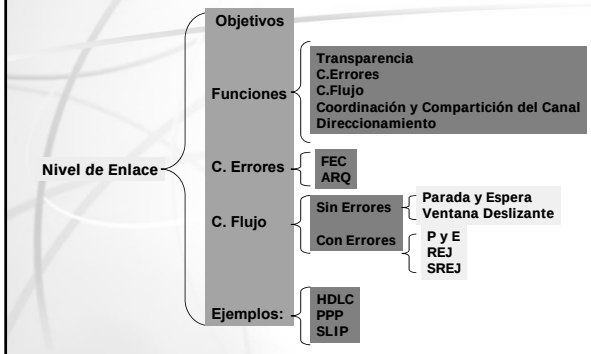
Nivel de Enlace

Prof. Dr. Jose Ignacio Moreno Novella

Departamento de INGENIERIA TELEMATICA



Funciones del Nivel de Enlace

- Objetivo: **Resolver los problemas derivados de la falta de fiabilidad de los circuitos físicos.**

  Transferencia fiable de bloques de información (tramas) entre equipos directamente conectados.
- Funciones Principales:
  - **Delimitación de trama**
  - **Transparencia**
  - **Coordinación y Compartición del canal**
  - **Control de flujo**
  - **Control de errores**
  - **Direccionamiento: LAN**

Definiciones

- **Mensaje: Secuencia de caracteres o bits que representa la información a enviar de un origen a un destino.**
- **Bloque: conjunto de caracteres o bits que se agrupan por razones técnicas para ser transmitidos como una unidad.**
- **Trama: estructura de datos que maneja el protocolo de nivel de enlace para enviar un bloque.**

Delimitación de Trama

- **Nivel Físico: Delimitación (sincronismo) de bit y de carácter (a veces)**
- **Delimitación de trama:**

  ¿Donde empieza/acaba una secuencia de datos?
- Soluciones:
  - **Utilización de tramas de tamaño fijo**
  - **Delimitación por carácter de longitud**
  - **Delimitación por carácter de principio y fin**
  - **Delimitación por guiones**

Delimitación de tramas

- **Tramas de tamaño fijo**
  - **Intrínsecamente transparente**
  - **Poco flexible. Rellenar tramas cortas (desperdicio del canal)**
- **Delimitación por longitud**
  - **Intrínsecamente transparente**

    Datos a enviar abcde

    Trama: 5abcde

    ¿Qué pasa si se produce un error en la información de longitud? Se pierde el sincronismo de todas las tramas hasta que se recupere ese error

Delimitación de tramas

- **Delimitación por carácter de principio y fin**
  - **Problemas de transparencia**
  - **Carácter de principio/fin:** $
  - **Datos a enviar:** abcdefghijk
  - **Trama:** $abcdefghijk$
  - **Datos a enviar:** abc$efg
  - **Trama:** abcefg**... ?????=> ver luego**
- **Delimitación con guiones en protocolos orientados a bit (HDLC)**

01111110 11010111011010011101001111110

Transparencia

Se dice que un protocolo es transparente si es capaz de enviar cualquier dato.

- **Delimitación por carácter de principio y fin**
  - **Los caracteres de control van precedidos por carácter especial (de escape)**
  - **El carácter de escape se duplica cuando aparece en los datos.**
    - ✓ **Carácter de ppio./fin: $**
    - ✓ **Carácter de escape: %**
    - ✓ **Datos a enviar ab%deg**
    - ✓ **Trama: %$ab%%$de$g%$**
    - ✓ **U otra forma: trama $ab%%%$d%%e%g**
  - **Eficiencia caso peor 50%.**

Transparencia

- **Delimitación con guiones en protocolos orientados a bit.**
  - **Se inserta un 0 por cada cinco 1 consecutivos en el campo de datos, independientemente del símbolo siguiente**

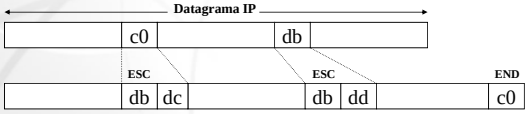
Datos a enviar: 00111111100111110

Trama: 01111110**00111110**1100111110**0**01111110

  - **En recepción se elimina siempre el 0 que sigue a cinco 1**
  - **Eficiencia en el caso peor: 5/6.**

Ejemplo Serial Link IP Protocol: SLIP

- **Definido para encapsular Datagramas IP sobre líneas serie (RFC1055).**
- **Muy difundido.**
- **Envía datagrama IP byte a byte añadiendo una marca de fin de Datagrama (0xc0).**



- **Usado principalmente para accesos a ISP**

Protocolos de Control de Errores: Técnicas FEC y ARQ

Departamento de INGENIERÍA TELEMÁTICA www.it.uc3m.es

Control de Errores

- **Conjunto de Técnicas que permiten resolver los problemas introducidos por los canales ruidosos con probabilidades de error inaceptables para las aplicaciones finales**
- **Fases**

1.- Detección de errores

Información redundante en cada trama

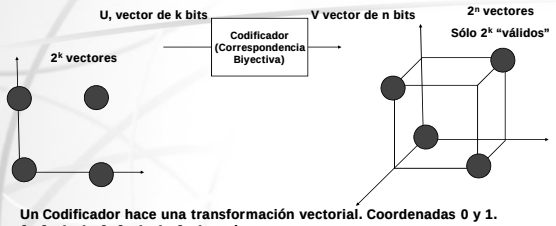
2.- Recuperación

  - **Corrección de errores en el destino (FEC) Información redundante en cada trama**
  - **Petición de retransmisión (ARQ)**

Detección/Corrección de Errores

- Método: añadir información redundante a los datos para detectar y, tal vez, corregir errores
- Se coge un bloque de k bits de datos.
- Ese bloque se pasa por un codificador y sale un bloque de n bits con n>k
- =>Códigos de bloque
- Si la salida del codificador contiene la entrada =>Códigos lineales
- Veremos sólo códigos de bloque lineales

Códigos lineales de bloque: geometría



Un Codificador hace una transformación vectorial. Coordenadas 0 y 1. 0+0=1+1=0, 0+1=1+0=1, no hay acarreo

Ejemplo: codificador paridad par al final, K=2, n=3, m=n-k=1

El espacio de las u está totalmente lleno. El espacio de las v está parcialmente lleno (redundancia), esto es lo que da capacidades de detección y, tal vez, de corrección al código

$v_{1*n}=u_{1*k}*G_{k*n}$ (transformación vectorial=transformación matricial)

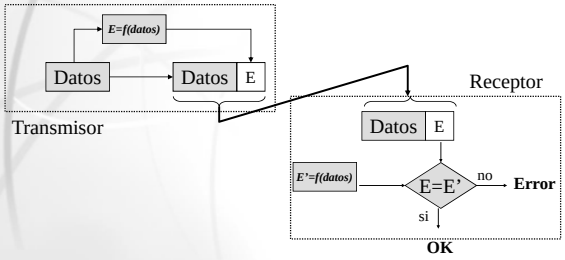
Además $G_{k*n}*H^t_{n*m}=0$ y $v*H^t=0$

Distancia mínima y propiedades de detección corrección

- Vi≠j, d(Vi,Vj) = nº be bits distintos entre Vi,Vj
- dmin=min(d(Vi,Vj)) Vi≠j
- C+D+1≤dmin con C≤D
- Detecto SIEMPRE D o menos errores
- Corrijo SIEMPRE C o menos errores
- Ejemplo k=1, n=4 m=3, cuadriplicar el bit
- dmin=d(1111,0000)=4
- Posibilidades (C=1, D=2) o (C=0,D=3)
- Llega 1000
  - Si C=1 D=2 detecto error y corrijo en 0000 (¿qué pasa si se txió 1111?)
  - Si C=0 D=3 detecto y pido rtx (ARQ)
- Llega 1010
  - Si (C=1, D=2) o (c=0,D=3) sólo puede detectar y pedir rtx (ARQ)

Códigos sistemáticos

Si el código de bloque lineal tiene todos los bits del mensaje al principio y los bits de redundancia todos al final es un código de bloque lineal y sistemático



Ejemplo de códigos de detección

- Paridad:

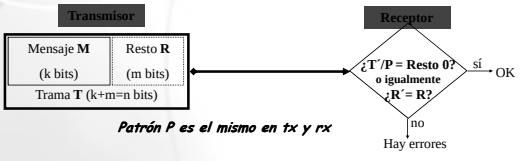
  Añadir un bit a una secuencia de datos indicando si el número de "0s" o "1s" es par o no

  - Dos tipos:
    - ✓ Impar: 1: numero par de "1s"
      0: numero impar de "1s"
      ¡¡Ojo viola la regla de la suma, no es lineal!!
    - ✓ Par: 1: numero impar de "1s"
      0: numero par de "1s"

  Sólo detecta errores impares

Ejemplos de códigos de detección: CRC

- CRC *Cyclic Redundancy Check*
  - El transmisor, dado un mensaje M (de k bits), genera un código R (o frame check sequence FCS, de m bits) con un algoritmo que usa un patrón o divisor P (de m+1 bits) y transmite la trama T (de k+m=n bits, concatenando M y R)
  - El receptor, con los k primeros bits (M´) de los recibidos (T´), genera R´ (m bits) utilizando el mismo algoritmo y patrón P : si R´=R ó, equivalentemente, R{T/P}=0: no hay errores (o no se pueden detectar)



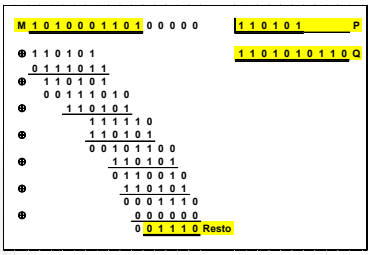
Patrón P es el mismo en tx y rx

CRC

- 4 formas de verlo
  - v=u*G, códigos cíclicos
  - Aritmética módulo 2
    (XOR: a⊕a=0, a⊕b=1 // 2^m = desplazar a la izqda y rellenar con 0s)
    En txt: R = Resto (2^m*(M=u) / P) ; T=v= 2^m*M ⊕ R
    En rx sin error: T´/P=T/P= (2^m*M ⊕ R)/P = Q⊕(R/P)⊕(R/P) = Q
    =>resto=0
    En rx con error: T´/P= (T ⊕ *Error*) / P = Q ⊕ *Error*/P.
    =>resto=R(*Error*/P) Si *Error* múltiplo de P, no se detecta
    (Significado de *Error:* 1 en el bit erróneo, 0 en el OK)
    Evidentemente Error es desconocido
  - División de Polinomios
    Misma idea, usando polinomios de grado m-1. x variable muda
    "m" bits 110101 = $1*X^5 + 1*X^4 + 0*X^3 + 1*X^2 + 1*X^1 + 1$
  - Lógica digital
    Este algoritmo debe ejecutarse de una forma rápida para todos los mensajes que se intercambien, preferentemente implementada en hardware ... Relacionado con V=u*G

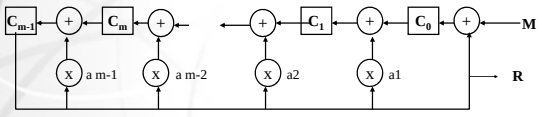
CRC

- Ejemplo
  M = 10 10 00 11 01
  P = 11 01 01
  P=x^5+x^4+x^2+1

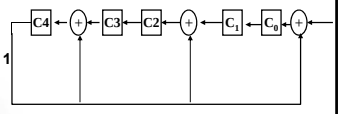


CRC con lógica digital

- Algoritmo implementable mediante lógica digital
  1.- Registros de desplazamiento para "m" bits, inicializados a "0" Ci
  2.- "m" puertas "or-exclusiva" XOR ⊕
  3.- Presencia o ausencia de puerta dependiendo de la presencia o ausencia del término en polinomio divisor P



- Ejemplo:
  Polinomio $P=X^5 + X^4 + X^2 + 1$
  Mensaje 1010001101

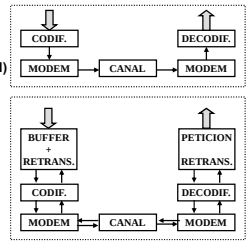


CRC

- Polinomios de patrón P habituales:
  - CRC-16 $x^{16}+x^{15}+x^2+1$ (P de 17 bits, R de 16 bits)
  - CRC-CCITT $x^{16}+x^{12}+x^5+1$ (P de 17 bits, R de 16 bits)
  - CRC-32 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$ (P de 33 bits, R de 32 bits)
  - CRC-12 $x^{12}+x^{11}+x^3+x^2+x+1$ (P de 13 bits, R de 12 bits)
- ¿Qué tipos de error se detectarán?
  - Todos los errores de un único bit
  - Todos los errores dobles, si P tiene al menos tres 1
  - Cualquier número impar de errores, siempre que P(X) contenga el factor (X+1)
  - Cualquier ráfaga de errores cuya longitud sea *menor* que la longitud de P, i.e. menor o igual que la longitud de FCS
  - La mayoría de las ráfagas de mayor longitud

Corrección de Errores

- Objetivo:
  Recuperación frente a errores detectados
- Dos Enfoques
  - Técnicas FEC
    Perror=P(no corregir o corregir mal)
    Se corrige aprovechando la redundancia
  - Técnicas ARQ
    Perror=P(no detectar)
    Sólo se detectan los errores y se pide retransmisión.
    Se estudiará luego



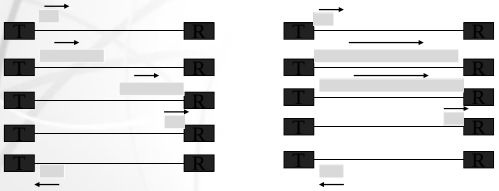
Técnicas de Control de Flujo

Control de Flujo

- Objetivo:
  limitar la cantidad de información que el transmisor puede enviar al receptor, al objeto de no saturar los recursos (memoria,..) disponibles.
- **Suposiciones**
  - **Ausencia de errores**
  - **Recepción ordenada**
- **Técnicas de Control**
  - **Parada y espera**
  - **Ventana Deslizante**

Parada y Espera

- **Fuente envía una trama y espera confirmación.**
- **Receptor envía confirmación.**

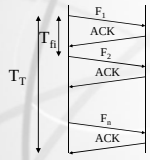


- **PROBLEMA: Sólo una trama en tránsito**
- **Eficiencia= f(tamaño trama)**

$\begin{cases} t_{tx} > t_{pro} & \text{Ineficiente} \\ t_{tx} < t_{pro} & \textbf{Muy Ineficiente} \end{cases}$

Análisis de Prestaciones

- **Suponemos: línea semiduplex sin errores. Tramas igual tamaño**



$T_{Fi}=T_{prop} + T_{tx} + T_{proc} + T_{prop} + T_{ack} + T_{proc}$

Suponemos: T_{ack} y T_{proc} despreciables

$T_{Fi}=T_{tx} + 2T_{prop}$ $\quad T_T=n\,[T_{tx} + 2T_{prop}]$

$$U = \frac{n*T_{tx}}{T_T} = \frac{T_{tx}}{T_{tx}+2T_{prop}} = \frac{1}{1+2a}; \quad a = \frac{T_{prop}}{T_{tx}}$$

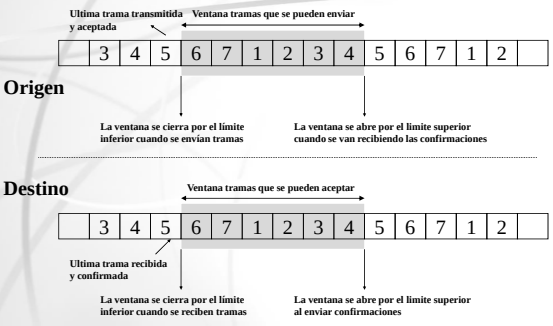
$$T_{prop} = \frac{Distancia}{Velocidad} \quad \frac{Enlace}{Medio}$$

$$T_{tx} = \frac{Longitud\,(bits)}{Velocidad} \quad \frac{Trama}{Enlace}$$

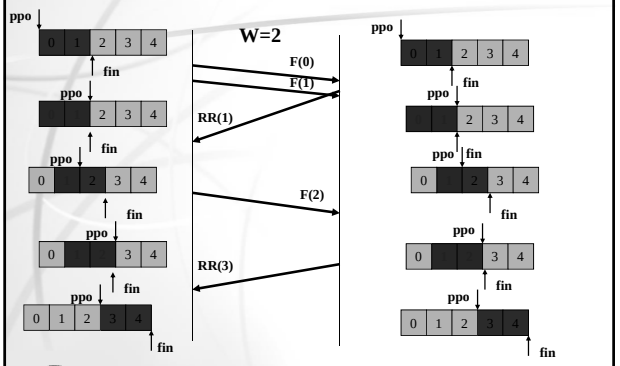
Ventana Deslizante

- **Permite el envío simultáneo de varias tramas en tránsito.**
- **El destino reserva n buffers para recepción de tramas.**
- **El origen puede enviar n tramas sin esperar confirmación.**
- **Las tramas deben numerarse mediante el uso de un campo de longitud finita (n) en la información de control.**
- **El tamaño máximo de la ventana es 2^n-1**

Ventana Deslizante



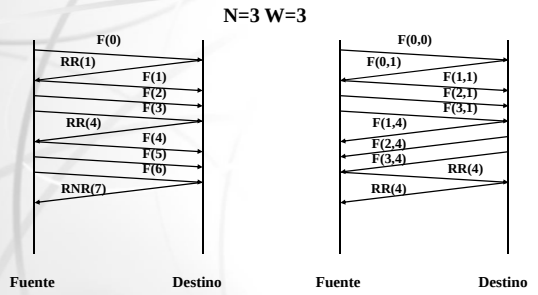
Ejemplo de Ventana Deslizante



Ventana Deslizante

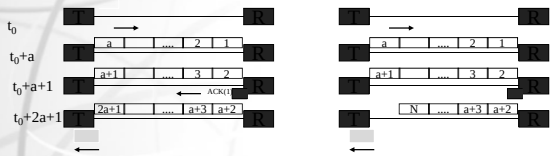
- **Tipos de Tramas (Notación)**
  - **Datos:** F_1, ...F_n
  - **ACK:** RR(n) **(He recibido hasta la trama n-1 espero recibir la n)**
  - **RNR** (n) **(Recibido correctamente hasta n-1 no soy capaz de recibir más temporalmente)**
- **Si enlace duplex, tramas de datos incorporan campos de asentimiento**
  - **Datos:** F(m,n) **(Envío trama m, asiento n-1).**

Ventana Deslizante



Análisis de Prestaciones

- **Suponemos T_{tx}= 1 T_{prop}=a**



$$U \begin{cases} 1 & si \quad N > 2a+1 \\ \dfrac{N}{2a+1} & si \quad N < 2a+1 \end{cases}$$

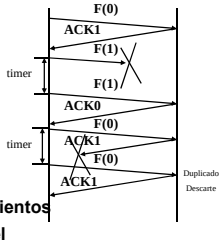
ARQ

ARQ

- **Petición de retransmisión en caso de errores. Sirve también para hacer control de flujo.**
- **Parada y Espera**
- **Ventana Deslizante con rechazo simple**
- **Ventana Deslizante con rechazo selectivo**

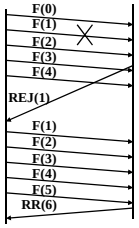
ARQ: Parada y Espera

- **Dos tipos de error:**
  - **Trama dañada o perdida**
    - ✓ **Detección y descarte de trama**
    - ✓ **Temporizador en fuente para retransmisión**
  - **ACK dañado o perdida**
    - ✓ **Numeración de tramas y asentimientos (0,1), para evitar duplicados en el receptor.**



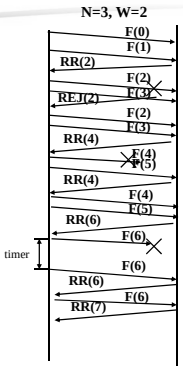
ARQ: Rechazo Simple

- Si el destino detecta error envía trama REJ(n). La estación destino descartará esa trama y las siguientes hasta recibir de nuevo la trama correctamente.
- El origen al recibir el REJ(n) retransmite la trama y las posteriores.

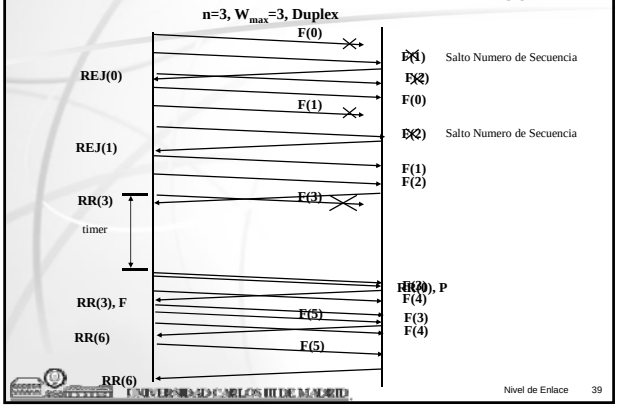


ARQ: Rechazo simple

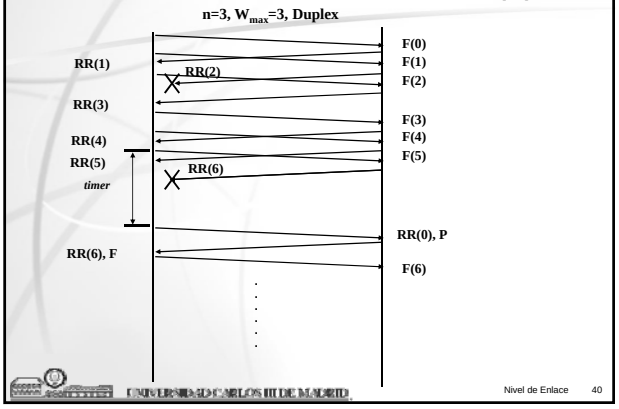
- Tipos de errores:
  - Trama dañada o perdida.
    - ✓ Si es detectada por receptor REJ
    - ✓ Si se pierde, alteración número de secuencia, REJ
    - ✓ Si se pierde y es la última, temporizador e interrogación de estado.
  - RR dañado o perdido
    - ✓ Si se recibe RR posterior no hay problema
    - ✓ Si no, tx solicita estado (RR).
  - REJ dañado o perdido
    - ✓ Interrogación de estado.



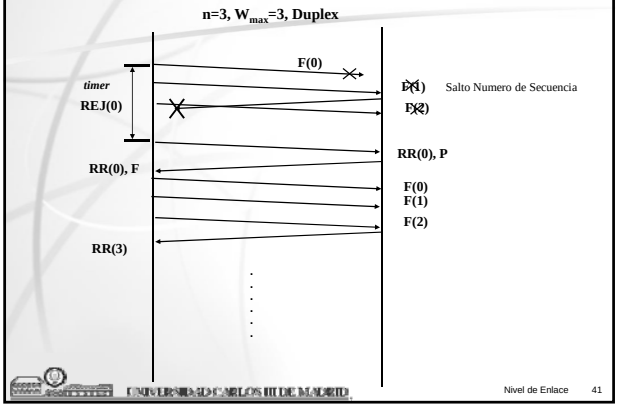
REJ: Detección de Errores (I)



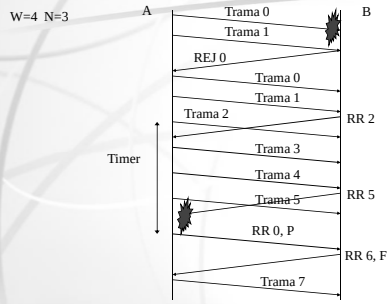
REJ: Detección de Errores (II)



REJ: Detección de Errores (III)

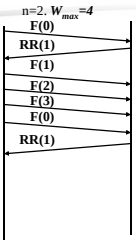


ARQ: rechazo simple



REJ: Tamaño Máximo de Ventana

- $W_{max}=2^n-1$
- Supogamos $W_{max}=2^n$



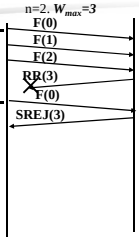
- ¿¿ Qué ha pasado ??
  - Se ha recibido todo bien ??
  - Se ha recibido todo (4 últimas tramas) mal ??
  - Ambiguo → Solución $W_{max}=2^n-1$

ARQ: Rechazo selectivo

- Rechazo selectivo de tramas dañadas. SREJ(n). Igual que REJ pero solo se retransmiten tramas dañadas.
- Mas memoria que rechazo simple.
- Lógica de reinserción y envío selectivo.
- Tamaño máximo de ventana 2^{n-1}

SREJ: Tamaño Máximo de Ventana

- $W_{max}=2^{n-1}$
- Supogamos $W_{max}=2^n-1$



- Receptor supone F(3) perdida, acepta F(0), envía SREJ(3)
- Transmisor supone que se perdieron todas las tramas
  - Ambiguo → Solución $W_{max}=2^{n-1}$

Probabilidad de error de bloque

- Probabilidad de error de bloque

$P_{eb}= 1- (1-p)^n$ con p=prob. error de bit
n=nº de bit/bloque

ARQ: Análisis de Prestaciones

- Parada y Espera
  - Los errores provocan retransmisiones

$$U = \frac{T_{tx}}{T_{total}} = \frac{T_{tx}}{N_t * (T_{tx} + 2T_{prop})} = \frac{1}{N_t(1+2a)}$$

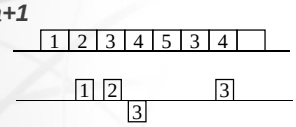
N_t	Prob
1	$1-P_{err}$
2	$P_{err}(1-P_{err})$
.....	.....
n	$P_{err}^{n-1}(1-P_{err})$

$$N_t = \sum_{i=1}^{\infty} i(1-P_{err})P_{err}^{i-1} = \frac{1}{1-P_{err}}$$

$$\boxed{U = \frac{1-P_{err}}{1+2a}}$$

ARQ: Análisis de Prestaciones

- Rechazo simple con envío continuo *N > 2a+1*



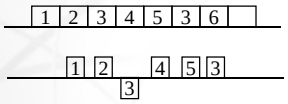
$$U = \frac{T_{tx}}{T_{total}} = \frac{T_{tx}}{N_r * (T_{tx} + 2T_{prop}) + T_{tx}} = \frac{1}{N_r(1+2a)+1}$$

$$N_r = N_t - 1 = \frac{P_{err}}{1-P_{err}}$$

$$\boxed{U = \frac{1-P_{err}}{1+2aP_{err}}}$$

ARQ: Análisis de Prestaciones

- Rechazo selectivo con envío continuo *N > 2a+1*



$$U = \frac{T_{tx}}{T_{total}} = \frac{T_{tx}}{N_t * T_{tx}} = 1 - P_{err}$$

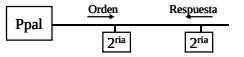
Ejemplos: Protocolo HDLC

Protocolos Orientados a Bit

- Operación independiente del código. No hay códigos de control.
- Adaptabilidad a varias configuraciones
  - 2,4 hilos,
  - punto a punto, multipunto
- Alto rendimiento (Datos/control)
- Alta seguridad. Tramas protegidas con mecanismos de control de errores.

HDLC: High Level Data Link Control

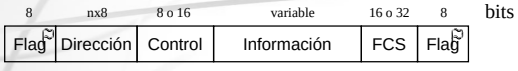
- Características Básicas
  - Protocolo de nivel de enlace orientado a bit
  - Define tres tipos de estaciones
    - ✓ primaria
    - ✓ secundaria
    - ✓ combinada
  - Dos configuraciones de enlace
    - ✓ no balanceada (primaria +nsecundarias)
    - ✓ balanceada (2 combinadas)
  - tres modos de operación
    - ✓ Respuesta normal (NRM)
    - ✓ Balanceado asíncrono (ABM)
    - ✓ respuesta asíncrono (ARM)



No Balanceada

Balanceada

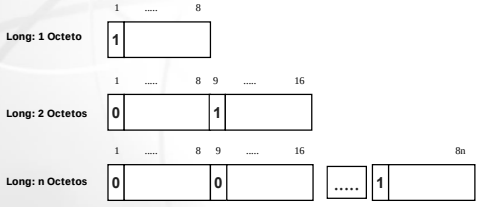
HDLC: Formato de Trama



- Tramas con formato único
- Flag (1 octeto): `01111110`
  transparencia mediante bit-stuffing
- Dirección variable origen o destino
- Control:Determina el tipo de trama
- CRC (2 o 4 octetos), utilizando CRC-CCITT o CRC-32

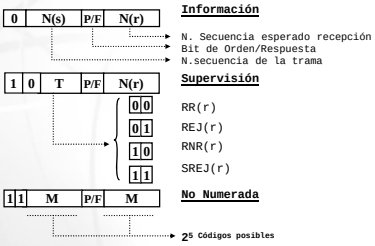
Campo Dirección

- Identifica a la estación secundaria que ha transmitido o que va a recibir la información.
- No necesario en enlaces punto a punto.
- Formato normal (8 bits) o ampliado (variable).



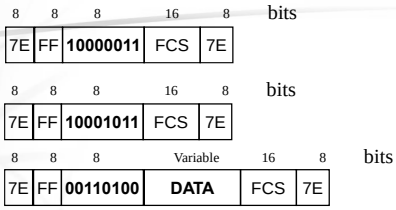
Campo de Control

- Longitud de 8 bits salvo negociación de numeración extendido.
- Tres tipos de tramas

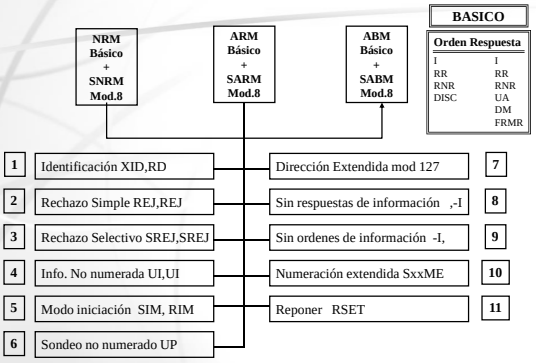


Ejemplos

- RR(2)
- RR(2), P
- F(3,4)
- . . . . . .

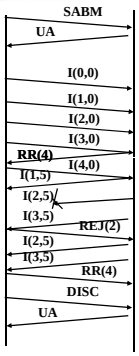


Procedimientos HDLC



Funcionamiento

- Establecimiento del enlace
- Transferencia de datos
- Desconexión



Ejemplos

- Protocolo de nivel de enlace en redes X.25 LAPB = HDLC BA 2, 8
- Utilizado en redes IP sobre enlaces punto a punto. HDLC BA 5. Utilizan tramas de información no numerada.

Limitaciones

- Orientado a entornos centralizados
- Múltiples versiones del protocolo
- Sin soporte multiprotocolo

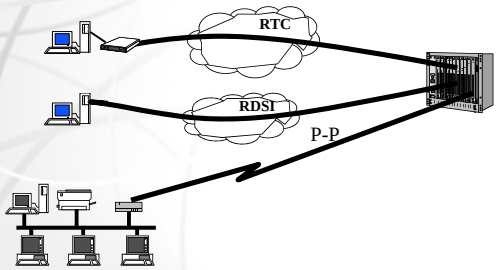
Ejemplos

- LAPD desarrollado por la UIT-T como parte de las recomendaciones para RDSI
- Proporciona el procedimiento para el control del enlace de datos sobre el canal D
- Se restringe solo a modo ABM
- El campo de dirección es de 16 bits

Ejemplos

- LLC es parte de la familia de estándares IEEE 802 para LAN
- La diferencia entre LLC y HDLC es el formato de trama
- En LLC las funciones para controlar el enlace se dividen en dos capas: MAC y LLC
- La capa MAC incluye dirección origen y destino
- La capa LLC contiene los puntos de acceso al servicio del origen y destino

Escenario



Point-to-Point Protocol: PPP

- Consiste en tres componentes:
  - Mecanismo de encapsulación (RFC 1548) sobre líneas síncronas y asíncronas (HDLC).
  - Protocolo de control de enlace (LCP): establecimiento, configuración (negociación de opciones) mantenimiento y "liberación" del enlace. (RFC 1548)
  - Opcionalmente protocolos de autenticación (PAP o CHAP)
  - Una familia de protocolos de control de red (NCP) para protocolos específicos. Existen normas para IP (RFC1332), OSI, DECNet y AppleTalk.

Escenario via RTC/RDSI

- Conexión al ISP través de la red (modem)
- Negociación del enlace (LCP)
- Autenticación (opcional)
- Negociación parámetros de Red (NCP). Ej: dirección IP.
- Transferencia de Datos con detección de errores y, opcionalmente, mecanismos de retransmisión (ARQ)
- Liberación de la conexión del nivel de red (NCP).
- Cierre ordenado del enlace (LCP).
- Desconexión del circuito (Módem)

PPP: Formato de trama

- Análogo a HDLC pero orientado a byte/carácter (envío múltiplos de 8 bits (1byte))

Bytes	1	1	1	1 o 2	variable	2 o 4	1
	Flag 7E	Address FF	Control 03	Protocol	Information	CRC	Flag 7E

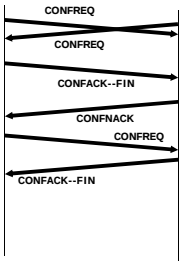
- IP: Protocol 0021 — Datagram IP
- LCP: Protocol C021 — Link Control Data
- NCP: Protocol 8021 — Network Control Data

- Flag=HDLC
- Address Broadcast (es punto a punto, solo dos estaciones, así no negocio direcciones nivel enlace)
- Mediante LCP se puede reducir el numero de bytes/trama: omisión de los campos de flag, dirección, reducción del tamaño del campo protocolo de 2 a 1 byte.

LCP

- Mensajes para negociación
- CONFREQ (lista de parametros propuestos)
- CONFREJ (no entiendo)
- CONFNACK (no soporto eso parámetros)
- CONFACK (ok)
- …

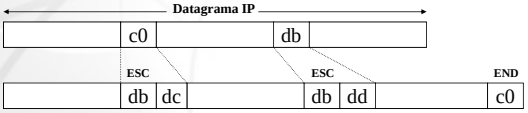
Ejemplo



Para NCP es igual

Ejemplo Serial Link IP Protocol: SLIP

- Definido para encapsular Datagramas IP sobre líneas serie (RFC1055).
- Muy difundido.
- Envía datagrama IP byte a byte añadiendo una marca de fin de Datagrama (0xc0).



- Usado principalmente para accesos a ISP

Ventajas frente a SLIP

- Soporta transferencias multiprotocolo.
- Soporta detección de errores (CRC)
- Soporta identificación de sistemas conectados mediante el uso del protocolo de control de red (NCP).
- Soporte de mecanismos de compresión
- Soporte de negociación de parámetros de enlace (LCP).
  - Autenticación (PAP, CHAP)
  - MultiLink PPP.
- Utilizado sobre RDSI, RTC y líneas P-P.