



Ingeniería Informática + ADE

Universidad de Granada (UGR)

Autor: Ismael Sallami Moreno

Asignatura: Fundamentos de Redes



Índice

| | |
|------------------------------------|------------|
| 1. Teoría | 3 |
| 1.1. Tema 1 | 3 |
| 1.2. Tema 2 | 24 |
| 1.3. Tema 3 | 55 |
| 1.4. Tema 4 | 77 |
| 1.5. Tema 5 | 96 |
| 2. Relaciones de Ejercicios | 149 |
| 2.1. Relación 1 | 149 |
| 2.2. Relación 2 | 155 |
| 2.3. Relación 3 | 168 |
| 2.4. Relación 4 | 182 |
| 2.5. Relación 5 | 188 |
| 3. Fuente | 199 |

1 Teoria

1.1. Tema 1

Tema 1

Introducción a los fundamentos de redes

Fundamentos de Redes

Grado en Ingeniería Informática y dobles grados

Curso 3º

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones

E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2022



1

Tema 1. Introducción a los fundamentos de redes



Esquema

1. Sistemas de comunicación y redes
2. Diseño y estandarización de redes
3. Terminología, conceptos y servicios
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

2





Objetivos del tema

- Conocer y comprender los principios básicos de las comunicaciones y los elementos que las conforman.
- Entender el diseño funcional en capas de las redes y los conceptos y terminología fundamentales involucrados.
- Comprender desde un punto de vista teórico-conceptual el modelo de referencia OSI y su correspondencia con el modelo de capas usado en Internet.



Bibliografía



Capítulo 1, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson, 2ª Edición. 2014, ISBN: 9788490354612.



Capítulo 1, James F. Kurose y Keith W. Ross. **REDES DE COMPUTADORES, UN ENFOQUE DESCENDENTE**, 7ª Edición, Addison-Wesley, 2017, ISBN: 9788490355282





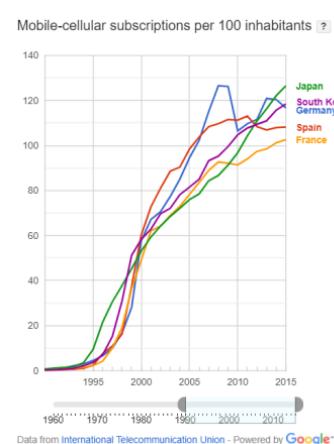
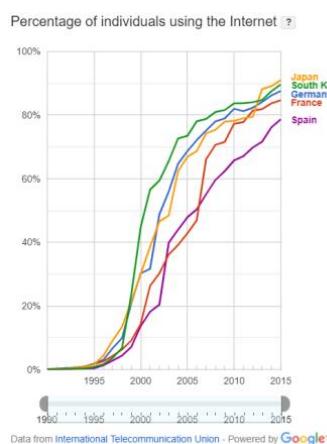
Esquema

1. **Sistemas de comunicación y redes**
2. Diseño y estandarización de redes
3. Terminología, conceptos y servicios
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios



1. Sistemas de comunicación y redes

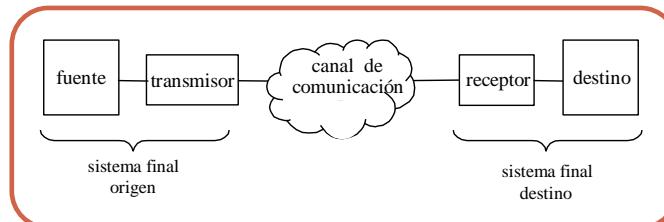
<http://www.itu.int/net4/itu-d/icteye/>





1. Sistemas de comunicación y redes

- **Sistema de comunicación:** infraestructura (*hard* y *soft*) que permite el intercambio de información



- **Información:** Conjunto de datos con significado

- **Red** (de computadores, de móviles, de dispositivos...): sistema de comunicación con sistemas finales (terminales) **autónomos** (con capacidad de procesar información) que facilita el intercambio **eficaz** y **transparente** de información



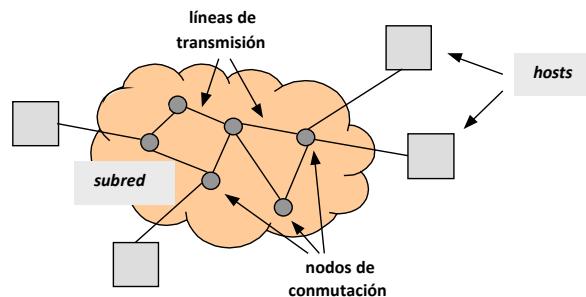
1. Sistemas de comunicación y redes

- Razones (motivación) para usar redes:
 - Compartir recursos
 - Escalabilidad
 - Fiabilidad, robustez → Duplicidad (redundancia)
 - Ahorro de costes (computación distribuida)
- Qué esperamos de una red (de computadores, de móviles, de dispositivos...)
 - Autonomía: capacidad de procesar información
 - Interconexión: mediante un sistema de comunicación
 - Intercambio de información, con eficacia y transparencia



1. Sistemas de comunicación y redes

- Estructura y elementos de una red:
 - **Hosts:** sistemas finales (terminales) autónomos
 - **Subred:** infraestructura para el transporte de información
 - Líneas de transmisión
 - Nodos o elementos de conmutación: *routers / switches*



1. Sistemas de comunicación y redes

■ Medios de transmisión:

■ Cable coaxial



■ Cable de par trenzado → UTP, STP, FTP



■ Fibra óptica





1. Sistemas de comunicación y redes

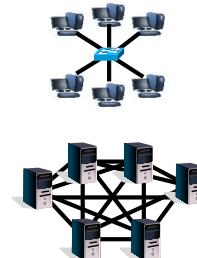
■ Topologías de redes: patrón de interconexión entre sus nodos

■ Física Vs lógica



■ Tipos:

- En bus
- En anillo
- En estrella
- En árbol
- Mallada
- Híbrida



1. Sistemas de comunicación y redes

■ Clasificación de redes

- Según tamaño y extensión: (PAN), LAN, MAN, WAN
- Según tecnología de transmisión: difusión, punto a punto
- Según el tipo de transferencia de datos: simple, half-duplex, full-duplex



Tema 1. Introducción a los fundamentos de redes

1. Sistemas de comunicación y redes

■ Nomenclatura típica en figuras (iconos)

| | | | |
|------------------------|--|-------------------------------------|--|
| HUB | | ROUTER CON CONMUTACIÓN DE ETIQUETAS | |
| BRIDGE | | SWITCH ATM | |
| SWITCH | | SWITCH MULTICAPA | |
| ROUTER | | PC | |
| ROUTER CON CORTAFUEGOS | | PORTÁTIL | |
| CORTAFUEGOS | | SERVIDOR | |
| NAT | | | |
| RED (NUBE) | | | |

Universidad de Granada

13

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

13

Tema 1. Introducción a los fundamentos de redes

1. Sistemas de comunicación y redes

- Estructura y elementos de una red (© Kurose, Ros):

Clave:

- Host (+ sistema terminal)
- Servidor
- Dispositivo móvil
- Router móvil
- Switch de nivel de interfaz
- Módem
- Estación base inteligente
- Teléfono inteligente
- Torre de telefonía móvil
- Tableta
- Semáforo
- Termostato
- Nevera
- Monitor de pantalla plana
- Teléfono

Universidad de Granada

14

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

14



Esquema

1. Sistemas de comunicación y redes
2. **Diseño y estandarización de redes**
3. Terminología, conceptos y servicios
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios



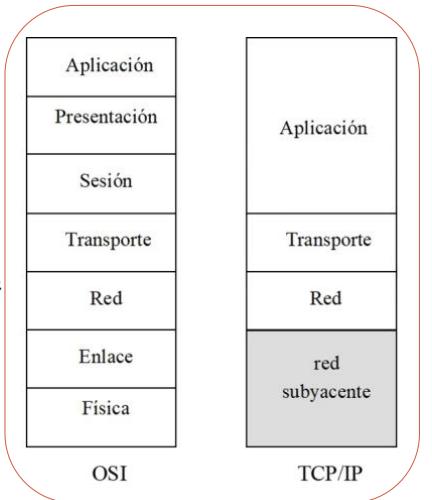
2. Diseño y estandarización de redes

- Problemas a resolver por la red (transparencia y eficacia):
 - ¿Cómo enviar físicamente la información?
 - Compartición del medio. Segmentación de la información
 - Control de flujo y de errores, salto a salto y también extremo a extremo
 - Control del encaminamiento (enrutamiento)
 - Control de congestión
 - Entrega ordenada de los mensajes
 - Gestión del diálogo o turno de palabra
 - Representación (sintaxis) de los datos
 - Significado (semántica) de los datos

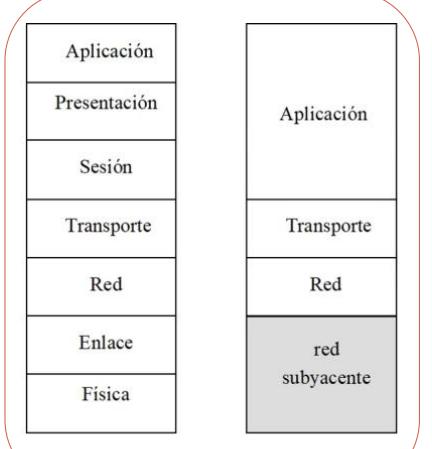
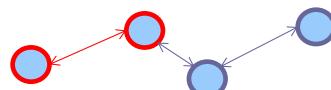




- Conceptos de diseño en redes:
 - Solucionar los problemas en capas
 - Concepto de "**Modelo de Referencia**"
→ definición de capas + funcionalidades
- Principios de diseño para el modelo de referencia:
 - Funcionalidades distintas deben estar en capas distintas
 - Minimizar el flujo de información entre capas
- Estándares internacionales:
 - 1) **Modelo OSI** (*Open System Interconnection*) de la ISO
 - 2) **TCP/IP** del Internet Engineering Task Force.
 - ¿Qué es un RFC?
<http://www.rfc-editor.org>



- **Modelo OSI**
 - Capa o nivel de aplicación
 - Capa o nivel de presentación
 - Capa o nivel de sesión
 - Capa o nivel de transporte
 - Capa o nivel de red
 - **Capa o nivel de enlace**
 - **Capa o nivel físico**



Tema 1. Introducción a los fundamentos de redes

2. Diseño y estandarización de redes

➤ **Modelo OSI**

- Capa o nivel de aplicación
- Capa o nivel de presentación
- Capa o nivel de sesión
- Capa o nivel de transporte
- **Capa o nivel de red**
- **Capa o nivel de enlace**
- **Capa o nivel físico**

| |
|--------------|
| Aplicación |
| Presentación |
| Sesión |
| Transporte |
| Red |
| Enlace |
| Física |

OSI

| |
|----------------|
| Aplicación |
| Transporte |
| Red |
| red subyacente |

TCP/IP

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

19

Tema 1. Introducción a los fundamentos de redes

2. Diseño y estandarización de redes

➤ **Modelo OSI**

- Capa o nivel de aplicación
- Capa o nivel de presentación
- Capa o nivel de sesión
- Capa o nivel de transporte
- **Capa o nivel de red**
- **Capa o nivel de enlace**
- **Capa o nivel físico**

| |
|--------------|
| Aplicación |
| Presentación |
| Sesión |
| Transporte |
| Red |
| Enlace |
| Física |

OSI

| |
|----------------|
| Aplicación |
| Transporte |
| Red |
| red subyacente |

TCP/IP

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

20



Tema 1. Introducción a los fundamentos de redes

2. Diseño y estandarización de redes

- **Modelo de referencia OSI**
 - Capa o nivel de aplicación
 - Capa o nivel de presentación
 - Capa o nivel de sesión
 - Capa o nivel de transporte
 - Capa o nivel de red
 - Capa o nivel de enlace
 - Capa o nivel físico
- **Modelo de referencia TCP/IP:**
 - Capa o nivel de aplicación
 - Capa o nivel de transporte
 - Capa o nivel de red
 - Red subyacente

| |
|--------------|
| Aplicación |
| Presentación |
| Sesión |
| Transporte |
| Red |
| Enlace |
| Física |

OSI

| |
|----------------|
| Aplicación |
| Transporte |
| Red |
| red subyacente |

TCP/IP

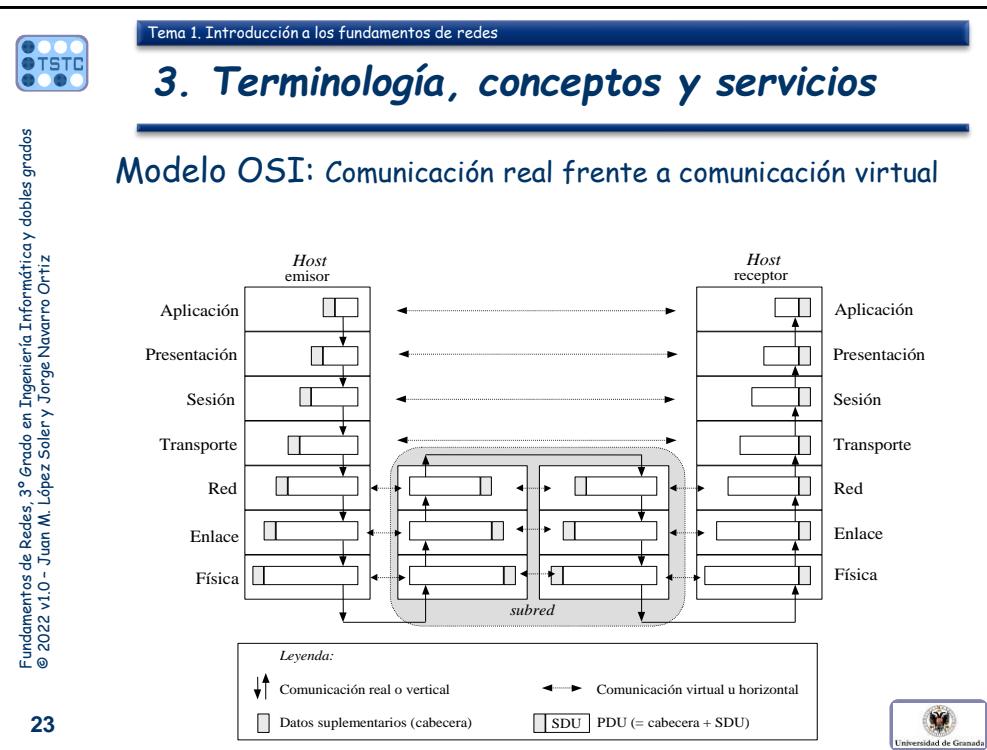
Universidad de Granada

Tema 1. Introducción a los fundamentos de redes

Esquema

1. Sistemas de comunicación y redes
2. Diseño y estandarización de redes
3. **Terminología, conceptos y servicios**
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios

Universidad de Granada



23

Tema 1. Introducción a los fundamentos de redes

3. Terminología, conceptos y servicios

➤ Terminología:

- **Comunicación real** (vertical)
- **Comunicación virtual** (horizontal)
- ○ **Entidad del nivel N** (N en OSI del 1 = físico al 7 = aplicación)
- ○ **Entidades pares**
- **Protocolo**
- ○ **Interfaz**
- **Servicio**
- ○ **Capa proveedora/usuario del servicio**
- **Pila de protocolos**
- **Arquitectura de red = Modelo de referencia + Pila de protocolos**
- ○ **Compartir una arquitectura de red extremo a extremo garantiza el "intercambio de información transparente" entre hosts.**

○ SAP (Service Access Point)
○ SDU (Service data Unit)
○ PDU (Protocol Data Unit)

Universidad de Granada

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

24

24

Tema 1. Introducción a los fundamentos de redes

3. Terminología, conceptos y servicios

Retardos en la comunicación (© Kurose, Ross)

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

25

Universidad de Granada

25

Tema 1. Introducción a los fundamentos de redes

3. Terminología, conceptos y servicios

Retardos en la comunicación

Física Física

(1) Transmisión $T_t = L(\text{bits})/V_t (\text{bps})$

(2) Propagación $T_p = D (\text{m})/V_p (\text{m/s})$

(3) Procesamiento (interno) + retardo en la cola + acceso al medio

t

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

26

Universidad de Granada

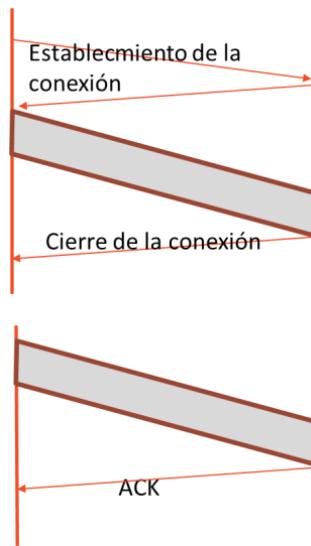
26



3. Terminología, conceptos y servicios

➤ Tipos de servicios

- Orientado a conexión (SOC)
- No orientado a conexión (SNOC)
- Confirmado (fiable)
- No confirmado (no fiable)



Esquema

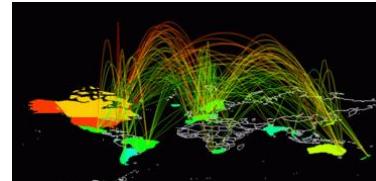
1. Sistemas de comunicación y redes
2. Diseño y estandarización de redes
3. Terminología, conceptos y servicios
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios



4. Internet: topología y direccionamiento

Historia de Internet:

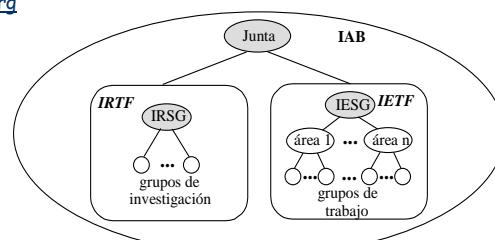
- 70s: DARPA inicia proyecto en redes con dos objetivos básicos:
 - Robustez en las comunicaciones.
 - Seguridad en las transmisiones.
- 1973: Metcalfe inventa Ethernet (tesis doctoral).
- 80s: La red creada se divide en dos:
 - ARPANET
 - MILNET
- 1983: Aparece el S.O. UNIX de BSD (Universidad de Berkeley), que incluye:
 - Nuevos protocolos: TCP/IP, el servicio de nombres DNS.
 - Utilidades de servicios de red.
 - La API socket.
- 1986: Aparece una nueva red troncal: NSFNET, motor impulsor de la actual Internet.
- 1989: Tim Berners Lee (CERN) crea el intercambio de hipertextos (HTTP, HTML).
- 1993: Primer navegador con interfaz gráfica (GUI): MOSAIC.
- 1996: Microsoft incorpora el "explorer" dentro del S.O. "...el ordenador es la red..."



4. Internet: topología y direccionamiento

Organización: ¿Quién fija las reglas?

- 1983: Se crea el IAB ("Internet Architecture Board") dependiente del "Department of Defense". <http://www.iab.org>
- 1989: IAB se independiza y se organiza en dos grupos.
 - IRTF ("Internet Research Task Force") . <http://www.irtf.org>
 - IETF ("Internet Engineering Task Force") 8 áreas, más de 100 grupos de trabajo. <http://www.ietf.org>



- Gestión de Internet: INTERNIC (www.internic.net) + IANA (www.iana.org, ahora es la ICANN (www.icann.org)).
- Normalización: DRAFT (borradores) que evolucionan a "Request For Comments" (www.rfc-editor.org). Ver RFC2026.



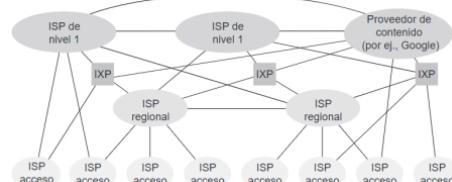


4. Internet: topología y direccionamiento

Topología jerárquica 3

Niveles:

- Intranets (Ethernet-WiFi) del usuario:
zona pública + zona privada
- Redes de acceso (xDSL, RDSI, FTTH, etc) del *Internet Service Provider* (ISP)
- Redes troncales (ATM, SDH, SONET, MPLS) de grandes operadores de telecomunicaciones
- Acuerdos de Peering y Tránsito
- Tier1, Tier2 y Tier3
- Puntos neutros ó PoP (Point of Presence) ó IXP (Internet eXchange Point)



<http://en.wikipedia.org/wiki/Peering>

http://en.wikipedia.org/wiki/Tier_1_network

http://en.wikipedia.org/wiki/Network_access_point

http://en.wikipedia.org/wiki/List_of_Internet_Exchange_Points_by_size

<http://espanix.net>



4. Internet: topología y direccionamiento

■ Redes Tier 1:

- De grandes operadores globales, al menos en 2 continentes
- Todas las redes Tier 1 están conectadas entre sí → backbone de Internet

■ Redes Tier 2:

- De ámbito más regional, necesitan pasar por una red Tier 1 para llegar a toda Internet. Ofrecen servicios de conectividad a operadores Tier 3.

■ Redes Tier 3:

- Operadores que dan servicio de conexión a Internet a usuarios y empresas → ISPs (*Internet Service Providers*)

■ Conexiones de tránsito Vs conexiones de peering

- IXP (*Internet eXchange Point*) → peering entre ISPs



Tema 1. Introducción a los fundamentos de redes

4. Internet: topología y direccionamiento

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

33

Universidad de Granada

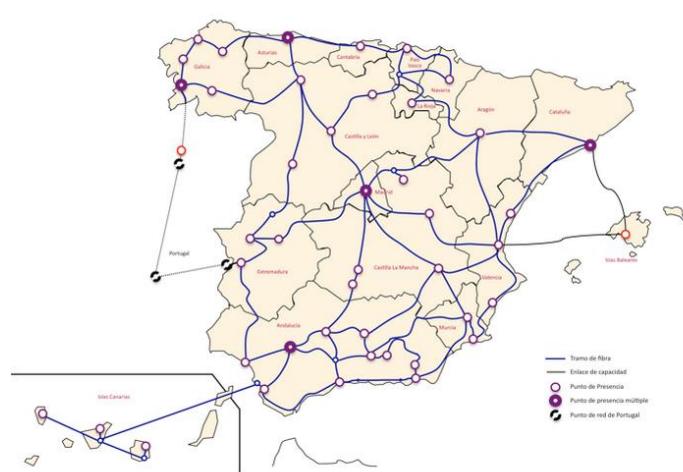
33

Tema 1. Introducción a los fundamentos de redes

4. Internet: topología y direccionamiento

Fundamentos de Redes, 3º Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

- Red Iris (www.rediris.es): Red académica e investigación



34

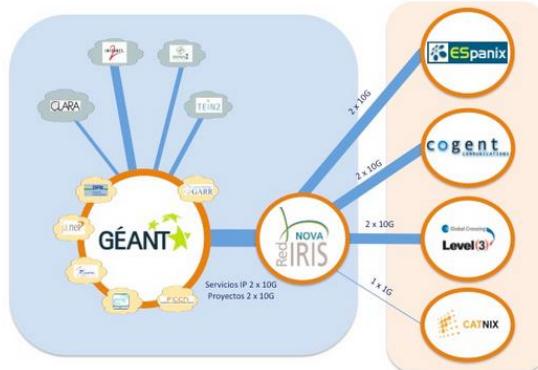
Universidad de Granada

34

Tema 1. Introducción a los fundamentos de redes

4. Internet: topología y direccionamiento

- Red Iris (www.rediris.es): Conexiones externas

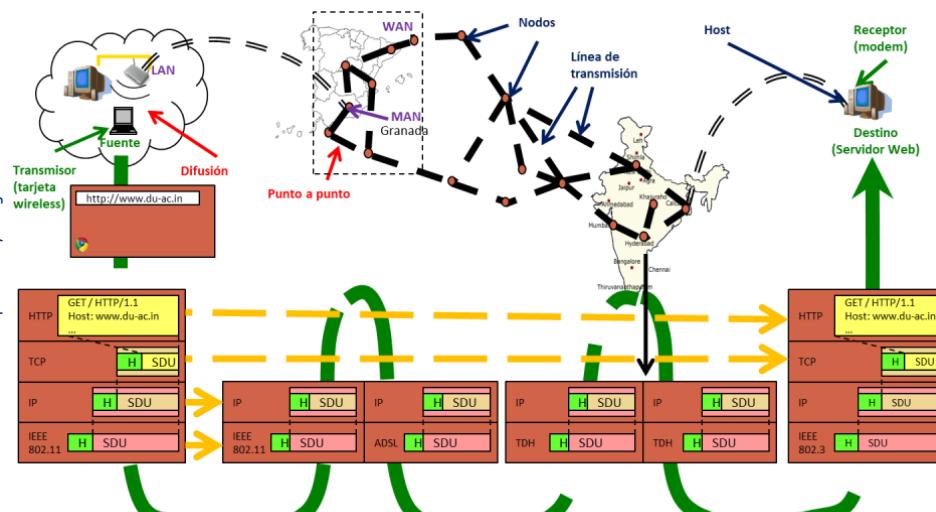


- Red Autonómica: RICA <https://www.cica.es/red-rica/trafico-rica/>



Tema 1. Introducción a los fundamentos de redes

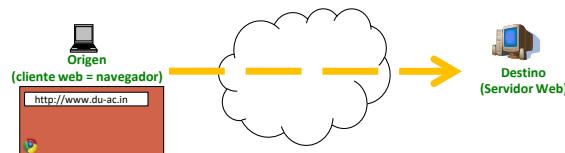
4. Internet: topología y direccionamiento





4. Internet: topología y direccionamiento

➤ Niveles de direccionamiento:



- URL → <http://www.du-ac.in/index.html> (nombre de dominio: du-ac.in) ➔ Capa de aplicación
- Puertos: identifica el proceso origen y destino ➔ Capa de transporte
- Dirección IP (identifica los hosts) ➔ Capa de red
 - Origen: 192.168.1.10
 - Destino: 69.162.68.236



Esquema

1. Sistemas de comunicación y redes
2. Diseño y estandarización de redes
3. Terminología, conceptos y servicios
4. Internet: topología y direccionamiento
5. Cuestiones y ejercicios





5. Cuestiones y ejercicios

1. Boletín de ejercicios resueltos Tema 1 página web de FR
2. Cuestiones y ejercicios del capítulo 1 de Kurose, Ross.
3. Cuestiones y ejercicios del capítulo 1 de García-Teodoro, Díaz-Verdejo, López-Soler



1.2. Tema 2



Tema 2 Capa de red

Fundamentos de Redes

Grado en Ingeniería Informática y dobles grados

Curso 3º

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones

E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2024



1

Tema 2. Capa de red



Esquema

1. Funcionalidades
2. Comutación
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

2





Objetivos del tema

- Comprender las funcionalidades y servicios de la capa de red :
 - Concepto de conmutación de paquetes y datagramas
 - Direcciónamiento en Internet
 - Encaminamiento salto a salto
 - Asociación con la capa de enlace a través del protocolo ARP
 - Señalización de errores mediante el protocolo ICMP



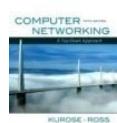
Bibliografía



Capítulo 6 y 9, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson, 2017, ISBN: 978-0-273-76896-8

- Apuntes de direcciónamiento IP en web de la asignatura

Para saber más...



Capítulo 4 James F. Kurose y Keith W. Ross. **COMPUTER NETWORKING. A TOP-DOWN APPROACH**, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675





Esquema

- 1. Funcionalidades**
2. Comutación
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)



1. Funcionalidades

- Funciones y servicios en TCP/IP
 - Encaminamiento
 - Comutación
 - Interconexión de redes
 - En OSI: control de congestión
- Ejemplos de protocolos de red:
 - X.25 https://es.wikipedia.org/wiki/Norma_X.25
 - IP





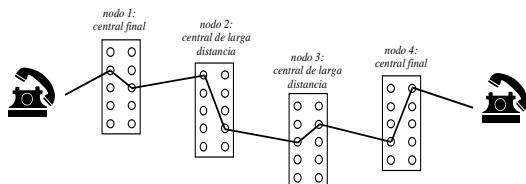
Esquema

1. Funcionalidades
2. **Comutación**
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)



2. Comutación

- Comutación = acción de establecer o determinar un camino que permite transmitir información extremo a extremo
- Esquemas de comutación
 - Circuitos
 - Paquetes: datagramas o circuitos virtuales
- Comutación de circuitos
 - Ej. Teléfono
 - Es un servicio orientado a conexión → exige un establecimiento de conexión previo a la transmisión



- Pasos: (i) Conexión, (ii) Transmisión, (iii) Desconexión
- Recursos dedicados. Facilita comunicaciones tiempo-real. No hay contención (contienda por acceder al medio).
- Retraso para establecimiento de la llamada. Poca flexibilidad para adaptarse a cambios. Poco tolerante a fallos.





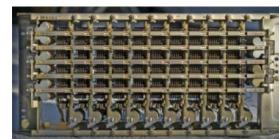
Tema 2. Capa de red

2. Conmutación

➤ Conmutación de circuitos

■ Ventajas

- La transmisión se realiza en **tiempo real**, adecuado para voz
- Uso permanente de recursos, el circuito se mantiene durante toda la sesión
- No hay **contención**, no hay contienda para acceder al medio
- El circuito es **fijo**, no hay decisiones de encaminamiento una vez establecido
- Simplicidad en la gestión de los nodos intermedios.



■ Desventajas

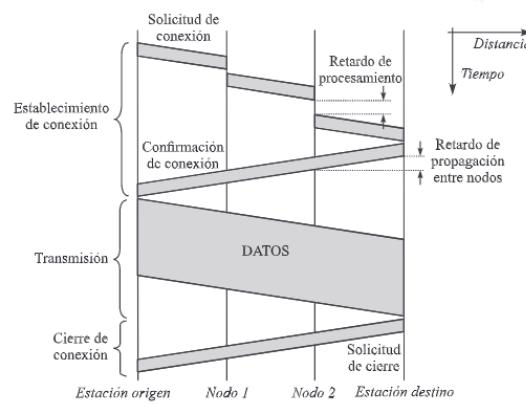
- Retraso en el inicio de la comunicación.
- En ocasiones uso no eficiente de recursos.
- El circuito es fijo. No se reajusta la ruta de comunicación.



Tema 2. Capa de red

2. Conmutación

➤ Conmutación de circuitos





Tema 2. Capa de red

2. Conmutación

Comutación de paquetes:

- Envío en bloques (paquetes)
- Conmutación mediante **datagramas**:
 - ej. IP
 - No hay conexión
 - Envío independiente, pueden seguir rutas diferentes
 - En cada salto: Almacenamiento y envío
 - Cada paquete debe contener las direcciones origen y destino
- Conmutación de paquetes con **circuitos virtuales**:
 - ej. ATM (troncales)
 - Pasos: (i) Conexión, (ii) Transmisión, (iii) Desconexión
 - Recursos no dedicados

estación origen nodo 1 nodo 2 estación destino

paquete 1 paquete 2 paquete 3

paquete 1 paquete 2 paquete 3

paquete 1 paquete 2 paquete 3

tiempo de procesamiento

tiempo

Universidad de Granada

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

11

11

Tema 2. Capa de red

2. Conmutación

4. Estime el tiempo involucrado en la transmisión de un mensaje de datos para la técnica de conmutación de paquetes mediante **datagramas (CDP)** considerando los siguientes parámetros:

- **M:** longitud en bits del mensaje a enviar.
- **V:** velocidad de transmisión de las líneas en bps.
- **P:** longitud en bits de los paquetes.
- **H:** bits de cabecera de los paquetes.
- **N:** número de nodos intermedios entre las estaciones finales.
- **D:** tiempo de procesamiento en segundos en cada nodo.
- **R:** retardo de propagación, en segundos, asociado a cada enlace.

Universidad de Granada

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

12

12



Para
comutación
de paquetes,
¿es válida la
respuesta de
ChatGPT?

2. Comutación

Fórmula de tiempo total involucrado:

1. Fragmentación del mensaje: Primero, el mensaje total M se fragmenta en paquetes de tamaño $P - H$ (donde H es la cabecera).
 - El número de paquetes sería aproximadamente $\lceil M/(P - H) \rceil$.
2. Tiempo de transmisión por paquete: Cada paquete tendrá un tiempo de transmisión en cada enlace de $\frac{P}{V}$ segundos.
3. Retardo de propagación: En cada enlace (entre nodos), habrá un retardo de propagación de R segundos.
4. Procesamiento en cada nodo: En cada nodo intermedio (total N nodos), hay un tiempo de procesamiento D .

Tiempo total estimado:

El tiempo total T sería:

$$T = \left(\frac{M}{P - H} \right) \times \left(\frac{P}{V} + R \right) + N \times D$$

Esto incluye:

- La transmisión de los paquetes a través de los enlaces, considerando el retardo de propagación R y la velocidad de transmisión V .
- El tiempo de procesamiento D en cada nodo intermedio.

Para
comutación
de circuitos,
¿es válida la
respuesta de
ChatGPT?

2. Comutación

1. Retardo de propagación total para el establecimiento del circuito:

El circuito debe establecerse entre los nodos antes de que se pueda transmitir el mensaje. Este proceso implica que la señal viaje de extremo a extremo.

El retardo total de propagación para el establecimiento sería:

$$T_{establecimiento} = (N + 1) \times R$$

Aquí, N es el número de nodos intermedios, y R es el retardo de propagación entre cada par de nodos.

2. Transmisión del mensaje:

Una vez que el circuito está establecido, se transmite el mensaje completo a través del enlace de datos sin dividirlo en paquetes. El tiempo de transmisión del mensaje será:

$$T_{transmisión} = \frac{M}{V}$$

Donde M es la longitud en bits del mensaje y V es la velocidad de transmisión de los enlaces en bits por segundo (bps).

3. Procesamiento en cada nodo:

Aunque no hay procesamiento de paquetes, puede haber un pequeño tiempo de procesamiento en cada nodo para manejar el paso de datos a través del circuito. Este tiempo sería:

$$T_{procesamiento} = N \times D$$

Donde D es el tiempo de procesamiento en cada nodo, y N es el número de nodos intermedios.

Tiempo total involucrado:

El tiempo total para la transmisión del mensaje usando comutación de circuitos sería la suma de los tres componentes:

$$T_{total} = T_{establecimiento} + T_{transmisión} + T_{procesamiento}$$

Sustituyendo las expresiones:

$$T_{total} = (N + 1) \times R + \frac{M}{V} + N \times D$$

Este tiempo incluye:

- El retardo de propagación para establecer el circuito.
- El tiempo de transmisión del mensaje a través del circuito.
- El tiempo de procesamiento en cada nodo. ↴



Esquema

1. Funcionalidades
2. Comutación
3. **El protocolo IP**
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)



3. El protocolo IP

- IPv4 está especificado en el RFC 791:
 - Es un protocolo para la **interconexión** de redes (también llamadas **subredes**).
 - Resuelve el **direcccionamiento** en Internet.
 - Realiza la **retransmisión salto a salto** entre **hosts** y **routers**. Ofrece un servicio **no orientado a conexión y no fiable**:
 - No hay negociación o "handshake", no hay una conexión lógica entre las entidades.
 - No existe control de errores ni control de flujo.
 - La unidad de datos (paquete) de IP se denomina **datagrama**.
 - IP es un protocolo de **máximo esfuerzo** ("best-effort"), es decir los datagramas se pueden perder, duplicar, retrasar, llegar desordenados.
 - IP gestiona la "**fragmentación**": adaptar el tamaño del datagrama a la diferentes **Maximum Transfer Units (MTUs)** de las subredes hasta llegar al destino.





3. El protocolo IP

➤ Direcciones IP:



Servidor
Webmail
130.206.192.39



www.youtube.com
172.194.34.206



Servidor
Spotify
78.31.8.101



Google España

www.google.com =
172.194.34.209



www.ugr.es =
150.214.204.25
dns3.ugr.es =
150.214.191.10
pop.ugr.es = 150.214.20.3



3. El protocolo IP

- Internet adopta un **direcciónamiento jerárquico** para simplificar el **routing**.
- Las direcciones IP (32 bits) tienen dos partes bien diferenciadas:
un identificador de la subred y **un identificador del dispositivo** dentro de esa subred.
- Cada subred tiene un identificador único en la intranet.
- Cada dispositivo tiene un identificador único en la subred.
- La **máscara de red** es un patrón que determina qué bits pertenecen al identificador de subred
 - a) Dirección IP → 200.27.4.112 = 11001000.00011011.00000100.01110000
Máscara → 255.255.255.0 = 11111111.11111111.11111111.00000000
 - b) La máscara se puede representar de forma compacta, por ejemplo 200.27.4.112/24
- Para obtener la dirección o identificador de la subred:

| | |
|---------------------|---------------------------------------|
| 200.27.4.112 | = 11001000.00011011.00000100.01110000 |
| & | & |
| 255.255.255.0 | = 11111111.11111111.11111111.00000000 |
| ----- | |
| Subred → 200.27.4.0 | = 11001000.00011011.00000100.00000000 |



Tema 2. Capa de red

3. El protocolo IP

➤ Podemos considerar Internet como un conjunto de subredes interconectadas

➤ ¿Qué es una **subred**? ¿Qué es un **switch**? ¿Qué es un **router**?

Computer Networking. A Top-down Approach. de James F. Kurose y Keith W. Ross:
"Para determinar las subredes, separe cada interfaz de los *hosts* y *routers*, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes."

Universidad de Granada

19

19

Tema 2. Capa de red

3. El protocolo IP

➤ ¿Qué es una **subred**? ¿Qué es un **switch**? ¿Qué es un **router**?

¿Quién tiene direcciones IP?
Los *hosts* y los *routers* tienen 1 IP por cada interfaz.
Los *switches* NO tienen direcciones IP

Computer Networking. A Top-down Approach. de James F. Kurose y Keith W. Ross:
"Para determinar las subredes, separe cada interfaz de los hosts y routers, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes."

Universidad de Granada

20

20

Tema 2. Capa de red

3. El protocolo IP

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

- ¿Cómo se elige la máscara? ➔ Según el **número de dispositivos** previsibles en la subred tal que se ajusta para no desaprovechar direcciones. Recuérdese: cada subred tiene un identificador único en nuestra intranet.

Dirección IP ➔ 200.27.4.112 = 11001000.00011011.00000100.01110000
Máscara ➔ 255.255.255.0 = 11111111.11111111.11111111.00000000

- **# dispositivos** = $2^{\# \text{ ceros}} - 2$ ➔ ej. 8 ceros (/24) permite 254 dispositivos
- El -2 viene de que la primera (000...0) y última (111...1) están reservadas.
Por ejemplo en la subred 200.27.4.0/24 no se pueden asignar como id. de dispositivo
 - 200.27.4.0 = 11001000.00011011.00000100.00000000 ➔ Reservada (subred)
 - 200.27.4.1 = 11001000.00011011.00000100.00000001 ➔ Dispositivo #1
 - ...
 - 200.27.4.254 = 11001000.00011011.00000100.11111110 ➔ Dispositivo #254
 - 200.27.4.255 = 11001000.00011011.00000100.11111111 ➔ Reservada (difusión)

21

21

Tema 2. Capa de red

3. El protocolo IP

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

- **Direcciones públicas**
 - Cada dirección se asigna a sólo 1 dispositivo en Internet.
Se asignan centralizadamente
- **Direcciones privadas**
 - Sólo en intranets. Se pueden repetir en distintas intranets.
Las asigna el usuario según su criterio.

22

22



3. El protocolo IP

➤ Direcciones IP: CLASES (ver RFC 1166)

- Los *hosts* y *routers* tienen una IP por cada una de sus interfaces.
- 32 bits, notación decimal con puntos. Ejemplo: 192.168.212.60
- 5 clases de direcciones IP
- Clases A,B,C → Jerárquicas a dos niveles:
identificador de red + identificador de dispositivo (*host*)

| | | | |
|---------|-----------|--|----------------|
| Clase A | 0 | red (7 bits) | host (24 bits) |
| Clase B | 1 0 | red (14 bits) | host (16 bits) |
| Clase C | 1 1 0 | red (21 bits) | host (8 bits) |
| Clase D | 1 1 1 0 | dirección grupo <i>multicast</i> (28 bits) | |
| Clase E | 1 1 1 1 0 | | uso futuro |



3. El protocolo IP

5 clases de direcciones (cont.):

Rangos:

A → 0.0.0.0-127.255.255.255 ⇒ 128 redes x 16.777.216 hosts
 B → 128.0.0.0-191.255.255.255 ⇒ 16.384 redes x 65.536 hosts
 C → 192.0.0.0-223.255.255.255 ⇒ 2.097.152 redes x 256 hosts
 D → 224.0.0.0-239.255.255.255 ⇒ para *multicast*
 E → 240.0.0.0-255.255.255.255 ⇒ usos futuros

Reglas especiales:

host = 00..0 ⇒ identifica a una red, nunca es una dirección origen, no se usa para dispositivos

host = 11..1 ⇒ difusión en la red especificada, es una dirección destino, no se usa para dispositivos

127.0.0.0 ⇒ autobucle (*loopback*)

Para evitar ambigüedades el identificativo de dispositivo no debe ser ni 255 ni 0

Reserva de direcciones privadas (RFC1918):

Clase A → 10.0.0.0 → 1 Red privada clase A

Clase B → 172.16.0.0 - 172.31.0.0 → 16 redes privadas clase B

Clase C → 192.168.0.0 - 192.168.255.0 → 256 redes privadas clase C

Gestión/asignación: IANA (www.iana.org) ahora gestionada por ICANN (www.icann.org)





3. El protocolo IP

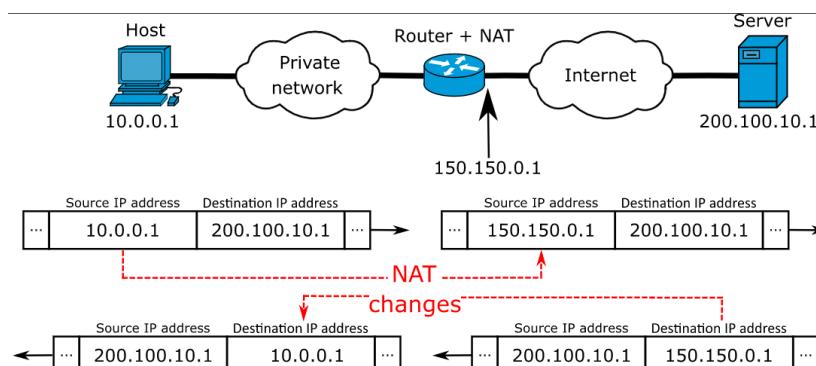
- Los bloques de **direcciones IPv4** se “agotaron” ya (Nov. 2019)!!!
- Sólo quedan disponibles bloques /24 (256 direcciones) a /32 (1 dirección).
- Se van recopilando direcciones de sitios obsoletos, empresas que hayan desaparecido, proyectos terminados, hosting que ya no está en uso...

IPv6

- IPv6 usa un esquema de **direcccionamiento de 128 bits**.
- Notación **hexadecimal**. 8 grupos de 4 dígitos, separados por “:”.
- Cada dígito hexadecimal corresponde a 4 dígitos en binario (4 bits).
- Rango: 0000:0000:0000:0000:0000:0000:0000:0000 a FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
- 340.282.366.920.938.463.463.374.607.431.768.211.456 (340 sextillones) direcciones diferentes.
- **Compatible con IPv4**.

3. El protocolo IP

NAT (Network Address Translation) (RFC 1631, 2663, 3022)



es un método para reasignar un espacio de direcciones IP (típicamente privadas) a otro (públicas) modificando la dirección IP de los paquetes mientras se retransmiten a través de un router



3. El protocolo IP

Network Address Translation (RFC 1631, 2663, 3022)

- Optimiza el uso de direcciones públicas mediante la utilización de direcciones privadas.
- Reemplaza las direcciones privadas origen salientes por públicas y al revés con las entrantes.



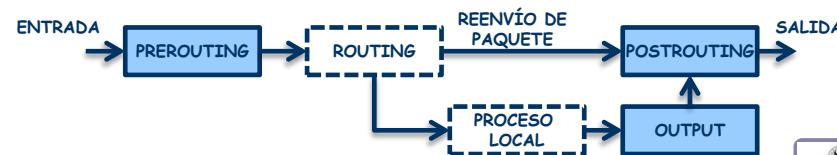
- Tabla de traducciones.
- IMPORTANTE: No se pueden implementar servidores detrás de un NAT. Por ello, se establece la zona pública (DMZ) y la zona privada.



3. El protocolo IP

Problema de escasez de direcciones IP

- Se necesitan m direcciones pero se disponen de n , siendo $n < m$.
- Si $n = 1$ se denomina enmascaramiento (**masquerading**).
- Se usa en ISPs, para así poder dar acceso a más usuarios que direcciones IP tenga el ISP. Se supone que no todos los usuarios acceden simultáneamente. Las direcciones se asignan a los usuarios de forma dinámica.
- **SNAT:** Source NAT → el origen de los datos está en la red privada; cambia la dirección IP de origen; se realiza tras el encaminamiento (*postrouting*)
- **DNAT:** Destination NAT → el origen de los datos está en la red pública; cambia la dirección IP de destino; requiere configurar en el router qué puerto irá dirigido a qué máquina; se realiza antes del encaminamiento (*prerouting*)



Tema 2. Capa de red

3. El protocolo IP

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

29

The diagram shows a client (10.0.0.3) connecting to a server (128.32.32.68) through a NAT device. Step 1: Client sends SYN packet to server. Step 2: NAT adds entry and modifies packet. Step 3: Server responds with SYN, ACK. Step 4: NAT translates response back to client.

| PROTO | TCP |
|-------|--------------|
| SADDR | 10.0.0.3 |
| DADDR | 128.32.32.68 |
| SPORT | 1049 |
| DPOR | 80 |
| FLAGS | SYN |
| CKSUM | 0x1636 |

1. El cliente intenta conectarse al servidor web 128.32.32.68 y envíe un paquete SYN con su dirección IP interna 10.0.0.3 (privada).

| PROTO | TCP |
|-------|--------------|
| SADDR | 24.1.70.210 |
| DADDR | 128.32.32.68 |
| SPORT | 40960 |
| DPOR | 80 |
| FLAGS | SYN |
| CKSUM | 0x2436 |

2. El dispositivo NAT ve la configuración del paquete, añade una nueva entrada a su tabla de traducción. Luego modifica el paquete usando su dirección IP externa (pública), cambia el puerto y el chequeo de integridad del paquete.

| PROTO | TCP |
|-------|--------------|
| SADDR | 128.32.32.68 |
| DADDR | 24.1.70.210 |
| SPORT | 80 |
| DPOR | 40960 |
| FLAGS | SYN, ACK |
| CKSUM | 0x8041 |

3. El servidor responde con un paquete SYN, ACK. El paquete se envía a la dirección IP externa (pública) del dispositivo NAT.

| Original | NAT |
|---------------|-------------------|
| 10.0.0.3:1049 | 24.1.70.210:40960 |
| ... | ... |

29

Tema 2. Capa de red

3. El protocolo IP

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

30

➤ Ejercicio: Asignar direcciones

- Subredes corporativas: 30 dispositivos, direcciones privadas 192.168.0.0
- Subred de acceso: dirección pública (ISP)

The network diagram shows three hosts (Host A, Host B, Host C) connected to a router (R1). R1 is connected to another router (R2), which is connected to the Internet. This represents a corporate network structure where multiple hosts share a single public IP address via a router.

Universidad de Granada

30

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Tema 2. Capa de red

3. El protocolo IP

➤ Ejercicio: Asignar direcciones

- Subredes corporativas: 30 dispositivos, direcciones privadas 192.168.0.0 → 5 ceros, /27
- Subred de acceso: dirección pública (ISP) → 2 ceros, /30, 150.214.190.0 (UGR)

Host A: 192.168.0.2, Subred = 192.168.0.0

Host B: 192.168.0.35, Subred = 192.168.0.32

Host C: 192.168.0.66, Subred = 192.168.0.64

R1: 192.168.0.1, Subred = 192.168.0.33

R2: 192.168.0.34, Subred = 150.214.190.0

Internet: 150.214.190.1, Subred = 150.214.190.0

31

Universidad de Granada

31

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Tema 2. Capa de red

3. El protocolo IP

➤ El encaminamiento

- Encontrar el mejor camino para llevar la información (paquetes) de un origen a un destino dado.
- Se decide **paquete a paquete** y **salto a salto** en función de la **IP destino del paquete** y de las **tablas de encaminamiento** residentes en cada uno de los **routers**.

Host A

Host B

Host C

R1

R2

Internet

Encaminamiento

Almacenamiento & Retransmisión

32

Universidad de Granada

32

Tema 2. Capa de red

3. El protocolo IP

➤ Retransmisión salto-a-salto:

- Resolución local del camino
- En el dispositivo origen y todos los intermedios

Host A → R1 → R2 → Host C

Internet

Host A: ¿Para llegar a C? Lo envío a R1

R1: ¿Para llegar a C? Lo envío a R2

R2: ¿Para llegar a C? Lo envío a C

Host C

33

Universidad de Granada

33

Tema 2. Capa de red

3. El protocolo IP

El encaminamiento se realiza **salto a salto** y **datagrama a datagrama** (IP es no orientado a conexión).

❑ Modos de encaminamiento: **directo** y **no directo**.

❑ Cada dispositivo (*host o router*) tiene una tabla de encaminamiento.

❑ Un *router* suele estar en varias redes distintas, un *host* suele estar en solo una

¿Falta algún destino?

192.100.13.154 192.100.13.176
H1 H2
192.100.13.0
192.100.13.129
R1
150.100.0.1 Hacia Internet
192.100.12.7 192.100.12.24 192.100.12.55
H3 H4
192.100.12.0
192.100.12.1
R2
192.100.15.54 192.100.15.11
H5
192.100.15.0

Tabla de R1, * = routing directo

| i | Destino (D_i) | Salto siguiente (S_i) | Máscara (M_i) | Flags | Interfaz(I_i) |
|---|-------------------|---------------------------|-------------------|-------|-------------------|
| 1 | 127.0.0.1 | * | 255.255.255.255 | H | lo |
| 2 | 192.100.12.0 | * | 255.255.255.0 | - | eth0 |
| . | 192.100.13.0 | * | 255.255.255.0 | - | eth1 |
| . | 192.100.15.0 | 192.100.12.1 | 255.255.255.0 | G | eth0 |
| N | Default | 150.100.0.222 | 0.0.0.0 | G | eth2 |

192.100.15.0
❑ En caso de conflicto se elige la ruta con máscara más larga

34

34

3. El protocolo IP

- Si no hay fragmentación y no hay "traducción de direcciones" (NAT) el datagrama (salvo el TTL, las opciones y el campo de comprobación) no se modifica en el camino.
- Proceso de encaminamiento en los nodos IP (salto a salto) por cada datagrama:
 - Se extrae la dirección destino: IP_DESTINO del datagram
 - Por cada entrada i con $i = 1, \dots, N$, de la tabla de encaminamiento se calcula

$$IP_i = IP_DESTINO \text{ AND}(\&) \text{ MASCARA}_i$$

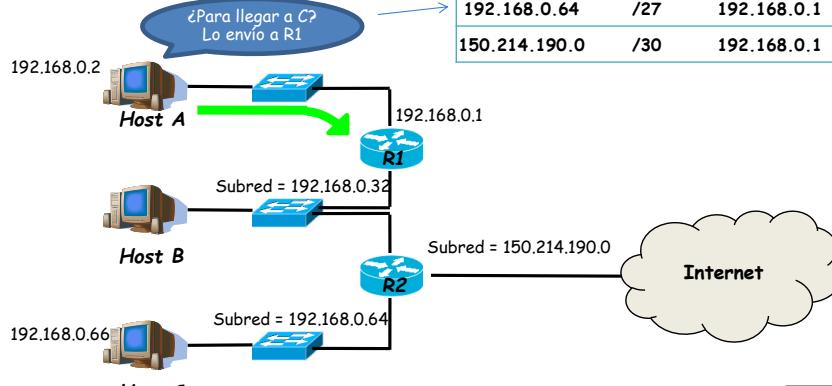
- Si $IP_i = Di$ y
si es routing directo (*) → reenviar el datagrama al destino final por la interfaz i
o si no es routing directo → reenviar el datagrama al salto siguiente por la interfaz i
- Si hay varias coincidencias se elige el destino con la máscara más larga
- Si se ha barrido toda la tabla y no hay coincidencia con ninguna fila → error (posible mensaje ICMP)
- Para encapsular el datagrama en la trama física correspondiente, se debe consultar la tabla ARP (ver más adelante) y en caso de no conocer la dirección física se envía un broadcast con protocolo ARP para obtener la dir. física.



3. El protocolo IP

➤ Tabla de encaminamiento:

| Dirección IP destino | Máscara | Siguiente nodo |
|----------------------|---------|----------------|
| 192.168.0.0 | /27 | - |
| 192.168.0.32 | /27 | 192.168.0.1 |
| 192.168.0.64 | /27 | 192.168.0.1 |
| 150.214.190.0 | /30 | 192.168.0.1 |





3. El protocolo IP

En el origen y en cada router se coteja la tabla:

- Dirección de destino (DD): 192.168.0.66
- Para cada entrada (fila en la tabla)
 - DD & Máscara = A
 - ¿A = Dirección de destino?
SI → elegir el "Siguiente Nodo" → consultar TABLA ARP
NO → seguir buscando
 - 192.168.0.66 & /27 =
11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
 - ¿192.168.0.64 = 192.168.0.0? NO
 - 192.168.0.66 & /27 =
11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
 - ¿192.168.0.64 = 192.168.0.32? NO
 - 192.168.0.66 & /27 =
11000000.10101000.00000000.010**00010** & /27 = 192.168.0.64
 - ¿192.168.0.64 = 192.168.0.64? SÍ ➔ Siguiente Nodo = 192.168.0.1
 - 192.168.0.66 & /30 =
11000000.10101000.00000000.010000**10** & /30 = 192.168.0.64
 - ¿192.168.0.64 = 150.214.190.0? NO
- 37 ➤ Si hay más de una coincidencia (colisión) se elige la entrada de máscara más restrictiva (+1s)

| Dirección IP destino | Máscara | Siguiente nodo |
|----------------------|---------|----------------|
| 192.168.0.0 | /27 | - |
| 192.168.0.32 | /27 | 192.168.0.1 |
| 192.168.0.64 | /27 | 192.168.0.1 |
| 150.214.190.0 | /30 | 192.168.0.1 |



37

37



3. El protocolo IP

➤ Tabla de encaminamiento:

➤ Problemas:

- La tabla del ejemplo NO direcciona Internet (ej. www.google.com = 172.194.34.209)
- La topología implica sólo un camino de salida desde A ➔ ¿necesitamos 4 entradas?

| Dirección IP destino | Máscara | Siguiente nodo |
|----------------------|---------|----------------|
| 192.168.0.0 | /27 | - |
| 192.168.0.32 | /27 | 192.168.0.1 |
| 192.168.0.64 | /27 | 192.168.0.1 |
| 150.214.190.0 | /30 | 192.168.0.1 |

!!Usar la entrada por defecto!! ➔ /0



38

38



Tema 2. Capa de red

3. El protocolo IP

➤ Tabla de encaminamiento:

➤ Problemas:

- La tabla del ejemplo NO dirige Internet (ej. www.google.com = 172.194.34.209)
- La topología implica sólo un camino de salida desde ➤ ¿necesitamos 4 entradas?

| Dirección IP destino | Máscara | Siguiente nodo |
|----------------------|---------|----------------|
| 192.168.0.0 | /27 | - |
| 0.0.0.0 | 0.0.0.0 | 192.168.0.1 |

R1 192.168.0.1 → Host C 192.168.0.66

Host C 192.168.0.66 → www.google.com = 172.194.34.209

Host C 192.168.0.66 → Microsoft Hotmail 130.206.192.39

Host C 192.168.0.66 → Servidor Webmail 78.31.8.101

Host C 192.168.0.66 → YouTube www.youtube.com 172.194.34.206

Host C 192.168.0.66 → Universidad de Granada dns3.ugr.es = 150.214.191.10
pop.ugr.es = 150.214.20.3

39

39

Tema 2. Capa de red

3. El protocolo IP

➤ Ejercicio: Diseñar la Tabla de encaminamiento en R2

- Incorporar todas las redes directamente conectadas.
- Incorporar la entrada por defecto
- Añadir todas las entradas adicionales necesarias.

| Dirección IP destino | Máscara | Siguiente nodo |
|----------------------|---------|----------------|
| 192.168.0.32 | /27 | - |
| 192.168.0.64 | /27 | - |
| 150.214.190.0 | /30 | - |
| 0.0.0.0 | /0 | 150.214.190.2 |
| 192.168.0.0 | /27 | 192.168.0.33 |

Host A Subred = 192.168.0.0 → **R1**

Host B Subred = 192.168.0.32 → **R1**

Host C Subred = 192.168.0.64 → **R1**

R1 → **R2**

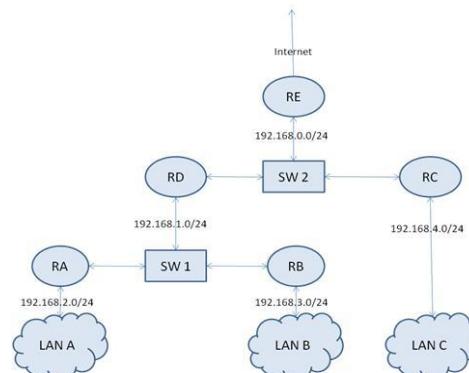
R2 → Internet Subred = 150.214.190.0

40

40

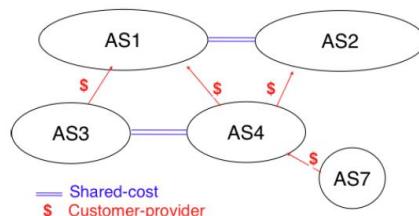
3. El protocolo IP

7. Imagine una situación donde hay cinco routers RA-RE. RA, RB y RC se conectan cada uno a una red local A, B y C, siendo cada router única puerta de enlace de cada red. RA, RB y RD están conectados entre sí a través de un switch. RC, RD y RE están conectados entre sí a través de un switch. RE conecta a Internet a través de la puerta de acceso especificada por el ISP. Especifique tablas de encaminamiento en los routers. Asigne a voluntad las direcciones IP e interfaces necesarias.



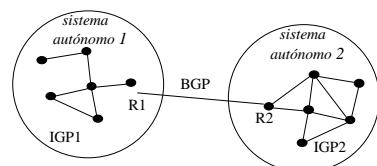
3. El protocolo IP

- ❑ Para facilitar la administración y aumentar la escalabilidad Internet se jerarquiza en **Sistemas Autónomos** (SA).
- ❑ Un **SA** es un conjunto de redes y routers administrados por una autoridad.
- ❑ Cada SA informa a los otros SA de las redes accesibles. Existe un router responsable, denominado **router exterior** (R1, R2, Rn).
- ❑ Cada SA se identifica por un entero de 16 bits (DESDE 2007 ES 32-BITS). Rediris = AS766



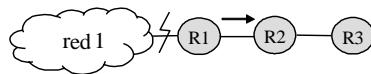


3. El protocolo IP



3. El protocolo IP

- **RIP** ("Routing Information Protocol" RFC 1058, 2453, 4822)
 - Protocolo de la capa de aplicación (opera sobre UDP puerto 520)
 - Adopta un algoritmo *vector-distancia* (métrica basada en número de saltos)
 - Períódicamente (por defecto cada 30 segundos) cada *router* RIP recibe de todos sus vecinos (dirección multicast 224.0.0.9) los vectores-distancia para todos los posibles destinos
 - De entre ellos, para un destino dado, se selecciona como sato siguiente el vecino que anuncie el menor coste, actualizando la métrica para ese destino sumando uno al coste anunciado
 - Problema de la convergencia lenta: las malas noticias tardan en propagarse
 - Problema de la "cuenta al infinito".
 - Soluciones:
 - Split horizon*
 - Hold down*
 - Poison reverse*
 - Ver > *man routed* (SO Linux)



Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

45

Tema 2. Capa de red

3. El protocolo IP

OSPF (RFC 2328)

- Basado en estado del enlace (coste $\alpha 1 / \text{velocidad del enlace}$)
- Permite rutas alternativas y balanceo de carga
- Gestión en base a áreas independientes
- Minimiza difusión mediante routers designados
- Mensajes: hello, database description, link status request/update/ack

Ejemplo para RIP y OSPF

Universidad de Granada

45

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

46

Tema 2. Capa de red

3. El protocolo IP

➤ Formato de datagrama

| | | | | | |
|----------------------|-----------|--------------|----------------|----|----------------|
| 0 | 4 | 8 | 16 | 19 | 31 |
| V | LC | TS | longitud total | | |
| | | | identificación | I | desplazamiento |
| TTL | protocolo | comprobación | | | |
| dirección IP origen | | | | | |
| dirección IP destino | | | | | |
| opciones | | | relleno | | |
| datos | | | | | |

cabecera

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: ip Expression... Clear Apply

No. 215 4: 848984 Time 88.188.158.15 Source 150.214.191.5 Destination Protocol Info

Frame 215: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
Ethernet II, Src: Cisco_b7:64:00 (00:07:0d:b7:64:00), Dst: Micro-St_a8:f7:63 (00:24:21:a8:f7:63)
Internet Protocol, Src: 88.188.158.15 (88.188.158.15), Dst: 150.214.191.5 (150.214.191.5)

Version: 4 Header Length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 75 Identification: 0x87c (29516)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0 Time to live: 100 Protocol: TCP (6)
Header checksum: 0xdb88 [correct]
Source: 88.188.158.15 (88.188.158.15)
Destination: 150.214.191.5 (150.214.191.5)
Transmission Control Protocol, Src Port: wap-push-http (4035), Dst Port: 23691 (23691), Seq: 36, Ack: 36, Data (35 bytes)

Universidad de Granada

46

Tema 2. Capa de red

3. El protocolo IP

➤ Formato de datagrama

| | | | | | |
|---|----|----|----------------------|-----------|----------------|
| | | | | | |
| V | LC | TS | longitud total | | |
| | | | identificación | I | desplazamiento |
| | | | TTL | protocolo | comprobación |
| | | | dirección IP origen | | |
| | | | dirección IP destino | | |
| | | | opciones | relleno | |
| | | | datos | | |

cabecera

➤ Fragmentación IPv4:

- Tamaño máximo del datagrama: $2^{16}-1 = 65.535$ bytes.
- Es necesario adaptarse a la **MTU** (Maximum Transfer Unit) de cada subred
- El ensamblado sólo se puede hacer en el destino final
- **desplazamiento**: offset respecto del comienzo del paquete.
- **indicadores (I)** : "Don't Fragment", "More Fragments".

| Nivel de enlace | MTU (bytes) |
|-----------------------|--------------------|
| PPP normal | 1500 |
| PPP bajo retardo | 296 |
| X.25 | 1600 (RFC 1356) |
| Frame Relay | 1600 (normalmente) |
| Ethernet DIX | 1500 |
| Ethernet LLC-SNAP | 1492 |
| Token Ring 4 Mb/s | 4440 (THT 8ms) |
| Classical IP over ATM | 9180 |

47

47

Tema 2. Capa de red

3. El protocolo IP

➤ Fragmentación IPv4:

red Ethernet MTU=1.500

red 1 R1 red 2

red 1 R1 red 2

datagrama IP n

cabecera 20 bytes datos 4.180 bytes

cabecera 20 bytes datos 1.480 bytes

cabecera 20 bytes datos 1.480 bytes

cabecera 20 bytes datos 1.220 bytes

fragmento 1, MF =1

fragmento 2, MF =1

fragmento 3, MF =0

Universidad de Granada

48

48



Tema 2. Capa de red

3. El protocolo IP

Diferencias entre IPv4 e IPv6:

| Característica | IPv4 | IPv6 |
|---------------------------|---------------------------------------|--|
| Longitud de la dirección | 32 bits | 128 bits |
| Espacio de direcciones | ~4,3 mil millones | 340 undecillones (casi ilimitado) |
| Formato | Decimal (ej. 192.168.0.1) | Hexadecimal (ej. 2001:0db8:8a2e:0370:7334) |
| Configuración | Manual o DHCP | Autoconfiguración sin estado (SLAAC) |
| Seguridad | Opcional (IPsec) | IPsec obligatorio |
| Fragmentación | Los routers pueden fragmentar | Solo el dispositivo emisor fragmenta |
| Encabezado de paquetes | Complejo y variable | Simplificado y fijo |
| QoS | Limitado (TOS) | Optimizado (Flow Label) |
| Compatibilidad | Amplia, pero limitada por direcciones | No compatible directamente con IPv4 |
| Optimización para móviles | Menos eficiente | Mejor rendimiento en redes móviles |

49

Tema 2. Capa de red

Esquema

1. Funcionalidades
2. Comutación
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (DHCP)

50



Tema 2. Capa de red

4. Protocolo ARP

➤ Direcciones MAC

- Tras la redirección IP → Enviar a la MAC del siguiente nodo

Internet

Host A

Host B

Host C

R1

R2

IEEE 802

Capa de Aplicación
Capa de Transporte
Capa de Red
Red Subyacente

Subcapa de enlace
Subcapa MAC
Capa Física

Universidad de Granada

51

51

51

Tema 2. Capa de red

4. Protocolo ARP

➤ Direcciones MAC

- Tras la redirección IP → Enviar a la Medium Access Control (MAC) del siguiente nodo. Se usan en redes Ethernet (cableadas) y Wifi
- Formato (6 bytes): HH-HH-HH-HH-HH-HH → e.j. 00-24-21-A8-F7-6A
- Son únicas, asignadas por IEEE en lotes de 2^{24} para cada fabricante
- Dirección de difusión (broadcast) FF-FF-FF-FF-FF-FF
- Protocolo: Address Resolution Protocol (ARP)
Obtener MAC a partir de IP: (a) y (b)
- Protocolo: Reverse ARP (RARP)
Obtener IP a partir de MAC: (a) y (c)

(a)

(b)

(c)

A

B

C

D

A

B

C

D

Universidad de Granada

52

52



Tema 2. Capa de red

4. Protocolo ARP

➤ **Formato ARP:**

| | | | |
|---------------------|------|---------------------|----|
| 0 | 8 | 16 | 31 |
| Htipo | | Ptipo | |
| Hlen | Plen | Operación | |
| Hemisor (bytes 0-3) | | | |
| Hemisor (bytes 4-5) | | Pemisor (bytes 0-1) | |
| Pemisor (bytes 2-3) | | Hsol (bytes 0-1) | |
| Hsol (bytes 2-5) | | | |
| Psol (bytes 0-3) | | | |

Sniffer Captura:

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: arp Expression... Clear Apply

No. Time Source Destination Protocol Info

6 0.106885 AsustekC_a2:68:bd Broadcast ARP who has 150.214.191.10? Tel

Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: AsustekC_a2:68:bd (90:e6:ba:a2:68:bd), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
[Is gratuitous: False]
Sender MAC address: AsustekC_a2:68:bd (90:e6:ba:a2:68:bd)
Sender IP address: 150.214.191.178 (150.214.191.178)
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 150.214.191.10 (150.214.191.10)

Universidad de Granada

53

53

Esquema

1. Funcionalidades
2. Comutación
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. **El protocolo ICMP**
6. Autoconfiguración de la capa de red (DHCP)

54



54



Tema 2. Capa de red

5. El protocolo ICMP

➤ ICMP (Internet Control Message Protocol)

- Informa sobre situaciones de error en IP → es un protocolo de señalización
- Suelen ir (excepto eco y solicitudes) hacia el origen del datagrama IP original
- ICMP se encapsula en IP
- Cabecera de 32 bits
 - Tipo (8 bits): tipo de mensaje
 - Código (8 bits): subtipo de mensaje
 - Comprobación (16 bits)

| | | |
|------|--------|--------------|
| 0 | 8 | 16 |
| tipo | código | comprobación |

➤ Mensajes ICMP:

| Campo tipo | Mensaje ICMP |
|------------|--|
| 8/0 | Solicitud/respuesta de eco (ping) |
| 3 | Destino inalcanzable |
| 4 | Ralentización del origen |
| 5 | Redirecciónamiento |
| 11 | Tiempo de vida excedido |
| 12 | Problema de parámetros |
| 13/14 | Solicitud/respuesta de sello de tiempo |
| 17/18 | Solicitud/respuesta de máscara de red |

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

55

55

Tema 2. Capa de red

5. El protocolo ICMP

➤ ICMP (Internet Control Message Protocol)

- Informa sobre situaciones de error → señalización
- Hacia el origen del datagrama IP.
- Se encapsula en IP
- Cabecera de 32 bits. Incluye la cabecera del datagrama que ha disparado el mensaje

| | | | | | | | | | | |
|--|----------|----------------|---------------|----------|--------------------------------|------------|-----------|-------|------|--|
| File | Edit | View | Go | Capture | Analyze | Statistics | Telephony | Tools | Help | |
| | | | | | | | | | | |
| Filter: icmp Expression... Clear Apply | | | | | | | | | | |
| No. | Time | Source | Destination | Protocol | Info | | | | | |
| 2 | 0.000719 | 150.214.20.130 | 150.214.191.5 | ICMP | Destination unreachable (Port) | | | | | |
| <pre> Frame 2: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) Ethernet II, Src: Cisco_b7:64:00 (00:07:0d:b7:64:00), Dst: Micro-st_a8:f7:63 (00:24:21:a8:f7:63) Internet Protocol Version 4, Src: 150.214.20.130 (150.214.20.130), Dst: 150.214.191.5 (150.214.191.5) Internet Control Message Protocol Type: 3 (Destination unreachable) Code: 3 (Port unreachable) Checksum: Oxfe7c [correct] Internet Protocol, Src: 150.214.191.5 (150.214.191.5), Dst: 150.214.20.130 (150.214.20.130) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) NetBIOS Name Service </pre> | | | | | | | | | | |

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2024 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

56

56



Esquema

1. Funcionalidades
2. Comutación
3. El protocolo IP
4. Asociación con la capa de enlace: el protocolo ARP
5. El protocolo ICMP
6. Autoconfiguración de la capa de red (**DHCP**)



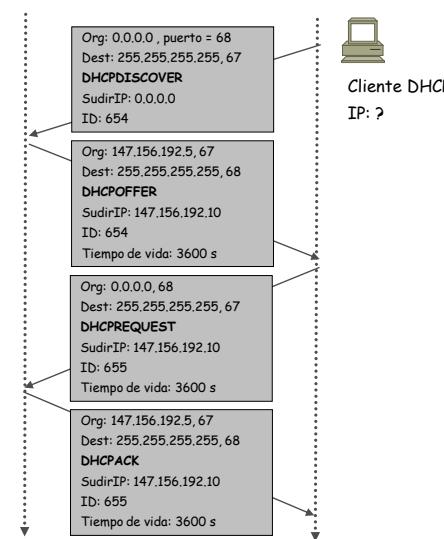
6. Autoconfiguración de la capa de red (DHCP)

DHCP (Dynamic Host Configuration Protocol)

Servidor DHCP
147.156.192.5

Para asignar las direcciones se usa
DHCP (RFC 2131-3396), protocolo
usuario de UDP (puerto 67)

- El host (cliente) envía un mensaje broadcast: "DHCP discover"
- El server DHCP responde con un mensaje "DHCP offer"
- El host solicita una dirección IP, mensaje "DHCP request"
- El server DHCP envía la dirección IP: mensaje "DHCP ack"



Tema 2. Capa de red

6. Autoconfiguración de la capa de red (DHCP)

DHCP

Configuración de un cliente MS Windows:

Wireless Network Connection Properties

Networking Sharing

Connect using: Intel(R) PRO/Wireless 2200BG Network Connection

This connection uses the following items:

- Client for Microsoft Networks
- VMware Bridge Protocol
- QoS Packet Scheduler
- File and Printer Sharing for Microsoft Networks
- Internet Protocol Version 6 (TCP/IPv6)
- Internet Protocol Version 4 (TCP/IPv4)
- Link-Layer Topology Mapper I/O Driver
- Link-Layer Topology Discovery Responder

Description: Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK Cancel

Internet Protocol Version 4 (TCP/IPv4) Properties

General Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically Use the following IP address:

IP address: Subnet mask: Default gateway:

Obtain DNS server address automatically Use the following DNS server addresses:

Preferred DNS server: Alternate DNS server:

OK Cancel Advanced...

Universidad de Granada

59

Tema 2. Capa de red

6. Autoconfiguración de la capa de red (DHCP)

DHCP

Configuración de un cliente Linux (Fedora Core distribution):

```
# Sample /etc/sysconfig/network-scripts/ifcfg-eth0:
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:0C:29:CE:63:E3
ONBOOT=yes
TYPE=Ethernet
```

Configuración de un servidor de Linux (dhcpd):

```
# Sample /etc/dhcpd.conf
default-lease-time 600;max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.100;
    range 192.168.1.150 192.168.1.200;
}

# Static IP address assignment
host haagen {
    hardware ethernet 08:00:2b:4c:59:23;
    fixed-address 192.168.1.222;
}
```

Universidad de Granada

60

1.3. Tema 3



Tema 3 ***Capa de transporte***

Fundamentos de Redes

Grado en Ingeniería Informática y dobles grados
Curso 3º

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones
E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2022



1

Tema 3. Capa de transporte en Internet



Esquema

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

1. Introducción.
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
 1. Multiplexación/demultiplexación.
 2. Control de conexión.
 3. Control de errores y de flujo.
 4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.

2





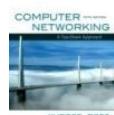
Objetivos del tema

- Comprender las funcionalidades y servicios de la capa de transporte :
 - Servicio de multiplexación/demultiplexación.
 - Servicio orientado a conexión frente a no orientado a conexión.
 - Cómo conseguir una transferencia de datos fiable.
 - Cómo proporcionar control de flujo.
 - Cómo proporcionar control de congestión.
 - Cómo se han implementado estas funcionalidades en Internet.

Bibliografía



Capítulo 10, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson, 2017, ISBN: 978-0-273-76896-8



Capítulo 3 James F. Kurose y Keith W. Ross. **COMPUTER NETWORKING. A TOP-DOWN APPROACH**, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675

- Agradecimientos:

Transparencias originales de **Juan Manuel López Soler, Pedro García Teodoro, Jorge Navarro Ortiz**, Departamento TSTC, UGR.





Esquema

1. **Introducción.**
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
 1. Multiplexación/demultiplexación.
 2. Control de conexión.
 3. Control de errores y de flujo.
 4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.



1. Introducción.

- Funciones y servicios de la capa de transporte:
 - Comunicación **extremo a extremo** (*end-to-end*).
 - Multiplexación/demultiplexación de aplicaciones → puerto.
- Protocolo UDP:
 - Multiplexación/demultiplexación de aplicaciones.
 - Servicio **no orientado a conexión, no fiable**.
- Protocolo TCP:
 - Multiplexación/demultiplexación de aplicaciones.
 - Servicio **orientado a conexión, fiable**:
 - Control de **errores y de flujo**.
 - Control de la **conexión**.
 - Control de **congestión**.
- Extensiones TCP



Tema 3. Capa de transporte en Internet

1. Introducción.

Comunicación **extremo a extremo (end-to-end)**:

host A

host B

Internet

APDU

TPDU

nodo intermedio

datagrama IP

red 1

red 2

aplicación

TCP/UDP

IP

Interfaz de red

aplicación

TCP/UDP

IP

Interfaz de red

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

7

Universidad de Granada

Tema 3. Capa de transporte en Internet

Esquema

1. Introducción.
2. **Protocolo de datagrama de usuario (UDP).**
3. Protocolo de control de transmisión (TCP).
 1. Multiplexación/demultiplexación.
 2. Control de conexión.
 3. Control de errores y de flujo.
 4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.

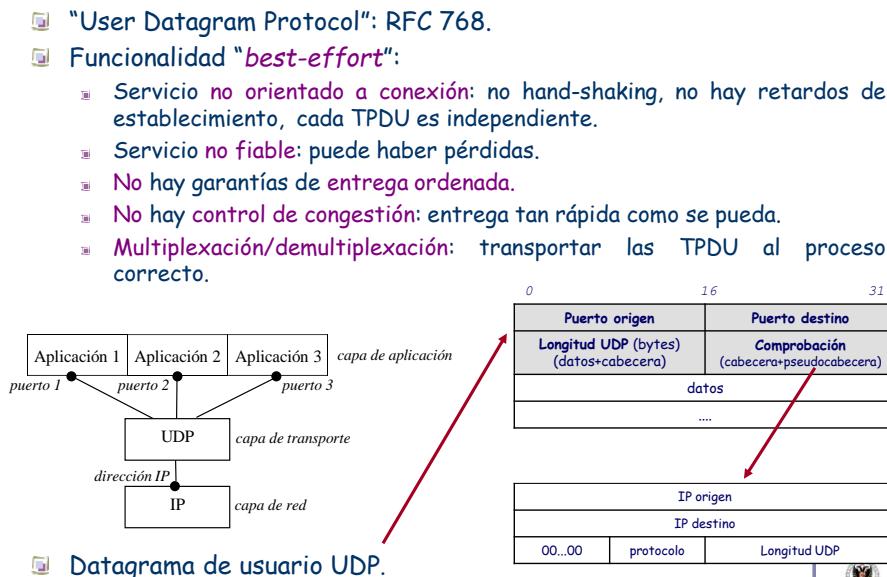
Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

8

Universidad de Granada



2. Protocolo de datagrama de usuario (UDP).



Datagrama de usuario UDP.

2. Protocolo de datagrama de usuario (UDP).

- Multiplexación/demultiplexación: transportar las TPDU al proceso correcto.

- Existen puertos preasignados con servicios normalizados:

Ejemplos de puertos UDP preasignados

| Puerto | Aplicación/Servicio | Descripción |
|--------|---------------------|----------------------------------|
| 7 | echo | Eco |
| 13 | daytime | Fecha |
| 37 | time | Hora |
| 42 | nameserver | Servicio de nombres |
| 53 | domain | Servicio de nombres de dominio |
| 69 | tftp | Transferencia simple de ficheros |
| 123 | ntp | Protocolo de tiempo de red |

- Otros puertos (>1024) están a libre disposición del desarrollador.

- UDP se usa frecuentemente para aplicaciones multimedia: tolerantes a fallos y sensibles a retardos.

- Cada segmento UDP se encapsula en un datagrama IP.





Esquema



3. Protocolo de control de transmisión (TCP).



Tema 3. Capa de transporte en Internet

3. Protocolo de control de transmisión (TCP).

Funcionalidades de TCP:

- Multiplexación/demultiplexación de aplicaciones.
- Control de la conexión (establecimiento y cierre).
- Control de errores y de flujo.
- Control de congestión.

TPDU TCP = Segmento TCP:

| | | | | | | |
|---|-----------------------|-----------|-----------------------------|--------------------------------|----|--|
| # de secuencia del primer byte del segmento | 0 | 4 | 10 | 16 | 31 | |
| | Puerto origen | | Puerto destino | | | |
| | Número de "secuencia" | | | | | |
| # del byte que se espera recibir (acumulativos) | | | Número de "acuse" de recibo | | | |
| Longitud de la cabecera TCP | Hlen (32 bits) | reservado | UAPRSF | "Ventana" del receptor (bytes) | | Cuenta de bytes (no segmentos) |
| | Comprobación | | | | | Control de flujo |
| | Opciones | | | | | Envío de datos urgentes fuera de banda |
| | datos | | | | | |
| | | | | | | |

13 Cada segmento TCP se encapsula en un datagrama IP.

Universidad de Granada

13

Tema 3. Capa de transporte en Internet

3.1. TCP. Multiplexación / demultiplexación.

Multiplexación/demultiplexación de aplicaciones:

- Transportar las TPDU al proceso correcto.
 - Existen puertos preasignados con servicios normalizados:

| Puerto | Aplicación/Servicio | Descripción |
|--------|---------------------|------------------------------------|
| 20 | FTP-DATA | Transferencia de ficheros: datos |
| 21 | FTP | Transferencia de ficheros: control |
| 22 | SSH | Terminal seguro |
| 23 | TELNET | Acceso remoto |
| 25 | SMTP | Correo electrónico |
| 53 | DNS | Servicio de nombres de dominio |
| 80 | HTTP | Acceso hipertexto (web) |
| 110 | POP3 | Descarga de correo |

- Otros puertos (>1024) están a libre disposición del desarrollador.
- La "conexión TCP" se identifica por: puerto e IP origen y puerto e IP destino.

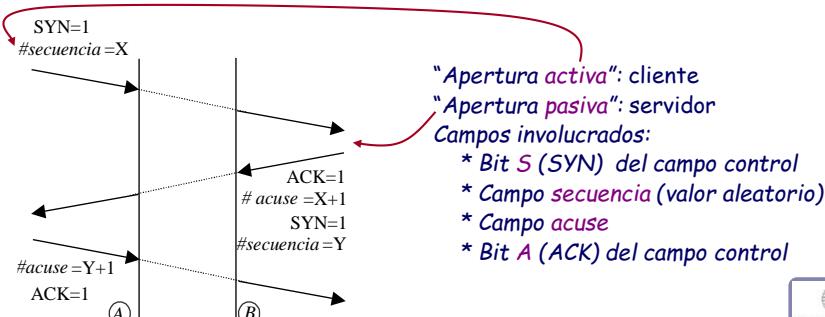
14 Universidad de Granada

14

Tema 3. Capa de transporte en Internet

300

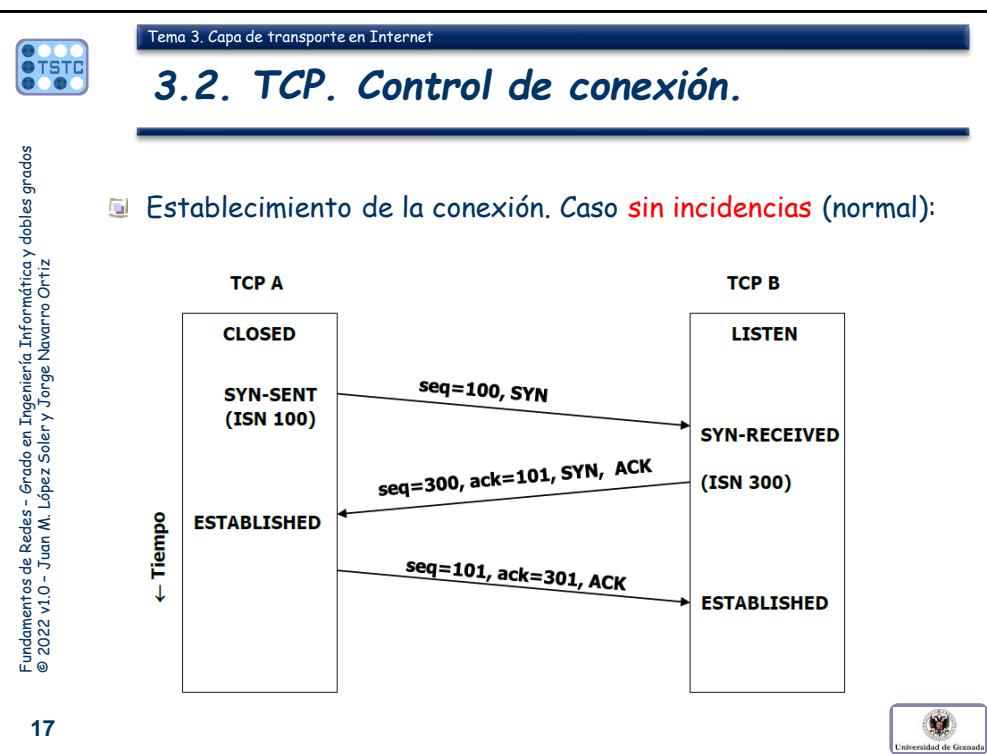
- TCP ofrece un servicio orientado a conexión.
 - El intercambio de información tiene tres fases:
 - Establecimiento de la conexión (sincronizar # de secuencia y reservar recursos).
 - Intercambio de datos (full-duplex).
 - Cierre de la conexión (liberar recursos).
 - ¿Es posible garantizar un establecimiento/cierre fiable de la conexión sobre un servicio (IP) no fiable? NO.
 - Establecimiento de la conexión: *three-way handshake*.



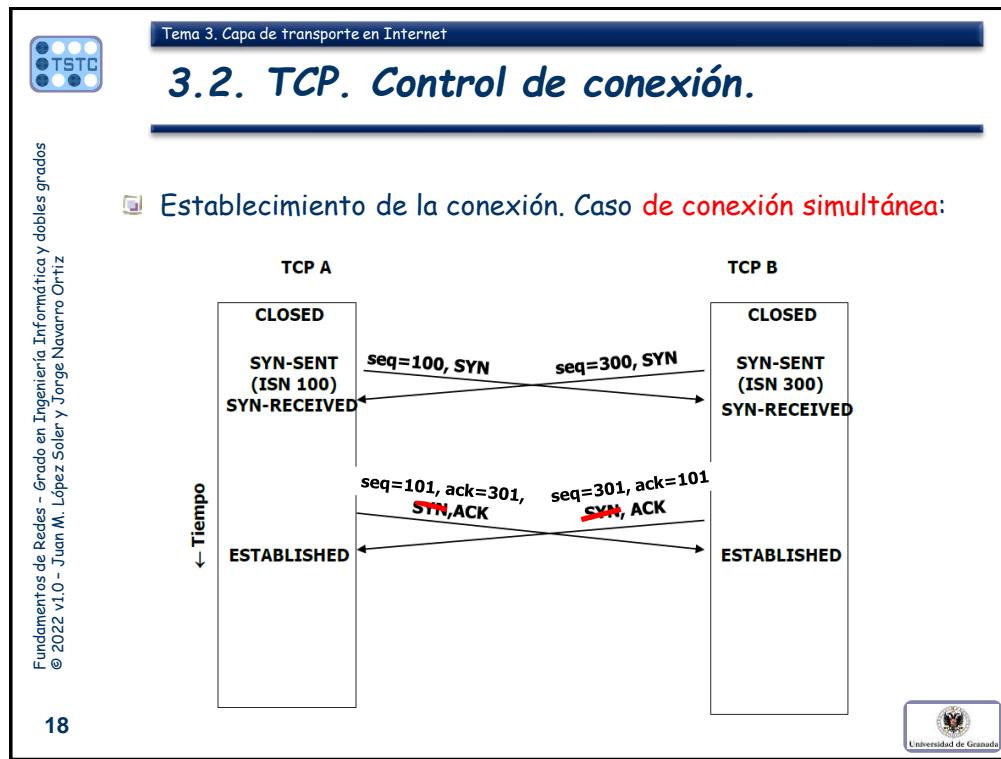
3.2. TCP. Control de conexión.

- Control de la conexión. Números de secuencia.
 - El **número de secuencia** es un campo de 32 bits que cuenta bytes en módulo 2^{32} (el contador se da la vuelta cuando llega al valor máximo).
 - El número de secuencia no empieza normalmente en 0, sino en un valor denominado **ISN** (Initial Sequence Number) elegido "teóricamente" al azar; para evitar confusiones con solicitudes anteriores.
 - El ISN es elegido por el sistema (cliente o servidor). El estándar sugiere utilizar un contador entero incrementado en 1 cada 4 μs aproximadamente. En este caso el contador se da la vuelta (y el ISN reaparece) al cabo de 4 horas 46 min.
 - El mecanismo de selección de los ISN es suficientemente fiable para proteger de coincidencias, pero no es un mecanismo de protección frente a sabotajes. Es muy fácil averiguar el **ISN** de una conexión e interceptarla suplantando a alguno de los dos participantes.
 - TCP **incrementa el número de secuencia** de cada segmento según los bytes que tenía el segmento anterior, con una sola excepción:
 - Los flags **SYN** y **FIN**, cuando están puestos, incrementan en 1 el número de secuencia.
 - La presencia del flag **ACK** no incrementa el número de secuencia.





17



18

Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

■ Establecimiento de la conexión. Caso con SYN retrasados y duplicados:

← Tiempo

19

Universidad de Granada

19

Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

■ Control de la conexión:

- Cierre de la conexión: liberación de recursos.

#secuencia = X

FIN=1

#acuse = Y+1

ACK=1

(A)

ACK=1

#acuse=X+1

FIN=1

#secuencia=Y

(B)

■ "Cierre activo"
■ "Cierre pasivo"
■ Campos involucrados:
* Bit F (FIN) del campo control
* Campo secuencia
* Campo acuse
* Bit A (ACK) del campo control

■ Para evitar bloqueos por pérdidas, una vez comenzado el procedimiento de CIERRE se usan timeouts (ver diagrama de estados: Maximum Segment LifeTime = 2 min).

20

Universidad de Granada

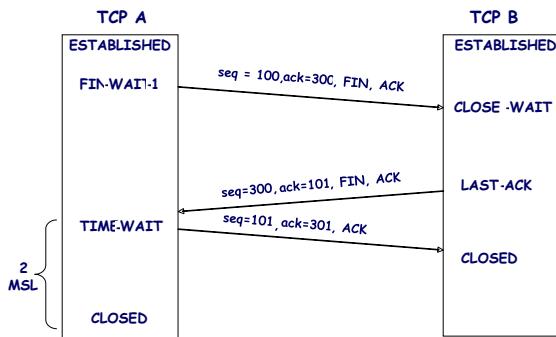
20

Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

 **Control de la conexión:**

-  **Cierre de la conexión: caso normal.**



MSL: Maximum Segment Lifetime (normalmente 2 minutos)

Hay otras posibilidades de cierre de la conexión (ver el diagrama de estados siguiente).



Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

21

21

Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

9. Se desea transferir con protocolo TCP un archivo de L bytes usando un MSS de 536.

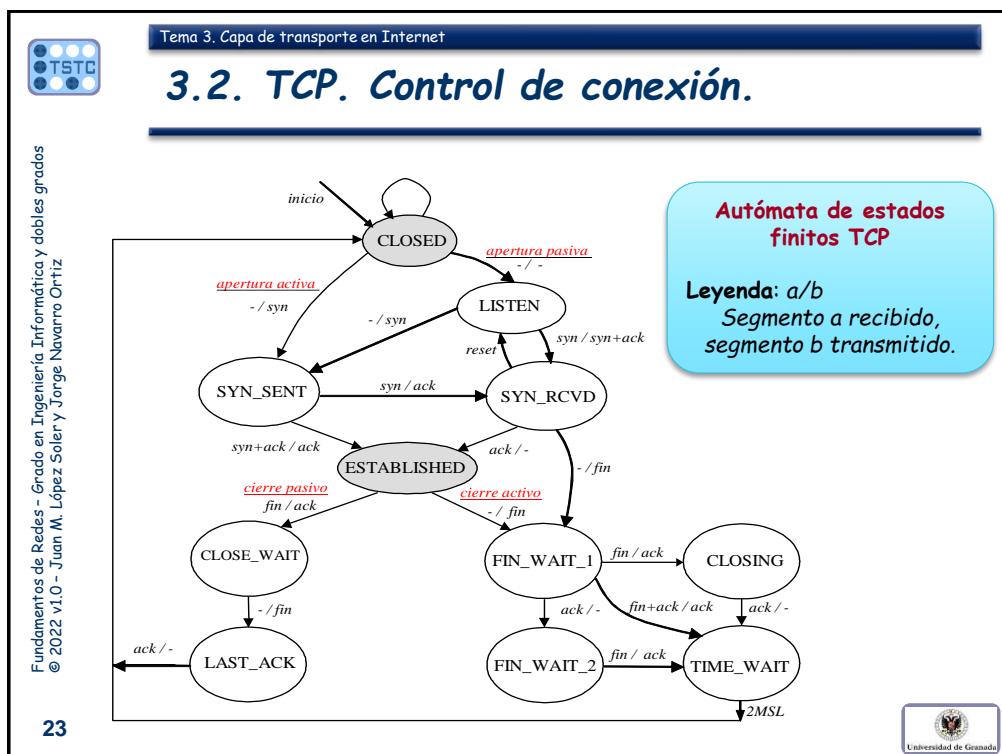
- a) **¿Cuál es el valor máximo de L tal que los números de secuencia de TCP no se agoten?**
- b) **Considerando una velocidad de transmisión de 155 Mbps y un total de 66 bytes para las cabeceras de las capas de transporte, red y enlace de datos, e ignorando limitaciones debidas al control de flujo y congestión, calcule el tiempo que se tarda en transmitir el archivo en A.**



Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

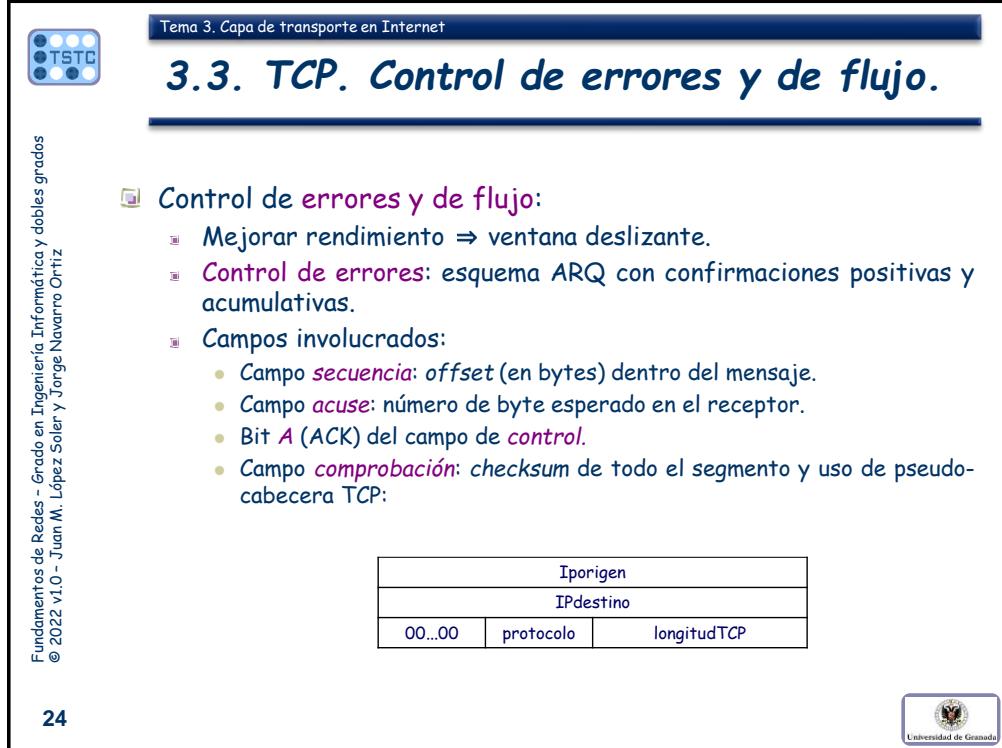
22

22



23

23



24

24

Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- Control de errores:** escenarios de retransmisión (gráficas © James F. Kurose).

Host A → Host B: Seq=92, 8 bytes data
Host B → Host A: ACK=100

Host A → Host B: Seq=92, 8 bytes data
Host B → Host A: ACK=100

timeout

pérdida X

Pérdida de ACK

tiempo

Host A → Host B: Seq=92, 8 bytes data
Host B → Host A: ACK=100

Host A → Host B: Seq=100, 20 bytes data
Host B → Host A: ACK=120

timeout=92 timeout

Seq=92, 8 bytes data

ACK=100

ACK=120

ACK=120

timeout prematuro y ACK acumulativo

25

25

Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- Control de errores:** generación de ACKs (RFC 1122, 2581).

| Evento | Acción del TCP receptor |
|--|---|
| Llegada ordenada de segmento, sin discontinuidad, todo lo anterior ya confirmado. | Retrasar ACK. Esperar recibir al siguiente segmento hasta 500 mseg. Si no llega, enviar ACK. |
| Llegada ordenada de segmento, sin discontinuidad, hay pendiente un ACK retrasado. | Inmediatamente enviar un único ACK acumulativo. |
| Llegada desordenada de segmento con # de sec. mayor que el esperado, discontinuidad detectada. | Enviar un ACK duplicado, indicando el # de sec. del siguiente byte esperado. |
| Llegada de un segmento que completa una discontinuidad parcial o totalmente. | Confirmar ACK inmediatamente si el segmento comienza en el extremo inferior de la discontinuidad. |

26

26



3.3. TCP. Control de errores y de flujo.

11. Los hosts A y B se están comunicando a través de una conexión TCP y B ya ha recibido y confirmado todos los bytes hasta el byte 126. Suponga que a continuación el host A envía dos segmentos seguidos a B que contienen, respectivamente, 70 y 50 bytes de datos. El envío de A es ordenado, el número de puerto origen en dichos segmentos es 302 y el de destino el 80. El host B envía una confirmación inmediata a la recepción de cada segmento de A, sin esperar el retardo de 500 ms del estándar.

- Especifique los números de secuencia de ambos segmentos.
- Si el primer segmento llega antes que el segundo ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que se envía?
- Si el segundo segmento llega antes que el primero ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que envía?
- Imagine que los segmentos llegan en orden pero se pierde el primer ACK.



3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- **Control de errores:** ¿Cómo estimar los "timeouts"?
 - Mayor que el tiempo de ida y vuelta (RTT).
 - Si es demasiado pequeño: timeouts prematuros.
 - Si es demasiado grande: reacción lenta a pérdida de segmentos.
 - Para situaciones cambiantes la mejor solución es la adaptable:

RTTmedido: tiempo desde la emisión de un segmento hasta la recepción del ACK.

$$RTT_{nuevo} = \alpha \cdot RTT_{viejo} + (1-\alpha) \cdot RTT_{medido}, \quad \alpha \in [0,1]$$

$$Desviacion_{nueva} = (1-x) * Desviacion_{vieja} + x * | RTT_{medido} - RTT_{nuevo} |$$

$$Timeout = RTT_{nuevo} + 4 * Desviacion$$

- Problema con ACKs repetidos: ambigüedad en la interpretación.
- Solución: **Algoritmo de Karn**, actualizar el RTT sólo para los no ambiguos, pero si hay que repetir un segmento incrementar el timeout:

$$tout_{nuevo} = \gamma \cdot tout_{viejo}, \quad \gamma = 2.$$





3.3. TCP. Control de errores y de flujo.

15. Si el RTT es 30 ms, la Desviación es 2 ms y se reciben ACKs tras 26, 32 y 24 ms, ¿Cuál será el nuevo RTT, Desviación y timeout? Usar $\alpha=0,125$ y $\beta=0,25$. ¿Y si los dos primeros ACKs tienen el mismo número de acuse y se usa el algoritmo de Karn?

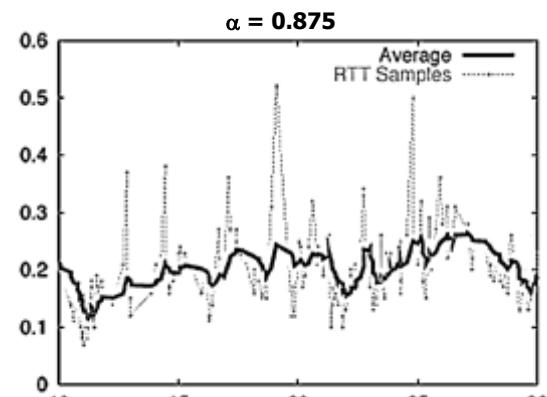


3.3. TCP. Control de errores y de flujo.

■ Control de errores y de flujo:

■ Control de errores: ¿cómo estimar los "timeouts"?

- Ejemplo de RTT medidos y estimados entre Amherst, Massachusetts y St. Louis, Missouri.



Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- Control de flujo:

- Procedimiento para evitar que el emisor **sature** al receptor con el envío de demasiada información y/o demasiado rápido.
- Es un **esquema crediticio**: el receptor informa al emisor sobre los bytes autorizados a emitir sin esperar respuesta.
- Se utiliza el campo **ventana**:

$$\text{ventana útil emisor} = \text{ventana ofertada receptor} - \text{bytes en tránsito}$$

31

31

Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- Control de flujo:

Emisor

Receptor

Buffer

0 4K

Vacio

2 KB

Lleno

La aplicación lee 2 KB

3 KB

La aplicación escribe 2 KB

La aplicación escribe 3 KB

Emisor Bloqueado

El emisor puede enviar hasta 2 KB

2 KB Seq = 0 Ack = 2048, Win = 2048

2 KB Seq = 2048 Ack = 4096, Win = 0

1 K Seq = 4096 Ack = 4096, Win = 2048

¿Alguna debilidad en el control de flujo?

¿Y si pierde el anuncio de WIN = 2048? → ¡Bloqueo! ¿Cómo evitarlo? → temporizador de persistencia

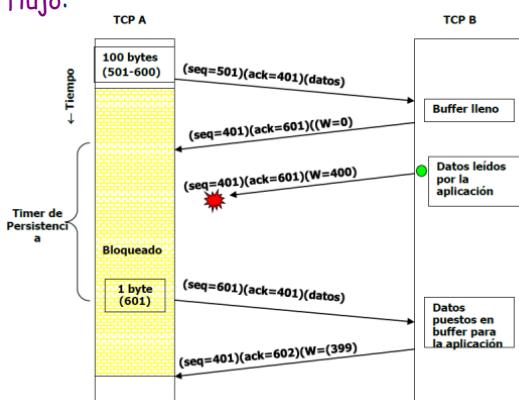
32

32

Tema 3. Capa de transporte en Internet

Control de errores y de flujo:

- ## ■ Control de flujo: Temporizador de persistencia



- Posible problema: síndrome de la ventana tonta (RFC 813) si se utilizan segmentos muy pequeños.
 - Posible mejora: la ventana optimista (RFC 813).
 - Es posible hacer entregas "no ordenadas": Bit **U** (URG), campo **puntero**.
 - Solicitar una entrega inmediata a la aplicación: bit **P** (PSH).



Tema 3. Capa de transporte en Internet

3.4. TCP. Control de congestión.

- ### Control de congestión (RFC 2001):

- Es un problema debido a la insuficiencia de recursos (ancho de banda de las líneas como buffer en routers y sistemas finales).
 - Es un problema diferente al control del flujo: involucra a la red y a los sistemas finales.
 - Tiene naturaleza adelante-atrás.
 - Se manifiesta en pérdidas y/o retrasos en las ACKs.
 - Solución: en la fuente limitar de forma adaptable el tráfico generado.



3.4. TCP. Control de congestión.

- ## Control de congestión:

En el emisor se utilizan dos ventanas y un umbral.

```
Bytes_permitidos_enviar =  
    min{VentanaCongestion, VentanaDelReceptor}
```

VentanaDelReceptor: utilizada para el control de flujo (de tamaño variable) según el campo "ventana" recibido.

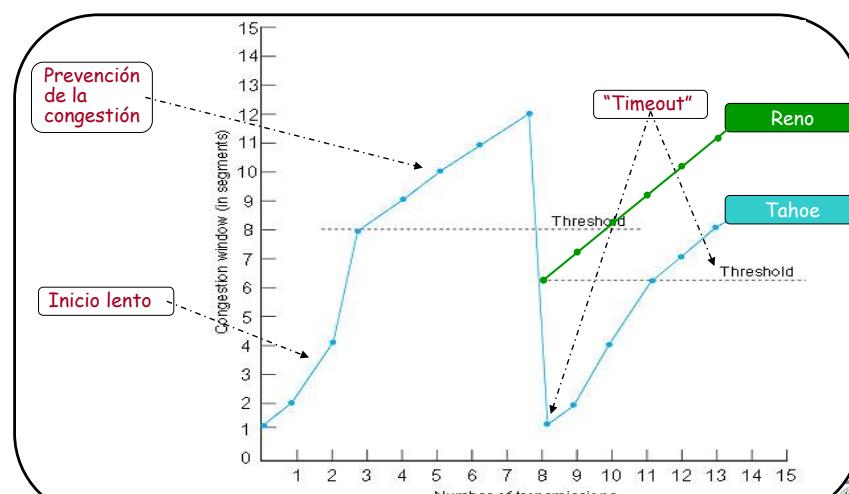
VentanaCongestion:

Inicialmente VentanaCongestion = 1 (num. segmentos)

Inicio lento Si $VentanaCongestion < umbral$, por cada ACK recibido $VentanaCongestion++$ (**crecimiento exponencial**)

Prevención de la congestión Si VentanaCongestion > umbral, cada vez que se recibe todos los ACKs pendientes VentanaCongestion++ (crecimiento lineal)

Si hay timeout entonces
umbral=VentanaCongestion/2 y VentanaCongestion = 1



Tema 3. Capa de transporte en Internet

Esquema

Universidad de Granada

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

1. Introducción.
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
 1. Multiplexación/demultiplexación.
 2. Control de conexión.
 3. Control de errores y de flujo.
 4. Control de congestión.
4. **Extensiones TCP.**
5. Ejercicios.

37

Tema 3. Capa de transporte en Internet

4. Extensiones TCP.

Universidad de Granada

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

- TCP se define con múltiples "Sabores"
- Los diferentes sabores no afectan a la interoperabilidad entre los extremos
- Desde cualquier versión de Linux con kernel mayor que la 2.6.19 se usa por defecto TCP CubIC

Fig. 50. Congestion window dynamics in CUBIC

38



Tema 3. Capa de transporte en Internet

4. Extensiones TCP.

- Adaptación de TCP a redes actuales (RFC 1323).
 - **Ventana escalada:**
 - Opción TCP en segmentos SYN:
 - Hasta $2^{14} \times 2^{16}$ bytes ($=2^{30}$ bytes=1GB) autorizados.
 - **Estimación RTT:**
 - Opción TCP de sello de tiempo, en todos los segmentos.
 - **PAWS ("Protect Against Wrapped Sequence numbers"):**
 - Sello de tiempo y rechazo de segmentos duplicados.
- Referencias:
 - RFCs
 - /usr/src/linux-2.../net/ipV4/tcp.c
 - /usr/include/netinet/tcp.h
 - <http://www.cs.columbia.edu/~hgs/internet/tcp.html>
 - Herramientas de análisis: ethereal/wireshark y tcpdump.

Universidad de Granada

39

39

Tema 3. Capa de transporte en Internet

Esquema

- 1. Introducción.
- 2. Protocolo de datagrama de usuario (UDP).
- 3. Protocolo de control de transmisión (TCP).
 - 1. Multiplexación/demultiplexación.
 - 2. Control de conexión.
 - 3. Control de errores y de flujo.
 - 4. Control de congestión.
- 4. Extensiones TCP.
- 5. Ejercicios.

Universidad de Granada

40

40



5. Ejercicios

10. Considere un enlace con una velocidad de transmisión de 1 Mbps, un tiempo de ida y vuelta (RTT) de 30 ms y segmentos fijos de 1500 bytes, incluyendo cabeceras y datos, ¿cuál tiene que ser el tamaño de la ventana para que la eficiencia en la transmisión (ratio entre el tiempo de transmisión y el tiempo total en el emisor) sea de al menos un 95%?



5. Ejercicios

17. Suponiendo que la ventana de congestión es 18 KB y que se dispara un timeout ¿Cuánto será la ventana de congestión si las 4 siguientes ráfagas de transmisiones, donde se envía la ventana completa, son exitosas? Suponed que el MSS es 1 KB.



1.4. Tema 4



Tema 4 **Seguridad en Redes**

Fundamentos de Redes

Doble Grado en Ingeniería Informática y Matemáticas

Doble Grado en Informática y ADE

Curso 2021 – 2022

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones

E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2022

1



Tema 2. Servicios y protocolos de aplicación en Internet



Esquema

1. Introducción
2. Cifrado
3. Autenticación
4. Funciones Hash
5. Firma digital y certificados digitales
6. Protocolos seguros

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

2





Objetivos del tema

- Comprender la importancia de la seguridad en las comunicaciones y aprender cómo desplegar mecanismos básicos de seguridad en redes de computadores e Internet.
- Conocer los aspectos de seguridad en redes: confidencialidad, autenticación, no repudio, integridad y disponibilidad
- Entender los conceptos básicos de la seguridad en redes, como el uso de algoritmos de clave secreta, de clave pública, intercambio de claves, etcétera
- Comprender qué son los certificados digitales y las autoridades de certificación, y los diferentes mecanismos que se pueden implementar con certificados
- Conocer algunos de los principales protocolos de comunicación seguros, como TLS e IPSec, y los mecanismos que utilizan



Bibliografía



Capítulo 8, James F. Kurose y Keith W. Ross.
COMPUTER NETWORKING. A TOP-DOWN APPROACH, 5^a Edición, Addison-Wesley, 2010,
ISBN: 9780136079675.

Capítulo 12, Pedro García Teodoro, Jesús Díaz Verdejo
y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson,
2^a Edición. 2014, ISBN: 9788490354612.





Esquema

1. **Introducción**
2. Cifrado
3. Autenticación
4. Funciones Hash
5. Firma digital y certificados digitales
6. Protocolos seguros



1. Introducción

- Una red de comunicaciones es **segura** cuando se garantizan **todos** los aspectos → no hay protocolos ni redes 100% seguros
- ¿Qué es seguridad? → múltiples aspectos:
 - **Confidencialidad/privacidad:** el contenido de la información es comprensible sólo por entidades autorizadas.
 - **Autenticación:** las entidades son quien dicen ser.
 - **Control de accesos:** los servicios están accesibles sólo a entidades autorizadas.
 - **No repudio o irrenunciabilidad:** el sistema impide la renuncia de la autoría de una determinada acción.
 - **Integridad:** el sistema detecta todas las alteraciones (intencionadas o no) de la información.
 - **Disponibilidad:** el sistema mantiene las prestaciones de los servicios con independencia de la demanda.





1. Introducción

- ¿En qué nivel/capa se debe situar la seguridad? en **TODOS.....el grado de seguridad lo fija el punto más débil**
- Ataque de seguridad: cualquier acción intencionada o no que menoscaba cualquiera de los aspectos de la seguridad
- Tipos de ataques:
 - **Sniffing** = vulneración a la confidencialidad, escuchas (husmear)
 - **Spoofing (phishing)** = suplantación de la identidad de entidades
 - **Man_in_the_middle** = hombre en medio (interceptación)
 - **Distributed Denial_of_Service (DDoS)** = denegación de servicio distribuido, ejemplo **Flooding** (inundación)
 - **Malware** = troyanos, gusanos, spyware, backdoors, rootkits, ransomware, keyloggers



1. Introducción

- Mecanismos de Seguridad
 - Cifrado (simétrico y asimétrico)
 - Autenticación con clave secreta (reto-respuesta)
 - Intercambio de Diffie-Hellman (establecimiento de clave secreta)
 - Funciones Hash. Hash Message Authentication Code (HMAC)
 - Firma Digital
 - Certificados digitales





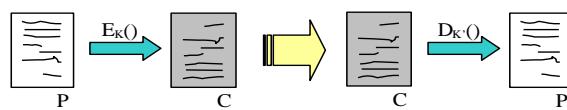
Esquema

1. Introducción
2. **Cifrado**
3. Autenticación
4. Funciones Hash
5. Firma digital y certificados digitales
6. Protocolos seguros



2. Cifrado

- **Cifrado de datos:**
 - Procedimiento para garantizar la **confidencialidad**
 - Texto llano/claro, $P \rightarrow$ texto cifrado, C
 - Se basa en la existencia de un **algoritmo** de cifrado/descifrado, **normalmente conocido** $E_K()$ y $D_{K'}()$. La dificultad reside en la existencia de un **clave de cifrado (K)**/**descifrado(K') desconocidas**



Tema 2. Servicios y protocolos de aplicación en Internet

2.1. Cifrado simétrico

- **Cifrado simétrico, algoritmos de clave secreta:**
 - Una sola clave para cifrar y descifrar ($k=k'$)
 - **DES** ("Data Encryption Standard", IBM 1975):

http://en.wikipedia.org/wiki/Feistel_network

http://en.wikipedia.org/wiki/Data_Encryption_Standard

(a)

(b)

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

11

Tema 2. Servicios y protocolos de aplicación en Internet

2.1. Cifrado simétrico

- **DES:** Es un esquema de sustitución monoalfabético
- Encadenamiento DES (para evitar que DES sea un algoritmo de sustitución):

(a)

(b)
- Mejorar la robustez: **DES doble y 3DES:**

cifrado K₁ K₂ K₁ → C C → K₁ descifrado

C → K₁ K₂ K₁ → P

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

12

Tema 2. Servicios y protocolos de aplicación en Internet

2.1. Cifrado simétrico

- **IDEA** ("International Data Encryption Algorithm"):
 - Simétrico: misma clave para cifrar y para descifrar
 - Claves de 128 bits
 - Opera en tiempo real (VLSI).

(a)

(b)

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

13

Tema 2. Servicios y protocolos de aplicación en Internet

2.2. Cifrado asimétrico

Cifrado asimétrico, algoritmos de clave pública/privada:

- Dos claves por usuario (*A*): una pública K_{PUB_A} y otra privada K_{PRI_A} distintas
- Conocida K_{PUB_A} es imposible conocer K_{PRI_A}
- Claves diferentes para cifrar y descifrar:

Cifrar $\rightarrow C = E_{K_{pubB}}(P)$ *Descifrar*: $P = D_{K_{priB}}(C)$

Y si enviamos $C = E_{K_{privA}}(P)$? \rightarrow autenticación

- **RSA** (Rivest, Shamir y Adleman):
 - Elegimos *p* y *q* primos grandes ($>10^{100}$)
 - $n = (p \cdot q)$ y $z = (p-1) \cdot (q-1)$ (función de Euler)
 - Elegimos *d* primo respecto de *z*
 - Calculamos *e* tal que $e \cdot d \bmod z = 1$ (algoritmo de Euclides)
 - $K_{pub} = (e, n)$ y $K_{priv} = (d, n)$, de modo que:
 - * $C = P^e \bmod n$
 - * $P = C^d \bmod n$

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

14



Tema 2. Servicios y protocolos de aplicación en Internet

2.2. Cifrado asimétrico

Ejemplo RSA:

- $p = 3, q = 11$
- $n = p \cdot q = 33, z = (n-1)(p-1) = 20 = 5 \times 2 \times 2$
- $d = 7$, primo respecto de z
- $e = 3, e \times d \bmod z = 1$
- $K_{pub} = (3, 33)$ y $K_{pri} = (7, 33)$

| Simbólico | Numérico | P^3 | $P^3 \bmod 33$ | C^7 | $C^7 \bmod 33$ | Simbólico |
|-----------|----------|-------|----------------|-------------|----------------|-----------|
| S | 19 | 6859 | 28 | 13492928512 | 19 | S |
| U | 21 | 9261 | 21 | 1801088541 | 21 | U |
| Z | 26 | 17576 | 20 | 1280000000 | 26 | Z |
| A | 01 | 1 | 1 | 78125 | 01 | A |
| N | 14 | 2744 | 5 | 78125 | 14 | N |
| N | 14 | 2744 | 5 | 78125 | 14 | N |
| E | 05 | 125 | 26 | 8031810176 | 05 | E |

$\underbrace{\hspace{10em}}_{P}$ $\underbrace{\hspace{10em}}_{C}$ $\underbrace{\hspace{10em}}_{P}$

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

15

Tema 2. Servicios y protocolos de aplicación en Internet

Esquema

1. Introducción
2. Cifrado
- 3. Autenticación**
4. Funciones Hash
5. Firma digital y certificados digitales
6. Protocolos seguros

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

16

Universidad de Granada

16

Tema 2. Servicios y protocolos de aplicación en Internet

3. Autenticación

- ❑ Autenticación y cifrado de clave secreta:
 - Esquema de reto-respuesta:

```

    graph LR
        A[A] -- "R_B" --> KABRB[K_AB(R_B)]
        KABRB --> B[B]
        B -- "R_A" --> KABA[K_AB(R_A)]
        KABA --> A
    
```

- ¿Ataque por reflexión?
- Usar espacios de claves disjuntos

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

17

Tema 2. Servicios y protocolos de aplicación en Internet

3. Autenticación

- Intercambio de Diffie-Hellman: permite establecer una clave secreta entre dos entidades a través de un canal no seguro.

A elige x, n, g

$$K_{AB} = (g^y \bmod n)^x \bmod n$$

B elige y

$$K_{BA} = (g^x \bmod n)^y \bmod n$$

- Ataque:
man-in-the-middle

The diagram shows entity A sending $n, g, g^x \bmod n$ to a middleman, who then sends $n, g, g^z \bmod n$ to B. Entity A receives $g^y \bmod n$ from the middleman, which is then sent back to the middleman as g^y . The middleman calculates g^z and sends $n, g, g^z \bmod n$ to B, while also sending g^y back to A.

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

18



Esquema

1. Introducción
2. Cifrado
3. Autenticación
- 4. Funciones Hash**
5. Firma digital y certificados digitales
6. Protocolos seguros



4. Funciones Hash

➤ **Funciones Hash (compendios).** Características de los compendios:

- Funciones unidireccionales (irreversibles) de cálculo sencillo
- Texto de entrada (M) de longitud variable
- $M \rightarrow H(M)$ siendo $H(M)$ de longitud fija (256 ó 512 bits)
- Imposible obtener M a partir de su resumen $H(M)$
- Invulnerables a ataques de colisión, dado M es imposible encontrar M' / $M \neq M'$ y $H(M) = H(M')$
- Ejemplos de funciones HASH: MD5, SHA-1, SHA-512
- Las funciones Hash se usan para garantizar integridad + autenticación Hash Message Authentication Code (HMAC): $M + H(K|M)$ pero para evitar ataques de extensión se usa $M + H(K | H(K | M))$



Tema 2. Servicios y protocolos de aplicación en Internet

4. Funciones Hash

MD5 ("Message Digest 5", RFC 1321):

- **Proceso (resumen de 128 bits):**
 - Relleno 100..0 de longitud máxima 448 bits
 - Adición de campo de longitud de 64 bits
 - División del mensaje en bloques de 512 bits
 - Procesamiento secuencial por bloques

Proceso MD5 general

Proceso para un bloque q

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

21

Tema 2. Servicios y protocolos de aplicación en Internet

4. Funciones Hash

SHA-1 ("Secure Hash Algorithm 1", NIST 1993):

- **Proceso (resumen de 160 bits):**
 - Relleno 100..0 de longitud máxima 448 bits
 - Adición de campo de longitud de 64 bits
 - División del mensaje en bloques de 512 bits
 - Procesamiento secuencial por bloques

Proceso SHA-1 general

Proceso para un bloque q

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

22



Esquema

1. Introducción
2. Cifrado
3. Autenticación
4. Funciones Hash
5. **Firma digital y certificados digitales**
6. Protocolos seguros

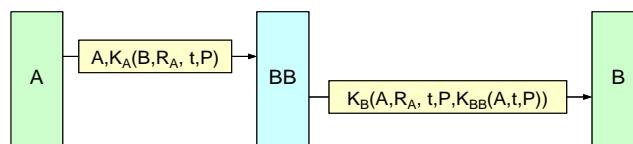


5. Firma digital y certificados digitales

➤ **Firma Digital: objetivos**

- El receptor pueda autenticar al emisor
- No haya repudio
- El emisor tenga garantías de no falsificación (integridad)

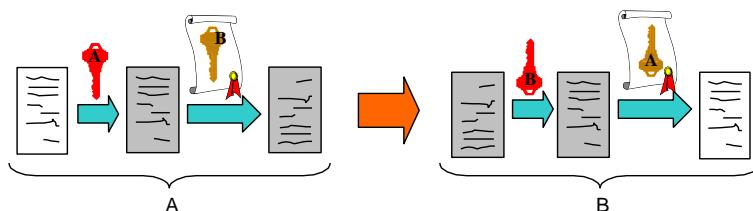
➤ **Firma digital con clave secreta: Big Brother:**



5. Firma digital y certificados digitales

➤ Firma digital con clave asimétrica. Doble cifrado:

- Uno para proporcionar privacidad, con KpubB
- Otro, previo, para autenticación, con KpriA
- Para firmar, enviar $KpubB(KpriA(T)) \rightarrow$
- En el receptor $KpubA(KpriB(KpubB(KpriA(T))))=T$



- Debilidad: para garantizar el no repudio se necesita garantizar la asociación fehaciente e indisoluble de la "identidad A" con su "clave pública KpubA" ($A \leftrightarrow KpubA$) ... ?
esto se consigue con un "certificado digital"



5. Firma digital y certificados digitales

□ Para garantizar la asociación "identidad-clave" certificados digitales

Autoridades de certificación (AC):

- **AC** = Entidad para garantizar la **asociación entre identidad y claves**:
 - El usuario obtiene sus claves pública y privada
 - Éste envía una solicitud, firmada digitalmente, a la AC indicando su identidad y su clave pública
 - AC comprueba la firma y emite el certificado solicitado:
 - * Identidad de AC, identidad del usuario, clave pública del usuario y otros datos como, por ejemplo, el período de validez del certificado
 - * Todo ello se firma digitalmente con la clave privada de AC con objeto de que el certificado no pueda falsificarse

- **Formato de certificados:** principalmente **X.509**

- **AC reconocidas:**

- ACE (www.ace.es)
- CAMERFIRMA (www.camerfirma.es)
- CERES (www.cert.fnmt.es)
- VeriSign (www.verisign.com)



Tema 2. Servicios y protocolos de aplicación en Internet

5. Firma digital y certificados digitales

 **Autoridades de certificación (AC):**

Campos de un certificado X.509

| Field | Explanation |
|-----------------|---|
| Version | Version number of X.509 |
| Serial number | The unique identifier used by the CA |
| Signature | The certificate signature |
| Issuer | The name of the CA defined by X.509 |
| Validity period | Start and end period that certificate is valid |
| Subject name | The entity whose public key is being certified |
| Public key | The subject public key and the algorithms that use it |


Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

27

Tema 2. Servicios y protocolos de aplicación en Internet

5. Firma digital y certificados digitales

 **Autoridades de certificación (AC):**

```

Certificate:
Data:
Version: 1 (0x0)
Serial Number: 7829 (0x1e95)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Capetown, O=Thawte Consulting cc,
        OU=Certification Services Division,
        CN=Thawte Server CA/Email=server-certs@thawte.com
Validity
    Not Before: Jul  9 16:04:02 1998 GMT
    Not After : Jul  9 16:04:02 1999 GMT
Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
        OU=Freesoft, CN=www.freesoft.org/Email=baccala@freesoft.org
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
        Modulus (1024 bit):
        00:b4:31:98:0a:c4:bc:f2:c1:88:a1:dc:b0:c8:bb:
        33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
        66:36:d0:8e:56:12:44:ba:75:eb:8e:81:c1:9c:5b:66:
        70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
        16:94:6e:ee:f4:d5:6f:df:ca:b3:i47:5e:1b:0c:7b:
        c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
        8f:a0:21:c7:4c:d0:16:65:00:cl:0f:d7:bf:8:80:e3:
        d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bcb:b8:
        e8:35:1c:9e:27:52:7e:41:8f
        Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
92:2e:4a:1b:8b:8c:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
0d:19:aa:ad:dd:9a:df:ab:97:50:65:fb:be:85:a6:ef:19:d1:
5a:de:9d:ea:63:cdb:cc:f6:d5:d0:01:85:b5:6d:c8:f3:d9:f7:
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
68:9f

```


Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

28



Resumen de mecanismos de seguridad

- Relación entre los mecanismos de seguridad y los servicios / aspectos de seguridad:
 - **Confidencialidad.** Se consigue mediante cifrado, bien sea simétrico (clave secreta) o asimétrico (clave pública)
 - **Autenticación.** Se consigue mediante mecanismos de autenticación como reto-respuesta y firma digital (big brother, doble cifrado).
 - **No repudio.** Se consigue mediante firma digital (big brother, doble cifrado).
 - **Integridad.** Se consigue añadiendo compendios o resúmenes a los mensajes, generados con funciones hash.
 - **Disponibilidad.** Los mecanismos vistos no proporcionan disponibilidad por sí solos. Serían necesarios sistemas antiataque, redundancia en las líneas de acceso, de los servidores, etcetera.



Esquema

1. Introducción
2. Cifrado
3. Autenticación
4. Funciones Hash
5. Firma digital y certificados digitales
6. **Protocolos seguros**



Tema 2. Servicios y protocolos de aplicación en Internet

6. Protocolos seguros

- Seguridad:
 - Seguridad Perimetral:
 - Firewalls, sistemas de detección de intrusiones (IDS) y de respuesta (IRIS)
 - Seguridad (criptográfica) en protocolos (¿dónde poner la seguridad?):
 - Capa de aplicación
 - Pretty Good Privacy (PGP)
 - Secure Shell (SSH)
 - Capa de sesión (entre aplicación y transporte)
 - Transport Secure Layer (TSL) (antes SSL) ➔ HTTPS, IMAPS, SSL-POP, VPN.
TLS = Handshake (negociar) + Record Protocol (operación).
TLS → Confidencialidad (K_{secreta} negociada) + Autenticación (para el server por defecto con $K_{\text{Pública}}$) + integridad (Con HMAC)
 - Capa de Red ➔ IPSec (VPN)

Universidad de Granada

31

Tema 2. Servicios y protocolos de aplicación en Internet

6. Protocolos seguros

- Pretty Good Privacy (PGP) - correo electrónico seguro

Emisor:

- $R = \text{MD5}(P)$
- $FD = K_{\text{pr}_A}(R)$
- $Z = \text{ZIP}(FD + P)$
- $C = \text{IDEA}_K(Z) + K_{\text{pu}_B}(K)$
- $M = \text{BASE64}(C)$

Receptor:

- $C = \text{B64}^{-1}(M)$
- $K = K_{\text{pr}_B}(K_{\text{pu}_B}(K))$
- $Z = \text{IDEA}_K^{-1}(C)$
- $FD + P = \text{ZIP}^{-1}(Z)$
- $R = K_{\text{pu}_A}(FD)$
- $R' = \text{MD5}(P)$
- $R' = R ??$

Universidad de Granada

32



6. Protocolos seguros

- Transport Secure Layer (TSL) (SSL) → HTTPS, IMAPS, SSL-POP, VPN.
- **SSL Record Protocol** encapsula los protocolos y ofrece un canal seguro con privacidad, autenticación e integridad
- **SSL Handshake Protocol**
 - Negocia el algoritmo de cifrado
 - Negocia la función Hash
 - Autentica al servidor con X.509
 - El cliente genera claves de sesión:
 - Aleatorias cifrada con K_{PUB_SERVER} ó Diffie-Hellman
- **SSL Alert protocol**
 - Informa sobre errores en la sesión
- **Change Cipher Spec Protocol**
 - Para notificar cambios en el cifrado

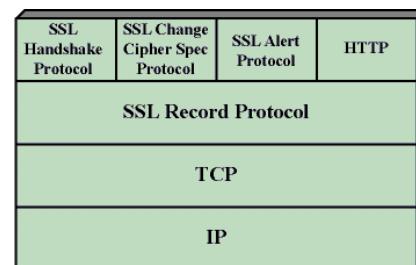
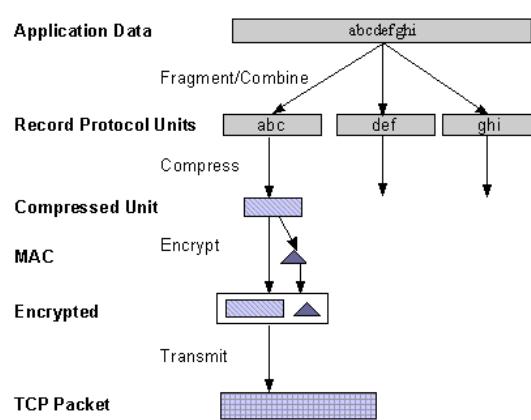


Figure 14.2 SSL Protocol Stack



6. Protocolos seguros

- Transport Secure Layer (TSL) (SSL) → HTTPS, IMAPS, SSL-POP, VPN.
- **SSL Record Protocol**





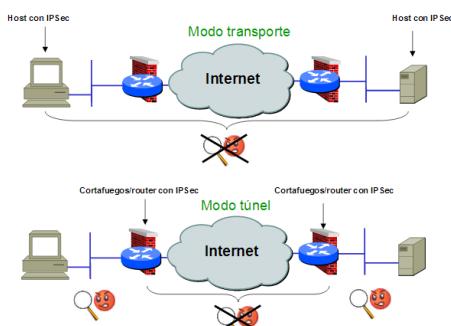
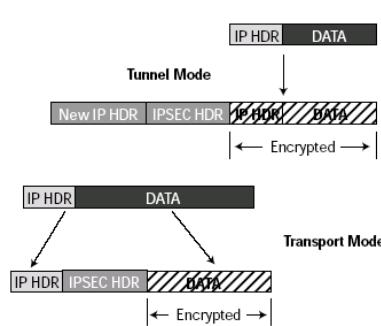
6. Protocolos seguros

- IPsec: su objetivo es garantizar autenticación, integridad y (opcionalmente) privacidad a nivel IP.
- IPsec son 3 procedimientos:
 - 1) Establecimiento de una "Asociación de seguridad": IKE =RFC 2409.
 - Objetivo: establecimiento de clave secreta (Diffie-Hellman)
 - Incluye previamente autenticación (con certificados) para evitar el ataque de persona en medio
 - Es *simplex*: la asociación de seguridad tiene un único sentido.
 - Se identifica con la IP origen + Security Parameter Index (32 bits)
 - Vulnera el carácter NO orientado a conexión de IP.
 - 2) Garantizar la autenticación e integridad de los datos:
protocolo de "Cabeceras de autenticación", RFC 2401
 - 3) (Opcional) Garantizar la autenticación e integridad y privacidad de los datos:
protocolo de "Encapsulado de seguridad de la carga", RFC 2411



6. Protocolos seguros

- IPsec tiene 2 modos de operación →
 - 1) **Modo Transporte**: la asociación se hace extremo a extremo entre en host origen y host destino
 - 2) **Modo túnel**: la asociación se hace entre dos routers intermediarios



1.5. Tema 5

Tema 5 *Capa de aplicación*

Fundamentos de Redes

Doble Grado en Ingeniería Informática y Matemáticas
Doble Grado en Informática y ADE
Curso 2022 – 2023

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones

E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2022



1

Tema 5. Capa de aplicación

Esquema

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación web
4. El correo electrónico
5. Aplicaciones multimedia
6. Cuestiones y ejercicios

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

2



2

Objetivos del tema

- Conocer las aplicaciones y servicios estándar en Internet, identificando los protocolos y servicios de usuario más relevantes a nivel de red, transporte y aplicación.
- Conocer el funcionamiento del modelo cliente/servidor.
- Desarrollar programas básicos de transmisión de datos

3



3

Bibliografía


Capítulo 2 (2.1, 2.2, 2.4, 2.5), James F. Kurose y Keith W. Ross. **COMPUTER NETWORKING. A TOP-DOWN APPROACH**, 5^a Edición, Addison-Wesley, 2010, ISBN: 9780136079675.

- Para saber más: capítulos 7 y 8



Capítulo 11, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson, 2^a Edición. 2014, ISBN: 9788490354612.

4



4

Tema 5. Capa de aplicación

Esquema

- 1. Introducción a las aplicaciones de red
- 2. Servicio de Nombres de Dominio (DNS)
- 3. La navegación web
- 4. El correo electrónico
- 5. Aplicaciones multimedia
- 6. Cuestiones y ejercicios

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

5

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

PROTOCOLOS TCP/IP ■

Estructura de protocolos

Diagrama jerárquico de los protocolos TCP/IP:

```

graph TD
    FTP[FTP] --> TELNET[TELNET]
    TELNET --> TCP[TCP]
    SMTP[SMTP] --> TCP
    SNMP[SNMP] --> UDP[UDP]
    TCP --> ICMP[ICMP]
    TCP --> EGP[EGP]
    UDP --> ARP[ARP]
    ARP --> Acceso[Acceso a la red]
    RARP[RARP] --> Acceso
    Acceso --> Internet[Internet]
    Internet --> Transporte[Transporte]
    Transporte --> Aplicacion[Aplicación]
  
```

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERACCIÓN CLIENTE/SERVIDOR

Arquitectura cliente-servidor

Servidor:

- Siempre en funcionamiento
- IP permanente & pública
- Agrupados en "granjas"
- <http://www.xatakandroid.com/mundo-android/la-imagen-de-la-semana-google-muestra-el-corazon-de-internet>
- <https://www.youtube.com/watch?v=zRwPSFpLX8I>

Clientes:

- Funcionando intermitentemente
- Pueden tener IP dinámica & privada
- Se comunican con el servidor
- No se comunican entre sí

Universidad de Granada

7

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

Proceso Cliente : proceso que inicia la comunicación

Proceso Servidor: proceso que espera a ser contactado
→ IP permanente & pública

➤ Proceso envía/recibe mensajes a/desde su **socket**

➤ Para recibir mensajes un proceso debe tener un **identificador** (IP + puerto)

Ej: servidor web **gaia.cs.umass.edu**:
Dirección IP: 128.119.245.12
Número de puerto: 80

8

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

La mayoría de las transacciones entre aplicaciones se basan en el paradigma cliente-servidor:

- Servidor:** programa que ofrece un servicio accesible a través de la red.
- Cliente:** programa que envía peticiones y espera respuestas del servidor a través de la red.

Diferencias:

- El servidor comienza antes (apertura pasiva).
- El servidor se ejecuta de forma permanente.
- El servidor usa puertos reservados.

Universidad de Granada

9

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

El software TCP/IP es parte del S.O.

Las aplicaciones lo usan a través de una **API** consistente en llamadas al sistema:
"la interfaz socket"

La definición de la **interfaz socket** no es parte de ningún protocolo.

Distintas implementaciones:

- Berkeley Socket Distribution,
- Winsock,
- Transport Layer Interface, etc.

Universidad de Granada

10

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

La interfaz socket (BSD):

- Punto de partida: la gestión de I/O del S.O. se basa en:
abrir-leer/escribir-cerrar
- Implementado mediante llamadas al sistema:
open, read, write, close
- Concepto importante: el **descriptor de fichero**.
- La interfaz socket extiende la I/O a conexiones en red.
- Se necesita añadir la identificación de los puntos finales:
IP local, IP remota, puerto local y puerto remoto
- Definimos **SOCKET** como un **descriptor** de una transmisión a través del cual la aplicación puede enviar y/o recibir información hacia y/o desde otro proceso de aplicación.
- Es una "**puerta**" de acceso entre la **aplicación** y los servicios de **transporte**.

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

11

11

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

- Definimos **SOCKET** como un **descriptor** de una transmisión a través del cual la aplicación puede enviar y/o recibir información hacia y/o desde otro proceso de aplicación.
- Es una "**puerta**" de acceso (metafóricamente) entre la **aplicación** y los servicios de **transporte**.
- En la práctica un **socket** es una variable tipo **puntero** a una estructura:

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

12

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

- Especificar la dirección IP y puerto LOCAL:

```
int bind (int socket, struct sockaddr *myaddr, int addresslen)
```

- Para pasar direcciones se define la estructura:

```
struct sockaddr_in {           /* INET socket addr info */
    short sin_family;          /* familia: AF_INET */
    u_short sin_port;          /* puerto: 16 bits, nbo */
    struct in_addr sin_addr;   /* dir IP de 32 bits */
    char sin_zero[8];          /* no usada*/
};

struct in_addr {
    u_long s_addr;             /* dir IP de 32 bits, nbo */
};
```

...es necesario hacer un casting :

| | | |
|----------------------------|--------------------------|----|
| 0 | 16 | 31 |
| familia | dirección ₀₋₁ | |
| dirección ₂₋₅ | | |
| dirección ₆₋₉ | | |
| dirección ₁₀₋₁₃ | | |

| | | |
|--------------|--------|----|
| 0 | 16 | 31 |
| AF_INET(2) | puerto | |
| dirección IP | | |
| 00..00 | | |
| 00..00 | | |

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

13

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

- Especificar la dirección IP y puerto REMOTOS:

```
int connect (int socket,
             struct sockaddr *toaddr,
             int addresslen)
```

- Si el socket es SOCK_STREAM, **connect** envía un SYN (TCP hand-shaking) a través de la red. Exige simultaneidad de los dos procesos.
- Si el socket es SOCK_DGRAM, **connect** NO envía nada a través de la red. Sólo especifica las dirección IP y puerto remoto. Esto facilita su utilización posterior.

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

14

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

■ Para poner un socket en modo pasivo en el servidor y definir el número de solicitudes de conexión pendientes que se encolarán:

```
int listen (int sockfd, int maxwaiting)
```

no es una llamada "bloqueante".

■ Para detener el flujo del programa y esperar hasta que llegue una solicitud de conexión en el servidor:

```
int accept (int sockfd,
            struct sockaddr *fromaddrptr,
            int *addresslen)
```

devuelve un socket nuevo conectado para comunicarse con los clientes.

15

Universidad de Granada

15

16

16

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

INTERFAZ SOCKET

Otras llamadas...

- read() / readv() / recmsg()
- write() / writev() / sendmsg()
- gethostbyname()
- getservbyname()
- getprotobynumber()
- htons() / htonl()
- ntohs() / ntohl()
- Ficheros cabecera típicos:
 - netinet/in.h
 - sys/types.h
 - sys/socket.h

17

Universidad de Granada

17

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

TIPOS DE SERVIDORES

Tipos de servidores:

Criterios clasificación:

- Orientados a Conexión - No Orientados a Conexión
- Iterativos - Concurrentes

Servidor iterativo no orientado a conexión:

(a) Diagrama de un servidor iterativo:

(b) Diagrama de flujo para un servidor iterativo:

```

graph TD
    cliente[cliente] -- "socket" --> sock1[sock]
    sock1 -- "bind(sock)" --> servidor[servidor]
    servidor -- "recvfrom" --> cliente
    cliente -- "sendto" --> servidor
    servidor -- "sendto" --> cliente
    cliente -- "close" --> cliente
    servidor -- "close" --> servidor
  
```

Universidad de Granada

18

18

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

TIPOS DE SERVIDORES

■ Servidor iterativo orientado a conexión:

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

(a)

(b)

19

19



19

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

TIPOS DE SERVIDORES

■ Servidores concurrentes orientados a conexión:

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

conurrencia real

conurrencia aparente

I/O asíncrona: select()

20

20

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

TIPOS DE SERVIDORES

Servidores multiprotocolo con I/O asíncrona (select):

- Un solo proceso iterativo ofrece varios protocolos.

socket de solicitudes UDP
socket de solicitudes TCP
socket de conexión y servicio TCP

```

servidor
sUDP = socket; sTCP = socket;
bind(sUDP); bind (sTCP);
listen(sTCP);
select();
sTCP: accept(sTCP...), recv(sTCP...), send(sTCP...), close(s2)
sUDP: recvfrom(sUDP...), sendto(sUDP...);

Universidad de Granada

```

21

21

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

TIPOS DE SERVIDORES

Servidores multiservicio:

- Ofrecen varios servicios con concurrencia.

(a)

sockets de servicio

(b)

sockets de solicitud socket para una conexión individual

(c)

sockets de solicitud sockets para conexiones individuales

(d)

sockets de solicitud sockets para conexiones individuales

22

22

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

CLIENTE TCP - SOCKETS BSD

```
/*
 * File: cliente.c
 */

#include <stdio.h>
#include <malloc.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/time.h>

#include <stdlib.h>
#include <stdio.h>

/*
 * Cliente TCP
 */
int
main(int argc, char** argv) {
    int socket_datos;
    struct sockaddr_in sockname;
    char buffer[82];

    // Obteniendo parámetros de la línea de comandos...
    if (argc<3)
        perror("Sintaxis: cliente IP_servidor puerto_servidor");
    exit(1);
}
```

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

SOCKETS - EJEMPLOS

```
char *servidor = argv[1]; // Dirección IP del servidor
int port = atoi(argv[2]); // Puerto de escucha del servidor

// Creando el socket de datos...
if((socket_datos=socket(AF_INET,SOCK_STREAM,0))==-1)
    perror("Cliente: error en la llamada a la función socket");
exit(1);

// Asignando puerto y dirección...
sockname.sin_family=AF_INET;
sockname.sin_addr.s_addr=inet_addr(servidor);
sockname.sin_port=htons(port);

// Conectándose con el servidor
if(connect(socket_datos,(struct sockaddr *)&sockname,
            sizeof(sockname))==-1)
    perror("Cliente: error en la llamada a la función connect");
exit(1);

// Bucle para enviar mensajes hasta introducir "FIN"...
do{
    printf("Teclee el mensaje a transmitir: \n");
    gets(buffer);

    printf ("Has tecleado: %s\n", buffer);

    // Mandando datos a través del socket...
    if(send(socket_datos,buffer,80,0)==-1)
        perror("Cliente: error en la llamada a la función send");
}while(strcmp(buffer,"FIN")!=0);

// Cerrando el socket de datos...
close(socket_datos);

exit (0);
}
```

Universidad de Granada

23

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

SERVIDOR ITERATIVO TCP - SOCKETS BSD

```
/*
 * File: servidor.c
 */

#include <stdio.h>
#include <malloc.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/time.h>

#include <stdlib.h>

/*
 * Servidor TCP iterativo
 */
int
main(int argc, char** argv) {
    int socket_control, from_len, socket_datos;
    struct sockaddr_in from,sockname;
    char buffer[82];
    int salir=0;

    // Obteniendo parámetros de la línea de comandos...
    if (argc<2)
        perror("Sintaxis: servidor puerto_servidor");
    exit(1);

    int port = atoi(argv[1]);

    // Creando el socket de control (para aceptar conexiones)...
    if((socket_control=socket(AF_INET,SOCK_STREAM,0))==-1)
        perror("Servidor: error en la llamada a la función socket");
    exit(1);

    // Asignando dirección y puerto...
    sockname.sin_family=AF_INET;
    sockname.sin_addr.s_addr=INADDR_ANY;
    sockname.sin_port=htons(port);
}
```

SOCKETS - EJEMPLOS

```
// Anunciándose como servidor.
if(bind(socket_control,(struct sockaddr *)&sockname,sizeof(sockname))==-1)
    perror("Servidor: error en la llamada a la función bind");
exit(1);

// Diciendo que será un socket de escucha...
if(listen(socket_control,1)==-1)
    perror("Servidor: error en la llamada a la función listen");
exit(1);

// Bucle infinito para aceptar peticiones...
do{

    // Aceptando conexiones de los diferentes clientes
    // (creándose un nuevo socket de datos)...
    from_len=sizeof(from);
    socket_datos=accept(socket_control,(struct sockaddr *)&from,&from_len);
    if(socket_datos== -1)
        perror("Servidor: error en la llamada a la función accept");
    exit(1);

    // Recibiendo un mensaje y escribiéndolo en pantalla (hasta recibir FIN)...
    do{
        int nbytes = recv(socket_datos,buffer,80,0);
        if(nbytes== -1)
            perror("Servidor:Recv");
        if(nbytes==0)
            perror("El cliente se ha desconectado");
        printf("El mensaje recibido fue: \n%s\n",buffer);
    }while(strcmp(buffer,"FIN")!=0);

    // Cerrando el socket de datos...
    close(socket_datos);
}while(!salir);

// Cerrando el socket de control...
close(socket_control);

// Terminando el programa...
exit(0);
}
```

24



Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

SERVIDOR CONCURRENTE TCP - SOCKETS BSD

```
/*
 * File: servidorconcurrente.c
 */

#include <stdio.h>
#include <malloc.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/time.h>

#include <stdlib.h>

/*
 * Servidor TCP concurrente
 */
int
main(int argc, char** argv) {
    int socket_control, from_len, socket_datos;
    struct sockaddr_in from,sockname;
    char buffer[82];
    int salir=0;

    // Obteniendo parámetros de la línea de
    // comandos...
    if (argc<2)
        perror("Sintaxis: servidor
puerto_servidor"),exit(1);

    int port = atoi(argv[1]);
    // Creando el socket de control (para aceptar
    // conexiones)...
    socket_control=socket(AF_INET,SOCK_STREAM,0);
    if((socket_control<=1))
        perror("Servidor: error en la llamada a la
función socket"),exit(1);
}
```

SOCKETS - EJEMPLOS

```
// Asignando dirección y puerto...
sockname.sin_family=AF_INET;
sockname.sin_addr.s_addr=INADDR_ANY;
sockname.sin_port=htons(port);

// Anunciándose como servidor...
if(bind(socket_control,(struct sockaddr*)&sockname,sizeof(sockname))<=1)
    perror("Servidor: error en la llamada a la
función bind"),exit(1);

// Diciendo que será un socket de escucha...
if(listen(socket_control,1)<=1)
    perror("Servidor: error en la llamada a la
función listen"),exit(1);

// Bucle infinito para aceptar peticiones...
do{
    // Aceptando conexiones de los diferentes
    // clientes (creándose un nuevo socket de
    // datos)...
    from_len=sizeof(from);
    if((socket_datos=accept(socket_control,
    (struct sockaddr*)&from,&from_len))<=1)
        perror("Servidor: error en la llamada a la
función accept"),exit(1);

    // Creación de un proceso hijo para atender al
    // cliente que se conectó...
    int pid; // Identificador del proceso padre
    if ((pid=fork())==0)
        // Proceso hijo =
        // PROCESO DE ATENCIÓN AL CLIENTE
    {
        // Cerrando el socket de control...
        close(socket_control);
        // Finalización del proceso de atención al
        // cliente...
        exit(0);
    }
    else {
        // Proceso padre =
        // PROCESO QUE ESPERA PETICIONES
        // No hace nada, el bucle hará que vuelva a
        // esperar una petición (accept()).
    }
}while(salir);

// Cerrando el socket de control...
close(socket_control);

// Terminando el programa...
exit(0);
}
```

25

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

CLIENTE TCP - JAVA

© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

```
/*
 * <p>Title: Minimo cliente TCP</p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2007</p>
 * <p>Company: UGR</p>
 * @author not attributable
 * @version 1.0
 */

import java.net.*;
import java.io.*;

public class MinimoClienteTCP {

    // Atributos de la clase:
    static Socket socket_datos;
    static String direccionServidor; // Nombre o dirección IP
    static int puerto;
    static PrintWriter out;
    static BufferedReader in;

    public MinimoClienteTCP() {
    }

    public static void main (String args[])
    boolean error=false;
    String mensajeSolicitud;
    String mensajeRespuesta = "";
    
```

SOCKETS - EJEMPLOS

```
// Se piden 3 argumentos: dirección del servidor, puerto y mensaje a enviar.
if (args.length>3) {
    System.err.println("Sintaxis: MinimoClienteTCP <direccion-servidor>
<puerto> <mensaje a enviar>");
    System.exit(-1);
}

// Dirección (IP o nombre) del servidor
direccionServidor = args[0];
// Puerto
puerto = Integer.parseInt(args[1]);
// Mensaje a enviar
mensajeSolicitud = args[2];

// 1 - Se abre el socket y se conecta a la dirección y puerto del servidor.
try {
    socket_datos = new Socket (direccionServidor, puerto);

    // Se obtienen los flujos de lectura y escritura para recibir y enviar
    // mensajes.
    out = new PrintWriter (socket_datos.getOutputStream(), true);
    in = new BufferedReader (new InputStreamReader(socket_datos.getInputStream()));

    } catch (UnknownHostException e) {
        System.err.println ("Error: no se pudo encontrar al servidor " +
direccionServidor);
        System.exit(-2);
    } catch (IOException e) {
        System.err.println("Error: no se pudo establecer la conexión con el
servidor");
    }

    // (Continua en la siguiente transparencia ...)
```

26

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

SOCKETS - EJEMPLOS

CLIENTE TCP - JAVA (continuación)

```
// (... continuación de MinimoClienteTCP)

// 2 - Se escribe el mensaje de solicitud y leemos la respuesta:
out.println(mensajeSolicitud);
try {
    mensajeRespuesta = in.readLine();
} catch (IOException e) {
    System.err.println("Error: no se pudo leer la respuesta.");
}

// 3 - Se cierra la conexión.
try {
    in.close();
    out.close();
    socket_datos.close();
} catch (IOException e) {
    System.err.println("Error: no se pudo cerrar la conexión.");
}

// Se muestra la respuesta:
System.out.println("El mensaje enviado fue: " + mensajeSolicitud);
System.out.println("El mensaje recibido fue: " + mensajeRespuesta);
}
```

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

27

27

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

SOCKETS - EJEMPLOS

SERVIDOR ITERATIVO TCP - JAVA

```
/** * <p>Title: Mínimo servidor iterativo TCP</p> */
import java.net.*;
import java.io.*;

public class MinimoServidorTCP {

    // Atributos de la clase:
    static ServerSocket socket_control;
    static Socket socket_datos;
    static int puerto;
    static PrintWriter out;
    static BufferedReader in;

    public MinimoServidorTCP() {
    }

    public static void main (String args[]) {
        boolean salir = false;
        boolean error = false;
        String mensajeSolicitud;
        String mensajeRespuesta;

        // El argumento es el puerto donde se
        // atenderá el servicio.
        if (args.length<1){
            System.err.println("Sintaxis:
                MinimoServidorTCP <puerto>");
            System.exit(-1);
        }

        // Se obtiene el puerto:
        puerto = Integer.parseInt(args[0]);
    }

    // 1 - Se abre el socket en modo "escucha"
    try {
        socket_control = new ServerSocket (puerto);
    } catch (IOException e) {
        System.err.println("Error: no se puede abrir el puerto
            indicado.");
        System.exit(-2);
    }

    // Es un servidor iterativo: acepta una conexión, la
    // procesa y la cierra. Después se acepta otra conexión
    // y así sucesivamente.
    do {
        // 2 - Se bloquea la hebra actual en "accept", y se
        // devuelve la conexión establecida con el cliente.
        try {
            socket_datos = socket_control.accept();
        }

        // Se obtienen los flujos de entrada y salida para
        // recibir y enviar mensajes.
        try {
            out = new PrintWriter
                (socket_datos.getOutputStream(), true);
            in = new BufferedReader (new InputStreamReader
                (socket_datos.getInputStream()));
        } catch (IOException e) {
            System.err.println("Error: no se pudo obtener un
                canal para los flujos");
            error = true;
        }

        // 3 - Código del servicio ofrecido.
        // 3a) Se lee una línea del cliente
        mensajeSolicitud = in.readLine();

        // 3b) Se aplica el servicio:
        procesaServicio(mensajeSolicitud);

        // 3c) Se envía la respuesta:
        out.println(mensajeRespuesta);

        // catch (IOException e){
        System.err.println("Error: no se pudo aceptar la
            solicitud de una conexión");
        error = true;
        }

        } while (!salir);
    }

    static String procesaServicio(String mensaje){
        // Aquí se podría poner el código que procesara el
        // mensaje recibido. Actualmente sólo lo devuelve tal
        // cual vino.

        return mensaje;
    }
}
```

28

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

SERVIDOR CONCURRENTE TCP - JAVA

```
/*
 * <p>Title: Mínimo servidor concurrente TCP</p>
 * <p>Description:</p>
 * <p>Copyright: Copyright (c) 2007</p>
 * <p>Company: UGR</p>
 * <p>Author: not attributable</p>
 * <version 1.0>
 */

import java.net.*;
import java.io.*;

public class MinimoServidorConcurrenteTCP {
    // Atributos de la clase
    static ServerSocket socket_control;
    static Socket socket_datos;
    static int puerto;
    static Servicio servicio;
    // Hebras para que el servicio sea concurrente

    public MinimoServidorConcurrenteTCP() {
    }

    static public void main (String args[]){
        boolean salir = false;
        boolean error = false;

        // Al menos un argumento, el puerto.
        if (args.length<1){
            System.err.println ("Sintaxis:  
MinimoServidorConcurrenteTCP <punto>");
            System.exit(-1);
        }

        // Se obtiene el puerto
        puerto = Integer.parseInt(args[0]);

        // 1 - Se abre el socket en modo "escucha".
        try{
            socket_control = new ServerSocket (puerto);
        } catch (IOException e){
            System.err.println ("Error: no se puede abrir el  
puerto indicado.");
            System.exit(-2);
        }
    }
}
```

// Bucle para aceptar conexiones. Por cada conexión
// aceptada se creará una hebra a la que se le pasará el
// socket para cursar el servicio.
do {
 // 2 - Se bloquea la hebra actual en "accept"
 // esperando una solicitud de conexión. Se devuelve
 // un nuevo socket con la conexión establecida.
 try {
 socket_datos = socket_control.accept();
 }
 // Se lanza una hebra para que sirva a este cliente
 // por "socket_datos".
 new Servicio(socket_datos).start();
} catch (IOException e) {
 System.err.println("Error: no se pudo aceptar la
solicitud de una conexión.");
 error = true;
}
} while (!salir);
}

class Servicio extends Thread {
 // Atributos de la clase
 Socket socket_datos;
 PrintWriter out;
 BufferedReader in;
 // El constructor recibirá como argumentos el socket
 // (abierto) que debe utilizar (se lo pasa la hebra
 // principal del servidor).
 public Servicio(Socket socket_datos_) {
 socket_datos = socket_datos_;
 }

 // Se obtienen los flujos de lectura y de escritura para
 // enviar y recibir mensajes.
 try {
 out = new PrintWriter
 (socket_datos.getOutputStream(), true);
 in = new BufferedReader(new InputStreamReader
 (socket_datos.getInputStream()));
 } catch (IOException e) {
 System.out.println("Error: no se pudo establecer la conexión.");
 return;
 }
}

public void run() {
 String mensajeSolicitud = "";
 String mensajeRespuesta = "";
 try {
 // 3 - Código del servicio ofrecido.
 // 3a - Se lee el mensaje del cliente.
 mensajeSolicitud = in.readLine();
 } catch (IOException e){
 System.err.println(this.getName() + " Error: no
se pudo leer el mensaje.");
 }
 // 3b - Se aplica el servicio.
 mensajeRespuesta = procesaServicio
 (mensajeSolicitud);
 // 3c - Se envía la respuesta.
 out.println(mensajeRespuesta);
 // 4 - Se cierra la conexión establecida con
 // el cliente.
 try {
 in.close();
 out.close();
 socket_datos.close();
 } catch (IOException e){
 System.err.println(this.getName() + " Error: no
se pudo cerrar la conexión.");
 }
}

static String procesaServicio (String mensaje){
 // Aquí se podría poner el código que procesara
 // el mensaje recibido.
 // Actualmente sólo lo devuelve tal cual vino.
 return mensaje;
}

SOCKETS - EJEMPLOS

29

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

RETARDO EN COLA

- Para estimar los retardos (tiempos) en cola se usa la teoría de colas:
- El uso de un servidor se modela con un sistema M/M/1 (ver **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**)

- El retardo en cola es:

$$R = \frac{\lambda \cdot (T_s)^2}{1 - \lambda \cdot T_s}$$

- donde T_s (distribución exponencial) es el tiempo de servicio y λ (Poisson) la ratio de llegada de solicitudes.
- Esta misma expresión se puede utilizar para calcular el retardo en cola en un router.

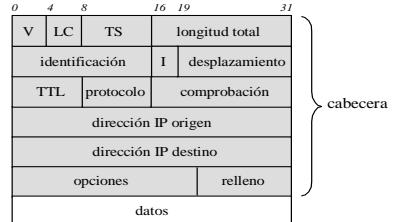
30

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

¿QUÉ DEFINEN LOS PROTOCOLOS DE APLICACIÓN?

- ¿Qué es y qué define un protocolo de aplicación?
 - El tipo de servicio
 - - Orientado o no orientado a conexión
 - - Realimentado o no
 - El tipo de mensaje
 - ej., request, response,
 - La sintaxis:
 - Definición y estructura de "campos" en el mensaje
 - En aplicación generalmente son orientados a texto (HTTP)
 - Aunque hay excepciones (DNS)
 - Tendencia : usar formato Type-Length-Value
 - La semántica:
 - Significado de los "campos"
 - Las reglas:
 - Cuándo los procesos envian mensajes/responden a mensajes
- Tipos de protocolos:
 - Protocolos de dominio público (Definidos en RFCs (ej., HTTP, SMTP)) versus propietarios → (ej., Skype, IGRP)
 - Protocolos in-band versus out-of-band
 - Protocolos stateless versus state-full
 - Protocolos persistentes versus no-persistentes (sobre servicios SOC)



Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

¿QUÉ DEFINEN LOS PROTOCOLOS DE APLICACIÓN?

- Tendencia: hacer los protocolos flexibles con
 - Una cabecera fija
 - Una serie de "trozos" (obligatorios y opcionales)

| | | | |
|---------------|---------------|------------------|----|
| 0 | 8 | 16 | 31 |
| Campo Fijo 1 | | Campo Fijo 2 | |
| Campo 3 | | | |
| Campo 4 | | | |
| Trozo 1 Tipo | Trozo 1 Flags | Trozo 1 Longitud | |
| Trozo 1 Datos | | | |
| | | | |
| Trozo N Tipo | Trozo N Flags | Trozo N Longitud | |
| Trozo N Datos | | | |

Cabecera

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

32

© 2022, v1.0

16

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

¿QUÉ DEFINEN LOS PROTOCOLOS DE APLICACIÓN?

- **Tendencia:** hacer los protocolos flexibles con
 - Una cabecera fija
 - Una serie de "trozos" (obligatorios y opcionales)
 - Los trozos pueden incluir una cabecera específica más una serie de datos en forma de parámetros:
 - Parámetros fijos: en orden
 - Parámetros de longitud variable u opcionales.
 - Para los parámetros se usa Formato TLV (Type-Length-Variable)

| | | | |
|---------------------|------------------------|----|----|
| 0 | 8 | 16 | 31 |
| Tipo de parámetro | Longitud del parámetro | | |
| Valor del parámetro | | | |

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

33

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

APLICACIONES DE RED: CARACTERÍSTICAS

- **Características/requisitos de las aplicaciones:**
 - **Tolerancia a pérdidas de datos (errores):** Algunas apps (e.g., audio) pueden tolerar algunas pérdidas de datos; otras (e.g. FTP, telnet, HTTP) requieren transferencia 100% fiable
 - **Exigencia de requisitos temporales:** Algunas apps denominadas *inelásticas* (e.g., telefonía Internet, juegos interactivos) requieren retardo (*delay*) acotado para ser efectivas, otras aplicaciones no
 - **Demandas de ancho de banda (tasa de transmisión o throughput):** Algunas apps requieren envío de datos a una tasa determinada (p. ejemplo un *codec* de vídeo), otras no
 - **Nivel de seguridad:** Los requisitos de seguridad para las distintas apps son muy variables (Encriptación, autenticación, no repudio, integridad...)
 - **Conclusión:** las distintas aplicaciones tienen requisitos HETEROGÉNEOS

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

34

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

REQUERIMIENTOS DE ALGUNAS APLICACIONES

| Application | Data loss | Throughput | Time Sensitive |
|-----------------------|---------------|--|----------------|
| file transfer | no loss | elastic | no |
| e-mail | no loss | elastic | no |
| Web documents | no loss | elastic | no |
| real-time audio/video | loss-tolerant | audio: 5kbps-1Mbps video:10kbps-5Mbps | yes, 100's ms |
| stored audio/video | loss-tolerant | same as above | yes, few s |
| interactive games | loss-tolerant | few kbps up | yes, 100's ms |
| instant messaging | no loss | elastic | yes and no |

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

35

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

PROTOCOLOS DE TRANSPORTE

The diagram illustrates the transport layer protocol stack and data delivery across a network. It shows two hosts, host A and host B, connected through an Internet backbone consisting of two clouds labeled 'red 1' and 'red 2'. Host A's stack includes Application, TCP/UDP, IP, and Interfaz de red layers. Host B's stack includes Application, TCP/UDP, IP, and Interfaz de red layers. Application-layer data (APDU) is exchanged between hosts. These APDUs are converted into Transport Protocol Data Units (TPDU) at the TCP/UDP layer. The TPDUs are then segmented into IP datagrams at the IP layer. These datagrams are transmitted through the network. At each node (host or router), the IP datagrams are handled by the IP and Interfaz de red layers. Finally, the datagrams reach host B, where they are reassembled into TPDUs and then delivered to the application layer.

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

36

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

PROTOCOLOS DE TRANSPORTE

Servicio TCP:

- Orientado a conexión
- Transporte fiable con control de errores
- Control de flujo
- Control de congestión

Servicio UDP:

- No orientado a conexión
- Transporte no fiable
- Sin control de flujo
- Sin control de congestión,
- ¿Para qué existe UDP?

- TCP y UDP (capa de transporte) al ser usuarios del protocolo IP (capa de red) **no garantizan Calidad de Servicio (QoS)**, es decir:
 - El **retardo** NO está acotado
 - Las **fluctuaciones en el retardo** NO están acotadas
 - No hay una **velocidad de transmisión** mínima garantizada
 - No hay una **probabilidad de pérdidas** acotada
- Tampoco hay garantías de seguridad.

Universidad de Granada

37

Tema 5. Capa de aplicación

1. Introducción a las aplicaciones de red

PROTOCOLOS DE TRANSPORTE

| Application | Application layer protocol | Underlying transport protocol |
|------------------------|--|-------------------------------|
| e-mail | SMTP [RFC 2821] | TCP |
| remote terminal access | Telnet [RFC 854] | TCP |
| Web | HTTP [RFC 2616] | TCP |
| file transfer | FTP [RFC 959] | TCP |
| streaming multimedia | HTTP (eg Youtube), RTP [RFC 1889] | TCP or UDP |
| Internet telephony | SIP, RTP, proprietary (e.g., Skype) | typically UDP |

Universidad de Granada

38

Tema 5. Capa de aplicación

Esquema

1. Introducción a las aplicaciones de red
2. **Servicio de Nombres de Dominio (DNS)**
3. La navegación web
4. El correo electrónico
5. Aplicaciones multimedia
6. Cuestiones y ejercicios

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

39

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

- La comunicación en Internet precisa de direcciones IP
- Los usuarios prefieren usar "nombres de dominio" (más de 300 x 10^6)
- DNS: traducción de nombres a direcciones IP (resolución de nombres)
dns.ugr.es <-----> **150.214.204.10**
- Estructura jerárquica en dominios:
Parte_local.dominio_niveln.dominio_nivel2.dominio_nivel1.
- Al dominio de nivel 1 se le denomina **dominio genérico** (.com .es .edu etc).
- El dominio raíz o "." está gestionado por el **ICANN** (Internet Corporation for Assigned Names and Numbers;
<http://www.icann.org>). ICANN delega la gestión de algunos dominios genéricos a centros regionales.

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

40

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

■ Lectura recomendadas

Tutorial sobre "los nombres de dominios":

- <https://www.icann.org/en/system/files/files/domain-names-beginners-guide-06dec10-es.pdf>

Instrucciones para registrar un nombre de dominio en .es:

- <https://www.dominios.es/es/registra-un-dominio/como-registrar-domino>

Instalación y ejemplos de ficheros configuración de named

- <https://www.tldp.org/HOWTO/DNS-HOWTO.html>

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

41

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

■ Inicialmente fueron definidos los siguientes 9 dominios genéricos (RFC 1591):

- .com -> organizaciones comerciales
- .edu -> instituciones educativas, como universidades, de EEUU.
- .gov -> instituciones gubernamentales estadounidenses
- .mil -> grupos militares de estados unidos
- .net -> proveedores de Internet
- .org -> organizaciones diversas diferentes de las anteriores
- .arpa -> propósitos exclusivos de infraestructura de Internet
- .int -> organizaciones establecidas por tratados internacionales entre gobiernos
- .xy -> indicativos de la zona geográfica (ej. es (España); pt (portugal); jp (Japón)...

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

42

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

- DNS es un **protocolo** de aplicación para el acceso a una base de datos distribuida con una gestión distribuida.
- 3 niveles de servidores:
 - Servidores raíz “.”
 - Servidores de dominio (Top-Level domain o TLD)
 - Servidores Locales
- `jcp.ugr.es` ➔ `www.google.com`
 - Consulta al “resolver” local
 - Conexión con DNS local con IP conocida: ¿cómo se conoce?
 - El DNS local realiza la “resolución” (ver página siguiente)
- Resolución iterativa o recursiva
- Para mejorar prestaciones se usan caches

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

43

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

44

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

- Gestión de la base de datos distribuida y jerárquica:
 - Está formada por un conjunto de servidores cooperativos que almacenan parcialmente la base de datos que se denomina BIND (Berkeley Internet Name Domain).
 - Cada servidor es responsable de lo que se denomina **ZONA**.
 - Una **zona** es un conjunto de nombres de dominio contiguos (por debajo de un nodo en el árbol) de los que un servidor tiene toda la información y es su **autoridad**.
 - Los **servidores autoridad (Start of Authority Servers)** deben contener **toda** (no "cacheada") la información de su zona.
 - La autoridad puede **delegarse** jerárquicamente a otros servidores

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

45

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

- Gestión de la base de datos DNS:
 - Cada zona debe tener **al menos** un servidor de autoridad.
 - En cada zona hay servidores **primarios** (almacenan una copia master de la db en discos locales) y servidores **secundarios** (obtienen la db por transferencia)
 - Además, existe un servicio de **cache** para mejorar prestaciones.
 - La **topología real** de servidores es complicada: existen **13 servidores raíz (A-M)** (ver <http://www.root-servers.org>)
 - El root-server F (y otros) tiene un servidor en Madrid (**Espanix: punto neutro**)
 - Cuando un cliente (a través de un *resolver local*) solicita una resolución de nombres a su servidor, puede ocurrir:
 - **Respuesta CON autoridad:** el servidor tiene autoridad sobre la zona en la que se encuentra el nombre solicitado y devuelve la dirección IP.
 - **Respuesta SIN autoridad:** el servidor no tiene autoridad sobre la zona en la que se encuentra el nombre solicitado, pero lo tiene en la cache.
 - **No conoce la respuesta:** el servidor preguntará a otros servidores de forma recursiva o iterativa. Normalmente se "eleva" la petición a uno de los servidores raíz.

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

46

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)



 Root-servers <http://www.root-servers.org/>

Servidor A: Network Solutions, Herndon, Virginia, USA.
 Servidor B: Instituto de Ciencias de la Información de la Universidad del Sur de California, USA.
 Servidor C: PSINet, Virginia, USA.
 Servidor D: Universidad de Maryland, USA.
 Servidor E: NASA, en Mountain View, California, USA.
 Servidor F: Internet Software Consortium, Palo Alto, California, USA.
 Servidor G: Agencia de Sistemas de Información de Defensa, California, USA.
 Servidor H: Laboratorio de Investigación del Ejército, Maryland, USA.
 Servidor I: NORDUnet, Estocolmo, Suecia.
 Servidor J: (TBD), Virginia, USA.
 Servidor K: RIPE-NCC, Londres, Inglaterra.
 Servidor L: (TBD), California, USA.
 Servidor M: Wide Project, Universidad de Tokio, Japón.





Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

47

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)



¿Cómo es la base de datos DNS?

- Todo dominio está asociado al menos a un registro **Resource Record**.
- Cada **RR** es una tupla con 5 campos:

Nombre del dominio: nombre del dominio al que se refiere el RR.
Tiempo de vida: tiempo de validez de un registro (para la cache).
Clase: en Internet siempre IN.
Tipo: Tipo de registro.

| | |
|-------|---|
| SOA | Registro (S tart O f A uthority) con la autoridad de la zona. |
| NS | Registro que contiene un servidor de nombres. |
| A | Registro que define una dirección IPv4. |
| MX | Registro que define un servidor de correo electrónico. |
| CNAME | Registro que define el nombre canónico de un nombre de dominio. |
| HINFO | Información del tipo de máquina y sistema operativo. |
| TXT | Información del dominio. |

Valor: Contenido que depende del campo tipo

- Existe una base de datos asociada de **resolución inversa** para traducir direcciones IP en nombres de dominio. (in-addr.arpa)



Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

48

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

Ejemplo de fichero de zona para el dominio **example.com**

```

; IPv4 zone file for example.com
$TTL 2d ; default TTL for zone
$ORIGIN example.com. ; base domain-name
; Start of Authority record defining the key characteristics
; of the zone (domain)
@ IN SOA ns1.example.com. hostmaster.example.com. (
    2003080800 ; se = serial number
    12h ; ref = refresh
    15m ; ret = refresh retry
    3w ; ex = expiry
    2h ; nx = nxdomainttl
)
; name servers Resource Records for the domain
    IN NS ns1.example.com.
; the second name server is
; external to this zone (domain).
    IN NS ns2.example.net.
; mail server Resource Records for the zone (domain)
; value 10 denotes it is the most preferred
    3w IN MX 10 mail.example.com.
; the second mail server has lower preference (20) and is
; external to the zone (domain)
    IN MX 20 mail.example.net.
; domain hosts includes NS and MX records defined previously
; plus any others required
ns1 IN A 192.168.254.2
mail IN A 192.168.254.4
joe IN A 192.168.254.6
www IN A 192.168.254.7
; aliases ftp (ftp server) to an external location
ftp IN CNAME ftp.example.net.

```

Universidad de Granada

49

Tema 5. Capa de aplicación

2. Servicio de Nombres de Dominio (DNS)

Formato mensajes DNS:

| | | |
|----------------|--------------|----|
| 0 | 16 | 31 |
| identificación | parametro | |
| nsolicitudes | nrespuestas | |
| nautoridades | nadicionales | |
| solicitud ... | | |
| srespuesta ... | | |
| sautoridad ... | | |
| sadicional ... | | |

- Consulta/respuesta.
- Reuesta con/sin autoridad.
- Consulta con/sin recursión deseada.
- Disponible/no disponible respuestas con recursión.
- Consulta directa/inversa.

| | | |
|----------------|----------|----|
| 0 | 16 | 31 |
| solicitud | | |
| tipo | clase | |
| recurso | | |
| tipo | clase | |
| tiempo | longitud | |
| datos | | |

DNS se ofrece en el puerto 53 mediante UDP normalmente o TCP (para respuestas grandes > 512 bytes).

Más información:

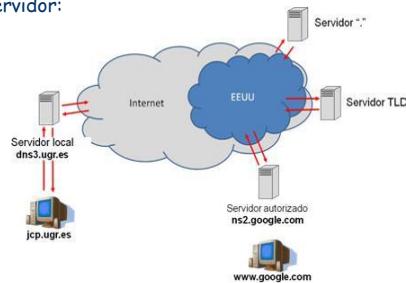
- RFC 1034 y RFC 1035 (actualizados 3597 y 3658)
- /usr/doc/HOWTO/trans/es/DNS-COMO
- man named, nslookup, resolver, host.conf, dig
- DNSSEC http://www.dominios.es/dominios/sites/dominios/files/1318333648229_0.pdf

Universidad de Granada

50

2. Servicio de Nombres de Dominio (DNS)

5. En la siguiente figura se ilustra un ejemplo de acceso DNS por parte de una máquina (`jcp.ugr.es`) que quiere acceder a los servicios de `www.google.com`. Para obtener la dirección IP del servidor, es necesario que la consulta pase por todos los servidores del gráfico. Considerando unos retardos promedio de $8 \mu s$ dentro de una red LAN, de 12 ms en cada acceso a través de Internet (4 ms si la conexión se restringe a EEUU) y de 1 ms de procesamiento en cada servidor:



Calcule el tiempo que se tardaría si la solicitud al servidor local es recursiva, pero el propio servidor local realiza solicitudes iterativas.

Especifique una política (recursiva-iterativa) más rápida de solicitudes y el tiempo que tardaría la solicitud en ser respondida. ¿Qué desventaja tiene sobre la solución anterior?



Esquema

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. **La navegación Web**
4. El Correo electrónico
5. Seguridad y protocolos seguros
6. Aplicaciones multimedia
7. Cuestiones y ejercicios



Tema 5. Capa de aplicación

3. La navegación web

➤ Una página Web es un fichero (HTML) formado por objetos:
 □ ficheros HTML, imágenes JPEG, Java applets, ficheros de audio, vídeo, etc

➤ Cada objeto se direcciona por una URL (o URI):

esquema://[/user[:password]@]dominio[:puerto]/[path]/[recurso][?solicitud][#fragment]

| Name | Used for | Example |
|--------|------------------|---|
| http | Hypertext (HTML) | http://www.cs.vu.nl/~ast/ |
| ftp | FTP | ftp://ftp.cs.vu.nl/pub/minix/README |
| file | Local file | file:///usr/suzanne/prog.c |
| news | Newsgroup | news:comp.os.minix |
| news | News article | news:AA0134223112@cs.utah.edu |
| gopher | Gopher | gopher://gopher.tc.umn.edu/11/Libraries |
| mailto | Sending e-mail | mailto:JohnUser@acm.org |
| telnet | Remote login | telnet://www.w3.org:80 |

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

53

Tema 5. Capa de aplicación

3. La navegación web

➤ Las páginas se sirven con el protocolo HTTP: Hyper Text Transfer Protocol
 □ Modelo cliente-servidor
 □ **cliente**: browser que solicita, recibe y muestra objetos web
 □ **servidor**: envía objetos web en respuesta a peticiones

➤ Las páginas web pueden ser **estáticas** (contenido invariable) o **dinámicas** (con contenido variable).

➤ Las páginas dinámicas pueden proporcionar contenido variable:
 □ Usando lenguajes de **scripting en el cliente**: JavaScript o Flash etc
 □ Usando lenguajes de **scripting en el servidor**: Perl, PHP, Ruby, Python etc. Se utilizan incrustando etiquetas dentro de la página web. Cuando el cliente solicita esa página web, el servidor web interpreta estas etiquetas para realizar acciones en el servidor generando contenido dinámico. Por ejemplo, insertando información de una base de datos.

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

54

Tema 5. Capa de aplicación

3. La navegación web

➤ **Características HTTP:**

- **Usa los servicios de TCP (S.O.C.) en el puerto 80**
 - Inicio de conexión TCP, envío HTTP, cierre de conexión TCP
- **HTTP es "stateless" → Cookies**
 - El servidor no mantiene información sobre las peticiones de los clientes (su estado) y así ahorra recursos aunque hace más compleja la interacción
- **Existen dos tipos de servidores**
 - No persistente → Se envía únicamente un objeto en cada conexión TCP.
 - Persistente → Pueden enviarse multiples objetos sobre una única conexión TCP entre cliente y servidor

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

55

Universidad de Granada

55

Tema 5. Capa de aplicación

3. La navegación web

MENSAJES HTTP

1. El cliente HTTP (navegador) solicita un objeto identificado por su URL, en el ejemplo www.ugr.es/pages/Universidad. Según la configuración del servidor, si no se especifica nada, por defecto se sirve el fichero index.html
2. El cliente consulta al resolver de DNS por la dirección IP de www.ugr.es
3. DNS contesta 150.214.27.71 (IP virtual de un servicio balanceado)
4. El cliente abre una conexión TCP al puerto 80 de 150.214.27.71 (3 bandas)
5. El cliente envía una petición "GET /pages/universidad/ ..." (más otra información adicional: cabeceras, cookies, variables, etc)
6. El servidor responde enviando el fichero "index.html" por la misma conexión TCP
7. Al usar TCP el cliente y servidor de HTTP reciben un servicio orientado a conexión, fiable, sin errores, con control de flujo, con control de congestión, etc. Es decir una comunicación TRANSPARENTE y FIABLE.
8. Si es persistente se siguen solicitando objetos de la página ("GET...") por la conexión
9. Se cierra la conexión TCP y se liberan recursos en el servidor y cliente
10. El cliente visualiza el contenido

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

56

Tema 5. Capa de aplicación

3. La navegación web

MENSAJES HTTP

Firefox browser window showing a search for "Universidad de Granada" on the website www.ugr.es.

Diagram illustrating the HTTP message exchange process:

- 1a. El Cliente HTTP inicia conexión TCP al servidor HTTP (proceso) en www.ugr.es en el puerto 80 (segmento SYNC de TCP)
- 1b. El Servidor HTTP acepta la conexión y solicita al cliente abrir la conexión (SYNC+ACK)
2. El Cliente HTTP envía request message para el objeto
- 1c. El cliente confirma (ACK)
3. El servidor HTTP devuelve la respuesta

tiempo

4. Si es persistente → Envío de más objetos por la misma conexión TCP
5. Cierre de conexión TCP (liberación de recursos)
6. Nuevas conexiones TCP

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

57

57

Tema 5. Capa de aplicación

3. La navegación web

TIPOS DE MENSAJES HTTP

HTTP define dos tipos de mensajes (*request, response*):

1. *HTTP request message* (solicitudes del cliente al servidor):

Línea de petición (GET, POST, HEAD)

Líneas de cabecera

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Carriage return + line feed (extra carriage return, line feed)

Indican fin del mensaje

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

58

58

Tema 5. Capa de aplicación

3. La navegación web

TIPOS DE MENSAJES HTTP

HTTP define dos tipos de mensajes (*request, response*):

2. **HTTP response message:** (respuestas del servidor al cliente):

Línea de estado

Líneas de cabecera

Datos, ej. fichero html

59

```

HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

```

200 OK
301 Moved Permanently
400 Bad Request
404 Not Found
505 HTTP Version Not Supported

59

60

60



Tema 5. Capa de aplicación

3. La navegación web

PROTOCOLO HTTP 1.1 (RFC 2616) ▶

• **MÉTODOS** (acciones solicitadas por los clientes en los *request messages*):

- **OPTIONS**: solicitud de información sobre las opciones disponibles
- **GET**: solicitud de un recurso (puede ser condicional)
- **HEAD**: igual que GET pero el servidor no devuelve el "cuerpo" sólo cabeceras
- **POST**: solicitud al servidor para que acepte y subordine a la URI especificada, los datos incluidos en la solicitud,
- **PUT**: solicitud de sustituir la URI especificada con los datos incluidos en la solicitud.
- **DELETE**: solicitud de borrar la URI especificada.

• **CÓDIGOS DE RESPUESTA** (para los *response messages* del servicio):

- **1xx** indican mensajes exclusivamente informativos
- **2xx** indican algún tipo de éxito
- **3xx** redireccion al cliente a otra URL
- **4xx** indican un error
- **5xx** indican un error

• **CABECERAS** (47 *request headers* y 49 *response headers*)

From: , User-Agent:, Content-Type:, Content-Length:.....

http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

61

61

Tema 5. Capa de aplicación

3. La navegación web

PROTOCOLO HTTP 1.1 (RFC 2616) ▶

➤ <https://developer.mozilla.org/es/docs/Web/HTTP/Headers>

➤ https://www.tutorialspoint.com/http/http_quick_guide.htm

✓ **Cabeceras comunes para peticiones y respuestas**

- **Content-Type**: descripción MIME de la información contenida en este mensaje.
- **Content-Length**: longitud en bytes de los datos enviados, expresado en base decimal.
- **Content-Encoding**: formato de codificación de los datos enviados en este mensaje. Sirve, por ejemplo, para enviar datos comprimidos o encriptados.
- **Date**: fecha local de la operación. Las fechas deben incluir la zona horaria en que reside el sistema que genera la operación. Por ejemplo: Sunday, 12-Dec-96 12:21:22 GMT+01. No existe un formato único en las fechas.

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

62

62

Tema 5. Capa de aplicación

3. La navegación web

PROTOCOLO HTTP 1.1 (RFC 2616)

✓ Cabeceras sólo para peticiones del cliente

- **Accept:** campo opcional que contiene una lista de tipos MIME aceptados por el cliente.
- **Authorization:** clave de acceso que envía un cliente para acceder a un recurso de uso protegido o limitado. La información incluye el formato de autorización empleado, seguido de la clave de acceso propiamente dicha. La explicación se incluye más adelante.
- **From:** campo opcional que contiene la dirección de correo electrónico del usuario del cliente Web que realiza el acceso.

63

Universidad de Granada

Tema 5. Capa de aplicación

3. La navegación web

PROTOCOLO HTTP 1.1 (RFC 2616)

- **If-Modified-Since:** permite realizar operaciones GET condicionales, en función de si la fecha de modificación del objeto requerido es anterior o posterior a la fecha proporcionada. Puede ser utilizada por los sistemas de almacenamiento temporal de páginas. Es equivalente a realizar un HEAD seguido de un GET normal.
- **Referer:** contiene la URL del documento desde donde se ha activado este enlace. De esta forma, un servidor puede informar al creador de ese documento de cambios o actualizaciones en los enlaces que contiene. No todos los clientes lo envían.
- **User-agent:** cadena que identifica el tipo y versión del cliente que realiza la petición. Por ejemplo, los browsers de Netscape envían cadenas del tipo User-Agent: Mozilla/3.0 (WinNT; I)

64

Universidad de Granada

Tema 5. Capa de aplicación

3. La navegación web

PROTOCOLO HTTP 1.1 (RFC 2616)

✓ **Cabeceras sólo para respuestas del servidor HTTP**

- **Allow:** informa de los comandos HTTP opcionales que se pueden aplicar sobre el objeto al que se refiere esta respuesta. Por ejemplo, Allow: GET, POST.
- **Expires:** fecha de expiración del objeto enviado. Los sistemas de cache deben descartar las posibles copias del objeto pasada esta fecha. Por ejemplo, Expires: Thu, 12 Jan 97 00:00:00 GMT+1. No todos los sistemas lo envían.
- **Last-modified:** fecha local de modificación del objeto devuelto. Se puede corresponder con la fecha de modificación de un fichero en disco, o, para información generada dinámicamente desde una base de datos, con la fecha de modificación del registro de datos correspondiente.

65

Universidad de Granada

Tema 5. Capa de aplicación

3. La navegación web

WEB CACHÉ

Objetivo: satisfacer el requerimiento del cliente sin involucrar al servidor destino.

- Usuario configura el browser: Acceso Web vía cache
- browser envía todos los requerimientos HTTP al cache
 - Si objeto está en cache: cache retorna objeto
 - Sino cache requiere los objetos desde el servidor Web, y retorna el objeto al cliente

66

Universidad de Granada

Tema 5. Capa de aplicación

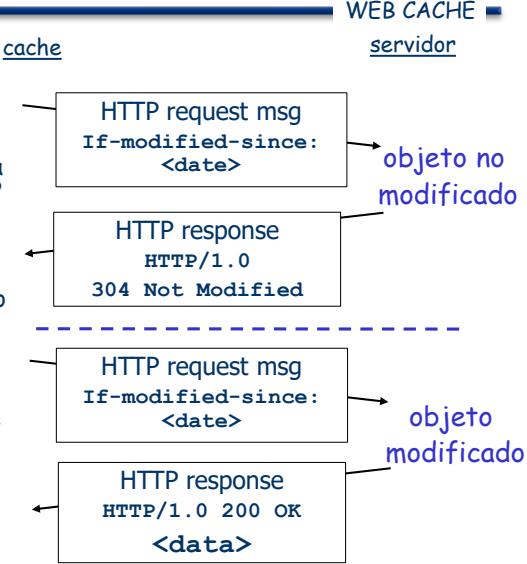
3. La navegación web

WEB CACHÉ

- Ejemplo de respuesta típica de un servidor configurado para gestionar cachés:

HTTP/1.1 200 OK
Date: Fri, 30 Oct 1998 13:19:41 GMT
Server: Apache/1.3.3 (Unix)
Cache-Control: max-age=3600
Expires: Fri, 30 Oct 1998 14:19:41 GMT
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
ETag: "3e86-410-3596fbbc"
Content-Length: 1040
Content-Type: text/html

- Las cabeceras se asocian al fichero en la caché local



Tema 5. Capa de aplicación

3. La navegación web

COOKIES

✓ Las **cookies** son pequeños ficheros de texto que se intercambian los clientes y servidores HTTP, para solucionar una de las principales deficiencias del protocolo: la falta de información de estado entre dos transacciones. Fueron introducidas por Netscape, y han sido estandarizadas en el RFC 2109.

- La primera vez que un usuario accede a un determinado documento de un servidor, éste proporciona una cookie que contiene datos que relacionarán posteriores operaciones.
- El cliente almacena la cookie en su sistema para usarla después. En los futuros accesos a este servidor, el navegador podrá proporcionar la cookie original, que servirá de nexo entre este acceso y los anteriores.
- Todo este proceso se realiza automáticamente, sin intervención del usuario.

69

Universidad de Granada

69

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Tema 5. Capa de aplicación

3. La navegación web

COOKIES

✓ Su aplicación más inmediata son los sistemas de **compra electrónica**. Estos supermercados virtuales necesitan relacionar el contenido de un pedido con el cliente que lo ha solicitado.

✓ Otro uso muy interesante son los **sistemas personalizados de recepción de información**, en los que es posible construir una página a medida, con información procedente de fuentes muy diversas. En accesos sucesivos, el cliente enviará la cookie, y el servidor podrá generar una página personalizada con las preferencias del usuario.

✓ Por último, algunas compañías emplean las cookies para realizar un seguimiento de los accesos a sus servidores WWW, identificando las páginas más visitadas, la manera en que se pasa de una a otra sección, etc.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>

70

Universidad de Granada

70

Fundamentos de Redes - Curso 2021 - 2022
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Tema 5. Capa de aplicación

3. La navegación web

COOKIES

- ✓ Un servidor HTTP envía los diferentes campos de una cookie con la nueva cabecera HTTP Set-Cookie:

Set-Cookie: Domain=www.unican.es; Path=/; Nombre=Luis; Expires Fri, 15-Jul-97 12:00:00 GMT

- Cuando se accede a una URL que verifica el par dominio/path registrado, el cliente enviará automáticamente la información de los diferentes campos de la cookie con la nueva cabecera HTTP Cookie:

```

sequenceDiagram
    participant Client
    participant Server
    Client->>Server: GET http://www.example.com/
    Note over Client: GET http://www.example.com/ HTTP/1.1
    Note over Server: HTTP/1.1 200 OK
    Note over Server: Set-Cookie: session-id=12345;
    Client->>Server: GET http://www.example.com/
    Note over Client: GET http://www.example.com/ HTTP/1.1
    Note over Client: Cookie: session-id=12345;
  
```

Universidad de Granada

71

71

3. La navegación web

ACceso RESTringido

- HTTP no es seguro, pero incluye cabeceras (`WWW-Authenticate` y `Authorization`) para restringir el acceso a recursos.
 - Es vulnerable a ataques por repetición.
- ```

sequenceDiagram
 participant Client
 participant Server
 Client->>Server: GET / HTTP/1.1
 Note over Client: GET / HTTP/1.1
 Note over Server: HTTP/1.1 401 Unauthorized
 Note over Server: WWW-Authenticate: Basic realm="Access to the staging site"
 Client->>Server: GET / HTTP/1.1
 Note over Client: GET / HTTP/1.1
 Note over Client: Authorization: Basic YXhhZGRpbjpvnGVuc2VxYWl1
 Note over Server: Check credentials
 Client->>Server:
 Note over Client:
 Note over Server: HTTP/1.1 200 OK or
 Note over Server: HTTP/1.1 403 Forbidden

```
- `WWW-Authenticate: <type> realm=<realm>[, charset="UTF-8"]`
  - `Authorization: <type> <credentials>`  
`<credentials> si <type> es BASIC incluye el username:password codificado en BASE64`



72

**Tema 5. Capa de aplicación**

**Esquema**

---

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación web
- 4. El correo electrónico**
5. Aplicaciones multimedia
6. Cuestiones y ejercicios

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

73

**Tema 5. Capa de aplicación**

**4. El correo electrónico**

---

- Elementos y protocolos principales:
  - Cliente de correo (Mail User Agent)
  - Servidor de correo (Mail Server ó Mail Transfer Agent)
  - Protocolo de envío:  
Simple Mail Transfer Protocol (SMTP)
  - Protocolos de descarga (o lectura):  
POP3, IMAP, HTTP
- Agente de usuario (MUA):
  - Compone, edita y lee mensajes de correo del buzón. Ej. Outlook, Thunderbird
- Servidor de correo (MTA)
  - Reenvía mensajes salientes y almacena en buzones los mensajes entrantes de cada usuario.

The diagram illustrates the email system architecture. It shows several 'mail server' boxes connected to each other and to 'user agent' boxes. Arrows indicate the flow of protocols: red arrows labeled 'SMTP' point from one mail server to another and from a mail server to a user agent. A blue arrow labeled 'POP3' points from a user agent back to a mail server. A red arrow labeled 'HTTP' points from a user agent to another user agent. A legend at the bottom right defines the icons: a green bar with yellow dots represents the 'outgoing message queue', and a yellow square represents a 'user mailbox'.

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

74

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**SMTP (RFC 2821)**

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

- SMTP se implementa mediante dos programas (incluidos ambos en cada *mail server*):
  - Cliente SMTP: se ejecuta en el *mail server (MTA)* que está enviando correo
  - Servidor SMTP: se ejecuta en el *mail server (MTA)* que está recibiendo correo
  - "sendmail" <http://en.wikipedia.org/wiki/Sendmail>
- SMTP usa TCP en el puerto 25. Es un protocolo orientado a texto.
- SMTP es un protocolo orientado a conexión, es *in-band* y es *state-full*: implica tres fases
  - Handshaking ("saludo")
  - Transferencia de mensajes
  - Cierre
- La interacción entre cliente SMTP y servidor SMTP se realiza mediante **commands / respuesta**
  - **comandos:** texto ASCII
  - **respuestas:** código de estado y frases explicativas
- Los mensajes deben estar codificados en ASCII de 7 bits!! ➔ Con la definición posterior de las extensiones MIME se pueden enviar ASCII de 8 bits y formatos enriquecidos

Universidad de Granada

75

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**SMTP (RFC 2821)**

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Pasos en el envío/recepción de correo

- 1) El usuario origen compone mediante su Agente de Usuario (MUA) un mensaje dirigido a la dirección de correo del usuario destino
- 2) Se envía con SMTP (ó HTTP) el mensaje al servidor de correo (MTA) del usuario origen que lo sitúa en la cola de mensajes salientes
- 3) El cliente SMTP abre una conexión TCP con el servidor de correo (MTA) (obtenido por DNS) del usuario destino
- 4) El cliente SMTP envía el mensaje sobre la conexión TCP
- 5) El servidor de correo del usuario destino ubica el mensaje en el *mailbox* del usuario destino
- 6) El usuario destino invoca su Agente de Usuario (MUA) para leer el mensaje utilizando POP3, IMAP ó HTTP

Universidad de Granada

76

Tema 5. Capa de aplicación

## 4. El correo electrónico

SMTP (RFC 2821)

*Propuesta de ejercicio:  
dibujar el diagrama de  
estados de SMTP*

```

S: 220 smtp1.ugr.es
C: HELO ugr.es
S: 250 smtp1.ugr.es
C: MAIL FROM: uno@ugr.es
S: 250 Ok
C: RCPT TO: dos@ugr.es
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: Correo estúpido
C: Tengo ganas de enviarte un correo...
C: ¿Te importa si lo hago?
C: .
S: 250 Ok: queued as KJSADHFFWDF
C: QUIT
S: 221 Bye

```

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

77

Tema 5. Capa de aplicación

## 4. El correo electrónico

SMTP (RFC 2821)

### >Comandos SMTP: cliente

*Propuesta de ejercicio:  
dibujar el diagrama de  
estados de SMTP*

| Comando           | Descripción                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HELO (ahora EHLO) | Identifica el remitente al destinatario.                                                                                                                                                                                                                             |
| MAIL FROM         | Identifica una transacción de correo e identifica al emisor.                                                                                                                                                                                                         |
| RCPT TO           | Se utiliza para <b>identificar un destinatario individual</b> . Si se necesita identificar múltiples destinatarios es necesario repetir el comando.                                                                                                                  |
| DATA              | Permite enviar una serie de líneas de texto. El tamaño máximo de una línea es de 1.000 caracteres. Cada línea va seguida de un retorno de carro y avance de línea <CR><LF>. <b>La última línea debe llevar únicamente el carácter punto "."</b> seguido de <CR><LF>. |
| RSET              | Aborta la transacción de correo actual.                                                                                                                                                                                                                              |
| NOOP              | No operación. <b>Indica al extremo que envíe una respuesta positiva. Keepalives</b>                                                                                                                                                                                  |
| QUIT              | Pide al otro extremo que envíe una respuesta positiva y cierre la conexión.                                                                                                                                                                                          |
| VRFY              | Pide al receptor que confirme que un nombre identifica a un destinatario valido.                                                                                                                                                                                     |
| EXPN              | Pide al receptor la <b>confirmación de una lista de correo</b> y que devuelva los nombres de los usuarios de dicha lista.                                                                                                                                            |
| HELP              | Pide al otro extremo información sobre los comandos disponibles.                                                                                                                                                                                                     |
| TURN              | El emisor pide que se <b>inviertan los papeles</b> , para poder actuar como receptor. El receptor puede negarse a dicha petición.                                                                                                                                    |
| SOML              | Si el destinatario está conectado, entrega el mensaje directamente al terminal, en caso contrario lo entrega como correo convencional.                                                                                                                               |
| SAML              | Entrega del mensaje en el buzón del destinatario. En caso de estar conectado también lo hace al terminal.                                                                                                                                                            |
| SEND              | Si el destinatario está conectado, entrega el mensaje directamente al terminal.                                                                                                                                                                                      |

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

78

Tema 5. Capa de aplicación

**4. El correo electrónico**

SMTP (RFC 2821)

➤ **Códigos de respuesta SMTP: servidor**

| Código | Descripción                                                                |
|--------|----------------------------------------------------------------------------|
| 211    | Estado del sistema.                                                        |
| 214    | Mensaje de ayuda.                                                          |
| 220    | Servicio preparado.                                                        |
| 221    | Servicio cerrando el canal de transmisión.                                 |
| 250    | Solicitud completada con éxito.                                            |
| 251    | Usuario no local, se enviará a <dirección de reenvío>                      |
| 354    | Introduzca el texto, finalice con <CR><LF>,<CR><LF>.                       |
| 421    | Servicio no disponible.                                                    |
| 450    | Solicitud de correo no ejecutada, servicio no disponible (buzón ocupado).  |
| 451    | Acción no ejecutada, error local de procesamiento.                         |
| 452    | Acción no ejecutada, insuficiente espacio de almacenamiento en el sistema. |
| 500    | Error de sintaxis, comando no reconocido.                                  |
| 501    | Error de sintaxis. P.ej contestación de SMTP a ESMTP                       |
| 502    | Comando no implementado.                                                   |
| 503    | Secuencia de comandos errónea.                                             |
| 504    | Parámetro no implementado.                                                 |
| 550    | Solicitud no ejecutada, buzón no disponible.                               |
| 551    | Usuario no local, pruebe <dirección de reenvío>. Si no se tiene cuenta     |
| 552    | Acción de correo solicitada abortada.                                      |
| 553    | Solicitud no realizada (error de sintaxis).                                |
| 554    | Fallo en la transacción.                                                   |

Universidad de Granada

79

Tema 5. Capa de aplicación

**4. El correo electrónico**

EXTENSIONES MIME

➤ **Multipurpose Internet Mail Protocol Extensions (MIME):**

- Nada cambia respecto a la arquitectura de correo anterior.
- Las extensiones de MIME van encaminadas a soportar:
  - Texto en conjuntos de caracteres distintos de US-ASCII;
  - Adjuntos que no son de tipo texto;
  - Cuerpos de mensajes con múltiples partes (multi-part);
  - Información de encabezados con conjuntos de caracteres distintos de ASCII.
- MIME está especificado en seis RFCs: RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 y RFC 2077.

Universidad de Granada

80

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**EXTENSIONES MIME**

➤ No se debe confundir los mensajes del protocolo con el formato de almacenamiento.

Versión MIME  
Método de codificación  
Datos multimedia  
Tipo, subtipo,  
...  
Datos codificados

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data
.....base64 encoded data
```

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

81

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**EXTENSIONES MIME**

➤ **Cabeceras de mensajes MIME**

| Cabecera                   | Descripción                                                                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| MIME-Version:              | Identifica la versión de MIME. Si no existe se considera que el mensaje es texto normal en inglés.                                              |
| Content-Description:       | Cadena de texto que describe el contenido. Esta cadena es necesaria para que el destinatario sepa si desea descodificar y leer el mensaje o no. |
| Content-Id:                | Identificador único, usa el mismo formato que la cabecera estándar Message-Id.                                                                  |
| Content-Transfer-Encoding: | Indica la manera en que está envuelto el cuerpo del mensaje.                                                                                    |
| Content-Type:              | Especifica la naturaleza del cuerpo del mensaje.                                                                                                |

➤ **Content-Transfer-Encoding**

- Indica la manera en que está envuelto el cuerpo para su transmisión, ya que podría haber problemas con la mayoría de los caracteres distintos de letras, números y signos de puntuación.
- Existen 5 tipos de codificación (RFC1521) : *ASCII 7, ASCII 8, codificación binaria, base64 y entrecomillada-imprimible.7.2*

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

82

**Tema 5. Capa de aplicación**

**4. El correo electrónico**

**EXTENSIONES MIME**

**MIME: Content-Type: tipos y subtipos**

- La lista inicial de tipos y subtipos especificada por el RFC 1521 es:

| <b>Tipo</b> | <b>Subtipo</b> | <b>Descripción</b>                              |
|-------------|----------------|-------------------------------------------------|
| Text        | Plain          | Texto sin formato.                              |
|             | Richtext       | Texto con comandos de formato sencillos.        |
| Image       | Gif            | Imagen fija en formato GIF.                     |
|             | Jpeg           | Imagen fija en formato JPEG.                    |
| Audio       | Basic          | Sonido.                                         |
| Video       | Mpeg           | Película en formato MPEG.                       |
| Application | Octet-stream   | Secuencia de bytes no interpretada.             |
|             | Postscript     | Documento imprimible PostScript.                |
| Message     | Rfc822         | Mensaje MIME RFC 822.                           |
|             | Partial        | Mensaje dividido para su transmisión.           |
|             | External-body  | El mensaje mismo debe obtenerse de la red.      |
| Multipart   | Mixed          | Partes independientes en el orden especificado. |
|             | Alternative    | Mismo mensaje en diferentes formatos.           |
|             | Parallel       | Las partes deben verse simultáneamente.         |
|             | Digest         | Cada parte es un mensaje RFC 822 completo.      |

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

**Tema 5. Capa de aplicación**

**4. El correo electrónico**

**EXTENSIONES MIME**

**MIME: Content-Type: tipo application:**

- El tipo **application** es un tipo general para los formatos que requieren procesamiento externo no cubierto por ninguno de los otros tipos.
- El subtipo **octet-stream** simplemente es una secuencia de bytes no interpretados, tal que a su recepción, un agente de usuario debería presentarla en pantalla sugiriendo al usuario que se copie en un archivo y solicitando un nombre de archivo.
- El subtipo **postscript**, se refiere al lenguaje PostScript de Adobe Systems. Aunque un agente de usuario puede llamar a un intérprete PostScript externo para visualizarlo, hacerlo no está exento de riesgos al ser PostScript un lenguaje de programación completo.

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

Tema 5. Capa de aplicación

## 4. El correo electrónico

EXTENSIONES MIME

**MIME: Content-Type: tipo message:**

- El tipo **message** permite que un mensaje esté encapsulado por completo dentro de otro. Este esquema es útil para reenviar correo electrónico.
- El subtipo **rfc822** se utiliza cuando se encapsula un mensaje RFC 822 completo en un mensaje exterior.
- El subtipo **partial** hace posible dividir un mensaje encapsulado en pedazos y enviarlos por separado. Los parámetros hacen posible ensamblar correctamente todas las partes en el destino. E.g. 1/3, 2/3, 3/3.
- El subtipo **external-body** puede usarse para mensajes muy grandes, por ejemplo, películas de vídeo. En lugar de incluir el archivo MPEG en el mensaje, se da una dirección de FTP y el agente del receptor puede obtenerlo a través de la red cuando se requiera.

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

85

Tema 5. Capa de aplicación

## 4. El correo electrónico

EXTENSIONES MIME

**MIME: Content-Type: tipo multipart**

- El tipo es **multipart**, que permite que un mensaje contenga más de una parte, con el comienzo y el fin de cada parte claramente delimitados.
- El subtipo **mixed** permite que cada parte sea diferente.
- El subtipo **alternative** indica que cada parte contiene el mismo mensaje, pero expresado en un medio o codificación diferente.
- El subtipo **parallel** se usa cuando todas las partes deben "verse" simultáneamente, por ejemplo, en los canales de audio y vídeo de las películas.
- El subtipo **digest** se usa cuando se juntan muchos mensajes en un mensaje compuesto.

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

86

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**EJEMPLO MULTIPART/MIXED**

```

From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble. It is to be ignored, though it
is a handy place for mail composers to include an
explanatory note to non-MIME compliant readers.
--simple boundary

This is implicitly typed plain ASCII text.
It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain ASCII text.
It DOES end with a linebreak.

--simple boundary--
This is the epilogue. It is also to be ignored.

```

Universidad de Granada

87

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**PROTOCOLOS DE ACCESO: POP3**

Ej: POP3 PROTOCOL TCP PORT = 110

**Fase de autorización**

Comandos del cliente:  
**user:** nombre de usuario  
**pass:** contraseña

Respuestas del servidor  
**+OK**  
**-ERR**

**Fase de transacción, cliente:**  
**list:** lista mensajes por número  
**retr:** obtiene mensajes por num.  
**dele:** borra  
**quit**

**Fase de actualización del servidor**  
(tras desconexión)

```

S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

```

Universidad de Granada

88

Tema 5. Capa de aplicación

## 4. El correo electrónico

PROTOCOLOS DE ACCESO: POP3

**Comandos POP3**

| Comando             | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USER identification | Este comando permite la autenticación. Debe estar seguido del nombre de usuario, es decir, una cadena de caracteres que identifique al usuario en el servidor. El comando USER debe preceder al comando PASS.                                                                                                                                                                                                                                                                                                      |
| PASS password       | El comando PASS permite especificar la contraseña del usuario cuyo nombre ha sido especificado por un comando USER previo.                                                                                                                                                                                                                                                                                                                                                                                         |
| STAT                | Información acerca de los mensajes del servidor                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| RETR                | Número del mensaje que se va a recoger                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| DELE                | Número del mensaje que se va a eliminar                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| LIST [msg]          | Número del mensaje que se va a mostrar                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| NOOP                | Permite mantener la conexión abierta en caso de inactividad                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| TOP <messageID> <n> | Comando que muestra <i>n</i> líneas del mensaje, cuyo número se da en el argumento. En el caso de una respuesta positiva del servidor, éste enviará de vuelta los encabezados del mensaje, después una línea en blanco y finalmente las primeras <i>n</i> líneas del mensaje.                                                                                                                                                                                                                                      |
| UIDL [msg]          | Solicitud al servidor para que envíe una línea que contenga información sobre el mensaje que eventualmente se dará en el argumento. Esta línea contiene una cadena de caracteres denominada <i>unique identifier listing</i> ( <i>lista de identificadores únicos</i> ) que permite identificar de manera única el mensaje en el servidor, independientemente de la sesión. El argumento opcional es un número relacionado con un mensaje existente en el servidor POP, es decir, un mensaje que no se ha borrado. |
| QUIT                | El comando QUIT solicita la salida del servidor POP3. Lleva a la eliminación de todos los mensajes marcados como eliminados y envía el estado de esta acción.                                                                                                                                                                                                                                                                                                                                                      |

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

89

Tema 5. Capa de aplicación

## 4. El correo electrónico

PROTOCOLOS DE ACCESO: IMAP

**IMAP4:**

- Permite trabajar con el correo como si fuese local.

**Estados:**

- No autenticado (NA)
- Autenticado (A)
- Seleccionado (S)
- Desconexión (D)

**Comandos (puerto 143):**

- CAPABILITY, NOOP, LOGOUT
- NA → LOGIN, AUTHENTICATE
- A → SELECT, CREATE, DELETE, LIST, APPEND, UN/SUBSCRIBE, ...
- S → CHECK, CLOSE, SEARCH, FETCH, STORE, COPY, ...

**Más información en: <http://www imap.org>**

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

Universidad de Granada

90

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**PROTOCOLOS DE ACCESO: IMAP**

```
#> telnet sal.ugr.es 143
* OK sal.ugr.es IMAP4rev1 v12.264 server ready
a001 LOGIN usuario clave
a001 OK LOGIN completed
a002 SELECT inbox
* 18 EXISTS
* 2 RECENT
* OK [UIDVALIDITY 3857529045] UID validity status
* OK [UIDNEXT 17] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
a002 OK [READ-WRITE] SELECT completed
a003 FETCH 12 full
* 12 FETCH (FLAGS (\Seen) INTERNALDATE "14-Jul-1993 02:44:25 -0700"
RFC822.SIZE 4282 ENVELOPE ("Wed, 14 Jul 1993 02:23:25 -0700 (PDT)"
"IMAP4 WG mtg summary and minutes"
(("Terry Gray" NIL "gray" "cac.washington.edu"))
((NIL NIL "imap" "cac.washington.edu"))
((NIL NIL "minutes" "CNRI.Reston.VA.US")
("John Klensin" NIL "KLENSIN" "INFOODS.MIT.EDU")) NIL NIL
"<B27397-0100000@cac.washington.edu>")
BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 3028 92))
a003 OK FETCH completed
a004 FETCH 12 RFC822.header
* 12 FETCH (RFC822.HEADER {346}
Date: Wed, 14 Jun 1993 02:23:25 -0700 (PDT)
From: Terry Gray <gray@cac.washington.edu>
Subject: IMAP4 WG mtg summary and minutes
To: imap@cac.washington.edu
cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@INFOODS.MIT.EDU>
Message-ID: <B27397-0100000@cac.washington.edu>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
)
a004 OK FETCH completed
a005 STORE 12 +flags \deleted
* 12 FETCH (FLAGS (\Seen \Deleted))
a005 OK STORE completed
a006 LOGOUT
* BYE sal.ugr.es IMAP4rev1 server terminating connection
a006 OK LOGOUT completed
#> -

```

Universidad de Granada

91

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**PROTOCOLOS DE ACCESO: IMAP**

**telnet capone.rutgers.edu 143**

```
Trying 192.168.5.240...
Connected to 192.168.5.240.
Escape character is '^'.
* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS STARTTLS AUTH=LOGIN] capone.rutgers.edu
IMAP4rev1 2003.339 at Wed, 13 Apr 2005 01:38:58 -0400 (EDT)
Client A1 LOGIN mailtest Password
Server A1 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS BINARY
UNSELECT SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND]
User mailtest authenticated
Client A2 SELECT Inbox
* 2 EXISTS
* 2 RECENT
* OK [UIDVALIDITY 1113370837] UID validity status
* OK [UIDNEXT 3] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags
* OK [UNSEEN 1] first unseen message in /var/mail/mailtest
Server A2 OK [READ-WRITE] SELECT completed
Client A3 FETCH 2 BODY[HEADER]
* 2 FETCH (BODY[HEADER] {670}
Return-Path:
X-Original-To: mailtest@capone.rutgers.edu
Delivered-To: mailtest@capone.rutgers.edu
```

Universidad de Granada

92

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**PROTOCOLOS DE ACCESO: IMAP**

```

Received: from node18.rutgers.edu (node18 [192.168.5.38])
 by capone.rutgers.edu (Postfix) with ESMTP id A291B2B15C
 for ; Tue, 12 Apr 2005 22:23:53 -0400 (EDT)
Received: from me?here.com (unknown [192.168.5.250])
 by node18.rutgers.edu (Postfix) with SMTP id 4653B14112
 for ; Tue, 12 Apr 2005 22:24:03 -0400 (EDT)
To: some_guru@somewhere.com
From: pp@pp.com
Subject: Forged e-mail
Message-Id: <20050413022403.4653B14112@node18.rutgers.edu>
Date: Tue, 12 Apr 2005 22:24:03 -0400 (EDT)

)
* 2 FETCH (FLAGS (\Recent \Seen))
Server A3 OK FETCH completed
Client A4 FETCH 2 BODY[TEXT]
* 2 FETCH (BODY[TEXT] {88}
Hey,
The "To:" and "From:" are non-existent, but you still get the e-mail.
bye, bye
)
Server A4 OK FETCH completed
Client A5 LOGOUT
* BYE capone.rutgers.edu IMAP4rev1 server terminating connection
Server A5 OK LOGOUT completed
Connection closed by foreign host.

```

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

93

**Tema 5. Capa de aplicación**

## 4. El correo electrónico

**PROTOCOLOS DE ACCESO**

- **Ventajas de IMAP4:**
  - Permite organización en carpetas en el lado del servidor (MTA)
  - Para ello, mantiene información entre sesiones (asociando *flags* a los mensajes).
  - Permite la descarga de partes de los mensajes.
  - Posible acceder con varios clientes (POP también, pero en modo descargar y guardar)
- **Ventajas de Web MAIL:**
  - Organización total en el servidor, accesible desde cualquier cliente con HTTP.
  - Seguridad: Uso extendido de HTTPS

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

94

## 4. El correo electrónico

➤ Listado de puertos relacionados con e-mail:

POP3 - port 110  
IMAP - port 143  
SMTP - port 25  
HTTP - port 80  
Secure SMTP (SSMTP) - port 465  
Secure IMAP (IMAP4-SSL) - port 585  
IMAP4 over SSL (IMAPS) - port 993  
Secure POP3 (SSL-POP) - port 995

## Esquema

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación web
4. El correo electrónico
5. **Aplicaciones multimedia**
6. Cuestiones y ejercicios

**Tema 5. Capa de aplicación**

## 5. Aplicaciones multimedia

➤ Conceptos: IP = "tecnología de convergencia" →



**Aplicaciones Multimedia:** audio, vídeo, juegos, real-time

**Calidad de servicio (QoS):** capacidad de ofrecer el rendimiento requerido para una aplicación

**IP ofrece mejor esfuerzo (best effort):** sin garantías de QoS

Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

97

**Tema 5. Capa de aplicación**

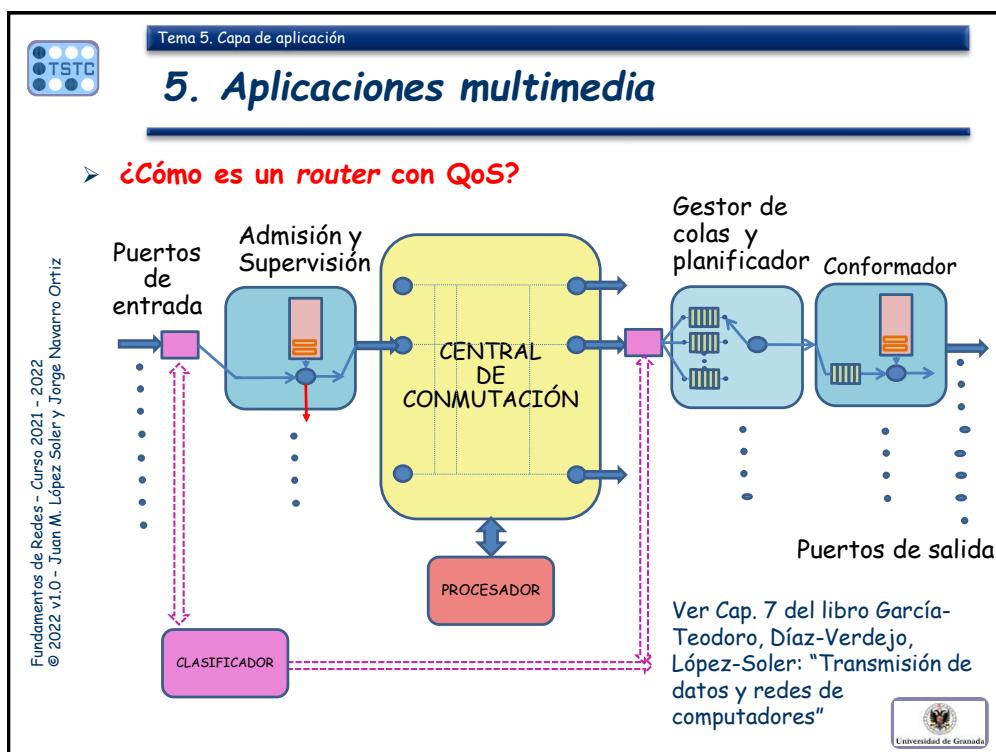
## 5. Aplicaciones multimedia

- **Tipos de aplicaciones**
  - Flujo de audio y vídeo (*streaming*) almacenado → Ej. YouTube
  - Flujo de audio y vídeo en vivo → Ej. emisoras de radio o IPTV
  - Audio y vídeo interactivo → Ej. Skype
- **Características fundamentales**
  - Elevado ancho de banda
  - Tolerantes relativamente a la pérdida de datos
  - Exigen *Delay* (retardo) acotado
  - Exigen *Jitter* (fluctuación del retardo) acotado
  - Se pueden beneficiar de usar de *multicast* (direcciones destino de grupo)

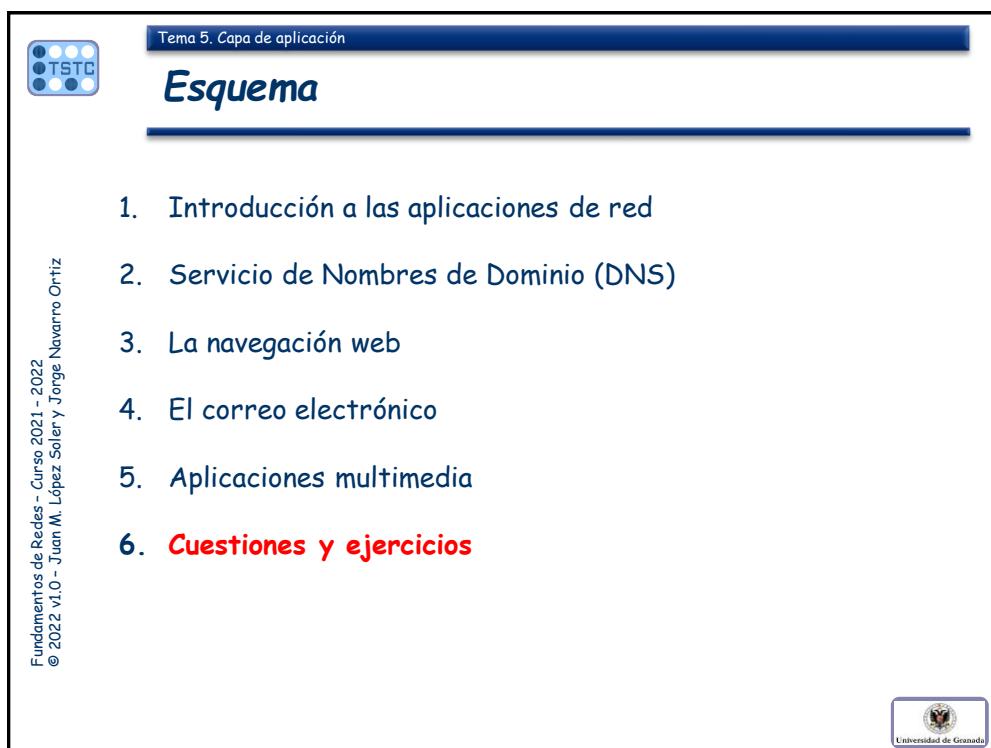
Universidad de Granada

Fundamentos de Redes - Curso 2021 - 2022  
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

98



99



100

## 7. Cuestiones y ejercicios

2. Discuta las características de las siguientes aplicaciones en términos de su tolerancia a la pérdida de datos, los requisitos temporales, la necesidad de rendimiento mínimo y la seguridad.

La telefonía móvil  
WhatsApp  
YouTube  
Spotify  
Comercio electrónico

## 7. Cuestiones y ejercicios

8. Una sucursal con 50 empleados en Granada tiene una red interna basada en *FastEthernet* (100Mbps) que se conecta a Internet con una red de acceso *ADSL* de 0,5 Mbps de subida y 1,5 Mbps de bajada. Cada empleado, en el desempeño de su trabajo, realiza un promedio de 2000 solicitudes de información a la hora a un servidor de Base de Datos ubicado en la central del banco, en Madrid, donde cada solicitud supone el envío por parte del servidor de 10 registros de 1KB cada uno. Adicionalmente, la modificación de datos tras algunas de estas solicitudes supone el envío de 100 actualizaciones, de 10 registros, a la hora desde la sucursal al servidor. El resto de los servicios telemáticos se restringe.

- Calcule la velocidad de transmisión requerida. ¿Es la velocidad del enlace de acceso suficiente?
- Si se dobla la velocidad del enlace? ¿Cuál sería el tiempo de cola que esperaría en promedio cada solicitud en el enlace descendente antes de ser enviada? Considere que cada registro se envía por separado, con una cabecera de tamaño despreciable
- Si, alternativamente, se diseña una caché que permite evitar un 70% de los accesos a la BD ¿cuál sería el tiempo de cola que esperaría en promedio cada solicitud en el enlace descendente? ¿Qué solución es mejor, la b. o esta?

Tema 5. Capa de aplicación



## 7. Cuestiones y ejercicios

1. Explicar por qué cuando solicitamos <http://www.google.com> desde nuestro navegador, se muestra la URL servida desde ([www.google.es](http://www.google.es))

¿qué relación tienen esos 2 nombres de dominio?  
¿guarda google información sobre nuestra localización? ¿cómo se obtiene?  
¿qué herramientas e información se necesita?  
¿qué ocurre y cómo influye si configuro en mi navegador como lenguaje preferido "francés"?  
¿pueden servirse páginas dependiendo de nuestra localización? ¿en su caso, con qué precisión?

Sugerencia: Usar el analizador <http://www.wireshark.org> para mostrar trazas

## 2 Relaciones de Ejercicios

### 2.1. Relación 1

# Boletín de Ejercicios nº1

1. Explique brevemente las funciones de cada una de las capas del modelo de comunicación de datos OSI.
2. Si la unidad de datos de protocolo en la capa de enlace se llama trama y la unidad de datos de protocolo en la capa de red se llama paquete, ¿son las tramas las que encapsulan los paquetes o son los paquetes los que encapsulan las tramas? Explicar la respuesta.

En este caso, son las tramas las que encapsulan los paquetes ya que la información de la capa de enlace se incorpora después en el emisor que la de la capa de red, por lo que la encapsula. Así mismo, en los nodos intermedios y el receptor se identifica primero el contenido de la capa de enlace para, posteriormente, ver su contenido, que comenzaría con la información de la capa de red.

3. Averigüe qué son los sistemas de representación de datos “*Little Endian*” y “*Big Endian*”. ¿Puede un *host* que utilice representación *Little Endian* interpretar mensajes de datos numéricos provenientes de un *host* que utilice representación *Big Endian* y viceversa? Discuta la respuesta.

Los términos “*Little Endian*” y “*Big Endian*” hacen referencia al formato en el que se almacenan los datos de más de un byte en un ordenador. “*Little Endian*” significa desde el menos significativo hasta el final, es decir, se comienza por el byte menos significativo. “*Big Endian*” significa lo contrario. Así, para el valor hexadecimal 0x44332211 tenemos:

*Little Endian*: 0x11, 0x22, 0x33, 0x44

*Big Endian*: 0x44, 0x33, 0x22, 0x11

El modelo de comunicación de datos OSI (*capa de Presentación*) ya contempla la necesidad de existencia de software de red que solucione la diferencia en los sistemas de representación de datos que pueda existir en los diferentes dispositivos de red (*hosts*) interconectados. Desde el punto de vista práctico lo que se hace es convertir los datos a un *formato estándar de red* antes de ser transmitidos hacia la red y convertir los datos de este formato de red al formato del host (“*Little Endian*” o “*Big Endian*”, en el caso que nos ocupa) al ser recibidos desde la red. En algunos lenguajes de programación (ej. JAVA) este proceso es transparente al programador; en otros (ej. C/C++) el programador debe realizar explícitamente la conversión (ej. a través de las funciones ntohs(), htons(), htonl(), htons() en el caso de C/C++).

4. Averigüe qué *ISPs* (*Internet Service Provider*) operan en España.
5. ¿Qué diferencia, en el contexto de una red de computadores, existe entre la tecnología de difusión y la tecnología punto-a-punto?

La tecnología de difusión es aquella donde los datos se difunden a todos los dispositivos de la red, típicamente debido al uso de un medio de comunicación compartido, como el aire. Son los dispositivos los que deciden qué datos o paquetes son destinados a ellos y por tanto los aceptan.

La tecnología punto a punto es aquella que establece un medio dedicado para la comunicación entre dos dispositivos, de forma que todos los datos o paquetes en ese medio tienen un destino implícito.

6. Un sistema tiene una jerarquía de protocolos de  $n$  capas. Las aplicaciones generan mensajes de  $M$  bytes de longitud. En cada capa se añade una cabecera de  $h$  bytes. ¿Qué fracción del ancho de banda de la red se llena con cabeceras? Aplique el resultado a una conexión a 512 Kbps con tamaño de datos de 1500 Bytes y 4 capas, cada una de las cuales añade 64 Bytes de cabecera.

La fracción del ancho de banda que se llena con cabeceras es (tomando únicamente dos decimales y redondeando con el tercero):

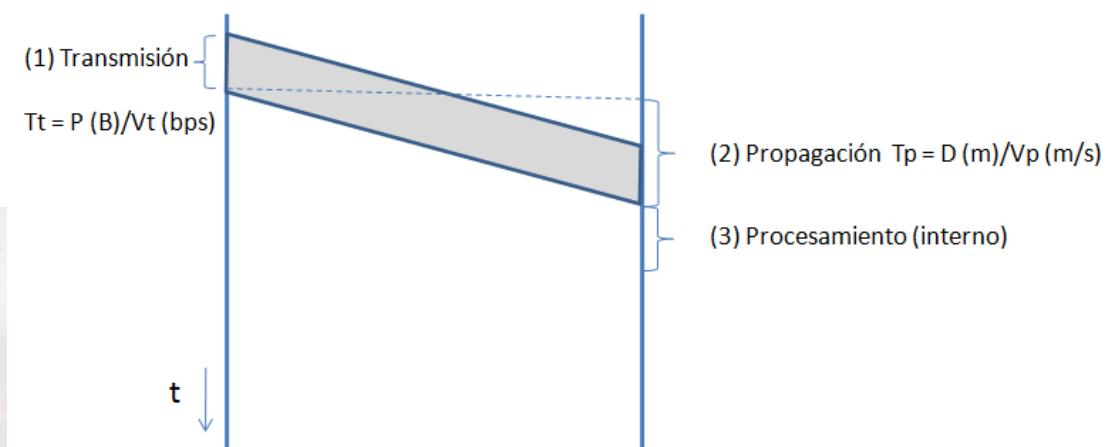
$$\frac{n \times h}{n \times h + M} = \frac{4 \times 64B}{4 \times 64B + 1500B} = 0,15 \text{ (15%)}$$

¿Qué velocidad real de envío de datos resulta?

$$v_{realdatos} = \left(1 - \frac{nxh}{nxh + M}\right) \times 512Kbps = 435,2Kbps$$

7. ¿Cuál es el tiempo necesario en enviar un paquete de 1000 Bytes, incluidos 50 Bytes de cabecera, por un enlace de 100 Mbps y 10Km? ¿cuál es el tiempo mínimo desde que se envía hasta que se recibe confirmación? ¿qué relación hay entre este tiempo y los temporizadores en, por ejemplo, las capas de enlace y transporte?

El retardo asociado a la comunicación del paquete incluye el retardo de transmisión y el retardo de propagación, ya que el retardo de procesamiento no afecta a la comunicación sobre la línea.



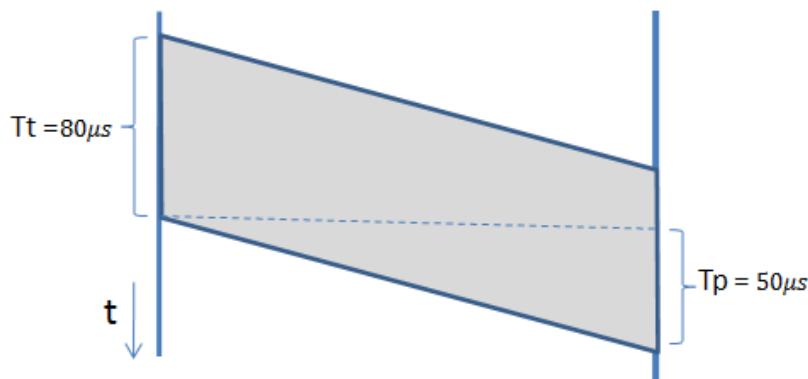
El retardo de transmisión es igual a:

$$T_t = \frac{1000 B \times 8 b/B}{100 Mbps} = 80\mu s$$

Y el de propagación será:

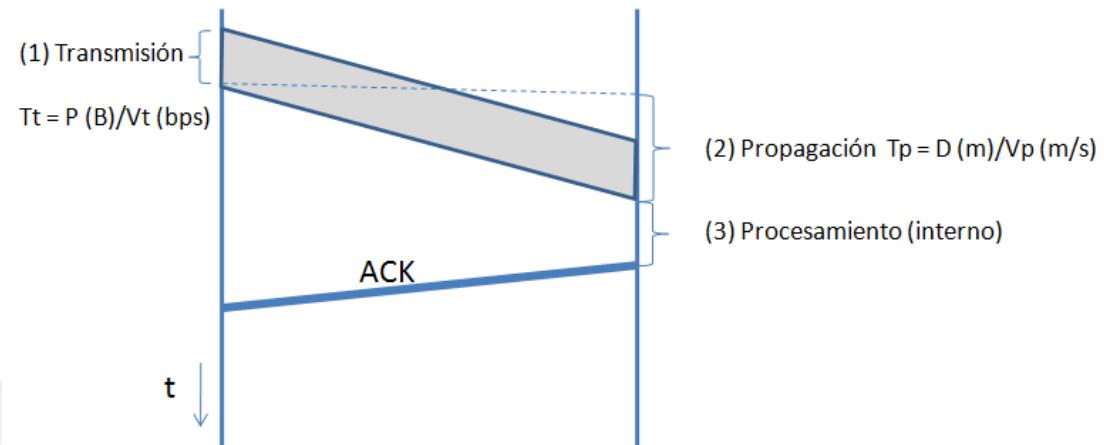
$$T_p = \frac{10 Km \times 1000 m/Km}{2 \cdot 10^8 m/s} = 50\mu s$$

Así, una figura más representativa de lo que está pasando sería la siguiente:



El tiempo necesario para el envío completo sería la suma de ambos retardos, es decir  $130\mu s$ .

Para calcular el tiempo desde que se envía hasta que se recibe la confirmación, sí se debe tener en cuenta el retardo de procesamiento, tras el cual se enviará un paquete de confirmación (ACK) que a su vez tendrá retardo de transmisión y de propagación.



El enunciado no especifica ninguna información sobre el retardo de procesamiento. Considerando la velocidad de procesamiento en los dispositivos modernos, una asunción típica es que el tiempo de procesamiento es varios órdenes de magnitud

menor que los otros retardos y, por tanto, despreciable. Es lo que haremos en esta ocasión, pensando además que nos piden el tiempo mínimo hasta que se recibe la confirmación.

El tiempo de propagación de cualquier bit en el mismo medio es el mismo, por lo que el retardo de propagación del ACK es el mismo que el del paquete de datos ( $50 \mu s$ ) No obstante, el paquete ACK normalmente sólo consta de cabecera, por lo que tiene un tamaño de 50 Bytes, y su tiempo de transmisión es:

$$T_{ACK} = \frac{50 B \times 8 b/B}{100 Mbps} = 4\mu s$$

El tiempo mínimo hasta que se recibe la confirmación sería:

$$T = T_t + 2 \cdot T_p + T_{ACK} = 184\mu s$$

Un temporizador de control de flujo en capa de enlace o de transporte debe ser suficientemente mayor a este tiempo mínimo para evitar un re-envío inmediato de paquetes ante cualquier eventualidad mínima en la red, como un retardo en las colas (mayor retardo de procesamiento) por un cierto nivel de congestión.

8. Cuando se intercambia un fichero entre dos *hosts* se pueden seguir dos estrategias de confirmación. En la primera, el fichero se divide en paquetes que se confirman individualmente por el receptor, pero el fichero en conjunto no se confirma. En la segunda, los paquetes individuales no se confirman individualmente, es el fichero entero el que se confirma cuando llega completo. Discutir las dos opciones.

En la primera opción sólo hay que retransmitir aquellos paquetes del fichero que no llegan correctamente al destino. En la segunda opción si alguno/s de los paquetes no llega/n correctamente se debe retransmitir todo el fichero (con el consiguiente consumo innecesario de ancho de banda).

9. Clasifique como difusión o punto a punto cada uno de los siguientes sistemas de transmisión:
  - a. Radio y TV
  - b. Redes inalámbricas (WLAN)
  - c. ADSL
  - d. Redes de Cable.
  - e. Comunicaciones móviles (p.e., GSM, UMTS).

Radio y TV: difusión

Redes inalámbricas (WLAN): difusión

ADSL: punto a punto

Redes de Cable: difusión o punto a punto

Comunicaciones móviles (p.e., GSM, UMTS): difusión

**10. Clasifique los siguientes servicios como orientados a conexión / no orientados a conexión y confirmados /sin confirmación. Justifique la respuesta.**

- a. Correo postal ordinario
- b. Correo certificado
- c. Envío y recepción de fax
- d. Conversación telefónica
- e. Domiciliación bancaria de recibos
- f. Solicitud de certificado de empadronamiento

Correo postal ordinario: no orientado a conexión y no confirmado

Correo certificado: no orientado a conexión y confirmado

Envío y recepción de fax: orientado a conexión y confirmado

Conversación telefónica: orientado a conexión y confirmado

Domiciliación bancaria de recibos: no orientado a conexión y confirmado

Solicitud de certificado de empadronamiento: no orientado a conexión y confirmado

## 2.2. Relación 2

# Ejercicios – Tema 2

## 1. ¿Para qué sirve el programa *ping*? ¿y el programa *traceroute*?

El programa *ping* se utiliza para comprobar el estado de un host utilizando el protocolo ICMP. El host origen manda un mensaje de solicitud de eco con un identificador. Si el destino está conectado, la comunicación es posible y está correctamente configurada, el host origen recibirá un mensaje de respuesta de eco con el mismo código. Este programa sirve asimismo para calcular empíricamente la latencia de la conexión.

El programa *traceroute* se utiliza para averiguar todos los nodos intermedios en la comunicación entre dos dispositivos y la latencia de cada salto. Para ello, se utilizan varios paquetes ICMP de solicitud de eco con un valor creciente del campo TTL (1,2,...) de la cabecera IP, de forma que se envía un paquete ICMP de TTL excedido en el primer nodo intermedio, el segundo, etc. hasta el nodo destino.

## 2. ¿Qué protocolos de un paquete puede cambiar un *router*? ¿En qué circunstancias?

Un *router* puede cambiar los protocolos situados debajo de la capa de red, siempre que sea necesario debido a que las redes que interconecta tengan dichos protocolos diferentes. Por ejemplo, una red doméstica típica es aquella basada en wifi (IEEE 802.11) y con acceso a Internet contratado con tecnología ADSL. En este caso, el router inalámbrico deberá modificar el protocolo de las capas físicas y de enlace convenientemente.

## 3. Describa brevemente la diferencia entre un *switch*, *router* y un *hub*.

Un *hub* o concentrador es un dispositivo de interconexión de hosts o dispositivos finales que trabaja puramente en difusión. Esto implica que cada vez que se recibe un paquete por una de sus interfaces de entrada, el mismo se reenvía por todas las de salida. Se puede considerar que opera en capa física del modelo OSI, ya que se limita a realizar la retransmisión de los paquetes.

Un *switch* o conmutador es un dispositivo de interconexión de hosts o dispositivos finales que puede trabajar inicialmente en difusión, como un *hub*, pero que a partir del envío de tráfico puede reconocer qué dispositivos están conectados en qué interfaces. Así, una red interconectada con un *switch* no opera en difusión, de forma que el paquete con un determinado destino es únicamente re-enviado por la interfaz de conexión apropiada. Opera en capa de enlace del modelo OSI.

Un *router* o enrutador es un dispositivo de interconexión de redes que opera en capa de red del modelo OSI, por lo que tiene funcionalidades asociadas al enrutamiento entre distintas redes interconectadas.



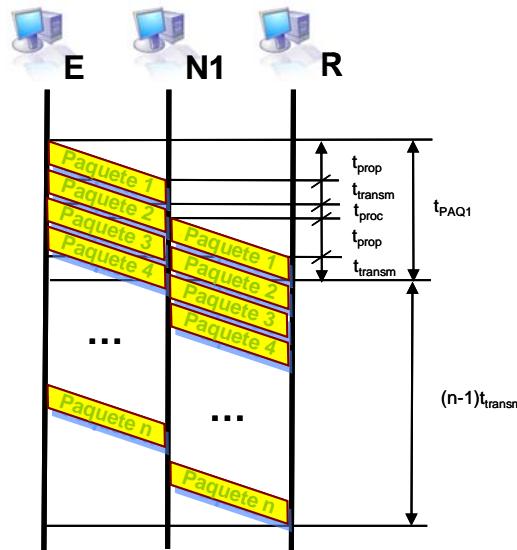
4. Estime el tiempo involucrado en la transmisión de un mensaje de datos para la técnica de conmutación de paquetes mediante datagramas (CDP) considerando los siguientes parámetros:

M: longitud en bits del mensaje a enviar.  
V: velocidad de transmisión de las líneas en bps.  
P: longitud en bits de los paquetes.  
H: bits de cabecera de los paquetes.  
N: número de nodos intermedios entre las estaciones finales.  
D: tiempo de procesamiento en segundos en cada nodo.  
R: retardo de propagación, en segundos, asociado a cada enlace.

Ejercicio resuelto en "Transmisión de Datos y Redes de Computadores", P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler, Pearson, 2007, página 164. Ver únicamente la solución de CDP.

5. Un mensaje de 64 kB se transmite a lo largo de dos saltos de una red. Ésta limita la longitud máxima de los paquetes a 2 kB y cada paquete tiene una cabecera de 32 bytes. Las líneas de transmisión de la red no presentan errores y tienen una capacidad de 50 Mbps. Cada salto corresponde a una distancia de 1000 km. ¿Qué tiempo se emplea en la transmisión del mensaje mediante datagramas?

Sean E el emisor, N1 el nodo intermedio y R el receptor. En el caso de enviar los paquetes mediante datagramas, se procederá al envío de todos los paquetes sin ningún establecimiento previo de conexión ni ninguna operación adicional. De esta forma, se enviarán consecutivamente todos los paquetes a partir del instante inicial, tal como se muestra en la figura.



Por tanto, podemos descomponer el tiempo empleado en el envío del mensaje,  $t_{mensaje}^{dat}$ , en dos contribuciones: el tiempo empleado por el primer paquete en alcanzar el destino,  $t_{PAQ1}$ , más el tiempo empleado en la transmisión de los restantes paquetes. Así, suponiendo que el mensaje se descompone en n paquetes, tendremos:

$$t_{mensaje}^{dat} = t_{PAQ1} + (n-1)t_{transm}$$



Universidad de Granada

## Fundamentos de Redes

### 3º del Grado en Ingeniería Informática



Dept. Teoría de la Señal,  
Telemática y Comunicaciones

donde  $t_{transm}$  corresponde al tiempo empleado en transmitir un paquete. A su vez, el tiempo empleado por el primer paquete será la suma del tiempo en llegar al primer nodo,  $t_{PAQ1}^{E \rightarrow N1}$ , más el tiempo en alcanzar el destino desde el primer nodo,  $t_{PAQ1}^{N1 \rightarrow R}$ , que puede ser expresado en función del tiempo de transmisión, el de propagación sobre los enlaces,  $t_{prop}$ , y el retardo de procesamiento,  $t_{proc}$ , de acuerdo a:

$$t_{PAQ1} = t_{PAQ1}^{E \rightarrow N1} + t_{PAQ1}^{N1 \rightarrow R} = t_{transm}^E + t_{prop}^{E \rightarrow N1} + t_{proc}^{N1} + t_{transm}^{N1} + t_{prop}^{N1 \rightarrow R}$$

donde los superíndices indican los sistemas implicados en cada caso.

Teniendo en cuenta que ambos enlaces tienen la misma longitud y que las velocidades de transmisión son idénticas, podemos agrupar términos de la forma,

$$t_{PAQ1} = t_{PAQ1}^{E \rightarrow N1} + t_{PAQ1}^{N1 \rightarrow R} = 2(t_{transm} + t_{prop}) + t_{proc}^{N1}$$

Por tanto, el tiempo total será

$$t_{mensaje}^{dat} = 2(t_{transm} + t_{prop}) + t_{proc}^{N1} + (n-1)t_{transm}$$

Podemos obtener los valores de los tiempos de propagación y transmisión a partir de los datos suministrados, sin más que considerar la velocidad de la luz en un medio guiado, de la forma

$$t_{prop} = \frac{d}{v_{prop}} = \frac{10^6 \text{ m}}{2 \cdot 10^8 \text{ m/s}} = 0,005 \text{ s}$$

$$t_{transm} = \frac{L}{R} = \frac{2 \text{ kB} \cdot 1024 \text{ B/kB} \cdot 8 \text{ b/B}}{50 \cdot 10^6 \text{ bps}} = 0,00033 \text{ s}$$

Por otra parte, el número de paquetes vendrá determinado por la longitud del mensaje a transmitir y el tamaño de la carga útil de cada paquete. De acuerdo al enunciado, la longitud de cada paquete es  $L=2 \text{ kB}$  con una cabecera de 32 bytes, por lo que, para enviar un mensaje de  $M=64 \text{ kB}$  se necesitarán

$$n = \left\lceil \frac{M}{L_{util}} \right\rceil = \left\lceil \frac{64 \text{ kB} \cdot 1024 \text{ B/kB}}{2 \text{ kB/paq} \cdot 1024 \text{ B/kB} - 32 \text{ B/paq}} \right\rceil = 33 \text{ paq}$$

donde hemos considerado que el último paquete se rellena hasta completar el tamaño indicado.

Finalmente, si consideramos el tiempo de procesamiento despreciable, obtendremos un tiempo total

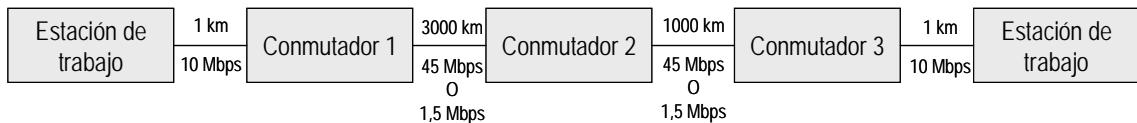
$$t_{mensaje}^{dat} = 2(5 + 0,33) + 32 \cdot 0,33 \text{ ms} = 21,22 \text{ ms}$$

6. Una aplicación audiovisual en tiempo real hace uso de commutación de paquetes para transmitir voz a 32 kbps y vídeo a 64 kbps a través de la conexión de red de la figura. Se consideran paquetes de voz e información de audio con dos longitudes distintas: 10 ms y 100 ms. Cada paquete tiene además una cabecera de 40 octetos.
  - a. Encuentre para ambos casos el porcentaje de bits suplementarios que supone la cabecera.
  - b. Dibuje un diagrama temporal e identifique todas las componentes del retardo extremo a extremo en la conexión anterior. Recuerde que un paquete no puede ser transmitido hasta que esté completo y que no se puede retransmitir hasta que no se haya recibido completamente. Suponga despreciables los errores a nivel de bit.
  - c. Evalúe todas las componentes del retardo de las que se dispone suficiente



información. Considere las dos longitudes de paquete aceptadas. Suponga que la señal se propaga a una velocidad de 1 km/5 microsegundos y considere dos velocidades para la red troncal: 45 Mbps y 1,5 Mbps. Resuma el resultado para los cuatro posibles casos en una tabla con cuatro entradas.

- d. ¿Cuál de las componentes anteriores implica la existencia de retardos de cola?



- a) Para obtener el porcentaje de bits suplementarios hemos de obtener, en primer lugar, el número de bits de datos en cada paquete. Así, para voz a 32 kbps con 10 ms de duración de paquetes, tendremos:

$$L_{voz} = tR = 10 \cdot 10^{-3} s \cdot 32 \cdot 10^3 \text{ kbps} \Rightarrow L_{voz} = 320 \text{ bits}$$

En el caso de vídeo,

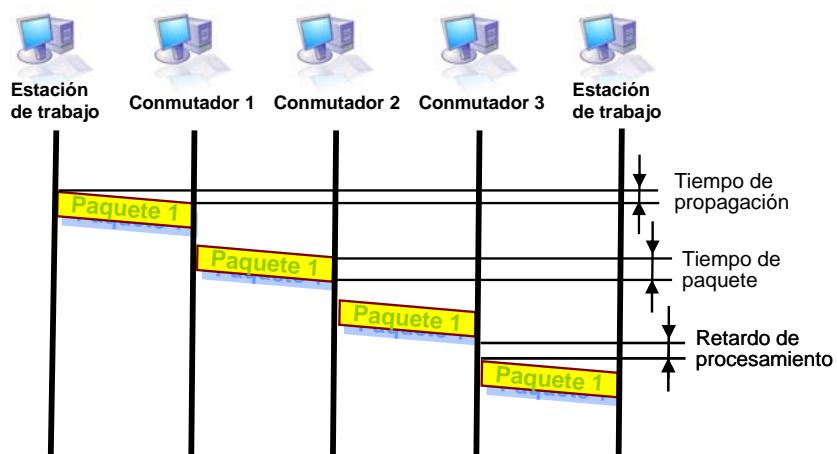
$$L_{video} = tR = 100 \cdot 10^{-3} s \cdot 64 \cdot 10^3 \text{ kbps} \Rightarrow L_{video} = 6400 \text{ bits}$$

Por lo que, dado que hay que añadir 40 octetos de cabecera, los porcentajes de bits suplementarios son:

$$\%_{video}^{sup} = \frac{40 \cdot 8}{6400 + 40 \cdot 8} \cdot 100 \Rightarrow \%_{video}^{sup} = 4,76 \%$$

$$\%_{voz}^{sup} = \frac{40 \cdot 8}{320 + 320} \cdot 100 \Rightarrow \%_{voz}^{sup} = 50 \%$$

- b) La transmisión de un paquete corresponderá, gráficamente, con la figura



en la que se han identificado los tres retardos implicados: el tiempo de propagación en cada enlace, que corresponderá con el tiempo empleado por la señal para propagarse de un extremo a otro de cada uno de los enlaces, el tiempo de paquete, necesario para que cada comutador reenvíe los paquetes tras recibirlos y procesarlos, y el tiempo de procesamiento, que corresponde al tiempo que los paquetes deben permanecer en la cola más el tiempo empleado en el encaminamiento y conmutación del paquete.

- c) De los tres retardos identificados, únicamente podemos evaluar el tiempo de



propagación en cada enlace y el tiempo de paquete, ya que no se proporcionan datos que permitan establecer el retardo de procesamiento. Además, este último se suele despreciar al ser comúnmente varios órdenes de magnitud menor que los otros.

Los tiempos de paquete se pueden obtener a partir de la longitud de los mismos y de las velocidades de los enlaces.

$$t_{paquete} = \frac{L_{paquete}}{R_{enlace}}$$

Aplicándolo a los enlaces y tamaños de paquete disponibles, obtenemos

$$\text{Enlace 1: } t_{paquete}^{voz,enlace1} = \frac{640 \text{ bits}}{10^7 \text{ bps}} = 0,064 \text{ ms} \quad t_{paquete}^{video,enlace1} = \frac{6720 \text{ bits}}{10^7 \text{ bps}} = 0,672 \text{ ms}$$

$$\text{Enlace 2(a): } t_{paquete}^{voz,enlace2A} = \frac{640 \text{ bits}}{45 \cdot 10^6 \text{ bps}} = 0,014 \text{ ms} \quad t_{paquete}^{video,enlace2A} = \frac{6720 \text{ bits}}{45 \cdot 10^6 \text{ bps}} = 0,149 \text{ ms}$$

$$\text{Enlace 2(b): } t_{paquete}^{voz,enlace2B} = \frac{640 \text{ bits}}{1,5 \cdot 10^6 \text{ bps}} = 0,427 \text{ ms} \quad t_{paquete}^{video,enlace2B} = \frac{6720 \text{ bits}}{1,5 \cdot 10^6 \text{ bps}} = 4,48 \text{ ms}$$

El enlace 3 presenta los mismos tiempos de paquete que el 2.

El enlace 4 presenta los mismos tiempos de paquete que el 1.

Los tiempos de propagación dependen de la longitud de los enlaces y de la velocidad de propagación:

$$t_{prop} = \frac{d_{enlace}}{v_{prop}} = \frac{d_{enlace}}{\cancel{1 \text{ km}} / \cancel{5 \mu\text{s}}} = \frac{d_{enlace}}{2 \cdot 10^8 \text{ m/s}}$$

Por tanto, los tiempos serán:

$$\text{Enlace 1: } t_{prop}^{enlace1} = \frac{d_{enlace1}}{2 \cdot 10^8 \text{ m/s}} = \frac{1000 \text{ m}}{2 \cdot 10^8 \text{ m/s}} \Rightarrow t_{prop}^{enlace1} = 5 \mu\text{s}$$

$$\text{Enlace 2: } t_{prop}^{enlace2} = \frac{d_{enlace2}}{2 \cdot 10^8 \text{ m/s}} = \frac{3 \cdot 10^6 \text{ m}}{2 \cdot 10^8 \text{ m/s}} \Rightarrow t_{prop}^{enlace2} = 15 \text{ ms}$$

$$\text{Enlace 3: } t_{prop}^{enlace3} = \frac{d_{enlace3}}{2 \cdot 10^8 \text{ m/s}} = \frac{10^6 \text{ m}}{2 \cdot 10^8 \text{ m/s}} \Rightarrow t_{prop}^{enlace3} = 5 \text{ ms}$$

El enlace 4 presenta el mismo tiempo de propagación que el 1.

El tiempo de propagación total es:

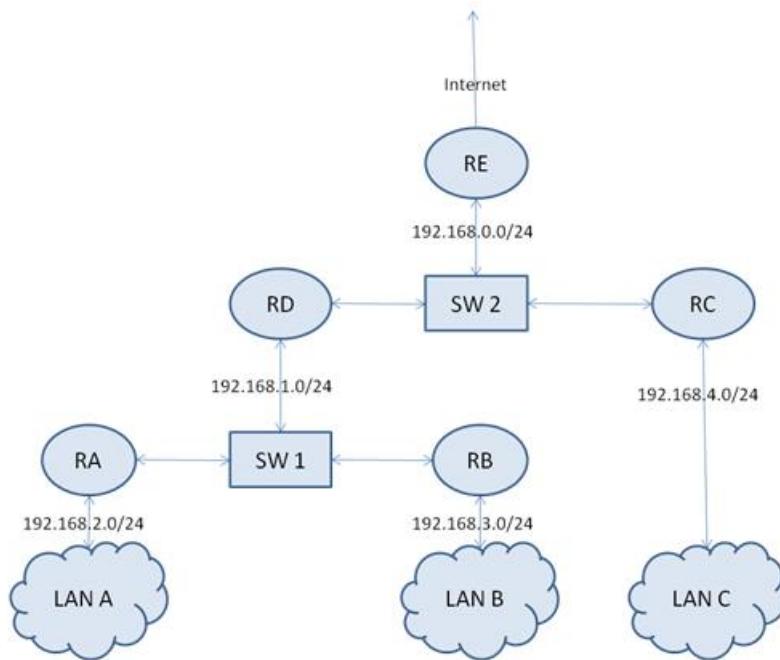
$$t_{prop}^{TOTAL} = t_{prop}^{enlace1} + t_{prop}^{enlace2} + t_{prop}^{enlace3} + t_{prop}^{enlace4} \Rightarrow t_{prop}^{TOTAL} = 20,01 \text{ ms}$$

Por tanto, los tiempos totales para las 4 posibilidades (voz-video, 1,5-45 Mbps) son:

|                  | Voz      | Vídeo    |
|------------------|----------|----------|
| Enlaces 1,5 Mbps | 20.98 ms | 30.3 ms  |
| Enlaces 45 Mbps  | 20.16 ms | 21.64 ms |

- d) Como se ha mencionado anteriormente, la componente que implica retardos en cola es el tiempo de procesamiento, que es la suma del tiempo en la cola y el tiempo empleado en el encaminamiento de los paquetes.

7. Imagine una situación donde hay cinco routers RA-RE. RA, RB y RC se conectan cada uno a una red local A, B y C, siendo cada router única puerta de enlace de cada red. RA, RB y RD están conectados entre sí a través de un switch. RC, RD y RE están conectados entre sí a través de un switch. RE conecta a Internet a través de la puerta de acceso especificada por el ISP. Especifique tablas de encaminamiento en los routers. Asigne a voluntad las direcciones IP e interfaces necesarias.



En primer lugar, para especificar una solución es necesario asignar direcciones IP en interfaces a los routers de la topología. Para ser un poco sistemáticos, haremos la siguiente asignación.

- Interfaces: Llamaremos Eth0 a la interfaz del enlace que sale de cada router hacia abajo, y Eth1 a la otra.
- Direcciones: En cada subred, cogeremos la dirección de host .1 para el router que se conecta al switch desde arriba, la .2 desde la izquierda y la .3 desde la derecha. Asumiremos que la IP pública del router de acceso es la 80.5.5.0/30.

Con estas asignaciones, las tablas de encaminamiento serán las siguientes. Para construir cada una de ellas, seguimos las recomendaciones de la teoría:

- Primero: añadir las subredes directamente conectadas
- Segundo: añadir la orden por defecto en la dirección con mayor número de IPs.
- Tercero: añadir las redes que faltan.

Donde sea posible, usaremos compresión (summarization) de rutas, de forma que el número de entradas en la tabla de encaminamiento sea mínimo.

**Tabla RA:**

| IP destino  | Máscara       | Siguiente Nodo   | Interfaz |
|-------------|---------------|------------------|----------|
| 192.168.2.0 | 255.255.255.0 | -                | Eth0     |
| 192.168.1.0 | 255.255.255.0 | -                | Eth1     |
| 0.0.0.0     | 0.0.0.0       | 192.168.1.1 (RD) | Eth1     |
| 192.168.3.0 | 255.255.255.0 | 192.168.1.3 (RB) | Eth1     |

**Tabla RB:**

| IP destino  | Máscara       | Siguiente Nodo   | Interfaz |
|-------------|---------------|------------------|----------|
| 192.168.3.0 | 255.255.255.0 | -                | Eth0     |
| 192.168.1.0 | 255.255.255.0 | -                | Eth1     |
| 0.0.0.0     | 0.0.0.0       | 192.168.1.1 (RD) | Eth1     |
| 192.168.2.0 | 255.255.255.0 | 192.168.1.2 (RA) | Eth1     |

**Tabla RC:**

| IP destino  | Máscara              | Siguiente Nodo   | Interfaz |
|-------------|----------------------|------------------|----------|
| 192.168.4.0 | 255.255.255.0        | -                | Eth0     |
| 192.168.0.0 | 255.255.255.0        | -                | Eth1     |
| 0.0.0.0     | 0.0.0.0              | 192.168.0.1 (RE) | Eth1     |
| 192.168.0.0 | <b>255.255.252.0</b> | 192.168.0.2 (RD) | Eth1     |

**Tabla RD:**

| IP destino  | Máscara       | Siguiente Nodo   |      |
|-------------|---------------|------------------|------|
| 192.168.0.0 | 255.255.255.0 | -                | Eth1 |
| 192.168.1.0 | 255.255.255.0 | -                | Eth0 |
| 0.0.0.0     | 0.0.0.0       | 192.168.0.1 (RE) | Eth1 |
| 192.168.2.0 | 255.255.255.0 | 192.168.1.2 (RA) | Eth0 |
| 192.168.3.0 | 255.255.255.0 | 192.168.1.3 (RB) | Eth0 |
| 192.168.4.0 | 255.255.255.0 | 192.168.0.3 (RC) | Eth1 |

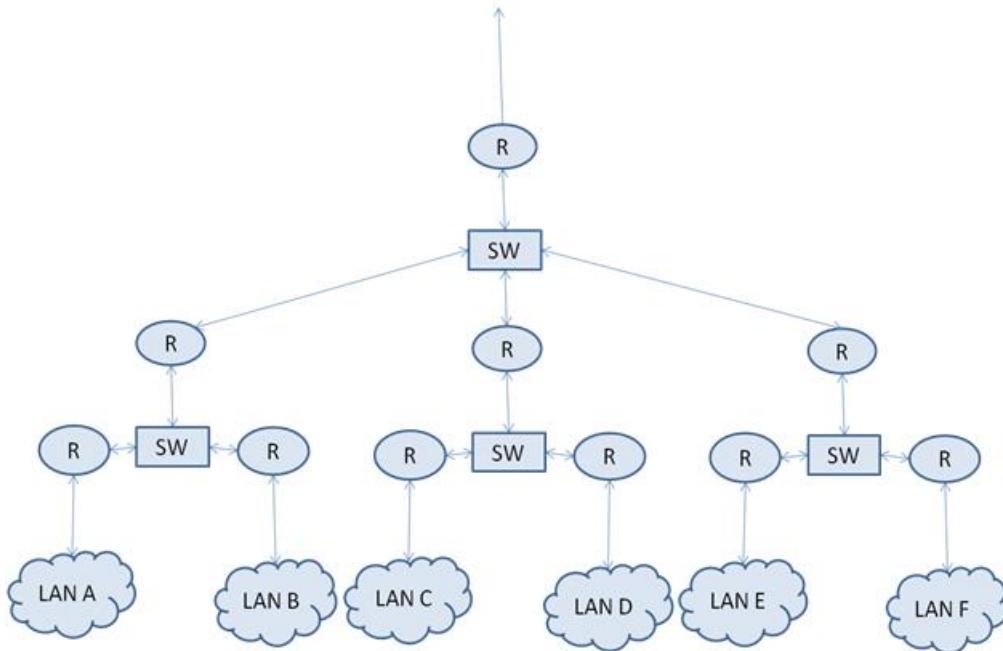
**Tabla RE:**

| IP destino  | Máscara              | Siguiente Nodo   |      |
|-------------|----------------------|------------------|------|
| 192.168.0.0 | 255.255.255.0        | -                | Eth0 |
| 80.5.5.5.0  | 255.255.255.252      | -                | Eth1 |
| 0.0.0.0     | 0.0.0.0              | 80.5.5.5.1       | Eth1 |
| 192.168.0.0 | <b>255.255.252.0</b> | 192.168.0.2 (RD) | Eth0 |
| 192.168.4.0 | 255.255.255.0        | 192.168.0.3 (RC) | Eth0 |

Donde en las tablas de RC y RE, en negrita, se señala la compresión de las tres rutas bajo el router RD.



8. Asigne las direcciones de subred en la siguiente topología a partir de 192.168.0.0 para minimizar el número de entradas en las tablas de encaminamiento, asumiendo que en las redes LAN puede haber hasta 50 PCs.

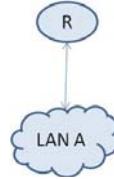


Para solucionar este ejercicio, la minimización de las entradas en las tablas de encaminamiento pasa por un adecuado diseño jerárquico de las direcciones de subred. Aquí se plantea una forma sistemática de resolución, que no es la única posible. Para ello, realizamos cuatro pasos:

- a) Establecer las máscaras de subred con la información suministrada.
  - b) Establecer las direcciones de subred.
  - c) Establecer todas las direcciones de dispositivo necesarias.
  - d) Obtener las tablas de encaminamiento.
- a) Para establecer las máscaras de subred, recorremos la topología de abajo hacia arriba parándonos en cada nodo de interconexión IP (cada router) para establecer la necesidad de direcciones y a partir de ahí la máscara asociada.
- i. Empezamos fijándonos en los routers de acceso de cada red LAN. Para cada LAN debajo del router de acceso necesitamos un total de 50 (PCs) + 1 (router) + 2 (subred y broadcast) direcciones IP, en total 53 direcciones. La potencia de 2 igual o inmediatamente superior es 64, que requiere de 6 bits, o lo que es lo mismo una máscara de red de  $32-6=26$  unos  $\Rightarrow /26$ . Estamos asumiendo que 50 PCs es lo máximo que va a tener cada LAN, incluyendo las posibles futuras incorporaciones. En el caso que se hablara de un crecimiento potencial de la red, se podría pensar en mayor cantidad de bits para PCs en la subred.

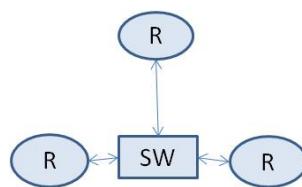


/26 ➔



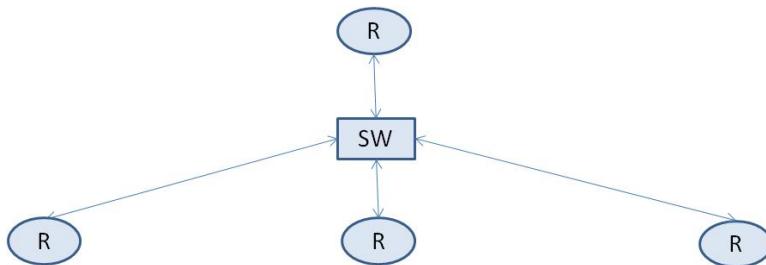
- ii. Seguimos por los routers en segundo nivel de jerarquía, que se interconectan cada uno a otros dos routers con una LAN asociada. En total, cada router deja por debajo tres subredes (dos LAN más la de interconexión), para las que como máximo se necesitan 6 bits de direcciones de acuerdo al punto anterior. La potencia de 2 igual o inmediatamente superior a 3 (subredes) es 4, que requiere de 2 bits más los 6 bits de direccionamiento en cada subred. Por tanto, la máscara de red es de  $32-8=24$  unos ➔ /24.

/24 ➔



- iii. El último nivel de jerarquía es el router de acceso de toda la red. En total, el router deja por debajo cuatro subredes /24. La potencia de 2 igual a 4 (subredes) es 4, que requiere de 2 bits más los 8 bits de direccionamiento en cada subred. Por tanto, la máscara de red es de  $32-10=22$  unos ➔ /22.

/22 ➔



- b) La asignación de subredes puede llevarse a cabo de la siguiente forma. Empezamos de arriba hasta abajo<sup>1</sup>.
- Empezamos asignando desde el principio la subred principal:
    - 192.168.0.0/22
  - El siguiente punto es asignar las tres subredes siguientes. Para ello, consideraremos que son /24 y que el primer espacio (192.168.0.0/24) servirá para las direcciones de la subred de conexión de i.:
    - 192.168.1.0/24

<sup>1</sup> En clase y en la resolución se ha visto una asignación de direcciones posible, empezando desde abajo hasta arriba. Por mostrar otra posibilidad, empezaremos en esta resolución desde arriba hasta abajo.

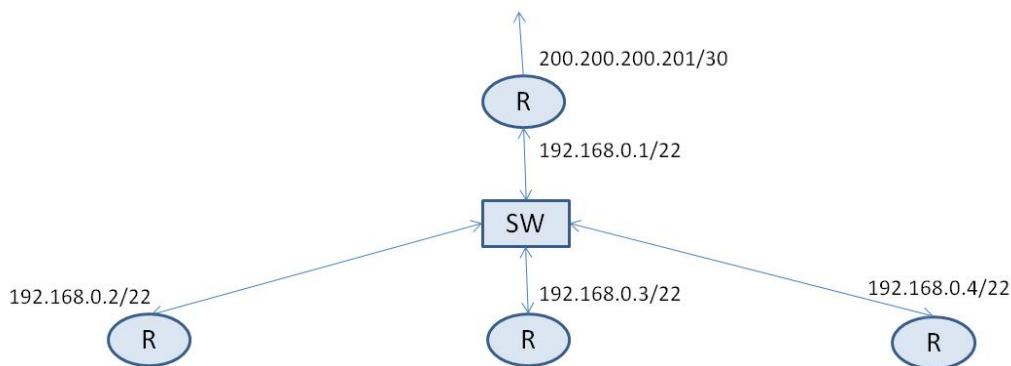
- b. 192.168.2.0/24
- c. 192.168.3.0/24

iii. Para cada una de las tres subredes, hemos de diseñar las subredes /26. De nuevo, volvemos a dejar el primer espacio /26 para las subredes de interconexión del punto anterior.

- a. LAN A: 192.168.1.64/26
- b. LAN B: 192.168.1.128/26
- c. LAN C: 192.168.2.64/26
- d. LAN D: 192.168.2.128/26
- e. LAN E: 192.168.3.64/26
- f. LAN F: 192.168.3.128/26

- c) Para evitar extender mucho esta explicación, haremos la asignación de hosts (punto c)) y la tabla de encaminamiento (punto d)) sólo en el router superior, el de acceso a Internet. En primer lugar, necesitamos establecer una subred pública de acceso a Internet, con direcciones públicas. Por ejemplo, la 200.200.200.200/30. Así, el router de acceso puede tener la dirección pública 200.200.200.201 y el del lado del ISP la 200.200.200.202.

Por otro lado, en el apartado b.i y b.ii hemos decidido que la subred ocuparía la primera porción de 192.168.0.0/22, por tanto una posible asignación nos queda:



- d) Con esta asignación, la tabla de encaminamiento en el router de acceso queda como sigue. Primero añadimos las redes directamente conectadas (2), luego la opción por defecto, hacia Internet, y finalmente todo lo que nos quede (3 subredes más) Por simplicidad, no se han añadido interfaces de red, ya que además el enunciado no lo pedía, y se ha utilizado la nomenclatura /n para las máscaras:

| IP destino      | Máscara | Siguiente Nodo  |
|-----------------|---------|-----------------|
| 200.200.200.200 | /30     | -               |
| 192.168.0.0     | /22     | -               |
| 0.0.0.0         | /0      | 200.200.200.202 |
| 192.168.1.0     | /24     | 192.168.0.2     |
| 192.168.2.0     | /24     | 192.168.0.3     |
| 192.168.3.0     | /24     | 192.168.0.4     |

9. Un datagrama de 4020 bytes pasa de una red Token Ring con THT 8 ms (MTU 4400) a una Ethernet (MTU 1500) y después pasa por un enlace PPP con bajo retardo (MTU 296). Si ese mismo datagrama pasara directamente de la red Token Ring al enlace PPP (sin pasar por la red Ethernet) ¿habría alguna diferencia en la forma como se produce la fragmentación? Especifique en ambos casos los fragmentos obtenidos.

Considerando datagramas IP sin opciones, tenemos un total de 20B de cabecera, el datagrama tiene un total de 4000B de datos en capa de red (que incluyen capa de transporte y aplicación). La red Token Ring no afecta al datagrama, ya que su MTU es superior al tamaño total del datagrama. Al pasar a la red Ethernet, el datagrama se fragmenta en el siguiente número de fragmentos:

$$N_f = \left\lceil \frac{4020 - 20}{1500 - 20} \right\rceil = 3$$

de forma que tendremos dos fragmentos de 1500 bytes más otro de:

$$F_3 = 4020 - 20 - 2(1500 - 20) + 20 = 1060$$

A su vez, cada fragmento de 1500 generará los siguientes fragmentos en la red PPP:

$$N_f = \left\lceil \frac{1500 - 20}{296 - 20} \right\rceil = 6$$

de forma que tendremos 5 fragmentos de 296B más uno de:

$$F_6 = 1500 - 20 - 5(296 - 20) + 20 = 120$$

El fragmento Ethernet de 1060 generará los siguientes fragmentos en la red PPP:

$$N_f = \left\lceil \frac{1060 - 20}{296 - 20} \right\rceil = 4$$

de forma que tendremos 3 fragmentos de 296B más uno de:

$$F_4 = 1060 - 20 - 3(296 - 20) + 20 = 232$$

En conclusión, tras pasar el datagrama de 4020B por la red Ethernet y luego por la PPP, se generan 13 fragmentos de 296B más 2 fragmentos de 120B y un fragmento de 232B. Podemos comprobar fácilmente que el resultado es correcto:

$$13 \cdot (296 - 20) + 2 \cdot (120 - 20) + (232 - 20) + 20 = 4020$$

Como vemos, hemos aumentado 16 veces el tamaño de bytes de cabeceras, al generar 16 datagramas a partir de 1 sólo.

Todos los fragmentos tendrán el mismo identificativo (el del datagrama original) y el flag MF a 1, excepto el último que lo tendrá a 0. El offset aparece en la siguiente tabla.

| # Frag. | offset |
|---------|--------|---------|--------|---------|--------|---------|--------|
| 1       | 0      | 5       | 1104   | 9       | 2032   | 13      | 2960   |
| 2       | 276    | 6       | 1380   | 10      | 2308   | 14      | 3236   |
| 3       | 552    | 7       | 1480   | 11      | 2584   | 15      | 3512   |
| 4       | 828    | 8       | 1756   | 12      | 2860   | 16      | 3788   |

Por otro lado, si consideramos el paso directo desde la red Token Ring hasta la PPP, el número de fragmentos resultante es:

$$N_f = \left\lceil \frac{4020 - 20}{296 - 20} \right\rceil = 15$$

de forma que tendremos 14 fragmentos de 296B más uno de:

$$F_{15} = 4020 - 20 - 14(296 - 20) + 20 = 156$$

De nuevo, podemos comprobar que el resultado es correcto:

$$14(296 - 20) + (156 - 20) + 20 = 4020$$

Como vemos, hemos aumentado en 15 veces el tamaño de bytes original de cabeceras, un fragmento menos que en el primer caso.

Todos los fragmentos tendrán el mismo identificativo (el del datagrama original) y el flag MF a 1, excepto el último que lo tendrá a 0. El offset aparece en la siguiente tabla.

| # Frag. | offset |
|---------|--------|---------|--------|---------|--------|---------|--------|
| 1       | 0      | 5       | 1104   | 9       | 2208   | 13      | 3312   |
| 2       | 276    | 6       | 1380   | 10      | 2484   | 14      | 3588   |
| 3       | 552    | 7       | 1656   | 11      | 2760   | 15      | 3864   |
| 4       | 828    | 8       | 1932   | 12      | 3036   |         |        |

- 10. ¿Cómo podría utilizar ICMP para hacer una estimación de la latencia entre dos entidades finales? ¿Y para estimar la latencia de un enlace en particular entre dos routers?**

En ambos casos, utilizando mensajes de solicitud/respuesta de sello de tiempo ICMP. En el primer caso, se puede utilizar el programa *ping*, como se explica en la primera relación. En el segundo, se podría utilizar *traceroute* desde un origen hasta un destino que incluya el enlace entre los routers.

## **2.3. Relación 3**



## Ejercicios – Tema 3

1. Suponga que el cliente A inicia una sesión Telnet con el servidor S. Aproximadamente en el mismo instante, el cliente B inicia otra conexión con el servidor.
  - a. Proporcione los posibles números de puerto origen y destino para los segmentos enviados de A a S.  
Puerto origen ∈ [1024,65535]  
Puerto destino=23,
  - b. Proporcione los posibles números de puerto origen y destino para los segmentos enviados de S a B.  
Puerto origen=23,  
Puerto destino ∈ [1024,65535]
  - c. Si A y B son hosts diferentes, ¿es posible que el número de puerto origen en los segmentos que van de A a S sea el mismo que en los segmentos de B a S?  
Sí.
  - d. ¿Y si A y B son el mismo host?  
No.
2. ¿Por qué TCP y UDP usan los números de puerto para identificar las entidades de transporte en lugar de usar el identificador del proceso?

El identificador de proceso podría usarse cuando las dos entidades de transporte que se comunican están en el mismo host. En el caso general, las entidades de transporte se encuentran en hosts diferentes, entonces no es posible conocer el identificador de proceso de la entidad contraria, y por lo tanto, no puede iniciarse la comunicación. Por ejemplo, no se puede conocer el identificador del proceso encargado de un servicio web, que además cambia cada vez que se reinicia el servidor. Para solucionar este problema, se usan los puertos bien conocidos (en el ejemplo del puerto 80)

3. Un proceso en un host 1 se le asigna el puerto p, y a un proceso en un host 2 se le asigna el puerto q. ¿Sería posible establecer 2 o más conexiones TCP simultáneas entre estos puertos?

No, únicamente podría darse una conexión. En primer lugar, al menos uno de los procesos debe iniciar la conexión mandando un paquete SYN. De acuerdo al diagrama de estados de TCP, en ese momento su puerto pasaría al estado SYN\_SENT y se podría establecer la primera conexión siguiendo el saludo en tres pasos (three-way handshake) Una vez establecida, el puerto del iniciador no aceptaría nuevas conexiones. La única posibilidad alternativa sería el envío simultáneo de mensajes SYN por parte de los procesos, que se resolvería en una sola conexión, como se discute en el ejercicio 6.



Universidad de Granada

Fundamentos de Redes

3º del Grado en Ingeniería  
Informática



Dept. Teoría de la Señal,  
Telemática y Comunicaciones

4. Además del campo de 32 bits para el acuse de recibo en los segmentos TCP hay un bit o flag ACK. ¿Realmente este bit sirve para algo? Justificar la respuesta.

Sí, el bit ACK sirve para diferenciar los segmentos que llevan confirmaciones de los que no las llevan. Por tanto, si el flag ACK no está activo, el campo de acuse de recibo no debe tenerse en cuenta.

5. Calcule la suma de comprobación en UDP y TCP de las siguientes palabras de 8 bits (observe que aunque UDP y TCP utilicen palabras de 16 bits, en este ejercicio se pide el mismo cálculo sobre palabras de 8 bits): 01010011, 01010100, 01110100.

La suma en complemento a uno de se realiza de dos en dos palabras, y al derivarse un acarreo se suma adicionalmente:

$$01010011 + 01010100 = 10100111$$

$$10100111 + 01110100 = 00011011 + 1(\text{acarreo}) = 00011100$$

El complemento a uno de 00011100 es 11100011.

- a. ¿Por qué UDP/TCP utilizan el complemento a uno de la suma complemento a uno, en lugar de directamente la suma en complemento a uno?

El checksum calculado es añadido al paquete. Cuando se re-calcula en el receptor, al incluir el complemento a uno del checksum del resto de las palabras, el resultado final es una palabra todo a 1s. Si ése no es el resultado, se detecta un error. Esta estrategia es más rápida computacionalmente que incluir una comparación, que sería lo necesario en el caso de no usar el complemento a 1.

- b. ¿cómo detecta el receptor los errores?

Si obtiene algún 0 en el checksum del receptor.

- c. ¿se detectan todos los errores de 1 bit?

Sí, todos los errores de 1 sólo bit generarían un 0 en el checksum del receptor, por lo que serían detectados.

- d. ¿se detectan todos los errores que afectan simultáneamente a 2 bits?

No todos. Por ejemplo, si variamos el mismo bit en la 1<sup>a</sup> y 2<sup>a</sup> palabras, coincidiendo que estos bits tienen valores distintos, este error no se detecta ya que su suma es la misma.

Ejemplo:

$$01010011 + 01010100 + 01110100 = 00011100$$

$$01010010 + 01010101 + 01110100 = 00011100 \text{ ¡Error en dos bits no detectado!}$$



6. Suponga un establecimiento de conexión TCP en el que ambos host mandan un mensaje SYN casi simultáneo. ¿Sería posible un bloqueo del sistema? Justificar la respuesta con un ejemplo.

No sería posible. El Establecimiento de Conexión TCP está diseñado también para el caso en el que los dos host realizan una apertura activa. En este caso, ambos hosts pasarían iterativamente por los estados:

CLOSED → SYN\_SENT → SYN\_RCVD → ESTABLISHED.

En todo caso, ante la incidencia de una posible desincronización de la conexión debido a la latencia variable que experimentan los paquetes en Internet, se mandarían paquetes RESET, para reiniciar la conexión.

7. Describir dos formas de llegar al estado SYN RCVD para TCP.

CLOSED → LISTEN → SYN\_RCVD

CLOSED → SYN\_SENT → SYN\_RCVD

8. Comparando UDP con TCP:

- a. ¿Con qué protocolo de transporte tiene una aplicación más control sobre qué datos se envían en un segmento/datagrama?

A nivel general en UDP, ya que al no implementar un servicio de conexión fiable le da mayor flexibilidad a la capa de aplicación para decidir el envío de los datos.

- b. ¿Con qué protocolo de transporte tiene una aplicación más control sobre cuándo se envía un segmento/datagrama?

Igual que a)

9. Se desea transferir con protocolo TCP un archivo de L bytes usando un MSS de 536.

- a. ¿Cuál es el valor máximo de L tal que los números de secuencia de TCP no se agoten?

Los números de secuencia, y por tanto de acuse de recibo, se agotan tras  $2^{32}$  bytes, ya que cada número está asociado a un byte de la capa de aplicación. Si consideramos además los 2 números de secuencia consumidos correspondientes al SYN inicial y al FIN final, el número máximo de L es  $2^{32} - 2$  bytes.

- b. Considerando una velocidad de transmisión de 155 Mbps y un total de 66 bytes para las cabeceras de las capas de transporte, red y enlace de datos, e ignorando limitaciones debidas al control de flujo y congestión, calcule el tiempo que se tarda en transmitir el archivo en A.



En primer lugar, es necesario calcular el número de segmentos necesarios para la transmisión del fichero completo:

$$n_{seg} = \left\lceil \frac{L}{MSS} \right\rceil \text{ seg.} = 8012999 \text{ seg.}$$

donde  $\lceil A \rceil$  significa el número entero inmediatamente superior a A.

A partir de este número, se calcula el número de bits transmitidos, que incluye para cada segmento la parte de datos (MSS Bytes) y las cabeceras (66 B):

$$n_{bit} = \left\lceil \frac{L}{MSS} \right\rceil \times (66 + MSS) \times 8 \text{ bits}$$

En este caso, se está asumiendo que todos los segmentos tienen el mismo tamaño de datos, cuando en realidad el último segmento puede ser menor. No obstante, considerando que el número de segmentos es tan grande, las implicaciones en el resultado final serían despreciables. En todo caso siempre se puede calcular el tamaño de datos del último segmento a partir de L y el número de segmentos (366 Bytes).

Por último, el tiempo de transmisión (sin incluir inicio y cierre de conexión) es:

$$t_{transmisión \ del \ fichero} = \frac{\#bits \ transmitidos}{velocidad \ de \ transmisión} \approx 249 \ sg.$$

- 10. Los hosts A y B se están comunicando a través de una conexión TCP y B ya ha recibido y confirmado todos los bytes hasta el byte 126. Suponga que a continuación el host A envía dos segmentos seguidos a B que contienen, respectivamente, 70 y 50 bytes de datos. El envío de A es ordenado, el número de puerto origen en dichos segmentos es 302 y el de destino el 80. El host B envía una confirmación inmediata a la recepción de cada segmento de A, sin esperar el retardo de 500 ms del estándar.**

- Especifique los números de secuencia de ambos segmentos.**  
127 y 197
- Si el primer segmento llega antes que el segundo ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que se envía?**

Confirmación del primer segmento:

- número de acuse: 197
- puerto origen: 80
- puerto destino: 302

- Si el segundo segmento llega antes que el primero ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que envía?**



Se produce una recepción fuera de orden, por lo tanto se envía una confirmación duplicada (RFC 1122, 2581) con las siguientes características:

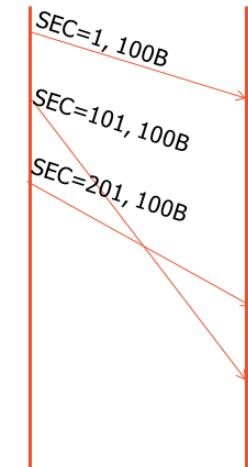
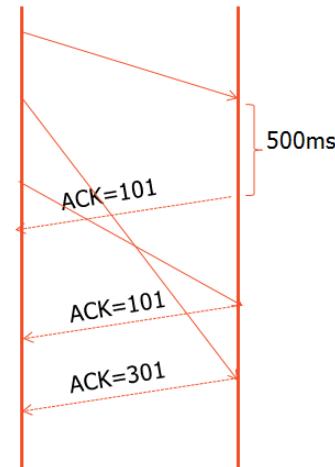
- número de acuse=127
- puerto origen=80
- puerto destino=302

d. **Imagine que los segmentos llegan en orden pero se pierde el primer ACK.**

Si se pierde el primer ACK no pasa nada, pues TCP utiliza confirmaciones acumulativas, es decir, la recepción del ACK del segundo segmento sirve para confirmar también el primer segmento. Si llegara a expirar el temporizador del primer segmento y éste fuera retransmitido, sería descartado en el receptor que ya guarda una copia válida de este segmento.

**11. Considere el diagrama de tiempo de la Figura 1 donde se ilustra el envío y recepción de tres segmentos de 100 bytes de datos desde el Host A al Host B. Especifique las confirmaciones que realizaría el Host B. Vuelva a repetir el ejercicio considerando que estos son los únicos paquetes de datos en la conexión y que A calcula ISN=40210.**

Las confirmaciones se muestran en la Figura 2. La primera confirmación (ACK=101) se enviaría tras esperar 500 ms, de acuerdo al estándar. Si el tiempo entre la llegada del primer y segundo segmento fuera inferior a 500 ms, simplemente esta primera confirmación no se enviaría y el resto sería igual. La segunda confirmación sería también ACK=101, debido a la llegada de un segmento fuera de orden, y se enviaría de manera inmediata de acuerdo al estándar. La tercera confirmación, ACK=301, sería acumulativa, y se enviaría de manera inmediata de acuerdo al estándar.

**A****B****Figura 1****A****B****Figura 2**



- 12.** Los hosts A y B están directamente conectados mediante un enlace de 100 Mbps. Existe una conexión TCP entre los dos hosts y el host A está transfiriendo al host B un archivo de gran tamaño. El host A puede enviar los datos de la capa de aplicación a su socket a 120 Mbps pero el host B sólo puede leer los datos de su socket a 60 Mbps. Describa el efecto del control de flujo TCP.

El control de flujo TCP hará que el host A envíe a su socket datos a una velocidad no superior a la que se leen los datos en el socket del host B (60 Mbps).

- 13.** En una conexión TCP, el nodo A tiene en su ventana los bytes 200, 201 y 202 enviados pero no confirmados, recibe un mensaje ACK con un valor de acuse de 202 y una ventana útil de 9. Si a continuación envía los bytes 203, 204 y 205 y si el valor de la ventana de congestión es 20.

- a. Muestre la nueva ventana en el nodo A.

La nueva ventana se muestra en la Figura3.

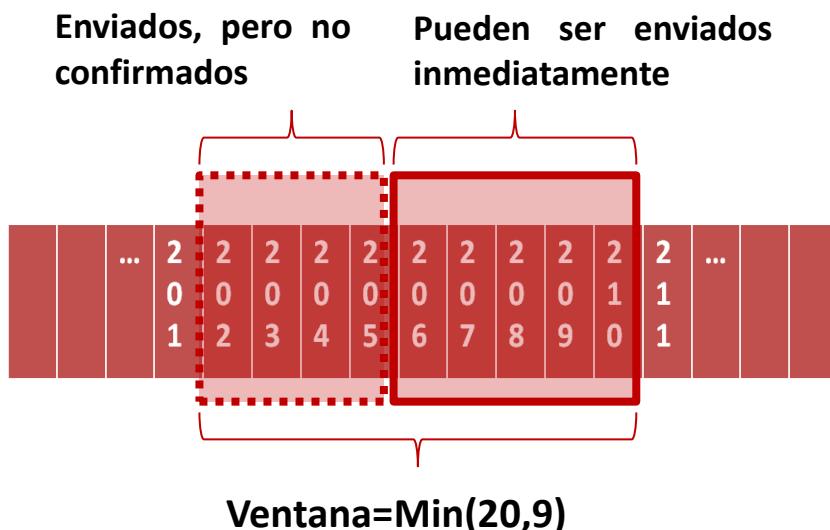


Figura 3

- b. El nodo A envía los bytes 206, 207, 208 y 209; y posteriormente recibe un paquete ACK con acuse 210 y ventana útil de 5. Si el valor de la ventana de congestión sigue siendo 20, muestre la ventana resultante.

La ventana resultante se muestra en la Figura 4.

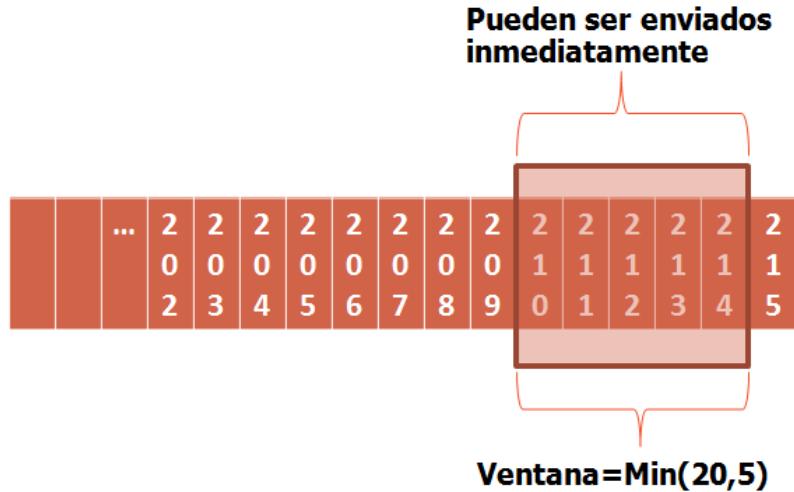


Figura 4

14. Si el RTT es 30 ms, la Desviación es 2 ms y se reciben ACKs tras 26, 32 y 24 ms, ¿Cuál será el nuevo RTT, Desviación y timeout? Usar  $\alpha=0,125$  y  $\beta=0,25$ .

-Inicialmente:

$$RTT_0 = 30ms$$

$$Desv_0 = 2ms$$

-Tras la llegada del primer ACK:

$$RTT_1 = (1 - \alpha) * RTT_0 + \alpha * 26 = 29,5ms$$

$$Desv_1 = (1 - \beta) * Desv_0 + \beta * |RTT_1 - 26| = 2,38ms$$

-Tras la llegada del segundo ACK:

$$RTT_2 = 0,875 * RTT_1 + 0,125 * 32 = 29,81ms$$

$$Desv_2 = 0,75 * Desv_1 + 0,25 * |RTT_2 - 32| = 2,33ms$$

-Tras la llegada del tercer ACK:

$$RTT_3 = 0,875 * RTT_2 + 0,125 * 24 = 29,08ms$$

$$Desv_3 = (1 - \beta) * Desv_2 + \beta * |RTT_3 - 24| = 3,02ms$$

$$tout = RTT_3 + 4 * Desv_3 = 41,16ms$$

¿Y si los dos primeros ACKs tienen el mismo número de acuse y se usa el algoritmo de Karn?

Los dos primeros no se utilizan en el cálculo:

$$RTT_1 = 0,875 * RTT_0 + 0,125 * 24 = 29,25ms$$



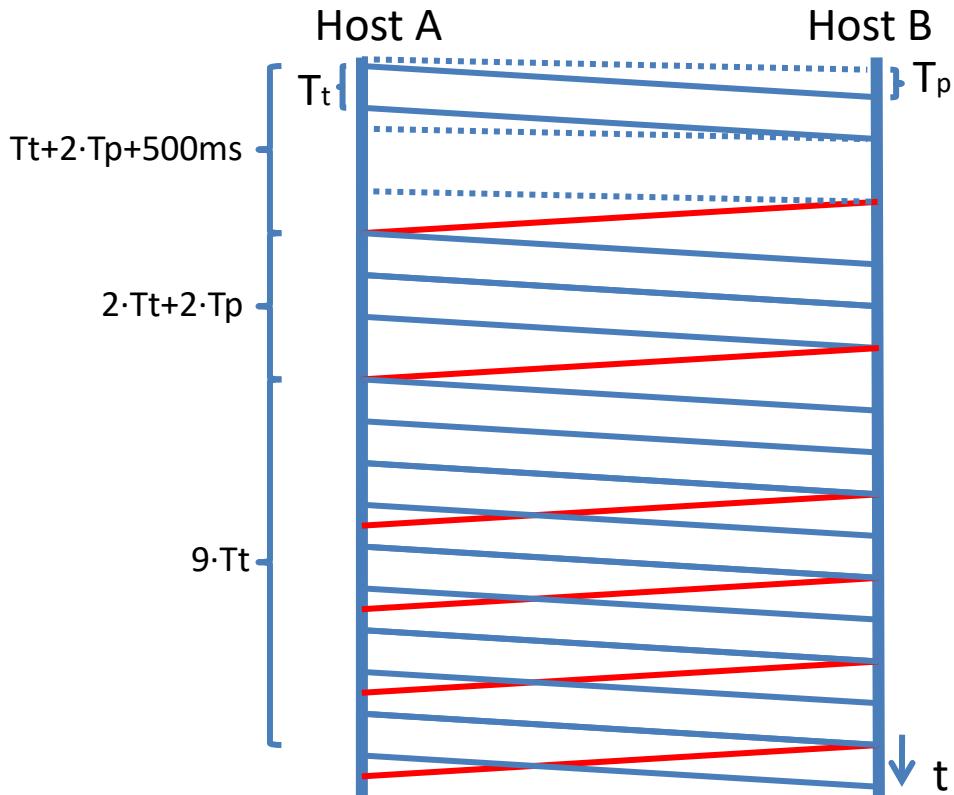
$$\text{Desv}_1 = (1 - \beta) * \text{Desv}_0 + \beta * |RTT_1 - 24| = 2,81\text{ms}$$
$$tout = RTT_1 + 4 * \text{Desv}_1 = 40,49\text{ms}$$

15. Teniendo en cuenta el efecto del inicio lento, en una línea sin congestión con 10 ms de tiempo de propagación, 1 Mbps de velocidad de transmisión y un MSS de 2KB, ¿cuánto tiempo se emplea en enviar 24 KB?

A partir de los parámetros, se puede calcular que:

- El número de segmentos, asumiendo siempre MSS datos, a enviar es de  $24/2=12$ .
- El tiempo de transmisión, despreciando cabeceras, es de  $2\text{KB}/1\text{Mbps} = 16,384\text{ ms}$ .
- Se desprecia el tiempo de transmisión de los paquetes ACK.

Con estos parámetros, el diagrama de tiempos es el siguiente:



Por tanto, el tiempo total en transmitir todos los datos desde el emisor es:

$$T_{\text{tot}} = 12 \cdot T_t + 4 \cdot T_p + 500\text{ms} = 736,608\text{ ms}$$



**16. Suponiendo que la ventana de congestión es 18 KB y que se dispara un *timeout*. ¿Cuánto será la ventana de congestión si las 4 siguientes ráfagas de transmisiones, donde se envía la ventana completa, son exitosas? Suponer que el MSS es 1 KB.**

- i. Inicialmente, VC es de 18 KB.
- ii. Tras el timeout, VC pasa a 1KB y el umbral a 9KB.
- iii. Tras la primera ráfaga de 1KB, VC pasa a 2KB.
- iv. Tras la segunda ráfaga de 2KB, VC pasa a 4KB.
- v. Tras la tercera ráfaga de 4KB, VC pasa a 8KB.
- vi. Tras la cuarta ráfaga de 8KB, VC pasa a 9KB. Esto ocurre de la siguiente forma. Tras el ACK del primer 1KB, VC ya pasa a 9KB y a partir de ese momento, por cada ventana enviada completa, aumentará un MSS. Es decir, tras recibir los siguientes 9KB, pasará a 10KB, pero la cuarta ráfaga sólo llega a 8KB.

**17. Dadas dos entidades TCP (A y B) conectadas por una red cuya velocidad de transmisión es 100 Mbps, suponga segmentos de 1024 bytes y un RTT (*Round Trip Time*) constante de 2 mseg. Si A transmite masivamente datos a B ¿Cuánto tiempo tardará en transmitir completamente 8 tramas? Incluya el número de secuencia y de acuse en todos los segmentos TCP. Haga las suposiciones que estime necesarias.**

$L \rightarrow$  Segmento TCP  $\rightarrow 1024$  bytes  
(+20 bytes de cabecera IP + X bytes de cabeceras de otras capas inferiores)

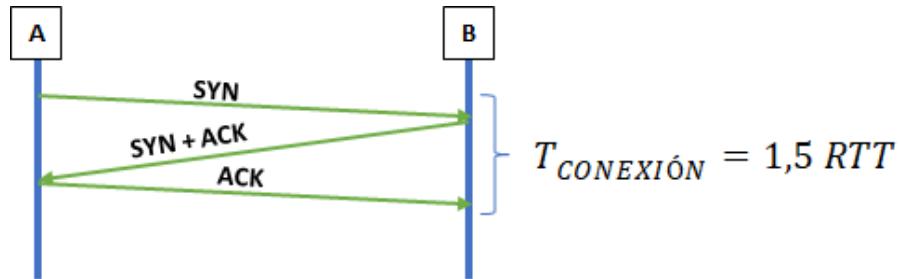
$$V_t = 100 \text{ Mbps}$$

$$RTT = 2 \text{ ms}$$

$$\text{Tiempo de transmisión} = L/V_t$$

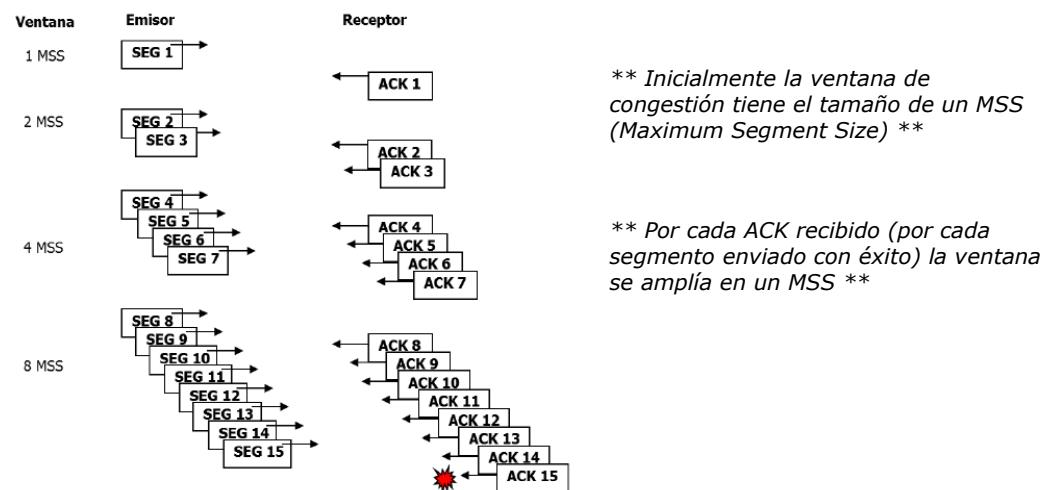
#### SUPOSICIONES:

- Tiempo ida = Tiempo vuelta (RTT/2)
  - Tiempo de generación/procesamiento de tramas/ACK  $\approx 0$
  - Consideramos los tamaños de cabecera de TCP=20 bytes e IP=20 bytes.
  - Se desprecia el tamaño de las cabeceras y colas de las capas inferiores a IP.
  - Consideramos que  $RTT = T_{\text{TRANSMISIÓN}} + 2 \times T_{\text{PROPAGACIÓN}}$
  - Sólo se considera como cabecera adicional la de la capa de red (cab. IP) ya que aún no conocen la capa de enlace.
- 
- Hay que considerar **el tiempo de establecimiento de conexión de TCP**
  - Hay que considerar **control de congestión de TCP**
  - Supondremos un tamaño de ventana de congestión inicial de 2, ya que será más eficiente (no habría que esperar los 500ms para confirmar el primer segmento en caso de que el tamaño de ventana de congestión fuese 1).
  - **Control de flujo:** La ventana ofrecida por el receptor será suficiente para recibir todos los segmentos en cada momento.

Establecimiento de conexión:

Envío de mensajes considerando el **inicio lento (slow start)** del control de congestión de TCP:

Inicio lento:

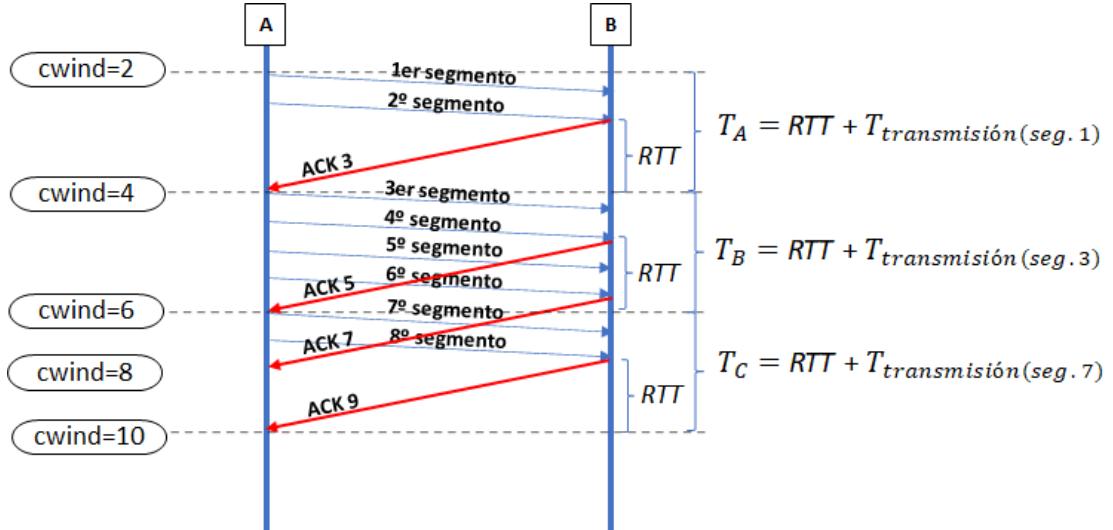


Según el protocolo de generación de ACKs:

- El receptor se espera a recibir el siguiente segmento 500ms, si no lo recibe envía el ACK.
- Si llega un segmento ordenado y ya había otro sin confirmar, se genera un ACK acumulado para los dos.



\*\* **cwind** se refiere al tamaño de la ventana de congestión \*\*



\*\* 20 → cabecera IP, 40 → cabeceras IP + TCP \*\*

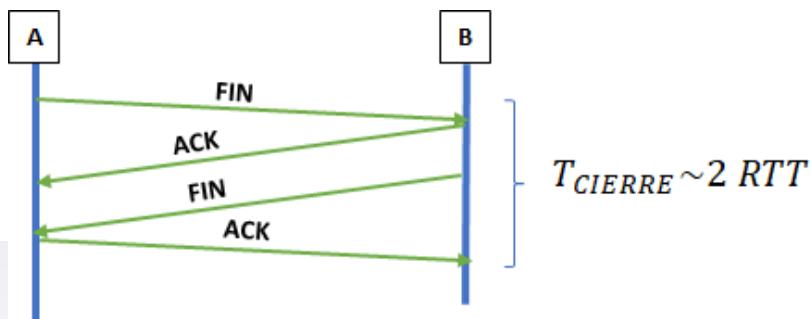
$$T_A = T_B = T_C = T_{ida} + 8 \cdot (1024+20)100 \cdot 106 + T_{vuelta} = RTT + 8 \cdot (1024+20)100 \cdot 106 = 2 \text{ ms} + 83,52 \mu\text{s} = 2,1 \text{ ms}$$

$$\begin{aligned} T_{\text{TOTAL}} &= T_{\text{CONEXIÓN}} + T_A + T_B + T_C = 3,0096 \text{ ms} + 2,1 \text{ ms} + 2,1 \text{ ms} + 2,1 \text{ ms} \\ &= 9,4 \text{ ms} \end{aligned}$$

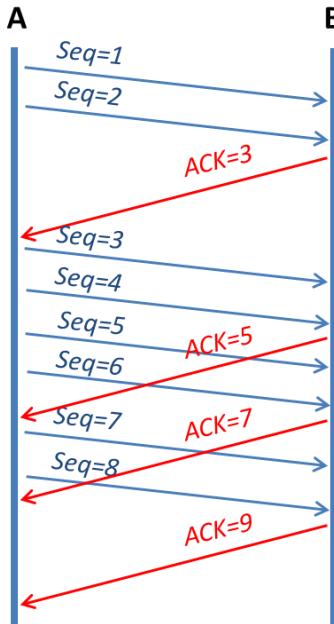
- A esto se le podría sumar el tiempo de generación de los segmentos.
- También el tiempo de generación y procesamiento de ACKs.  
Además, en cada suma de tamaños de cabeceras, habría que considerar el tamaño de las cabeceras de capas inferiores.

También podemos considerar que el envío de los 8 segmentos finaliza cuando se han enviado correctamente los 8 segmentos y se cierra la conexión.

Por tanto, se puede sumar al tiempo total el tiempo de cierre de la conexión TCP. Este cierre puede ser igual a la conexión (FIN, FIN-ACK, ACK) o, si consideramos que las entidades tienen que terminar envíos antes de hacer el cierre, se podría calcular como:



$$\begin{aligned} T_{\text{TOTAL}} &= T_{\text{CONEXIÓN}} + T_A + T_B + T_c + T_{\text{CIERRE}} = 3 \text{ ms} + 2,1 \text{ ms} + 2,1 \text{ ms} + 2,1 \text{ ms} \\ &+ 4 \text{ ms} = 13,3 \text{ ms} \end{aligned}$$

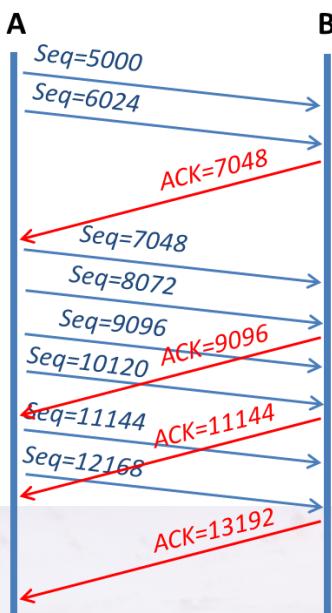
Números de secuencia y acuse (numeración simple):

Usando una numeración simple para los números de secuencia y considerando que el receptor B no enviaría datos a A (sólo confirmaciones), el intercambio de segmentos y acuses sería el mostrado en la figura.

(Piense en un servicio de subida de archivos, por ejemplo, en el que el servidor B únicamente confirma que se han subido correctamente los datos)

Debemos recordar que se trata de confirmaciones acumulativas, es decir, un acuse confirma varios segmentos recibidos.

El acuse se referirá al número de secuencia del siguiente segmento a recibir.

Números de secuencia y acuse (numeración realista):

Consideramos que el ISN es actualmente 5000, por lo que el primer segmento se asocia a ese número.

Cada segmento tiene un tamaño de 1024 bytes, según el enunciado.

Consideramos nuevamente que el receptor B no enviaría datos a A (sólo confirmaciones).

Se trata de confirmaciones acumulativas.

El acuse se referirá al número de secuencia (byte de comienzo) del siguiente segmento a recibir.

\*\* Hay que destacar que, tanto este, como el diagrama anterior no son precisos en cuanto a la representación de los tiempos, ya que, por ejemplo, el tiempo de transmisión del ACK será en realidad menor que el del envío de los segmentos de datos.  
Se ha hecho de esta forma para mayor claridad. \*\*



Universidad de Granada

Fundamentos de Redes

3º del Grado en Ingeniería  
Informática



Dept. Teoría de la Señal,  
Telemática y Comunicaciones

**17. Suponga una conexión TCP entre dos entidades ¿Qué ocurre en las dos entidades al detectarse una pérdida?**

**18. Suponga el envío de un fichero grande sobre una conexión TCP que tiene un RTT constante. La velocidad de transmisión puede suponerse muy elevada.**

- a) Si CongWin es inicialmente 2 MSS, ¿cuánto tiempo como mínimo se necesitará para que CongWin sea 10 MSS? Suponga que no entra en la zona de prevención de congestión.
- b) ¿Cuál será el throughput medio tras 5 RTTs?
- c) Si CongWin es 50 MSS y está en la zona de prevención de congestión, ¿cuánto tardará CongWin en llegar a 58 MSS? ¿Cuál será el throughput medio en ese momento?

## **2.4. Relación 4**



## Ejercicios – Tema 4

1. Suponga un posible escenario para la entrega telemática de la Declaración del Impuesto de la Renta de Personas Físicas (I.R.P.F.) que contempla su pago inmediato a través de Internet. Los agentes implicados serán la persona que presenta la declaración (P), la Agencia Estatal de Administración Tributaria (AT) y el banco donde la persona tiene una cuenta (BP).

En este escenario hipotético se intercambian los mensajes indicados debajo, donde certificado\_digital<sub>X</sub> se refiere al certificado digital de X, Kpriv<sub>X()</sub> al cifrado mediante la clave privada de X, Kpúb<sub>X()</sub> al cifrado mediante la clave pública de X, datos\_fiscales<sub>X</sub> a los datos de la declaración de I.R.P.F. de X, importe a la cantidad a pagar como resultado de la declaración de I.R.P.F. de X, código\_para\_pagar\_IRPF es un código indicado por la AEAT para que la persona realice el pago en su banco y código\_IRPF\_pagado es un código indicado por el banco a la persona como comprobante de su pago.

```
P → AT: certificado_digitalP
AT → P: certificado_digitalAT
P → AT: KprivP (KpúbAT (datos_fiscalesP, importe))
AT → P: KprivAT (KpúbP (código_para_pagar_IRPF))
P → BP: certificado_digitalP
BP → P: certificado_digitalBP
P → BP: KprivP (KpúbBP (importe, código_para_pagar_IRPF))
BP → P: KprivBP (KpúbP (código_IRPF_pagado))
P → AT: KprivP (KpúbAT (certificado_digitalBP, código_IRPF_pagado))
AT → BP: KprivAT (KpúbBP (identidadP, código_para_pagar_IRPF))
BP → AT: KprivBP (KpúbAT (identidadP, código_IRPF_pagado))
AT → P: KprivAT (KpúbP (mensaje_declaración_correcta))
```

Todos los certificados digitales han sido expedidos por una Autoridad de Certificación fiable (e.g. la Fábrica Nacional de Moneda y Timbre). Además, la AEAT conoce la identidad de los bancos a través de los cuales se puede realizar el pago telemático de la declaración de I.R.P.F. Responda razonadamente las siguientes cuestiones:

- a) ¿Qué servicios de seguridad se proporcionan en la transacción indicada?  
b) ¿Qué debilidades/vulnerabilidades presenta el esquema y, en su caso, cómo podrían solucionarse?
2. Explique el objetivo que se persigue al utilizar firmas digitales. Exponga detalladamente los mecanismos de firma digital que conozca.
3. Suponga una transacción comercial en Internet con cuatro entidades involucradas: C (cliente), P (proveedor), Bc (entidad bancaria del cliente) y Bp (entidad bancaria del proveedor). Entre ellas se intercambian los mensajes indicados abajo a la derecha; donde Kpb<sub>X</sub> se refiere al cifrado con la clave pública de X, K<sub>X-Y</sub> al cifrado con la clave secreta entre X e Y, producto a la identificación del producto adquirido/vendido, importe a su valor económico, R a un reto, C, P, Bc y Bp a la identidad de las entidades correspondientes y datos\_X a la información bancaria correspondiente a X-Bx.  
Aceptadas la disponibilidad y validez de las claves públicas involucradas en base a la existencia de una entidad superior confiable, responda justificadamente a las siguientes cuestiones:



- a) ¿Qué servicios de seguridad se proporcionan en la transacción indicada?
- b) ¿Qué debilidades/vulnerabilidades presenta el esquema propuesto y, en su caso, cómo podrían solucionarse?
- C→P:  $Kpb_P(producto, importe, datos_C)$   
P→Bp:  $Kpb_{Bp}(importe, datos_C, P)$   
Bp→P:  $Kpb_P(datos_P, R)$   
P→Bp:  $Kpb_{Bp}(datos_P, K_{P-Bp}(R))$   
Bp→Bc:  $Kpb_{Bc}(importe, datos_C, P)$   
Bc→C:  $Kpb_C(importe, datos_C, P, R')$   
C→Bc:  $Kpb_{Bc}(importe, datos_C, P, K_{C-Bc}(R'))$   
Bc→Bp:  $Kpb_{Bp}(importe, datos_C, P)$   
Bp→P:  $Kpb_P(importe, datos_C)$   
P→C: ...entrega del producto...
4. ¿Es posible autenticar mutuamente con garantías dos entidades A y B, tal que A dispone de certificado digital y B no? Explique la respuesta adoptando las suposiciones que estime necesarias.
5. Describa el funcionamiento del protocolo de aplicación PGP (*Pretty Good Privacy*). Describa los pasos para el envío y la recepción de un mensaje, incluyendo qué aspectos de seguridad se garantizan y cómo.
6. ¿Qué tres objetivos fundamentales tiene la firma digital? Describa tres procedimientos para realizar una firma digital.
7. ¿Son DES o IDEA algoritmos de sustitución o trasposición? Explique un esquema para evitarlo.
8. Explique cómo establecer una clave secreta a través de un canal no seguro. ¿qué debilidades tienes? Ponga un ejemplo de protocolo estandarizado en el que se use ese procedimiento.
9. Suponga un protocolo que por cada mensaje en texto plano M, envía  $(M, H(M) \oplus KS)$ , donde  
H(x) es un compendio o Hash de x  
(a  $\oplus$  b) es la X-OR de a y b  
 $KS$  es una clave secreta compartida entre los dos extremos.  
¿Qué aspectos de seguridad y cuáles no garantiza? Justifique la respuesta y proponga en su caso una alternativa –con las mismas herramientas– que sea más segura.
10. Explique detalladamente qué es un certificado digital y **qué información contiene**. Describa cómo se podría, **UTILIZANDO CERTIFICADOS DIGITALES**, garantizar la autenticación, la integridad y el no repudio en las comunicaciones entre dos entidades con certificados digitales emitidos por entidades de certificación fiables.
11. Explique detalladamente cómo se puede utilizar certificados digitales para realizar firmas digitales (únicamente firmas digitales **USANDO CERTIFICADOS DIGITALES**). Para ese procedimiento concreto, explique qué aspectos de seguridad se garantizan.
12. La figura y mensajes siguientes describen un hipotético protocolo utilizado para permitir el acceso de un cliente a Internet a través de un Servidor de Acceso a Red (NAS). El Servidor de Autenticación (AS) guarda en una base de datos las claves secretas que se solicita a los usuarios para poder acceder a Internet.



**PC → NAS:** KpubNAS (peticion\_acceso + usuario)  
**NAS → PC:** desafío  
**PC → NAS:** KpubNAS(MD5(usuario:KPC-AS:desafío))  
**NAS → AS:** peticion\_autenticacion + usuario + desafío + MD5(usuario:KAS-PC:desafío))  
**AS → NAS:** peticion\_aceptada + KsesionPC-NAS + KPC-AS(KsesionPC-NAS)  
 (ó peticion\_rechazada)  
**NAS → PC:** KprivNAS (peticion\_aceptada + KPC-AS(KsesionPC-NAS))  
 (ó KprivNAS (peticion\_rechazada))  
**PC → NAS:** KsesionPC-NAS (datos\_a\_enviar)  
**NAS → hacia Internet:** datos\_a\_enviar  
**Desde Internet → NAS:** datos\_de\_respuesta  
**NAS → PC:** KsesionPC-NAS (datos\_de\_respuesta)

Siendo:

- Kpubx cifrado con la clave pública de X
- Kprivx cifrado con la clave privada de X
- K<sub>X-Y</sub> la clave secreta entre X e Y
- MD5 es una función hash

Aceptadas la disponibilidad y validez de las claves públicas involucradas en base a la existencia de una entidad superior confiable, responda razonadamente:

- a) ¿Qué servicios de seguridad se proporcionan en el protocolo descrito?  
 b) ¿Qué debilidades presenta el esquema propuesto? En su caso, ¿cómo podrían evitarse?
13. Un protocolo de reto-respuesta...
  - 3.1. ¿Qué es y para qué sirve?
  - 3.2. Suponiendo la existencia de una clave secreta compartida ponga un ejemplo de mensajes intercambiados.
  - 3.3. Identifique sus posibles debilidades.
  - 3.4. ¿Sería posible realizarlo si dispusiera de certificados digitales? En su caso ¿cómo?
14. Suponga una transacción comercial en Internet con cuatro entidades involucradas: C (cliente), P (proveedor), Bc (entidad bancaria del cliente) y Bp (entidad bancaria del proveedor). Entre ellas se intercambian los mensajes indicados abajo a la derecha; donde KpbX se refiere al cifrado con la clave pública de X, K<sub>X-Y</sub> al cifrado con la clave privada entre X e Y, producto a la identificación del producto adquirido/vendido, importe a su valor económico, R a un reto, C, P, Bc y Bp a la identidad de las entidades correspondientes y datos\_X a la información bancaria correspondiente a X-Bx.

Aceptadas la disponibilidad y validez de las claves públicas involucradas gracias a la existencia de una entidad superior confiable (es decir, al uso de certificados digitales), responda justificadamente a las siguientes cuestiones:

¿Qué servicios de seguridad se proporcionan en la transacción indicada?

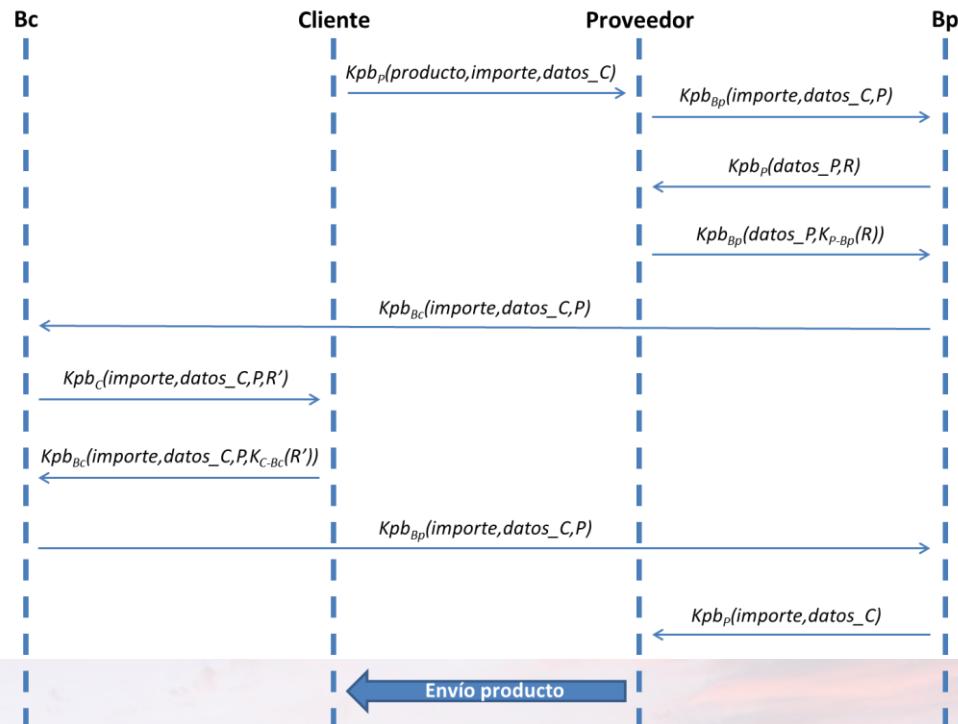
¿Qué debilidades/vulnerabilidades presenta el esquema propuesto y, en su caso, cómo podrían solucionarse?

C→P:  
 $KpbP(\text{producto}, \text{importe}, \text{datos}_C)$   
 P→Bp:  $KpbBp(\text{importe}, \text{datos}_C, P)$   
 Bp→P:  $KpbP(\text{datos}_P, R)$   
 P→Bp:  $KpbBp(\text{datos}_P, Kp-Bp(R))$   
 Bp→Bc:  $KpbBc(\text{importe}, \text{datos}_C, P)$   
 Bc→C:  $KpbC(\text{importe}, \text{datos}_C, P, R')$   
 C→Bc:  
 $KpbBc(\text{importe}, \text{datos}_C, P, KC-Bc(R'))$   
 Bc→Bp:  $KpbBp(\text{importe}, \text{datos}_C, P)$   
 Bp→P:  $KpbP(\text{importe}, \text{datos}_C)$   
 P→C: ...entrega del producto...

#### MENSAJES:

- $KpbX$  → cifrado con la clave pública de X
- $K_{X-Y}$  → cifrado con la clave privada entre X e Y
- **producto** → identificación del producto adquirido/vendido
- **importe** → valor económico de un producto
- **R** → reto
- **datos\_X** → información bancaria correspondiente a X-Bx

#### EL PROTOCOLO SERÍA:



#### a) ¿Qué servicios de seguridad se proporcionan en la transacción indicada?

- **Confidencialidad** → sí, ya que todos los mensajes están cifrados con clave pública, por tanto, sólo el dueño de la clave privada puede obtener su contenido.



- **Integridad** → no, ya que no se usan funciones hash.
- **Autenticación** → sólo el cliente/proveedor con sus bancos respectivos, mediante el envío cifrado del reto propuesto ( $R$  y  $R'$ ). Sin embargo, los bancos no se autentican entre ellos ni con sus clientes.
- **No repudio** → no, ya que el cliente no tiene ninguna prueba de que el proveedor haya aceptado la transacción que implica cierto producto y su importe. Ni siquiera de que haya realizado el pago, ya que su banco no le envía la confirmación de la operación con algún campo que sólo hubiese podido incluir él.
- **Disponibilidad** → no, ya que la red podría dejar de funcionar en cualquier momento, por ataques en capas inferiores o por fallos de la misma.

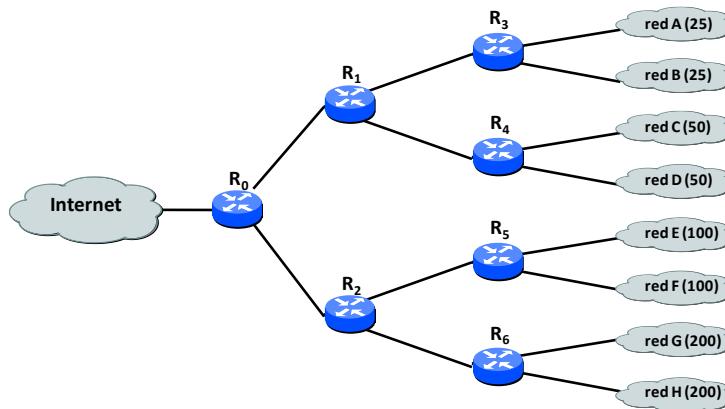
#### b) ¿Qué debilidades/vulnerabilidades presenta el esquema propuesto y, en su caso, cómo podrían solucionarse?

- **Integridad** → se podría usar una función compendio (hash) para comprobar la integridad de los datos.
- **Autenticación** → podría haber autenticación entre los bancos el cliente/proveedor mediante un reto propuesto por  $C$  a  $B_C$  y por  $P$  a  $B_P$ . También podría haber autenticación entre los bancos proponiéndose un reto cada uno.
- **No repudio** → tanto cliente como proveedor podrían firmar digitalmente sus mensajes antes de transmitirlos (con su clave privada) y el receptor del mensaje lo desencriptaría con la clave pública correspondiente. Igualmente, el banco podría mandar una confirmación de la operación realizada firmada digitalmente con su clave privada.
- **Disponibilidad** → el enunciado no da información que permita indicar si hay problemas de disponibilidad (Ej: redundancia de conexiones, posibles problemas ante ataques en capas inferiores, etcétera).

## **2.5. Relación 5**

## Ejercicios – Tema 5

- En la figura siguiente suponga que  $R_0$  corresponde con el servidor DNS raíz de un espacio de nombres de dominio ficticio,  $R_1$  y  $R_2$  son los servidores de los dominios  $.R1$  y  $.R2$  respectivamente,  $R_3$  corresponde con el servidor del dominio  $.R3.R1$ , etc. Suponiendo resolución recursiva, describa paso a paso **los mensajes DNS intercambiados** para enviar un correo desde un MUA situado en la red A, a un destinatario cuya MTA estuviera instalada en MTA.R6.R2.



- ¿Qué tienen en común HTTP y SMTP?
- Discuta breve y razonadamente los principales retos y soluciones para el desarrollo de servicios IP multimedia (voz, vídeo, ...) desde la perspectiva de: (a) consumo de ancho de banda, (b) retardo de transmisión y (c) confidencialidad de los datos.
- Explique detalladamente (incluyendo los mensajes de resolución de nombres que sean necesarios suponiendo resolución recursiva) todos los mensajes de aplicación intercambiados en el envío y recepción de un correo electrónico (suponga IMAP) entre dos MUAs.
- Suponga dos usuarios de correo electrónico,  $user1$  y  $user2$ , situados en sendos puestos de trabajo  $H1$  y  $H2$ . Las estafetas de correo correspondientes son  $mailserver.com$  y  $servidormail.es$ . Describa todos los pasos y protocolos involucrados en los siguientes procesos:
  - Redacción y envío de un email de  $user1$  a  $user2$ , desde el punto de vista del primero de ellos.
  - Recepción del mensaje en  $servidormail.es$ .
  - Descarga y lectura del correo por parte de  $user2$ .
- ¿Qué diferencias o semejanzas hay entre POP e IMAP?
- Desde un ordenador se arrancan tres navegadores diferentes, Internet Explorer, Mozilla Firefox y Google Chrome, y se accede desde los tres a un servidor web en la dirección 147.156.1.4 (el mismo desde los tres) ¿Cuántos sockets y cuantas conexiones TCP están implicados, tomando en cuenta tanto el lado servidor como el cliente?
- Discuta las características de las siguientes aplicaciones en términos de su tolerancia a la pérdida de datos, los requisitos temporales, la necesidad de rendimiento mínimo y la seguridad.



- a. La telefonía móvil
- b. WhatsApp
- c. YouTube
- d. Spotify
- e. Comercio electrónico

Nótese que el diseño de una aplicación siempre debe tener en cuenta todas las características anteriores. No obstante, varias de estas características son difíciles de conseguir de forma simultánea, cuando no son antagónicas. Por este motivo, el diseño de una aplicación debe considerar qué características primar en detrimento de otras.

Se propone la siguiente tabla de asignación de prioridades, utilizando la siguiente notación:

Requisito fundamental: ↑      Requisito relevante: ↔      Requisito secundario: ↓

|                      | Tolerancia a<br>pérdidas | Requisitos Temporales<br>(Delay) | Rendimiento<br>(Throughput) | Seguridad |
|----------------------|--------------------------|----------------------------------|-----------------------------|-----------|
| Telefonía móvil      | ↓                        | ↑                                | ↑                           | ↔         |
| WhatsApp             | ↑                        | ↓                                | ↓                           | ↔         |
| YouTube              | ↓                        | ↔                                | ↑                           | ↓         |
| Spotify              | ↓                        | ↔                                | ↑                           | ↓         |
| Comercio Electrónico | ↑                        | ↔                                | ↔                           | ↑         |

12. ¿Es posible que un host tenga varias direcciones IP y un único nombre de dominio? Discuta un caso.

Sí. Por ejemplo un host que tenga varias interfaces de red.

¿Es posible que un host tenga varios nombres de dominio y una única dirección IP? Discuta un caso.

Sí, típicamente un host tiene un nombre canónico y varios *alias*.

¿Es posible que varios host tengan el mismo nombre de dominio, aunque direcciones IP distintas? Discuta un caso.

Si, esto puede hacerse para que el servidor DNS realice balanceo de tráfico, rotando su respuesta ante solicitudes de un determinado nombre de dominio (ej. [www.google.com](http://www.google.com) es, en realidad, un grupo de hosts con diferentes direcciones IP)

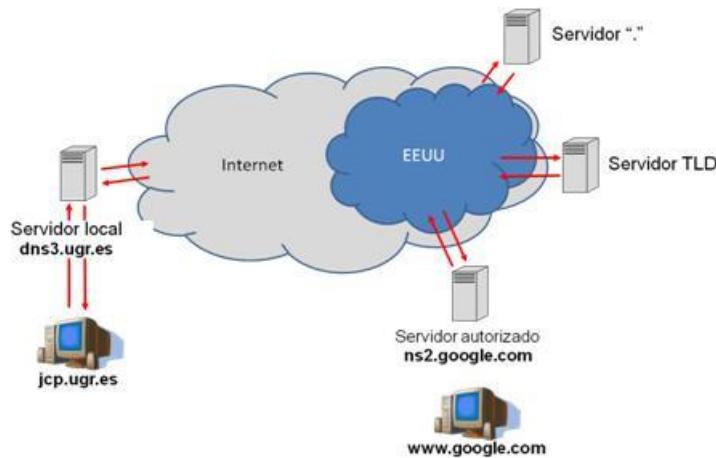
#### 13. ¿Qué carga más un servidor DNS, accesos iterativos o recursivos?

Desde el punto de vista de un servidor DNS que debe solicitar la traducción de un nombre a otro servidor, al primero le cargarán más accesos iterativos que recursivos, pues en el primer caso es él mismo el que se encarga de ir contactando con los respectivos servidores DNS hasta resolver completamente la petición del cliente. Si bien es cierto que un servidor que resuelve iterativamente está liberando de carga a los servidores DNS con los que contacta.

¿Cuál de estos dos accesos carga más la red?

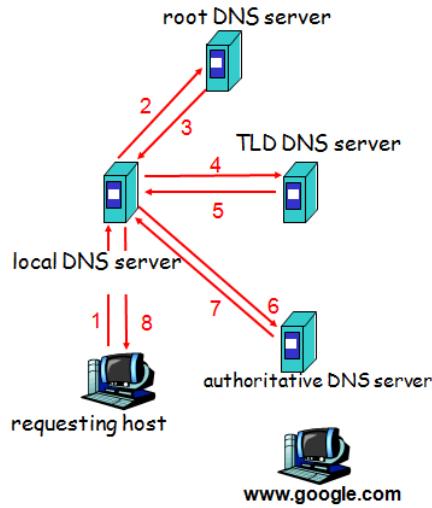
Los dos por igual, ya que el número de solicitudes y respuestas necesario para resolver una determinada solicitud de DNS es la misma.

#### 14. En la siguiente figura se ilustra un ejemplo de acceso DNS por parte de una máquina (jcp.ugr.es) que quiere acceder a los servicios de [www.google.com](http://www.google.com). Para obtener la dirección IP del servidor, es necesario que la consulta pase por todos los servidores del gráfico. Considerando unos retardos promedio de 8 µs dentro de una red LAN, de 12 ms en cada acceso a través de Internet (4 ms si la conexión se restringe a EEUU) y de 1 ms de procesamiento en cada servidor:



- Calcule el tiempo que se tardaría si la solicitud al servidor local es recursiva, pero el propio servidor local realiza solicitudes iterativas.

En definitiva, el esquema que se propone analizar es el siguiente:



El tiempo total invertido en la resolución es:

$$r_{resolución} = ret_{LAN} + ret_{Internet};$$

Donde:

$$\begin{aligned} ret_{LAN} &= 2 \cdot r_{LAN} + r_{servidor}; \\ ret_{Internet} &= 6 \cdot r_{España-EEUU} + 3 \cdot r_{servidor}; \end{aligned}$$

Teniendo en cuenta los datos del ejercicio:

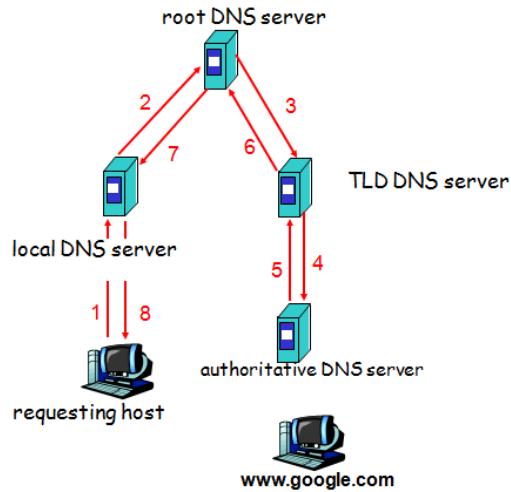
$$\begin{aligned} r_{LAN} &= 8\mu s; \\ r_{España-EEUU} &= 12 \cdot 10^3 \mu s; \quad r_{EEUU-EEUU} = 4 \cdot 10^3 \mu s; \quad r_{servidor} = 1 \cdot 10^3 \mu s \end{aligned}$$

Tenemos que:

$$r_{resolución} = 76,016 ms$$

- b. Especifique una política (recursiva-iterativa) más rápida de solicitudes y el tiempo que tardaría la solicitud en ser respondida. ¿Qué desventaja tiene sobre la solución anterior?

Una política más rápida en el supuesto que nos ocupa es esta:



En este caso tenemos que:

$$ret_{Internet} = 2 \cdot r_{España-EEUU} + 4 \cdot r_{EEUU-EEUU} + 3 \cdot r_{servidor}$$

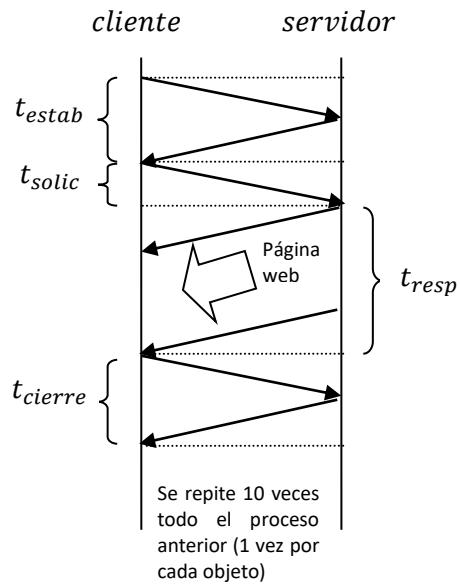
Con lo cual:

$$r_{resolución} = 44,016 \text{ ms}$$

El principal problema de esta estrategia es la sobrecarga del servidor raíz, aunque en todo caso se ha intentado minimizar haciendo a su vez una solicitud recursiva al servidor TLD.

15. Compare el rendimiento en términos temporales de HTTP persistente y no persistente considerando los siguientes parámetros:
- Descarga de una página web con 10 objetos incrustados
  - Tiempo de Establecimiento de conexión TCP → 5 ms
  - Tiempo de Cierre de conexión TCP → 5 ms
  - Tiempo de solicitud HTTP → 2 ms
  - Tiempo de respuesta HTTP (página web u objeto) → 10 ms

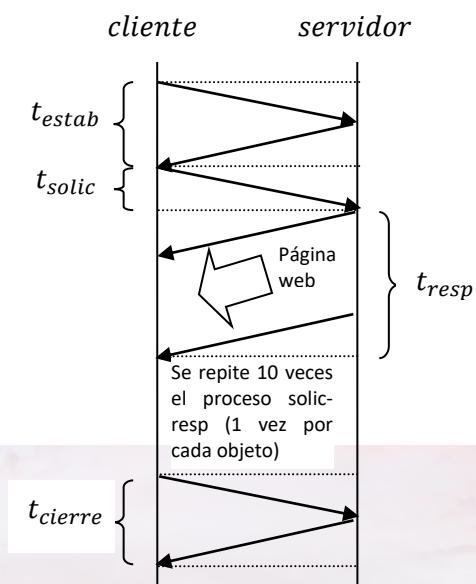
Para HTTP no persistente tenemos que el proceso de descarga sería el siguiente:



El tiempo de descarga para el modo HTTP no persistente es por lo tanto:

$$r_{descarga}^{np} = 11 \times (t_{estab} + t_{solic} + t_{resp} + t_{cierre}) = 242ms$$

Para HTTP persistente tenemos que el proceso de descarga sería el siguiente:



El tiempo de descarga para el modo HTTP persistente es por lo tanto:



$$r_{descarga}^p = t_{estab} + 11 \times (t_{solic} + t_{resp}) + t_{cierre} = 142ms$$

16. Una sucursal con 50 empleados en Granada tiene una red interna basada en *FastEthernet* (100Mbps) que se conecta a Internet con una red de acceso ADSL de 0,5 Mbps de subida y 1,5 Mbps de bajada. Cada empleado, en el desempeño de su trabajo, realiza un promedio de 2000 solicitudes de información a la hora a un servidor de Base de Datos ubicado en la central del banco, en Madrid, donde cada solicitud supone el envío por parte del servidor de un promedio de 10 registros de 1KB cada uno. Adicionalmente, la modificación de datos tras algunas de estas solicitudes supone el envío promedio de 100 actualizaciones, de 10 registros de media, a la hora desde la sucursal al servidor. El resto de los servicios telemáticos se restringe.

Calcule el promedio de la velocidad de transmisión requerida. ¿Es la velocidad del enlace de acceso suficiente?

Despreciando los paquetes de solicitud y confirmación, que contarán simplemente con cabeceras, podemos ver la velocidad requerida en promedio:

$$\begin{aligned} v_{download} &= 10 \frac{KB}{sol} \times (8 \times 1024) \frac{b}{KB} \times 2000 \frac{sol}{emp \times hora} \times 50 emp \\ &\quad \times \frac{1}{3600} \frac{hora}{s} \end{aligned}$$

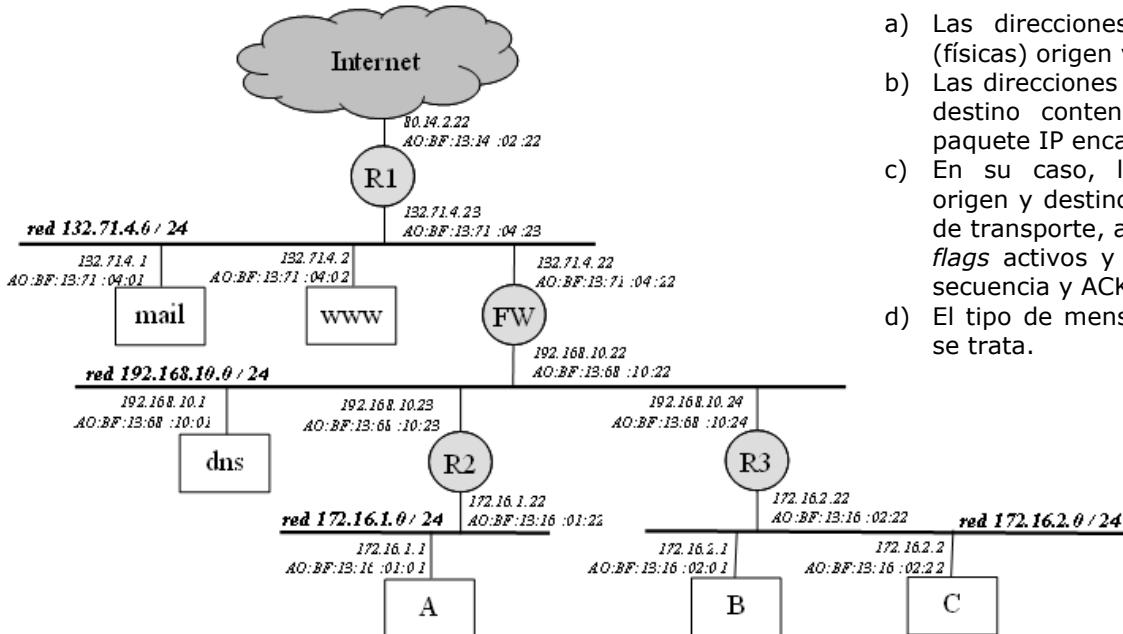
$$v_{download} = 2,28 \cdot 10^6 bps = 2,28Mbps$$

$$\begin{aligned} v_{upload} &= 10 \frac{KB}{act} \times (8 \times 1024) \frac{b}{KB} \times 100 \frac{act}{emp \times hora} \times 50 emp \\ &\quad \times \frac{1}{3600} \frac{hora}{s} \end{aligned}$$

$$v_{upload} = 0,11Mbps$$

La velocidad del enlace es insuficiente, ya que el download requerido en promedio es menor que el de la red.

17. Dada la topología adjunta correspondiente a una red corporativa, en la que se especifican tanto las direcciones IP como las MAC de cada uno de los dispositivos que la forman, analice el tráfico generado al hacer un acceso de correo electrónico desde el host "B" al servidor "mail", especificando en una tabla, y para cada trama Ethernet generada:

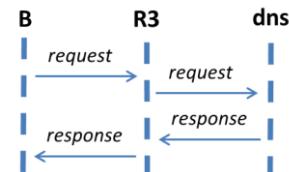


- a) Las direcciones hardware (físicas) origen y destino.
- b) Las direcciones IP origen y destino contenidas en el paquete IP encapsulado.
- c) En su caso, los puertos origen y destino de la PDU de transporte, así como los flags activos y campos de secuencia y ACK.
- d) El tipo de mensaje de que se trata.

**NOTA:** suponga todas las tablas ARP son conocidas y, por simplicidad utilice sólo el último de los 6 octetos de las direcciones físicas de las NIC (interfaces o tarjetas de red)

#### 1) PASO 1: Petición DNS y Respuesta

Es una **petición sobre UDP** □ no hay establecimiento de conexión previo



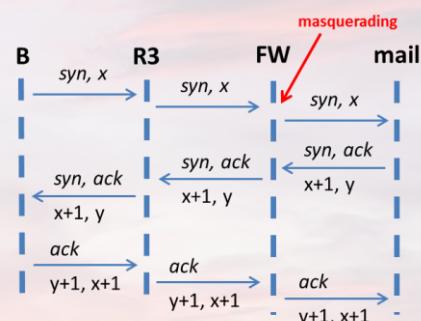
| ETH ORI. | ETH DES. | IP ORI.            | IP DEST.           | PORT ORI. | PORT DES. | FLAGS | MENSAJE                     | COMENTARIOS         |
|----------|----------|--------------------|--------------------|-----------|-----------|-------|-----------------------------|---------------------|
| 01 (B)   | 22 (R3)  | 172.16.2.1 (B)     | 192.168.10.1 (dns) | (1*)      | 53        | ---   | Solicitud DNS. Dominio mail | A través de R3      |
| 24 (R3)  | 01 (dns) | 172.16.2.1 (B)     | 192.168.10.1 (dns) | (1*)      | 53        | ---   | Solicitud DNS. Dominio mail | Retransmisión a dns |
| 01 (dns) | 24 (R3)  | 192.168.10.1 (dns) | 172.16.2.1 (B)     | 53        | (2*)      | ---   | Respuesta DNS IP de mail    | A través de R3      |
| 22 (R3)  | 01 (B)   | 192.168.10.1 (dns) | 172.16.2.1 (B)     | 53        | (2*)      | ---   | Respuesta DNS IP de mail    | Retransmisión a B   |

(1\*) Asignado por el S.O. . (2\*) Puerto elegido en (1\*)

#### 2) PASO 2: Establecimiento conexión TCP

**SMTP** → sobre TCP en el puerto 25

**Masquerading** es una traducción de IPs entre subredes. Hay que hacerlo para poder salir a la zona de direcciones públicas de la red





| ETH ORI.             | ETH DES.   | IP ORI.           | IP DEST.          | PORT ORI. | PORT. DES. | FLAGS           | MENSAJE                                | COMENTARIOS                                 |
|----------------------|------------|-------------------|-------------------|-----------|------------|-----------------|----------------------------------------|---------------------------------------------|
| 02:01 (B)            | 02:22 (R3) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (1*)      | 25         | SYN x (3*)      | Solicitud estab. TCP a mail            | A través de R3                              |
| 10:24 (R3)           | 10:22 (FW) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (1*)      | 25         | SYN x (3*)      | Solicitud estab. TCP a mail            | Retransmisión a FW                          |
| 04:22 (FW)<br>(mail) | 04:01 (FW) | 132.71.4.22 (FW)  | 132.71.4.1 (mail) | (5*)      | 25         | SYN x (3*)      | Solicitud estab. TCP a mail            | Masquerading (4*) FW entrega a mail         |
| 04:01 (mail)         | 04:22 (FW) | 132.71.4.1 (mail) | 132.71.4.22 (FW)  | 25        | (5*)       | SYN, ACK x+1, y | Aceptación y estab. en el otro sentido | mail hacia FW                               |
| 10:22 (FW)           | 10:24 (R3) | 132.71.4.1 (mail) | 172.16.2.1 (B)    | 25        | (2*)       | SYN, ACK x+1, y | Aceptación y estab. en el otro sentido | Deshace Masquerading (4*) FW retransm. a R3 |
| 02:22 (R3)           | 02:01 (B)  | 132.71.4.1 (mail) | 172.16.2.1 (B)    | 25        | (2*)       | SYN, ACK x+1, y | Aceptación y estab. en el otro sentido | R3 retransm. a B                            |
| 02:01 (B)            | 02:22 (R3) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (2*)      | 25         | ACK x+1, y+1    | Aceptación en el otro sentido          | A través de R3                              |
| 10:24 (R3)           | 10:22 (FW) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (2*)      | 25         | ACK x+1, y+1    | Aceptación en el otro sentido          | Retransmisión a FW                          |
| 04:22 (FW)<br>(mail) | 04:01 (FW) | 132.71.4.22 (FW)  | 132.71.4.1 (mail) | (5*)      | 25         | ACK x+1, y+1    | Aceptación en el otro sentido          | Masquerading (4*) FW entrega a mail         |

(1\*) Asignado por el S.O.      (2\*) Puerto elegido en (1\*)      (3\*) Num. Aleatorio elegido por el emisor

(4\*) FW al hacer masquerading mapea

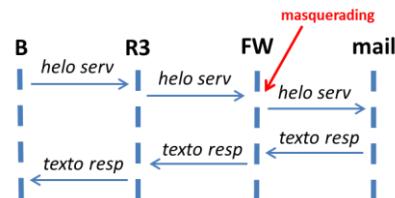
[IP intranet, puerto host intranet] → [IP pública FW, puerto libre en FW]

(5\*) Puerto elegido por FW en (4\*)

### 3) PASO 3: Acceso a correo electrónico

SMTP → sobre TCP en el puerto 25

**Masquerading** es una traducción de IPs entre subredes.



Sólo se inicia la conexión...Se podrían enviar más mensajes

| ETH ORI.             | E TH DES.  | IP ORI.           | IP DEST.          | P ORT ORI. | . PORT DES. | FLAGS                | MENS AJE                 | COMENTARIOS                                      |
|----------------------|------------|-------------------|-------------------|------------|-------------|----------------------|--------------------------|--------------------------------------------------|
| 02:01 (B)            | 02:22 (R3) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (2*)       | 25          | x+1                  | helo servidor            | Conexión inicial a servidor SMTP. A través de R3 |
| 10:24 (R3)           | 10:22 (FW) | 172.16.2.1 (B)    | 132.71.4.1 (mail) | (2*)       | 25          | x+1                  | helo servidor            | Retransmisión a FW                               |
| 04:22 (FW)<br>(mail) | 04:01 (FW) | 132.71.4.22 (FW)  | 132.71.4.1 (mail) | (5*)       | 25          | x+1                  | helo servidor            | Masquerading (4*) FW entrega a mail              |
| 04:01 (mail)         | 04:22 (FW) | 132.71.4.1 (mail) | 132.71.4.22 (FW)  | 25         | (5*)        | ACK x+1+NB(helo) y+1 | texto respuesta servidor | mail hacia FW                                    |
| 10:22 (FW)           | 10:24 (R3) | 132.71.4.1 (mail) | 172.16.2.1 (B)    | 25         | (2*)        | ACK x+1+NB(helo) y+1 | texto respuesta servidor | Deshace Masquerading (4*) FW retransm. a R3      |
| 02:22 (R3)           | 02:01 (B)  | 132.71.4.1 (mail) | 172.16.2.1 (B)    | 5          | 2           | (2*)                 | ACK x+1+NB(helo) y+1     | R3 retransm. a B                                 |

(2\*) Puerto elegido por el S.O. en el paso anterior

(4\*) FW al hacer masquerading mapea

[IP intranet, puerto host intranet] → [IP pública FW, puerto libre en FW]

(5\*) Puerto elegido por FW en (4\*)

NB: Número de bytes del mensaje

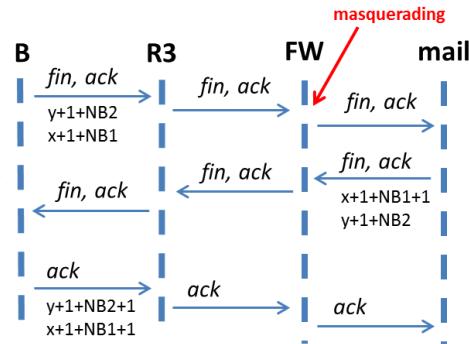
#### 4) PASO 4: Cierre de la conexión TCP

**Se envía confirmación del último mensaje del servidor, junto con la solicitud de cierre de conexión**

NB1 → longitud en bytes del mensaje "helo"  
NB2 → longitud en bytes de la respuesta

- La tabla y los campos son iguales que los del establecimiento de la conexión, salvo los flags, números de acuse y acks que se muestran en la figura.

\*\* Primero se indican los acuses y luego los identificadores (números de secuencia) de cada segmento \*\*



### **3 Fuente**

- Diapositivas de clase.