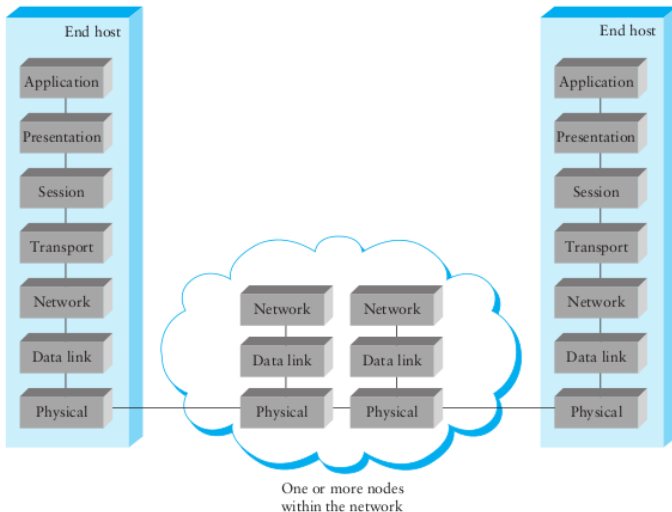


# Protocolos punto a punto

Teoría de la Comunicaciones

30 de Agosto de 2017

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



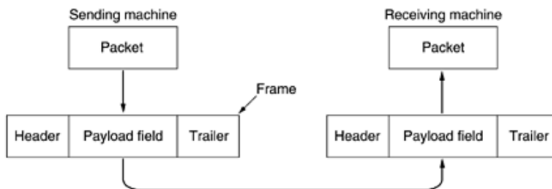
## Conceptos

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

## Objetivos

- **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)

**Figure 3-1. Relationship between packets and frames.**



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

- a. ¿Cuántos bits entran en el canal?
- b. Asumiendo que se separan los bits en frames de largo fijo de 1kB ¿Cuántos Frames entran en el cable? ¿Y si usara frames de 2kB?

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

- Largo fijo
- Largo en el encabezado
- Delimitadores con *bit-stuffing*
- (Violación de código)

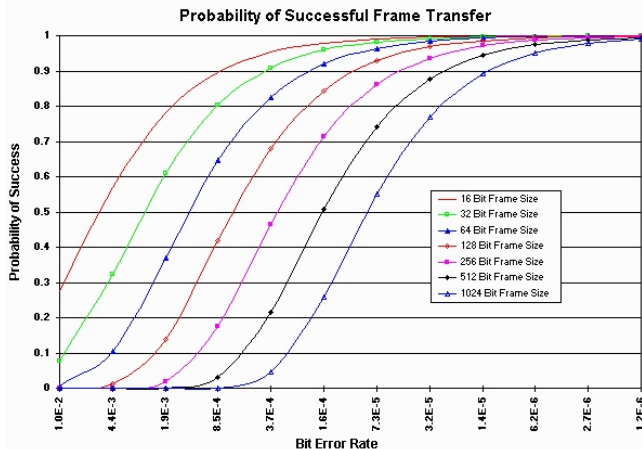
### Eficiencia de frame

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

Calcule el 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*

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





- Detección y Corrección de errores

- ★ Bit de paridad
- ★ CRC
- ★ Checksum
- ★ Hamming
- ★ Reed-Solomon
- ★ MD5

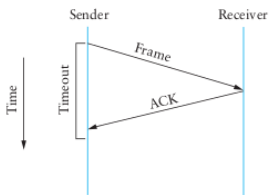
- Retransmisiones

- ★ Explícitas (mensajes de control específicos para pedir un datos nuevamente)
- ★ Implícitas (cuando ocurre un time-out se asume que el dato se perdió)

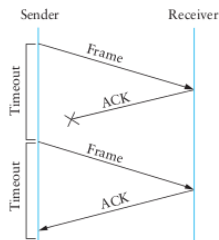
Una estrategia para detectar y/o corregir errores es transmitir los datos en bloques de  $n$  filas con  $k$  bits por fila y agregar un bit de paridad a cada fila y columna. ¿Qué tipos de errores detecta este esquema? ¿Qué tipos de errores corrige?

- 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)

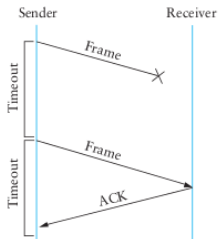
# Transmisión confiable: Stop and Wait



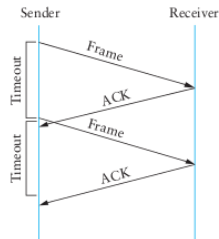
(a)



(c)



(b)



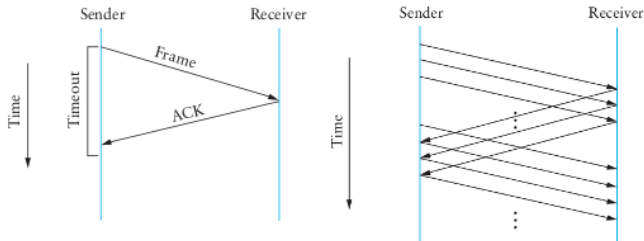
(d)

- 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.

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

## Motivación: llenar el canal



★ **Ventana de emisión:**  $SWS = V_{tx} * RTT / |Frame|$

★ **Enviar según:**  $UltimoFrameEnviado \leq UltimoFrameReconocido + SWS$

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

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

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

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

Y para distinguir reencarnaciones

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



¿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.*

Un protocolo sobre un enlace punto a punto de 1Mbps y 0.25 segundos de propagación, 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. ¿Cuántos bits hacen falta para secuenciar los frames?
- c. Calcule cuánto tiempo es necesario para transmitir 20Mb de datos asumiendo que no hay errores.

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?