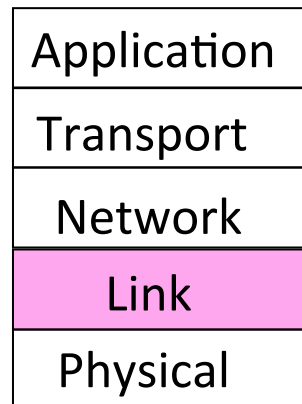


Nivel de Enlace de Datos

Introducción (1)

- Este Nivel busca lograr una comunicación:
 - Confiable
 - Eficiente
 - Entre máquinas adyacentes (físicamente conectadas entre sí)



Introducción (2)

- 1) Servicios Posibles de Nivel 2
- Servicios Sin Conexión
 - Sin confirmación
 - Máquina Origen envia tramas a Máquina Destino sin pedir confirmación
 - No se establece conexión
 - Si se pierde trama (por ruido) no se intenta recuperarla
 - Se le deja a un Nivel Superior resolver esto
 - Se usa en LANs con BER bajo

Introducción (3)

- Servicio Sin Conexión con Confirmación
 - Sin conexión
 - Pero confirma cada trama
 - Canales inestables
 - Ej. Redes inalámbricas
- Servicio Con Conexión con Confirmación
 - Orientado a conexión: 3 fases
 - Establecimiento
 - Transferencia
 - Desconexión

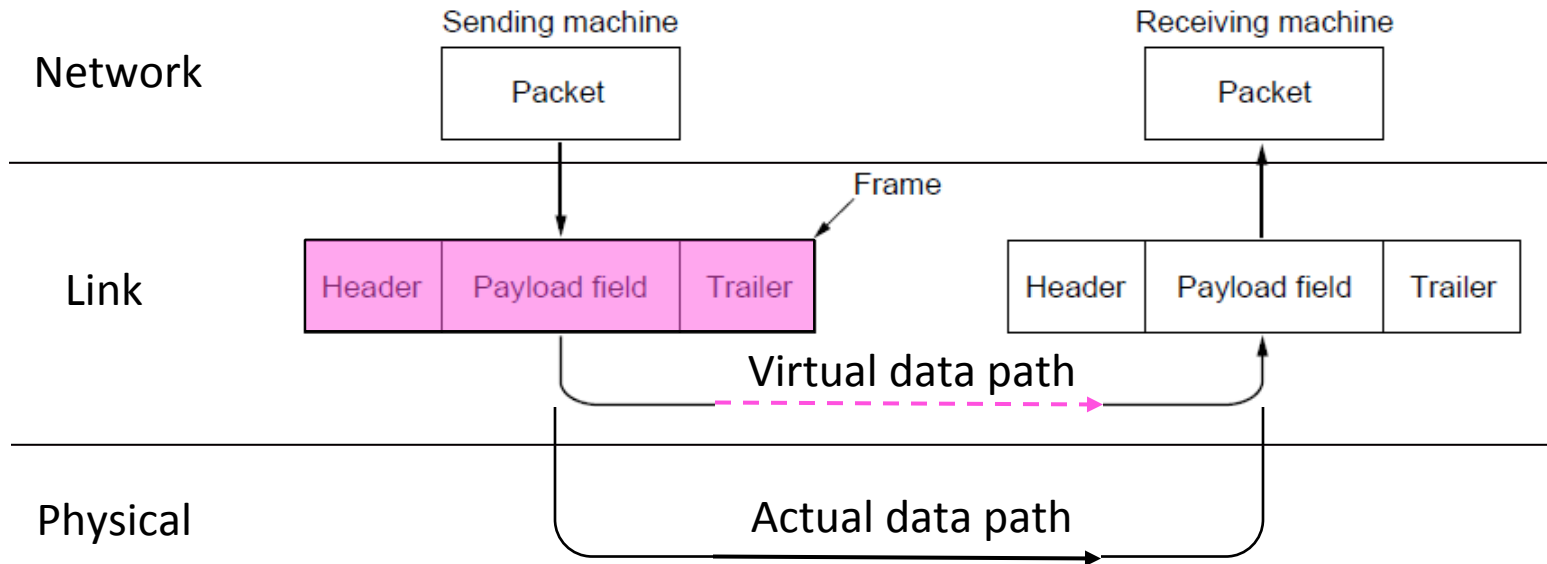
Introducción (4)

- Servicio Con Conexión con Confirmación...
 - Mayor control
 - Tramas numeradas
 - Se garantiza llegada de cada trama
 - No se mandan tramas repetidas a nivel superior
 - Las tramas se pasan al nivel superior en el orden adecuado
 - Ej.: Redes WAN
 - » Ruteadores requieren mayor control en conexiones punto a punto

Introducción (5)

– 2) Definición de Trama

- Los datos vienen de Nivel 3
- La trama se transmite por medio del Nivel 1



Introducción (6)

— 2) Definición de Trama

- La construcción de la trama se da en nivel 2 para que Nivel 1 solo la transmita
 - Define límites de trama
 - Significado de bits en la trama
 - Define el número de bits de la trama
 - Separa una trama de la otra mediante banderas de inicio y final
 - » En protocolos orientados a bits
 - Violaciones de codificación en Nivel 1
 - » Voltajes prohibidos

Introducción (7)

- 3) Control de Errores
 - Detección de Errores
 - Corrección de Errores
- 4) Mecanismos de retransmisiones
 - Acks
 - Temporizadores (Time-outs)

Introducción (8)

- 5) Control de Flujo
 - Resuelve el problema cuando un TX quiere transmitir más tramas de las que puede manejar el RX
 - Ej. PC a Impresora

- 6) Direccionamiento
 - Opcional
 - Ej . Direcciones MAC (MAC addresses)

Control de Errores (1)

- Los errores son causados por diferentes factores:
 - Ruido Térmico (siempre presente)
 - Ruido Impulsivo (perturbaciones mayores a 10 ms)
- Ruido y Afectación de Señal
 - La afectación de señal depende de la velocidad de Transmisión
 - Por ejemplo una perturbación mayor a 10 ms
 - Si $R_b = 10$ bps, entonces $T_b = 100$ ms
 - » Si acaso afecta 1 bit

Control de Errores (2)

— Ruido y Afectación de Señal

— Si $R_b = 10 \text{ Kbps}$, entonces $T_b = 0.1 \text{ ms}$

» Ahora afecta 100 bits

- Una forma de atenuar los efectos del ruido es disminuir la velocidad

— Tipos de Errores

- Aislados: independientes y más fácil de tratar
- En grupos (bursts): dependientes y más difícil de tratar

Control de Errores (3)

— Tratamiento de Ruido

- Muchas veces es muy complicado
- Ruido térmico sí es tratado matemáticamente (como vimos antes)
- Otros tipos de ruidos, debido a sus muchas causas es muy difícil de tratar
- Se usan estudios empíricos
 - Distribuciones de Errores en vez de análisis matemático
- BER (bit error rate): Tasa promedio de errores
 - Ejemplo de Helados

Control de Errores (4)

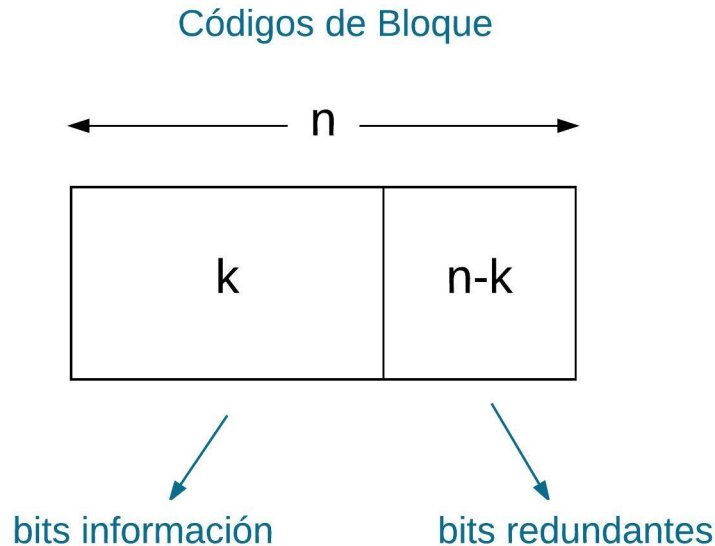
- Ejemplo de BER
 - Voz: 1×10^{-3}
 - Datos: 1×10^{-5} a 1×10^{-6}
 - Video: 1×10^{-4} a 1×10^{-5}
- Si un canal no tiene el S/N adecuado para lograr el BER buscado?
 - Se recurre a códigos de control de errores
 - Ventaja de sistemas digitales de comunicaciones

Control de Errores (5)

- Control de Errores
 - Corrección de Errores
 - Detección de Errores
- Tipos de códigos
 - Bloque (el que veremos)
 - Convolución

Control de Errores (6)

— Códigos de Bloque



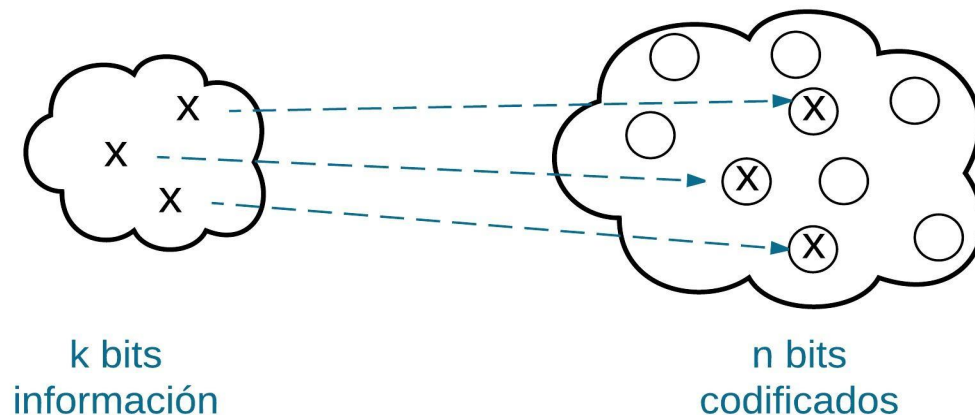
- bits de información y bits redundantes
- Se conoce como código (n,k)

Control de Errores (7)

— Códigos de Bloque.....

- Los bits redundantes son utilizados por el RX para ver si la palabra codificada de n bits llegó OK

Principio de
Funcionamiento de
control de errores



Control de Errores (8)

— Control de Errores....

- Cuando se introducen bits redundantes se debe aumentar la capacidad del sistema

$$R_c = \frac{n}{k} R$$

» R_c : velocidad cuando se codifica

» R : velocidad sin codificar

- Ejemplo:
1 → 111
0 → 000
1 → 111
0 → 000

Control de Errores (9)

- Distancia de Hamming:

- Número de posiciones en que 2 palabras codificadas (codewords) difieren:

111

000

$$d = 3$$

- Distancia mínima de Hamming

- La distancia menor entre todas las palabras posibles

$$d_{min} = 3$$

Control de Errores (10)

- Distancia de Hamming...
 - En un código, la distancia mínima de Hamming determina el número de errores que se pueden detectar o corregir
- Capacidad de Detección

$$E_D = d_{min} - 1$$

- Capacidad de Corrección

$$E_C = \left[\frac{d_{min} - 1}{2} \right]$$

$[] \rightarrow \text{entero menor}$

Control de Errores (11)

— Ejemplo: $1 \rightarrow 111$
 $0 \rightarrow 000$

- Se transmite un 0

RX	Corrección		Detección	
<i>000</i>	<i>0</i>	✓	<i>0</i>	✓
<i>001</i>	<i>0</i>	✓	<i>Retx</i>	✓
<i>010</i>	<i>0</i>	✓	<i>Retx</i>	✓
<i>011</i>	<i>1</i>	✗	<i>Retx</i>	✓
<i>100</i>	<i>0</i>	✓	<i>Retx</i>	✓
<i>101</i>	<i>1</i>	✗	<i>Retx</i>	✓
<i>110</i>	<i>1</i>	✗	<i>Retx</i>	✓
<i>111</i>	<i>1</i>	✗	<i>1</i>	✗

Control de Errores (12)

— Ejemplo...

- $d_{\min} = 3$
- $E_C = [(3-1)/2] = 1$
 - Corrije hasta 1 error
- $E_D = 3 - 1 = 2$
 - Detecta hasta 2 errores
- Un código siempre tiene una capacidad máxima
 - En detección no se puede detectar 3 errores
- Siempre se podrá detectar más errores de los que se pueden corregir

Códigos de Bloque (1)

— Códigos de Bloque Cíclicos

- Un tipo de códigos de bloque es el cíclico o polinomial
- Es muy popular porque se implementa fácilmente con registros desplazantes

$$C(x) = 1 + x + x^2 + x^3 + x^4 + \dots$$

- Los polinomios se usan para representar la palabra del código
- Este código es definido por un polinomio generador, $g(x)$, de orden “m”

Códigos de Bloque (2)

— Códigos de Bloque Cíclicos....

- m: representa el número de bits redundantes
- D: representa la palabra sin codificar (bits)
- D(x): el polinomio de datos
- Entonces, se define:

$$\frac{x^{n-k} \cdot D(x)}{g(x)} = Q(x) , \quad R(x)$$

Códigos de Bloque (3)

— Códigos de Bloque Cíclicos....

$Q(x)$: polinomio cociente

$g(x)$: polinomio generador

$R(x)$: polinomio residuo

$C(x)$: polinomio palabra codificada

- Además, se hace que:

$$C(x) = x^{n-k} \cdot D(x) + R(x)$$

Códigos de Bloque (4)

— Códigos de Bloque Cíclicos....

- Se tiene que:



- Entonces,

$$\frac{C'(x)}{g(x)} = Q'(x), \quad R'(x)$$

Códigos de Bloque (5)

— Códigos de Bloque Cíclicos....

- Por lo tanto,

Si $R'(x) = 0 \rightarrow$ no hay errores

Si $R'(x) \neq 0 \rightarrow$ sí hubo errores

— Se conoce como el cálculo del síndrome

- Los polinomios generadores son parte de un estandar. Por ejemplo:

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

Códigos de Bloque (6)

— Ejemplo Códigos de Bloque Cíclicos

- Se tiene un código (7,4), datos = 1011 y $g(x) = x^3 + x + 1$
- Entonces,

$$D = d_0 d_1 d_2 d_3 = 1011$$

$$D(x) = 1 + x^2 + x^3$$

$$D(x) = 1 \cdot x^0 + 0 \cdot x^1 + 1 \cdot x^2 + 1 \cdot x^3$$

$$x^{n-k} = x^{7-4} = x^3$$

Códigos de Bloque (7)

— Ejemplo Códigos de Bloque Cíclicos....

- Se tiene

$$\Rightarrow x^{n-k} \cdot D(x) = x^6 + x^5 + x^3$$

- Se resuelve la división usando operaciones en módulo 2

Códigos de Bloque (8)

- Ejemplo Códigos de Bloque Cíclicos....

$ \begin{array}{r} x^6 + x^5 + + x^3 \\ x^6 + + + x^3 \\ \hline x^5 + x^4 \\ x^5 + + + x^2 \\ \hline x^4 + x^3 + x^2 \\ x^4 + + + x \\ \hline x^3 + x \\ x^3 + x + 1 \\ \hline 1 \end{array} $	$ \begin{array}{r} x^3 + x + 1 \\ \hline x^3 + x^2 + x + 1 \end{array} $
--	--

Códigos de Bloque (9)

— Ejemplo Códigos de Bloque Cíclicos....

$$\Rightarrow R = r_0 r_1 r_2$$

$$\Rightarrow R(x) = 1 = 1.x^0 + 0.x^1 + 0.x^2$$

$$\Rightarrow R = 100$$

$$C(x) = x^6 + x^5 + x^3 + 1$$

$$C(x) = 1.x^0 + 0.x^1 + 0.x^2 + 1.x^3 + 0.x^4 + 1.x^5 + 1.x^6$$

$$\Rightarrow C = 100\ 1011$$

Códigos de Bloque (10)

- Ejemplo Códigos de Bloque Cíclicos....

$$C'(x) = C(x)$$

$$\Rightarrow R(x) = R'(x) = 0, \text{ sin errores}$$

Códigos de Bloque (11)

- Métodos de Control de Errores
 - Existen 2 métodos
 - FEC: Forward Error Correction
 - Corrección de errores
 - ARQ: Automatic Retransmission reQuest
 - Retransmisiones automáticas

Códigos de Bloque (12)

- Métodos de Control de Errores....
 - FEC
 - Corrección de errores
 - Intenta corregir errores
 - Mejora BER del canal directamente
 - Si aumentan mucho los errores la confiabilidad baja (siempre se corrigen menos errores de los que se detectan)
 - Se usa mucho en aplicaciones de tiempo real
 - Canales Simplex (un solo sentido)

Códigos de Bloque (13)

- Métodos de Control de Errores....
 - ARQ
 - Detección de Errores
 - Pide retransmisiones
 - Requiere canal de reversa
 - Si aumenta mucho los errores la confiabilidad es más o menos constante
 - » Mejor que FEC
 - Pero aumentan los retardos
 - » No es bueno para las aplicaciones en tiempo real
 - » Se usa más para datos

ARQ(1)

— Principio de Funcionamiento ARQ

- Después de enviar una trama, el TX espera una respuesta del RX
- La respuesta puede ser una de tres posibles
- ACK (acknowledgment) trama fue recibida correctamente
- NAK (not acknowledgement) trama fue recibida con errores
- Time Out (temporizador)
 - Si no se recibe ninguna respuesta del RX en un tiempo predeterminado
 - TX retransmite la trama

ARQ(2)

- Principio de Funcionamiento ARQ....
 - Se usa numeración cíclica
 - Se numeran tramas, ACKs y NAKs
 - NAK es opcional: podría no ser implementado

Stop & Wait (1)

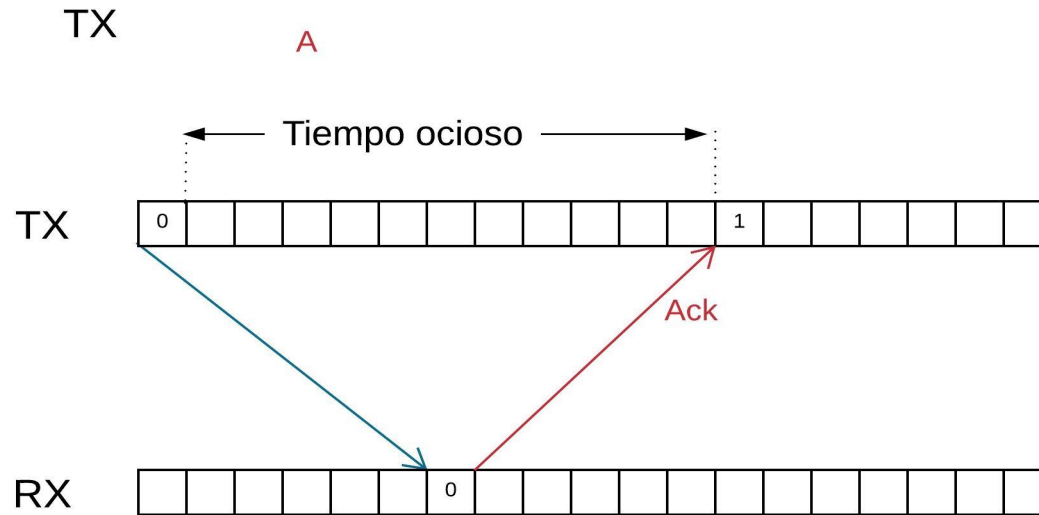
— Funcionamiento S&W

- Se envia una trama
- Se espera recibir un Ack antes de continuar
- Durante este tiempo el TX está ocioso
- Si se recibe un Nak o T.O. entonces se retransmite

Stop & Wait (2)

— Funcionamiento S&W....

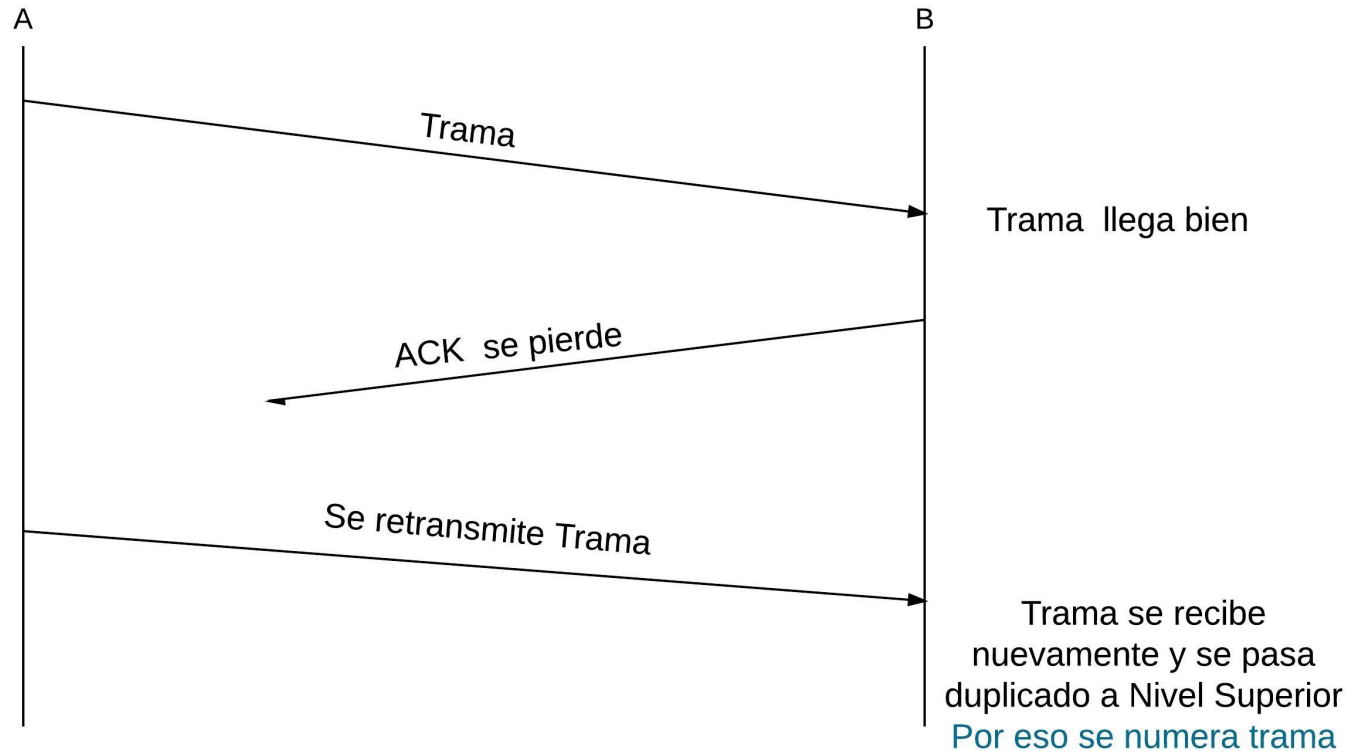
- Diagrama de Tiempo



- Dependiendo de t_p , puede ser ineficiente

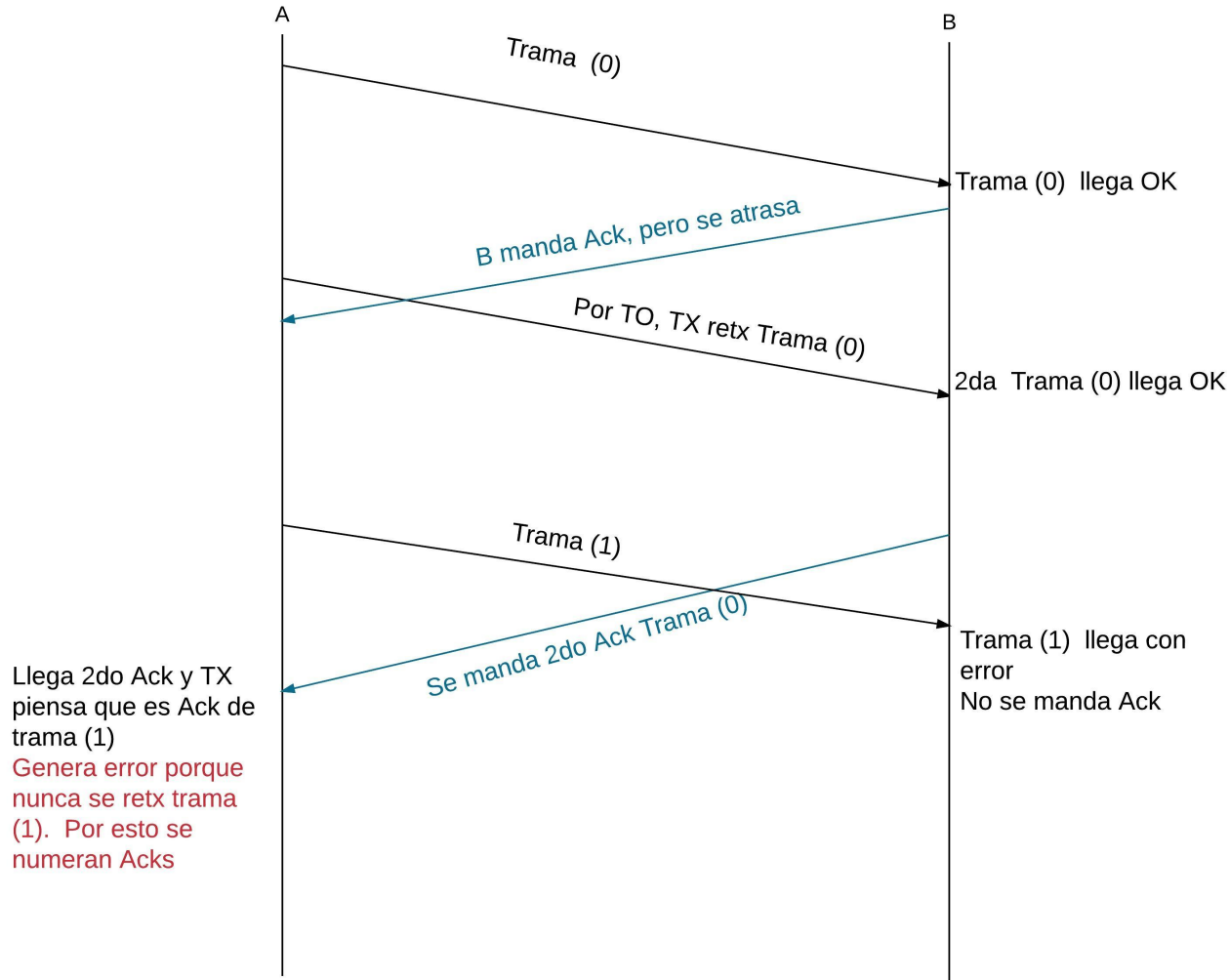
Stop & Wait (3)

Numerar Tramas



Stop & Wait (4)

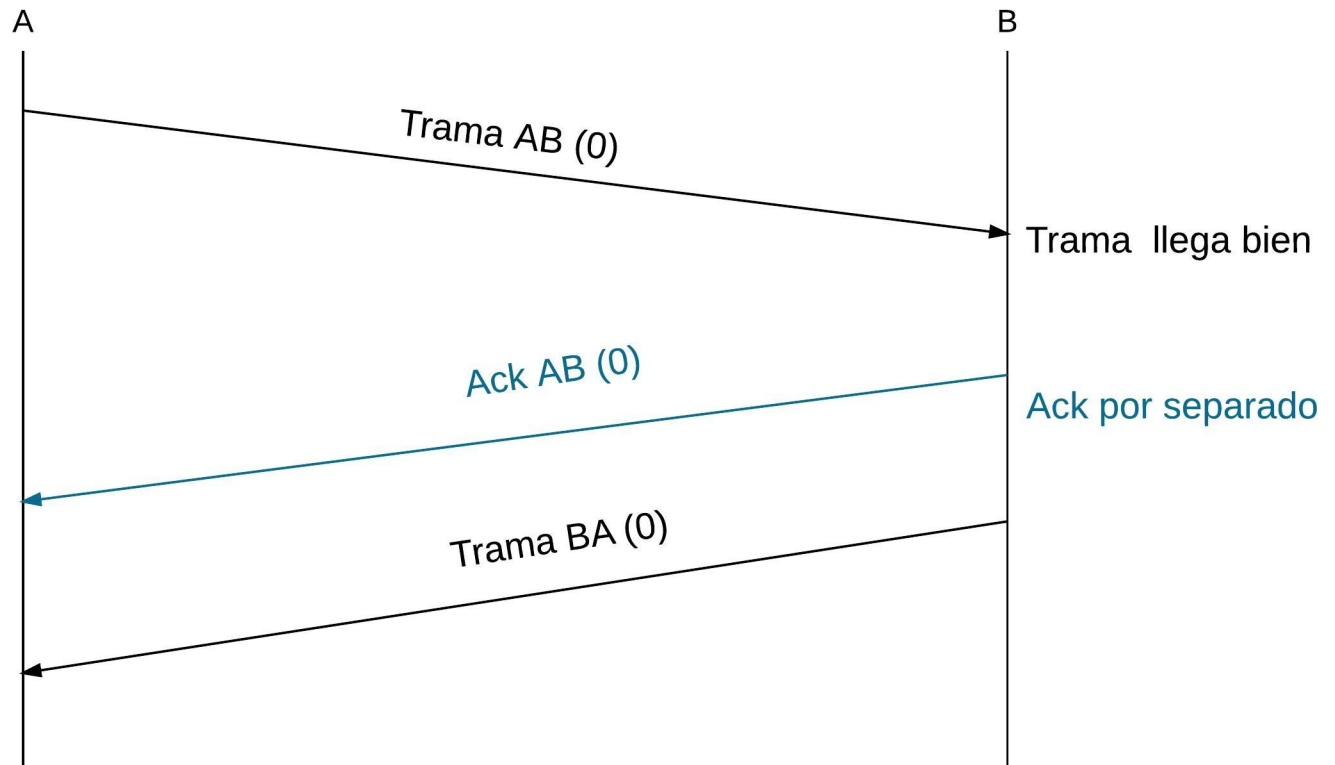
Numerar Acks



Stop & Wait (5)

- Acks normales y en piggyback

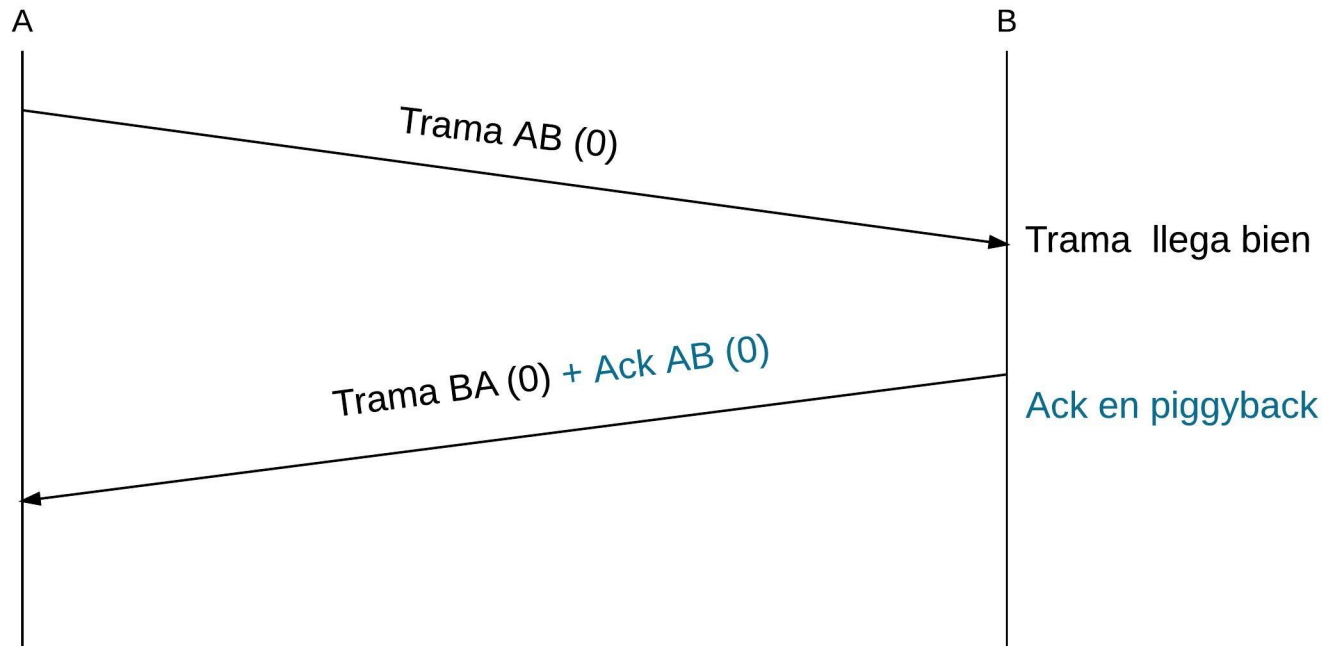
Acks Normales



Stop & Wait (6)

- Acks normales y en piggyback....

Acks en Piggyback



Stop & Wait (7)

- Protocolos del Libro
 - Empieza con el más simple: Utopia
 - A transmite a B solamente
 - Sin ruido
 - Sin errores
 - Después S&W
 - Half Duplex
 - Full Duplex
 - Limitaciones

Stop & Wait (8)

- Protocolos del Libro....
 - Finalmente
 - Protocolos que funcionan
 - Protocolo 4: S&W
 - Protocolo 5: GBN
 - Protocolo 6: SR
 - Se definen subrutinas para:
 - Enviar o Recibir tramas
 - » From Physical Layer
 - » To Physical Layer

Stop & Wait (9)

- Protocolos del Libro....
 - Se definen subrutinas para:.....
 - Enviar o Recibir paquetes
 - » From Network_Layer
 - » To Network_Layer
 - Se definen eventos
 - Time Out
 - » Start_Timer
 - » Stop_Timer
 - Acks en Piggyback
 - » Start_Ack_Timer
 - » Stop_Ack_Timer

Stop & Wait (10)

- Protocolos del Libro....
 - Se define estructura de tramas o paquetes
 - Data, Ack, Nak: `frame_kind`
 - Numeración trama: `seq_nr_seq`, `seq_nr_Ack`, etc.
 - Sirve para entender las funciones a realizar en Nivel 2 y para hacer simulaciones de estos protocolos

Protocolos con Ventanas (1)

- Antes de proseguir vamos a introducir otro concepto
- Protocolos con Ventanas (sliding window protocols)
- Estrategia del TX (Transmisor)
 - Mantiene una lista de los números de secuencia que puede enviar
 - Se llama Ventana del TX
 - Las tramas de la ventana del TX representan tramas que han sido enviadas pero que no han sido confirmadas

Protocolos con Ventanas (2)

— Ventana TX....

- La ventana del TX es de tamaño variable
- El TX conserva todas las tramas de la ventana en caso de que tenga que retransmitir alguna de las tramas
- El número de buffers = número de campos de la ventana
- Si se llena la ventana del TX, se para la transmisión hasta que llegue un Ack y libere un espacio

Protocolos con Ventanas (3)

— Ventana RX

- El RX tiene una ventana que contiene los números de secuencia de las tramas que puede aceptar
- Si llega una trama con un número de secuencia fuera de la ventana se descarta
- El número de buffers = número de campos de la ventana
- La ventana del RX es de tamaño constante

Protocolos con Ventanas (4)

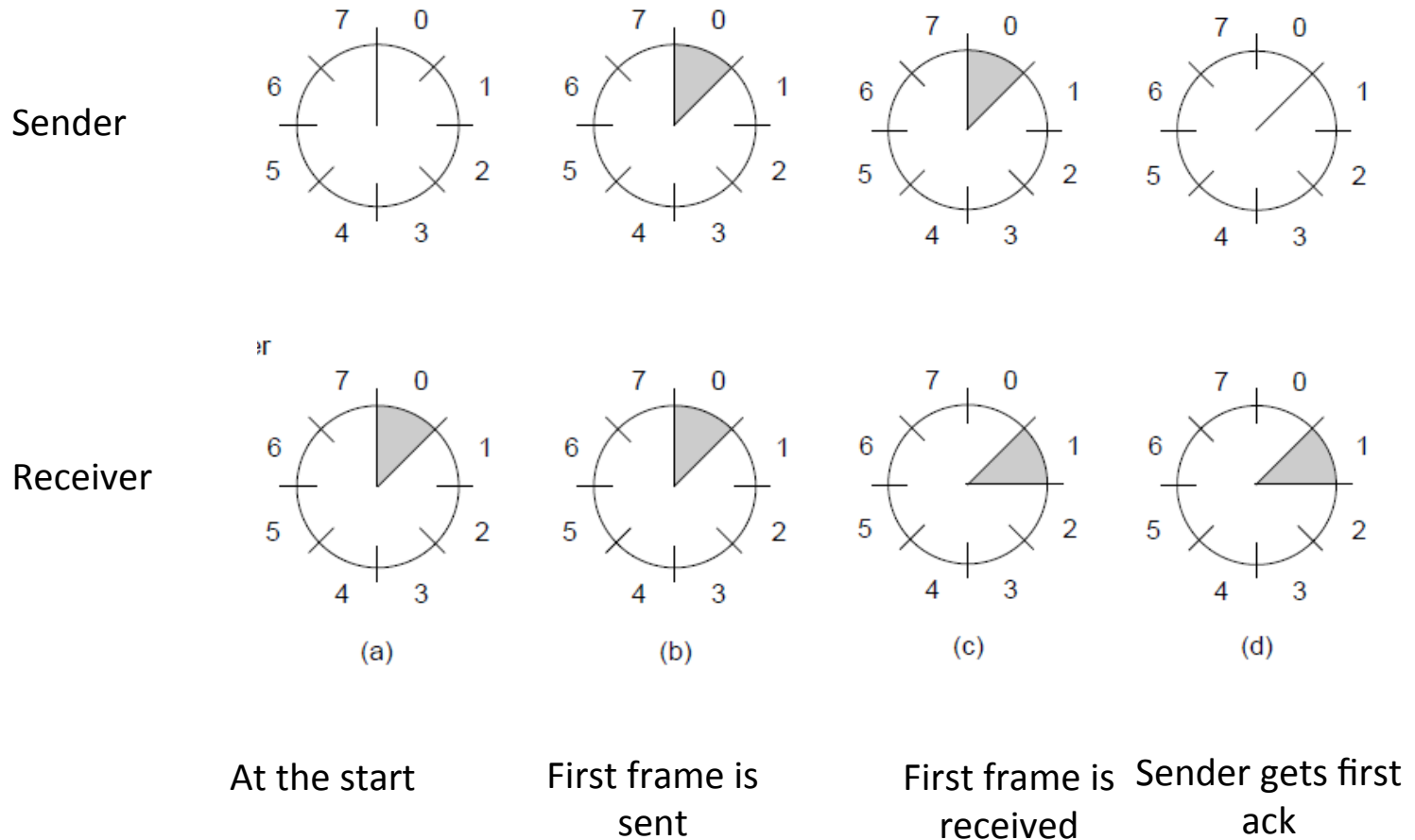
- Ventana RX....

- Si se recibe una trama con el número de secuencia igual al límite inferior de la ventana:
 - Se pasa trama al nivel superior
 - Se manda Ack al TX
 - La ventana del RX rota en 1
- No es necesario que la Ventana del TX sea igual a la Ventana del Rx

Protocolos con Ventanas (5)

- Las ventanas permiten a los nodos cambiar el orden de enviar o recibir las tramas
- Sin embargo, a la hora de pasarselas al nivel superior se debe de hacer en orden
- Ejemplo:
 - Veamos el siguiente ejemplo
 - Ventana del TX = 1
 - Ventana del RX = 1
 - Básicamente estamos hablando del Stop&Wait
 - Se usa 3 bits de número de secuencia

Protocolos con Ventanas (6)

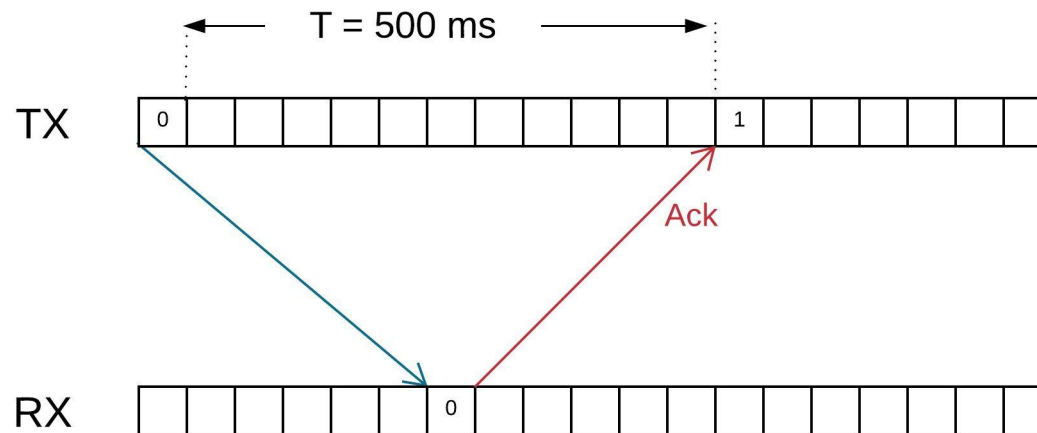


Esquemas Continuos (1)

- Problema de Eficiencia del S&W
 - Supongamos los siguientes datos
 - T = tiempo de ida y vuelta
 - t_p = tiempo de propagación
 - $T = 2 t_p$
 - Ejemplo:
 - $R = 50$ kbps
 - $T = 500$ ms
 - Trama: $n = 1000$ bits

Esquemas Continuos (2)

— Problema de Eficiencia del S&W....



- $t = 0$ inicio
- $t = 20 \text{ ms}$ se transmite trama(0)
- $t = 520 \text{ ms}$ llega Ack(0)

Esquemas Continuos (3)

- Problema de Eficiencia del S&W....
 - Se puede definir eficiencia del TX como:
 - % del tiempo que está activo o transmitiendo

$$E_{TX} = \frac{n/R}{n/R + T} = \frac{20}{20 + 500} = \frac{20}{520}$$

$$E_{TX} = 4\%$$

$$Si \ n < T.R \Rightarrow \ E_{TX} < 50\%$$

Esquemas Continuos (4)

- t_p influye en la eficiencia del TX en un ARQ
- Esquema Continuo
 - En este tipo de ARQ el TX enviará más de una trama antes de requerir un Ack (es como tener un crédito)
 - En el ejemplo anterior si tuvieramos un esquema continuo:
 - Ventana del TX = 26 tramas
 - $(20 \text{ ms}) \cdot (26) = 520 \text{ ms}$

$$E_{TX} = 100\%$$

Esquemas Continuos (5)

— Esquema Continuo....

- Existen 2 tipos
 - Go Back N: (GBN) Regrese N
 - Selective Repeat (SR) Repetición Selectiva

— GBN

- TX: debe transmitir N-1 tramas antes de requerir un Ack
- Donde N:

$$N = \left\lceil \frac{T \cdot R}{n} + 1 \right\rceil$$

Esquemas Continuos (6)

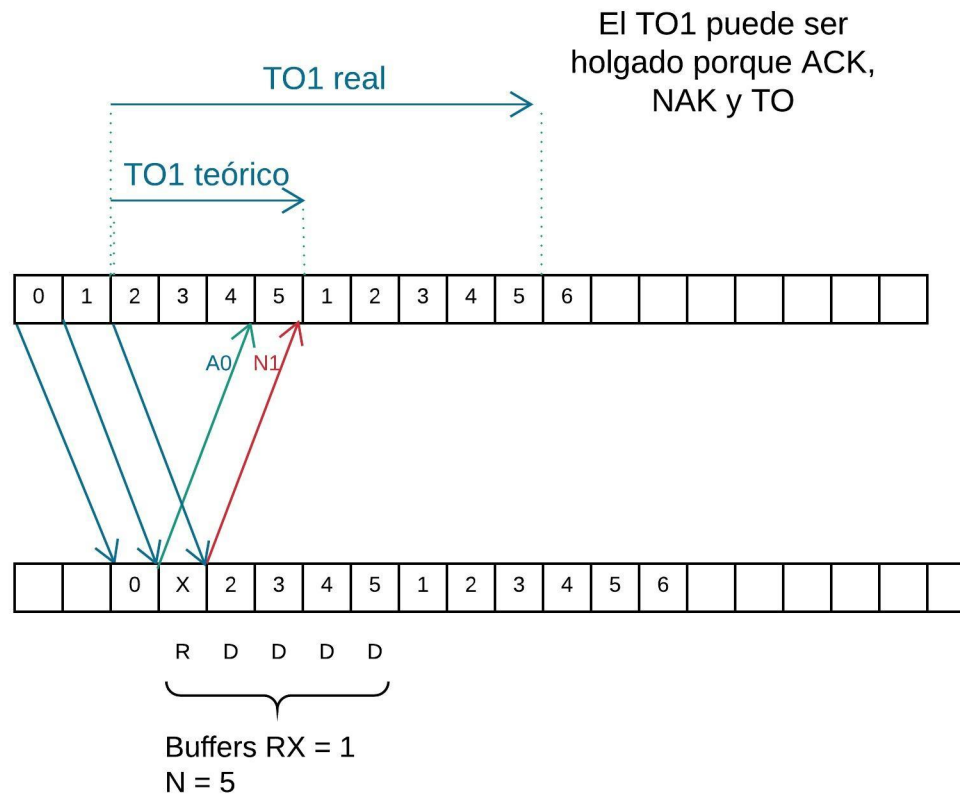
— GBN....

- n = número de bits de la trama
- $T = 2 t_p$
- R = velocidad en bps
- $[]$ = significa próximo entero ej. $[2.1] = 3$
- RX solo acepta tramas en orden
- Si la trama(i) tiene errores, se debe retransmitir la trama(i) + N-1 tramas sucesoras a la trama(i)
- Tengan o no errores

Esquemas Continuos (7)

— GBN....

- Ejemplo con Ack, Nak, TO, Vent-Tx=N=5

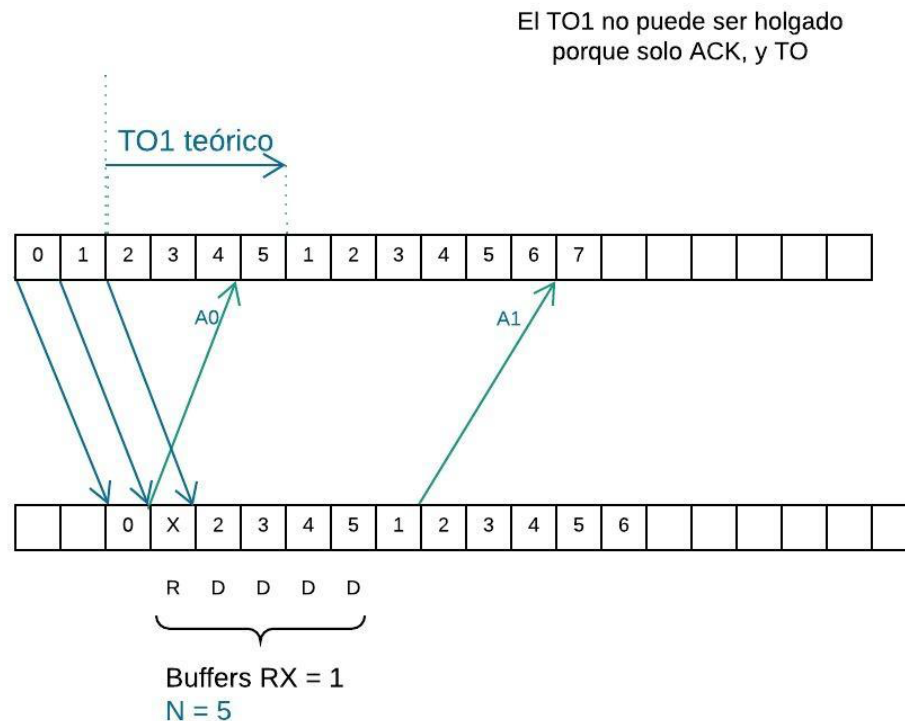


R: reservado
D: descarta

Esquemas Continuos (8)

— GBN....

- Ejemplo con Ack, TO, Vent-Tx=N=5, TO Teórico

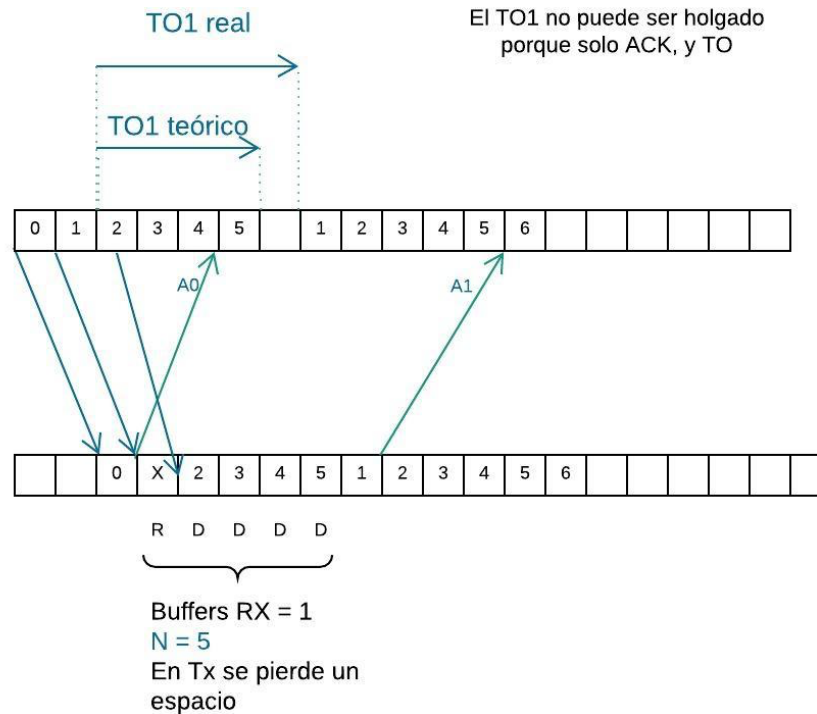


R: reservado
D: descarta

Esquemas Continuos (9)

— GBN....

- Ejemplo con Ack, TO, Vent-Tx=N=5, TO Real



Esquemas Continuos (10)

— GBN....

- Como se ven en las últimas dos diapositivas al no implementar Nak el tiempo de TO es más crítico
 - Sí es muy corto: posibles retransmisiones de tramas que habían llegado OK por atrasos en los Acks
 - Sí es muy largo, se puede perder eficiencia, es decir, que deje de ser un esquema continuo
 - » Se ve en diapositiva con $N = 5$, deja de ser continuo

Esquemas Continuos (11)

— GBN....

- Un punto importante es ver los Números de Secuencia
- Supongamos:
 - m : bits para números de secuencia
 - $m = 3$, entonces podríamos tener Vent-TX = 8
 - Vent-Tx = [0,7]
 - Se puede transmitir entonces: 0,1,2,3,4,5,6,7
 - Se recibe Ack(7): Todas llegaron OK
 - Se transmite un segundo grupo de 8 tramas (9-16)
 - Este segundo grupo: 0,1,2,3,4,5,6,7
 - Ya que la Vent-TX = 8

Esquemas Continuos (12)

— GBN....

- Supongamos:.....

- Después de que se transmitió el segundo grupo de tramas

- » Se recibe Ack(7)

- » Que significa??

- » 1. Llegó bien el segundo grupo de 8?

- » 2. Solo llegó bien el primer grupo de 8?

- » No se puede saber

- Por lo tanto, para corregir esto se hace:

$$Vent - TX = 2^m - 1$$

Esquemas Continuos (13)

— GBN....

- Resolvamos nuevamente las supocisiones anteriores:
 - $m = 3$, entonces podríamos tener $Vent-TX = 7$
 - $Vent-Tx = [0,6]$
 - Se puede transmitir entonces: 0,1,2,3,4,5,6
 - Se recibe $Ack(6)$: Todas llegaron OK
 - Se transmite un segundo grupo de 7 tramas (8-14)
 - Este segundo grupo: 7,0,1,2,3,4,5
 - Ya que la $Vent-TX = 7$
 - Se recibe $Ack(5)$

Esquemas Continuos (14)

— GBN....

- Resolvamos nuevamente las supocisiones anteriores:.....
 - Después de que se transmitió el segundo grupo de tramas
 - » Que se recibe??
 - » 1. Llegó bien el segundo grupo de 7?
 - Se recibe Ack(5)
 - » 2. Solo llegó bien el primer grupo de 7?
 - Se recibe Ack(6)
 - Por lo tanto, ya no habría incertidumbre de lo que pasó: **NO SE TRASLAPAN LAS VENTANAS**

Esquemas Continuos (15)

— Mecanismo de Acks

- Antes de ver SR, revisemos este concepto
- Se debe de notar que se está trabajando con Acks acumulativos
- Que significa?
 - Del ejemplo anterior, el Ack(6) reconoce las siguientes tramas 0,1,2,3,4,5,6 (el primer grupo)
 - Ack(5) reconoce 7,0,1,2,3,4,5 (el segundo grupo)
- Que significa Ack(6) si llega después de transmitir segundo grupo?
 - Está fuera de la ventana por lo tanto no reconoce a ninguna

Esquemas Continuos (16)

— Mecanismo de Acks....

- Este concepto da mayor confiabilidad a los Acks
- Ya que no hacemos Acks de los Acks
- Por ejemplo, si se pierde un Ack, puede tener poco efecto porque el siguiente Ack puede reconocer la trama actual y la anterior

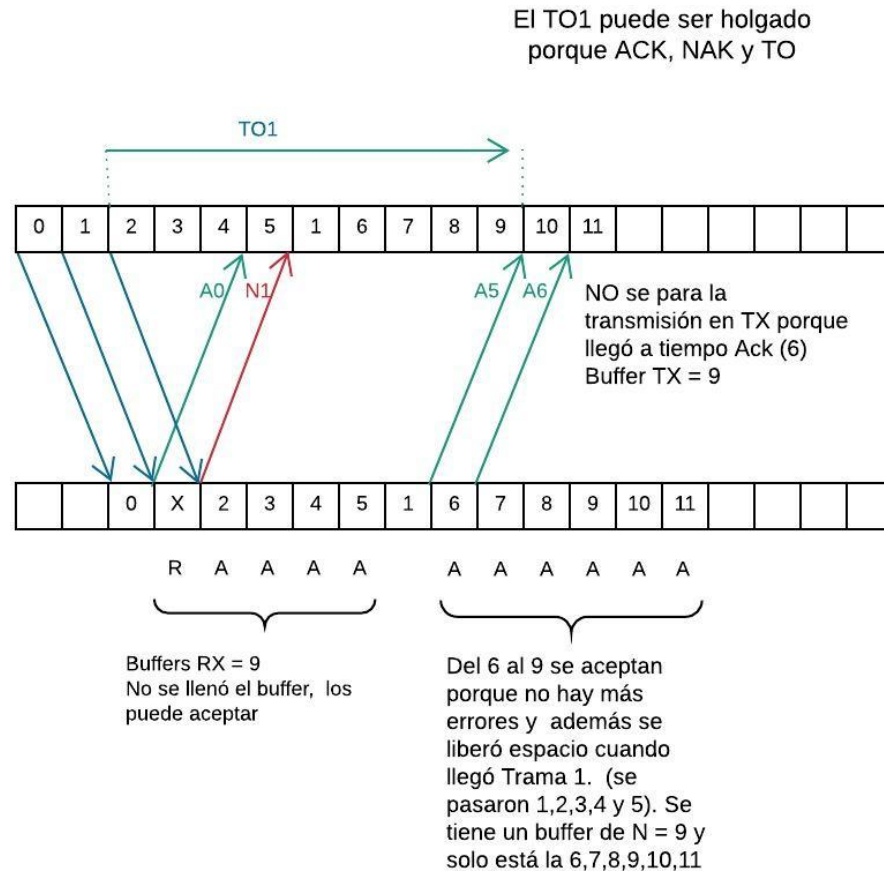
Esquemas Continuos (17)

— SR

- TX debe de ser capaz de transmitir $N-1$ tramas antes de requerir un Ack
- RX, si detecta un error, solo pide la retransmisión de esa trama (Repetición Selectiva)
- La Vent-RX debe de ser capaz de almacenar las otras $N-1$ tramas siguientes al error
- Los buffers de RX debe ser igual a N
- Veamos siguiente ejemplo con Ack, Nak, TO, $N=9$

Esquemas Continuos (18)

— SR: $N = 9$



R: reservado
D: descarta
A: Puede aceptar trama

Esquemas Continuos (19)

— SR....

- Problemas del SR

- La implementación no es tan fácil
- La memoria era cara hace algun tiempo
- Sí hay más de 1 error, el tamaño N de la Ventana del TX y RX debe ser mayor a $2t_p$ ya que sino deja de ser continuo
- Dado que se usan Acks acumulativos, también afecta la continuidad a este ARQ
 - » En el ejemplo anterior, que pasa si la trama(3) tiene errores?
 - » No se puede mandar el Ack(2) o otros y esto seguro provocará un problema de TO

Esquemas Continuos (20)

— SR....

- Problemas del SR....
 - Que pasa si la trama(1) vuelve a tener un error?
 - » Se perdería la continuidad
- En teoría, el SR es el que debería de dar mejor rendimiento (throughput) **un nuevo concepto**
- Ya que es el que en teoría desperdicia menos
- Entonces, se debe de distinguir
 - SRI: SR ideal
 - SRR: SR real (con la memoria que se implemente)
 - Podría requerir mucha mayor memoria para el SRI

Esquemas Continuos (21)

— Manejo de Números de Secuencia SR

- Iniciemos con $2^m - 1$ como el GBN
- Supongamos $m = 3$

- **TX**
- TX: 0,1,2,3,4,5,6
- Vent-TX: [0,6]

- Se pierden todos los Acks
- Vent-TX: [0,6]
- TO(0)
- Se retx trama(0)

- **RX**
- Vent-RX [0,6]
- Se recibe todo OK
- Se mandan Ack(0)... Ack(6)
- Vent-RX: [7,5]

Esquemas Continuos (22)

— Manejo de Números de Secuencia SR.....

- | | |
|--|---|
| <ul style="list-style-type: none">— TX— Vent-TX [0,6]— Llega Ack(6) por piggyback— Vent-TX [7,5]— Ahora se sigue con la transmisión— Se transmite: 7,0,1,2,3,4,5 | <ul style="list-style-type: none">— RX— Vent-RX

— Vent-RX[7,5]— Llega trama(0) vieja— Se acepta porque está en ventana RX— Llega trama(7)— Se acepta |
|--|---|

Esquemas Continuos (23)

- Manejo de Números de Secuencia SR.....

-

TX

-

RX

- Se pasa trama(0) vieja y trama(7) nueva
- Protocolo falla porque paso 2 veces la vieja trama(0)
- Se traslapan las ventanas

Esquemas Continuos (24)

- Manejo de Números de Secuencia SR....

$$Vent - TX/RX = \frac{2^m}{2}$$

- Para evitar esto:

- **TX**

- Vent-TX = 4, Vent-TX [0,3]
- TX: 0,1,2,3
- Vent-TX: [0,3)

- Se pierden todos los Acks
- Vent-TX: [0,3]
- TO(0)
- Se retx trama(0)

- **RX**

- Vent-RX [0,3]
- Se recibe todo OK
- Se mandan Ack(0)... Ack(3)
- Vent-RX: [4,7]

Esquemas Continuos (25)

— Manejo de Números de Secuencia SR.....

— TX

- Se retransmite trama(0)

- Llega Ack(3), por piggyback
- Vent-TX [4,7]
- Se transmite siguiente trama
- Trama(4)

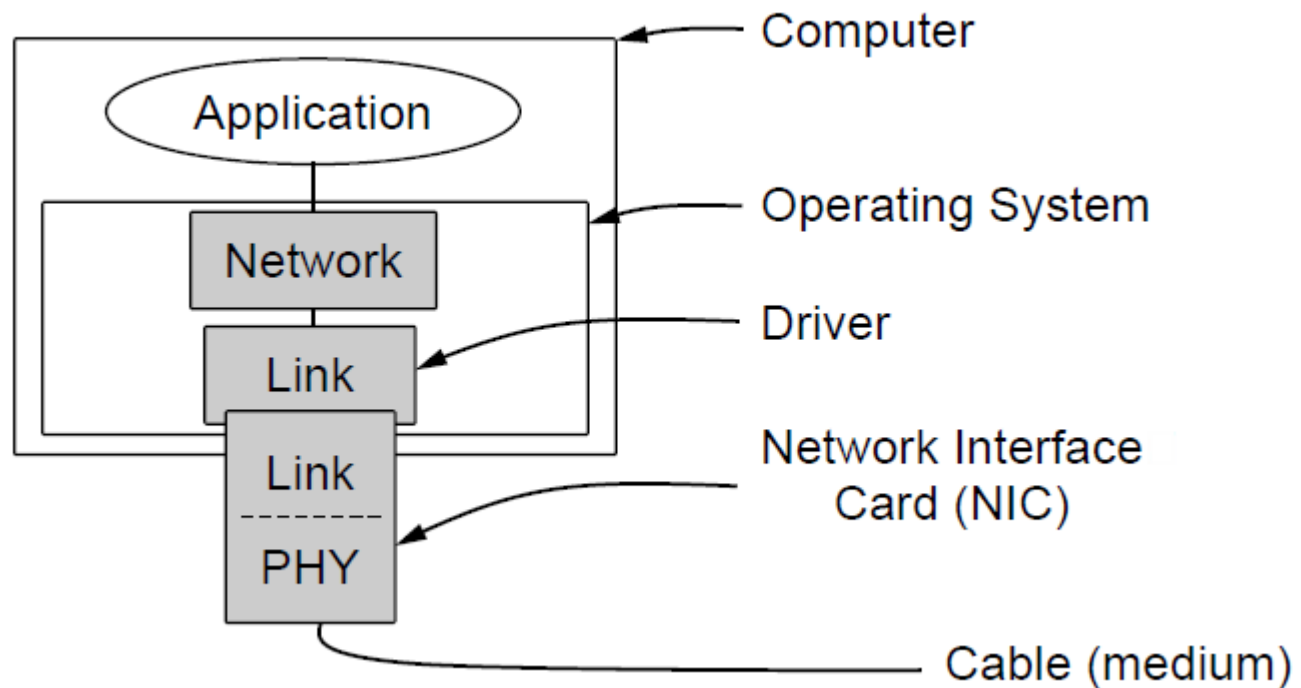
— RX

- Se recibe trama(0) vieja
- No se acepta por estar fuera de la ventana

- Llega trama(4), y se acepta (dentro ventana)

HDLC (1)

- Primero, donde se implementa el nivel 2?
 - NICs y drivers del sistema operativo del Host



HDLC (2)

— Protocolo real: HDLC.....

- En los 70's se empiezan a trabajar en protocolos orientados a bits (antes eran orientados a bytes)
- IBM desarrolló SDLC (Synchronous Data Link Control)
- Parte de Arquitectura SNA de IBM
- IBM se lo entrega a ANSI para buscar hacerlo un estandar en EUA
- Luego, la ISO lo modifica y saca HDLC (High level Link Control)

HDLC (3)

— Protocolo real: HDLC.....

- Luego la CCITT (hoy la UIT de Naciones Unidas) lo transforma a LAP y luego LAP-B (Link Access Protocol – Balanced)
- Finalmente, Comité IEEE-802 lo modifica y saca el LLC (Logical Link Control) para LANs
- En conclusión, todos muy parecidos entre sí

HDLC (4)

- Protocolo real: HDLC.....
 - Tipos de Estaciones
 - Primaria: Genera comandos (amo)
 - Secundaria: Responde a comandos (esclavo)
 - Combinada: Ambos
 - Configuración de Línea
 - Unbalanced: punto a punto y multipunto (1 primario y 1 o más secundarios)
 - Full Duplex, Half-Duplex
 - Balanceado: punto a punto, 2 estaciones combinadas
 - Full Duplex, Half Duplex

HDLC (5)

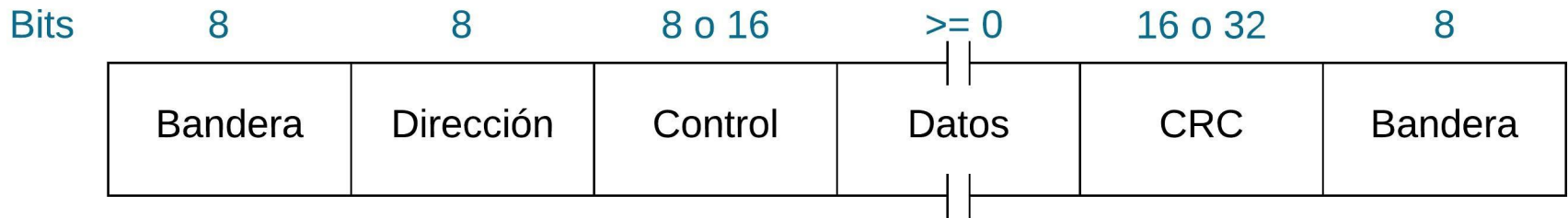
- Protocolo real: HDLC.....
 - Modos de Operación
 - Normal Response Mode (NMR)
 - » No balanceado
 - » Primario genera
 - » Secundario responde
 - Asynchronous Balanced Mode (ABM)
 - » Balanceado
 - Asynchronous Response Mode (ARM)
 - » No Balanceado
 - Amo genera
 - Secundario responde y genera

HDLC (6)

— Protocolo real: HDLC.....

- Formato de trama

Trama HDLC



- Encabezado (header): Bandera, Dirección, Control
- Trailer: CRC y Bandera

HDLC (7)

- Protocolo real: HDLC.....
 - Usa transmisión sincrónica
 - Bandera: Inicio y final de trama
 - Para sincronización del RX con TX
 - Bandera: 0111 1110
 - Se puede usar una única bandera fin e inicio de la siguiente trama si esta es inmediata
 - La secuencia de la bandera está prohibida dentro de los datos
 - Para lograr esta independencia de los datos se usa una técnica de **Inserción de Bits (bit stuffing)** en el campo de datos

HDLC (8)

- Protocolo real: HDLC.....
 - Inserción de Bits (bit stuffing)
 - » TX: siempre inserta un 0, después de 5 unos consecutivos
 - » RX: siempre quita un 0, después de 5 unos consecutivos
 - » 1111 11 → 1111 101
 - » Si ocurren 7 bits consecutivos en 1: condición de aborto
- Dirección
 - No se necesita en enlaces de punto a punto
 - Solo se necesita en enlaces multipunto

HDLC (9)

- Protocolo real: HDLC.....
 - Dirección....
 - Se incluye dirección en enlaces de punto a punto por uniformidad
 - Tramas comando: dirección indica a cual secundario
 - Tramas respuestas: cual secundario envia
 - La dirección puede ser de más de 8 bits
 - » Si LSB = 1 indica que es último byte
 - » Si LSB = 0 indica que no es el último byte
 - Control
 - Identifica la función de la trama y los números de secuencia

HDLC (10)

- Protocolo real: HDLC.....
 - Control....
 - Existen 3 tipos de trama
 - » I: Trama de Información (lo normal)
 - » S: Trama de supervisión: tipo de ARQ y números de secuencia
 - » U: Trama sin numerar (unnumbered) sirve para control interno o configuraciones
 - Existen dos modos de Operación:
 - » Modo normal: 8 bits, números de secuencia de 3 bits
 - » Modo extendido: 16 bits, números de secuencia de 7 bits

HDLC (11)

— Protocolo real: HDLC.....

- Control....

Bits	1	7	1	7	modo extendido
	1	3	1	3	modo normal
I	0	N(s)	P/F	N(R)	
S	1	0 S ₁ S ₂	P/F	N(R)	
U	1	1 U1 U2	P/F	N(R)	

» N(s): Número de secuencia

» N(R): Número de Ack

HDLC (12)

- Protocolo real: HDLC.....
 - Control....
 - Tramas U: establecen o terminan una conexión
 - » Selecciona Modo de Operación (NRM, ABM o ARM)
 - » Selecciona módulo (normal o extendido)
 - » Sondeo (polling) de estaciones
 - » Reporte de estatus
 - Tramas I: transfieren los datos
 - » N(S): 3 o 7 bits
 - » N(R): 3 o 7 bits
 - » Numeración Ack
 - Última trama recibida
 - Primera trama no recibida

HDLC (13)

— Protocolo real: HDLC.....

- Control....

- Tramas S: tipo ARQ, hay 4 tipos

- » Tipo 0: Receive Ready: Ack: significa que trama sí llegó bien al RX y no se uso piggyback

- » Tipo 1: Nak del GBN

- N(R): a partir de cual trama se retransmite

- » Tipo 2: Ack, pero le dice a TX que pare transmisión temporalmente (problemas de buffer)

- Control de flujo

- Cuando se soluciona se envia trama Tipo 0, Receive Ready

HDLC (14)

- Protocolo real: HDLC.....
 - Control....
 - Tramas S:....
 - » Tipo 3: Selective Reject: NAK de Selective Repeat
 - N(R) indica cual trama retransmitir
 - SDLC, LAP-B y LLC no lo implementan por la poca ganancia de rendimiento
 - Bit P/F: (Poll/Final)
 - » Mecanismo de chequeo
 - » Se usa para Sondeo (polling)
 - P: bit trama comando
 - F: bit trama respuesta
 - F=1 en última trama respuesta
 - Siempre se usa en pares

HDLC (15)

- Protocolo real: HDLC.....
 - CRC:

$$C(x) = x^{16} + x^{12} + x^5 + 1$$