

## UNIDAD 3

# PROTOSCOLOS DE TRANSPORTE Y RUTEO DINÁMICO

EL5107 – Tecnología de Información y Comunicación

# CONTENIDOS DE LA UNIDAD

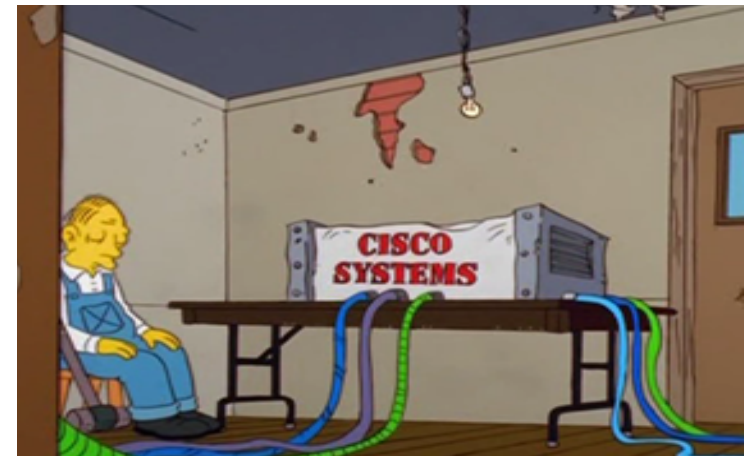
## Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Protocolos y Encabezados End-to-End
- ⊙ UDP
- ⊙ Corrección de Errores
- ⊙ TCP
- ⊙ Anycast & Multicast
- ⊙ Ruteo Interno
- ⊙ Ruteo Externo
- ⊙ Seguridad
- ⊙ DNS



# PROTOCOLOS Y ENCABEZADOS END-TO- END

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

- ⊙ End-to-End Argument.
- ⊙ Transferencia de Archivos.

**EL5107**  
*Tecnologías de  
Información y  
Comunicación*



# END-TO-END ARGUMENT (1)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

#### 3.2. UDP

#### 3.3. Corrección de Errores

#### 3.4. TCP

#### 3.5. Anycast & Multicast

#### 3.6. Ruteo Interno

#### 3.7. Ruteo Externo

#### 3.8. Seguridad

#### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Como se dijo anteriormente uno de los principios básicos del protocolo IP es realizar la inteligencia y el manejo de las conexiones en los extremos de la red de comunicaciones.
  - ⊙ Por eso se dice que las redes TCP/IP son redes “tontas”.
- ⊙ Lo anterior es lo que se conoce como *End-to-End Argument*.
- ⊙ La primera vez que se introdujo este concepto fue en un artículo de 1981 de Saltzter, Reed y Clark del MIT.

# END-TO-END ARGUMENT (2)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

### 3.2. UDP

### 3.3. Corrección de Errores

### 3.4. TCP

### 3.5. Anycast & Multicast

### 3.6. Ruteo Interno

### 3.7. Ruteo Externo

### 3.8. Seguridad

### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ En la práctica se traducen en que se trata de minimizar la intervención de los routers en los encabezados de los paquetes, dejando todo el procesamiento posible en los nodos extremos de la comunicación (el cliente y el servidor).
- ⊙ Algunos casos en que los routers intervienen en los encabezados de los datagramas son:
  - ⊙ Al realizar fragmentación de un datagrama.
    - ⊙ Aquí además de dividir los paquetes es necesario recalculer el Checksum de los paquetes generados.
  - ⊙ Al recalculer el TTL (o HopLimit en IPv6).

# END-TO-END ARGUMENT (3)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Al implementar un sistema que cuente con un subsistema de comunicaciones se trata de definir una interface fija entre éste y el resto del sistema.
  
- ⊙ Al realizar lo anterior existen funciones que pueden ser implementadas de muchas formas distintas:
  - ⊙ Por el subsistema de comunicaciones.
  
  - ⊙ Por la aplicación subyacente.
  
  - ⊙ Entre ambos niveles.

# END-TO-END ARGUMENT

## (4)

### Transporte y Ruteo Dinámico

#### 3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Para orientar la decisión de dónde llevar a cabo la implementación de la función se formuló el siguiente argumento:
  - ⊙ *Una función puede ser implementada correctamente sólo con el conocimiento y ayuda de la aplicación que se encuentre en los extremos del sistema de comunicación. Por lo tanto, proveer dicha función como una característica del sistema de comunicación en sí mismo no es posible.*

# TRANSFERENCIA DE ARCHIVOS (1)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

### 3.2. UDP

### 3.3. Corrección de Errores

### 3.4. TCP

### 3.5. Anycast & Multicast

### 3.6. Ruteo Interno

### 3.7. Ruteo Externo

### 3.8. Seguridad

### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Para ilustrar lo anterior se considerará un ejemplo clásico relacionado con la transferencia de archivos.
- ⊙ Considérese dos computadores (A y B) conectados por un sistema de comunicación.
- ⊙ Para copiar un archivo de A a B es necesario:
  - ⊙ Leer el archivo desde el disco duro de A.
  - ⊙ Copiar el archivo a la memoria RAM y entregárselo al sistema de comunicación.
  - ⊙ El sistema de comunicación de A debe enviar el archivo al sistema de comunicación de B.
  - ⊙ B debe copiar el archivo recibido a memoria RAM y finalmente escribir el archivo en el disco duro.



# TRANSFERENCIA DE ARCHIVOS (2)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

### 3.2. UDP

### 3.3. Corrección de Errores

### 3.4. TCP

### 3.5. Anycast & Multicast

### 3.6. Ruteo Interno

### 3.7. Ruteo Externo

### 3.8. Seguridad

### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ En cada una de las actividades necesarias para copiar un archivo desde A a B pueden ocurrir fallas a nivel del hardware o software de bajo nivel.
- ⊙ Si se quisiera implementar una aplicación de transferencia segura de archivos habría que lidiar con todas estas posibles fallas.
- ⊙ Una aproximación al problema anterior sería revisar en cada etapa la integridad de la información.
  - ⊙ Esto sin embargo puede resultar tremendamente ineficiente.
- ⊙ Otra aproximación sería añadir un checksum al archivo y verificarlo una vez que ya está copiado en el disco duro de la máquina B. Si la verificación del checksum falla simplemente se reenvía el archivo.

# TRANSFERENCIA DE ARCHIVOS (3)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

### 3.2. UDP

### 3.3. Corrección de Errores

### 3.4. TCP

### 3.5. Anycast & Multicast

### 3.6. Ruteo Interno

### 3.7. Ruteo Externo

### 3.8. Seguridad

### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Lo anterior soluciona el problema de manera sencilla y probablemente casi siempre la transferencia de archivos se logrará exitosamente al primer intento.
- ⊙ A veces se necesitarán dos o tres intentos para enviar el archivo correctamente. De ser necesario más intentos se podría sospechar fuertemente que existe alguna falla en el sistema.
- ⊙ Alguien podría querer implementar este checksum como parte del protocolo de comunicaciones. Sin embargo si se hace esto no se eliminan los riesgos relacionados a las fallas que puedan ocurrir fuera del alcance de éste (por ejemplo al escribir el archivo en el disco duro).

# TRANSFERENCIA DE ARCHIVOS (4)

## Transporte y Ruteo Dinámico

### 3.1. Protocolos End-to-End

#### 3.2. UDP

#### 3.3. Corrección de Errores

#### 3.4. TCP

#### 3.5. Anycast & Multicast

#### 3.6. Ruteo Interno

#### 3.7. Ruteo Externo

#### 3.8. Seguridad

#### 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ De ahí el argumento, para que la aplicación tenga una transmisión confiable de archivos tiene que proporcionar un método específico para garantizar la integridad del archivo de extremo a extremo.
- ⊙ Que el sistema de comunicaciones además implemente la misma función tiene poco sentido ya que no se libera a la aplicación de realizar el chequeo por si misma.

# INTRODUCCIÓN (1)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Las aplicaciones esperan ciertas características del servicio entregado por la capa de transporte, como:
  - ⊙ Despacho garantizado de mensajes.
  - ⊙ Despacho de los mensajes en el orden original.
  - ⊙ Despacho de a lo más una copia de cada mensaje.
  - ⊙ Envío de mensajes de largo arbitrario.

# INTRODUCCIÓN (2)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107

Tecnologías de  
Información y  
Comunicación



- ⊙ Las aplicaciones esperan ciertas características del servicio entregado por la capa de transporte, como:
  - ⊙ Sincronización entre emisor y receptor.
  - ⊙ El receptor debe ser capaz de aplicar control de flujo al emisor.
  - ⊙ Múltiples aplicaciones por host.

# INTRODUCCIÓN (3)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Existe dos líneas principales de servicios entregados por la capa de transporte (en el modelo de Internet)
  - ⊙ TCP: Transmission Control Protocol
    - Orientado al flujo confiable de bytes
    - Provee control de flujo, control de errores, orden.
  - ⊙ UDP: User Datagram Protocol
    - Orientado principalmente a la mensajería.
    - Es asíncrono.
    - No confiable, sin control de flujo ni orden.

# UDP

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

*EL5107*  
*Tecnologías de*  
*Información y*  
*Comunicación*



- ⊙ Introducción
- ⊙ Protocolo UDP
- ⊙ Multiplexión por Puertos

# PROTOCOLO UDP (1)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Protocolo de transporte muy simple.
- ⊙ Extiende las características de datagramas IP entre hosts a la comunicación entre procesos.
- ⊙ Orientado a los mensajes y a los sistemas que funcionan en base a request/reply. En éstos casos, el establecimiento de una conexión es un costo que no vale la pena asumir.
- ⊙ El control de flujo, orden y recuperación de errores queda en manos de la aplicación.



# PROTOCOLO UDP (2)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

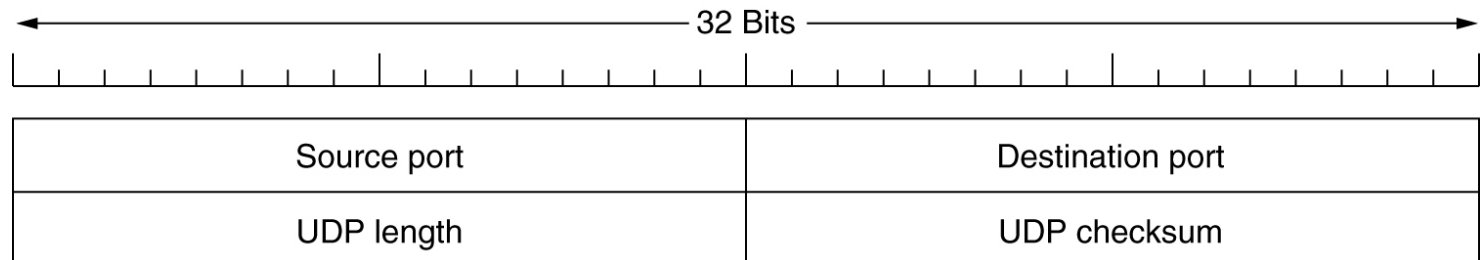
3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ El checksum se calcula sobre el header UDP mas un *pseudoheader*: campos del header IP como Protocolo, IP origen, IP destino y el largo del payload.



# MULTIPLEXIÓN POR PUERTOS (1)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Los servicios de la capa de transporte requieren que el emisor y receptor establezcan un socket. Cada socket se identifica por el par (IP, puerto).
- ⊙ Las conexiones se identifican por los identificadores de los sockets a ambos lados.
- ⊙ Esto permite asociar una aplicación con una conexión, mediante una tabla interna propia del S.O.

PID	Socket
1234	1
4563	2
8890	3

Socket	IP	Puerto
1	10.0.45.3	3535
2	10.0.45.4	5050
3	10.0.45.4	8080

# MULTIPLEXIÓN POR PUERTOS (2)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

### 3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Los puertos son números de 16 bits (por lo que existen 65535 posibilidades).
- ⊙ Se dividen en dos categorías: Puertos bien conocidos (*well-known ports*, aquellos menores al 1024) y puertos no privilegiados (todos los otros)
- ⊙ Los puertos bien conocidos están reservados para los servicios estándares. Así, un cliente conoce exactamente a qué puerto dirigir un requerimiento.
- ⊙ Cada flujo se identifica por la tupla ( $Dir_{origen}, Puerto_{origen}, Dir_{destino}, Puerto_{Destino}, Protocolo$ )

# MULTIPLEXIÓN POR PUERTOS (3)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

## 🎯 Listado de puertos

Servicio	Puerto	Protocolo	Servicio	Puerto	Protocolo
FTP	21	TCP	POP3	110	TCP
SSH	22	TCP	SunRPC	111	TCP y UDP
Telnet	23	TCP	NNTP	119	TCP
SMTP	25	TCP	NTP	123	TCP y UDP
Whois	43	TCP y UDP	SNMP	161	TCP y UDP
DNS	53	TCP y UDP	BGP	179	TCP
HTTP	80	TCP	HTTPS	443	TCP

EL5107

Tecnologías de  
Información y  
Comunicación



# CORRECCIÓN DE ERRORES

## Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores**
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

- ⊙ Transmisión Confiable
- ⊙ Stop-and-Wait
- ⊙ Go Back N
- ⊙ Sliding Windows
- ⊙ Números de Secuencia

EL5107  
Tecnologías de  
Información y  
Comunicación



# TRANSMISIÓN CONFIABLE

## (1)

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Transmisión Confiable se traduce en el mecanismo para asegurar que todo lo que se envía es recibido, en el mismo orden original.
- ⊙ Después de ver los mecanismos para detectar errores, quedó pendiente conocer el procedimiento en caso de que un error fuera detectado.
- ⊙ Cuando un error aparecía, la capa debía descartar el frame o paquete, y por tanto, recuperarse posteriormente de esa situación.

# TRANSMISIÓN CONFIABLE

## (2)

### Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Esto se logra mediante la combinación de los mecanismos: *acknowledgments (ACK)* y *timeout*.
- ⊙ Los ACK son pequeños paquetes de control que un protocolo envía para emisor para notificar la recepción satisfactoria de un paquete previo.
  - ⊙ Estos paquetes de control son sólo el encabezado (*header*) y no contienen datos. Por lo mismo, el receptor puede incluir un ACK en un paquete de datos que vaya en el sentido contrario. A este mecanismo se le conoce como *piggybacking*.

# TRANSMISIÓN CONFIABLE

## (3)

### Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Si después de una cantidad razonable de tiempo el ACK no se ha recibido, el emisor retransmite el paquete que no fue confirmado. El tiempo máximo de espera por un ACK se conoce como timeout.
- ⊙ La estrategia que usa ACK's y timeouts para implementar transmisión confiable es llamado *automatic repeat request* (ARQ).
- ⊙ Revisaremos tres tipos de ARQ.
  - ⊙ Stop-and-Wait
  - ⊙ Go-Back N
  - ⊙ Sliding Window



# STOP-AND-WAIT (1)

## Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



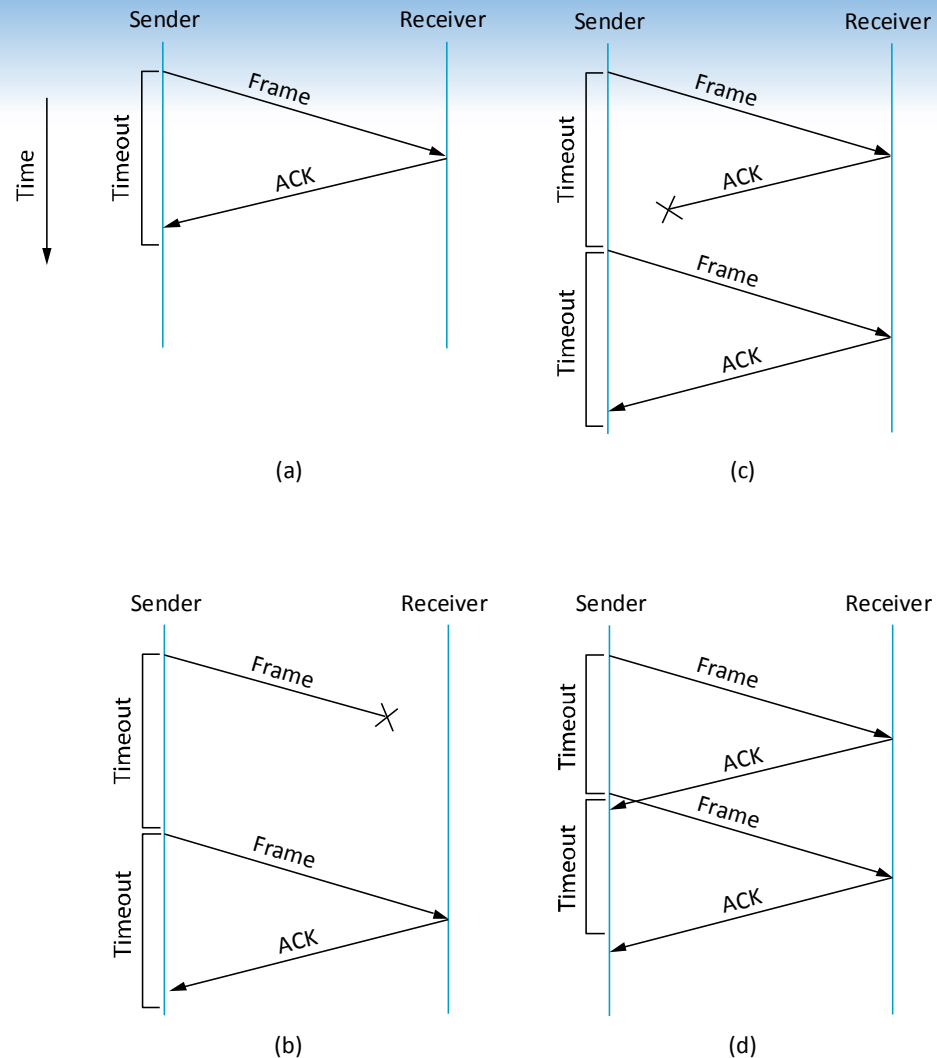
- ⊙ La idea es muy simple: Envío un paquete y me quedo esperando por el ACK respectivo antes de enviar otro paquete.
- ⊙ Si después de cierta espera el ACK no llega, se retransmite.
- ⊙ Revisemos algunos casos que se pueden producir durante la operación de este mecanismo.

# STOP-AND-WAIT (2)

## Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores**
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



# STOP-AND-WAIT (3)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ En (a) se muestra el envío del paquete, y la posterior recepción del ACK.
- ⊙ En (b) se muestra el envío de un paquete y su posterior pérdida, lo que genera una retransmisión correcta.
- ⊙ En (c) se muestra el envío de un paquete, su recepción y generación del ACK, el cual se pierde antes de llegar al emisor, lo que genera una retransmisión incorrecta.
- ⊙ En (d) se muestra un caso parecido a (c), pero el ACK no se pierde, sino que se retrasa. Esto genera una retransmisión incorrecta y un ACK duplicado.

# STOP-AND-WAIT (4)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



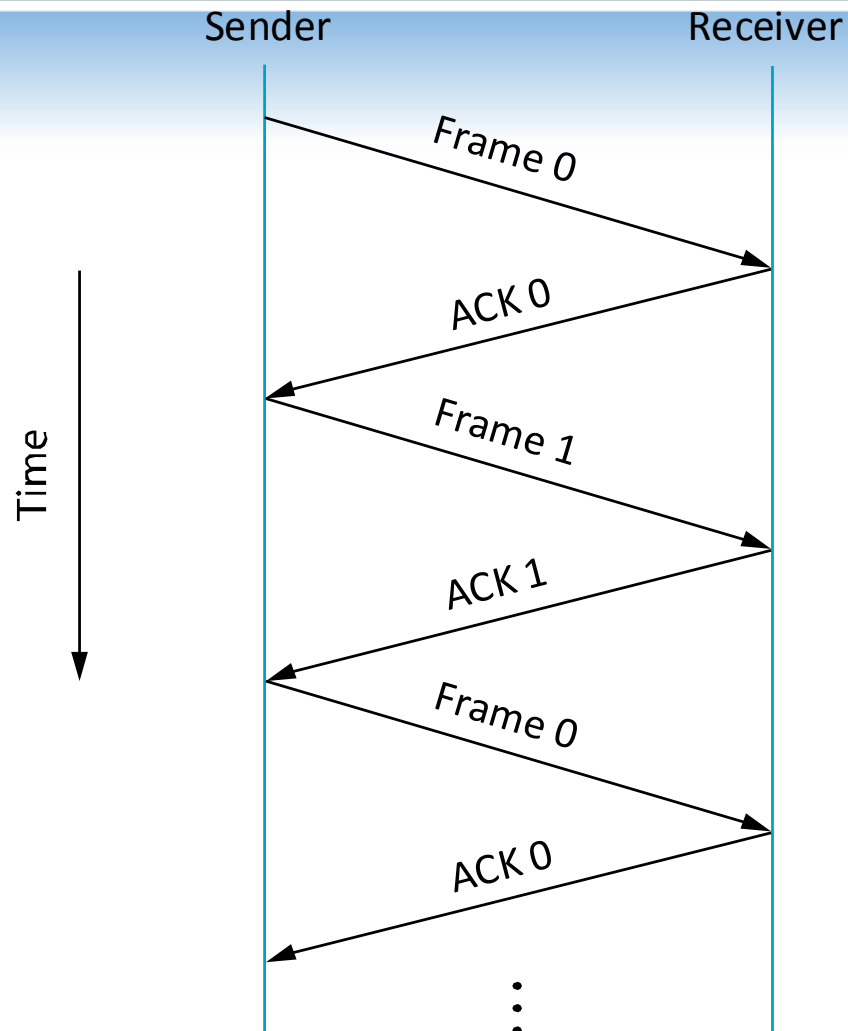
- ⊙ (b), (c) o (d) indistinguibles
- ⊙ En los casos (c) y (d) el emisor reenviará el paquete original, pero el receptor creerá que se trata de un nuevo paquete, lo que generará una duplicación.
- ⊙ En (d) se duplica el ACK, haciendo creer al enviador que se recibió un nuevo paquete OK.
- ⊙ Por ello, es necesario identificar el frame **Y** el ACK, por lo que se utilizan números de secuencia.
- ⊙ Para este caso, basta un número de secuencia de 1 bit.

# STOP-AND-WAIT (5)

## Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores**
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



# STOP-AND-WAIT (6)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ La gran desventaja de Stop-and-Wait es que solo permite que un paquete de datos viaje al mismo tiempo, lo que se puede traducir en un uso ineficiente de la capacidad del enlace.
- ⊙ Ejemplo:
  - ⊙ Suponga un enlace de 1.5 Mbps con un RTT de 45 ms. El BDP (*bandwidth-delay product*) será de 67.5 Kb, aprox. 8KB.
  - ⊙ Si un emisor envía un paquete de 1KB cada RTT, tendremos:

BitsPorPaquete ÷ TiempoPorPaquete

$(1024 \times 8) \div 0.045$

182 Kbps

# STOP-AND-WAIT (7)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ El peor comportamiento lo tiene en *Long Fat Networks (LFN)*
- ⊙ También llamadas “Elefantes”
- ⊙ Ej: enlace satelital de 100Mbps, RTT de 600 ms
- ⊙ BDP: ~6 Mbytes
- ⊙ Es lo que cabría en el enlace hasta que llega el primer ACK
- ⊙ Deberíamos entonces tener 'BDP' buffers de transmisión en espera de ACKs

# SLIDING WINDOW (1)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

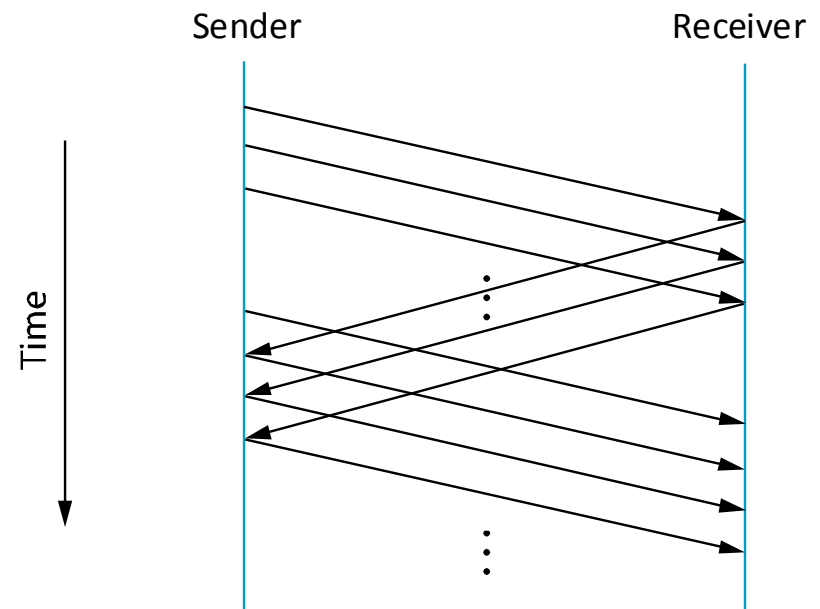
3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ En el caso de BDP 8KB y paquetes de 1KB quisiéramos enviar 8 paquetes a la vez, y poder enviar el noveno en cuanto llegue el ACK del primer paquete.





# SLIDING WINDOW (2)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

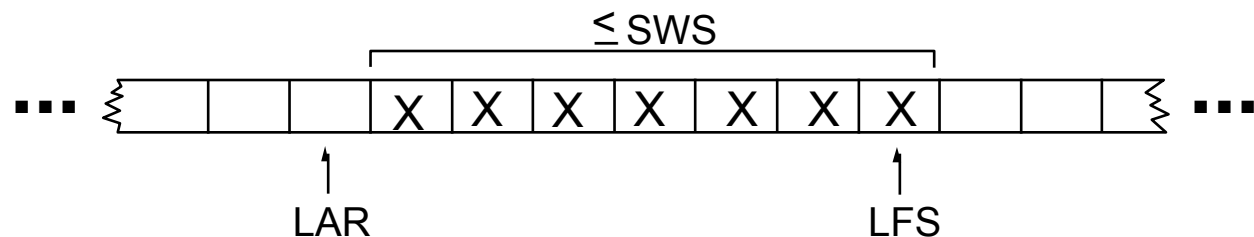
3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- El mecanismo de *Sliding Window* define lo siguientes elementos:
  - El emisor le asigna un *número de secuencia* a cada paquete que envía **SeqNum**.
  - El emisor mantiene tres variables. **SWS**=*send window size* (debería aproximar BDP), que define el límite superior de paquetes sin confirmar que se pueden transmitir; **LAR**=*last ack received*, que indica el número de secuencia del último ACK recibido; **LFS**=*last frame sent*, que indica el número de secuencia del último paquete enviado.
  - El emisor mantiene el invariante:



# SLIDING WINDOW (3)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Cuando se recibe un ACK, el emisor “mueve” LAR a la derecha, lo que permite enviar otro paquete.
- ⊙ El emisor tiene un timer asociado a cada paquete enviado (aprox RTT).
- ⊙ Si expira antes de la recepción del respectivo ACK, debe retransmitir. Esto se traduce en que el emisor debe ser capaz de mantener SWS paquetes en caso de que los necesite para retransmisión.



# SLIDING WINDOW (4)

## Go-back-N

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ El receptor mantiene una ventana de tamaño 1:
  - ⊙  $\text{ExpectedSeq} == \text{paquete esperado}$
  - ⊙ Todo paquete distinto a ese es descartado
  - ⊙ Pero siempre enviamos ACK para  $\text{ExpectedSeq}-1$
- ⊙ El emisor retransmite la ventana completa si hay un timeout (*Go-Back-N*)

# SLIDING WINDOW (4)

## Go-back-N

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Al recibir un ACKn con  $n > LAR+1$ 
  - ⊙ Puedo dar por recibidos OK todos los paquetes enviados desde LAR+1 hasta n (no me habrían enviado este ACK si no los hubieran recibido bien)
  - ⊙ Se define ACKn  $\longrightarrow$  ACKi  $i \leq n$

Al recibir un paquete con número de secuencia  $>$  ExpectedSeq

- Puedo dar por perdidos los paquetes entre medio
- Enviar NACK para ExpectedSeq y apurar la retransmisión

# SLIDING WINDOW (4)

## Go-back-N

### Transporte y Ruteo Dinámico

- 3.1. Protocolos End-to-End
- 3.2. UDP
- 3.3. Corrección de Errores**
- 3.4. TCP
- 3.5. Anycast & Multicast
- 3.6. Ruteo Interno
- 3.7. Ruteo Externo
- 3.8. Seguridad
- 3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Mucho más eficiente que Stop&Wait si la ventana de emisión es suficiente para llenar el BDP de la conexión
- ⊙ Frente a pérdidas: retransmite la ventana completa, problemas si es muy grande y hay muchos errores
- ⊙ En enlaces de alto BDP y alta pérdida necesitamos una mejor solución: aprovechar los paquetes transmitidos OK y no retransmitirlos.



# SLIDING WINDOW (4)

## Selective Repeat

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⊙ Ahora el receptor mantiene ventana también, con tres variables:
  - ⊙ **RWS**=receive window size, que define el límite máximo de paquetes desordenados que puede aceptar.
  - ⊙ **LAF**=largest acceptable frame, que define el número de secuencia más grande que puede aceptar.
  - ⊙ **LFR**=last frame received, que define el número de secuencia del último paquete recibido tal que todos los paquetes con secuencia  $\leq$  LFR ya se recibieron OK.
- ⊙ El receptor mantiene el siguiente invariante:
$$LAF - LFR \leq RWS$$

# SLIDING WINDOW (5)

## Selective Repeat

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

**3.3. Corrección de Errores**

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107

Tecnologías de  
Información y  
Comunicación



- ⊙ Cuando un paquete con número SeqNum llega, el receptor hace lo siguiente:
  - ⊙ Si  $\text{SeqNum} \leq \text{LFR}$  o  $\text{SeqNum} > \text{LAF}$ , se considera fuera de la ventana de recepción y se descarta.
  - ⊙ Si  $\text{LFR} < \text{SeqNum} \leq \text{LAF}$ , entonces el paquete está dentro de la ventana de recepción y es aceptado.
  - ⊙  $\text{SeqNum} \leq \text{LFR}$  es una retransmisión
  - ⊙  $\text{SeqNum} > \text{LAF}$  es un paquete que no puedo recibir ya que mi ventana de recepción es muy pequeña

# SLIDING WINDOW (6)

## Selective Repeat

### Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



- ⦿ Cuando un paquete con número SeqNum llega, el receptor hace lo siguiente (Continuación):
  - ⦿ El receptor envía el ACK para el paquete recibido (SeqNum) si  $\text{SeqNum} \leq \text{LAF}$
  - ⦿ Si  $\text{SeqNum} > \text{LAF}$ , envía ACK para LRF
  - ⦿ Si  $\text{SeqNum} == \text{LRF} + 1$ , ajusta la ventana hasta el próximo paquete no recibido.
  - ⦿ **Ahora un ACKn no implica Acks  $a < n$**



# SLIDING WINDOW (9)

## Transporte y Ruteo Dinámico

3.1. Protocolos End-to-End

3.2. UDP

3.3. Corrección de Errores

3.4. TCP

3.5. Anycast & Multicast

3.6. Ruteo Interno

3.7. Ruteo Externo

3.8. Seguridad

3.9. DNS

EL5107  
Tecnologías de  
Información y  
Comunicación



## ⊙ Notas

- ⊙ El tamaño de la ventana de emisión se calcula en base al BDP.
- ⊙ El tamaño de la ventana de recepción puede ser cualquiera.
  - ⊙ Si  $RWS=1$ , el receptor no aceptará paquetes desordenados.
  - ⊙ Si  $RWS=SWS$ , el receptor puede aceptar cualquier paquete que el emisor envíe.
  - ⊙ Si  $RWS>SWS$ , no tiene mucho sentido, pues no pueden más de  $SWS$  paquetes desordenados.