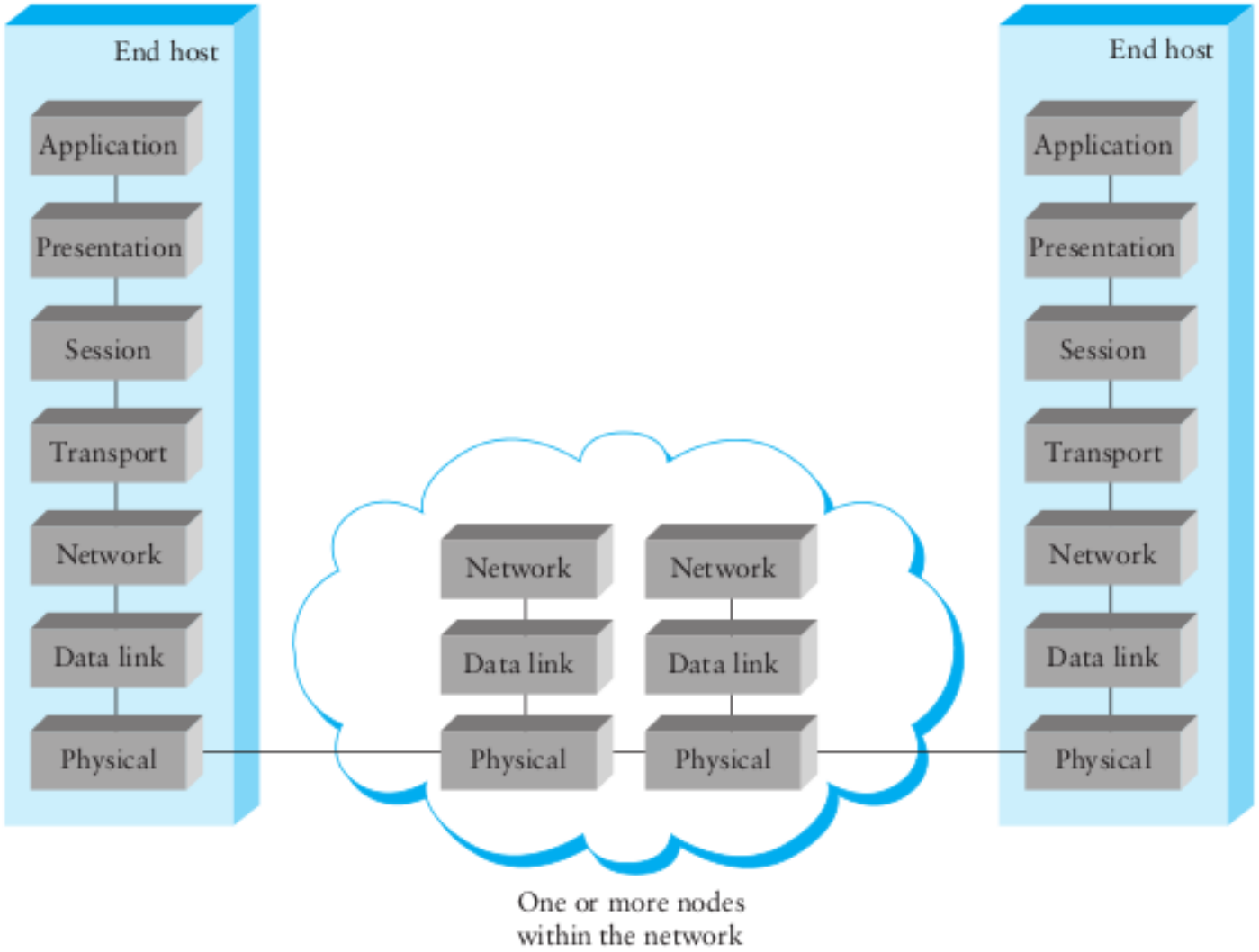
Protocolos punto a punto

Teoría de la Comunicaciones

08.04.2025

Arquitectura en capas

Las comunicaciones se dan en capas que se brindan servicios entre sí



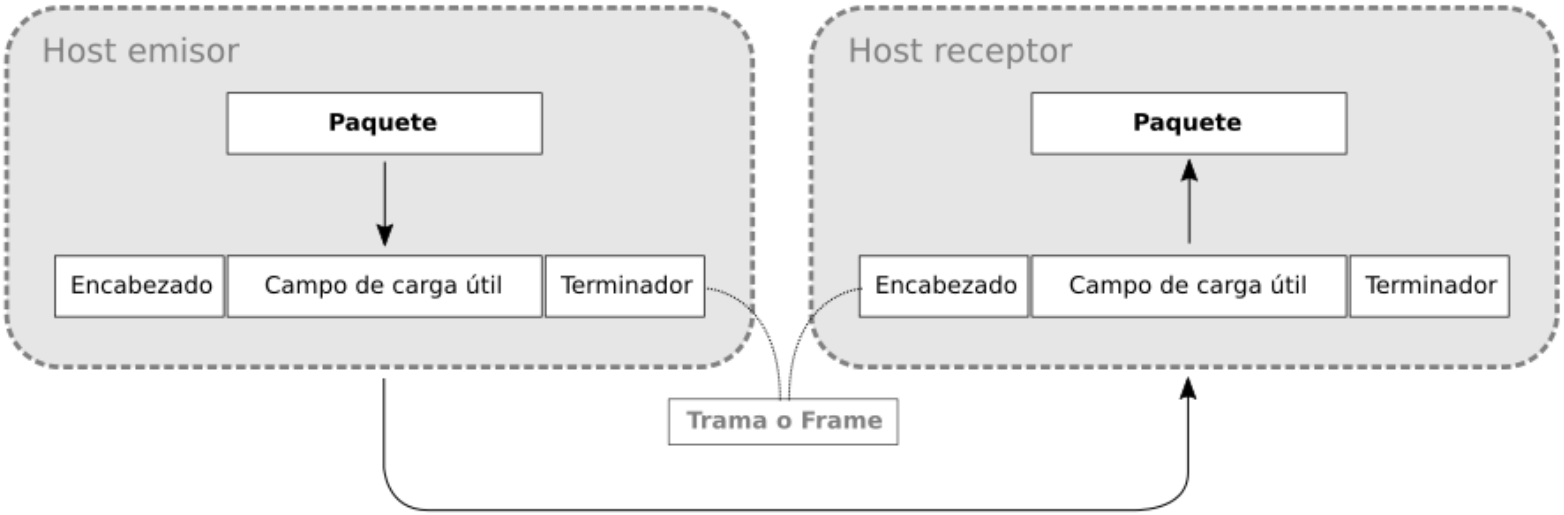
Contexto

- **Caño serial** (no hay desorden)
- **Vulnerable a ruido impulsivo** Lo que se recibe puede no ser lo que se envió (*error de transmisión*)

Conceptos y objetivos a resolver

- **Framing** - Encapsular los bits en frames agregando información de control - ¿Cómo los codifico/decodifico?
- **Proveer servicio a la capa superior** - *¿Confiable o no confiable?*
- **Control de Errores** - ¿Se produjo algún error? ¿Que hacemos con los errores?
- **Control de Flujo** - (Más adelante: en nivel de transporte)

Framing



¿Qué es un **frame**?

Ejercicio

Dado un enlace punto a punto a la luna de 1Mbps con un delay de 1.25 segundos

a. ¿Cuántos bits entran en el enlace?
b. Asumiendo que se separan los bits en frames de largo fijo de 1Kb ¿Cuántos Frames entran en el enlace?

¿Cómo se separan los frames en un tren de bits consecutivos?

- Largo fijo
- Largo en el encabezado
- Delimitadores con *bit-stuffing*

¿Cómo se separan los frames en un tren de bits consecutivos?

- Largo fijo
- Largo en el encabezado
- Delimitadores con *bit-stuffing*

Eficiencia de frame

$$\eta_{frame} = \frac{largo\ \ de\ \ los\ \ datos}{largo\ \ total\ \ del\ \ frame}$$

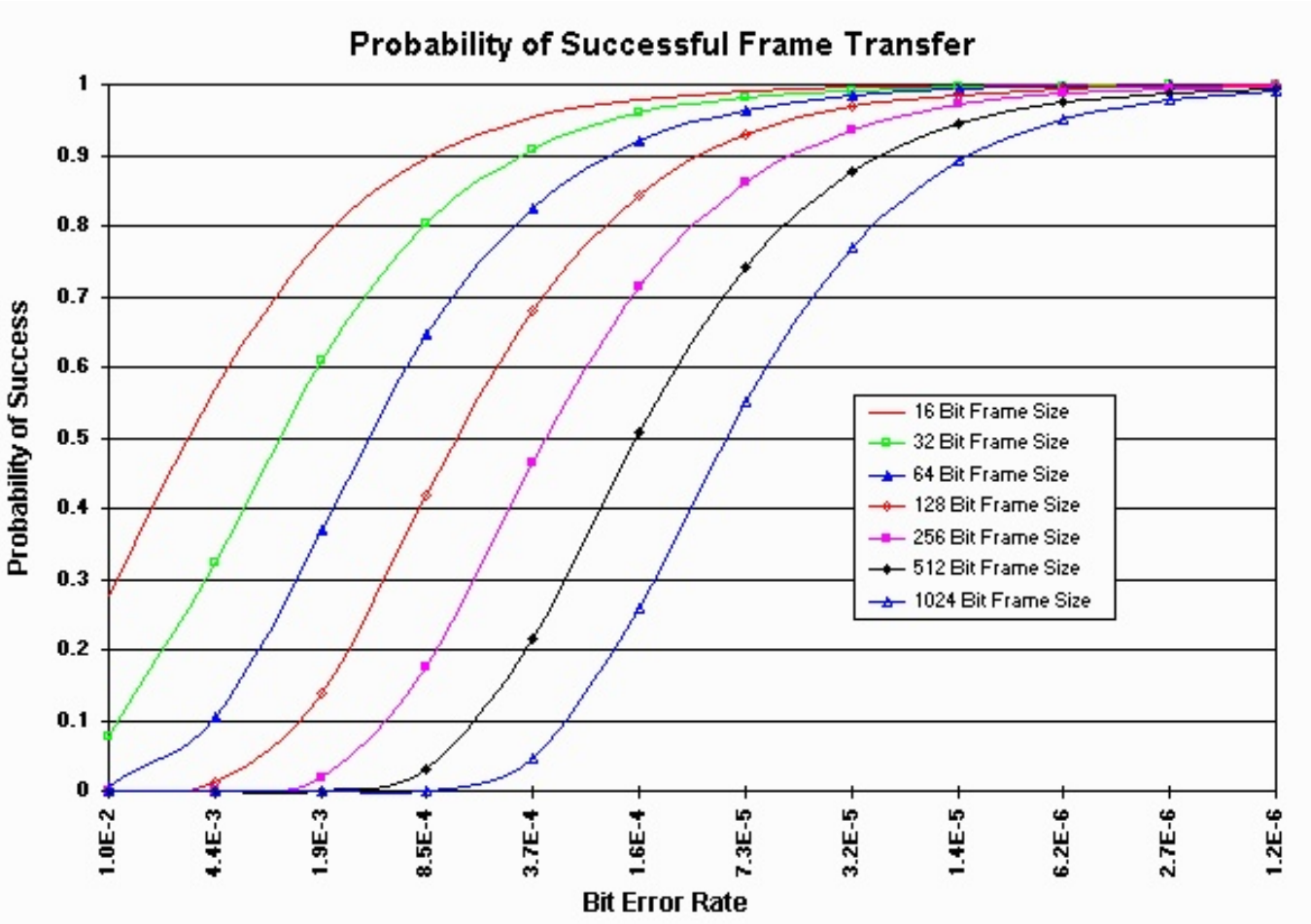
Ejercicio

Calcule la eficiencia del frame tomando en cuenta solo el overhead impuesto por las siguientes técnicas de framing:

a. Largo fijo
b. Campo de 16 bits en el encabezado indicando el largo del frame
c. Delimitadores de 8 bits usando *bit-stuffing*

Frames de largo fijo: probabilidad de error

La probabilidad de que el frame llegue bien depende del largo del frame.



Control de errores

- Detección y Corrección de errores
  - ★ Bit de paridad
  - ★ CRC
  - ★ Checksum
  - ★ Hamming
  - ★ Reed-Solomon
  - ★ MD5

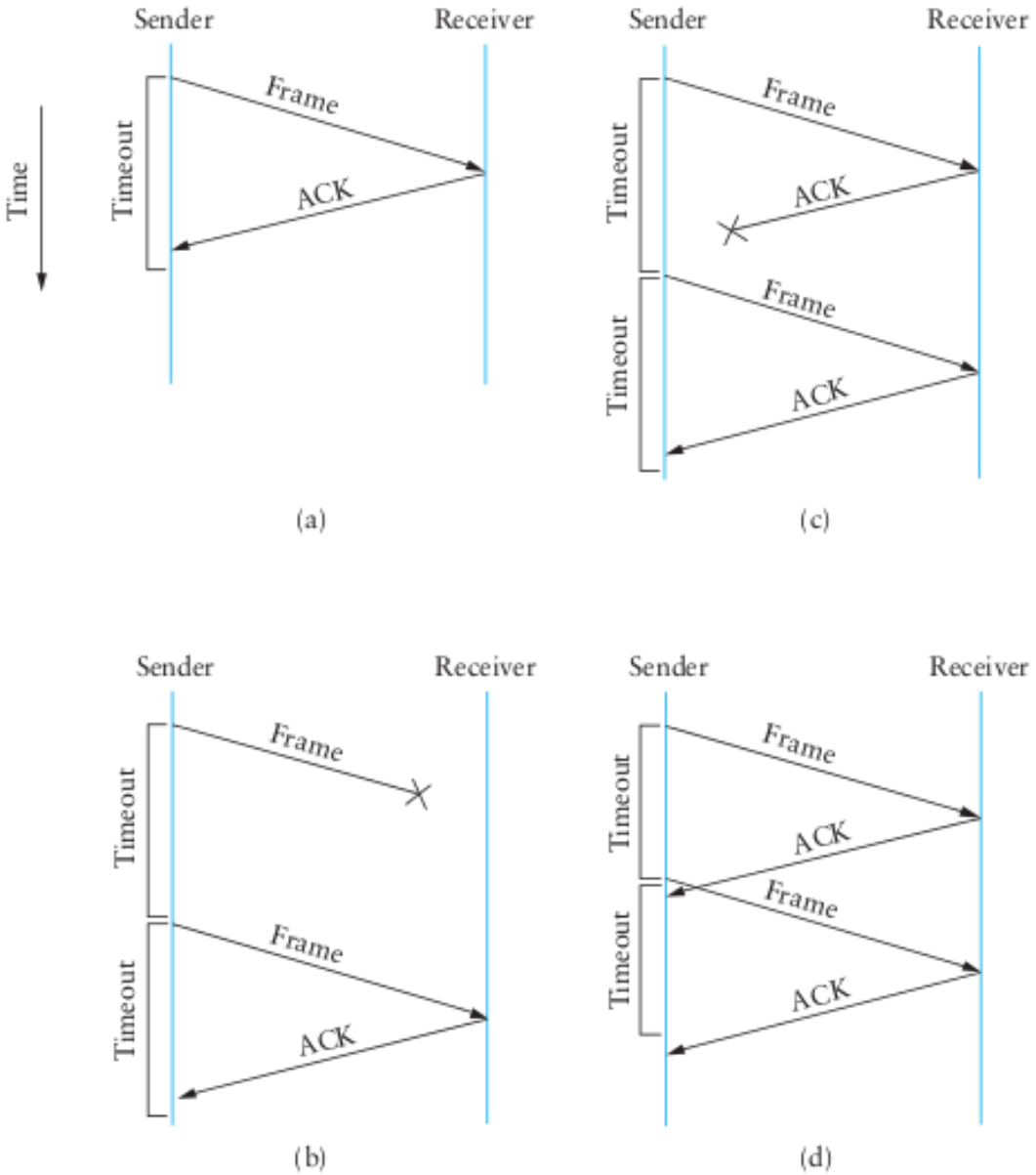
Control de errores

- Detección y Corrección de errores
  - ★ Bit de paridad
  - ★ CRC
  - ★ Checksum
  - ★ Hamming
  - ★ Reed-Solomon
  - ★ MD5
- Retransmisiones
  - ★ **Explícitas:**
    mensajes de control específicos para pedir un datos nuevamente
  - ★ **Implícitas:**
    cuando ocurre un time-out se asume que el dato se perdió

Tipos de Servicio

- Sin conexión y sin reconocimiento
  - Los datos se envían sin necesidad de saber si llegan bien.
- Sin conexión y con reconocimiento
  - Los datos se envían y se asegura la correcta recepción mediante el aviso explícito (ACKs)
- Orientado a conexión
  - Además de asegurar la correcta recepción de los datos, se mantiene un *estado* de conexión (una sesión)

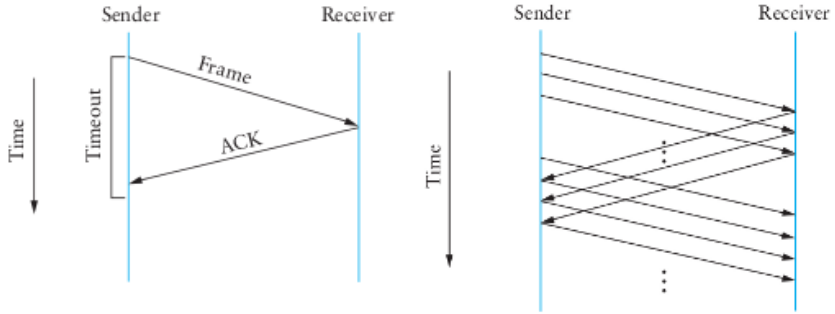
Transmisión confiable: Stop and Wait



Transmisión confiable: Stop and Wait

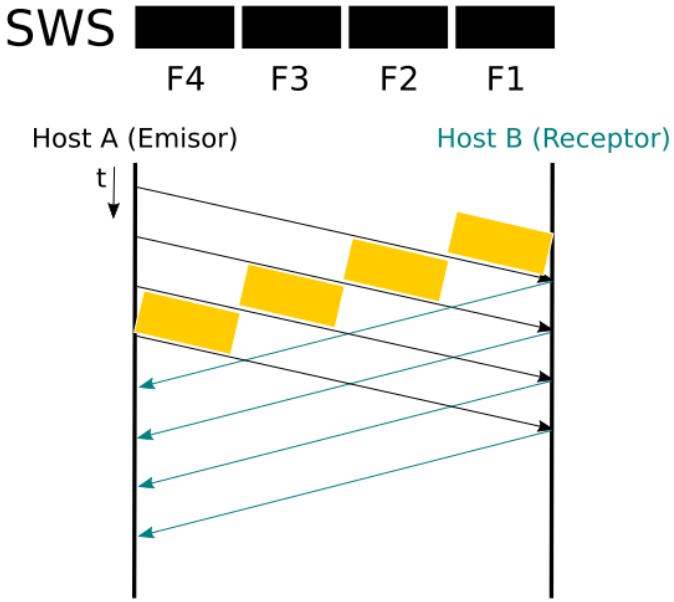
- Cada Frame debe ser reconocido por el receptor
- Surge la necesidad de secuenciar los frames para evitar el problema de las reencarnaciones
- ★ Para Stop & Wait se necesita poder secuenciar 2 frames.

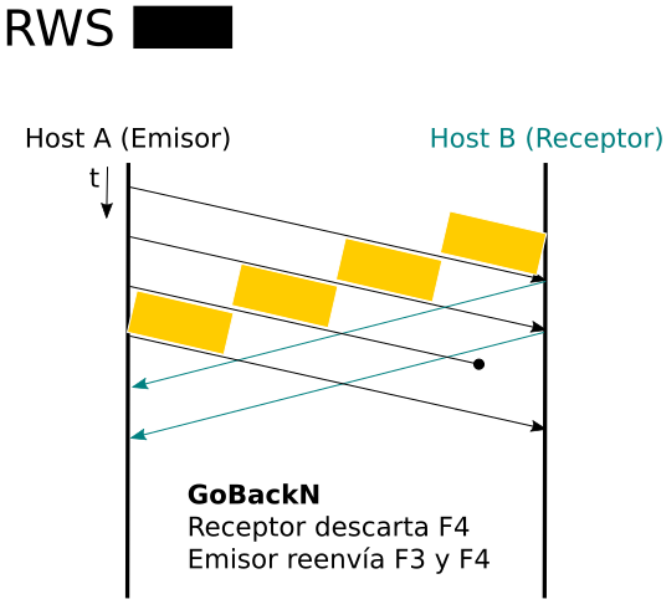
Motivación: llenar el canal



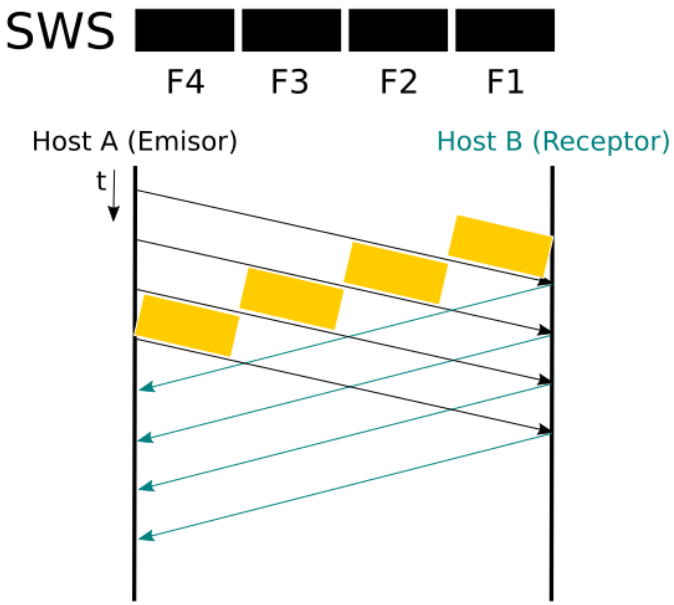
- ★ **Ventana de emisión:** $SWS = V_{tx} * RTT / |Frame|$
- ★ **Enviar según:**
  $UltimoFrameEnviado \leq UltimoFrameReconocido + SWS$

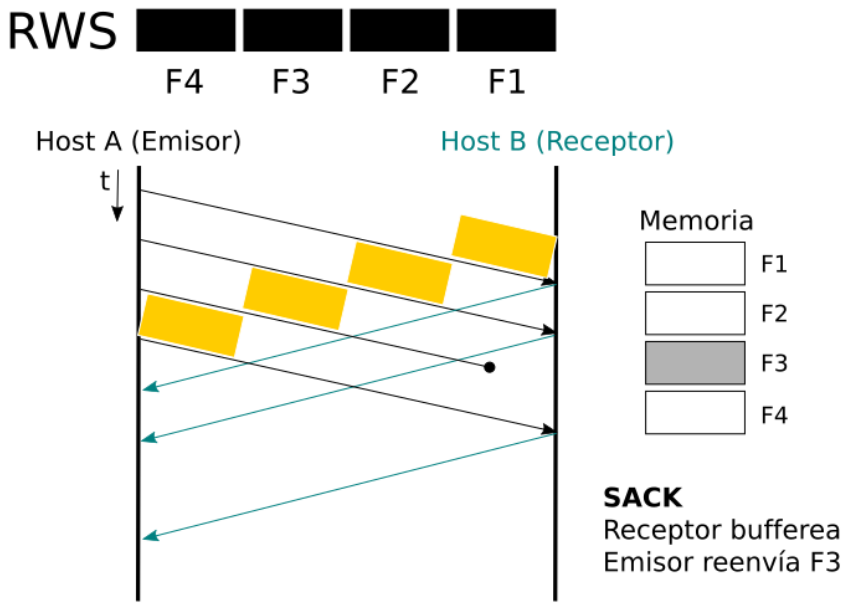
Sliding Window





Sliding Window





Ejercicio

Diseñe posibles conjuntos de frames para los siguientes tipos de protocolos, asumiendo que se detectan errores usando CRC. (No hace falta aclarar el largo de los campos)

a. Stop & Wait
b. Sliding Window con GoBackN usando *Piggybacking*
c. Sliding Window con ACK Selectivo

Debemos aumentar la ventana de emisión para aprovechar mejor el canal:

$$\bigstar\ SWS = V_{tx} * RTT / |Frame|$$

Si el receptor dispone de buffers, se puede usar SACK. Sino, no aporta a la eficiencia del protocolo.

$$\bigstar\ RWS = \begin{cases} SWS & \text{Si hay SACK} \\ 1 & \text{Si no} \end{cases}$$

Y para distinguir reencarnaciones

$$\bigstar\ \#frames \geq SWS + RWS$$

Eficiencia

¿Cuánto tiempo se está transmitiendo con respecto al tiempo bloqueado esperando?

$$Eficiencia = \frac{T_{tx}(V)}{RTT(F)}$$

Con $T_{tx}(V)$ el tiempo de transmisión de una ventana y $RTT(F)$ el tiempo de ida y vuelta de un frame.

★ *Aumentar la eficiencia es estar bloqueado lo menos posible.*

Ejercicio

Un protocolo sobre un enlace punto a punto de 1Mbps y 0.25 segundos de delay, trabaja con ventana deslizante con GoBackN usando frames de largo fijo 2Kb y un CRC de 16bits para detectar errores.

a. Calcule cuáles son los tamaños de ventana de emisión y recepción óptimos.
b. ¿Cuantos bits hacen falta para secuenciar los frames?
c. Calcule cuánto tiempo es necesario para transmitir 20Mb de datos asumiendo que no hay errores.

b. ¿Cuantos bits hacen falta para secuenciar los frames?

$\lceil \log_2(SWS + RWS) \rceil$

c. Calcule cuánto tiempo es necesario para transmitir 20Mb de datos asumiendo que no hay errores.

$|Datos|$ = $|Frame|$ - $|SEQ|$ - $|Checksum|$ = X bits

Entonces podemos calcular $\frac{20Mb}{Xb}$ frames.

Luego, $Delay(\#frames)$ =
$T_{tx}(\#frames) + T_{prop} = \frac{|frame| * \#frames}{1Mbps} + 0.25s$

Mas el último ACK,
$\frac{|frame| * \#frames}{1Mbps} + 0.5s$

Ejercicio de parcial

Un protocolo confiable punto a punto que usa sliding window, opera sobre un canal de 10 Mbps, usa SACK y un frame emisor de 5Kb como el siguiente:

#SEQ (8bits) ; Datos ; Checksum

a. Proponga un frame para el receptor.
b. ¿Cuál es el valor de delay para el cual el protocolo presenta un 100% de eficiencia?
c. Si el delay fuera de 1 seg ¿Cuántos bits deberían ocupar los números de secuencia de manera de maximizar la eficiencia?

Bibliografía

- Capítulo 2 “Direct Link Networks”, Sección 2.3 (Framing) hasta Sección 2.5 (Reliable Transmission) inclusive - Computer Networks: A Systems Approach - Larry Peterson and Bruce Davie. Elsevier, 2012. License CC BY 4.0 (https://github.com/SystemsApproach/book)
- Capítulo 3 “La Capa de Enlace de Datos” - A.Tanenbaum.