

Unidad 4: Nivel de enlace (punto a punto)

Medidas de eficiencia

Ancho de banda y latencia

- El **ancho de banda** (o **throughput**) es la cantidad de bits que se pueden transmitir en una red en determinado período de tiempo. Ejemplo: 10Mbps
- La **latencia** es cuánto demora un mensaje en llegar de un extremo al otro. Se mide en unidades de tiempo.
- El **roundtrip time** (*RTT*) es cuánto tarda un mensaje en ir y volver.
- Latency = Propagation + Transmit + Queue
- Propagation = Distance / SpeedOfLight
- Transmit = Size / Bandwidth

Producto delay × ancho de banda

Se puede pensar el canal de comunicación como un tubo, donde la longitud es el delay y el diámetro el ancho de banda. En este caso el producto de ambas métricas nos da el volumen, que se corresponde con la cantidad de bits presentes en el canal en un determinado momento.

Esta métrica es útil porque da la idea de cuántos bits se deben transmitir hasta que el receptor comienza a recibir información. A su vez, el doble de esta magnitud es lo que se puede llegar a enviar hasta que se tiene una primera respuesta del receptor.

Framing

En la capa física se resuelve el problema de transmitir secuencias de bits a través de un medio. Si queremos implementar una red de switcheo de paquetes, el siguiente problema que nos encontramos es cómo darnos cuenta cuándo empiezan y terminan los frames.

Protocolos orientados a byte

Se entiende el flujo como una secuencia de bytes, y se interpretan utilizando algún charset predefinido (ASCII, por ejemplo).

Sentinel

- utilizar marcas de inicio y fin de la sección de datos del frame
- apariciones del marcador de fin de datos se escapa cuando aparece dentro del contenido
- se incluye un CRC (ver métodos de detección de errores) para el frame

- se incluyen campos extra para lograr confiabilidad

Byte count

- en vez de utilizar marcas de inicio y fin, se puede declarar el tamaño del paquete al inicio del mismo
- se introduce *framing error* si se corrompe el campo de tamaño.

Protocolos orientados a bit

- Utilizan una secuencia especial que indica el inicio de un frame.
- A su vez, esta secuencia se envía constantemente cuando no se usa el canal para mantener emisor y receptor sincronizados.
- Se utilizan técnicas de *bit stuffing* para escapar apariciones dentro de los datos de la secuencia especial.

Detección de errores

El mecanismo fundamental para detectar errores es agregar un pequeño número de bits redundantes a cada paquete, que se calculan a través de un algoritmo definido y permiten que el receptor de un paquete pueda verificar con un alto nivel de certeza la ausencia de errores de bits (tras recalcular utilizando el mismo algoritmo).

Bit de paridad bidimensional

- Agregar un bit al final indicando si la cantidad de 1s en el resto del mensaje es par o impar.
- Agregar un byte extra de paridad, donde cada posición indica paridad entre todos los bits de esa posición.
- Protege contra todos los errores de 1, 2 y 3 bits.
- Protege contra la mayoría de los errores de 4 bits.

Internet checksum

- Sumar el contenido de todas las words de un paquete y usar la suma como código de verificación
- No se usa directamente en nivel de enlace
- Se suele usar por encima de otros mecanismos más efectivos

CRC (cyclic redundancy check)

- Interpreta frames como polinomios con coeficientes 0 y 1
- Utiliza propiedades aritméticas de divisibilidad de polinomios
- Se pueden elegir distintos polinomios divisores para detectar distintos tipos de errores
- Fácil de implementar

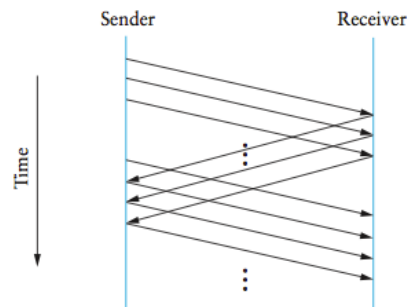
Transmisión confiable

Al detectar errores no corregibles en frames se querrá forzar el reenvío de dichos frames. A su vez, se debería contar con un mecanismo para reenviar mensajes que nunca llegaron.

Stop and wait

- Emisor espera el ACK de cada frame antes de enviar el siguiente
- Si el ACK no llega (o se demora) retransmite
- En caso de que un ACK se pierda o llegue tarde, se retransmite. Para evitar que el receptor confunda el frame retransmitido con el siguiente, se utiliza un número de secuencia de 1 bit.
- Problema: no se utiliza al máximo el enlace.

Sliding window



Emisor:

- Genera números de secuencia consecutivos para los mensajes
- *SWS* (send window size): cantidad máxima de mensajes sin ACK
- *LAR* (last acknowledgement received)
- *LFS* (last frame sent)
- Invariante: $LFS - LAR \leq SWS$

Receptor:

- *RWS* (receiver window size): cantidad máxima de mensajes fuera de orden que se aceptan
- *LAF* (largest acceptable frame)
- *LFR* (largest frame received)
- Invariante: $LAF - LFR \leq RWS$

Propiedades:

- En ausencia de errores aprovecha al máximo la capacidad.
- El emisor tiene un mecanismo de timeout para retransmitir.
- *SWS* se ajusta dependiendo de la capacidad del enlace.
- *RWS* indica cuánto está dispuesto a guardar en buffer el receptor.

Utilidad:

- Problema de pérdida de frames
- Problema de orden incorrecto de frames
- Problema de control de flujo: el receptor puede limitar el ritmo del emisor

Extensión: selective ACKs

- En caso de que el primer frame de un bloque grande no llegue, el receptor no puede enviar ACK del resto

- Esto fuerza a retransmitir innecesariamente muchos frames que el receptor ya tiene
- Si se utilizan **ACK selectivos**, el receptor puede enviar confirmación de bloques de frames ya recibidos para evitar este problema.