Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3067 Redes

Laboratorio 2 - Segunda parte

Esquemas de detección y corrección de errores

1 Antecedentes

Los errores de transmisión suceden en toda comunicación y es parte de los retos al momento de implementar este tipo de sistemas el manejar adecuadamente las fallas que puedan ocurrir. Por lo tanto, a lo largo de la evolución del Internet se han desarrollado distintos mecanismos que sirven tanto para la detección como para la corrección de errores.

2 Objetivos

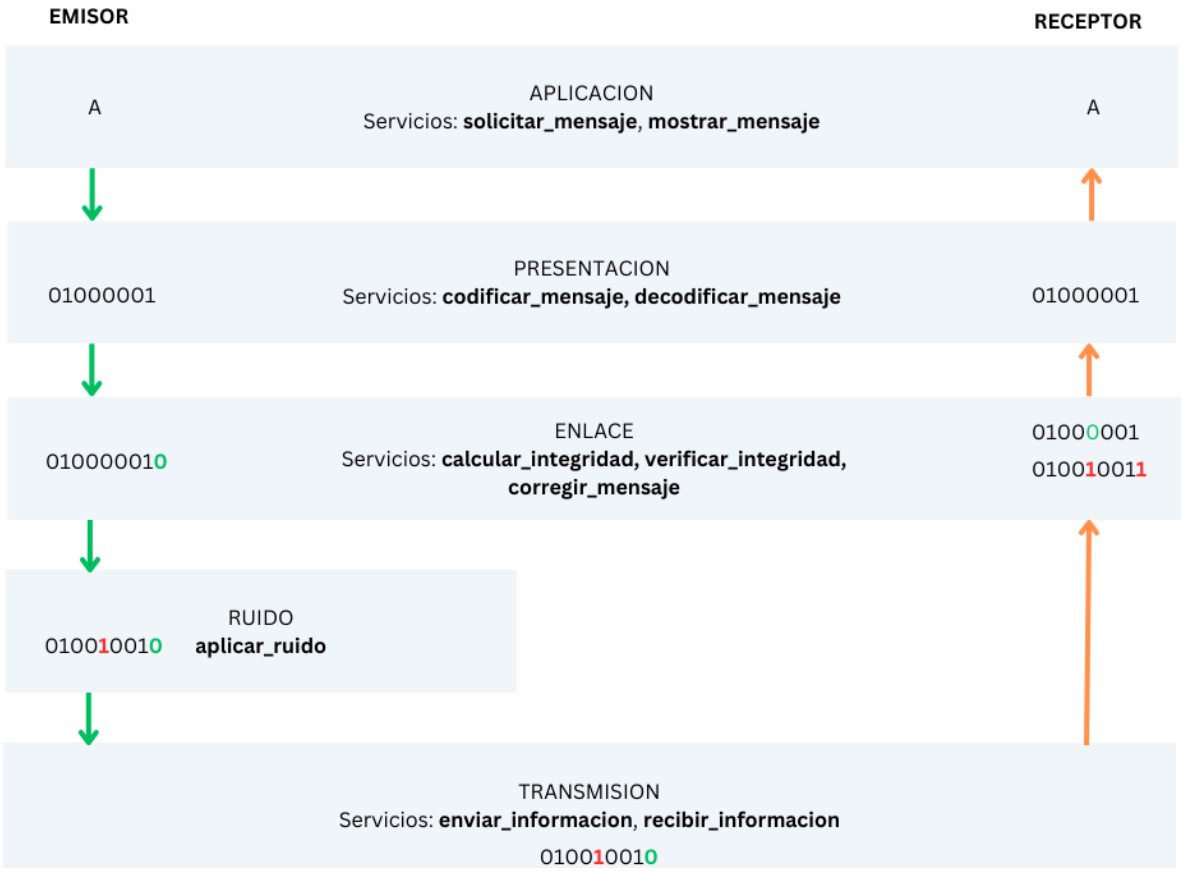
- Comprender el funcionamiento de un modelo de capas y sus servicios
- Implementar sockets para la transmisión de información
- Experimentar la transmisión de información expuesta a un canal no confiable
- Analizar el funcionamiento de los esquemas de detección y corrección

3 Desarrollo

En la primera parte del laboratorio se implementaron por lo menos dos algoritmos, uno de detección y otro de corrección. En esta segunda parte se desarrollará una aplicación para la transmisión y recepción de mensajes, en base a una arquitectura de capas con distintos servicios. Los mismos grupos de la primera parte trabajarán en la segunda parte del laboratorio.

a. Arquitectura de capas

La arquitectura cuenta con las siguientes capas y servicios:



Descripción de los servicios:

1. APLICACIÓN
    - **Solicitar mensaje**: solicita el texto a enviar al emisor. También solicita el algoritmo a utilizar para comprobar la integridad.
    - **Mostrar mensaje**: muestra el mensaje al receptor (sin errores). Si se detectaron errores y no fue posible corregirlos, se debe indicar con un mensaje de error.
2. PRESENTACIÓN
    - **Codificar mensaje**: codifica cada caracter individual en ASCII binario. Por ejemplo, para el carácter A el código binario ASCII es 01000001.
    - **Decodificar mensaje**: si no se detectan errores, se debe decodificar el ASCII binario a los caracteres correspondientes. Si se detecta algún error, se debe indicar a la capa de aplicación.
3. ENLACE
    - **Calcular integridad**: utilizando el algoritmo indicado en el servicio *solicitar_mensaje*, calcular la información de integridad. Concatenar la información al mensaje en binario original.

- **Verificar integridad**: el algoritmo seleccionado debe calcular la información del lado del receptor y compararla contra la proporcionada por el emisor para detectar posibles errores. Debe indicar esto a la capa de presentación. Aquí es donde se deben integrar los algoritmos implementados en la primera parte del laboratorio.
- **Corregir mensaje**: si el algoritmo tiene la capacidad de corregir los errores detectados debe corregirlos.

4. RUIDO
   - **Aplicar ruido**: el ruido no es una capa como tal, pero a fin de simular posibles interferencias se tratará como una capa del lado del emisor, y se aplicará ruido a la trama proporcionada por la capa de enlace. La forma de determinar si cada bit se voltea (bit = !bit) se basará en cierta probabilidad expresada en errores por bits transmitidos (por ejemplo, 1/100=0.01 es un error por cada 100 bits). Note que ¡los bits de paridad también son propensos al ruido!
5. TRANSMISIÓN
   - **Enviar información**: envía la trama de información a través de sockets mediante el puerto elegido.
   - **Recibir información**: recibe la trama de información a través de sockets mediante el puerto elegido. El receptor siempre debe estar "escuchando" en el puerto elegido a la espera de recibir datos (i.e.: a modo "server" y el emisor a modo "client").

La aplicación del lado del emisor debe estar implementada en el mismo lenguaje de programación en que fueron implementados los algoritmos de detección y corrección para la versión del emisor.

La aplicación del lado del receptor debe estar implementada en el mismo lenguaje de programación en que fueron implementados los algoritmos de detección y corrección para la versión del receptor.

b. Pruebas

Utilizando los algoritmos implementados realizar **varias** pruebas (10k, 100k+) de envío y recepción, donde se logre evidenciar el funcionamiento de los algoritmos. Busque automatizar el proceso de pruebas, análisis estadístico y despliegue de gráficas. Para estas pruebas cada grupo deberá de elegir cómo las realizará y generar gráficas que reflejen estos datos y respalden su discusión y conclusiones. La cantidad y contenido de las gráficas queda a discreción del grupo, no obstante, deben de ser realizadas variando el tamaño de las cadenas enviadas, la probabilidad de error, el algoritmo utilizado y la redundancia/taza de código para que el algoritmo sea efectivo (ej: *r* bits de paridad, en Hamming, o taza 2:1, 3:1, r:1 en convolucional).

Algunas preguntas que pueden ayudar a la discusión son:

- ¿Qué algoritmo tuvo un mejor funcionamiento y por qué?
- ¿Qué algoritmo es más flexible para aceptar mayores tasas de errores y por qué?
- ¿Cuándo es mejor utilizar un algoritmo de detección de errores en lugar de uno de corrección de errores y por qué?

4 Desarrollo

Al finalizar la actividad debe de realizarse un reporte **grupal** donde se incluyan las siguientes secciones:

1. Nombres y carnés
2. Título de la práctica
3. Descripción de la práctica y metodología utilizada
4. Resultados
5. Discusión
6. Comentario grupal sobre el tema (opcional)
7. Conclusiones
8. Citas y Referencias

c. Rúbrica de evaluación

Elemento	Excelente (1-0.9 pt.)	Aceptable con mejoras (0.9-0.5 pts.)	Inaceptable (0.5-0 pts)
Implementación	La codificación y decodificación, y el uso de sockets es apropiado. Los algoritmos funcionan bien bajo la simulación de errores.	La comunicación entre sockets falla en ocasiones, la decodificación no interpreta en ocasiones la trama devuelta por la capa de enlace.	Los sockets, codificación/decodificación, ruido y los algoritmos no fueron integrados, no compilan o la cantidad de fallas es muy alta.
Elemento	**Excelente (0.5)**	**Aceptable con mejoras (0.25 pts.)**	**Inaceptable (0 pts)**
Resultados/Discusión	Las explicaciones reflejan efectivamente lo realizado y aprendido en el laboratorio.	Las explicaciones omiten detalles importantes, pero en general expresan la idea central.	No es posible entender lo realizado en el laboratorio a partir de las explicaciones.
Formato del reporte/Calidad de conclusiones	Las conclusiones son reflejo del análisis preparado en la discusión. El reporte está ordenado, legible y con buen formato.	Las conclusiones no se encuentran totalmente fundamentadas en la discusión. El reporte cuenta con un formato y legibilidad aceptable.	Se concluye elementos de la teoría o no existe referencia alguna en la discusión. El reporte no sigue un formato y no es legible.

** La asistencia y participación es obligatoria, una ausencia injustificada anula la nota del laboratorio

** Se debe cuidar el formato y ortografía del reporte

d. Entregar en Canvas

- **Reporte** en formato PDF
- **Todo el código** involucrado y cualquier elemento para su compilación (makefiles, etc).
- **Link a su repositorio**, el cual debe ser privado hasta el momento de entrega
    - El **repositorio también debe tener el Reporte** PDF.