

UNIVERSITAT DE BARCELONA

# XARXES DE COMUNICACIONS

---

---

7º Semestre - EET

---

Carlos Marín

Curs 2015 - 2016



## **ÍNDICE:**

INTRODUCCIÓN.....	2
CLASIFICACIÓN REDES DE COMUNICACIONES.....	3
SUBSISTEMA DE COMUNICACIONES.....	9
MODELO DE REFERENCIA ISO PARA OSI.....	11
CAPA FÍSICA.....	12
CAPA DE ENLACE.....	13
CAPA DE RED.....	14
CAPA DE TRANSPORTE.....	15
CAPA DE SESIÓN.....	17
CAPA DE PRESENTACIÓN.....	18
CAPA DE APLICACIÓN.....	18
SERVICIOS EN EL MODELO OSI.....	21
CONTROL DE ERRORES.....	31
VERIFICACIÓN DE SUMA DE BLOQUE.....	37
VERIFICACIÓN DE REDUNDANCIA CÍCLICA.....	39
CÓDIGOS CRC.....	41
COMPRESIÓN DE DATOS.....	46
GRADO DE UTILIZACIÓN DEL ENLACE.....	54
PIGGYBACKING.....	64
VENTANA DESLIZANTE.....	65
EXAMEN ENERO 2016 SOLUCIONADO.....	66
PREGUNTAS RANDOM DE EXÁMENES ANTERIORES.....	72



Miguel Ángel Morumó

ma motsumo @ ub.edu



Bibliografía

\* IMPORTANTE  
RECOMENDABLE.

- Fred Halsall, "Comunicación de datos, redes de sistemas abiertos", Ed. Addison Wesley Iberoamericana (computadores) y
- Andrew S. Tanenbaum, "Redes de ordenadores", Ed. Prentice Hall
- William Stallings, "Local & Metropolitan Area Networks", Ed. Prentice-Hall
- Jesús García Tomás, Santiago Ferrando, Mario Piattini, "Redes para proceso distribuido", Ed. Ra-ma.

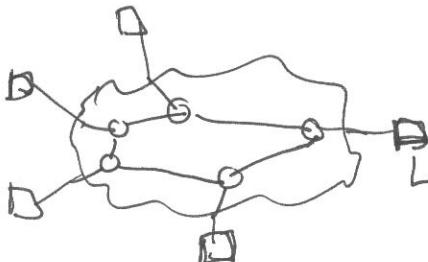
NECESIDAD DE LAS REDES DE COMUNICACIONES

- Ordenadores inmenos y caros. [poco].
- No hay necesidad de interconectarlos.
- Descenso del costo del computador.
- Conjunto de nuevas aplicaciones: generación de información.
- Compartir recursos:
  - Recursos Hardware: compartir una misma impresora. Acceder des de cualquier equipo.
  - Recursos Software: compartir datos y programas.
- Tener una red: entorno más fiable.
  - ↑ Fidabilidad.
  - Varios equipos para el mismo proceso.
  - Escalabilidad: añadir más equipos para tener mayor potencia de cálculo.

DEFINICIÓN DE RED DE ORDENADORES

↳ Formada por una serie de elementos / equipos que queremos interconectar

- LÍNEAS DE COMUNICACIÓN
- ○ NODOS



□ RED DE ORDENADORES  
 ↳ HOST  
 DTE CDia Terminal Equipment  
 ESTACIONES  
 SISTEMA TERMINAL  
 ○ SUBRED ○  
 RED DE COMUNICACIONES

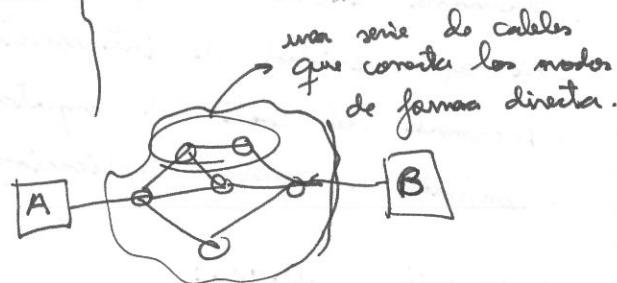
- + HOST: ordenador o dispositivos que ejecutan programas.
- + SUBRED: información fluye de origen a destino.
- Líneas de Transmisión: medio físico que permite transmitir bits.
- Dispositivos de conmutación: [NODES] - ROUTERS.
  - Moen información.
  - Ordenadores especializados en el enrutado de la información. Su objetivo es que sus datos llegan al destino. Moen rápidamente la información.
- NODOS: ROUTERS  $\Rightarrow$  trabajan como MESSAGE STORE & FORWARD.

### CLASIFICACIÓN DE REDES DE COMUNICACIÓN

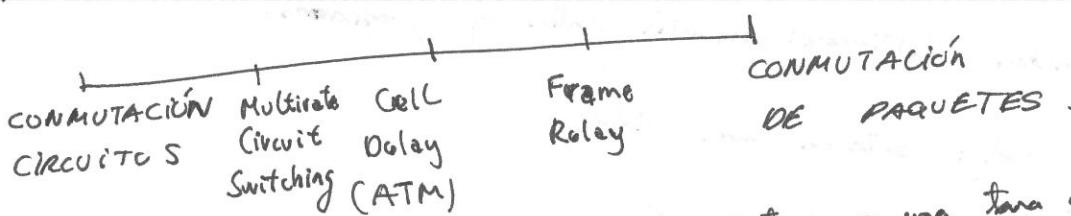
[En función de]:

- Diseño de la SUBRED DE COMUNICACIONES.
- Alcance geográfico.

- DISEÑO SUBRED
  - Canales punto a punto
  - línea de transmisión
  - Canales de difusión.



Técnicas de conmutación de datos:



(Igualada): Sistema a una tasa de velocidad definida (fija) y más sencillo.

(Varada): Diferentes velocidades pero computacionalmente más complejo.

- CONMUTACIÓN DE CIRCUITOS: fijo un camino fijo de extremo a extremo.
- Tiene 3 fases: → necesita reservar unos recursos.

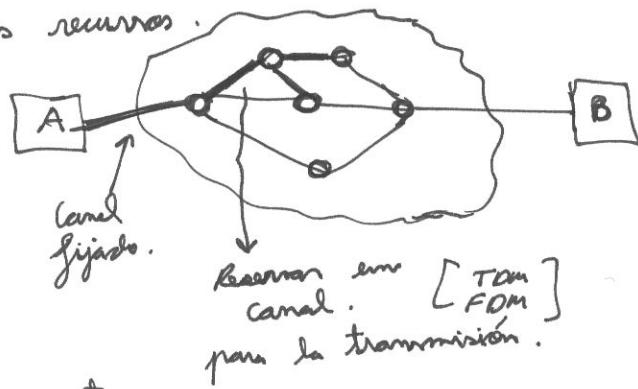
+ 1º: Establecimiento de enlace.

- Necesita características de conmutación interna.

- El destino B debe aceptar la comunicación.

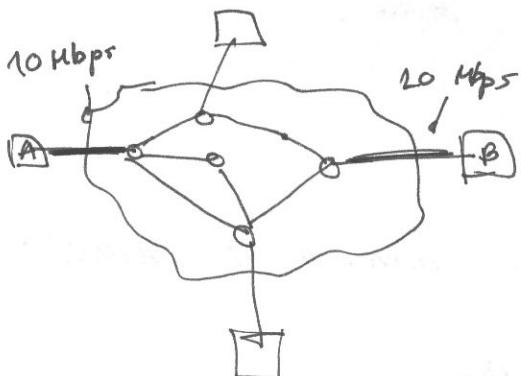
+ 2º: Transmisión. La red es transparente.  
No hay más retrasos que al Daley.

+ 3º: Finalizar la transmisión.  
Librar los recursos reservados.





Esguema de una red de ordenes de voz



Conjunto Completo

SUBRES

ESTACIONES OTE, HOST

NODOS DE COMUNICACIÓN, IMP'S.

LÍNEAS DE TRANSMISIÓN.

Canal punto a punto

Redes conmutadas.

CONMUTACIÓN DE CIRCUITOS

CONMUTACIÓN DE PAQUETES

(+ Flexible)  
(+ complejidad)

1º Enlace dedicado.

2º Dividen la ruta para alcanzar el destino.

3º Reservan en slot de tiempo. Buscan un modo libre y reservarán un canal.

Tasa de datos fija. [Reserva = Transmisión]

4º Desconexión del circuito. Libera los canales reservados.

CONMUTACIÓN DE CIRCUITOS (Redes de telefonía - voz)

- Uso de todos los recursos. No es posible nueva comunicación.

CONMUTACIÓN DE CIRCUITOS

- Tasa de datos fija.

- Retraso inicial en la fase de establecimiento del circuito. Retraso de transmisión.

- (Cable directo).

- Recursos reservados para una comunicación en concreto. (Voz no problema, datos no es tan buena).

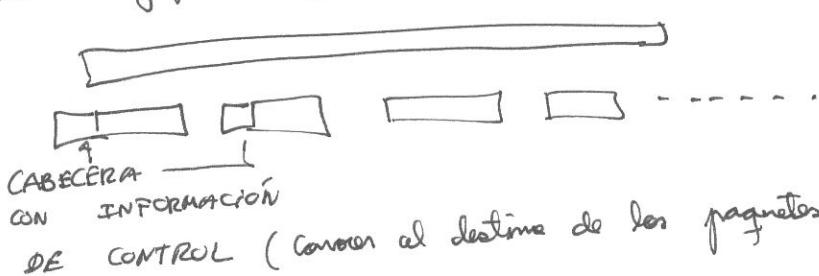
\* La transmisión de datos no es continua.

- Inflexible.

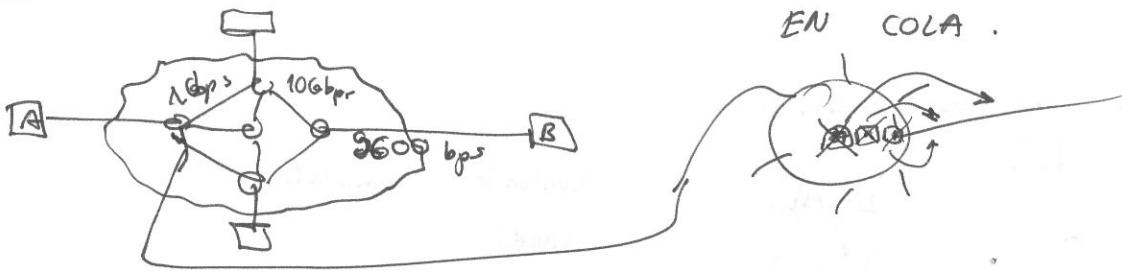
(Transmisión de datos)

CONMUTACIÓN DE PAQUETES

- Fragmentar al flujo de información, datos en bloques llamados paquetes.



PAQUETES



EN COLA.

- 2) A comienza una secuencia de paquetes.

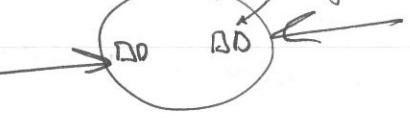
2) Nodo almacena y decide a que lado enviarlo. (ALMACENA Y REENVÍO).  
 MESSAGE STORE & FORWARD  
 { Almacena → EN COLA → DISPONIBILIDAD → ENVÍO.

3) Llegan al destino y deshace la fragmentación.  
 + ↑↑↑ Eficiencia.  
 { Nodos / Router de memoria, almacenan datos }

CABECERA (DESTINO)  
 NODOS (CAMINO)

  - Paquetes en cola de salida se puede utilizar el enlace / recurso.
  - Una vez recibido el mensaje, está en cola y luego se reenvíe a la velocidad del siguiente nodo o estación.
  - Se puede enviar más información a un nodo.

Paquetes en cola.



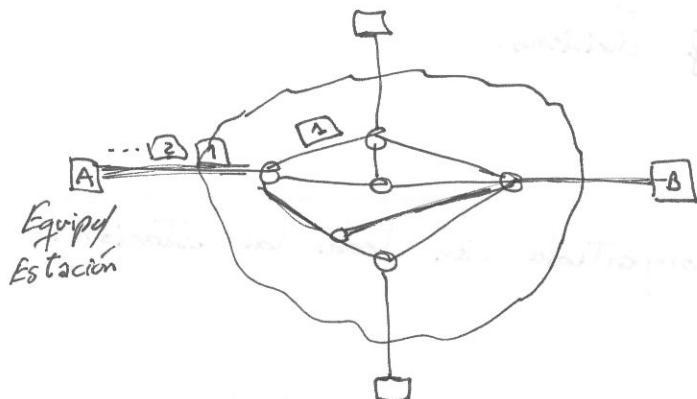
FULL-DUPLEX  
 HALF-DUPLEX

  - Es posible hacer conversiones de velocidad.
  - Si hay congestión se almacenan en la cola, pero se pueden aceptar paquetes. Tránsito saturado.
  - Establecen prioridades. FIFO (sin prioridades).
  - Canal libre, el siguiente paquete es aquel de mayor prioridad.

CANALES PUNTO A PUNTO

- Recursos reservados sin utilizar.
- Comunicación a diferentes velocidades no posible, tasa fija.
- No establece prioridades

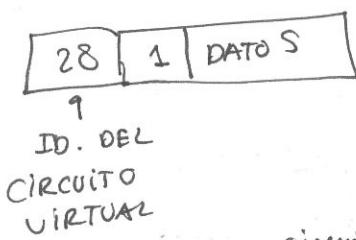
CONMUTACIÓN  
DE CIRCUITOS



→ Paquete =  $2K$ ,  $3K$  (3 paquetes)  
Almacena en un buffer.

→ Por donde viajan los paquetes. Encaminamiento de los paquetes.

- DATAGRAMA: los paquetes se tratan de forma independiente.  
El modo donde por donde envía los paquetes.  
Los paquetes pueden llegar desordenados.  
El destino **B** reordena los paquetes [TTFP].  
Los errores serán detectados de forma tardía ya que los datos son desordenados (DESORDENADOS).
- CIRCUITOS VIRTUALES: se establece una ruta predeterminada que tendrán que seguir todos los paquetes.  
Tendrán que seguir la ruta establecida del circuito virtual.  
Fase previa de establecimiento del circuito virtual.  
(No es conmutación de circuitos).  
Paquetes se almacenan en buffer, están en cola y se envían posteriormente, la ruta está indicada para el canal no está reservado solo está decidido el camino.



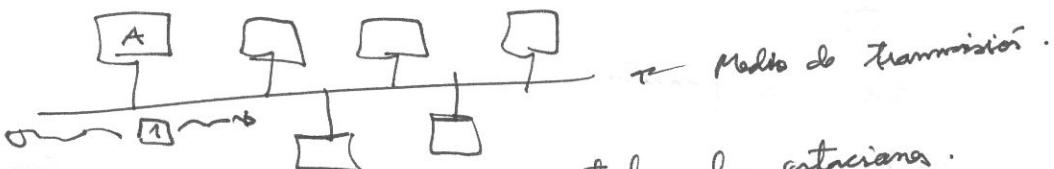
(Cabecera más compleja)  
/ No mira si hay camino más rápido o si este está saturado /

- Equipo/Estación: puede tener varios circuitos virtuales.
  - + Circuitos virtuales: es secuencial, los paquetes llegan en orden ya que es mismo camino para todos.  
Tienen que llevar al N° de paquete.

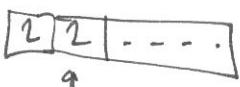
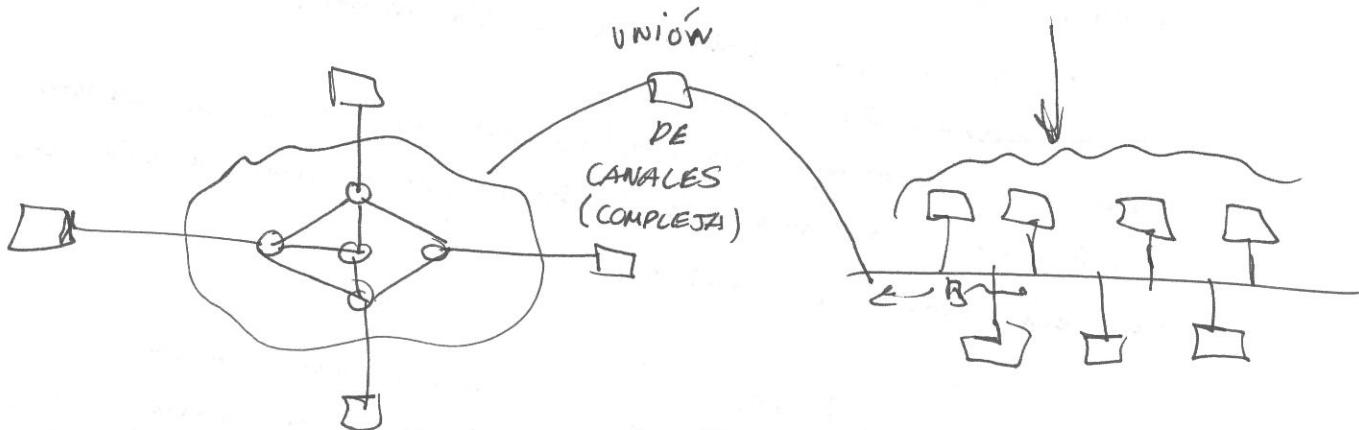
- + Control de errores: solicitud de envío.
  - + Servicios de control de flujo.
- Si **B** se queda sin memoria puede detener / parar la transmisión.
- + Puede incluir más rápido ya que al establecer la ruta el nodo de commutación no tiene que tomar decisiones.

## CANALES DE DIFUSIÓN

→ un único medio de transmisión compartido con todas las estaciones.

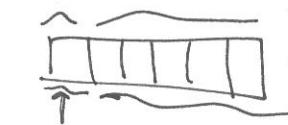


- El paquete se difunde y lo reciben todas las estaciones.  
Solo exceptúan aquellos que van dirigidos a ellas. (Identificador de ESTACIONES)
- DIFUSIÓN TOTAL:  
→ Un único paquete dirigido a todas las estaciones (BROADCAST).
- DIFUSIÓN RESTRINGIDA: Un único mensaje a unos cuantos estaciones.
- No hay mecanismo de enrutado.



Mensajes  
señalados al

grupu 1  
(Máquina se  
envíe a tantos  
grupos como quiera)



DIFUSIÓN  
1 => Dirección individual (PUNTO A PUNTO)  
2 => DIFUSIÓN

1 1 1 1 1 1 → Todos unir & FF  
(BROADCAST). Difusión para toda.

## CANALES DE DIFUSIÓN

↳ Si queremos que todos los equipos puedan comunicarse deberían utilizar la misma tara.

- RED LOCAL

### CLASIFICACIÓN EN FUNCIÓN DEL ALCANCE GEOGRÁFICO

- REDES DE AREA EXTENSA (WAN: Wide AREA NETWORK)

- REDES DE AREA LOCAL (LAN: LOCAL AREA NETWORK)

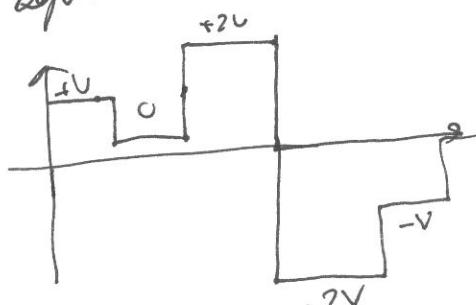
- REDES DE AREA METROPOLITANA (MAN: METROPOLITAN AREA NETWORK)

- REDES DE AREA PERSONAL (PAN: PERSONAL AREA NETWORK)  
(Bluetooth)

↳ EXAMEN

(Tara de Baudio Baudrate)

- Depende los niveles de la señal.



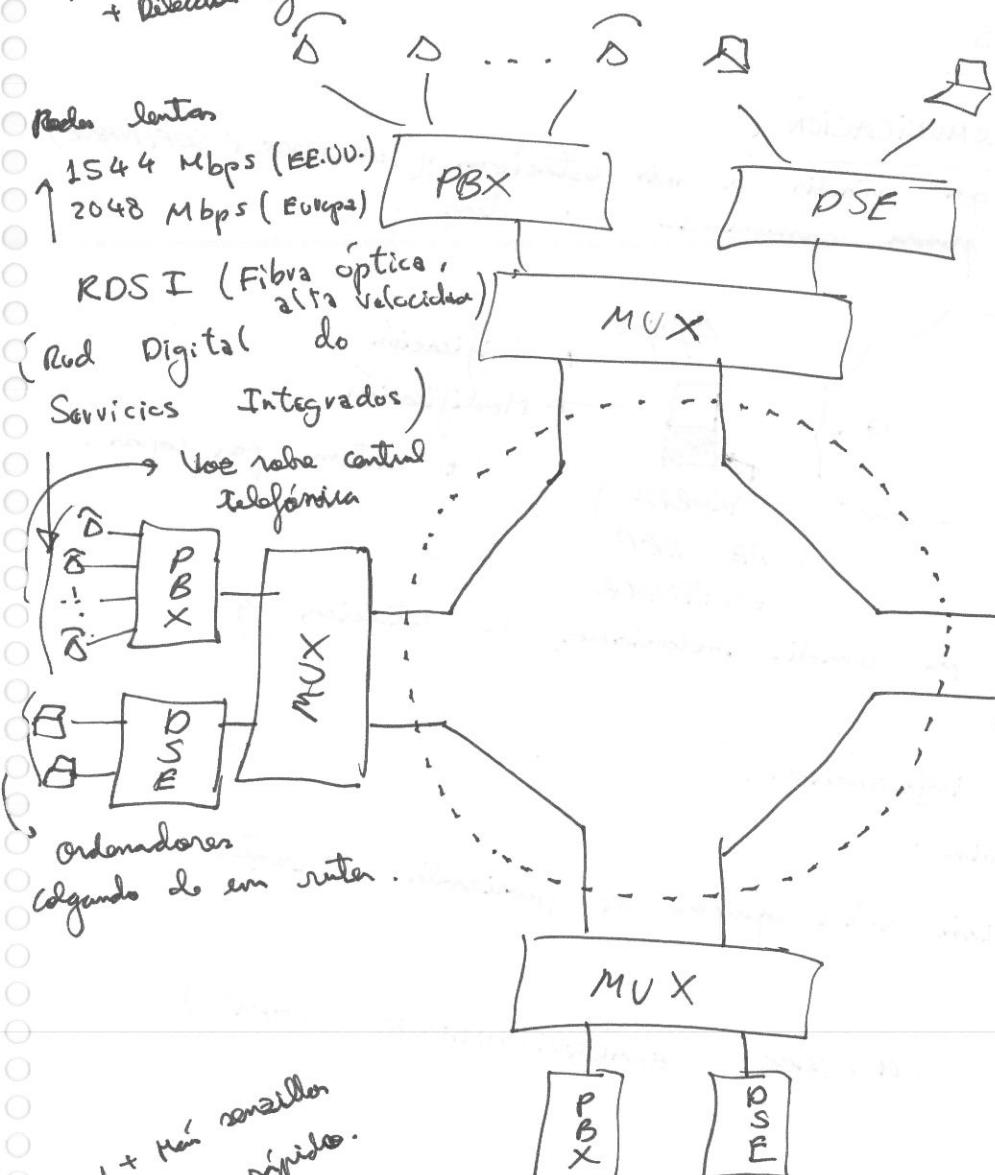
BINARIO

Baudrate = bits.



- WAN (Wide Area Network) : Red de área extensa.

- + Más compleja
- + Más robusta
- + Dirección y conexión de errores. Red privada corporativa



→ Para transmitir voz.

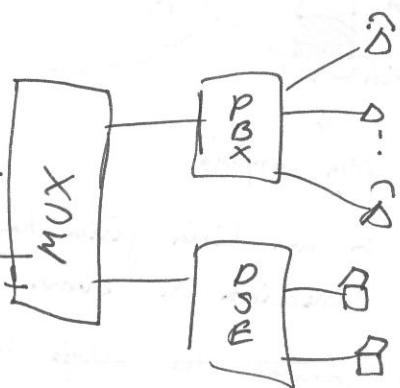
PSTN (Public Switched Telephone Network)

→ Cuando los costos son elevados y había poca información a transmitir.

PSDN (Public Switched Data Network)

→ Redes para transmitir datos

RETD (Red Española de Transmisión de Datos)



+ más sencillo.  
+ más rápido.

- LAN (Local Area Network)

(Referencias con WAN)  
+ Alcance geográfico.

+ Velocidad de transmisión.

+ Propiedad privada de los medios.

+ BER menor. (tasa de errores es menor)

↓  
Fiabilidad.

Netwrok) : Red de corto alcance.  
Velocidades altas.

(Estaciones + red):

la propia actividad o propietaria de los ordenadores y la red (LAN), así WAN no

## MAN (Metropolitán Área Network) :

- ↳ Tipo de red intermedia
- ↳ Transmisión de datos rápida
- { CAN rápido  
WAN lento }  
que cubre la red local intermedia dentro de una ciudad.

## ARQUITECTURA DE REDES

### SUBSISTEMA DE COMUNICACIÓN

- ↳ Es aquello que tengo que añadir a mi estación. (HARDWARE + SOFTWARE)
- ↳ a un DTE para que pueda comunicarse con otras.



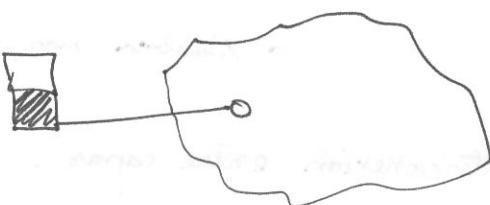
- Muy complejo
- En una línea física hay que añadir mecanismos de detección y corrección de errores.
- Control del flujo de información.
- Encriptado de información.
- Problemas de interconexión entre equipos de fabricantes diferentes.

### SISTEMA DE CAPAS (MÓDULO JERÁRQUICAMENTE - CAPAS)

- Detectar errores.
- Corregir errores.
- Enrutar datos.

## ARQUITECTURA DE REDES

### SUBSISTEMA DE COMUNICACIONES



TARGETS DE REP

DRIVERS

CARGAR LA PILA TTFP.

SOFTWARE (PROTOCOLOS)

### SISTEMA MONOLITICO

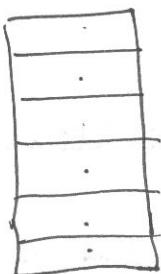
- Problemas:

+ Verificación (funciona y no hay errores)

+ Modificación (parar de línea a fibra)

- Dividir el problema en subproblemas (capas). Cada capa será especialista de una parte de las comunicaciones.  
El objetivo de capa es ofrecer unos recursos a su capa inmediatamente superior.

Capas de la torre OSI.

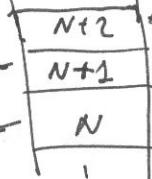


N (Capas)

Capa proveedora de servicios

Capa usuaria de servicios

Protocolos invertidos



Hacer uso de transmitir bits  
Transmitir bits

+ Entre la capa N y N+2 no hay ningún tipo de contacto.

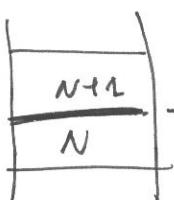
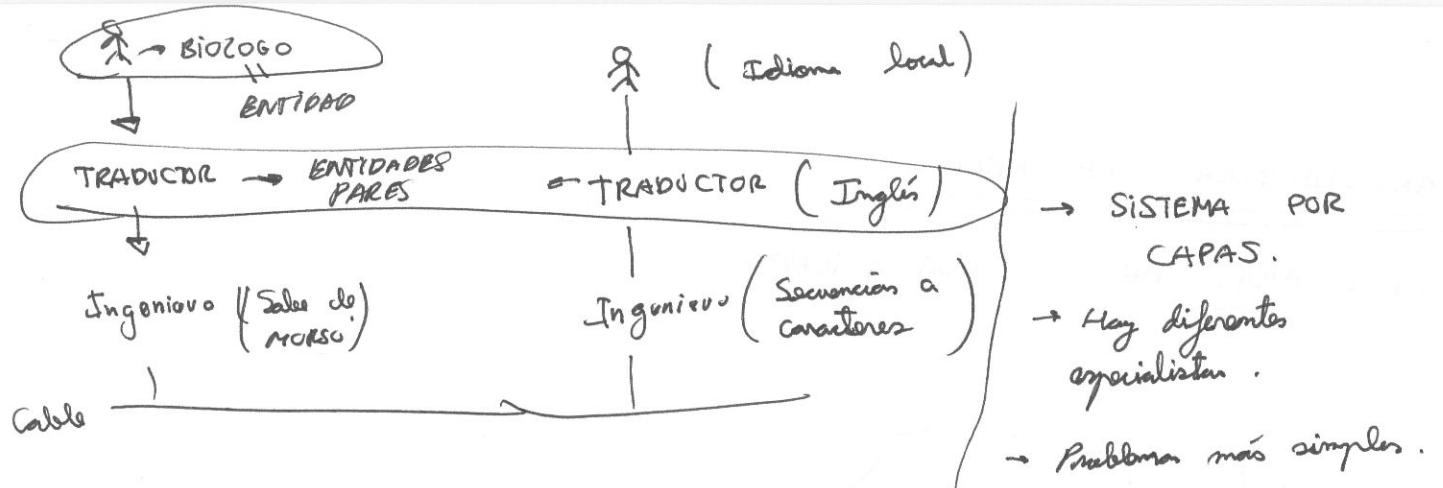
BIOLOGO  
DE  
KENYA

BIOLOGO  
INDONESIO

+ Entender lenguajes (protocolos)

Sistema  
monolítico.

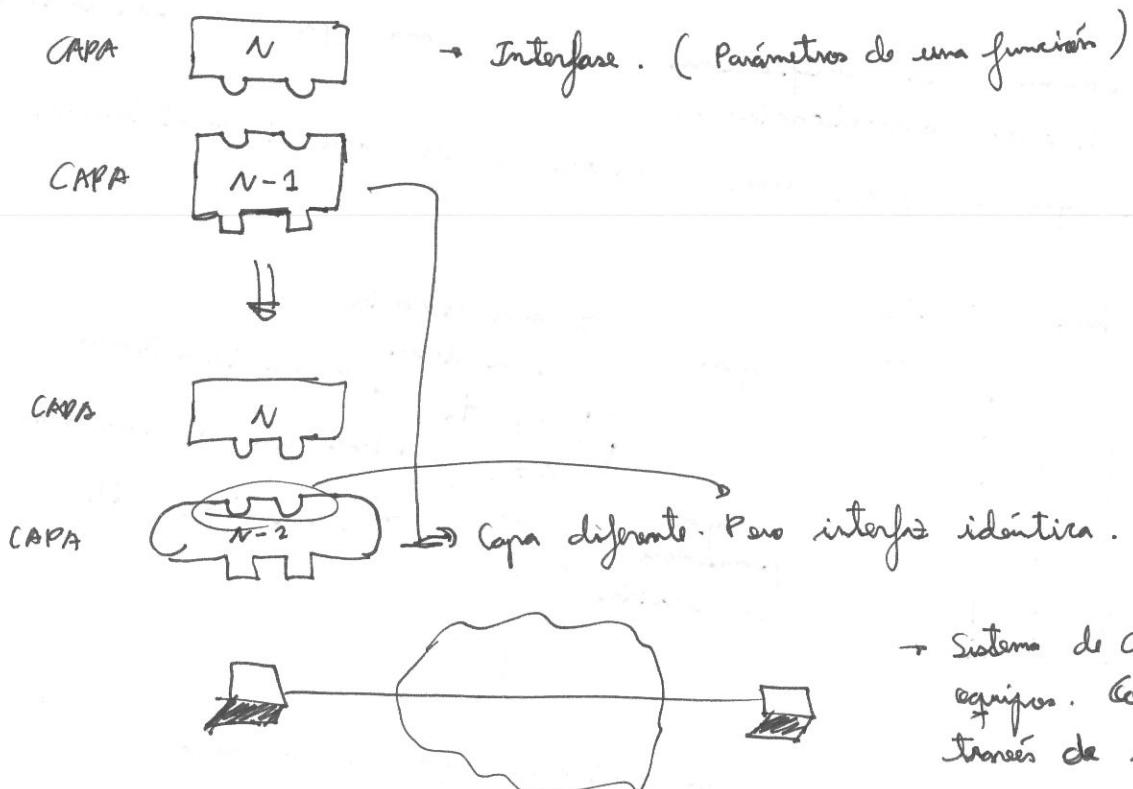
Monajes via telegrama.  
Código MORSE.  
FDICIAS  
MORSE  
BICR060



→ Interface: punto de interconexión entre capas.  
 - Definición clara de las funciones de cada una de las capas.

### SISTEMA POR CAPAS

- Proceso de comunicación más sencillo.
- Modificación más sencilla.  
 (COBRE A FIBRA, Cambiar TARGETA DE RED)



→ Sistema de Capas en ambos equipos. Comunicación a través de protocolos.

**ENTIDAD :** Elementos activos que se encuentran en cada uno de los capas. ( Software o Hardware )

Aquello que implementa la función asignada a una capa.

**ENTIDADES PARES :** Aquellas entidades que forman las capas correspondientes de máquinas correspondientes.

Entidades en la misma capa en diferentes máquinas.

Aquellas que se hablan.

Aquellos que se hablan.

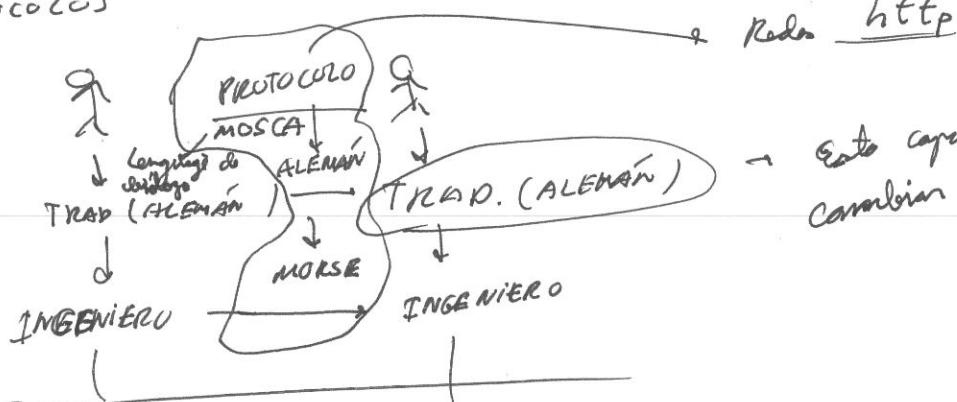
En los 2 extremos habrá el mismo número de capas.



**PROTOCOLO :** Conjunto de reglas, normas o convenciones utilizadas por las entidades pares.

( NORMAS ) → CAPA  $n$  → CAPA  $n$

↓  
PROTOCOLOS



Redes http

→ Estas capas se pueden combinar

**ARQUITECTURA DE RED o PROTOCOLOS :** conjunto de capas en el que se ha dividido el subsistema de comunicación y los protocolos que se implementan en cada una de las capas.

- N° de capas
- Protocolos de cada capa.

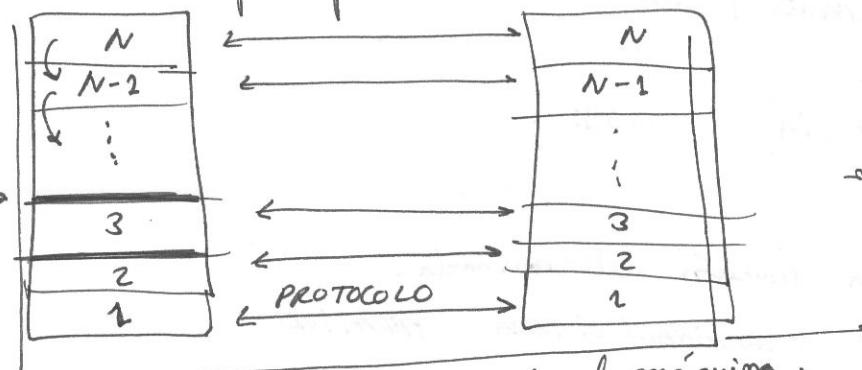
} ARQUITECTURA DE RED



## COMUNICACIÓN VIRTUAL - COMUNICACIÓN REAL

Subsistemas:

Protocolos entre capas pares.

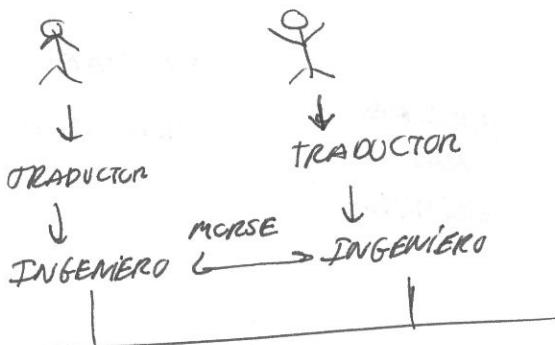


→ Puedo hacer capas vacías? una capa que haga la misma que 2 capas del otro sistema.

→ Capas pares hablan entre ellas.

Comunicación real en vertical.

→ Las interfaces quedan ocultas dentro de la máquina.



MODELO DE REFERENCIA (ISO DE REFERENCIA) ( MODELO DE REFERENCIA OSI ) (OSI)

→ Sistema monolítico y de capas.  
- Cada fabricante define sus capas. SISTEMAS PROPIETARIOS: únicamente los equipos del mismo fabricante pueden comunicarse.  
Limita la posibilidad de interconexión entre ordenadores.

SOLUCIÓN: Adoptar un conjunto de normas acordadas internacionalmente. (ESTÁNDARES)  
de forma que cualquier equipo que las adopte pueda comunicarse.  
ABIERTO.

SISTEMA ABIERTO. [open source].

→ ISOS: señal, norma (guía)

Organización Internacional de Normas (ISO)  
International Standard Organization for Standardization  
La organización ISO definió un modelo de referencia para la interconexión de sistemas abiertos: Open Systems Interconnection (OSI)

## ISO :

- N° de capas de un subsistema de comunicación.
- Identificar la función de cada una de las capas.

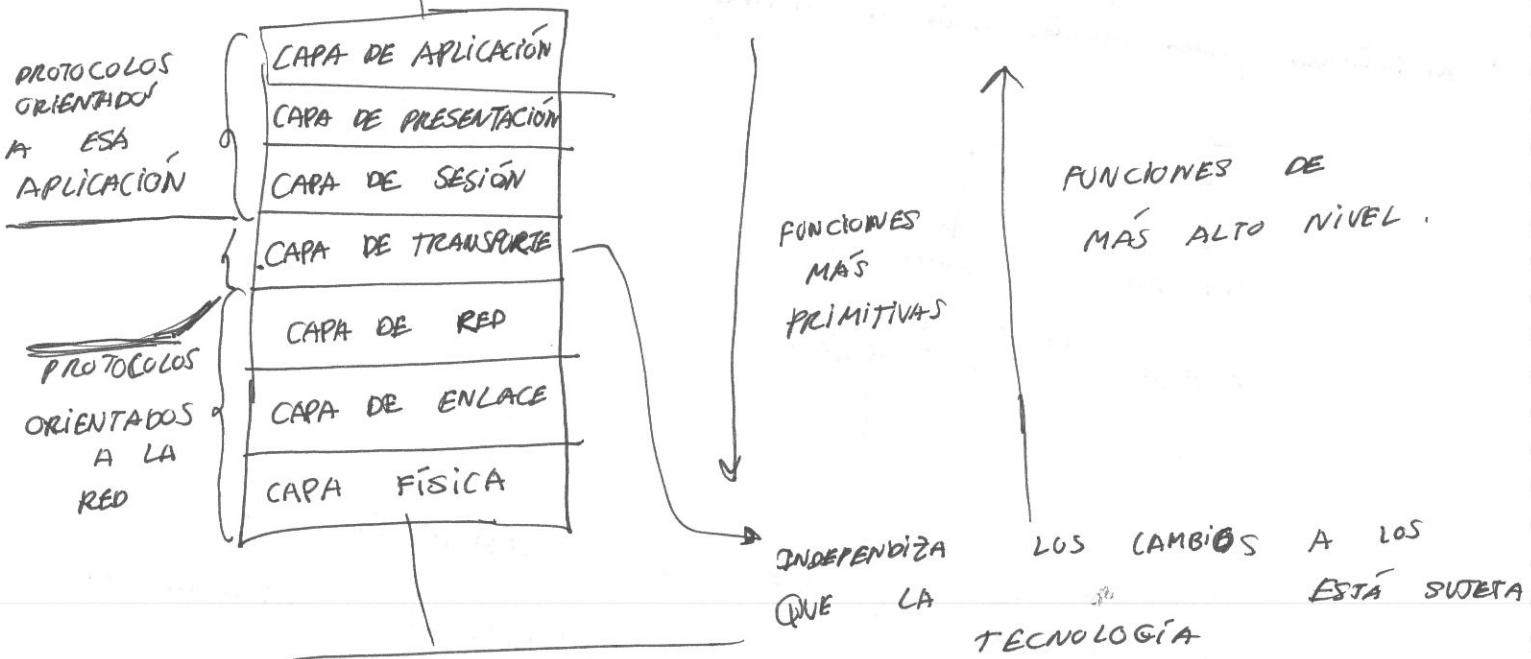
CAPAS ↑↑ , ↑↑ INTERPASES : LENTITUD ↑↑

## • MODELO DE REFERENCIA OSI .

+ Tiene ↑ capas .

+ Cada capa tiene una función determinada .

TERMINAL / SISTEMA OPERATIVO

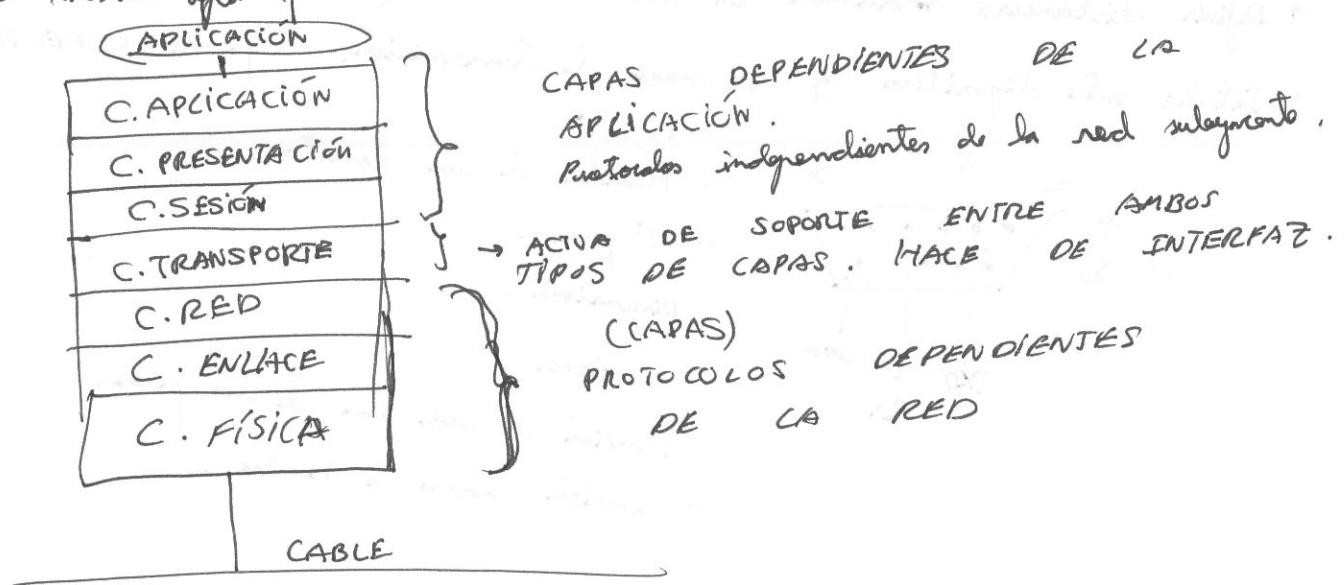


## EXAMEN !!!!!!

- El modelo de referencia OSI no es una arquitectura y de red ya que los protocolos no están definidos. (las capas si lo están)

ISO EN SUMODELO DE REFERENCIAOSI

- ↳ Modelo para evitar que cada fabricante utilice sus normas y protocolos tener alternativos para no tener que aplicar estas normas o saltárselas. (No depender de las normas del fabricante).
- En una misma red pueden haber muchos estandares.



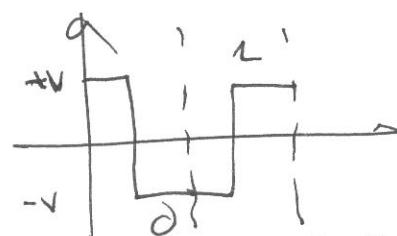
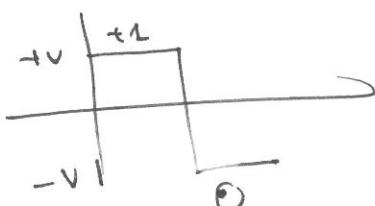
- CAPA FÍSICA (obligatoria)
- ↳ Cubre la interfaz física entre el dispositivo de transmisión y el medio a comunicar y las reglas mediante las cuales puede enviar bits a través del cable / medio de transmisión.

→ Transmisión de bits a través del medio físico de transmisión de una forma lo más óptima posible.

(poner el bit y asegurar que se recibe ese mismo bit).

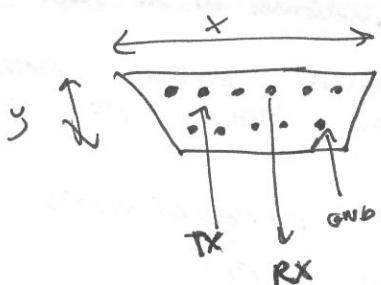
→ Tengo que decidir con cuantos voltios se representan un 1 y un 0.

- Codificación: que representa cada uno de los niveles.



→ Codificación Manchester.

- Cuanto tiempo tengo que mantener una señal para entender un 1 o una 0.
- Potencia de la señal.
- Tipo de codificación y modulación.
- Intentando asegurar que los bits enviados se reciben correctamente.
- Definir distancias máximas en un cableado. (Debido a atenuación).
- Interfaz entre dispositivo y al medio de transmisión. Tipos de CONECTOR.



- Formas
- Medidas de un conector.
- N° pines.
- Dimensiones.
- Conexiones.
- Función de cada uno de los pines.
- Conexión macho o hembra.

- Como me conecto y como transformo la señal del cable en bits.
- Definición de las características mecánicas - { Forma  
Medidas}

- Definición de las características eléctricas. | Codificación  
Representación de cada uno de los bits

- Protocolos de cada física: | V. 24,  
X. 21,  
RS-232C

- INTERFASE = CONECTOR entre CABLE
  - Interfase queda oculta
  - No interviene los protocolos.
- \* CAPA FÍSICA

F50 EN SU

---

MODELO DE REFERENCIA

---

OSI

- Cuanto más software más lento.
- Cuanto más hardware más cara.

CAPA DE ENLACE

→ ISO : si quieras hacer algunas funciones que yo digo tienen que usar esta capa.

- Le llegan los bits de la capa física.



→ N : una servicio.

→ N-1 : proceso servicio.

- Función : hacer o convertir en finales un medio de transmisión no fiables. Ofrecerlo a la capa de red en medio de transmisión ideal donde no hay errores.

- Fragmentará la información.

→ Anadir información adicional : mecanismos de detección y corrección de errores.

- Si detecta errores , se protocolo deberá pedir retransmisiones al origen. Puedo recibir el mismo fragmento 2 veces . (novo mensaje se puede perder).

- Duplicado otro tipo de error.

→ Errores :

Suprimir bits	Anadir bits
---------------	-------------

- La capa de enlace debe solucionar estos errores.

→ Existe que son transmisor muy rápido世家 un receptor que no puede procesar los tramas a la velocidad que se envían . ( CONTROL DE FLUJO)

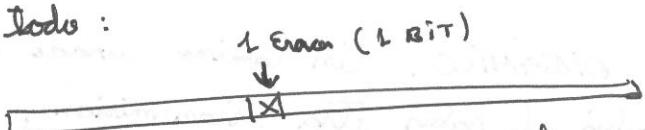


CAPA DE ENLACE (FRAMES) (TRAMA) - Unidad de datos.

Intentan ofrecer un medio de transmisión libre de errores a la capa de red.

- Entramado / encapsulado de información.
- Hacen fragmentos pequeños si la información es larga y añaden bits para saber donde empieza y acaba una trama.
- Lo tiene que hacer de forma que el receptor no se confunde.
- Bits adicionales de códigos detectores y correctores de errores.
- Capa de red usa en servicio / recurso libre de errores.

- Envío todo:



→ si fragmento s error, solo afecta a un fragmento.

MTU: 1500 bits. (MÁX).

Pila OSI  
(conjunto de 7 capas)

→ Capa de red ( Paquetes ).

→ Overhead: enviar mucha información y grandes datos.

→ Datos en orden y sin duplicación.

→ Control de flujo: en TX (transmisor) rate en RX (receptor) muy lento.

→ Consentir un fiel medio de transmisión no fiable.

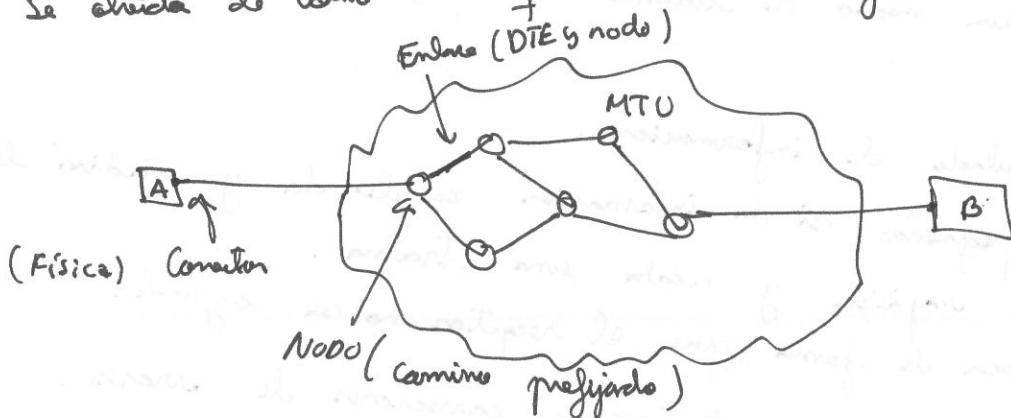
→ Control de flujo.

→ Ejemplos protocolos de capa de enlace:

+ LAPB, HDLC, BSC

## CAPA DE RED

↳ Se encarga de como hacer que los datos lleguen correctamente.



→ Capa de red: gestión de la subred de comunicaciones.

Encaminamiento o enrutamiento de paquetes.

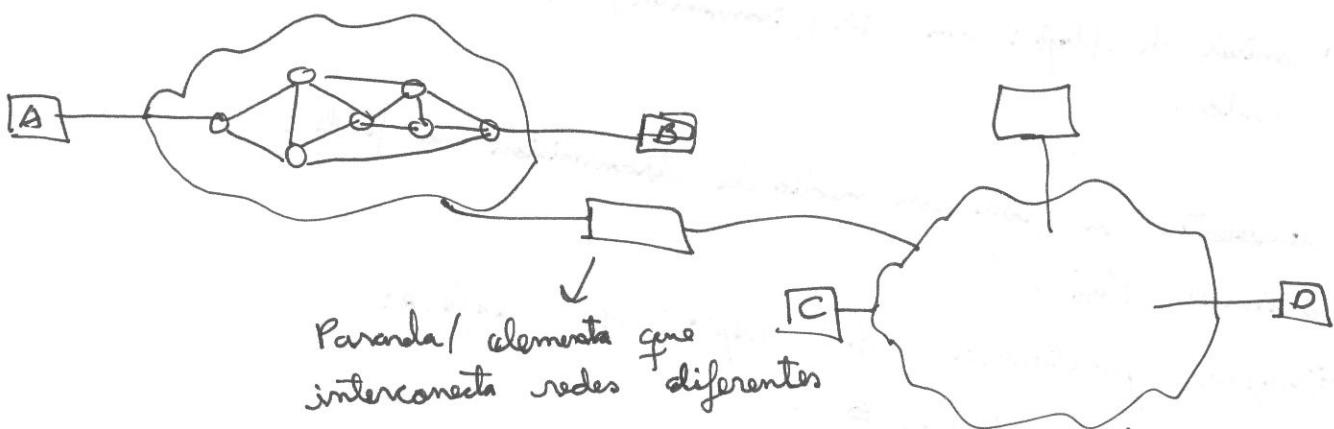
- Nodo (Camino prefijado). TABLAS DE ENCAMINAMIENTO ESTÁTICAS.

- ESTABLECIMIENTO DE RUTA DINÁMICO. Un camino usado durante el tiempo de transmisión. Cambia al haber otra transmisión (la ruta).  
Modelo datagrama.

→ Capa de red asegura un camino para alcanzar tu destino.

→ Control de la congestión o intento de evitar dicha congestión.

→ Conocer el tráfico / consumo de los usuarios.

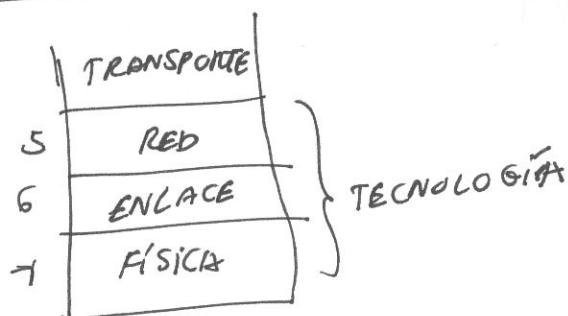


→ Se encarga de resolver problemas de conexión entre redes heterogéneas.

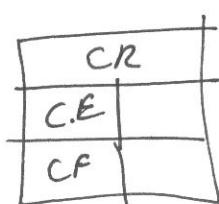
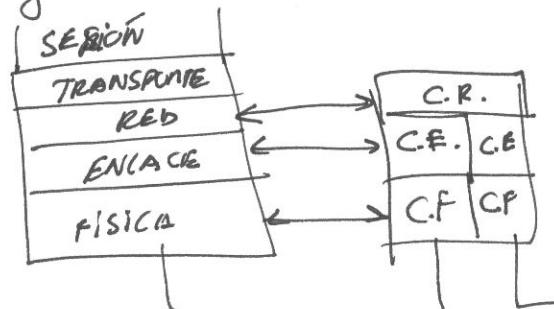
## CAPA DE RED

- Control del enrutamiento / encaminamiento entre DTE's en comunicación.
- Control del tráfico en la red.
- Tarificación.
- Intercconexión de redes heterogéneas.
- Ejemplos: X.25 nivel 3  
IP [ Internet Protocol ]
  - Cómo direcciona cada una de las estaciones.
  - Cómo se establecen rutas.

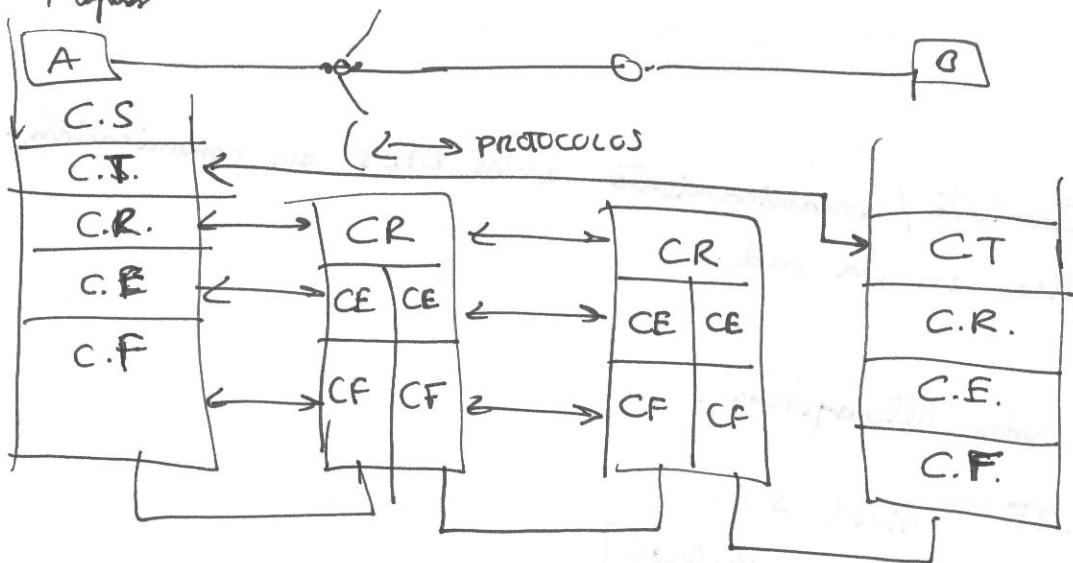
## CAPA DE TRANSPORTE



- Capa de Transporte:
    - ↳ Actua de interfaz entre capas 2, 2, 3 y 5, 6, 7.
    - ↳ Es una capa que llamaremos de extremo a extremo a origen/destino.
  - Protocolos (CAPA FÍSICA / ENLACE / RED) que se comunican con él.
- CONMUTADOR (tiene 3 capas)
- Nodo de conmutación de 3 capas.

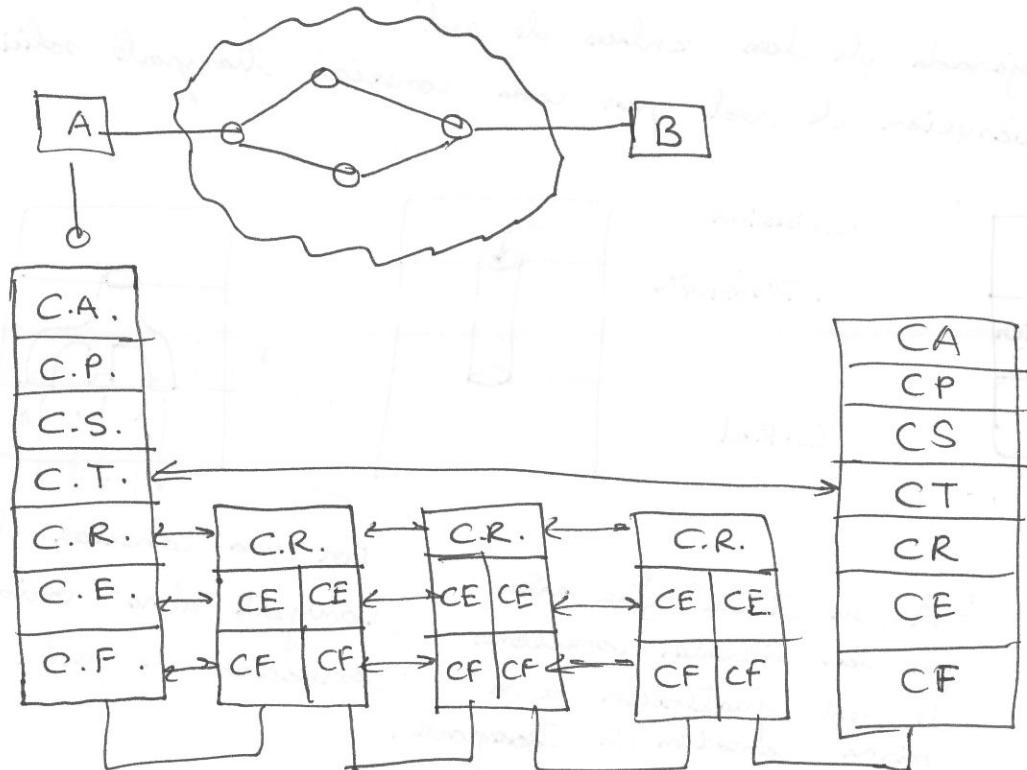


7 Copiar



C.T. (Copa destino - origen)

{ C.R.  
C.E.  
C.F. } → No en copa destino - origen tiene protocolos entre sí antes de llegar de equipo a equipo.

CAPA DE TRANSPORTE

- ↳ Primera capa origen/ destino extremo a extremo
- Asignará los mensajes de la capa de sesión, los dividirá en lo que sea necesario.
- Asegurará que los paquetes lleguen correctamente al destino, en transporte fiable.

**Funciones**

Llegan al destino.

Sin duplicados.

En la secuencia correcta.

- Al ser la primera capa extremo a extremo comprueba si la secuencia es correcta.

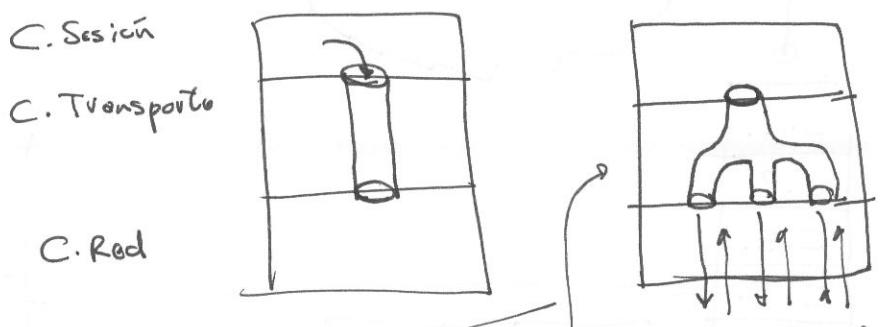
Pretende asegurar que no haya errores extremo a extremo.

Si puntualmente ha hecho errores extremo a extremo.

CAPA ENLACE (PUNTO A PUNTO)  
CAPA TRANSPORTE (EXTREMO A EXTREMO)

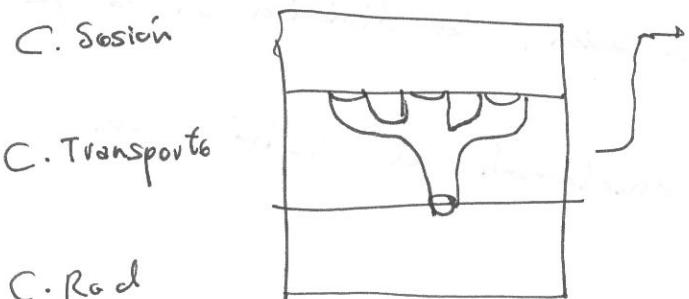
} Evitan errores

- Si la tecnología es fiable no necesitaré algoritmos complejos.  
Si los paquetes llegan ordenados no hay que reordenarlos.
- Utilización mejorada de los enlaces de red.  
Se abre una conexión de red por cada conexión transporte solicitada.



A su tendré que saber que las diferentes conexiones de red pertenecen a la misma conexión de transporte.

Por una conexión de transporte abro varias conexiones de red.



Capa de sesión pide varias conexiones de transporte.

Abro una sola conexión de red para minimizar costos.

Este conexión de red se corresponde con varias de transporte.

- + Mejorar el caudal.
- + Aprovechar la conexión de red.

VARIOS FLUJOS EN UNO, O UNO DIVIDIDO EN VARIOS.

- Muchas conexiones de transporte abiertas.  
A que conexión/applicación acaban siendo.  
FUNCIONES DE CONTROL DE FLUJO. Para evitar interrupción de TX a RX.

CAPA DE TRANSPORTE

- Aísla capas superiores de la tecnología de red.
- 1º Capa extremo a extremo.
- Garantiza la transmisión extremo a extremo.
- Control de flujo de extremo a extremo
- Optimización de rendimiento. (económico o de velocidad)

CAPA DE SESIÓN

- Ofrece otros servicios que a veces son necesarios y a veces no lo son.
- Sesión: cualquier tiempo en el que un usuario / aplicación accede a una máquina remota.

- Permite ofrecer gestiones de interacciones, un control del diálogo entre equipos.

En muchos casos con los servicios de la capa de transporte será suficiente.

- Control del diálogo. (Half Duplex o Full Duplex)
- Este función HD o FD es propia de la capa de sesión.

- A veces una comunicación Full-Duplex FD si se quiere realizar la misma operación, habrá un testigo que elegirá que equipo realiza la operación.

TESTIGO: controla que equipo puede realizar la operación.

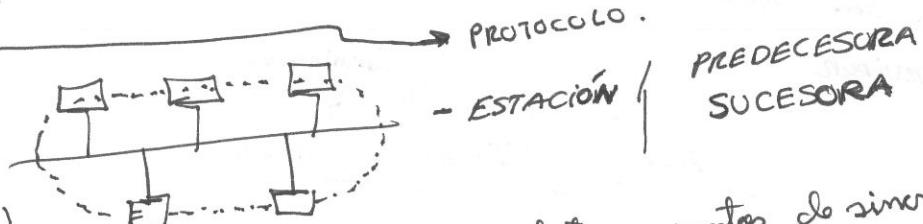
- Podría estar vacía si no se quiere controlar quien envía.

CAPA DE SESIÓN

(Puedo no ser necesaria).

- Niveles OSI.
- Capas dirigidas a aplicaciones. Servicios adicionales a aquello que ofrece un transporte ordinario.  
(capa de transporte).
- Controlar el diálogo entre los extremos.
- Queremos controlar la gestión de interacciones, control del diálogo.
- Controla quien tiene el turno en cada momento.
- Ejemplo: no procesará el mensaje que tiene hasta que se envíe el que ya tiene.
- Agrupamiento de datos.
- No hay obligación de tener protocolos en todas las capas.
- Gestión de interacciones.
- Gestión de testigos (tokens). Solo el equipo que posea el testigo puede realizar la operación. (Solo uno puede hacerlo en un momento dado).

token bus: es una red con un medio de transmisión compartido que se monta en anillo lógico y se pasa un testigo.

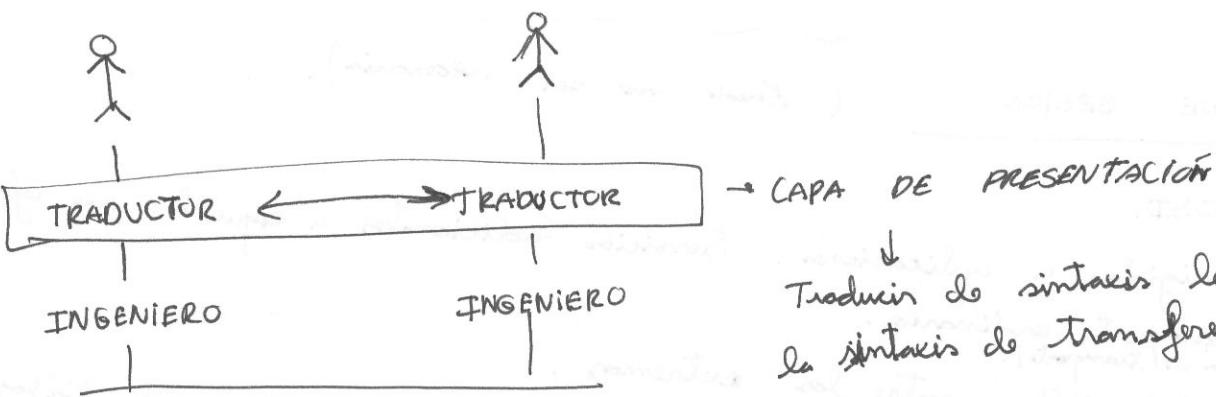


(SINCRONIZACIÓN).

- Puedo establecer en un transporte de datos puntos de sincronización para reestablecer una comunicación si esta se ha perdido.

CAPA DE PRESENTACIÓN

- ↳ Representación de los datos → <sup>'ábs'</sup> <sup>'á'</sup>
- Envío de caracteres / ASCII, EBCDIC, ISO - 8859
- Números C<sub>01</sub>, C<sub>02</sub>
- Negociando una sintaxis de transferencia convenida entre ambos extremos.

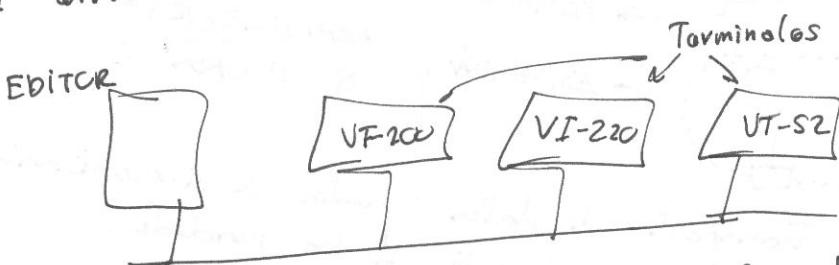
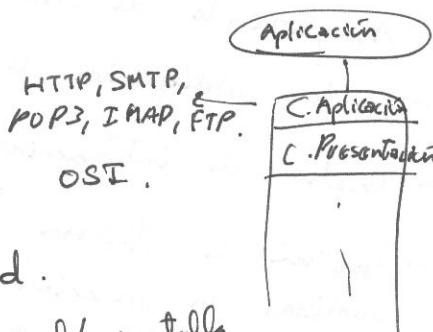


↓  
Traducir de sintaxis local a la sintaxis de transferencia.

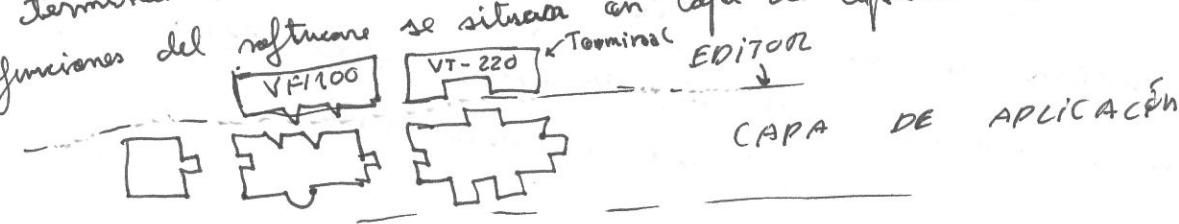
- Compresión de datos; para que estos ocupen menos.
- Afecta a la representación de la información.
- Negociar la sintaxis de transferencia.
- Encriptación de datos. Afecta a la representación de los datos.
- Entorno de interconexión de sistemas de transferencia.

### CAPA DE APPLICACIÓN

- Lo punto de acero de las aplicaciones con el sistema OSI.
- Interfaz de usuario a servicios distribuidos de red.
- Red con diferentes terminales: editar orientado a terminal / pantallas que envíen los servicios de control.



- Terminal virtual de red donde escribe el editor.
- Terminal (tiene un software/protocolo) capaz de generar comandos para un terminal virtual de red.
- Las funciones del software se situarán en capa de aplicación.

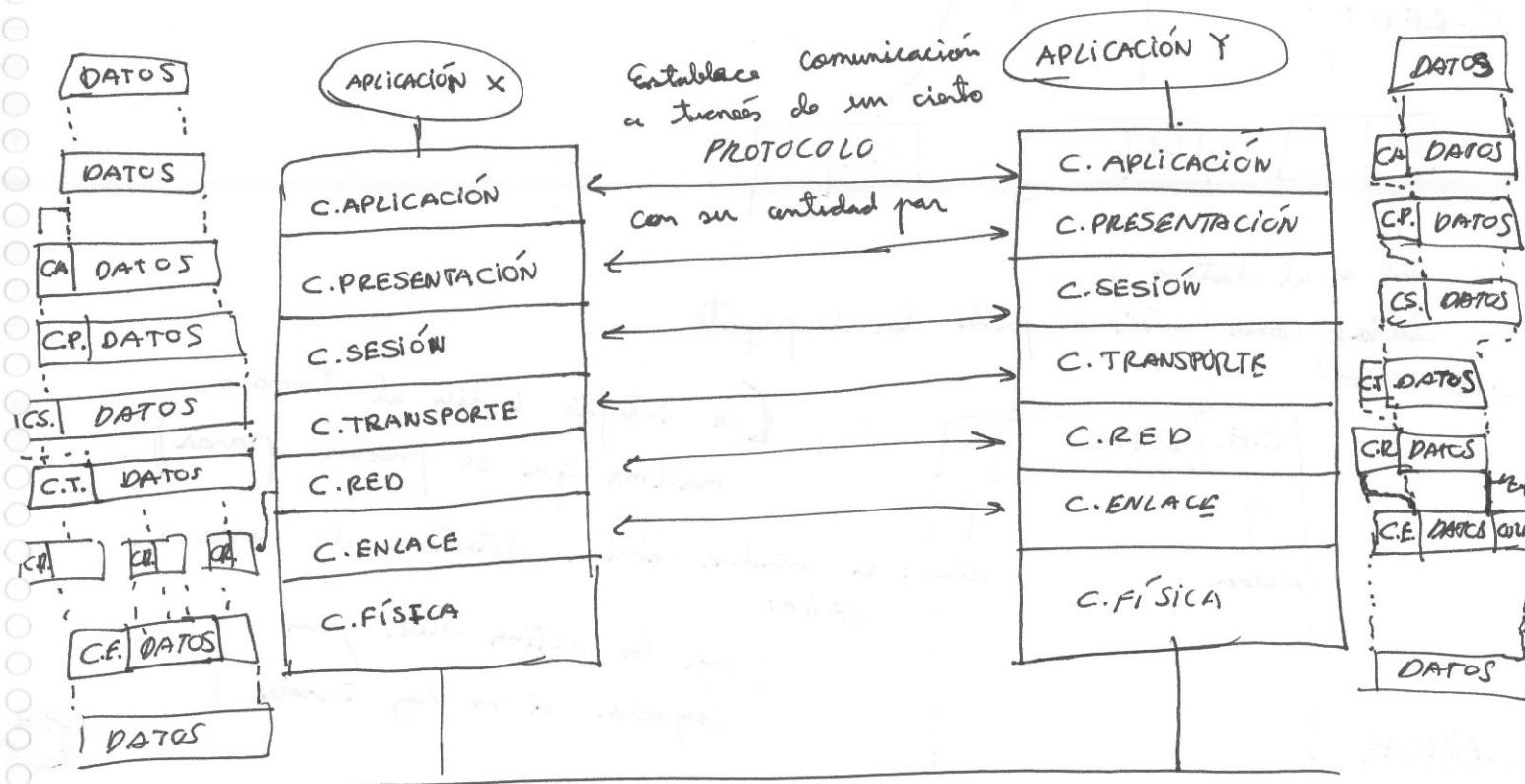


- Protocolos para interpretar código HTML
- Transmitir, modificar ficheros.

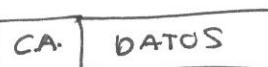
# TRANSMISIÓN DE DATOS EN EL MODELO OSI

Capa de aplicación: capa de contacto entre aplicaciones y el subsistema de comunicación.

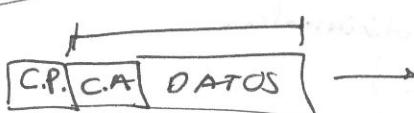
C. = CAPA



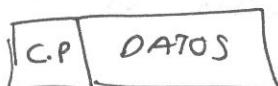
C. APLICACIÓN: Añade una cabecera. / Transporta datos o un mensaje de control  
 ↑  
 Información de cabecera.



C. PRESENTACIÓN: Envía el logotipo completo.



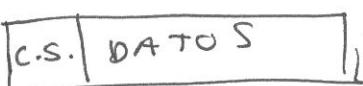
Para C.Presentación



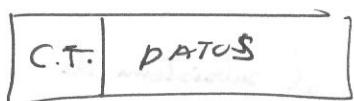
DATOS ≠ del mensaje original.

C.A y DATOS son DATOS  
 (Así para toda la capa)  
 → Le para el mensaje completo  
 a la C. Sesión.

C. SESIÓN:

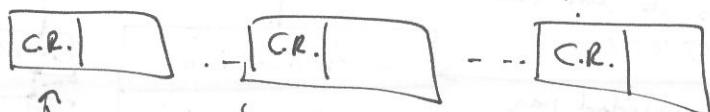


### C. TRANSPORTE:



→ Rebe ayeran los paquetes y recomponer el orden.

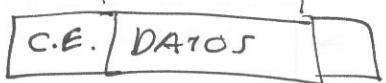
### C. RED:



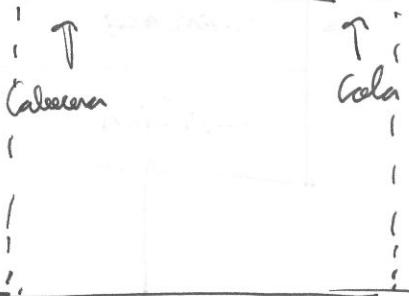
¿Cuál es el destino.

Saltos como el máximo puede dar el paquete.

### C. ENLACE:



(La interfaz indica el tamaño máximo que se pueden pasar).



Cola: se añaden códigos detectores de errores.

RECIBE:

(Coge los últimos bits para comprobar si no hay errores)

### C. FÍSICA:



\* Algunas de las capas pueden no necesitar cabeceras.

\* Ninguna cabecera de una capa inferior jamás pasa a una superior.

\* Comunicación real en vertical



\* Para la capa de aplicación el mensaje también es transparente.

- En cada una de las capas se añaden bits adicionales.

- Ancho de Banda / Aumentado en control }

200 Mbps no implica 2s → 200 Mb, ya que hay cabeceras.

Puede tardar un 20% más.

Eficiencia & Errores.

- La tasa de errores no depende de la pila TCP.

PDU (Protocol Data Unit)

↳ Mensajes entre capas . Unidades de datos de protocolo .

A - PDU

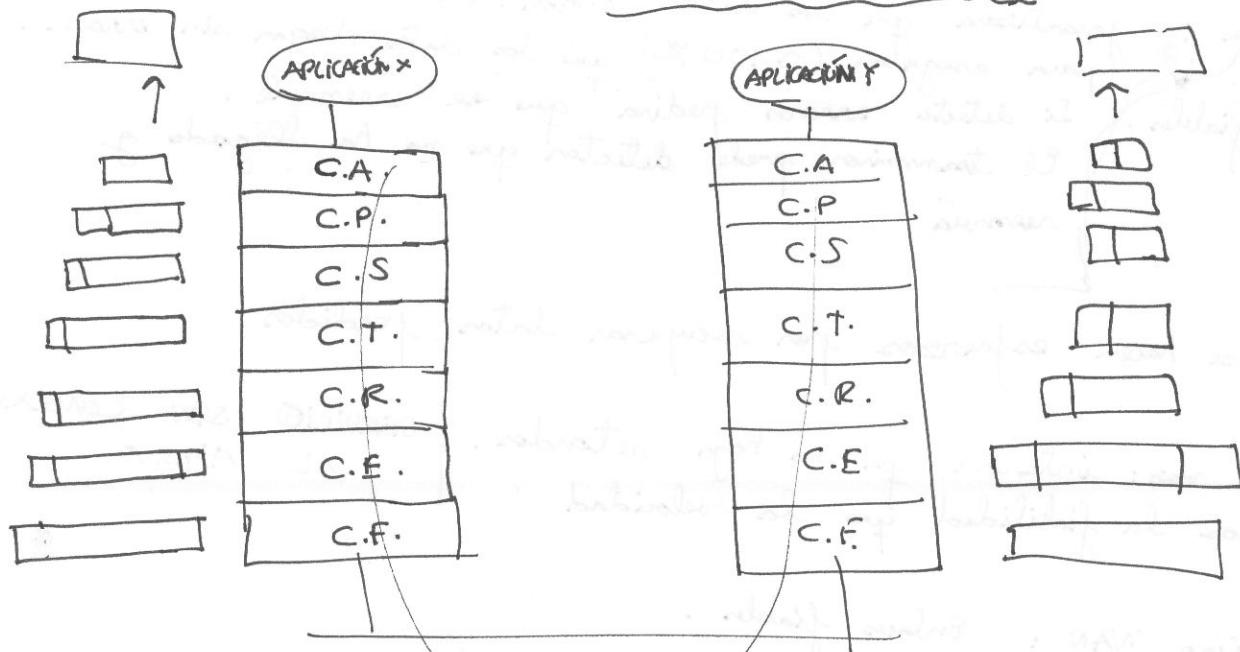
P - PDU

S - PDU

C. Física: Bits

C. Enlace: Tramo ( Frame ) → ( En el Switch )

C. Red: Paquetes → ( En el router ) .



- Los cabeceras añadidos a una capa, nunca pasan a otras superiores.
- Cabeceras y protocolos. [200 Mbps, 100 Mb no se enviaría en 1 segundo].

### SERVICIOS EN EL MODELO OSI

↳ Se accede a través de unos puntos llamados SAP.

SAP = SERVICE ACCESS POINT.

↳ Interfaz a través de capa superior que le ofrece la capa inferior (capa proveedora).

Redes : SAP (Sockets) - Puertos. Se identifican con unos números.

Puerto 21 → Servidor FTP

Puerto 25 → SMTP.

- Servicio / Modo de funcionamiento orientado a conexión.

- Servicio / Modo de funcionamiento sin conexión.

↳ Modelado como un sistema postal.  
Cada mensaje se envía de forma independiente del resto de mensajes.  
Los datos pueden llegar fuera de orden.

Modelado como el sistema telefónico.

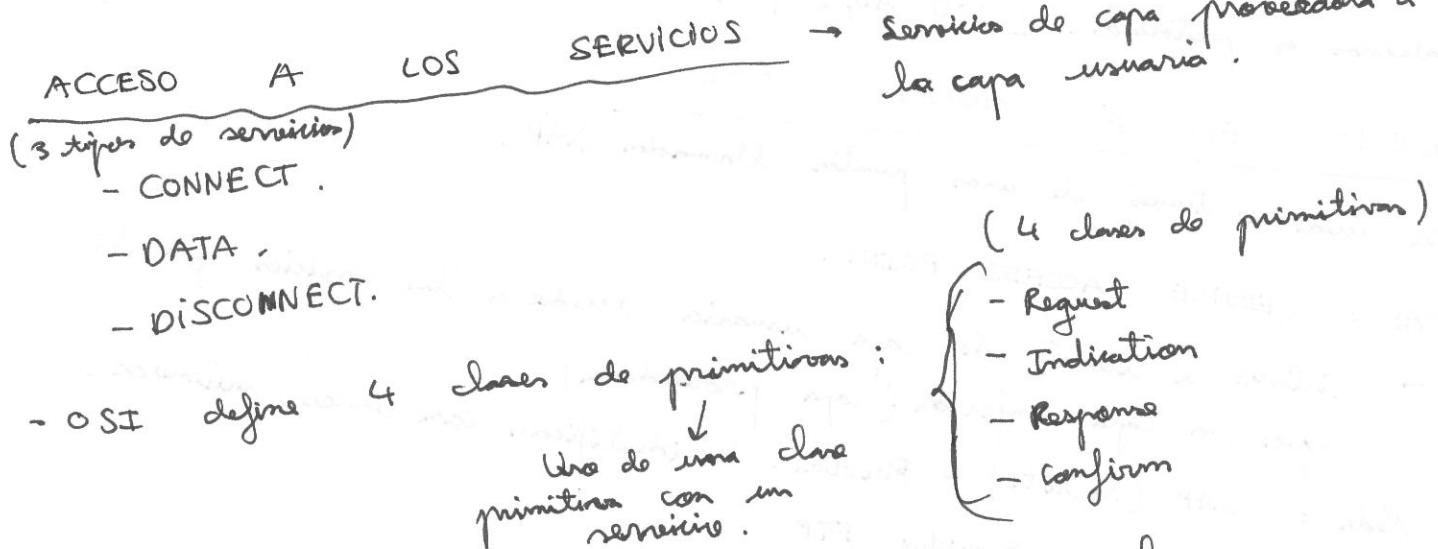
- Se establece una conexión.
- Tránsito de datos.
- Se finaliza la conexión.
- Datos se reciben en el orden en el que se enviaron.

- fiables: garantizará que los datos llegan. Proveerá los mecanismos para asegurar ( $\geq 100\%$ ) que los datos llegan sin errores.
- no fiables. Si detecta errores pedirá que se reenvíe. El transmisor debe detectar que no ha llegado y reenvíe.

→ No se hacen esfuerzos por recuperar datos perdidos.

- Reenvíos de voz: interverá que no haya retardos.
  - Para mermar la fiabilidad que la velocidad
- |            |              |
|------------|--------------|
| SERVICIO   | SIN CONEXIÓN |
| NO FIABLE. |              |

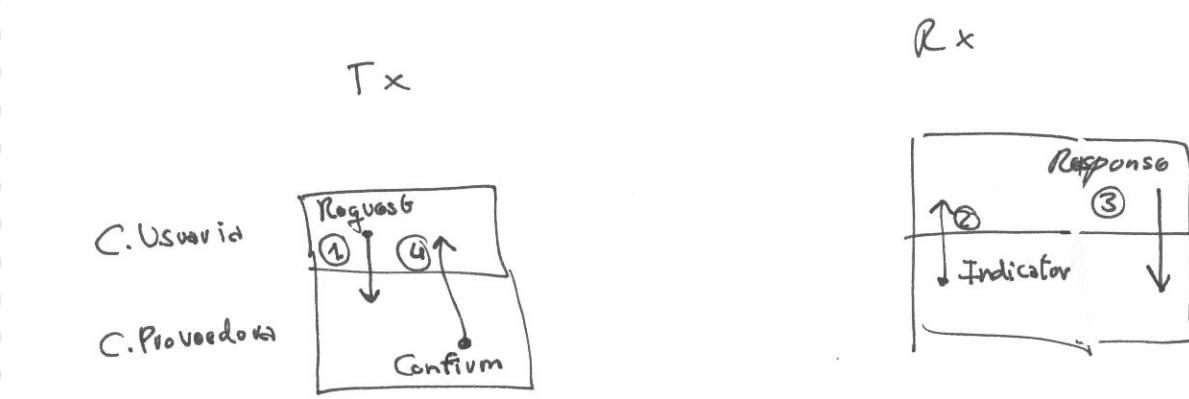
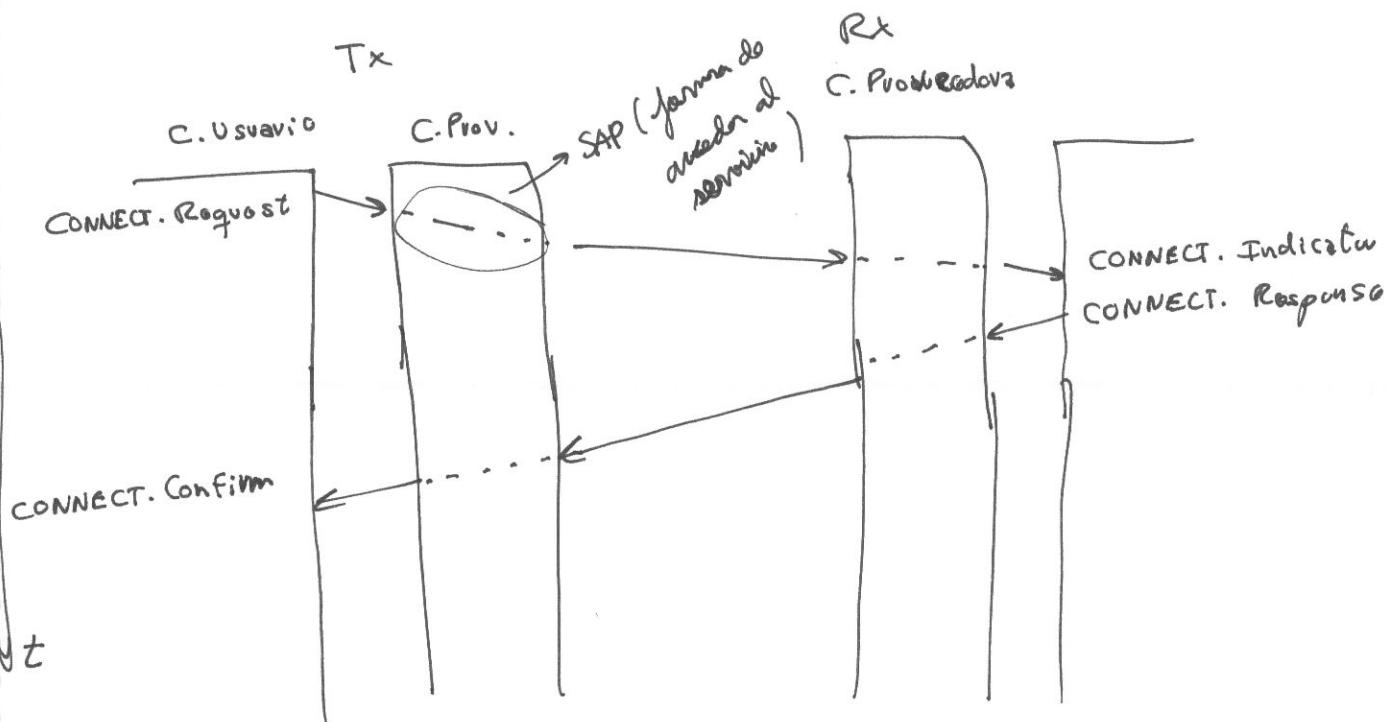
- En redes tipo WAN: Enlaces fiables.
- En redes tipo LAN: Enlace no fiables. Ya que los errores son muy pocos debido al medio de transmisión.



- (4 clases de primitivas)
- Request
  - Indication
  - Response
  - Confirm

- Request: sirven a la capa usuario para indicar a la capa precedente que se ejecuta algún tipo de acción.
- Indication: la capa usuario para recibir una indicación. Alguien quiere corregirte contigo.
- Response: la capa que en el destino remoto puede responder.
- Confirm: la capa usuario que habrá hecho al request sabe como ha quedado su solicitud.

- DIAGRAMA DE SECUENCIA TEMPORAL



SERVICIOS

- **Confirmados:** Tiene las 4 clases de primitivas
- **No Confirmado:** Tiene 2 clases de primitivas
  - Un servicio no confirmado puede ser fiable.

Request  
Confirm  
Response  
Indicator

Request  
Indication

EXAMEN

→ Si el servicio de la capa de inferior ofrece ya fiabilidad.

SERVICIOS

CONNECT

DATA

DISCONNECT

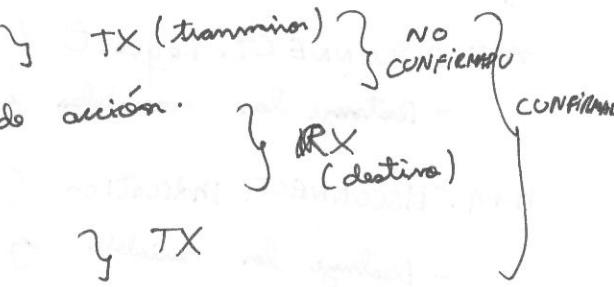
PRIMITIVAS

Request (Solicitud)

Indication (Aviso) : inicio de acción.

Response (Respuesta)

Confirm (Confirmación)

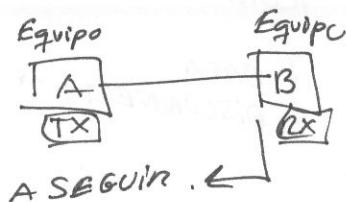


- Servicio no confirmado puede ser fiable o puede no serlo.
- Servicio confirmado fiable por naturaleza.

Transmitir un fichero.

→ orientado a conexión. → FIABLE.

→ PASOS



**TX** 1.- CONNECT.request (B, A, transferir fichero 50 KB a 200 Mbps)  
 SAP origen, quien soy yo, donde debo mandar.  
 SAP destino  
 Agre puesto, máquina, socket.

- + No identifico.
- + Lo que quiero (velocidad).

+ Destino.

**RX** 2.- CONNECT.indication (A, B, transferir fichero 50 KB a 200 Mbps)

→ (Rechazo)

**RX** 3.- CONNECT.response (A, B, OK a 20 Mbps)  
 → porque no pude ir a 200 Mbps.  
 - Si propone 20 Mbps es porque no pude ir a 200 Mbps.  
 Si hay discordancia, se establece la velocidad menor.

**TX** 4.- CONNECT.confirm (B, A, OK a 20 Mbps)  
 → (Rechazo)  
 - Si hubiera rechazo, terminaría aquí la conexión. (No habrá falta DISCONNECT).

**TX** 5.- DATA.request (B, A, datos)

**RX** 6.- DATA.indication (A, B, datos)

→ Tantas veces como sean necesarias para enviar 50 KB.  
 DATA.request      DATA.request (Así hasta acaben).  
 DATA.indication    DATA.indication

← Podriamos usar (Resposta i' Contum) en data.

n.- DISCONNECT. Request (B, A, <motivo>)

- Destruye las variables y libera los recursos del TX.

n+1.- DISCONNECT. Indication (A, B, <motivo>)

- Destruye las variables y libera los recursos del RX.

CONNECT → Servicio confirmado

{ DATA  
DISCONNECT → Servicio no confirmado.

## SERVICIOS DE CADA CAPA

### CAPA FÍSICA

↳ Tipos de medios de transmisión

- Tipos de codificación:

- + Manchester
- + Manchester diferencial.
- + Bipolar.

Par trenzado  
Coaxial  
Fibra óptica  
Aire

Es un medio imperfecto, con errores.

- Tipos de modulación:

- |     |              |
|-----|--------------|
| A8K | (Amplitud)   |
| FSK | (Frecuencia) |
| PSK | (Fase)       |

- La señal sufre con la distancia, tiene retardos.

### CAPA ENLACE

↳ Crear un medio de transmisión perfecto, libre de errores.

Hacer un medio de transmisión perfecto que por naturaleza es imperfecto.

↳ Interfaz de servicios definidos a la capa de red.

- Interfaz de servicios definidos a la capa de red.

- Entramado y encapsulamiento de los datos.

- Fragmentar los datos a enlace. El RX debe saber localizar tramas.

2 0 0 0 1 0 1 0 . . . 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1

↑ ↑ ↑

¿Cuál es el límite de inicio de trama?  
+ Métodos fiables y eficientes para encontrar el inicio y el final de la trama sin que ninguna combinación posible de errores.

- Sincronización de bits.

- Detección y corrección de errores.

- Mecanismos de control de errores.

→ Agregar bits de CRC.

Si hay errores pedir que se retransmita.

- Si me llega la trama.

- Aplicado de la trama.

- TX a velocidad mayor que RX  
(CONTROL DE FLUJO)

CONTROL DE ERRORES

- En capa de enlace, 2 equipos físicamente conectados.



- + Controlar la tasa de transmisión de tramas.
- + TX debe saber si RX tiene memoria y recursos disponibles.

- Gestión de enlace.
  - Variables a inicializar y destruir cuando se haya acabado la transmisión. Control de recursos.

- Interfaz de servicios a Capa de Red.

- + Servicios sin conexión - sin arentamiento
- + Servicio sin conexión - con arentamiento
- + Servicio orientado a conexión (fiable)

(Datagrama)

Cada trama tiene la dirección completa del destino, no hay condición previa, no se liborean recursos, no hay garantía que los datos hayan llegado.

Ventaja

Transmisión Rápida.

Inconvenientes

Si una trama se pierde no se hace ningún esfuerzo por recuperarla.

No sabe si ha llegado o si ha llegado bien.

Es útil cuando la tasa de errores es muy baja.

La conexión de errores delegada a capas superiores.

Transmisión de voz/video en tiempo real.

- ① - Servicio sin conexión sin asentimiento.  $\rightarrow$  CAPA FÍSICA SIN ERRORES
- ② - Servicio sin conexión con asentimiento.
- ③ - Servicio orientado a conexión.

④ Capa de Enlace no vale si la trama llega solo la envía.  
 Capa de Enlace si llega no vale si llega corrupta o con errores.  
 Servicio Datagrama.

⑤ No se establece conexión previa y no se libera la conexión.  
 Se confirma la recepción de cada trama individualmente.

Capa de Enlace asegura que las tramas llegan.

Capa de Enlace asegura que las tramas llegan sin errores.

Es más lento, espera confirmaciones.

El receptor puede recibir dos veces la misma trama. (DUPLICADO).

Servicio Datagrama con asentimientos.

⑥ Servicio fiable y más sofisticado que puede ofrecer la capa de enlace a la capa de red.

Las tramas llegan.

Las tramas llegan sin errores.

Las tramas llegan sin duplicados.

Las tramas llegan en orden.

Ofrece un medio de transmisión libre de errores (fiable).

Ofrece un medio de transmisión libre de errores (fiable).

Capa de red es un equivalente de un flíjio de bits fiable.

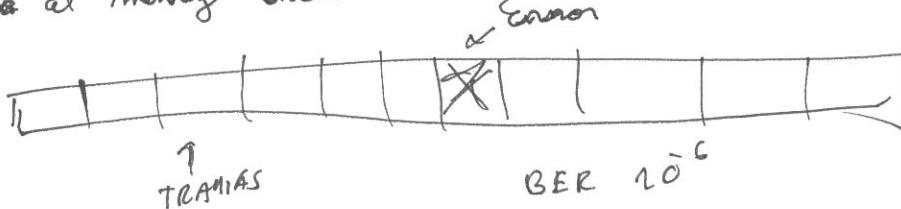
dejar

Servicio no confirmado no quiere que sea fiable.  $\rightarrow$  EXAMEN

### CAPA DE ENLACE $\rightarrow$ ENTRAMADO

Lo fragmenta el flíjio continuo de bits de la capa física.

(TRAMAS o BLOQUES): Para si hay errores, enviar solo una trama y no todos al mensaje entero.



→ Camas marcan los límites de una trama:

200 110010 ...

↑ ↑

↓ Dónde empieza la trama?

① - Inclusión de espacios de tiempo.

② - Cuento de caracteres.

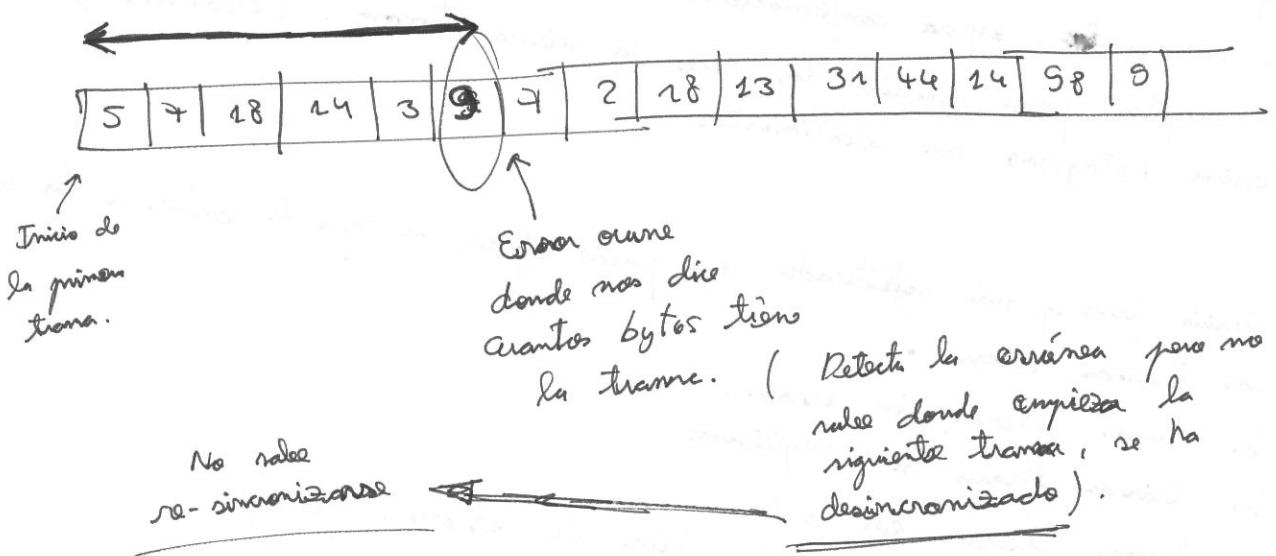
③ - Caracteres de inicio y final con inserción de carácter.

④ - Banderas de inicio y final con inserción de bytes.

⑤ - Violación de código en la capa física.

① → Rezar un tiempo entre tramas no vale. ✗

② → Añadir una calecera indicando el número de bytes. ✗



ENTREMAZO

- - Inclusión espacios en blanco.
- - Cuenta de caracteres.
- (3) - Caracteres de inicio y final con inserción de carácter.
- - Banderas de inicio y final con inserción de bit.
- (5) - Violación de código en capa física.

① ✗ NO.

② Problema de re-sincronización. MÉTODO NO FIABLE.

✗ NO

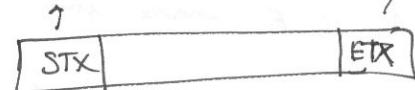
③ Conseguir que Rx encuentre un inicio lógico de trama.

- Inicio de trama con carácter STX.

START TEXT : STX      Ascii: 2

- Final de trama con carácter ETX.

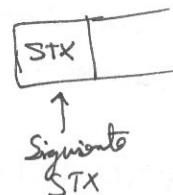
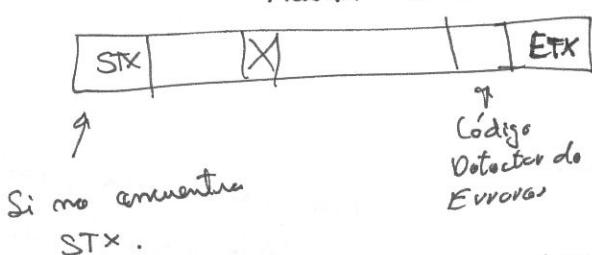
END TEXT : ETX      Ascii: 3



- STX, ETX son caracteres no impresibles para encapsular texto.  
(caracteres impresibles)

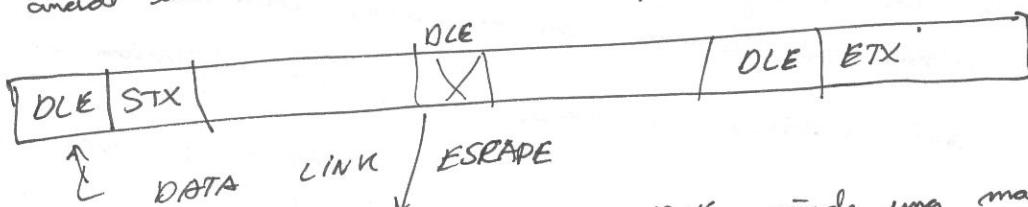
TRAMA n

TRAMA n-1

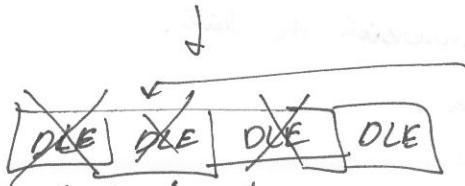
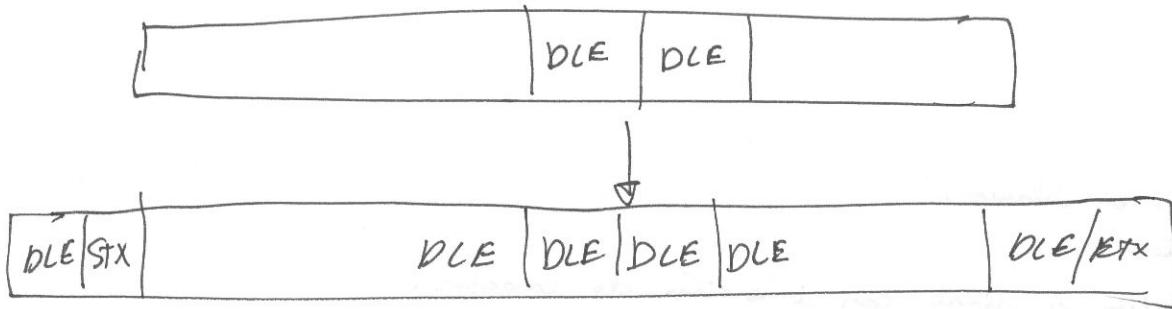


Verá al siguiente STX y se resincronizará.

- Si se envían datos binarios puros. [000 00011]  
Se puede confundir con ETX intermedio al encapsular datos binarios puros.
- Se añade un DLE. (Se hace menos probable el error).



Si encuentra un DLE añade una marca previa a ese carácter.



Detecta que el primero son datos y no marca de datos.

→ En este caso DLE no hay problemas de resincronización.

RX: DLE añade un DLE.

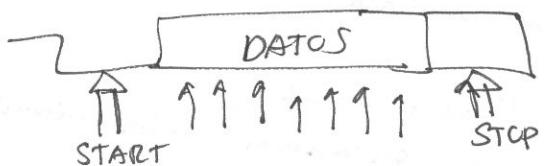
BISYNC (FBM) ← PROTOCOLO.

Sincronización a nivel de trama.

→ DLE | STX | DLE/ETX :

- Previo hay que la sincronía de bit, la sincronía de byte y posteriormente la sincronía de trama.

### TRANSMISIÓN ASÍNCRONA



Rx tiene una fuente de reloj múltiplo natural de la tasa de transmisión.

→ En el centro del bit muestra.

→ 32 flancos de reloj generan un pulso de lectura.

→ Nunca será un múltiplo exacto.

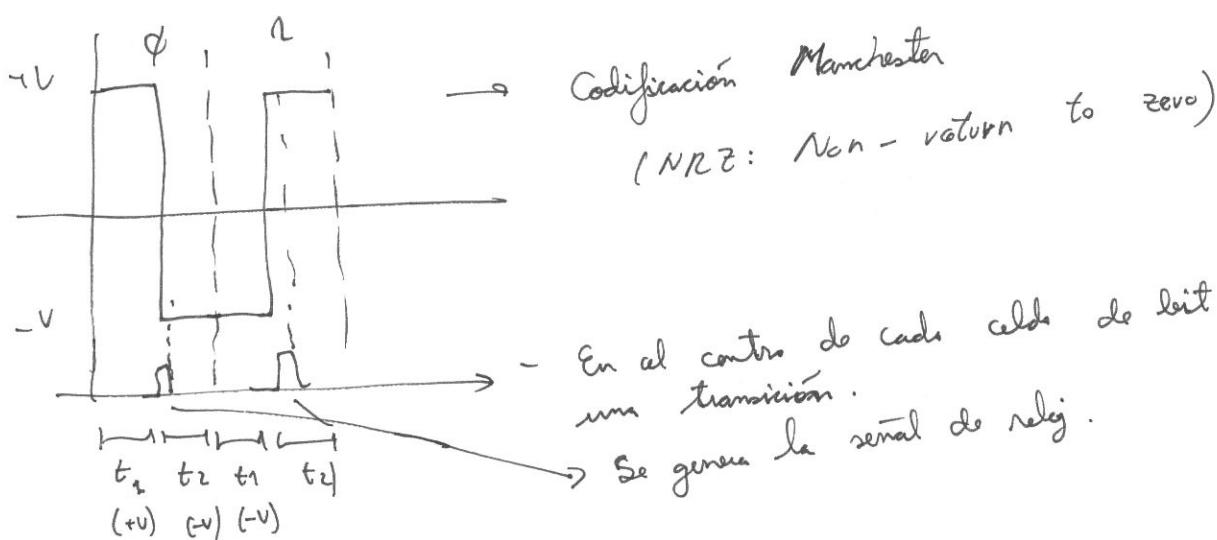
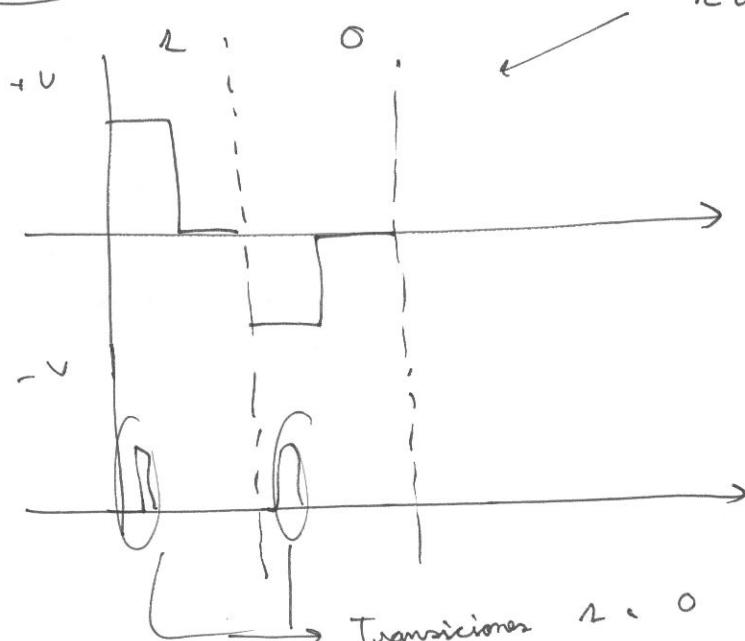
→ Reloj Rx trabaja en sincronía con la señal recibida.

→ Señal de reloj enviada a través de: { la codificación de la señal.

DPLL : Genera transiciones de 1's y 0's para sincronizar continuamente.

### TRANSMISIÓN SÍNCRONA

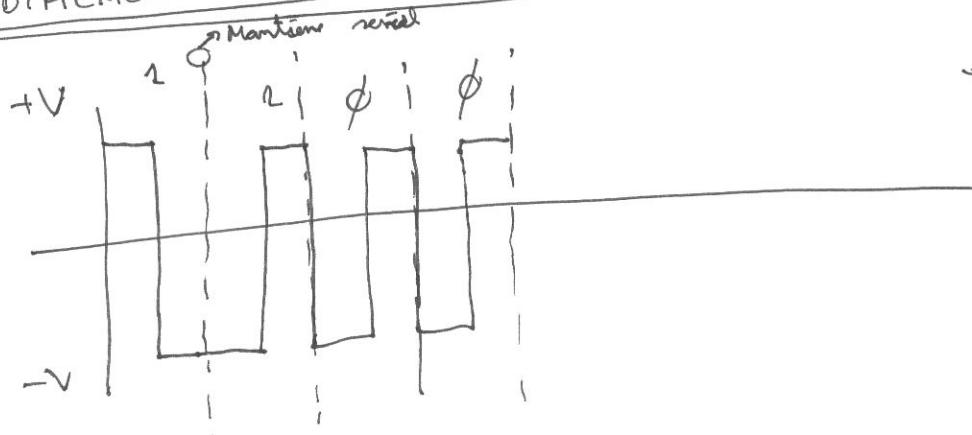
### TRANSMISIÓN SÍNCRONA



$$\Delta t = \frac{\Delta t}{2}$$

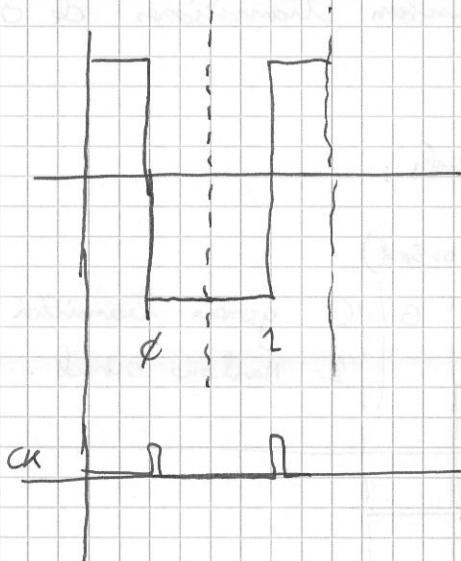
$$t_2 = \frac{\Delta t}{2}$$

### CODIFICACIÓN MANCHESTER DIFERENCIAL



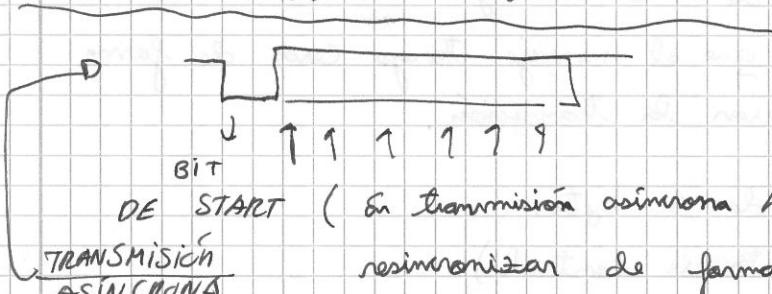
- } 1 : No hay transición
- 0 : Hay transición a principio de celda de bit.
- En el centro de cada celdas de bit hay transición, hay reloj sincronizado con los datos.

Como me sincroniza a nivel de bit

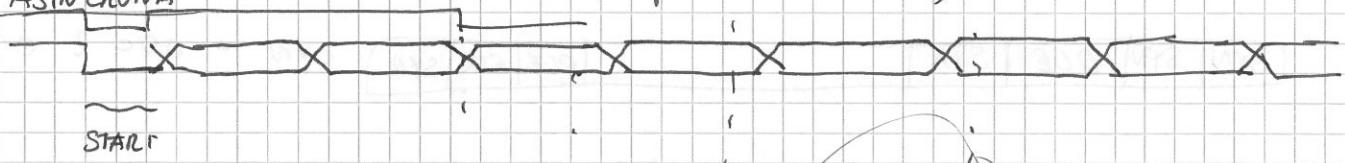


DPLL → DIGITAL PHASE LOCK LOOP.

↳ Lazo de fijación de fase digital



TRANSMISIÓN ASÍNCRONA (En transmisión asíncrona hay bit de start para resintonizar de forma continua.)



→ Reloj múltiplo de la tasa de transmisión (local)

(entre (un poquito a la derecha)



\* 32

M M M M  
(32 pulsos)  
después.

↓  
Nos hemos saltado al final.

- Se transmiten pocos bits y la secuencia de desplazamiento no ocurrirá.



(Aprovechan las transiciones 1 a 0 y 0 a 1 para registrarla).

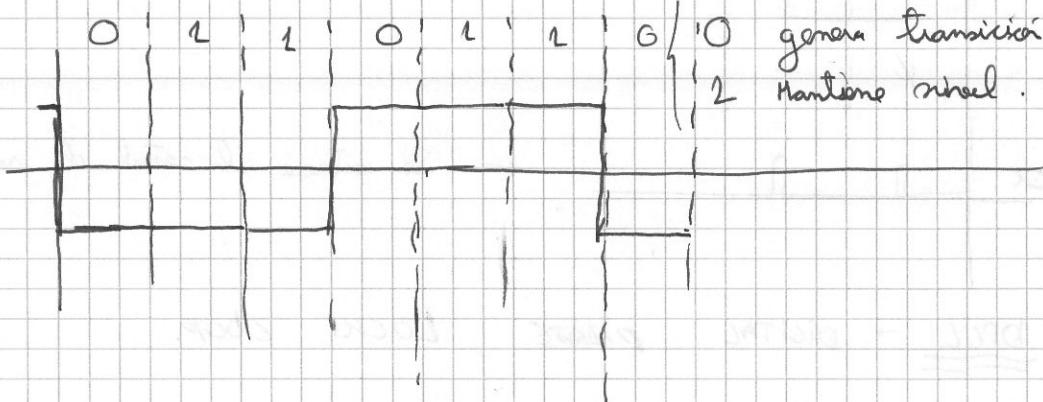
## TRANSMISIÓN SÍNCRONA

Si al DLL detecta una transición y detecta desplazamiento, es decir no son 16 pulsos registra el número de pulsos.

Lo que para que esto ocurra se necesitan transiciones de 0 a 1 y de 1 a 0.

→ Para ello se utiliza la codificación:

NRZI (Non-Return to Zero Invertido)



→ Es una secuencia de 1's no hay variación de señal.

Hay que garantizar que el mensaje tenga ceros de forma frecuente para generar la transición.

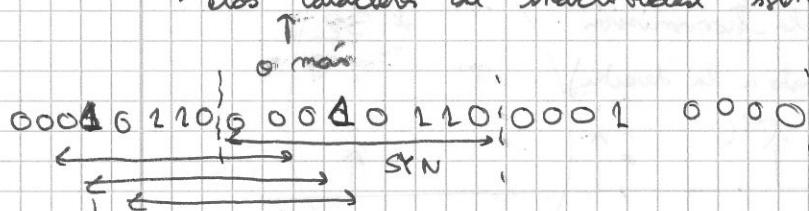
## TRANSMISIÓN SÍNCRONA

→ Detectar los límites de 1 byte.

Carácteres SYN (Carácter de control).



→ Dos caracteres de inactividad síncrona.



→ ASCII extendido 8 bits.

→ El receptor entra en modo de búsqueda.

Abre una ventana de 8 bits, cogiendo los 8 primeros bits.

Compara los 8 bits con el carácter de sincronía. (SÍNCRONÍA).

Comprueba si es SYN (descarta el último bit y coge al siguiente)

Primer SYN: sincronización a nivel de bit

Segundo SYN: sincronización a nivel de byte.

→ DCE ambiental en medio anulado uno.

→ No se aprovecha la tasa de transmisión, no es eficiente.

- Límites a concatenar ASCII.

## TRANSMISIÓN SÍNCRONA ORIENTADA A CARÁCTERES

- El número mínimo es un carácter (byte) ↳ SYN

## TRANSMISIÓN SÍNCRONA ORIENTADA A BITS

↳ Se utiliza en enlaces punto a punto.

Inicio y final de trama:

[01111110] [01111110] [01111110]

↓  
byte bandera  
patrón bandera

{ 01111110 }

↑  
RX piensa que  
la trama ha  
acabado  
Evitar que esta  
secuencia ocurra en  
la parte de datos.

[ 01111110 ]

- Si encuentra 5 1's seguidos inserta un cero.

011111010 --- Continuar

Caso de Enlace OKE encontrada, OKE insertar.

0111110101

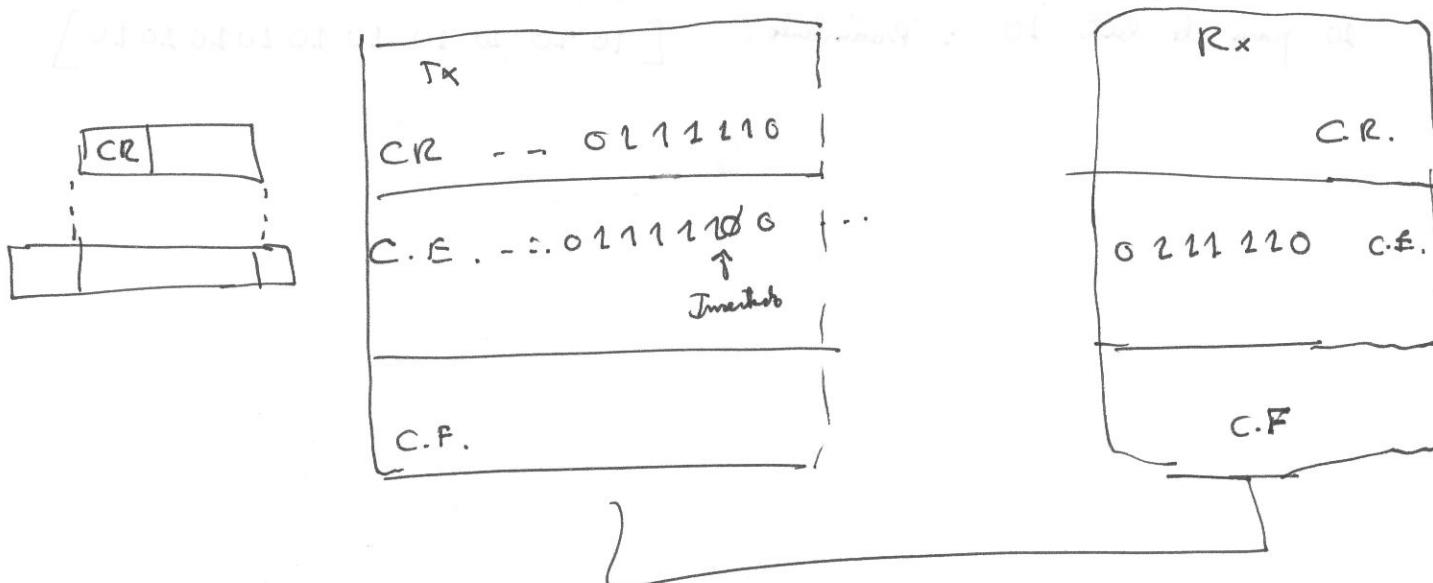
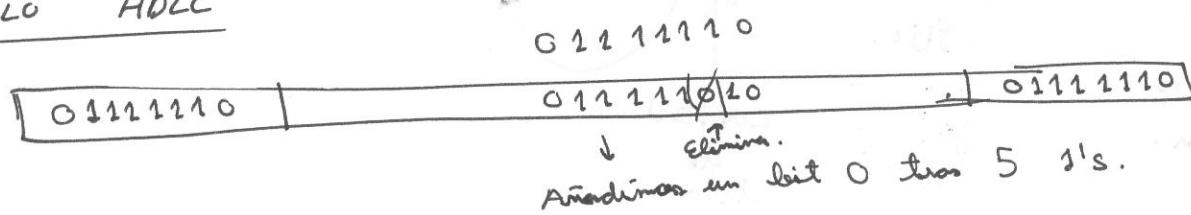
↳ Cero después de 5 1's cero que elimino.



## BANDERAS DE INICIO Y FIN Y CON INSERCIÓN DE BIT

- Mecanismos de transmisión orientados a bytes. Secuencia de bytes especiales.

### PROTOCOLO HDLC



NRZI → PPLL.

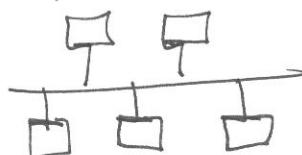
▼

- Independiente del código utilizado.
- 1 bit adicional, por cada 5 unos consecutivos. 1 bit para lograr la transparencia de los datos.
- Tramas de un número cualquiera de bytes. Trama de longitud cualquiera. (No son de un número múltiplo de 8).
- Resincronización mediante marcadores de inicio y final.

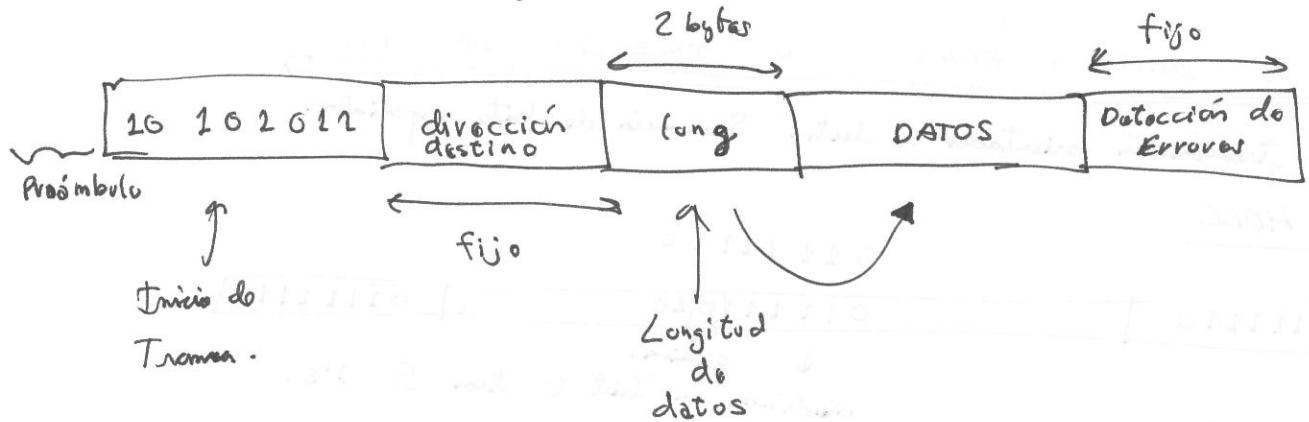
## BANDERAS DE INICIO

### DELIMITADOR DE INICIO + CUENTA DE CARACTERES

- Se utiliza en redes LAN de difusión. En aquellas en que al medio de transmisión se comparte por todas las estaciones.

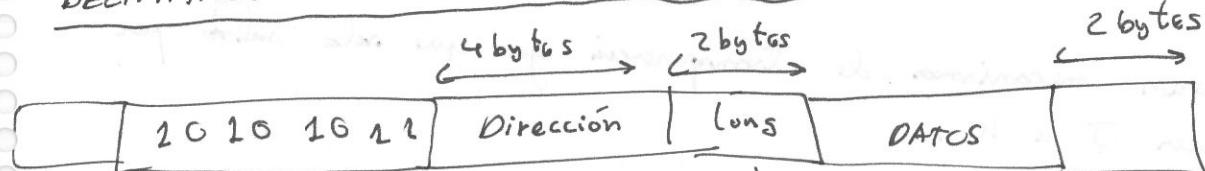


- Delimitador de inicio : formato de trama de Ethernet . 802.3



20 pares de bits 10 : Preámbulo . [ 10 10 10 10 10 10 10 10 10 10 ]

## DELIMITADOR DE INICIO + CUENTA DE CARACTERES



Preamble  
10 bits  
10 para que haya  
resincronización en  
caso de problemas.

Para lograr  
resincronización de lect.,  
no si hay errores no  
afecta.

↓  
802.3  
Ethernet.

↑ Retención de  
errores.

Los elementos son  
bytes, no puedo tener  
un número arbitrario  
de bits.

- No se necesitan mecanismos de transparencia de datos ya que disponemos  
de la longitud.

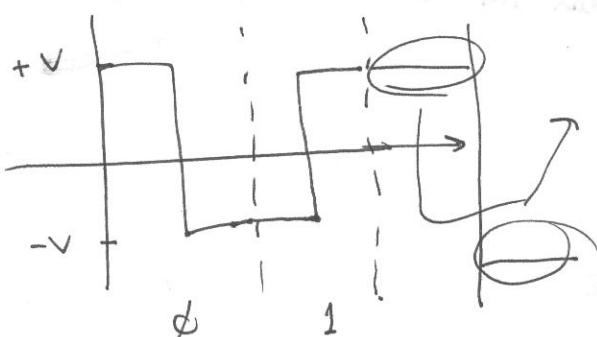
↳ EXAMEN

## VIOLACIÓN DE CÓDIGO EN CADA FÍSICA

→ Se utiliza cuando la codificación utilizada en capa física  
redundancias. Token Ring (802)

continúa

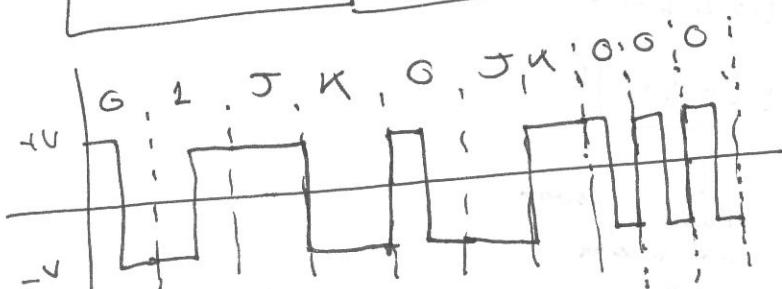
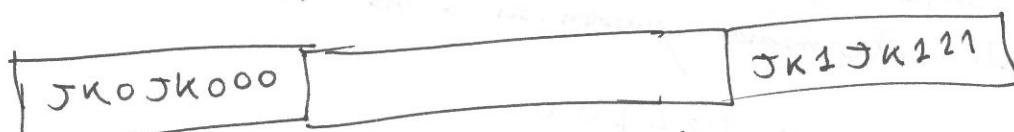
Manchester:



Codificaciones apropiadas para marcar inicio  
y finales de trama.

J → Mantener nivel durante todo el  
tiempo de celda de lect.

K → Nivel contrario durante todo el  
tiempo de celda de lect



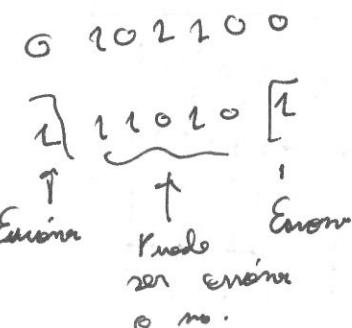
- Transmisión: añadir coras por en media para indicar que el RX creó que la trama ha terminado.
- No se requieren mecanismos de transmisión ya que solo sabes por error apretar J o K.
- Más eficiente que el modo 

DLE / STX	
-----------	--

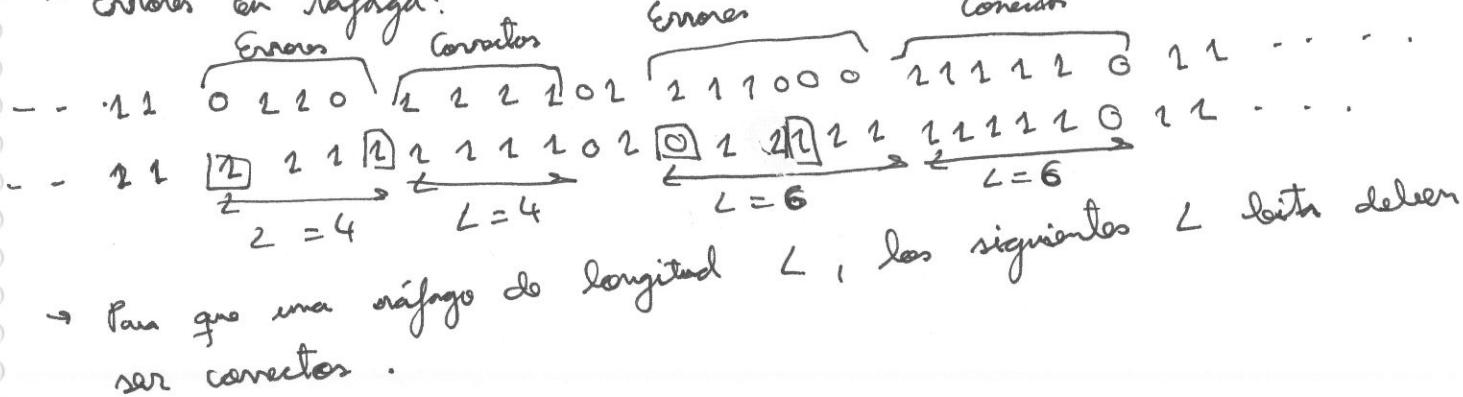
DLE / ETX
-----------

### CONTROL DE ERRORES

- Asegurar que los tramos llegan correctamente:
  - + Ruido Térmico: ocurre en todos los medios de transmisión debido a la agitación térmica.
  - + Diafonía: cruce de señales. Asoplamiento capacitivo entre 2 cables.
    - NEXT: diafonía de extremo cercano. Un transmisor donde la señal enviada es potente y la recibida es débil.
  - + Ruido de impulso: impulso eléctrico externo que es captado por mi medio de transmisión.
  - Trenzados
- Tipos de errores:
  - Errores aislados: que afecta a un bit
  - Errores en ráfaga: una serie de bits se ven afectados.
  - Ráfaga: conjunto que empieza y acaba con un bit erróneo y cuyos bits intermedios pueden ser o no erróneos.



- Errores en ráfaga:



- Tramas : 1000 bits

- BER :  $10^{-3}$

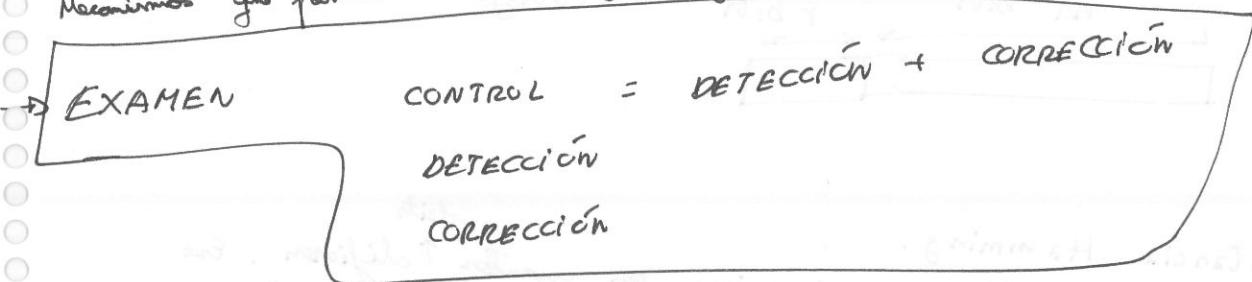
- CONTROL DE ERRORES = DETECCIÓN + CORRECCIÓN

+ Detección + corrección de errores por parte del receptor.

## CONTROL DE ERRORES

- Errores aislados.
- Errores en ráfaga.

Mecanismos que permiten detectar y corregir errores: CONTROL DE ERRORES.



- Control de errores hacia adelante; añadimos suficientes bits redundantes a la trama como para que la capa de enlace destino sepa que ha habido un error y sepa de forma autónoma corregir los errores.

↳ Detectar y corregir  $\Rightarrow$  Ambas cosas en Capa Enlace destino.  
(Ideal para transmisión de satélite).

- Control de errores por retroalimentación; se va añadir por cada trama (mensaje) un número de bits redundantes para que la capa de enlace, después que la trama es enviada a correcta y su forma de corrección consiste en pedir un reenvío de la trama. (más utilizada)

↳ Detección (C. Enlace destino).

↳ Corrección (C. Enlace destino solicita retransmisión).

Errores Aislados:  
Tramas: 1000 bits

$$\text{BER: } 2^{-6}$$

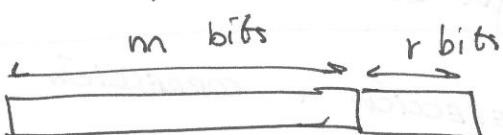
Fichero: 1 Mb

- Control de errores hacia adelante: MÉTODO DE HAMMING. (de más)
  - Corregir un bit: 10 bits por trama.
  - ↳ 1000 tramas  $\rightarrow$  Se corrían 10.000 bits redundantes (2 BER trama (BER))
  - ↳ 1000 tramas  $\rightarrow$  20 bits trama para código correcto. (corrección de 2 bits)

- Control de errores por retroalimentación:
  - 1000 tramas  $\rightarrow$  1 bit por trama  $\rightarrow$  1.000 bits redundantes (paridad)  $\rightarrow$  1 frame (BER)
  - Trama errónea  $\rightarrow$  + 1001 bits (trama retransmitida)
  - ↳ (de más)

→ Def: Palabra Código.

Dado un mensaje a transmitir de  $m$  bits hay se añade un código de detección - corrección de  $r$  bits redundantes. Este mensaje completo ( $m+r$ ) bits se le denomina palabras.

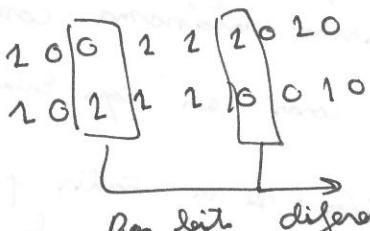


Palabra

$$\text{Código} = m+r$$

→ Def: Distancia Hamming.

Tenemos 2 palabras código que midan en cuantos bits difieren. Ese número de bits en que difieren se conocen como distancia Hamming.



$$\text{Distancia Hamming} = 2$$

→ Def: Distancia Hamming del Código.

Código: totalidad de palabras código. Conjunto de símbolos diferentes que puede utilizar.

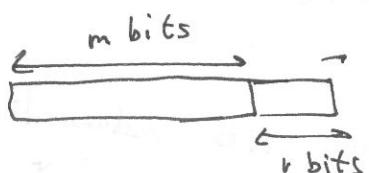
La distancia hamming del código: dada una lista de palabras código la distancia hamming del código es la distancia hamming entre cualesquier palabras código de mi código.

Código

$= 2 \left\{ \begin{array}{l} 00 \\ 01 \end{array} \right\} \rightarrow = 1$  (Distancia Hamming de (Código)

$= 2 \left\{ \begin{array}{l} 00 \\ 01 \\ 10 \end{array} \right\} = 2$  La menor de las distancias hamming teniendo en cuenta todos los símbolos.

$1 = \left\{ \begin{array}{l} 00 \\ 01 \\ 10 \\ 11 \end{array} \right\} \rightarrow$  La menor es 2.



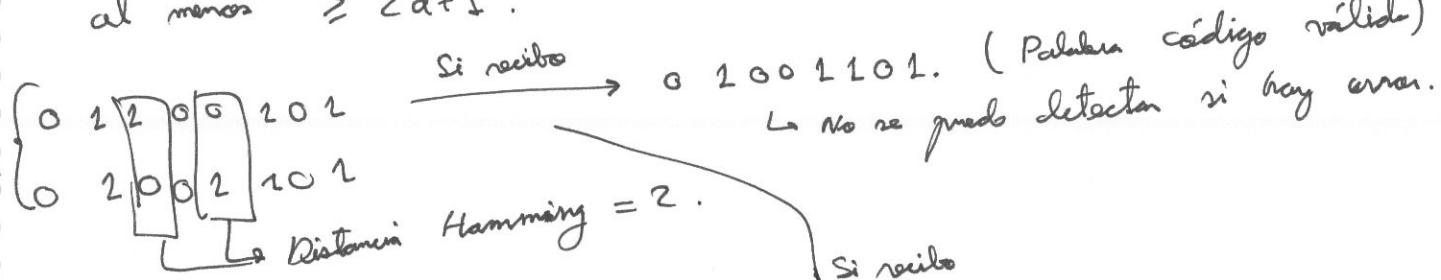
$2^m$  combinaciones.

$2^r$ : No todas son válidas las combinaciones ( $2^r$ )  
 ↳ Válidas (símbolos que se pueden dar)  
 ↳ Inválidas (símbolos que ~~no~~ se pueden dar)

→ Def: Distancia Hamming del código.

→ Para poder detectar el errores se necesita un mecanismo cuya distancia Hamming del código sea al menos  $\geq d+1$ .

→ Para corregir  $d$  errores, la distancia Hamming del código debe ser al menos  $\geq 2d+1$ .



2 opciones

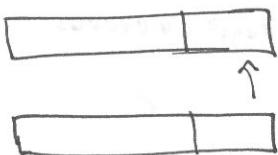
→ Si recibo  $01001101$ . (Palabra código válido)  
↳ No se puede detectar si hay error.

→ Si recibo  
 $01000101$  (Palabra código inválida)  
↳ Se detecta el error.

→ Copia de Enlace debe intentar  $\approx 100\%$  de no errores.

Control de errores por sobrecarga: añade menos bits.

Palabra Código:

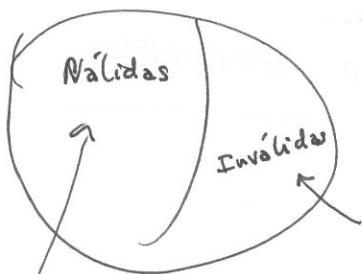


Distancia Hamming: bits que difieren dos palabras código.

Distancia Hamming del Código: distancia mínima del código.  
(menor de las distancias Hamming).

Bits redundantes:

→ Detectar d errores  $\Rightarrow$  Distancia Hamming del código  $\geq d+1$



Error

→ Detectar y corregir errores  $\Rightarrow$  Distancia Hamming del código  $\geq 2d+1$

La tendré que dar por buena.

Capacidad de detección = 0

ALFABETO: Distancia  
 $r=0$

000

001

010

011

100

101

110

111

Hamming del Código = 1.

Envío  $\xrightarrow{\text{Tx}}$  010 → 010 ← Palabra código válida.

$\times 02 \rightarrow 002 \leftarrow$  Palabra código válida

(Error no detectado)

(esta dentro del alfabeto).

$v=1$

→ Distancia Hamming de (v: N° de bits redundantes).

0000

0011

0101

0110

1001

1010

1100

1111

Tx

0110

0110

Rx

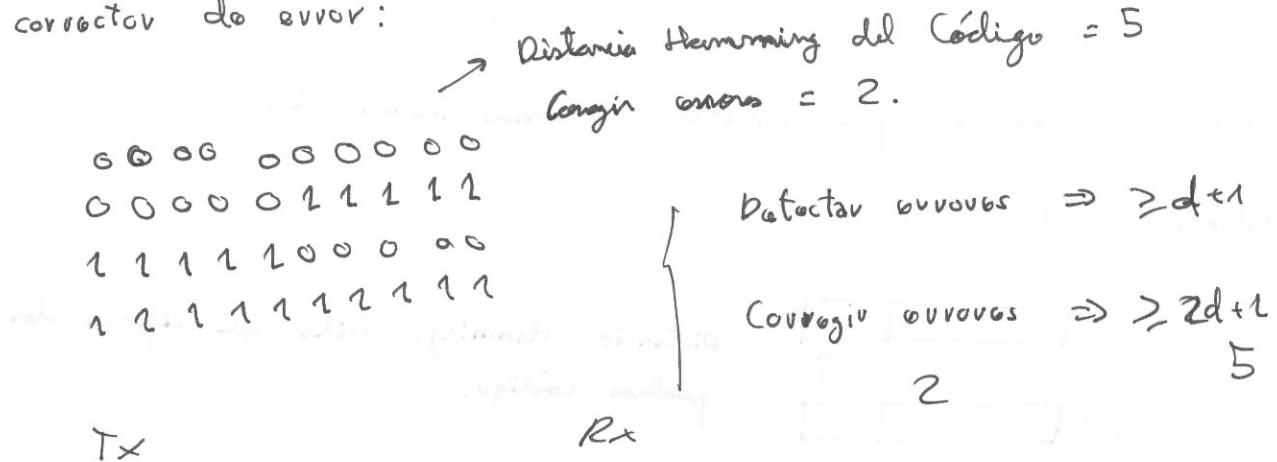
0100

0101

→ Capacidad de Detección = 1  
(Error detectado)  
Palabra código inválida  
(no es un símbolo del alfabeto)

↓  
Error no detectado.  
Palabra código válida  
(es un símbolo del alfabeto)

Código corrector de error:



$$\begin{array}{l} 000000000000 \rightarrow 00000000011 \\ 000000000000 \rightarrow 00000000011 \end{array}$$

↑  
No tiene capacidad para detectar 3 errores.

### MECANISMOS O CÓDIGOS DE DETECCIÓN DE ERRORES.

- Paridad
- Verificación de suma de bloques (BCC).
- Código de redundancia clínica (CRC).
- ↳ Añadir  $r$  bits redundantes de forma que a priori sepa que resultado debe dar cierta operación matemática.
- Paridad: Método típico con transmisiones asíncronas y de sincronas orientadas a caracteres.  
ASÍNCRONO



Bit de paridad { par → N° de 1's par.  
impar → N° de 1's impar.

PARIDAD → 0110011

PAR → 0110011 0

IMPAR → 0110011 1

{ N° de 1's incluye la paridad os par.

{ N° de 1's incluye bit de paridad os impares.

Tx Rx

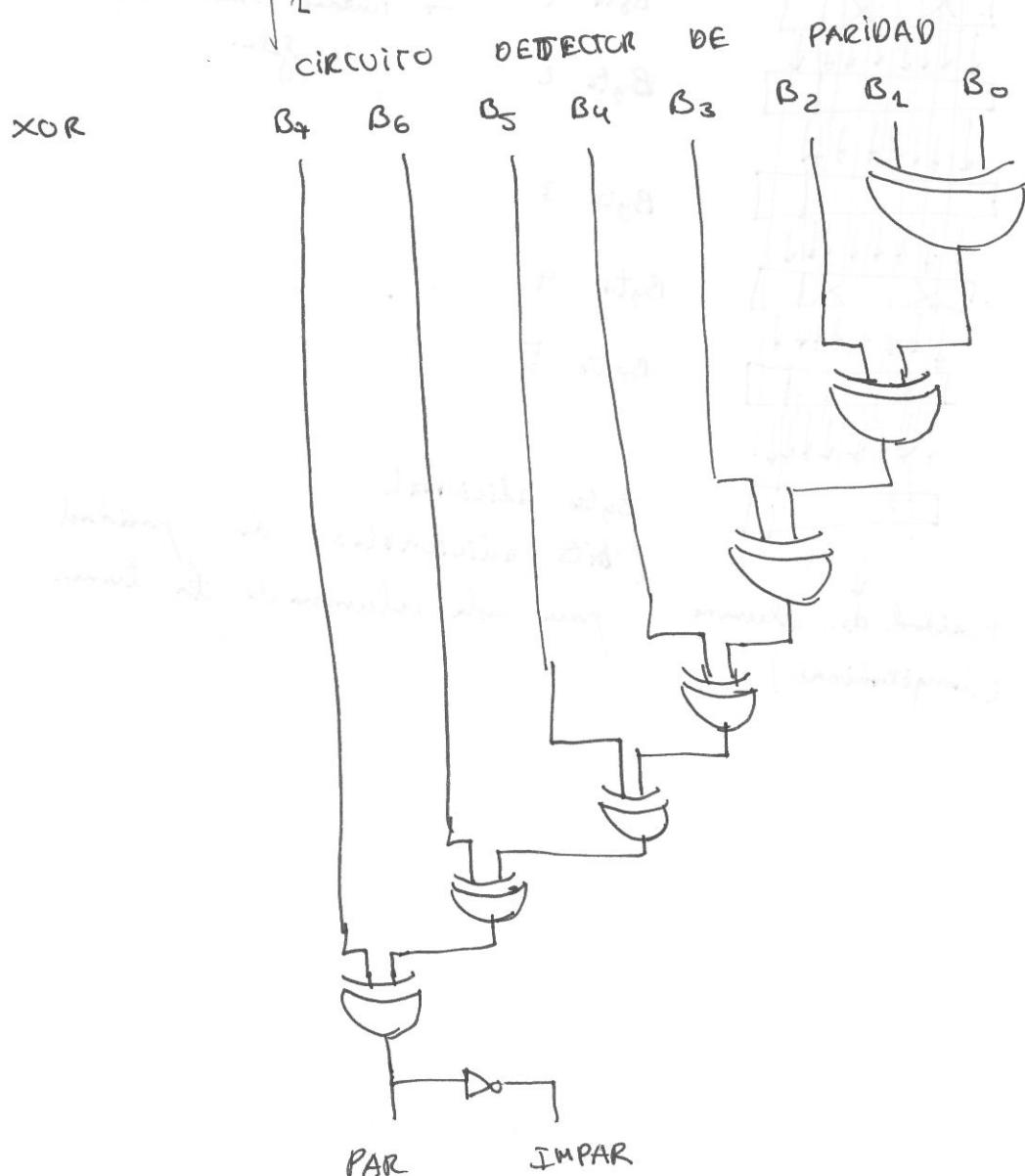
0110011 0 → 0110011

0110011 0 → Error N° de 1's impar.

- Paridad: distancia hamming = 2.

→ También detecta N° impar de errores.

0 0 0	0
0 0 1	1
0 1 0	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1



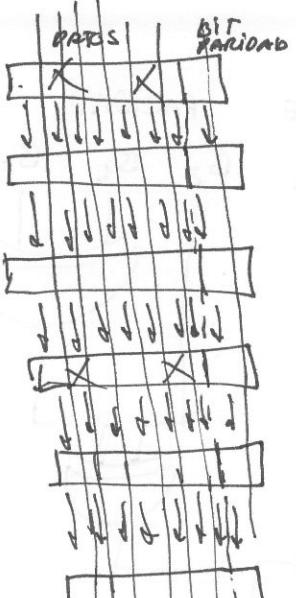
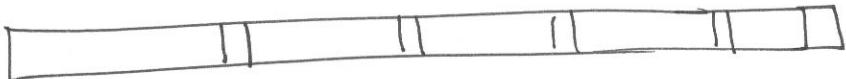
VERIFICACIÓN DE SUMA DE BLOQUES (BCC: BLOCK CHECK CHARACTER)

↳ Al transmitir tramas puede alcanzar una detección mayor de errores obtenida con el bit de paridad. Añadiendo un conjunto adicional de bits de paridad que tiene en cuenta toda la trama.

1 bit de paridad para cada byte

+  
1 bit de paridad por cada posición de bits de mi  
Estructura.

(Paridad de todos los bytes de una misma posición).



en otro byte  
el mismo error en  
los misma posiciones,  
error no detectable,  
pero menor probabilidad.

Byte 1 → Paridad transversal o  
de fila.

Byte 2

Byte 3

Byte 4

Byte 5

Byte adicional

(bits adicionales de paridad  
para cada columna de la trans-

Paridad de columna  
(longitudinal).

## VERIFICACIÓN DE SUMA DE BLOQUE

- Teniendo en cuenta la totalidad de la trama. Paridad de fila y columna.
- Aritmética del complemento a 1.
- Byts Núm. sin signo.

Algoritmo Tx:

- 1º Suma en complemento a 1 ( $C_1$ ) de todos los bytes.
- 2º Resultado de la suma calculamos el complementario e invertido.  $\Rightarrow BCC$ .

Algoritmo Rx:

- 1º Suman todos los bytes, BCC incluido.
- 2º Resultado conocido = 0.

EJEMPLO:

- Se envía la siguiente trama:

Stx  
0000 0010

20110111

1101100

Etx  
0000011

Caracter de 7 bits:  
se suman 8 bits.

0000010  
2011011  
1101100  
0000011

000 00 20  
1 0 1 2 0 1 0  
2 2 0 1 2 1 0  
1 2 0 1 2 0 0  
1 2 0 1 2 0 1

111 101  
000 0010  
2011011  
1101100  
0000011

Carry  
no realmente  
realmente  
1  
+  
1001100

2002201

→ Suma.  
→ Complementario (Invertido)

Añadimos a la trama como byte de verificación de suma de bloque.

→ Ca1.

$$\begin{array}{r}
 \text{Rx} \rightarrow \text{Suma Todo} \\
 \begin{array}{r}
 \cancel{0} \quad \cancel{0} \quad \cancel{0} \quad \cancel{0} \quad \cancel{0} \quad 2 \quad 0 \\
 + 1 \quad 0 \quad 2 \quad 1 \quad 0 \quad 2 \quad 2 \\
 \hline
 1 \quad 1 \quad 0 \quad 2 \quad 2 \quad 0 \quad 0 \\
 + 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \\
 \hline
 0 \quad 1 \quad 2 \quad 0 \quad 0 \quad 2 \quad 0
 \end{array}
 \end{array}$$

Cal { + Ø  
- Ø

$$② \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0$$

$$\begin{array}{r}
 \cancel{1} \quad \cancel{1} \quad \cancel{1} \quad \cancel{1} \quad \cancel{1} \quad \cancel{1} \\
 + 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\
 \hline
 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 1
 \end{array}$$

Cal → Todos 2 1 1 1 1 1 1 (Almox)  
as cero y Toda cero es uno.

### códigos polinómicos

↳ También llamados: códigos de REDUNDANCIA FRAME CHECK SEQUENCE (FCS).

↳ Verificación de secuencia de trama.

CÍCLICA (CRC).

- A la trama completa se le añaden los bits del CRC
- Bits de la trama van a representar coeficientes de un polinomio. Cuyos valores son 0 y 1.

$$a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_1 \cdot x^1 + a_0 \cdot x^0, a_i \in \{0, 1\}$$

Trama  $K$  bits  $\Rightarrow$  polinomio de grado  $K-1$ .

$$\begin{array}{r}
 4 \quad 2 \quad 1 \quad 0 \\
 1 \quad 0 \quad 0 \quad 1 \quad 1
 \end{array}$$

MSB  $\rightarrow a_n$ .

LSB  $\rightarrow a_0$

$$\begin{array}{r}
 \downarrow \\
 x^4 + x + 1
 \end{array}$$

- Aritmética de polinomios (MÓDULO 2). No hay términos de errores.  
Equivale a XOR.

- El divisor cabrá en un dividendo siempre y cuando tengamos el mismo grado.

$G(x) \rightarrow$  Polinomio Generador

$$\hookrightarrow a_n = 1$$

$$a_0 = 1$$

$\rightarrow$  La suma, los lleva al final representación un polinomio ~~multiple~~  $\rightarrow$  multiple de  $G(x)$ .

1 100  $\cdots$  1  $\overline{xxxx}$   
 CRC  
 $\hookrightarrow$  Polinomio divisible por  $G(x)$   
 $\uparrow$   
 Generador.

$\rightarrow$  Si Resto = 0.  $\Rightarrow$  OK.

Si Resto  $\neq$  0.  $\Rightarrow$  K.O.  $\rightarrow$  Error Detectado.

## VERIFICACIÓN DE REDUNDANCIA CÍCLICA

$$a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0 \cdot x^0, \text{ where } \{a_i\}_{i=0}^n$$

↑  
LSB

$G(x)$  : Polinomio genérico.

→ dado un polinomio, anadir letr redundantes de forma que el polinomio sea divisible por  $G(x)$ .



$$\left. \begin{array}{l} \text{Ratio} \neq \phi \Rightarrow \text{Error} \\ \text{Ratio} = \phi \Rightarrow \text{No key error.} \end{array} \right\}$$

m bits  $\Rightarrow$   $M(x)$   
 r bits  $\Rightarrow$  redundantes, polinomio generador de grado  $r$ .  
 BITS DE DATOS  $\rightarrow$  BITS REDUNDANTES  
 $m > r$

TX

TX 1- Añadir r bits 0 para la parte menos significativa. (LSB).

↳  $x^r \cdot M(x)$

$$2: \frac{x^r \cdot M(x)}{G(x)} \rightarrow \begin{cases} Q(x) \\ R(x) \end{cases} \rightarrow \text{Rest zu}$$

3.- Restarle el resto al dividendo.

$$x^r \cdot M(x) - R(x) \quad \left\{ \begin{array}{l} \text{Operación} \\ (\text{Polinómico}) \end{array} \right. \begin{array}{l} \text{múltiple de} \\ G(x) \end{array}$$

4 - La Thoma a unión es:

$$t(x) = x^r \cdot m(x) - R(x)$$

R(x)

1: Va a verleir  $T'(x)$ , que divide  $T(x)$  o  $T(x)$  con error.

$$\frac{T'(x)}{G(x)} \left\{ \begin{array}{l} \text{Si Resto} \neq 0 \Rightarrow \text{Hay errores.} \\ \text{Si Resto} = 0 \Rightarrow \text{No hay errores. (}\geq 100\% \text{)} \end{array} \right.$$

- El error, si lo hubiera es:  $E(x)$ .  $\rightarrow$  Polinomio de  $x$ .

+ Términos con ' $x^3$ ', error en esa posición.

Suma módulo 2. (XOR)

$$T'(x) = T(x) + E(x)$$

*Malo un error*

$$T(x) \equiv 100110$$

$$E(x) \equiv x^2 \rightarrow 100$$

$$\begin{array}{r} 100110 \\ + 100 \\ \hline 1000110 \end{array}$$

$$\begin{array}{r} 100110 \\ 0 \\ \hline 100110 \end{array}$$

2.<sup>o</sup>

$$\frac{T(x)}{G(x)} = \frac{T(x) + E(x)}{G(x)} = \frac{\cancel{T(x)}}{\cancel{G(x)}} + \frac{E(x)}{G(x)}$$

*Resto = 0*

$$\frac{E(x)}{G(x)} \rightarrow \text{Malo el error.}$$

$E(x)$  tiene como factor  $G(x)$   
no podemos detectar el error.

$$E(x) \bmod G(x) \left\{ \begin{array}{l} \text{Resto} \neq 0 \Rightarrow \text{Error (100%)} \\ \text{Resto} = \emptyset \left\{ \begin{array}{l} E(x) = 0 \Rightarrow \text{No hay error} \\ E(x) = K(x) \cdot G(x) \Rightarrow \text{Error No Detectado.} \end{array} \right. \end{array} \right.$$

→ Polinomio generador tenga muy pocos múltiplos para disminuir los errores no detectados.

$$E(x) = x^i \rightarrow \text{Error de lect}$$

$$G(x) \rightarrow 2 \text{ términos} \rightarrow G(x) = x^2 + x \rightarrow \text{Se capturan todos los errores de lect.}$$

VERIFICACIÓN DE REDUNDANCIA CÍCLICA (II)

→ Capturar los errores en un número impar de bits.

$E(x)$  tendrá un número impar de términos.

↗  $(x+1) \rightarrow$  Puede facturar de un número impar de términos? No.

$$E(x) = Q(x) \cdot (x+1)$$

$$E(1) = (1+1) \cdot Q(x) = 0 \cdot Q(x) = 0$$

$G(x)$  permite:

→ Capturar todos los errores de 1 bit.

→ Capturar todos los errores dobles.

→ Capturar todos los errores en un número impar de términos.

→ Capturar todos los ráfagos menores ( $\leq$ ) o iguales a  $r$ .

→ Capturar casi todos los ráfagos de  $(r+1)$  bits. La probabilidad de que una trama incorrecta se dé por casualidad  $\Rightarrow \left(\frac{1}{2}\right)^{r-1}$

$$P = \left(\frac{1}{2}\right)^{r-1} \rightarrow r: \text{bits}$$

→ Para ráfagos mayores ( $>$ ) de  $(r+1)$  bits. La probabilidad de que una trama incorrecta se dé por casualidad  $\Rightarrow \left(\frac{1}{2}\right)^r$

Ejemplos:

$$\text{CRC-16} = x^{16} + x^{15} + x^7 + 1$$

→ Añadimos  $r$  bits.

$$r = 16 \rightarrow 16 \text{ bits.}$$

$$\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$$

$$\text{CRC-32} = x^{32} + x^{26} + x^{23} + x^{26} + x^{27} + x^{21} + x^{20} + x^8 + x^7 + x^5 + x^4 + x^3 + x + 1$$

EXAMEN

CÓDIGOS POLINÓMICOS (POLINOMIALES) (CRC)

RX

$$1^{\circ} \quad x^v \cdot M(x)$$

$$2^{\circ} \quad \frac{x^v \cdot M(x)}{G(x)} \rightarrow \begin{array}{l} Q(x) \\ R(x) \end{array}$$

$$3^{\circ} \quad T(x) = x^v \cdot M(x) - G(x)$$

RX

$$2^{\circ} \quad T'(x) = T(x) + E(x)$$

$$\frac{T'(x)}{G(x)} = \frac{T(x)}{G(x)} + \frac{E(x)}{G(x)}$$

III

Resto =  $\emptyset$ 
 $E(x) \neq 0 \Rightarrow$  Error (100%)

 $E(x) \bmod G(x)$   
 (residuo)

 $E(x) = \emptyset \Rightarrow$  No hay error

$$E(x) = K(x) \cdot G(x)$$

 Si  
 $E(x) \neq 0$   
 → Error no detectado

→ Queremos enviar la siguiente trama:

$$\text{TRAMA} = 1102011011011$$

$$\text{Polinomio Generador} \rightarrow G(x) = x^4 + x + 1$$

$$\begin{array}{r} \text{Calcular CRC} \\ \text{9845343210} \\ \text{1102011011011} \end{array}$$

$$T(x) = 1102011011011$$

$$T(x) = x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$$

$$x^4 \cdot (x^5 + x^8 + x^6 + x^4 + x^3 + x + 1)$$

$$x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4$$

$$\boxed{x^4 + x + 1}$$

$$\begin{array}{r}
 x^{13} + x^{12} + x^{10} + x^8 + x^7 - x^5 + x^4 \\
 - (x^{12} + x^9 + x^8 + x^7 + x^5) \\
 \hline
 -x^{18} + x^7 + x^5
 \end{array}$$

MAL!

$$x^4 - x^2$$

$$x^8 + x^6$$

$$\begin{array}{r}
 x^4 (x^9 + x^8 + x^6 + x^4 + x^3 + x^2) \\
 - (x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4) \\
 \hline
 + x^{12} - x^5 + x^8 \\
 - (x^{12} + x^8 + x^6) \\
 \hline
 -x^{18} \\
 \left( \begin{array}{r} + x^7 + x^5 + x^4 + x^3 \\ + x^7 + x^5 + x^4 + x^3 \end{array} \right) \\
 - (x^5 + x^2 + x) \\
 \hline
 (-x^{18} - x^3 - x^2 - x)
 \end{array}$$

$$(x^4 + x - 2)$$

$$x^4 + x - 2$$

$$x^8 + x^6 + x^3 + x$$

X.C.

$$\begin{array}{l} 1+1=0 \\ 1+0=1 \\ 0+0=0 \\ 0+1=1 \end{array}$$

Trama  $\Rightarrow M(x) = 1101011011$

$G(x) = x^4 + x + 1 \Rightarrow$  Grado 4  $\Rightarrow r = 4$  bits

$$\begin{array}{r} 13 \cdot 11 \cdot 10 = 89654 = 210 \\ 11010110110000 \end{array}$$

$G(x) = 10011$  *Anadimos ceros (no valen)*

$$\begin{array}{r} 11010110110000 \\ - 10011 \\ \hline 010021 \\ - 10011 \\ \hline 00000010110 \\ - 10011 \\ \hline 001020 \\ - 10011 \\ \hline 001100 \end{array}$$

Suma  
 Restas  
 Módulo 2  
 es (o  
 mismo  
 11  
 XOR 2).

$$\begin{array}{r} 10011 \rightarrow \text{Divisor } D(x) \\ \hline 1100001010 \\ \downarrow \\ G(x) \end{array}$$

$$R(x) = x^3 + x$$

$R(x) < D(x) \rightarrow$  Resultado posible.

3.º  $x^v \cdot M(x) - R(x)$

Solo modifico los bits redundantes.

$$\begin{array}{r} 11010110110110 \\ - \\ \hline 11010110110110 \end{array} \quad \begin{array}{l} \downarrow \\ \text{Resto lógico} \\ \hline \underbrace{1120}_{\text{CRC}} \end{array} \quad \equiv T(x)$$

→ EXAMEN!!!

Probabilidad correcta  $= \left(\frac{1}{2}\right)^v$  Trama incorrecta paso como

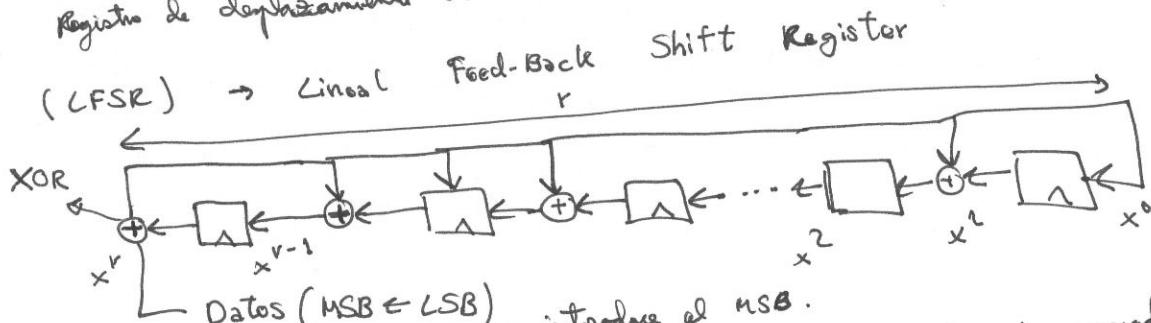
Códigos CRC

- Detecta errores de  $r$  bits.
- Algoritmo de división, computacionalmente complejo.

Peterson de Brown (1961)

- Interesa conocer el resto.
- Mecanismo rápido del cálculo del resto.
- Mecanismo rápido del cálculo del resto.

Registro de desplazamiento de realimentación lineal.



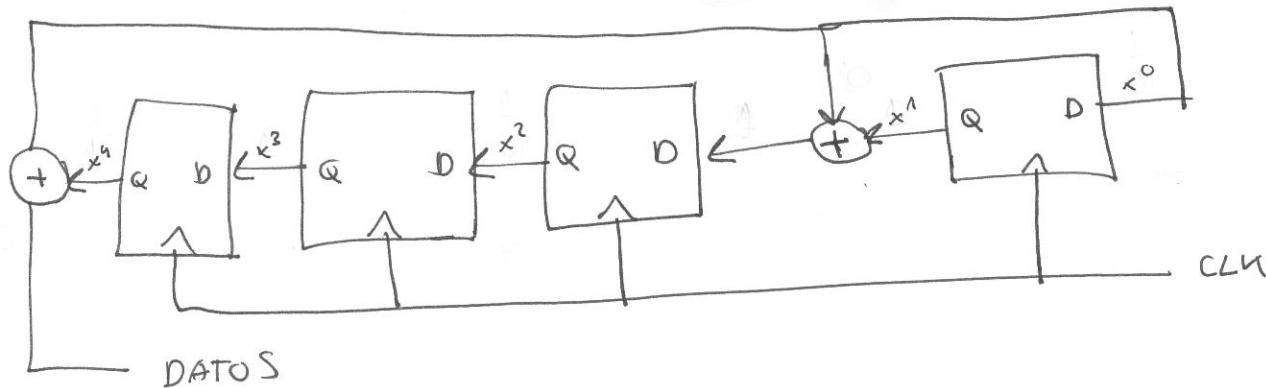
Datos (MSB  $\leftarrow$  LSB)  
Lo primero se introduce al MSB.  
Tanto como el polinomio generador (grado).

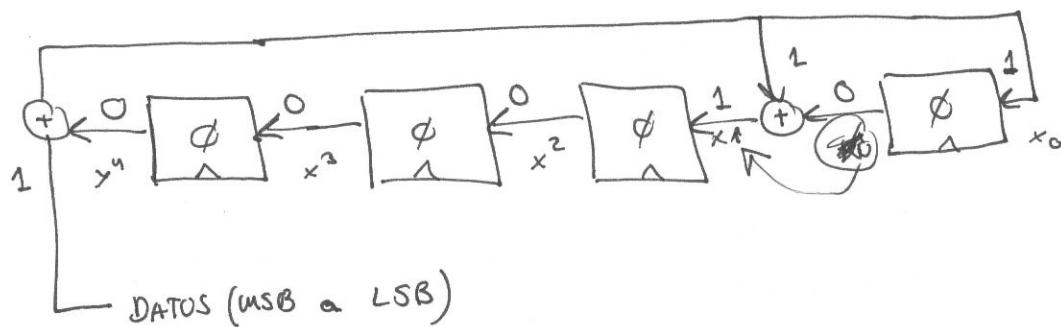
- FF's tantas como bits de CRC. (Tanto como el polinomio generador (grado)).

$\oplus = \text{XOR}$   
 $G(x)$   
Si el coeficiente 1; XOR y realimentación.  
Si el coeficiente 0; desplazamiento.

$$\begin{cases} a_n = 1 \\ a_0 = 1 \end{cases}$$

$$\begin{aligned} M(x) &\rightarrow \text{Mensaje} \\ M(x) &= \begin{array}{ccccccccc} \downarrow & \downarrow \\ 1 & 1 & 0 & 2 & 0 & 2 & 1 & 0 & 1 & 1 \end{array} \quad 9 \ 32 \ 20 \\ G(x) &= x^4 + x + 1 = 1 \ 0 \ 0 \ 2 \ 1 \end{aligned}$$





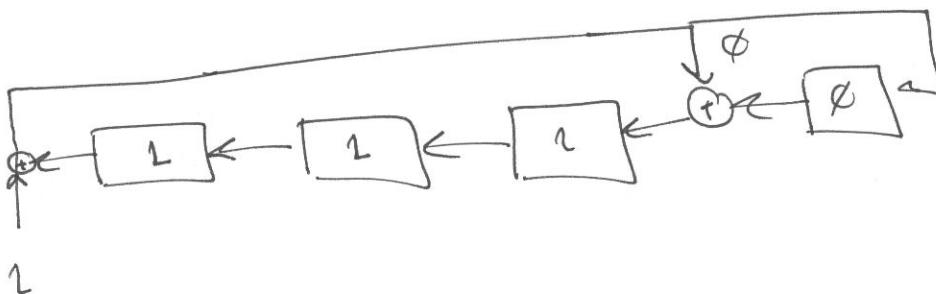
$$G(x) = x^4 + x + 1$$

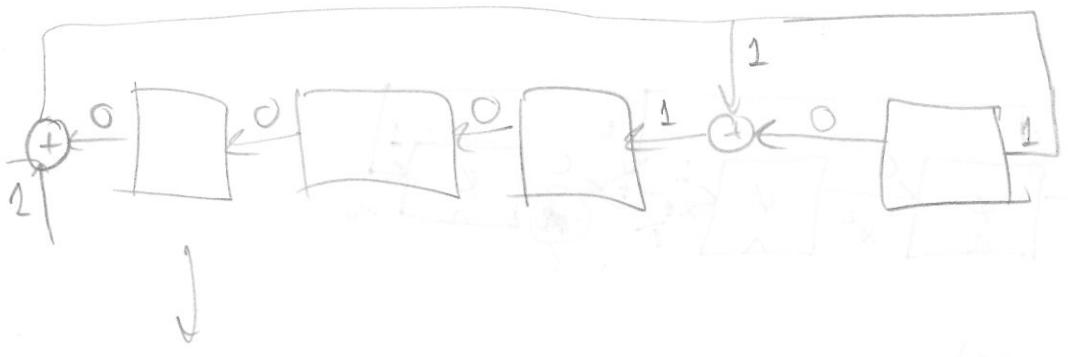
$\begin{smallmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{smallmatrix}$



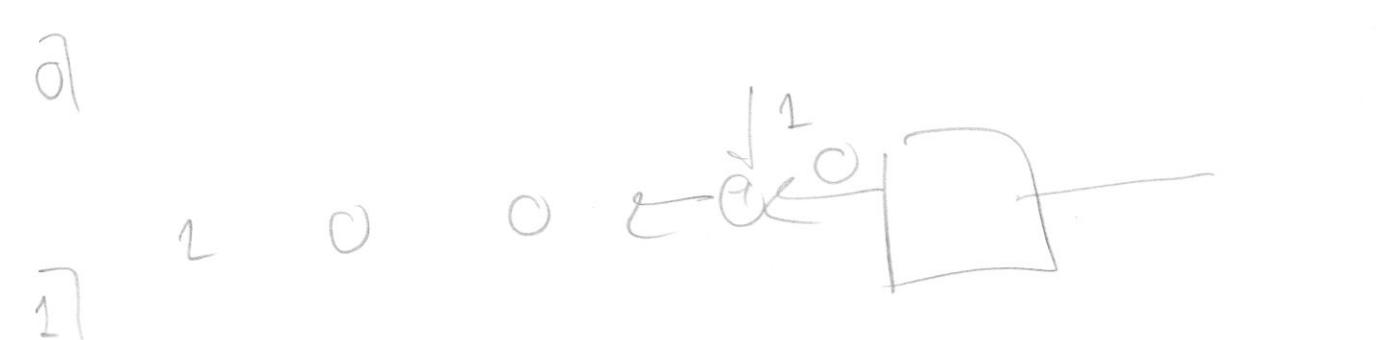
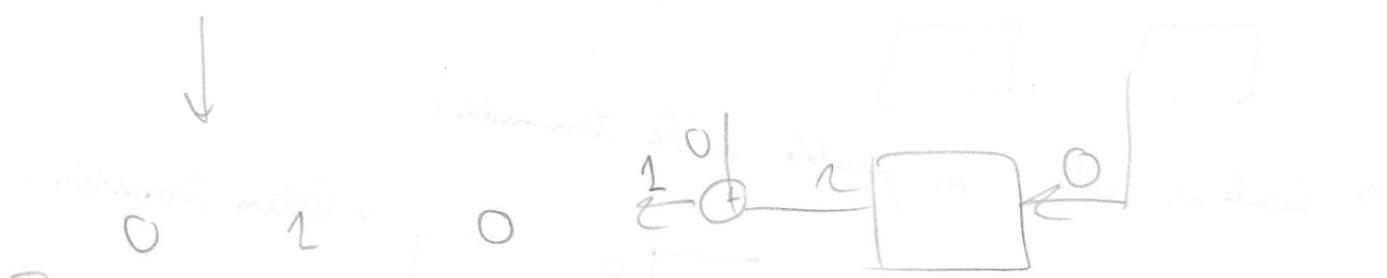
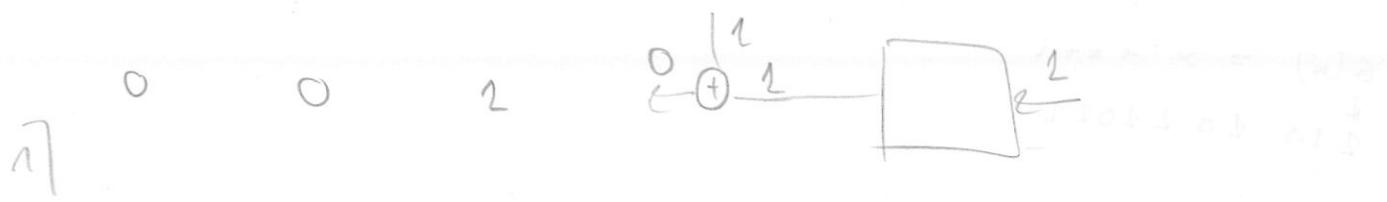
→ Calcula el resto en paralelo a la transmisión:

→ Última transmisión:





( $A_1 \wedge A_2 \wedge A_3 \wedge A_4$ )  $\rightarrow A_5$



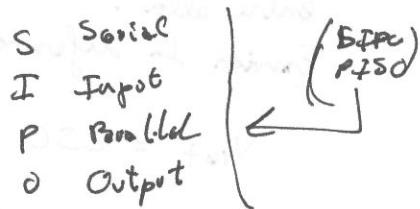
CLK	DATOS	$x^4$	$x^3$	$x^2$	$x^1$	$x^0$
0	-	0	0	0	0	0
1	1	0	0	1	1	1
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	1	0	1	0	0	0
5	0	1	0	0	0	0
6	1	1	0	0	0	1
7	1	0	0	1	1	0
8	0	0	1	1	0	1
9	2	1	1	1	1	0
10	1	2	1	1	1	0
11						
12						
13						
14						

EMISOR

$$\hookrightarrow g(x) = x^4 + x^3 + 1$$

CONTROL 1 → Envíar datos.

CONTROL 0 → Envíar CRC.

RECEPTOR

→ Recibe datos y va calculando el CRC.

COMPRESIÓN DE DATOS

→ Enviamos los datos originales.

- Recorrimos:

- + Codificación decimal empaqetada.
- + Codificación relativa.
- + Codificación Huffman.

+ Codificación decimal empaqetada

→ Útil cuando las tramas contienen muchos datos numéricos. Podemos

codificar los números con BCD (Binary Code of Decimal). Solo quedarse con la parte de BCD.

BCD.

$0 \rightarrow 0000$

$1 \rightarrow 0001$   
↑  
3 bits de ASCII (Eliminados).

Combinación 03 como byte se puede interpretar como ETX.

STX	CTRL	Nº de BCD's	'0'	'1'	'2'	...	ETX
-----	------	-------------	-----	-----	-----	-----	-----

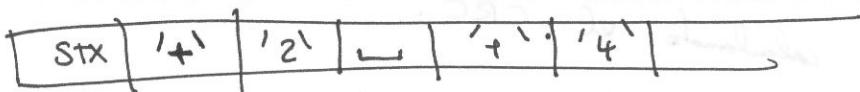
→ Muchos datos numéricos.

### + Codificación relativa:

- ↳ Alternativa BCD cuando los números presentan pequeñas diferencias entre ellos.
- Envian la diferencia respecto a un valor de referencia.

$$V_{ref} = 250 \rightarrow \text{Enviamos } [+2, -5, \dots]$$

\* Utilizamos transmisión orientada a caracteres.



\* Transmisión orientada a bits.



↑  
Bandera de inicio.

### + Codificación Huffman:

- ↳ Símbolos del alfabeto que no aparecen con la misma frecuencia.

Pocos bits para símbolos habituales. (5 bits)

Muchos bits para símbolos menos frecuentes. (10 bits)

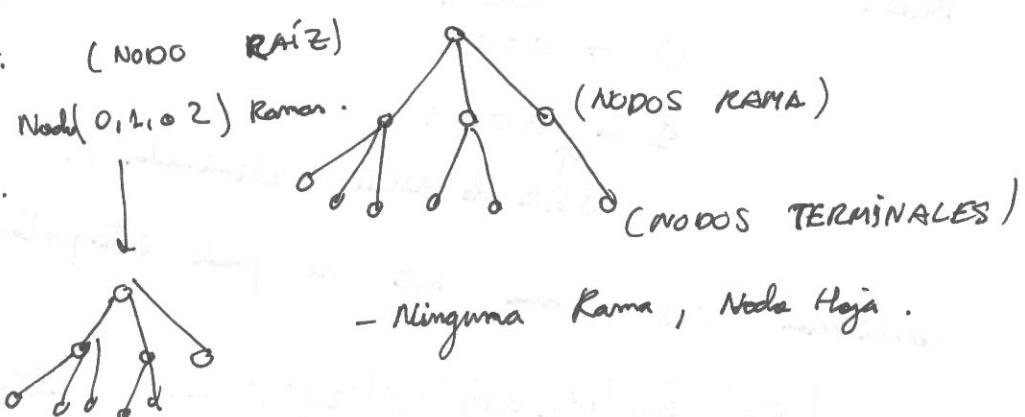
- Requiere saber la estadística de los símbolos.

- Solo disponible, orientada a bits.

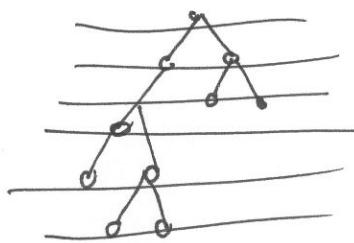
- Árbol Huffman: (NODO RAÍZ)

+ Binario: cada Nod(0,1,0,2) Ramas.

+ Desbalanceado.



5 Nodos.

**- DESBALANCEADO:**

- Grado del sub-árbol izquierdo diferente del sub-árbol derecho.

ALGORITMO

- ① Determinar los símbolos del alfabeto y su frecuencia relativa de aparición (probabilidad de aparición).
- ② Ordenar los símbolos en función de su frecuencia relativa de aparición.  
en orden ascendente o descendente
- ③ Eligimos los 2 símbolos cuya frecuencia relativa de aparición sean menores.
- ④ Crear nuevo símbolo:
  - Como símbolo la concatenación de los 2 elegidos.
  - Como frecuencia relativa de aparición la suma de ambas.
- ⑤ Asignar '0' al símbolo con frecuencia relativa de aparición mayor.  
(a la norma que va)
   
Asignar '1' al símbolo con frecuencia relativa de aparición menor.  
(a la norma que va)
   
(Siempre se aplica el mismo criterio mientras creamos el árbol, puede ser al revés).
- ⑥ Desectan los 2 símbolos usados y consideran el nuevo símbolo creado.
- ⑦ Volver al ③.
- ⑧ La codificación

- ③ La codificación de cada símbolo será la concatenación de los dígitos binarios al recorrer el árbol desde el modo raíz a los nodos hoja (símbolos del alfabeto).



CODIFICACIÓN HUFFMAN

X-C.

23-14-15

- 1.- Determinar símbolos del alfabeto y sus frecuencias relativas de aparición.
  - 2.- Ordenar símbolos (ascendente o descendente) según frecuencias relativas.
  - 3.- Seleccionar los 2 símbolos con frecuencia relativa de aparición menor.
  - 4.- Crear nuevo símbolo:
    - ¿Símbolo? concatenación de ambos.
    - ¿Frecuencia Relativa? suma de la de ambos.
  - 5.- Asignan 0 al símbolo con frecuencia relativa mayor } o al revés  
Asignan 1 al símbolo con frecuencia relativa menor }
  - 6.- Repetir los 2 símbolos seleccionados y considerar el nuevo generado.
  - 7.- Volvemos a 3.
  - 8.- Codificación: concatenar 0's y 1's desde la raíz a las hojas.
- Probabilidad
- |   |      |   |
|---|------|---|
| A | 0,3  | ✓ |
| B | 0,1  | ✓ |
| C | 0,05 | ✓ |
| D | 0,2  | ✓ |
| E | 0,25 | ✓ |
| F | 0,1  | ✓ |

- 
- 1)
  - 2) 1) A 0,3
  - 2) E 0,25
  - 3) D 0,2
  - 4) B/F 0,1

$$5) C 0,05 = P + P + P + 0,2 + 0,1 + 0,1$$

$$= 0,05 + 0,1 + 0,1 + 0,2 + 0,1 + 0,1 = 0,8$$

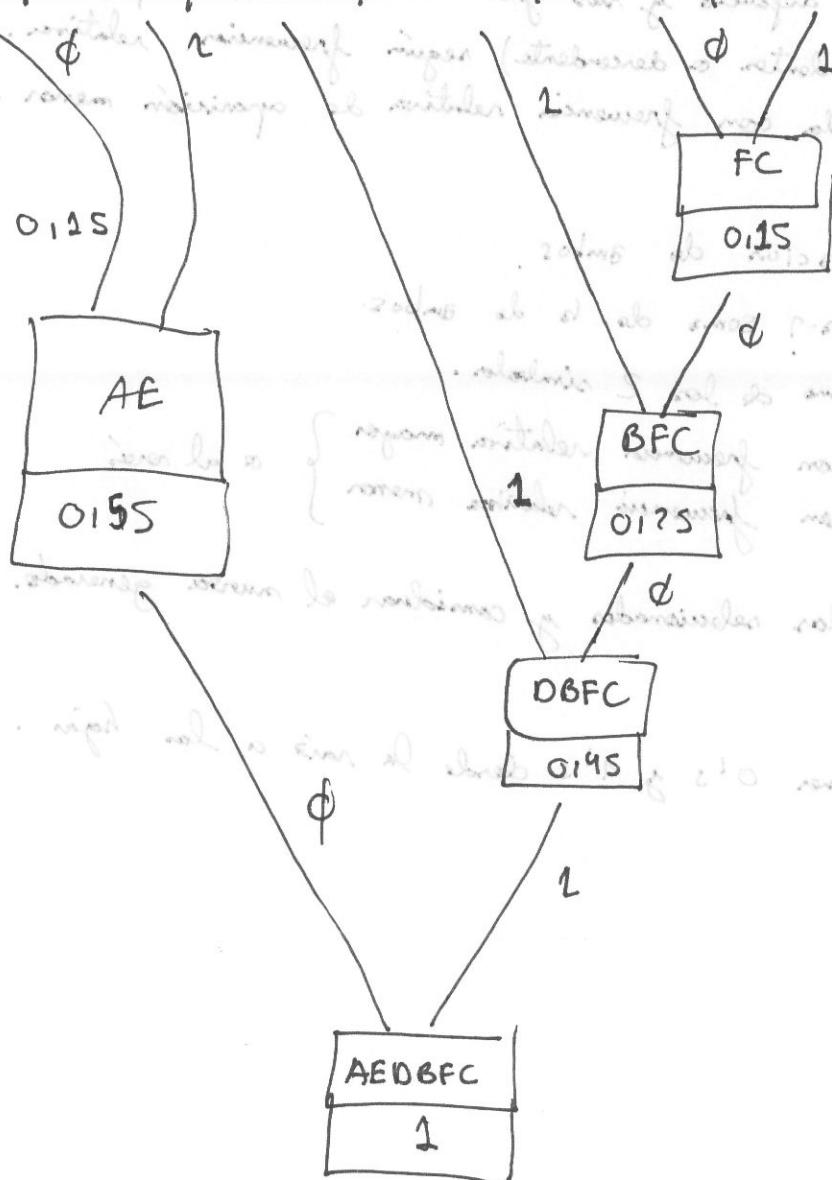
$$0,3 + 0,25 + 0,2 + 0,1 + 0,1 = 1 \text{ en codificación binaria}$$

$$0,05 + 0,1 = 0,15 \text{ en codificación binaria}$$

A	E	D	B	F	C
0,3	0,25	0,12	0,1	0,1	0,05

3) FC.

4)  $FC \rightarrow 0,15$



$A = 00$   
 $E = 01$   
 $D = 11$   
 $B = 101$   
 $F = 1000$   
 $C = 1001$

#### CODIFICACIÓN HUFFMAN

+ Más bits, menos probables.

+ Menos bits, más probables.

Nº bits a codificar:

Mensaje = AAAA BB EEEE DD FC

$$(2 \times 4) + (2 \times 3) + (4 \times 2) + (2 \times 2) + (4 \times 1) + (4 \times 1) =$$

$$= 8 + 6 + 8 + 4 + 4 + 4 =$$

$$= 34 \text{ bits} \rightarrow \text{Huffman}$$

ASCII: 14 símbolos  $\times 7$  bits = 98 bits  $\rightarrow$  ASCII

Nº bits codificación binaria = 14 símbolos  $\times 3$  = 42 bits  $\rightarrow$  binario

$$N_p = \sum_{i=1}^n N_i \cdot P_i \rightarrow \text{Probabilidad } i\text{-ésimo símbolo.}$$

↓

Nº bits  
símbolo

Número de  
los  $i$ -ésimo  
símbolo

$$N_p = 2.(0,3) + 3.(0,12) + 4.(0,05) + 2.(0,2) + 2.(0,25) + 4 \cdot 0,1 =$$

$$N_p = 2,4 \text{ bits / símbolo} \rightarrow \text{HUFFMAN}$$

3 bits  
símbolo → BINARIO .

definida.

→ Dada un alfabeto con frecuencia de aparición existe un número medio mínimo de bits técnicos por palabra - código : ENTROPIA ( $H$ )

$$H = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

$$H = - (0,3 \cdot \log_2(0,3) + 0,12 \cdot \log_2(0,12) + 0,05 \cdot \log_2(0,05) +$$

$$+ 0,2 \cdot \log_2(0,2) + 0,25 \cdot \log_2(0,25) + 0,1 \cdot \log_2(0,1)) =$$

$$= + 0,522 + 0,332 + 0,126 + 0,1464 + 0,1500 + 0,1332 =$$

$$= 2,365$$

→ Nº bits mínimo para codificar.

$$H = 2,365$$

bits  
símbolo

EFICIENCIA [μ]

$$\mu = \frac{H}{N_p} = \frac{2,365}{2,4} \frac{\text{bits}}{\text{símbolo}} = 0,985 \rightarrow 98,5\%$$

•  $\text{Jodatmung} = \text{H}_2\text{O} + \frac{1}{2}\text{O}_2$

$\downarrow$   
• animal  
animal Teil,  
abstrakt

$\frac{\text{Zahl}}{\text{Wert}}$

$$= 1.10.1 + (25.10).S + (5.10).S + (200.10).H + (1.10).S + (10.10).S = H$$

MAMMEN  $\leftarrow$   $\frac{\text{Zahl}}{\text{Wert}}$   $\rightarrow$   $\text{H}_2\text{O}$

•  $\text{Oxidation} = \frac{\text{Zahl}}{\text{Wert}}$

durch ab animal. abstr. animal.  $\rightarrow$  abstrakt  
animal. animal.  $\rightarrow$  abstrakt  
 $\rightarrow$  animal. animal.  $\rightarrow$  abstrakt  
(H) Alkohol: oxidierendes reag. auf animal.

$$(H)_{\text{sofol.}; 10} \frac{S}{d=3} = H$$

$$\Rightarrow (5.10)_{\text{sofol.} 10} + (10)_{\text{sofol.} 5.10} + (8.10)_{\text{sofol.} 8.10} = H$$

$$= 500.10 + 002.10 + 400.10 + 0.10 + 500.10 + 0.10 + 0.10 + =$$

•  $\text{Volumen} = \frac{\text{Zahl}}{\text{Wert}}$   $\rightarrow$   $S_{\text{sofol.}} =$

$$\boxed{\frac{\text{Zahl}}{\text{Wert}} = \frac{200}{1000} = \frac{200}{1000} = H}$$

$$\cdot \text{Op. Z. } S = \frac{\frac{200}{1000} \text{ Z. } S}{\frac{200}{1000} \text{ H. } P_{10}} = \frac{H}{P_{10}} = M$$

ENTROPIA

$$H = - \sum_{i=1}^n P_i \cdot \log_2 (P_i)$$

Eficiencia  $\rightarrow \mu = \frac{H}{N_p}$

HUFFMAN

→ Huffman en un fichero comprimido:

- Se approxima mejor.

↳ Cuando la probabilidad de aparición de los símbolos es diferente y cuanta más diferente mejor.

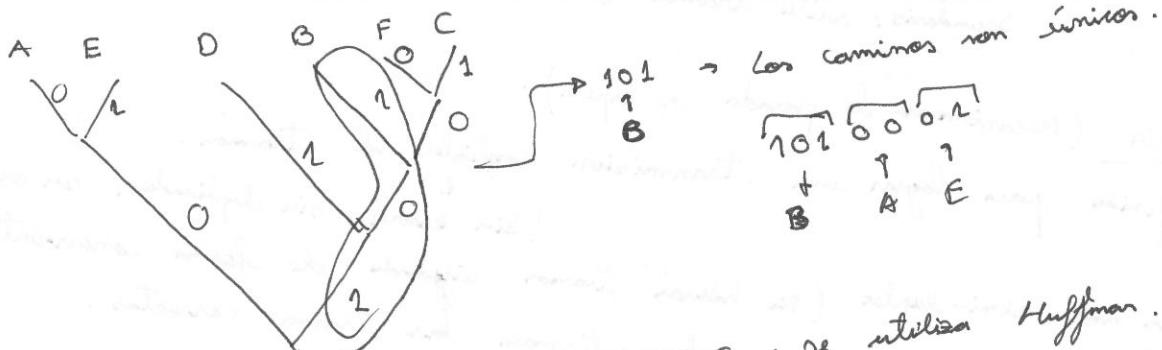


→ Símbolos , serán los leídos.

$$A = 101$$

Urga:

$\overbrace{101 \dots}^{\text{el símbolo codificado jamás es el inicio de la codificación de otro símbolo.}}$  (PROPIEDAD DEL PRE-Fijo).



$\overbrace{101}^B \overbrace{00}^A \overbrace{01}^E$

→ Los caminos son únicos.

→ IMÁGENES COMPRIMIDAS EN JPEG, se utiliza Huffman.  
JPEG: pone a cero el contenido frecuencial, donde errores más probables que otros.

PROTOCOLOS DE CONTROL DEL ENLACE DE DATOS

$$\text{CONTROL ERRORES} = \text{DATOS} + \text{CORRECCIÓN}$$

↳ Control de errores.

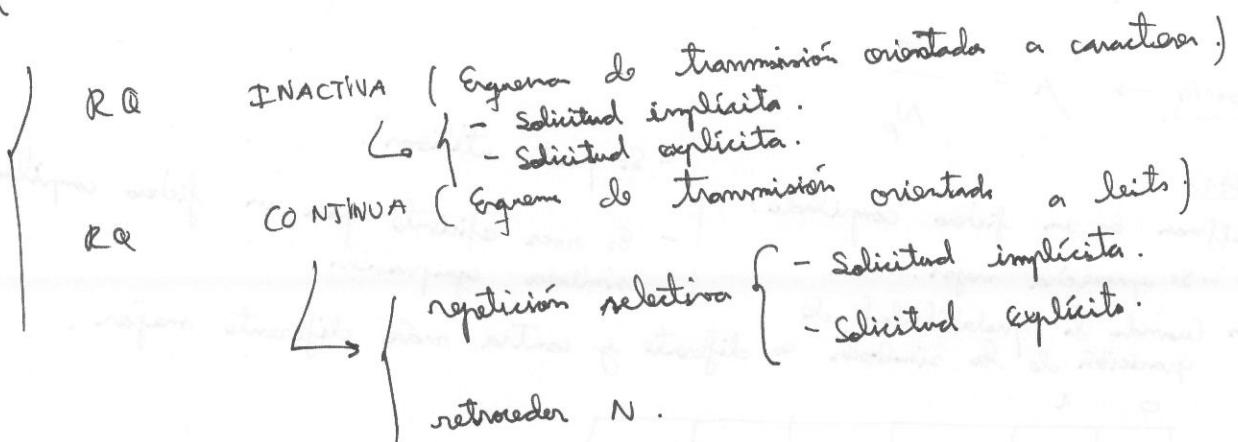
↳ Control de flujo.

↳ Gestión del enlace.

## CONTROL DE ERRORES

- 2 equipos transmitiendo tramas. Rx envía paquete aviso de errores. (Solicita repetición de trama).

ARQ (Automatic Repeat Request) → MECANISMO



NOMENCLATURA: Tramas I (tramas de información, transportan datos).

Tramas de control (Necesarias para implementar al protocolo, mensajes de confirmación).

Primaria: envía trama (Tx).

Secundaria: recibe trama. (Rx)

RQ INACTIVA (Mecanismos de parada y espera).

→ Está definido para lograr una transmisión confiable de tramas.  
(Sin errores, sin duplicados, en orden)

Trabaja en modo semi-duplex (No habrá tramas restringidas de forma concurrente).

- Solicitud implícita: el secundario solo confirmará las tramas correctas.

TRAAMA CORRECTA → ENVÍA CONFIRMACIÓN

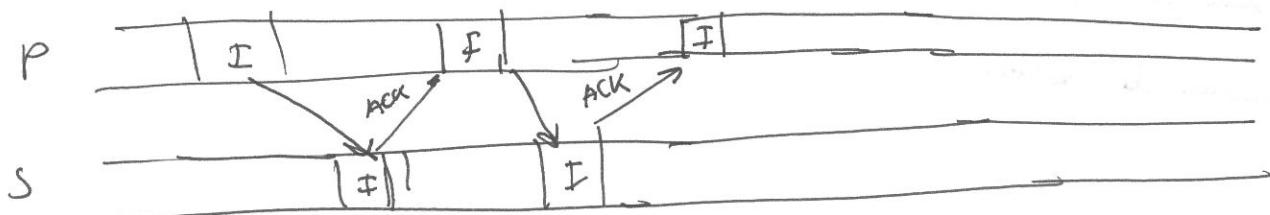
TRAAMA INCORRECTA → NO RESPONDE

- Solicitud explícita: responderá aunque la trama sea incorrecta, pedirá un reenvío.

TRAAMA CORRECTA → ENVÍA CONFIRMACIÓN

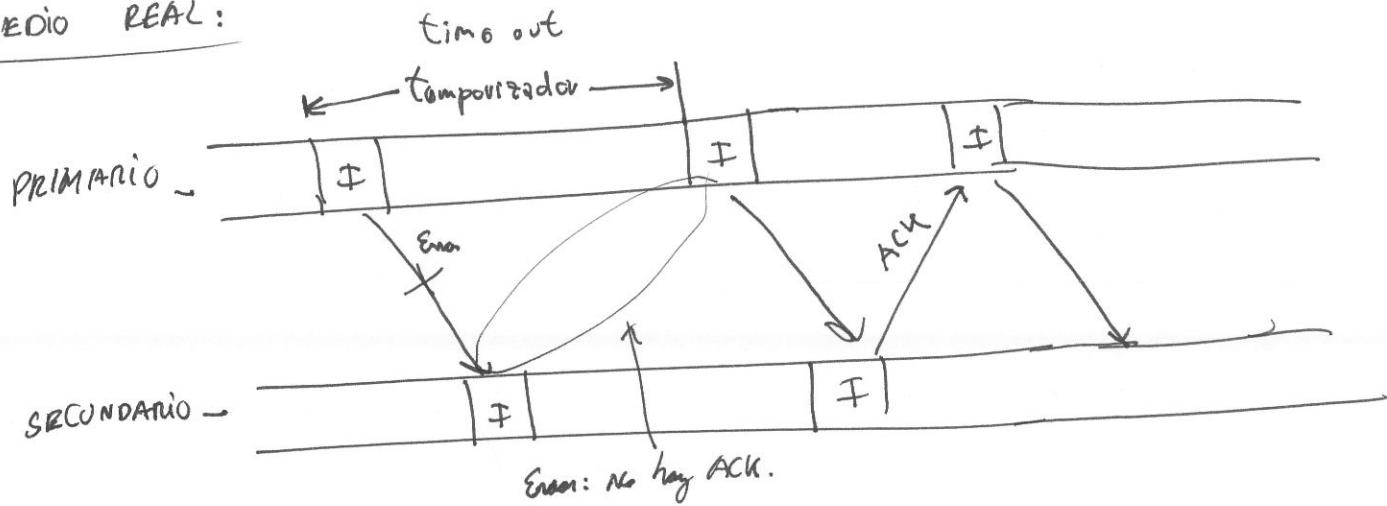
TRAAMA INCORRECTA → PIDE REENVÍO.

MÉDIO IDEAL

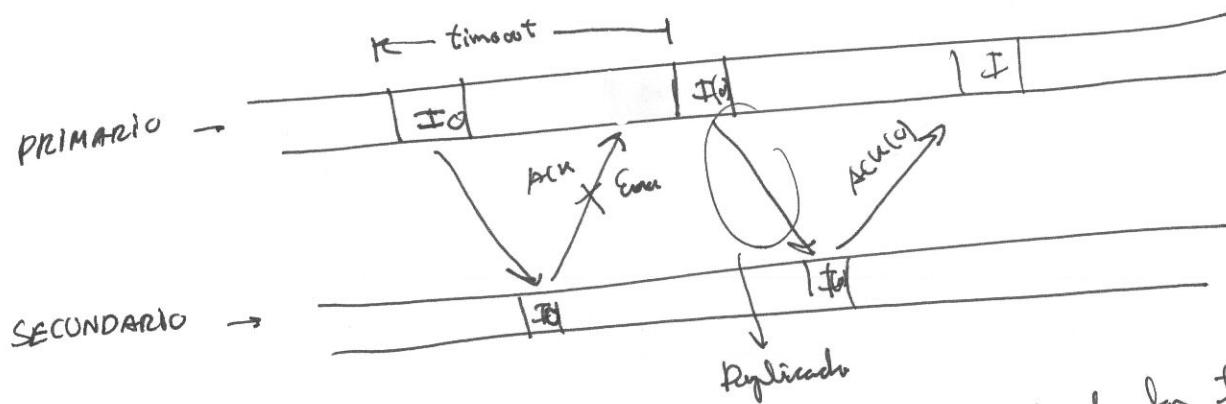


RQ INACTIVA - SOLICITUD IMPÍCITA

MÉDIO REAL:



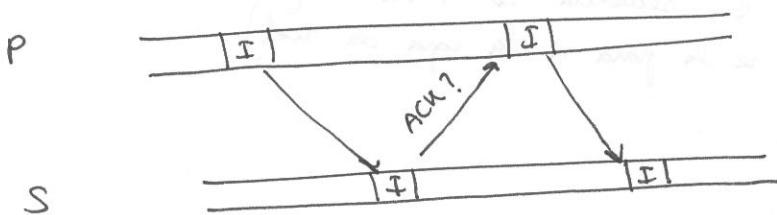
→ ACK también puede ser erroneo.



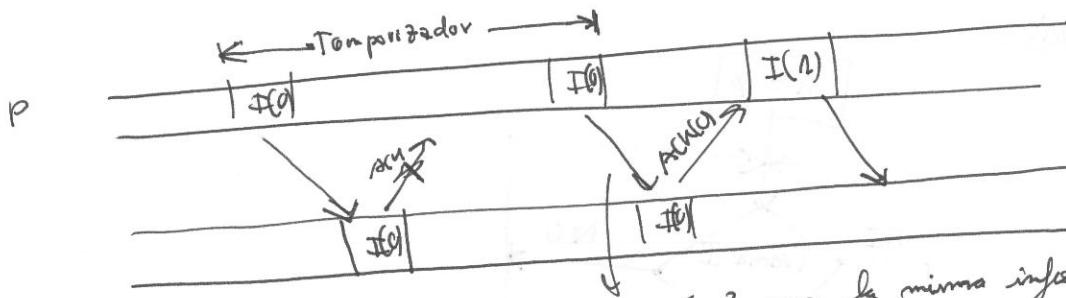
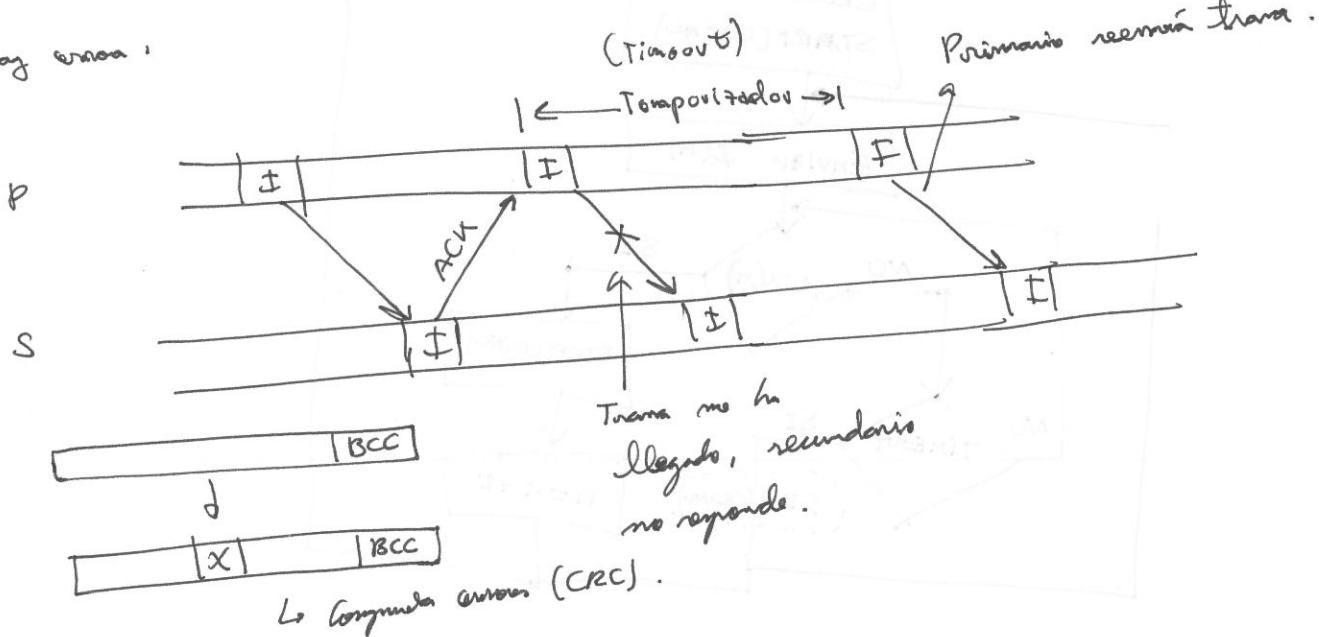
- Evitar el problema con un prefijo, cambiando los tramas numerados.
- El secundario detecta el trama duplicada y responde a confirmar la trama (ACK).

RQ INACTIVA

- Métodos de respuesta inventiva . (ARQ) .



- Mientras no reciba la confirmación ACK no envía la trama siguiente.
  - Si hay error:
    - (Timeout)
    - Primario re



Protegerá 2 veces la misma información.  
Solución: numerar los trámites.  
Duplicado detectado.

PRIMARIO (Mecanismos de parada y apertura) Solic.

PRIMARIO (Mecanismos de parada y freno) RQ Inactiva por retrotransmisión de solicitud implícita. (I<sub>(n)</sub>) y se queda en la espera de respuesta.

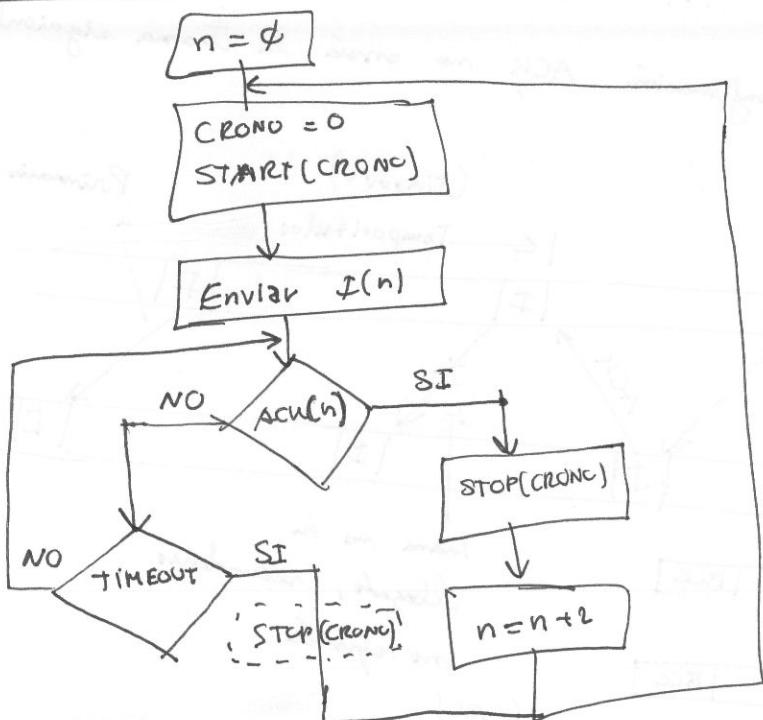
- Inician temporizadores.
  - Si recibe ACK dañada  $\Rightarrow$  Resinton.
  - Si recibe ACK libre de errores  $\Rightarrow$  Transmite la siguiente trama.
  - Si vence el temporizador  $\Rightarrow$  se retransmite  $\neq (n)$ .

## SECUNDARIO

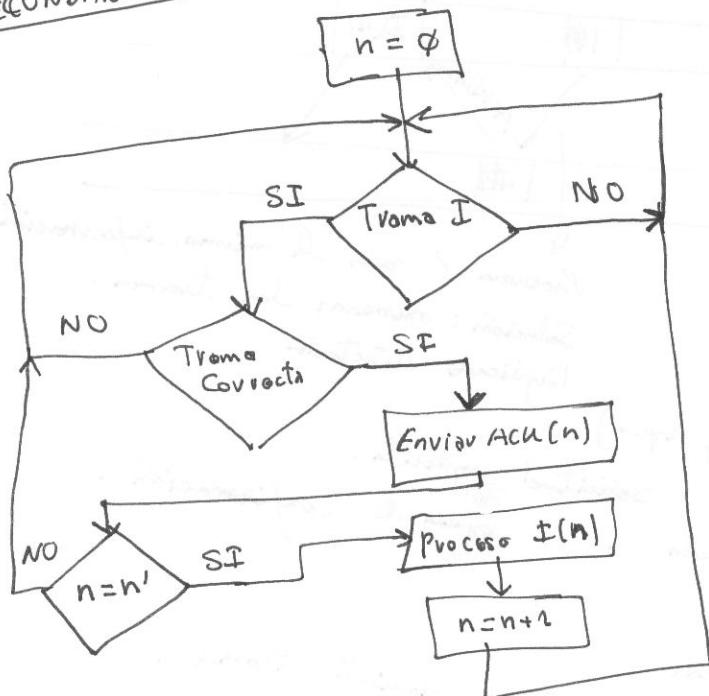
- Si recibe trama I dañada  $\Rightarrow$  Desear.
- Si recibe trama I libre de errores  $\Rightarrow$  Confirma (ACK):
  - + Si es un duplicado  $\Rightarrow$  Desear.
  - + Si es la siguiente en secuencia  $\Rightarrow$  Procesar (Anota cabecera y cola y la parte de datos se la pasa a la cola de red).

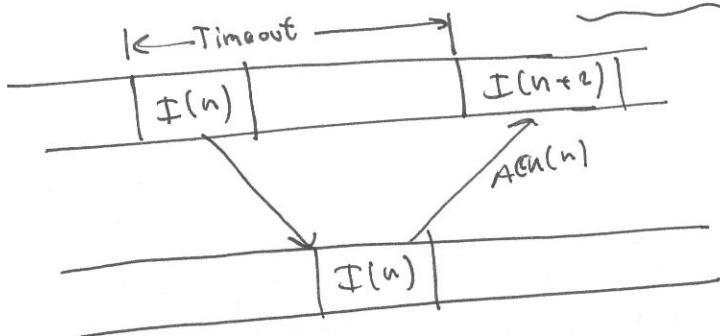
Diagrama de secuencias:

## PRIMARIO



## SECUNDARIO





Timeout: nunca debe reenviar antes de que llegue una confirmación.

↑ No debe ser demasiado corto.

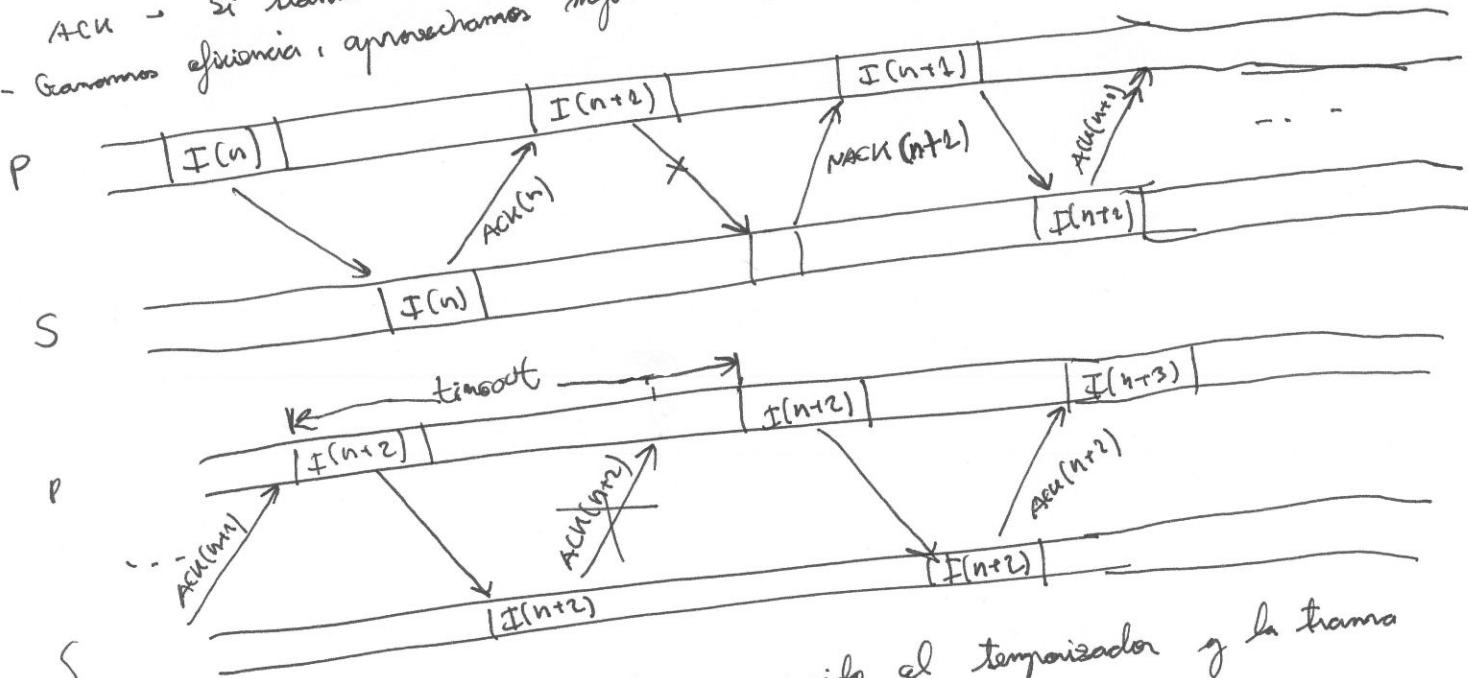
RE INACTIVA DE SOLICITUD EXPLÍCITA

→ El secundario se puede mejorar si la envía de que la trama llega dañada.

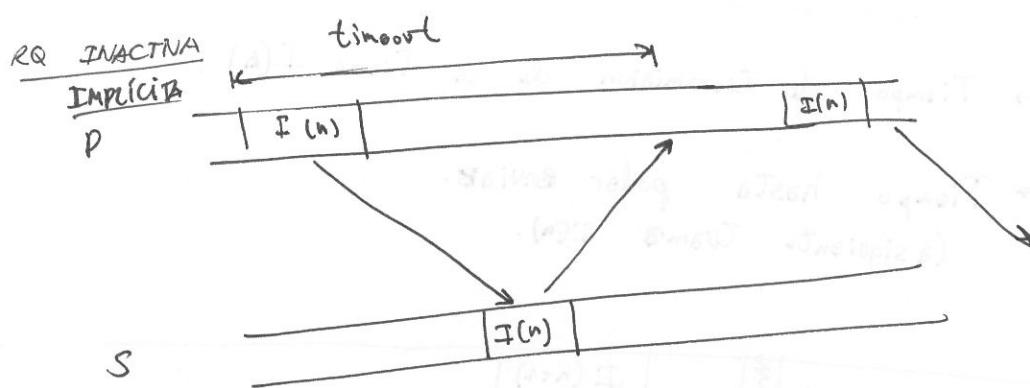
NACK → Trama I dañada.

ACK → Si trama OK.

- Gravamos eficiencia, aprovechamos mejor la capacidad de retransmisión del enlace.



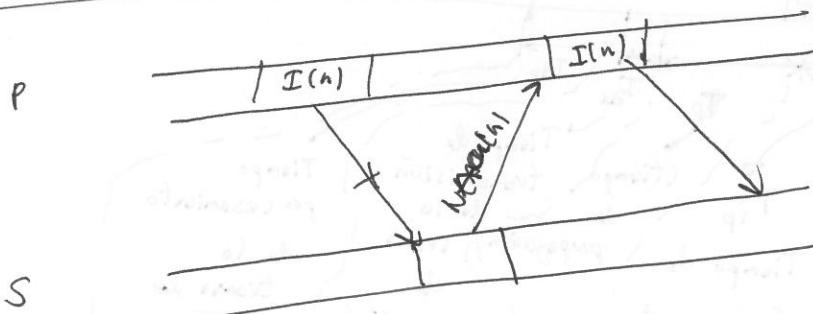
→ Si NACK hubiera estado dañado, hubiera reenviado el temporizador y la trama hubiera sido reenviada.



BER

↳ La eficiencia dependerá de la tasa de errores.

RQ INACTIVA EXPLÍCITA

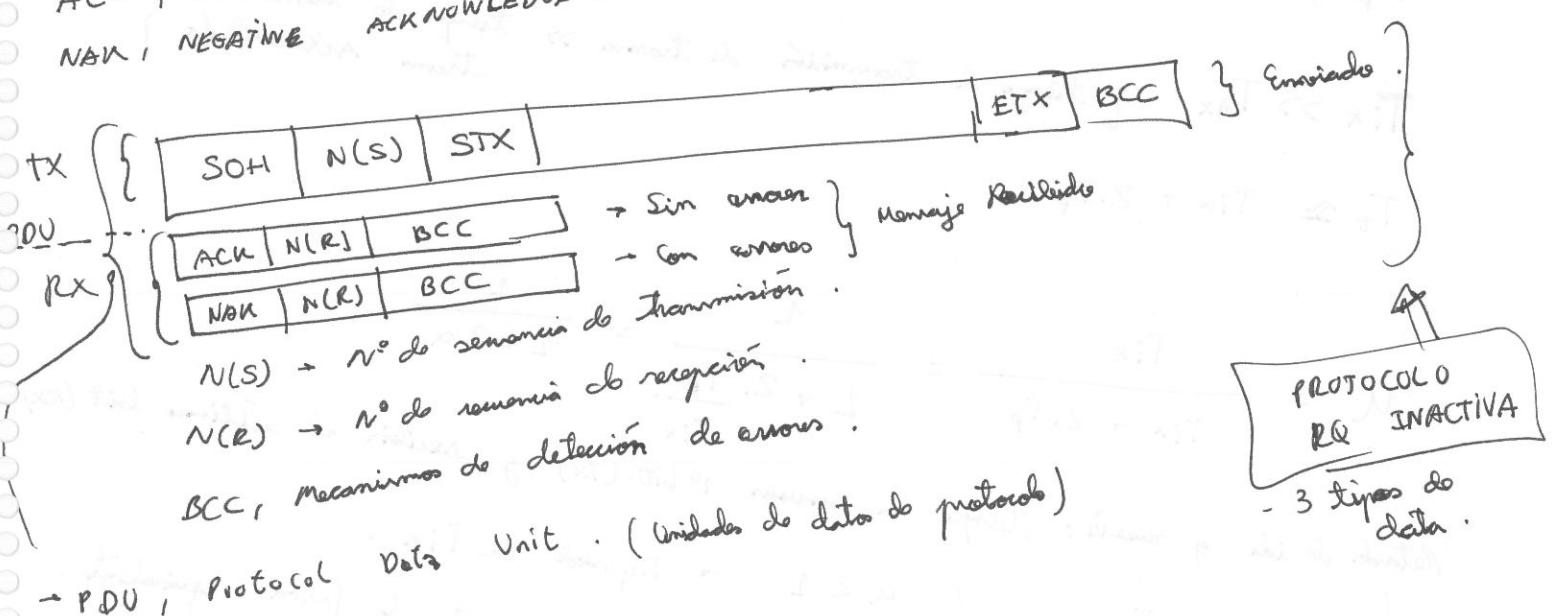


STX, ETX, DLE.

SOH → Start Of Header (Inicio de cabecera)

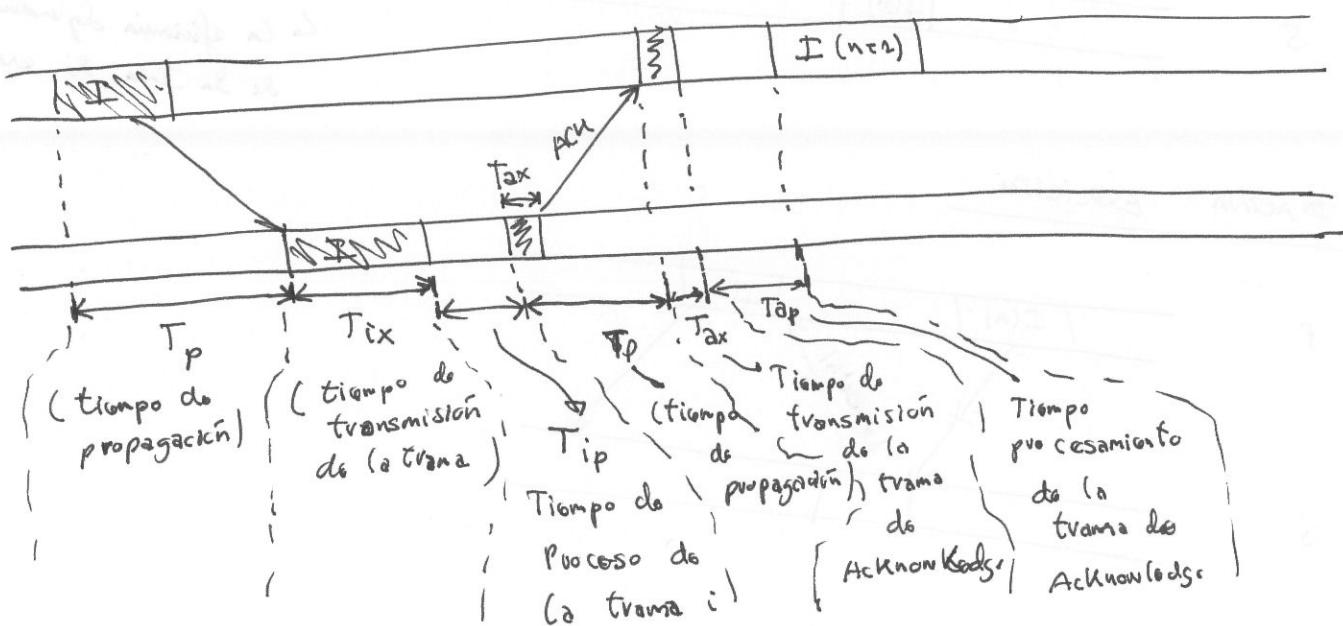
ACK, ACKNOWLEDGE.

NACK, NEGATIVE ACKNOWLEDGE.



# GRADO DE UTILIZACIÓN DEL ENLACE

$$U = \frac{T_{ix}}{T_t} \rightarrow \begin{array}{l} \text{Tiempo de transmisión de la Trama I.} \\ \text{Tiempo hasta poder enviar.} \\ (\text{las siguientes Tramas I.}) \end{array}$$



$$\left. \begin{array}{l} T_{ip} \\ T_{ap} \end{array} \right\} \text{Despreciables} \quad \text{resiste} \propto T_p \quad \left\{ \begin{array}{l} T_p \gg T_{ip} \\ T_p \gg T_{ap} \end{array} \right.$$

$$T_{ik} \gg T_{ax} \quad [ \text{tiempo de transmisión de trama} \gg \text{tiempo de transmisión de trama ACKnowledged} ]$$

$$T_t \cong T_{1\alpha} + 2 \cdot T_p$$

$$u = \frac{T_{ix}}{T_{ix} + 2 \cdot T_p} = \frac{\lambda}{\lambda + 2 \cdot \frac{T_p}{T_{ix}}} = \frac{\lambda}{\lambda + 2 \cdot a}$$

Retardo de ida y vuelta: tiempo en avion 10 bit (Tx)  $\Rightarrow$  ~~salida~~  
Retardo de Tx.

$$a = \frac{T_p}{T_{ix}}$$

$a < 1$  → Depende de  $T_{ix}$ .  
 $a \approx 1$  →  $T_p, T_{ix}$  afetam de forma equivalente.  
 $a > 1$  → Depende de  $T_p$ .

Calcular al grado de utilización del Enlace: [u]

Lo transmitir tramas de 1000 bits en 3 situaciones diferentes:

a) Enlace con par trenzado de 1 Km.

b) Enlace de fibra óptica de 200 Km.

c) Enlace satélite a 50.000 Km.

i) Tasas de transmisión

i) 1 kbps

ii) 2 Mbps

$$u = \frac{T_{tx}}{T_{tx} + T_p}$$

$$u = \frac{T_{tx}}{T_{tx} + 2 \cdot T_p}$$

$$v_p = 2 \cdot 10^8 \text{ m.s}^{-2}$$

ESTA MÁL !!!  
ES DIVIDIDO.

$$T_{tx} = \left( \begin{array}{l} \text{bits a} \\ \text{transmitir} \end{array} \right) \times \left( \begin{array}{l} \text{Velocidad} \\ \text{de} \\ \text{Transmisión} \end{array} \right) = 1000 \times \left( \begin{array}{l} \text{= 2 Mbps} \end{array} \right) = 1 \text{ s}$$

$$T_p = \frac{\text{(Longitud del medio)}}{\text{(Velocidad del medio)}} = \frac{2 \text{ km}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} =$$

$$= \frac{1000 \text{ m}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{10^3 \text{ m}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{1}{2} \cdot 10^{-5} \text{ s}$$

$$= 50 \mu\text{s}$$

→ En caso muy particular la inactiva es suficiente.

$$2) \frac{1 \text{ Mbps}}{u} = \frac{T_{tx}}{T_{tx} + 2 \cdot T_p} = \frac{1 \text{ s}}{1 \text{ s} + 2(50 \mu\text{s})} = \frac{1 \text{ s}}{(1.000100) \text{ s}} = 0,5555 \text{ s}$$

$$\frac{2 \text{ Mbps}}{u} = \left( \begin{array}{l} \text{bits a} \\ \text{transmitir} \end{array} \right) \times \left( \begin{array}{l} \text{Velocidad} \\ \text{de} \\ \text{Transmisión} \end{array} \right) = 1000 \times (2 \text{ Mbps}) = 1000 \text{ s}$$

$$u = \frac{1000 \text{ s}}{1000 \text{ s} + 2(50 \mu\text{s})} = 0,5555555 \text{ s}$$

$$b) \text{ Mbps} \quad t_p = \frac{\text{Longitud del medio}}{\text{Velocidad del medio}} = \frac{200 \text{ Km}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{200 \cdot 10^3}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{2 \cdot 10^3}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \\ = 10^{-3} \text{ s} = 0,001 \text{ s} = 1 \text{ ms}$$

$$\tau_{tx} = 1 \text{ s} \times 1 \text{ ms}$$

$$u = \frac{\tau_{tx}}{\tau_{tx} + 2 \cdot t_p} = \frac{1 \text{ s}}{1 \text{ s} + 2 \cdot (1 \text{ ms})} = \frac{1 \text{ s}}{1 \text{ s} + 2 \cdot (1 \text{ ms})} = 0,998$$

$\xrightarrow{x(1ms)}$   
 $\uparrow y(1ms)$

c) Mbps

$$\tau_{tx} = 1000 \text{ s} \rightarrow \tau_{tx} = 1 \text{ ms}$$

$$u = \frac{\tau_{tx}}{\tau_{tx} + 2 \cdot t_p} = \frac{1000 \text{ s}}{1000 \text{ s} + 2 \cdot (1 \text{ ms})} = 0,999998$$

c)

Mbps

$$\tau_{tx} = 1 \text{ s}$$

$$t_p = \frac{50.000 \text{ Km}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{5 \cdot 10^4 \cdot 10^3 \text{ m}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{5 \cdot 10^7 \text{ m}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = \frac{5}{2} \cdot \frac{1}{10^1} \text{ s} = \\ = \frac{5}{2} \cdot 10^{-1} \text{ s} = 0,125 \text{ s}$$

$$u = \frac{\tau_{tx}}{\tau_{tx} + 2 \cdot (t_p)} = \frac{1 \text{ s}}{1 \text{ s} + 2 \cdot (0,125 \text{ s})} = \frac{1 \text{ s}}{1 \text{ s} + 0,25 \text{ s}} = \frac{1 \text{ s}}{1,25 \text{ s}} = 0,8$$

Mbps

$$\tau_{tx} = 1000 \text{ s} \rightarrow 1 \text{ ms}$$

$$u = \frac{1000 \text{ s} (1 \text{ ms})}{1000 \text{ s} + 2 \cdot (t_p)} = \frac{1000 \text{ s}}{1000 \text{ s} + 2 \cdot (1 \text{ ms})} = \frac{1000 \text{ s}}{1000 \text{ s} + 2 \cdot 10^{-3} \text{ s}} = 0,999998$$

X.C.

Transm 1000 bits

- a) Enlace por tronzado de 1 Km.  
 b) Enlace de fibra óptica de 200 Km.  
 c) Enlace satélite de 50000 Km.

$$v_p = 2 \cdot 10^8 \text{ m/s}$$

- i) Tasas de transmisión  $\begin{cases} \text{i)} 1 \text{ Kbps} \\ \text{ii)} 1 \text{ Mbps} \end{cases}$

d)

i)

$$\mu = \frac{l}{1 + 2 \cdot \frac{T_p}{T_{ix}}}$$

$$T_{ix} = 1s = 100 \text{ bits} \cdot \frac{1 \text{ seg}}{1000 \text{ bits}} = 1s$$

$$T_p = 1 \text{ Km} / 2 \cdot 10^8 \frac{\text{m}}{\text{s}} = \frac{1}{2} \cdot 10^{-5} \text{ s}$$

$$\mu = \frac{l}{1 + 2 \cdot \frac{\frac{1}{2} \cdot 10^{-5}}{1}} = \frac{l}{1 + 10^{-5}} \approx 1$$

ii)

$$T_{ix} = 1000 \text{ bit} \cdot \frac{1 \text{ seg}}{10^6 \text{ bits}} = 10^{-3} \text{ segundos}$$

$$T_{ix} = \frac{\text{bits a transmitir}}{\text{Velocidad de transmisión}}$$

$$T_p = \frac{\text{Longitud del medio}}{\text{Velocidad en el medio}}$$

$$\mu = \frac{l}{1 + 2 \cdot \frac{\frac{1}{2} \cdot 10^{-5}}{10^{-3}}} = \frac{l}{1 + 10^2} \approx 1$$

iii)

$$T_p = \frac{200 \text{ Km}}{2 \cdot 10^8 \frac{\text{m}}{\text{s}}} = 1 \text{ ms}$$

$$\mu = \frac{l}{1 + 2 \cdot \frac{10^{-3}}{1}} = \frac{l}{1 + 2 \cdot 10^{-2}} \approx 1$$

ii)

$$T_{tx} = 1000 \text{ bits} \cdot \frac{1 \text{ sec}}{10^6 \text{ bits}} = 10^{-3} \text{ s}$$

$$u = \frac{1}{1 + 2 \cdot \frac{10^{-3}}{10^{-3}}} = \frac{1}{3} \quad \rightarrow \text{El queachamiento no es óptimo.}$$

c)

$$T_{tx} = 1000 \text{ bits} \cdot \frac{1 \text{ sec}}{1000 \text{ bits}} = 1 \text{ s}$$

i)

$$T_p = 5 \cdot 10^9 \text{ m} \cdot \frac{1 \text{ sec}}{2 \cdot 10^8 \text{ m}} = \frac{5}{2} \cdot 10^{-2} \text{ s}$$

$$u = \frac{1}{1 + 2 \cdot \frac{5/2 \cdot 10^{-2} \text{ s}}{1 \text{ s}}} = \frac{1}{1 + \frac{1}{2}} = \frac{2}{3}$$

ii)

$$T_{tx} = 100 \text{ bits} \cdot \frac{1 \text{ sec}}{10^6 \text{ bits}} = 10^{-3} \text{ s}$$

$$u = \frac{1}{1 + 2 \cdot \frac{\frac{5}{2} \cdot 10^{-2}}{10^{-3}}} = \frac{1}{1 + 5 \cdot 10^2} \approx 0$$

En enlaces cortos el grado de queachamiento del enlace es independiente de la tasa de transmisión a no ser que sean exageradas, (máderas mejor)

RQ

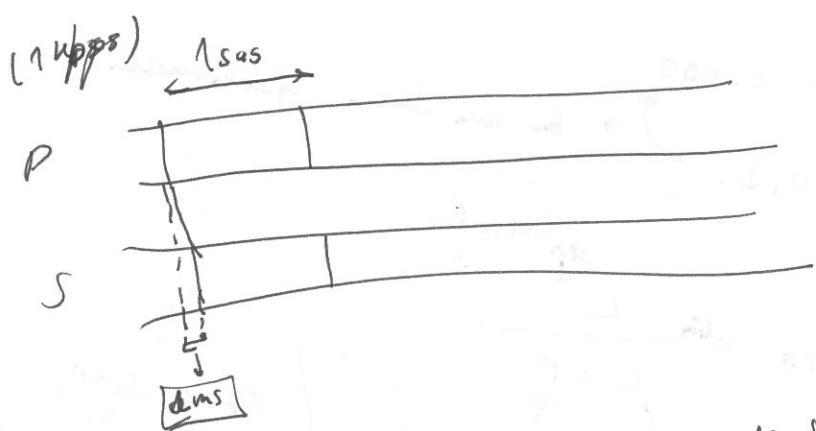
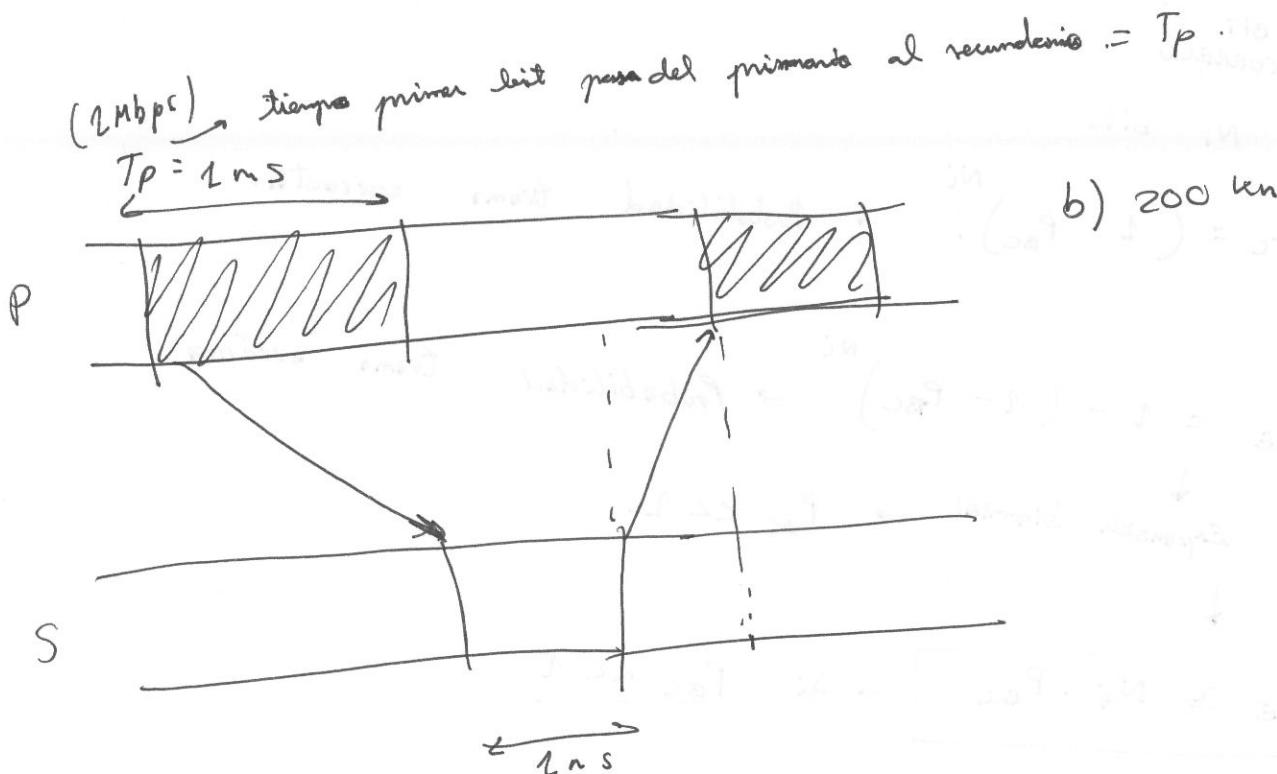
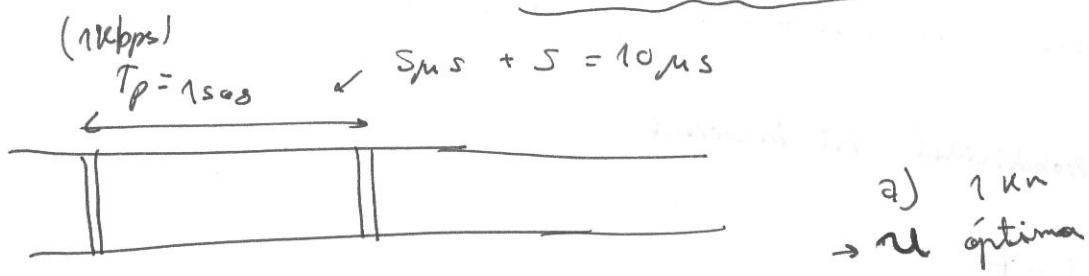
INACTIVA

+ Distancia corta, u aceptable

- Distancia media, u Mala si velocidad 99

- Distancia larga, u Mala (independiente de velocidad)

+ Distancia larga,



RQ INACTIVA solo eficiente en casos particulares.  
 $N_r$ : N° de intentos (promedio) para hacer que una trama se envíe correctamente.

↳  $\alpha = \frac{1}{N_r(1+2\alpha)}$

BER  $\rightarrow$  Probabilidad bit transmitida llegado con errores.

$P \approx$  Probabilidad bit incorrecto

$$P_{BC} = 1 - P$$

↑

BIT CORRECTO

TRAMA  $N_i$  bits

$$\hookrightarrow P_{TC} = (1 - P_{BC})^{N_i} \rightarrow \text{Probabilidad trama correcta.}$$

↓

$$P_{TE} = 1 - (1 - P_{BC})^{N_i} \rightarrow \text{Probabilidad trama errónea.}$$

↓  
Expansión binomial  $\rightarrow P_{BC} \ll 1$ .

$$P_{TE} \approx N_i \cdot P_{BC} \rightarrow \text{Si } P_{BC} \ll 1$$

TRAMA 1000 bits

$$10^{-4} \rightarrow 1 - (1 - 10^{-4})^{1000} = 0,095 \quad \begin{matrix} \uparrow \\ \text{aproximación} \end{matrix}$$

$$1000 \cdot 10^{-4} = 0,1.$$

$$P_{TC} = 1 - P_{TE} \rightarrow 1 - P_{TE} = \frac{\text{Correctos}}{\text{Correctos} + \text{Erróneos}}$$

$$Nr = \frac{C+F}{C}$$

$$\frac{1 - P_{TE}}{C} = \frac{1}{C+F} \Rightarrow \frac{Nr}{C+F} = \frac{1}{C} = \frac{4}{1 - P_{TE}}$$

$$u = \frac{1}{Nr \cdot (1+2a)} = \frac{1 - (P_{TE})^{N_i} \cdot P_{BC}}{1+2a}$$

RQ CONTINUA

+ Eliminar la restricción que tiene el primario de tener que esperar un Acknowledgment.

Tramas 1000 bits

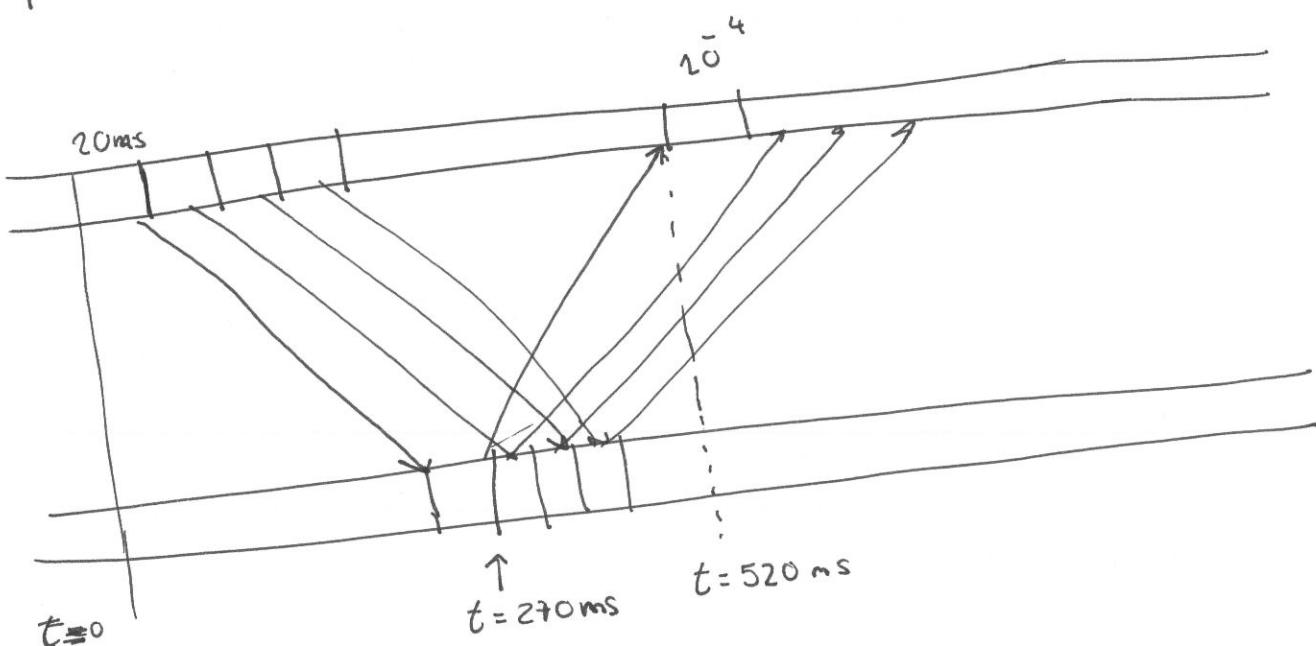
Transmitir a 50 kbps

$$\text{Tiempo de propagación} = \left( \frac{\text{distancia}}{2} \right) = \frac{500 \text{ ns}}{2} = 250 \text{ ns}$$

$$\text{Tiempo Ido/Vuelta} = 500 \text{ ns}$$

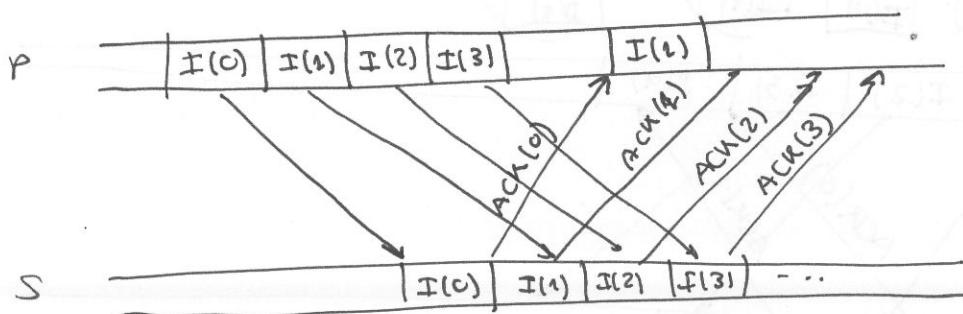
$$t_{ix} = 20 \text{ ms}$$

$$T_p = 250 \text{ ms}$$



$$\frac{500 \text{ ms}}{520 \text{ ms}} = 0,96 \rightarrow \text{Estoy parado.}$$

RQ CONTINUA



### VENTAJAS

- ↑ Eficiencia .
- Transmisiones Full-Duplex .  
(Imprescindibles)
- ↳ "no es propriamente una ventaja".

### INCONVENIENTES

- ↑ Espacio de almacenamiento.  
(Debo guardar más tramas.)
- Mas espacio en el Primario y en algún caso en el Secundario.
- Algoritmos más complejos .

→ LISTA DE RETRANSMISIÓN  
↳ Una estructura de datos transmitidas y no

en la que el Primario almacena las tramas confirmadas. Opera como una FIFO.

### LISTA DE RECEPCIÓN

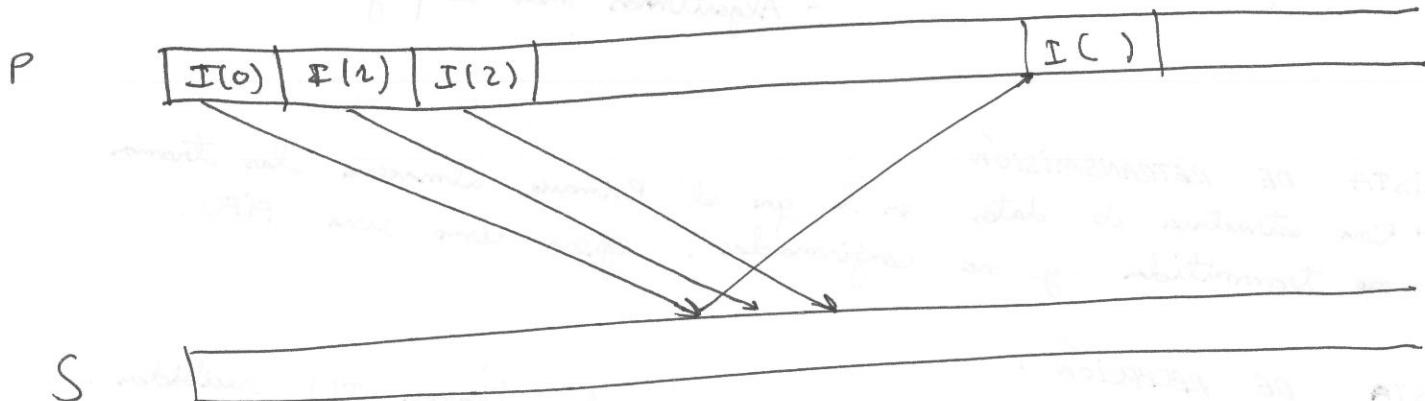
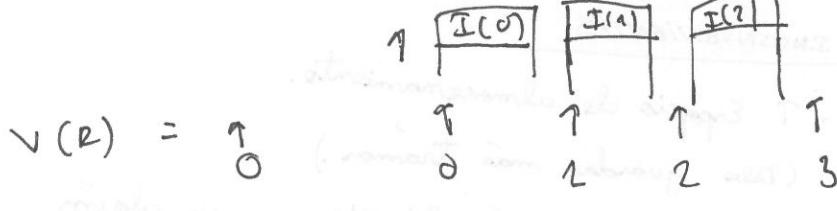
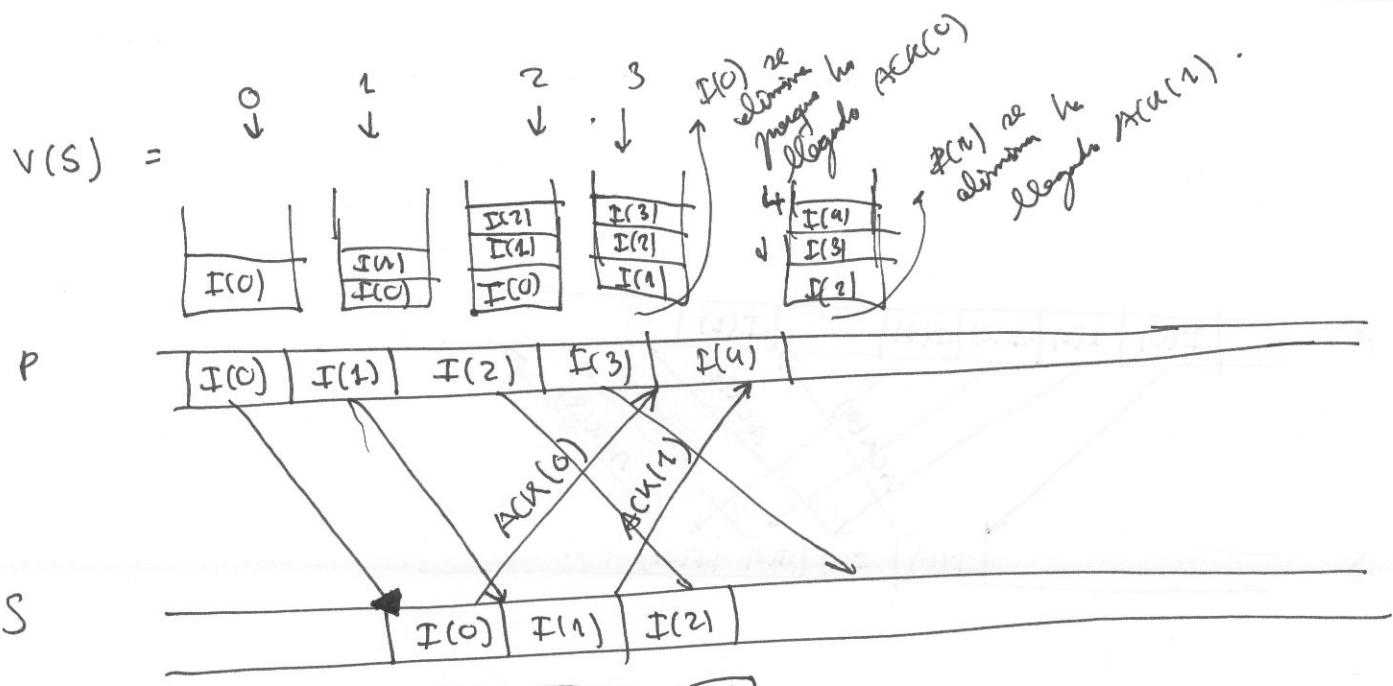
↳ Una estructura de datos donde se almacenan las tramas I(1) recibidas .

### TRANSMISIÓN (V(S))

VARIABLE SECUENCIA DE TRANSMISIÓN (V(S))  
↳ Donde se registra el número de secuencia que tendrá la siguiente trama I(1).

### (V(R))

VARIABLE SECUENCIA DE RECEPCIÓN (V(R))  
↳ Variable que tiene el secundario que indica cual es la siguiente trama I(1) en la secuencia correcta .



→ 2 formas de abordar el problema : + Retroceder N.

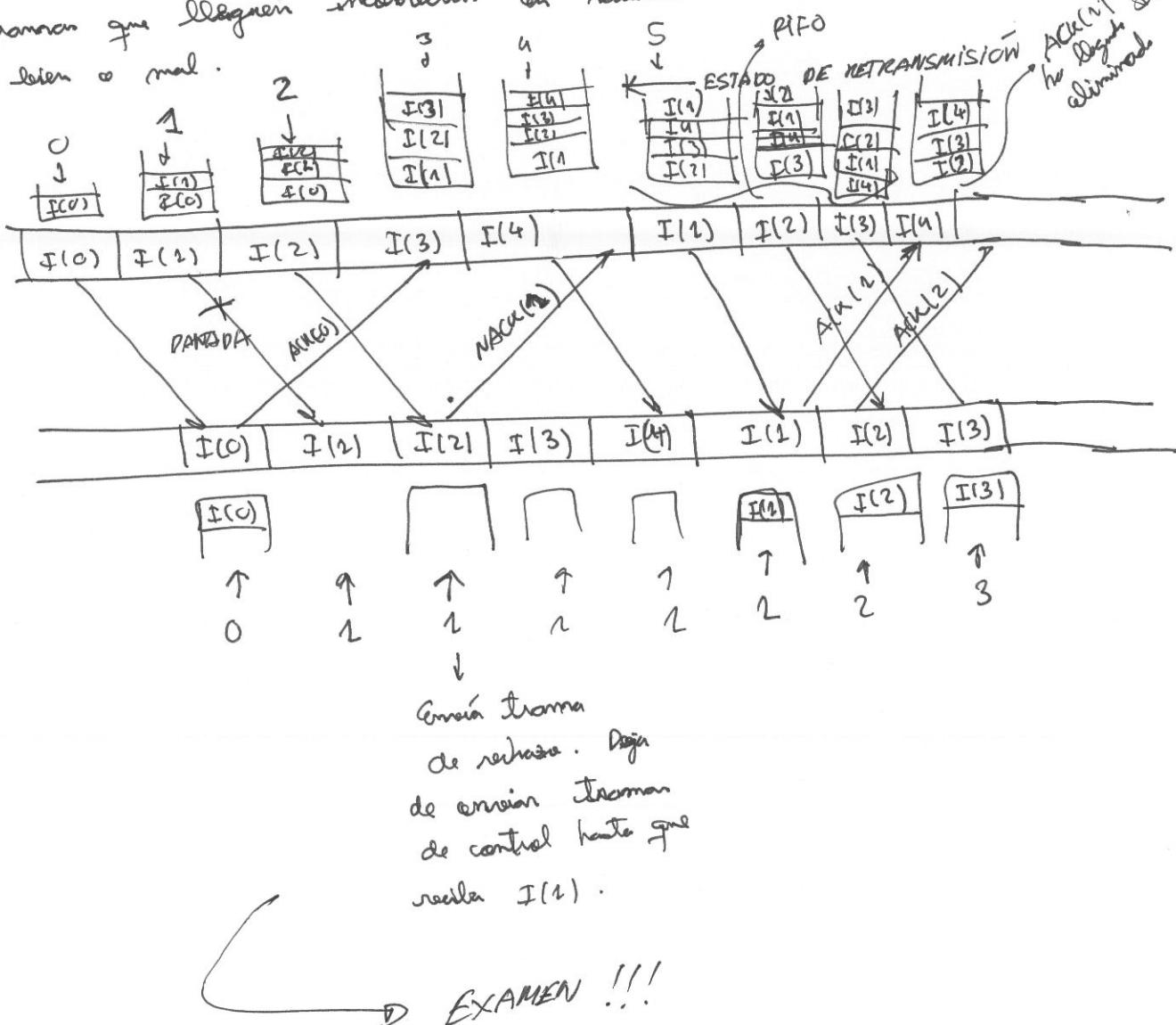
+ Retransmisión de todas las tramas des de aquella que no se ha recibido.

+ Repetición Selectiva :

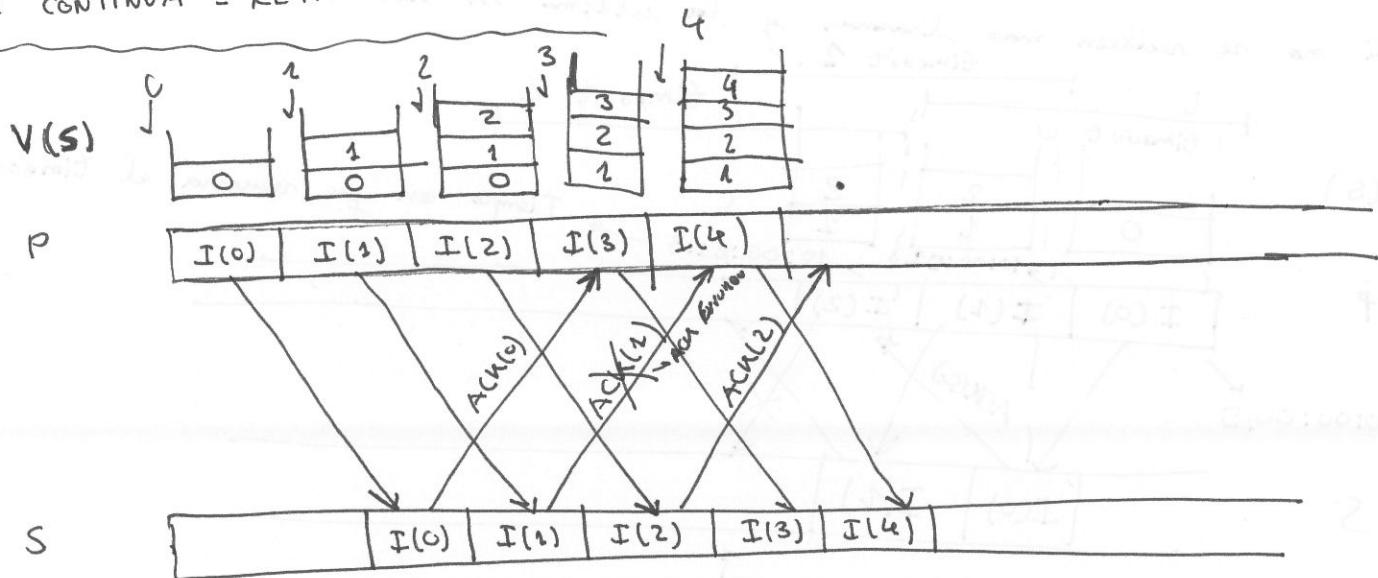
secundario detecta que le falta alguna trama, pedirá al primario que retransmita la trama que le falta en secuencia.

## RETROCEDER N

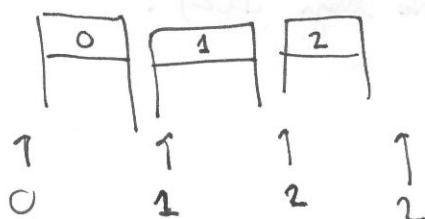
• El secundario solo acepta tramas correctas y en secuencia. Los tramas que lleguen incorrectas en secuencia las va a desechar estén bien o mal.



RQ CONTINUA - RETROCEDER N

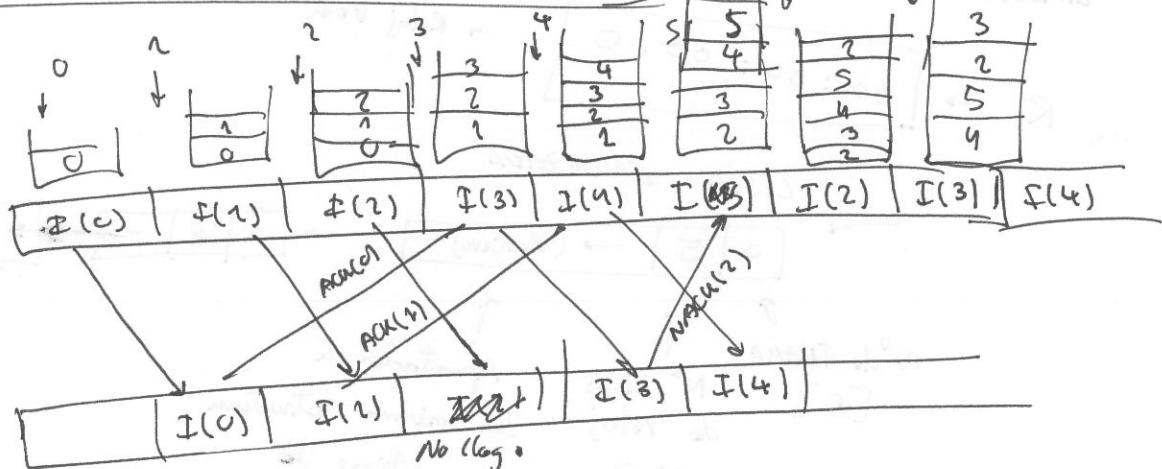


$$V(R)$$

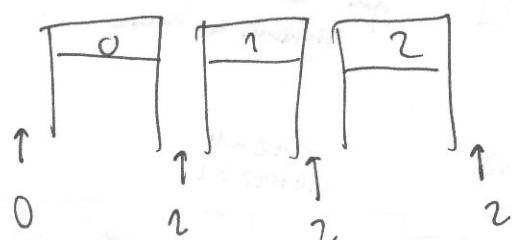


- Solo puedo recibir una trama de control si el secundario ha recibido la trama actual y todas las anteriores.

$$V(s) =$$



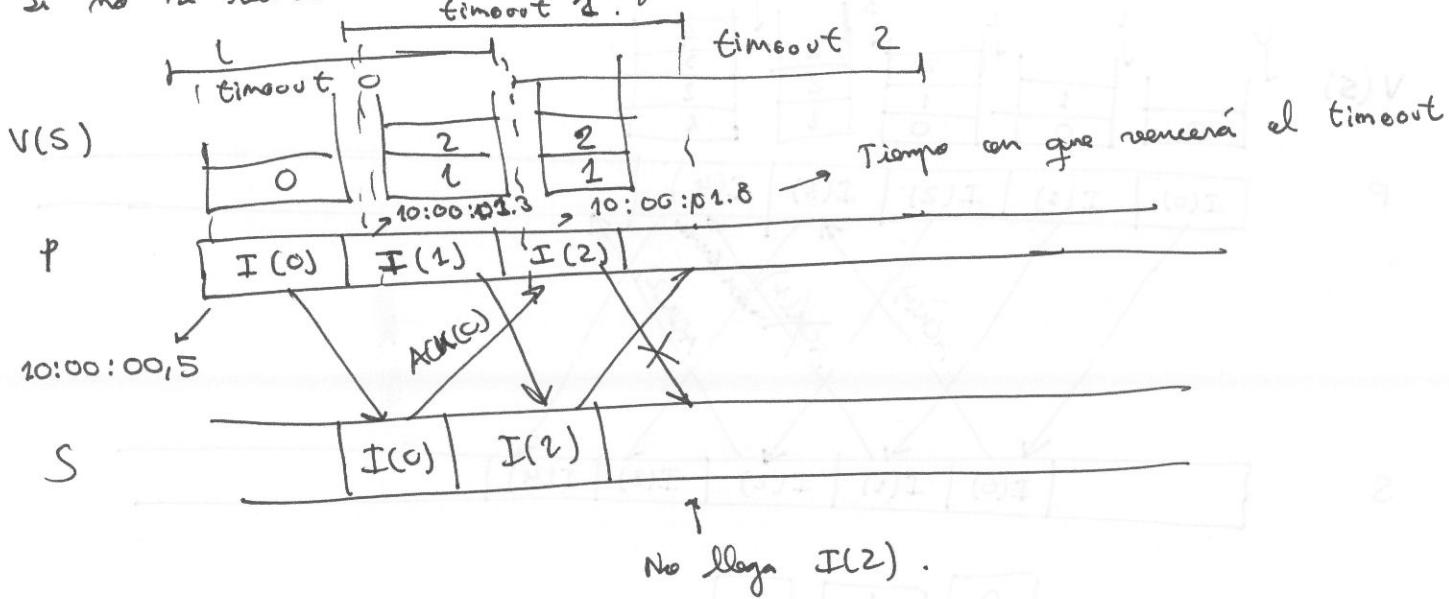
$$v(r)$$



- Se descomponen fracciones correctas porque no han llegado ~~a~~ en la secuencia correcta.

→ Si se recibe la siguiente trama correcta se aviva al promotor.

→ Si no se reciben más tramas y la última ha sido incorrecta.



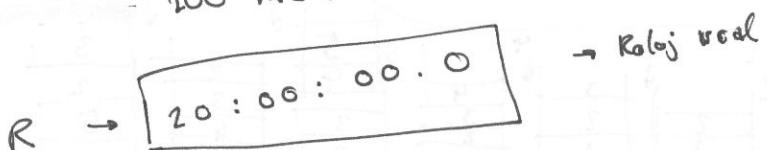
V(R)

- Tanto temporizadores como tramas pendientes de retransmisión. ¿?

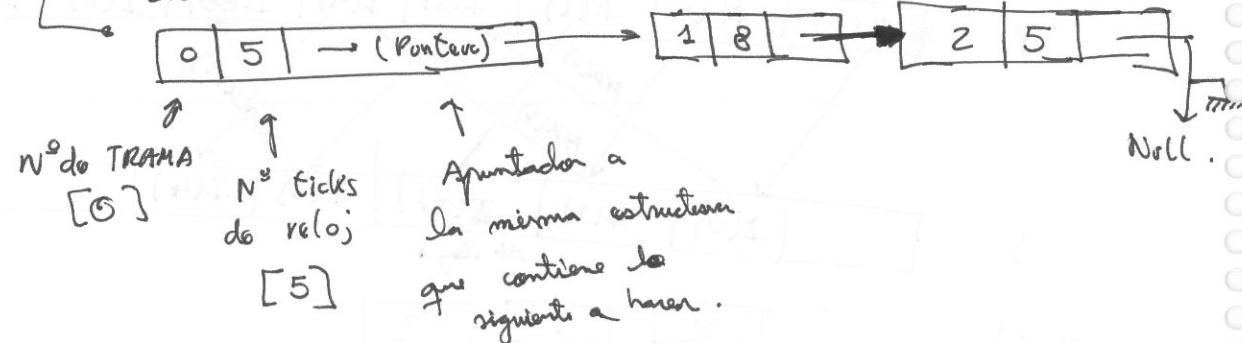
1 temporizador (hardware), controlado por Software.

↳ LISTA ENLAZADA.

- 100 ms = Reloj sistema

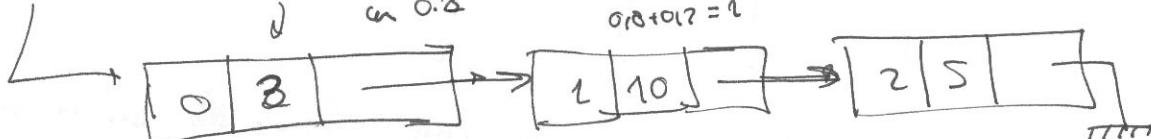


↳ LISTA ENLAZADA



Ha llegado  
↓ en 0.8

$$\begin{aligned} &0+2=N \\ &0,8+0,7=1 \end{aligned}$$



REPETICIÓN SELECTIVA

- ↳ Se retransmiten las tramas dañadas o aquellas que faltan en secuencia.
- ↳ 2 formas de implementación.
  - Retransmisión implícita: ha llegado bien o mal. El Primario supondrá que ha habido un problema porque le faltaron ACK.
  - Solicitud explícita: el secundario pedirá que le retransmitan concretamente una trama.

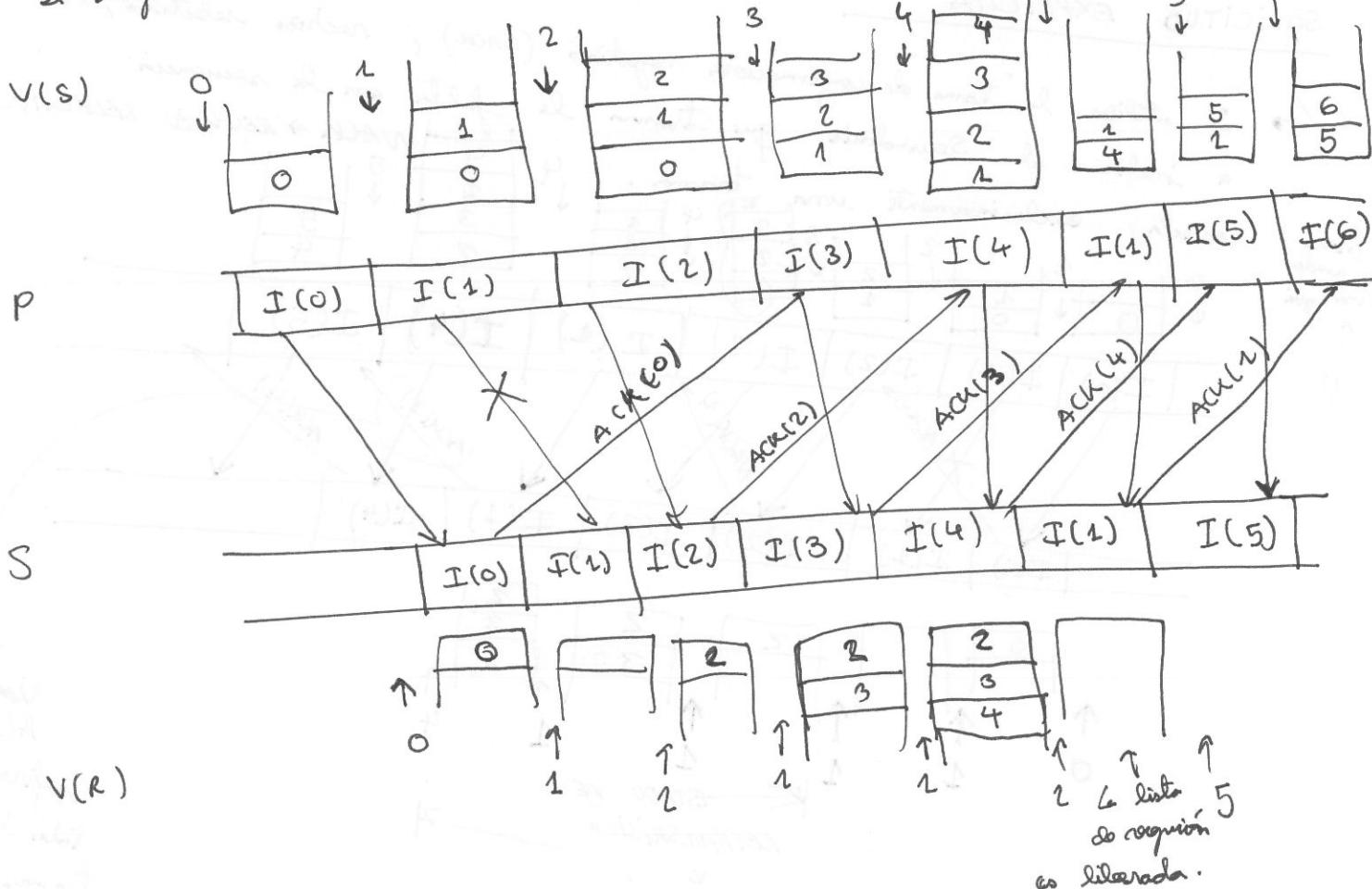
REPETICIÓN      SELECTIVA

- 5 NOMBRE DEL CRC 16.

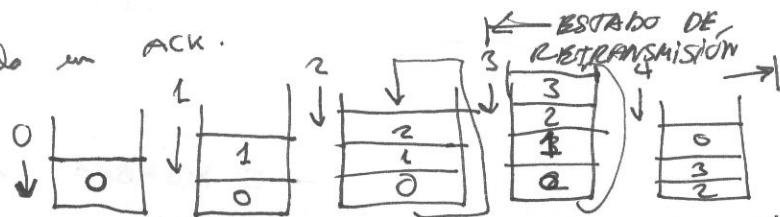
- ↳ Retransmisión implícita
- ↳ Solicitud explícita

RETRANSMISIÓN IMPLÍCITA

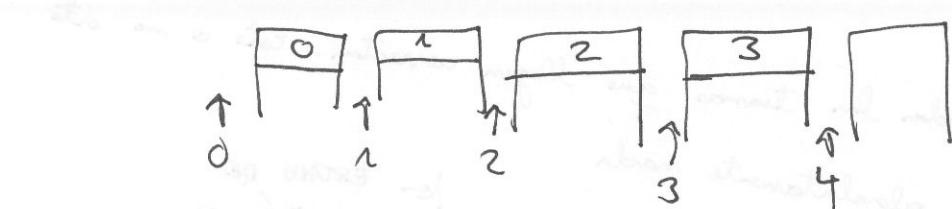
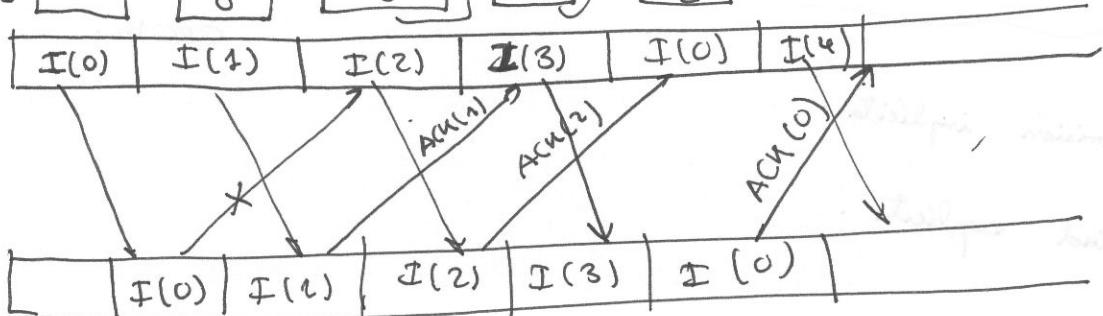
- El secundario confirma todos los tramas que llegan correctas. Este es un arte en secuencia.
- Si llega dañada no hace absolutamente nada.



Si falso en ACK.

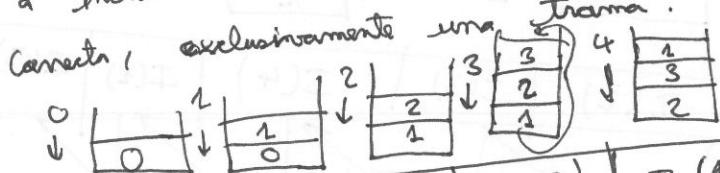


P

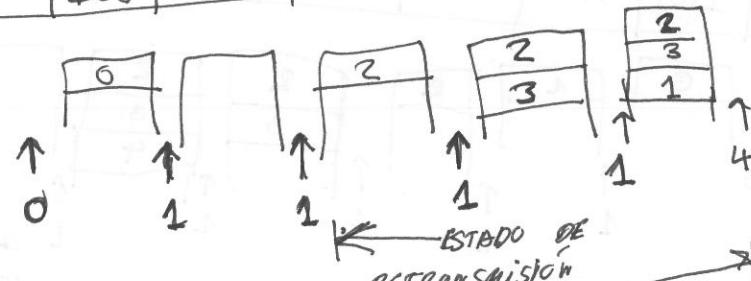
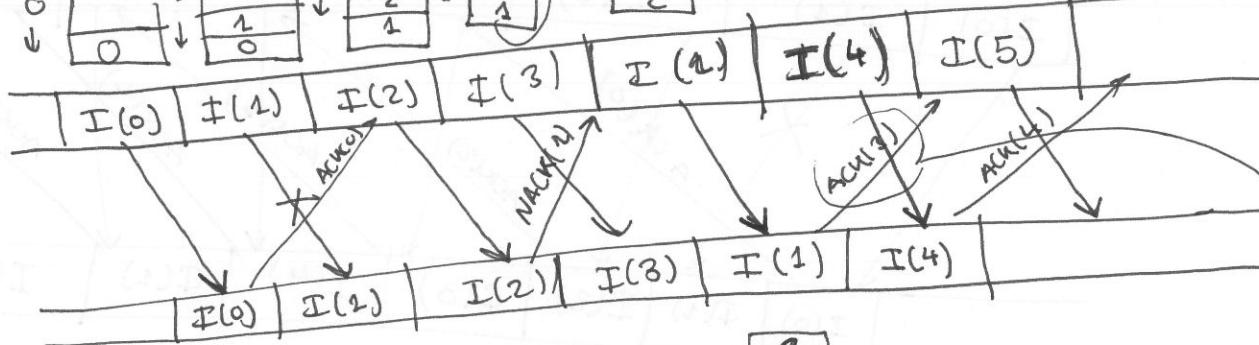


### SOLICITUD EXPLÍCITA

Se utiliza la trama de confirmación negativa (NACK), rechazo selectivo, para indicar al Secundario que trama le falta en la secuencia. NACK → RECHAZO SELECTIVO.



Correcto, exclusivamente una trama.



ESTADO DE RETRANSMISIÓN

Régimen de enviar confirmaciones

NACK / Secundario deja de enviar tramas hasta que reciba la que expresa su retransmisión.

Una ACK(n)

confirma todas las tramas anteriores

PIGGY BACKING (SUPERPOSICIÓN DE A SENTIMIENTOS)

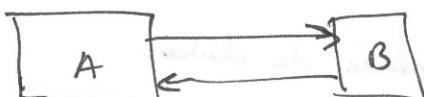


→ Datos en 2 sentidos  
Traman en 2 sentidos



→ Datos en 2 sentidos.  
Traman en 2 sentidos

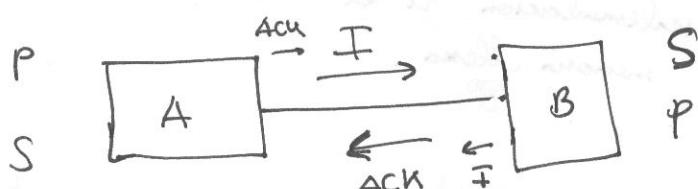
Opción 2



→ 2 circuitos físicos cada uno de ellos para transmisión de datos unidireccional.

Canal expedidor (envía)  
Canal receptor

Opción 2



→ Necesitar en el canal tránsito de información (I) y ACK (control).

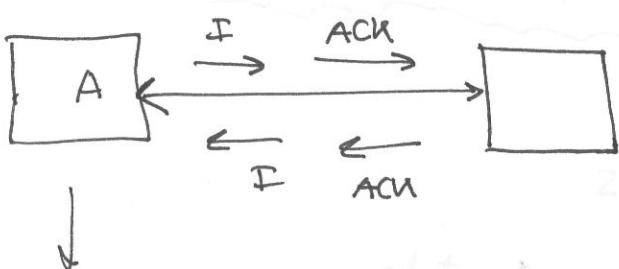
→ 1 circuito de datos. Trasmisión de datos bidireccional.



I o ACK.

## OPCIÓN 3

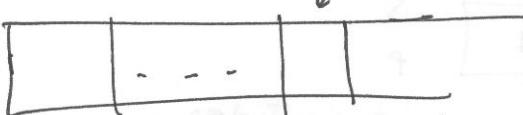
(ACK=ACK, NACK).



→ Aprovechar una trama de información.

A envía I y ACK al mismo tiempo. Para ello en la trama de I, superpongo ACK.

ACK → Envío ACK & I en la misma trama.



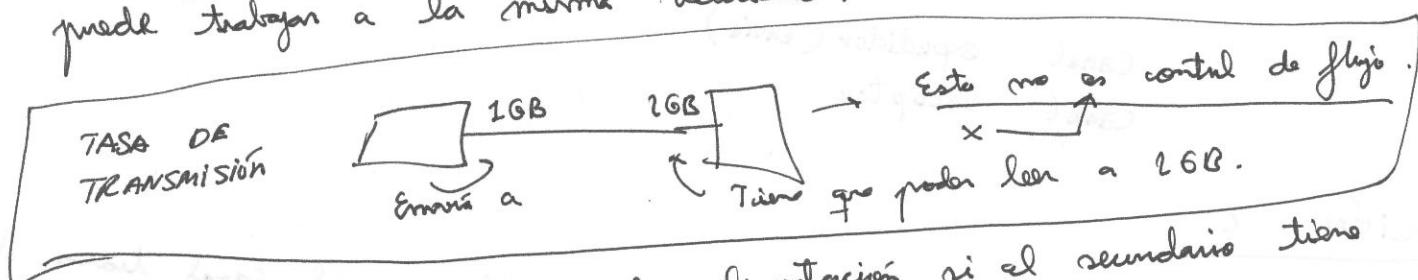
- Uso eficiente del ancho de banda.

- Retener el ACK mucho tiempo puede implicar que salte el timeout.

## CONTROL DE FLUJO

↳ Mecanismo que forma parte del control de enlace de datos.

↳ Evita que un transmisor muy rápido envíe a un receptor que no puede trabajar a la misma velocidad.



- Implementar un mecanismo de realimentación si el secundario tiene recursos necesarios o tiene la memoria llena.

X-ON

↓  
Estoy disponible.

Vuelve a enviarme datos.

X-OFF

↓  
Canción de control especial.

Sobrecarga; digo de enviarme porque no puedo procesar.

VENTANA DESLIZANTE

$\rightarrow$  N° de tramas

- Definir un número máximo de tramas  $K$  que el Primario puede almacenar en la lista de retransmisión.

$K$ : Tamaño de Ventana.

Si el Primario tiene pendiente de confirmación  $K$  tramas, se bloqueará.

Ese número

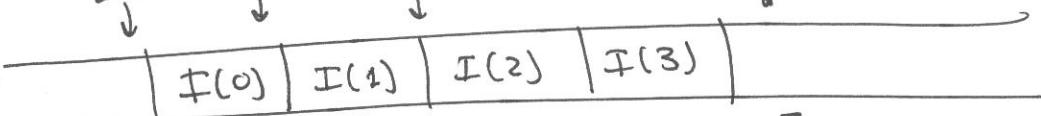
- Ventana de transmisión:  $\downarrow$  de tramas pendientes de confirmación.

de recepción:  $\downarrow$  los números de secuencia que al secundario está autorizado a aceptar

UWE  $\rightarrow$  Upper Window Edge  
 LWE  $\rightarrow$  Lower Window Edge

Primario  $\rightarrow$  Bando superior de la ventana  
 - UWE + 1 (Reíllo I)  $\downarrow$  (CF)  
 - LWE + 1 (Reíllo ACK)  $\downarrow$  (GWE + 1) (ACK)  
 $\rightarrow$  Bando inferior de la ventana.

$\rightarrow$  Si  $UWE - LWE = K \rightarrow$  Tamaño de memoria máxima alcanzada.  
 $\downarrow$  Si esto ocurre el P se bloquea.



Basta que no tengo la confirmación que ha recibido me envíe más.

$UWE(1+, \text{ Información})$

$LWE(1+, \text{ ACK})$

- Si trama I  $\Rightarrow UWE + L$

- Si ACK  $\Rightarrow LWE + ^\wedge$

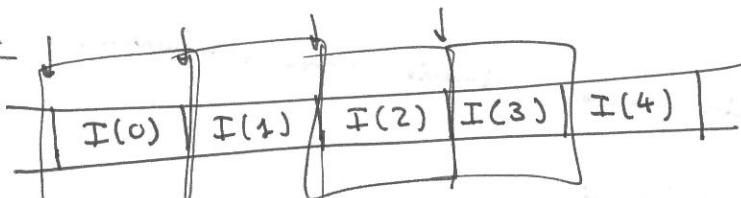
Si  $UWE - LWE$

$< K \Rightarrow$  Acepta paquetes de CR. (Capa de Red)  
 $= K \Rightarrow$  Bloquea C.R.

Rx

- Si I(n)
  - cae dentro de ventana del Rx  $\Rightarrow$  Acepta
  - cae fuera de ventana al Rx  $\Rightarrow$  Rechaza.
- Si  $n = V(R)$   $\Rightarrow$  Desplaza ventana de Rx.

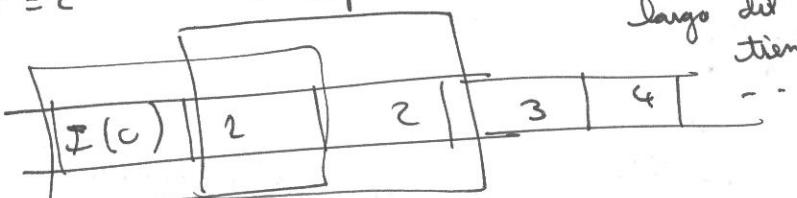
Tx



Ventana tamaño 3. La ventana va desplazándose.

Rx

$K' = 2 \rightarrow$  Es dobleza.



Ventana tamaño 2

→ Ventana fija que se desplaza a lo largo del conjunto de tramas que tiene que recibir.

## NÚMEROS EN SECUENCIA

→ Pueden identificar de forma única una trama.

V(S)

K : Tamaño de ventana de transmisión (del transmisor).

↳ Espacio de memoria máximo para almacenar tramas.

↳ Limitan el número de identificadores necesarios para identificar de manera única las tramas.

control de retransmisión. (RQ continua, inactiva)

nº de identificadores depende de

- Del organismo de retransmisión.
- Tamaño de ventana.

## RQ INACTIVA

↳ Trama la que tiene un reconocimiento o un duplicado.

Si necesitan 2 identificadores.

Número mínimo de identificadores = 2. → Mínimo 1 bit.

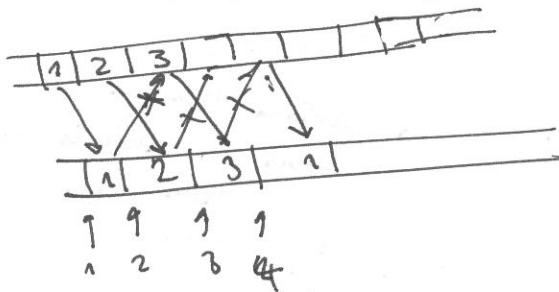
## Retroceder N

↳ Puede enviar hasta K tramas sin que el primario llegase a saturarse.

K TRAMAS

→ (K+1) IDENTIFICADORES.

↳ Si no se traganía duplicados.



## Repetición Selectiva

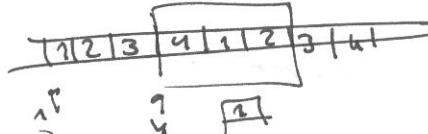
TX 1 1 2 3 4 1 1 2 3 4 1

K Tramas

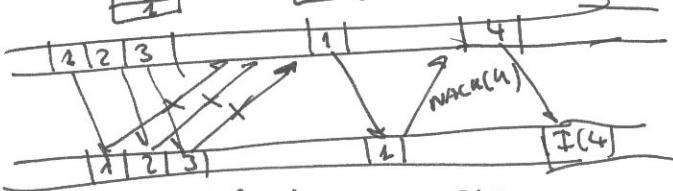
↓(K+1) Identificadores

1
3
2

1x



P



S

→ El número mínimo de identificadores es 2K.

Esquina	Ventana Transmisión	Ventana Recopilación	Nº mínimo Identificador
RQ INACTIVA	1	1	2
RQ CONTINUA	K	1	$K+1$
Repetición Selectiva	K	K	$2K$

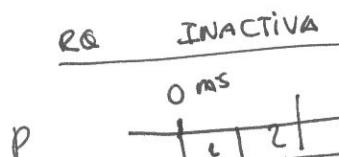
El secundario  
trama correcta  
y en memoria

### GRADO DE UTILIZACIÓN DEL ENLACE

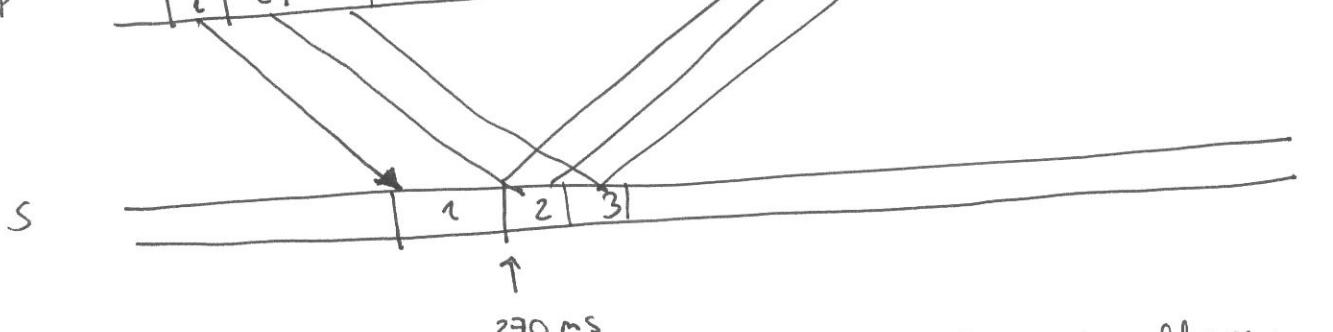
$$50 \text{ Kbps} \quad \left. \right\} T_{tx} = 20 \text{ ms}$$

Trama 1000 bits

$$T_p = 250 \text{ ms}$$

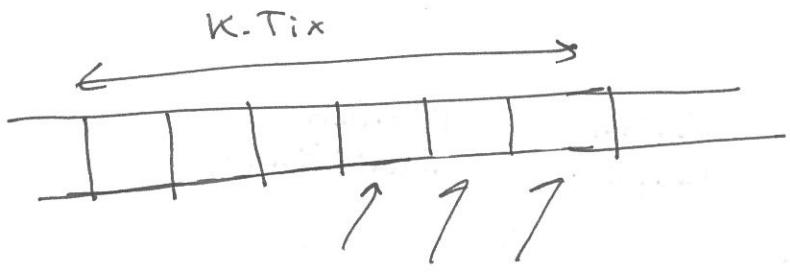


(+250 ms)  
520 ms



$$\frac{520 \text{ ms}}{20 \text{ ms}} = 26 \text{ Tramas}$$

→ Primario se retira y se bloquera.  
Pero justo llega el ACK y resuelve a  
enviar y por tanto no se bloquera.



→ Si no hay errores nunca se bloqueará.

$$T_T = T_p + T_{ix} + T_p$$

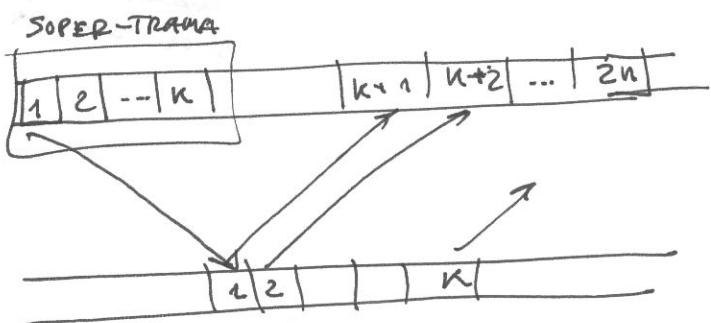
Si  $T_T \leq K \cdot T_{ix}$

→ Tiempo que tarda el ACK en llegar

$$\alpha = 100\%$$

↳ Grado de aprovechamiento del ancho.

$$K \geq \frac{T_{ix} + 2T_p}{T_{ix}} \Rightarrow K \geq 1 + 2 \cdot \frac{T_p}{T_{ix}}$$



$$K \geq 1 + 2\alpha$$

Nº de tramas que caben  
en ~~el espacio~~ ≤ que al  
número de tramas a enviar.

$$T_{ix}^S \rightarrow S = \text{SUPER-TRAMA}$$

$$\rightarrow T_{ix}^S = K \cdot T_{ix}$$

$$\alpha = \frac{\frac{S}{T_{ix}}}{T_T} = \frac{K \cdot T_{ix}}{2T_p + T_{ix}} = \frac{K}{1 + 2\alpha}$$

EXAMEN XARXES DE COMUNICACIÓNS ENERO 2016 – SOLUCIONADO

**1) El modelo OSI de la ISO es:**

- a) Una arquitectura de red.
- b) Un marco de referencia para la definición de arquitecturas de red.
- c) Un subsistema de comunicaciones monolítico.
- d) Ninguna de las anteriores.

**2) Las entidades de protocolo de la capa N:**

- a) Entablan comunicación mediante cierto protocolo con su capa proveedora.
- b) Entablan comunicación mediante cierto protocolo con su capa usuaria.
- c) Entablan comunicación mediante cierto protocolo con su capa par.
- d) Ninguna de las anteriores.

**3) Si se implementa un esquema RQ continua con repetición selectiva, el modo de funcionamiento de la capa de enlace tendría que ser:**

- a) Modo de funcionamiento datagrama.
- b) Modo de funcionamiento datagrama confirmado.
- c) El modo de funcionamiento de la capa de enlace es independiente del mecanismo de control de errores.
- d) Ninguna de las anteriores. (a i b falsas ya que modo funcionamiento datagrama → no orientado a conexión)

**4) ¿Qué ventaja aporta un modo de funcionamiento orientado a conexión frente a uno sin conexión con asentimiento?:**

- a) Asegura que las tramas llegan al destino.
- b) Las tramas llegan sin errores.
- c) Las tramas llegan en el orden correcto.
- d) Ninguna de las anteriores. (pueden llegar desordenadas aunq luego el secundario las ordene)

5) El SAP de capa N es un elemento que conceptualmente se encuentra en:

- a) La interfaz entre capa N y capa N+1, cuando la capa N actúa como usuaria.
- b) La interfaz entre capa N y capa N-1, cuando la capa N actúa como proveedora.
- c) No se encuentra en la interfaz.
- d) Ninguna de las anteriores. La interfaz entre capa N y capa N+1, cuando la capa N actúa como proveedora.

6) ¿Qué problema aparece en un modo de funcionamiento sin conexión con asentimiento frente a un modo sin conexión – sin asentimiento?

- a) La posibilidad de la llegada de tramas fuera de secuencia.
- b) La posibilidad de la llegada de tramas duplicadas.
- c) La posibilidad de no llegada (pérdidas) de una trama.
- d) Ninguna de las anteriores.

7) ¿Cuál de los siguientes métodos de entramado es el más eficiente (razón  $\frac{\text{bits datos}}{\text{bits transmitidos}}$  mayor)?:

- a) Caracteres de inicio y final con inserción de carácter.
- b) Banderas de inicio y final con inserción de bit.
- c) Violación de código en la capa física.
- d) b) y c) presentan la misma eficiencia.

8) Las variables de secuencia de transmisión y de recepción aparecen en:

- a) Modo de funcionamiento sin conexión.
- b) Modo de funcionamiento sin conexión con asentimiento.
- c) Modo de funcionamiento orientado a conexión.(misma razón que pregunta 3.)
- d) Las V(R) Y V(S) son independientes del modo de funcionamiento.

**9) Un servicio confirmado necesita de la utilización de:**

- a) CONNECT, DATA, DISCONNECT.
- b) CONECTA, DATA.
- c) DATA, DISCONNECT.
- d) Ninguna de las anteriores. (Necesita Response, confirm, request e indication)

**10) Si el campo de datos de una trama es 000001011110111111 y se utiliza la técnica de encapsulamiento “bandera de inicio y final con inserción de bit”, ¿Cómo quedaría para lograr la transparencia de los datos?**

- a) 00000110111110111111.
- b) 00000101111101111101.
- c) 000001011111001111101. (Después de 5 unos se pone un cero)
- d) 0000011011111001111101.

**11) Se quiere implementar un control de errores hacia adelante sobre un enlace en el que se presentan errores en ráfaga. ¿Cuál de los siguientes códigos de detección de errores utilizarías?**

- a) Paridad.
- b) BCC modificado.
- c) CRC-16 o CRC-32.
- d) Ninguna de las anteriores.(datos aleatorios que no tienes información para poder codificar) respuestas diferentes a lo que pide la pregunta.

**12) Para corregir d errores:**

- a) Se necesitan códigos con distancia de Hamming d.
- b) Se necesitan códigos con distancia de Hamming d+1.
- c) Se necesitan códigos con distancia de Hamming 2d.
- d) Ninguna de las anteriores. d (distancia 2d+1)

**13) ¿Por qué no se suelen utilizar los métodos de control de errores hacia adelante?**

- a) Porque hay muchos bits redundantes.
- b) Porque muchos de los enlaces son simplex.
- c) Porque utilizan códigos computacionalmente muy costosos.
- d) Porque son métodos de reciente aparición.

**14) Si se utiliza BCC como código detector de errores podemos decir que:**

- a) Asegura la detección de cualquier número par de errores.
- b) Asegura la detección de cualquier número impar de errores.
- c) La capacidad de detección de errores es como la del método de paridad.
- d) El tipo de paridad utilizado (par, impar) en las filas y en las columnas tiene que ser la misma.

**15) r bits redundantes → polinomio grado r**

**16) En el método de CRC, suponiendo G(X) el polinomio generador y E(X) el polinomio Error:**

- a) Si  $E(X) \bmod G(X)=0$  seguro que la trama ha llegado sin errores.
- b) Si  $E(X) \bmod G(X) \neq 0$  seguro que la trama ha llegado con errores.
- c) Si  $E(X) \bmod G(X) \neq 0$  no es seguro que la trama haya llegado sin errores.
- d) Ninguna de las anteriores.

**17) ¿A qué es debido que el receptor pueda descomprimir sin problemas un mensaje comprimido mediante codificación Huffman?**

- a) A que la transmisión es orientada a bit.
- b) A la propiedad del prefijo.
- c) A la entropía H del mensaje (Shannon).
- d) Ninguna de las anteriores.

**18) El control de errores con RQ inactiva necesita como mínimo de un enlace:**

- a) Símplex.
- b) Semidúplex.**
- c) Full dúplex.
- d) Funciona con cualquiera de los anteriores.

**19) El piggybacking es un mecanismo que permite:**

- a) Aprovechar las tramas de asentimiento (positivo o negativo) para enviar las tramas de Información.
- b) Aprovechar las tramas I (Información) para enviar información de asentimientos (positivos o negativos).**
- c) Reducir el índice de errores en la transmisión de tramas ACK.
- d) Reducir el número de asentimientos (positivos o negativos) del P (primario) al enviar al S (secundario).

**20) CRC-6 con muy baja probabilidad de errores:**

- a)  $x^5 + 1$
- b)  $x^5 + x^4 + 1$
- c)  $x^5 + x^4 + x^3 + 1 \rightarrow \text{Multiplo de } (x + 1)$
- d) Ninguna de las anteriores (Tiene que ser de grado 6)**

**21) RQ Inactiva:**

- a) Implícita solo da ACK cuando la trama llega correcta y en secuencia.
- b) Explícita solo da NACK cuando la trama llega mal o fuera de secuencia.
- c) La trama nunca puede llegar con errores o fuera de secuencia.
- d) Ninguna de las anteriores. (a y b son correctas para RQ continua y c es incorrecta)**

**22) Comprimir un archivo .exe en una LAN.**

- a) BCD modificado.
- b) Codificación relativa.
- c) Codificación Huffman.
- d) Ninguna de las anteriores.(datos aleatorios que no tienes información para poder codificar)

- 1) capa usuaria y proveedor diferencia
- 2) protocolo entidad par y comunicación real/virtual
- 3) modelo OSI, como está repartido (3 bloques, 7 capas)
- 4) RJ45 a qué capa pertenece → capa física
- 5) uso de servicio connect, data, disconnect
- 6) los servicios que dan conexión confirmada
- 7) y no confirmada
- 8) diferencia entre servicios sin conexión con o sin asentimiento
- 9) carácter inicial final e inserción carácter
- 10) código Hamming lo que diste tú [d: distancia de error]
- 11) propietat del prefix
- 12) grado eficiencia en los enlaces
- 13) Nr = número mitja d'intents de transmisión

No se necesitan mecanismos de  
transparencia de dato ya que  
disponemos de la longitud.

Nr: Número medio de  
intents de transmisión.



→ No hay contacto entre capa (N+2) y N.  
Capa usuaria (arriba) N+1  
Capa proveedor (abajo). N

1)

Con qué se comunica el protocolo de una entidad?

Con su par.

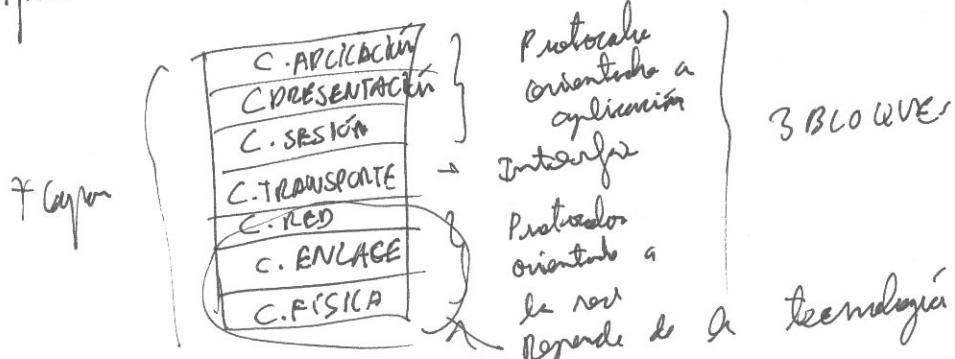
Círculo virtual: ruta predefinida que tendrán que seguir todos los paquetes.

El canal me está reservado, los paquetes solo están decidido el camino.

Los paquetes llegan en orden.

- Comunicación real en vertical. Capas adyacentes.
- Comunicación virtual. Capas iguales.

3) MOPFLIO OSI. No es una arquitectura de red ya que los protocolos no están definidos. Las capas si lo están.



c) Transporte (Copia de extremo a extremo)

↳ Evitar errores extremo a extremo.

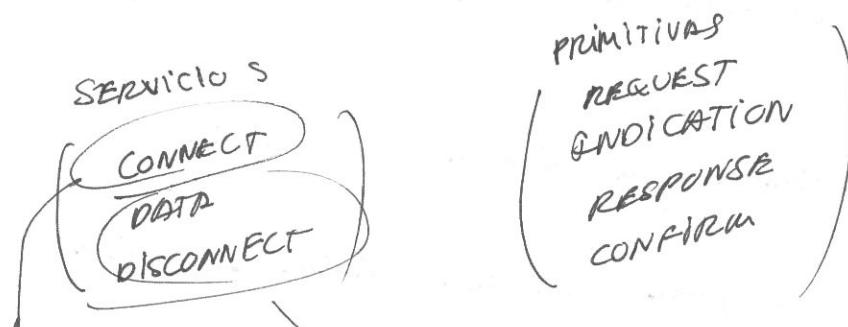
Testigo: control que ningún retraso de generación.

ii) RJ45 → Copia Física.

5) SERVICIO DATA, CONNECT, DISCONNECT.

Servicio no confirmado { - con asentimiento. (fiable)  
- sin asentimiento. (no fiable)

Confirmado (fiable)  
orientado a conexión (en orden)  
sin conexión (fuera de orden)



Confirmado      no confirmado

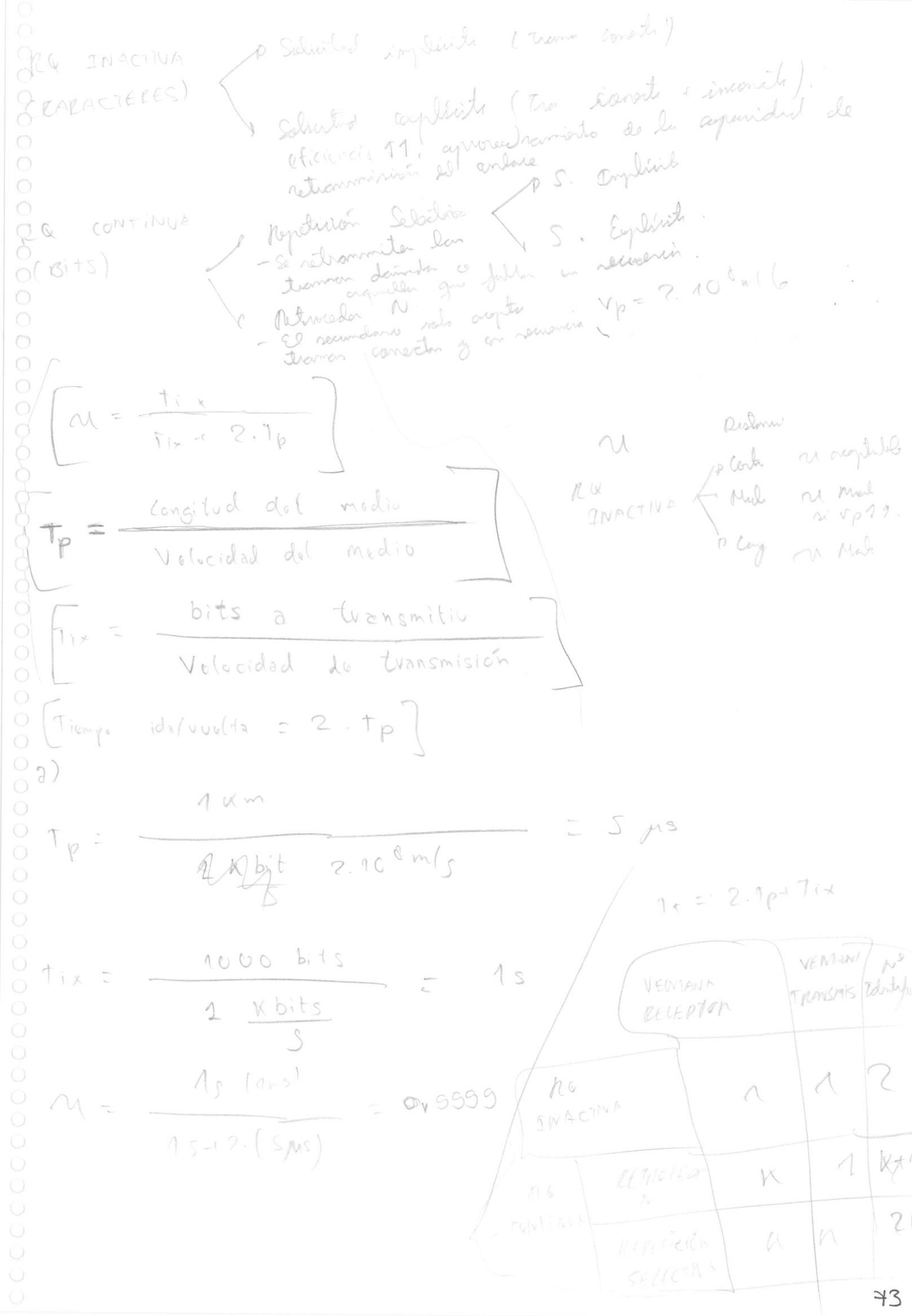
Ritmo & siguientes & ritmo correcto.

d) Propiedad del prefijo: al símbolo codificado jamás será el símbolo de la codificación de otro símbolo.

Nr = Número de intentos promedio que hacen que una trama sea recibida correctamente.

$$N_r = \frac{T_D}{T_{fix}}$$

→ En la CANTINVA → con relación explícita a ACK - confirm

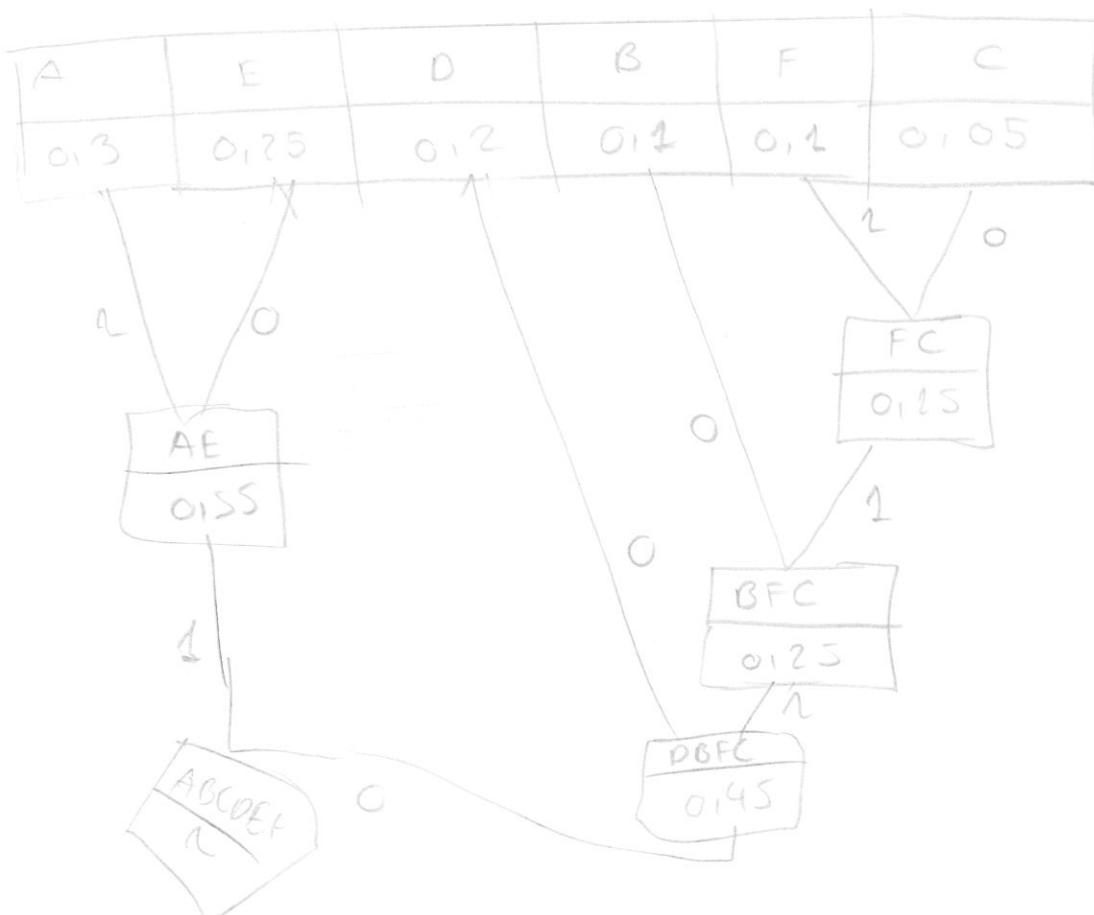




11840110110000      20021  
 10011111  
 01111111  
 - 100111  
 01100111

A 0,3  
 B 0,2  
 C 0,05  
 D 0,2  
 E 0,25  
 F 0,1

A 0,3  
 E 0,25  
 D 0,2  
 B 0,2  
 F 0,1  
 C 0,05



IX

$$x^v \cdot M(x)$$

$$\frac{x^v \cdot M(x)}{G(x)} \quad \begin{matrix} Q(x) \\ \swarrow \\ G(x) \end{matrix} \quad \rightarrow P(x)$$

$$x^v \cdot M(x) = R(x)$$

$$T(x) = x^v \cdot M(x) - R(x)$$

$v$  = grado del polinomio generado.

$$c_{n+1} \cdots c_6 = x^{16} + x^{15} + x^{12} + 1$$

$$\frac{x^{16}(x^{16} + x^{15} + x^{12} + 1)}{x^{16}}$$

$$\text{TRAMA} = \begin{array}{ccccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 5 & 6 & 4 & 6 & 5 & 9 & 3 & 2 & 1 \end{array}$$

$$M(x) = x^5 + x^4 + x^6 + x^9 + x^3 + x + 1$$

$$R(x) = x^9 + x + 1$$

$$x^v \cdot M(x) = x^9(x^5 + x^4 + x^6 + x^9 + x^3 + x + 1)$$

$$\begin{array}{r} 10110 \\ 011100 \\ \hline 1.11010 \end{array}$$

C. FÍSICA  $\Rightarrow$  Bits

C. FÍSICA  $\rightarrow$  TRAMAS

(SWITCH)

C. RDP  $\rightarrow$  PAQUETES (MUTEX)

$\rightarrow$  Un sensor o informado pide ver trama.  
 $\rightarrow$  Pone una marca de longitud y la respuesta es de acuerdo.

Control de errores = detección + corrección

Parity  $\geq (d+1)$

Parity & Checksum  $\geq (2d+1)$

Parity & Checksum  $\geq (2d+1)$

1000

tramas

$\frac{\text{bits}}{\text{trama}}$

Palabra código en cuenta bits difusos

o distorsión bit error

Control = palabra código

SWR  $\rightarrow$  respondiendo

1 2 3 4

?

1

$$0+0=0$$

$$1+0=1$$

$$0+1=1$$

$$1+1=0 \quad (\text{a})$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 11010101 \\ \times 10000101 \\ \hline 1100001010 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$$

$$000\ 0010$$

$$\begin{array}{r} 0-0=0 \\ 1-0=1 \\ 0-1=1 \\ 1-1=0 \end{array} \quad (\text{a})$$