Tema 3. Nivel de transporte

Introducción a las redes de ordenadores

Boni García

Curso 2017/2018

Índice de contenidos

Índice de contenidos

1. Introducción al nivel de transporte

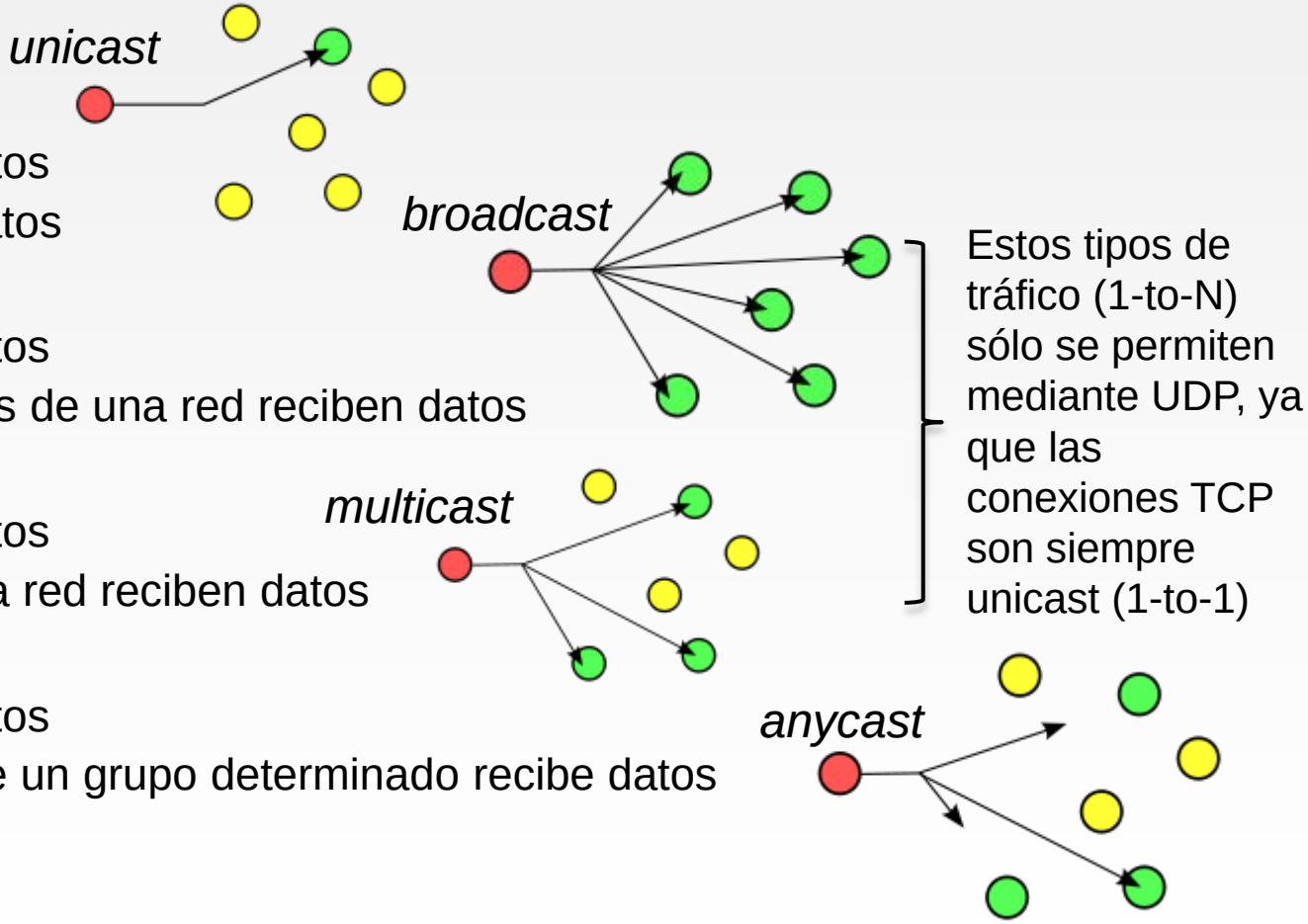
Servicio de transporte

- La capa de transporte proporciona comunicación lógica entre procesos de aplicación que se ejecutan en diferentes equipos terminales
- Ni TCP ni UDP ofrecen cifrado. Para ello se usa TLS
- Direcciones de nivel transporte = Puerto. Numeración lógica que se asigna a las entidades de transporte (16 bits)
  - 0-1023: Puerto de sistema (*well-known ports*). Reservados para servicios distribuidos en Internet (correo electrónico, web, etc). Son usados por procesos de sistema (necesitan permisos de administrador)
  - 1024-49151: Puertos de usuario, de libre utilización
  - 49152-65535: Puertos efímeros, de uso dinámico. Se utilizan sobre todo por los clientes al conectar con servidores
- Lista de puertos conocidos y registrados en IANA: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

1. Introducción al nivel de transporte

Tipos de tráfico

- *Unicast*
  - 1 entidad envía datos
  - 1 entidad recibe datos
- *Broadcast*
  - 1 entidad envía datos
  - Todas las entidades de una red reciben datos
- *Multicast*
  - 1 entidad envía datos
  - N entidades de una red reciben datos
- *Anycast*
  - 1 entidad envía datos
  - 1 entidad dentro de un grupo determinado recibe datos



Índice de contenidos

2. UDP

Servicio proporcionado por UDP

- UDP (*User Datagram Protocol*, definido en la RFC 768) es protocolo de transporte que ofrece un servicio **no orientado a la conexión:**
- UDP no proporciona fiabilidad (no hay confirmación datos, no implementa temporizador de retransmisión, puede haber desorden/duplicación segmentos), no proporciona control de flujo ni control de congestión
- UDP sólo proporciona la capacidad de **direccionamiento** de aplicaciones (procesos) y un mecanismo de integridad de datos basado en la suma de verificación (***checksum***) de la cabecera y la carga útil
- Cualquier tipo de garantías para la transmisión de la información deben ser implementadas en capas superiores
- Se dice que un canal UDP es ***full-dúplex*** (puede existir envío bidireccional simultáneo entre los extremos de la comunicación)

2. UDP

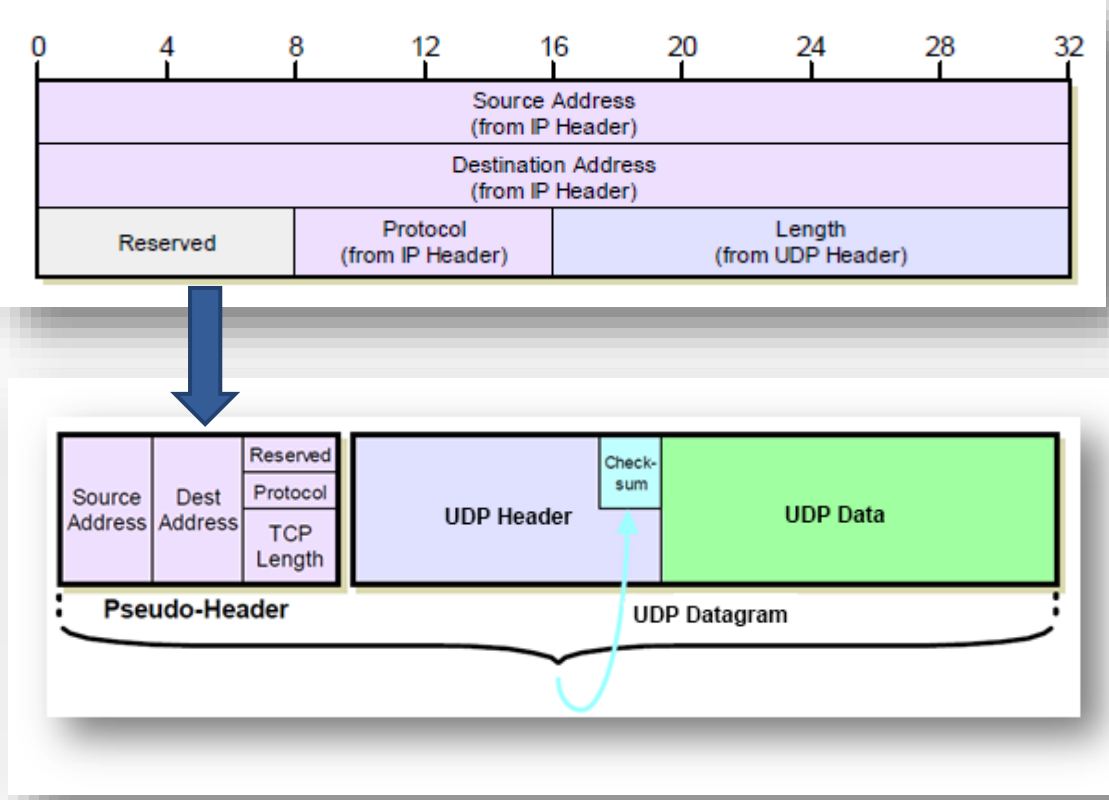
Datagrama UDP

- La PDU de nivel de transporte referido al protocolo UDP de llama **datagrama**. El formato de los datagramas UDP es el siguiente:
  - Puerto origen (opcional): si no se especifica (va a 0) es porque no se requiere respuesta
  - Puerto destino (obligatorio)
  - Longitud (obligatoria): tamaño en bytes del datagrama UDP (cabecera + datos)
  - Suma de verificación o *checksum* (opcional): para el cálculo del checksum se usa la pseudo-cabecera IP que se transporta (se llama así porque no forma parte del segmento, sólo se usa para el cálculo del checksum)

+	Bits 0 - 15	16 - 31
0	Puerto origen	Puerto destino
32	Longitud del Mensaje	Suma de verificación
64	Datos	

2. UDP

Checksum



El checksum es el complemento a 1 (cambiar 1 por 0 y 0 por 1) de la suma del contenido en palabras de 16 bits

Se calcula en origen y después en destino sobre los datos recibidos. Se garantiza la integridad de los datos si el resultado de sumar ambos checksum es igual a todo 1's

La razón del cálculo de checksum de esta forma tiene motivos históricos. En un principio los protocolos TCP/IP eran un único protocolo (TCP) que posteriormente fue dividido en dos

Índice de contenidos

3. TCP

Servicio proporcionado por TCP

- El protocolo TCP (*Transmission Control Protocol*, definido en las RFCs 793 y 1323) ofrece un **servicio orientado a la conexión** (suministro garantizado de los mensajes de la capa de aplicación al destino)
- Proporciona **segmentación** (división de los mensajes largos en segmentos más cortos)
- Ofrece un mecanismo de **control del flujo** (adaptación de las velocidad de transmisión para evitar desbordar al receptor)
- Proporciona un mecanismo de **control de congestión** (adaptación de la velocidad de transmisión si la red está congestionada)
- Se dice que un canal TCP es ***full-dúplex*** (puede existir envío bidireccional simultáneo entre los extremos de la comunicación)

3. TCP

Segmento TCP

- La PDU de nivel de transporte referido al protocolo TCP de llama **segmento**. El formato de los segmentos TCP es el siguiente:

Offsets	Octeto	0								1								2								3							
Octeto	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Puerto de origen																Puerto de destino															
4	32	Número de secuencia																															
8	64	Número de acuse de recibo (si ACK es establecido)																															
12	96	Longitud de Cabecera				Reservado				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Tamaño de Ventana														
16	128	Suma de verificación																Puntero urgente (si URG es establecido)															
20 ...	160 ...	Opciones (Si la Longitud de Cabecera > 5, relleno al final con "0" bytes si es necesario) ...																															

3. TCP

Segmento TCP

- Puerto origen (16 bits) y puerto destino (16 bits)
- Número de secuencia (32 bits). Las conexiones TCP están orientadas a bytes, o sea, se cuentan los datos de aplicación que encapsula. El número de secuencia identifica los bytes enviados por el emisor
- Número de acuse de recibo (32 bits). Número de bytes asentidos por el extremo receptor
- Longitud cabecera (4 bits): En unidades de 4 bytes (32 bits). Por defecto vale el mínimo, 5 (20 bytes). El máximo es 15 (60 bytes)
- Reservados (3 bits): Para usos futuros. Actualmente 000

TCP no proporciona el mecanismo para saber el tamaño de la carga que transporta

Para ello hay que usar los campos del paquete IP (longitud cabecera y longitud de paquete)

3. TCP

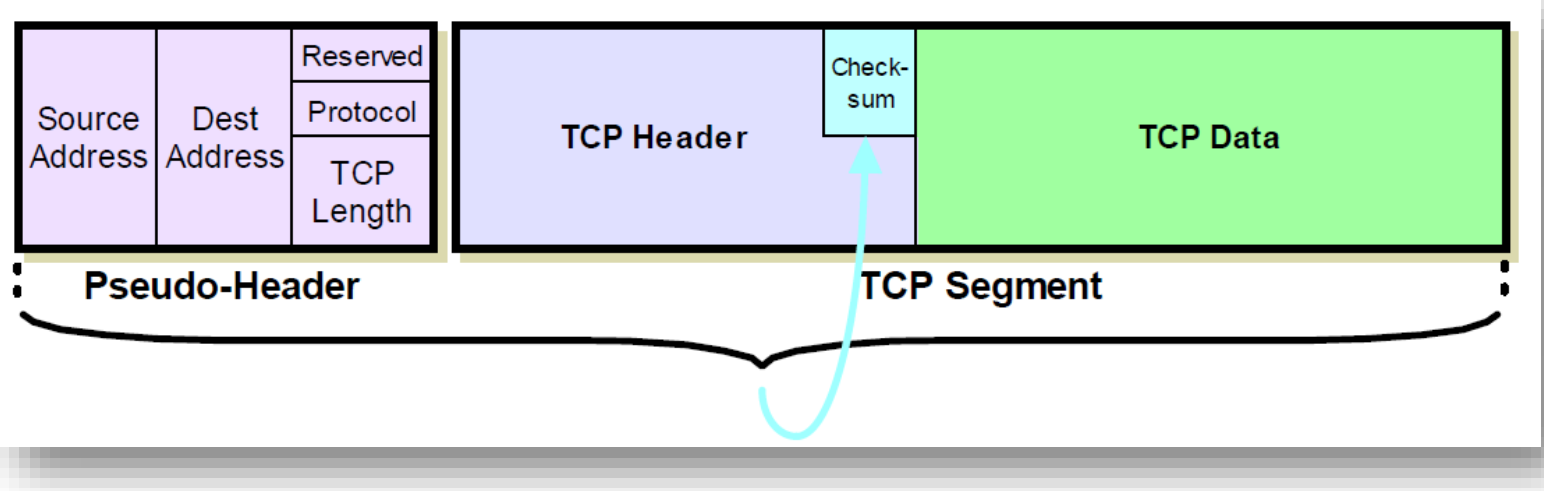
Segmento TCP

- Flags (9 bits) son bits de control que se usan para implementar diferentes funciones del protocolo
- Los flags más importantes son:
  - **SYN** (1 bit): establecimiento de conexión
  - **ACK** (1 bit): asentimiento de datos (campo de nº de acuse de recibo)
  - **FIN** (1 bit): liberación de conexión
- El resto de flags no los vamos a estudiar en clase:
  - ECN (3 bits) *Explicit Congestion Notification* (RFC 3168), NS (1 bit) *Nonce Sum,* CWR (1 bit) *Congestion Window Reduced*, ECE (1 bit) ECN-Echo, URG (1 bit), PSH (1 bit), RST (1 bit)

3. TCP

Segmento TCP

- Tamaño de ventana de recepción (*rwnd*), usada para el control de flujo
- Checksum. Se calcula igual que en UDP (usando la pseudo-cabecera IP) y el segmento TCP con datos:



3. TCP

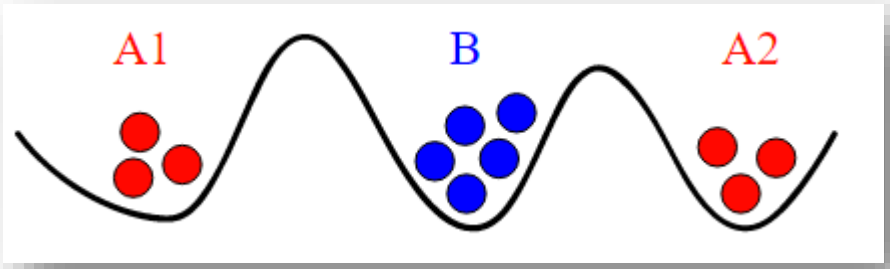
Segmento TCP

- Puntero para datos urgentes. Usado en conjunción con el flag URG, se usa para la transferencia de datos urgentes. El valor de este campo contienen el número de secuencia del último byte de datos urgentes. Por una discrepancia entre las RFC 793 y 1122 el envío de datos urgentes no se usa en la actualidad en Internet
- Opciones (múltiplo de 4 bytes): Funciones extras a TCP. Cada opción puede tener un tamaño fijo de 1byte o ser de tamaño variable. Algunos ejemplos de opciones:
  - Cambio del tamaño máximo de segmento
  - Cambio del tamaño de ventana

3. TCP

Conexión TCP

- TCP está divido en tres fases: **conexión**, **transferencia de datos**, **desconexión**
- Para el establecimiento de conexión TCP es necesario diseñar un mecanismo de coordinación
- Al funcionar TCP siempre sobre IP (canal no fiable), aparece lo que se conoce como "*problema de los dos generales*"



Ejercicios A1 y A2 tienen que fijar una hora para atacar a B

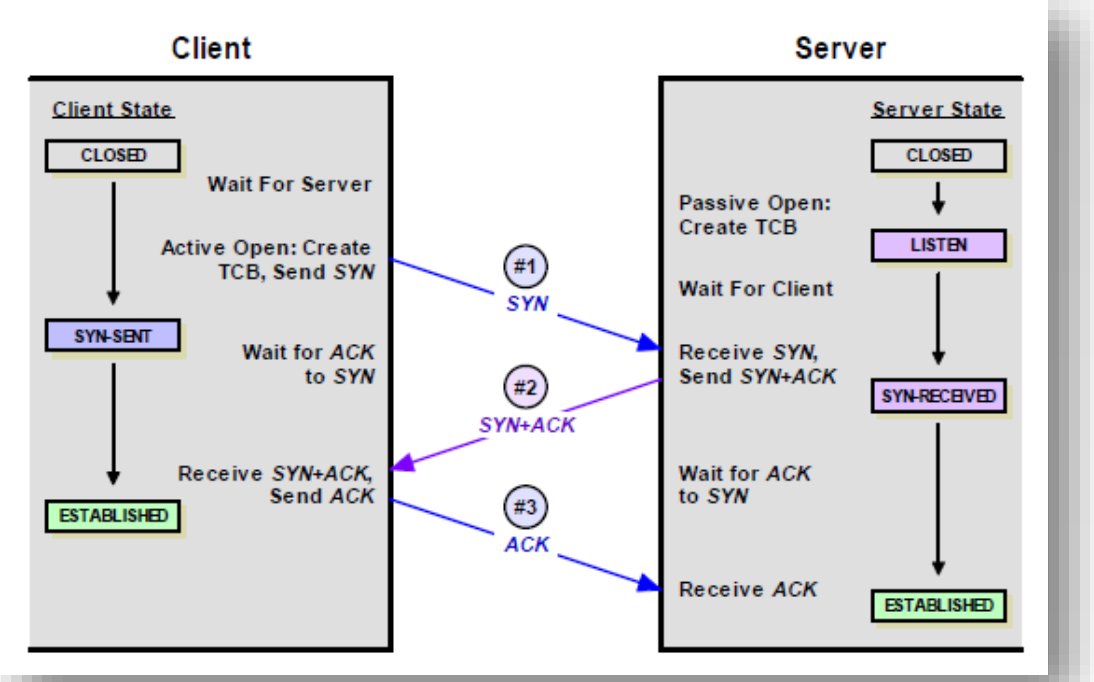
Para llegar a ese acuerdo, pueden mandar mensajeros que atraviesa B (éste puede ser interceptado)

No es posible llegar a un acuerdo confirmado al 100%

3. TCP

Conexión TCP

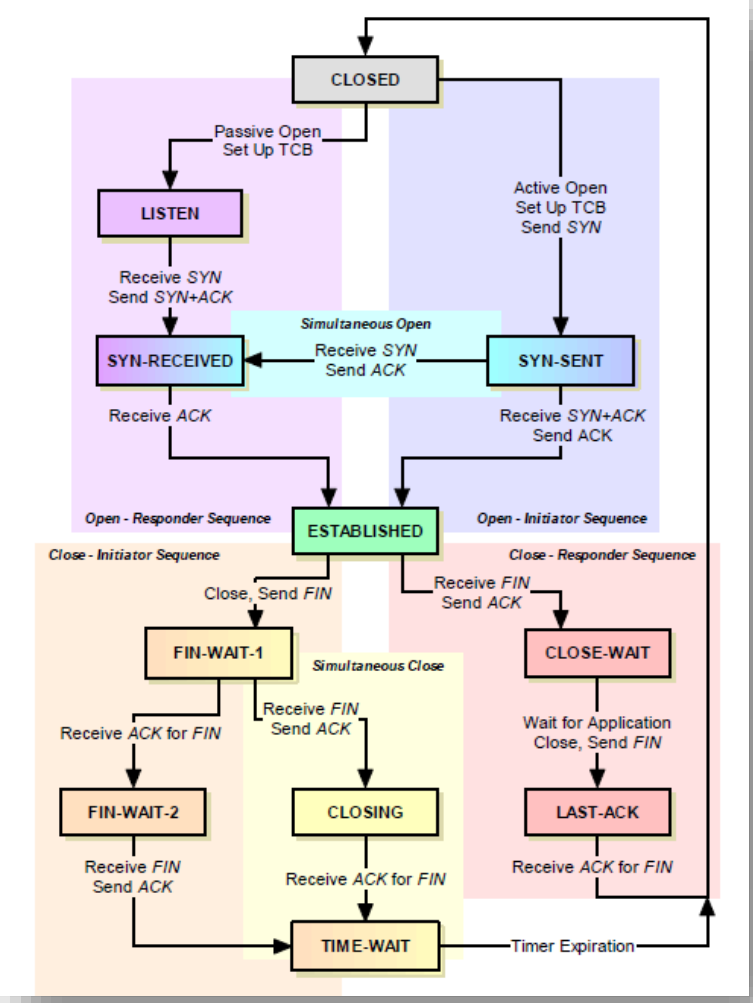
- Como medida de compromiso, para la conexión TCP se usa negociación en 3 pasos (***three-way handshake***):



3. TCP

Conexión TCP

- Cada entidad TCP implementa una máquina de estados (FSM, *finite state machine*)
- Cuando se establece una conexión las entidades pasan al estado `ESTABLISHED`
- Existen temporizadores (*timeouts*) entre algunos estados para detectar situaciones de error



3. TCP

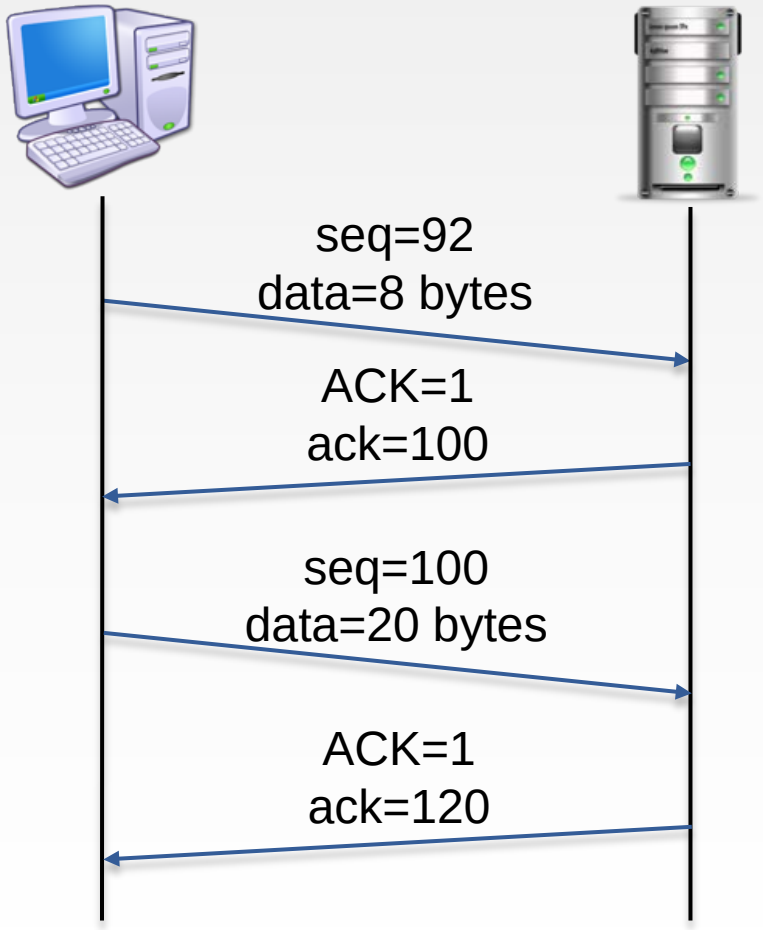
Transferencia de datos

- TCP proporciona un servicio de transporte fiable de un flujo de **bytes**
- La entidad que envía datos tiene que **secuenciar** los datos que envía
  - Campo nº de secuencia (*seq*)
- El receptor de segmentos de datos tiene que **asentir** los que recibe
  - Flag ACK=1
  - Campo nº acuso de recibo (*ack*): Número de secuencia que el receptor está esperando (confirma bytes hasta nº-1)
- El esquema de asentimiento más utilizado en TCP es ACK acumulativo inmediato

3. TCP

Transferencia de datos

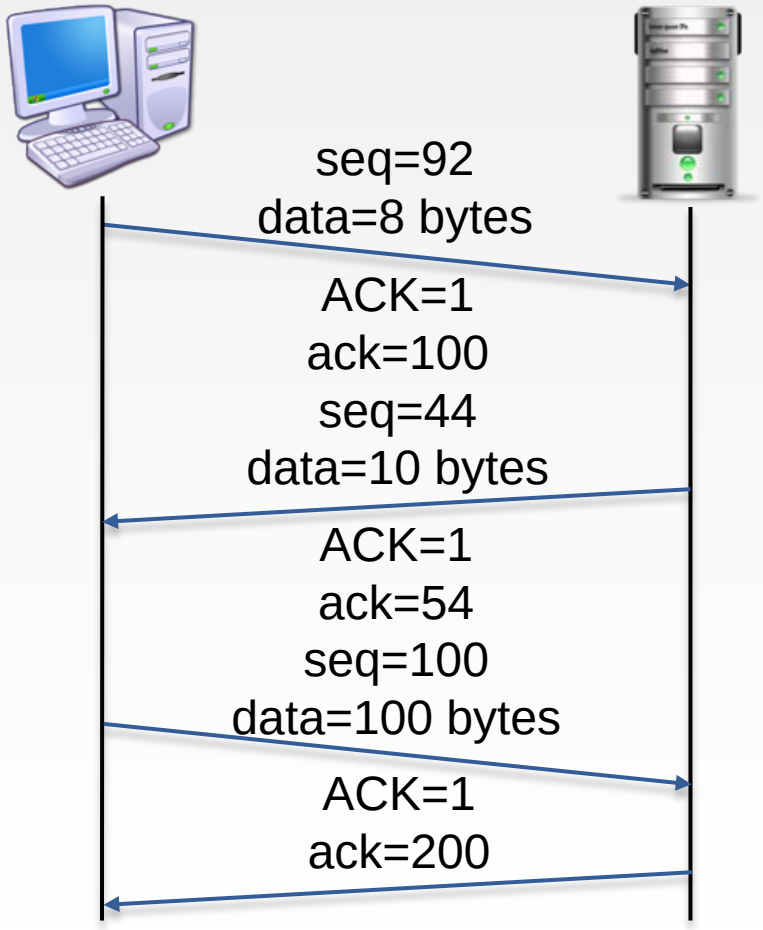
- Ejemplo: datos enviados y recibidos correctamente



3. TCP

Transferencia de datos

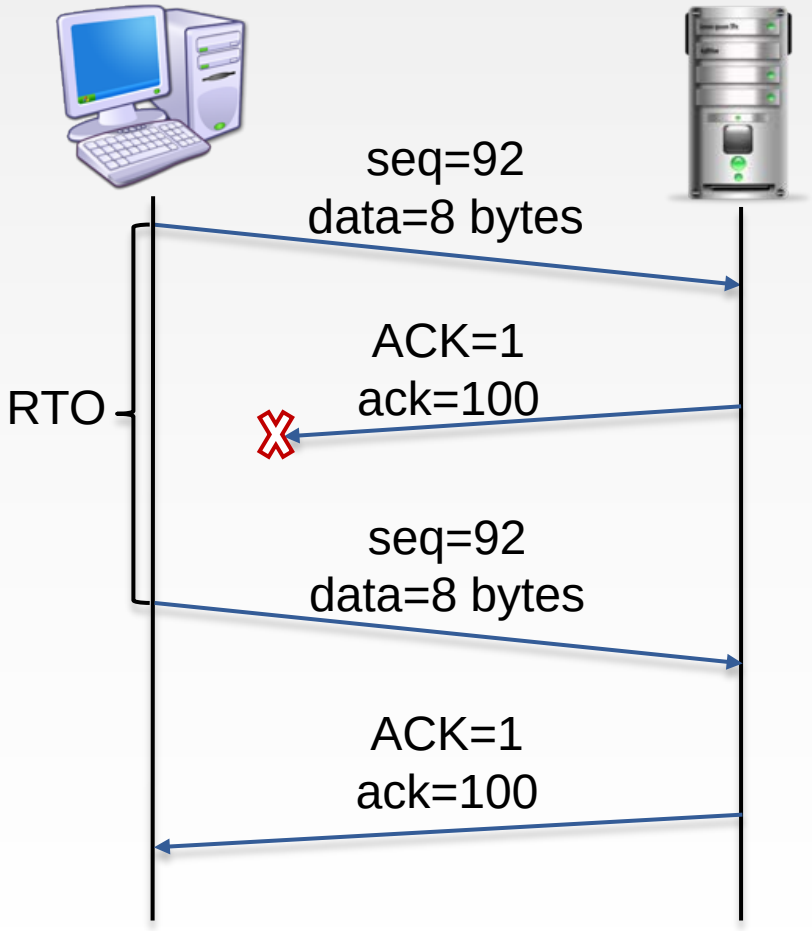
- Ejemplo: datos enviados y recibidos correctamente de forma simultánea (*piggybacking*)



3. TCP

Transferencia de datos

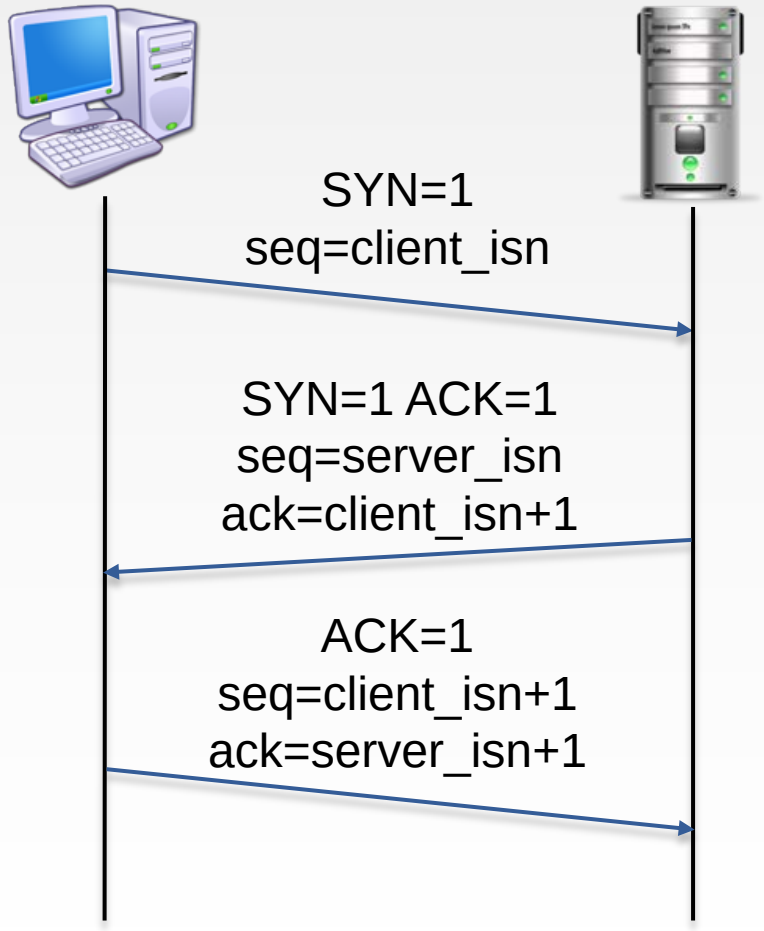
- Ejemplo: segmento perdido en la red (retransmisión)
- El tiempo de espera de retransmisión (RTO, *Retransmission Timeout*) es un valor que varía en función de la latencia de ida y vuelta (RTT, *Round-Trip delay Time*)
  - Definido en la RFC 2988
  - Tiene un valor inicial de 3 segundos



3. TCP

Transferencia de datos

- El número de secuencia inicial (ISN) se fija en la fase de establecimiento de la conexión (SYN=1)
- ISN se obtiene de un forma aleatoria. Se hace así por varios motivos:
  - Por robustez: para evitar mezclar segmentos de diferentes conexiones
  - Por seguridad: para evitar ataques por predicción de secuencia TCP (por ejemplo, el "Mitnick attack")



3. TCP

Transferencia de datos

- Otros modos de asentimiento:
  - ACK selectivo (RFC 2018): SACK
    - Permite realizar asentimientos fuera de secuencia
  - ACK retardado: se espera hasta 200 ms para mandar ACK
    - Permite retrasar asentimiento ante la llegada de una ráfaga de segmentos
  - Algoritmo de Nagle (RFC 896)
    - Método heurístico (conjunto de reglas basadas en la experiencia) cuyo objetivo es mejorar la eficiencia de los canales de comunicación TCP

3. TCP

Transferencia de datos

- TCP proporciona el servicio de segmentación (división y reensamblado de un flujo de bytes grande)
- El tamaño máximo de segmento TCP se conoce como **MSS** (*Maximum Segment Size*)
- El cálculo de MSS se hace a nivel de sistema operativo:

```
MSS = MTU – cabecera_TCP – cabecera_IP = MTU – 40
```

- MTU (*Maximum Transmission Unit*) es el valor máximo del tamaño de las tramas (nivel 2, enlace)
- El valor de MTU en redes Ethernet es de **1500 bytes**
  - Este valor lo fija Ethernet para maximizar la tasa de transferencia efectiva del enlace (*throughput*)

3. TCP

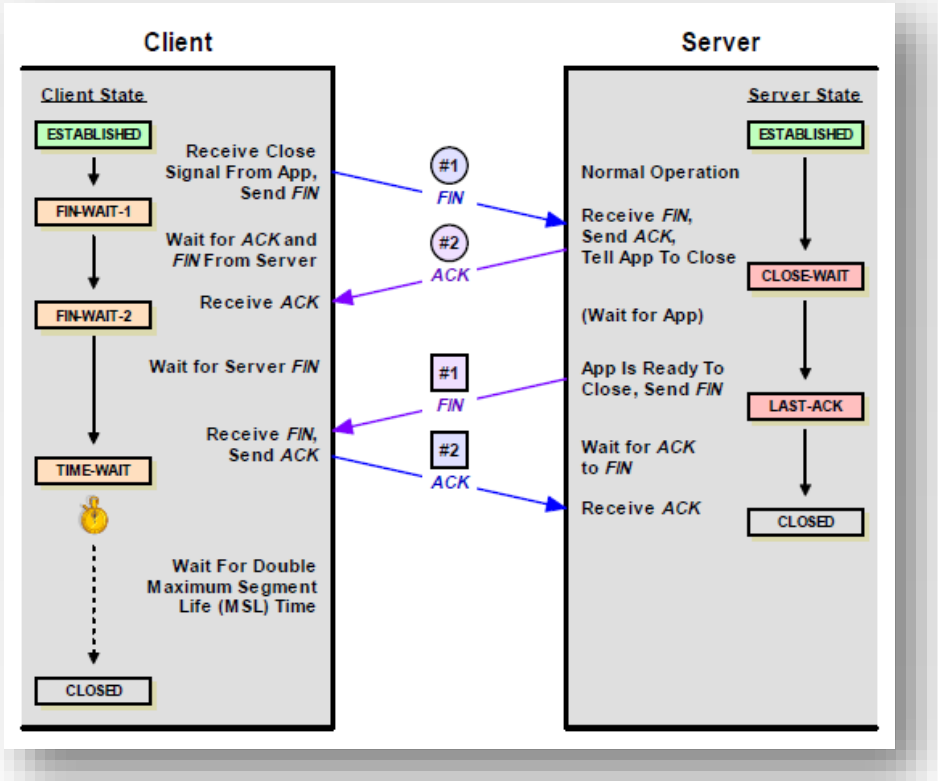
Desconexión TCP

- Para la desconexión TCP se usa negociación en 4 pasos (***four-way handshake***):

TCP define un tiempo de espera al finalizar la conexión:
$t_{TIME\text{-}WAIT}$ = 2* MSL = 4 minutos

MSL (*Maximum Segment Life*) = 2 minutos (es el tiempo máximo que puede existir un segmento)

Este tiempo de espera sirve para:
1. Proporciona tiempo para ACK
2. Evitar colisión de segmentos con conexiones subsiguientes



3. TCP

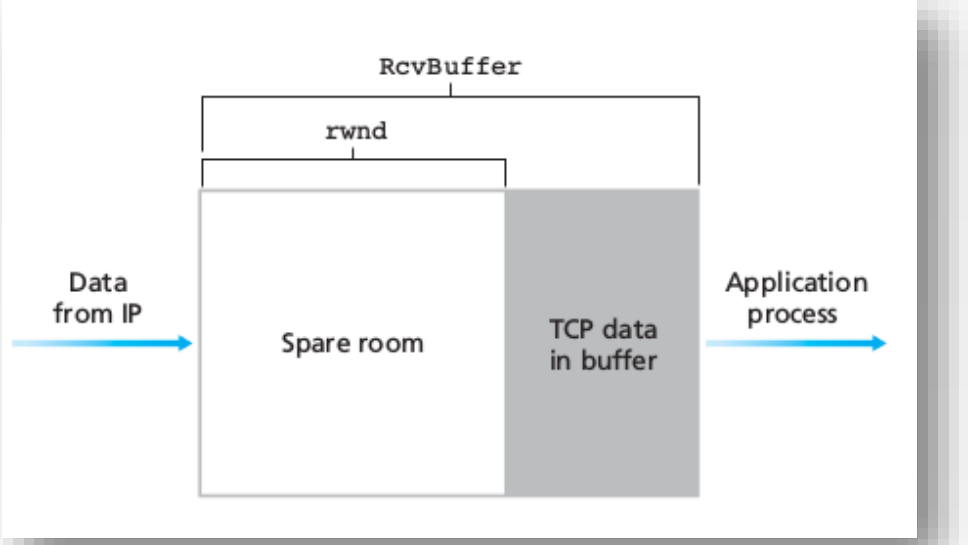
Control de flujo

- TCP usa el método de **ventana deslizante** para el control de flujo
- Para el control de flujo se define una variable llamada **ventana de recepción** (**rwnd**)
- Esta variable se envía en cada segmento TCP y mediante ella se especifica el **número de bytes que pueden recibir**
- Se llama ventana deslizante porque:
  - **rwnd** disminuye (ventana se cierra) al confirmar bytes (ACK)
  - **rwnd** aumenta (ventana se abre) al procesar bytes (se el libera buffer de recepción)

3. TCP

Control de flujo

- Cuando el receptor indica un tamaño de ventana=0, el transmisor detiene el envío y pone un contador de tiempo
- Cuando expira el contador, el transmisor manda paquetes pequeños al receptor (para evitar interbloqueo)



3. TCP

Control de flujo

- La ventana de recepción (**rwnd**) se calcula usando esta fórmula:

```
rwnd = RcvBuffer - [LastByteRcvd - LastByteRead]
```

  - `RcvBuffer` = Tamaño buffer de recepción
  - `LastByteRcvd - LastByteRead` = Número de datos recibidos pero no procesados
  - `LastByteRcvd` = Número de secuencia del último byte recibido y confirmado
  - `LastByteRead` = Número de secuencia del último byte procesado

3. TCP

Control de congestión

- El control de congestión se implementa con un mecanismo conocido como **ventana de congestión** (**cwnd**)
- La ventana de congestión impone una **restricción en el flujo de envío de datos** (limita el número de bytes que puede haber enviados y sin confirmar)

$$\mathbf{LastByteSent - LastByteAcked \leq \min\{cwnd, rwnd\}}$$

  - `LastByteSent - LastByteAcked` = Nº de datos enviados y sin confirmar
  - `LastByteSent` = Número de secuencia del último byte enviado
  - `LastByteAcked` = Número de secuencia del último byte confirmado
- Suponiendo que rwnd es lo suficientemente grande (no hay problemas de recepción, caso habitual), el flujo de envío estará determinado por el valor de la ventana de congestión (cwnd)

3. TCP

Control de congestión

- El valor de cwnd varía según el algoritmo de control de congestión en TCP (RFC 2581)
- Este algoritmo tiene tres fases:
  - Arranque lento (*slow start*)
  - Evitación de congestión (*congestion avoidance*)
  - Detección de congestión

3. TCP

Control de congestión

- **Arranque lento**. Ocurre en el inicio de la transferencia de datos
  - Ni el transmisor ni el receptor saben cual es el volumen máximo de datos que puede transmitir por la red
  - Por eso se comienza enviando segmentos con un tamaño pequeño, que irá aumentando paulatinamente
  - Con cada ACK que llega, cwnd dobla su tamaño (comportamiento exponencial) hasta que se alcanza un valor máximo llamado SST (*Slow Start Threshold*)
- **Evitación de congestión**. Una vez alcanzado SST, el crecimiento de cwnd pasa a ser lineal:

```
cwnd(superado SST) = cwnd(anterior) + MSS/cwnd(antiguo)
```

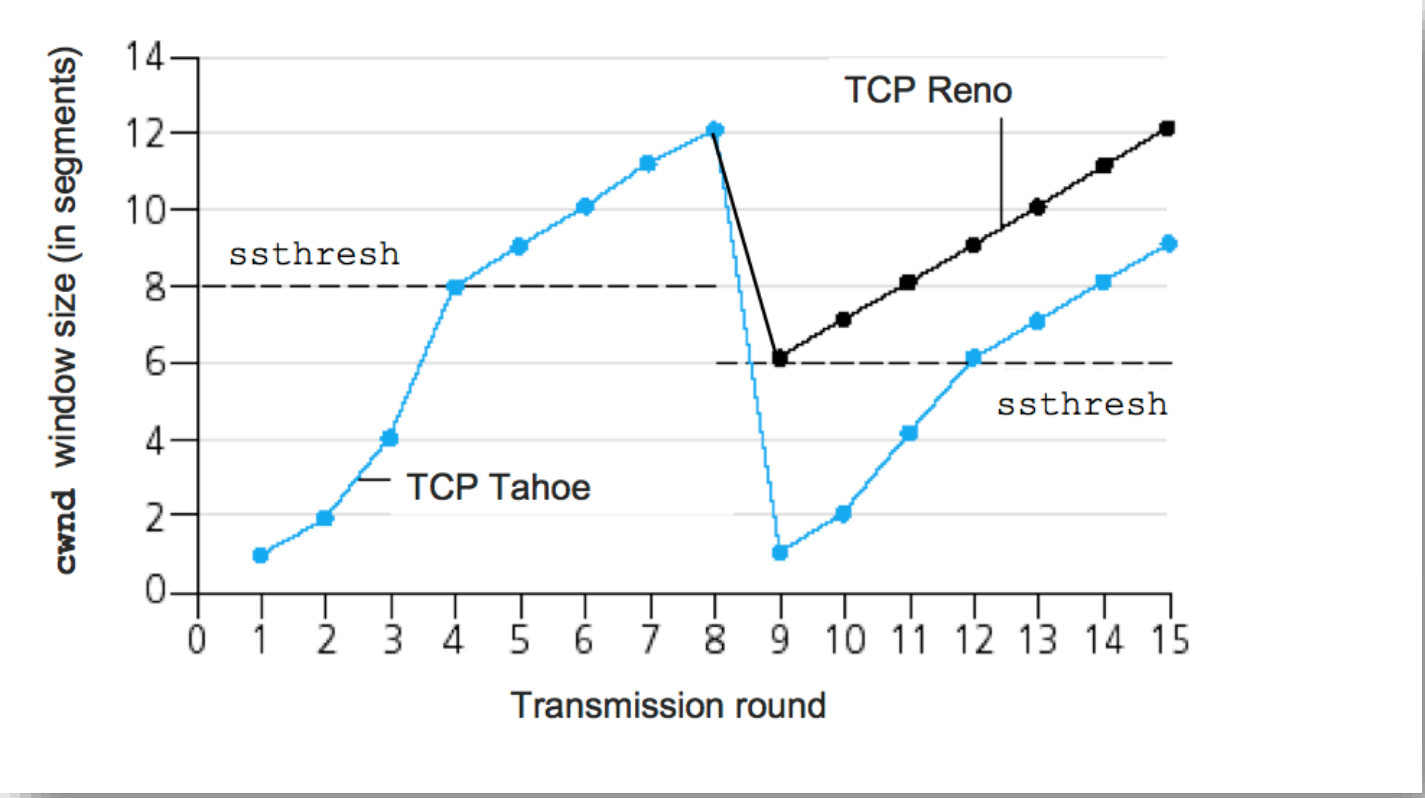
3. TCP

Control de congestión

- **Detección de congestión**. En cualquiera de las dos fases anteriores se puede detectar un problema de congestión en la red. Se detecta de forma diferente dependiendo de la implementación de TCP:
  - TCP Tahoe: Un problema de congestión en esta implementación se produce expira un temporizador (*timeout*) en la recepción de un ACK. En este caso, se reduce el valor de SST=cwnd/2 y cwnd=1, y se vuelve a la fase de arranque lento
  - TCP Reno: Un problema de congestión en esta implementación se puede produce cuando se reciben **tres ACK repetidas** diferentes al último segmento enviado. En este caso, se reduce el valor de SST=cwnd/2 y cwnd=cwnd/2 sin volver a la fase de arranque lento (esto se conoce como recuperación rápida, *fast recovery*)

3. TCP

Control de congestión



Índice de contenidos

4. Seguridad en redes de datos

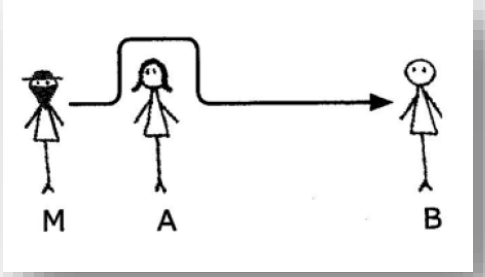
Conceptos de seguridad

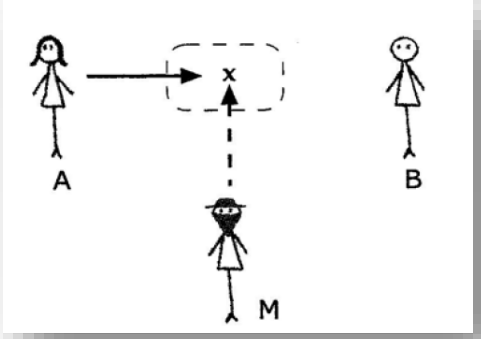
- La **seguridad** en redes de datos son un conjunto de técnicas que tienen el objetivo de minimizar las **vulnerabilidades** en redes telemáticas
- El objetivo de hacer seguros los servicios distribuidos es conseguir que el coste de la consecución indebida de un recurso sea superior al bien que se intente proteger
- No existe la seguridad total: ante cualquier medida de protección, siempre se podrá encontrar un elemento capaz de romperla
- Un **ataque** (*attack*) es una amenaza intencionada por parte de un sujeto malicioso que pretende hacer un uso indebido de un servicio

4. Seguridad en redes de datos

Conceptos de seguridad

- Tipos de ataques en función del usuario malintencionado:
  - Suplantación de personalidad (*spoofing*). Se produce cuando una persona o entidad suplanta la personalidad de otra con fines ilícitos. Cuando la suplantación ocurre en un mensaje de correo electrónico o sitio web intenta hacer creer al usuario que es la entidad suplantada (*phishing*)
  - Denegación del servicio (*denial of service*, DoS, *distributed denial-of-service*, DDoS). Se produce cuando se consigue, con fines fraudulentos, que una entidad no cumpla deliberadamente su cometido, o se impide que otras entidades cumplan con las funcionalidades que tienen encomendadas

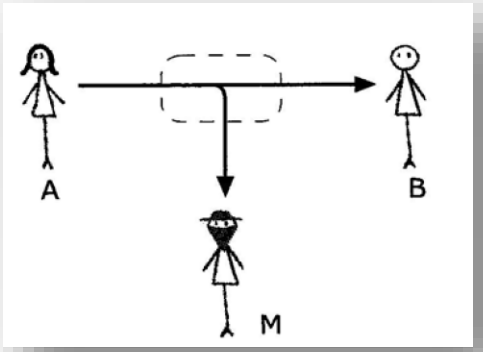


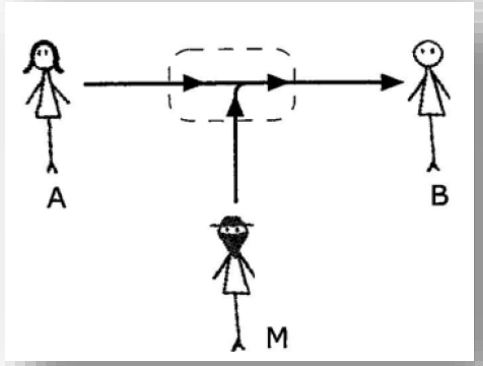


4. Seguridad en redes de datos

Conceptos de seguridad

- Tipos de ataques en función del usuario malintencionado:
  - Divulgación o repetición del contenido (*replay*). Ocurre cuando una entidad repite un mensaje, o parte de él, para dirigirlo a un destinatario no autorizado. En otras palabras, es el típico “pinchazo” de una línea de comunicación para obtener información de forma fraudulenta
  - Modificación de mensajes (*modification*). Consiste en la alteración del contenido de un mensaje (evitando que esta alteración sea detectada) con la finalidad de que su destinatario adopte una decisión o tenga una percepción de la realidad distinta de aquella que se produciría caso de haber recibido el mensaje tal y como fue emitido





Estos tipos de ataque también se conocen como ***man-in-the-middle***

4. Seguridad en redes de datos

Conceptos de seguridad

- Tipos de software malicioso (***malware***):
  - Trampilla (*backdoor*/*trapdoor*): secuencia especial por la cual se pueden romper las medidas de seguridad. Puede ser un error en un sistema (*exploit*) pero también puede darse cuando un sistema es alterado para permitir a un atacante concreto un uso no autorizado
  - Troyano (*Trojan Horse*): un software malicioso que se presenta al usuario como un programa aparentemente legítimo e inofensivo, pero que, al ejecutarlo, le brinda a un atacante acceso remoto al equipo infectado
  - Virus: programas bien conocidos que, una vez instalados, destruyen parte de la información almacenada o de los recursos del sistema
  - Gusano (*worm*): programas autorreplicables que se transmiten por las redes y provocan la saturación de los sistemas (negación del servicio). La diferencia con respecto a los virus es que pueden viajar sin la intervención humana

4. Seguridad en redes de datos

Conceptos de seguridad

- Los **piratas informáticos** son personas que “rompen” los sistemas de seguridad:
  - Hacker: se les presume una cierta intención lúdica o respaldada por un convencimiento moral de que “las redes son para los que las trabajan”
    - El término “hacker” también hace referencia a la comunidad entusiasta de desarrollo de software libre. Para más información leer el manifiesto *How to become a hacker* de Eric S. Raymond
  - Cracker: tienen una intención más perversa (por ejemplo, con un fines lucrativos). También llamado *cibercriminales*
  - Phreaker: son hacker/crackers que atacan las compañías telefónicas (por ejemplo para obtener llamadas gratis)
  - Wannabe o lamer: son aprendices que aspiran a ser hackers

4. Seguridad en redes de datos

Servicios de seguridad

- Un **servicio de seguridad** protege las comunicaciones de los usuarios ante determinados ataques
  - Un servicio de seguridad es proporcionado por un nivel N, se garantiza a las entidades (N+1) usuarias una determinada protección
- Los principales tipos se servicios de seguridad en redes de datos son:
  - **Confidencialidad** (*data confidentiality*) proporciona protección para evitar que los datos sean revelados a un usuario no autorizado
  - **Autenticación** (*authentication*): sirve para garantizar que una entidad (persona o máquina) es quien dice ser
  - **Autorización** (*authorization*): sirve para discernir si un usuario (ya autenticado) tiene acceso a un recurso
  - **Integridad** (*data integrity*): garantiza al receptor del mensaje que los datos recibidos coinciden exactamente con los enviados por el emisor

4. Seguridad en redes de datos

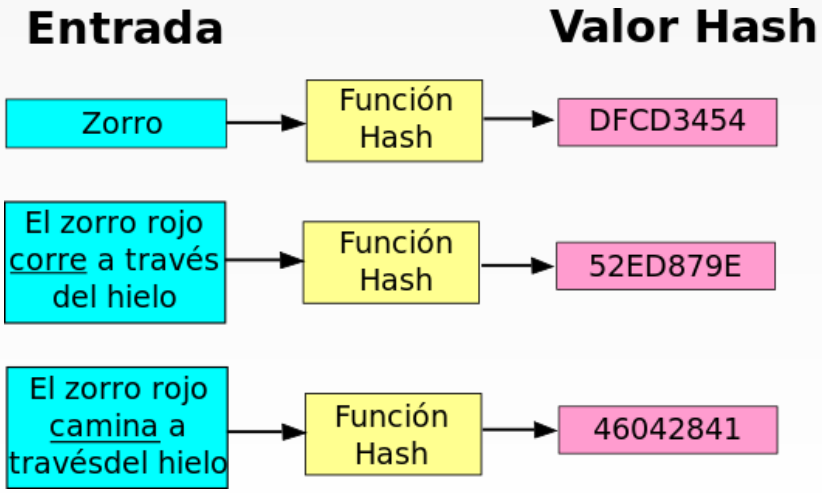
Servicios de seguridad

- La **autenticación** se consigue mediante:
  - Algo que sabes. Por ejemplo, unas credenciales login-password
  - Algo que tienes. Por ejemplo, una tarjeta de acceso o un token de seguridad
  - Algo que eres. Por ejemplo, cualidades biométricas (huella digital, patrón del iris)
- La **autorización** discrimina el acceso a un determinado recurso en base a permisos, roles, tokens, etc
  - A veces la autorización se produce con autenticación previa (AAA, *Authentication, Authorization and Accounting*)

4. Seguridad en redes de datos

Servicios de seguridad

- La **integridad de datos** típicamente se consigue integridad de datos con funciones ***hash*** (también llamado resumen):
  - Son funciones computables mediante un algoritmo, que convierte una entrada (típicamente una cadena) a un rango de salida finito (típicamente cadenas de longitud fija)



4. Seguridad en redes de datos

Servicios de seguridad

- La **integridad de datos** típicamente se consigue integridad de datos con funciones ***hash*** (también llamado resumen):
  - Es mucho más robusto que el mecanismo de *checksum*
  - La posibilidad de colisión (dos textos diferentes que producen el mismo hash) es muy pequeña
  - Es una función unidireccional (dado un hash es “imposible” sintetizar un texto que lo produzca
  - Implementaciones típicas de función hash:
    - SHA (*Secure Hash Algorithm*)
    - MD5 (*Message-Digest Algorithm 5*)
    - DSA (*Digital Signature Algorithm*)

4. Seguridad en redes de datos

Criptografía

- El **servicio de confidencialidad** se consigue mediante sistemas criptográficos (criptosistemas)
- La **criptografía** es la técnica que se ocupa del cifrado o codificado destinadas a alterar las representaciones de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados
- La palabra “criptografía” proviene del griego *criptos* (“oculto”) y *grafé*, (“escritura”), literalmente “escritura oculta”

4. Seguridad en redes de datos

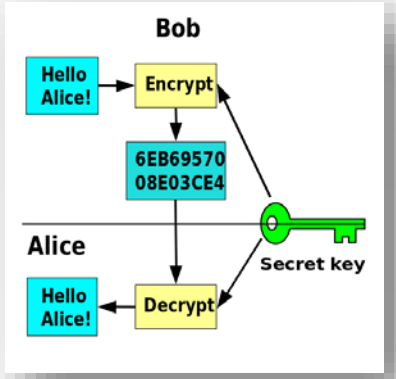
Criptografía

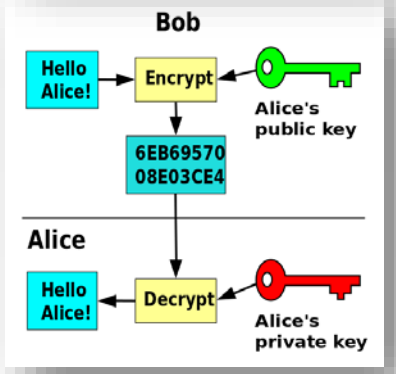
- Esquema genérico de un sistema criptográfico
  - Un subsistema produce un mensaje cifrado a partir de un mensaje original, mediante la aplicación de una transformación de éste que depende de un parámetro denominado clave de cifrado
  - En el extremo opuesto, el mensaje original puede ser recuperado en otro subsistema mediante la aplicación de la clave de descifrado

4. Seguridad en redes de datos

Criptografía

- Tipos de sistemas criptográficos:
  - **Criptosistemas de clave secreta**. En ellos, la clave de cifrado y de descifrado es la misma: es una clave secreta que comparten el emisor y el receptor del mensaje. Debido a esta característica son denominados también criptosistemas **simétricos**
  - **Criptosistemas de clave pública**. Se distinguen porque cada usuario o sistema final dispone de dos claves: una privada, que debe mantener secreta, y una pública, que debe ser conocida por todas las restantes entidades que van a comunicar con ella. Se los conoce también como criptosistemas **asimétricos**

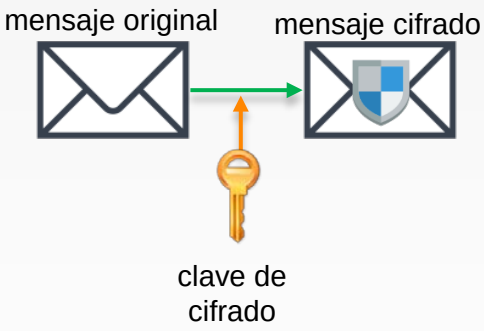


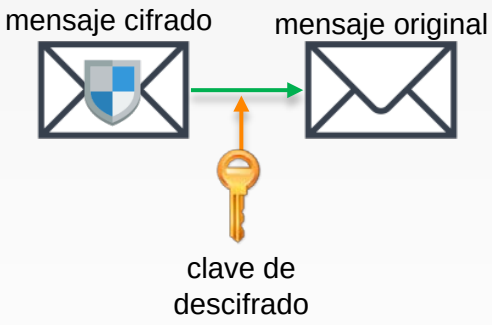


4. Seguridad en redes de datos

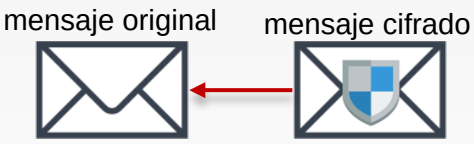
Criptografía

- Los criptosistemas de clave pública se apoyan en la **teoría de la complejidad** mediante el uso de algoritmos computacionalmente complejos en un sentido pero simples en el otro





Clase P (determinista en tiempo polinómico)



Clase NP (no determinista en tiempo polinómico) o NP-completo

La longitud de la clave (en bits) es critica para evitar ataque por fuerza bruta

4. Seguridad en redes de datos

Criptografía

- Algunos ejemplos de algoritmos usados en criptosistemas asimétricos:
  - Test de primalidad. Se basa en la facilidad de multiplicar dos números primos grandes para obtener un número compuesto y en la dificultad de hallar la descomposición en factores primos dado un número obtenido de la multiplicación de dos primos grandes
  - Factorización de enteros. Se basa en la dificultad de descomponer números enteros muy grandes en sus factores primos
    - Un intento reciente de factorizar un número de 200 dígitos (RSA-200) tardó 18 meses
  - Problema del logaritmo discreto. Es un número entero *k* que soluciona la ecuación $b^k = g$ donde *b* y *g* son elementos de un grupo finito (la exponenciación discreta es una operación sencilla mientras que el logaritmo para ciertos grupos es irresoluble)

4. Seguridad en redes de datos

Criptografía

- Algunos ejemplo de sistemas criptográficos:

Criptosistemas asimétricos

- RSA (Rivest, Shamir y Adleman)
- Diffie-Hellman
- ElGamal
- Criptografía de curva elíptica

Criptosistemas simétricos

- AES (*Advanced Encryption Standard*)
- ES (*Data Encryption Standard*)
- IDEA (*International Data Encryption Algorithm*)
- 3DES
- RC2, RC4, RC5
- Blowfish

4. Seguridad en redes de datos

Criptografía

- ¿Cómo podemos hacer pública una clave pública?

1. Mediante un conjunto de entidades y procedimientos a los que llamamos Infraestructura de Clave Pública (PKI, *Public Key Infrastructure*) destinados a provisionar servicios de seguridad
   - En este modelo existen entidades de confianza denominadas Autoridad de Certificación (CA) que emiten **certificados digitales**
   - El certificado digital es una pieza de información que asocia a una entidad con su clave pública
2. Mediante servidores de claves públicos
   - Las claves públicas de los usuarios se almacenan en un servidor de claves (*key servers*)
   - Por ejemplo: keys.gnupg.net, pool.sks-keyservers.net, pgp.mit.edu

4. Seguridad en redes de datos

Criptografía

- El tipo de certificado más utilizado: X.509 (estándar del ITU-T)
- Algunas extensiones comunes de archivo de certificados X.509 son:
  - `.cert`, `.crt`, `.der`: Certificado codificado mediante el estándar ASN.1 (protocolo de nivel de presentación en el modelo OSI)
  - `.pem`: Certificado DER codificado en Base64, encerrado entre `"-----BEGIN CERTIFICATE-----"` y `"-----END CERTIFICATE-----"`
  - `.p12`: Certificado PKCS#12 (*Public-Key Cryptography Standards*), puede contener certificado(s) y claves privadas
- Otras extensiones para archivos de claves:
  - `.ppk`: Formato de archivo de claves implementado por la aplicación PuTTY (requerido por FileZilla por ejemplo)
- A veces los certificados pueden estar protegidos con una frase de contraseña (*pass phrase*) para controlar el acceso

4. Seguridad en redes de datos

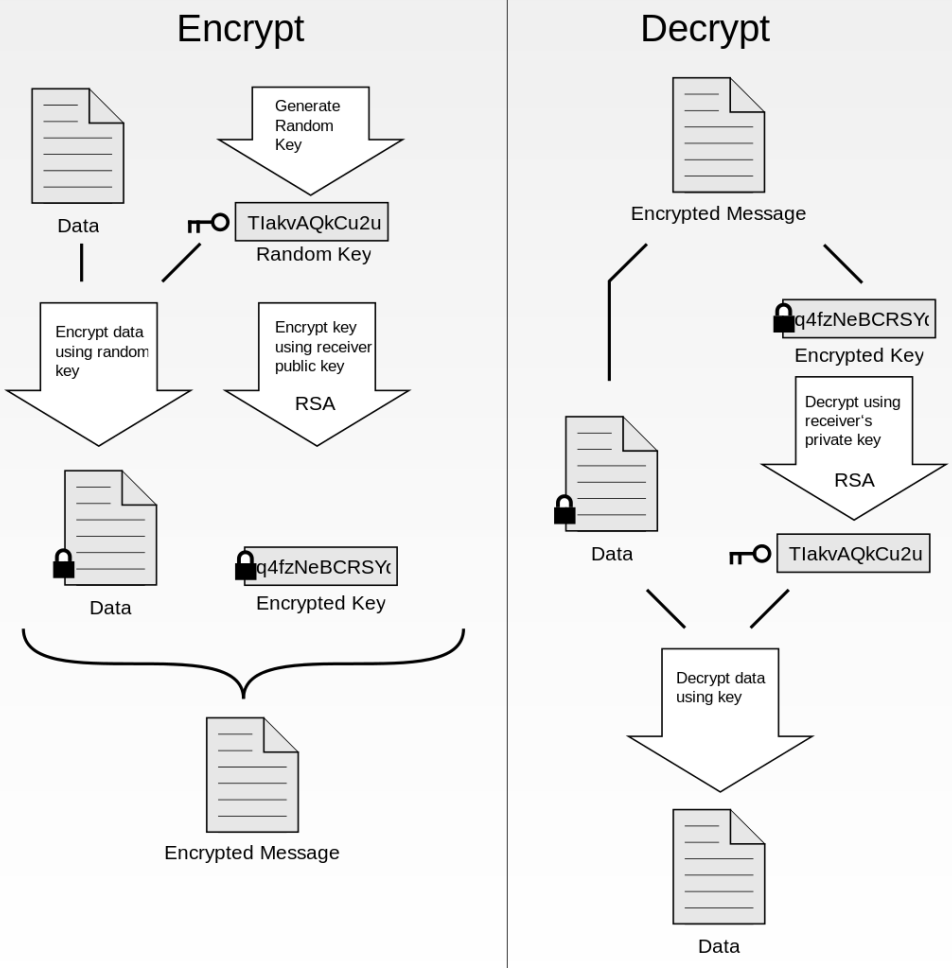
PGP

- La seguridad extremo a extremo en el correo electrónico sólo se puede conseguir cifrando el contenido del mensaje al ser enviado
- Esto se puede hacer mediante PGP (*Pretty Good Privacy*), que es un estándar de facto para cifrar mensajes de correo electrónico
  - Fue creado por Phil Zimmermann en 1991
  - Es la base del estándar OpenPGP (RFC 4880)
- PGP es un criptosistema híbrido que combina técnicas de criptografía simétrica y criptografía asimétrica
- Las claves públicas de los usuarios se almacenan en un servidor de claves
- Una alternativa *open source* a PGP es GPG (*GNU Privacy Guard*). Su funcionamiento es equivalente (está basado en OpenPGP)

4. Seguridad en redes de datos

PGP

- Esquema de funcionamiento de PGP:



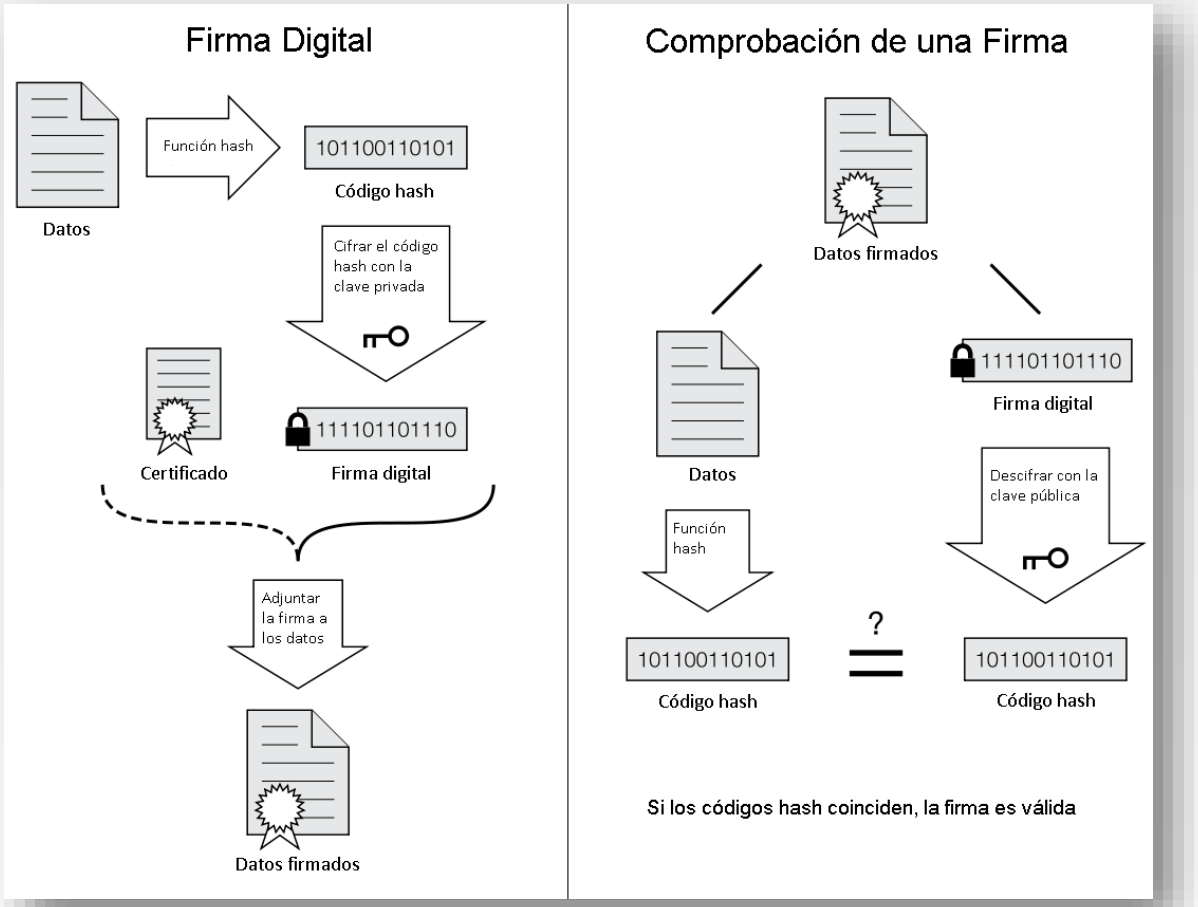
4. Seguridad en redes de datos

Firma digital

- Hasta ahora hemos visto que la clave pública de una entidad puede ser usada para cifrar un mensaje dirigido a dicha entidad (confidencialidad)
- Otra forma de proceder es usar la **clave privada** para cifrar un mensaje con el objetivo de proporcionar **autenticación**
- Este mecanismo se conoce con el nombre de **firma digital**
- Para firmar digitalmente un documento se usa la clave privada para cifra es el resumen (*hash*) del mensaje
  - El receptor del mensaje firmado, mediante comparación, puede adquirir suficientes garantías acerca de quién es el autor (autenticación) y sobre la integridad del mensaje recibido

4. Seguridad en redes de datos

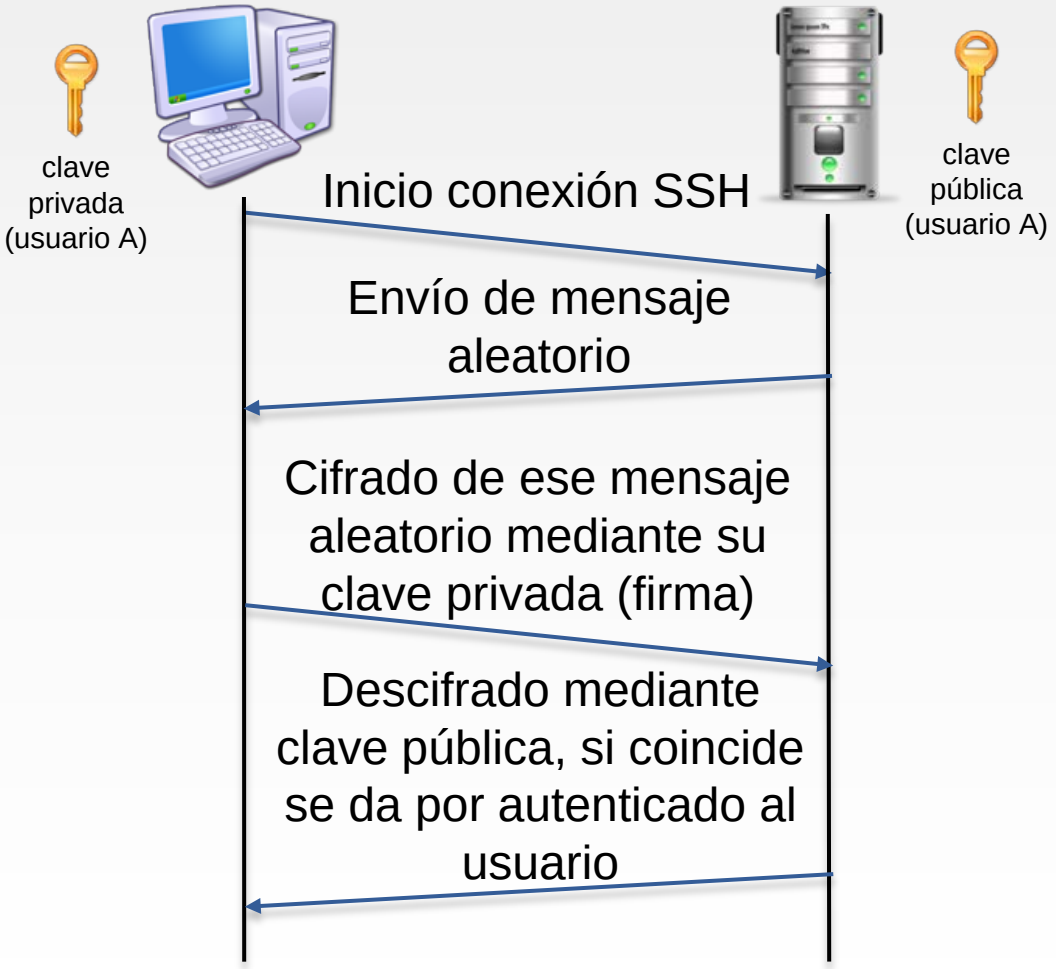
Firma digital



4. Seguridad en redes de datos

Firma digital

- El mecanismo de firma se puede usar para autenticar clientes en servidores SSH



4. Seguridad en redes de datos

TLS

- SSL (*Secure Sockets Layer*) y su sucesor TLS (*Transport Layer Security*) son protocolos criptográficos que proporcionan comunicaciones seguras por una conexión TCP
- Existe una variante de TLS que trabaja sobre UDP llamada DTLS (*Datagram Transport Layer Security*)

Versión	Año
SSL 2.0	1995
SSL 3.0	1996
TLS 1.0	1999
TLS 1.1	2006
TLS 1.2	2008
TLS 1.3	2017

4. Seguridad en redes de datos

TLS

- Los servicios de seguridad ofrecidos por TLS son:
  - Confidencialidad (*data confidentiality*) →se cifra el intercambio de datos a nivel TCP
  - Autenticación (*authentication*) → entidades pueden (opcionalmente) confirmar su identidad. En un protocolo cliente-servidor sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar
  - Integridad (*data integrity*) →se usa una función hash para garantizar la integridad de datos

4. Seguridad en redes de datos

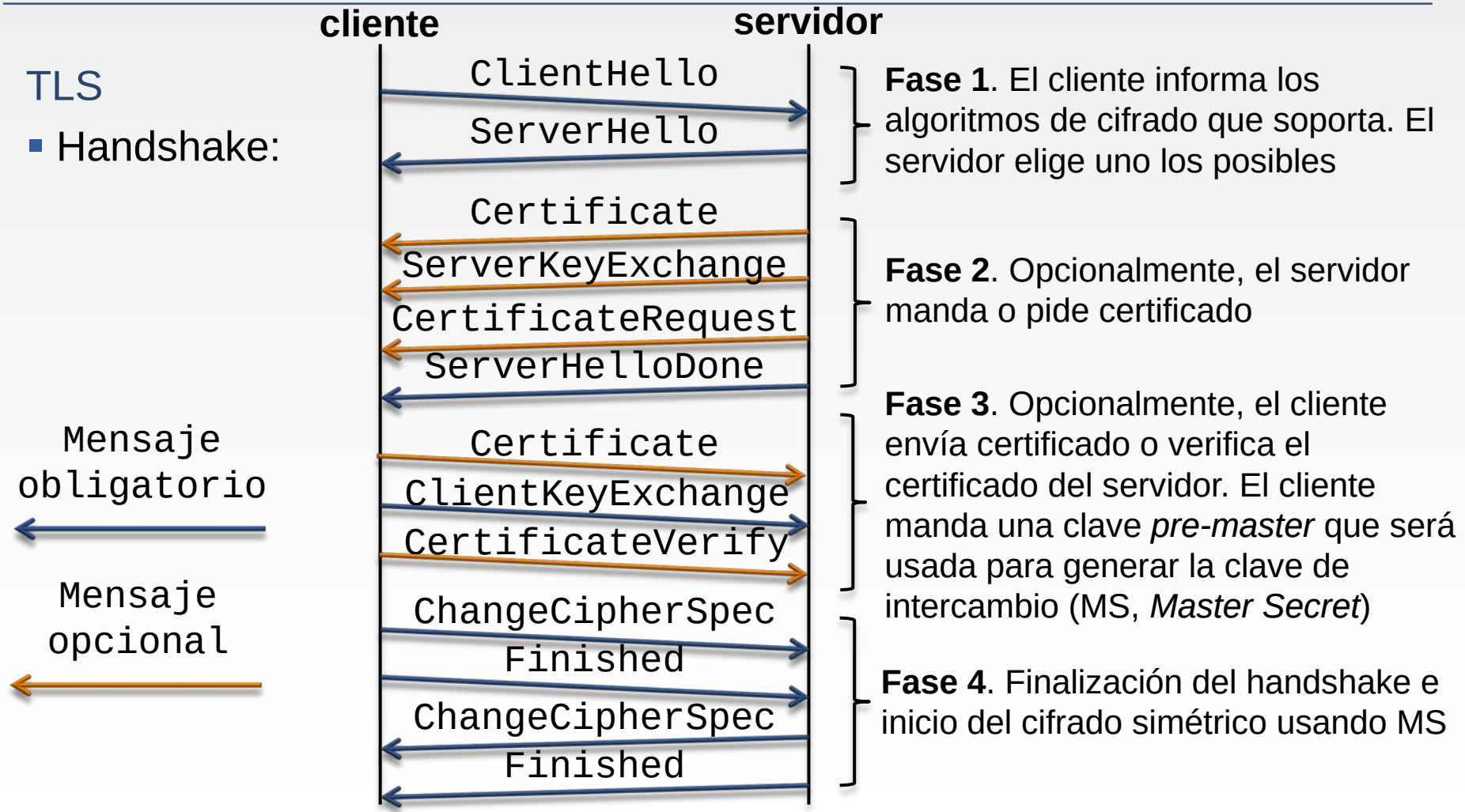
TLS

- Para establecer un canal seguro cifrado, las entidades tienen que llegar a un acuerdo (***handshake***)
- Este acuerdo implica una serie de fases:
  - Negociar entre las partes el algoritmo que se usará en la comunicación
  - Intercambio de claves públicas (certificados digitales)
  - Establecimiento de una clave maestra (MS, *Master Secret*) que será usada para cifrar todos los datos de la sesión
  - Cifrado del tráfico mediante cifrado simétrico

4. Seguridad en redes de datos

TLS

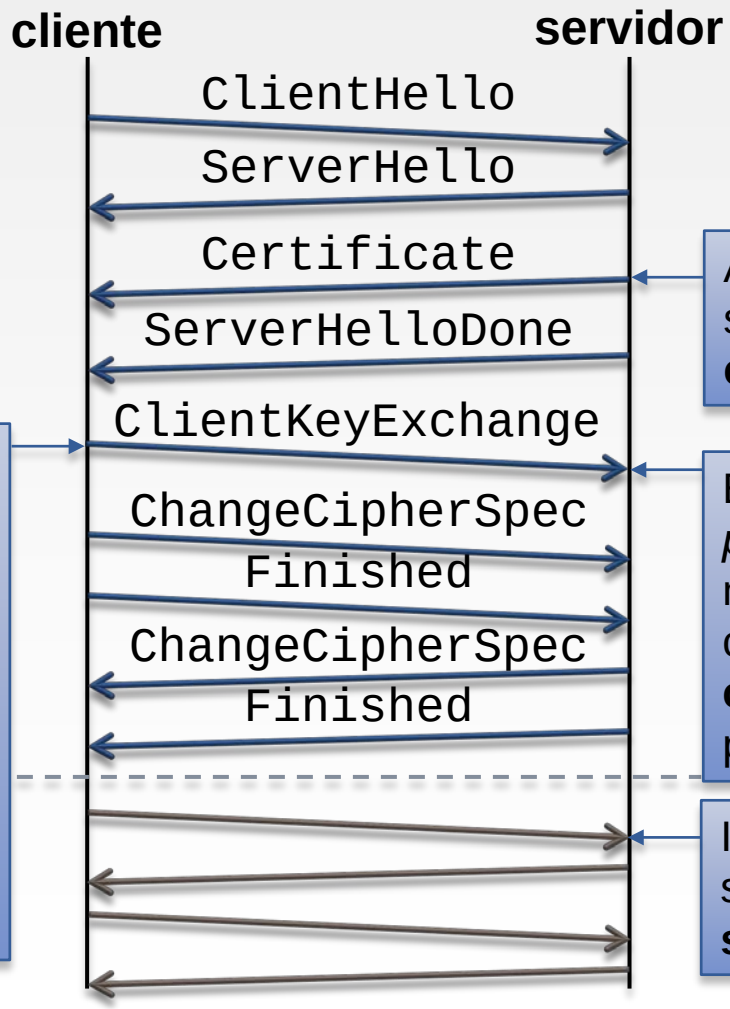
- Handshake:



4. Seguridad en redes de datos

TLS

- Ejemplo: aplicación segura cliente-servidor (HTTPS, etc)



El cliente genera una clave *pre-master* que será usada generar la clave maestra MS con la que se cifrarán todos los datos de la sesión. Esta clave se envía cifrada con la **clave pública del servidor**, obtenida a partir del certificado

Al enviar el certificado el servidor está mandando su **clave pública** al cliente

El servidor obtiene la clave *pre-master* descifrando el mensaje enviado por el cliente, usando para ello **su clave privada**. A partir de la pre-master se obtiene la MS

Intercambio de datos en canal seguro mediante cifrado **simétrico**

Índice de contenidos

5. Introducción a IP

- IP (*Internet Protocol*) es el protocolo de nivel de red del modelo de referencia TCP/IP (Internet). Está definido en las RFCs 791 y 2460
- Proporciona un **servicio no orientado a la conexión**
  - Servicio IP es también llamado como "mejor esfuerzo" (*best effort*): lo hará lo mejor posible, pero sin garantías de entrega ni orden
- La PDU de IP se llama **paquete**
- Versiones IP: IPv4 e IPv6 →para las prácticas en Java usaremos IPv4
- La conexión de un host a una red se llama **interfaz de red** (*Network Interface Card*, NIC)
- Para identificar un host en una red IP usamos su **dirección IP**
- Dirección IPv4 = 32bits; Dirección IPv6 = 128bits

172	16	254	1

Notación decimal con puntos: dividimos los 32 bits en 4 y lo representamos en decimal

5. Introducción a IP

- Direcciones IP privadas:
  - Usadas en redes de área local (LAN)
  - La salida a Internet se realiza mediante una dirección IP pública

Rangos de direcciones IP privadas

Inicio	Fin	Nº de direcciones
`10.0.0.0`	`10.255.255.255`	16777216
`172.16.0.0`	`172.31.255.255`	1048576
`192.168.0.0`	`192.168.255.255`	65536

- Direcciones IP especiales:
  - `127.0.0.1` : Dirección de la máquina local (*localhost*)
  - `0.0.0.0` : Dirección especial no enrutable. Si un servicio escucha en la dirección `0.0.0.0`, se escuchan peticiones en todas las interfaces de red