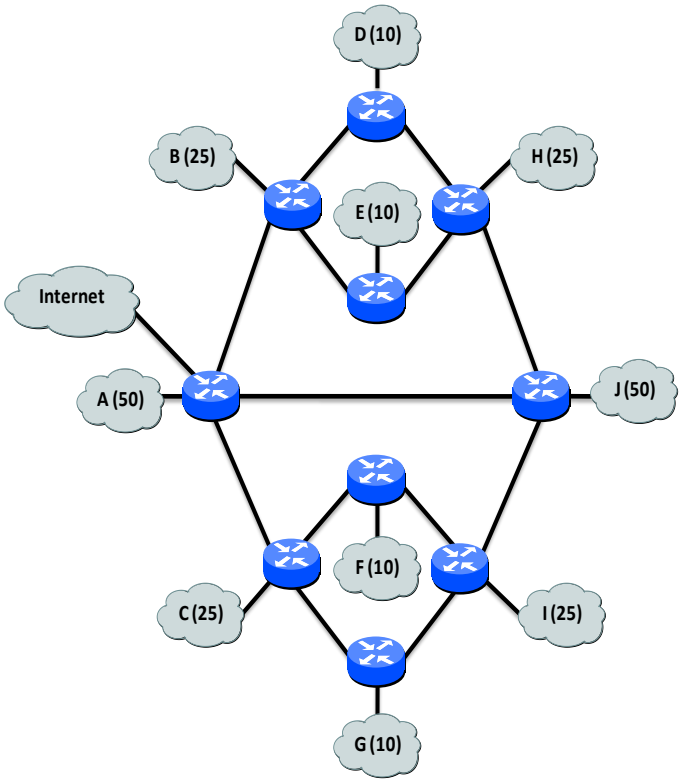
TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES II

– 4º curso de Ingeniería Informática –

Examen de teoría[1] – 6 de Julio de 2009

Apellidos y nombre: _____

1. (*2 puntos*) La siguiente figura muestra una red ficticia conectada a Internet. Se indica el número de ordenadores conectados a cada una de las subredes entre paréntesis.



a) Si se dispone únicamente del rango 10.10.10.0/24, ¿podría asignar todas las direcciones IP necesarias? En caso afirmativo, exponga dicha asignación de direcciones. En caso negativo, utilice además el rango de direcciones IP, consecutivas al rango anterior, que considere necesario.

b) Con la asignación de direcciones anterior, suponga que:
- La subred A es una red inalámbrica.
- Los usuarios que llegan a dicha red deben configurarse de forma automática.
- Existe un servidor de nombres en la subred D (elija su dirección IP).
- Los usuarios de esta red utilizan un servidor de HTTP en una máquina en la red J (elija su dirección IP).

Exponga la siguiente información para las tramas intercambiadas (niveles de red y superiores) cuando un usuario llega a la red inalámbrica, se conecta y envía un carga una página en su navegador.
- Direcciones físicas de origen y destino (utilice etiquetas representativas)
- Direcciones IP de origen y destino
- En su caso, puertos origen y destino
- En su caso, flags activos, campo de secuencia y de acuse
- Tipo de mensaje

Para este apartado, haga las suposiciones razonables que sean necesarias y justifíquelas. Exponga únicamente las tramas que se transmitan dentro de esta red.

2. (*2 puntos*) 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.

3. (*1,5 puntos*) Explicar el funcionamiento de TODOS los procedimientos que hacen que TCP sea adapatable y su impacto en las prestaciones finales.

4. (*1,5 puntos*) 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.

[1] ➔ La calificación de esta parte de la asignatura supondrá 7 puntos sobre el total de 10.

NOTA: Esta resolución del examen de la convocatoria de Junio de 2009 contiene indicaciones para ayudar al alumno a comprender cómo se realizan los ejercicios. No haría falta incluir todos los razonamientos sino emplearlos. Algunas aclaraciones, suposiciones, etc, sí deberían ser incluidos en el examen. Puede contener errores.

Ejercicio 1

Apartado a. Asignación de **todas** las direcciones IP necesarias usando el rango 10.10.10.0/24 (y rangos consecutivos si hiciese falta). **Aclaración**: el rango 10.10.10.0/24 es un rango de 256 direcciones **privadas** (la red 10.0.0.0/8 contiene las direcciones privadas de clase A).

Cada red (A, B, … J) tendrá un número de direcciones IP que será una potencia de 2, debido al uso de máscaras de subred. Dentro del rango elegido, deberán contemplarse:

- Las direcciones IP de los ordenadores conectados (50, 25, 10, …).
- La dirección IP del interfaz del *router* conectado a esa red.
- La dirección de subred (todos los bits de *host* a 0) y la dirección de difusión (todos los bits de *host* a 1).

Así, por ejemplo, la red A necesitará 50 + 1 + 2 = 53 direcciones IP. La potencia de 2 inmediatamente superior es 64, lo que requieren 32 (bits de una dirección IP) – 6 (bits necesarios para 64 direcciones, i.e. bits de *host*) = 26 bits de máscara de subred.

Las direcciones IP de una subred no pueden empezar en cualquier valor, ni tampoco terminar. En el ejemplo de la red A se necesitarían 6 bits para asignar 64 direcciones ($2^6 = 64$). Esto significa que los últimos 6 bits identificarán al equipo concreto (o a las direcciones de subred y difusión). La primera dirección (dirección de subred) tendrá esos bits a 0. Y la última dirección (dirección de difusión) tendrá esos bits a 1.

Por ejemplo, una dirección de subred inválida sería 10.10.10.13/26, ya que no tiene los últimos 6 bits (32 bits de dirección IP – 24 bits de máscara) a 0 ($13_{decimal} = 00001101_{binario}$). Una dirección de red válida podría ser 10.10.10.64/26, ya que los últimos 6 bits sí están a 0 ($64_{decimal} = 01000000_{binario}$).

Ahora bien, **¿es necesario un rango mayor, o es suficiente con el dado?** Como se ha comentado, cada red (A, B, … J) necesita al menos *X+3* direcciones IP, donde *X* es el número de ordenadores conectados. Y siempre un valor que sea una potencia de 2 (superior o igual al valor anterior). Así, para A hará falta un rango con 64 direcciones, para B un rango con 32 direcciones, para D un rango con 16 valores, etcétera.

Direcciones necesarias para las redes A … J = 64 (A) + 32 (B) + 32 (C) + 16 (D) + 16 (E) + 16 (F) + 16 (G) + 32 (H) + 32 (I) + 64 (J) = **320 direcciones IP**

Además, **son necesarias las direcciones para las subredes entre *routers***. Los interfaces hacia las redes A, B, … J ya están incluidas en las 320 direcciones comentadas. Pero entre cada dos *routers* conectados se forma una subred. Cada una tendrá 2 direcciones (una por *router* conectado) más las direcciones de subred y de difusión, en total 4 direcciones (máscara /30).

Así, habrá 12 subredes con 4 direcciones (subred entre los *routers* A y B, entre los *routers* A y C, etcétera). En total 48 direcciones IP que se suman a las 320 anteriores, dando un **total de 368 direcciones IP**.

De esta forma, queda claro que **con el rango dado** (256 direcciones) **no es suficiente para soportar estas 368 direcciones**. A continuación, siguiendo los criterios comentados para la creación de subredes (primera dirección siempre con todos los bits a 0, última dirección con todos los bits a 1), se propone un ejemplo de asignación de direcciones. Existen múltiples soluciones válidas, siempre que respeten estos criterios. A modo de ejemplo, para la red A se especifican la

primera dirección asignable a un equipo, la última dirección asignable a un equipo, la dirección de subred y la dirección de difusión.

NOTA: A veces resulta más complejo y tedioso realizar la asignación de direcciones sobre el dibujo del ejercicio, ya que hay menos sitio, se puede olvidar más fácilmente un rango ya utilizado, etcétera. Pero es válido igualmente.

Red A → 10.10.10.0/26

- Dirección de subred: 10.10.10.0
- Dirección de difusión: 10.10.10.63
- Primera dirección asignable a un equipo: 10.10.10.1
- Última dirección asignable a un equipo: 10.10.10.62
- Número de direcciones asignables a equipos: 62 (64 menos las direcciones de subred y difusión)

Red B → 10.10.10.64/27
Red C → 10.10.10.96/27
Red D → 10.10.10.128/28
Red E → 10.10.10.144/28
Red F → 10.10.10.160/28
Red G → 10.10.10.176/28
Red H → 10.10.10.192/27
Red I → 10.10.10.224/27

A partir de aquí se termina el rango dado en el enunciado, y hay que utilizar rangos de direcciones consecutivas a éste:

Red J → 10.10.11.0/26

Las subredes entre *routers* se nombrarán con los *routers* que conectan:

RA – RB → 10.10.11.64/30
RA – RC → 10.10.11.68/30
RA – RJ → 10.10.11.72/30
RB – RD → 10.10.11.76/30
RB – RE → 10.10.11.80/30
RC – RF → 10.10.11.84/30
RC – RG → 10.10.11.88/30
RD – RH → 10.10.11.92/30
RE – RH → 10.10.11.96/30
RF – RI → 10.10.11.100/30
RG – RI → 10.10.11.104/30
RH – RJ → 10.10.11.108/30
RI – RJ → 10.10.11.112/30

De esta forma, se ha utilizado además el rango 10.10.11.0 → 10.10.11.115.

Apartado b. Muestre las tramas (y los campos relevantes) entre un ordenador que se autoconfigura en la red A (inalámbrica) y un servidor de HTTP en la red J. Hay un servidor de nombres en la subred D.

Posibles suposiciones:

- Los *routers* tienen sus tablas ARP actualizadas, y por tanto no necesitan realizar peticiones/respuestas para averiguar las direcciones físicas asociadas a las direcciones IP. O se hace esta suposición (razonable), o hay que incluir la primera petición/respuesta asociada a cada dirección IP.

- No es razonable suponer que ocurre lo mismo con el ordenador de la red A, ya que acaba de llegar y utilizará DHCP para autoconfigurarse. De esta forma, no ha tenido peticiones/respuestas ARP anteriores sobre el router inalámbrico. Sin embargo, el enunciado dice explícitamente que se muetren las tramas de nivel de red y superiores por lo que no se muestran las tramas ARP en este ejercicio (ARP está entre el nivel de enlace y el nivel de red).
- El protocolo DNS suele utilizar UDP (podría ser también TCP), con lo que no es necesario iniciar o cerrar la conexión. Es razonable suponer que tanto la petición como la respuesta (incluyendo las cabeceras de todos los protocolos) no tendrán un tamaño superior a una MTU (*maximum transfer unit*) de las redes implicadas.
- Se pueden incluir todos los mensajes necesarios del protocolo HTTP. Sin embargo, bastaría con enviar una petición de una página web, e.g. GET y recibir otra trama. Se puede suponer, por simplicidad, que la página web es sencilla y entra en una trama HTTP.

La forma más sencilla de realizar este apartado es utilizando una tabla. Algunos aspectos a considerar:

- Los procedimientos a seguir son los siguientes:

    1) Adquisición de la configuración (dirección IP, máscara, dirección IP del DNS, …) a través del protocolo DHCP.
    2) [Uso de ARP para conocer la dirección física del *router* inalámbrico → no es necesario por el enunciado.]
    3) Petición/respuesta DNS para conocer la dirección IP del servidor HTTP.
    4) Petición de una página web y respuesta. Previamente hay que realizar el establecimiento de una conexión TCP. Y posteriormente hay que realizar el cierre de dicha conexión.

Algunas aclaraciones antes de realizar este apartado:

- Las direcciones físicas (MAC) cambian salto a salto, identificando al origen y al destino en ese tramo de red.
- Cada interfaz de red tiene una dirección física diferente. Así, un *router* recibe una trama en un interfaz con una dirección, y la retransmite por un interfaz con una dirección diferente. No es correcto por tanto suponer siempre la misma dirección física para un *router* (o equipos con más de un interfaz de red).
- Las direcciones IP identifican el origen y el destino extremo a extremo. Durante las retransmisiones de una misma trama nunca cambian, salvo que se utilice NAT (*Network Address Translation*). El uso de NAT es necesario cuando una red privada se conecta a una red pública (e.g. Internet). En este ejercicio no sucede esto, ni el enunciado indica que se ningún *router* utilice NAT.
- Los *flags* y los números de secuencia y acuse son propios de TCP. Nunca se utilizan con UDP (no existen dichos campos en la cabecera).
- El concepto de puerto se utiliza a nivel de transporte, o sea, en las capas TCP y UDP. En las tramas que portan mensajes del nivel de aplicación se especifican los puertos porque dichos mensajes se envían en segmentos del nivel de transporte. Sin embargo, los protocolos de capas inferiores (e.g. ARP, ICMP, …) no utilizan puertos ya que sus mensajes no son enviados sobre ningún nivel de transporte.

Por sencillez en la lectura se van a utilizar etiquetas para nombrar a las direcciones físicas e IP implicadas. Se podría coger cualquier dirección asignada a las subredes como se explicó en el apartado anterior. Para los interfaces de red de los *routers* se considerará el interfaz ETH0 como el conectado a la red con los equipos (A, B, …) y los interfaces estarán ordenados respecto a éste moviéndose en el sentido de las agujas del reloj.

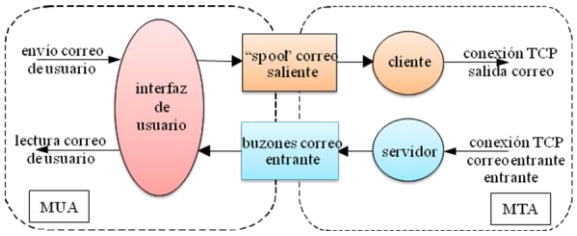
DIR. MAC ORIGEN	DIR. MAC DESTINO	DIR. IP ORIGEN	DIR. IP DESTINO	PUERTO ORIGEN	PUERTO DESTINO	FLAGS/SECUENCIA /ACUSE	TIPO DE MENSAJE
CONFIGURACIÓN MEDIANTE DHCP							
MAC_USUARIO	DIFUSIÓN	0.0.0.0	255.255.255.255	(*1) ASIGNADO POR S.O.	67	-	DHCP DISCOVER
MAC_SERV_DHCP	DIFUSIÓN	IP_SERV_DHCP	255.255.255.255	67	(*1)	-	DHCP OFFER
MAC_USUARIO	DIFUSIÓN	0.0.0.0	255.255.255.255	(*1)	67	-	DHCP REQUEST
MAC_SERV_DHCP	DIFUSIÓN	IP_SERV_DHCP	255.255.255.255	67	(*1)	-	DHCP ACK
CONSULTA/RESPUESTA DNS							
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_DNS	(*2) ASIGNADO POR S.O	53	-	CONSULTA DNS
MAC_RA_ETH2	MAC_RB_ETH3	IP_USUARIO	IP_SERV_DNS	(*2)	53	-	CONSULTA DNS
MAC_RB_ETH1	MAC_RD_ETH2	IP_USUARIO	IP_SERV_DNS	(*2)	53	-	CONSULTA DNS
MAC_RD_ETH0	MAC_SERV_DNS	IP_USUARIO	IP_SERV_DNS	(*2)	53	-	CONSULTA DNS
MAC_SERV_DNS	MAC_RD_ETH0	IP_SERV_DNS	IP_USUARIO	53	(*2)	-	RESPUESTA DNS
MAC_RD_ETH2	MAC_RB_ETH1	IP_SERV_DNS	IP_USUARIO	53	(*2)	-	RESPUESTA DNS
MAC_RB_ETH3	MAC_RA_ETH2	IP_SERV_DNS	IP_USUARIO	53	(*2)	-	RESPUESTA DNS
MAC_RA_ETH0	MAC_USUARIO	IP_SERV_DNS	IP_USUARIO	53	(*2)	-	RESPUESTA DNS
PETICIÓN Y RESPUESTA HTTP							
ESTABLECIMIENTO DE CONEXIÓN TCP							
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_HTTP	(*3) ASIGNADO POR S.O.	80	SYN, SEQ=X	PETICIÓN DE CONEXIÓN TCP
MAC_RA_ETH3	MAC_RJ_ETH2	IP_USUARIO	IP_SERV_HTTP	(*3)	80	SYN, SEQ=X	PETICIÓN
MAC_RJ_ETH0	MAC_SERV_HTTP	IP_USUARIO	IP_SERV_HTTP	(*3)	80	SYN, SEQ=X	PETICIÓN
MAC_SERV_HTTP	MAC_RJ_ETH0	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SYN, ACK, SEQ=Y, ACUSE=X+1	ACK + PETICIÓN EN OTRO SENTIDO
MAC_RJ_ETH2	MAC_RA_ETH3	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SYN, ACK, SEQ=Y, ACUSE=X+1	ACK + PETICIÓN
MAC_RA_ETH0	MAC_USUARIO	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SYN, ACK, SEQ=Y, ACUSE=X+1	ACK + PETICIÓN
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+1, ACUSE=Y+1	ACK DE PETICIÓN
MAC_RA_ETH3	MAC_RJ_ETH2	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+1, ACUSE=Y+1	ACK DE PETICIÓN
MAC_RJ_ETH0	MAC_SERV_HTTP	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+1, ACUSE=Y+1	ACK DE PETICIÓN

DIR. MAC ORIGEN	DIR. MAC DESTINO	DIR. IP ORIGEN	DIR. IP DESTINO	PUERTO ORIGEN	PUERTO DESTINO	FLAGS/SECUENCIA /ACUSE	TIPO DE MENSAJE
PETICIÓN Y RESPUESTA HTTP							
MENSAJES HTTP							
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_HTTP	(*3)	80	SEQ=X+1 ACUSE=Y+1	GET URI PROTOCOLO (NB1 BYTES)
MAC_RA_ETH3	MAC_RJ_ETH2	IP_USUARIO	IP_SERV_HTTP	(*3)	80	SEQ=X+1 ACUSE=Y+1	GET URI PROTOCOLO (NB1 BYTES)
MAC_RJ_ETH0	MAC_SERV_HTTP	IP_USUARIO	IP_SERV_HTTP	(*3)	80	SEQ=X+1 ACUSE=Y+1 ACK (de la petición)	GET URI PROTOCOLO (NB1 BYTES)
MAC_SERV_HTTP	MAC_RJ_ETH0	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SEQ=Y+1 ACUSE=X+1+NB1 ACK (de la petición)	ENVÍO DE PÁGINA WEB (NB2 BYTES)
MAC_RJ_ETH2	MAC_RA_ETH3	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SEQ=Y+1 ACUSE=X+1+NB1 ACK (de la petición)	ENVÍO DE PÁGINA WEB (NB2 BYTES)
MAC_RA_ETH0	MAC_USUARIO	IP_SERV_HTTP	IP_USUARIO	80	(*3)	SEQ=Y+1 ACUSE=X+1+NB1 ACK (de la petición)	ENVÍO DE PÁGINA WEB (NB2 BYTES)
CIERRE DE CONEXIÓN TCP							
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_HTTP	(*3)	80	FIN, SEQ=X, ACUSE=Y+1+NB2 ACK (de la respuesta)	PETICIÓN DE CONEXIÓN TCP
MAC_RA_ETH3	MAC_RJ_ETH2	IP_USUARIO	IP_SERV_HTTP	(*3)	80	FIN, SEQ=X, ACUSE=Y+1+NB2 ACK (de la respuesta)	PETICIÓN
MAC_RJ_ETH0	MAC_SERV_HTTP	IP_USUARIO	IP_SERV_HTTP	(*3)	80	FIN, SEQ=X, ACUSE=Y+1+NB2 ACK (de la respuesta)	PETICIÓN
MAC_SERV_HTTP	MAC_RJ_ETH0	IP_SERV_HTTP	IP_USUARIO	80	(*3)	FIN, ACK, SEQ=Y, ACUSE=X+2+NB1	ACK + PETICIÓN EN OTRO SENTIDO
MAC_RJ_ETH2	MAC_RA_ETH3	IP_SERV_HTTP	IP_USUARIO	80	(*3)	FIN, ACK, SEQ=Y, ACUSE=X+2+NB2	ACK + PETICIÓN
MAC_RA_ETH0	MAC_USUARIO	IP_SERV_HTTP	IP_USUARIO	80	(*3)	FIN, ACK, SEQ=Y, ACUSE=X+2+NB1	ACK + PETICIÓN
MAC_USUARIO	MAC_RA_ETH0	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+2+NB1, ACUSE=Y+2+NB2	ACK DE PETICIÓN
MAC_RA_ETH3	MAC_RJ_ETH2	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+2+NB1, ACUSE=Y+2+NB2	ACK DE PETICIÓN
MAC_RJ_ETH0	MAC_SERV_HTTP	IP_USUARIO	IP_SERV_HTTP	(*3)	80	ACK, SEQ=X+2+NB1, ACUSE=Y+2+NB2	ACK DE PETICIÓN

Ejercicio 2

Explique los mensajes para enviar/recibir un correo electrónico (IMAP para recepción), incluyendo los mensajes del protocolo DNS (resolución recursiva), entre dos MUAs.

Esta primera figura indica la relación entre el cliente de correo electrónico (MUA, *Mail User Agent*) y su estafeta de correo (máquina que actúa de servidor de correo electrónico, MTA, *Mail Transfer Agent*).

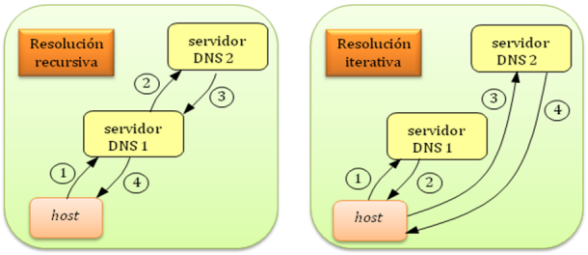


Esta segunda figura señala los protocolos utilizados para el envío y recepción de un correo electrónico:

Para el envío entre MUA y MTA siempre se utiliza el protocolo SMTP (*Simple Mail Transfer Protocol*), protocolo de aplicación que se transmite sobre el puerto 25 de TCP. Este protocolo se utiliza siempre también para el envío de mensajes entre estafetas (MTAs) de correo electrónico.

Para la recepción de correo electrónico, el destinatario se conecta a través de su cliente de correo electrónico (que implementa su MUA) a su servidor de correo (MTA), utilizando el protocolo POP (*Post Office Protocol*, puerto 110 de TCP) ó IMAP (*Internet Message Access Protocol*, puerto 143 de TCP). El primero suele utilizar una política *down-and-delete* (se descarga el correo del servidor y se borra de éste), mientras que el segundo suele trabajar directamente desde el servidor como si su información fuese local. En el enunciado se especifica que se utilice IMAP.

El enunciado también dice que se explicite el uso del protocolo DNS, y que éste usará resolución recursiva. El servicio DNS puede utilizar tanto TCP como UDP, siendo éste último el protocolo más habitual. En ambos casos se utiliza el puerto 53. La siguiente figura muestra el uso de los dos tipos posibles de resolución de nombres de dominio, recursiva e iterativa.



La **resolución recursiva** se caracteriza por el redireccionamiento de las peticiones/respuestas por parte de los diferentes servidores. Si un servidor DNS no conoce la respuesta, reenviará la petición a un DNS de nivel superior, y así sucesivamente hasta que alguno conozca la respuesta. En ese caso, la respuesta se reenviará a través de los servidores DNS preguntados.

La resolución iterativa se caracteriza por el protagonismo del cliente DNS. Si un servidor DNS no conoce la respuesta, envía un mensaje al cliente indicándole este hecho y cuál es su servidor

DNS de nivel superior. El cliente preguntará a este último servidor, y así sucesivamente hasta que alguno conozca la respuesta.

Después de estas generalidades, se muestra el proceso paso a paso del envío y recepción de un correo electrónico.

1. El usuario que escribe el correo electrónico (origen) utiliza para ello un cliente de correo electrónico. Rellenará los campos habituales (*from, to, subject, content, …*).
2. Cuando envía el correo (pulsa el botón "enviar" o realiza una acción similar), el cliente de correo electrónico comprueba la configuración de la cuenta de correo electrónico configurada. Entre otras cosas, tendrá configurado cuál es su servidor de correo electrónico entrante (e.g. imap.ugr.es) y su servidor de correo electrónico saliente (e.g. smtp.ugr.es). No es habitual que se especifiquen las direcciones IP, sino sus nombres de dominio.
3. Al no conocer la dirección IP del servidor de correo saliente, el MUA origen realizará una petición DNS (con resolución recursiva como se explicó previamente) para averiguar dicha dirección IP. Se preguntará por un servidor de correo electrónico (registro MX, *Mail eXchange*) de ese nombre de dominio. Se enviará una petición y se recibirá una respuesta.
4. Una vez conocida, el MUA origen enviará el correo electrónico a su servidor de correo electrónico (MTA), utilizando el protocolo SMTP. Los mensajes más importantes son (véase las transparencias 41, 42 y 43 del tema 4):
    - HELO: indica quién es el *host* cliente
    - MAIL: indica quién es el remitente
    - RCPT: indica quién es el destinatario
    - DATA: envía la información del mensaje
    - QUIT: finaliza la sesión
    - Las respuestas a estos mensajes siguen el esquema habitual *<XYZ texto explicativo>*, donde XYZ es un número de 3 cifras que indica el estado de la transacción.

    De esta forma, su MTA (origen) recibe el mensaje y lo pone en su *spool* de correo saliente.
5. El MTA origen enviará este correo al MTA del destinatario. Para ello, comprueba el nombre de dominio de la cuenta de correo del destinatario (e.g. si la cuenta es *destino@dominiodestino.es*, el nombre de dominio sería *dominiodestino.es*). Como no conoce la dirección IP del servidor de correo electrónico de ese dominio, realizará una petición DNS (nuevamente recursiva) para averiguarla (registro *MX* del dominio *dominiodestino.es* en el ejemplo). Recibirá la respuesta DNS y así ya conocerá la dirección IP del servidor de correo electrónico del destinatario.
6. El MTA origen envía el correo al MTA destino utilizando el protocolo SMTP, usando los mensajes anteriormente comentados. El MTA destino colocará el mensaje en el buzón de correo entrante de ese usuario (previamente se ha comprobado que es un usuario válido en ese servidor), a la espera de que éste se conecte y solicite sus correos.
   NUNCA EL SERVIDOR ENVÍA EL CORREO DIRECTAMENTE AL DESTINATARIO. EL USUARIO SIEMPRE ABRIRÁ SU CLIENTE Y SOLICITARÁ DESCARGARSE LOS CORREOS. El concepto de servidor de correo tiene sentido porque los usuarios no están conectados todo el tiempo, y solicitarán asíncronamente (cuando quieran) recibir sus correos.
7. Cuando el destinatario abra su cliente de correo electrónico (que implementa su MUA), solicitará recuperar sus correos electrónicos. En este caso (ya que lo dice el enunciado) utilizará IMAP, por lo que verá los correos electrónicos pero éstos seguirán en el servidor, no en la máquina del destinatario.
8. De nuevo, el usuario no conoce la dirección IP de su servidor de correo electrónico. En este caso preguntará por su servidor de correo entrante (por ejemplo, imap.dominiodestino.es). Nuevamente se usará resolución recursiva, y se obtendrá la dirección IP de su servidor.
9. Una vez obtenida su IP, se solicitará recibir sus correos electrónicos mediante el protocolo IMAP. Los mensajes más habituales de IMAP son los siguientes:
    - LOGIN: para autenticarse en el servidor

- SELECT: se selecciona el buzón correspondiente
- FETCH: obtiene datos de los mensajes
- STORE: modifica datos de los mensajes (e.g. mensaje leido, mensaje borrado, …)
- LOGOUT: se desconecta del servidor

De esta forma, el MUA origen envía un correo y el MUA destino lo recibe.

Ejercicio 3

Que algo sea "adaptable" significa que se adapta. ¿A qué? Lógicamente a los cambios (si no cambiara nada, no habría a qué adaptarse). Los procedimientos más importantes que hacen que TCP sea adaptable son:

a) **Control de flujo**. Gracias a la ventana ofertada (y la ventana útil), ***el emisor se adapta al espacio disponible en el buffer del receptor***. Así, no se envían mensajes que el receptor tendría necesariamente que descartar (no queda espacio en su memoria). Y **mejora el rendimiento** de la red porque no se transmiten mensajes innecesarios (reduciendo una posible congestión y evitando desperdiciar ancho de banda).

b) **Control de congestión**. Gracias a la ventana de congestión (y los mecanismos asociados como slow start, congestion avoidance…) ***el emisor se adapta a la capacidad del canal de transmisión***. TCP utiliza todo el canal disponible, y usa el control de congestión para adaptarse. Por ejemplo, si se utiliza una red de 10Mbps, TCP se adaptará a dicha velocidad. Si es de 100Mpbs, se ajustará a esta velocidad gracias al control de congestión. TCP aumenta su velocidad hasta que detecta la congestión (al no recibir mensajes ACK) y reduce la velocidad. Según el tipo de TCP (Tahoe, Vegas, Reno…) lo realiza de una manera diferente, pero siempre se adapta a la capacidad del canal.

   Es evidente la relación entre el rendimiento y el control de congestión. Si no hubiese control de congestión, TCP no podría modificar su velocidad de transmisión y adaptarse al ancho de banda del canal. Esto es lo que ocurre con el protocolo UDP, cuya velocidad depende de la aplicación (e.g. transmisión de vídeo) y no de la red (no se puede adaptar a su ancho de banda).

   NOTA: La *"congestión"* significa que los *routers* se quedan sin espacio libre en sus *buffers* y empiezan a descartar los nuevos paquetes (por falta de memoria para almacenarlos). A veces se utiliza la expresión "la red se congestiona" sin saber qué significa exactamente.

c) **Adaptación de los *timeouts*** (tiempos de expiración de los temporizadores) asociados a la recepción de mensajes ACKs **en el control de errores**. Es evidente que el control de errores es muy importante en TCP, ya que permite garantizar que los datos llegan correctamente (para evitar los efectos de que lleguen desordenados se utiliza el control de flujo). En el control de errores, es muy importante el tiempo asignado a los temporizadores, que depende del *round trip time* (RTT). En función del destino (y las redes atravesadas) ***hay que adaptarse al nuevo RTT***, que variará con el tiempo ya que las condiciones de red varían. Si el valor del *timeout* fuese muy pequeño, no daría tiempo a que llegase el mensaje ACK. Y si fuese muy grande, se tardaría demasiado en reaccionar por la falta de un mensaje ACK. Por tanto, hay que adaptarse a las variaciones de RTT en la red para que el control de errores funcione de forma eficiente.

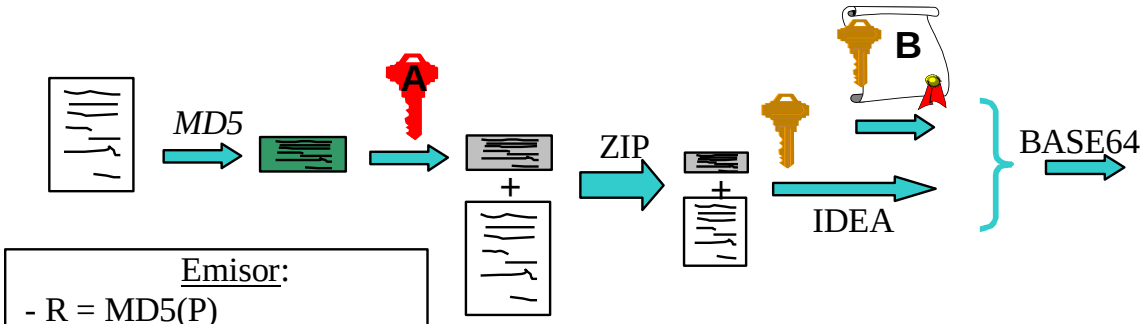
   En el primer caso (*timeout* muy pequeño) se dispararían los mecanismos del control de congestión (e.g. inicio lento en TCP Tahoe) reduciéndose drásticamente la velocidad de transmisión, afectando así al rendimiento. En el segundo caso, se tardaría en detectar la congestión (descarte de paquetes en los *routers* por falta de espacio libre en sus *buffers*),

por lo que muchos paquetes serían descartados antes de que se hiciese nada por solucionarlo. Este descarte implica la retransmisión de esos paquetes (control de errores), por lo que también se reduciría la velocidad de transmisión efectiva y bajaría el rendimiento de la red.

Otros aspecto menos importante pueden ser el uso de la *ventana escalada*, que permite que TCP se adapte a redes de alta velocidad (escalado de la ventana ofertada). Puede haber otros aspectos que tengan un menor impacto en la adaptabilidad de TCP.

Ejercicio 4 (PGP)

Un esquema del funcionamiento de PGP (transparencias) sería (el dibujo sólo representa la parte emisora, la parte receptora sería igual pero en orden inverso y con funcionalidad inversa, como se comenta debajo):



Emisor:
- R = MD5(P)
- FD = $Kpr_A(R)$
- Z = ZIP(FD + P)
- C = $IDEA_K(Z)$ + $Kpu_B(K)$
- M = B64(C)

Receptor:
- C = $B64^{-1}(M)$
- K = $Kpr_B(Kpu_B(K))$
- Z = $IDEA_K^{-1}(IDEA_K(Z))$
- FD + P = $ZIP^{-1}(Z)$
- R = $Kpu_A(FD)$
- R' = MD5(P)
- R' = R ??

Proceso en el emisor:

1. A partir de un texto plano *P* se obtiene su resumen *R* usando la función *hash* MD5.
2. Se cifra con la clave privada del emisor (Kpr_A), creándose así una firma digital FD.
   - Sólo el emisor puede haberlo cifrado con su clave privada → **autenticidad + no repudio**.
   - La firma digital (resumen cifrado con la clave privada del emisor) garantiza que el mensaje no puede ser modificado sin que el receptor lo perciba → **integridad**.
3. La firma digital se comprime (mediante el algoritmo ZIP) junto con el mensaje en texto plano. La compresión no afecta a la seguridad.
4. El resultado se cifra mediante un algoritmo de clave secreta (IDEA). La clave K utilizada se cifra con la clave pública del receptor Kpu_B.
   - Sólo el receptor puede obtener la clave secreta, ya que va cifrada con su clave pública (y se necesita su clave secreta para descifrar).
   - El mensaje enviado va cifrado → **confidencialidad**.
5. Finalmente se codifica mediante la codificación BASE64. Esta codificación no afecta a la seguridad.

Proceso en el receptor:

Se realiza el proceso contrario (orden inverso y operación inversa).

1. Se decodifica la codificación BASE64.
2. Se obtiene la clave secreta, descifrando con la clave privada del receptor.
3. Se descifran los datos comprimidos (ZIP) usando dicha clave secreta.
4. Se descomprime mediante el algoritmo ZIP, obteniéndose la firma digital y el texto plano.
5. Se descifra la firma digital con la clave pública del emisor, obteniéndose el resumen original R.
6. Con el texto plano, se calcula un nuevo resumen R' usando la función *hash* MD5.
7. Se compara R con R'. Si son iguales, el mensaje no ha sido modificado. Si son diferentes, el mensaje ha sido modificado desde su emisión.

Resumiendo, el uso de PGP garantiza los siguientes aspectos de seguridad: **confidencialidad** (nadie externo puede ver el mensaje en texto plano), **autenticación** (hay seguridad sobre la identidad del emisor), **integridad** (el mensaje no ha sido modificado) **y no repudio** (hay pruebas, la firma digital, de que el emisor participó en la transacción).