



Universidad de Granada
Departamento de Teoría de la Señal,
Telemática y Comunicaciones



FUNDAMENTOS DE REDES

– 3er. curso del Grado de Ingeniería Informática –
Examen de teoría – Enero 2013

Apellidos y nombre: _____ Grupo: _____

1. (1 pto.: 10x0,1) Marque como verdaderas (V) o falsas (F) las siguientes afirmaciones:

(Nota: una respuesta errónea anula una correcta)

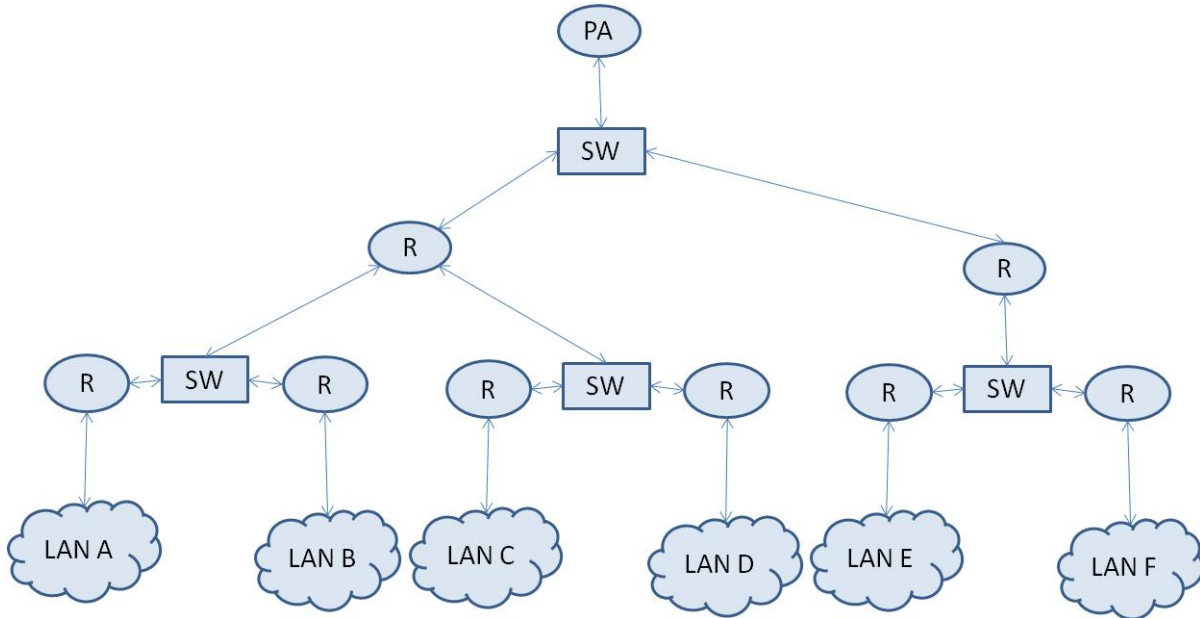
		V	F
a)	El protocolo Secure Socket Layer es exclusivo para HTTPS.	<input type="checkbox"/>	<input type="checkbox"/>
b)	Un servicio confirmado debe ser necesariamente orientado a conexión.	<input type="checkbox"/>	<input type="checkbox"/>
c)	El estándar sugiere obtener el ISN en TCP a partir de un temporizador.	<input type="checkbox"/>	<input type="checkbox"/>
d)	El checksum en TCP y UDP se calcula únicamente sobre el valor de las cabeceras de dichos protocolos.	<input type="checkbox"/>	<input type="checkbox"/>
e)	El three-way handshake en TCP no es el único mecanismo para establecer la conexión.	<input type="checkbox"/>	<input type="checkbox"/>
f)	Las aplicaciones de transferencia de ficheros no son sensibles a requisitos temporales.	<input type="checkbox"/>	<input type="checkbox"/>
g)	La técnica de ventana deslizante es una solución al problema de la ventana tonta.	<input type="checkbox"/>	<input type="checkbox"/>
h)	Los servidores DNS Top-Level Domain también son llamados servidores raíz o punto.	<input type="checkbox"/>	<input type="checkbox"/>
i)	El campo Time To Live (TTL) en IP especifica el número de segundos tras los cuales se descarta un paquete en la red..	<input type="checkbox"/>	<input type="checkbox"/>
j)	Uno de los protocolos de comunicación entre servidores de correo es POP3.	<input type="checkbox"/>	<input type="checkbox"/>

2. (1 pto.) Al inicio de una conexión TCP, en una línea sin congestión con 10 ms de tiempo de propagación y 10 Mbps de velocidad de transmisión, ¿cuánto tiempo se emplea en enviar y recibir confirmación de 30 KB con las siguientes asunciones (añada cualquier asunción adicional que crea conveniente)? Realice el diagrama de tiempos de la transmisión.

- a) Ventana ofertada de control de flujo de 10 KB
- b) Todos los segmentos se ajustan a un MSS de 2KB
- c) Umbral de congestión de 8 KB
- d) Respuesta ACK retardada en el receptor de acuerdo a la teoría.

3. (1.5 pto.) Se desea transmitir un mensaje de M bits entre dos estaciones origen y destino separadas entre sí S enlaces, sobre una red de conmutación de paquetes mediante datagramas. D es el retardo de propagación en cada línea (en s.), R el tiempo de procesamiento en cada nodo (en s.) y P la longitud total de cada paquete (en bits), con H bits de cabecera. Calcule el tiempo total involucrado en la transmisión del mensaje M si se supone que la velocidad de cada enlace (expresada en bps) es tal que $V_1 < V_2 < \dots < V_S$.

4. (1.5 pts) Asigne las direcciones de subred en la siguiente topología a partir de 192.168.0.0 de forma que el número de entradas en la tabla de encaminamiento de PA, incluyendo las redes directamente conectadas, sea mínimo. Asuma que en las redes LAN puede haber hasta 25 PCs y que la dirección pública del PA pertenece a la red 200.200.200.200/30.



Ejemplo de examen.

Ejercicio 1.

a) El protocolo Secure Socket Layer es exclusivo para HTTPS.

F. Tema 2. Transparencia de protocolos seguros. También para IMAPS, SSL-POP.

b) Un servicio confirmado debe ser necesariamente orientado a conexión.

F. Tema 1. Transparencia Terminología y servicios. Hay 4 tipos.

c) El estándar sugiere obtener el ISN en TCP a partir de un temporizador.

V. Tema 3. Transparencia Transmission Control Protocol (TCP). Es teóricamente al azar, pero se sugiere usar un contador entero.

d) El checksum en TCP y UDP se calcula únicamente sobre el valor de las cabeceras de dichos protocolos.

F. Tema 3. Transparencia Transmission Control Protocol (TCP) control de errores de flujo. El campo de comprobación: Tanto en TCP como en UDP, el checksum se calcula con la cabecera, la pseudo-cabecera IP y los datos.

e) El three-way handshake en TCP no es el único mecanismo para establecer la conexión.

V. Tema 3. Transparencia Transmission Control Protocol (TCP) autómata de estados finitos TCP. Ejercicio 7 de la relación 3. Se puede realizar la conexión normal y otra forma es la conexión simultánea.

f) Las aplicaciones de transferencia de ficheros no son sensibles a requisitos temporales.

V. Tema 3. Transparencia User Datagram Protocol (UDP) aplicación/servicio TFTP. En general, las aplicaciones de T.F. son no tolerantes a fallos y no sensibles a retardos.

g) La técnica de ventana deslizante es una solución al problema de la ventana tonta.

F. Tema 3. Transparencia Transmission Control Protocol (TCP). Posible mejora: la ventana optimista. La ventana deslizante es una solución de control de flujo y no al problema de la ventana tonta.

h) Los servidores DNS Top-Level Domain también son llamados servidores raíz o punto.

F. Tema 2. Transparencia Servicio de Nombres de Dominio (DNS). Resolución distribuida: 1-Servidores punto. 2-Servidores de dominio (Top-Level Domain o TLD). 3-Servidores Locales.

i) El campo Time To Live (TTL) en IP especifica el número de segundos tras los cuales se descarta un paquete en la red.

F. Tema 4. Va por saltos, restando 1 al campo TTL en cada salto. Hacerlos por tiempo sería más complejo ya que se tendría que calcular el tiempo de propagación, de transmisión...

j) Uno de los protocolos de comunicación entre servidores de correo es POP3.

F. Tema 2. Transparencia El Correo Electrónico. El dibujo indica que entre servidores es el protocolo SMTP pero entre el servidor y el usuario se utiliza el POP3, SMTP o HTTP.

Ejercicio 2.

2. (1 pto.) Al inicio de una conexión TCP, en una línea sin congestión con 10 ms de tiempo de propagación y 10 Mbps de velocidad de transmisión, ¿cuánto tiempo se emplea en enviar y recibir confirmación de 30 KB con las siguientes asunciones (añada cualquier asunción adicional que crea conveniente)? Realice el diagrama de tiempos de la transmisión.

- a) Ventana ofertada de control de flujo de 10 KB
- b) Todos los segmentos se ajustan a un MSS de 2KB
- c) Umbral de congestión de 8 KB
- d) Respuesta ACK retardada en el receptor de acuerdo a la teoría.

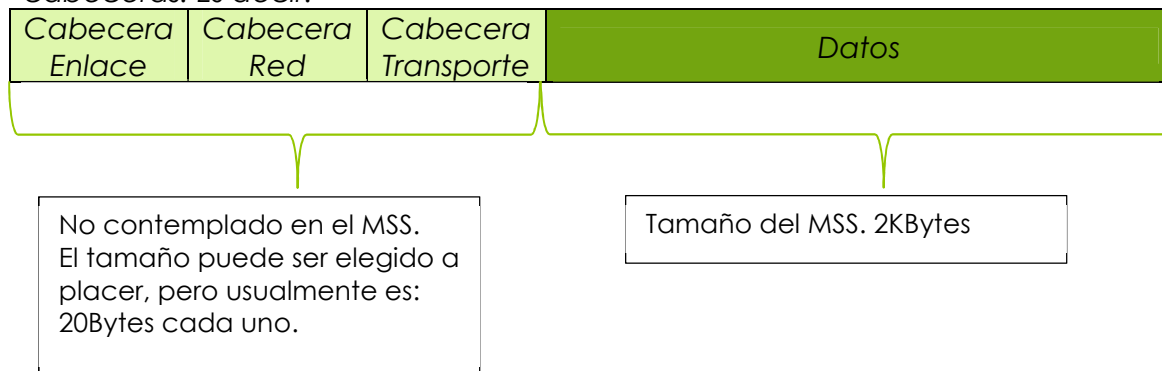
Lo primero a tener en cuenta es que se nos pide cuánto se tarda en **enviar y recibir confirmación** luego tenemos que calcular hasta que se realiza la recepción del envío completa.

Se comienza con inicio lento (con 1 MSS).

$$T_t = \frac{\text{Tamaño de los datos}}{\text{Velocidad}} = \frac{\text{MSS}}{V_t} \quad T_p = 10\text{ms}$$

Sea T_p tiempo de propagación.
 Sea T_t tiempo de transmisión.
 Sea V_t velocidad de transmisión.

Asunción 1a. Se asume que 1 MSS es la parte solo de los datos, faltan las cabeceras. Es decir:



Luego el

$$T_t = \frac{\text{MSS} + 60\text{Bytes}}{V_t}$$

Asunción 1b. Se asume que 1 MSS es la parte solo de los datos, pero se consideran que el tamaño de las cabeceras es despreciable.

$$T_t = \frac{\text{MSS}}{V_t} = \frac{2\text{KB} \cdot 1024\text{Bytes/KBytes} \cdot 8\text{bits/Byte}}{10^7 \text{ bits/seg}} = 1,64\text{ms} = T_t$$

Se continuará la resolución del ejercicio tomando la **asunción 1b.**

Entonces si tamaño_{cabeceras} = 0Bytes → $T_{t_{ack}} = 0$

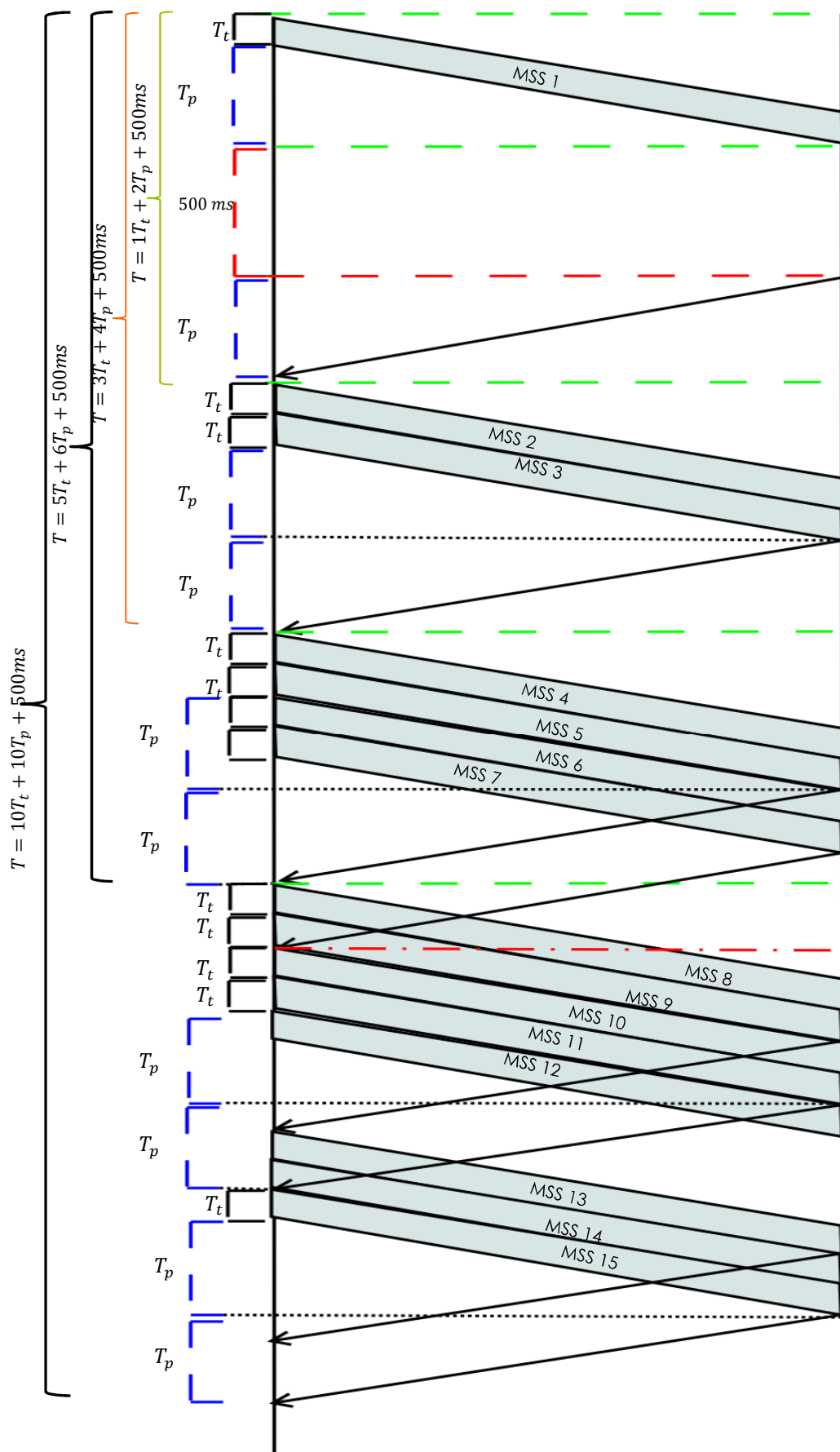
Más datos de que disponemos son:

$$\begin{aligned} V_{CF} &= 10\text{KB} \\ \text{MSS} &= 2\text{KB} \\ U_c &= 8\text{KB} \end{aligned}$$

Nótese:
 V_{CF} Ventana de control de flujo.
 U_c Umbral de congestión.

Emisor

Receptor



Inicio de conexión. Inicio lento.

$V_c = 1 \text{ MSS} = 2 \text{ KB}$

$V_E = \min(U_c, V_c) = 2 \text{ KB}$

Mirando cuando se envía y cuando llega el último bit se observa el tiempo de propagación.

Tiempo de espera. 500 ms porque se espera a que se reciba el siguiente paquete. Para enviar 1 ACK acumulando la confirmación del 1^{er} y 2^o paquetes. Pero no se envía porque $V_E = 2 \text{ KB}$ es decir solo se puede mandar $2 \text{ KB} = 1 \text{ MSS}$

Envío de ACK.

Como su tamaño es 0 no hay T_t solo T_p

Recepción de ACK.

Al recibir el ACK se aumenta la ventana en

1 MSS = 2 KB con lo que tenemos $V_c = 4 \text{ KB}$

Y podemos enviar exactamente 2 MSS = 4 KB

Recepción de ACK.

Al recibir el ACK de ya 4 KB es decir otra vez la ventana entera entonces se vuelve a doblar el tamaño de la ventana.

Y se envía 8 KB.

$V_c = 8 \text{ KB} = U_c$ como tenemos que la ventana es igual al umbral pasamos a **evitación de congestión** es decir hay que esperar a recibir la confirmación de la ventana entera para poder ampliar esta en 1 MSS.

Envío de ACK.

Aquí se manda la confirmación de 2 MSS una vez que ya han llegado.

Posteriormente se envía la de los otros dos (7 y 6)

Recepción de ACK.

Al recibir el ACK se hace un hueco en la ventana lo cual nos permite poder mandar 4 KB más, que es lo confirmado.

Recepción de ACK.

Al recibir este ACK se hace la $V_c = 10 \text{ KB}$ porque ya se confirman los 8 KB de la ventana anterior (ventana entera). Y por tanto se aumenta la ventana en 1 MSS = 2 KB

Recepción de ACK.

Como se envió la ventana entera, no se puede enviar nada hasta no recibir algún ACK, ahora se recibe el ACK de los MSS 8 y 9. Lo cual hace un hueco en la ventana de $4 \text{ KB} = 2 \text{ MSS}$ y por tanto se envían.

Recepción de ACK.

Ahora se recibe el ACK de los MSS 10 y 11. Lo cual hace un hueco en la ventana de $4 \text{ KB} = 2 \text{ MSS}$. Pero como solo queda por mandar 1 MSS ya que **nos indican en el enunciado 30 KB** entonces $30 \text{ KB} = 15 \text{ MSS}$.

Recepción de ACK.

Ahora se recibe el ACK de los MSS 12 y 13. Haría un hueco en la ventana pero no hay nada que mandar.

Recepción de último ACK.

Ahora se recibe el ACK de los MSS 14 y 15. Y con esto finalizaríamos el envío y confirmación de los paquetes.

Si vemos en el margen izquierdo podemos observar que la cuenta en tiempo nos ha llevado a un total de

$$T_{total} = 10 T_t + 10 T_p + 500 ms$$

Téngase en cuenta que $T_p = \frac{Distancia}{V_{propagacion}}$ donde $V_{propagación} = 3 \cdot 10^8$ en el vacío, $V_{propagación} = 2,8 \cdot 10^8$ en el aire y $V_{propagación} = 2 \cdot 10^8$ por cable.

Ejercicio 3.

3. (1.5 pts) Se desea transmitir un mensaje de M bits entre dos estaciones origen y destino separadas entre sí S enlaces, sobre una red de conmutación de paquetes mediante datagramas. D es el retardo de propagación en cada línea (en s.), R el tiempo de procesamiento en cada nodo (en s.) y P la longitud total de cada paquete (en bits), con H bits de cabecera. Calcule el tiempo total involucrado en la transmisión del mensaje M si se supone que la velocidad de cada enlace (expresada en bps) es tal que $V_1 < V_2 < \dots < V_S$.

Estamos en la capa de red y los datos aquí a tener en cuenta son:

Mensaje $\rightarrow M$ bits

Enlaces $\rightarrow S$ enlaces

Tiempo de propagación $\rightarrow T_{prop} = D$ seg

Tiempo de procesamiento $\rightarrow T_{proc} = R$ seg

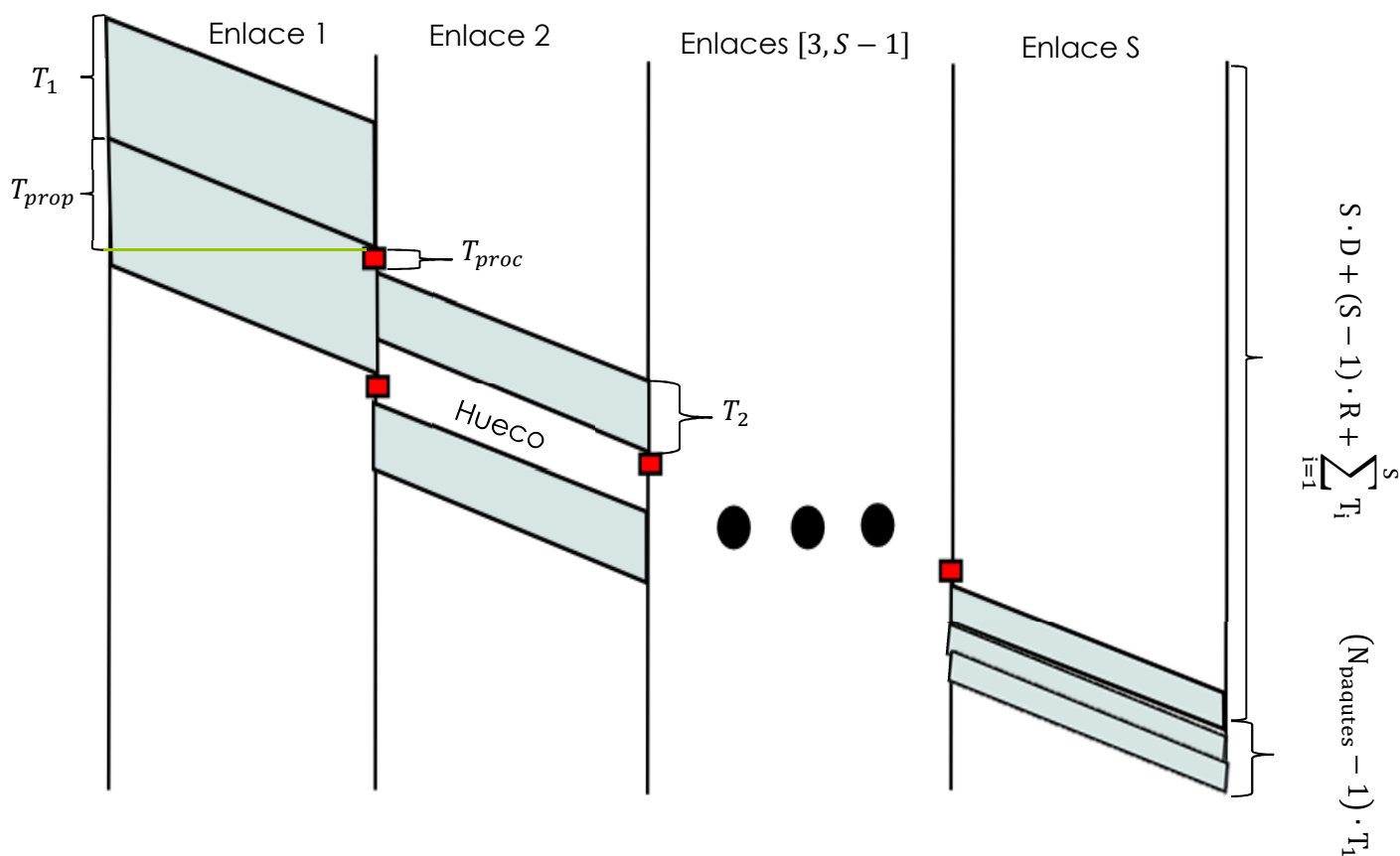
Longitud de paquete $\rightarrow P$ bits

Cabeceras $\rightarrow H$ bits donde se ha de tener en cuenta que los H bits están incluidos en los P bits. $\rightarrow H \subset P$

También se ha de tener en cuenta que $V_1 < V_2 < V_3 \dots < V_S$ y por lo tanto si la velocidad es mayor el **tiempo de transmisión** será más chico $\rightarrow T_1 > T_2 > \dots > T_S$

Asunción 1. Se puede realizar procesamiento en paralelo.¹

Asunción 2. NO se puede realizar propagación en paralelo.



¹ En realidad, siempre asumimos que el procesamiento se puede hacer en paralelo, aunque para muchos problemas no es necesario. Esto tiene sentido, considerando que el tiempo de procesamiento incluye tiempos de cola, reserva de recursos, etc. El tiempo de transmisión, en la salida de un nodo a un enlace nunca puede ser en paralelo.

Para el tiempo total, lo que hay que tener en cuenta es que el tiempo que más hace influencia es el más lento, en este caso T_1 luego todos los paquetes tienen que tardar mínimo $(N_{\text{paquetes}} - 1) \cdot T_1$

Además se tendría que añadir el tiempo que tarda en propagarse por todos los enlaces $S \cdot D$, y el tiempo de procesamiento de los $S - 1$ paquetes: $(S - 1) \cdot R$.

Y finalmente el tiempo de transmisión acumulativo de todos los enlaces:

$$\sum_{i=1}^S T_i$$

Lo que haría un total de:

$$(N_{\text{paquetes}} - 1) \cdot T_1 + S \cdot D + (S - 1) \cdot R + \sum_{i=1}^S T_i$$

Hemos de tener en cuenta que

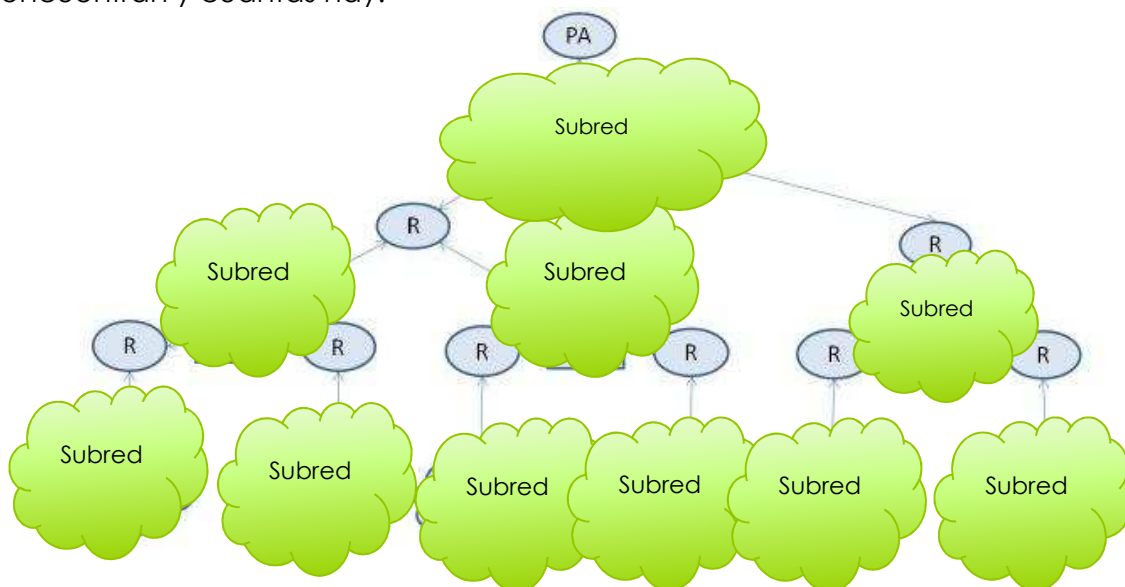
$$N_{\text{paquetes}} = \left\lceil \frac{M}{P - H} \right\rceil$$

Porque tenemos que dividir el mensaje grande en $P - H$ paquetes chicos ya que se tiene que excluir las cabeceras de los mismos, porque éstas tienen obligatoriamente que incluirse en todos los paquetes, haciendo en resumidas cuentas una parte útil, para meter el paquete de M bits, de $P - H$ bits.

Ejercicio 4.

4. (1.5 pts) Asigne las direcciones de subred en la siguiente topología a partir de 192.168.0.0 de forma que el número de entradas en la tabla de encaminamiento de PA, incluyendo las redes directamente conectadas, sea mínimo. Asuma que en las redes LAN puede haber hasta 25 PCs y que la dirección pública del PA pertenece a la red 200.200.200.200/30.

Lo primero que vamos a hacer es identificar las subredes que se encuentran y cuántas hay.

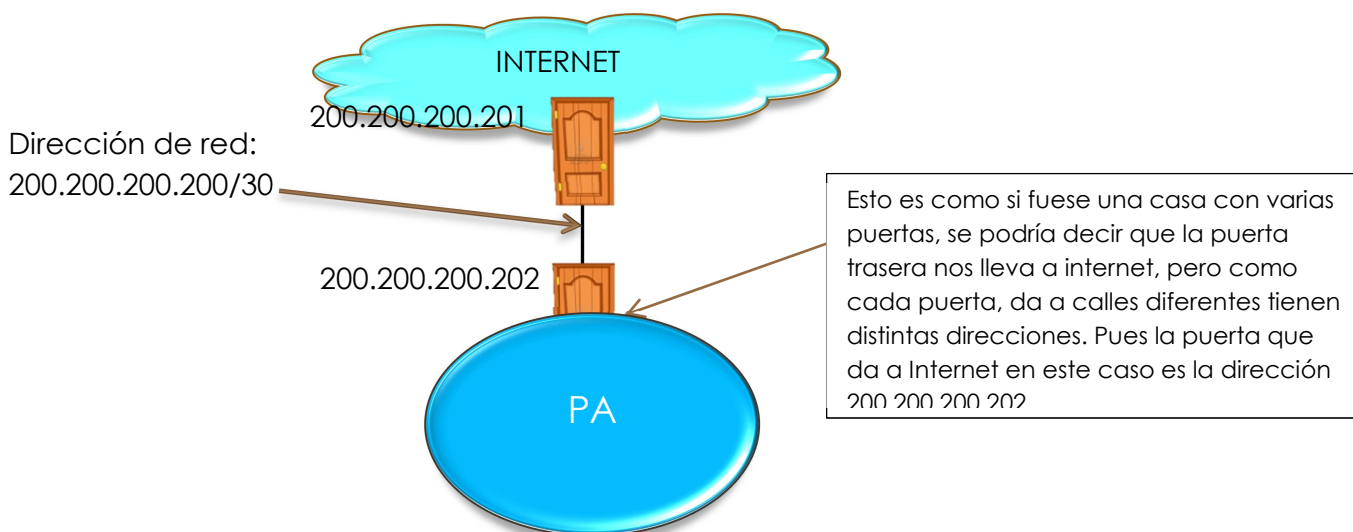


Y si las contamos obtendríamos un total de 10 subredes.

Nótese que las redes LAN pueden tener hasta 25 PCs.

Ahora vamos a hacer una *asunción* sobre con quién está conectado con PA, como se nos dice que la dirección pública de PA pertenece a la red 200.200.200.200/30.

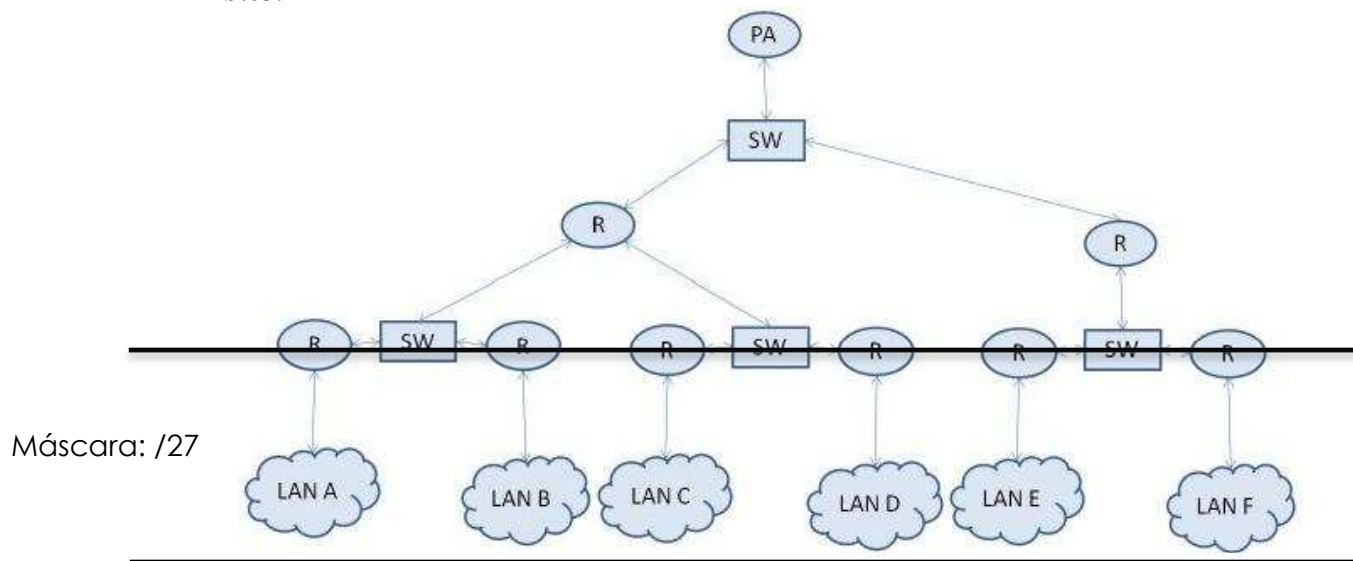
A continuación se elige una de las posible formas.



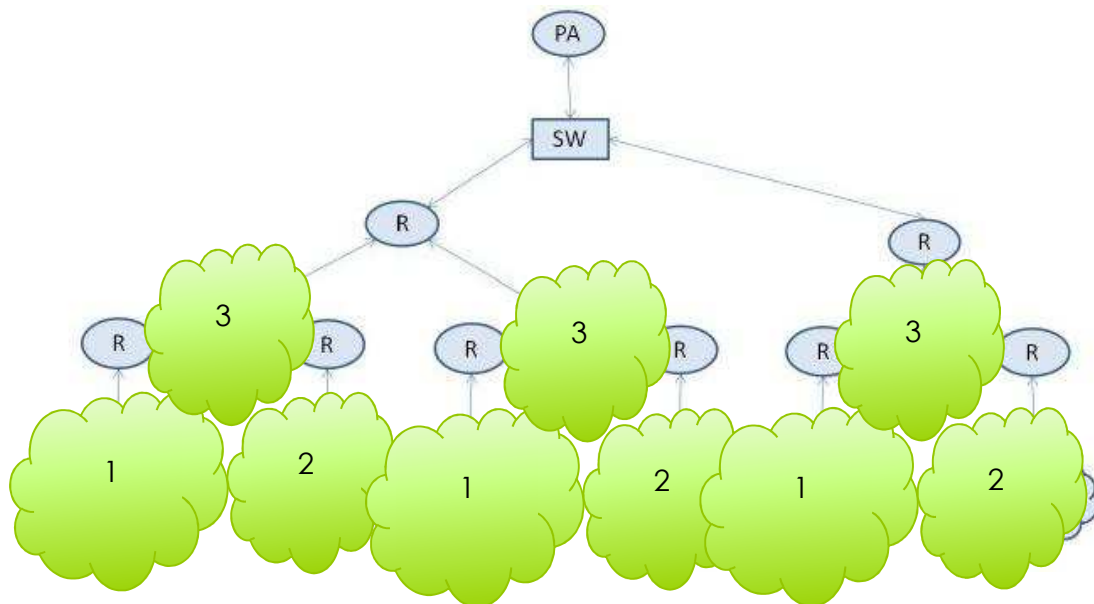
Lo siguiente a realizar va a ser seleccionar las máscaras de subred teniendo en cuenta el número de redes, dispositivos... que hay que poder direccionar.

Por ejemplo, como hay hasta 25 PC en las LAN tenemos que poder direccionar de forma inequívoca a los 25 (tienen que tener distinta dirección), pero además se reservan 1 dirección para el router, y 2 direcciones reservadas, lo que hace un total de $25 + 2 + 1 = 28$ direcciones.

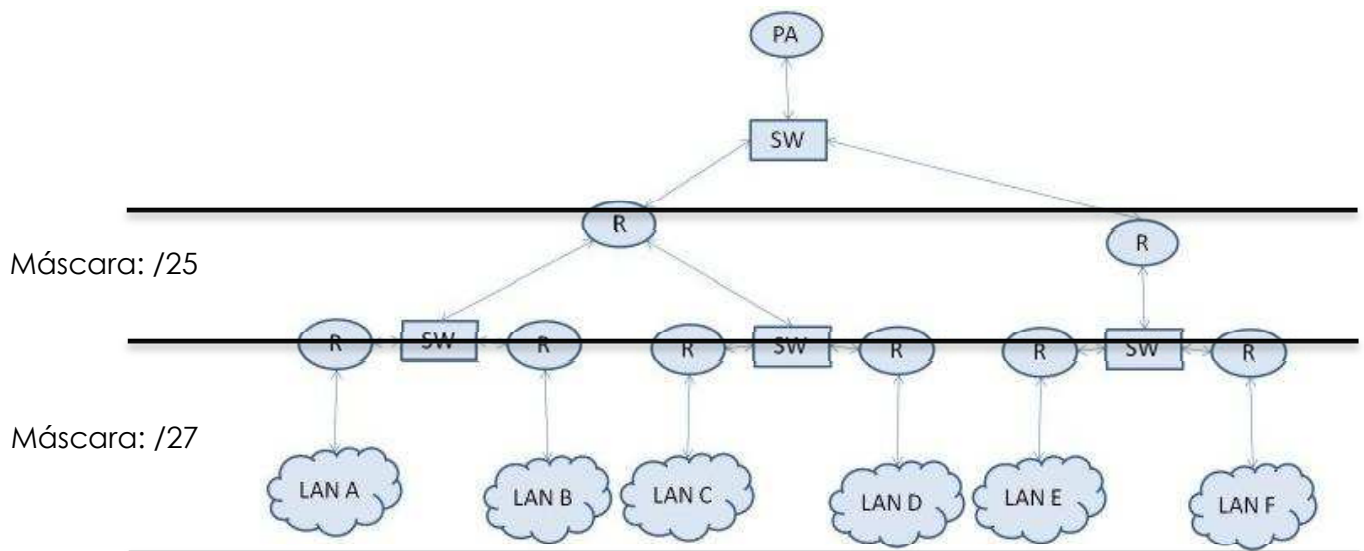
Y para ello necesitamos: $\lceil \log_2 28 \rceil = 4.8 \rightarrow 5$ bits. Pero contamos con un máximo de 32 bits para direcciones. Luego la máscara será de $32 - 5 = 27$ bits.



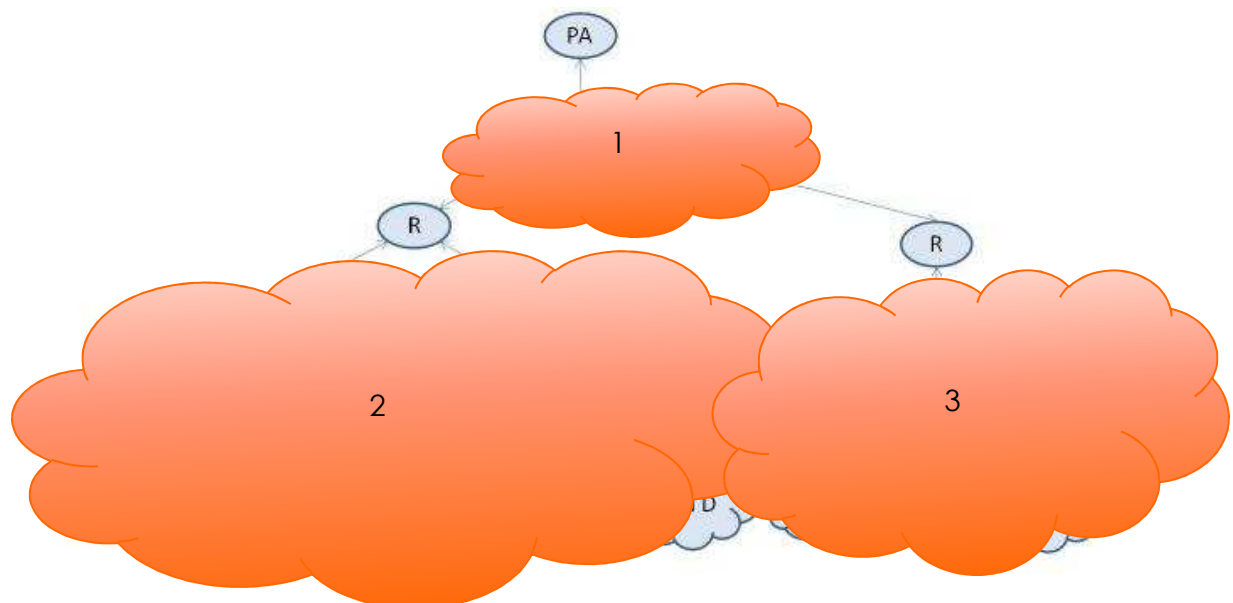
Ahora vamos al piso superior. Como hay 3 subredes distintas:



Entonces necesitamos $\lceil \log_2 3 \rceil = 2$ bits luego necesitaríamos una máscara con dos bits menos a 1. Es decir una máscara $/27 - 2$ bits = $/25$ lo que nos quedaría:

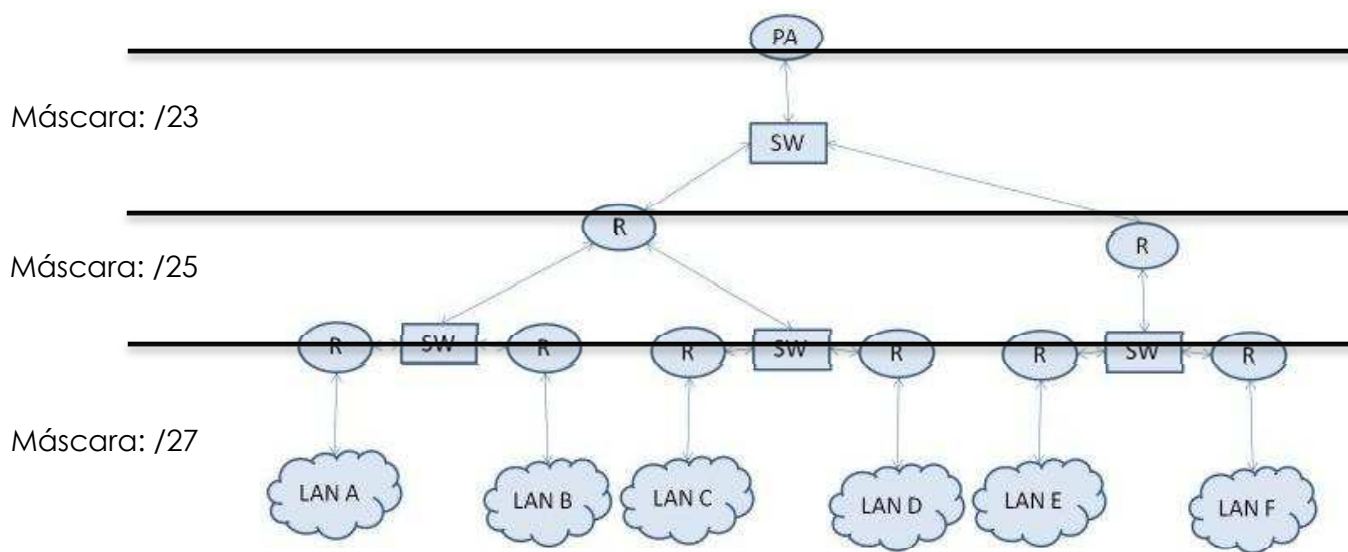


La última máscara tendrá que poder direccionar otras 3 redes distintas:



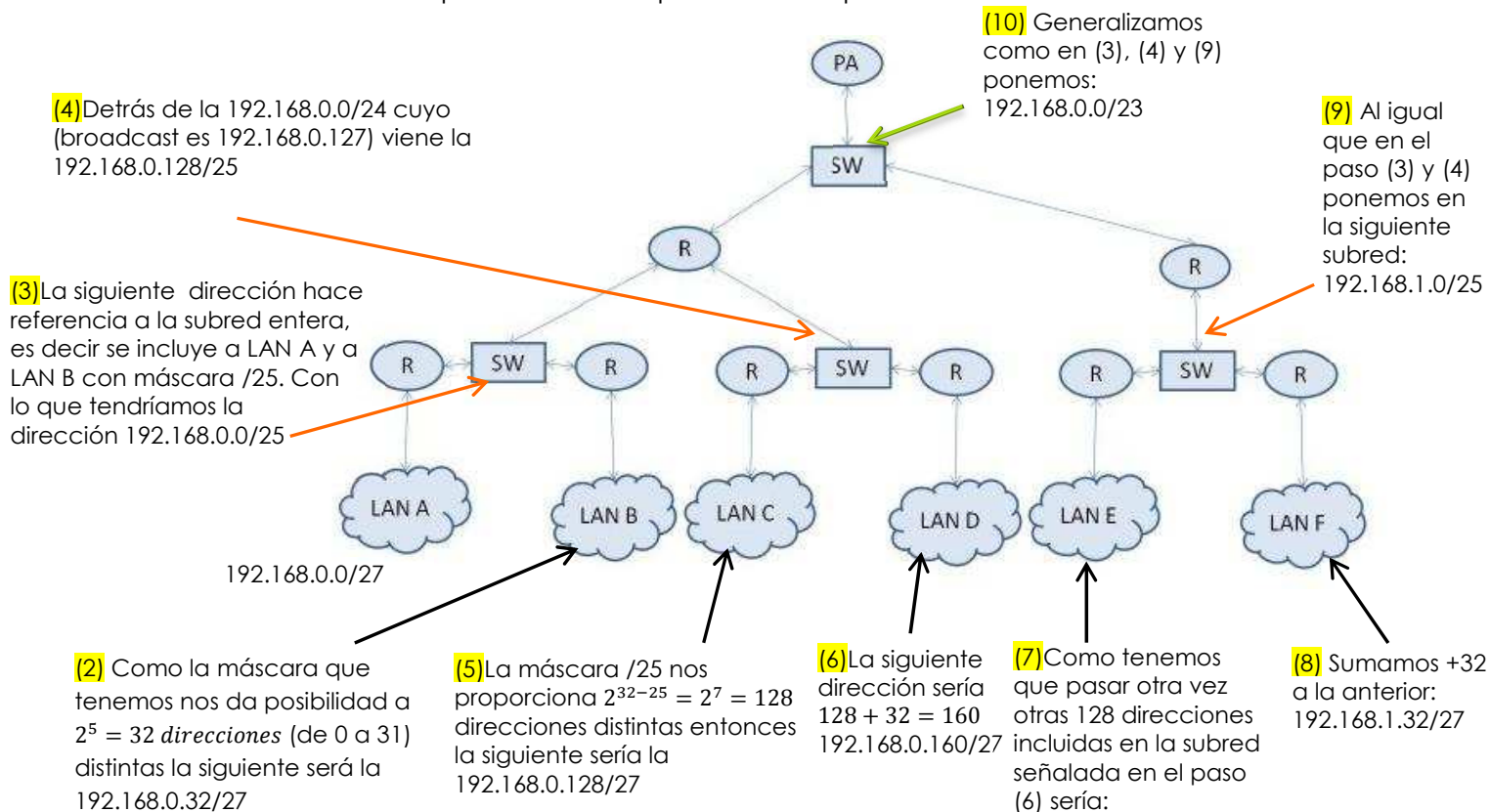
Luego se necesitarían otros 2 *bits*. En definitiva una máscara /23.

Quedándose entonces la siguiente configuración:



Y ahora se van a comenzar a dar direcciones válidas a los distintos dispositivos. Vamos a seguir un algoritmo que no nos da la solución óptima, pero se le aproxima.²

- 1) Comenzamos por las subredes de abajo a la izquierda (o bien a la derecha es indiferente). Se nos dice que empecemos por la dirección 192.168.0.0 pues esa es la que llevará la primera:

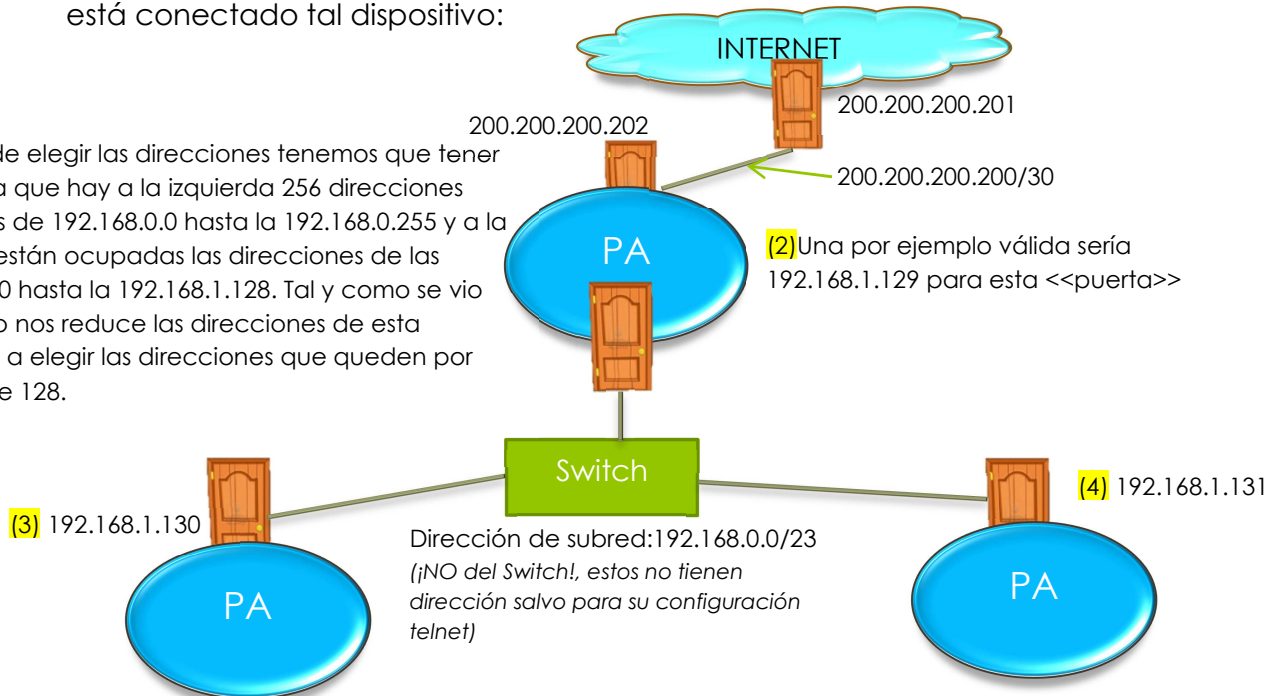


² La solución óptima es difícil de establecer. Al menos nos da una solución que cumple los objetivos (número de entradas mínimo en las tablas de encaminamiento) sin desperdiciar demasiadas direcciones.

Lo último que vamos a realizar va a ser la tabla de encaminamiento que se nos pide la del dispositivo PA.

Asunción. Antes tenemos que asignar direcciones a los nodos a los que está conectado tal dispositivo:

(1) Antes de elegir las direcciones tenemos que tener en cuenta que hay a la izquierda 256 direcciones ocupadas de 192.168.0.0 hasta la 192.168.0.255 y a la derecha están ocupadas las direcciones de las 192.168.1.0 hasta la 192.168.1.128. Tal y como se vio antes. Esto nos reduce las direcciones de esta <<calle>> a elegir las direcciones que queden por encima de 128.



Finalmente se realiza la tabla:

(Subred) Dirección.	Máscara	Siguiente Nodo.
200.200.200.200	/30	
192.168.0.0	/23	
0.0.0.0	/0	200.200.200.201
192.168.0.0	/24*	192.168.1.130
192.168.1.0	/25	192.168.1.131

Nótese que la "línea" denota que está directamente conectado.

Se podría poner por defecto cualquier otro nodo, pero lo lógico sería echarlo hacia internet.

*Se elige la máscara /24 ya que hay dos posibilidades irse a la izquierda o a la derecha. Donde entonces se nos quedaría a la izquierda las 256 primeras direcciones porque tendríamos $2^{32-24} = 256$ direcciones –de 192.168.0.0 hasta la 192.168.0.255– que son exactamente las direcciones que definimos anteriormente.