CAPA 2, Control de Errores y Control de Flujo

Taller de Redes Inalámbricas

Tecnólogo en Telecomunicaciones

Agenda

Agenda

¿por qué se producen los errores?

¿por qué se producen los errores?

Medio Físico no confiable

1. Distorsión de la señal o pérdidas por atenuación
2. Perturbaciones eléctricas o electromagnéticas (interferencia)

Puede darse:

- error de bit
- error de ráfaga

Gestión de los errores de transmisión

Es un tema relevante en las redes inalámbricas

Estrategias de manejo de errores:

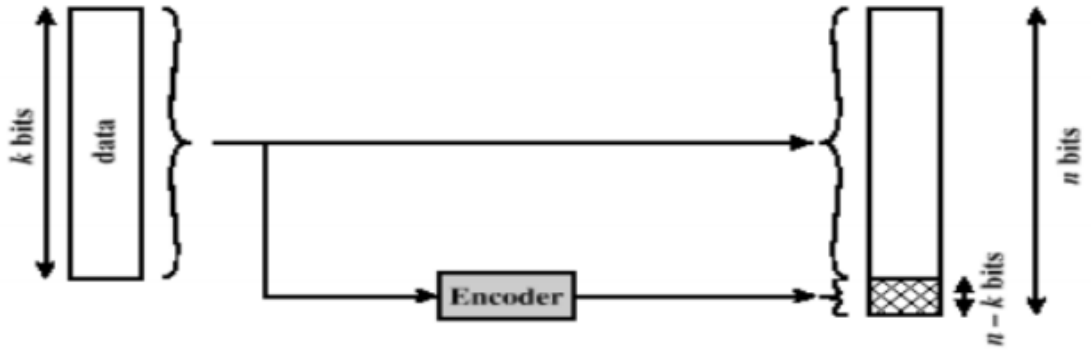
- no se tienen en cuenta los errores
- **Códigos de detección de errores:** diseñados para detectar la presencia de errores en la transmisión
- **Códigos de corrección de errores:** diseñados para detectar y corregir errores. FEC: Forward Correction Codes
- **Automatic repeat request (ARQ):**
  - Es una técnica usada conjuntamente con la detección/corrección de errores
  - Los datos con errores (que no se pueden corregir) son descartados
  - El transmisor retransmite ese bloque de datos

Probabilidad de detección de errores

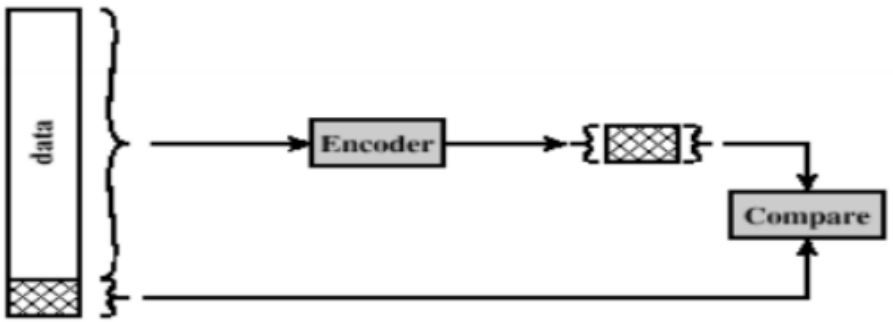
Definiciones:

- P_b: Probabilidad de un error en un sólo bit (BER)
- P_t: Probabilidad de que una trama arribe sin errores, para una trama de F bits, $P_t = (1 - P_b)^F$
- Los bits son corrompidos de manera independiente y con la misma probabilidad

El proceso de detección de errores



(a) Sender



(b) Receiver

Ejemplo simple: control de paridad

- Un bit de paridad se agrega al bloque de datos
- **Paridad par:** el bit agregado asegura un número par de 1s
- **Paridad impar:** el bit agregado asegura un número impar de 1s
- Ejemplo 7-bits [1110001]
  - paridad par: [11100010]
  - paridad impar: [11100011]

Ejemplo simple: control de paridad

- Un bit de paridad se agrega al bloque de datos
- **Paridad par:** el bit agregado asegura un número par de 1s
- **Paridad impar:** el bit agregado asegura un número impar de 1s
- Ejemplo 7-bits [1110001]
  - paridad par: [11100010]
  - paridad impar: [11100011]
- Este método **detecta** un número impar de errores

Ejemplo: Chequeo de redundancia cíclico (CRC)

Transmisor:

- Para un bloque de k bits, el transmisor genera una trama de chequeo de secuencia de $(n - k)$ bits
- La trama resultante de n bits es exactamente divisible por un número predeterminado

Receptor:

- Divide la trama que recibió por un número predeterminado
- Si el resto es 0, asume que no hubo errores.

También se los conoce como códigos polinomiales.

CRC - explicación con un ejemplo

Información a enviar: 10110101101 $x^{10}+x^8+x^7+x^5+x^3+x^2+1$
Polinomio generador: 10011 x^4+x+1
Polinomio a dividir: 101101011010000 $(x^{10}+x^8+x^7+x^5+x^3+x^2+1)\cdot x^4$ $x^{14}+x^{12}+x^{11}+x^9+x^7+x^6+x^4$
División en módulo 2:

```
101101011010000 |10011          x14 x12 x11 x9 x7 x6 x4 | x4+x+1
10011           10101010010     x14     x11 x10         x10+x8+x6+x4+x
 01011                              x12 x10 x9
 00000                              x12     x9 x8
  10110                                 x10 x8 x7 x6
  10011                                 x10     x7 x6
   01011                                    x8     x4
   00000                                    x8     x5 x4
    10111                                         x5
    10011                                         x5     x2 x
     01000                                                x2 x
     00000
      10001
      10011
       00100
       00000
        01000
        00000
         10000
         10011
          00110
          00000
           0110
```

Resto obtenido: 0110 x^2+x
Trama transmitida: 101101011010110 $x^{14}+x^{12}+x^{11}+x^9+x^7+x^6+x^4+x^2+x$

CRC - versiones ampliamente utilizadas de polinomio generador

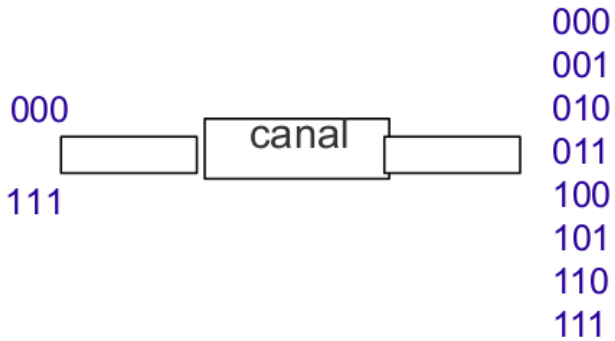
- CRC–12
  - $X^{12} + X^{11} + X^3 + X^2 + X + 1$
- CRC–16
  - $X^{16} + X^{15} + X^2 + 1$
- CRC – CCITT
  - $X^{16} + X^{12} + X^5 + 1$
- CRC – 32
  - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Errores en redes inalámbricas

- La detección de errores requiere retransmisiones
- La detección/retransmisión genera problemas en aplicaciones sobre redes inalámbricas
  - La tasa de error en redes inalámbricas puede ser alta, resultando en un gran número de retransmisiones
  - Genera retardos de transmisión largos comparado con el retardo de propagación.
- Habitualmente se emplean códigos de corrección de errores y CRC combinados. Si no hay "muchos" errores se corrigen y sino sólo se detectan.
- Nos concentraremos en códigos de corrección de errores

Códigos de Corrección de Errores

Si se tiene un canal con probabilidad p de error
A los efectos de aumentar la confiabilidad se decide repetir 3 veces cada bit:



¿Cuál podría ser la regla de decisión para asignar a cada posible palabra de salida la palabra enviada?
¿Cuál es la probabilidad de que no existan errores, de que se cometa uno y solo un error, dos o tres errores?
¿Cuál es la probabilidad de interpretar erróneamente el mensaje a la salida?

Códigos de Corrección de Errores

Si tomamos como referencia la tasa de información transmitida en el caso en que la información se codifica en sólo un bit.
¿Cuál es la tasa de información a la que se transfiere en este caso?

Analice las mismas preguntas para el caso que se utilizan en lugar de repetir 3 bits, se repiten 5 o 7 bits.

Generalizando: ¿se puede de esta forma hacer tan pequeño como se desee la probabilidad de error en la decodificación? ¿a qué costo?

Códigos de Corrección de Errores

Del ejemplo anterior surge la pregunta, dada una tasa de error ¿es posible conseguir códigos más eficientes que la simple repetición?

La respuesta la dio Shannon en la década del 50 y es que si y además nos dice en su segundo teorema cuanto mejor podemos hacerlo.

El segundo teorema de Shannon dice que: Para cualquier tasa de codificación menor que la capacidad del canal C, existen códigos tales que la probabilidad de mensaje erróneo sea tan pequeña como se desee.

Veremos un ejemplo de como construir códigos más eficientes que los anteriores

Códigos de Corrección de Errores

En este ejemplo fijaremos el orden de la extensión del alfabeto (3,5,7 en el ejemplo anterior) y variaremos el número de palabras utilizadas.

Tomemos el caso de 3 bits.

- Si utilizo solo las secuencias 000 y 111 como palabras de entrada ya vimos cual era la probabilidad de error y la tasa de transmisión de información.
- Si utilizo las 8 posibles palabras ¿cual es la probabilidad de error? ¿cuál es la tasa de transmisión de información?

Códigos de Corrección de Errores

Si utilizo las palabras: 000, 011, 101, 110.
¿cuál sería la "mejor" forma de decodificación?

Códigos de Corrección de Errores

Si utilizo las palabras: 000, 011, 101, 110.
¿cuál sería la "mejor" forma de decodificación?

- 000 y 001 como 000
- 010 y 011 como 011
- 101 y 100 como 101
- 110 y 111 como 110

¿cuál es la probabilidad de error? ¿cuál es la tasa de transmisión de información?

Códigos de Corrección de Errores

Si utilizo las palabras: 000, 011, 101, 110.
¿cuál sería la "mejor" forma de decodificación?

- 000 y 001 como 000
- 010 y 011 como 011
- 101 y 100 como 101
- 110 y 111 como 110

¿cuál es la probabilidad de error? ¿cuál es la tasa de transmisión de información?

¿Por qué conviene tomar estas 4 palabras y no por ejemplo: 000, 001, 010, 100?

Códigos de Corrección de Errores

- Si en lugar de 3 se usan N y se eligen M palabras de las 2^N. ¿cómo varía la probabilidad de error al variar M?
- ¿influye la forma en que elijo esas M palabras en la probabilidad de error? ¿cuál sería la mejor forma de elegirlas?
- En lo que sigue veremos algunas formas posibles de seleccionar las palabras de código

Códigos de Corrección de Errores

Transmisor:

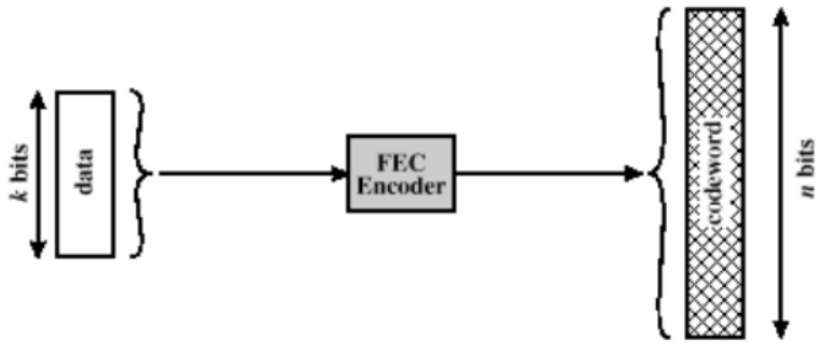
- Códigos de bloques: un codificador con corrección de errores (FEC) mapea cada bloque de k-bits en bloques de n-bits (palabra de código). Se habla entonces de códigos(n,k).
- Las palabras del código (n-bits) son transmitidas.

Receptor:

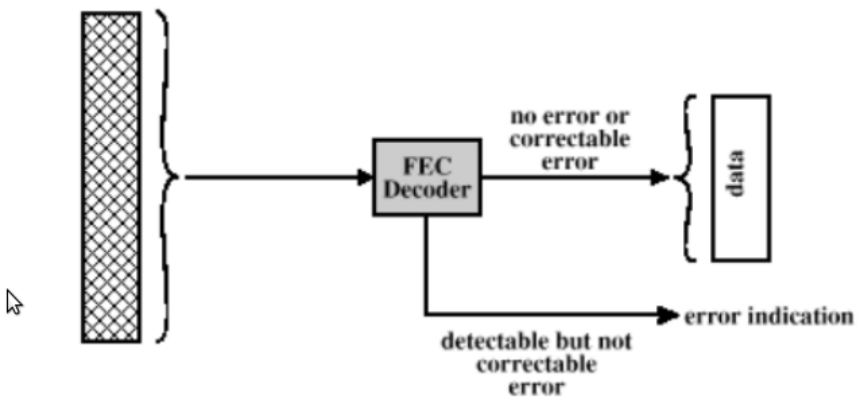
- la señal recibida es demodulada
- El bloque es pasado a través de un decodificador FEC.

Nota:

- Cuanto más bits de paridad más cantidad de errores puedo detectar/corregir pero más ancho de banda es requerido.
- Un objetivo es elegir los bits de paridad de forma de detectar la mayor cantidad de errores y teniendo una eficiencia en términos de ancho de banda lo mejor posible.



(a) Sender



(b) Receiver

Posibles salidas del decodificador FEC

- No hay errores presentes
- El decodificador detecta y corrige errores
- El decodificador detecta pero no puede corregir errores
- El decodificador no detecta errores pero sin embargo hay errores

Códigos de bloques - Ideas y definiciones

- Distancia de Hamming H: para dos secuencias de n-bit es el número de bits diferentes; ej: v1=011011, v2=110001, H(v1,v2)=3.
- Redundancia: relación entre bits redundantes y bits de datos $(n-k)/k$
- Tasa de codificación: cociente entre bits de datos y bits totales k/n
- Mínima distancia de Hamming de un código (d): Es la menor distancia entre dos palabras cualesquiera.
- Si un código tiene distancia d:
  - capacidad de detección de errores: hasta $(d-1)$
  - capacidad de corrección de errores: hasta $(d-1)/2$, cuando corrijo mapeo a la palabra válida con menor H.
  - ¿por qué?

Códigos de bloques - más Ideas y definiciones

- Código lineal: un código es lineal si la suma (módulo 2 sin acarreo) de dos palabras de código es también una palabra de código.
- Código sistemático: $n - k$ bits de paridad, k bits de información

obs: todo código lineal puede escribirse de forma sistemática
Ej: 000,101,011 no es un código lineal

Codificador lineal - matriz generadora

- m es el vector de bits de información que se desea transmitir. $m = [m_0 m_1 \dots m_{k-1}]$
- c es la palabra correspondiente codificada $c = [c_0 c_1 \dots c_{n-1}]$
- al ser un código lineal, c se obtiene a partir de m como una combinación lineal.
- $c = m * G$, $c = m * g_0 + m * g_1 + \dots + m * g_{k-1}$
- G es la llamada matriz generadora del código, G es $k * n$ y de rango k. g_i representa la i-ésima fila de G (las filas de G son linealmente independientes ya que se asume de rango k).
- obs: La matriz G no es única

Ejemplo:

Código de Hamming (7,4)

La ecuación de codificación para este código es:

$c_0 = m_0$, $c_1 = m_1$, $c_2 = m_2$, $c_3 = m_3$, $c_4 = m_0 + m_1 + m_2$,
$c_5 = m_1 + m_2 + m_3$, $c_6 = m_0 + m_1 + m_3$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Código Sistemático

Un código lineal es sistemático si su matriz generadora G, puede escribirse: $G = [I_k|P]$, donde:

- I_k es la matriz identidad de orden k
- P es una matriz $k * (n - k)$ que determina los bits de paridad

Matriz de Chequeo de Paridad

Para $G = [I_k|P_{k,n-k}]$, se define H la matriz de chequeo de paridad como $H = [P^T_{n-k,k}|I_{n-k}]$ (el tamaño de H es $(n-k) * n$. Se puede ver que:

- $GH^T = 0$
- como $c = mG$, entonces $cH^T = mGH^T = 0$, o sea, las palabras del código verifican la ecuación anterior.

Codificación usando la matriz H

$$[c1\ c2\ c3\ c4\ c5\ c6\ c7] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = 0$$

$$c1 + c4 + c6 + c7 = 0$$
$$c2 + c4 + c5 + c6 = 0$$
$$c3 + c5 + c6 + c7 = 0$$

Decodificación

- Sea c la secuencia transmitida y r la recibida.
- $r = c + e$, siendo e el patrón de error
- si de alguna forma puedo determinar el patrón de error, se puede decodificar: $c = r + e$

Decodificación - ejemplos

En el caso del código (7,4):

- Sea 1101000 lo transmitido y 1100000 lo recibido. Entonces: e = 0001000 (un error en la cuarta posición)
- Sea r = 1110100. Qué se transmitió?

c	e
0110100	1000000
1101000	0011100
1011100	0101000

Decodificación - ejemplos

En el caso del código (7,4):

- Sea 1101000 lo transmitido y 1100000 lo recibido. Entonces: e = 0001000 (un error en la cuarta posición)
- Sea r = 1110100. Qué se transmitió?

c	e
0110100	1000000
1101000	0011100
1011100	0101000

el primer escenario es el más probable

El Síndrome

- Se define síndrome $s = rH^T = (c + e)H^T = eH^T$
- El síndrome depende solo del patrón de error y no de la palabra transmitida.
- Cada patrón de error tiene un síndrome asociado. Cada error de un solo bit corresponde a una columna de H^T
- En general $s = eH^T$ son $n - k$ ecuaciones y n incógnitas. Hay $2k$ patrones de error que dan el mismo síndrome. En general se asume que el patrón mas probable es aquel con mínimo peso.

El Síndrome- procedimiento de decodificación

- Para el vector r recibido, calcular el síndrome $s = rH^T$
- Identificar el patrón de error e
- Agregar e a r para recuperar la palabra transmitida c

Código de Hamming

- Diseñado para corregir errores de un solo bit
- Familia de (n, k) códigos de bloque con parámetro m:
  - largo del bloque: $n = 2^m - 1$
  - número de bits de datos: $2^m - m - 1$
  - número de bits de chequeo: $n - k = m$
  - distancia mínima: $d_{min} = 3$
  - La matriz H tiene en sus columnas todos los posibles vectores binarios de largo m, menos el nulo.
- ej: Hamming (7,4)

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Código Cíclico

- Un código C(n,k) es llamado código cíclico si cada desplazamiento de un vector del código C es también un vector del código C.
  O sea, si Si $c = [c_0, \ldots, c_{n-1}]$ es palabra de código entonces: $c = [c_{n-1}, c_0, \ldots, c_{n-2}]$ es también palabra de código.
- ejemplos: Hamming codes, Golay Codes, BCH codes, RS codes

Códigos Convolucionales

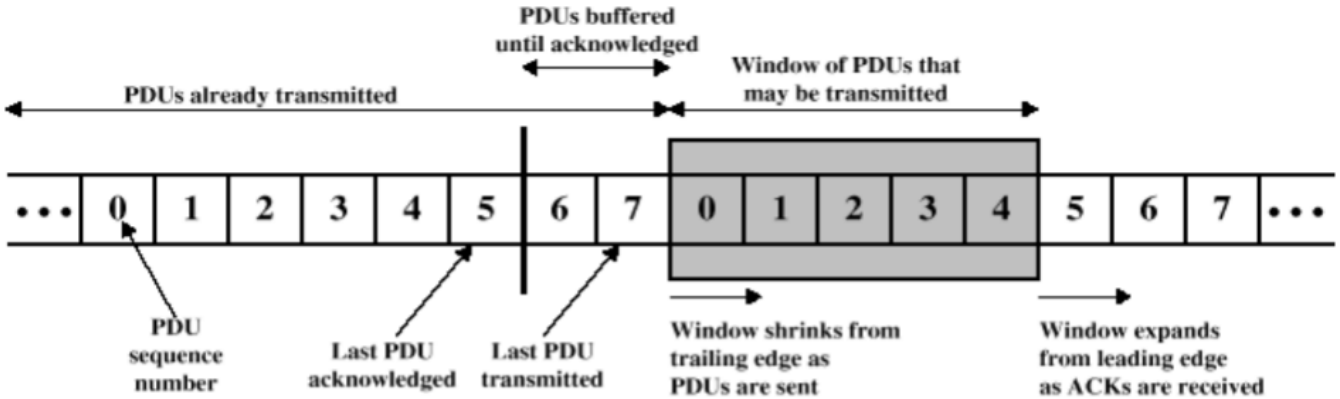
(opcional)

Agenda

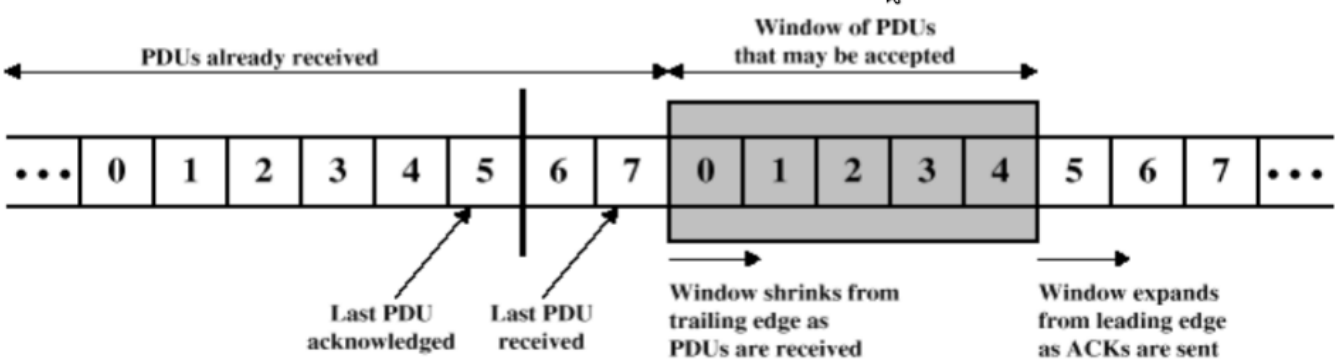
Control de Flujo

- Asegura que el nodo que transmite no sobrecarga con datos al nodo que recibe. La idea es que el receptor necesita tiempo para el procesamiento de los datos recibidos y la idea de estos mecanismos es asegurar que el el receptor no se desborde de tramas para procesar.
- Protocolos con mecanismos de control de flujo habilitan a que existan múltiples PDUs en tránsito.
- Las PDU (protocol data unit) arriban en el mismo orden en que fueron enviadas
- Técnicas: parada y espera o ventana deslizante (sliding window)

Ventana Deslizante



(a) Sender's perspective



(b) Receiver's perspective

Automatic Repeat Request

- Es un mecanismo utilizado en la capa de enlace
- Se basa en algún mecanismo de detección de errores (como CRC), ¿qué pasa cuando no puedo corregir los errores?

Automatic Repeat Request

- Es un mecanismo utilizado en la capa de enlace
- Se basa en algún mecanismo de detección de errores (como CRC), ¿qué pasa cuando no puedo corregir los errores? Se debe pedir la retransmisión de dichos datos. Cuando esta retransmisión se hace automática es lo que se conoce como ARQ. Este pedido se hace con el envío de ACKs y NAKs.

Automatic Repeat Request

- Es un mecanismo utilizado en la capa de enlace
- Se basa en algún mecanismo de detección de errores (como CRC), ¿qué pasa cuando no puedo corregir los errores? Se debe pedir la retransmisión de dichos datos. Cuando esta retransmisión se hace automática es lo que se conoce como ARQ. Este pedido se hace con el envío de ACKs y NAKs.
- Hay varias variantes:
  - Stop & Wait
  - Go back N
  - Selective Repeat

Stop & Wait ARQ

Una vez que se envía una trama, no se envía la siguiente hasta que se recibe el ACK correspondiente.
Asegura que las tramas no se pierden y que llegan en el orden correcto.

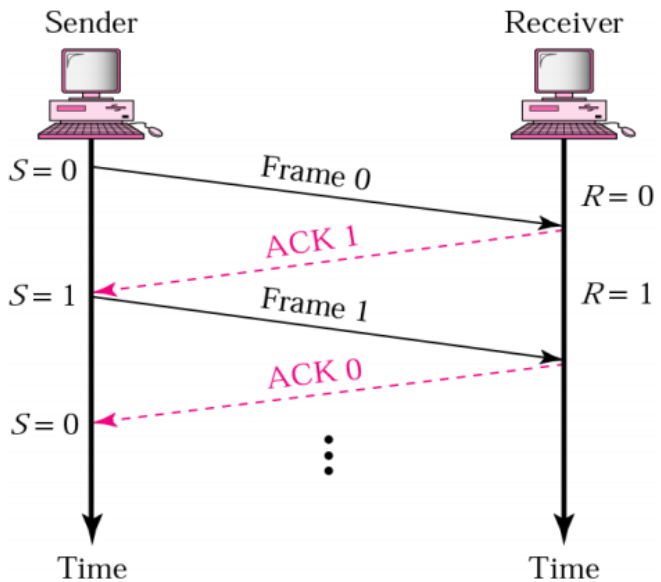


Figura: Operación Normal, fuente: [1]

Go back N ARQ

- Cuando la calidad del enlace es buena, Stop & Wait ARQ pasa a ser ineficiente (puedo enviar una sóla trama por vez).
- Se introduce el concepto de "sliding window"
- El receptor envía un ACK por cada trama recibida en correcto orden

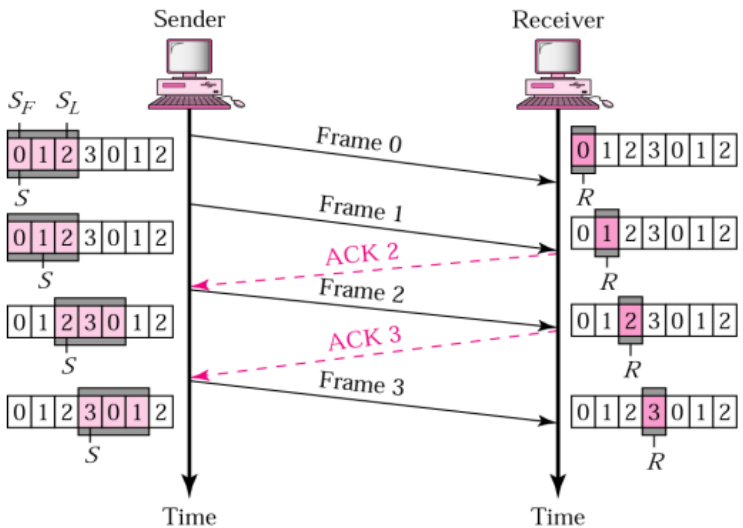


Figura: Operación Normal, fuente: [1]

Selective Repeat ARQ

- En general en canal es ruidoso y se generan pérdidas (y/o errores que no se pueden corregir) con frecuencia.
- Go back N ARQ ineficiente
- La idea de este método (que es producto de la evolución de los anteriores) es enviar sólo las tramas "perdidas".
- Para ello se utiliza el NACK (reconocimiento negativo)

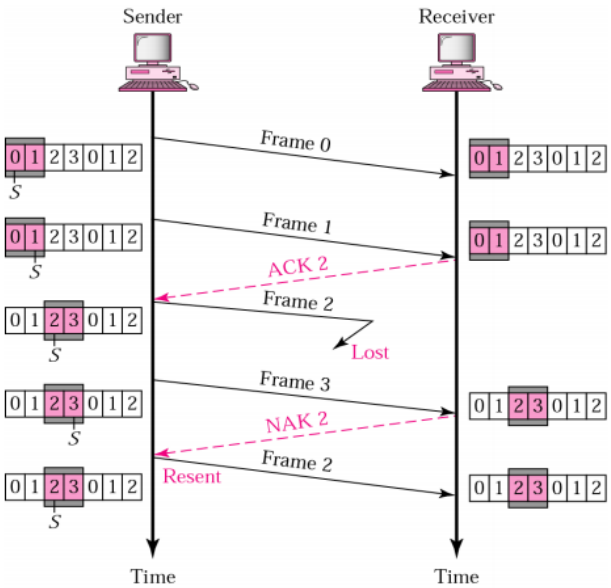


Figura: Operación Normal con pérdida de una trama, fuente: [1]

recapitulando...

- Tenemos mecanismos para detectar y/o corregir errores de transmisión
- Los errores pueden ser:
  - se pierde una PDU
  - se daña una PDU
- Existen técnicas para que se realice una retransmisión en los casos que se necesite:
  - Timeouts
  - Reconocimientos (ACK)
  - Reconocimientos negativos (NACK)

  obs: Los protocolos ARQ permiten controlar la transmisión y el flujo, es decir, evitan que un receptor lento sea saturado.

Referencias

1. Behrouz A. Forouzan, “Data Communications and Networking”, (third edition), published by McGraw-Hill Education, 2004.
2. Go back N arq, demo online: https://www.youtube.com/watch?v=9BuaeEjIeQI
3. Selective Repeat ARQ, demo online: https://www.youtube.com/watch?v=Cs8tR8A9jm8
4. Redes de Computadoras, Andrew S. Tanenbaum, 4ta edición
5. Hamming Code: https://www.youtube.com/watch?v=V8qLNcJGQzo

¿Preguntas?