



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

El ruido y 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

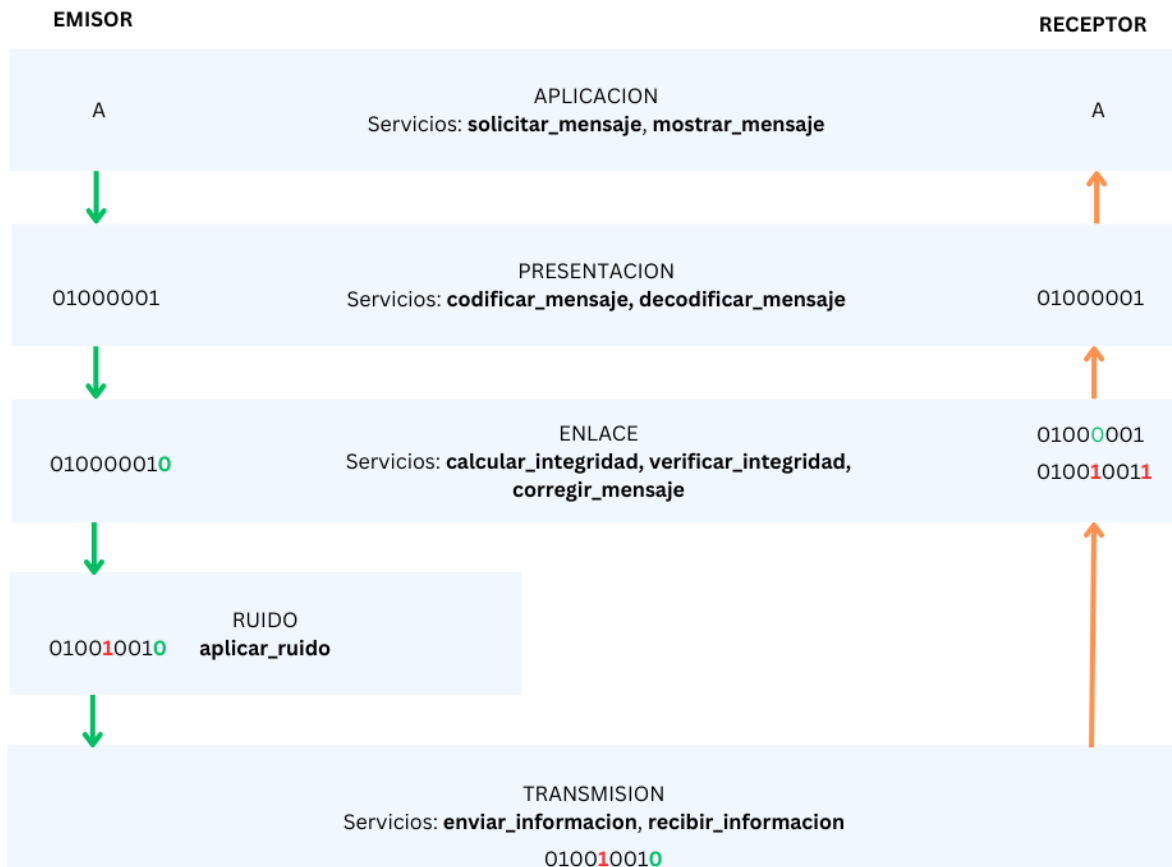
- Comprender el funcionamiento de un modelo de capas y sus servicios
- Implementar sockets para la transmisión de información
- Experimentar la transmisión de información expuesta a un canal no confiable
- Analizar el funcionamiento de los esquemas de detección y corrección

#### 3 Desarrollo

En la primera parte del laboratorio se implementó un algoritmo de corrección. En esta segunda parte se implementará un algoritmo de detección y 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. PRESENTACION

- **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 error, se debe indicar de alguna forma 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 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 sufre un flip se basará en cierta probabilidad expresada en errores por bits transmitidos (por ejemplo, 1/100 es un error por cada 100 bits). Esta tasa debe ser solicitada el momento de enviar un mensaje. Recuerde que la información de redundancia (ej: bits de paridad) también están sujetos al ruido.

#### 5. TRANSMISION

- **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 data.

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 pruebas de envío y recepción, donde se logre evidenciar el funcionamiento de los algoritmos. Para estas pruebas cada grupo deberá de elegir cómo las realizará y generar gráficas que respalden estos datos. 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 el *overhead* (cantidad de información extra que debe ser enviada como redundancia para que el algoritmo sea efectivo).

Algunas preguntas que pueden ayudar a la discusión son:

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

## 4 Desarrollo

Entrega inicial (**hoy antes de las 21:30 horas**)

- Implementación de las capas del lado del Emisor (Aplicación hasta ruido): mostrar con screenshots como su aplicación solicita un mensaje, y este es codificado en ASCII, se le añade los bits de paridad/check, y se le aplica ruido al mensaje.

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
4. Resultados
5. Discusión
6. Comentario grupal sobre el tema (opcional)
7. Conclusiones
8. Citas y Referencias

### c. Rúbrica de evaluación

- Entrega inicial: 20%\*\*
- Implementación de algoritmo de detección, y comunicación con receptor: 30%
- Reporte: 50%
  - Formato: 5%
  - Pruebas: 15%
  - Discusión: 20%
  - Conclusiones: 10%

\*\* 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

- **Todo el código** involucrado y cualquier elemento para su compilación (makefiles, etc).
- **Link a su repositorio**, el cual si es privado debe otorgar permisos de acceso al profesor y auxiliares
  - El **repositorio debe tener el Reporte** en formato PDF.