Universidad del Valle de Guatemala

Facultad de ingeniería

Redes

Catedrático: Carlos Boix

Laboratorio 2 - Esquemas de detección y corrección

Christian Echeverría 221441

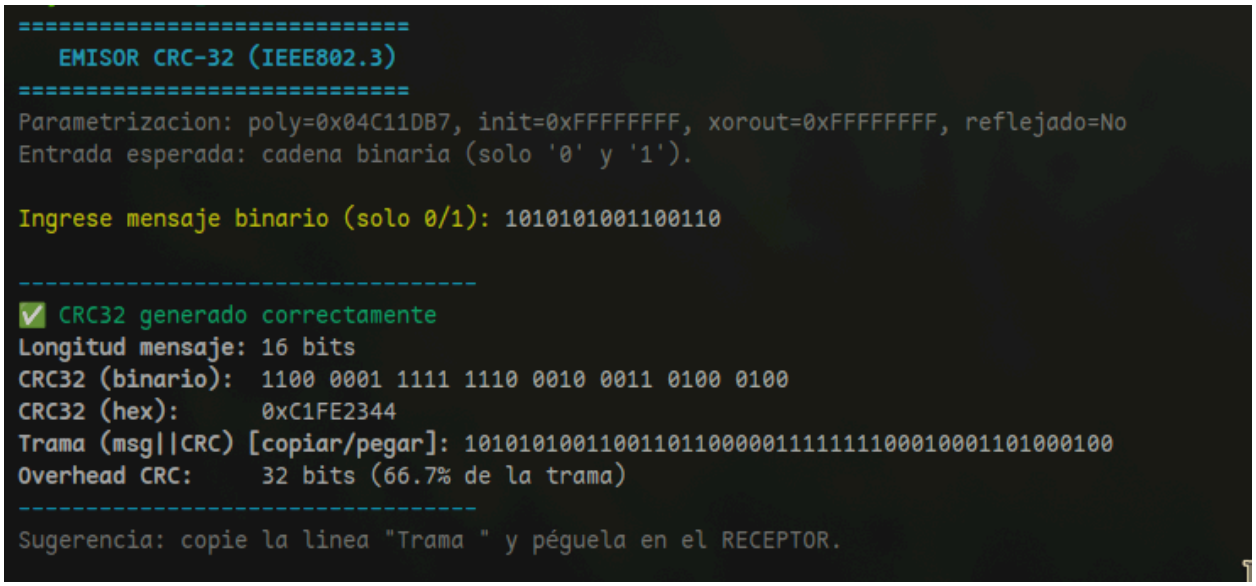
Gabriel Alberto Paz González 221088

Primera parte:

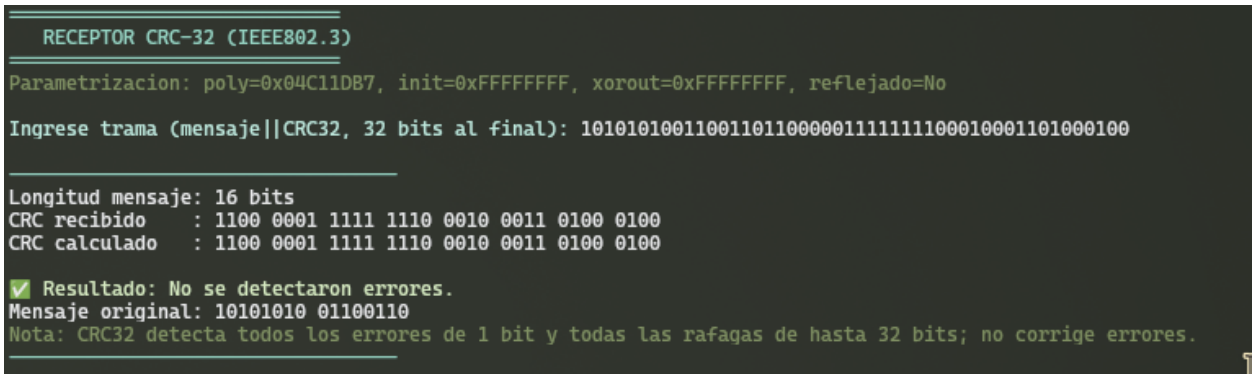
CRC32

Mensaje Correcto:

Emisor



Receptor



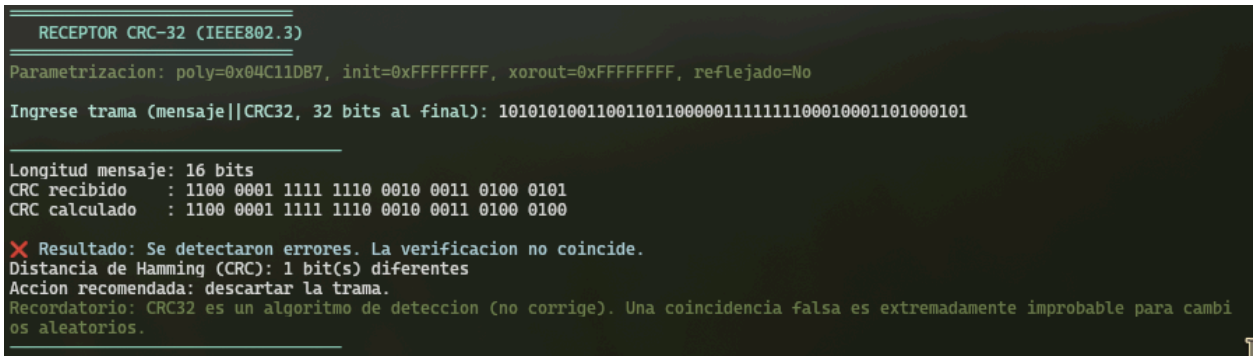
Nota: Si que detecta que ningun bit fue cambiado.

Mensaje con bit cambiado:

Emisor

10101010011001101100000111111110001000110100010 1 (ultimo bit cambiado)

Receptor



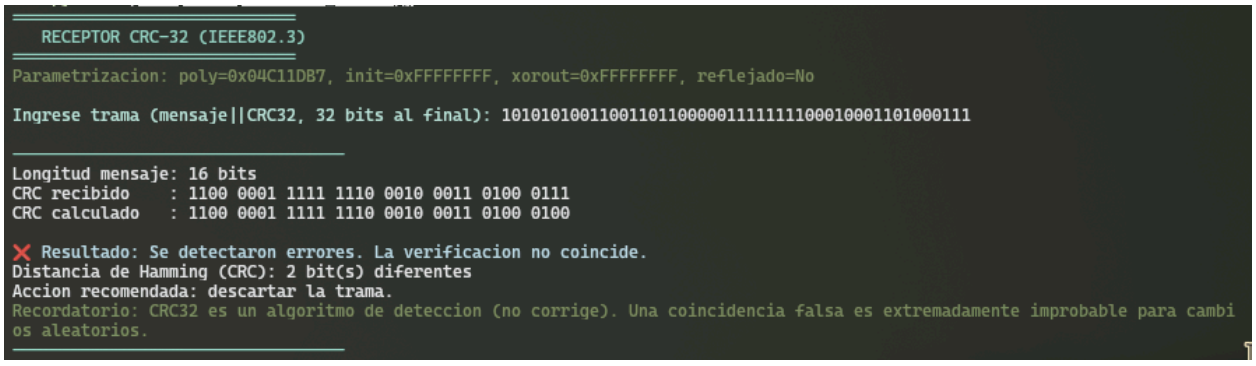
Nota: el bit si fue detectado

Mensaje con 2 bits cambiados:

Emisor

10101010011001101100000111111110001000110100011 1 (último y penúltimos bits cambiados)

Receptor

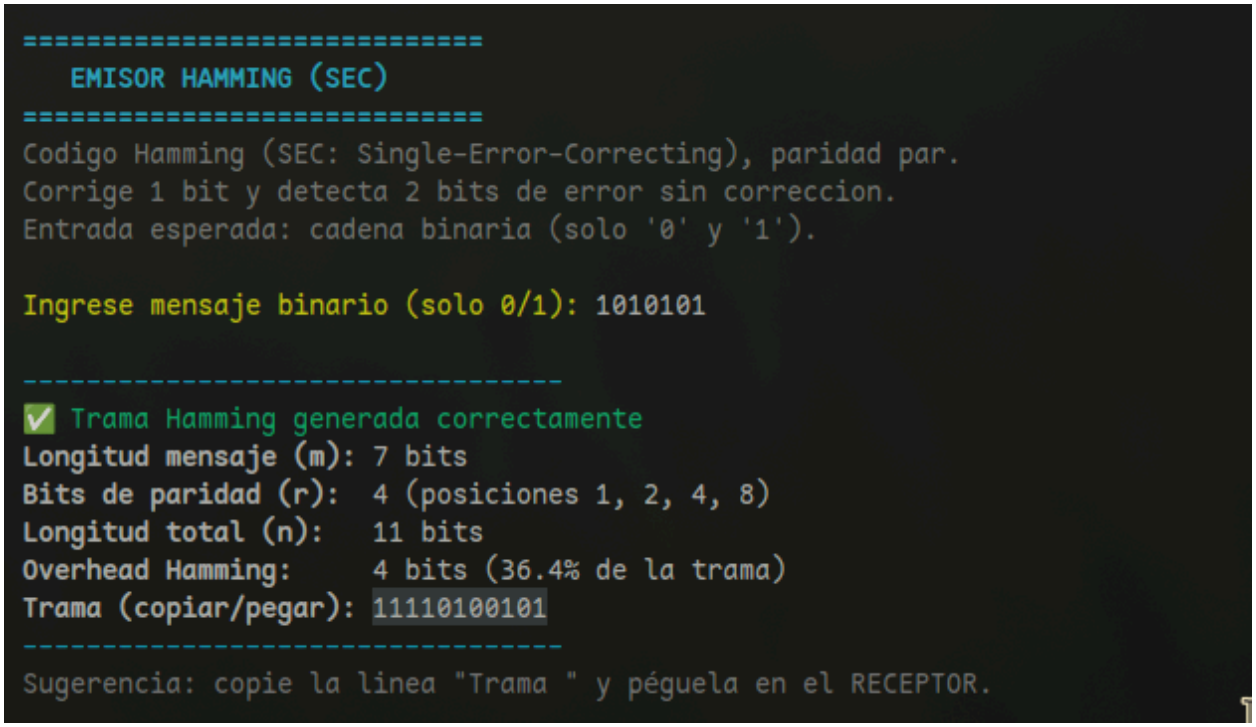


Nota: si detecto los dos bits cambiados

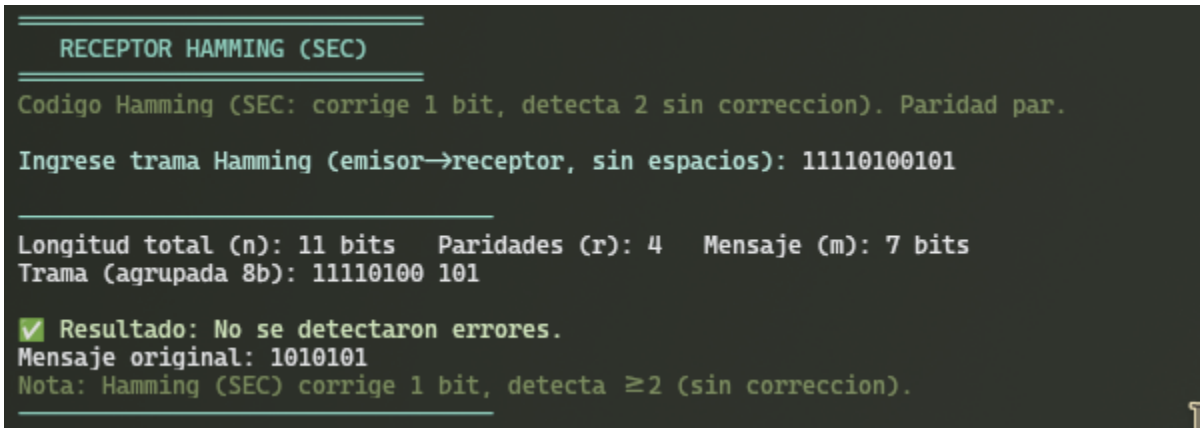
Hamming

Mensaje correcto

Emisor



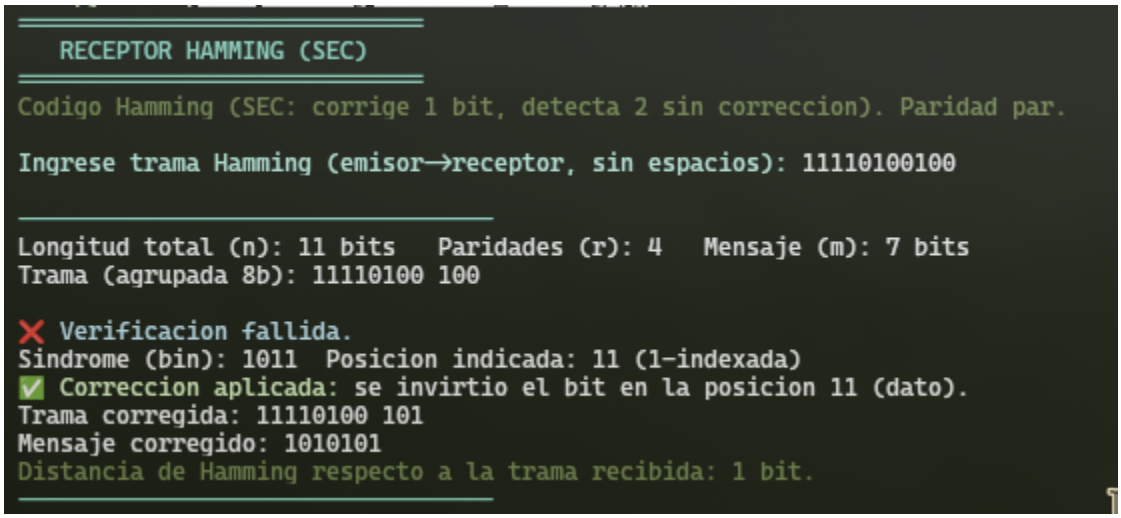
Receptor



Mensaje con un bit cambiado:
Emisor

11110100100 (ultimo bit cambiado)

Receptor

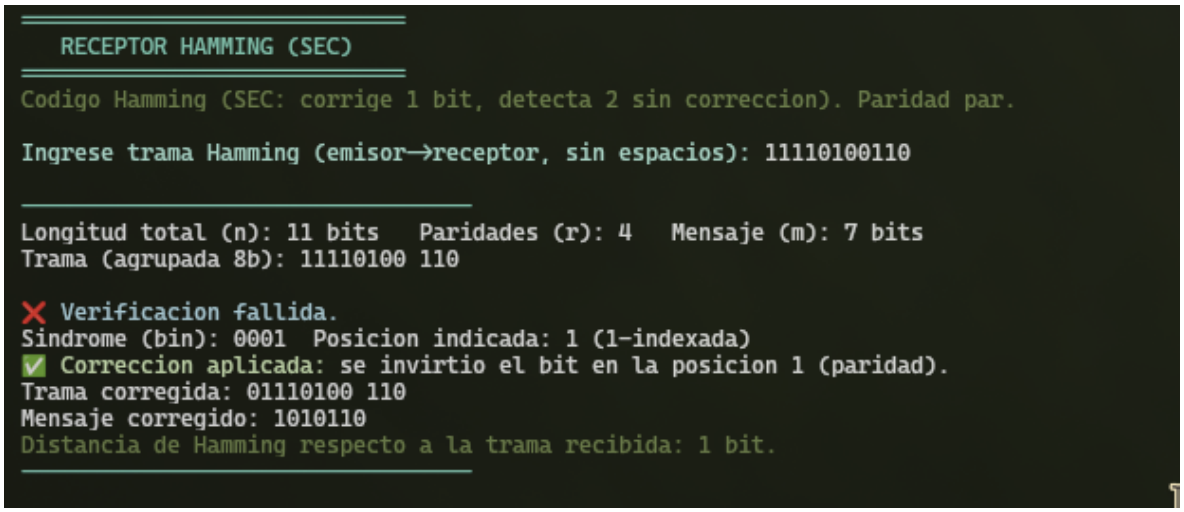


Mensaje con dos bits cambiados

Emisor

11110100110 (ultimo y penultimo bits cambiados)

Receptor

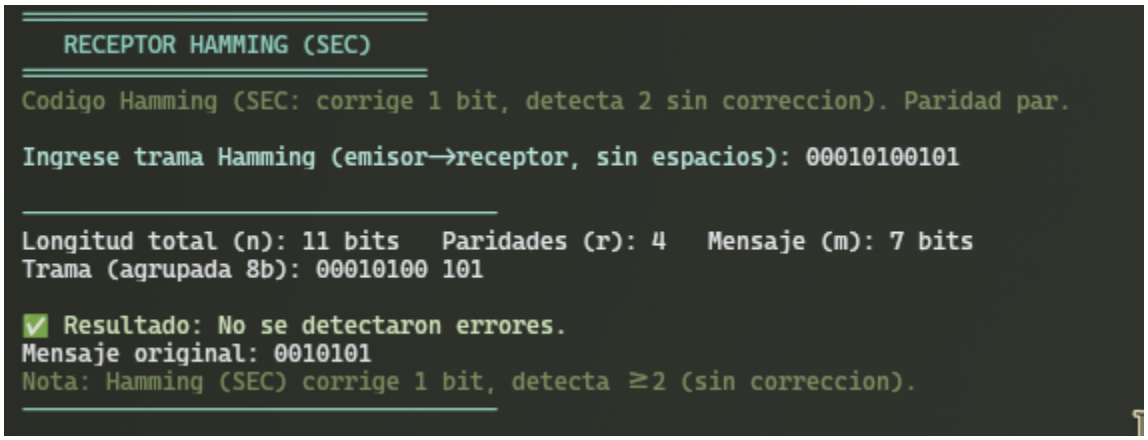


¿Es posible manipular los bits de tal forma que el algoritmo seleccionado no sea capaz de detectar el error? ¿Por qué sí o por qué no? En caso afirmativo, demuestrelo con su implementación.
Hamming:
Sí. Mostramos un caso indetectable con flip en posiciones 1, 2 y 3 de la trama (síndrome 0), lo que cambia el mensaje sin que el receptor lo note. Esto se debe a que el Hamming SEC tiene d=3, y existen ternas de columnas cuya XOR es 0.

Mensaje original:
11110100101

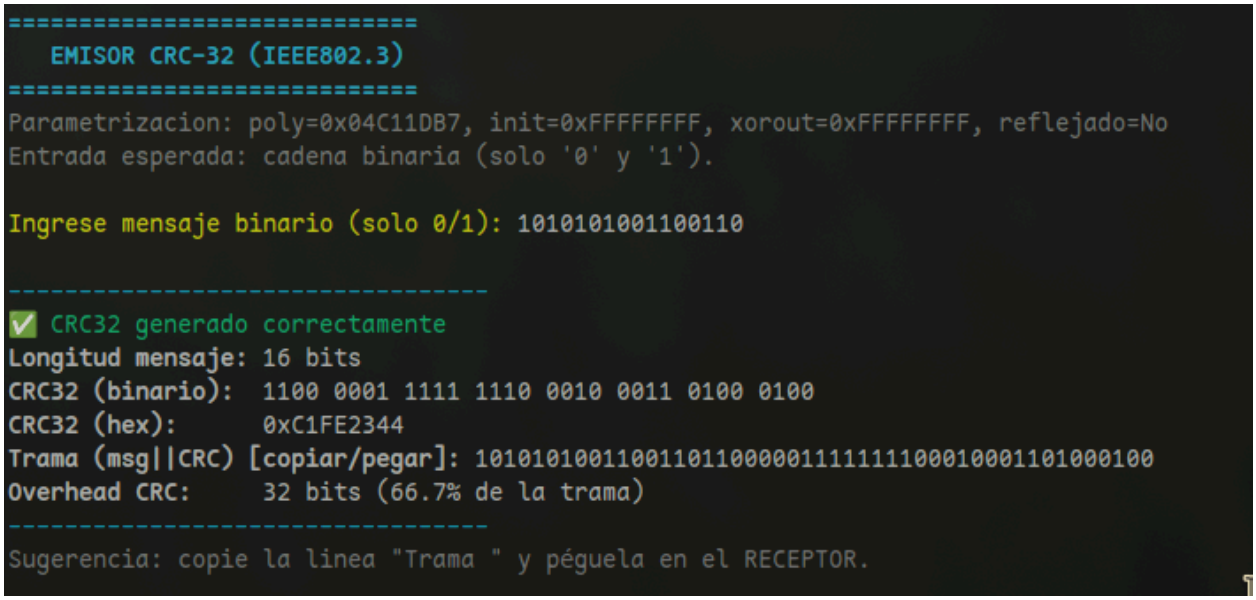
Mensaje con las 3 primeras posiciones flipeadas:
00010100101

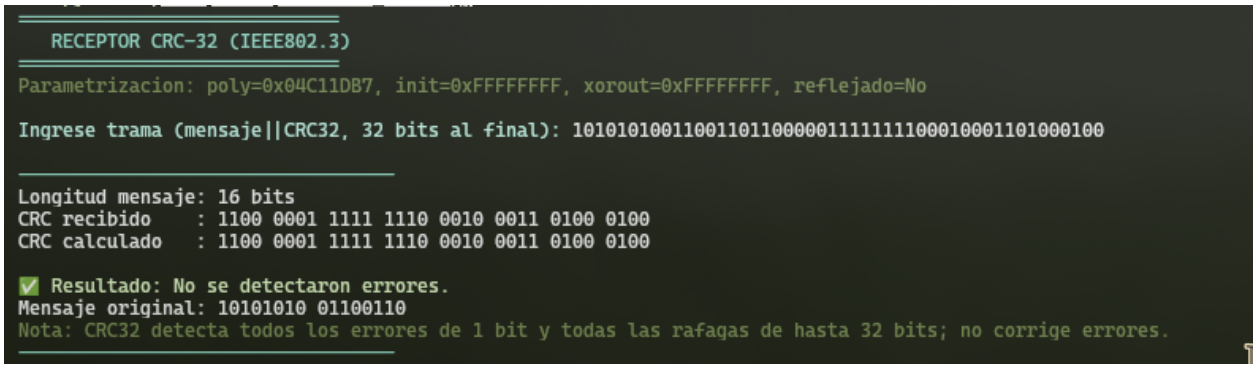
Receptor



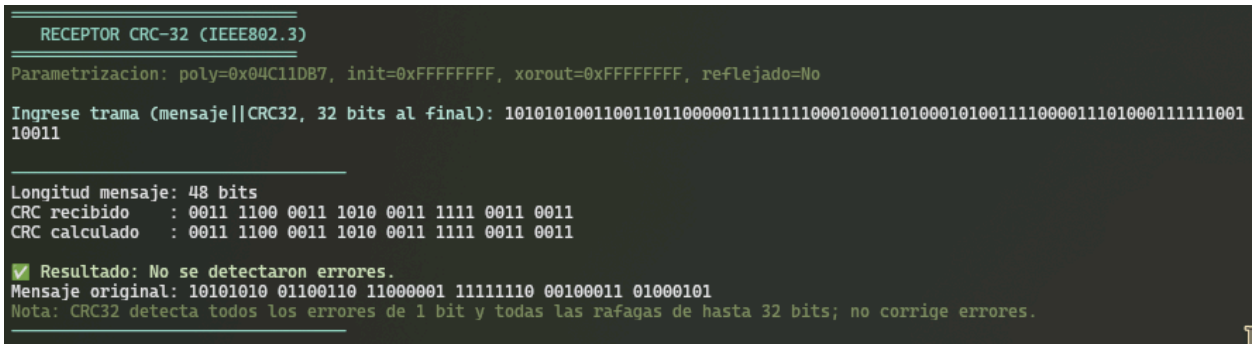
Si se puede truquear

CRC32:
En canal aleatorio, la probabilidad de no detección es ~$1/2^{32}$ para errores fuera de sus garantías; en la práctica, no lo observamos. Sin embargo, un atacante que recalcula el CRC tras modificar el payload pasa: CRC no es autenticación. Lo demostramos generando una trama válida para otro mensaje con el emisor.





Cambiamos un bit de este mensaje saliente y lo volvemos a enviar al CRC y después la salida de eso lo volvemos a pasar al Receptor y si lo aceptara



CRC-32 detecta errores aleatorios con probabilidad de fallo ~$1/2^{32}$, pero no auténtica: si un atacante modifica el mensaje y recalcula el CRC, el receptor lo acepta. Lo demostramos generando una trama válida para otro mensaje (B) y el receptor la aceptó.

Ventajas y desventajas

Hamming (SEC)

Ventajas

- Corrige 1 bit de error automáticamente.
- Overhead pequeño y predecible: rrr bits con 2r≥m+r+12^r \ge m+r+12r≥m+r+1.
- Implementación simple y muy rápida.

Desventajas

- Con ≥2 bits, puede fallar e incluso miscorregir (silencioso).
- Existen patrones de 3 bits indetectables (p. ej. 1,2,3).
- Requiere bloques con tamaños compatibles (ajustar m,rm, rm,r).

Cuándo usarlo

- Cuando necesitas corrección (SEC) y el canal no es muy ruidoso; payload corto/medio.

CRC-32

Ventajas

- Altísima capacidad de detección: 1-bit, 2-bits y ráfagas ≤32 bits garantizadas; prob. de fallo aleatorio ~1/2^32.
- Overhead fijo de 32 bits (relativamente bajo para mensajes largos).
- Muy eficiente (soporte HW/bitwise rápido) y sirve para cualquier longitud de mensaje.

Desventajas

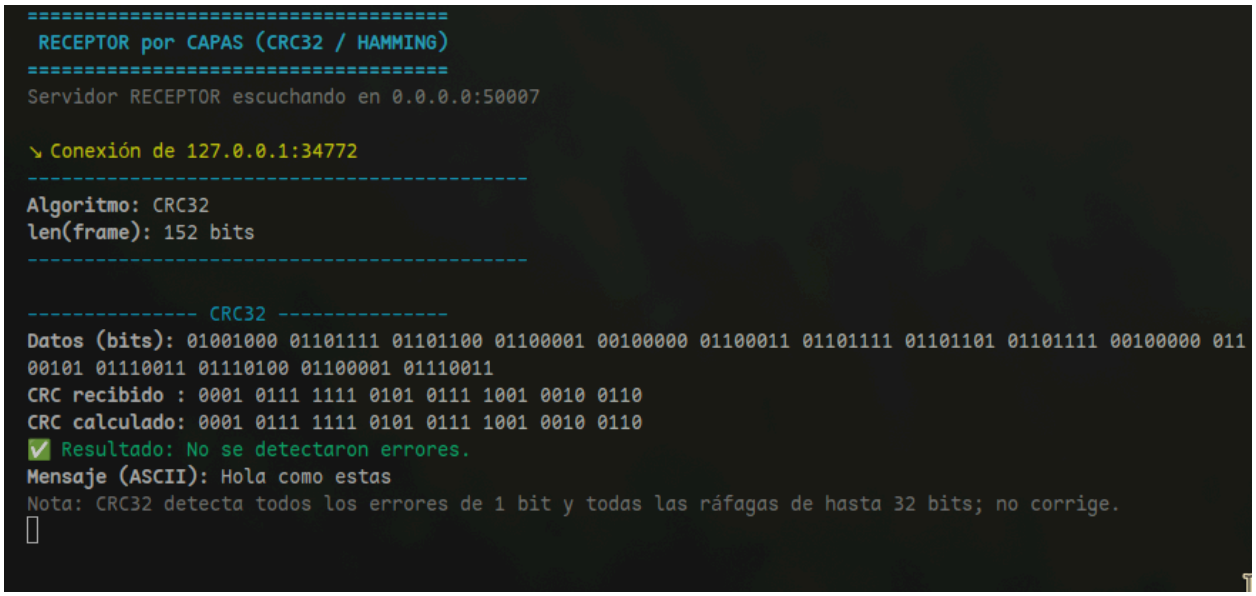
- No corrige (solo detecta).
- No autentica: un atacante que recalcula el CRC tras modificar el payload lo hace pasar.
- En mensajes muy cortos, 32 bits de overhead puede ser alto.

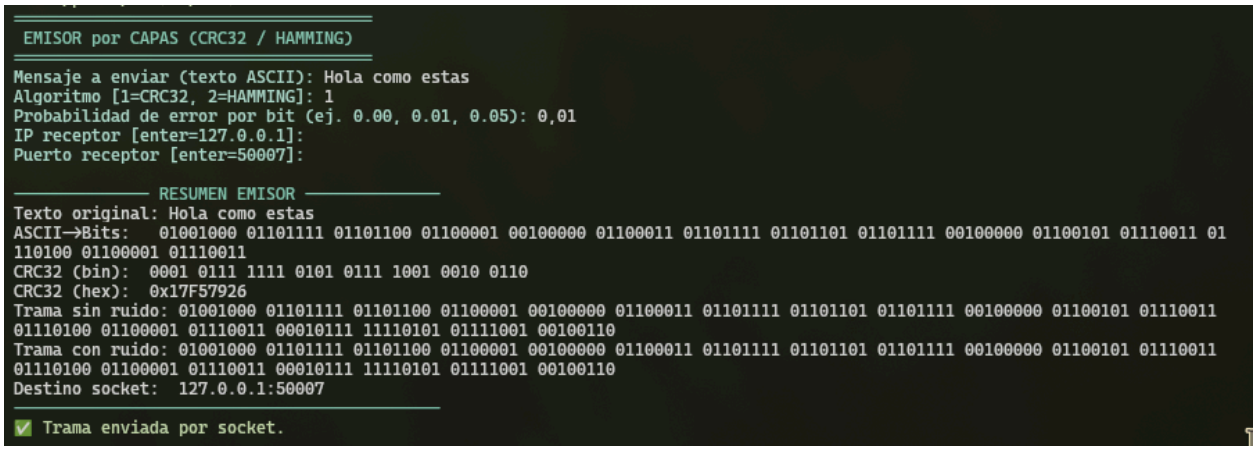
Cuándo usarlo

- Cuando necesitas detectar con fiabilidad en tramas medianas/largas y combinar con reintentos/ARQ.

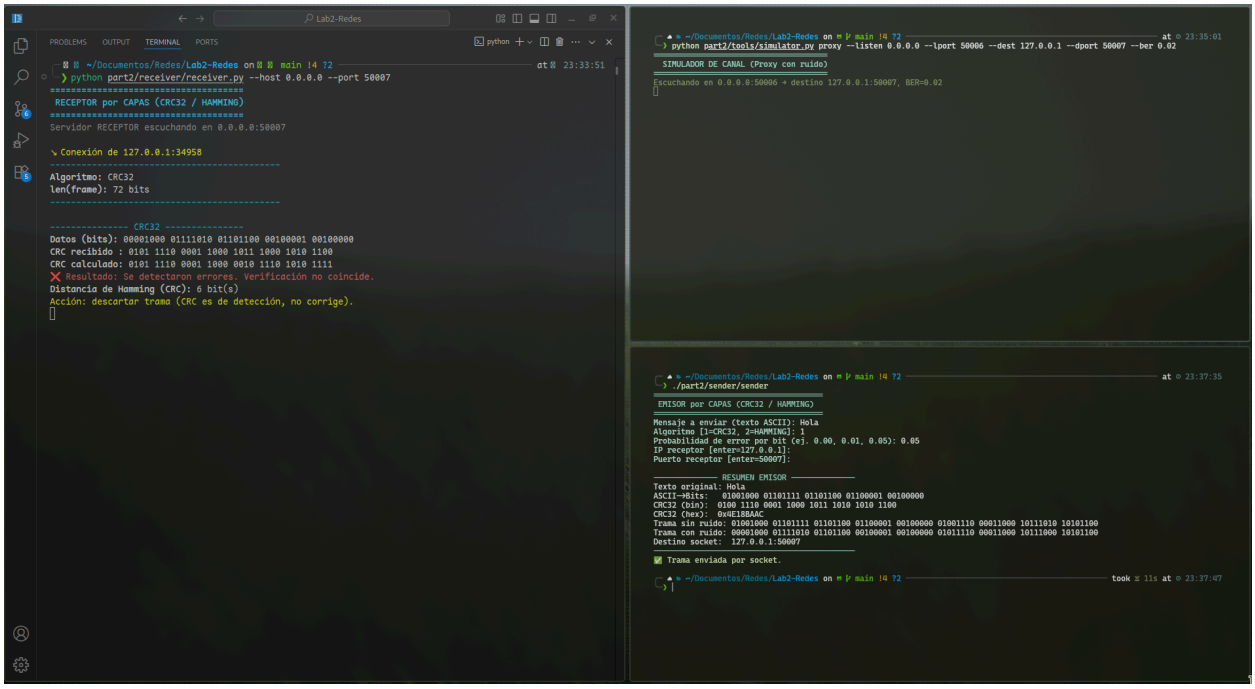
Segunda parte:

Pruebas de emisor receptor

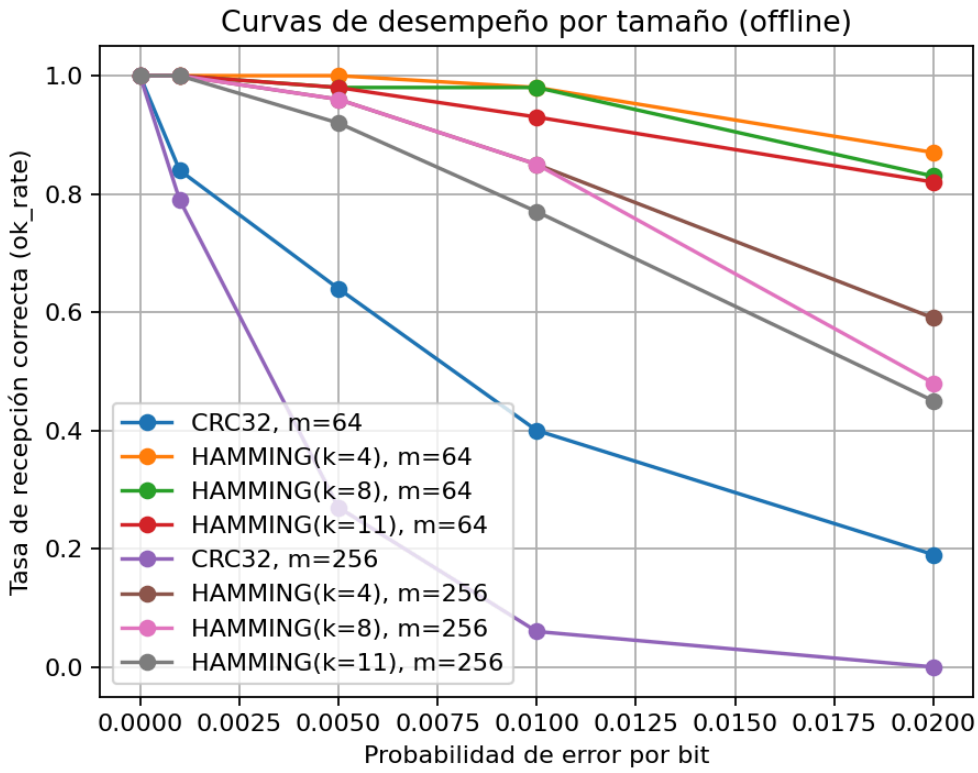




Con proxy emitiendo ruido para correción de errores



Métricas de estrés en modo offline:



Resultados
Los datos obtenidos coinciden con la teoría y permiten comparar directamente el comportamiento de CRC-32 y del código Hamming SEC.

• CRC-32 (solo detección). La proporción de tramas aceptadas (ok) iguala la probabilidad de que no ocurra ningún flip en todo el paquete: (1-p)^(m+32). Para m=64 bajamos de 0,91 a 0,38 y luego a casi 0 al pasar de p=0,001 a 0,01 y 0,05; para m=256 la caída es aún más pronunciada. Esto confirma que cualquier error, aunque sea mínimo, provoca el rechazo de la trama.

• Hamming SEC (corrección de 1 bit). A BER bajos (0,1 %–1 %) el ratio ok se mantiene por encima de 0,95 porque la mayoría de los errores afectan a un solo bit y se corrigen. El contador corr_avg crece con p, reflejando más correcciones efectivas. Al aumentar p o el tamaño de bloque (k=8/11) crece la probabilidad de que aparezcan dos o más errores en un mismo bloque, lo que genera uncor_avg>0 y reduce ok; sin embargo, incluso a p=0,05 el código sigue aceptando un 34–48 % de tramas (m=64), superior al desempeño de CRC-32 en ese punto.

Conclusiones
CRC-32 ofrece una detección muy confiable, pero exige retransmisión cuando aparece cualquier error; su tasa de éxito decae conforme (1-p)^(m+32). El código Hamming SEC, por el contrario, incrementa la tasa de recepción sin retransmisión en entornos con baja o moderada tasa de errores, gracias a la capacidad de corregir 1 bit por bloque; esta ventaja se pierde cuando la probabilidad de ≥2 errores por bloque se vuelve apreciable, especialmente en bloques largos.