UNIVERSIDAD DEL VALLE DE GUATEMALA
Redes - CC3067
Sección 11
Ing. Miguel Novella Linares

Laboratorio 2 - Parte 2

Detección y corrección de errores

José Pablo Orellana 21970
Diego Alberto Leiva 21752

GUATEMALA, 01 de agosto del 2024

Descripción de la práctica y metodología utilizada

En este laboratorio se desarrollaron programas con el objetivo de poder transmitir y recibir mensajes utilizando dos algoritmos de detección y corrección de errores. Hamming y CRC-32 correspondientemente. Los programas se estructuraron en una arquitectura de capas que incluyen las siguientes.

- **Aplicación:**
  - Solicitar mensaje: El emisor solicita al usuario el mensaje que desea enviar. También solicita el algoritmo de detección y corrección de errores que se utilizará (Hamming o CRC-32).
  - Mostrar mensaje: El receptor muestra el mensaje decodificado. Si se detectaron errores que no pudieron ser corregidos, se debe indicar con un mensaje de error.

- **Presentación:**
  - Codificar mensaje: La capa de presentación convierte el mensaje a su representación binaria ASCII. Por ejemplo, el carácter 'A' se convierte a '01000001'.
  - Decodificar mensaje: Convierte los bits binarios de vuelta a caracteres ASCII si no se detectan errores. Si se detecta un error, se notifica a la capa de aplicación.

- **Enlace:**
  - Calcular integridad: Utilizando el algoritmo seleccionado (Hamming o CRC-32), se calcula la información de integridad y se concatena al mensaje binario original.
  - Verificar integridad: En el receptor, se recalcula la información de integridad y se compara con la proporcionada por el emisor para detectar posibles errores. Se informa a la capa de presentación sobre los resultados de la verificación.
  - Corregir mensaje: Si el algoritmo es capaz de corregir los errores detectados, esta capa realiza la corrección.

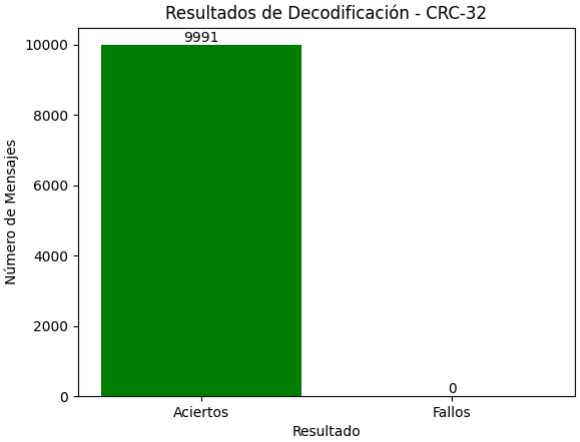
- **Ruido:**
  - Aplicar ruido: Para simular un canal no confiable, se introduce ruido en la trama binaria generada por la capa de enlace. La probabilidad de que cada bit se voltee está determinada por un parámetro de error (por ejemplo, 1/100).

- **Transmisión:**
  - Enviar información: Utilizando sockets TCP, el emisor envía la trama de información al receptor.
  - Recibir información: El receptor está en modo "server", escuchando en un puerto específico para recibir la trama de información.

Resultados

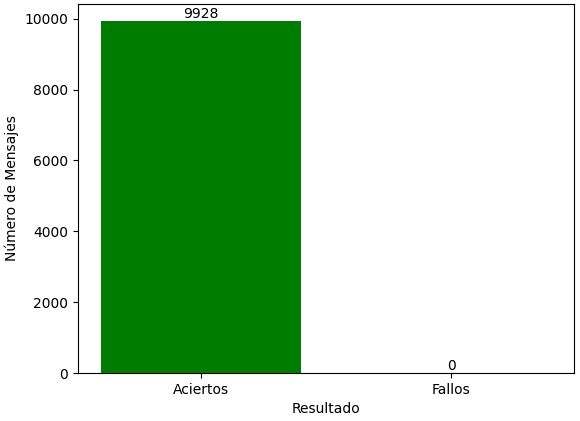
CRC-32

Cadenas de longitud 1 con probabilidad 0.001 de fallo

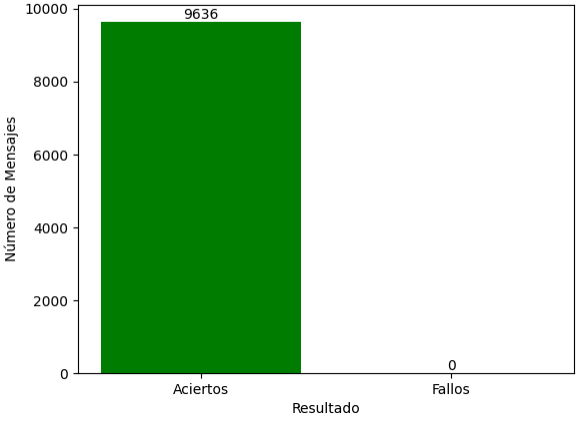


Cadenas de longitud 30 con probabilidad 0.001 de fallo

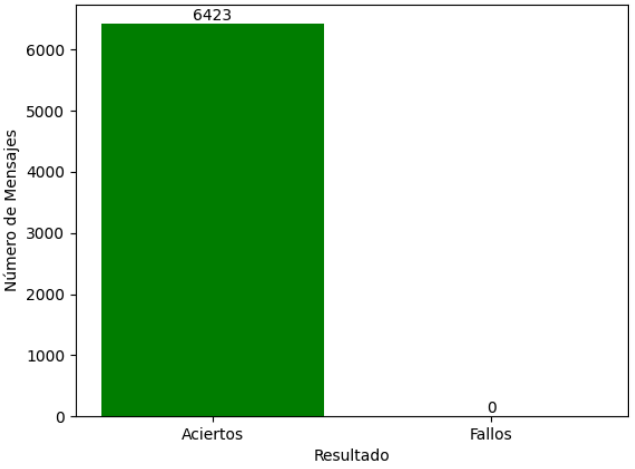
Cadenas de longitud 5 con probabilidad 0.01 de fallo



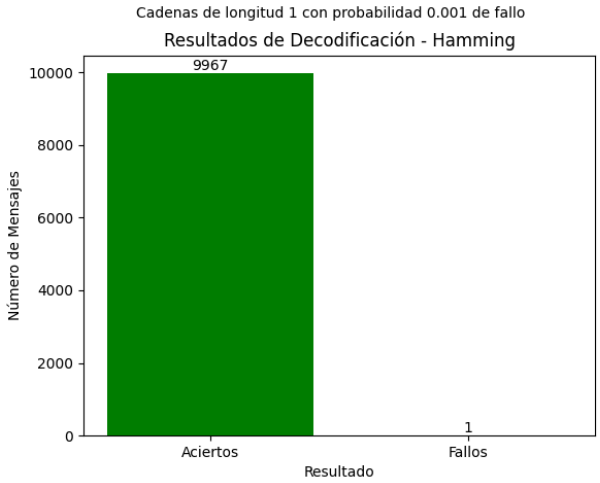
Cadenas de longitud 10 con probabilidad 0.1 de fallo

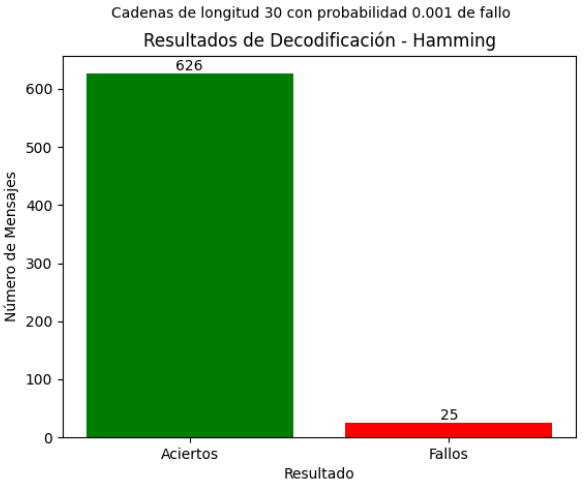


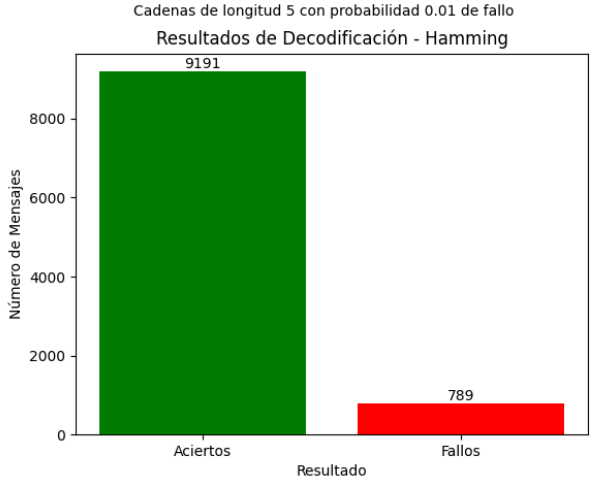
Cadenas de longitud 25 con probabilidad 0.005 de fallo

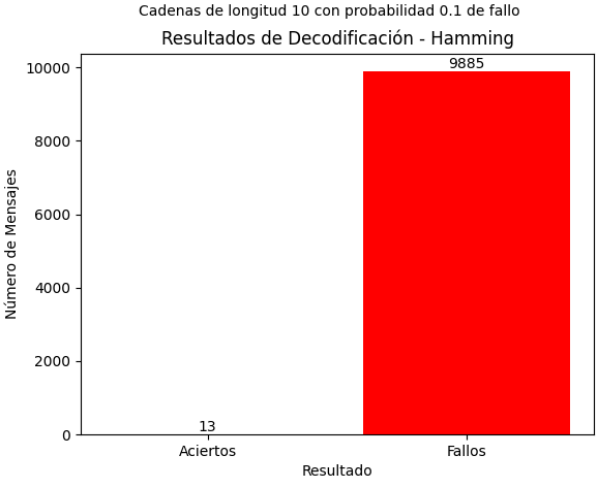


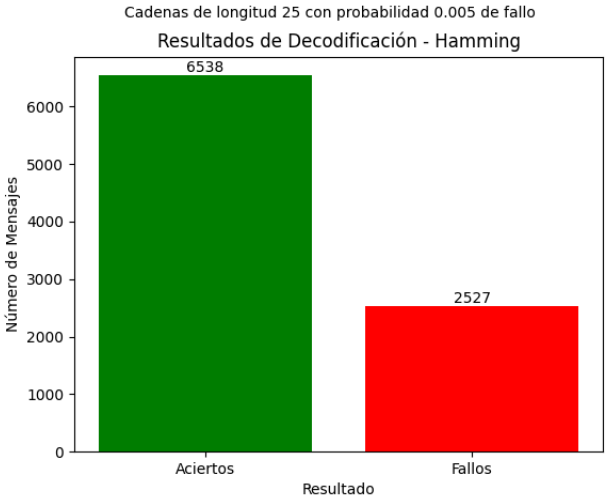
Hamming


Cadenas de longitud 1 con probabilidad 0.001 de fallo
Resultados de Decodificación - Hamming
Número de Mensajes
Resultado
Aciertos
Fallos
9967
1


Cadenas de longitud 30 con probabilidad 0.001 de fallo
Resultados de Decodificación - Hamming
Número de Mensajes
Resultado
Aciertos
Fallos
626
25


Cadenas de longitud 5 con probabilidad 0.01 de fallo
Resultados de Decodificación - Hamming
Número de Mensajes
Resultado
Aciertos
Fallos
9191
789


Cadenas de longitud 10 con probabilidad 0.1 de fallo
Resultados de Decodificación - Hamming
Número de Mensajes
Resultado
Aciertos
Fallos
13
9885


Cadenas de longitud 25 con probabilidad 0.005 de fallo
Resultados de Decodificación - Hamming
Número de Mensajes
Resultado
Aciertos
Fallos
6538
2527

Discusión

Análisis de Resultados de CRC-32
Pruebas con diferentes longitudes de cadena y probabilidades de fallo:

- **Cadena de longitud 1 con probabilidad de fallo 0.001:**
  - **Resultados:** 9991 aciertos, 0 fallos.
  - **Interpretación:** La probabilidad de fallo baja y la corta longitud de la cadena resultaron en una detección de errores muy efectiva. CRC-32 mostró un rendimiento robusto, sin fallos detectados en 10,000 mensajes.

- **Cadena de longitud 30 con probabilidad de fallo 0.001:**
  - **Resultados:** 734 aciertos, 0 fallos.
  - **Interpretación:** Con una longitud de cadena mayor, CRC-32 siguió demostrando su alta capacidad de detección de errores. La ausencia de fallos en este escenario indica que CRC-32 puede manejar adecuadamente errores incluso en mensajes más largos con una baja tasa de error.

- **Cadena de longitud 5 con probabilidad de fallo 0.01:**
  - **Resultados:** 9928 aciertos, 0 fallos.
  - **Interpretación:** Incrementar la probabilidad de fallo a 0.01 y mantener la longitud de cadena en 5 bits, CRC-32 mantuvo su capacidad de detección de errores, sin presentar fallos en 10,000 pruebas.

Tomando en cuenta que en las dos gráficas restantes se obtuvieron resultados similares y en términos generales, podemos decir que el CRC-32 mostró una alta eficiencia en cuanto a la detección de errores, independientemente de la longitud de la cadena y con diferentes probabilidades de fallo. La robustez que presenta en la detección de errores múltiples hace que sea una buena elección para la verificación de la integridad de datos en transmisiones largas.

Análisis de Resultados de Hamming
Pruebas con diferentes longitudes de cadena y probabilidades de fallo:

- **Cadena de longitud 30 con probabilidad de fallo 0.001:**
  - **Resultados:** 626 aciertos, 25 fallos.
  - **Interpretación:** Hamming mostró cierta debilidad frente a errores en cadenas más largas. Aunque detectó y corrigió muchos errores, algunos fallos fueron inevitables debido a la limitación del algoritmo para corregir solo un error por bloque.

- **Cadena de longitud 5 con probabilidad de fallo 0.01:**
    - **Resultados:** 9191 aciertos, 789 fallos.
    - **Interpretación:** La mayor probabilidad de fallo incrementó el número de fallos detectados. Hamming es menos efectivo con tasas de error más altas y cadenas más largas debido a su incapacidad para manejar múltiples errores en un solo bloque.

- **Cadena de longitud 10 con probabilidad de fallo 0.1:**
    - **Resultados:** 13 aciertos, 9885 fallos.
    - **Interpretación:** Con una alta probabilidad de fallo, Hamming demostró ser inadecuado. La mayoría de los mensajes presentaron errores que no pudieron ser corregidos, resaltando su limitación en ambientes con alta tasa de errores.

- **Cadena de longitud 25 con probabilidad de fallo 0.005:**
    - **Resultados:** 6538 aciertos, 2527 fallos.
    - **Interpretación:** A pesar de una tasa de fallo relativamente baja, la longitud mayor de la cadena afectó significativamente la eficacia de Hamming. Más del 27% de los mensajes presentaron fallos que no pudieron ser corregidos.

En base a los resultados presentados por las gráficas podemos decir que Hamming es capaz de corregir errores de un solo bit y es adecuado para mensajes cortos con baja probabilidad de error. Presenta limitaciones significativas con mensajes largos y altas tasas de error, donde múltiples errores pueden ocurrir.

Rendimiento y Eficiencia:

Hamming es ideal para entornos con baja probabilidad de errores donde la corrección de los mismos es esencial y la simplicidad es una ventaja. Su rendimiento decae significativamente conforme crece la longitud del mensaje y la tasa de error. Por otro lado CRC-32 es preferible en escenarios con alta probabilidad de error y donde se prioriza la detección rápida de los errores por encima de la corrección, la robustez y flexibilidad lo hacen adecuado en aplicaciones que nos requieran alta demanda en cuanto a la integridad de datos.

Conclusiones

- Eficacia de Algoritmos: CRC-32 demostró una capacidad superior para detectar errores en una variedad de condiciones de prueba, mientras que Hamming mostró limitaciones significativas en la corrección de errores en condiciones de alta tasa de fallo.
- Robustez de CRC-32: La robustez de CRC-32 para detectar errores múltiples lo hace una opción más confiable para aplicaciones donde la integridad de los datos es crítica, aunque no proporciona mecanismos de corrección.
- Limitaciones de Hamming: La eficacia de Hamming está restringida a entornos con bajas tasas de error y mensajes cortos. Su capacidad de corrección de errores se limita a errores de un solo bit, lo que resulta inadecuado en condiciones de transmisión más adversas.

Repositorio: https://github.com/LeivaDiego/Redes-Lab2

Referencias

Forouzan, B. A. (2007). Data Communications and Networking. McGraw-Hill.

Stallings, W. (2007). Data and Computer Communications. Pearson Prentice Hall.