

Ejercicios redes

Mateo Díaz Pérez

EJERCICIO 2.

$$v = \infty \Rightarrow t_{prop} = \frac{d(Km)}{v(Km/s)} = 0$$

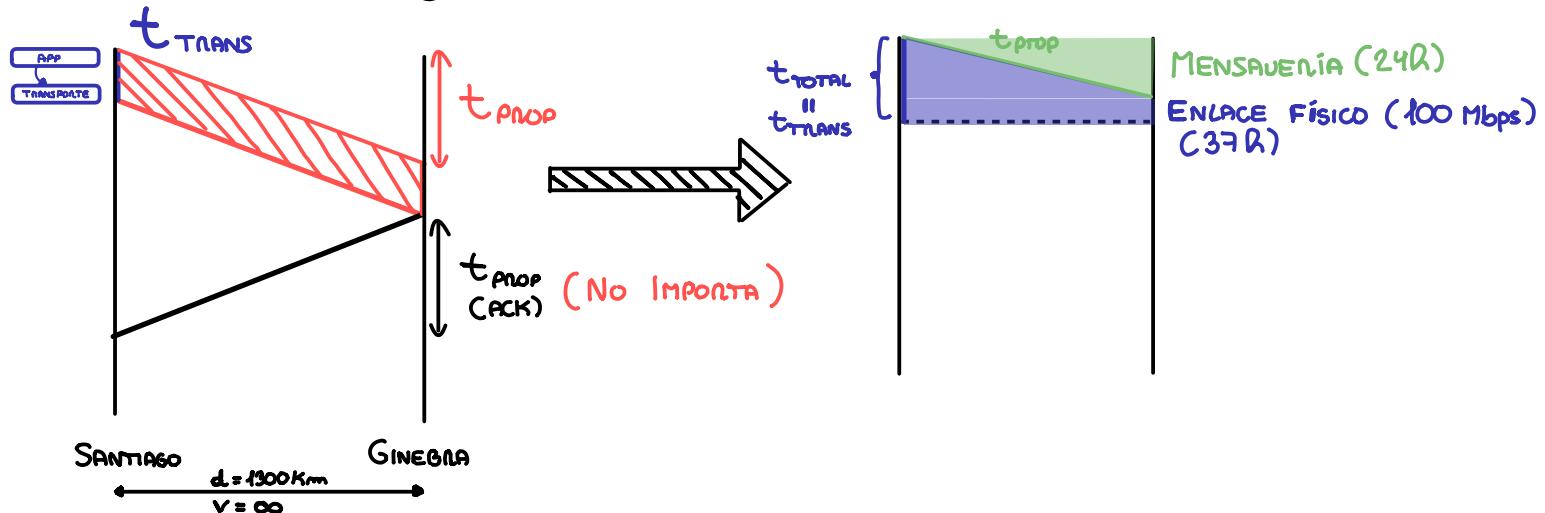
"size of package"

$$t_{trans} = \frac{R}{L} = \frac{4 \times 10^{13} \text{ BYTES}}{1.25 \times 10^7 \text{ BYTES/s}} \approx 37 \text{ DIAS} > 24 \text{ DIAS} \Rightarrow \text{MEJOR LA EMPRESA DE MENSAJERIA}$$

"Wi-Fi SPEED"

$$R = 100 \text{ Mbps} = 10^8 \div 8 \text{ BYTES/s} = 1.25 \times 10^7 \text{ B/s}$$

1 CONEXIÓN ES:



EJERCICIO 3.

$$R = 10 \text{ Mbps}$$

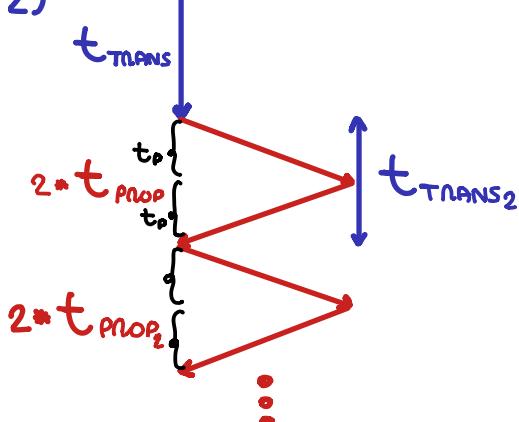
$$d = 36000 \text{ Km}$$

$$v = 2,4 \times 10^8 \text{ m/s}$$

c) $L = \infty \rightarrow$ ¿EL ENLACE ESTÉ TRANSMITIENDO CONTINUAMENTE?

$L_{\text{tono}} \text{ PNG}$

2)



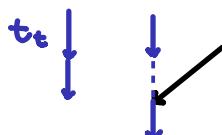
$$\text{a) } t_{prop} = d/v = \frac{3,6 \times 10^7 \text{ m}}{2,4 \times 10^8 \text{ m/s}} = 0.15 \text{ s}$$

b) $\overset{\text{DELAY}}{\underset{\text{BANDWIDTH}}{\text{BDP}}} \rightarrow \text{Nº BYTES ENTRE A Y B.}$

$$\text{BDP} = R \cdot t_{prop} = 10 \text{ Mbps} \times 0.15 \text{ s} = 1.5 \text{ Mb} \approx 187,5 \text{ KB}$$

1) $\xrightarrow{t_{TRANS} = 1 \text{ s}}$

2) $\bullet U = 1,$



C1) $t_{TRANS} = 1 \text{ min} (\text{TIEMPO HASTA ENVIAR LA SIGUIENTE IMAGEN})$

$$t_{TRANS} = \frac{L}{R} = \frac{x \text{ BYTES}}{1.25 \times 10^6 \text{ BYTES/s}} = 60 \text{ s} \Rightarrow x = 75 \text{ MB TAMAÑO IMAGEN}$$

C2) VERSIÓN MÁS OPTIMA CON SLIDING WINDOW

$$U = \frac{t_{TRANS}}{t_{TRANS} + RTT} = 1 \Rightarrow t_{TRANS} = t_{TRANS} + RTT \quad \left. \begin{array}{l} \\ \end{array} \right\} \begin{aligned} RTT &= 2t_{prop} + t_{TRANS} \\ t_{TRANS} &= 2t_{prop} + 2t_{TRANS} \end{aligned}$$

$\text{NO}!!$
 $\Rightarrow -t_{TRANS} = 2t_{prop}$

CONCLUSIÓN: EN LA 1

EJERCICIO CLASE *PENDIENTE

ISP: 68.80.0.0/13

Facultad: 68.85.2.0/24

Google: 64.233.160.0/19

PORTÁTIL: 68.85.2.101

ENLACE

MAC.O
MAC.D

IP

0000...00
1111...11

UDP

68

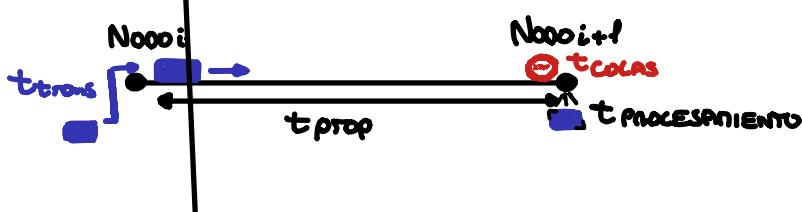
DHCP
DISCOVER

ESTÁ EN LAS TRASMAS DEL
ÚLTIMO TIEMPO, MUY BUENO PARA
REPASAR

EJERCICIO 4.

$$T_{TOTAL} = \sum_{i=0}^{n-1} \underbrace{T_{\text{Nodo}i - \text{Nodo}i+1}}$$

TIEMPO ENTRE DOS NODEOS



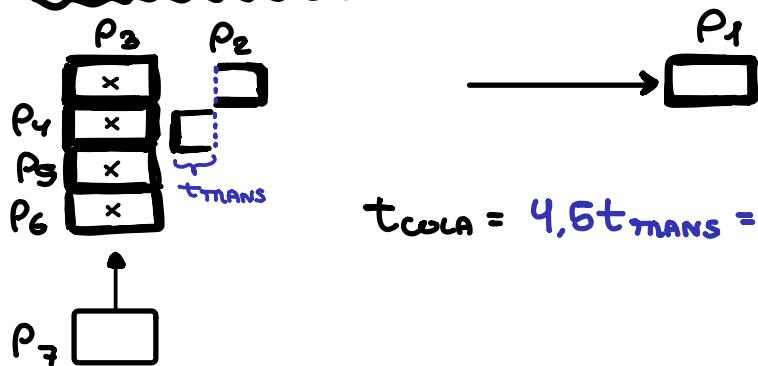
$$t_{prop} = \frac{d}{v} = \frac{2000 \text{ Km}}{200000 \text{ Km/s}} = 0.01 \text{ s} = 10.000 \mu\text{s}$$

$$t_{trans} = \frac{L}{B} = \frac{5 \times 10^6 \text{ b}}{5 \times 10^6 \text{ b/s}} = 1 \text{ s} = 1000.000 \mu\text{s}$$

$$t_{colas} = 2 \mu\text{s}, t_{procesa} = 1 \mu\text{s}$$

$$T_{total} = 11(1) + 0,01 + 10(0,000003) = 11,010003$$

EJERCICIO 5

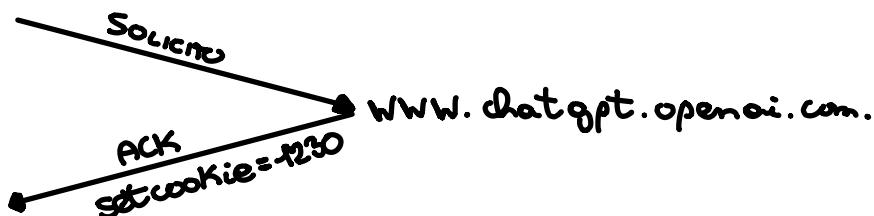


$$t_{cola} = 4,5 t_{trans} = 4,5 \frac{L}{B} = \frac{4,5 \times 1500 \text{ Bytes}}{2 \times 10^6 \text{ bits/s}} = 0,275$$

8 bits/byte

EJERCICIO 6

Cookies:



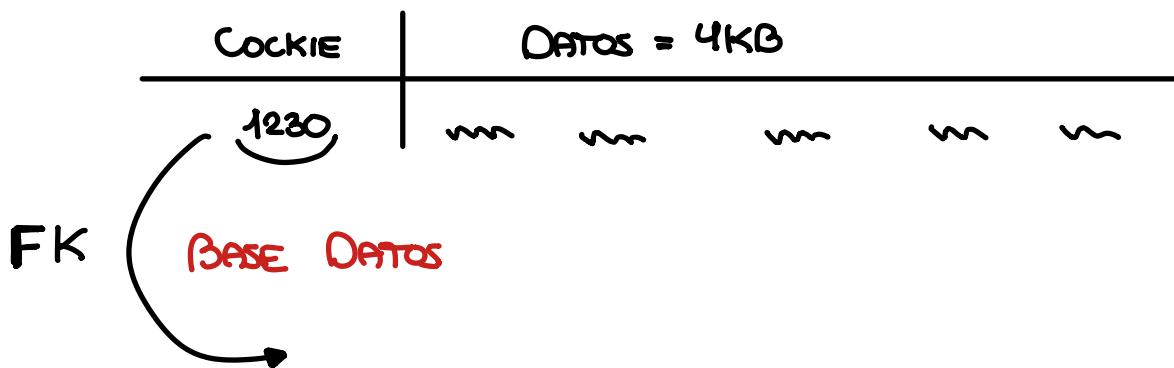
ChatGPT	1230
www.linkedin.com	2738
:	

{PK} 8 LINKEDIN $\times \text{KB} \gg 4$
2738: búsquedas: .., .., .. ooo

UTILIZAR COOKIES	NO UTILIZARLAS
VARIAS COOKIES PARA EL MISMO DNS.	NECESITANOS PROTOCOLO CON ESTADO, OTCP

REALIDAD

FireFox



EL NOMBRE ALMACENASE SOBRE LA CLAVE DE LA COOKIE Y QUE EL MESTRO DE INFORMACIÓN SE GUARDA EN LA BASE DE DATOS DE LA EMPRESA.

EJERCICIO 7

DNS & DOMAIN NAME SYSTEM

dominio web → IP

CAPPA DE NED

SEGMENTO:

DNS



a) VERDADERA: CADA SEGMENTO DNS A PARTE DE PROPORCIONAR INFORMACIÓN SOBRE LA IP DOMINIO PEDIDO, SE ENTREGAN IP's DE VARIOS SERVIDORES AUTORIZADOS PARA MANTENER ESE DOMINIO.

HTTP(s)://www.domain.extension
segundo IP

b) ~~VERDADERO~~, PARA PODER UTILIZAR EL PROTOCOLO HTTP PARA RECIBIR ARCHIVOS DE HYPertexto DE LA WEB, SE NECESITA PODER ACCEDER A LA IP DEL SERVIDOR QUE ALLOJA LOS ARCHIVOS HYPERTEXTU DE DICHA PÁGINA WEB.

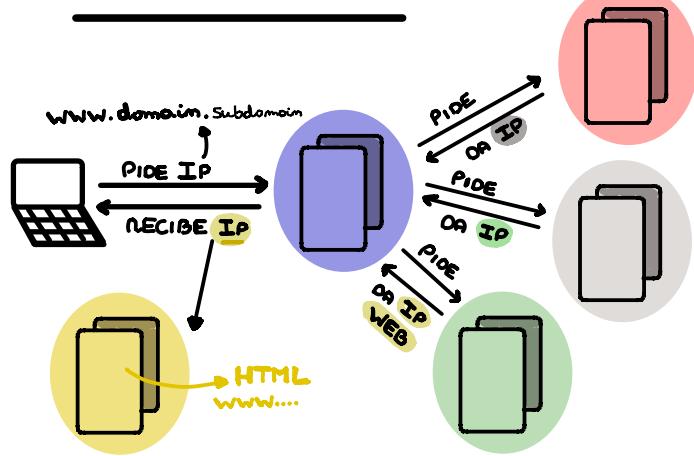
FAlsa, EN CASO DE TENER LA IP NO NECESITARÍAN UTILIZAR DNS

c) CIERTO, NAT PERMITE AHORRAR DIRECCIONES IPv4 ENMASCARANDO TODOS LOS HOST DE UNA LAN TRAS UNA SOLA IPv4 DEL ROUTER FRONTERA.

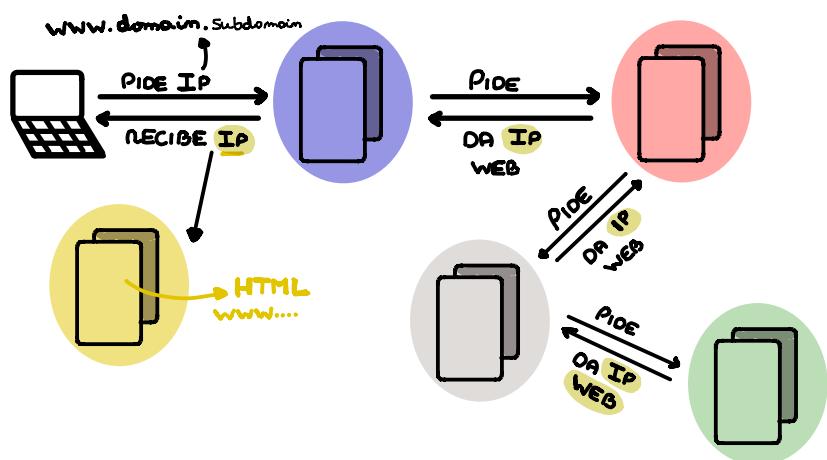


d.)

ITERATIVO



RECUSIVO

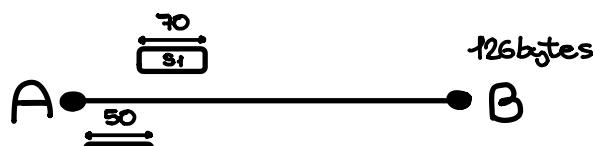


FALSO, EN UNA CONSULTA RECUSIVA EL SERVICIO ISP LOCAL EN CASO DE NO TENER LA RESPUESTA ALMACENADA EN SU CACHE DNS, PIDE AL ROOT LA IP CORRESPONDIENTE, ESTE HACE LA PETICIÓN AL SIGUIENTE SERVICIO, HASTA ENCONTRAR EL ISP AUTORIZADO, RETROCEDIENDO LA INFORMACIÓN POR TODOS LOS SERVICIOS HASTA EL ISP LOCAL.

EJERCICIO 8

PASO	ORIGEN	DESTINO	TIPO	INFORMACIÓN
1	jefe.empresa.com	dns.empresa.com	Consulta	¿IP www.servicio.com?
2	dns.empresa.com	root	Consulta	¿IP www.servicio.com?
3	root	tld.gr	Consulta	¿IP www.servicio.gr?
4	tld.gr	root	Respuesta	A.B.C.D
5	root	.dns.empresa.com	Respuesta	A.B.C.D
6	.dns.empresa.com	jefe.empresa.com	Respuesta	A.B.C.D

EJERCICIO 9



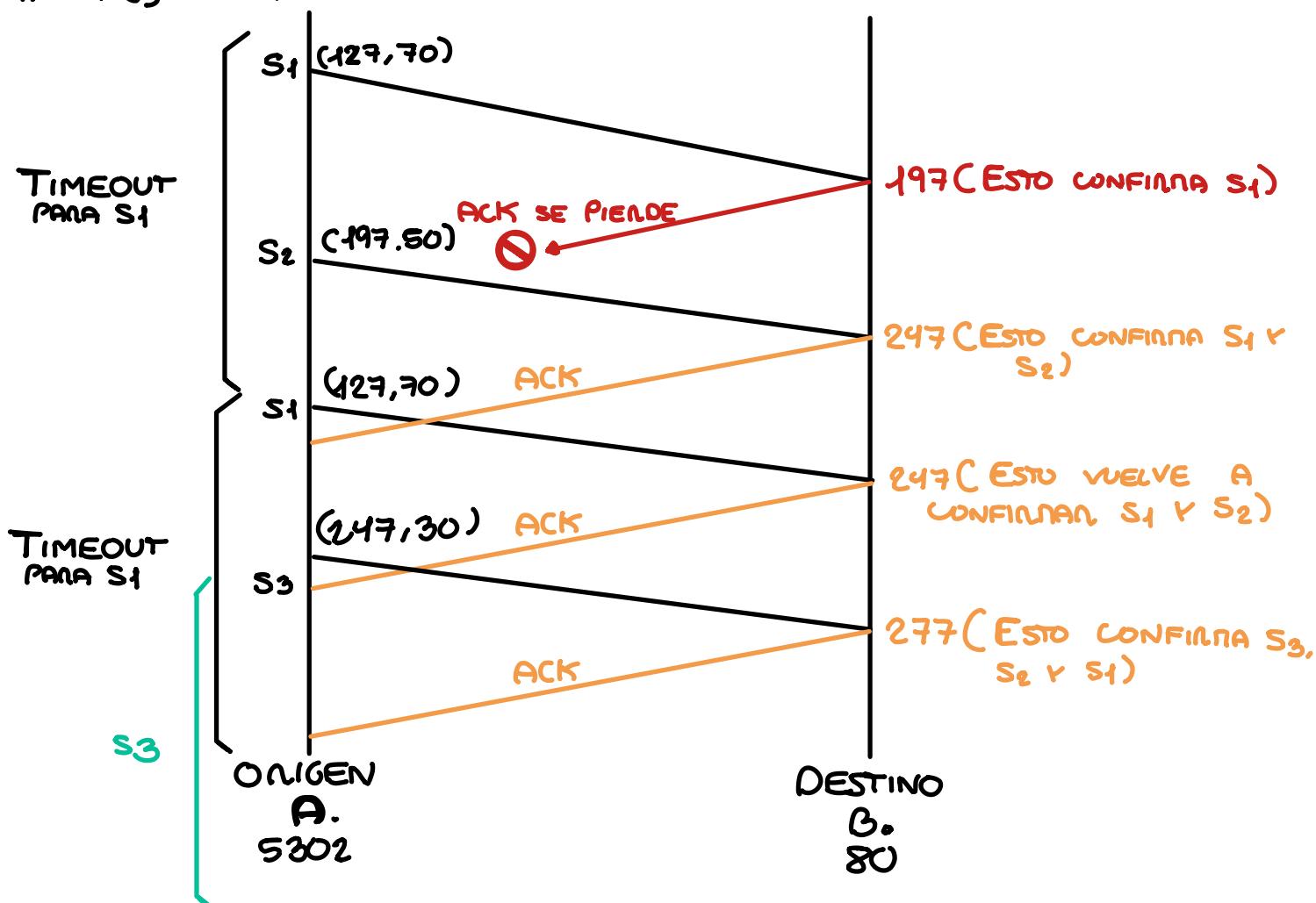
Nº	PUERTO ORIGEN	PUERTO DESTINO
a) S ₁ : 127	S302	80
b) S ₂ : 197	S302	80
ACK: 197	80	S302

c) SI S_2 LLEGA ANTES QUE S_1 , TCP DETECTARÁ QUE FALTA EL SEGMENTO S_1 . POR LO QUE MANDARÁ TRES ACK's DÚPLICADOS, TENIENDO EL PRIMERO - LA SIGUIENTE ESTRUCTURA:

ACK: 127 80 5302
 ↘ Ultimo Byte recibido.

d) ACK₁ SE PIERDE Y ACK₂ LLEGA DESPUÉS DE TIMEOUT.

PARA RESOLVERLO HAY QUE SUPONER UNA VENTANA DESLIZANTE PARA QUE S_1, S_2, S_3 PUEDAN ENVIANSE ANTES DE LLEGAR SU ACK



TCP FUNCIONA MEDIANTE LA RETRASIÓN DE DATAGRAMAS NO CONFIRMADOS PARA ASEGURAR LA FIABILIDAD DE LA TRANSMISIÓN, PARA ELLA EL Host QUE RETRASATE LLEVA UN TEMPORIZADOR DEL ÚLTIMO PAQUETE NO CONFIRMADO (BIEN PUEDE PON FALLO EN LA ENTREGA O EN LA CONFIRMACIÓN), RETRASITIENDO EL DATAGRAMA MÁS VIEJO NO CONFIRMADO ANTES DEL TIEMPO DE "TIMEOUT" O TEMPORIZACIÓN. AL PERDERSE EL ACK = 197 EL SEGMENTO S₁ NO ESTÁ CONFIRMADO EN A. SI EL ACK = 247 LLEGASE ANTES DEL FINAL DEL TEMPORIZADOR SE CONFIRMAN AN S₁ Y S₂ DIRECTAMENTE, EN ESTE CASO NOS INDICAN QUE LLEGA DESPUES, PUN LO TANTO, LO QUE SUCDE ES QUE A RETRASATE EL SEGMENTO MÁS ANTIGUO, S₁, (Y SE REINICIA EL CONTADOR), B AL RECIBIR EL DATAGRAMA DEVUELVE OTRA VEZ EL ACK 247 INDICANDO QUE RECIBIÓ S₁ Y S₂

POR TANTO CUALQUIERA OEL SEGUNDO O TERCER ACK CONFIRMAN LOS DOS SEGMENTOS DENTRO DEL NUEVO TIMEOUT, Y SE PASA A PROPAGAR EL SEGMENTO S₃, REINICIANDO SU CONTADOR Y ACTUALIZANDO EL ACK DEVUELTO POR B INDICANDO QUE LOS 30 BYTES PROCESADOS DE S₃ SE RECIBIERON CORRECTAMENTE.

EJERCICIO 10

a) Como nos dicen que hay 4 bytes:

$$2^{8 \times 4} = 2^{32} \text{ POSIBILIDADES}$$

xxxxxx xxxx xx xxxx xx xxxx xx xxxx

PARA EL CAMPO DEL NÚMERO DE SUCUENCIA SE PUEDEN REPRESENTAR A LO SUMO 2^{32} POSIBILIDADES, SUPONIENDO QUE LA UNIDAD A REPRESENTAR ES EL MSS (MAXIMUM SEGMENT SIZE), OBTENDRÍAMOS EL TAMAÑO MÁXIMO DEL PAQUETE como $2^{32} \times \text{MSS} = 2^{32} \times 536 = 2^{30} \times 4 \times 536 = 2144 \times 2^{30} \text{ bytes} = 2144 \text{ GiB.}$ ✗ FALLO TÍPICO

$$1 \text{ GiB} = 2^{30} \text{ bytes}$$

✓ CORRECTO

SI EL NÚMERO DE SEGMENTO REPRESENTASE Nº bytes Leídos EN LUGAR DEL NÚMERO DE SEGMENTO A ENVÍAR (Como IDENTIFICATIVO DEL PAQUETE) SERÍAN $2^{32} = 4 \text{ GiB}$ DE TAMAÑO. EN ESTE CASO NOS DICE LA TEORÍA QUE LA OPCIÓN CORRECTA ES ESTA.

b) PRIMERO TENEMOS QUE CALCULAR EN Nº SEGMENTOS QUE VAMOS A ENVÍAR PARA SUMARLE A CADA UNO LOS SEGMENTOS LOS 66 BYTES DE CABECERAS.

TAMAÑO MENSAJE: 2^{32} bytes

MÁXIMO TAMAÑO DE SEGMENTO: 536 bytes

MÍNIMO NÚMERO DE SEGMENTOS A ENVÍAR: $\frac{2^{32}}{536} = 8012998.68 \rightarrow 8012999$

BYTES ANADIDOS EN TOTAL POR CABECERAS: $66 \times 801299 = 528867934 \text{ bytes}$

TAMAÑO TOTAL A ENVÍAR: $2^{32} + 528867934 = 4823825230 \text{ bytes}$

TAMAÑO EN Mbits: $\frac{4823825230 \text{ bytes} \times 8 \text{ bits}/\text{byte}}{10^6 \text{ bits}/\text{mbit}} = 3890.6 \text{ mb}$

$$\text{TIEMPO DE TRANSMISIÓN: } t_{\text{TRANS}} = \frac{L}{B} = \frac{38590.6 \text{ Mb}}{155 \text{ Mbps}} = 248.975$$

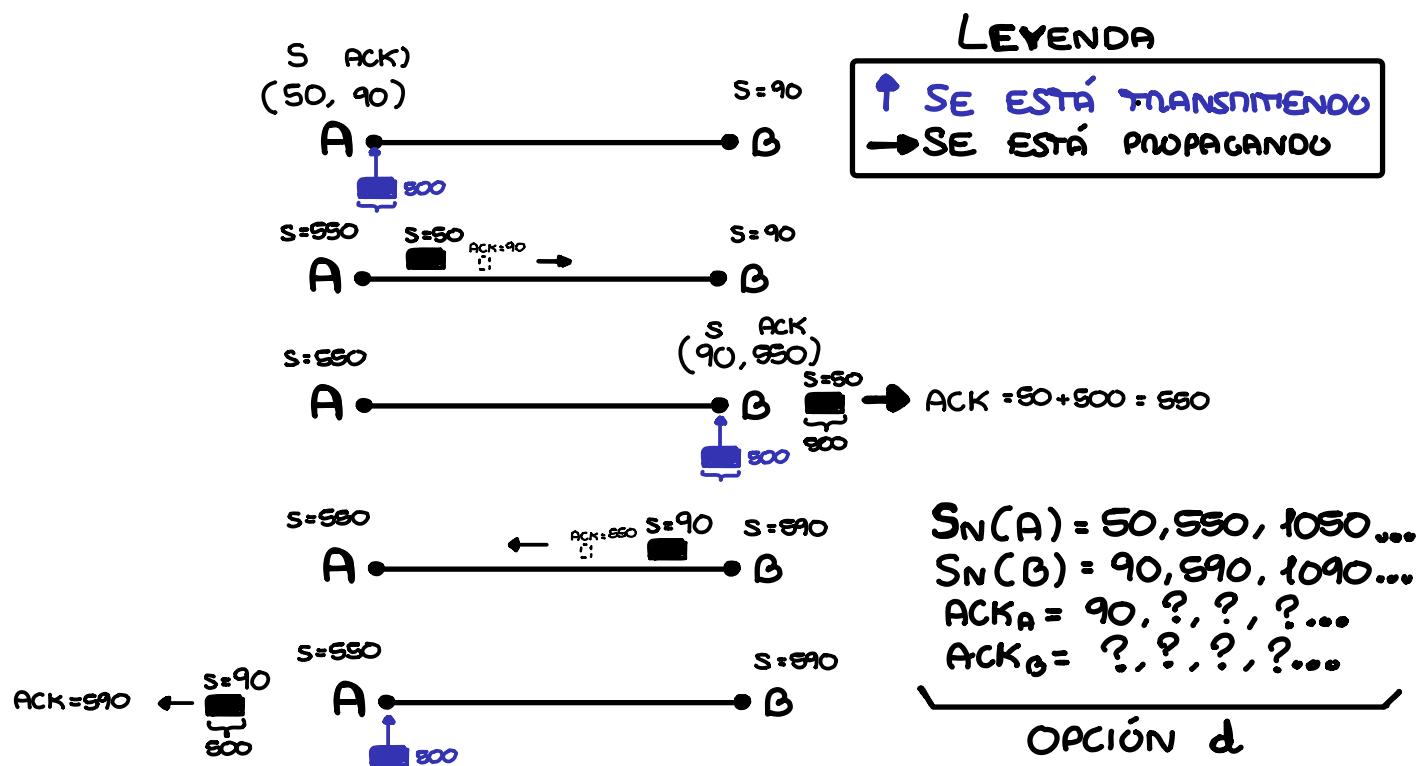
EL TIEMPO REAL SERÍA LA SUMA DE TRANSMITIR TODOS LOS PAQUETES MÁS EL TIEMPO DE PROPAGAR EL PAQUETE DE A A B.

PERO COMO NO NOS DAN LA DISTANCIA QUE SEPARA LOS NODOS NI LA VELOCIDAD EN EL MEDIO QUE LOS SEPARA NO SE PUEDE CALCULAR t_{prop} .

(SE SUPONE QUE SOLO NOS PIDEN EL TIEMPO DE "TRANSMISIÓN")

EJERCICIO 11.

SI NOS DICE QUE ESTA CONEXIÓN TCP UTILIZA PIGGYBACKING, BASICAMENTE NOS DICE QUE A Y B CADA VEZ QUE ENVÍAN UN ACK, MANDAN TAMBÉN UN NUEVO SEGMENTO



LOS ACKS DEL DIAGRAMA SON EL "CASO FÁCIL" CUANDO LA VENTANA ES TAMAÑO 1, PERO COMO NO SE SABE A B PUEDEN LLEGAR DOS PAQUETES ANTES DE QUE ENVÍE SU PRIMER PAQUETE, COMO LOS ACK SON ACUMULATIVOS PASARÍA DIRECTAMENTE DE 50, 550, 1050... EN N° PAQUETES YA ENTREGADOS SERÍA EL NÚMERO DE POSICIONES QUE SE SAXTA EN LA SUCECIÓN.

ESTO SE VUEVE TUDAVÍA MÁS PROBABLE A MEDIDA QUE AVANZAN LOS ACK, POR ESO LOS MARCAMOS TODOS CON '?'.

EJERCICIO 12



$$\text{MENSAJE} = 15S; S = \text{MSS}$$

CADA SEGMENTO TIENE 1 MSS, OSEA S bytes, ADÉNDS LA VELOCIDAD DE TRANSMISIÓN ES R bytes/s, POR LO TANDEM $t_{trans} = S/R$.

CASO a) $\text{RTT} > 3S/R \Rightarrow \text{RTT} > 3t_{trans} \Rightarrow$ EN LO QUE VA EL PAQUETE Y VUEVE EL ACK SE PUEDEN IR TRANSMIENDO MÍNIMO 3 PAQUETES. EN ESTE CASO PONE UNA FOTO, ES FACILITO PERO BASICAMENTE AUMENTA LA VENTANA INFINITAMENTE HASTA ACABAN LOS SEGMENTOS.

SOLICITUD: 1 RTT (VENTANA = 1)

1º SEGMENTO: 1 RTT + 1 t_{trans}

VENTANA *2:

2º - 3º SEGMENTO: 1 RTT + 1 t_{trans} (EL 3 SE TRANSMITE MIENTRAS VIAJA ACK)

VENTANA *2:

4º - 7º SEGMENTO: 1 RTT + 1 t_{trans} (" 5,6,7 ")

VENTANA *2:

8º - 15º SEGMENTO: 1 RTT + ~~1 t_{trans}~~ (" 9,...,15 SE TRANSMITEN 8 t_{trans} MIENTRAS VIAJA UN ACK QUE NO NOS IMPORTA)

TOTAL: 4 RTT + 11 t_{trans}

ESTE SOLO

a) BIEN RAZONADO): $R_{TT} > 3S/R \Rightarrow R_{TT} > 3t_{TRANS}$

\Rightarrow PARA "AHORRARSE" 1 RTT LUEGO DE QUE SALGA ACK, TIENEN QUE QUEDAR MÁS DE 3 PAQUETES POR TRANSMITIR (+ EL PRIMER PAQUETE SON 4). LUEGO CUANDO LA VENTANA CRECE EN POTENCIAS DE 2: (1,2,4,8...) ESTO SOLO PASARÁ CUANDO SE ENVÍEN 8 PAQUETES. AHÍ "SOLO TANCA" 8 RTTANS

b) $R_{TT} < S/R$

NUEVAMENTE EN ESTE CASO HACE UN

$S/R \approx 1,5 R_{TT}$, PERO ES UN LARGO

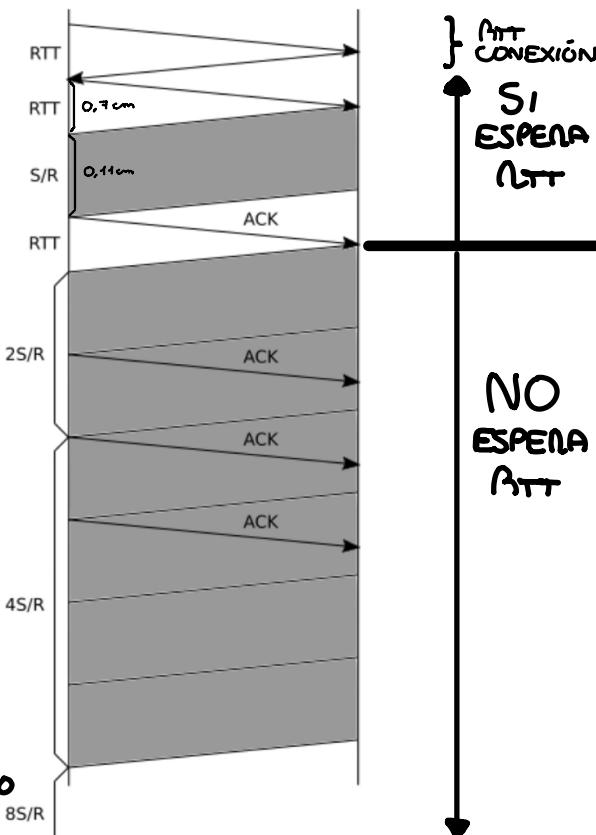
INFINITO PODRÍA SER EL DOBLE, EL

TRIPLE... Y SE SEGUÍA CUMPLIENDO

SIEMPRE Y CUANDO $R_{TT} > S/R$.

(EL DIAGRAMA PUEDE A OJO)

LO ÚNICO QUE SUcede ES QUE A PARTIR DE UNA VENTANA DE TAMAÑO 2, EL ACK VA Y VUELVE CON LOS SIGUIENTES SEGMENTOS ANTES DE QUE SE TRANSMITA EL SIGUIENTE PAQUETE SIN TENER QUE ESPERAR R_{TT} TIEMPO PARA TENER MÁS PAQUETES LISTOS PARA SER TRANSMITIDOS EN EL BUFFER DEL RECEPTOR (MIENTRAS LA VENTANA DE RECEPCIÓN R_{wmd} PERMITA ACUMULAR TANTOS PAQUETES PENDIENTES).



CONEXIÓN: 1 RTT (VENTANA ENVÍO $S_{Env} = 1$)

1º PAQUETE: $2t_{prop} + t_{TRANS} = R_{TT} + S/R$

VENTANA $\times 2$ YA VOLVIÓ ACK*

2º, 3º PAQUETE: $2t_{prop} + \underbrace{t_{TRANS}}_{ACK*} + \underbrace{t_{TRANS}}_{3º} + \underbrace{t_{TRANS}}_{4º} = R_{TT} + S/R + S/R$
VENTANA $\times 2$

4º - 7º: $4 \times t_{TRANS} = 4 S/R$

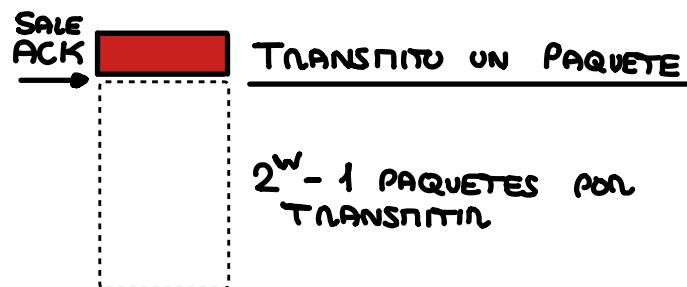
VENTANA $\times 2$

8º - 15º: $8 \times t_{TRANS} = 8 S/R$

TOTAL: $15 S/R + 3 R_{TT}$

$$c) \quad 3S/\beta > \beta_{TT} > S/\alpha$$

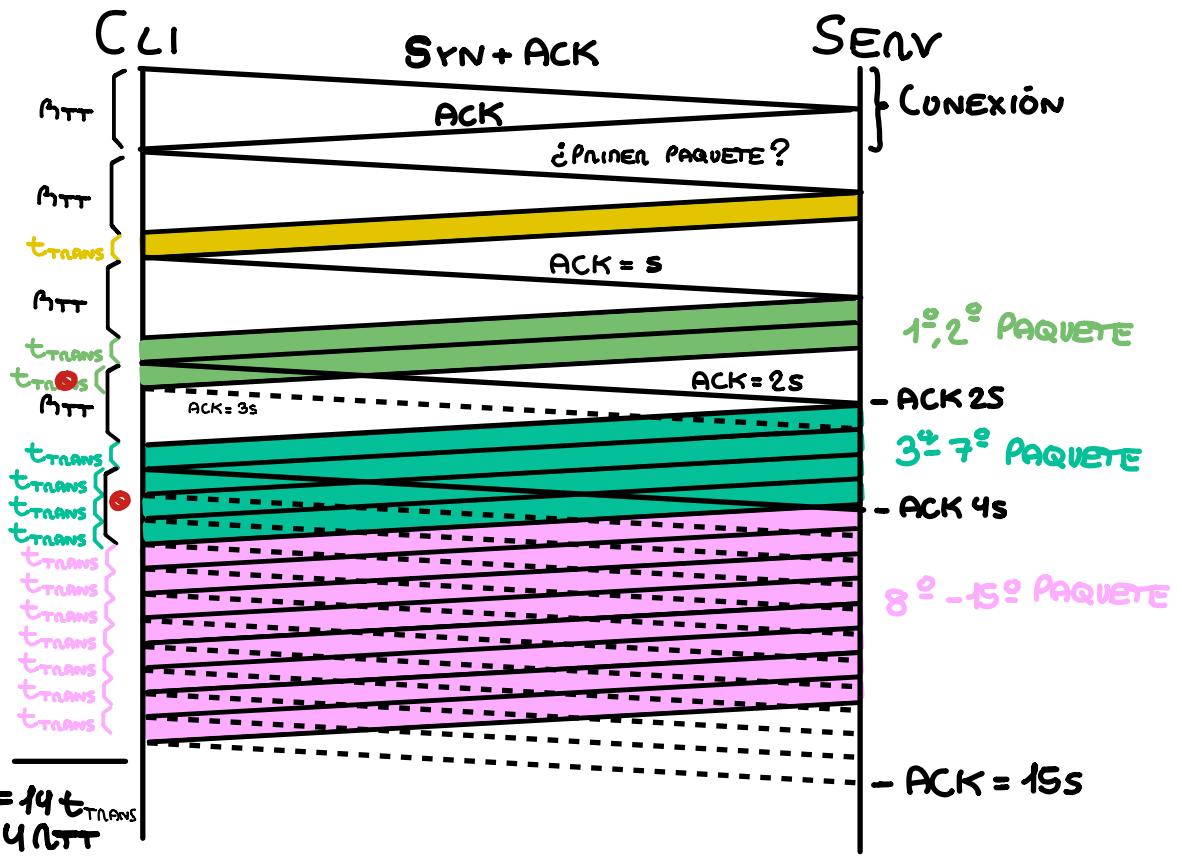
ESTE CASO ES EL QUE NENOS
DUAS GENERA YA QUE UN ACK
TANDE O BIEN MÁS U NENOS
DE DOS TRANSMISIONES DE
PAQUETES, PERO NUNCA PASA DE
TRES ENTONCES CUANDO LA
VENTANA CRECE EN POTENCIAS
DE 2.



PARA AHORRARLOS UN PITT TIENE
 QUE QUEDAR \geq d PAQUETES SI
 $P_{IT} = \frac{d}{13}$; $d \in \{1, 3\} \rightarrow$ EN EL
 CASO $n_1 \neq n_0$ AL MENOS n_1 DE 1
 Y EN EL CASO $n_1 = n_0$ CON n_1

DE LOS LLEGA (NO ESTÁ INCLUIDO EL TRES) $\Rightarrow 2^W - 1 > 2$
 $\Rightarrow 2^W > 3 \Rightarrow W > \log_2(3)$

MINIMO VENTANA DE 4 PAQUETES
 PARA APROXIMAR RTT.



$$= 14S/\rho + 4\pi r^2$$

(trans)

EJERCICIO 13

EN TCP HAY 2 CONTROLES DE LA CANTIDAD DE PAQUETES QUE SE PUEDEN ENVÍAR CON 2 VENTANAS DIFERENTES:

- CONTROL DE FLUJO ("SE ADAPTA A LO QUE ESTÉ EL RECEPTOR")

→ AL ESTABLECER LA CONEXIÓN TCP SE INTERCAMBIAN INFORMACIÓN SOBRE EL NÚMERO DE SEGMENTO (S_N) Y EL TAMAÑO DE VENTANA DE RECEPCIÓN (R_{wnd}) INICIALES.

→ LA R_{wnd} IDENTIFICA EL NÚMERO DE bytes QUE UN HOST PUEDE RECIBIR DEL OTRO HOST EN UN MOMENTO DETERMINADO.

→ A MEDIDA QUE LOS BUFFERS DE UN HOST SE VAN LLENANDO O LA CAPA DE APLICACIÓN VA LEYENDO Y VALIDANDO ESTA VA CAMBIANDO.

→ EN CADA ACK SE ACTUALIZA AL OTRO HOST EL VALOR ACTUAL DE SU R_{wnd} . MIENTRAS LA VENTANA DE ENVÍO DEL OTRO HOST (S_{wnd}) PUEDE CRECER COMO SE LO PERMITA LA RÉD, PERO NO MÁS DE LA R_{wnd} DE QUIEN RECIBE.

- CONTROL DE CONGESTIÓN ("SE ADAPTA A LO QUE ESTÉ LA RÉD") EN ESTE EJERCICIO NO LO PIDE.

→ CADA HOST TCP TIENE UNA VENTANA DE ENVÍO (C_{wnd}), PERO SI UNO SÓLO RECIBE DATOS PUES NO LA UTILIZARÁ, PARA REPRESENTAR EN N.º bytes QUE PUEDE ENVÍAR AL OTRO HOST SIN RECIBIR ACK.

→ LA C_{wnd} EMPIEZA SIENDO 1MSS (NORMALMENTE LOS PAQUETES SON MUCHO MENORES QUE MSS, POR ESO NO ENVÍAN SEGMENTOS < MSS), ES DECIR, UN SEGMENTO.

1. ESTA C_{wnd} O VENTANA DE CONGESTIÓN INTENTA APROXIMAR LA CONGESTIÓN DE LA RÉD, SE INCREMENTA EN UNO POR CADA VEZ QUE SE RECIBE UN ACK NUEVO, PERO ESO SI TODOS LOS ACK SON NUEVOS EL NÚMERO DE SEGMENTO SE DUPLICARÁ HASTA QUE:

1.1. C_{wnd} SUPERA UN LÍmite/UMbral, AL PRINCIPIO SE FIJA A UN VALOR MUY GRANDE Y PASA INCREMENTAR SU TAMAÑO LINEALMENTE CADA RTT (EVITACIÓN DE LA CONGESTIÓN)

1.1.1 \rightarrow Fórmula $VC = VC + \pi MSS^2 / VC$ (VC : Cwnd) por cada ACK!Nuevo

1.1.2 \rightarrow SI LLEGAN 3 ACK DUPLICADOS ES PORQUE SE HA PERDIDO UN SEGMENTO, PUES SE RETRANSMITE Y EL UMbral PASA A SER LA MITAD DE LA Cwnd.

1.1.2.1 Si tenemos TCP RENO Cwnd también se divide en 2 y le sumamos 3MSS (por los tres ACK duplicados), esto se conoce como REUPERACIÓN RÁPIDA

- * 1.1.2.1.1 HASTA QUE LLEVE UNO NUEVO, AVANZAMOS 1MSS CADA ACK LA Cwnd

- * 1.1.2.1.2 SI ES NUEVO VOLVEMOS A 1.1.1 (EVITACIÓN DE LA CONGESTIÓN)

- * SI HAY TIMEOUT LEER 1.1.3 (ES LO MISMO)

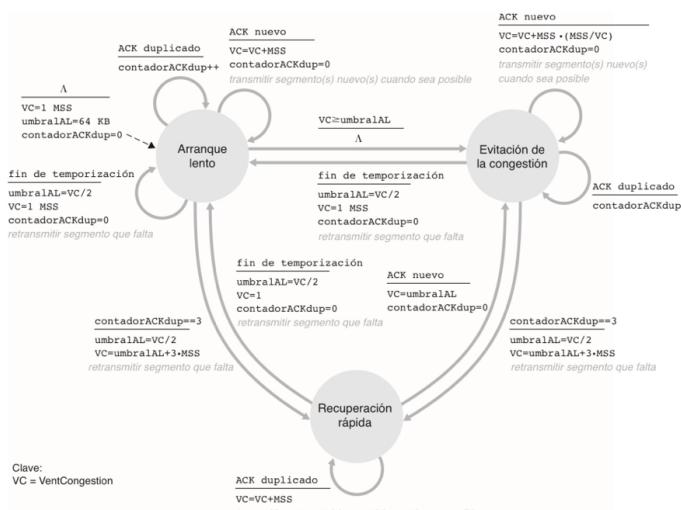
1.1.2.2 Si tenemos TCP TAHOE Cwnd pasa a ser 1 y vuelve a 1 (INICIO LENTO)

1.1.3 \rightarrow TANTO EN TAHOE como EN RENO SI LLEGA UN TIMEOUT (PAU TIEMPO DESPES SIN RECIBIR EL ACK QUE QUEREMOS), IMPLICA QUE NUESTRO SEGMENTO MÁS VIEJO PROBABLEMENTE NO LLEGO AL DESTINO O ESTÁ MUY CONGESTIONADA LA RED, ENVIANOS ESE SEGMENTO, EL UMbral = Cwnd/2 Y LA VENTANA Cwnd = 1, VOLVEMOS A 1 (INICIO LENTO).

1.3 SI HAY TIMEOUT RETRASMITIMOS EL SEGMENTO MÁS ANTIGUO SIN CONFIRMAR, FIJAMOS UMbral = Cwnd/2 Y VOLVEMOS A EMPEZAR EN 1 como ANTES.

IMPORTANTE: Los ACK Duplicados son seguidos, si llega uno nuevo, todo vuelve a empezar.

ESTA PARAFENALIA SE RESUME EN:



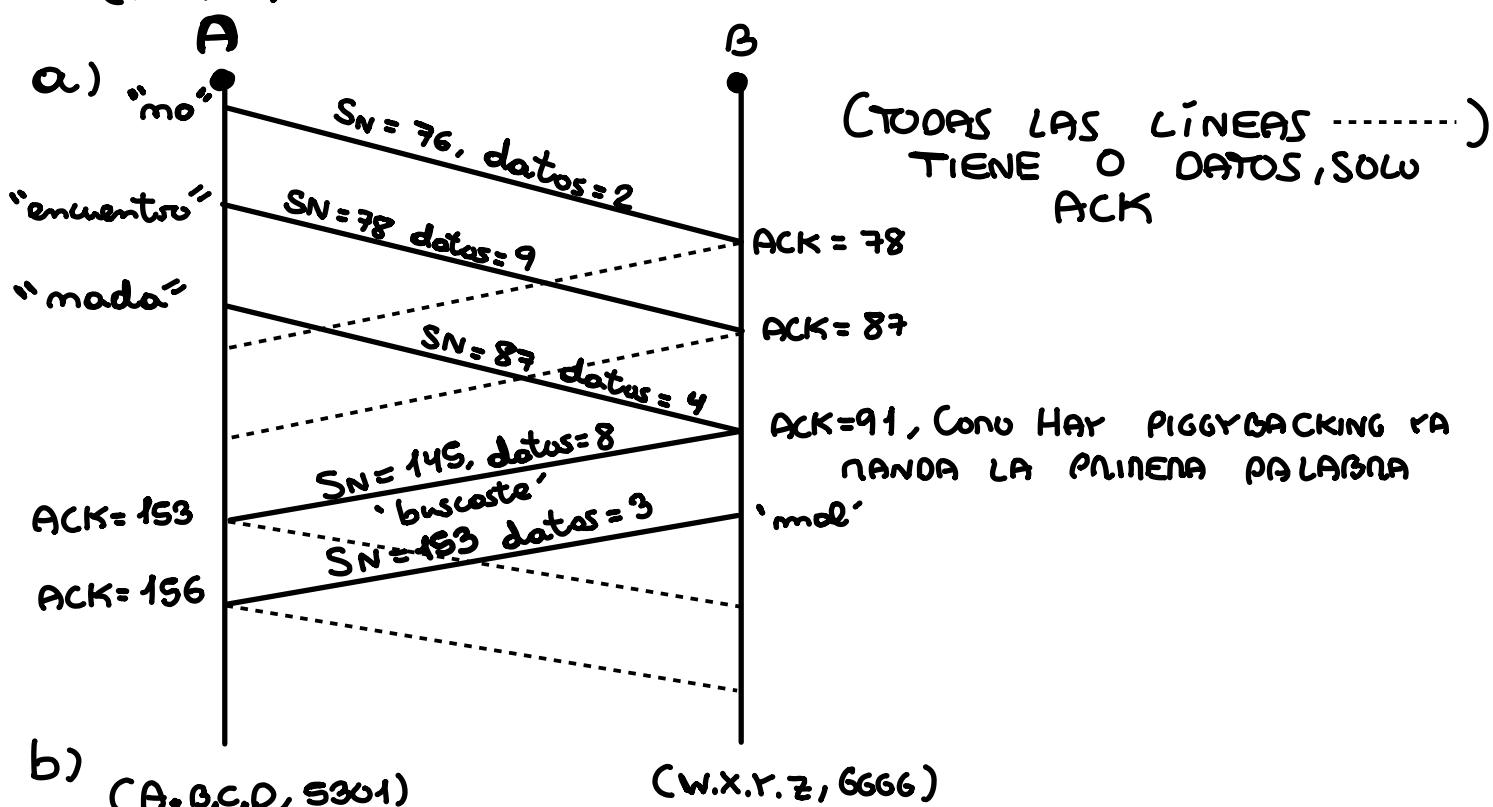
FINALMENTE EL ENIGMA ENVIA TANTOS BYTES COMO LA SWND INDIQUE CON EL NÚMERO ENTRE SU CWRD Y LA FRRD DEL RECEPTOR.

EJERCICIO 14

Como nos dice que hace PUSH después de cada palabra. LE INDICAMOS A TCP TRANSMITE LO QUE HAY EN EL BUFFER CUANDO QUIERAS, SEPARANDO ASÍ CADA PALABRA EN UN SEGMENTO AUNQUE SEAN MÚLTIPLOS BYTES.

(A.B.C.D, S301)

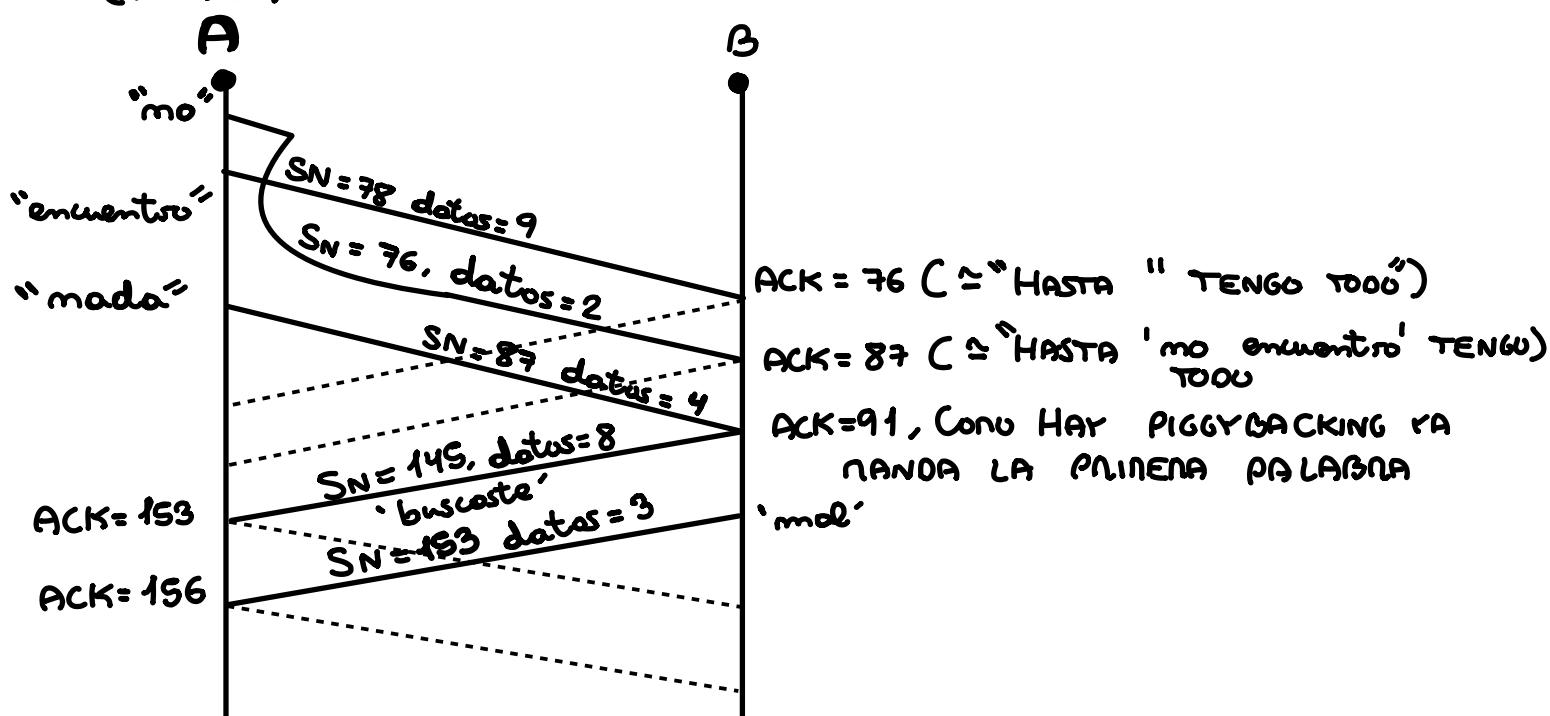
(W.X.Y.Z, GGGG)



b)

(A.B.C.D, S301)

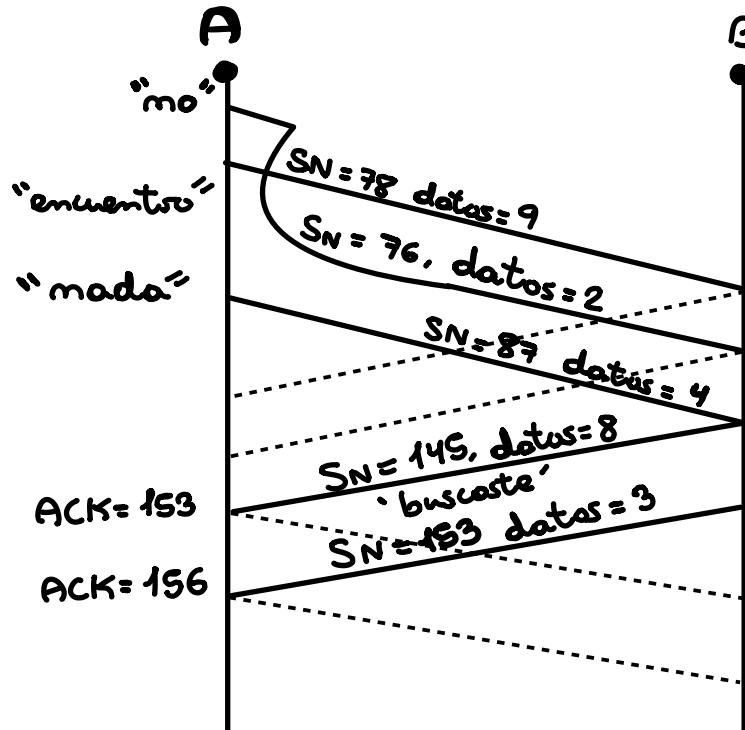
(W.X.Y.Z, GGGG)



C)

(A.B.C.D, 5301)

(W.X.Y.Z, 6666)



ACK = 76 (\approx "HASTA " TENO TOOO")
 ACK = 87 (\approx "HASTA 'mo encuentro' TENO TOOO")
 ACK = 91, Con HAR PIGGYBACKING YA
 PANDA LA PRIMERA PALABRA
 'mol'

EJERCICIO 15

10 Mbps \rightarrow velocidad transmisión

SEGMENTO = 1500 bytes = 12000 bits \rightarrow información a transmitir

$$t_{trans} \text{ de un segmento} = \frac{12000 \text{ bits}}{10000000 \text{ bits/s}} = 1.2 \text{ ms CADA SEGMENTO}$$

bufor recepción \gg Cwnd \Rightarrow $T_{wmd} \gg Cwnd = \frac{\max(Cwnd)}{\max(Cwnd)}$
 $\beta_{TT} = 100 \text{ ms}$

a) TENEMOS 100 ms (β_{TT}) PARA TRANSMITIR X PAQUETES,
 TIENDO CADA UNO 1.2 ms

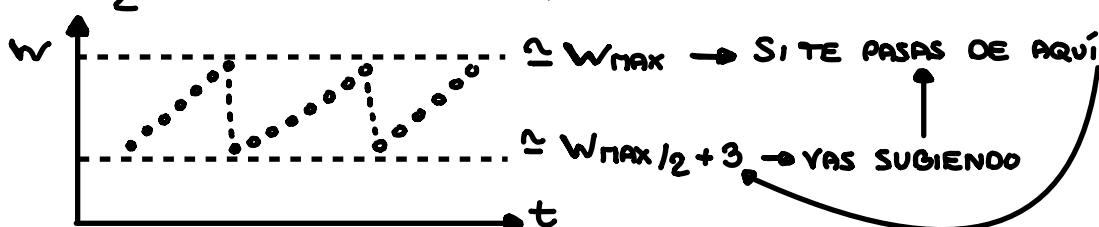
$$\Rightarrow \max(w_{\text{segmentos}}) = \left\lceil \frac{100}{1.2} \right\rceil = \underline{\underline{84 \text{ SEGMENTOS}}}$$

b) SUPONIENDO EL ESTADO EN QUE LA CONEXIÓN YA FINALIZÓ LA ETAPA DE ARRANQUE LENTO, LA TASA DE TRANSMISIÓN SE MANTIENE CONSTANTE ENTRE EL LÍMITE DE CONGESTIÓN (W), VALOR EN EL QUE YA HAY MÁS SEGMENTOS DE LOS QUE ESA RED PERmite, Y EL VALOR DE "REINICIO" (ASÍ LE LLANO YO) QUE ES WANDU

SE DETECTAN PERÍODOS POR 3 ACK DUPLICADOS ($W/2 + 3$)

$$W_{\text{MEDIA}} = \frac{W_{\text{MAX}} + (\frac{W_{\text{MAX}}}{2} + 3)}{2} = \frac{3W_{\text{MAX}}}{4} + 1,5 = \frac{3*84}{4} + 1,5 = 64,5$$

DIAGRAMA



AHORA PARA CALCULAR LA TASA DE TRANSMISIÓN EN bps SERÍA LA VELOCIDAD MEDIA A LA QUE TRABAJARÍA NUESTRO ENLACE DE ANCHO DE BANDA 10Mbps.

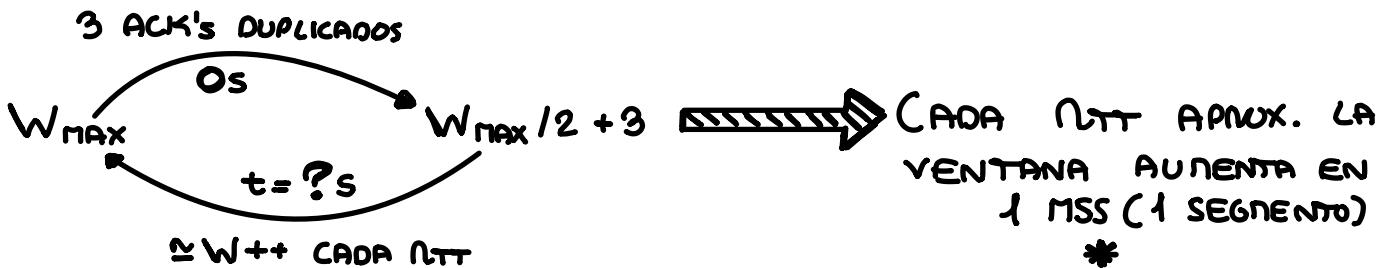
$$\begin{aligned} \beta_{\text{MEDIA}} &= \frac{\text{bits promedio}}{\text{RTT}} = \frac{W_{\text{MEDIA}} \text{ segmentos} \times \frac{\text{Nº bytes}}{\text{segmento}} \times \frac{8 \text{ bits}}{\text{byte}}}{\text{RTT segundos}} \\ &= \frac{W_{\text{MEDIA}} \text{ segmentos} \times \frac{\text{bits}}{\text{segmento}}}{\text{RTT segundos}} = \frac{64,5 \times 12000}{0,1} = 7740000 \frac{\text{bps}}{\text{bps}} \\ &\approx 7,74 \text{ Mbps} \end{aligned}$$

$\approx 77\%$

CAPACIDAD

ENLACE

c) AHORA NOS PIDEN EL TIEMPO EN LLEGAR DE W_{MAX} A W_{MAX} OTRA VEZ COMO EN EL DIAGRAMA ANTERIOR.

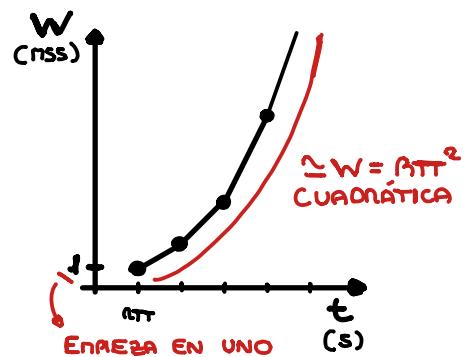


Por tanto para pasar de $W_{\text{MAX}}/2 + 3$ a W_{MAX} hacen falta $W_{\text{MAX}}/2 - 3$ ciclos (β_{RTT}) \Rightarrow Nº Ciclos = $84/2 - 3 = 39$ ciclos
 $\Rightarrow \underline{t = 39 \times 100 \text{ ms} = 3,9 \text{ s}}$

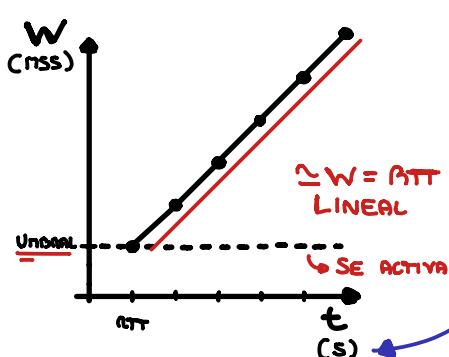
* ESTA CONCLUSIÓN ES UNA SIMPLIFICACIÓN PORQUE LA FÓRMULA REAL ES $W+ = \text{MSS}^2/W$ (EJERCICIO 13) POR CADA ACK, PERO ESTA SIMPLIFICACIÓN SUELE SER CORRECTA PARA ESTOS EJERCICIOS PORQUE ES COHERENTE CON LA TEORÍA DE QUE CRECE LINEALMENTE CON RESPECTO A β_{RTT} .

EJERCICIO 16

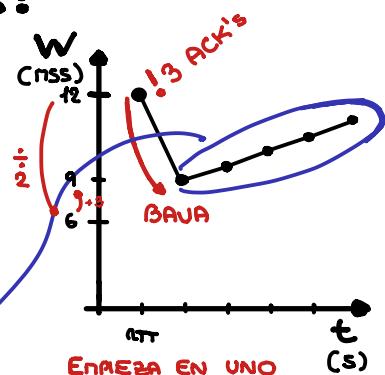
TCP RENO TIENE TIEMPOS POSIBLES ESTADOS:



ANILANQUE LENTO



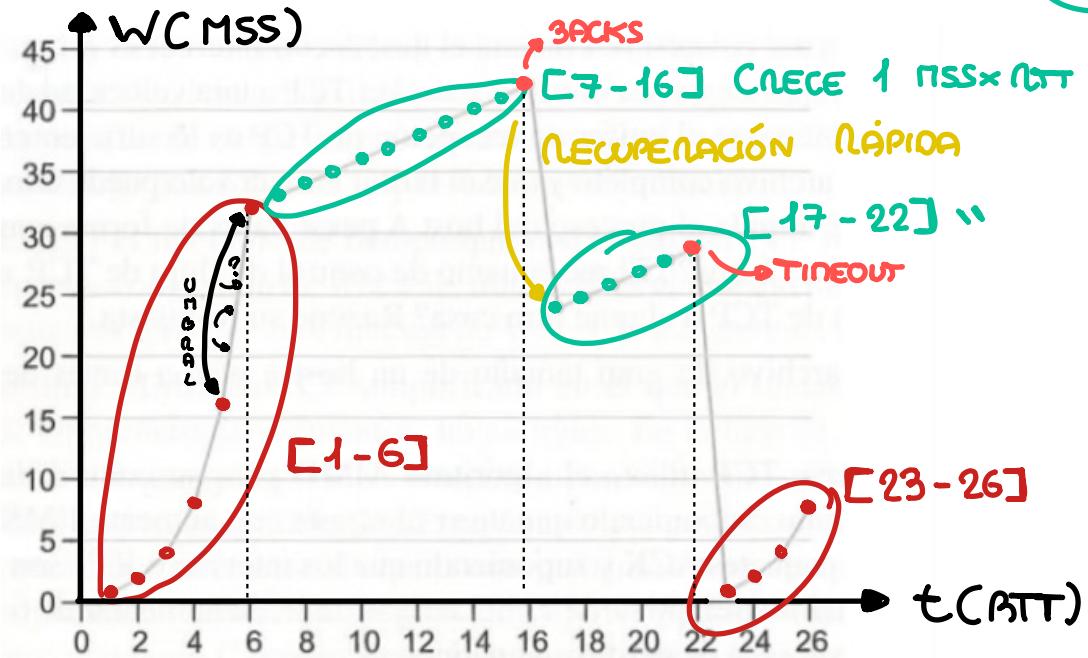
EVITACIÓN DE LA CONGESTIÓN



RECUPERACIÓN RÁPIDA

a) ANILANQUE LENTO SE PARECE MÁS A LO **NODEADO EN NOJO**.

b) EVITACIÓN DE LA CONGESTIÓN SE PARECE MÁS A **ESTO**



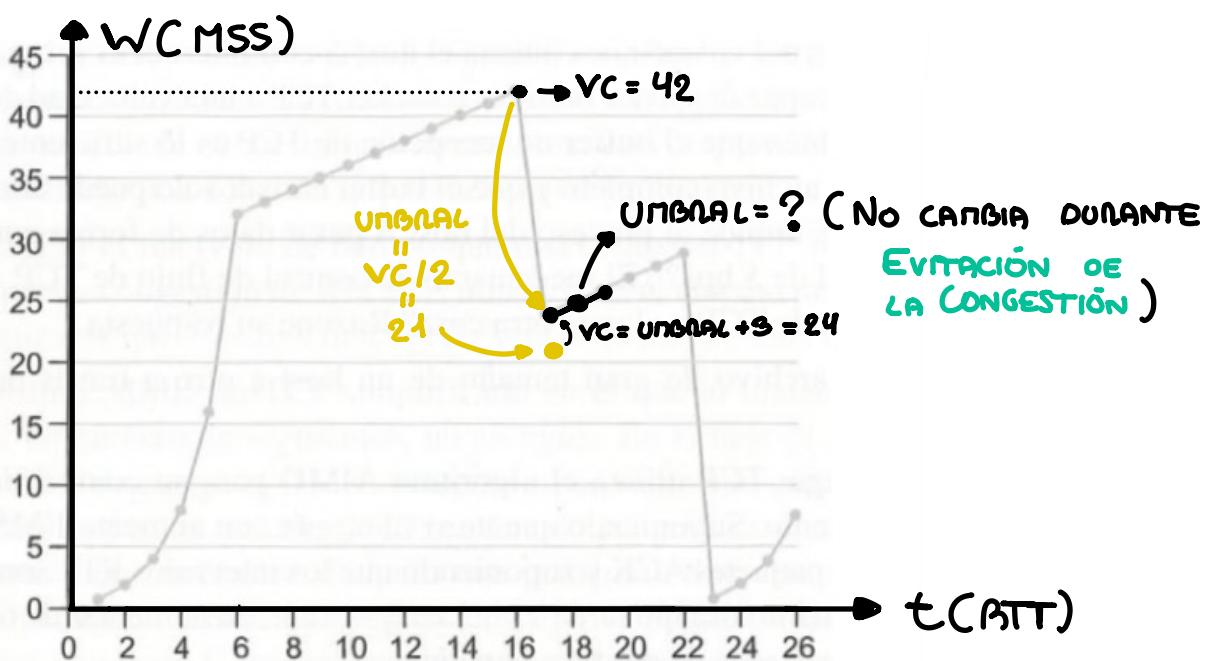
c) CICLO 16 CONECTA DOS EVITACIONES DE LA CONGESTIÓN A TRAVÉS DE UNA RECUPERACIÓN RÁPIDA, PRODUCIDA DE LA LLEGADA DE 3 ACK Duplicados (PERÍODO PAQUETE)

d) SIEMPRE QUE TCP (TAHUE TAMBÍEN) PASA DE CUALQUIER ESTADO A ANILANQUE LENTO ES PORQUE SE HA PRODUCIDO UN TIMEOUT (FIN DE TEMPORIZACIÓN).

e) EXACTAMENTE 32 YA QUE TCP TRANSICIONA DESDE ANILANQUE LENTO A EVITACIÓN DE LA CONGESTIÓN CUANDO SE RECIBE EL

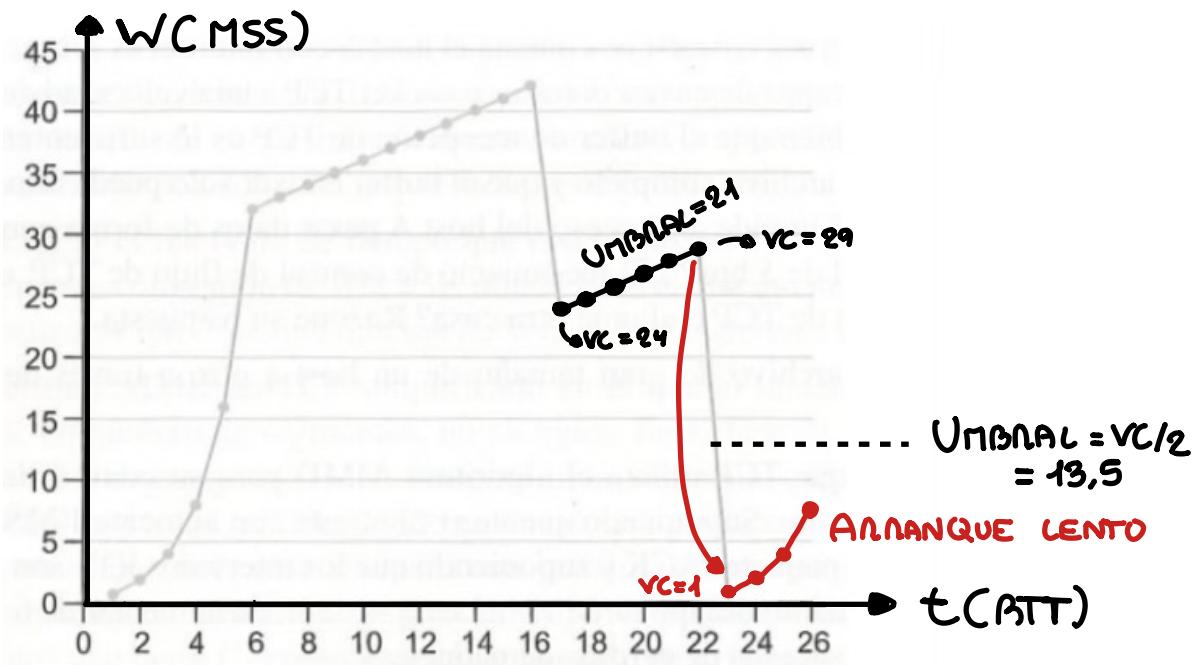
ACK QUE PERmite que $W = \text{UMBRAL}$, si RESTAN ACK POr LLEGAR EN ESE CICLO como SIMPLIFICAMOS QUE **EVITACIÓN DE LA CONGESTIÓN** AUMENTABA 1 MSS POR CADA RTT como YA PASÓ EL RTT PARA LA VEZ QUE LLEGARON LOS ACK "SE QUEDA AHÍ ESTANCIADO".

f)



$$\text{UMBRAL} (\text{Ciclo 18}) = 21$$

g)



$$\text{UMBRAL} (\text{Ciclo 24}) = 13,5 \rightarrow \text{REDONDEO } 15$$

h) EN QUE CICLO SE ENVÍA EL SEGMENTO N° 100?

ARRANQUE LENTO: $1-6 \Rightarrow 2^6 - 1 = 63$ (1, 2, 4, 8, 16, 32...)
 $N \rightarrow 2^{N-1}$

EVITACIÓN CONGESTIÓN:

$$7 \rightarrow +33 \Rightarrow 63 + 33 = 96 < 100$$

$$8 \rightarrow +34 \Rightarrow 96 + 34 = 130 > 100 \rightarrow \text{CICLO 8}$$

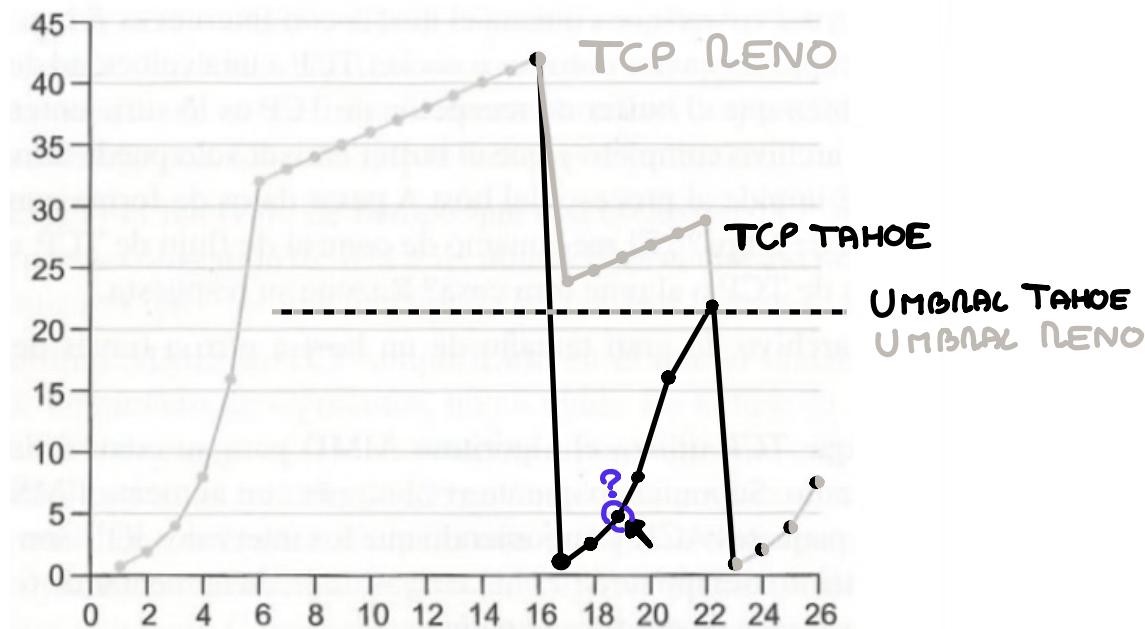
i) EN 26 RTT LLEVAVANOS 4 CICLOS DE ARRANQUE LENTO.

⇒ LA VC EN ESE CICLO ERA 8.

$$\text{NUEVO UNBRAL} = \text{VC}/2 = 8/2 = 4$$

$$\text{NUEVA VENTANA} = \text{NUEVO UNBRAL} + 3 = 7$$

j)



$$VC = 42 \Rightarrow \text{UNBRAL IGUAL QUE TAHOE} = 21$$

VENTANA NUEVA = 1 PORQUE CAMBIA A ARRANQUE LENTO.
(Como es ciclo 19 VC=4)

K) SI SE PRODUCE UN FALLO EN 22 TENEMOS EL DIAGRAMA DIBUJADO ANTERIORMENTE (PORQUE NO HAY QUE OLVIDAR LOS 3 ACK's DUPLICADOS, LA EJECUCIÓN ES CON TAHOE LOS 22 CICLOS ANTERIORES)

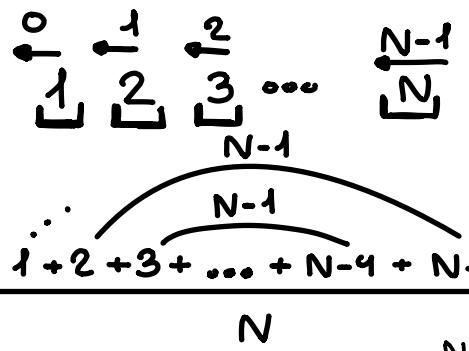
ARRANQUE LENTO : 5 PASOS $(17-21) = 2^5 - 1 = 31$ SEGMENTOS.

LUEGO EN EL CICLO 22 TAHOE SUPERA AL UNBRAL Y SE DETIENE
EN $CV = \text{UNBRAL} = 21$ SEGMENTOS

$$\begin{aligned} &\text{CICLOS } 17-21 \\ &\text{TOTAL} = 31 + 21 = 52 \\ &\quad \downarrow \\ &\quad \text{CICLO 22} \end{aligned}$$

EJERCICIO 17

CADA PAQUETE (Nº i DEL TOTAL DE N) TENDRÁ EXACTAMENTE $i-1$ PAQUETES DELANTE, ES DECIR $(i-1)*t_{trans}$ DE TIEMPO DE ESPERA EN LA COLA.



→ PEDIA DE PAQUETES POR DELANTE
DE CADA UNO DE LOS N PAQUETES
SENÁ

$$M_p = \frac{0 + 1 + 2 + 3 + \dots + N-4 + N-3 + N-2 + N-1}{N} = \frac{\frac{N}{2}N - 1}{N} = \frac{N-1}{2}$$

$$t_{\text{NEOIO-COLA}} = t_{\text{NEOIO-PAQUETE}} * M_p = t_{\text{TRANS}} \times \frac{N-1}{2} = \frac{L(N-1)}{2B}$$

$\frac{L \text{ bytes}}{B \text{ bytes/s}}$

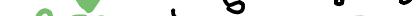
EJERCICIO 18

- ### a) RANGO DE DIRECCIONES DESTINO

- INTERFAZ

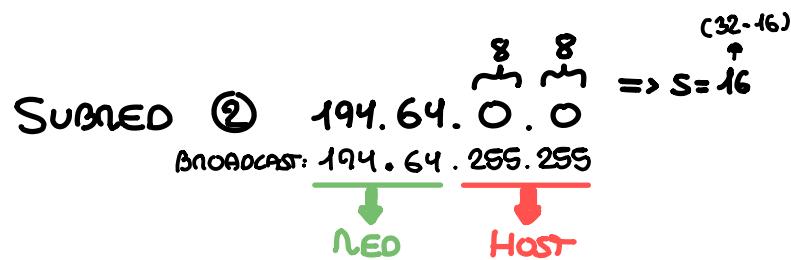
- ④ 194.32.0.0 - 194.63.255.255
⑤ 194.64.0.0 - 194.64.255.255
⑥ 194.65.0.0 - 195.127.255.255

- 0123

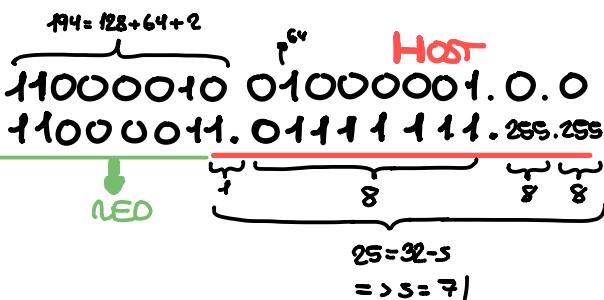
Subnet ①  → Host

* LA PARTE CONSTANTE ENTRE LA RED BASE Y LA DE BROADCAST
ES LA PARTE IDENTIFICATIVA DE LA SUBRED. LA PARTE DE RED
NUNCA PUEDE CAMBIAR PARA UNA DIRECCIÓN DE LA SUBRED.

(4) 194.32.0.0/11 → 0

SUBRED ② $194.64.0.0$ $\xrightarrow[8]{\quad}$ $\xrightarrow[8]{\quad}$ $\Rightarrow S=16$
 $(32-16)$
 Broadcast: $194.64.255.255$


② $194.64.0.0/16 \rightarrow 1$

SUBRED ③ $194 = 128 + 64 + 2$
 $11000010 01000001.0.0$ Host
 $11000011.0111111.255.255$
 Broadcast:


$194.0.0.0/7 \rightarrow 2$

AHORA HAY UN PROBLEMA HAY VARIAS SUBREDES CONTENIDAS EN ③ QUE NO FORMAN PARTE DEL RANGO ESPECIFICADO, TENEMOS QUE IDENTIFICAR HUECOS DONDE HAYA OTRAS REDES CONTENIDAS

DIVIDIDOS EN 2.

RANGO $[194.65.0.0 - 195.127.255.255]$

$194.0.0.0/8 \rightarrow$ No TODO \times
 $195.0.0.0/8 \rightarrow$ TANPOW, QUEDA PENDIENTE \times

DIVIDIDOS EN 2 LA PRIMERA.

$194.0.0.0/9 \rightarrow$ No TODO \times
 $194.128.0.0/9 \rightarrow$ TODO CONTENIDO \checkmark

DIVIDIDOS EN 2 LA PRIMERA.

$194.0.0.0/10 \rightarrow$ AQUÍ ESTÁ LA SUBRED ① ✓
 ANTES HAY UN HUECO

$194.64.0.0/10 \rightarrow$ AQUÍ ESTÁ LA SUBRED ②, PERO
 YA ESTÁ PEGADA LA SUBRED ③, NO HAY HUECO.

DIVIDIDOS EN 2 LA PRIMERA.

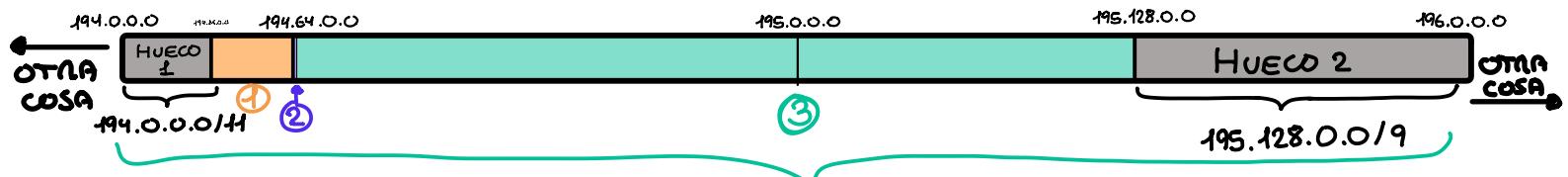
$194.0.0.0/11 \rightarrow$ Es HUECO NO FORMA PARTE DE ③

$194.32.0.0/11 \rightarrow$ SUBRED ① OCUPA TODO

DIVIDIENDOS AHORA LA SEGUNDA O EL PRINCIPIO QUE ESTABA PENDIENTE:

195.0.0.0/9 → TODO CONTENIDO ✓
 195.128.0.0/9 → NADA, ESTE ES NUESTRO SEGUNDO HUECO ✗

AHORA APLICAMOS UN RESTA DE CONJUNTOS PARA EXCLUIR ESTOS HUECOS DE ③, ASOCIANDOLO A LA TABLA COMO OTRA ENTRADA MÁS CON INTERFAZ DE SALIDA 3



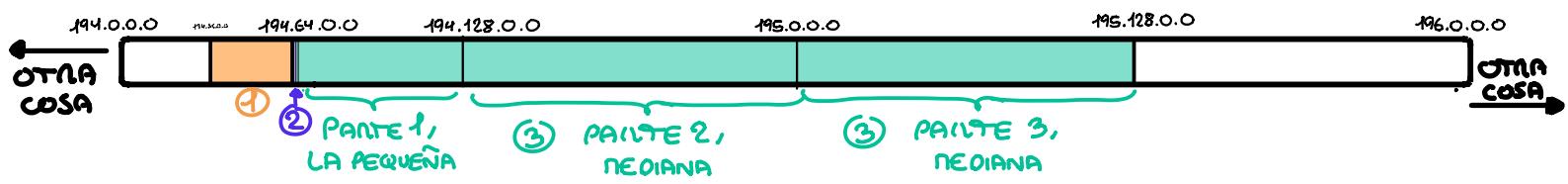
③ OTRA VEZ DESDE EL PRINCIPIO SI NO PONEROS ESE "HUECO" COMO OTRA SUBRED MÁS. POR EL PRINCIPIO DEL PREFIJO MÁS LARGO TODO LO QUE ENTRE EN HUECO NO ENTRA EN ③.

TABLA DE ENVÍO

SUBRED	MÁSCARA	INTERFAZ	CONTENIDO
194.32.0.0/11	255.224.0.0	0	"SUBRED ①"
194.64.0.0/16	255.255.0.0	1	"SUBRED ②"
194.0.0.0/11	255.224.0.0	3	"HUECO 1"
194.0.0.0/7	254.0.0.0	2	"SUBRED 3"
195.128.0.0/9	255.128.0.0	3	"HUECO 2"
0.0.0.0/0	0.0.0.0	3	"OTRO CASO"

ESTA TABLA ES EQUIVALENTE A LA DE LA SOLUCIÓN DE MI PROFESOR SOLO QUE EL DIVIDE LA SUBRED 3 EN 3 REGIONES Y HACE LA UNIÓN

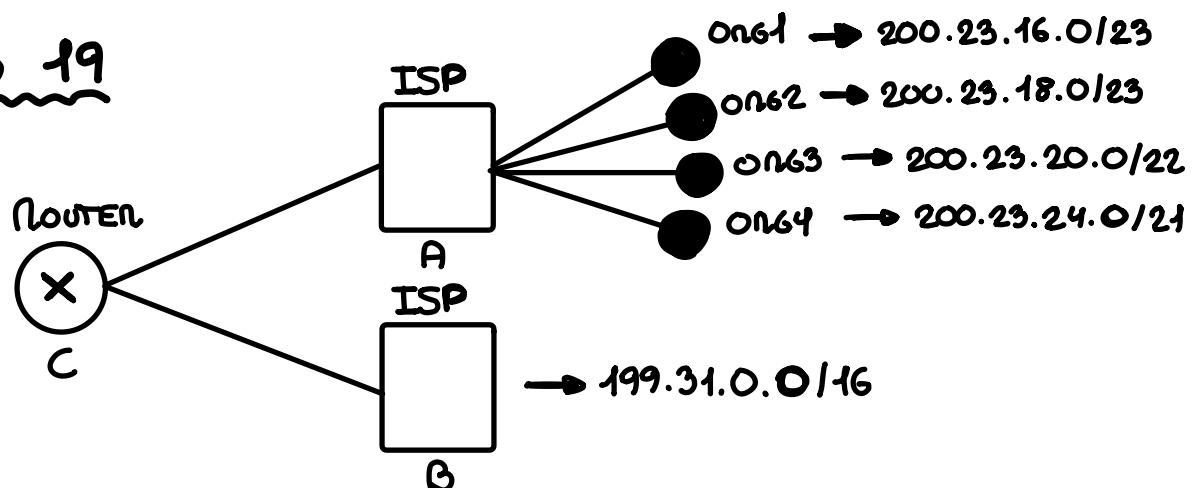
SUBRED	MÁSCARA	INTERFAZ	CONTENIDO
194.32.0.0/11	255.224.0.0	0	"SUBRED ①"
194.64.0.0/16	255.255.0.0	1	"SUBRED ②"
194.64.0.0/10	255.192.0.0	2	"SUBRED ③ P1"
194.128.0.0/9	255.128.0.0	2	"SUBRED ③ P2"
195.0.0.0/9	255.128.0.0	2	"SUBRED ③ P3"
0.0.0.0/0	0.0.0.0	3	"OTRO CASO"



b)

- 1) 194.145.81.85 SI LE APLICAMOS $\&\&$ (254.0.0.0) NOS DA 194.0.0.0,
PARA LA PRIMERA TABLA Y SI LE APLICAMOS $\&\&$ (255.128.0.0) NOS DA
194.128.0.0 \Rightarrow Subred 3, INTERFAZ 2.
 - 2) 194.64.95.60 SI LE APLICAMOS $\&\&$ (255.255.0.0) NOS DA 194.64.0.0
PARA LA PRIMERA TABLA Y SO MISMO SUcede PAlA LA SEGUNDA
TABLA \Rightarrow Subred 2, INTERFAZ 1.
 - 3) 194.28.13.255 SI LE APLICAMOS $\&\&$ (255.224.0.0) NOS DA 194.0.0.0
PARA LA PRIMERA TABLA Y PAlA LA SEGUNDA AL HACER EL AND LÓGICO
NO COINCIDE CON NINGUNA RED BASE EXCEPTO LA POR DEFECTO (0.0.0.0)
POR TANTO \Rightarrow "HUECO 1/OTRO CASO", INTERFAZ 3.

EJERCICIO 19



VAMOS A INTENTAR UNIFICAR TODAS LAS DIRECCIONES DEL ISP A, BAJO UNA SUBRED QUE SERÁ LA ENTRADA DEL ROUTER.

200.00010111.00010000.00000000
 8 8 3 HOS
 23 => PLANGO 200.23.16.0 2 Dos
 200.23.17.255 2 CLASE C

ENTONCES 200.23.18.0/23 EMPIEZA EN DONDE PLAGA EL ANTERIOR BLOQUE DE DIRECCIONES, COMO TIENE EL MISMO SUFIJO ABARCA DOS REDES CLASE C LLEGANDO HASTA 200.23.20.0 (NO INCLUIDO) DONDE EMPIEZA EL BLOQUE DE DIRECCIONES DE LA ORGANIZACIÓN 3, QUE OCUPA 4 REDES CLASE C DEJANDO TODO EL BLOQUE DE DIRECCIONES 200.23.16.0 - 200.23.31.255 UBICARIO POR EL ISP 1. LO QUE ES EQUIVALENTE A 16 REDES CLASE C.

Org. 1 → RED 123 } SUPERRED 122 }
 Org. 2 → RED 123 } SUPERRED 121 }
 Org. 3 → RED 122 }
 Org. 4 → RED 121 } SUPERRED 120 }

VAMOS A COMPROBAR AHORA CON LA DIRECCIÓN BASE Y LA DIRECCIÓN DE BROADCAST SI PODEROS ENCONTRAR TODO BAJO UNA SOLA SUPERRED.

CANTIDAD

BASE 8 11001000.00010111.0001 0000.00000000

BROADCAST 8 11001000.00010111.0001 1111.00000000

4 HOST 8

$32 - 5 = 27 \Rightarrow S = 20$ => COINCIDE CON EL SUFIXO QUE CALUANOS ANTES

ENCONTRAMOS TODO EL ISP 1 BAJO LA RED 200.23.16.0/20
EL ISP 2 SOLO TIENE UNA RED, YA SERÁ LA MISMA QUE LA ENTRADA DEL ROUTER.

TABLA DE REENVÍO

RED	MÁSCARA	DESTINO
200.23.16.0/20	255.255.240.0	ISP1
199.31.0.0/16	255.255.0.0	ISP2
0.0.0.0/0	0.0.0.0	DESCONOCIDO

b) SI LA ORGANIZACIÓN 1 PASA A DEPENDER DEL ISP2, POR EL PRINCIPIO DEL PREFIJO MÁS LARGO PODEROS PONER SU RED 200.23.18.0/23 COMO OTRA ENTRADA CON DESTINO AL ISP 2

TABLA DE REENVÍO

RED	MÁSCARA	DESTINO
200.23.16.0/20	255.255.240.0	ISP1
199.31.0.0/16	255.255.0.0	ISP2
200.23.18.0/23	255.255.254.0	ISP2
0.0.0.0/0	0.0.0.0	DESCONOCIDO

c) PAQUETE DESTINO 200.23.19. 60 PRUEBA A HACER

200.23.00010011.60
& 255.255.1110000.0

= 240.23.00010000.0 QUE ES EXACTAMENTE LA

RED BASE, PERTENECE A ESA SUBRED.

A DIFERENCIA DEL EJEMPLO CON LA SEGUNDA TABLA NO HAY MARQUES PREFIUDS EN LA TABLA ASÍ QUE NO VA A PODER SALIR POR OTRO INTERFAZ DIFERENTE A ISP4.

SI TENEMOS LA SEGUNDA TABLA LUEGO LA PRIMERA RED COINCIDE, LA SEGUNDA NO Y EN LA TERCERA.

$$\begin{array}{r} \text{200.23.00010011.160} \\ \text{255.255.11111110.0} \\ \hline \end{array}$$

200.23.18.0 → RED BASE => SALE POR ESTA YA QUE EL PREFIJO 23 > 20.

EJERCICIO 20

VAMOS A ASIGNAR LAS REDES POR ORDEN (A-F), COMO TENEMOS UN BLOQUE 193.144.130.0/23 TENEMOS 512 INTERFACES TOTALES, NOS LLEGARÁ PERFECTAMENTE PARA HACER LA SIGUIENTE ASIGNACIÓN.

A → 256 INTERFACES → 1 RED CLASE C CON 256 INTERFACES, LA PRIMERA SUBRED /24 DE NUESTRO BLOQUE: 193.144.130.0/24.

B → 120 INTERFACES, COMIENZA EN DONDE ACABA A Y LE ENTREGAMOS UN BLOQUE /25 CON 128 INTERFACES: 193.144.131.0/25.

C → 60 INTERFACES, COMIENZA DONDE ACABA B Y LE ENTREGAMOS UN BLOQUE /26 CON 64 INTERFACES: 193.144.131.128/26.

PARA D, E Y F COMO SÓLO NECESITAN 2 INTERFACES LLEGA CON 3 SUBREDES /32 CONTIGUAS A PARTIR DE C (COGENOS CADA UNA CON 4 DIRECCIONES PARA NO OCUPAR LA BASE NI LA BROADCAST).

D: 193.144.131.192/30

E: 193.144.131.196/30

F: 193.144.131.200/30

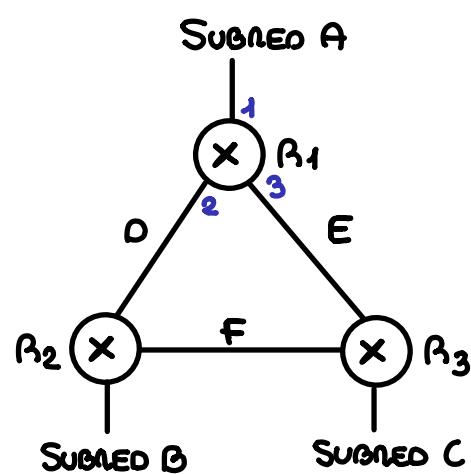
Ocupando el rango [193.144.130.0 - 193.144.131.203] del total [193.144.130.0 - 193.144.131.255]

TABLA SUBREDES

SUBRED	RANGO DE DIRECCIONES	IDENTIFICATIVO
193.144.130.0/24	[193.144.130.0 - 193.144.130.255]	SUBRED A
193.144.131.0/23	[193.144.131.0 - 193.144.131.127]	SUBRED B
193.144.131.128/22	[193.144.131.128 - 193.144.131.191]	SUBRED C
193.144.131.192/30	[193.144.131.192 - 193.144.131.195]	SUBRED D
193.144.131.196/30	[193.144.131.196 - 193.144.131.199]	SUBRED E
193.144.131.200/30	[193.144.131.200 - 193.144.131.203]	SUBRED F

AHORA EN EL ROUTER R1

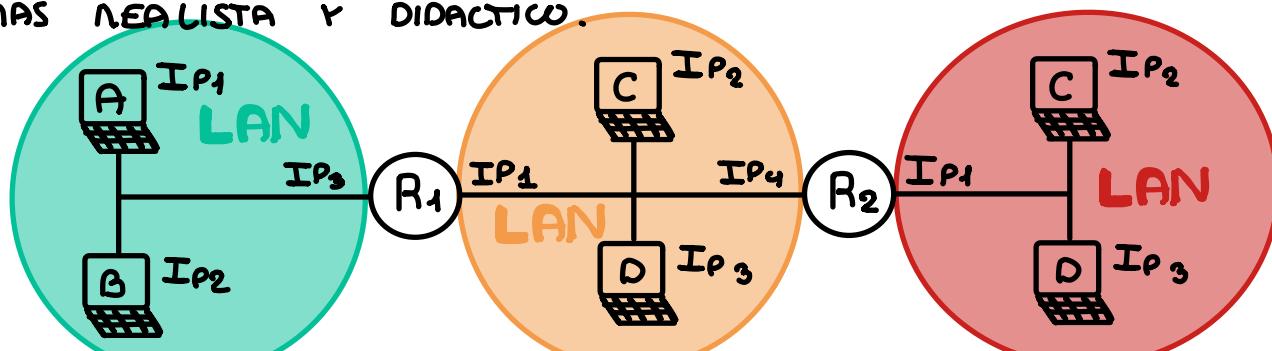
ENTRADA	INTERFAZ
193.144.130.0/24	1
193.144.131.0/23	2
193.144.131.128/22	3
193.144.131.192/30	2
193.144.131.196/30	3
193.144.131.200/30	X
0.0.0.0	



EJERCICIO 21

PARA CADA RED LAN DEBEMOS ASEGURARNOS QUE EL TAMAÑO DE LA SUBRED PERMITA CUBRIR EL NÚMERO DE HOST + INTERFACES DE ROUTERS CON DIRECCIONES UTILIZABLES. COMO CADA LAN DESEA DE TIERRA A DERECHA 3, 4 Y NUEVAMENTE 3 DIRECCIONES IP SE SELECCIONARON 3 REDES /29 QUE ANUNCIAN 6 DIRECCIONES UTILIZABLES (8 TOTALES - BASE Y BROADCAST) DE MANERA CONSECUITIVA CON EL OBJETIVO DE MINIMIZAR EL NÚMERO DE DIRECCIONES RESERVADAS.

OJO ESTO NO LO HACE TAN PRECISO EL PROFE EN LA SOLUCIÓN PERO AYUDA A ENTENDER MEJOR SUBNETTING Y QUE SEA UN EJEMPLO MÁS REALISTA Y DIDÁCTICO.

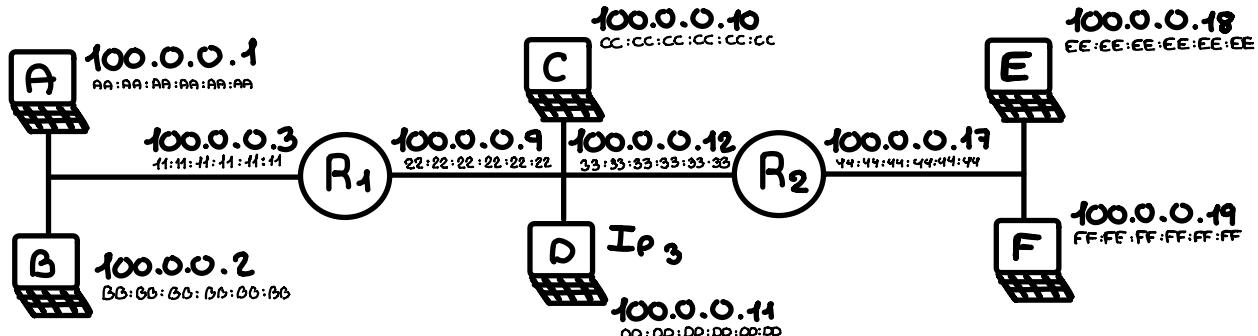


SUBRED 1: 100.0.0.0/29 (A, B, B₁)

SUBRED 2: 100.0.0.8/29 (B₁, C, D, B₂)

SUBRED 3: 100.0.0.16/29 (B₂, E, F)

LAS MAC LAS ASIGNANENOS DE MANERA ALEATORIA



LAN₁ 100.0.0.0/29

LAN₂ 100.0.0.8/29

LAN₃ 100.0.0.16/29

b) MENSAJE: A → F ≡ 3 TRAMAS (A → B₁, B₁ → B₂, B₂ → F)

b1) A → R₁

Formato Trama

MAC DESTINO 11:11:11:11:11:11	MAC ORIGEN AA:AA:AA:AA:AA:AA	IP ORIGEN 100.0.0.1	IP DESTINO 100.0.0.19	CABECERA ENLACE	DATOS	COLA ENLACE
----------------------------------	---------------------------------	------------------------	--------------------------	--------------------	-------	----------------

b1) B₁ → R₂

Formato Trama

MAC DESTINO 33:33:33:33:33:33	MAC ORIGEN 22:22:22:22:22:22	IP ORIGEN 100.0.0.1	IP DESTINO 100.0.0.19	CABECERA ENLACE	DATOS	COLA ENLACE
----------------------------------	---------------------------------	------------------------	--------------------------	--------------------	-------	----------------

b1) B₂ → F

MAC DESTINO FF:FF:FF:FF:FF:FF	MAC ORIGEN 44:44:44:44:44:44	IP ORIGEN 100.0.0.1	IP DESTINO 100.0.0.19	CABECERA ENLACE	DATOS	COLA ENLACE
----------------------------------	---------------------------------	------------------------	--------------------------	--------------------	-------	----------------

EJERCICIO 22

SUBNED	SIGUIENTE SALTO
128.96.164.0/22	Interfaz 0
128.96.170.0/23	Interfaz 1
128.96.168.0/23	R ₂
128.96.166.0/23 <por defecto>	R ₃
	R ₄

a) D_{IR} = 128.96.171.92 → (171 = 128 + 0 + 32 + 0 + 8 + 0 + 2 + 1)

SUBNED	MÁSCARA	MÁSCARA × D _{IR}	¿ == RED BASE?	PREFIJO
128.96.164.0/22	255.255.252.0	128.96.168.0	✗	/
128.96.170.0/23	255.255.254.0	128.96.170.0	✓	<u>23</u>
128.96.168.0/23	255.255.254.0	128.96.170.0	✗	/
128.96.166.0/23	255.255.254.0	128.96.170.0	✗	/

SALIDA: INTERFAZ 1

b) D_{IR} = 128.96.167.151 → (167 = 128 + 0 + 32 + 0 + 0 + 4 + 2 + 1)

SUBNED	MÁSCARA	MÁSCARA × D _{IR}	¿ == RED BASE?	PREFIJO
128.96.164.0/22	255.255.252.0	128.96.164.0	✓	<u>22</u>
128.96.170.0/23	255.255.254.0	128.96.166.0	✗	/
128.96.168.0/23	255.255.254.0	128.96.166.0	✗	
128.96.166.0/23	255.255.254.0	128.96.166.0	✓	<u>23</u> MÁXIMO

SALIDA: INTERFAZ Router R₃

c) $D_{IR} = 128.96.163.151 \rightarrow (163 = 128 + 0 + 32 + 0 + 0 + 0 + 1)$

SUBREO	MÁSCARA	$MÁSCARA \times D_{IR}$	$\overset{?}{=} MED\ BASE?$	PREFIJO
128.96.164.0/22	255.255.252.0	128.96.160.0	✗	/
128.96.170.0/23	255.255.254.0	128.96.162.0	✗	/
128.96.168.0/23	255.255.254.0	128.96.162.0	✗	/
128.96.166.0/23	255.255.254.0	128.96.162.0	✗	/

SALIDA: Por defecto ya que ninguna coincide \rightarrow Router 3y

d) $D_{IR} = 128.96.169.192 \rightarrow (169 = 128 + 0 + 32 + 0 + 8 + 0 + 0 + 1)$

SUBREO	MÁSCARA	$MÁSCARA \times D_{IR}$	$\overset{?}{=} MED\ BASE?$	PREFIJO
128.96.164.0/22	255.255.252.0	128.96.168.0	✗	/
128.96.170.0/23	255.255.254.0	128.96.168.0	✗	/
128.96.168.0/23	255.255.254.0	128.96.168.0	✓	<u>23</u>
128.96.166.0/23	255.255.254.0	128.96.168.0	✗	/

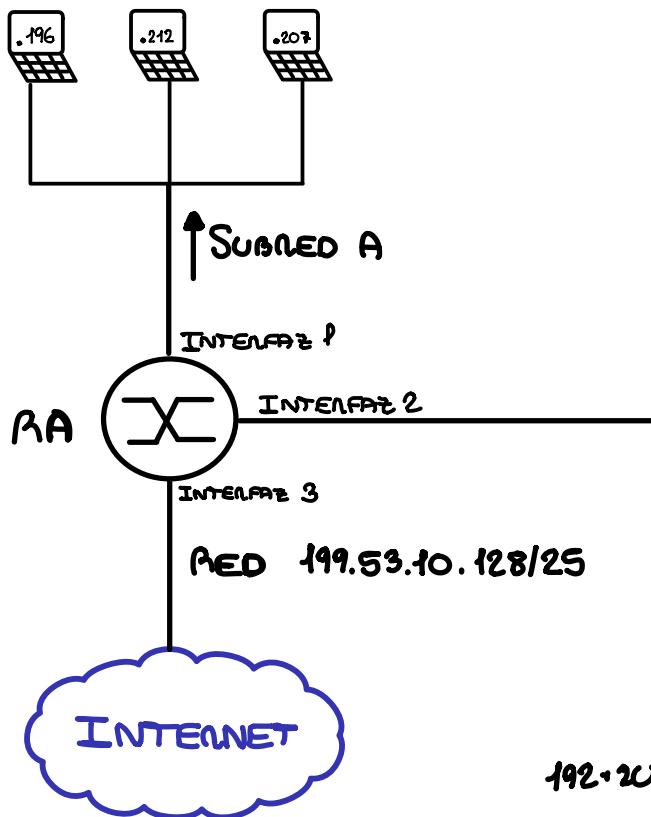
SALIDA: INTERFAZ Router 2

e) $D_{IR} = 128.96.165.121 \rightarrow (165 = 128 + 0 + 32 + 0 + 0 + 4 + 0 + 1)$

SUBREO	MÁSCARA	$MÁSCARA \times D_{IR}$	$\overset{?}{=} MED\ BASE?$	PREFIJO
128.96.164.0/22	255.255.252.0	128.96.164.0	✓	<u>22</u>
128.96.170.0/23	255.255.254.0	128.96.164.0	✗	/
128.96.168.0/23	255.255.254.0	128.96.164.0	✗	/
128.96.166.0/23	255.255.254.0	128.96.164.0	✗	/

SALIDA: INTERFAZ O

EJERCICIO 23



Subred A: Cono mínimo alcanzan [199.53.10.196 - 199.53.10.215]

$$\begin{array}{r} 128+64 +4 = 196 \\ \text{MIN: } 199.53.10.11000100 \\ \text{MAX: } 199.53.10.11010100 \\ \hline 3 \quad S \end{array} \Rightarrow S = 32 - 5 = 27$$

Conún: 199.53.10.192/27

ESTA SUBRED NOS OFRECE 30 DIRECCIONES EFECTIVAS/UTILIZABLES.

=> PARA 1 INTERFAZ Y 3 Host nos llega

a.) DETERMINA TODAS LAS POSIBLES OPCIONES

No hay ningún otro host con IP > 199.53.10.192

Por tanto se podrían hacer /27 o /26 incluso pero siempre con red base 199.53.10.192.

→ OJO!! /25 no podría ser porque 199.53.10.11000000 \leq 199.53.10.128

Sólo podemos estirar máscara hacia izquierda cuando haya 0's en la red base.

b) LA SUBRED D CONTIENE LOS Host .161 y .166 SI COLLOCAMOS UNA SUBRED CON 8 DIRECCIONES DESDE .0 HASTA .8 NINGUNO DE ELLAS QUEDARÁ CON OIR BASE NI BROADCAST ADÉMAS QUE NOS QUEDARÁN 3 DIRECCIONES LIBRES A NACIONES PARA ASIGNARLE UNA A LA INTERFAZ NC2 DEL ROUTER RC.

MIN: 193.53.10.10100000
MAX: 193.53.10.10100110

129 [193.53.10.160 - 193.53.10.167] ✓

128 [193.53.10.160 - 193.53.10.175] ✓

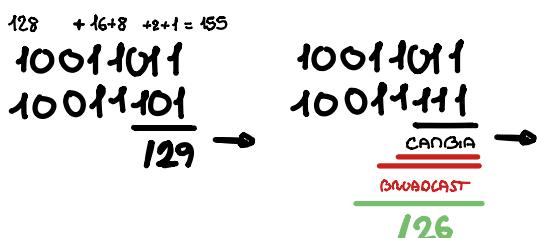
127 [193.53.10.160 - 193.53.10.192] ⊗

↳ PROBLEMA: INCLUIR C,E DENTRO DE D

TENEMOS DOS OPCIONES 193.53.10.160/8
193.53.10.160/9

c) ¿ 193.53.10.155 → 193.53.10.159 ?

SIN NINGÚN TIPO DE PROBLEMA YA QUE .159 NO ES UNA IP CONTENIDA EN NINGUNA OTRA SUBRED, PERO SI CAMBIARÍA LA MÁSCARA DE ESTA SUBRED DE /29 A /26 YA QUE SI NO LA IP 193.53.10.159 ESTARÍA OCUPANDO LA DIRECCIÓN DE BROADCAST CON CUALQUIER PREFIJO MAYOR.



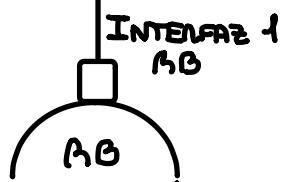
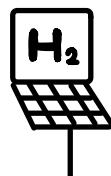
Como en el apartado anterior NEGAMOS PISAR OTRAS SUBREDES CON UNA SUBRED, EN ESTE CASO TAMBÉN YA QUE LA NUEVA B ESTARÍA INCLUYENDO A E, D, C. => NO SE PUEDE

d) CONECTANDO CON EL APARTADO ANTERIOR SI B TUVESE LOS HOST

193.53.10.155



193.53.10.157



CON EL PREFIJO MINIMO TENEMOS LA SUBRED.

193.53.10.10011000
B H

193.53.10.152/29

SI AVANZÓ UN BIT HACIA LA IZQUIERDA HACIA LA MÁSCARA 128, ESTARÍA INCLUYENDO LA DIRECCIÓN

193.53.10.144 - 193.53.10.160

NINGÚN PROBLEMA

EL PROBLEMA SURGIRÍA EN 193.53.10.128/26

BASE: 193.53.10.10000000

↳ Esto YA COBERTURA TODAS LAS IPs DE LOS HOST DE LA SUBRED C,D Y E

ENTONCES SUBRED B CON TAMAÑO MÁXIMO 193.53.10.128/27

RANGO: [193.53.10.128 - 193.53.10.159] → Son miembros de B ✓

EN ESTE CASO EL PROFESOR ELIGE 193.53.10.152/29 COMO LA PARTE CLÁSICA, ENTONCES PODEMOS CONCLUIR QUE SI CABALEIRO PREGUNTA POR LA PARTE CLÁSICA SE REFIERE AL PARTE SUFijo. (YO HABÍA ENTENDIDO LA SUBRED MÁS GRANDE POSIBLE)

e)

SUBRED E:

199.53.10.10110110
199.53.10.10110011
_____ NED Host

199.53.10.176/29

SUBRED D:

199.53.10.10100001
199.53.10.10100101
199.53.10.10100110
_____ NED Host

199.53.10.160/29

Como son dos /29 NO CONSECUTIVAS YA QUE ESTÁ

199.53.10.168/29 ENTRE NEDIPS

MÍNIMA FORMA DE AGUARDAR D Y E ES BAJO UNA /27



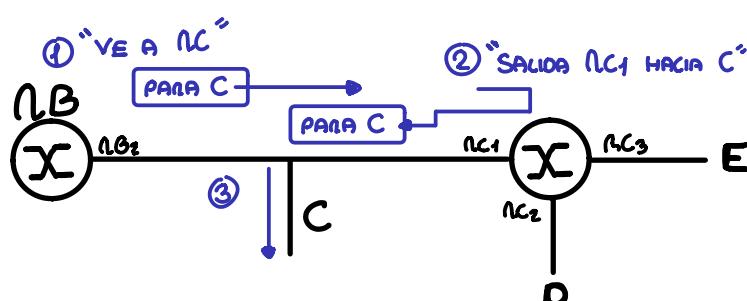
SUBRED C:

199.53.10.10111001
199.53.10.10111100
_____ NED Host
199.53.10.184/29

LA SOLUCIÓN DEL PROFESOR PASA POR REENVÍAR TODO LO PERTENECIENTE AL BLOQUE 199.53.10.160/27 HACIA EL ROUTER RC, PARA QUE SI SE TRATA DE UN DATAGRAMA DIRIGIDO A C RC LO REENVÍE POR LA INTERFAZ RC1.

IMPORTANTE: UN ROUTER SI PUEDE "NEGOTIAR" (REENVÍAR UN DATAGRAMA POR LA MISMA INTERFAZ DE SALIDA POR LA QUE HABÍA ENTRADO) YA QUE SOLO SE BASA EN SU TABLA DE ENRUTAMIENTO Y NO EXISTE NINGÚN MECANISMO QUE LO PROHIBA. SIN ENCONTRAR UN SWITCH NO PUEDE NEGOTIAR UNA TRAMA POR LA MISMA INTERFAZ QUE HABÍA LLEGADO, YA QUE SE DESCARTA INTERNAMENTE.

EFFECTO NEGOTIATE:



EN ESTE CASO VAMOS A AGREGAR LA SALIDA DE LA SUBRED C AL ROUTER NB PARA EVITAR TRÁFICO EXTRA Y ASÍ APLICANDO EL PRINCIPIO DEL PREFIJO MÁS LARGO AHORRARNOS TODAVÍA 1 ENTRADA PERO EVITANDO TRÁFICO INNECESARIO.

SIN AGREGAR	SOL. PROFESOR	MI SOLUCIÓN
{ ENTRADAS ANTIGUAS }	{ } 199.53.10.176/29, NB2 199.53.10.160/29, NB2 199.53.10.184/29, NB2	{ } 199.53.10.160/27, NC2 "TODA LO DE C VA A N3" "W DE C VA A LA 129, LO OTRO A NC"

(LA SOLUCIÓN DEL PROFESOR ES COMPLETAMENTE CORRECTA FORMALMENTE, ESTA SOLO ES UNA ALTERNATIVA DE DISEÑO PARA ENTENDER MEJOR CÓMO FUNCIONAN LOS PREFIJOS Y LOS ROUTERS).

f) PARTIENDO DE C-E-D AGREGADA Y B CONO 199.53.10.152/29, A PUEDE SER COMO ENTRADA POR DEFECTO (TODA LO QUE VAYA AL RESTO DE INTERNET O DIRIGIDO A A VA A PASAR POR RA) O COMO ENTRADA PROPIA.

COMO DICE EL PROFESOR NO ES NECESARIO YA QUE LA SUBRED A YA SERÍA UNA ENTRADA DE A Y ASÍ TE AHORRARÍAS UNA ENTRADA EN LA TABLA DE NB.

g) POR SUPUESTO AGREGADA YA QUE NB TAMBÍEN LA TENDRÁ AGREGADA Y SERÁ NC QUIEN DECIDA O SI NB NO LA TIENE AGREGADA YA SERÁ QUIEN DECIDA A TRAVÉS DE SU TABLA.

PODRÍAMOS INCLUSO AGGRUPAR B CON E-C-D EN RA



Q) LA IP DESTINO DENTRO DE LA SUBRED E SERÁ LA IP DE LA INTERFAZ 3 DEL ROUTER NC.

HOST 199.53.10.182 - NC3, NC1 - NB2, NB3 - RA2, RA1 - Host 199.53.10.212

SUBRED E

SUBRED C

SUBRED F

SUBRED A

EJERCICIO 24

PARA SIMPLIFICAR UNA DIRECCIÓN IPv6 SE PUEDEN REALIZAR LOS SIGUIENTES CAMBIOS:

- ELIMINAR LOS 0 INICIALES DE UN BLOQUE DE 4 DÍGITS HEXADECIMALES 00a4→a4
- SUSTITUIR SECUENCIAS DE BLOQUES 0000₁₆ POR :: SOLO UNA VEZ POR DIRECCIÓN, YA QUE NO SE INDICAN CUANTOS BLOQUES SE ABREVIAN CON UN :: (DESDE 1 A 7 TEÓRICAMENTE)
- SEGÚN LA REGULACIÓN RFC 5952 SI TIENES DOS BLOQUES CON '0000' SEPARADOS SE SIMPLIFICA:
 - El de mayor tamaño ↴rfc:0000:dc01:ab72:0000:0000:f127
 - El de más a la izquierda ↴rfc:0000:0000:ab72:0000:0000:f127
 - Los demás bloques cumplen la primera regla :0000:→::0:

a) ::0f53:6382:a**b**00:67db:bb27:7332
 ^

Comprueba dos 0000

Dir. REAL: 0000:0000:0f53:6382:a**b**00:67db:bb27:7332

CORRECTA ✓

b) 7803:42f2:::88ec:d4ba:b75d:f1cd
 ^

ESTE SINOOOW **No EXISTE**, SERÍA ::

Dir. REAL: ✗

INCORRECTA ✗

c) ::4ba8:95cc::db97:4eab
 ^ ^

! HAY DOS :: ! NO SE PUEDE SABER CUANTOS 0000 VAN EN CADA UNO

Dir. REAL: ✗

INCORRECTA ✗

d) $\text{74dc} :: \text{02ba}$

COMPRISE SEIS 0000

DIR. REAL:

$74\text{dc}:0000:0000:0000:0000:0000:0000:\text{02ba}$

CONNECTA ✓ (AUNQUE EL 02ba PODRÍA SER 2ba.)

e) $::ffff:128.112.92.116$ → ESTO ES LA TRANSFORMACIÓN DE UNA IPv4 A IPv6.
8 bits => dir IPv4 ocupa 8 BITS HEXADECIMALES (32 BINARIOS)
COMPRISE 5 '0000'

DIR. REAL:

$0000:0000:0000:0000:0000:\text{ffff}:8070:\text{5c74}$

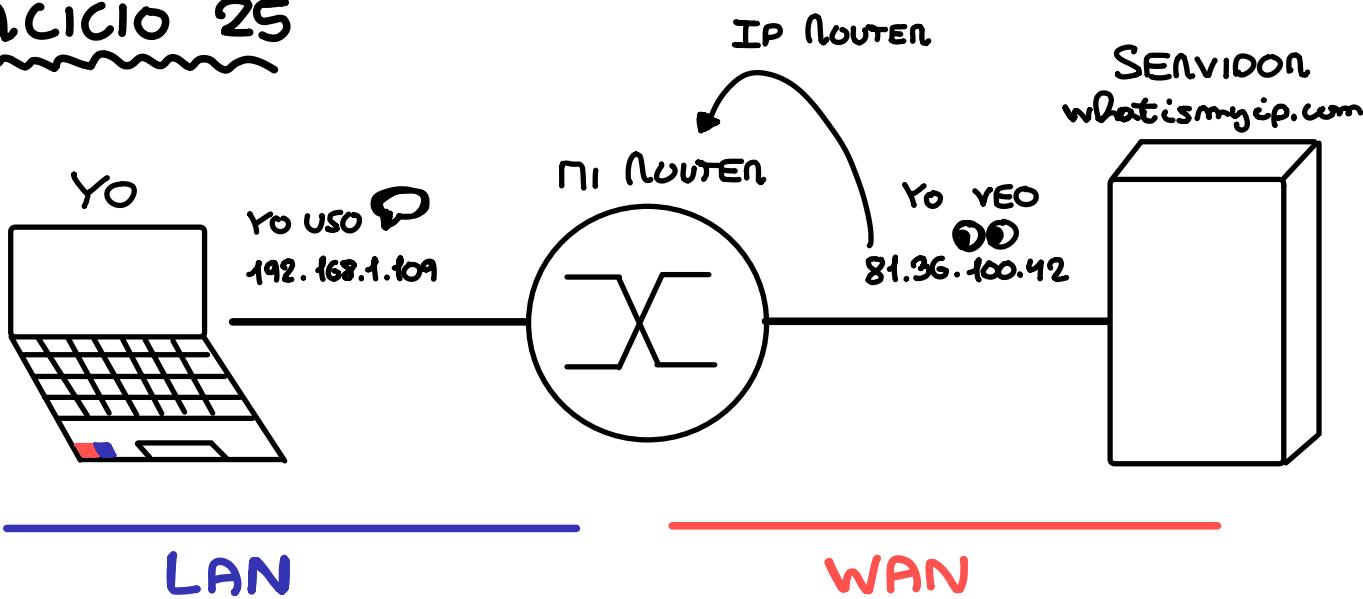
CONNECTA ✓

f) $::1$ → ES LA DIRECCIÓN LOOPBACK $\approx 127.0.0.1$ /IPv4

COMPRISE 7 '0000'

DIR. REAL: $0000:0000:0000:0000:0000:0000:0000:0001$

EJERCICIO 25



c) ANTES DE RESPONDER, LO ANTERIOR ESTE MECANISMO ES NAT (NETWORK ADDRESS TRANSLATION) PERMITE AGUARDAR HASTA UN TOTAL DE $2^{16} - 1024 = 64512$ HOST YA QUE LA DUPLA (IP, PUERTO₁) DE PUERTOS RESERVADOS UNA CONEXIÓN SE TRADUCE A (IP_{ROUTER}, PUERTO₂), EL LÍMITE ES EL NÚMERO DE PUERTOS POSIBLES EN EL ROUTER Y LA TABLA AUMENTA POR CADA CONEXIÓN QUE ABRE CADA HOST. DENTRO DE LA LAN EL HOST TIENE SU IP Y PUERTO Y SUS DATAGRAMAS NO REFLEJAN. FUERA DE LA

LAN ESTÁ LA WAN Y TODOS LOS DATAGRAMAS DEL HOST TIENEN LA IP DEL ROUTER Y EL PUERTO DE LA TABLA TRADUCIDO.

a) y b)

SI BUSCO www.cesga.es Y LA IP DEL SERVIDOR DE DONDE VAY A DESCARGAR EL HTML ES 193.144.34.236.

EN MI ROUTER SE TRADUCIRÁ MI IP Y EL PUERTO QUE UTILICE:

IP ORIGEN	PUERTO ORIGEN	IP TRADUCIDA	PUERTO TRADUCIDO
192.168.1.109	X	81.36.100.42	Y!

LAN
 (b), (o QUE VIAJA POR CASA)
 PRE ROUTER

WAN
 (c) (O QUE HAY FUERA)
 POST ROUTER

ESTE
 PUERTO DEBE
 SER ÚNICO EN LA
 TABLA

LA IP DESTINO Y PUERTO DESTINO NO CAMBIA, ES 193.144.34.236 (IP SERVICIO WEB www.cesga.es) Y 80 (PUERTO PARA CONSULTAS HTTP) O 443 (PUERTO CONSULTAS HTTPS) LOS DOS APARTADOS.

EJERCICIO 26

LA CAPA DE APLICACIÓN EN A ENTREGA A LA CAPA DE TRANSPORTE UN NENSAJE 4160 bytes (4096 ARCHIVO + 64 CABECERA).

LA CAPA DE TRANSPORTE PUEDE IMPLEMENTAR DOS PROTOCOLOS DISTINTOS.

TCP

NECESITA ENVÍAR DOS SEGMENTOS TCP DE CONTROL SYN Y ACK.

Como NO SUPONEMOS PIGGYBACKING NO ENVIANOS EL SIGUIENTE SEGMENTO CON DATOS UNIDO AL ACK.

Los dos PRIMEROS SEGMENTOS TCP:

SYN: 20 bytes (Sola CABECERA)
ACK: 20 bytes (Sola CABECERA)

ADENÁS LLEVAN EL BIT NF=1

UDP

ENPAQUETA TODO EN UN SOLO SEGMENTO UDP:

SEGMENTO 1: $\underbrace{4160}_{\text{DATOS}} + \underbrace{8}_{\text{CABECERA UDP}} = 4168 \text{ bytes}$

-
- LEYENDA —
- CONEXIÓN
 - ENTREGA DATOS

PARA INDICARLE A LA CAPA DE RED QUE NO PUEDE FRAGMENTARLOS

LOS 4160 BYTES DE DATOS LOS DIVIDE EN SEGMENTOS DE TAMAÑO MSS EXCEPTO EL ÚLTIMO.

$$\text{Nº SEGMENTOS: } \left\lceil \frac{\text{TAM MENSAGE}}{\text{TAM SEGMENTO}} \right\rceil$$
$$\left\lceil \frac{4160}{1400} \right\rceil = 3$$

LOS DOS PRIMEROS SEGMENTOS:

SEGMENTO 1: $\underbrace{1400}_{\text{DATOS}} + \underbrace{20}_{\text{CABECERA TCP}} = 1420$ bytes

SEGMENTO 2: $\underbrace{1400}_{\text{DATOS}} + \underbrace{20}_{\text{CABECERA TCP}} = 1420$ bytes

EL ÚLTIMO SEGMENTO TENDRÁ EL SIGUIENTE TAMAÑO

$$4160 - 1400 * 2 = 1360$$

SEGMENTO 3: $\underbrace{1360}_{\text{DATOS}} + \underbrace{20}_{\text{CABECERA TCP}} = 1380$ bytes

CAPA RED

AHORA ANGOS PROTOCOLOS ENTREGAN SUS SEGMENTOS A LA CAPA DE RED, PUEDE UTILIZAR DATAGRAMAS IPr4 o IPr6.

TCP + IPr4

SE AGREGAN 20 BYTES A CADA SEGMENTO PARA FORMAR LOS DATAGRAMAS.

ACK: 40
SYN: 40
S₁: 1440
S₂: 1440
S₃: 1400

< MTU ✓

UDP + IPr4

SE AGREGAN 20 BYTES DE CABECERA IPr4 PERO CUANDO EL DATAGRAMA SUPEREA EL MTU.

$$4188 >> 1500$$

$$\text{Nº FRAGMENTOS: } \left\lceil \frac{\text{TAM FRAGMENTO}}{\text{MTU}} \right\rceil$$

$$\left\lceil \frac{4188}{1500} \right\rceil = 3$$

SI ALGÚN DATAGRAMA SE PASASE DEL MTU ALGÚN ROUTER PODRÍA FRAGMENTARLO (NF=0) O ENVIAN UN ICMP "FRAGMENTATION NEEDED" PARA QUE TCP AVISTE

LOS PRIMEROS DOS FRAGMENTOS TENDRÁN UN TAMAÑO DE 1500 BYTES INCLUIDA LA CABECERA DE IPr6 Y EL BIT MF=1 PARA INDICAR QUE ESE DATAGRAMA ESTÁ COMPOSTO DE MÁS FRAGMENTOS.

EL MSS(NF=1). EN ESTE CASO, LOS DATOS DAN PARA QUE ESTÉ BIEN Y POR ESO EL PROFE PONE 'X', PORQUE DA IGUAL QUE CONVICTARIANDO TOME.

TCP + IPr6

SUCEDA LO MISMO QUE CON IPr4 PERO LAS CABECERAS OCUPAN 40 bytes.

ACK: GU
SYN: GU
S1: 1460
S2: 1460
S3: 1420

< MTU ✓

CON LA DIFERENCIA DE QUE IPr6 EL BIT NF NO EXISTE YA QUE NO PERMITE FRAGMENTAR DATAGRAMAS DE LA CAPA DE RED EN ROUTERS INTERMEDIOS. ENVIANDO UN MENSAJE ICMP, PARA QUE TCP ADAPTE EL MSS, AL HOST ORIGEN.

TCP IPr4

DATAGRAMA	id	TAMAÑO	MF	NF	offset
1	356	40	0	1	0
2	357	40	0	1	0
3	358	1440	0	X	0
4	359	1440	0	X	0
5	360	1400	0	X	0

IDENTIFICA SI FUE FRAGMENTADO DATAGRAMA SI QUEDAN MAS FRAGMENTOS

FRAGMENTO 1: $1480 + 20 = 1500$ (NF=1)
FRAGMENTO 2: $\underbrace{1480}_{\text{DATOS}} + \underbrace{20}_{\text{CABECERA}} = 1500$
DATOS CABECERA IPr6

FRAGMENTO 3: $4188 - \underbrace{3000}_{\text{DATOS SEGMENTO RESTO}} = 188$ (NF=0)

DATOS SEGMENTO RESTO UDP FRAGMENTOS

ADENAS EL CAMPO OFFSET INDICA EN BLOQUES DE 8 bytes LA POSICIÓN DE INICIO DEL FRAGMENTO CON RESPECTO AL SEGMENTO DE LA CAPA DE TRANSPORTE.

OFFSET SEGMENTO 1: 0 (PRIMER SEGMENTO)

SEGMENTO 2: $1480 / 8 = 185$

SEGMENTO 3: $1480 * 2 / 8 = 185 * 2 = 370$

UDP IPr4

DATAGRAMA	id	TAMAÑO	MF	NF	offset
1	356	1480	1	0	0
2	356	1480	1	0	185
3	356	1188	0	0	370

IGUAL PORQUE SON FRAGMENTOS DEL MISMO SEGMENTO UDP ¿QUEDAN FRAGMENTOS? DESPLAZAMIENTO BLOQUES 8 bytes

(TAMAÑO CON CABECERA IPr4, LO ÚNICO QUE CAMBIA ES QUE IPr6 TIENE 20 bytes MÁS Y NO EXISTIRÍAN LOS CAMPOS MF, NF Y OFFSET).

EN UDP IPr6 UNO NO SE PERMITE LA FRAGMENTACIÓN EL PRIMER ROUTER EN NUESTRA RED GENERARÍA UN MENSAJE ICMP AL VER QUE EL DATAGRAMA TIENE MAYOR TAMAÑO DEL QUE PERMITE SU ENLACE CON EL SIGUIENTE NODO (TODOS LOS ENLACES TENÍAN UNA MTU)

EJERCICIO 27

CSMA/CD (CARRIER SENSE MULTIPLE ACCESS / COLLISION DETECTION)

SRAN: TIEMPO DE ESCUCHA MÍNIMO ANTES DE HABLAR AL RÉGIO

CRAN: TIEMPO DESDE RECIBIR UNA TRAMA DE DATOS A ENVIAR UNA TRAMA DE CONTROL CON EL ACK.

$$t_{\text{trama}} > 2t_{\text{prop}}$$



CUANDO VUELVE LA TRAMA AL EQUIPO QUE LA ENVIA ES CUANDO ESTE DETECTA CONGESTIÓN Y PUEDE PARAR DE RETRANSMITIR LA TRAMA, PARA ELLA $t_{\text{trama}} > t_{\text{RTT}} = t_{\text{trama}} > 2t_{\text{prop}}$

$$\begin{aligned} d &= 500 \text{ m} \\ v &= 0.2 \mu\text{s}/\text{m} \Rightarrow t_{\text{prop}} = d/v = 500 \text{ m} \times 0.2 \mu\text{s}/\text{m} = 10 \mu\text{s} = 0.01 \text{ ms} \end{aligned}$$

$$\begin{aligned} R &= 5 \text{ Mbps} \Rightarrow \text{TENEMOS 2 DESIGUALDADES} \quad t_{\text{trans}} < 1 \text{ ms} \\ L? & \qquad \qquad \qquad t_{\text{trama}} > 2t_{\text{prop}} \Rightarrow t_{\text{trama}} > 0.02 \text{ ms} \end{aligned}$$

Como $t_{\text{trans}} = t_{\text{trama}}$, TENEMOS UNA DESIGUALDAD TAL $t_{\text{min}} < t_{\text{trama}} < t_{\text{max}}$

$$\text{EQUIVALENTE A: } t_{\text{min}} \times R < L_{\text{trama}} < t_{\text{max}} \times R \Rightarrow L_{\text{min}} < L_{\text{trama}} < L_{\text{max}}$$

$$0.02 < t_{\text{trama}} < 1 \text{ (En ms)}$$

$$\begin{aligned} 5 \text{ Mbps} &= 5 \text{ Kb/ms} \\ &= 5000 \text{ bits/ms} \end{aligned}$$

$$0.02 \times 5000 < L_{\text{trama}} < 1 \times 5000 \text{ (En bits)}$$

$$\begin{aligned} \text{100 bits} &< L_{\text{trama}} < \text{5000 bits} \\ \text{TAN PEQUEÑO} & \qquad \qquad \qquad \text{TAN GRANDE} \end{aligned}$$

EJERCICIO 28

$$* 1 \text{ Gbps} = 1 \text{ b/ms}$$

$$\begin{aligned} d &= D \text{ m} \\ v &= 5 \text{ ms/m} \Rightarrow t_{\text{prop}} = 5D \text{ ms} \end{aligned}$$

$$64 \text{ bytes} < L_{\text{trama}} < 1518 \text{ bytes} \text{ (En bytes)}$$

Ahora como $t_{\text{trama}} > 2t_{\text{prop}}$.

$$1024 \text{ bits} < L_{\text{trama}} < 12144 \text{ bits} \text{ (En bits)}$$

$$2t_{\text{prop}} < 512 \text{ ms} < t_{\text{trama}}$$

$$\Rightarrow 0 < 2t_{\text{prop}} < 512 \text{ ms}$$

NO EXISTE NINGUNA YA QUE LA DESIGUALDAD SE CUMPLE PARA t_{prop} INFINITO

Cuando $2t_{\text{prop}} < t_{\text{trama}}$ NUNCA SE PODRA SUPERAR EL NÚMERO IMPRESO

$$\frac{1024 \text{ bits}}{1 \text{ bits/ms}} < t_{\text{trama}} < \frac{12144 \text{ bits}}{1 \text{ bits/ms}} \text{ (En s)}$$

$$2t_{\text{prop}} < 512 \text{ ms}$$

$$\frac{1024 \text{ bits}}{1 \text{ bits/ms}} < t_{\text{trama}} < \frac{12144 \text{ bits}}{1 \text{ bits/ms}} \text{ (En ms)} *$$

$$\frac{1024 \text{ bits}}{1 \text{ bits/ms}} < t_{\text{trama}} < \frac{12144 \text{ bits}}{1 \text{ bits/ms}} \text{ (En ms)} > 2t_{\text{prop}}$$

$$512 \text{ ms} < t_{\text{trama}} < 12144 \text{ ms} \text{ (En ms)}$$

$$0 < D < 51,2 \text{ m}$$

EJERCICIO 29

UN SWITCH/COMBINADOR ETHERNET INICIALMENTE NO CONOCE QUE INTERFAZ SE ASOCIA A CADA NODO Y CADA VEZ QUE LE LLEGA UNA TRAMA ASOCIA LA MAC ORIGEN CON LA INTERFAZ DE ENTRADA PERMITIÉNDOLE APlicar EL MECANISMO DE AUTO APRENDIZAJE.

a) $B \rightarrow E$ TABLA =

DIRECCIÓN MAC	INTERFAZ
---------------	----------

El COMBINADOR ~~NO~~ SABE LA MAC DEL Nodo E, ENTONCES REEMITIRÁ POR TODAS LAS INTERFAZES Y SE APUNTA LA INTERFAZ POR LA QUE ENTRÓ LA TRAMA CON MAC ORIGEN CORRESPONDIENTE AL NODO B.

b) $E \rightarrow B$ TABLA =

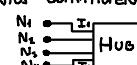
DIRECCIÓN MAC	INTERFAZ
MAC NODO B	INTERFAZ B

EL COMBINADOR AHORA YA DISPONE LA INTERFAZ DE SALIDA PARA LA MAC DESTINO CORRESPONDIENTE AL NODO B. ADENAS AHORA GUARDA INFORMACIÓN SOBRE E.

c) $A \rightarrow B$ TABLA =

DIRECCIÓN MAC	(*) INTERFAZ
MAC NODO B	INTERFAZ B
MAC NODO E	INTERFAZ E

(*) CON INTERFAZ «Nombre_Nodo»
SOLO ES UNA FORMA DE DECIR:
«LA INTERFAZ POR LA QUE ENTRAN
LOS PAQUETES DE B»
EN TIPOLOGÍA ESTRELLA NUNCA DOS NODOS
DISTINTOS COMPARTEN INTERFAZ.



NUEVAMENTE EL COMBINADOR CONOCE LA INTERFAZ DE SALIDA PARA LLEGAR AL NODO B, ADENAS GUARDA LA INTERFAZ DE ENTRADA ASOCIADA A LA MAC NODO A.

d) $B \rightarrow A$ TABLA =

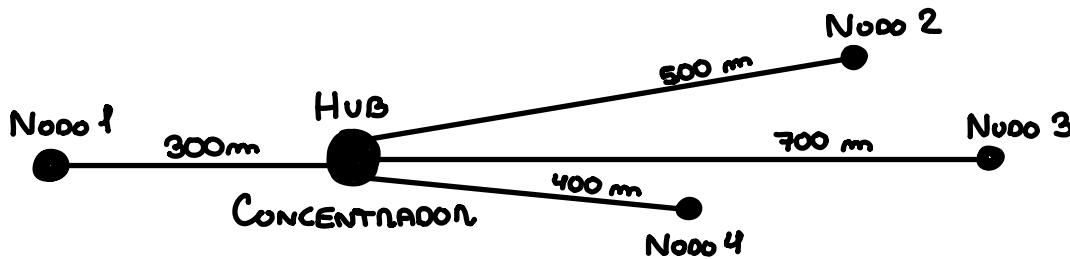
DIRECCIÓN MAC	INTERFAZ
MAC NODO B	INTERFAZ B
MAC NODO E	INTERFAZ E
MAC NODO A	INTERFAZ A

EL COMBINADOR YA CONOCÍ LA INTERFAZ PARA REEMITIR TRAMAS HACIA A CON LA TRAMA ANTERIOR UTILIZANDO ESA. ADENAS REFRESA LA INTERFAZ ASOCIADA A B O LA ACTUALIZA SI SU INTERFAZ HA SU NUEVO VALOR

TABLA FINAL:

DIRECCIÓN MAC	INTERFAZ
MAC NODO B	INTERFAZ B ₂
MAC NODO E	INTERFAZ E
MAC NODO A	INTERFAZ A

EJERCICIO 30



PARA GARANTIZAR QUE ALGUNO DE LOS NODOS PUEDE DETECTAR LA CONGESTIÓN MIENTRAS TRANSMITE HAY QUE GARANTIZAR QUE EN EL PEOR CASO LA CONGESTIÓN PUEDA DETECTARSE.

UN NODO SÓLO DETECTA LA CONGESTIÓN MIENTRAS TRANSMITE^(*), Y POR TANTO ESTA TRANSMISIÓN TIENE QUE DURAR EL TIEMPO MÁXIMO ENTRE DOS NODOS, COMO LA RED NO ES SIMÉTRICA DEBENOS COLOCAR LA DISTANCIA MÁXIMA.

Nodo 2 → Hub	Hub → Nodo 3
MIENTRAS NODO 3 TRANSMITE (COMPARA)	

t_{prop}

t_{trama}

$$t_{N2-Hub} = \frac{500 \text{ m}}{2 \times 10^8 \text{ m/s}} = 2,5 \times 10^{-6} \text{ s} = 2,5 \text{ ns}$$

$$t_{Hub-N3} = \frac{700 \text{ m}}{2 \times 10^8 \text{ m/s}} = 3,5 \times 10^{-6} \text{ s} = 3,5 \text{ ns}$$

"TIEMPO MÁXIMO QUE TANDEM UNA SEÑAL EN VIAJAR ENTRE LOS NODOS MÁS DISTANTES"

EN ESTE CASO DISTINTO A:

"TIEMPO MÁXIMO QUE TANDEM UNA SEÑAL EN VIAJAR ENTRE DOS NODOS CUALESQUIERA"

$$t_{prop} = t_{N2-Hub} + t_{Hub-N3} = 2,5 \text{ ns} + 3,5 \text{ ns} = 6 \text{ ns}$$

$$\Rightarrow L_{trama} > 2t_{prop} \times R = 12 \times 10^{-6} \text{ s} \times 10^7 \text{ b/s} = 120 \text{ bits}$$

TANROCOS EXISTE TANRÍO MÁXIMO EN ESTE EJEMPLO.

¿CÓMO SE DETECTA LA COLISIÓN EN CSMA/CD?

(*) UN NODO COMPARA LA SEÑAL QUE HAY EN EL MEDIO (ESCUCHA) CON LA SEÑAL QUE TRANSMITE (SÓLO MIENTRAS HABLA) Y SI SON IGUALES, (ALGUNA PRESENCIA DE OTRA SEÑAL PERTURBARÍA LA SEÑAL TRANSMITIDA AL ENTRAR AL MEDIO (NO ESCUCHA LO QUE EL ESTÁ HABLANDO)).

EJERCICIO 31

SI LA VELOCIDAD DE ENLACE (β) AUMENTA EL TIEMPO DE TRANSMITIR (L/β) LA TRAMA DISMINUYE PROBABLEMENTE QUE PARA UN TIEMPO DE PROPAGACIÓN CONSTANTE EL TAMAÑO MÍNIMO DE LA TRAMA AUMENTE PARA GARANTIZAR QUE $t_{trama} > 2t_{prop}$.

PARA QUE SE CUMPLA LA DESIGUALDAD ANTERIOR SOLO AUMENTANDO (β) Y SIN MODIFICAR EL TAMAÑO DE LA TRAMA (L) SE PUEDE REDUCIR EL t_{prop} AUMENTANDO LA VELOCIDAD DE PROPAGACIÓN DEL MEDIO (v) UTILIZADO POR ETHERNET EN LA MITAD QUE β O REDUCIR LAS DISTANCIAS ENTRE NUDOS (d) EN LA MITAD DEL AUMENTO DE β , O LO QUE NUESTRO PROFESOR SUGIERE AÑADIR CONUTADORES.

