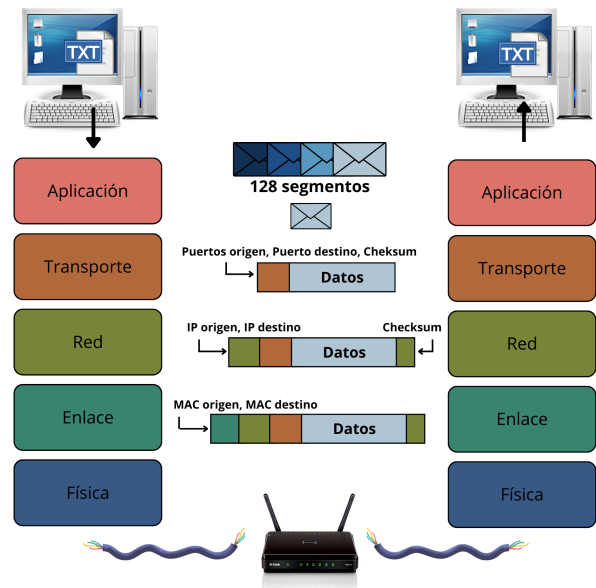


Figure 1: Representación de la Transmisión de Datos en cada Capa dentro de la Simulación

El usuario selecciona un archivo de texto para enviar desde una computadora a otra, y opcionalmente puede modificar la probabilidad de error. Sin embargo, a pesar que la implementación permite el cambio de probabilidad para cada error (pérdida y corrupción), desde la interfaz solo es posible modificar un parámetro que luego se repite para ambos. Una vez subido el archivo, este se divide en segmentos de 128, cada uno etiquetado con un identificador único y encapsulado en una estructura que incluye información de

control como el índice del segmento, el contenido y un marcador de finalización. Estos segmentos son procesados por la capa de aplicación y enviados a las capas inferiores. Cada segmento es entonces encapsulado con encabezados y tráileres necesarios para garantizar el correcto enrutamiento de los datos.

Cada capa se comunica con la capa adyacente mediante colas de datos, y los hilos permiten que estos datos se transfieran de una capa a otra simultáneamente. Los errores en la simulación se introducen según un parámetro de probabilidad para cada tipo de error y puede ser modificado por el usuario. Los hilos también simulan el envío aleatorio de datos.

Al final del proceso, los datos son transmitidos al router, que los envía a la computadora de destino. En el extremo receptor, el proceso se invierte, los datos recibidos se almacenan y se muestran en la interfaz gráfica, que indica el progreso de la recepción y cualquier error detectado. También existe la opción de poder visualizar gráficas comparando la integridad de los paquetes obtenida con la integridad teórica calculada de la probabilidad de error.

3.3 Justificación de los Criterios de Diseño

El diseño de este proyecto se basa en la necesidad de una simulación integral que represente tanto las capas estándar del modelo TCP/IP como la capa física, con el objetivo de evaluar de manera realista la transmisión de datos. Se eligió segmentar el archivo en partes de 128 por su simplicidad y manejabilidad, aunque en la práctica se podrían usar tamaños mayores, el establecido es suficiente para fines educativos.

Además, se emplearon hilos para gestionar la transmisión de datos, lo que es crucial para permitir que las capas del modelo operen en paralelo, y simular el envío fuera de orden.

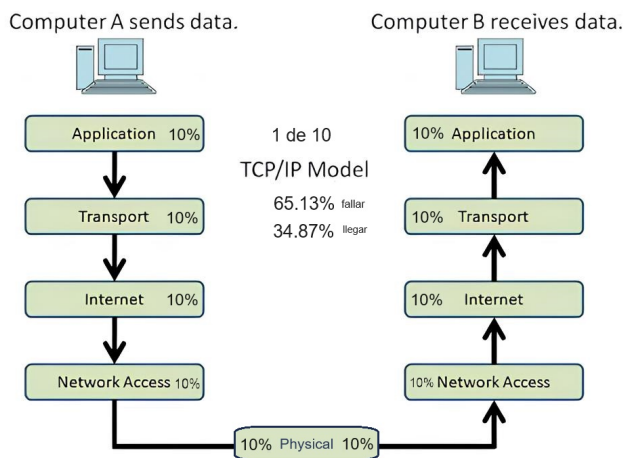


Figure 2: Ejemplificación de la Inyección de Errores

Finalmente, la inyección de errores, como la pérdida de paquetes y la corrupción de datos, se basa en parámetros probabilísticos modificables para una simulación controlada e interactiva. Esta probabilidad se aplica a cada capa individualmente, de modo que para calcular la probabilidad total de llegada, se debe multiplicar este parámetro por cada capa que atraviesan los datos durante la

transmisión, de la siguiente manera:

$$\text{Probabilidad de Llegada} = (1 - \text{Probabilidad de Error})^{10}$$

Este enfoque modular ayuda a descomponer y analizar el impacto de errores en cada capa, proporcionando una comprensión más detallada del proceso de transmisión.

4 Experimentos

Para evaluar la simulación, se realizaron experimentos enviando archivos de diferentes tamaños y con distintas configuraciones de error. Cada experimento se repitió varias veces para obtener resultados consistentes y representativos del comportamiento. A continuación, se detallan los experimentos planteados y sus respectivos parámetros:

- **Experimento 0. Sin errores:** Este experimento evalúa la transmisión de mensajes sin la introducción de errores. Su objetivo es obtener una línea base para comparar la efectividad de la simulación en un entorno ideal sin pérdida ni corrupción.
— **Porcentaje de error a evaluar:** 0%
- **Experimento 1. Impacto de la probabilidad de error (Pérdida y Corrupción de Paquetes):** Este experimento agrupa las simulaciones con diferentes probabilidades de pérdida y corrupción de paquetes para observar el efecto acumulativo de pequeños errores en la transmisión.
— **Porcentaje de error a evaluar:** 1%, 5%, 10%, 20%, 50%
- **Experimento 2. Evaluación del tamaño del mensaje:** Este experimento examina cómo el tamaño del mensaje influye en la integridad del mensaje recibido, lo que permitirá analizar si tiene relación con la degradación de la transmisión.
— **Tamaños a evaluar:** 4KB, 50KB, 100KB

5 Resultados

Después de realizar los experimentos de simulación, tal como se muestra en la Figura 3, se obtuvieron resultados que permiten extraer diversas conclusiones importantes.

- En el primer experimento, se observó que todos los mensajes fueron recibidos de manera íntegra, independientemente del tamaño del mensaje, lo que confirma que la simulación funciona correctamente en condiciones ideales.
- El segundo experimento, que incluyó variaciones en las probabilidades de error, mostró un impacto significativo en la integridad de los mensajes recibidos, pues disminuía progresivamente a medida que aumentaban las probabilidades de error. Este comportamiento es en cierta medida esperado y demuestra que el sistema es sensible a errores incluso en tasas bajas. Lo que valdría destacar es que, a partir de un parámetro de 50% de errores, los mensajes apenas conservaron su integridad.
- El tercer experimento reveló que los mensajes más grandes son más susceptibles a la pérdida de integridad en comparación con los mensajes más pequeños. Este resultado sugiere que el manejo grandes volúmenes de datos tiene mayor riesgo de sufrir pérdidas importantes. Por lo tanto, la fragmentación de mensajes grandes en segmentos más pequeños podría ser una estrategia efectiva para mitigar su pérdida.

No. Exp	Porcentaje de Pérdida de Paquetes en cada capa (%)	Porcentaje de Corrupción de Paquetes en cada capa (%)	Tamaño del mensaje	Datos Enviados	Datos Recibidos	Paquetes enviados	Paquetes recibidos	Paquetes perdidos	Paquetes corruptos	Mensaje recibido Práctico (%)	Mensaje recibido Teórico (%)
0	0	0	4KB	3840	3840	30	30	0	0	100%	100%
			50KB	51200	51200	400	400	0	0	100%	
			100KB	51200	51200	400	400	0	0	100%	
1/2	1	1	4KB	3840	3584	30	28	2	5	93.33%	90.44%
			50KB	51200	47872	400	374	26	20	93.50%	
			100KB	101888	95104	796	743	53	65	93.34%	
	5	5	4KB	3840	2816	30	22	8	6	73.33%	59.87%
			50KB	51200	35200	400	275	125	133	68.75%	
			100KB	101888	66304	796	518	278	266	65.08%	
	10	10	4KB	3840	2048	30	16	14	18	53%	34.87%
			50KB	51200	17792	400	139	261	162	34.75%	
			100KB	101888	43008	796	336	460	414	42.21%	
	20	20	4KB	3840	1280	30	3	27	23	10%	10.74%
			50KB	51200	9088	400	71	329	292	17.75%	
			100KB	101888	15488	796	121	675	559	15.20%	
	50	50	4KB	3840	0	30	0	30	23	0.75%	0.1%
			50KB	51200	384	400	3	397	230	0.75%	
			100KB	101888	128	796	1	795	401	0.13%	

Figure 3: Resultados de los Experimentos

Además, al comparar los resultados teóricos y prácticos, se observó que, aunque las predicciones teóricas ofrecen una aproximación útil para prever el comportamiento general de la simulación, las condiciones reales pueden llevar a diferencias significativas en los resultados, especialmente en situaciones de alta probabilidad de error. Esta discrepancia subraya la necesidad de realizar pruebas prácticas para ajustar los modelos teóricos.

6 Conclusiones

La simulación implementada es efectiva para fines educativos, puesto que permite interactuar con los errores inyectados y refleja cómo impactan a la integridad de los archivos enviados. Sin embargo, según los resultados de los experimentos, se evidencia la importancia de implementar mecanismos adicionales de tolerancia a fallos, como protocolos de retransmisión o esquemas avanzados de corrección de errores, para mejorar la confiabilidad en redes con alta probabilidad de pérdida y corrupción de paquetes.

7 Limitaciones

El proyecto presenta algunas limitaciones que podrían ser abordadas en futuras versiones.

En primer lugar, no se ha implementado ningún protocolo estándar del modelo TCP/IP, lo que limita la simulación a un enfoque más

básico. Aunque se emplea un mecanismo de checksum para la verificación de integridad de los datos, este podría ser más robusto como SHA-256 (Secure Hash Algorithm 256) o MD5 (Message-Digest Algorithm 5). Además, la simulación carece de medidas de seguridad como la encriptación de datos, lo cual sería esencial para simular una transmisión más realista en escenarios donde la seguridad es crítica.

Si bien la simulación permite la representación de errores, no incluye la capacidad de corregirlos, lo que podría añadirse como una mejora significativa. Asimismo, la red simulada permite la escalabilidad, pero actualmente no cuenta con esas funcionalidades lo que impide la inclusión de más computadoras en la red. Finalmente, la segmentación actual se realiza a nivel de caracteres, cuando una implementación a nivel de bits podría ofrecer mayor precisión y similitud con el comportamiento de redes reales.

8 Repositorio de la Simulación

<https://github.com/leodamac/Proyecto-Redes.git>

References

- [1] M Fernández Barcell. 2014. *Protocolo TCP/IP*. Retrieved Septiembre 02, 2024 from https://rodin.uca.es/bitstream/handle/10498/16833/temaIII_tcpip.pdf
- [2] Jacquenet C. Boucadair, M. [n. d.]. *IP Address Management Principles and Practice*. ([n. d.]).

- [3] J. A. Castillo. 2020. *Protocolo TCP/IP - ¿Qué es y cómo funciona?* Retrieved Septiembre 02, 2024 from <https://repositorio.usam.ac.cr/xmlui/handle/11506/localhost/xmlui/handle/11506/2181>
- [4] Douglas Comer. 1998. *Internetworking with TCP/IP* (5th. ed.). Pearson.
- [5] D. E. Comer. 2013. *Internetworking with TCP/IP: Principles, Protocols, and Architecture* (6th. ed.). Pearson.
- [6] CSFG. s.f.. *Transport layer protocols - TCP, UDP*. Retrieved Septiembre 02, 2024 from <https://www.csfieldguide.org.nz/en/chapters/network-communication-protocols/transport-layer-protocols/>
- [7] Jacobs D. 2022. *Intro to encapsulation and decapsulation in networking*. Retrieved Septiembre 01, 2024 from <https://www.techtarget.com/searchnetworking/tip/Intro-to-encapsulation-and-decapsulation-in-networking>
- [8] FaterCapital. s.f.. *Tipos De Corrupción De Datos*. Retrieved Septiembre 04, 2024 from <https://fatercapital.com/es/tema/tipos-de-corrupci%C3%B3n-de-datos.html>
- [9] Keith W. Ross James F. Kurose. 2013. *Computer Networking: a Top Down Approach* (6th. ed.). Pearson.
- [10] sirolf2009. 2018. *FXGraph*. Retrieved Septiembre 03, 2024 from <https://github.com/sirolf2009/fxgraph>