

Comunicaciones y redes de computadores

Problemas y ejercicios resueltos

Eduardo Magaña Lizarrondo
Edurne Izkue Mendi
Manuel Prieto Miguez
Jesús Villadangos Alonso

R.782.6
(P)

681.32
C.M.
Semana

COMUNICACIONES Y REDES DE COMPUTADORES

Problemas y ejercicios resueltos

Eduardo Magaña Lizarrondo

Edurne Izkue Mendi

Manuel Prieto Miguez

Jesús Villadangos Alonso

Departamento de Automática y Computación
Universidad Pública de Navarra



Madrid • México • Santafé de Bogotá • Buenos Aires • Caracas • Lima • Montevideo •
San Juan • San José • Santiago • São Paulo • White Plains

MAGAÑA LIZARRONDO, E.; IZKUE MENDI, E.; PRIETO MIGUEZ, M.; VILLADANGOS ALONSO, J.

COMUNICACIONES Y REDES DE COMPUTADORES
Problemas y ejercicios resueltos

PEARSON EDUCACIÓN, S.A., Madrid, 2003

ISBN: 84-205-3920-1

Materia: Computadores 681.3

Formato: 190 × 270 mm

Páginas: 216

Todos los derechos reservados. Queda prohibida, salvo excepción prevista en la ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra sin contar con autorización de los titulares de la propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. Código Penal*).

DERECHOS RESERVADOS

© 2003 por PEARSON EDUCACIÓN, S.A.

Ribera del Loira, 28

28042 Madrid (España)

COMUNICACIONES Y REDES DE COMPUTADORES. Problemas y ejercicios resueltos

MAGAÑA LIZARRONDO, E.; IZKUE MENDI, E.; PRIETO MIGUEZ, M.; VILLADANGOS ALONSO, J.

ISBN: 84-205-3920-1

Depósito legal: M. 35.361-2003

PEARSON PRENTICE HALL es un sello editorial autorizado de PEARSON EDUCACIÓN, S.A.

Equipo editorial:

Editor: David Fayerman Aragón

Técnico editorial: Ana Isabel García Borro

Equipo de producción:

Director: José Antonio Clares

Técnico: José Antonio Hernán

Diseño de cubierta: Equipo de diseño de PEARSON EDUCACIÓN, S.A.

Composición: JOSUR, TRATAMIENTO DE TEXTOS, S.L.

Impreso por: Imprenta Fareso, S. A.

IMPRESO EN ESPAÑA - PRINTED IN SPAIN



CONTENIDO

Prólogo	VII
Capítulo 1. Transmisión de datos	1
Capítulo 2. Medios de transmisión	9
Capítulo 3. Codificación de datos	15
Capítulo 4. La interfaz en las comunicaciones de datos	29
Capítulo 5. Control del enlace de datos	33
Capítulo 6. Multiplexación.....	41
Capítulo 7. Conmutación de circuitos	55
Capítulo 8. Conmutación de paquetes	75
Capítulo 9. Transferencia en modo asíncrono y retransmisión de tramas	97
Capítulo 10. Congestión en redes de datos	101
Capítulo 11. Tecnologías LAN	109
Capítulo 12. Sistemas LAN	121
Capítulo 13. Protocolos de interconexión de redes	139
Capítulo 14. Funcionamiento de la interconexión de redes	151
Capítulo 15. Protocolos de transporte	171
Capítulo 16. Seguridad en redes.....	185
Capítulo 17. Aplicaciones distribuidas	195
Índice de figuras y tablas	203

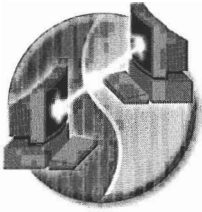
PRÓLOGO

Muchas veces, el estudio de una materia se hace particularmente difícil por la ausencia de ejercicios resueltos que permitan comprender los conceptos teóricos. La comprensión de éstos permite enfrentarse con mayor confianza a la resolución de problemas complejos.

La presente obra nace con la intención de ayudar a los estudiantes en el aprendizaje de los conceptos básicos de las redes de comunicaciones. En este sentido, se ha considerado que la mejor manera de apoyar al estudiante en el aprendizaje de las redes de comunicaciones, ha sido la de realizar una obra donde se plantean ejercicios relacionados con los conceptos teóricos más elementales de las redes de comunicaciones. Estos ejercicios se acompañan de la solución de los mismos, para que el estudiante pueda trabajar de forma personal su resolución.

El libro se estructura en diferentes capítulos siguiendo el esquema del título *Comunicaciones y redes de computadores* de William Stallings, por lo que esta obra puede servir de libro de apoyo para la comprensión de los conceptos introducidos en dicha obra. Esta orientación no limita el uso de esta obra, ya que los ejercicios propuestos se refieren a conceptos elementales sobre las redes de comunicaciones.

Así, en este libro se proporcionan ejercicios correspondientes a conceptos de los diferentes niveles OSI, desde la transmisión y codificación de datos, pasando por ejercicios sobre conmutación de paquetes y circuitos, hasta problemas sobre aplicaciones distribuidas. En concreto, el Capítulo 1 se dedica a la transmisión de datos, el Capítulo 2, a los medios de transmisión y el Capítulo 3, a la codificación de datos. Los Capítulos 4, 5 y 6 tratan la interfaz de comunicación de datos, el control de enlace de datos y la multiplexación, respectivamente. La conmutación de circuitos y paquetes, así como la transmisión en modo asíncrono, se ven en los Capítulos 7, 8 y 9. A continuación, en el Capítulo 10, se presentan ejercicios sobre la congestión en las redes de datos. Los Capítulos 11 y 12 se dedican a las tecnologías y sistemas LAN para, después, tratar la interconexión de sistemas y los protocolos de interconexión en los Capítulos 13 y 14. El Capítulo 15 presenta ejercicios sobre los protocolos de transporte. Finalmente, los Capítulos 16 y 17 tratan la seguridad en las redes de comunicaciones y las aplicaciones distribuidas.



TRANSMISIÓN DE DATOS



Indicar las diferencias entre el Ancho de Banda absoluto y el Ancho de Banda efectivo. Para ello usar el siguiente ejemplo de señal:

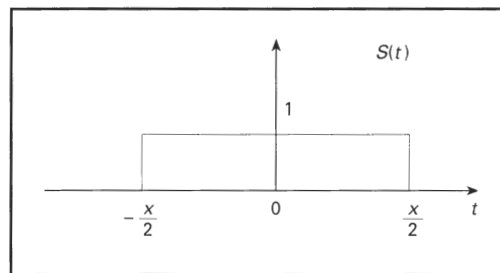


Figura 1.1. Señal en el dominio del tiempo.

SOLUCIÓN

Una vez identificada la señal del ejemplo, la función de $S(t)$ en el dominio frecuencial (mediante la Transformada de Fourier) es como la siguiente figura.

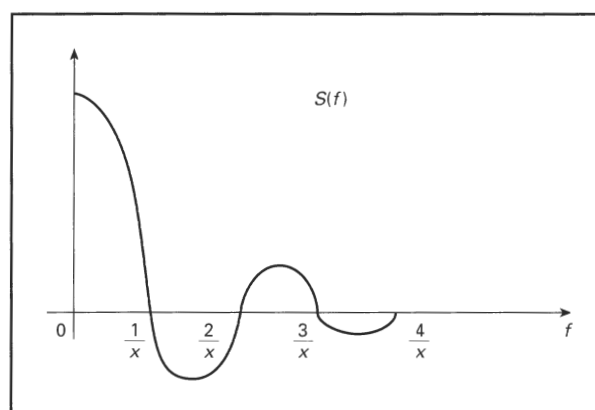


Figura 1.2. Señal en el dominio frecuencial.

Como norma general, las señales discretas en el dominio temporal son señales continuas en el dominio frecuencial y viceversa; las señales continuas en el dominio temporal son señales discretas en el dominio frecuencial.

Claro ejemplo es la señal senoidal en el dominio temporal (señal continua), cuya Transformada de Fourier es un pulso en la frecuencia de la senoidal, en el dominio frecuencial (señal discreta).

El ancho de banda absoluto de la señal del ejemplo $S(f)$ es infinito, porque es el ancho de banda en el que $S(f)$ tiene valores diferentes de cero, no nulos.

El ancho de banda efectivo es el ancho de banda que se considera suficiente para la representación, es decir, despreciando aquellos rangos de frecuencia en los que $S(f)$ es más o menos nulo, un valor despreciable.

1.2.

La representación de las señales binarias para la transmisión digital con un menor ancho de banda se hace ajustando a una señal senoidal, aproximando una señal cuadrada a una senoidal pura (de una sola frecuencia de $1/T$), suficiente para que el receptor sea capaz de diferenciar entre un cero y un uno binario. Se supone un sistema de transmisión digital capaz de transmitir señales con un ancho de banda de 4 MHz. Se pretende transmitir una secuencia de unos y ceros alternativamente con una señal cuadrada, ¿qué velocidad de transmisión se puede conseguir?

SOLUCIÓN

La transformada de una señal senoidal en el tiempo es discreta en frecuencia, con un único pico en la frecuencia f , donde f es la frecuencia de la señal senoidal $1/T$ (siendo T el período para transmitir un uno y un cero).

La señal en el dominio frecuencial no tiene ancho de banda, puesto que es una única frecuencia y no un rango. Por lo que la velocidad de transmisión dependerá únicamente del período T de tiempo de la señal temporal en la que se transmite un uno y un cero. Por ser 2 bits por período, la velocidad de transmisión resultaría:

$$v = \frac{2 \text{ (bits)}}{T(\text{sg})} \text{ (bps)}$$

1.3.

Se pretende transmitir a una velocidad de 4 Mbps. La señal a transmitir será una secuencia de unos y ceros alternantes, como una onda cuadrada, en el dominio temporal. ¿Qué ancho de banda se necesitaría en los siguientes casos?

- La señal cuadrada se puede aproximar de modo eficiente para que el receptor pueda discernir el uno (amplitud igual a 1) y el cero (amplitud igual a 0), con la segunda frecuencia fundamental de la Transformada de Fourier.
- La señal cuadrada se aproxima con la tercera frecuencia fundamental.

SOLUCIÓN

- La señal cuadrada en el dominio temporal se representaría, con la segunda frecuencia a un dominio frecuencial, de la siguiente manera.

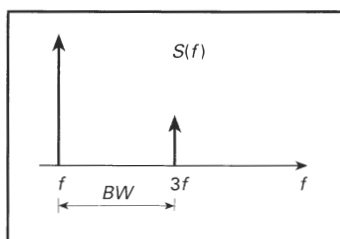


Figura 1.3. Señal hasta el segundo fundamental en el dominio frecuencial.

El cálculo de esta señal en el dominio frecuencia se realiza mediante la Transforma de Fourier de la señal cuadrada, dando como resultado la siguiente señal.

$$S(f) = \frac{4}{\pi} \left[\text{sen}(2\pi ft) + \frac{1}{3} \text{sen}(2\pi(3f)t) \right]$$

La frecuencia fundamental será la inversa del período,

$$f = \frac{1}{t}$$

con 2 bits por cada período. La velocidad de transmisión resultaría:

$$v = \frac{2 \text{ (bits)}}{T \text{ (sg)}} = 2f = 4 \text{ Mbps}$$

Esto da como resultado una frecuencia:

$$f = 2 \text{ MHz } [T = 0,5 \mu\text{sg}]$$

Si f son 2 MHz, el ancho de banda mínimo del sistema de transmisión para una señal cuadrada aproximada al segundo fundamental, necesita:

$$BW = 3f - f = 2f = 4 \text{ MHz}$$

- b) Si la misma señal se necesita aproximar hasta la tercera frecuencia fundamental, el ancho de banda necesario es mayor.

La señal cuadrada aproximada al tercer fundamental tiene una señal en el dominio frecuencial del siguiente tipo:

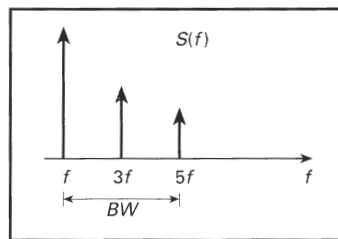


Figura 1.4. Señal en el dominio frecuencial.

La señal resultante de la Transformada de Fourier aproximada hasta el tercer fundamental es como la siguiente:

$$S(f) = \frac{4}{\pi} \left[\text{sen}(2\pi ft) + \frac{1}{3} \text{sen}(2\pi(3f)t) + \frac{1}{5} \text{sen}(2\pi(5f)t) \right]$$

Por lo que el ancho de banda mínimo es:

$$BW = 5f - f = 4f = 8 \text{ MHz}$$

Del caso a) se ha resuelto que la frecuencia fundamental es $f = 2 \text{ MHz}$ y por lo tanto para el caso b) se necesita un ancho de banda de $BW = 8 \text{ MHz}$, cuando para el caso a) se necesitaba $BW = 4 \text{ MHz}$, la mitad.

1.4.

Dada una temperatura efectiva de ruido de 8.000 °K, en un sistema con un ancho de banda de 4 MHz, ¿qué nivel de ruido hay a la salida?

SOLUCIÓN

Consideremos que el ruido térmico es $N_0 = k \cdot T$ por cada ancho de banda $BW = 1$ Hz.

En este caso se tiene un ancho de banda $BW = 4$ MHz, por lo que el ruido térmico resultaría, $N_0 = k \cdot T \cdot BW$, donde las partes integrantes son las siguientes:

- k es la Constante de Boltzmann, cuyo valor es $1,3803 \cdot 10^{-23}$ J/°K.
- T es la temperatura efectiva, que en este caso el valor es 8.000 °K.
- BW es el ancho de banda en hercios, y en este caso es 4 MHz.

De estos valores se calcula el ruido térmico resultante:

$$N_0 = 1,3803 \cdot 10^{-23} \text{ J/°K} \cdot 8.000 \text{ °K} \cdot 4 \text{ MHz} = 4,417 \cdot 10^{-16} \text{ W}$$

Y expresado el mismo ruido en decibelios, resulta:

$$N_{dB} = 10 \log N = -153,55 \text{ dBW}$$

1.5.

En un sistema de transmisión con un ancho de banda de 8 MHz, y una capacidad de 48 Mbps, ¿hasta cuántos niveles de tensión se pueden transmitir? O en otras palabras, ¿cuántas señales discretas se pueden transmitir?

SOLUCIÓN

Según el ancho de banda de Nyquist, se tiene la siguiente ecuación:

$$C = 2 B \log_2 M$$

Con una capacidad de $C = 48$ Mbps, y un ancho de banda $B = 8$ MHz, se calcula la cantidad de señales discretas que se pueden transmitir, M :

$$C = 48 \text{ Mbps} = 2 \cdot 8 \text{ MHz} \cdot \log_2 M \rightarrow \log_2 M = 3 \rightarrow M = 8$$

Por lo que son 8 las señales discretas que se pueden transmitir.

1.6.

Se han implantado dos sistemas de comunicaciones de las mismas características en Egipto y en Moscú. Si se comprueba que en ambos sistemas la SNR en el equipo receptor es idéntica, ¿es posible determinar cual de los equipos transmisores emite una mayor potencia?

En caso de que la respuesta anterior sea afirmativa, indique cuál es la relación entre las potencias emitidas por cada uno de los transmisores.

(Suponga que la temperatura en Egipto es de unos 35 °C y en Moscú de -10 °C.)

SOLUCIÓN

A pesar de que los sistemas instalados en ambos lugares tienen las mismas características, el ruido térmico que afecta a ambos sistemas es distinto debido a que la temperatura es distinta. Por ello, para mantener la relación SNR constante, el lugar con mayor ruido térmico (aquel con una mayor temperatura), Egipto en este caso, será el que requerirá que el equipo transmisor emita con una mayor potencia.

Para calcular la relación entre las potencias emitidas en ambos lugares por los equipos transmisores, empezamos planteando la condición de que la SNR es igual en ambos sistemas.

$$\frac{S_{\text{Egipto}}}{N_{\text{Egipto}}} = \frac{S_{\text{Moscu}}}{N_{\text{Moscu}}}$$

Por lo que,

$$\frac{S_{\text{Egipto}}}{S_{\text{Moscu}}} = \frac{N_{\text{Egipto}}}{N_{\text{Moscu}}} = \frac{KT_{\text{Egipto}}B}{KT_{\text{Moscu}}B} = \frac{T_{\text{Egipto}}}{T_{\text{Moscu}}} = \frac{273 + 35}{273 - 10} = \frac{308}{263} = 1,17$$

Es decir, la potencia transmitida en Egipto debe de ser un 17% mayor que en Moscu.

$$S_{\text{Egipto}} = (1 + 0,17) \cdot S_{\text{Moscu}}$$

La relación anterior expresada en decibelios es la siguiente:

$$S_{\text{Egipto}} (\text{dB}) = 10 \cdot \log_{10}(1,17 \cdot S_{\text{Moscu}}) = S_{\text{Moscu}}(\text{dB}) + 10 \cdot \log_{10} 1,17$$

$$S_{\text{Egipto}} (\text{dB}) = S_{\text{Moscu}} (\text{dB}) + 0,69 (\text{dB})$$

Nota: Se ha considerado que la única fuente de ruido que afecta al sistema es el Ruido térmico.

1.7.

Se pretende implementar un sistema de comunicaciones que tenga una eficiencia espectral superior a 2,5.

¿Cuál es el número mínimo de niveles que se deben emplear?

¿Cuál es la eficiencia espectral a partir de la cual se requieren más de dos niveles?

SOLUCIÓN

Según el límite de la velocidad de transmisión que impone el Teorema de Nyquist:

$$C = 2 \cdot B \cdot \log_2 M$$

O lo que es lo mismo,

$$\frac{C}{B} = 2 \cdot \log_2 M$$

Para conseguir una eficiencia espectral $\left(\frac{C}{B}\right)$ superior a 2,5 debe satisfacerse que:

$$\frac{C}{B} = 2 \cdot \log_2 M > 2,5$$

Por lo que se deberá de cumplir:

$$M > 2^{1,25} = 2,38$$

De esto se deduce que deberán emplearse al menos tres niveles para poder conseguir la eficiencia espectral deseada.

La máxima eficiencia espectral que puede conseguirse con dos niveles es:

$$\frac{C}{B} = 2 \cdot \log_2 2 = 2 \cdot 1 = 2$$

Por lo que para conseguir eficiencias espectrales superiores a 2 será necesario emplear más de dos niveles.

1.8.

Se ha diseñado un sistema de comunicaciones que emplea un ancho de banda de 1 MHz y en que se consigue una SNR de 24 dB. Se pretende que el sistema diseñado alcance los 4 Mbps.

¿Es viable el proyecto?

SOLUCIÓN

Según Shannon, la capacidad máxima de un canal verifica la siguiente expresión:

$$C_{\text{máx}} = B \cdot \log_2 (1 + \text{SNR})$$

Sustituyendo los datos que se proporcionan en la expresión anterior:

$$\text{SNR} = 24 \text{ dB} = 10^{24/10} = 251,19$$

$$C_{\text{máx}} = B \cdot \log_2 (1 + \text{SNR}) = 10^6 \log_2 (1 + 251,19) = 7,98 \cdot 10^6 \text{ bps} = 7,98 \text{ Mbps}$$

Por lo tanto, dado que la capacidad máxima del canal es superior a los 4 Mbps que se pretendían alcanzar, se concluye que el proyecto es viable.

1.9.

Un sistema de comunicaciones trabaja a 100 Mbps y utiliza un ancho de banda de 70 MHz.

¿Cuál es la cota inferior de la relación SNR para que el sistema pueda funcionar?

SOLUCIÓN

Según el Teorema de Shannon, la máxima capacidad de transmisión de un sistema de comunicaciones con un ancho de banda B y una relación señal ruido SNR sigue la expresión:

$$C_{\text{máx}} = B \cdot \log_2 (1 + \text{SNR})$$

De la ecuación anterior se extrae que la mínima SNR necesaria para conseguir una capacidad de transmisión $C_{\text{máx}}$ es

$$\text{SNR} = -1 + 2^{\frac{C_{\text{máx}}}{B}} = -1 + 2^{\frac{100 \cdot 10^6}{70 \cdot 10^6}} = -1,6918 = 2,28 \text{ dB}$$

1.10.

¿Cuál es la eficiencia espectral máxima que se puede conseguir con una SNR de 0 dB (cuando la potencia de señal es igual a la potencia de ruido)?

SOLUCIÓN

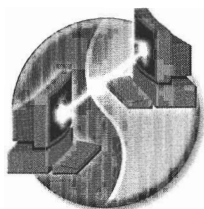
Según el Teorema de Shannon:

$$C_{\text{máx}} = B \cdot \log_2 (1 + \text{SNR})$$

Por lo tanto, la eficiencia espectral máxima será:

$$\frac{C_{\text{máx}}}{B} = \log_2 (1 + \text{SNR}) = \log_2 \left(1 + 10^{\frac{0}{10}} \right) = \log_2 (1 + 1) = 1 \text{ bit/Hz}$$

Como ha podido comprobarse, cuando la potencia de la señal es igual que la potencia de ruido, se puede transmitir 1 bit/Hz.



MEDIOS DE TRANSMISIÓN

2.1.

Complete la siguiente tabla suponiendo que la longitud de la antena del receptor debe de ser $\lambda/3$ (donde λ es la longitud de onda de la señal portadora)

Aplicación	Frecuencia	Longitud de la antena
Radio transoceánica	100 KHz	
Radio AM	1 MHz	
Radio FM	100 MHz	
Telefonía GSM	1 GHz	
Radioenlace	10 GHz	

Tabla 2.1. Longitud de la antena en función de la frecuencia de la portadora.

Represente gráficamente el tamaño que debe tener una antena en función de la frecuencia de emisión.

SOLUCIÓN

Teniendo en cuenta que $\lambda = \frac{c}{f}$, la longitud de antena necesaria en función de la frecuencia de transmisión será:

Aplicación	Frecuencia	Longitud de la antena
Radio transoceánica	100 KHz	1 km
Radio AM	1 MHz	100 m
Radio FM	100 MHz	1 m
Telefonía GSM	1 GHz	10 cm
Radioenlace	10 GHz	1 cm

Tabla 2.2. Longitud de la antena en función de la frecuencia.

Representando gráficamente la longitud de la antena L en función de la frecuencia tenemos:

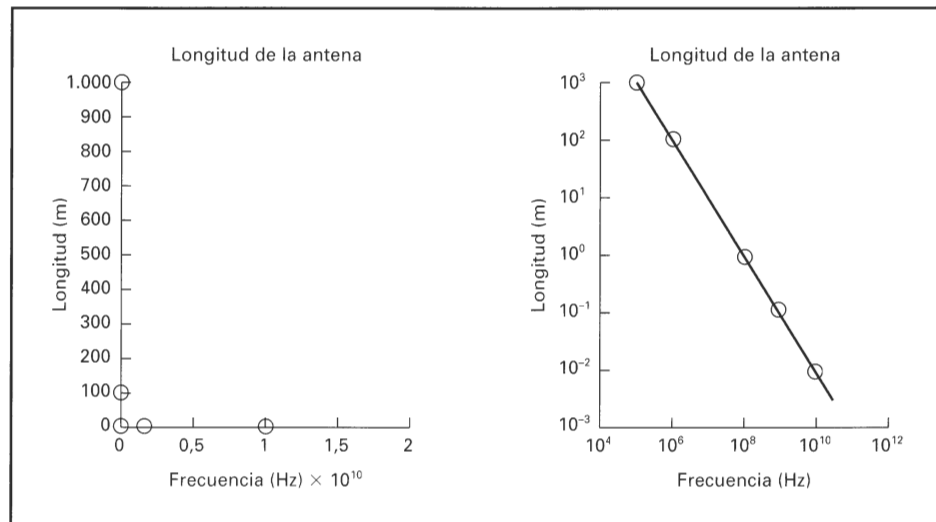


Figura 2.1. Longitud de la antena en función de la frecuencia de la portadora.

Observe que el tamaño de la antena crece tan rápidamente conforme la frecuencia de transmisión se reduce, que si no se utiliza una escala logarítmica resulta imposible distinguir la curva de los ejes.

2.2.

Habitualmente se considera que la distancia máxima entre dos antenas de microondas está limitada a $d = 7,14 \sqrt{K \cdot h}$ (km). Tenemos que h es la altura de las antenas (en metros) y K un factor de corrección que tiene en cuenta el hecho de que las microondas se refractan en la atmósfera siguiendo trayectorias cóncavas (habitualmente se considera $K = \frac{4}{3}$).

- a) Suponiendo que las microondas no se refractasen en la atmósfera, deduzca la expresión que permite calcular la distancia máxima entre dos antenas situadas ambas a una altura de h metros del suelo. (Considérese que el radio de la Tierra es de 6.371 km.)

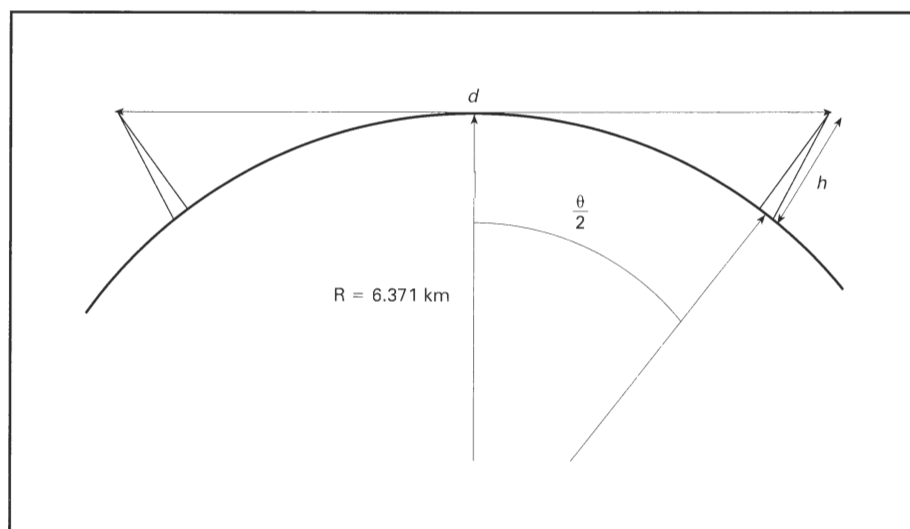


Figura 2.2. Diagrama de posicionamiento de dos antenas de altura h .

- b) Manteniendo la suposición de que las microondas no se refractan en la atmósfera, ¿cuál será la expresión que determina la distancia máxima entre dos antenas situadas a h_1 y h_2 metros de suelo?

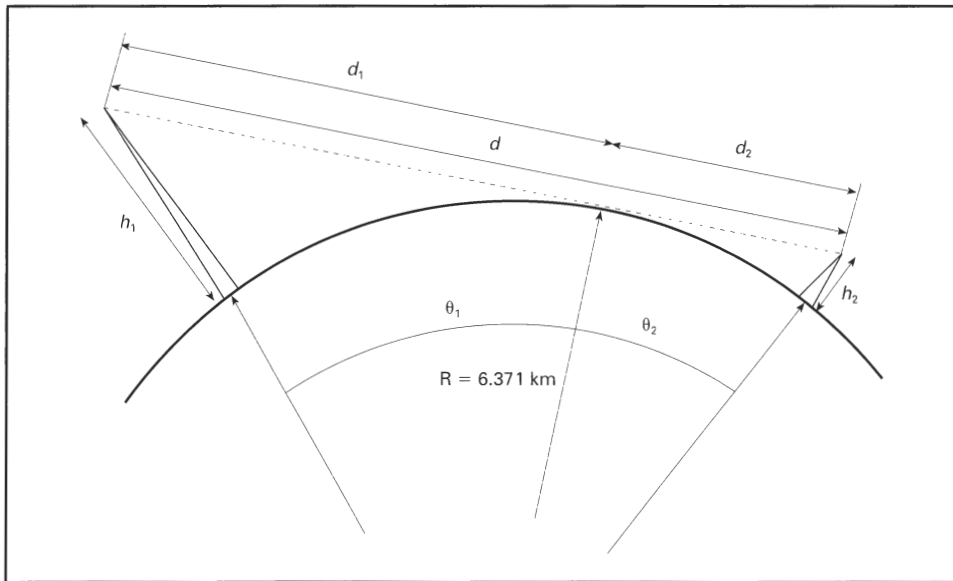


Figura 2.3. Diagrama de posicionamiento de dos antenas de altura h_1 y h_2 respectivamente.

- c) ¿En qué factor se reduce la altura a la que deben estar situadas las antenas para que un enlace de microondas pueda alcanzar una distancia dada por el hecho de que las microondas se refracten en la atmósfera?
- d) ¿Cuánto hay que aumentar la altura de las antenas para lograr doblar la distancia entre antenas?
- e) ¿A qué distancia podrán estar situadas las antenas de un enlace de microondas si se utilizan antenas de 100 m de altura?

SOLUCIÓN

- a) Aplicando algo de geometría, se deducen fácilmente las siguientes expresiones:

$$\frac{d}{2} = (R + h) \cdot \sin \frac{\theta}{2}$$

$$R = (R + h) \cdot \cos \frac{\theta}{2}$$

y despejando:

$$\begin{aligned} d &= 2 \cdot (R + h) \cdot \sin \frac{\theta}{2} = 2 \cdot (R + h) \cdot \sqrt{1 - \cos^2 \frac{\theta}{2}} = 2 \cdot \sqrt{(R + h)^2 - (R + h)^2 \cos^2 \frac{\theta}{2}} = \\ &= 2 \cdot \sqrt{(R + h)^2 - R^2} \end{aligned}$$

Teniendo en cuenta que $R \gg h$:

$$d = 2 \cdot \sqrt{(R + h)^2 - R^2} = 2 \cdot \sqrt{h^2 + 2Rh} \approx 2 \cdot \sqrt{2 \cdot R \cdot h}$$

Y si se desea que d se mida en kilómetros y h en metros:

$$d \approx 2 \cdot \sqrt{2 \cdot R \cdot \frac{h}{1.000}} = 7,14 \cdot \sqrt{h} \text{ (km)}$$

- b) Aplicando algo de geometría, de modo análogo al caso anterior, se llega a las siguientes expresiones:

$$(R + h_1) \cdot \sin \theta_1 = d_1$$

$$(R + h_2) \cdot \sin \theta_2 = d_2$$

$$(R + h_1) \cdot \cos \theta_1 = R$$

$$(R + h_2) \cdot \cos \theta_2 = R$$

$$d = d_1 + d_2$$

Suponiendo que $h_1, h_2 \ll R$ y despejando d se obtiene:

$$d \approx \sqrt{2 \cdot R} \cdot (\sqrt{h_1} + \sqrt{h_2})$$

Puesto que desea que d esté expresado en kilómetros y h_1 y h_2 en metros:

$$d \approx \sqrt{2 \cdot R} \cdot \left(\sqrt{\frac{h_1}{1.000}} + \sqrt{\frac{h_2}{1.000}} \right) = 3,57 \cdot (\sqrt{h_1} + \sqrt{h_2}) \text{ (km)}$$

Como puede comprobarse, se sigue verificando que si $h_1 = h_2$ entonces:

$$d \approx 3,57 \cdot 2 \cdot \sqrt{h} = 7,14 \cdot \sqrt{h}$$

- c) La altura h a la que deben estar situadas las antenas para alcanzar una distancia d será:

$$h = \frac{1}{K} \left(\frac{d}{7,14} \right)^2$$

En caso de que las microondas no se refractasen la altura sería:

$$h' = \left(\frac{d}{7,14} \right)^2$$

Por tanto, se deduce que el factor en el que se reduce la altura es:

$$\frac{h' - h}{h'} = 1 - \frac{1}{K} = 0,25 = 25\%$$

- d) La distancia que puede alcanzar una señal de microondas cuando la altura de las antenas es h es:

$$d = 7,14 \sqrt{h}$$

Para lograr que la señal de microondas alcance una distancia $2d$, se tiene que:

$$d' = 2d = 7,14\sqrt{h'}$$

Por lo que,

$$\frac{d'}{d} = \frac{2d}{d} = 2 = \frac{7,14\sqrt{h'}}{7,14\sqrt{h}} = \sqrt{\frac{h'}{h}}$$

$$h' = 4 \cdot h$$

Es decir, para doblar la distancia entre las antenas hay que utilizar unas antenas cuatro veces más altas.

- e) La distancia a la que podrán estar situadas las antenas de un enlace de microondas que emplee antenas de 100 m de altura será:

$$d = 7,14\sqrt{h} = 7,14 \cdot \sqrt{100} = 71,4 \text{ km}$$

2.3.

¿Cuánto se incrementan las pérdidas de un enlace de microondas al doblar la distancia?

¿Y en una fibra óptica?

SOLUCIÓN

En un enlace de microondas, la expresión que permite calcular las pérdidas para una distancia d es la siguiente:

$$L(d) = 10 \cdot \log_{10} \left(\frac{4 \cdot \pi \cdot d}{\lambda} \right)^2 \text{ (dB)}$$

Por lo tanto, para una distancia el doble de la anterior ($2d$):

$$L(2d) = 10 \cdot \log_{10} \left(\frac{4 \cdot \pi \cdot 2d}{\lambda} \right)^2 = 10 \cdot \log_{10} \left(\frac{4 \cdot \pi \cdot d}{\lambda} \right)^2 + 10 \cdot \log_{10} 2^2 \text{ (dB)}$$

$$L(2d) = L(d) + 6,02 \text{ (dB)}$$

Es decir, las pérdidas aumentan en aproximadamente 6 dB cuando se dobla la distancia de un enlace de microondas.

En el caso de que el enlace fuese un enlace de fibra óptica, la expresión que permitiría calcular las pérdidas para una distancia d sería la siguiente:

$$L(d) = \alpha \cdot d \text{ (dB)}$$

Por lo tanto, para una distancia el doble de la anterior ($2d$):

$$L(2d) = \alpha \cdot 2 \cdot d = 2 \cdot L(d) \text{ (dB)}$$

Es decir, las pérdidas se duplican cuando se dobla la distancia de un enlace de fibra óptica.

2.4.

En los enlaces de microondas por satélite se suele emplear una frecuencia f_1 en el enlace ascendente y una frecuencia f_2 en el enlace descendente. Razone por qué habitualmente $f_1 > f_2$.

SOLUCIÓN

En microondas, la atenuación debida a la propagación se rige por la expresión:

$$L = 10 \cdot \log \left(\frac{4\pi d}{\lambda} \right)^2 = 10 \cdot \log \left(\frac{4\pi d f}{c} \right)^2 \text{ dB}$$

Puesto que el logaritmo es una función monótona creciente, se cumplirá que a mayor frecuencia mayor atenuación, por lo que haciendo que $f_1 > f_2$ se consigue que un receptor situado en la Tierra, que es donde el ruido es mayor, reciba una mayor potencia que si $f_1 < f_2$. Otra ventaja es que el satélite podrá transmitir con menor potencia (por lo que el consumo será menor) si la señal emitida se va a atenuar menos.

2.5.

En un enlace de microondas, se logra cubrir una distancia de 200 km trabajando a una frecuencia f .

¿Qué distancia logrará cubrirse utilizando una frecuencia $2f$?

SOLUCIÓN

Suponiendo que el radio enlace está limitado por la relación SNR recibida, la nueva distancia que logrará cubrir el radio enlace será aquella a la que se obtenga la misma SNR. Por lo tanto, deberá cumplirse que:

$$\text{SNR}(f) = S_{\text{tx}} - L(f) - N = \text{SNR}(2f) = S_{\text{tx}} - L(2f) - N \Rightarrow L(f) = L(2f)$$

Es decir, a aquella distancia a la que las pérdidas de propagación sean idénticas. Teniendo en cuenta que en microondas la atenuación debida a la propagación se rige por la expresión:

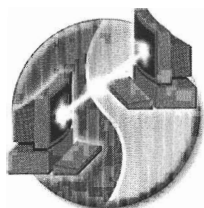
$$L = 10 \cdot \log \left(\frac{4\pi d}{\lambda} \right)^2 = 10 \cdot \log \left(\frac{4\pi d f}{c} \right)^2 \text{ dB}$$

Debe cumplirse que:

$$L(f) = 10 \cdot \log \left(\frac{4\pi d f}{c} \right)^2 = L(2f) = 10 \cdot \log \left(\frac{4\pi d' f}{c} \right)^2 \text{ dB} \Rightarrow d' = d \frac{f}{f'} = d \frac{f}{2f} = \frac{d}{2}$$

Es decir, la distancia que se consigue alcanzar al doblar el frecuencia es la mitad.

Observe sin embargo, que no podría asegurarse que utilizando una frecuencia $\frac{f}{2}$ se lograra doblar la distancia a no ser que las antenas tuviesen altura suficiente para ello.



CODIFICACIÓN DE DATOS

3.1.

Dibuje un diagrama de estados para la codificación NRZI y proponga una estructura que realice dicha codificación.

SOLUCIÓN

Teniendo en cuenta que la codificación NRZI (*No Return to Zero, Ivert on ones*) codifica los datos mediante la presencia o ausencia de transición al principio del intervalo de duración del bit (las transiciones se realizan cuando se desea transmitir un 1 binario), el diagrama de estados será el siguiente:

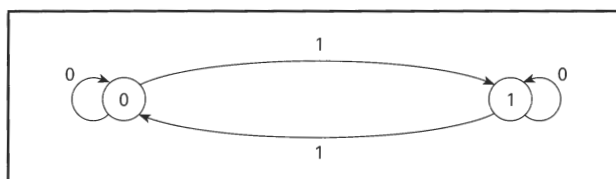


Figura 3.1. Diagrama de estados de la codificación NRZI.

O expresado en modo de tabla de la verdad:

Anterior estado	Próximo bit	Próximo estado
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 3.1. Tabla de la verdad de la codificación NRZI.

Según esto, la función lógica que realiza la codificación es la función XOR. Por lo tanto, una posible estructura que realizará la codificación NRZI será:

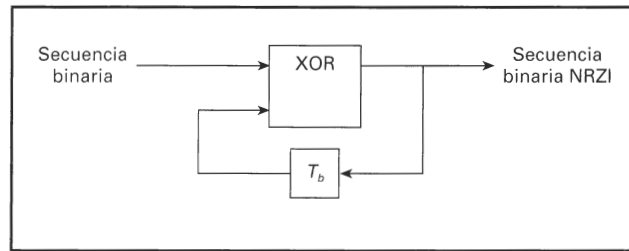


Figura 3.2. Codificador NRZI.

Donde la estructura T_b representa un retardo igual al tiempo de bit.

3.2.

Enumere las ventajas e inconvenientes de los formatos de codificación digital:

- NRZ,
- Manchester,
- HDB3.

SOLUCIÓN**Codificación NRZ**

Ventajas:

- Sencillo de implementar.
- Uso eficiente del ancho de banda.

Inconvenientes:

- La señal de reloj es difícilmente recuperable.
- Contiene nivel de continua.

Codificación Manchester

Ventajas:

- La información de reloj va incorporada en la propia señal.
- No contiene nivel de continua.
- Relativamente fácil de implementar.

Inconvenientes:

- Hace uso del doble de ancho de banda que las otras dos codificaciones.

Codificación HDB3

Ventajas:

- La información de reloj se puede recuperar de un modo sencillo.
- No contiene nivel de continua.
- Uso eficiente del ancho de banda.

Inconvenientes:

- Implementación más compleja.

3.3.

En la terminación de un enlace de fibra óptica que tiene un $\text{BER} = 10^{-7}$ y emplea una modulación digital ASK se pretende insertar, utilizando un acoplador 50 : 50, un instrumento que permite monitorizar el estado del enlace.

- ¿Cuál será el BER tras insertar el sistema de monitorización?
- Suponiendo que en lugar de un acoplador 50 : 50 se utilizase uno 90 : 10 y que se conectase la salida que recibe el 10% de la potencia al sistema de monitorización, ¿cuál será la relación entre el BER monitorizado y cuál el del enlace de comunicaciones?

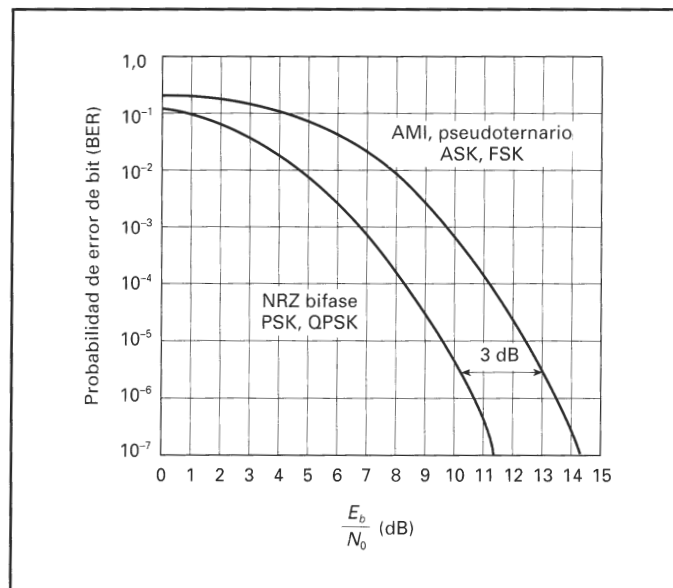


Figura 3.3. Probabilidad de error de bit (BER) en función de la relación E_b/N_0 .

SOLUCIÓN

- Según la Figura 3.3 la relación $\frac{E_b}{N_0}$ de un sistema que emplea una modulación ASK y tiene un BER de 10^{-7} vale 14,25 dB.

Por otro lado, dado que se inserta un acoplador 50 : 50 la señal que llegará al receptor se verá reducida a la mitad (pierde 3 dB).

Por lo tanto, la relación $\frac{E_b}{N_0}$ en el receptor tras insertar el acoplador 50 : 50 será:

$$\frac{E'_b}{N_0} = \frac{E_b}{N_0} - 3 = 14,25 - 3 = 11,25 \text{ dB}$$

Utilizando de nuevo la gráfica, se obtiene que el BER tras insertar el acoplador 50 : 50 será:

$$\text{BER} = 2 \cdot 10^{-4}$$

Observe que al introducir el acoplador para monitorizar el enlace, la tasa de errores se ha incrementado notablemente pasando de 10^{-7} a $2 \cdot 10^{-4}$ (ha aumentado varios órdenes de magnitud).

b) Recibir el 10% de la potencia supone unas pérdidas de inserción de:

$$10 \log_{10} 0,1 = -10 \text{ dB}$$

Por lo que en el sistema de monitorización la relación valdrá $\frac{E_b}{N_o}$:

$$\frac{E'_b}{N_o} = \frac{E_b}{N_o} - 10 = 14,25 - 10 = 4,25 \text{ dB}$$

Y por lo tanto se tendrá un BER:

$$\text{BER} = 10^{-1}$$

Recibir el 90% de la potencia supone unas pérdidas de inserción de:

$$10 \log_{10} 0,9 = -0,46 \text{ dB}$$

Por lo que en el sistema de monitorización la relación $\frac{E_b}{N_o}$ valdrá:

$$\frac{E'_b}{N_o} = \frac{E_b}{N_o} - 10 = 14,25 - 0,46 = 13,79 \text{ dB}$$

Y por lo tanto se tendrá un BER:

$$\text{BER} = 2 \cdot 10^{-6}$$

Observe que mientras que a pesar de que el sistema de comunicaciones tendrá una tasa de errores baja ($\text{BER} = 2 \cdot 10^{-6}$) el sistema de monitorización, sin embargo, no funcionará correctamente debido a una tasa de errores muy elevada ($\text{BER} = 10^{-1}$).

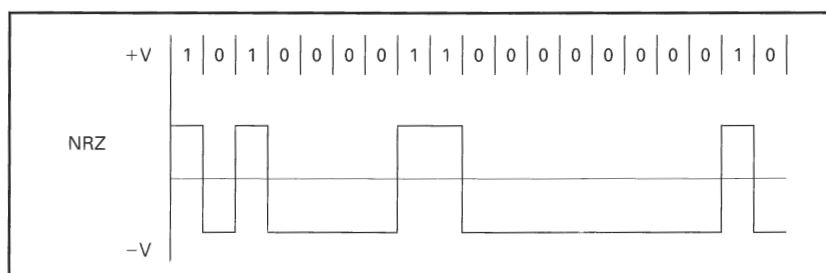
3.4.

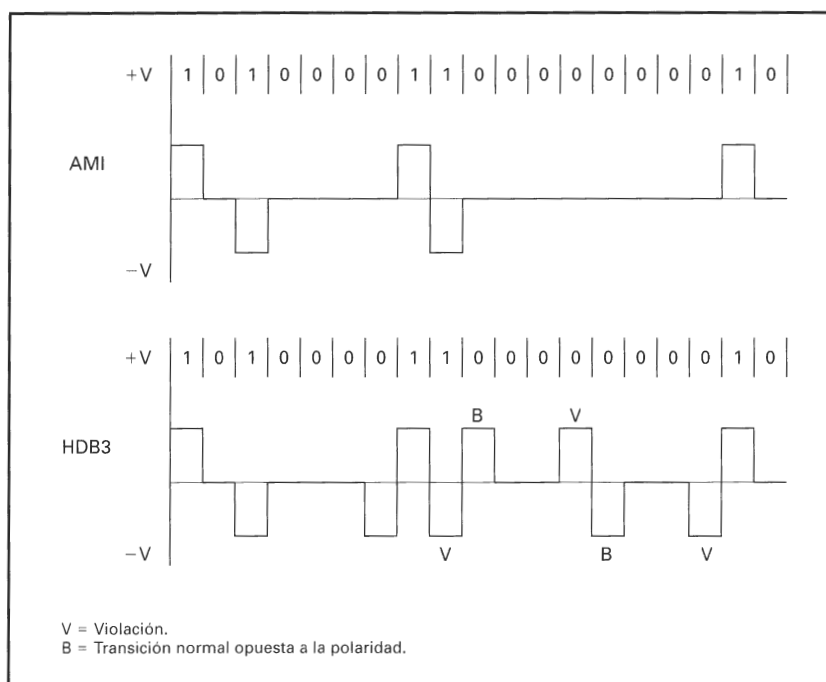
Dada la siguiente secuencia de bits:

1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Indique los niveles de señal que se obtendrían si se codificase utilizando un código NRZ, AMI y HDB3.

SOLUCIÓN





3.5.

Calcule la velocidad de transmisión de un sistema que transmite a 1.000 baudios si la modulación utilizada es:

- PSK,
- QPSK.

SOLUCIÓN

a) Teniendo en cuenta las siguientes definiciones:

$D \equiv$ velocidad de modulación (baudios),

$R \equiv$ velocidad de transmisión (bps),

$b \equiv$ número de bits por elemento de señalización.

Y teniendo en cuenta que se relacionan mediante la siguiente expresión:

$$D = \frac{R}{b}$$

Puesto que en modulación PSK cada elemento de señalización puede tomar dos valores distintos, tenemos que en cada elemento de señalización se transmite 1 bit ($\log_2 2 = 1$).

Y, por lo tanto:

$$R = D \cdot b = 1.000 \cdot 1 = 1.000 \text{ bps}$$

b) Si se utilizase una modulación QPSK en lugar de PSK, cada elemento de señalización podría tomar cuatro valores distintos, y por lo tanto cada elemento de señalización transportará 2 bits ($\log_2 4 = 2$).

Y, por lo tanto:

$$R = D \cdot b = 1.000 \cdot 2 = 2.000 \text{ bps}$$

3.6.

En el equipo receptor de un sistema de comunicaciones que emplea una modulación PSK, se está midiendo una SNR de 10dB y una probabilidad de error de 10^{-5} . ¿Cuál será la eficiencia espectral del sistema de comunicaciones?

Y si la probabilidad de error fuese 10^{-7} , ¿cuál sería la eficiencia espectral en ese caso?

¿Sería mayor o menor que cuando la probabilidad de error era 10^{-5} ?

Suponga ahora que en vez de utilizarse una modulación PSK se emplease una modulación ASK. ¿Cuál sería la eficiencia espectral para una probabilidad de error de 10^{-7} ?

SOLUCIÓN

Utilizando la Figura 3.3, el valor de la relación $\frac{E_b}{N_o}$ necesaria para obtener una probabilidad de error de 10^{-5} y 10^{-7} dada será:

P_e	$\left. \frac{E_b}{N_o} \right _{\text{PSK}} \text{ (dB)}$
10^{-5}	9,5
10^{-7}	11,3

Despejando $\frac{R}{B_T}$ de la siguiente expresión:

$$\left. \frac{E_b}{N_o} \right|_{P_e} = \text{SNR} - \frac{R}{B_T}$$

Llegamos a:

$$\frac{R}{B_T} = \text{SNR} - \left. \frac{E_b}{N_o} \right|_{P_e}$$

Por lo que la eficiencia espectral será:

$$\frac{R}{B_T} = 10^{\frac{\text{SNR} - \left. \frac{E_b}{N_o} \right|_{P_e}}{10}} = 10^{\frac{10 - 9,5}{10}} = 1,12$$

En el caso de que la probabilidad de error fuese 10^{-7} , la eficiencia espectral sería:

$$\frac{R}{B_T} = 10^{\frac{\text{SNR} - \left. \frac{E_b}{N_o} \right|_{P_e}}{10}} = 10^{\frac{10 - 11,3}{10}} = 0,74$$

La eficiencia espectral se vería por lo tanto reducida si se requiriese que la probabilidad de error fuese de 10^{-7} en lugar de 10^{-5} .

Utilizando de nuevo la Figura 3.3, el valor de la relación $\frac{E_b}{N_o}$ para la modulación ASK cuando la probabilidad de error es 10^{-7} será:

P_e	$\frac{E_b}{N_o} \Big _{\text{ASK}} \text{ (dB)}$
10^{-7}	14,3

Por lo que la eficiencia espectral será:

$$\frac{R}{B_T} = 10^{\frac{\text{SNR} - \frac{E_b}{N_o} \Big|_{P_e}}{10}} = 10^{\frac{10 - 14,3}{10}} = 0,37$$

3.7.

Dado un sistema de telefonía analógica que emplea un BW de 4 KHz y tiene una SNR de 50 dB.

¿Qué SNR percibiría un usuario si se digitalizasen los datos y se transmitiesen utilizando una codificación que lograra transmitir a la tasa máxima fijada por el Teorema de Shannon?

SOLUCIÓN

Según el Teorema de Shannon, la tasa máxima a la que se puede transmitir utilizando un canal de 4 KHz y con una SNR de 50 dB es:

$$R_{\text{máx}} = B \cdot \log_2 (1 + \text{SNR}) = 4 \cdot 10^3 \log_2 (1 + 10^{\frac{50}{10}}) = 66.438 \text{ bps}$$

Dado que al digitalizar la voz es necesario emplear a una frecuencia de muestreo mayor o igual a dos veces la máxima frecuencia de la señal (8 KHz), el número de bits por muestra del que se dispondrá será:

$$\frac{R_{\text{máx}}}{f_s} = \frac{66.438}{8.000} = 8,3 \text{ bits/muestra}$$

Pero dado que el número de bits por muestra ha de ser un número entero, cada muestra se deberá codificar utilizando 8 bits.

Por lo que según la expresión:

$$\text{SNR} = 1,76 + 6,02 \cdot N$$

La relación SNR que se conseguirá será:

$$\text{SNR} = 1,76 + 6,02 \cdot N = 1,76 + 6,02 \cdot 8 = 49,92 \text{ dB}$$

3.8.

El *dithering* es una técnica que trata de mejorar la calidad de la señal digital que se obtiene en el proceso de conversión A/D. La técnica consiste en añadir a la señal a digitalizar un pequeño ruido de amplitud igual a la mitad del intervalo de cuantificación.

¿Cómo es posible que la señal digitalizada sea de mayor calidad al aplicar esta técnica que añade ruido a la señal?

¿Qué señal tendrá una mayor relación SNR la digitalizada "normalmente" o la digitalizada empleando la técnica de *dithering*?

SOLUCIÓN

Cuando se digitaliza una señal, en el proceso de cuantificación, los infinitos valores que puede tomar una señal analógica pasan a ser representados por un conjunto finito de valores. Este efecto puede modelarse como un ruido:

$$S_{\text{continua}} = S_{\text{discreta}} + N_{\text{cuantificación}}$$

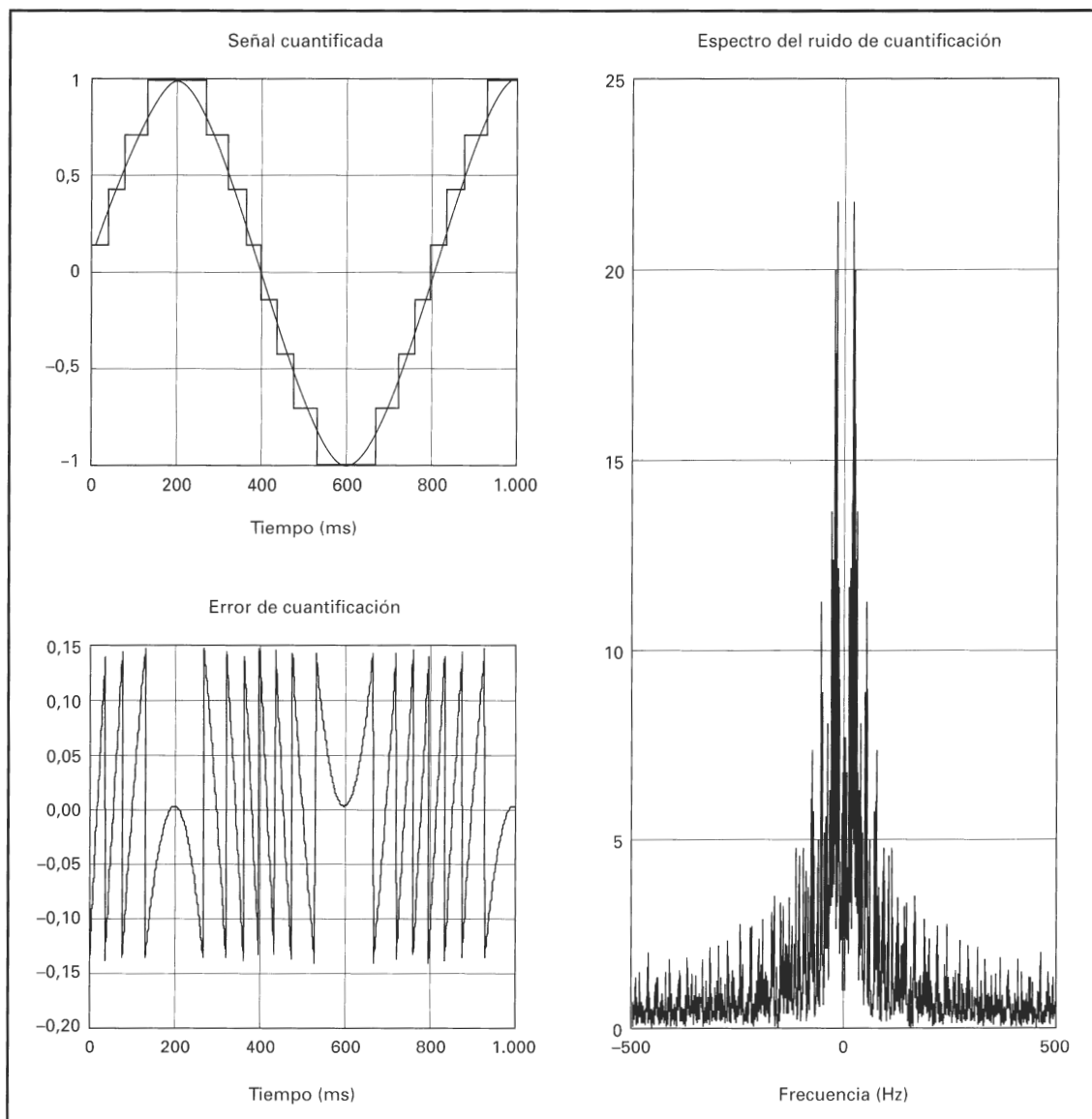


Figura 3.4. Cuantificación lineal: señal, error y espectro del ruido.

Como puede observarse en la figura, este ruido de cuantificación no es un ruido blanco. Es decir, el error que se comete al cuantificar la señal en un instante dado está relacionado con el que se comete al cuantificarla en instantes próximos. O dicho de otro modo, el ruido de cuantificación de una muestra está correlado con el de las muestras próximas.

El hecho de que no se trate de un ruido blanco tiene consecuencias negativas en muchas aplicaciones.

Para resolver esto, es posible añadir un ruido blanco de amplitud igual a la mitad del intervalo de cuantificación. Con ello se consigue que el ruido de cuantificación sea ruido blanco. Es decir, se consigue que el error que se comete al cuantificar la señal en un instante dado no este relacionado con el que se comete al cuantificarla en instantes próximos.

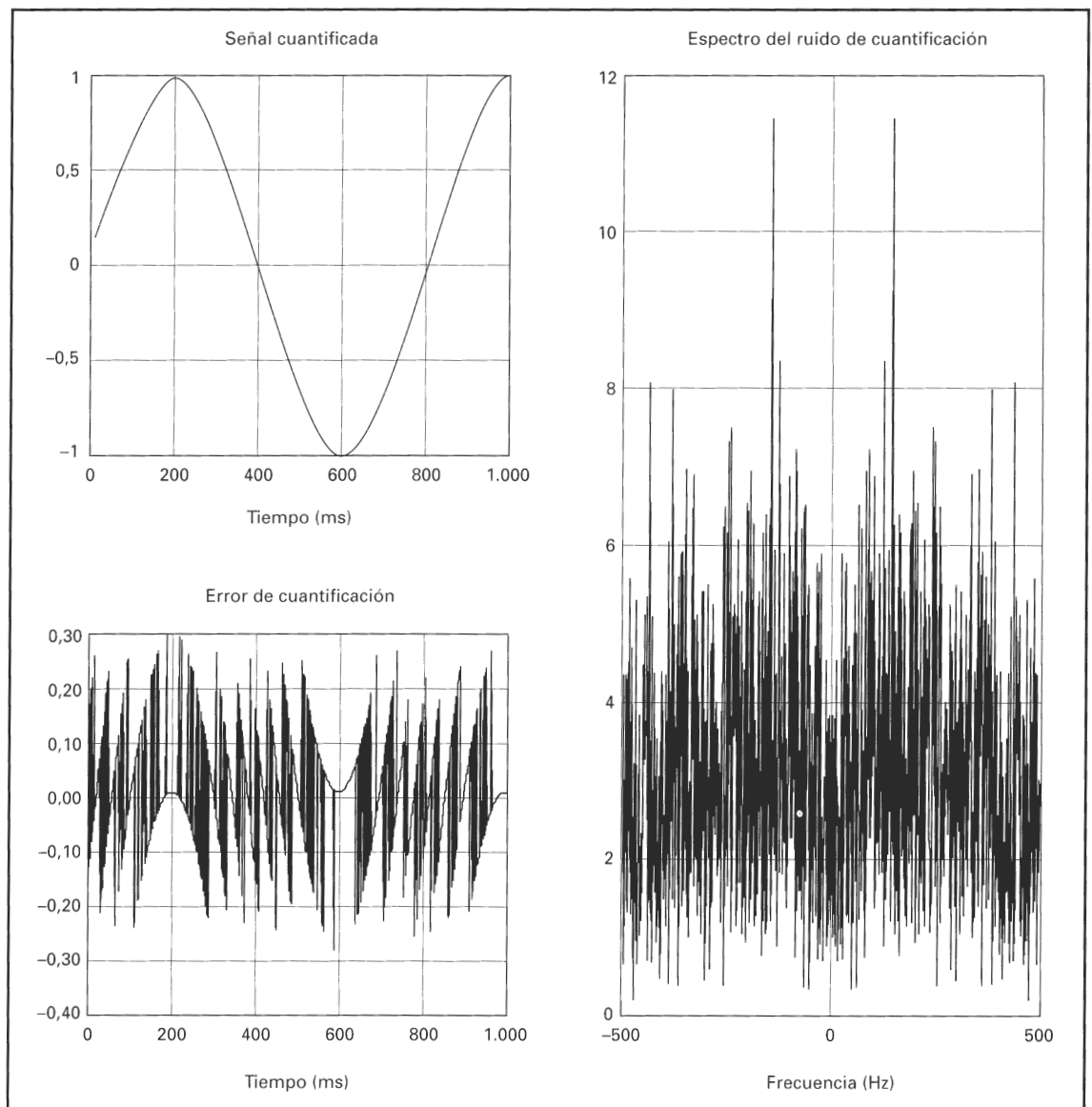


Figura 3.5. Cuantificación lineal (técnica de *dithering*): señal, error y espectro del ruido.

Es por esto por lo que la señal digitalizada empleando la técnica de *dithering* se considera que es de mayor calidad para muchas aplicaciones.

Sin embargo, si consideramos únicamente la relación SNR y no tenemos en cuenta el tipo de ruido, siempre se cumple que:

$$\text{SNR} > \text{SNR}_{\text{dithering}}$$

Esto es así porque al haber añadido un nivel de ruido a la señal, el error de cuantificación será de mayor amplitud.

Por lo tanto, evitar que el ruido de cuantificación no esté correlado no será una operación gratuita. El precio que habrá que pagar será el de aumentar el ruido de cuantificación.

3.9.

En el proceso de cuantificación de una señal, los infinitos valores que puede tomar una señal analógica pasan a ser representados por un conjunto finito de valores. Esta aproximación se modela habitualmente como un ruido. Es el denominado ruido de cuantificación.

Existen diversos métodos de cuantificación, el más simple de ellos es la cuantificación lineal que aproxima la señal con un conjunto finito de valores equiespaciados.

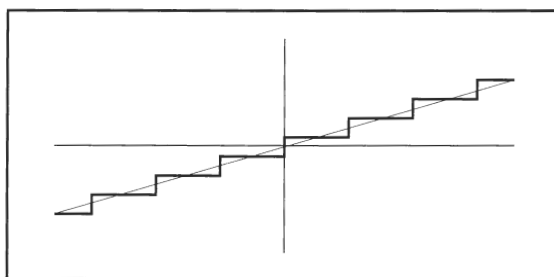


Figura 3.6. Cuantificación lineal.

Otro método bastante utilizado es la cuantificación no lineal. En él, la señal se aproxima con un conjunto de valores que están más espaciados entre sí cuanto mayor sea el nivel de señal que representen.

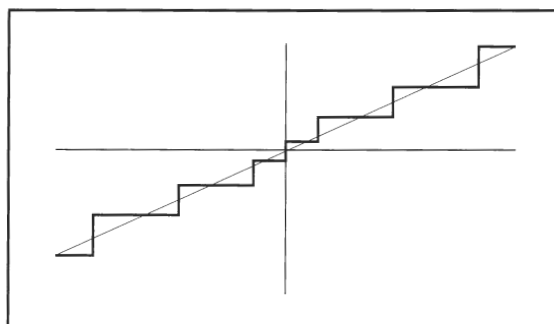


Figura 3.7. Cuantificación no lineal.

Comente qué ventajas aporta la cuantificación no lineal.

SOLUCIÓN

En la cuantificación lineal, la esperanza del error absoluto de cada muestra es independiente del nivel de señal de la muestra. Ello hace que el error relativo de las muestras de las señales de pequeña amplitud sea muy superior al de las muestras de las señales de gran amplitud. Visto de otro modo, las señales de pequeña amplitud están más distorsionadas que las de gran amplitud.

Utilizando una cuantificación no lineal es posible conseguir mantener constante el error relativo de cada muestra independientemente del nivel de señal de la muestra. Es decir, las señales se verán distorsionadas por igual independientemente de su amplitud.

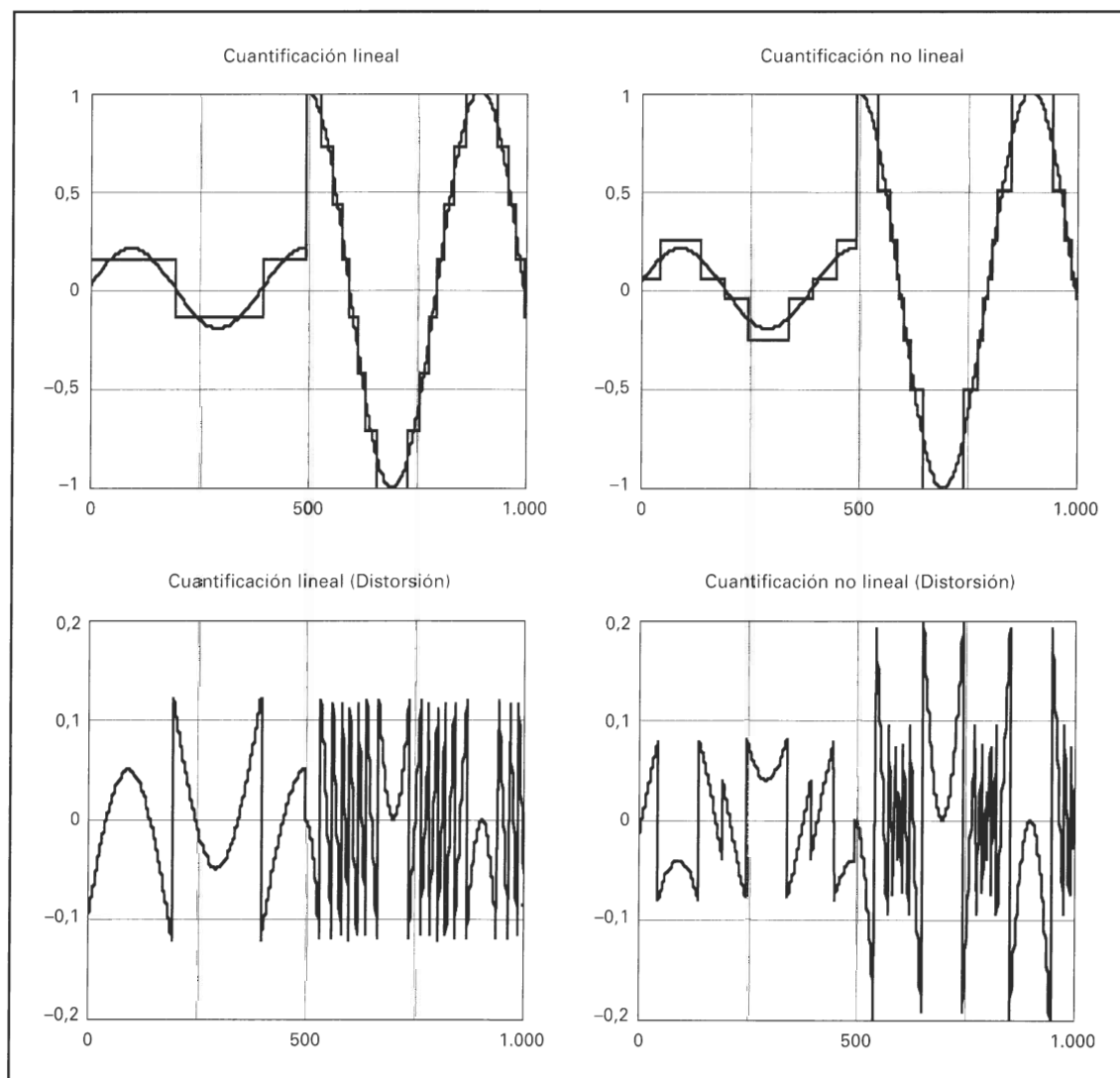


Figura 3.8. Cuantificación lineal, no lineal y su distorsión.

3.10.

Dada una señal $x(t)$ con las siguientes características:

$$E[x(t)] = 0$$

$$\text{Máx}[x(t)] = X_{\text{máx}}$$

$$E[x^2(t)] = X^2$$

- Calcule el máximo porcentaje de potencia que transportará información si dicha señal se modula en AM DSBTC (doble banda lateral con portadora) si se desea que el índice de modulación sea menor o igual que 1.
- Calcule el máximo porcentaje de potencia que transportará información en el caso en que la señal que se desee modular sea un tono $x(t) = X_{\text{máx}} \cdot \cos(w_1 t)$

SOLUCIÓN

- a) Para que el índice de modulación de la señal modulada sea menor que 1, la señal modulada deberá estar descrita por la expresión:

$$S(t) = A \cdot \left(1 + \frac{X(t)}{X_{\text{máx}}} \right) \cdot \cos(w_c t) = A \cdot \cos(w_c t) + A \cdot \frac{X(t)}{X_{\text{máx}}} \cdot \cos(w_c t)$$

Donde:

w_c es la frecuencia de la portadora.

A es la amplitud de la portadora.

Teniendo en cuenta que la señal $x(t)$ es independiente de la portadora $A \cdot \cos(w_c t)$, se cumplirá que:

$$E[\cos^2(w_c t) \cdot x(t)] = E[\cos^2(w_c t)] \cdot E[x(t)] = \frac{1}{2} \cdot 0 = 0$$

$$E[\cos(w_c t) \cdot x^2(t)] = E[\cos(w_c t)] \cdot E[x^2(t)] = 0 \cdot X^2 = 0$$

Por lo que la potencia de la señal modulada vendrá dada por:

$$P_s = P_{\text{portadora}} + P_x$$

$$P_{\text{portadora}} \propto E\left[(A \cdot \cos(w_c t))^2\right] = \frac{A^2}{2}$$

$$P_x \propto E\left[\left(A \cdot \frac{x(t)}{X_{\text{máx}}} \cdot \cos(w_c t)\right)^2\right] = \frac{A^2}{X_{\text{máx}}^2} \cdot E[x^2(t)] \cdot E[\cos^2(w_c t)] = \frac{A^2}{X_{\text{máx}}^2} \cdot X^2 \cdot \frac{1}{2}$$

Y, por lo tanto:

$$\frac{P_x}{P_s} = \frac{P_x}{P_{\text{portadora}} + P_x} = \frac{\frac{A^2 X^2}{2 X_{\text{máx}}^2}}{\frac{A^2}{2} + \frac{A^2 X^2}{2 X_{\text{máx}}^2}} = \frac{A^2 X^2}{A^2 X_{\text{máx}}^2 + A^2 X^2} = \frac{X^2}{X_{\text{máx}}^2 + X^2}$$

Teniendo en cuenta que siempre se cumplirá que:

$$X^2 \leq X_{\text{máx}}^2 \Rightarrow \frac{X^2}{X_{\text{máx}}^2} \leq 1$$

El máximo porcentaje de potencia que transportará información será menor del 50%.

- b) En el caso en que la señal $x(t)$ sea un tono, se cumplirá que:

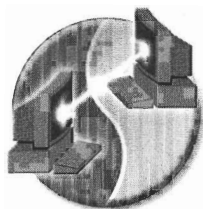
$$E[x(t)] = 0$$

$$\text{Máx } [x(t)] = X_{\text{máx}}$$

$$E[x^2(t)] = \int X_{\text{máx}}^2 \cdot \cos^2(wt) \cdot dt = \frac{X_{\text{máx}}^2}{2} = X^2$$

Y, por lo tanto, el máximo porcentaje de potencia que transportará información será:

$$\frac{P_x}{P_s} = \frac{X^2}{X_{\text{máx}}^2 + X^2} = \frac{\frac{X_{\text{máx}}^2}{2}}{X_{\text{máx}}^2 + \frac{X_{\text{máx}}^2}{2}} = \frac{1}{3} = 33\%$$



LA INTERFAZ EN LAS COMUNICACIONES DE DATOS

4.1

¿Cuál es la máxima desviación de frecuencia aceptable entre los relojes utilizados en un sistema de comunicación asíncrona?

(Suponga que cada carácter es de 8 bits y que se utiliza 1 bit de comienzo y 1 bit de parada.)

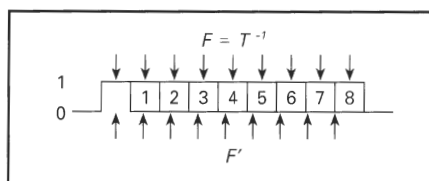


Figura 4.1. Muestreo de un carácter.

SOLUCIÓN

Consideremos que F ($T = F^{-1}$) es la frecuencia del reloj del transmisor, F' ($T' = F'^{-1}$) es la frecuencia del reloj del receptor y supongamos que el bit de comienzo se muestrease justo en la mitad del tiempo de bit ($t = \frac{T}{2}$), entonces el sistema podría tolerar una diferencia de frecuencia de reloj

tal que cuando se fuera a recibir el último bit del carácter la desincronización no hiciese que el muestreo se realizase fuera del tiempo de bit.

Teniendo en cuenta que cada carácter está compuesto de 8 bits y que se utiliza 1 bit de comienzo, el sistema deberá de ser capaz de mantener la sincronización durante nueve veces el tiempo bit.

Todo esto puede resumirse en la siguiente expresión:

$$-\frac{T}{2} < 9 \cdot (T - T') < \frac{T}{2}$$

Operando, tenemos que:

$$T \cdot \left(9 + \frac{1}{2}\right) > 9 \cdot T' \Rightarrow T' < \frac{19}{18} \cdot T \Rightarrow \frac{F'}{F} > \frac{18}{19}$$

$$T \cdot \left(9 - \frac{1}{2}\right) < 9 \cdot T' \Rightarrow T' > \frac{17}{18} \cdot T \Rightarrow \frac{F'}{F} < \frac{18}{17}$$

Y, por lo tanto:

$$\frac{18}{19} - 1 < \frac{F'}{F} - 1 < \frac{18}{17} - 1 \Rightarrow -0,05 \approx \frac{-1}{19} < \frac{F'}{F} - 1 < \frac{1}{17} \approx 0,06$$

Es decir, el máximo porcentaje que se puede desviar la frecuencia de los dos relojes es el 5%.

Nota: Debe tenerse en cuenta que la cota obtenida es un límite bastante optimista, ya se está considerando que:

- El bit de comienzo se muestrea justo en la mitad.
- Es aceptable muestrear el último bit tan cerca como se quiera del comienzo o del final del bit.

4.2.

Se desea diseñar un sistema de comunicaciones basado en transmisión síncrona, en el que cada trama está formada por 16 bits de delimitadores de trama y 32 bits de control.

Para conseguir una eficiencia superior al 80%, ¿cuál debe ser la longitud de trama?

SOLUCIÓN

La longitud de una trama puede expresarse como:

$$L = L_{\text{datos}} + L_{\text{delimitadores}} + L_{\text{control}} = L_{\text{datos}} + 16 + 32 = L_{\text{datos}} + 48$$

Y la eficiencia como:

$$\text{Eficiencia (\%)} = \frac{L_{\text{datos}}}{L} \cdot 100$$

Por lo que según se indica en el enunciado (Eficiencia > 80%) se tiene que:

$$L_{\text{datos}} > 0,8 \cdot L$$

Aplicando esto en la primera expresión se tiene que:

$$L = L_{\text{datos}} + 48 > 0,8 \cdot L + 48$$

Y, por lo tanto:

$$L > 240$$

Es decir, para longitudes de trama superiores a 240 bits se conseguirá la eficiencia exigida.

4.3.

Se desea construir un sistema de comunicaciones basado en transmisión síncrona en el que cada trama sea de 240 bits.

- ¿Es posible construir el sistema utilizando relojes a $1 \text{ MHz} \pm 200 \text{ Hz}$?
- ¿Cuál será la trama de mayor tamaño que se podrá enviar utilizando relojes con esta desviación de frecuencia?

SOLUCIÓN

- a) Suponiendo que en el primer bit el tiempo de muestreo se hiciese justo en la mitad del tiempo de bit (en $t = \frac{T}{2}$), el sistema podría tolerar una diferencia de frecuencia entre los relojes del transmisor y del receptor tal que tras los L bits ($L = 240$) de cada trama el retraso acumulado fuese inferior a $\frac{T}{2}$ (para que el instante de muestreo siga perteneciendo al bit correcto).

Considerando que T y T' son la inversa de las frecuencias de dos relojes cuya desviación en frecuencia es máxima, debe cumplirse que:

$$\frac{T}{2} > (T' - T) \cdot L$$

Por lo que,

$$\frac{T'}{T} < \frac{1}{2 \cdot L} + 1$$

De modo que dos sistemas cuyos relojes tengan la máxima desviación de frecuencia posible (según se especifica en el enunciado del problema) tendrán un período:

$$T = \frac{1}{1.000.200} = 0,9998 \cdot 10^{-6}$$

$$T' = \frac{1}{999.880} = 1,0002 \cdot 10^{-6}$$

Y, por lo tanto, dado que se verifica:

$$\frac{T'}{T} = \frac{1,0002 \cdot 10^{-6}}{0,9998 \cdot 10^{-6}} = 1,0004 < 1,002 = \frac{1}{2 \cdot 240} + 1$$

Se concluye por lo tanto, que sí será posible construir un sistema como el especificado utilizando relojes a $1 \text{ MHz} \pm 200 \text{ Hz}$.

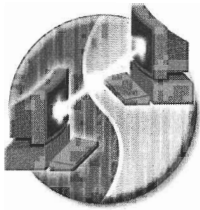
- b) Según se ha deducido, para que una trama se pueda recibir correctamente se debe cumplir que:

$$\frac{T'}{T} < \frac{1}{2 \cdot L} + 1$$

Y, por lo tanto:

$$L < \frac{T}{2 \cdot (T' - T)} = 1.249,75$$

Es decir, no podrán enviarse paquetes de más de 1.249 bits.



CONTROL DEL ENLACE DE DATOS

5.1.

- a) Con el objetivo de comparar el rendimiento del protocolo de control de flujo de parada y espera con el de ventana deslizante (de tamaño 7), calcule en los siguientes escenarios el rendimiento de cada uno de los protocolos.

(Suponga para todos los casos, que el tamaño de la trama de datos es de 4.000 bits, el tamaño de los ACK de 500 bits y la probabilidad de error es despreciable.)

a.1) En un enlace de 10.000 km $\left(t_{\text{propagación}} = \frac{10.000 \cdot 10^3 \text{ m}}{2 \cdot 10^8 \text{ m/s}} = 50 \cdot 10^{-3} \text{ s}\right)$ en el que la velocidad de transmisión es de 250 kbps.

a.2) En un enlace de 100 m $\left(t_{\text{propagación}} = \frac{100 \text{ m}}{2 \cdot 10^8 \text{ m/s}} = 50 \cdot 10^{-8} \text{ s}\right)$ en el que la velocidad de transmisión es de 10 Mbps.

- b) A la vista de los resultados obtenidos, ¿en qué condiciones es aceptable utilizar el protocolo de control de flujo de parada y espera?

SOLUCIÓN

a.1) Control de flujo de parada y espera:

$$t_{\text{transmisión}} = \frac{4.000 \text{ bits}}{250 \text{ kbps}} = 16 \cdot 10^{-3} \text{ s}$$

$$t_{\text{transmisión, ACK}} = \frac{500 \text{ bits}}{250 \text{ kbps}} = 2 \cdot 10^{-3} \text{ s}$$

$$\rho = \frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{prop}} + t_{\text{tx, ACK}} + t_{\text{prop, ACK}}} \approx 0,14 = 14\%$$

Control de flujo de ventana deslizante:

Puesto que $7 \cdot t_{\text{tx}} < t_{\text{tx}} + t_{\text{tx, ACK}} + t_{\text{prop, ACK}}$, el tamaño de la ventana no permite transmitir sin que haya que parar en algún instante para esperar la llegada de ACKs, por lo que:

$$\rho = \frac{7 \cdot t_{\text{tx}}}{t_{\text{tx}} + t_{\text{prop}} + t_{\text{tx, ACK}} + t_{\text{prop, ACK}}} \approx 0,95 = 95\%$$

a.2) Control de flujo de parada y espera:

$$t_{\text{transmisión}} = \frac{4.000 \text{ bits}}{10 \text{ Mbps}} = 40 \cdot 10^{-5} \text{ s}$$

$$t_{\text{transmisión, ACK}} = \frac{500 \text{ bits}}{10 \text{ Mbps}} = 5 \cdot 10^{-5} \text{ s}$$

Control de flujo de ventana deslizante:

Puesto que $7 \cdot t_{\text{tx}} > t_{\text{prop}} + t_{\text{tx, ACK}} + t_{\text{prop, ACK}}$, el tamaño de la ventana permite que no haya que dejar de transmitir en ningún instante, por lo que:

$$\rho = 1 = 100\%$$

b) En los casos en los que el retardo sea muy pequeño frente al tiempo de transmisión, será aceptable utilizar el protocolo de parada y espera, ya que en ese caso:

$$\rho = \frac{t_{\text{transmisión}}}{t_{\text{transmisión}} \left(1 + \frac{t_{\text{transmisión, ACK}}}{t_{\text{transmisión}}} + 2 \frac{t_{\text{propagación}}}{t_{\text{transmisión}}} \right)} \approx \frac{t_{\text{transmisión}}}{t_{\text{transmisión}} \left(1 + \frac{t_{\text{transmisión, ACK}}}{t_{\text{transmisión}}} \right)}$$

De este modo, el rendimiento supera el 50% y podrá estar muy cercano al 100% en el caso de que el tamaño del ACK sea muy pequeño frente al de las tramas de datos.

5.2. ¿Cuál debe ser el valor mínimo del *timeout* que emplea un protocolo de control de flujo?

SOLUCIÓN

Para que un protocolo de control de flujo funcione correctamente el tiempo mínimo que hay que esperar antes de retransmitir una trama deberá ser mayor que el tiempo que tardaría en recibirse el ACK de la trama que arranca el temporizador:

$$t_{\text{timeout}} > t_{\text{tx}} + t_{\text{tx, ACK}} + t_{\text{prop}} = 2 \cdot t_{\text{prop}} + t_{\text{tx}} \left(1 + \frac{t_{\text{tx, ACK}}}{t_{\text{tx}}} \right)$$

5.3. En un enlace con una velocidad de transmisión de 100 kbps y con una probabilidad de error de 10^{-4} en el que los paquetes que se envían tienen una cabecera de 300 bits, ¿qué longitud de paquete maximizará el número de bits transmitidos?

SOLUCIÓN

El número de bits transmitidos por unidad de tiempo será:

$$BW_{\text{efectivo}} = (\text{n.º de paquetes/s}) \cdot (\text{n.º bits útiles/paquete}) \cdot (\text{Probabilidad de que un paquete llegue sin errores})$$

Y, dado que:

$$\text{número de paquetes/sec} = \frac{BW}{L}$$

$$\text{número de bits útiles/paquete} = (L - L_c)$$

$$\text{probabilidad de que un bit sea erróneo} = P_e$$

$$\text{probabilidad de que un paquete llegue sin errores} = (1 - P_e)^L$$

Se llega a:

$$BW_{\text{efectivo}} = \frac{BW}{L} (L - L_c) \cdot (1 - P_e)^L$$

Donde:

- L es la longitud del paquete.
- L_c es la longitud de la cabecera.

Observe que en la expresión:

$$BW_{\text{efectivo}}(L) = \frac{BW}{L} (L - L_c) \cdot (1 - P_e)^L$$

Se pueden distinguir dos términos: $\frac{L - L_c}{L}$ y $(1 - P_e)^L$.

El primero de ellos tiende a 0 rápidamente cuando L se reduce y tiende a L_c . Este término modela el porcentaje de bits útiles que se transportan en cada paquete.

El segundo tiende a 0 cuando L crece y modela la probabilidad de que un paquete se reciba sin errores.

Para localizar el máximo derivamos e igualamos a 0:

$$\frac{dBW_{\text{efectivo}}}{dL} = 0 = BW \cdot \frac{L_c}{L^2} (1 - P_e)^L + BW \cdot \left(1 - \frac{L_c}{L}\right) \cdot (1 - P_e)^L \log_e (1 - P_e)$$

$$0 = \frac{L_c}{L^2} + \left(1 - \frac{L_c}{L}\right) \log_e (1 - P_e)$$

Despejando L se obtienen los dos posibles valores de L :

$$L_1 = 1.888 \quad \text{y} \quad L_2 = -1.588$$

Puesto que el tamaño del paquete únicamente puede tomar valores positivos, el tamaño del paquete que maximizará el número de bits transmitidos será $L_1 = 1.888$.

Observe que en la siguiente gráfica, en la cual se ha representado la relación $\frac{BW_{\text{efectivo}}}{BW}$ en función de L (el tamaño del paquete), se puede comprobar que el máximo se alcanza efectivamente cuando $L = 1.888$.

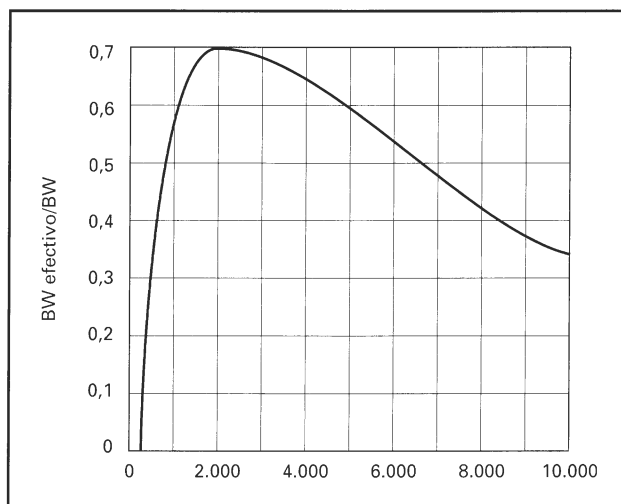


Figura 5.1. Eficiencia del enlace en función del tamaño del paquete ($\text{BER} = 10^{-4}$).

5.4.

¿En qué medida se incrementa la probabilidad de que una trama contenga errores debido a añadir un CRC de 32 bits en las tramas de 64 bytes cuando la probabilidad de error es 10^{-4} ?

¿Y en las tramas de 1.500 bytes?

SOLUCIÓN

Utilizando las siguientes definiciones:

P_e = Probabilidad de que 1 bit sea erróneo.

$P_{et}(L)$ = Probabilidad de que una trama de longitud L sea errónea.

El primer paso será encontrar la expresión que relaciona $P_{et}(L)$ y P_e :

$$P_{et}(L) = 1 - \text{Probabilidad de recibir } L \text{ bits sin errores} = 1 - (1 - P_e)^L$$

Utilizando esta expresión se llega a:

$$P_{et}(L = 8 \cdot 64) = 1 - (1 - 10^{-4})^{8 \cdot 64} = 0,0499$$

$$P_{et}(L = 8 \cdot 64 + 32) = 1 - (1 - 10^{-4})^{8 \cdot 64 + 32} = 0,0529$$

De donde se deduce que para tramas de 64 bytes la probabilidad de que una trama sea errónea se incrementa en:

$$\frac{P_{et}(8 \cdot 64 + 32) - P_{et}(8 \cdot 64)}{P_{et}(8 \cdot 64)} = \frac{0,0529 - 0,0499}{0,0499} \cdot 100 = 6\%$$

En el caso de que la trama fuese de 1.500 bytes:

$$P_{et}(L = 8 \cdot 1.500) = 1 - (1 - 10^{-4})^{8 \cdot 1.500} = 0,6988$$

$$P_{et}(L = 8 \cdot 1.500 + 32) = 1 - (1 - 10^{-4})^{8 \cdot 1.500 + 32} = 0,6998$$

Es decir, la probabilidad de que la trama sea errónea se incrementa en:

$$\frac{P_{et}(8 \cdot 1.500 + 32) - P_{et}(8 \cdot 1.500)}{P_{et}(8 \cdot 1.500)} = \frac{0,6998 - 0,6988}{0,6988} \cdot 100 = 0,1\%$$

Como puede apreciarse, la probabilidad de que una trama sea errónea crece rápidamente al crecer el tamaño de la trama. Sin embargo, como puede comprobarse en el ejemplo, el hecho de añadir un CRC de 32 bits no incrementa notablemente la probabilidad de que una trama sea errónea incluso cuando la trama es de tan sólo 64 bits.

5.5.

Dada la secuencia de bits $M = 111011101$ y el divisor $P = 110011$, ¿cuál será la longitud del CRC?

Calcule dicho CRC.

SOLUCIÓN

Teniendo en cuenta que el CRC es el resultado de dividir $M \cdot 2^n$ entre P (siendo $n + 1$ el número de bits de P), el CRC tendrá n bits.

En este caso, dado que $n + 1 = 6$, tendremos que el CRC estará constituido por 5 bits.

Para obtener el CRC realizamos las siguientes operaciones:

Se multiplica M por 2^n :

$$M \cdot 2^n = 111011101 \cdot 2^n = 11101110100000$$

Se divide $M \cdot 2^n$ entre P :

$$\begin{array}{r}
 Q = 101111110 \\
 P = 110011 \quad \overline{M \cdot 2^5 = 11101110100000} \\
 \underline{110011} \\
 00100010 \\
 \underline{110011} \\
 0100011 \\
 \underline{110011} \\
 0100000 \\
 \underline{110011} \\
 0100110 \\
 \underline{110011} \\
 0101010 \\
 \underline{110011} \\
 0110011 \\
 \underline{110011} \\
 R = 000010
 \end{array}$$

Figura 5.2. Cálculo del CRC.

El resto obtenido al realizar esta operación es el CRC que se pretendía calcular.

Es posible verificar que efectivamente ése es el CRC dividiendo $M \cdot 2^n + \text{CRC}$ entre P y comprobando que el resto de la operación es 0.

$$\begin{array}{r}
 Q = 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \\
 P = 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ \Big| \ M \cdot 2^5 + \text{CRC} = \begin{array}{r} 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ \hline R = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \end{array}
 \end{array}$$

Figura 5.3. Cálculo del CRC.

5.6.

Demuestre mediante un ejemplo que no es posible utilizar la técnica ARQ con vuelta atrás empleando una ventana mayor que la cantidad de los números de secuencia disponibles menos 1.

SOLUCIÓN

En el siguiente ejemplo, se ha considerado un sistema que utiliza 2 bits para numerar las tramas (por lo tanto, dispone de cuatro números de secuencia) y un tamaño de ventana 4.

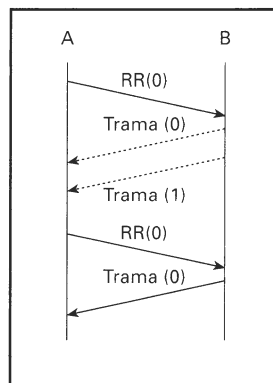


Figura 5.4. Error del protocolo ante la pérdida de dos tramas.

Como puede observarse, cuando el transmisor (B) recibe RR(0) no es capaz de darse cuenta de que las dos tramas que ha enviado se han perdido. Es más, considera que se están confirmando.

Por lo tanto, dado que en el ejemplo se ha utilizado la menor ventana posible entre todas las ventanas mayores que la cantidad de los números de secuencia disponibles menos 1, se concluye que efectivamente es cierta la proposición que se pretendía probar.

5.7.

Dadas dos estaciones A y B que se comunican utilizando el protocolo de enlace de datos HDLC y teniendo en cuenta las siguientes consideraciones:

Una trama errónea se representa como

$N(S)$ es el número de secuencia enviada.

$N(R)$ es el número de secuencia recibida.

$I(A,B) = I(N(S), N(R))$.

- a) Complete la secuencia de tramas suponiendo que ni A ni B tienen más tramas de información que enviarse.

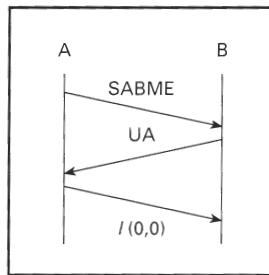


Figura 5.5. Transmisión de una trama de información.

- b) Indique cómo se resuelve la siguiente situación suponiendo que ni A ni B tienen más tramas de información que enviarse.

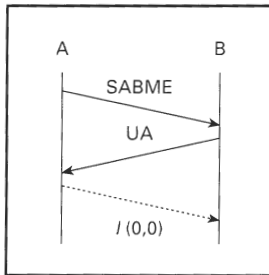


Figura 5.6. Pérdida de una trama de información.

- c) Indique cómo se resuelve la siguiente situación suponiendo que ni A ni B tienen más tramas de información que enviarse.

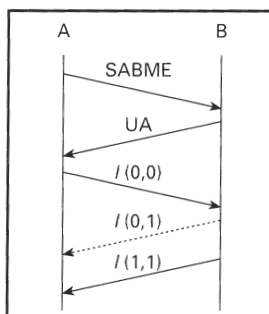


Figura 5.7. Pérdida de una trama de información.

SOLUCIÓN

a)

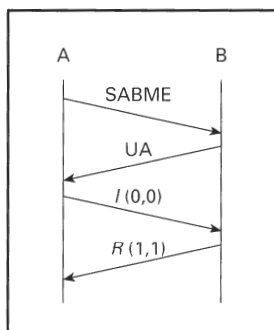


Figura 5.8. Confirmación de una trama de información.

b)

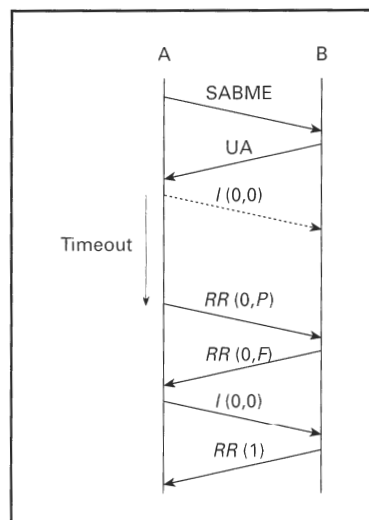


Figura 5.9. Recuperación de la pérdida de una trama de información.

c)

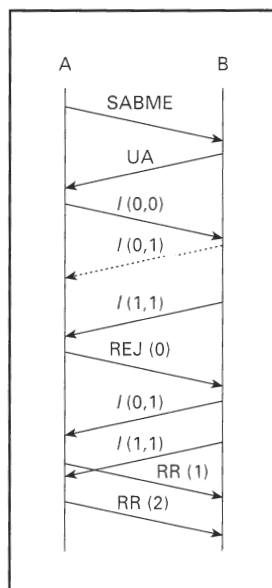
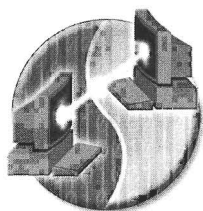


Figura 5.10. Recuperación de la pérdida de una trama de información.



MULTIPLEXACIÓN

6.1.

El sistema global para comunicaciones móviles (GSM) es un estándar europeo para telefonía celular. Este sistema ocupa la banda de frecuencias 935-960 MHz por lo que se refiere al canal de envío, que se divide en canales separados 200 KHz. A cada estación se le asigna el uso de una o más portadoras en su celda, cada una de las cuales transmite una señal digital que transporta canales de tráfico y control. La señal portadora se divide en multitramas de 120 ms; cada multitrama consta de 26 tramas y cada trama tiene 8 slots en los que se transmiten 114 bits por cada slot. Dos de las tramas de la multitrama son de control.

Determinar la tasa máxima de transferencia de datos de un canal de tráfico.

SOLUCIÓN

Un canal de tráfico a velocidad completa usa una sola ranura temporal en cada una de las tramas de tráfico en una multitrama, por lo que la velocidad de transmisión de un canal de este tipo es:

$$\begin{aligned}\text{Velocidad de canal de tráfico} &= 24 \text{ slots temporales/multitrama} \cdot 114 \text{ bits/slot} \cdot \\ &\cdot (1 \text{ multitrama}/120 \text{ ms}) = 22.800 \text{ bps}\end{aligned}$$

6.2.

Se utiliza un multiplexor por división de tiempo con mezcla de caracteres para combinar cadenas de datos de una serie de terminales asíncronos a 110 bps para transmitir datos a través de una línea digital a 2.400 bps. Cada terminal envía caracteres de 7 bits de datos, 1 bit de paridad, 1 bit de comienzo y 2 bits de parada. Supóngase que se envía un carácter de sincronización cada 19 caracteres de datos y que al menos el 3% de la capacidad de la línea se reserva para la inserción de pulsos, permitiendo así diferentes velocidades para los distintos terminales.

- Determinar el número de bits del carácter.
- Determinar el número de terminales que se pueden conectar al multiplexor.

SOLUCIÓN

Dado que se tienen 2.400 bps y las fuentes transmiten a 110 bps y que al menos un 3% de los bits transmitidos son bits insertados (72 bps), sólo puede haber un máximo de 21 fuentes transmitiendo a 110 bps ($2.400 - 72 = 2.328$ y $21 \cdot 110 = 2.310$ bps). Una de éstas, además, es utilizada por el canal de sincronización. Así, resulta que nos quedan 18 bps para repartir entre los terminales conectados.

Por tanto, el tamaño del carácter varía entre 11 bits (terminal transmite a 110 bps) y 29 bits (terminal transmitirá a 128 bps).

6.3.

Supóngase que dos terminales a 600 bps, cinco terminales a 300 bps y una serie de terminales a 150 bps se van a multiplexar en el tiempo usando un formato con mezcla de caracteres en una línea digital a 4.800 bps. Los terminales envían 10 bits/carácter y se inserta un carácter de sincronización cada 99 caracteres de datos. Todos los terminales son síncronos y se reserva un 3% de la capacidad de la línea para la inserción de pulsos para dar cabida a variaciones de los relojes de los terminales.

Determinar el número de terminales a 150 bps que se pueden conectar.

SOLUCIÓN

La línea dispone de 4.800 bps, de los cuales al menos un 3% se utiliza para inserción de bits. En tal caso, para transmitir datos quedan libres del orden de $4.800 (1 - 0,03) = 4.656$ bps.

Por otra parte, sabiendo que los caracteres tienen un tamaño de 10 bits cada uno se tiene que el número máximo de caracteres que puede transmitir la línea de 4.800 bps en un segundo es de $4.656/10 = 465$.

Una fuente a 600 bps transmitiendo caracteres de 10 bits requiere transmitir 60 caracteres en un segundo. Del mismo modo, una fuente a 300 bps requiere transmitir 30 caracteres en un segundo. Por tanto, las dos fuentes de 600 bps van a ocupar cada segundo 120 caracteres del total de 465 disponibles y las cinco fuentes de 300 bps van a ocupar 150. En total, esto supone que sobran 195 caracteres ($465 - 120 - 150 = 195$) para ser transmitidos por las fuentes de 150 bps.

Cada fuente de 150 bps requiere transmitir 15 caracteres cada segundo. Por tanto, en 195 posibles huecos se pueden situar 13 fuentes de 150 bps ($195/15 = 13$).

6.4.

Describir el funcionamiento de los elementos de conmutación temporal de los conmutadores.

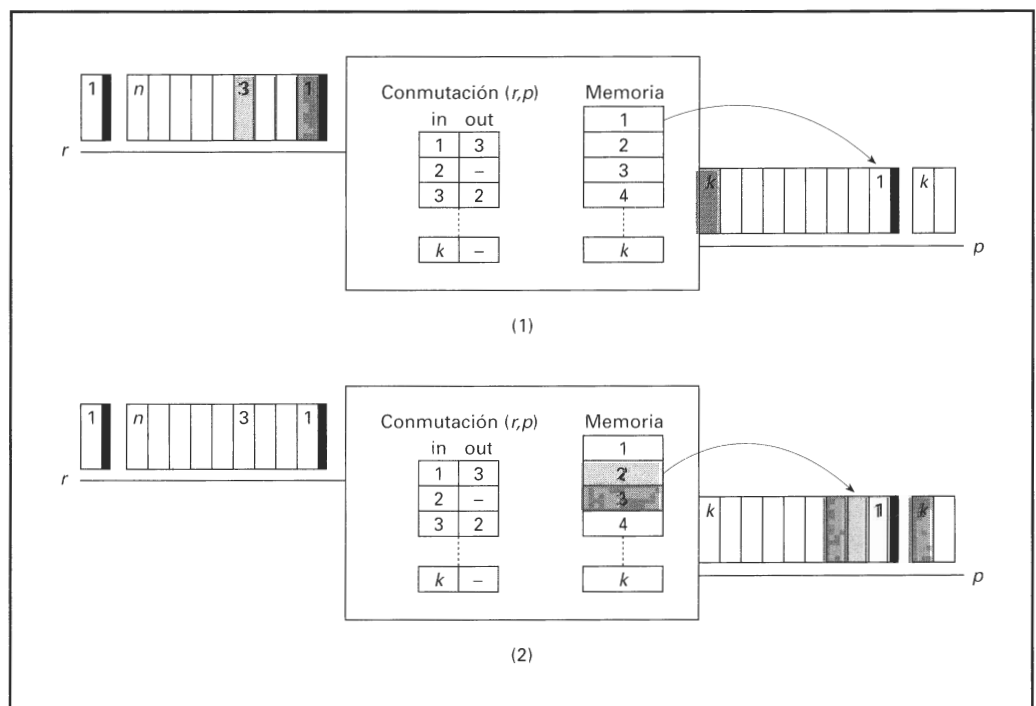


Figura 6.1. Conmutación temporal.

SOLUCIÓN

Los elementos de conmutación temporal tienen por objeto conmutar los canales en el tiempo. Este tipo de elementos forman parte, por ejemplo, de la etapa de entrada y salida de los conmutadores de tres etapas T-S-T. Su funcionamiento se ilustra en la figura anterior (Figura 6.1).

En este tipo de elementos se requiere almacenamiento donde depositar el contenido de los slots temporales y una tabla de conmutación de canales, con la cual se sabe qué slot de salida le corresponde a cada slot de entrada. En concreto, sobre la figura (Figura 6.1.(1)) se muestra cómo inicialmente la memoria no almacena ningún dato y la matriz de conmutación contiene una relación entre canales de entrada para la combinación entrada r y salida p del elemento de conmutación temporal. La trama de salida no contiene información porque, como se ha dicho previamente, la memoria no tiene información. Por otra parte, la entrada tiene información en los canales, pero se depositará a la salida una vez que se haya almacenado en la memoria.

En la siguiente figura (Figura 6.1.(2)) se muestra cómo se ha conmutado el contenido de los canales sobre la memoria, de modo que el canal de salida lo que hace es leer ordenadamente los elementos de memoria para saber qué datos se deben transmitir.

6.5.

Determinar la condición para que no haya bloqueos en un conmutador de tres etapas T-S-T.

SOLUCIÓN

Un conmutador T-S-T se puede ilustrar como muestra la siguiente figura (Figura 6.2). Se dispone de N/n elementos de conmutación temporal en las etapas de entrada y salida. Cada uno de estos elementos tiene una línea de entrada de n canales de entrada y k canales de salida. La etapa intermedia interconecta el total de los $k \cdot N/n$ de entrada con otros tantos canales de salida y que en grupos de tamaño k se distribuyen entre los elementos de conmutación de la etapa de salida.

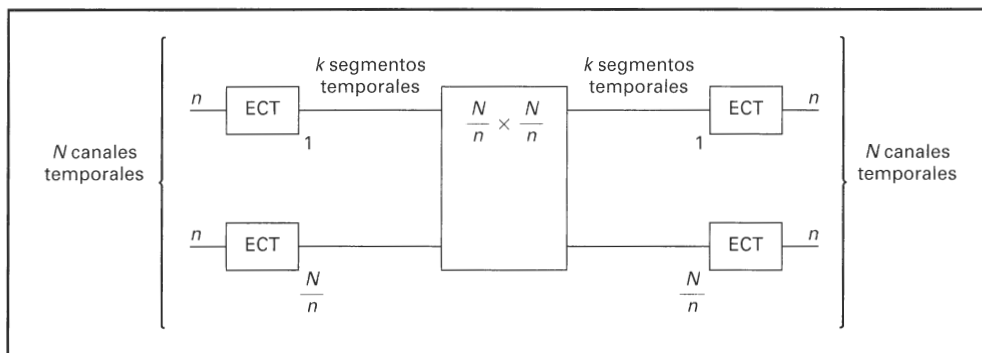


Figura 6.2. Estructura de los conmutadores T-S-T.

Vamos a suponer que están ocupados $(n - 1)$ de los n segmentos de tiempo en uno de los elementos de conmutación temporal (ECT) de la figura anterior (Figura 6.2). Éstos, a su vez, usan $(n - 1)$ de las k diferentes conexiones en tiempo del conmutador central. Supóngase que llega una llamada en el segmento del tiempo restante del mismo conmutador de segmento de tiempo de entrada, y que es dirigida a uno de los N/n conmutadores de segmento de tiempo de salida. Supóngase también que $(n - 1)$ segmentos de tiempo de salida de este conmutador de segmento de tiempo de salida están ocupados, o con llamadas ya establecidas. Éstos también corresponden a $(n - 1)$ de las k diferentes conexiones en tiempo del conmutador central. Supóngase, para el estudio del peor caso, que los dos conjuntos de $(n - 1)$ conexiones de cruces en el conmutador central están des-

conectados. Esta situación se muestra en la siguiente figura (Figura 6.3). En la Figura 6.2.(1) se tiene que el elemento de la capa de entrada tiene ocupados $(n - 1)$ elementos de los k posibles. Por otra parte, el elemento de conmutación temporal de salida tiene mapeados los canales de entrada sobre $(n - 1)$ canales de salida de los k posibles. Esto se ilustra en la Figura 6.3.(2), donde el eje de tiempos se muestra en línea con el del elemento de conmutación temporal de entrada. Los $(n - 1)$ canales a la salida están ocupados, pero no tienen que corresponder con los canales de la entrada que estamos considerando, pueden ser cualesquiera canales de entrada.

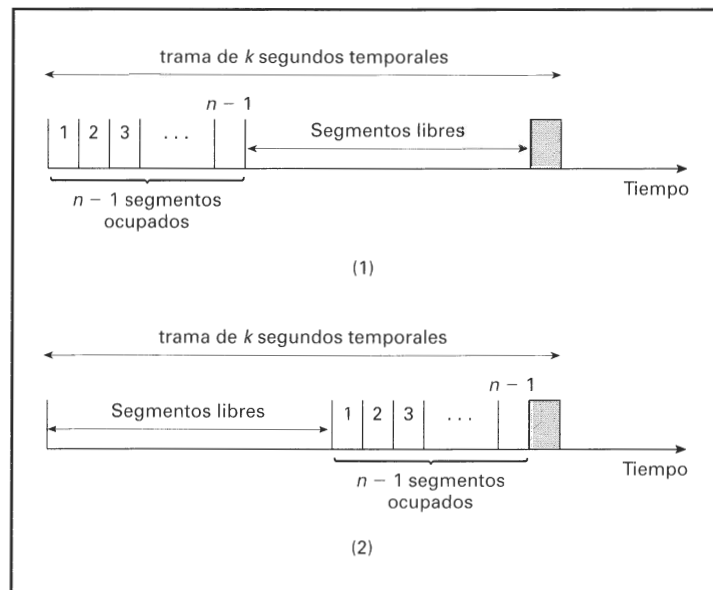


Figura 6.3. Ocupación de canales en la etapa de entrada (1) y de salida (2) en el peor caso.

Para que se establezca una comunicación entre los dos elementos de conmutación considerados (de entrada y salida), de manera simultánea, se requiere por lo menos un segmento libre en cada uno de los dos elementos de conmutación temporal.

La condición para que esto suceda, evitándose así el bloqueo de llamadas, es:

$$k = (n - 1) + (n - 1) + 1 = 2n - 1$$

6.6.

Determinar el número de cruces necesarios en un conmutador sin bloqueo con una sola etapa y T-S-T para conmutar 950 entradas con 950 salidas ($N = 950$).

SOLUCIÓN

Con un conmutador central de una sola etapa se requieren 950×950 cruces, del orden del millón de cruces.

En el caso de un conmutador T-S-T, sea $n = 125$ segmentos de tiempo de entrada los que se van a usar en los conmutadores de segmento de tiempo de entrada. Se requieren entonces $N/n = 8$ conmutadores de segmento de tiempo a la entrada, y un número igual a la salida. La condición de no bloqueo exige que haya $k = 2n - 1 = 249$ conexiones diferentes del conmutador central. Como $N/n = 8$, se necesita un conmutador central de 8×8 , lo que reduce bastante el número de cruces frente a los necesarios para un conmutador central de una sola etapa (10^6 cruces).

6.7. Determinar la probabilidad de bloqueo para un conmutador de tres etapas, homogéneo y con N entradas y salidas. Considerar que el bloqueo es debido únicamente a la etapa intermedia.

SOLUCIÓN

Por una parte, para que haya bloqueo es necesario que no se cumpla la condición de conmutador libre de bloqueo, por tanto, $k < 2n - 1$. Por otra parte, se tiene que tener que $k > n$ para que en las etapas de entrada y salida no haya bloqueo.

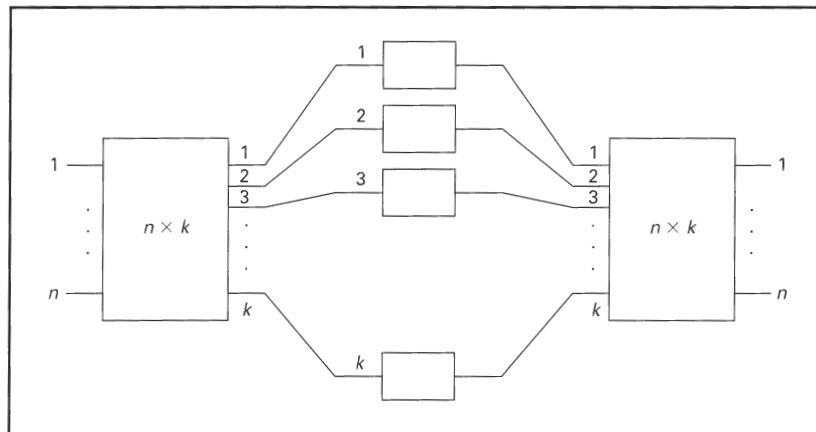


Figura 6.4. Cálculo de la probabilidad de bloqueo en un conmutador de tres etapas.

En la figura anterior (Figura 6.4) se muestra un elemento de la etapa de entrada, otro de la etapa de salida y los elementos intermedios. Supóngase que a es la ocupación media de un enlace. Al considerar un sistema homogéneo, se tiene que la ocupación de un canal de salida será también de valor a . A su vez, se va a considerar que el tráfico de entrada está uniformemente distribuido sobre los k enlaces entre etapas. Así, la probabilidad de que un enlace entre etapas esté ocupado es entonces $p = a \cdot n/k$.

Considerando que las probabilidades de ocupación de enlaces del conmutador son independientes, entonces la probabilidad de bloqueo es la probabilidad de que no exista una trayectoria libre entre el canal de entrada y el de salida.

El valor $(1 - p)$ es la probabilidad de que no esté ocupado un canal entre etapas. Cada “camino” dentro del conmutador necesita de dos de estos enlaces. Por tanto, la probabilidad de que no haya dos enlaces ocupados será $(1 - p)^2$. Así, la probabilidad de que uno de los k posibles caminos internos al conmutador esté ocupado es $(1 - (1 - p)^2)$. Si queremos conocer la probabilidad de bloqueo total, tenemos que tener en cuenta que hay k posibles caminos. Por tanto, la probabilidad de bloqueo se puede expresar del siguiente modo:

$$P_B = [1 - (1 - p)^2]^k$$

siendo:

$$p = a \cdot \frac{n}{k}$$

6.8.

Determinar el número de cruces necesario para un conmutador de tres etapas que conmuta 960 entradas con 960 salidas ($N = 960$), que proporcione una probabilidad de bloqueo menor de 0,1 o bien una probabilidad de bloqueo menor de 0,01 siendo la utilización del enlace de 0,7 Erlangs.

SOLUCIÓN

A continuación se muestra una tabla (Tabla 6.1) donde se relacionan los diferentes valores de probabilidad de bloqueo para diferentes posibles configuraciones del conmutador.

En la Tabla 6.1 se muestra la carga de los enlaces (a), el número de entradas o slots de tiempo (N) de la primera etapa del conmutador, el número de elementos de la primera etapa (también el número de entradas de la segunda etapa, (N/n), el número de salidas de los elementos de la primera etapa (k), el número de cruces para la segunda etapa (cruces), un valor intermedio ($p = a \cdot n/k$) usado en el cálculo de la probabilidad de bloqueo (P_b).

a	n	N/n	k	cruces	P	P_b
0,7	10	96	8	73.728	0,875	0,1181
0,7	10	96	9	82.944	0,778	0,0109
0,7	10	96	10	92.160	0,7	0,0008
a	n	N/n	k	cruces	P	P_b
0,7	20	48	15	34.560	0,933	0,1249
0,7	20	48	16	36.864	0,875	0,0139
0,7	20	48	17	39.168	0,824	0,0014
a	n	N/n	k	cruces	P	P_b
0,7	30	32	22	22.528	0,955	0,1319
0,7	30	32	23	23.552	0,913	0,0152
0,7	30	32	24	24.576	0,875	0,0016
a	n	N/n	k	cruces	P	P_b
0,7	40	24	29	16.704	0,966	0,1345
0,7	40	24	30	17.280	0,933	0,0156
0,7	40	24	31	17.856	0,903	0,0018
a	n	N/n	k	cruces	P	P_b
0,7	120	8	85	5.440	0,988	0,1284
0,7	120	8	86	5.504	0,977	0,0183
0,7	120	8	87	5.568	0,966	0,0024
a	n	N/n	k	cruces	P	P_b
0,7	160	6	113	4.068	0,991	0,1296
0,7	160	6	114	4.104	0,982	0,0159
0,7	160	6	115	4.140	0,974	0,0023
a	n	N/n	k	cruces	P	P_b
0,7	192	5	135	3.375	0,996	0,3389
0,7	192	5	136	3.400	0,988	0,0375
0,7	192	5	137	3.425	0,981	0,0052
a	n	N/n	k	cruces	P	P_b
0,7	240	4	169	2.704	0,994	0,1308
0,7	240	4	170	2.720	0,988	0,0165
0,7	240	4	171	2.736	0,982	0,002

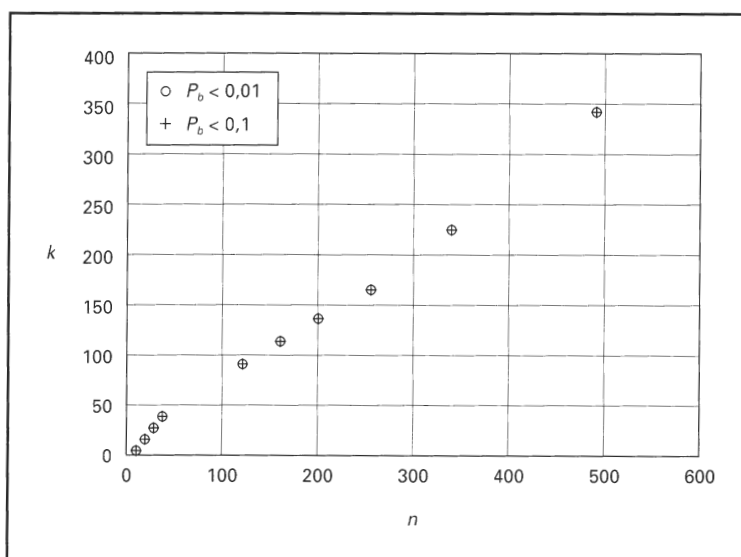
Tabla 6.1. Selección de k para obtener una probabilidad de bloqueo menor de 0,1 y 0,01.

a	n	N/n	k	cruces	P	P_b
0,7	320	3	225	2.025	0,996	0,1647
0,7	320	3	226	2.034	0,991	0,0168
0,7	320	3	227	2.043	0,987	0,0026
a	n	N/n	k	cruces	P	P_b
0,7	480	2	337	1.348	0,997	0,132
0,7	480	2	338	1.352	0,994	0,0171
0,7	480	2	339	1.356	0,991	0,0022

Tabla 6.1. (Continuación).

En este caso, se tiene que a medida que aumentan las entradas en los elementos de la primera etapa, se reduce la complejidad de la segunda etapa, aunque se requiere que los elementos de la primera etapa aumenten su complejidad (aumenta el valor de k para un mismo valor de probabilidad de bloqueo).

Estos resultados se resumen en las siguientes figuras (Figura 6.5, Figura 6.6). La siguiente figura (Figura 6.5) muestra la relación entre el valor de n y k para las diferentes probabilidades de bloqueo. Se ve que el valor de k para satisfacer las especificaciones de probabilidad de bloqueo son muy similares, por eso no se distingue la diferencia entre ambas sobre la gráfica. En la tabla anterior (Tabla 6.1) se ve que esta diferencia es de una unidad.

Figura 6.5. Relación entre n y k en conmutador de tres etapas ($N = 960$).

La siguiente figura (Figura 6.6) muestra la complejidad (medida en números de puntos de cruce) que supone el conmutador del problema en función del valor n . Se puede ver cómo la complejidad de la Etapa 1 aumenta linealmente con el valor de n . También se aprecia que la Etapa 2 no sigue esa tendencia, sino que se reduce a medida que aumenta el valor de n . Es decir, al complicarse la estructura de la Etapa 1, la Etapa 2 no requiere gran complejidad. Esto hace que la complejidad total siga la forma que se muestra en la figura. A partir de un valor inicial, desciende a un valor a partir del cual crece continuamente. Este valor mínimo para la complejidad del conmutador se puede utilizar como punto de diseño del conmutador.

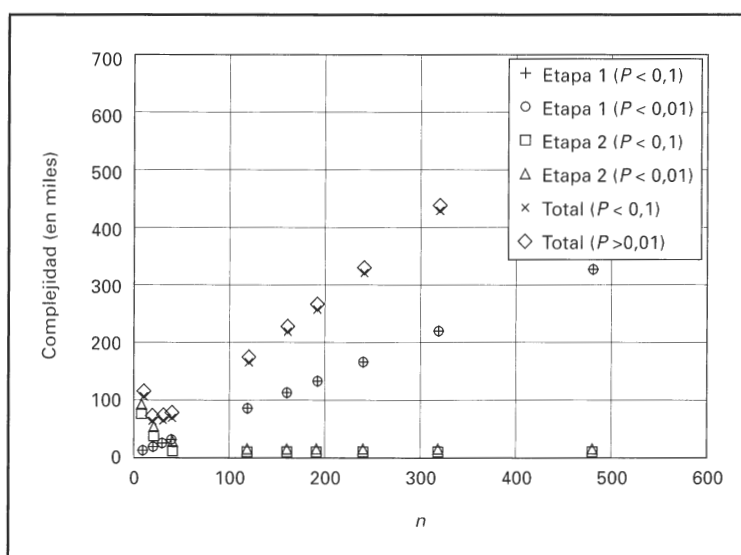


Figura 6.6. Complejidad de un conmutador de tres etapas ($N = 960$, $P.1$ y $P.01$).

6.9.

Determinar el número de cruces necesario para un conmutador de tres etapas que conmute 960 entradas con 960 salidas, con una probabilidad de bloqueo de 0,1 para una utilización del canal entre 0,5 y 0,9 Erlangs.

SOLUCIÓN

En este caso, al igual que en el problema anterior, se muestra la tabla (Tabla 6.2) que relaciona los diferentes parámetros del conmutador de tres etapas. En ella se muestra, para diferentes valores de carga del enlace (0,5-0,6-0,7-0,8-0,9) cuál es el valor de k que permite tener un valor de probabilidad de bloqueo menor o igual a 0,1 y la complejidad de dicho conmutador (medida en número de cruces de un conmutador espacial de tres etapas).

a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	10	96	7	64.512	0,714	0,009	71.232
0,6	10	96	8	73.728	0,750	0,010	81.408
0,7	10	96	9	82.944	0,778	0,010	91.584
0,8	10	96	10	92.160	0,800	0,011	101.760
0,9	10	96	11	101.376	0,818	0,012	111.936
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	20	48	12	27.648	0,833	0,012	39.168
0,6	20	48	14	32.256	0,857	0,013	45.696
0,7	20	48	16	36.864	0,875	0,013	52.224
0,8	20	48	18	41.472	0,889	0,014	58.752
0,9	20	48	20	46.080	0,900	0,014	65.280
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	30	32	17	17.408	0,882	0,014	33.728
0,6	30	32	20	20.480	0,900	0,014	39.680
0,7	30	32	23	23.552	0,913	0,015	45.632
0,8	30	32	26	26.624	0,923	0,015	51.584
0,9	30	32	29	29.696	0,931	0,015	57.536

Tabla 6.2. Efecto de la carga del enlace al seleccionar el valor k de un conmutador de tres etapas.

a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	40	24	22	12.672	0,909	0,015	33.792
0,6	40	24	26	14.976	0,923	0,015	39.936
0,7	40	24	30	17.280	0,933	0,015	46.080
0,8	40	24	34	19.584	0,941	0,016	52.224
0,9	40	24	38	21.888	0,947	0,0159	58.368
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	120	8	62	3.968	0,968	0,017	63.488
0,6	120	8	74	4.736	0,973	0,017	75.776
0,7	120	8	86	5.504	0,977	0,018	88.064
0,8	120	8	98	6.272	0,980	0,019	100.352
0,9	120	8	110	7.040	0,982	0,018	112.640
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	160	6	82	2.952	0,976	0,018	81.672
0,6	160	6	98	3.528	0,980	0,019	97.608
0,7	160	6	114	4.104	0,982	0,015	113.544
0,8	160	6	130	4.680	0,985	0,019	129.480
0,9	160	6	146	5.256	0,986	0,016	145.416
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	192	5	98	2.450	0,980	0,019	96.530
0,6	192	5	117	2.925	0,985	0,029	115.245
0,7	192	5	136	3.400	0,988	0,037	133.960
0,8	192	5	155	3.875	0,991	0,060	152.675
0,9	192	5	174	4.350	0,993	0,086	171.390
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	240	4	122	1.952	0,984	0,019	119.072
0,6	240	4	146	2.336	0,986	0,016	142.496
0,7	240	4	170	2.720	0,988	0,016	165.920
0,8	240	4	194	3.104	0,990	0,020	189.344
0,9	240	4	218	3.488	0,991	0,019	212.768
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	320	3	162	1.458	0,988	0,020	156.978
0,6	320	3	194	1.746	0,990	0,020	187.986
0,7	320	3	226	2.034	0,991	0,016	218.994
0,8	320	3	258	2.322	0,992	0,015	250.002
0,9	320	3	290	2.610	0,993	0,017	281.010
a	n	N/n	k	cruces	P	P_b	Complejidad
0,5	480	2	242	968	0,992	0,020	233.288
0,6	480	2	290	1.160	0,993	0,017	279.560
0,7	480	2	338	1.352	0,994	0,017	325.832
0,8	480	2	385	1.540	0,997	0,098	371.140
0,9	480	2	434	1.736	0,995	0,012	418.376

Tabla 6.2. (Continuación).

Estos datos se pueden resumir gráficamente y así obtenemos las siguientes figuras (Figuras 6.7 y 6.8).

En la Figura 6.7 se presenta la relación entre n y k para los diferentes valores de la carga del enlace. En este caso se puede apreciar que para una carga constante del enlace, k debe aumentar proporcionalmente con n .

Por otra parte, la variación de la carga en los enlaces tiene como efecto que la proporción que debe aumentar k es mayor cuanto mayor es la carga en los enlaces. Esto tiene como consecuencia que para un valor de $n = 480$ se tiene que para una carga de red $a = 0,5$ se requiera un valor $k = 250$ y para $a = 0,9$, un valor aproximado de $k = 450$. Sin embargo, para valores de n menores que 50 se tiene que las diferencias en los valores de k no son tan acusadas. Esto va a tener su reflejo en la complejidad del sistema.

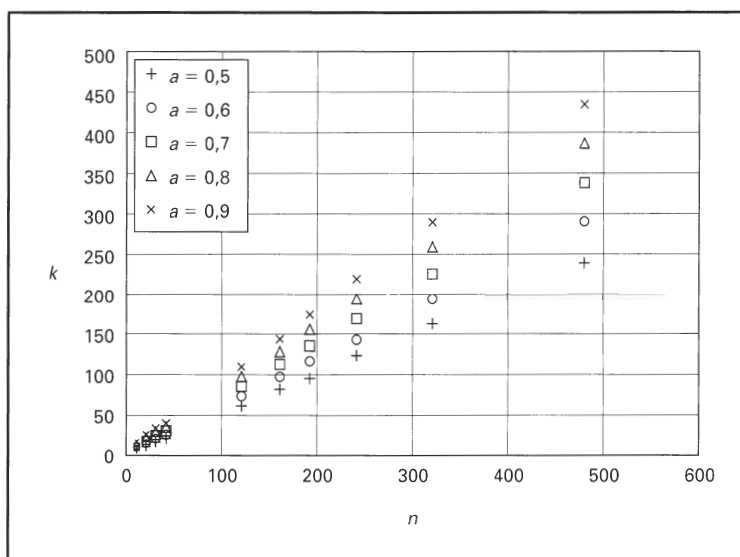


Figura 6.7. Relación entre n y k para diferentes cargas de los enlaces de entrada ($N = 960$, $P.1$).

A partir de la tabla anterior (Tabla 6.2) se puede relacionar gráficamente (Figura 6.8) el valor de n con la complejidad del sistema total para cada una de las posibles cargas de enlace consideradas.

En este caso se ve cómo para cada carga de enlace la complejidad crece al aumentar n , pero cruzando un mínimo antes de empezar a crecer continuamente.

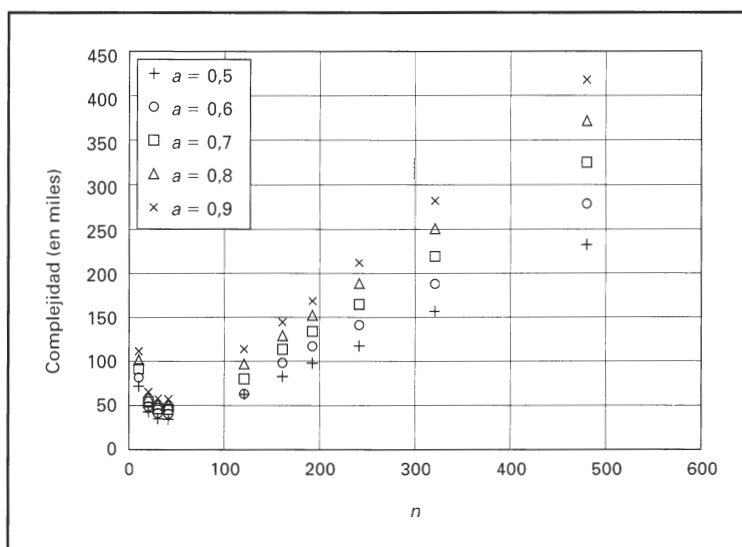


Figura 6.8. Complejidad del conmutador para diferentes cargas de red ($N = 960$, $P.1$).

6.10. SONET define una jerarquía de niveles de señalización denominados Señales de Transporte Síncrona (STS). Cada nivel STS (STS-1 a STS-192) soporta una cierta tasa de datos. Los enlaces físicos definidos para transportar cada nivel STS se denominan portadoras ópticas (OC). Por otra parte, SDH especifica un sistema similar denominado Módulo de Transporte Síncrono (STM). Determinar la relación entre estos tres elementos: STS, OC y STM.

SOLUCIÓN

La relación entre los elementos STS, OC y STM se muestra en la siguiente tabla (Tabla 6.3).

STS	OC	Velocidad (Mbps)	STM
STS-1	OC-1	51,840	
STS-3	OC-3	155,520	STM-1
STS-9	OC-9	466,560	STM-3
STS-12	OC-12	622,080	STM-4
STS-18	OC-18	933,120	STM-6
STS-24	OC-24	1.244,160	STM-8
STS-36	OC-36	1.866,230	STM-12
STS-48	OC-48	2.488,320	STM-16
STS-96	OC-96	4.976,640	STM-32
STS-192	OC-192	9.953,280	STM-64

Tabla 6.3. Relación STS, OC y STM.

6.11. Un sistema SONET se divide en tres capas: secciones, caminos y líneas. Identifique en la Figura 6.9 dichas capas.

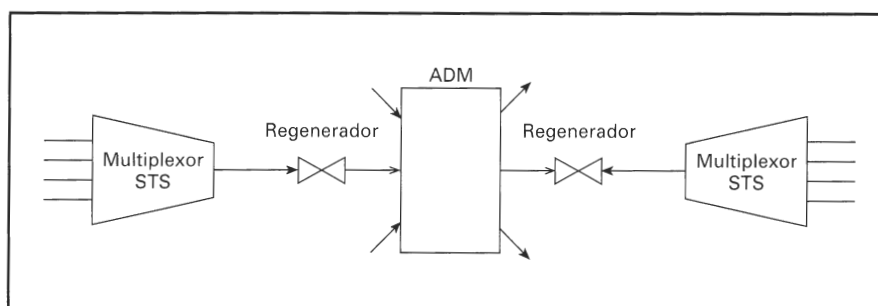


Figura 6.9. Ejemplo de segmento de red SONET.

SOLUCIÓN

Una sección es el trozo de fibra entre dos dispositivos adyacentes y se encarga de la transmisión de una señal STS- n a través del medio físico. Una línea se refiere al enlace entre dos multiplexores adyacentes y en general comprende varias secciones. Un camino corresponde al sistema entre dos terminales SONET en los extremos del sistema, y en general, comprende una o más líneas.

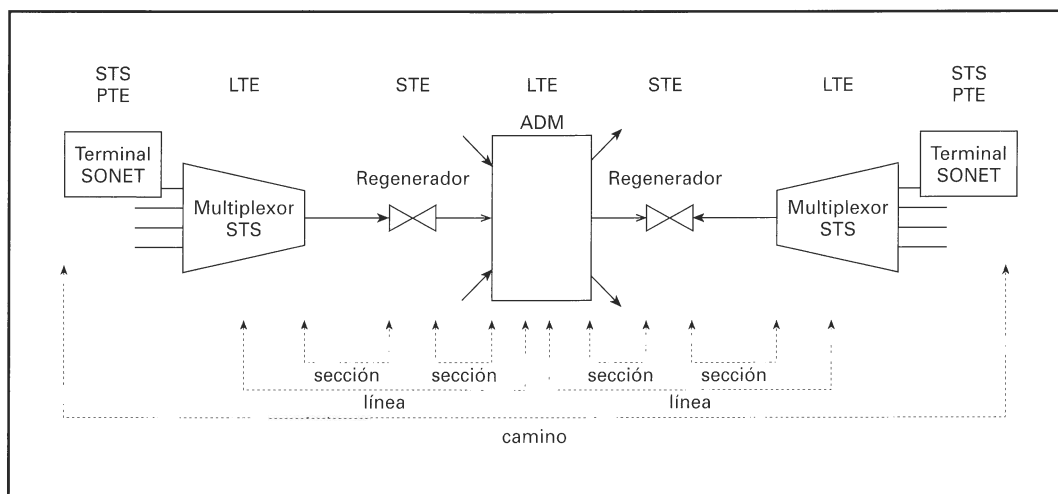


Figura 6.10. Capas SNET sobre segmento de red.

STE: equipo terminal de sección; por ejemplo un repetidor.

LTE: equipo terminal de línea; por ejemplo, un multiplexor STS-1 a STS-3.

PTE: equipo terminal de camino; por ejemplo, un multiplexor STS-1.

6.12.

Determinar el tamaño en bytes de la carga útil de una trama SNET (SPE – *Synchronous Payload Envelope*).

SOLUCIÓN

La trama SNET tiene la estructura de una matriz de bytes de 9 filas por 90 columnas que se transmite en 125 microsegundos. Cada fila contiene tres bytes (los tres primeros de cada fila) que se usan para implementar las cabeceras de sección y línea, esto se refleja en la Figura 6.11.

Por tanto, la carga útil se calcula fácilmente como:

$$\text{Carga útil} = 9 \text{ filas} \cdot 87 \text{ bytes por fila} \cdot 8 \text{ bits por byte} \cdot 8.000 \text{ veces por segundo} = 50,122 \text{ Mbps}$$

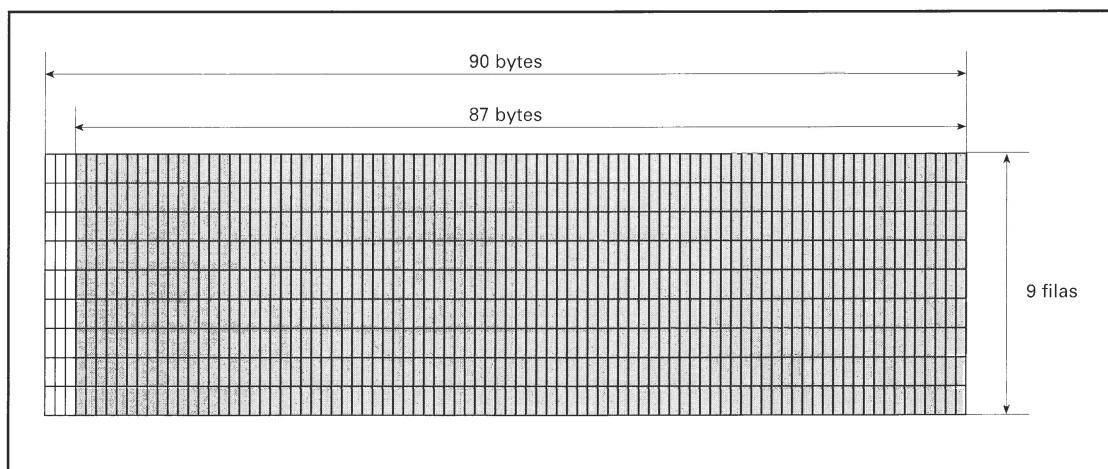


Figura 6.11. Estructura de trama SNET.

6.13. SONET tiene que soportar la compatibilidad hacia atrás con las jerarquías actuales. Para ello, SONET propone un sistema de tributarias virtuales (VTs). ¿Cómo hace SONET para soportar las jerarquías digitales americana DS-1 (1,544 Mbps) y europea E-1 (2,048 Mbps)?

SOLUCIÓN

Una tributaria virtual es una carga parcial que se puede insertar en una STS-1 y combinar con otras cargas parciales para rellenar la trama. En lugar de usar todas las 86 columnas de la carga de una trama STS-1 para los datos que provienen de su origen, se puede dividir la SPE y llamar a cada componente tributaria virtual.

Se han definido varias tributarias virtuales las cuales se diferencian en el número de columnas que utilizan cada una de ellas.

$$\begin{aligned}
 \text{VT1.5 (1,5} \cdot 2 = 3 \text{ columnas)} &= 8.000 \text{ tramas/s} \cdot 3 \text{ cols} \cdot 9 \text{ filas} \cdot 8 \text{ bits} = 1,728 \text{ Mbps} \\
 \text{VT2 (2} \cdot 2 = 4 \text{ columnas)} &= 8.000 \text{ tramas/s} \cdot 4 \text{ cols} \cdot 9 \text{ filas} \cdot 8 \text{ bits} = 2,304 \text{ Mbps} \\
 \text{VT3 (3} \cdot 2 = 6 \text{ columnas)} &= 8.000 \text{ tramas/s} \cdot 6 \text{ cols} \cdot 9 \text{ filas} \cdot 8 \text{ bits} = 3,456 \text{ Mbps} \\
 \text{VT6 (6} \cdot 2 = 12 \text{ columnas)} &= 8.000 \text{ tramas/s} \cdot 12 \text{ cols} \cdot 9 \text{ filas} \cdot 8 \text{ bits} = 6,912 \text{ Mbps}
 \end{aligned}$$

Como se puede ver, con una tributaria VT1.5 se puede alojar una señal T-1 y para alojar una señal E-1 se requiere al menos una tributaria VT2.

6.14. ¿Cuál es la estructura de la trama STS-3?

SOLUCIÓN

Para construir la trama STS-3 se parte de tres tramas STS-1. Cada una de ellas se sincroniza con el reloj local del multiplexor. Para ello se eliminan las cabeceras de sección y línea de la señal STS-1 de entrada y la carga útil (SPE) se transforma en una nueva trama STS-1, sincronizada con el reloj local. Este procedimiento garantiza que todas las tramas STS-1 entrantes se transforman en tramas STS-1 sincronizadas entre sí.

La trama STS-3 se obtiene alternando los bytes de las 3 tramas STS-1 sincronizadas de forma que se construya una trama con 9 filas, $9 (3 \times 3)$ columnas de cabeceras de sección y línea y $261 (87 \times 3)$ columnas de carga útil.

En general, la trama STS- n se obtiene alternando los bytes de las n tramas STS-1 sincronizadas de forma que se construya una trama con 9 filas, $9 (3 \times n)$ columnas de cabeceras de sección y línea y $261 (87 \times n)$ columnas de carga útil.

Por otra parte, para multiplexar k señales STS- n en una trama STS- kn , las señales de entrada son previamente descompuestas en tramas STS-1 y luego se aplica el procedimiento anterior (Figura 6.12).

La estructura de la trama STS- n en general tiene la forma siguiente:

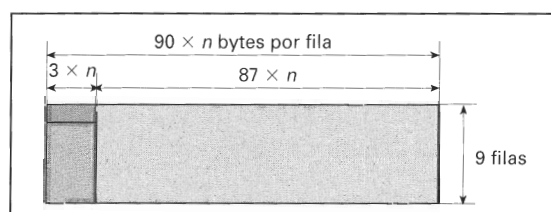


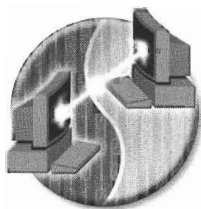
Figura 6.12. Estructura de la multiplexación de tramas STS.

6.15.**Calcule el número de canales de voz que se pueden transportar por señales STS-1, STS-3 y STS-12.****SOLUCIÓN**

Considerando el sistema europeo de jerarquía (E-1), se requiere una tributaria virtual VT2 para transportar la señal E-1. Es decir, 30 canales de voz. Esto supone el uso de 4 columnas de la señal STS-1. Por tanto, el número de señales E-1 que pueden transportarse en una señal STS-1 es 21 ($87/4$). En total, por tanto, $30 \cdot 21 = 630$ canales de voz.

Para el caso de señales STS-3, éstas transportan tres veces más canales de voz que una señal STS-1. Por tanto, 1.890 canales de voz. La señal STS-12 supone la multiplexación de 32 señales STS-1, por tanto, 7.560 canales de voz.

En el caso del sistema americano, la señal T-1 requiere el uso de una tributaria virtual VT1.5 que requiere 3 columnas para su transmisión. En tal caso, se tiene que en una señal STS-1 se pueden transportar hasta 29 ($87/3$) señales T-1 y en total $24 \cdot 29 = 696$ canales de voz. La señal STS-3 transportará $3 \cdot 696 = 2.088$ canales de voz y la señal STS-12 transportará $12 \cdot 696 = 12.362$ canales de voz.



CONMUTACIÓN DE CIRCUITOS

7.1.

Cuántas conversaciones de voz puede soportar: (1) un sistema de conmutación con 40 circuitos DTMF de registro de línea simultáneamente; (2) un conmutador por división de tiempo con bucles de 30 canales y (3) una oficina central con una concentración 5 : 1 aproximadamente y 640 circuitos de línea.

SOLUCIÓN

El número de usuarios marcando al mismo tiempo es de 40. Es decir, en cada instante como mucho, puede haber 40 usuarios utilizando el circuito. Debido al corto espacio de tiempo que se va a usar el registro se puede considerar que cada circuito se va a usar por 60 usuarios en el período de una hora. A esta tasa, si las llamadas se reparten a lo largo de la hora, 40 circuitos pueden dar servicio a 2.400 usuarios en una hora.

En cuanto al conmutador con matriz de conmutación por división de tiempo, se tiene que el número de conversaciones está limitado por los canales disponibles. En este caso, se tiene que por cada bucle se dispone de 30 canales. A su vez, cada conversación requiere un canal de voz para cada participante (dos canales de voz, uno para cada participante en una conversación entre dos). Por tanto, el número de conversaciones que se pueden gestionar por bucle es de 15. En una PBX pequeña, lo habitual es disponer de dos bucles de 30 canales. En oficinas centrales lo habitual es equipar a los conmutadores con bucles de 30 canales por cada 150 líneas de teléfono atendidas.

En cuanto a centrales grandes, la norma es la de proporcionar 120 canales de voz por cada 640 circuitos de línea, lo que resulta en una concentración algo superior a 5 : 1.

7.2.

En este problema se trata de calcular el valor correspondiente en Erlangs de 6 CCS y los CCS que corresponden a 6 Erlangs.

SOLUCIÓN

En telecomunicaciones, el tráfico se expresa en Erlangs. Un Erlang se corresponde con una hora de tráfico. Por ejemplo, si un circuito se usa durante una hora, éste ha tenido un tráfico/carga de 1 Erlang.

Otra unidad para medir el tráfico tiene en cuenta los cientos de segundos de la llamada, y se denomina CCS (100 *Call Seconds* – 100 en números romanos es C).

La relación entre ambas se puede ver a partir de considerar que 1 Erlang se corresponde a una hora y ésta tiene 3.600 s. Por ejemplo, 1 Erlang son 36 CCS.

Aplicando los resultados anteriores se tiene que 6 CCS corresponden a 0,16 Erlangs y que 6 Erlangs corresponden a 216 CCS.

7.3.

Calcular la demanda máxima (Erlangs y CCS) para uno de los bucles de un conmutador por división de tiempo que mantiene 15 conversaciones.

SOLUCIÓN

Como se ha visto en el problema anterior (Problema 7.2), 15 conversaciones de voz requieren 30 canales. La demanda máxima se tiene cuando todos los canales están ocupados a lo largo del tiempo. Por tanto, la demanda máxima supone 30 Erlangs. Esto equivale a $30 \text{ Erlangs} \cdot 36 \text{ CCS} / 1 \text{ Erlang} = 1.080 \text{ CCS}$.

7.4.

Determinar las expresiones para la carga ofrecida, carga cursada y carga bloqueada en un conmutador telefónico, así como para el número total de llamadas ofrecidas. Considerar el caso de una duración media de llamada de 3 minutos, donde los usuarios realizan, de media, 80 intentos de llamada por hora, lo cual hace que todos los circuitos del conmutador estén ocupados el 2% del tiempo dando lugar a que un 10% de las llamadas no se pueda establecer.

SOLUCIÓN

La carga ofrecida es el total de la carga cursada más la carga bloqueada. La carga cursada se refiere al conjunto de comunicaciones que se han realizado, lo cual equivale a un tiempo total dependiente de la media de duración de la llamada (AHT – *Average Holding Time*) y del número de llamadas realizadas (PC – *Peg Count*). Por su parte, la carga bloqueada está determinada por el número de llamadas que se han perdido porque al llegar éstas el conmutador tenía todas sus líneas ocupadas (ATB – *All Trunk Busy*). En general, estas variables están relacionadas del siguiente modo:

$$\begin{aligned} \text{demanda o carga ofrecida} &= \text{carga cursada} + \text{carga bloqueada} \\ \text{demanda o carga ofrecida} &= (\text{PC} \cdot \text{AHT}) + (\text{ATB} \cdot \text{AHT}) \end{aligned}$$

Por tanto:

$$\text{carga ofrecida} = (\text{PC} + \text{ATB}) \cdot \text{AHT}$$

En cuanto al número total de llamadas ofrecidas se puede expresar del siguiente modo:

$$\text{número total de llamadas ofrecidas} = \text{PC} + \text{ATB}$$

En concreto, para el caso que plantea el problema, las variables toman los siguientes valores: AHT = 3 minutos y número total de llamadas = 80 llamadas cada hora. Por tanto, la carga ofrecida es de 240 Erlangs.

La carga cursada se obtiene tras determinar el número de llamadas cursadas. Es decir, del total de intentos de llamada, se tiene que un 10% no se ha podido cursar. Por tanto, $\text{PC} = 80 - 10\% 80 = 72$ llamadas cursadas. Esto permite determinar finalmente que:

$$\text{carga cursada (en cada hora)} = 72 \text{ llamadas por hora} \cdot 3 \text{ minutos} = 216 \text{ Erlangs}$$

$$\text{carga bloqueada (en cada hora)} = 8 \text{ llamadas por hora} \cdot 3 \text{ minutos} = 24 \text{ Erlangs}$$

7.5.

Las tres variables fundamentales utilizadas en ingeniería de tráfico son: (1) demanda o carga, (2) número de circuitos y (3) el grado de servicio (GoS). La demanda es la carga ofrecida. La carga medida es la carga cursada. Si la demanda supera los recursos existentes, entonces se produce el bloqueo de la carga ofrecida. El grado de servicio representa los usuarios afectados por el bloqueo. Una solución es aumentar los recursos para reducir el bloqueo de las comunicaciones. Sin embargo, esto, normalmente, supone un coste muy elevado.

En ingeniería de tráfico, a la hora de determinar la demanda, se promedia utilizando las diez horas de mayor ocupación a lo largo de todo el año (ABBH – Average Bouncing Busy Hour). Este es el parámetro de mayor peso a la hora de determinar la capacidad del sistema.

Para el diseño de oficinas centrales se utiliza el número de llamadas que puede gestionar un conmutador en una hora. En este caso, la carga ofrecida o demanda es función de la duración media de las llamadas (AHT – Average Holding Time) y del número total de llamadas que se producen (PC – Peg Count): $\text{demanda} = \text{AHT} \times \text{PC}$.

Dada la siguiente tabla, donde se muestra el estudio de tráfico recogido durante los días y horas laborales de una semana determinar la carga ofrecida con la que tiene que trabajar el ingeniero para dimensionar los equipos de conmutación. En la tabla (Tabla 7.1) se muestran para cada día de la semana las llamadas cursadas (PC), las llamadas perdidas (ATB) y el tráfico cursado en CCS. Cada fila se corresponde con las medidas en cada hora. Por ejemplo, la primera fila corresponde a los datos medidos entre las 8:00 y las 9:00.

	Lunes			Martes			Miércoles			Jueves			Viernes		
Hora	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB
8:00	250	400	15	140	400	10	150	300	10	240	300	10	150	200	20
9:00	230	430	18	250	500	25	250	400	25	250	390	25	190	300	40
10:00	285	500	20	230	400	20	275	570	30	275	530	35	230	550	55
11:00	270	480	15	200	380	20	240	400	27	250	380	25	210	400	25
12:00	280	450	17	190	300	15	250	390	25	200	400	25	150	350	35
13:00	220	370	12	200	390	18	190	370	20	190	390	25	220	200	20
14:00	170	300	9	190	270	22	150	270	10	150	250	25	120	300	10
15:00	190	430	13	210	250	20	250	250	10	250	200	10	150	250	10
16:00	210	390	16	195	290	15	150	210	10	290	190	10	100	200	5

Tabla 7.1. Estudio de tráfico.

SOLUCIÓN

En primer lugar hay que determinar las horas cargadas, ya que son las que van a determinar la carga ofrecida para dimensionar el sistema. Esta evaluación se refleja en la Tabla 7.2.

A la hora de determinar la carga ofrecida los datos necesarios se pueden resumir en la Tabla 7.3.

A partir de la Tabla 7.3 se puede ver fácilmente que la carga media cursada en la hora cargada promedio es de $530 \text{ CCS} = 14,9 \text{ Erlangs}$. En dicha hora, se han producido en promedio 299 llamadas de las cuales 263 se han atendido (ABBH PC) y 33 se han bloqueado (ABBH ATB).

Para calcular la carga ofrecida, tenemos que calcular el tiempo medio de duración de la llamada (AHT). Este valor se calcula dividiendo la carga ofrecida entre el número de llamadas. Así

	Lunes			Martes			Miércoles			Jueves			Viernes		
Hora	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB	PC	CCS	ATB
8:00	250	400	15	140	400	10	150	300	10	240	300	10	150	200	20
9:00	230	430	18	250	500	25	250	400	25	250	390	25	190	300	40
10:00	285	500	20	230	400	20	275	570	30	275	530	35	230	550	55
11:00	270	480	15	200	380	20	240	400	27	250	380	25	210	400	25
12:00	280	450	17	190	300	15	250	390	25	200	400	25	150	350	35
13:00	220	370	12	200	390	18	190	370	20	190	390	25	220	200	20
14:00	170	300	9	190	270	22	150	270	10	150	250	25	120	300	10
15:00	190	430	13	210	250	20	250	250	10	250	200	10	150	250	10
16:00	210	390	16	195	290	15	150	210	10	290	190	10	100	200	5
BH	285	500	20	250	500	25	275	570	30	275	530	35	230	550	55
Total	2.105	3.750	135	1.805	3.180	165	1.905	3.160	167	2.095	3.030	190	1.520	2.750	220

Tabla 7.2. Valores totales de tráfico y de hora cargada (BH – *Busy Hour*).

Día	Hora cargada	ABBH CCS	ABBH PC	ABBH ATB
Lunes	10:00	500	285	20
Martes	9:00	500	250	25
Miércoles	10:00	570	275	30
Jueves	10:00	530	275	35
Viernes	10:00	550	230	55
Total para las 5 horas		2.650	1.315	165
ABBH		530 CCS	263 PC	33 ATB

Tabla 7.3. Resumen de valores ABBH para el estudio de tráfico.

tenemos que $AHT = CCS / PC = 530 / 263 = 2,01$ CCS, aproximadamente 2 CCS o bien 200 s por llamada, lo cual corresponde a 3 minutos y 20 segundos por llamada.

Una vez conocido el tiempo medio de duración de la llamada se puede estimar el tráfico ofrecido como la suma del tráfico cursado más el tráfico bloqueado. El tráfico bloqueado se calcula como: *tráfico bloqueado* = $AHT \cdot ATB = 2 \text{ CCS} \cdot 33 \text{ llamadas} = 66 \text{ CCS}$. Así, el tráfico ofrecido resulta: *tráfico ofrecido* = tráfico cursado + tráfico bloqueado = $530 \text{ CCS} + 66 \text{ CCS} = 596 \text{ CCS}$. Esta carga en Erlangs se corresponde a 16,55 Erlangs.

Otra manera de calcular el tráfico ofrecido es calcular el número total de llamadas (PC, ATB) y multiplicarlo por el tiempo medio de duración de la llamada (ABBH AHT). En el estudio anterior, ABBH PC = 263 y ABBH ATB 33. El número total de intentos de llamada = $263 + 33 = 299$ llamadas, y por tanto, el *tráfico ofrecido* = $299 \cdot 2 \text{ CCS AHT} = 598 \text{ CCS}$. La diferencia entre los dos resultados es debida al redondeo que se ha realizado al determinar el valor del tiempo medio de duración de la llamada.

7.6.

A la hora de realizar la estimación del tráfico, una medida de interés es la del porcentaje del tráfico diario que se ha ofrecido durante la hora cargada (ABH – *Average Busy Hour*). Utilizando el valor de ABH, estimar el tráfico telefónico de una empresa considerando que el estudio de tráfico que refleja el uso de los recursos se corresponde con la Tabla 7.3 del Problema 7.5. Considerar que el mes tiene 22 días de actividad laboral y que las llamadas requieren un tiempo de establecimiento de 20 segundos.

SOLUCIÓN

Para estimar el número de horas mensual de uso telefónico hay que hacer uso del valor del tráfico cursado, ya que a la empresa le facturan por el tráfico cursado. Por otra parte, el tiempo de facturación se corresponde con el tiempo desde que se establece la llamada hasta que ésta finaliza. Es decir, el tiempo de establecimiento del canal de voz no se factura.

Para estimar el tráfico convertimos el valor medio de tráfico cursado ABBH CCS a un valor mensual:

$$\text{ABBH CCS mensual} = \frac{\text{tráfico cursado}}{\text{ABH} \cdot 22 \text{ días de trabajo}}$$

Donde ABH es el porcentaje del tráfico diario que se ha ofrecido durante la hora cargada. Este valor se calcula sumando el tráfico total cursado en la hora cargada de cada día: $(500 + 500 + 570 + 530 + 550) = 2650 \text{ CCS}$ y dividiendo el resultado entre el total del tráfico cursado en el total de los días: $(3.750 + 3.180 + 3.160 + 3.030 + 2.750) = 18.900$. En este caso, $\text{ABH} = 14\%$. Es decir, el 14% del tráfico ocurre durante las 5 horas más cargadas. Se puede calcular el valor de ABH% para cada día dividiendo el valor de la hora cargada de cada día entre el total del tráfico cursado en ese día, pero la aproximación anterior es suficiente para el propósito que estamos buscando de estimar el tráfico mensual.

Con objeto de estimar el tráfico mensual a cursar aplicamos la expresión anterior para el tráfico cursado y tenemos en cuenta que evaluamos para 22 días de trabajo. Por tanto, la estimación resulta: $(14,9 \text{ Erlangs} / 0,14) \cdot 22 \text{ días} = 2.341 \text{ Erlangs u horas}$. Sin embargo, en este tiempo calculado se está acumulando tanto el tiempo de uso del canal de voz como el tiempo de establecimiento de la llamada. A continuación vamos a eliminar dicho tiempo para tener una estimación más ajustada para el tiempo de uso de recursos que se aplicará a la empresa. Teniendo en cuenta que el tiempo de setup es de 20 segundos, entonces:

$$\begin{aligned} \text{ABBH PC} &= 263 \text{ llamadas} \\ \text{Setup time} &= 263 \cdot 0,20 \text{ CCS} = 52,6 \text{ CCS} \\ \text{Tiempo de conversación} &= \text{tráfico cursado} - \text{setup} \\ \text{CCS} &= 530 - 52 = 478 \text{ CCS} = 13,3 \text{ Erlangs} \\ \text{Total horas estimadas} &= \frac{13,3}{0,14} \cdot 22 = 2.090 \text{ horas} \end{aligned}$$

Eliminar el tiempo de setup ha reducido el número total de horas en $251 \cdot (2.341 - 2.090)$.

7.7.

Se trata de determinar el valor de la hora cargada conociendo cuál es la facturación en horas que tiene una empresa. La situación es la siguiente, a una empresa le facturan 2.090 horas, 9.430 llamadas y se sabe que la relación entre el tráfico en la hora cargada y el tráfico total está dada por un valor ABH% del 14%.

SOLUCIÓN

La carga cursada se calcula de la siguiente manera:

$$ABBH = \frac{\text{tráfico facturado} + \text{núm. llamadas} \cdot \text{setup time}}{\text{núm. días de trabajo}} \cdot ABH\%$$

Hay que tener en cuenta que el tráfico cursado de 2.090 horas son $2.090 \times 36 = 75.240$ CCS. Así:

$$ABBH = \frac{75.240 + 9.430 \cdot 0,2}{22} \cdot 0,14 = 490 \text{ CCS}$$

Este valor de 490 CCS para el tráfico cursado es menor que el que se había calculado en el problema anterior (Problema 7.6). Esto es debido a que el valor de ABBH PC% no es realmente el 14% del total de llamadas (PC). Si se calcula dicho valor teniendo en cuenta el estudio de tráfico, se obtiene un valor de $1.315 / = 13,9\%$ frente a $14,01\%$. En general, ABBH PC% y ABBH CCS% no son siempre iguales, pero son suficientemente próximos como para que el uso de ABBH CCS% sea una buena aproximación cuando no se dispone de un estudio de tráfico.

7.8.

Se dispone de un conmutador que cursa entre sus circuitos un tráfico de 30 Erlangs, siendo AHT = 2 minutos el tiempo medio de duración de las conversaciones de voz. Determinar la tasa de llegada de las llamadas.

SOLUCIÓN

El conjunto cursa un tráfico de 30 Erlangs, que corresponden a 1.080 CCS. Entonces, 1.080

$$\text{CCS} = 1.800 \text{ min} \left(\frac{108.000}{60} = 1.800 \text{ min} \right). \text{ Si el tiempo de duración de la llamada (AHT) es de}$$

$$2 \text{ in } (1,2 \text{ CCS}), \text{ entonces se tendrían } 900 \text{ llamadas} \left(\text{PC} = \frac{\text{tráfico cursado}}{\text{AHT}} \right). \text{ Es decir,}$$

$$\frac{1.080 \text{ CCS}}{1,2 \text{ CCS}} = 900 \text{ llamadas, o de otra manera } \frac{1.800 \text{ min}}{2 \text{ min}} = 900 \text{ llamadas. Si se tienen 900 llamadas (PC) en una hora (60 min), entonces se tienen que tratar } \frac{900}{60} \text{ llamadas por minuto, lo cual resulta en una tasa de llegada de 15 llamadas por minuto.}$$

Se puede realizar el cálculo de manera más directa haciendo la división del total del tráfico cursado entre el tiempo de duración de la llamada en minutos. Es decir, la tasa de llegada = tráfico cursado (Erlangs) / AHT (min) = $30/2 = 15$ llamadas por minuto.

7.9.

Justificar por qué se puede realizar el cálculo directo que se ha indicado en el problema anterior (Problema 7.8) para determinar la tasa de llegada de las llamadas.

SOLUCIÓN

El cálculo realizado se basa en el uso de las siguientes expresiones:

$$\text{ABBH PC (llamadas) en 60 minutos} = \frac{\text{tráfico cursado en 60 minutos}}{\text{AHT por minuto}}$$

$$T \text{ (tasa de llegada de llamadas por minuto)} = \frac{\text{llamadas en 60 minutos}}{60 \text{ minutos}}$$

La segunda expresión determina el número de llamadas por minuto. Sustituyendo en esta expresión el valor del número de llamadas, se obtiene lo siguiente para la tasa de llegada de llamadas por minuto (T):

$$T \text{ (llamadas por minuto)} = \frac{\text{tráfico cursado en 60 min/AHT por min}}{60 \text{ min}}$$

Teniendo en cuenta que el tráfico cursado está en Erlangs, la expresión anterior resultante es:

$$T \text{ (llamadas por minuto)} = \frac{\text{Erlangs en 60 minutos / AHT por minuto}}{60 \text{ minutos}}$$

$$T \text{ (llamadas por minuto)} = \frac{\text{Erlangs} \times 60 \text{ min/AHT min}}{60 \text{ minutos}}$$

Finalmente, simplificando la expresión anterior se obtiene:

$$T \text{ (llamadas por minuto)} = \frac{\text{Erlangs}}{\text{AHT min}}$$

7.10.

Se dispone de un conmutador que cursa entre sus circuitos un tráfico de 30 Erlangs, siendo AHT = 2 minutos el tiempo medio de duración de las conversaciones de voz. Determinar el tiempo medio entre llamadas.

SOLUCIÓN

Una vez conocida la tasa de llegada de las llamadas (Problema 7.9), se puede calcular el tiempo medio entre llamadas como el inverso de la tasa de llegada. Así, en este caso se tiene que el tiempo medio entre llamadas es $1/15 = 0,066$ min. Es decir, llega una llamada cada 4 s, así se tendrá que llegan 15 llamadas cada 15 minutos.

7.11.

Al dimensionar los sistemas telefónicos se hace uso de diferentes tablas: Poisson, Erlang B y Erlang C entre otras. Indicar en qué situaciones se aplica cada una de ellas.

SOLUCIÓN

A la hora de diseñar los sistemas telefónicos se hace uso de tablas en las que se relaciona la carga, el grado de servicio o probabilidad de bloqueo (GoS) y el número de circuitos que puede soportar la carga para ese grado de servicio. El uso de una tabla depende principalmente del tratamiento que se esté considerando para las llamadas bloqueadas. Así tenemos que las llamadas pueden almacenarse en una cola hasta que se atiendan (BCH — *Blocked Calls Held*). En este caso se aplican las Tablas de Poisson.

Otra posibilidad para tratar a las llamadas bloqueadas es eliminarlas del sistema (BCC — *Blocked Calls Cleared*). En este caso, se aplican las Tablas de Erlang B . En este caso también se pueden usar las Tablas de Neal-Wilkinson. Este último conjunto de tablas considera que las llamadas no llegan aleatoriamente, sino que tiene en cuenta que hay situaciones de sobrecarga puntuales.

Si se considera que las llamadas bloqueadas se reintentan inmediatamente, se pueden utilizar tanto las Tablas de Poisson, como las denominadas Tablas de Erlang *B* extendidas. Estas tablas fueron desarrolladas por Henry Jacobsen. Estas tablas se utilizan para tener en cuenta que no todos los usuarios reintentan la llamada inmediatamente. En tal caso, se podría usar Erlang *B*.

Si se considera que las llamadas bloqueadas se almacenan en una cola o se les introduce retardo, entonces se utilizan las tablas de Erlang *C*. Las fuentes de retardo para las llamadas pueden ser, por ejemplo, los sistemas automáticos de distribución de llamadas (ACD – Automatic Call Distribution). Estas tablas asumen que los usuarios esperan indefinidamente a que se les atienda y, por tanto, el tráfico ofrecido y cursado serán iguales.

En resumen, cuando se bloquea un intento de llamada, el sistema redirige la llamada a otro grupo de circuitos, coloca la llamada en una cola o devuelve al usuario una señal indicando que no hay recursos disponibles. Un grupo de circuitos que redirige las llamadas cuando todos los circuitos están ocupados se dimensiona bien con Poisson, o bien con Erlang *B*. El dimensionamiento de grupos de circuitos que van a encolar las llamadas se realiza mediante Erlang *C*. Cuando se considera que el usuario reintenta la llamada inmediatamente tras recibir la señal de tono ocupado se puede utilizar Poisson o Erlang *B*. En este caso, también se puede utilizar la metodología de reintentos para dimensionar el sistema, en la que se aplican las tablas Erlang *B* extendidas.

7.12.

Se dispone de un conmutador telefónico que cursa un tráfico de 18,75 Erlangs (675 CCS) y se desea obtener un grado de servicio (GoS) de *P*.10 (probabilidad de bloqueo de un 1%). Determinar el número de circuitos de los que debe disponer el sistema considerando que es un sistema troncal final o bien que es un grupo de gran uso. Repetir el cálculo para un GoS de *P*.01.

SOLUCIÓN

A la hora de dimensionar un grupo troncal final se utiliza la Tabla de Poisson, en el otro caso se utiliza la Tabla Erlang *B*.

Para el primer caso, se busca en la Tabla de Poisson el GoS deseado (se reproduce a continuación una parte de la Tabla de Poisson - Tabla 7.4). En la columna ERL se busca el primer valor mayor o igual que el valor en Erlangs de tráfico cursado (18,75 Erlangs en nuestro caso). Sobre la Tabla 7.4,

GoS troncales	B.01		B.05		B.10	
	ERL	CCS	ERL	CCS	ERL	CCS
23	13,33	480	15,72	566	17,11	616
24	14,08	507	16,56	596	17,97	647
25	14,86	535	17,39	626	18,83	678
26	15,61	562	18,22	656	19,72	710
27	16,38	590	19,06	686	20,58	741
28	17,16	618	19,92	717	21,47	773
29	17,97	647	20,75	747	22,36	805
<u>30</u>	18,75	675	21,61	778	23,22	836

Tabla 7.4. Valores totales de tráfico y de hora cargada (BH – Busy Hour).

GoS troncales	P.01		P.05		P.10	
	ERL	CCS	ERL	CCS	ERL	CCS
22	13,64	491	17,14	617	19,69	709
23	14,47	521	18,08	651	20,78	748
24	15,28	550	19,03	685	21,78	784
25	16,11	580	20	720	22,83	822
26	16,97	611	20,94	754	23,89	860
27	17,81	641	21,89	788	24,94	898
28	18,64	671	22,86	823	26	936
29	19,49	702	23,83	858	27,05	974

Tabla 7.5. Tabla de Erlang *B*.

el valor que cumple lo anterior es 18,83. Mirando sobre la primera columna en la fila donde se encuentra este valor, se tiene que el número de troncales necesarias es 25. Es decir, con 25 troncales se puede dar un GoS de probabilidad de bloqueo de un 1% a un tráfico cursado entre 17,98 y 18,83 Erlangs. Del mismo modo, para un GoS de P.01 se tiene que el número de troncales necesario sería 30.

En el segundo caso, se aplica la Tabla de Erlang *B* (se reproduce a continuación una parte de la Tabla de Erlang *B* - Tabla 7.5). En este caso, en la citada tabla se puede ver que para un GoS de P.1 se requieren 22 circuitos, mientras que para un GoS de P.01 se requieren 29 circuitos.

Comparando ambos resultados se puede ver que para un GoS de P.1 en el primer caso (Poisson) se requieren 25 troncales mientras que para el segundo caso (Erlang *B*) se requieren 22 troncales. Según la aproximación poissoniana se sobredimensiona el sistema en tres troncales. Esto también se puede apreciar para un GoS de P.01 aunque no es tan clara la diferencia.

El uso de la aproximación poissoniana tiende a sobredimensionar el sistema. Es decir, se ahorran circuitos si se usa la aproximación de Erlang *B*. Sin embargo, el uso de las diferentes aproximaciones no responde a criterios de economizar circuitos, sino al tratamiento que se sigue con las llamadas que alcanzan un sistema saturado (llamadas bloqueadas). En caso de que las llamadas no se eliminen ni se pongan en un buffer, se utiliza Poisson (es el caso de los grupos finales); mientras que si las llamadas se eliminan del sistema y se pasan a otro sistema para su realización, se usa Erlang *B*.

7.13.

El dimensionamiento de los sistemas telefónicos requiere conocer la carga ofrecida. En este problema se trata de determinar la carga ofrecida siguiendo la metodología de reintentos de H. Jacobsen para calcular la carga ofrecida. Se considera que estudios de tráfico anteriores indicaban que el tráfico cursado promedio en la hora cargada (ABBH) era 150 CCS; el número de intentos de llamada (PC) era 72; el número de llamadas bloqueadas (ATB) era de 8; y se asume que para el 50% de las llamadas bloqueadas se realizan reintentos de llamada inmediatamente. Determinar el número de circuitos para tener un grado de servicio P.10 y P.01.

Resolución

La aplicación de la metodología de reintentos tiene por objetivo la obtención del tráfico ofrecido. En este caso se denomina *tráfico en el primer intento* y corresponde con el tráfico ofrecido. Si no

hay bloqueo, el tráfico se cursa en el primer intento y el tráfico ofrecido y cursado son iguales. La expresión que propone H. Jacobsen para determinar el tráfico en el primer intento tiene en cuenta los reintentos. Esta expresión se muestra a continuación, donde TPI denota el valor del tráfico en el primer intento:

$$\text{TPI} = \text{tráfico cursado} \cdot [1 - (\text{reintentos\%} \cdot \text{bloqueos\%}) / (1 - \text{bloqueos\%})]$$

Para aplicar los datos del problema a la expresión sabemos que:

$$\text{tráfico cursado} = 150 \text{ CCS}$$

$$\text{bloqueos\%} = 0,5 \text{ (50\%)}$$

$\text{reintentos\%} = 8$ llamadas se bloquean del total de 80 llamadas ($\text{PC} + \text{ATB} = 72 + 8 = 80$). Así, $\text{reintentos\%} = 8/80 = 0,1$ (10%).

Aplicando los datos sobre la expresión del tráfico en el primer intento se obtiene lo siguiente:

$$\begin{aligned} \text{TPI} &= \text{tráfico cursado} \cdot \left(\frac{1 - \text{reintentos\%} \cdot \text{bloqueos\%}}{1 - \text{bloqueos\%}} \right) = \\ &= 150 \text{ CCS} \cdot \left(\frac{1 - (0,5 \cdot 0,1)}{1 - 0,5} \right) = 150 \cdot 1,055 = 158,33 \text{ CCS} \end{aligned}$$

Ahora se debe consultar en la tabla Erlang B extendida para un 50% de reintentos (se muestra a continuación un extracto de la misma –Tabla 7.6). La tabla usa horas (Erlangs), por tanto hay que convertir el valor anterior a Erlangs. 158,33 CCS equivalen a $158,33/36 \text{ Erlangs} = 4,398 \text{ Erlangs}$. Este valor no está en la Tabla 7.6 y el primer número mayor es el de 4,40. Este valor, en la Tabla 7.6 que considera un 50% de reintentos, tiene asociado un número de circuitos dependiendo del GoS que se desee.

Tráfico en primer intento	Tráfico cursado	GoS (%)	Circuitos
4,20	4,19	P.07%	10
4,40	3,58	P.315%	5
<u>4,40</u>	<u>4,17</u>	<u>P.098</u>	<u>7</u>
4,40	4,29	P.049	8
<u>4,40</u>	<u>4,38</u>	<u>P.01</u>	<u>10</u>

Tabla 7.6. Erlang B extendida (50%).

Para un GoS P.10 (probabilidad de bloqueo del 1%), la Tabla 7.6 indica que son necesarios siete circuitos.

Para un GoS P.01 (probabilidad de bloqueo del 1%), la Tabla 7.6 muestra que son necesarios diez circuitos.

En la Tabla 7.6 también se muestra el tráfico cursado por el sistema (recordar que se dimensiona en función del tráfico ofrecido). En este caso, para un GoS de P.10 se tiene que la tabla da un va-

lor de 4,17 Erlangs para el tráfico cursado. Según los datos del enunciado se puede calcular el tráfico cursado como $150 \text{ CCS} / 36 = 4,166 \text{ Erlangs}$, que se ajusta bastante al valor del calculado en la tabla. Por tanto, se puede decir que en este caso la tabla de H. Jacobsen proporciona el tráfico cursado en horas y el tráfico ofrecido en horas de primer intento.

7.14. Interpretar la expresión de H. Jacobsen para el caso en el que se realiza un 100% de reintentos.

SOLUCIÓN

En principio, lo que cabe esperar es que el tráfico cursado sea igual que el tráfico en el primer intento, ya que al realizar intentos para todas las llamadas bloqueadas, éstas finalmente podrán ser cursadas. Por tanto, el tráfico en el primer intento (TPI) será

$$\begin{aligned} \text{TPI} &= \text{tráfico cursado} \cdot 1 - \frac{\text{reintentos\%} \cdot \text{bloqueos\%}}{1 - \text{bloqueos\%}} \\ &= \text{tráfico cursado} \cdot \frac{1 - \text{bloqueos\%}}{1 - \text{bloqueos\%}} = \text{tráfico cursado} \end{aligned}$$

7.15. Se considera que estudios de tráfico anteriores indicaban que el tráfico cursado promedio en la hora cargada (ABBH) era 150 CCS; el número de intentos de llamada (PC) era 72; el número de llamadas bloqueadas (ATB) era de 8; y se asume que para el 70% de las llamadas bloqueadas se realizan reintentos de llamada inmediatamente. Determinar el número de circuitos para tener un grado de servicio P.01.

SOLUCIÓN

El valor de 70% de reintentos es el típico para los sistemas telefónicos y los valores para este porcentaje son los que originalmente desarrolló H. Jacobsen. Esta tabla (Tabla 7.7) se denomina de Jacobsen en su honor y los valores de tráfico están en CCS. Aplicando la expresión de H. Jacobsen para calcular el tráfico en el primer intento y utilizando los datos del problema, se obtiene que el tráfico en el primer intento (TPI) es:

$$\begin{aligned} \text{TPI} &= \text{tráfico cursado} \cdot 1 - \frac{\text{reintentos\%} \cdot \text{bloqueos\%}}{1 - \text{bloqueos\%}} \\ &= 150 \cdot 1 - \frac{0,7 \cdot 0,1}{1 - 0,4} = 150 \cdot 1,033 \end{aligned}$$

Tráfico en el primer intento = 155 CCS

Tráfico en primer intento	P.01	
	Tráfico ofrecido	Tráfico cursado
8	108	107
9	131	130
10	154	154
11	179	178

Tabla 7.7. Tabla de H. Jacobsen (70%).

Así, mirando en la Tabla 7.7 se puede ver que para el grado de servicio requerido se necesitan diez circuitos.

7.16.

Dado un sistema donde se pueden encolar las llamadas (cola infinita) y que cursa un tráfico de 72 CCS, en 36 llamadas (PC) y que dispone de tres circuitos, se quiere saber:

- ¿Cuál es la probabilidad de retardo?
- ¿Cuál es el retardo medio de todas las llamadas?
- ¿Cuál es el retardo medio de las llamadas que se retardan?
- ¿Cuál es el número medio de llamadas en cola en cualquier instante de tiempo?
- ¿Cuál es el número medio de llamadas en cola cuando el sistema está ocupado?
- ¿Cuál es la probabilidad de que una llamada tenga que esperar 50 s?

SOLUCIÓN

En primer lugar, se tiene que considerar una tabla Erlang C (Tabla 7.8) para resolver el problema, ya que estamos considerando que las llamadas se encolan y esperan hasta que finalmente son atendidas. En tal caso, la carga cursada y ofrecida será la misma. Por tanto, la carga ofrecida se puede saber ya que el sistema cursa 72 CCS (2 Erlangs). Mirando en la Tabla 7.8 Erlang C para 2 Erlangs y fijándose en la fila asociada a tres circuitos, se responde a las preguntas del enunciado.

N	P	D1	D2	Q1	Q2	P8	P4	P2	P1	PP
3	.444	.44	1.0	.89	2.0	.39	.35	.27	.16	.16
4	.1739	.09	.50	.17	1.0	.14	.11	.06	.02	.00
5	.0597	.02	.33	.04	.67	.04	.03	.01	.00	.00

Tabla 7.8. Extracto de la tabla de Erlang C para 2 Erlangs.

- ¿Cuál es la probabilidad de retardo?

$$P = 0,4444 \quad \text{o} \quad 44,44\%$$

- ¿Cuál es el retardo medio de todas las llamadas?

Para resolver esta cuestión es necesario utilizar el valor D1 de la Tabla 7.8. D1 representa el retardo medio para todas las llamadas en unidades de tiempo de duración de la llamada (D1 es un factor multiplicador). Así, el tiempo medio de duración de la llamada (AHT) es 2 CCS (72 CCS / 36 PC) y $2 \cdot 0,44$ (D1) = 0,88 CCS. Esto equivale a 88 s o bien 1 min y 28 s.

- ¿Cuál es el retardo medio de las llamadas que se retardan?

Se requiere conocer cuál es valor del retardo para las llamadas retrasadas. El valor D2 de la Tabla 7.8 proporciona el retardo medio para las llamadas retrasadas (D2 es un factor multiplicador). Así, dado que el tiempo medio de duración de la llamada (AHT) es 2 CCS (72 CCS/36 PC). Corrigiendo este valor según propone D2, entonces 2×1.0 (D2) = 2 CCS. Esto equivale a 200 s o bien 3 min y 20 s.

d) ¿Cuál es el número medio de llamadas en cola en cualquier instante de tiempo?

Este valor lo indica el parámetro Q1 de la Tabla 7.8 de Erlang C para 2 Erlangs.

e) ¿Cuál es el número medio de llamadas en cola cuando el sistema está ocupado?

Este valor lo indica el parámetro Q2 de la Tabla 7.8 de Erlang C para 2 Erlangs.

f) ¿Cuál es la probabilidad de que una llamada tenga que esperar 50 s?

El tiempo medio de duración de la llamada (AHT) es 2 CCS. Por otro lado, 50 s son 0,5 CCS, que corresponde a un cuarto de la duración de la llamada (2 CCS). En la Tabla 7.8 de Erlang C el parámetro P4 representa la probabilidad de esperar al menos un cuarto del tiempo de duración de la llamada. El valor de P4 en este caso es 0,35. Por tanto, la probabilidad que se busca es de un 35%.

7.17.

Determinar el número de circuitos necesarios para tener un retardo medio de 90 s sabiendo que se cursan 72 CCS en 36 llamadas (PC).

SOLUCIÓN

En este caso hay que determinar qué parámetros de la Tabla 7.9 nos sirven para calcular retardos. Por un lado, los valores de P están relacionados con el bloqueo o probabilidad de retrasarse, pero no da idea de retardos. Q1 y Q2 se refieren al número de llamadas en cola. Los valores de P8 a PP se refieren a la probabilidad de esperar desde 1/8 al doble del tiempo medio de duración de la llamada. Por tanto, tampoco dan información acerca del retardo. Sólo los valores D1 y D2 se refieren al retardo.

N	P	D1	D2	Q1	Q2	P8	P4	P2	P1	PP
3	.444	.44	1.0	.89	2.0	.39	.35	.27	.16	.16
4	.1739	.09	.50	.17	1.0	.14	.11	.06	.02	.00
5	.0597	.02	.33	.04	.67	.04	.03	.01	.00	.00

Tabla 7.9. Extracto de la tabla de Erlang C para 2 Erlangs.

De acuerdo con la Tabla 7.9 de Erlang C para 2 Erlangs, que corresponde a la duración media de la llamada (AHT), $72 \text{ CCS} / 36 \text{ PC} = 2 \text{ Erlangs}$, o lo que es lo mismo 3 min y 20 s.

Como se ha visto en el problema anterior (Problema 7.16) el retardo medio para todas las llamadas se calcula multiplicando el factor que aparece en la Tabla 7.9 por el tiempo medio de duración de la llamada. Así, en caso de mirar sobre la columna D1 se tiene que el retardo medio por llamada es del orden de $0,44 \times 2 \text{ CCS} = 0,88 \text{ CCS}$ (88 s) que cumple lo que buscamos (inferior o igual a 90 s). Por tanto, si atendemos al retardo para el conjunto de llamadas con tres circuitos tenemos suficiente.

Sin embargo, si atendemos al retardo para las llamadas que se retrasan (D2) se tiene que si usamos tres circuitos, entonces el retardo medio para las llamadas es $1,00 \cdot 2 \text{ CCS} = 2 \text{ CCS} = 3 \text{ min } 20 \text{ s}$, que es mucho mayor que los 90 s que tenemos por objetivo. Si consideramos un circuito más, cuatro en total, se tiene que el retardo obtenido es $0,50 \cdot 2 \text{ CCS} = 1 \text{ CCS} = 100 \text{ s}$, también superior a los 90 s o menos deseados. En el caso de utilizar cinco circuitos obtenemos que $0,33 \times 2 \text{ CCS} = 0,66 \text{ CCS} = 66 \text{ s}$, lo cual cumple los requisitos.

Como se ha visto, hay dos posibilidades a la hora de determinar el retardo. En cada caso se usará la que más se ajuste a las especificaciones del diseño. En este caso, no se ha especificado claramente y por eso no podemos decantarnos por una de ellas. De todas formas, el valor D2 es más representativo del retardo ya que tiene en cuenta los retardos debido a las llamadas que se retardan. El valor de D1 se calcula con la contribución tanto de llamadas retrasadas como no retrasadas.

7.18.

Determinar el número de circuitos necesarios para tener un número medio de una llamada en cola sabiendo que se cursan 72 CCS en 36 llamadas (PC).

SOLUCIÓN

Procediendo del mismo modo que en el problema anterior (Problema 7.17), se tiene que sobre la Tabla 7.10, Erlang C para 2 Erlangs, ya que se corresponde con el tráfico ofrecido ($72 \text{ CCS} / 36 = 2 \text{ Erlangs}$).

N	P	D1	D2	Q1	Q2	P8	P4	P2	P1	PP
3	.444	.44	1.0	.89	2.0	.39	.35	.27	.16	.16
4	.1739	.09	.50	.17	1.0	.14	.11	.06	.02	.00
5	.0597	.02	.33	.04	.67	.04	.03	.01	.00	.00

Tabla 7.10. Extracto de la tabla de Erlang C para 2 Erlangs.

En esta Tabla 7.10 se muestra que para tener en media una llamada en cola y considerando el parámetro Q1 se necesitan tres circuitos. Así se tendrían del orden de 0,89 llamadas en colas, menor que el objetivo que teníamos planteado. Por otra parte, si consideramos el parámetro Q2 de la tabla, vemos que son necesarios cuatro circuitos para poder tener, en promedio, una llamada en cola. Menor número de circuitos nos van a aumentar las llamadas en cola al doble (considerando el parámetro Q2).

En este caso ocurre como en el problema anterior (Problema 7.17), que no se ha especificado claramente cuál es el valor medio que queremos satisfacer: el número medio de llamadas en cola (1) en cualquier instante de tiempo, o (2) en los instantes en que el sistema está ocupado. Al igual que en el caso anterior, el valor que mejor refleja la situación de las llamadas en cola es el del parámetro Q2 (Tabla 7.10).

7.19.

Dado un sistema donde se encolan las llamadas y en el que se cursa un tráfico de 126 CCS y 105 llamadas (PC), determinar el grado de servicio para el sistema de modo que el retardo medio de las llamadas retrasadas sea 45 s o menos.

SOLUCIÓN

En este caso, el tráfico ofrecido es igual que el cursado. Así, se tiene que el tráfico ofrecido es de $126 \text{ CCS} / 36 = 3,5 \text{ Erlangs}$. Por otra parte, el tiempo medio de duración de la llamada (AHT) es, mirando en la tabla Erlang C para 3,5 Erlangs (Tabla 7.11), de $126 \text{ CCS} / 105 \text{ PC} = 1,2 \text{ CCS} = 2 \text{ min}$.

Observando la Tabla 7.11 correspondiente, se puede ver que el valor que nos interesa no es 0,4 que resulta en un retardo medio para las llamadas con retraso de $0,4 \cdot 1,20 \text{ CCS} = 0,48 \text{ CCS} = 48 \text{ s}$.

En caso de elegir 0,29 se obtiene, siguiendo el mismo procedimiento (Tabla 7.11), un retardo de 0,348 CCS o bien 35 s aproximadamente, lo cual cumple las especificaciones. Con lo que se obtiene un grado de servicio de P.0762.

Otra manera de seleccionar más rápidamente el valor que nos interesa es realizar la siguiente operación:

$$\text{Factor de retardo} = \frac{\text{retardo global}}{\text{duración media de la llamada}}$$

Para realizar esta operación ponemos todos los elementos en las mismas unidades, así 45 s para el retardo global se corresponden con 0,45 CCS. Por lo tanto,

$$\begin{aligned}\text{Factor de retardo} &= \frac{\text{retardo total}}{\text{AHT}} = \\ &= \frac{0,45}{1,2} = 0,375\end{aligned}$$

Este valor nos permite acudir a la Tabla 7.11 y seleccionar el valor más fácilmente. El resultado es el mismo que el comentado anteriormente. Con esta operación previa lo que se consigue es determinar rápidamente la posición sobre la tabla.

N	P	D1	D2	Q1	Q2	P8	P4	P2	P1	PP
5	.3778	.25	.67	.88	2.33	.31	.26	.18	.08	.02
6	.1775	.07	.40	.25	1.40	.13	.09	.05	.01	.00
7	.0762	.02	.29	.08	1.00	.05	.03	.01	.00	.00

Tabla 7.11. Extracto de la tabla de Erlang C para 3,5 Erlangs.

7.20.

Dado un sistema donde se encolan las llamadas por tiempo indefinido que cursa un tráfico de 72 CCS y 36 llamadas (PC), y que dispone de cinco circuitos, calcular:

- ¿Cuál es la probabilidad de que ocurran retrasos?
- ¿Cuál es el retardo medio de las llamadas?
- ¿Cuál es el retardo medio de las llamadas que se retrasan?
- ¿Cuál es el número medio de llamadas en cola?
- ¿Cuánto tráfico se desborda?
- ¿Cuánto tráfico cursa el circuito tres?

SOLUCIÓN

En este caso, el tráfico cursado es igual que el ofrecido y resulta ser de valor $72 \text{ CCS} / 36 = 2$ Erlangs. Así, la tabla que tenemos que usar es la de Erlang C para 2 Erlangs, y para responder a las preguntas no nos vale la tabla que se utilizaba anteriormente (Problema 7.19). La Tabla 7.12 es la que se debe utilizar.

Las horas en el primer intento se corresponden con el tráfico ofrecido. Las horas conectado equivalen al tráfico cursado. Ambas son iguales porque consideramos almacenamiento de llamadas

Horas primer intento	Horas conectado	Horas overflow	Retardo	Factor de longitud de retardo por llamada	Líneas	Horas cursadas por circuito					
						1	2	3	4	5	6
2.00	2.00	.00	.444	1.000	3	.889	.667	.444			
2.00	2.00	.00	.174	.500	4	.870	.609	.348	.174		
<u>2.00</u>	<u>2.00</u>	<u>.00</u>	<u>.060</u>	<u>.333</u>	<u>5</u>	<u>.866</u>	<u>.597</u>	<u>.326</u>	<u>.149</u>	<u>.060</u>	
2.00	2.00	.00	.018	.250	6	.865	.594	.324	.144	.054	.018
2.00	2.00	.00	.005	.200	7	.865	.594	.324	.143	.053	.017
2.00	2.00	.00	.005	.200	8	.865	.594	.324	.143	.053	.017

Tabla 7.12. Extracto de la tabla de Erlang C para 2 Erlangs.

bloqueadas. Por tanto, el desbordamiento o bloqueo (horas overflow) es nulo, ya que todo el tráfico ofrecido se va a cursar. El retardo indica la probabilidad de que se retarden las llamadas y la longitud de retardo marca el retardo medio de las llamadas retrasadas. Las líneas indican los circuitos del sistema. Finalmente, las horas cursadas por circuito indican la cantidad de tráfico cursada por cada línea comenzando en Erlangs.

Teniendo en cuenta la Tabla 7.12, vamos a responder a las preguntas del enunciado:

- a) ¿Cuál es la probabilidad de que ocurran retrasos?

Como indica la fila para cinco circuitos (Tabla 7.12), este valor es 0,06 o 6%.

- b) ¿Cuál es el retardo medio de las llamadas?

La Tabla 7.12 no proporciona esta información, necesitamos el valor del parámetro D1.

- c) ¿Cuál es el retardo medio de las llamadas que se retrasan?

En la columna del factor de la longitud del retardo (Tabla 7.12) se ve que se tiene el valor de 0,333. Aplicándolo al valor de AHT resulta un valor de 0,666 CCS, o bien 67 s aproximadamente.

- d) ¿Cuál es el número medio de llamadas en cola?

La Tabla 7.12 no proporciona esta información, necesitamos el valor del parámetro Q1.

- e) ¿Cuánto tráfico se desborda?

0 Erlangs, ya que la aproximación de Erlang C supone que las llamadas esperan indefinidamente. Es decir, todo el tráfico ofrecido se cursa. Por tanto, no se desborda nada de tráfico.

- f) ¿Cuánto tráfico cursa el circuito tres?

Mirando en la Tabla 7.12 se ve rápidamente que el valor es de 0,326 Erlangs.

7.21.

Dado un sistema donde se encolan las llamadas durante un minuto como máximo, que cursa un tráfico de 72 CCS y 36 llamadas (PC), y que dispone de cuatro circuitos, calcular:

- a) ¿Cuál es la probabilidad de que ocurran retrasos?

- b) ¿Qué cantidad de tráfico se cursa?
- c) ¿Qué cantidad de tráfico se desborda?
- d) ¿Cuál es el retardo medio de las llamadas retrasadas?
- e) ¿Cuántas llamadas abandonarán la cola?

SOLUCIÓN

En este caso, se tiene un sistema que almacena las llamadas bloqueadas pero no indefinidamente, sino en una cola finita. Aquí, las llamadas esperan como máximo 1 minuto en la cola. Por otro lado, el tráfico cursado es de $72 \text{ CCS} / 36 = 2 \text{ Erlangs}$. Por tanto, tenemos que utilizar la tabla de Erlang C para un tráfico cursado de 2 Erlangs y cola finita de 1 minuto. Esta tabla (Tabla 7.13) se presenta a continuación:

Horas primer intento	Horas conectado	Horas overflow	Retardo	Factor de longitud de retardo por llamada	Líneas	Horas cursadas por circuito					
						1	2	3	4	5	6
2.00	1.704	.296	.271	.196	3	.695	.576	.433			
<u>2.00</u>	<u>1.833</u>	<u>.117</u>	<u>.111</u>	<u>.084</u>	<u>4</u>	<u>.678</u>	<u>.551</u>	<u>.401</u>	<u>.253</u>		
2.00	1.959	.041	.040	.030	5	.671	.540	.387	.238	.123	

Tabla 7.13. Extracto de la tabla de Erlang C para cola finita (1 min) y 2 Erlangs.

- a) ¿Cuál es la probabilidad de que ocurran retrasos?

El valor en la columna de retardo (Tabla 7.13) indica el porcentaje de llamadas retrasadas o la probabilidad de retardo. En este caso es del 11%.

- b) ¿Qué cantidad de tráfico se cursa?

En este caso, el valor que indica la Tabla 7.13 es de 1.833 Erlangs, es menor que 2 (el tráfico ofrecido) debido a que ahora algunas llamadas pueden abandonar la cola sin haber sido cursadas.

- c) ¿Qué cantidad de tráfico se desborda?

Se indica en la Tabla 7.13 sobre la columna de overflow y resulta 0,117 Erlangs. Notar que de la suma del tráfico desbordado y del tráfico cursado resulta el tráfico ofrecido.

- d) ¿Cuál es el retardo medio de las llamadas retrasadas?

El factor de longitud de retardo es en este caso 0,084 (Tabla 7.13). Multiplicando por el valor del tiempo medio de duración de la llamada ($AHT = 72 \text{ CCS} / 36 \text{ PC} = 2 \text{ CCS}$) se tiene que $2 \text{ CCS} \cdot 0,084 = 0,168 \text{ CCS} = 16,8 \text{ s}$. Por tanto, el resultado es aproximadamente 17 s.

- e) ¿Cuántas llamadas abandonarán la cola?

El tráfico de desbordamiento es del orden de 0,117 Erlangs o 4,212 CCS (Tabla 7.13). Si dividimos el tiempo medio de duración de las llamadas tendremos el número de llamadas: $4,212 / 2 \text{ CCS}$, aproximadamente 2 llamadas.

7.22.

Dado un sistema donde se encolan las llamadas durante 10 minutos como máximo, que cursa un tráfico de 72 CCS y 36 llamadas (PC), y que dispone de cuatro circuitos, calcular:

- ¿Cuál es la probabilidad de que ocurran retrasos?
- ¿Qué cantidad de tráfico se cursa?
- ¿Qué cantidad de tráfico se desborda?
- ¿Cuál es el retardo medio de las llamadas retrasadas?
- ¿Cuántas llamadas abandonarán la cola?

SOLUCIÓN

En este caso se tiene un sistema que almacena las llamadas bloqueadas pero no indefinidamente, sino en una cola finita. Las llamadas esperan como máximo 10 minutos en la cola. Por otro lado, el tráfico cursado es de $72 \text{ CCS} / 36 = 2$ Erlangs. Por tanto, tenemos que utilizar la tabla de Erlang C para un tráfico cursado de 2 Erlangs y cola finita de 10 minutos (Tabla 7.14). Esta tabla se presenta a continuación:

Horas primer intento	Horas conectado	Horas overflow	Retardo	Factor de longitud de retardo por llamada	Líneas	Horas cursadas por circuito					
						1	2	3	4	5	6
2.00	1.754	.246	.886	1.221	2	.888	.866				
2.00	1.959	.041	.359	.541	3	.749	.661	.548			
<u>2.00</u>	<u>1.989</u>	<u>.011</u>	<u>.128</u>	<u>.195</u>	<u>4</u>	<u>.694</u>	<u>.576</u>	<u>.433</u>	<u>.286</u>		
2.00	1.997	.003	.042	.065	5	.676	.547	.396	.248	.130	

Tabla 7.14. Extracto de la tabla de Erlang C para cola finita (10 min) y 2 Erlangs.

- ¿Cuál es la probabilidad de que ocurran retrasos?

El valor en la columna de retardo (Tabla 7.14) indica el porcentaje de llamadas retrasadas o la probabilidad de retardo. En este caso, es del 12,8%. Este valor es mayor que en el problema anterior (Problema 7.21) debido a que ahora la cola tiene más probabilidad de estar ocupada por llamadas que esperan a ser atendidas.

- ¿Qué cantidad de tráfico se cursa?

En este caso, el valor que indica la Tabla 7.14 es de 1.989 Erlangs. Este valor es mayor que en el problema anterior (Problema 7.21) debido a que ahora algunas esperan durante tiempo suficiente para poder ser cursadas.

- ¿Qué cantidad de tráfico se desborda?

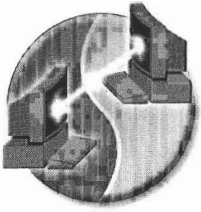
Se indica en la Tabla 7.14 sobre la columna de overflow y resulta 0,011 Erlangs. Este valor representa la cantidad de tráfico que va a abandonar la cola.

- d) ¿Cuál es el retardo medio de las llamadas retrasadas?

El factor de longitud de retardo (Tabla 7.14) es en este caso 0,195. Multiplicando por el valor del tiempo medio de duración de la llamada ($AHT = 72 \text{ CCS} / 36 \text{ PC} = 2 \text{ CCS}$) se tiene que $2 \text{ CCS} \cdot 0,195 = 0,39 \text{ CCS} = 39 \text{ s}$.

- e) ¿Cuántas llamadas abandonarán la cola?

El tráfico de desbordamiento es del orden de 0,011 Erlangs o 0,396 CCS (Tabla 7.14). Si le dividimos el tiempo medio de duración de las llamadas tendremos el número de llamadas: $0,396 / 2 \text{ CCS} = 0,198$ llamadas.



CONMUTACIÓN DE PAQUETES

8.1.

Dos usuarios (A y B) se comunican a través de una red y entre ellos hay tres saltos. Evaluar el tiempo que se tarda en transmitir un fichero de T bytes en caso de utilizar conmutación de paquetes, conmutación de circuitos y circuitos virtuales. Considerar que los paquetes disponen de una cabecera de C bytes siendo su tamaño máximo de M bytes. Por otra parte, al establecer un circuito se requiere el envío de un mensaje de SETUP, cuyo tamaño es S bytes y el cual se confirma con un mensaje ACK, de un tamaño C bytes, para indicar la disponibilidad del enlace. Además que considera nula la probabilidad de bloqueo en el establecimiento del circuito. La longitud de cada enlace es de l_1, l_2, l_3 y l_4 metros y su capacidad es de valor c_1, c_2, c_3 y c_4 .

SOLUCIÓN

A la hora de determinar el tiempo total necesario para realizar la transmisión del fichero hay que tener en cuenta el tipo de conmutación que se utiliza para transmitir la información.

Conmutación de circuitos: Vamos a empezar considerando el caso de la transmisión por conmutación de circuitos. En tal caso, la transmisión tiene tres partes claramente diferenciadas: establecimiento del circuito, transmisión de los datos y finalización del circuito. El tiempo necesario para disponer en B de los datos transmitidos desde A es el necesario para establecer el circuito ($t_{\text{est} - \text{CC}}$) más el tiempo de transmisión del fichero sobre el circuito establecido (t_{fich}). El tiempo de finalización del circuito va a afectar al tiempo en el que estarán disponibles los recursos en la red para establecer un nuevo circuito. Por tanto, el tiempo utilizado en conmutación de circuitos viene dado por la siguiente expresión:

$$t_{\text{CC}} = t_{\text{est} - \text{CC}} + t_{\text{fich}}$$

- a) *Establecimiento del circuito:* El tiempo requerido para el establecimiento de la conexión suma los tiempos necesarios para transmitir los paquetes de establecimiento del circuito. Considerando que se utiliza un mensaje de SETUP más otro de ACK para aceptar la conexión, se tiene como se muestra en la siguiente figura, un mensaje de SETUP desde el origen al destino más el mensaje de respuesta (ACK) de aceptación del circuito. La transmisión de cada mensaje supone un tiempo total de $t_{\text{msj}} = t_{\text{prop}} + t_{\text{trx} - \text{msj}} + t_{\text{proc}}$, donde:

$$t_{\text{prop}} = \frac{\text{longitud del enlace}}{\text{velocidad de transmisión}}$$

Durante el establecimiento de los circuitos se utilizan dos tipos de mensajes,

$$t_{\text{trx} - \text{setup}} = t_{\text{SETUP}} = \frac{\text{tamaño del mensaje SETUP (bytes)}}{\text{capacidad del enlace}}$$

$$t_{\text{trx} - \text{ACK}} = t_{\text{ACK}} = \frac{\text{tamaño del mensaje ACK (bytes)}}{\text{capacidad del enlace}}$$

Prop representa el tiempo de procesamiento de los mensajes que se considera despreciable en este caso. Por tanto, el establecimiento del circuito supone un tiempo total de:

$$\begin{aligned} t_{\text{est-CC}} &= t_{\text{mss} - 1} + t_{\text{mss} - 2} + t_{\text{mss} - 3} + t_{\text{mss} - 4} = 2 \cdot (t_{\text{prop} - 1} + t_{\text{SETUP}} + t_{\text{ACK}}) + 2 \cdot (t_{\text{prop} - 2} + \\ &+ t_{\text{SETUP}} + t_{\text{ACK}}) + 2 \cdot (t_{\text{prop} - 3} + t_{\text{SETUP}} + t_{\text{ACK}}) + 2 \cdot (t_{\text{prop} - 4} + t_{\text{SETUP}} + t_{\text{ACK}}) = \\ &= \sum_{i=1}^4 \left(2 \cdot \frac{l_i}{c} + \frac{S}{c_i} + \frac{C}{c_i} \right) \end{aligned}$$

- b) *Transmisión de la información:* Una vez finalizado el establecimiento del circuito se transmiten los datos a través de él, a los cuales no es necesario añadir datos adicionales (cabecera). El tiempo que se tarda en transmitir el fichero viene determinado por el tiempo de propagación en recibir el primer bit más el tiempo que se tarda en recibir el fichero:

$$t_{\text{fich}} = t_{\text{prop} - (1,2,3,4)} + \frac{\text{tamaño del fichero (bytes)}}{\text{capacidad del enlace (bytes/s)}}$$

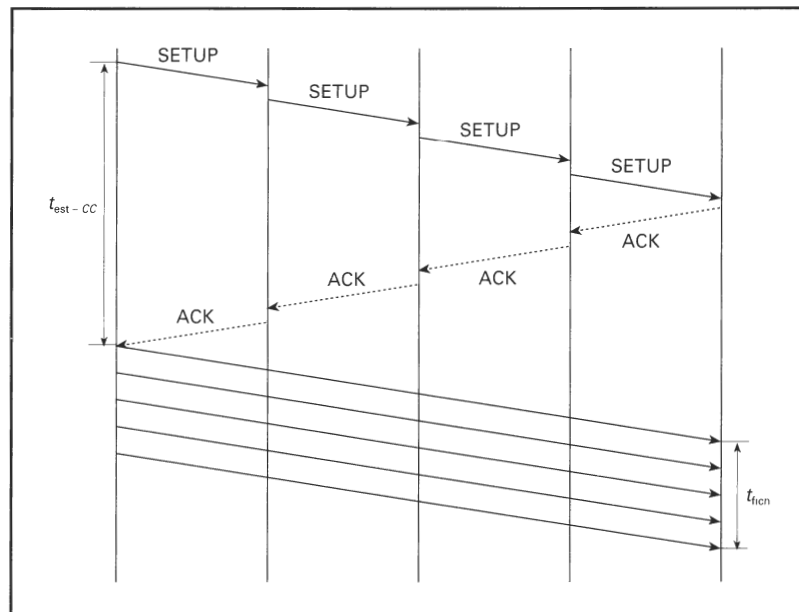


Figura 8.1. Transmisión de fichero por conmutación de circuitos.

Conmutación de paquetes: En este caso los datos se transmiten agrupados en paquetes. Por un lado hay que determinar el número de paquetes que se van a crear para transmitir el total del fichero. Estos paquetes van a circular a través de los enlaces que unen los elementos

intermedios. Por tanto, cada uno de los paquetes va a realizar cuatro saltos. El tiempo total requerido para transferir por completo el fichero será el necesario para transmitir todos los paquetes a lo largo de los cuatro enlaces que separan a los usuarios. Se asume en este caso que la red está funcionando en baja carga y por tanto es despreciable el tiempo de espera en cola en cada uno de los saltos.

Vamos a calcular el número de paquetes necesarios para transmitir el fichero completo. Cada paquete tiene capacidad para almacenar un máximo de:

$$D = M \text{ bytes (tamaño máximo del paquete)} - C \text{ bytes (cabecera)}$$

ya que la cabecera es absolutamente necesaria. En tal caso, el número de mensajes total para transferir el fichero es:

$$n = \left\lfloor \frac{T}{D} \right\rfloor$$

que representa el mayor entero menor o igual que el cociente entre T y D . Si el fichero no se puede agrupar en un número entero de paquetes de tamaño máximo, entonces el último de los paquetes tendrá un tamaño menor y requerirá un tiempo menor de transmisión.

Los paquetes de tamaño M bytes requieren un tiempo de $t_M = t_{\text{prop}} + t_{\text{trx}-M}$ para ser transmitidos en un salto. El paquete final será de un tamaño $m = C + (T - n \cdot D)$, y requiere un tiempo $t_m = t_{\text{prop}} + t_{\text{trx}-m}$ para ser transmitido. El tiempo necesario para transmitir el fichero en un salto supone un tiempo de:

$$t_{\text{hop}-i} = (n-1) t_{M-i} + t_{m-i} = (n-1) \left[\frac{l_i}{c} + \frac{M}{c_i} \right] + \left[\frac{l_i}{c} + \frac{m}{c_i} \right],$$

siendo c la velocidad de transmisión en el medio físico.

Por tanto, el tiempo total para transferir el fichero utilizando conmutación de paquetes resulta ser:

$$t_{CP} = \sum_{i=1}^4 t_{\text{hop}-i}$$

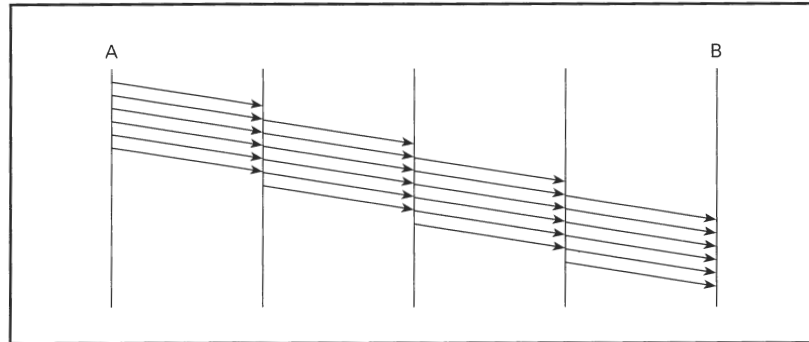


Figura 8.2. Transmisión por conmutación de paquetes.

Circuitos virtuales: El funcionamiento del sistema requiere que se establezca en primer lugar el circuito virtual de un modo similar al caso de la conmutación de circuitos. Sin embargo, la transmisión de la información se realiza en forma de paquetes. Es decir, se divide la información

en bloques a los que se añade una cabecera de modo similar a la conmutación de paquetes. La gran diferencia con respecto a la conmutación de paquetes es que, en este caso, los paquetes siguen el camino determinado por el circuito virtual establecido, mientras que en la conmutación de paquetes pueden seguir rutas diferentes hasta llegar a su destino.

En este caso, la transmisión de datos se diferencia de la conmutación de paquetes en que la conmutación es más rápida en cada salto, y la conexión tiene reservados unos recursos, por tanto tiene limitado el retardo total de la transmisión. La diferencia es que requiere el establecimiento del circuito. Por tanto, el tiempo de transmisión será la suma del tiempo de establecimiento de un circuito (t_{SETUP}) más el tiempo de transmisión de los paquetes (t_{CP}). En total, $t_{CV} = t_{\text{est-CC}} + t_{CP}$.

8.2.

En este problema vamos a centrarnos en la arquitectura interna de cada uno de los saltos. La Figura 8.3 muestra la arquitectura de cada salto. Cada elemento dispone de una memoria de entrada, memoria de salida y un elemento de conmutación que se encarga de la asociación entrada/salida en función de un algoritmo de encaminamiento. Para analizar las prestaciones del router se requiere conocer dos medidas: el retardo de encaminamiento y la latencia del control de flujo.

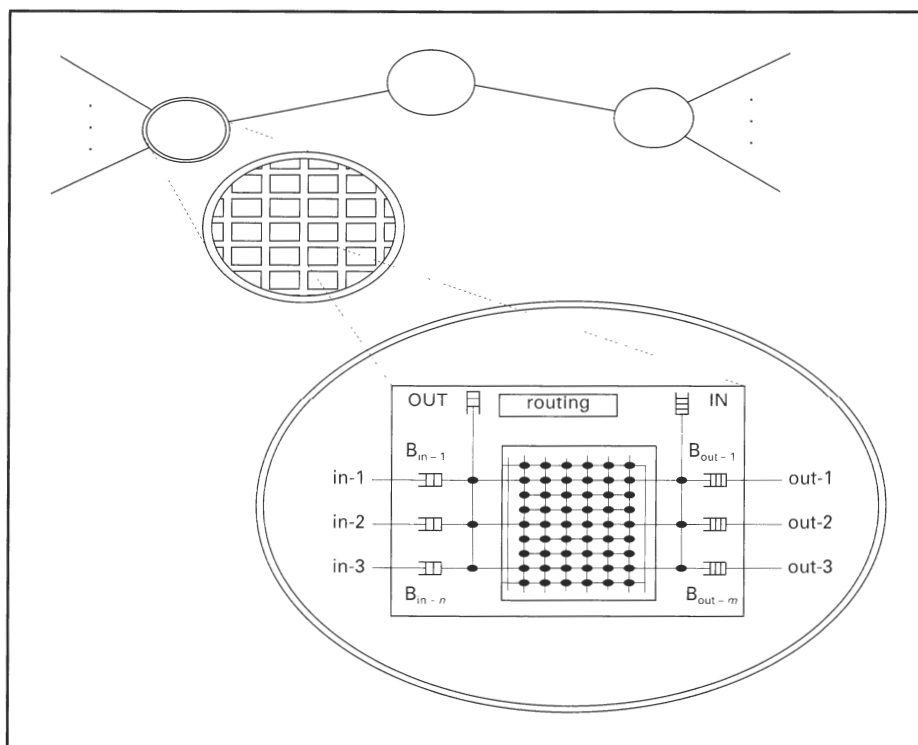


Figura 8.3. Arquitectura de un nodo de comunicaciones.

El retardo de encaminamiento se determina como el tiempo necesario para seleccionar el enlace de salida por el que se va a transmitir un mensaje que acaba de entrar en un procesador, una vez que se ha establecido el camino en él a través del elemento de conmutación interno, la tasa a la que se transmiten los mensajes internamente está determinada por el retardo de propagación a través del conmutador (retardo intrarouter) y la tasa de la sincronización para la transferencia interna entre los buffers de entrada y salida. Este retardo se llama latencia del control de flujo interno. Además, al retardo hay que añadir el retardo por la transmisión de los mensajes entre los procesadores (retardo interrouter). El retardo de encaminamiento

más la latencia del control de flujo determinan la latencia de los mensajes a través del conmutador y, junto con el retardo de adquisición de los enlaces físicos, van a determinar el throughput de cada salto.

En este problema se trata de dibujar esquemáticamente el diagrama de bloques del sistema de conmutación y determinar los tiempos que intervienen a la hora de transmitir un mensaje a través de ellos, tanto para el caso de que los mensajes se transmitan según la conmutación de paquetes como de circuitos.

SOLUCIÓN

Los tiempos definidos en el enunciado del problema se describen en la Figura 8.4. Cada salto está formado por un conjunto de procesadores que toman decisiones de encaminamiento para llevar el pq de la entrada a la salida del salto. El procesador toma la decisión de encaminar en t_r unidades de tiempo. El canal físico entre dos routers opera a BW bits por segundo. El retardo de propagación del enlace se denota por $t_w = 1 / BW$. Una vez que se ha establecido el camino a través de un procesador, el retardo intrarrouter o retardo de conmutación se denota por t_s .

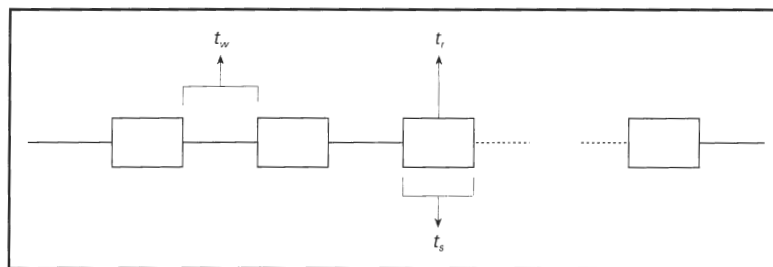


Figura 8.4. Tiempos de conmutación interna

8.3.

Determinar el tiempo requerido para transmitir un paquete en cada uno de los nodos de conmutación para transmitir un paquete.

SOLUCIÓN

Los nodos de conmutación de paquetes pueden actuar de diferente modo para realizar la conmutación del paquete. Algunas alternativas posibles para realizar el procesamiento interno de los paquetes en los nodos de comunicaciones son la conmutación de paquetes, de circuitos y la conmutación cut-through.

Normalmente, las redes de interconexión de los nodos de comunicación no trabajan con el paquete completo, sino que lo dividen en trozos mas pequeños (flips) que son los que se transfieren a nivel físico. Por tanto, al transmitir un paquete entre el buffer de entrada al de salida, éste se transmite a trozos. Estos elementos requieren a su vez, en algunos casos, añadirles una cabecera para determinar el buffer de salida que tienen como destino.

A continuación se va a determinar la latencia de la red de interconexión para cada una de las posibles formas de conmutar los paquetes desde los buffers de entrada a los de salida.

En el caso de conmutación de paquetes, el paquete se recibe en una cola asociada a un enlace de entrada si hay espacio disponible para él. A partir de la información contenida en la cabecera se ejecuta localmente la función de encaminamiento para determinar el enlace de salida para el paquete. A través del elemento de conmutación propiamente dicho (crossbar) se realiza la transmisión del paquete a la cola de salida correspondiente para ese paquete. Una vez en ella, el paquete se transferirá a su destino cuando haya espacio para él. Esto se muestra en la Figura 8.5, donde se indica cómo los diferentes enlaces de la red interconexión están ocupados en diferentes momentos. En concreto,

se muestra en esta figura que el paquete se transfiere en varios flips, cada uno con su cabecera, lo cual va a permitir realizar la conmutación interna. Esto hace que en cada momento uno de los enlaces esté ocupado transmitiendo la información.

Nota: Hay que tener claro que estas cabeceras no se corresponden con las cabeceras de los paquetes que se transmiten. Son necesarias para la conmutación interna. En este caso, el tiempo medio de transmisión de cada mensaje se calcula como:

$$t_{\text{paquete}} = D \left\{ t_r + (t_s + t_w) \left\lceil \frac{L + W}{W} \right\rceil \right\}$$

donde $\lceil x \rceil$ denota el mayor entero menor o igual que x .

Este valor de t_{paquete} representa el tiempo necesario para transferir un paquete de longitud $L + W$ bits a lo largo de un canal (t_w) desde el buffer de entrada al buffer de salida del router (t_s). Esta expresión asume que existen buffers a la entrada y a la salida del router. Es importante destacar que el retardo es proporcional a la distancia entre la entrada y la salida.

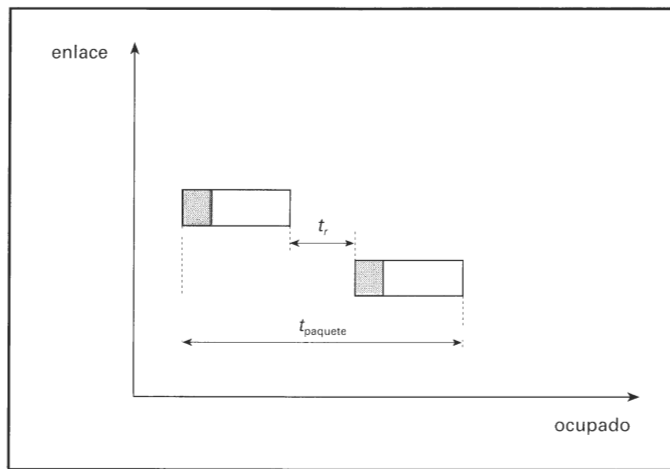


Figura 8.5. Diagrama temporal de ocupación de enlaces para la conmutación de paquetes.

En el caso de **conmutación de circuitos**, el paquete se encuentra en la cola del enlace de entrada. Entonces, se toma la decisión de encaminamiento y establece un circuito entre la cola donde se almacena el paquete y la cola correspondiente al enlace de salida para dicho paquete. En la Figura 8.6 se muestra una primera parte en la que se establece el canal de comunicaciones. Para ello se transmiten los mensajes de SETUP que se confirman con los ACKs. Cada mensaje va ocupando uno de los enlaces. Una vez establecido el camino se transfiere el paquete (Figura 8.6). En este caso, se ocupan varios de los enlaces de la red de interconexión durante la transferencia del mensaje. El tiempo medio de conmutación se determina mediante la siguiente expresión:

$$t_{cc} = t_{\text{setup}} + t_{\text{trx}}$$

$$t_{\text{setup}} = D [t_r + 2 \cdot (t_s + t_w)]$$

$$t_{\text{trx}} = 1/B \cdot \left\lceil \frac{1}{W} \right\rceil$$

En el caso de la **conmutación cut-through**, propuesta por M. Schwartz, se tiene que el paquete según llega a la cola de entrada toma la decisión de encaminamiento y se transmite a la cola de salida.

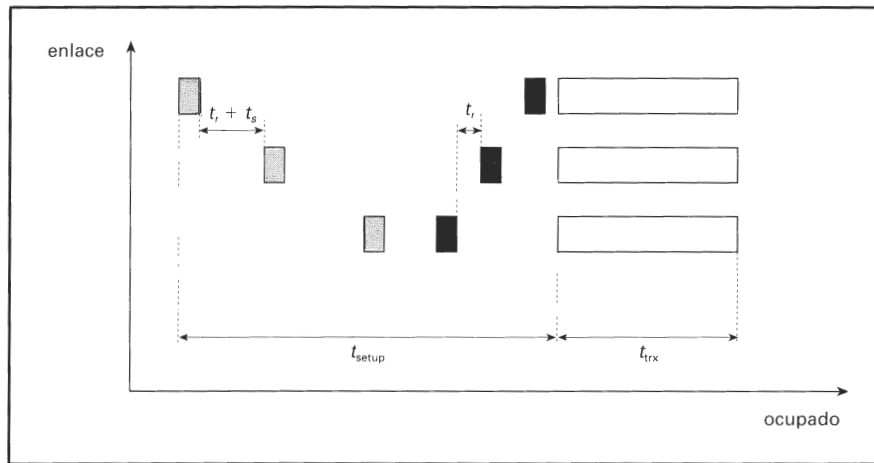


Figura 8.6. Diagrama temporal de ocupación de enlaces en conmutación de circuitos.

Es decir, no se espera a que sea recibido completamente el paquete para ser conmutado al enlace de salida. De este modo se puede llegar a tener, dependiendo del tamaño de paquete admitido, un paquete ocupando varios elementos de interconexión (Figura 8.7). En este caso, el tiempo de conmutación del paquete se expresa del siguiente modo:

$$t_{VCT} = D \cdot (t_r + t_s + t_w) + \max(t_s, t_w) \cdot \left\lceil \frac{L}{W} \right\rceil$$

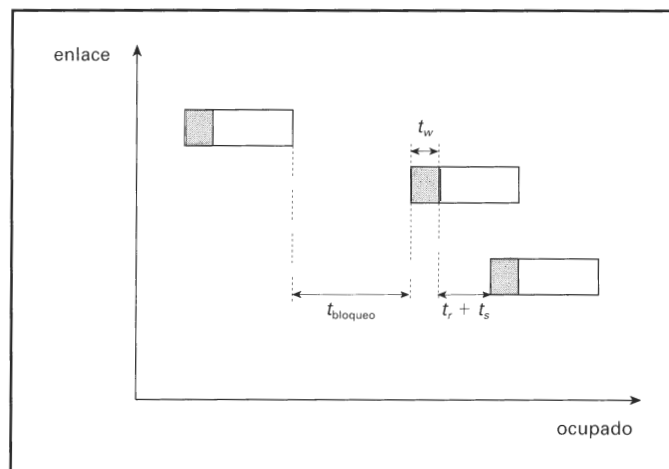


Figura 8.7. Diagrama temporal de ocupación de enlaces para conmutación cut-through.

En la Figura 8.7 se muestra cómo durante la transmisión puede ocurrir que el paquete se bloquee por no existir recursos disponibles para continuar con la transmisión del paquete. Y a su vez, se muestra cómo dos enlaces pueden estar transmitiendo simultáneamente datos de un mismo paquete, con la ventaja que ello supone.

Considerar un router con una entrada y una salida, a través del cual se transmiten mensajes de tamaño fijo. El router tiene un buffer tanto para la entrada como para la salida con capacidad para un sólo mensaje. Los mensajes se transmiten según un protocolo ARQ, con el que se manda un ACK al origen del mensaje cada vez que se almacena en el buffer de entrada del router destino. En caso contrario se manda un NACK y se reintentará enviar el paquete.

En esta situación, buscar una sucesión de estados para ilustrar la formación del interbloqueo. La situación de partida se ilustra en la Figura 8.8:

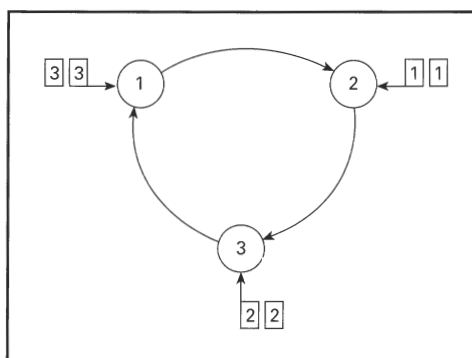


Figura 8.8. Estudio del interbloqueo de comunicaciones.

SOLUCIÓN

La Figura 8.8 muestra tres saltos desde los que se quieren transmitir dos mensajes después de dos saltos en la red. El estado del sistema lo definimos como el valor de las variables del sistema. Es decir, en este caso está determinado por el contenido de los buffers de cada uno de los saltos. Consideramos como variables las definidas en el Problema 8.2 (Figura 8.3) donde se describe la estructura de un router. Estas variables van a ser IN^i , OUT^i , B_{in}^i y B_{out}^i de cada uno de los nodos ($i = 1 \dots 3$).

A continuación vamos a presentar una secuencia posible de estados del sistema que van a derivar en una situación de interbloqueo en las comunicaciones.

En el estado inicial (s_0) se tiene que los buffers están vacíos, salvo el IN de cada nodo que contiene los mensajes a transmitir. Esta situación se ilustra en la Tabla 8.1.

Estado	Nodo	IN	OUT	B_{in}	B_{out}
S_0	1	$\langle 3, 3 \rangle$	—	—	—
	2	$\langle 1, 1 \rangle$	—	—	—
	3	$\langle 2, 2 \rangle$	—	—	—

Tabla 8.1. Estado inicial del sistema.

A partir de este estado ocurre que se manda un paquete a cada nodo siguiente. Estamos considerando una red asíncrona, pero sin pérdida de generalidad vamos a considerar un comportamiento particular para ésta, que es un comportamiento síncrono. Es decir, vamos a considerar que los nodos transmiten al mismo tiempo los mensajes. Por tanto, en este caso, se tiene que en el siguiente estado, tras la transmisión de un mensaje, se alcanza el estado de la Tabla 8.2.

A partir de este estado cada nodo, internamente, conmuta su mensaje al buffer del enlace de salida y con ello se llega al estado de la Tabla 8.3.

Estado	Nodo	IN	OUT	B_{in}	B_{out}
S_1	1	<3>	–	<3>	–
	2	<1>	–	<1>	–
	3	<2>	–	<2>	–

Tabla 8.2. Cada nodo ha transmitido un mensaje a su vecino.

Estado	Nodo	IN	OUT	B_{in}	B_{out}
S_2	1	<3>	–	–	<3>
	2	<1>	–	–	<1>
	3	<2>	–	–	<2>

Tabla 8.3. Transmisión del mensaje.

Estado	Nodo	IN	OUT	B_{in}	B_{out}
S_2	1	–	–	<3>	<3>
	2	–	–	<1>	<1>
	3	–	–	<2>	<2>

Tabla 8.4. Estado de interbloqueo.

A partir de este estado, los nodos pueden hacer dos cosas. O bien se transmite el mensaje al siguiente nodo, o bien se almacena en el buffer de entrada el mensaje inyectado desde el propio nodo. Vamos a considerar que el nodo almacena en el buffer de entrada el mensaje inyectado en el nodo. El estado alcanzado sería el de la Tabla 8.4.

En el estado final alcanzado, el sistema se encuentra en una situación en la que cada mensaje en un buffer de entrada quiere alcanzar un buffer de salida, pero no puede hacerlo porque no hay espacio para su almacenamiento. Por otra parte, al no poder vaciarse el buffer asociado al enlace de entrada de cada nodo, se tiene que los paquetes en la cola del enlace de salida de cada nodo tampoco pueda avanzar. Es decir, se ha llegado a una situación de bloqueo permanente de todos los nodos del sistema: el sistema está interbloqueado.

Para resolver esta situación las alternativas son las de utilizar algoritmos de encaminamiento que eviten el interbloqueo o bien algo tan simple como tirar alguno de los mensajes. El problema es determinar cuándo se produce dicha situación de interbloqueo.

8.5

Siguiendo el Algoritmo de Dijkstra, determinar la tabla de encaminamiento para el nodo 1 de la siguiente configuración de red.

SOLUCIÓN

El Algoritmo de Dijkstra se puede describir como sigue:

- N , número de nodos de la red.
- s , nodo fuente.
- M , conjunto de nodos incorporados por el algoritmo.

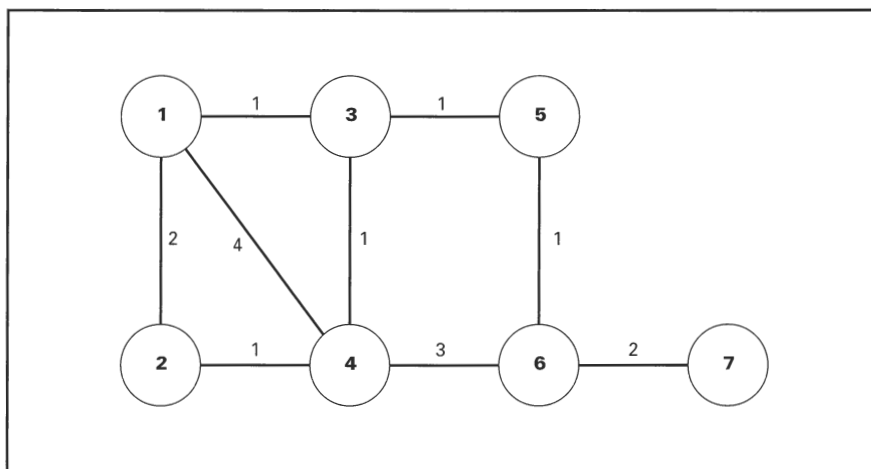


Figura 8.9. Ejemplo de configuración de red.

- $d_{i,j}$, costo del enlace entre los nodos i y j ; $d_{i,i} = 0$ y $d_{i,j} = \infty$ si los dos nodos no se encuentran conectados directamente; $d_{i,j} \geq 0$ si los dos nodos se encuentran conectados directamente.
- D_n , costo en curso obtenido por el algoritmo para el camino más corto desde el nodo s al nodo n .

El algoritmo consta de tres pasos, repitiéndose los pasos 2 y 3 hasta que el cardinal de M es igual a N .

Paso 1. Iniciación

$$M = \{s\}$$

$$D_n = d_{s,n} \text{ siendo } s \neq n$$

Paso 2. Encontramos el nodo vecino que no esté en M con el camino de menor coste desde el nodo s y lo incorporamos a M , lo que puede expresarse como:

Encontrar $w \notin M$ tal que $D_w = \min (D_j, j \notin M)$, entonces añadir w a M

Paso 3. Actualizamos los caminos de mínimo coste:

$$D_n = \min [D_n, D_w + d_{w,n}] \text{ para todo } n \notin M$$

Aplicamos el algoritmo descrito a continuación, partiendo de la iniciación del algoritmo.

(s0) El algoritmo comienza estableciendo el valor inicial de las variables.

Elementos del sistema:

$$M = \{1\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	–
1	3	3
1	4	–
1	5	–
1	6	–
1	7	–

Cálculo de costes:

En el estado inicial se calcula el coste con los nodos alcanzables directamente.

$$D(2) = 2, D(3) = 1, D(4) = 4, D(5) = \infty, D(6) = \infty, D(7) = \infty$$

(s1) En el siguiente estado se selecciona uno de los nodos a los que suponga un menor coste llegar a ellos. En este caso, el nodo 3 satisface dicha condición. Así que, en el siguiente estado se tiene que:

$$M = \{1, 3\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	—
1	3	3
1	4	—
1	5	—
1	6	—
1	7	—

La tabla de encaminamiento se actualiza indicando que para llegar al nodo 3 desde 1, esto se hace directamente.

Cálculo de costes:

$$D(2) = \min(2, D(3) + d_{3,2}) = \min(2, 1 + \infty) = 2$$

$$D(4) = \min(4, D(3) + d_{3,4}) = \min(4, 1 + 1) = 2$$

$$D(5) = \min(\infty, D(3) + d_{3,5}) = \min(\infty, 1 + 1) = 2$$

$$D(6) = \infty, D(7) = \infty$$

En este caso, tres nodos tienen un coste igual. Es decir, al nodo 1 le supone un mismo coste alcanzar a los nodos 2, 4 y 5. Por tanto, se selecciona uno de ellos aleatoriamente.

(s2) En este estado se añade el nodo 2 al conjunto de nodos analizados y se evalúan los costes para el resto de nodos.

$$M = \{1, 2, 3\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	—
1	5	—
1	6	—
1	7	—

La tabla de encaminamiento se actualiza indicando que para llegar al nodo 2 desde 1, esto se hace directamente.

Cálculo de costes:

$$D(4) = \min(4, D(2) + d_{2,4}, D(3) + d_{3,4}) = \min(4, 2 + 1, 1 + 1) = 2$$

$$D(5) = \min(\infty, D(2) + d_{2,5}, D(3) + d_{3,5}) = \min(\infty, 2 + \infty, 1 + 1) = 2$$

$$D(6) = \infty, \quad D(7) = \infty$$

(s3) En este estado se añade el nodo 4 al conjunto de nodos analizados y se evalúan los costes para el resto de nodos.

$$M = \{1, 2, 3, 4\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	4
1	5	—
1	6	—
1	7	—

La tabla de encaminamiento se actualiza indicando que para llegar al nodo 4 desde 1 esto sea hace directamente.

Cálculo de costes:

$$D(5) = \min(\infty, D(2) + d_{2,5}, D(3) + d_{3,5}, D(4) + d_{4,5}) = \min(\infty, 2 + \infty, 1 + 1, 2 + \infty) = 2$$

$$D(6) = \min(\infty, D(2) + d_{2,6}, D(3) + d_{3,6}, D(4) + d_{4,6}) = \min(\infty, 2 + \infty, 1 + \infty, 2 + 3) = 5$$

$$D(7) = \min(\infty, D(2) + d_{2,7}, D(3) + d_{3,7}, D(4) + d_{4,7}) = \infty$$

(s4) En este estado se añade el nodo 5 al conjunto de nodos analizados y se evalúan los costes para el resto de nodos. Por otra parte, se tiene que el menor coste para alcanzar se realiza a través de un salto en el nodo 3, y la tabla de encaminamiento reflejará esta situación.

$$M = \{1, 2, 3, 4, 5\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	4
1	5	3
1	6	—
1	7	—

Cálculo de costes:

$$D(6) = \min(\infty, D(2) + d_{2,6}, D(3) + d_{3,6}, D(4) + d_{4,6}, D(5) + d_{5,6}) = \\ = \min(\infty, 2 + \infty, 1 + \infty, 2 + 3, 2 + 1) = 3$$

$$D(7) = \min(\infty, D(2) + d_{2,7}, D(3) + d_{3,7}, D(4) + d_{4,7}, D(5) + d_{5,7}) = \\ = \min(\infty, 2 + \infty, 1 + \infty, 2 + \infty, 2 + \infty) = \infty$$

(s5) Ahora se añade el nodo 6 al conjunto de nodos analizados y se evalúan los costes para el resto de nodos. Ahora el nodo 6 será alcanzado desde el nodo 5 y para llegar al nodo 5, el nodo que está creando la tabla de encaminamiento necesita mandar los mensajes al nodo 3 (como indica la tabla desde que el nodo 5 fue insertado), lo cual tiene que quedar reflejado en la tabla de encaminamiento.

$$M = \{1, 2, 3, 4, 5, 6\}$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	4
1	5	3
1	6	3
1	7	—

Cálculo de costes:

$$D(7) = \min(\infty, D(2) + d_{2,7}, D(3) + d_{3,7}, D(4) + d_{4,7}, D(5) + d_{5,7}, D(6) + d_{6,7}) = \\ = \min(\infty, 2 + \infty, 1 + \infty, 2 + \infty, 2 + \infty, 3 + 2) = 5$$

(s6) Finalmente, se inserta el nodo 7 y se actualiza la tabla de encaminamiento teniendo en cuenta que al nodo 7 se llega a través del nodo 6 (y al nodo 6 a través del nodo 5 y a éste por el nodo 3). El resultado del algoritmo se resume a continuación.

$$M = \{1, 2, 3, 4, 5, 6, 7\}$$

Costes:

D(2)	D(3)	D(4)	D(5)	D(6)	D(7)
2	1	2	2	3	5

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	4

Tabla de encaminamiento (continuación)		
Origen	Destino	Out
1	1	3
1	6	3
1	7	3

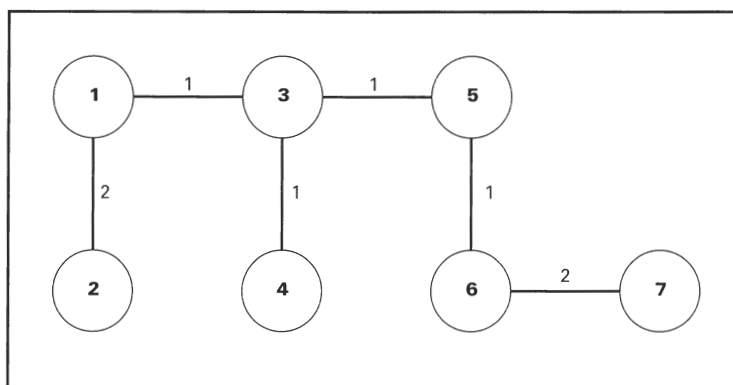


Figura 8.10. Árbol de alcanzabilidad resultante.

8.6.

Aplicar el Algoritmo Bellman-Ford para establecer la tabla de encaminamiento para el nodo 1 de la configuración de red que se muestra en la siguiente Figura 8.11.

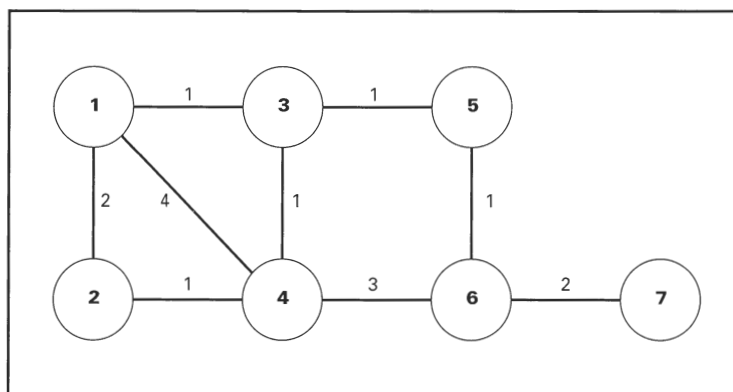


Figura 8.11. Ejemplo de configuración de red

SOLUCIÓN

El Algoritmo Bellman-Ford consigue encontrar los caminos más cortos desde un nodo fuente dado, con la condición de que éstos contengan como mucho un enlace; a continuación encontrar aquellos caminos más cortos con la condición de que contengan a lo sumo dos enlaces, y así sucesivamente. Este algoritmo se describe a continuación:

- s , nodo fuente.
- $d_{i,j}$, costo del enlace entre los nodo i y j ; $d_{i,i} = 0$ y $d_{i,j} = \infty$ si los dos nodos no se encuentran conectados directamente; $d_{i,j} \geq 0$ si los dos nodos se encuentran conectados directamente.
- h , número máximo de enlaces en un camino en el paso actual del algoritmo.

$D_n^{(h)}$, coste del camino de menor costo desde el nodo s hasta el nodo n con la condición de no más de h enlaces.

El algoritmo consta de los siguientes pasos, repitiéndose el Paso 2 hasta que ninguno de los costos cambia:

Paso 1. Iniciación

$$D_n^{(0)} = \infty, \text{ para todo } s \neq n.$$

$$D_s^{(0)} = 0, \text{ para todo } h.$$

Paso 2. h ($h \geq 0$)

$$D_n^{(h+1)} = \min [D_j^{(0)} + d_{j,i}], \text{ para todo } j$$

Vamos a aplicar el algoritmo a la configuración de red que se indica en el enunciado.

(s0) Inicialización, según indica el algoritmo,

$$D_1^{(0)} = 0$$

$$D_2^{(0)} = D_3^{(0)} = D_4^{(0)} = D_5^{(0)} = D_6^{(0)} = D_7^{(0)} = \infty$$

Tabla de encaminamiento		
Origen	Destino	Out
1	2	—
1	3	—
1	4	—
1	5	—
1	6	—
1	7	—

(s1) Primera iteración sobre una distancia de un salto ($h = 0$)

$$D_2^{(1)} = \min [D_1^{(0)} + d_{1,2}, D_2^{(0)} + d_{2,2}, D_3^{(0)} + d_{3,2}, D_4^{(0)} + d_{4,2}, D_5^{(0)} + d_{5,2}, D_6^{(0)} + d_{6,2}, D_7^{(0)} + d_{7,2}] = \min (0 + 2, \infty, \infty, \infty, \infty, \infty, \infty) = 2$$

$$D_3^{(1)} = \min [D_1^{(0)} + d_{1,3}, D_2^{(0)} + d_{2,3}, D_3^{(0)} + d_{3,3}, D_4^{(0)} + d_{4,3}, D_5^{(0)} + d_{5,3}, D_6^{(0)} + d_{6,3}, D_7^{(0)} + d_{7,3}] = \min (0 + 1, \infty, \infty, \infty, \infty, \infty, \infty) = 1$$

$$D_4^{(1)} = \min [D_1^{(0)} + d_{1,4}, D_2^{(0)} + d_{2,4}, D_3^{(0)} + d_{3,4}, D_4^{(0)} + d_{4,4}, D_5^{(0)} + d_{5,4}, D_6^{(0)} + d_{6,4}, D_7^{(0)} + d_{7,4}] = \min (0 + 4, \infty, \infty, \infty, \infty, \infty, \infty) = 4$$

$$D_5^{(1)} = \min [D_1^{(0)} + d_{1,5}, D_2^{(0)} + d_{2,5}, D_3^{(0)} + d_{3,5}, D_4^{(0)} + d_{4,5}, D_5^{(0)} + d_{5,5}, D_6^{(0)} + d_{6,5}, D_7^{(0)} + d_{7,5}] = \min (0 + \infty, \infty, \infty, \infty, \infty, \infty, \infty) = \infty$$

$$D_6^{(1)} = \min [D_1^{(0)} + d_{1,6}, D_2^{(0)} + d_{2,6}, D_3^{(0)} + d_{3,6}, D_4^{(0)} + d_{4,6}, D_5^{(0)} + d_{5,6}, D_6^{(0)} + d_{6,6}, D_7^{(0)} + d_{7,6}] = \min (0 + \infty, \infty, \infty, \infty, \infty, \infty, \infty) = \infty$$

$$D_7^{(1)} = \min [D_1^{(0)} + d_{1,7}, D_2^{(0)} + d_{2,7}, D_3^{(0)} + d_{3,7}, D_4^{(0)} + d_{4,7}, D_5^{(0)} + d_{5,7}, D_6^{(0)} + d_{6,7}, D_7^{(0)} + d_{7,7}] = \min (0 + \infty, \infty, \infty, \infty, \infty, \infty, \infty) = \infty$$

(s2) El resultado tras una nueva iteración ($h = 1$)

$$D_2^{(2)} = \min [D_1^{(1)} + d_{1,2}, D_2^{(1)} + d_{2,2}, D_3^{(1)} + d_{3,2}, D_4^{(1)} + d_{4,2}, D_5^{(1)} + d_{5,2}, D_6^{(1)} + d_{6,2}, D_7^{(1)} + d_{7,2}] = \\ = \min (0 + 2, 2 + 0, 1 + \infty, 4 + 1, \infty, \infty, \infty) = 2$$

$$D_3^{(2)} = \min [D_1^{(1)} + d_{1,3}, D_2^{(1)} + d_{2,3}, D_3^{(1)} + d_{3,3}, D_4^{(1)} + d_{4,3}, D_5^{(1)} + d_{5,3}, D_6^{(1)} + d_{6,3}, D_7^{(1)} + d_{7,3}] = \\ = \min (0 + 1, 2 + \infty, 1 + 0, 4 + 1, \infty, \infty, \infty) = 1$$

$$D_4^{(2)} = \min [D_1^{(1)} + d_{1,4}, D_2^{(1)} + d_{2,4}, D_3^{(1)} + d_{3,4}, D_4^{(1)} + d_{4,4}, D_5^{(1)} + d_{5,4}, D_6^{(1)} + d_{6,4}, D_7^{(1)} + d_{7,4}] = \\ = \min (0 + 4, 2 + 1, 1 + 1, 4 + 0, \infty, \infty, \infty) = 2$$

$$D_5^{(2)} = \min [D_1^{(1)} + d_{1,5}, D_2^{(1)} + d_{2,5}, D_3^{(1)} + d_{3,5}, D_4^{(1)} + d_{4,5}, D_5^{(1)} + d_{5,5}, D_6^{(1)} + d_{6,5}, D_7^{(1)} + d_{7,5}] = \\ = \min (0 + \infty, 2 + \infty, 1 + 1, 4 + \infty, \infty, \infty, \infty) = 2$$

$$D_6^{(2)} = \min [D_1^{(1)} + d_{1,6}, D_2^{(1)} + d_{2,6}, D_3^{(1)} + d_{3,6}, D_4^{(1)} + d_{4,6}, D_5^{(1)} + d_{5,6}, D_6^{(1)} + d_{6,6}, D_7^{(1)} + d_{7,6}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 4 + 3, \infty, \infty, \infty) = 7$$

$$D_7^{(2)} = \min [D_1^{(1)} + d_{1,7}, D_2^{(1)} + d_{2,7}, D_3^{(1)} + d_{3,7}, D_4^{(1)} + d_{4,7}, D_5^{(1)} + d_{5,7}, D_6^{(1)} + d_{6,7}, D_7^{(1)} + d_{7,7}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 4 + \infty, \infty, \infty, \infty) = \infty$$

(s3) El resultado tras una nueva iteración ($h = 2$)

$$D_2^{(3)} = \min [D_1^{(2)} + d_{1,2}, D_2^{(2)} + d_{2,2}, D_3^{(2)} + d_{3,2}, D_4^{(2)} + d_{4,2}, D_5^{(2)} + d_{5,2}, D_6^{(2)} + d_{6,2}, D_7^{(2)} + d_{7,2}] = \\ = \min (0 + 2, 2 + 0, 1 + \infty, 2 + 1, 2 + \infty, \infty, \infty) = 2$$

$$D_3^{(3)} = \min [D_1^{(2)} + d_{1,3}, D_2^{(2)} + d_{2,3}, D_3^{(2)} + d_{3,3}, D_4^{(2)} + d_{4,3}, D_5^{(2)} + d_{5,3}, D_6^{(2)} + d_{6,3}, D_7^{(2)} + d_{7,3}] = \\ = \min (0 + 1, 2 + \infty, 1 + 0, 2 + 1, 2 + 1, \infty, \infty) = 1$$

$$D_4^{(3)} = \min [D_1^{(2)} + d_{1,4}, D_2^{(2)} + d_{2,4}, D_3^{(2)} + d_{3,4}, D_4^{(2)} + d_{4,4}, D_5^{(2)} + d_{5,4}, D_6^{(2)} + d_{6,4}, D_7^{(2)} + d_{7,4}] = \\ = \min (0 + 4, 2 + 1, 1 + 1, 2 + 0, 2 + \infty, \infty, \infty) = 2$$

$$D_5^{(3)} = \min [D_1^{(2)} + d_{1,5}, D_2^{(2)} + d_{2,5}, D_3^{(2)} + d_{3,5}, D_4^{(2)} + d_{4,5}, D_5^{(2)} + d_{5,5}, D_6^{(2)} + d_{6,5}, D_7^{(2)} + d_{7,5}] = \\ = \min (0 + \infty, 2 + \infty, 1 + 1, 2 + \infty, 2 + 0, \infty, \infty) = 2$$

$$D_6^{(3)} = \min [D_1^{(2)} + d_{1,6}, D_2^{(2)} + d_{2,6}, D_3^{(2)} + d_{3,6}, D_4^{(2)} + d_{4,6}, D_5^{(2)} + d_{5,6}, D_6^{(2)} + d_{6,6}, D_7^{(2)} + d_{7,6}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 4 + 3, 2 + 1, \infty, \infty) = 3$$

$$D_7^{(3)} = \min [D_1^{(2)} + d_{1,7}, D_2^{(2)} + d_{2,7}, D_3^{(2)} + d_{3,7}, D_4^{(2)} + d_{4,7}, D_5^{(2)} + d_{5,7}, D_6^{(2)} + d_{6,7}, D_7^{(2)} + d_{7,7}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 7 + 2, \infty) = 9$$

(s4) El resultado tras una nueva iteración ($h = 3$)

$$D_2^{(4)} = \min [D_1^{(3)} + d_{1,2}, D_2^{(3)} + d_{2,2}, D_3^{(3)} + d_{3,2}, D_4^{(3)} + d_{4,2}, D_5^{(3)} + d_{5,2}, D_6^{(3)} + d_{6,2}, D_7^{(3)} + d_{7,2}] = \\ = \min (0 + 2, 2 + 0, 1 + \infty, 2 + 1, 2 + \infty, 3 + \infty, \infty) = 2$$

$$D_3^{(4)} = \min [D_1^{(3)} + d_{1,3}, D_2^{(3)} + d_{2,3}, D_3^{(3)} + d_{3,3}, D_4^{(3)} + d_{4,3}, D_5^{(3)} + d_{5,3}, D_6^{(3)} + d_{6,3}, D_7^{(3)} + d_{7,3}] = \\ = \min (0 + 1, 2 + \infty, 1 + 0, 2 + 1, 2 + 1, 3 + \infty, \infty) = 1$$

$$D_4^{(4)} = \min [D_1^{(3)} + d_{1,4}, D_2^{(3)} + d_{2,4}, D_3^{(3)} + d_{3,4}, D_4^{(3)} + d_{4,4}, D_5^{(3)} + d_{5,4}, D_6^{(3)} + d_{6,4}, D_7^{(3)} + d_{7,4}] = \\ = \min (0 + 4, 2 + 1, 1 + 1, 2 + 0, 2 + \infty, 3 + 3, \infty) = 2$$

$$D_5^{(4)} = \min [D_1^{(3)} + d_{1,5}, D_2^{(3)} + d_{2,5}, D_3^{(3)} + d_{3,5}, D_4^{(3)} + d_{4,5}, D_5^{(3)} + d_{5,5}, D_6^{(3)} + d_{6,5}, D_7^{(3)} + d_{7,5}] = \\ = \min (0 + \infty, 2 + \infty, 1 + 1, 2 + \infty, 2 + 0, 3 + 1, \infty) = 2$$

$$D_6^{(4)} = \min [D_1^{(3)} + d_{1,6}, D_2^{(3)} + d_{2,6}, D_3^{(3)} + d_{3,6}, D_4^{(3)} + d_{4,6}, D_5^{(3)} + d_{5,6}, D_6^{(3)} + d_{6,6}, D_7^{(3)} + d_{7,6}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 4 + 3, 2 + 1, 3 + 0, \infty) = 3$$

$$D_7^{(4)} = \min [D_1^{(3)} + d_{1,7}, D_2^{(3)} + d_{2,7}, D_3^{(3)} + d_{3,7}, D_4^{(3)} + d_{4,7}, D_5^{(3)} + d_{5,7}, D_6^{(3)} + d_{6,7}, D_7^{(3)} + d_{7,7}] = \\ = \min (0 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 3 + 2, 9 + 0) = 5$$

(s5) A continuación se realiza la iteración para $h = 4$. Ésta será la última, ya que no se modifica nada con respecto a la iteración anterior.

$$D_2^{(5)} = \min [D_1^{(4)} + d_{1,2}, D_2^{(4)} + d_{2,2}, D_3^{(4)} + d_{3,2}, D_4^{(4)} + d_{4,2}, D_5^{(4)} + d_{5,2}, D_6^{(4)} + d_{6,2}, D_7^{(4)} + d_{7,2}] = \min (0 + 2, 2 + 0, 1 + \infty, 2 + 1, 2 + \infty, 3 + \infty, \infty) = 2$$

$$D_3^{(5)} = \min [D_1^{(4)} + d_{1,3}, D_2^{(4)} + d_{2,3}, D_3^{(4)} + d_{3,3}, D_4^{(4)} + d_{4,3}, D_5^{(4)} + d_{5,3}, D_6^{(4)} + d_{6,3}, D_7^{(4)} + d_{7,3}] = \min (0 + 1, 2 + \infty, 1 + 0, 2 + 1, 2 + 1, 3 + \infty, \infty) = 1$$

$$D_4^{(5)} = \min [D_1^{(4)} + d_{1,4}, D_2^{(4)} + d_{2,4}, D_3^{(4)} + d_{3,4}, D_4^{(4)} + d_{4,4}, D_5^{(4)} + d_{5,4}, D_6^{(4)} + d_{6,4}, D_7^{(4)} + d_{7,4}] = \min (0 + 4, 2 + 1, 1 + 1, 2 + 0, 2 + \infty, 3 + 3, \infty) = 2$$

$$D_5^{(5)} = \min [D_1^{(4)} + d_{1,5}, D_2^{(4)} + d_{2,5}, D_3^{(4)} + d_{3,5}, D_4^{(4)} + d_{4,5}, D_5^{(4)} + d_{5,5}, D_6^{(4)} + d_{6,5}, D_7^{(4)} + d_{7,5}] = \min (0 + \infty, 2 + \infty, 1 + 1, 2 + \infty, 2 + 0, 3 + 1, \infty) = 2$$

$$D_6^{(5)} = \min [D_1^{(4)} + d_{1,6}, D_2^{(4)} + d_{2,6}, D_3^{(4)} + d_{3,6}, D_4^{(4)} + d_{4,6}, D_5^{(4)} + d_{5,6}, D_6^{(4)} + d_{6,6}, D_7^{(4)} + d_{7,6}] = \min (0 + \infty, 2 + \infty, 2 + \infty, 4 + 3, 2 + 1, 3 + 0, 5 + 2) = 3$$

$$D_7^{(5)} = \min [D_1^{(4)} + d_{1,7}, D_2^{(4)} + d_{2,7}, D_3^{(4)} + d_{3,7}, D_4^{(4)} + d_{4,7}, D_5^{(4)} + d_{5,7}, D_6^{(4)} + d_{6,7}, D_7^{(4)} + d_{7,7}] = \min (0 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 2 + \infty, 3 + 2, 5 + 0) = 5$$

La tabla de encaminamiento resulta ser la misma que en el problema anterior y los caminos que van a seguir los mensajes al ser transmitidos desde el nodo 1 se corresponden con el árbol de alcanzabilidad que se muestra en la Figura 8.12.

Tabla de encaminamiento		
Origen	Destino	Out
1	2	2
1	3	3
1	4	4
1	5	3
1	6	3
1	7	3

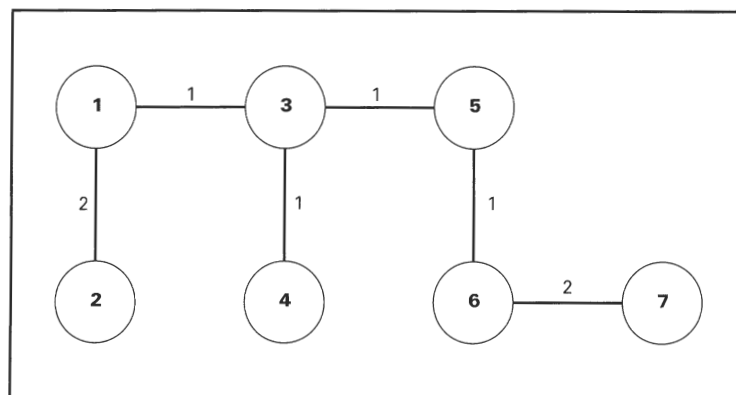


Figura 8.12. Árbol de alcanzabilidad resultante

8.7.

Determinar la tabla de encaminamiento y el árbol de alcanzabilidad para la siguiente configuración de red (Figura 8.13). Aplicar los Algoritmos de Dijkstra y Bellman-Ford.

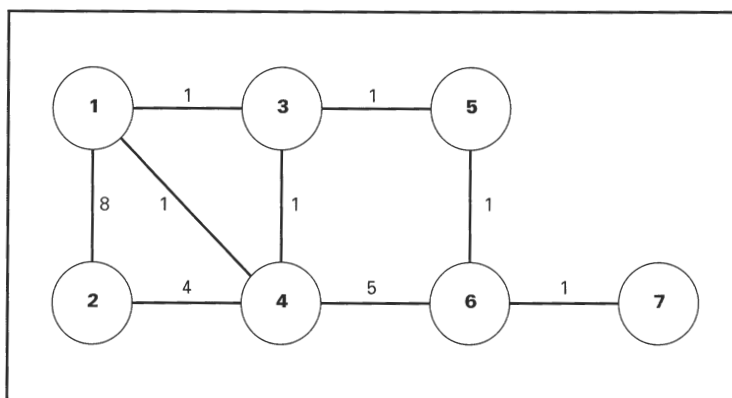


Figura 8.13. Ejemplo de configuración de red

SOLUCIÓN

El resultado de aplicar el Algoritmo de Dijkstra se resume en la Tabla 8.5.

Paso	M	Coste					
		2	3	4	5	6	7
0	{1}	8	1	1	∞	∞	∞
1	{1, 3}	8	1	1	2	∞	∞
2	{1, 3, 4}	8	1	1	2	6	∞
3	{1, 3, 4, 5}	5	1	1	2	3	∞
4	{1, 3, 4, 5, 6}	5	1	1	2	3	4
5	{1, 3, 4, 5, 6, 7}	5	1	1	2	3	4
6	{1, 2, 3, 4, 5, 6, 7}	5	1	1	2	3	4

Tabla 8.5. Resumen de resultados al aplicar el Algoritmo de Dijkstra.

En el caso de aplicar el Algoritmo de Bellman-Ford, la Tabla 8.6 resume el resultado de la ejecución del algoritmo.

Paso	Coste					
	2	3	4	5	6	7
0	∞	∞	∞	∞	∞	∞
1	8	1	1	∞	∞	∞
2	5	1	1	2	6	∞
3	5	1	1	2	3	7
4	5	1	1	2	3	4

Tabla 8.6. Resumen de resultados al aplicar el Algoritmo de Bellman-Ford.

El árbol de alcanzabilidad y la tabla de encaminamiento resultantes en ambos casos se presentan a continuación.

Tabla de encaminamiento			
Origen	Destino	Out	Ruta
1	2	4	1-4-2
1	3	3	1-3
1	4	4	1-4
1	5	3	1-3-5
1	6	3	1-3-5-6
1	7	3	1-3-5-6-7

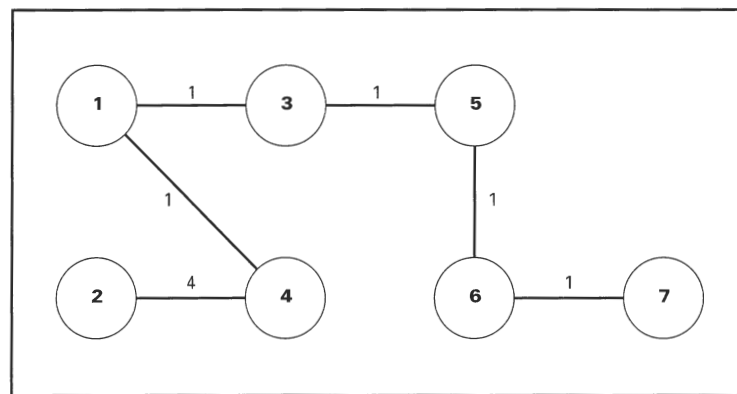


Figura 8.14. Árbol de alcanzabilidad resultante

La mayor desventaja de este tipo de algoritmos es que hacen que algunos de los enlaces no se utilicen para la transmisión de la información de los nodos. En este caso, el nodo 1 no va a transmitir información a través de los enlaces (1, 2), (3, 4) y (4, 6).

8.8.

En la literatura se han propuesto otros algoritmos que en lugar de "eliminar" enlaces, como en los algoritmos anteriores, lo que hacen es prohibir el paso de información entre dos enlaces. Por ejemplo, utilizando la Figura 8.15, los mensajes del nodo 1 que hayan circulado por el enlace (4, 2) no se permite que circulen por el enlace (2, 1). En este caso se dice que se ha creado el turn (1, 4, 2).

Esta filosofía de funcionamiento consigue evitar los ciclos en las transmisiones y que aumente la utilización media de los enlaces, haciendo que el retardo medio máximo de los mensajes sea menor que al usar algoritmos como los anteriores (Dijkstra y Bellman-Ford).

A continuación se presenta la propuesta de algoritmo dada por Starobinski para crear turns. Aplíquelo a la configuración de red que se muestra en la Figura 8.15 indicando los turns creados por el algoritmo.

SOLUCIÓN

Versión simplificada del Algoritmo de Starobinski. En este caso, se repiten los siguientes pasos hasta que ya no haya nodos con los que trabajar.

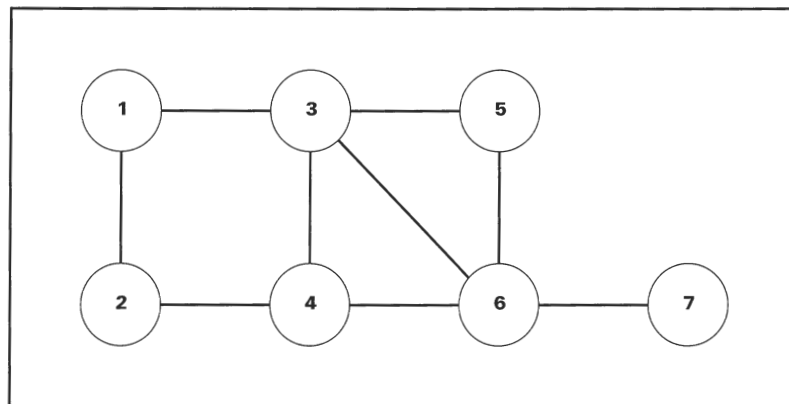


Figura 8.15. Ejemplo de configuración de red.

Paso 1. Seleccionar un nodo de grado mínimo (el que tenga menor número de enlaces). En lo que sigue se le denota por a . En caso de haber varios se elige uno de ellos al azar.

Paso 2. Prohibir todos los turns a su alrededor. Es decir, prohibir todos los turns tipo (b, a, c) .

Paso 3. Permitir todos los turns que empiezan en el nodo a . Es decir, permitir todos los turns tipo (a, b, c) .

Paso 4. Eliminar el nodo a y los enlaces que llegan a él.

Inicialmente, el nodo elegido es el nodo 7, ya que su grado es 1. Tras los pasos 2 y 3 resulta que tenemos que considerar una nueva configuración de red como la que se muestra en la Figura 8.16.(1).

En el primer paso, desde el nodo 7, no se crea ningún turn en la red.

Para realizar el siguiente paso se tiene que elegir entre varios nodos con el mismo grado (2) y estos son los nodos 1, 2 y 5. Elegimos el 2. En este caso, el nodo 2 tiene que crear el turn $(4, 2, 1)$ y el resto queda permitido (Figura 8.16.(3)). Como efecto de este ciclo del algoritmo, el nodo 1 pasa a ser el que menor grado tiene (1) y, por tanto, es el elegido para realizar el siguiente ciclo del algoritmo.

Para el nodo 1 la situación es similar a lo ocurrido para el nodo 7, y tras su eliminación la configuración de la red resulta ser la mostrada en la Figura 8.16.(4).

Ahora se elige al nodo 4 para continuar la ejecución del algoritmo. Éste creará el turn $(3, 4, 6)$ como se muestra en la Figura 8.16.(5). La eliminación de este nodo hace que los nodos 5 y 6 sean los de grado mínimo. Elegimos el nodo 5. La ejecución del algoritmo hace que éste cree el turn $(3, 5, 6)$ como se muestra en la Figura 8.16.(6). Finalmente, quedan los nodos 3 y 6 junto con un enlace que los une. Al ejecutar un nuevo ciclo del algoritmo se elimina uno de los nodos y el enlace entre ellos y finalmente en el siguiente paso se elimina el nodo restante.

La interpretación de los turns creados es que la información que se transmite del nodo 1 al nodo 4 va a través del nodo 3 y nunca se dejará que atraviese por el nodo 2. Sin embargo, la información del nodo 2 al 4 sí que viajará por el enlace entre ellos. En definitiva tenemos un árbol de alcanzabilidad entre los nodos de la red en el que no hay ciclos ni enlaces prohibidos.

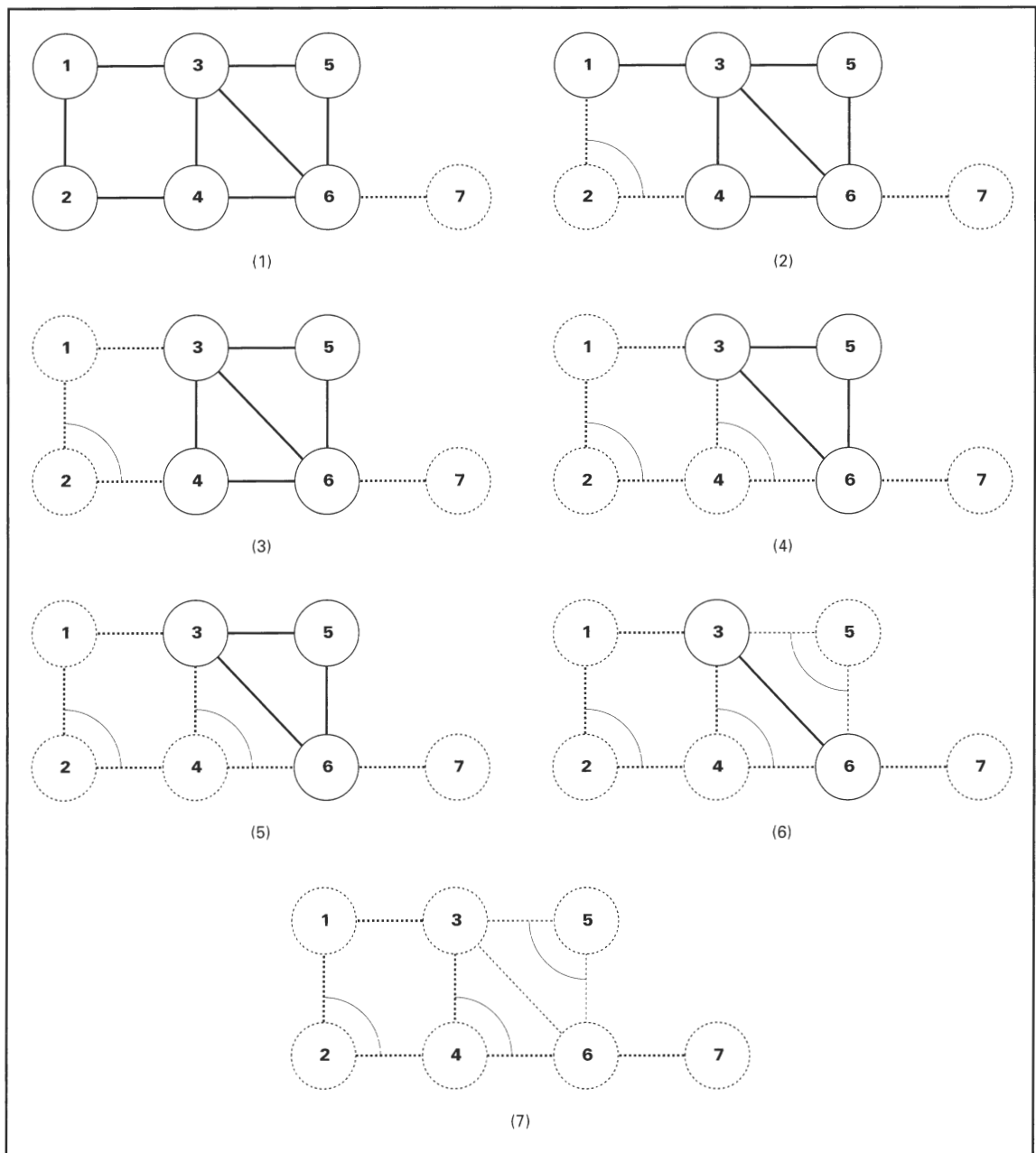
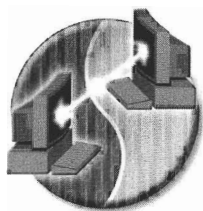


Figura 8.16 Resultado paso a paso de la ejecución del Algoritmo de Starobinski.



TRANSFERENCIA EN MODO ASÍNCRONO Y RETRANSMISIÓN DE TRAMAS

9.1. Mostrar el efecto de la multiplexación de celdas ATM y la multiplexación TDM en el siguiente sistema (Figura 9.1).

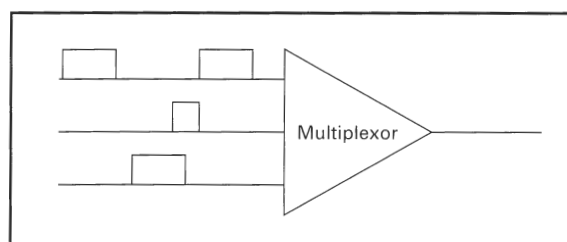


Figura 9.1. Esquema de multiplexor.

SOLUCIÓN

En este caso, la diferencia estriba en que TDM tiene asignados slots para los diferentes usuarios y sólo ellos pueden hacer uso de ellos. Sin embargo, los conmutadores ATM realizan multiplexación estadística. La diferencia entre ambos tipos de multiplexación se ilustra en el siguiente ejemplo (Figura 9.2).

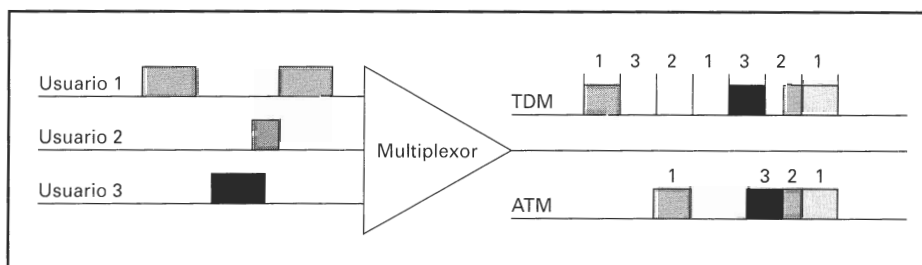


Figura 9.2. Comparación TDM/Multiplexación estadística.

Se aprecia claramente la diferencia entre ambos tipos de multiplexación. En el caso de TDM se requiere que cada comunicación ocupe su slot temporal, mientras que con la multiplexación estadística las comunicaciones pueden proseguir si el enlace está libre.

9.2.

Un mensaje de m bytes se transmite en una red de conmutación de celdas. Cada celda, de tamaño fijo, tiene un tamaño de C bytes de cabecera y D bytes de datos, añadiendo bytes de relleno si fuera necesario. Se define la eficiencia como la relación entre lo que se desea transmitir y lo que realmente se transmite.

Calcular la expresión de la eficiencia en función de m , C y D y dibujar la eficiencia considerando que $m = 16k$ bytes ($k = 0 \dots 10$) y que estamos utilizando celdas ATM.

SOLUCIÓN

En este caso, la eficiencia está determinada por los bytes a transmitir por el usuario (m bytes) y el número de total de bytes que se transmiten. Este último corresponde a la suma de los bytes que tienen todas las celdas que se transmiten (cabeceras más datos). Por tanto, necesitamos conocer cuántas celdas se transmiten. Este valor se puede calcular del siguiente modo N (Número celdas totales) = $\lceil \text{bytes totales} / \text{tamaño de la celda en bytes} \rceil$, donde $\lceil x \rceil$ determina el mayor entero más próximo a x . Así, tenemos que:

$$N = \lceil m / (C + D) \rceil$$

Entonces, el total de bytes transferidos resulta ser:

$$B \text{ (total bytes transferidos)} = N \cdot \text{tamaño de la celda en bytes}$$

$$B = N \cdot (C + D)$$

Por tanto, la eficiencia se calcula como:

$$E \text{ (eficiencia)} = \frac{m}{B} = \frac{m}{(N(C + D))} = \frac{m}{\left\lceil \frac{m}{C + D} \right\rceil (C + D)}$$

Aplicando los datos del problema: $C = 5$ bytes, $D = 48$ bytes y $m = 16k$ ($k = 0 \dots 40$) se obtiene la siguiente gráfica (Figura 9.3).

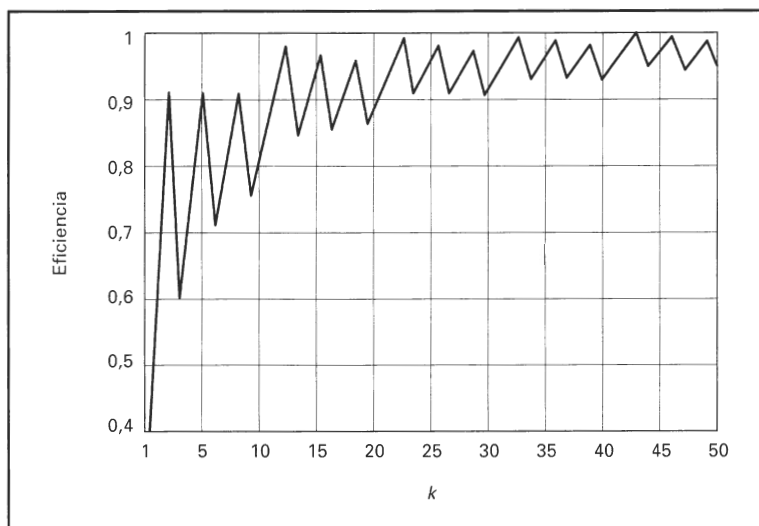


Figura 9.3. Eficiencia de la transmisión con celdas.

En esta gráfica se ve que la eficiencia para tamaños bajos oscila mucho debido a que son necesarias pocas celdas y si alguna va “medio” llena va a reducir en gran medida la eficiencia. A medida que aumenta el tamaño de datos a transferir ocurre que la relación entre celdas completas frente a parcialmente ocupadas con datos va a aumentar y, por tanto, también la eficiencia.

La reducción de la eficiencia es muy clara para transferencias de pocos datos.

Nota: Transferir un paquete de tamaño 1.500 bytes supone enviar 32 celdas; mientras que uno de 64 bytes supone 2 celdas.

9.3.

Estudiar el comportamiento de un cubo con permisos (token bucket) cuyas características son las siguientes: los tamaños de las celdas son de dos unidades de tiempo, al llegar se almacenan en una cola infinita, la capacidad del cubo es de dos tokens y la tasa a la que se recibe el token es de 1 token cada 3 unidades de tiempo.

SOLUCIÓN

El cubo se llena a una tasa de 1 token cada 3 unidades de tiempo. Ese intervalo de tiempo es suficiente, en este caso, como para que la mayor parte del tiempo la cola del sistema (buffer) esté ocupada por celdas que tienen que transmitirse.

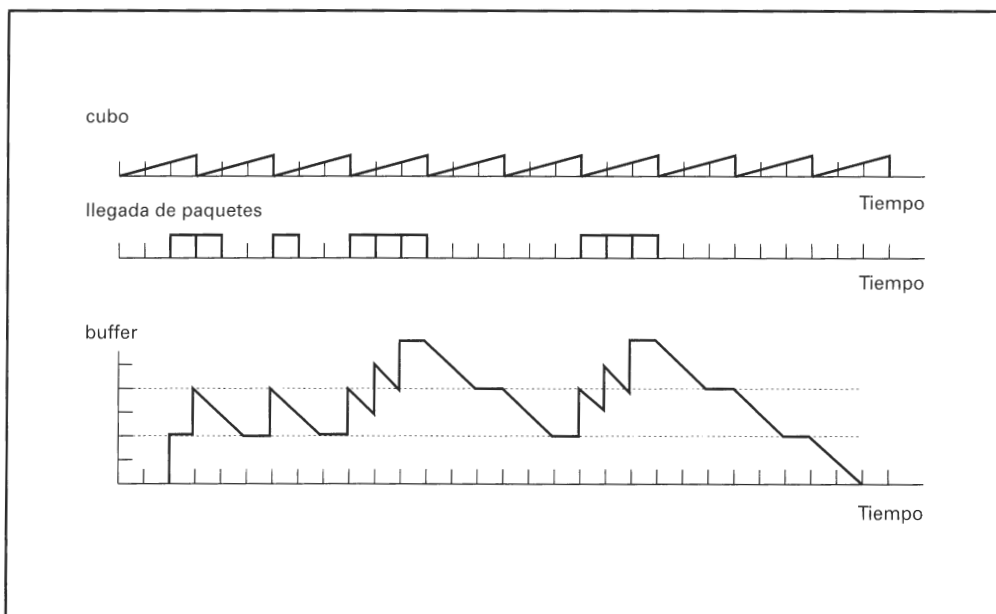


Figura 9.4. Funcionamiento de un token bucket con buffer infinito.

9.4.

Considerar, como en el caso anterior (Problema 9.3), que se tiene un cubo con permisos, pero esta vez se tiene que la cola es finita y no puede almacenar más de dos celdas. Dibujar el diagrama de transmisión de las celdas.

SOLUCIÓN

En este caso, se ve que una celda al llegar no encuentra sitio en el buffer y, por tanto, se elimina. Perder esta celda hace que las restantes celdas tengan un menor retardo extremo a extremo, pero también va a exigir que se retransmita un conjunto de celdas si el contenido de ésta era parte de un paquete mayor.

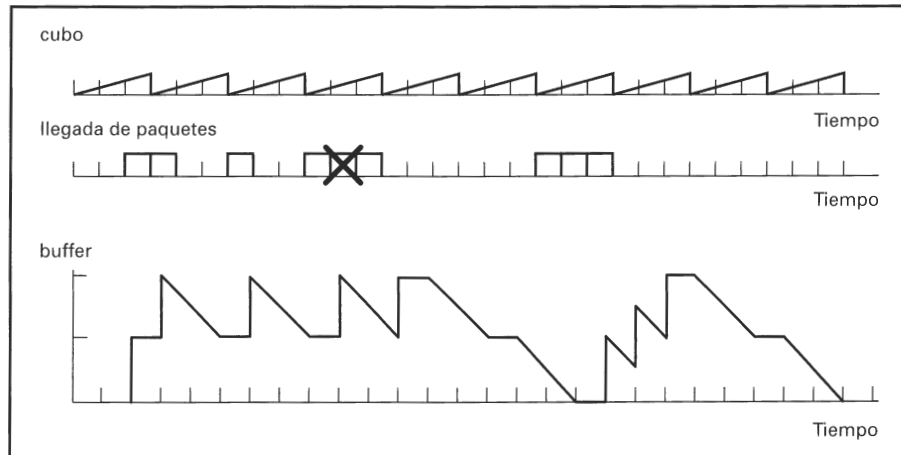


Figura 9.5. Funcionamiento del token bucket con buffer finito (2 celdas).

9.5.

Definir un proceso que cuente la llegada de paquetes y determinar las condiciones para que satisfaga un limitador de tipo cubo con permisos con parámetros (σ, ρ) , donde σ es la capacidad del cubo y ρ la tasa media a la que llegan los tokens al cubo.

SOLUCIÓN

El proceso de cuenta de las llegadas en tiempo continuo se puede definir como $A(t)$ que acumula las llegadas que se van produciendo a lo largo del tiempo. Inicialmente, $A(0) = 0$ y evidentemente es un proceso creciente $A(s) \leq A(t)$, para todo $s \leq t$.

La condición que se puede exigir a este proceso para satisfacer un cubo con permisos es la que imponga que el conjunto de llegadas sea similar al proceso de salidas que produce un cubo con permisos. Sean s y t dos instantes de tiempo cualesquiera, entonces

$$A(t) - A(s) \leq \rho(t - s) + \sigma, \text{ para todo } 0 \leq s \leq t$$

Se dice que el proceso $A(t)$ es (σ, ρ) -limitado.

9.6.

Sea $a(t) = A(t) - A(t - 1)$ el número de llegadas en un tiempo t del proceso de llegadas A . Este proceso de llegadas se dice que es periódico con período $p > 0$ si $a(t + p) = a(t)$ para todo $t > 0$. Determinar si este proceso es (σ, ρ) -limitado.

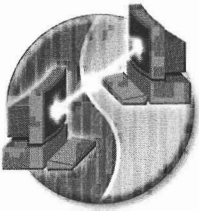
SOLUCIÓN

Sea $A(p)$ el número total de llegadas en un período p . Así,

$$A(t) - A(s) \leq \left\lceil \frac{t-s}{p} \right\rceil A(p) \leq \left(\frac{t-s}{p} + 1 \right) A(p) = \frac{A(p)}{p}(t-s) + A(p)$$

donde $\lceil x \rceil$ es la función que determina el mayor entero más cercano al valor x .

El resultado muestra que el proceso de llegadas periódico con período p es (σ, ρ) -limitado con $\sigma = A(p)$ y $\rho = A(p)/p$.



CONGESTIÓN EN REDES DE DATOS

10.1.

Suponga un enlace ATM punto a punto a 155 Mbps y 500 m de longitud. Determine el tamaño mínimo de buffer necesario en el transmisor para poder garantizar una transmisión fiable de información si los errores en el medio son muy pequeños y nunca se reciben dos paquetes erróneos seguidos.

SOLUCIÓN

Si suponemos que el receptor detecta una celda n errónea o perdida tan pronto como se recibe la siguiente $n + 1$ y que entonces manda una celda de vuelta solicitando retransmisión tendremos que tener en cuenta los bits en la red desde que se envía n hasta que se recibe $n + 1$ en el origen.

Como el origen tiene que conservar la celda n hasta que recibe la celda de petición de reenvío, el tiempo total que tendrá que mantener la celda n suponiendo que se envía la celda $n + 1$ inmediatamente después será el tiempo de transmisión de la celda $n + 1$, más el tiempo de propagación de la celda $n + 1$, más el tiempo de propagación de vuelta de la celda de petición de reenvío:

$$t = t_{tx} + t_{prop} + t_{prop} = 7,73 \cdot 10^{-6} \text{ s}$$

$$t_{tx} = 53 \cdot 8 / 155 \cdot 10^6 = 2,73 \cdot 10^{-6} \text{ s}$$

$$t_{prop} = 500 / 2 \cdot 10^8 = 2,5 \cdot 10^{-6} \text{ s}$$

$$\text{tamañobuffer}_{\text{bits}} = t \cdot R = 7,73 \cdot 10^{-6} \cdot 155 \cdot 10^6 = 1.198,15 \rightarrow 1.199 \text{ bits}$$

$$\text{tamañobuffer}_{\text{celdas}} = \text{tamañobuffer}_{\text{bits}} / \text{tamañoocelda}_{\text{bits}} = 1.199 / (53 \cdot 8) = 2,82$$

→ 3 celdas aproximadamente

10.2.

Si un enlace punto a punto de 100 Mbps y con un *RTT* (Round Trip Time) de 10 ms utiliza un sistema de control de congestión basado en paquete de obstrucción (choke packet), ¿qué tamaño mínimo de buffer es necesario en el receptor para no perder paquetes suponiendo que el paquete de obstrucción se envía cuando el receptor tiene ocupado la mitad del buffer?

SOLUCIÓN

Si B es el tamaño del buffer en recepción, $B/2$ ha de ser capaz de almacenar los paquetes que sigan llegando en el tiempo que le cueste al paquete de obstrucción llegar al origen.

$$\frac{B}{2} = \frac{RTT}{2R} = \frac{10 \cdot 10^{-3}}{2} \cdot 100 \cdot 10^6 = 500 \cdot 10^3 \text{ bits} = 62,5 \cdot 10^3 \text{ bytes}$$

Por tanto:

$$B = 125 \cdot 10^3 \text{ bytes} = 122 \text{ Kbytes}$$

Para asegurarnos que el último paquete se pudiera recibir habría que sumar a este tamaño de buffer un margen de seguridad que sería el tamaño del paquete de longitud máxima posible en esa red.

10.3.

Suponga un medio de transmisión a 10 Mbps compartido por varias máquinas en el que el control de acceso al medio se realiza por detección de colisiones. Estime cuál será el efecto de la congestión en esa red.

SOLUCIÓN

Al ser una red con colisiones sobre un medio compartido, el período de congestión moderado se alcanzará antes (en torno al 60-80% de utilización de la red) que con una red de medio no compartido. Por tanto, los retardos medios serán mayores durante estos períodos de congestión de lo que lo serían en una red con un medio no compartido.

10.4.

Suponga la red de la figura que posee una señalización por congestión explícita basada en tasa. Todos los enlaces son de 1 Mbps, se supone que el reparto es proporcional a su tasa deseada de transmisión y en el instante inicial se intentan establecer los siguientes flujos:

- $f_{BD} = 1 \text{ Mbps}$
- $f_{AD} = 1 \text{ Mbps}$
- $f_{CD} = 500 \text{ Mbps}$

¿Cuál será la tasa final de estos flujos?

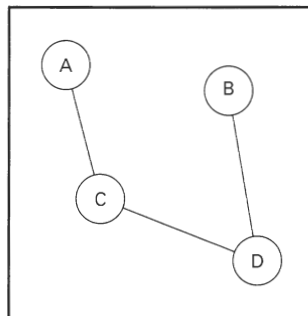


Figura 10.1. Red con señalización por congestión.

SOLUCIÓN

El flujo f_{BD} no comparte el enlace, por lo que obtendrá todo el ancho de banda deseado:

$$f_{BD} = 1 \text{ Mbps}$$

Los flujos f_{AD} y f_{CD} comparten el enlace CD, por lo que éste será el cuello de botella. El ancho de banda de este enlace deberá ser compartido proporcionalmente entre todos los flujos que pasen por él. En este caso son los dos flujos, con lo que:

$$f_{AD} = 1 \text{ Mbps} / 1,5 \text{ Mbps} \cdot 1 \text{ Mbps} = 666 \text{ Kbps}$$

$$f_{CD} = 0,5 \text{ Mbps} / 1,5 \text{ Mbps} \cdot 1 \text{ Mbps} = 333 \text{ Kbps}$$

10.5.

Sea un enlace ATM punto a punto de 155 Mbps entre dos nodos A y B. El nodo A transmite 2 celdas por milisegundo y el nodo B consume 1 celda por milisegundo. ¿Cuánto tiempo tardará el nodo B en congestionarse si no existe ningún mecanismo de control de congestión y suponiendo que el nodo B posee un buffer de 10 celdas de capacidad?

SOLUCIÓN

Suponiendo el tiempo de propagación despreciable, cada milisegundo el nodo B irá llenando en una celda su buffer (2 celdas / ms - 1 celda / ms). Como el buffer del nodo B tiene capacidad para 10 celdas, B se congestionará en 10 milisegundos:

$$\frac{10 \text{ celdas}}{(2 - 1) \text{ celdas / ms}} = 10 \text{ ms}$$

A partir del milisegundo 10,5 el nodo B empezará a tirar celdas.

10.6.

Se dispone de un enlace de 500 Kbps y una fuente de tráfico que transmite ráfagas de 1 s de duración a 1 Mbps cada 2 segundos. Determine:

- La mejor técnica de control de congestión que se puede aplicar.
- El retardo que ésta introducirá en el peor caso.
- Si se quiere garantizar un retardo máximo añadido de 0,5 s, ¿qué técnica aplicaría?

SOLUCIÓN

- a) Si cada 2 s tenemos una ráfaga de 1 Mbps, tendremos una tasa media de:

$$1 \text{ Mb} / 2 \text{ s} = 500 \text{ Kbps}$$

Por tanto, la capacidad del enlace es suficiente y el mecanismo de control de congestión más adecuado será un *traffic shaping* (conformado de tráfico) a 500 Kbps.

- b) El peor caso corresponderá al retardo sufrido por el último bit de cada ráfaga que se retrasará 1 s:

$$t_{\text{retardo}} = t_{\text{TXráfaga}} - t_{\text{salida último bit}} = 1 \text{ Mb} / 500 \text{ Kbps} - 1 = 1 \text{ s}$$

- c) El *traffic shaping* a 500 Kbps, que es el máximo que permite el enlace, se descarta porque introduce un retardo máximo como hemos visto de 1 s. Esto supone que la única manera para reducir el retardo consiste en tirar ciertos paquetes dentro de cada ráfaga. En concreto, si deseamos 0,5 s de retardo para el peor caso, que es el del último bit (que se transmite en el instante 1 s de cada ráfaga), supondrá que disponemos de un $t_{\text{TXráfaga}} = 1 + 0,5 = 1,5 \text{ s}$.

Por tanto, se podrán transmitir con éxito a la máxima tasa del enlace:

$$\text{bits}_{\text{TXéxito}} = r \cdot t_{\text{TXráfaga}} = 500 \text{ Kbps} \cdot 1,5 \text{ s} = 750 \text{ Kbits}$$

y, por tanto, será necesario descartar:

$$\text{bits}_{\text{TXdescartados}} = \text{bits}_{\text{ráfaga}} - \text{bits}_{\text{TXéxito}} = 1 \text{ Mbit} - 750 \text{ Kbits} = 250 \text{ Kbits}$$

10.7.

Determine la tasa media, tasa de pico, tamaño de ráfaga, mínimo y máximo retardo para un *token bucket* (cubo de permisos) con los siguientes parámetros medidos en celdas:

Tasa tokens = ρ

Capacidad del cubo en tokens = β

Capacidad del buffer = K

Velocidad del enlace de salida = R

¿Cuántos paquetes se transmitirían en t segundos si había en ese momento g tokens en el cubo? Dibuje la tasa instantánea de salida para ese caso.

SOLUCIÓN

Supongamos cierto instante t_0 en el que tenemos α tokens, con $\alpha \leq \beta$. En el instante $t_0 + t$, el cubo acumula ρt tokens más.

El número total de tokens en $t_0 + t$ es: $\alpha + \rho t$ (con $\alpha + \rho t \leq \beta + \rho t$)

Tasa media = tasa tokens = ρ

Tasa de pico = velocidad de enlace de salida = R

Tamaño de ráfaga: $t_{\text{ráfaga}} = \frac{\beta}{R - \rho}$

Mínimo retardo: hay token, con lo que la celda sale inmediatamente, $t_{\text{mín}} = 0$

Máximo retardo: que el buffer esté lleno K y el cubo vacío. Entonces la celda nueva se transmitirá

en $t_{\text{máx}} = \frac{K}{\rho}$

Si había γ tokens al comenzar la transmisión, en t segundos se transmitirían: $\gamma + \rho t$ celdas. Esta expresión se corresponde con la siguiente figura:

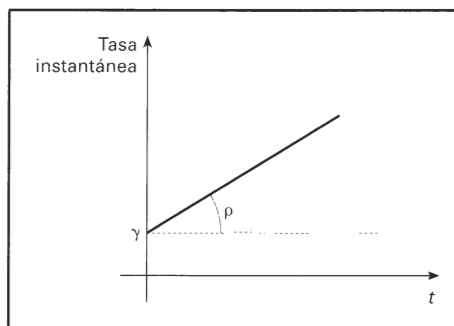


Figura 10.2. Tasa instantánea frente al tiempo.

10.8.

Suponga un *token bucket* (cubo de permisos) con tokens de 100 bytes y capacidad para 1.000 bytes de tokens y tasa de 4 tokens/s.

- ¿Puede este regulador manejar paquetes de 2.000 bytes?
- Si el paquete es de 500 bytes y llega cuando el cubo tiene 300 bytes y no hay otro paquete esperando para servicio, ¿cuál es el retardo medio que sufrirá antes de transmitirlo?

SOLUCIÓN

- No, el máximo tamaño de ráfaga es de 1.000 bytes, que es la capacidad en tokens del regulador.
- Depende de cuándo llegue el paquete respecto al token siguiente. Si llega justo antes del token, como necesitamos 2 tokens más para llegar a los 500 bytes ($300 + 100 + 100$) y sólo hay que esperar por el segundo token:

$$t = 0 + 0,25 \text{ s/token} = 0,25 \text{ s}$$

Si llega justo después del último token, necesita esperar 2 tokens completos:

$$t = 0,25 \text{ s/token} + 0,25 \text{ s/token} = 0,5 \text{ s}$$

Por tanto, el tiempo medio será:

$$t_{\text{medio}} = \frac{0,25 + 0,5}{2} = 0,375 \text{ s}$$

10.9.

Sea un *token bucket* (cubo de permisos) de tamaño β bytes, ρ bytes/s y con una tasa de salida máxima de R bytes/s. Calcule la máxima tasa de ráfaga y la longitud de la ráfaga a tasa máxima.

SOLUCIÓN

La máxima tasa de ráfaga es R bytes/s.

Para la longitud de la ráfaga a tasa máxima, en t segundos se transmiten $R \cdot t$ bytes. Si una ráfaga vacía el token en ese período: $\beta + \rho t = Rt$ y despejando:

$$t = \frac{\beta}{(R - \rho)}$$

10.10.

Suponga un *token bucket* (cubo de permisos) con una capacidad de 2.000 bytes, tokens (permisos) de 100 bytes, una tasa de 5 tokens/s y un canal de salida de 128 Kbps.

- ¿Cuál es la tasa de pico, tasa media y ráfaga más grande de este regulador?
- Si un paquete de 1.000 bytes llega cuando la cesta contiene 400 bytes y no hay paquetes esperando servicio, ¿cuál es el retardo medio que sufre?
- ¿Puede este regulador soportar el manejo de paquetes de 3.000 bytes?

SOLUCIÓN

- a) La tasa de pico está acotada porque ráfagas del tamaño máximo pueden enviarse tan rápido como el canal lo permita. Por tanto, en nuestro caso la tasa de pico será 128 Kbps.

$$\text{tasa media: } \rho = 5 \text{ tokens/s} \cdot 100 \text{ bytes/token} = 4 \text{ Kbps}$$

$$\text{tamaño máximo de ráfaga} = \text{capacidad token bucket} = 2.000 \text{ bytes}$$

- b) El retraso depende de cuándo llegue el paquete. Se necesitan tokens equivalentes a $1.000 - 400 = 600$ bytes, es decir, 6 tokens.

Si llega justo antes del token, deberá esperar sólo a 5 tokens completos:

$$t_{\text{token}} = 1/5 = 0,2 \text{ s}$$

$$t = 5 \cdot t_{\text{token}} = 1 \text{ s}$$

Si llega justo después de cuando llega el token:

$$t = 6 \cdot t_{\text{token}} = 1,2 \text{ s}$$

En media, el retraso será:

$$t_{\text{medio}} = \frac{1 + 1,2}{2} = 1,1 \text{ s}$$

- c) No puede soportar el manejo de paquetes de 3.000 bytes porque el tamaño de ráfaga está limitado a 2.000 bytes. Por tanto, nunca habrá suficientes tokens para enviar el paquete.

10.11.

Un aspecto distintivo de ATM es el tamaño fijo de celda. Demuestre el por qué de este valor contestando las siguientes cuestiones:

- Calcule la eficiencia de una red ATM suponiendo una transmisión continua en la que la red está completamente ocupada de celdas, en función del tamaño de datos por celda D y considerando la cabecera por celda constante de 5 bytes.
- Calcule el retardo de paquetización introducido por la necesidad de almacenar en un buffer los bits de datos hasta completar una celda y mandarla por la red a una tasa R .
- Teniendo en cuenta que la voz está típicamente codificada a 64 Kbps y que ATM optimiza la transferencia de este tipo de tráfico, justifique la elección de celdas de 48 bytes de datos teniendo en cuenta los resultados de a) y b).

SOLUCIÓN

- a) El tamaño de una celda vendrá dado por $D + 5$, de lo que D representa el tamaño de datos útiles. Este D en ATM es de 48 bytes. Por tanto, la eficiencia para una red con tráfico para transmitir será dependiente de la sobrecarga introducida por la cabecera de la celda:

$$\eta = \frac{D}{D + 5}$$

- b) El retardo de paquetización será:

$$T_{\text{paq}} = \frac{8 \cdot D}{R}$$

- c) Si se grafican las expresiones de los apartados a) y b) en función de D (la Gráfica b) con valores pequeños de tiempo en la parte superior del eje de ordenadas y con $R = 64$ Kbps, vemos que el punto de compromiso entre eficiencia y tiempo de paquetización anda en torno a los 48 bytes que es el punto elegido por ATM.

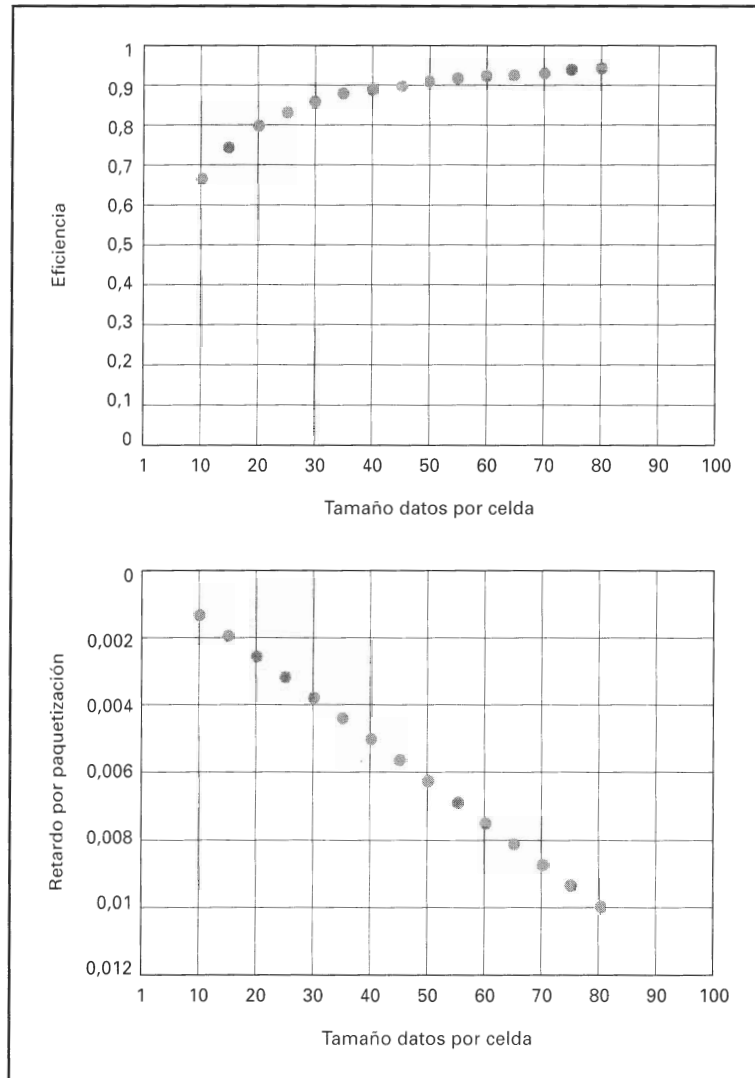


Figura 10.3. Gráficas de las expresiones de los apartados a) y b).

10.12.

Sea una transmisión ATM que tiene una tasa de error por bit (BER) de P y cada celda tiene C bits. Calcular la tasa de celdas erróneas (TCE).

SOLUCIÓN

La probabilidad de que no haya error en una celda será la probabilidad de que no haya ningún bit erróneo. Como la probabilidad de que un bit no sea erróneo es $(1 - P)$, tendremos que la probabilidad de que no haya error en una celda es igual a que los C bits no sean erróneos:

$$(1 - P)^C$$

Por tanto, la probabilidad de que haya un error en la celda será:

$$\text{TCE} = 1 - (1 - P)^C$$

10.13.

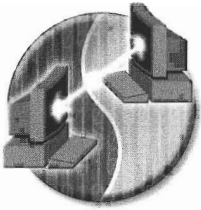
Una conexión ATM se caracteriza por su tasa de pico (PCR) cuyo inverso es un tiempo T , un δ que es el tiempo necesario para mandar una celda de 53 bytes, y un factor de tolerancia τ que es el tiempo que se admite de anticipación de que una celda puede llegar antes de lo esperado. Si $\tau > T - \delta$, más de una celda está dentro de la tasa contratada. Determinar cuántas celdas se admiten con determinado PCR si el enlace esta trabajando a tasa máxima.

SOLUCIÓN

En el algoritmo de gestión de tasa de tráfico utilizado por ATM, temporalmente se pueden cursar más celdas que las que se indican en la tasa sostenida contratada (SCR), pero no se puede exceder la tasa de pico en ningún momento. Sin embargo, si una celda mandada consume un tiempo T , tras ella deja un slot de tiempo de $T - \delta$ durante el cual se puede seguir enviando celdas, siempre que el parámetro τ lo permita.

Si se considera T y τ medidos en unidades de δ , el número de intervalos T permitidos por adelantado depende de $\tau / (T - \delta)$, luego el número de celdas consecutivas a tasa de pico PCR será:

$$N = \text{entero} \left(1 + \frac{\tau}{(T - \delta)} \right)$$



TECNOLOGÍAS LAN

11.1.

Si se dispone de dos estaciones (PCA y PCB) conectadas a través de un puente, ¿cuáles serán las direcciones MAC origen y destino de los paquetes que circulen por esa red?

SOLUCIÓN

El puente trabaja de forma transparente por lo que las direcciones MAC que nos encontraremos en los paquetes que circulan por esa red serán las correspondientes a las direcciones MAC de las propias estaciones PCA y PCB como direcciones MAC origen o destino.

11.2.

Determine el rendimiento máximo de las siguientes topologías de redes basadas en par trenzado a 10 Mbps, suponiendo tramas de 10.000 bits y 6 estaciones conectadas:

- Topología en bus.
- Topología en anillo con 100 m de circunferencia en la que sólo puede haber una trama a la vez.
- Estrella con conmutador central.

SOLUCIÓN

- $R = 10$ Mbps, ya que no hay ningún mecanismo de detección de colisión o confirmación.
- Suponemos que cada nodo transmite el paquete hacia el siguiente nodo tan pronto como lo recibe. El nodo origen para transmitir un segundo paquete deberá esperar a que llegue de vuelta el primero (de tamaño P bytes a una tasa V). Este tiempo de espera será:

$$t = t_{tx} + t_{prop}$$

$$t_{tx} = \frac{P}{V} = \frac{10 \cdot 10^3}{10 \cdot 10^6} = 10^{-3} \text{ s}$$

$$t_{prop} = \frac{D}{c} = \frac{100}{210^8} = 500 \cdot 10^{-9} \text{ s}$$

$$R = \frac{P}{t} = \frac{10 \cdot 10^3}{10^{-3} + 500 \cdot 10^{-9}} = 9,995 \text{ Mbps}$$

- c) El mejor caso será en el que haya comunicaciones dos a dos a través del conmutador, es decir, 3 comunicaciones, todas ellas a la máxima velocidad de 10 Mbps.

$$R = 30 \text{ Mbps}$$

11.3.

Se dispone de una cabecera de nivel MAC con los siguientes campos: dirección MAC origen, dirección MAC destino y control, de 6,6 y 2 bytes respectivamente. ¿En qué orden dispondría dichos campos dentro de la cabecera de nivel MAC para facilitar el diseño de puentes rápidos?

SOLUCIÓN

Para reenviar un paquete, un puente se fija en la dirección MAC destino, ignorando en la mayoría de los casos la dirección MAC origen y el campo de control. Por tanto, interesará colocar el campo dirección MAC destino en el comienzo de la cabecera de nivel MAC, para así poder reenviar el paquete tan pronto como se reciban los 6 primeros bytes (no hace falta recibir el paquete completo para empezar a reenviarlo).

En redes Ethernet esta disposición es la que se utiliza en el nivel de enlace, con la dirección MAC destino como primer campo. También se facilita el trabajo de las tarjetas de red de los equipos terminales, porque cuando leen un paquete de la red deben comprobar si va dirigido hacia ellos. Al estar la dirección MAC destino al comienzo del paquete, esta labor se facilita.

11.4.

Suponga dos redes LAN 10Base-5 unidas con un repetidor o con un puente. Determine la velocidad máxima obtenible en el conjunto y la longitud máxima alcanzable por el conjunto de ambas redes, según se encuentren unidas por un repetidor o un router.

SOLUCIÓN

Con un repetidor: $Vel_{\text{máx}} = 10 \text{ Mbps}$ (del estándar), porque el repetidor trabaja a nivel físico y se limita a regenerar señales eléctricas sin realizar ningún tipo de filtrado de paquetes a nivel de enlace.

$$Long_{\text{máx}} = 500 \text{ m entre las dos redes (del estándar)}$$

Con un puente: $Vel_{\text{máx}} = 10 + 10 = 20 \text{ Mbps}$ (suponiendo el mejor caso de comunicaciones internas dentro de cada LAN a máxima velocidad), porque únicamente saca por cada puerto los paquetes dirigidos a las máquinas que cuelgan de esos puertos. Por tanto, trabaja a nivel de enlace teniendo en cuenta las direcciones MAC de los paquetes que circulan por la red.

$$Long_{\text{máx}} = 500 \text{ m cada red (dominios de colisión diferentes)}$$

11.5.

Suponga un puente con encaminamiento estático que une 3 redes de 5 estaciones cada una. Calcule las entradas que tendrá como mínimo y máximo la tabla de reenvío de ese puente.

SOLUCIÓN

La tabla de reenvío debe almacenar las direcciones MAC de las estaciones que se encuentran en cada puerto. Al encender el puente no habrá detectado todavía ninguna estación por lo que su tamaño

será 0 (mínimo). Cuando el puente haya visto pasar tráfico con origen o destino todas las estaciones de la red, contendrá $3 \cdot 5 = 15$ entradas (máximo). El puente utilizará estas tablas para saber por qué puerto reenviar determinado paquete con cierta dirección MAC destino.

11.6. Determine si un puente con árbol de expansión funciona correctamente en las siguientes situaciones:

- a) El nodo origen nunca transmite.
- b) El nodo destino nunca transmite.
- c) El nodo origen nunca había transmitido antes.

SOLUCIÓN

- a) No hay problema, si no se transmite no hay paquetes con los que tener problemas.
- b) Tampoco hay problemas, ya que en este caso el algoritmo que sigue el puente siempre inunda porque nunca aprenderá esa dirección del nodo destino, ya que nunca ve una respuesta de éste, un paquete con la dirección MAC de éste como dirección MAC origen en el paquete.
- c) Si es la primera vez que el nodo transmite algo, se almacenará su dirección MAC en la tabla de reenvío asociada al puerto del puente en que se encuentra conectado ese nodo. Al ver ese paquete sobre la red con esa dirección como origen, se está seguro del puerto donde se encuentra conectado ese nodo.

11.7. Considere la transferencia de un fichero de 1 Mbyte entre 2 estaciones. Calcule la velocidad efectiva y el tiempo consumido para una topología en estrella con un centro conmutador y con una velocidad de transmisión en el medio de 64 Kbps, si de manera simultánea existe una transferencia de 48 Kbps entre otro par de nodos. ¿Qué ocurriría si el centro fuese concentrador?

SOLUCIÓN

Como la topología es en estrella con un conmutador, las comunicaciones entre pares de nodos diferentes no interfieren entre sí por lo que tendremos una velocidad efectiva que es la máxima disponible, 64 Kbps. El tiempo consumido en la transmisión a esta velocidad será:

$$t = \frac{P}{V} = \frac{10^6 \cdot 8}{64 \cdot 10^3} = 125 \text{ s}$$

Si el centro fuese concentrador, la velocidad se reduciría a $64 - 48 = 16$ Kbps porque el ancho de banda sería entonces compartido entre todos los flujos establecidos.

11.8. Considere una transferencia de un fichero de 1 Mbyte entre dos estaciones separadas 1 Km en una topología en anillo con una velocidad de transmisión de 1 Mbps y paquetes de 1 Kbyte. La longitud de la circunferencia del anillo es de 20 Km y existen 20 estaciones repartidas uniformemente en ese perímetro. La confirmación de recepción en el destino se realiza en la estación destino permitiendo pasar los paquetes hacia la estación origen (una vez comprobado que el paquete es correcto). Además existen tres repetidores en el anillo que introducen un retardo de un bit. Calcular la velocidad de transmisión efectiva de ese fichero.

SOLUCIÓN

El número de paquetes a transmitir para completar el fichero será:

$$n = \frac{F}{P} = \frac{10^6}{10^3} = 10^3$$

Por cada paquete necesitaremos el siguiente tiempo:

$$t = t_{tx} + t_{p1} + t_{tx} + t_{p2} + 3 t_{repetidor} = 2 t_{tx} + t_{p1} + t_{p2} + 3 t_{repetidor}$$

donde t_{tx} es el tiempo de transmisión para paquete de datos y de confirmación, t_{p1} es el tiempo de propagación del paquete de datos, t_{p2} el tiempo de propagación del paquete de confirmación y $t_{repetidor}$ es el retardo introducido por cada repetidor.

Tiempo de transmisión, tanto del paquete de datos como del de confirmación:

$$t_{tx} = 10^3 \cdot \frac{8}{10^6} = 8 \cdot 10^{-3}$$

Tiempo de propagación del paquete de datos del nodo origen al destino:

$$t_{p1} = \frac{10^3}{2 \cdot 10^8} = 5 \cdot 10^{-6}$$

Tiempo de propagación del paquete de confirmación del nodo destino al origen:

$$t_{p2} = \frac{19 \cdot 10^3}{2 \cdot 10^8} = 95 \cdot 10^{-6}$$

Retardo introducido por un repetidor:

$$t_{repetidor} = \frac{1}{10^6} = 10^{-6}$$

Sustituyendo, nos queda el tiempo necesario para transmitir un paquete:

$$t = 2 \cdot 8 \cdot 10^{-3} + 5 \cdot 10^{-6} + 95 \cdot 10^{-6} + 3 \cdot 10^{-6} = 16,103 \cdot 10^{-3} \text{ s}$$

Como los paquetes de datos son de 1 Kbyte, la velocidad de transmisión efectiva resultante será:

$$R = \frac{10^3 \cdot 8}{16,103 \cdot 10^{-3}} = 496,80 \text{ Kbps}$$

11.9.

Considere un bus en banda base con 19 estaciones equidistantes, una velocidad de transmisión de 10 Mbps y una longitud del bus de 20 Km.

- Calcule el tiempo medio para enviar una trama de 1.000 bits a otra estación, medido desde el comienzo de la transmisión hasta el final de la recepción.
- Si dos estaciones consecutivas comienzan a transmitir exactamente al mismo tiempo, sus paquetes interfieren entre sí. Si cada estación transmisora monitoriza el bus durante la transmisión, ¿cuánto tiempo tardarán en darse cuenta de esta colisión? ¿Cuántos bits habrán transmitido para entonces?

SOLUCIÓN

- a) El tiempo total de recepción de una trama vendrá dado por:

$$t = t_{tx} + t_p + t_{rx}$$

Según la distancia a la que se encuentre la estación destino tendremos valores diferentes de tiempo de propagación. Para calcular el tiempo medio necesitaremos el tiempo medio de propagación que vendrá dado por la distancia media entre estaciones:

$$D_{media} = \frac{1 + 20}{2} = 10,5 \text{ Km}$$

$$t_{p-media} = \frac{10,5 \cdot 10^3}{2 \cdot 10^8} = 52,5 \cdot 10^{-6}$$

$$t_{tx} = t_{rx} = \frac{1.000}{10 \cdot 10^6} = 100 \cdot 10^{-6}$$

El tiempo medio resultante será:

$$t_{medio} = 100 \cdot 10^{-6} + 52,5 \cdot 10^{-6} + 100 \cdot 10^{-6} = 252,5 \cdot 10^{-6} \text{ s}$$

- b) Si ambas estaciones comienzan a transmitir a la vez y se encuentran separadas 1 Km, el tiempo que tardará en llegar la señal de la una a la otra será el tiempo de propagación para esa distancia:

$$t_p = \frac{10^3}{2 \cdot 10^8} = 5 \cdot 10^{-6} \text{ s}$$

Éste será el tiempo que tarden las estaciones en darse cuenta de la colisión tras comenzar su transmisión. Los bits que habrán transmitido para entonces:

$$bits = \frac{t_p}{t_{bit}} = t_p \cdot V = 5 \cdot 10^{-6} \cdot 10 \cdot 10^6 = 50 \text{ bits}$$

11.10.

¿Cuál es el máximo tamaño de trama posible en una red en anillo de 1 Km de longitud circular y 1 Mbps de velocidad de transmisión, si la propia estación transmisora es la encargada de eliminar la trama del anillo y para ello no puede estar transmitiendo a la vez que recibe?

SOLUCIÓN

Mientras se está recibiendo la trama de vuelta se podrá comenzar a transmitir una nueva trama, por lo que el tamaño máximo de trama vendrá determinado exclusivamente por el tiempo de propagación de los bits en esa red:

$$P = t_p \cdot V = \frac{L}{c} \cdot V = \frac{10^3}{2 \cdot 10^8} \cdot 10^6 = 5 \text{ bits}$$

11.11.

Suponga una red inalámbrica 802.11b que ofrece 6 Mbps de velocidad de transmisión por canal. Si en la figura tenemos una comunicación A-B y otra C-A, y suponiendo que el acceso al medio es justo, ¿qué velocidad alcanzará cada flujo?

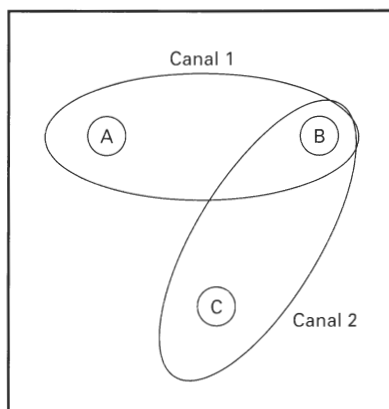


Figura 11.1. Red inalámbrica.

SOLUCIÓN

La comunicación A-B será directa, y la comunicación C-A seguirá un primer salto C-B en el Canal 2 y otro salto B-A en el Canal 1. Por tanto, en el Canal 1 tenemos ambas comunicaciones A-B y C-A. Este Canal 1 será el cuello de botella y por tanto habrá que repartir equitativamente el ancho de banda entre las dos, obteniendo $6/2 = 3$ Mbps. Ambas comunicaciones se limitarán a 3 Mbps suponiendo un reparto equitativo. Aunque en el tramo C-B haya de esta forma recursos disponibles, no se utilizarán porque el buffer en el nodo B va a ser siempre muy limitado, y por la propiedad conservativa el flujo tendrá a estabilizarse al valor del cuello de botella.

11.12.

Si se dispone de un concentrador (hub) de 12 puertos a 10 Mbps y un conmutador (switch) de 12 puertos a 10 Mbps, ¿cuál será la velocidad máxima que necesitaremos en un nuevo puerto de subida para ambos equipos?

SOLUCIÓN

Concentrador: 10 Mbps, porque es un medio compartido y no va a generarse más tráfico entre todos los puertos que los 10 Mbps. Internamente entre los puertos, el tráfico total generable es de 10 Mbps, por lo que en el peor caso todo este tráfico será de salida y por tanto nunca superior a 10 Mbps.

Conmutador: 120 Mbps, porque en el caso extremo las máquinas conectadas a cada puerto accederán a la máxima velocidad a la troncal, y como no interfieren entre sí, este valor de acceso a la troncal es de $12 \cdot 10$ Mbps, cada puerto consumiendo su velocidad para salida a la troncal.

11.13.

Cite las ventajas e inconvenientes que tiene el utilizar un tamaño de paquete grande en una LAN.

SOLUCIÓN

Ventajas: El utilizar un tamaño de paquete grande en una LAN supone una mayor utilización porque se reduce el tiempo de guarda y la duración de los intervalos de contienda por byte de datos. También el rendimiento de la red será mejor, porque se tendrá menor porcentaje del paquete dedi-

cado a cabeceras de protocolos y por tanto, mayor porcentaje del paquete dedicado a la transmisión de datos útiles.

Inconvenientes: Al aumentar el tamaño del paquete se produce un mayor retardo medio. Si un nodo transmite continuamente, el resto de nodos tendrá que esperar más para poder hacer uso del canal. Este hecho se agudiza en casos de utilización alta de la red porque además se le sumarán fenómenos como la existencia de colisiones.

11.14.

Sea una red basada en un protocolo parada y espera (stop & wait) para comunicar dos nodos con una velocidad de transmisión de 128 Kbps, tamaño de paquete 1.000 bits y 10 ms de propagación extremo a extremo. Calcule el rendimiento máximo conseguible.

SOLUCIÓN

En un protocolo de parada y espera, cuando un nodo transmite un paquete se ha de esperar a recibir la confirmación antes de mandar el siguiente. Por tanto, el tiempo invertido en la transmisión del paquete incluye dos tiempos de propagación, uno de ida del paquete de datos y otro de vuelta del paquete de confirmación. Consideramos el tamaño del paquete de confirmación despreciable y por tanto también despreciable el tiempo de transmisión de este paquete.

$$R = \frac{t_{tx}}{t_{tx} + 2t_p} = 0,7961 \rightarrow 79,61\%$$

$$T_{tx} = \frac{P}{V} = \frac{1.000}{128 \cdot 10^3} = 7,81 \cdot 10^{-3}$$

$$t_p = 10 \text{ ms}$$

11.15.

Suponga una configuración de red como la de la figura y escriba la tabla final de reenvío que alcanzarán los puentes P1 y P2, si A, B, C, D, E, F y G son estaciones que se intercomunican entre sí todas con todas. ¿Qué pasaría si las estaciones E y G fuesen la misma, es decir, corresponderían a dos interfaces de la misma estación? ¿Qué ocurriría si se añadiera un tercer puerto al puente P1 conectado a la red de la máquinas F y G?

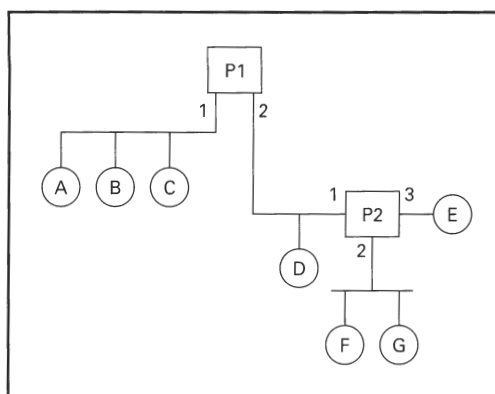


Figura 11.2. Redes unidas por puentes.

SOLUCIÓN

Las tablas de reenvío consisten en especificar el puerto de salida para cada paquete con determinada dirección MAC destino. Estas tablas serán:

Puente P1		Puente P2	
Dirección MAC	Puerto	Dirección MAC	Puerto
A	1	A	1
B	1	B	1
C	1	C	1
D	2	D	1
E	2	E	3
F	2	F	2
G	2	G	2

Tabla 11.1. Tablas de reenvío de los puentes.

Si E y G fuesen interfaces de la misma estación no pasaría nada extraordinario porque las comunicaciones se seguirían refiriendo a uno u otro interfaz, siempre que esta estación con dos interfaces no reenviara paquetes como un puente. Si hiciese de puente, entonces sí podría haber problemas al formarse un ciclo con el puente P2. Un algoritmo de reenvío correcto solucionaría este problema de los ciclos.

Si se añade un tercer puerto a P1 conectado a la red de las máquinas F y G podrían producirse otra vez ciclos. Los puentes deberían aplicar algún algoritmo para evitarlos (por ejemplo, intercambiarse mensajes para obtener el árbol de expansión de mínimo coste).

11.16.

Sea una red de R bps que une dos nodos, origen y destino, a través de N nodos intermedios, de forma que un paquete se envía primero del nodo origen al primer nodo intermedio, entonces el primer nodo intermedio una vez recibido el paquete completo, lo reenvía al segundo nodo intermedio y así sucesivamente. Calcular el valor del tamaño de los paquetes T a los que se deben añadir 20 bytes de cabeceras que minimiza el tiempo empleado en la transferencia de un fichero de tamaño F del nodo origen al destino, siendo este tamaño de fichero F mucho mayor que el tamaño del paquete T .

SOLUCIÓN

El tiempo de transmisión de un paquete es: $T + 20 / R$.

Si $F \gg T$, la transmisión del fichero consume F/T paquetes en cada salto, pero la transmisión entre nodos intermedios se encuentra solapada, de forma que cuando el nodo origen envía el segundo paquete, el primer nodo intermedio manda el primer paquete hacia el segundo nodo intermedio.

Por tanto, tendremos que esperar el reenviar $F/T + N$ paquetes, con lo que el tiempo total será:

$$t = \left(\frac{F}{T} + N \right) \left(\frac{T + 20}{R} \right)$$

Como queremos calcular el valor de T que minimiza este tiempo, derivamos con respecto a R e igualamos a 0:

$$\frac{dt}{dR} = 0 = \frac{-F}{R^2} \left(\frac{T + 20}{R} \right) + \left(\frac{F}{R} + N \right) \frac{1}{R}$$

Despejando T :

$$T = \sqrt{\frac{20 F}{N}}$$

11.17. Sea un enlace satélite con un retardo de propagación de 290 ms a cada una de las estaciones terrenas que une y una velocidad de transmisión de 1 Mbps. Determine qué protocolo de acceso al medio se podría utilizar: CSMA / CD y/o ALOHA.

SOLUCIÓN

El método CSMA / CD no sería recomendable porque exige un tamaño mínimo de los paquetes tal que el tiempo de transmisión sea $t_{tx} > 2t_p$, siendo t_p el tiempo de propagación. Por tanto:

$$2 \cdot 290 \cdot 10^{-3} < \frac{P}{10^6} \rightarrow P = 580.000 \text{ bits} = 72.500 \text{ bytes}$$

Este tamaño de paquete sería demasiado grande: supondría mucho tiempo de paquetización, elevado retardo al tener que esperar transmisiones de otros usuarios y mucha pérdida de eficiencia del canal en caso de colisión, por lo que no es recomendable utilizar CSMA / CD.

En cambio, el mecanismo ALOHA se podría utilizar porque no tiene ninguna exigencia parecida a la de CSMA / CD, debido a que no pretende realizar ninguna detección de colisión. La eficiencia de ALOHA será inferior, pero permitirá su utilización en entornos como el descrito.

11.18. Suponga una red en la que las estaciones conectadas usan un protocolo de reserva de forma que para cada paquete que deseen mandar deben solicitar una reserva que consume K segundos por un canal de señalización que se supone sin colisiones.

- ¿Cuál será la máxima eficiencia de esa red?
- ¿Y si se solicita la reserva de N paquetes por cada mensaje de señalización?

SOLUCIÓN

- Suponiendo que el retardo de propagación es despreciable y el tiempo de transmisión del paquete de datos es t_{tx} , la transmisión de cada paquete de datos requerirá $t_{tx} + K$ segundos. El rendimiento máximo se obtendrá cuando todas las estaciones quieran transmitir y lo hagan una tras la otra:

$$E_{\text{máx}} = \frac{t_{tx}}{t_{tx} + K}$$

- b) Si se reserva tiempo para transmitir N paquetes con cada mensaje de señalización, en el mejor de los casos la eficiencia será:

$$E_{\text{máx}} = \frac{N \cdot t_{\text{tx}}}{N \cdot t_{\text{tx}} + K} = \frac{t_{\text{tx}}}{t_{\text{tx}} + K/N}$$

Es decir, será mejor para este último caso.

11.19.

Considere una red que utiliza multiplexación temporal TDM como control de acceso al medio. Suponga que las tramas son de tamaño P , que existen N nodos en esa red, L es la longitud entre los puntos más lejanos y la velocidad de transmisión es R Mbps.

- a) Calcule la tasa de transmisión efectiva.
b) ¿Qué ocurre con un número elevado de nodos o una distancia máxima entre nodos grande?

SOLUCIÓN

- a) Para una transmisión efectiva de una trama, el tiempo de slot TDM ha de ser como mínimo igual a este tamaño de trama para permitir la transmisión de una trama completa en cada slot. Por tanto, sin colisión el tiempo de slot será:

$$t_{\text{slot}} = \frac{L}{c'} + \frac{P}{R}$$

Donde c' es la velocidad de propagación en esa red ($2 \cdot 10^8$ m/s).

Para que N nodos puedan transmitir serán necesarios N slots dentro de la supertrama TDM, y por tanto será necesario esperar cada vez el paso de los $N-1$ slots para poder transmitir en el propio.

$$R_{\text{efectivo}} = \frac{P}{N \cdot T} = \frac{P}{N \left(\frac{L}{c'} + \frac{P}{R} \right)}$$

- b) Si se eleva el número de nodos o la longitud de la red, la tasa de transmisión efectiva decae en gran medida. Por tanto, este tipo de acceso al medio no será recomendable para interconectar un gran número de nodos y a grandes distancias.

11.20.

Determinar el tamaño de fichero de M bits que hace que sea más rápido utilizar una conexión virtual ATM (con un tiempo de establecimiento correspondiente a E bits) comparado con una transmisión de paquetes IP de tamaño constante (considerando sólo el tamaño de la cabecera IP y no de niveles inferiores). Suponer una velocidad de transmisión de R bps.

SOLUCIÓN

La transmisión con paquetes IP no posee tiempo de establecimiento como el circuito virtual ATM, pero sin embargo las cabeceras en cada paquete son muy superiores al tamaño de las cabeceras de las celdas ATM.

Consideremos:

- D_{ATM} : bytes de datos de la celda ATM (48 bytes).
- C_{ATM} : bytes de la cabecera de la celda (3 bytes).
- D_{IP} : bytes de datos del paquete IP.
- C_{IP} : bytes de la cabecera del paquete IP (20 bytes aproximadamente).

El tiempo consumido en la transmisión del fichero se compondrá de un tiempo de establecimiento de la conexión y otro de transmisión de las celdas correspondientes. El número de celdas será el entero redondeado por arriba del cociente $M / 8D_{ATM}$:

$$t_{ATM} = \frac{E}{v} + \left\lceil \frac{M}{8 \cdot D_{ATM}} \right\rceil \frac{8(D_{ATM} + C_{ATM})}{v}$$

Para la transmisión IP el tiempo consumido será:

$$t_{IP} = \left\lceil \frac{M}{8 \cdot D_{IP}} \right\rceil \frac{8(D_{IP} + C_{IP})}{v}$$

Será mejor el circuito virtual ATM cuando se cumpla que su tiempo de transmisión sea menor, es decir:

$$\frac{E}{v} + \left\lceil \frac{M}{8 \cdot D_{ATM}} \right\rceil \frac{8(D_{ATM} + C_{ATM})}{v} < \left\lceil \frac{M}{8 \cdot D_{IP}} \right\rceil \frac{8(D_{IP} + C_{IP})}{v}$$

Despejando el tamaño del fichero M :

$$M < \frac{E}{\frac{C_{IP}}{D_{IP}} - \frac{C_{ATM}}{D_{ATM}}}$$

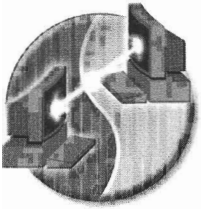
11.21.

¿Por qué la detección de colisiones no funciona correctamente en redes inalámbricas?

SOLUCIÓN

Por un lado, el equipo de radio no puede transmitir a la vez que se recibe, por limitaciones del circuito electrónico de transmisión/recepción.

Por otro lado, los nodos que conforman la red inalámbrica no están completamente conectados todos con todos. De forma que una transmisión generada por un nodo puede no ser escuchada por un nodo en el otro extremo, pero sí por un nodo intermedio a medio camino entre éstos. Si los nodos extremos transmiten a la vez, el nodo intermedio verá la colisión pero los nodos extremos no porque la señal del otro nodo no llega o llega muy atenuada.



SISTEMAS LAN

12.1.

Sea una red como la de la figura, compuesta por dos segmentos, uno inalámbrico y el otro un bus de cable. Se interconectan mediante un repetidor que funciona de forma transparente. La velocidad de transmisión en las dos redes es de 10 Mbps. La velocidad de propagación en el bus es de $2 \cdot 10^8$ m/s y la velocidad de propagación de las ondas de radio es $3 \cdot 10^8$ m/s. La máxima distancia que admite el bus es de 500 m y la máxima distancia desde un terminal inalámbrico al repetidor inalámbrico es de 1 Km.

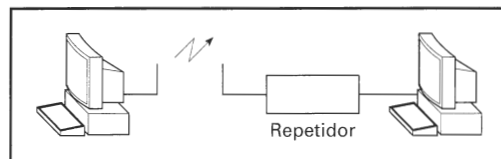


Figura 12.1. Esquema de red.

Sobre esta red se pretende emplear un protocolo de envío con confirmación: la estación emisora envía paquetes de datos de 1.000 bytes y no volverá a enviar un paquete hasta que no haya recibido la confirmación del anterior.

- ¿Cuál es la máxima velocidad de transmisión que se puede alcanzar entre un terminal inalámbrico y uno fijo en el peor caso?
- ¿Cuál es la longitud mínima que deben tener los paquetes de confirmación si se utiliza un protocolo de acceso al medio CSMA / CD?

SOLUCIÓN

- Peor caso:* Transmisión de las estaciones más lejanas inalámbrica y fija.

El tiempo total de transmisión del paquete de datos y vuelta de la confirmación será (el t_{tx} de la confirmación es despreciable):

$$t_{total} = t_{tx} + t_{p1} + t_{p2} + t_{p1} + t_{p2} = t_{tx} + 2 \cdot t_{p1} + 2 \cdot t_{p2}$$

t_{tx} = tiempo transmisión paquete de datos:

$$t_{tx} = \frac{1.000 \cdot 8}{10 \cdot 10^6}$$

t_{p1} = tiempo propagación en el aire:

$$t_{p1} = \frac{1.000}{3 \cdot 10^8}$$

t_{p12} = tiempo propagación en el bus:

$$t_{p2} = \frac{500}{2 \cdot 10^8}$$

Finalmente:

$$t_{\text{total}} = 8,1166 \cdot 10^{-4} \text{ s}$$

Para la velocidad máxima, en cuanto se recibe la confirmación de un paquete se manda el siguiente, por lo que:

$$V_{\text{tx}} = \frac{P}{t_{\text{total}}} = \frac{1.000 \cdot 8}{8,1166 \cdot 10^{-4}} = 9,8536 \text{ Mbps}$$

b) *Tamaño mínimo de la confirmación.*

En CSMA / CD para detectar la colisión la máquina que transmite el paquete ha de estar transmitiendo durante el tiempo mínimo en el que se puede producir la colisión. Este tiempo es de dos veces el tiempo de propagación (el peor caso de colisión es que se produzca cuando el paquete esté a punto de llegar al receptor y justo en ese instante el receptor mande otro paquete que llegaría al receptor otro tiempo de propagación después, en total $2 t_p$).

El tiempo de propagación peor resulta ser entre dos inalámbricos lo mas alejados, que será de $1 + 1 = 2 \text{ Km}$.

$$t_p = \frac{2.000}{3 \cdot 10^8}$$

$$t_{\text{tx}} = 2 \cdot t_p = 13,33 \cdot 10^{-6} \text{ s}$$

El tamaño del paquete de confirmación será:

$$P_{\text{confirmación}} = V_{\text{tx}} \cdot t_{\text{tx}} = 10 \cdot 10^6 \cdot 13,33 \cdot 10^{-6} = 133,3 \text{ bits} = 16,6 \text{ bytes} \approx 17 \text{ bytes}$$

12.2.

En un edificio se han implementado tres LANs interconectadas entre sí. El esquema de la conexión es el siguiente:

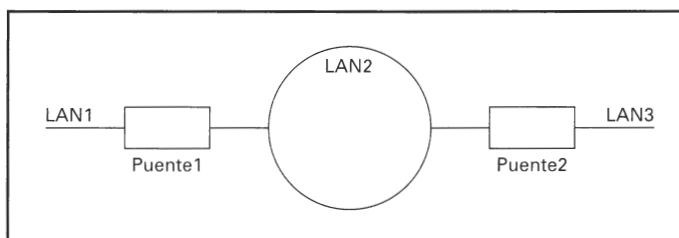


Figura 12.2. Esquema de red.

La red LAN2 es una red *token ring*. La longitud física del anillo es 500 m, y los puentes distan 250 m de cable entre sí. La velocidad de transmisión en esta red es de 16 Mbps.

Las redes LAN1 y LAN3 son Ethernet 10BASE-T de 100 m de longitud. Los puentes almacenan y reexpiden las tramas. Las estaciones origen y destino se colocan en los puntos extremos de la red, y no existen más estaciones conectadas.

Para asegurar un correcto funcionamiento entre las dos estaciones, éstas transmiten un paquete y no transmiten otro hasta que han recibido el ACK correspondiente, es decir, se utiliza un protocolo de confirmación. La longitud de los paquetes es de 1.000 bytes.

- Calcular la velocidad de transmisión efectiva en el mejor y peor de los casos.
- Describir cómo se podría mejorar el protocolo (manteniendo la fiabilidad) para conseguir una mayor velocidad de transmisión efectiva.

SOLUCIÓN

a) *Mejor caso*: Que los puentes tengan el testigo cuando quieran transmitir.

El tiempo total de transmisión del paquete y recepción de la confirmación será:

t_{tx-eth} = tiempo transmisión paquete de datos en la Ethernet:

$$t_{tx} = \frac{1.000 \cdot 8}{10 \cdot 10^6}$$

t_{tx-tr} = tiempo transmisión paquete de datos en la *token ring*:

$$t_{tx} = \frac{1.000 \cdot 8}{16 \cdot 10^6}$$

t_{p-eth} = tiempo propagación en la Ethernet:

$$t_{p2} = \frac{100}{2 \cdot 10^8}$$

t_{p-eth} = tiempo propagación en la *token ring*:

$$t_{p2} = \frac{250}{2 \cdot 10^8}$$

$$t_{total} = t_{datos} + t_{confirmación}$$

$$t_{datos} = t_{tx-eth} + t_{p-eth} + t_{tx-tr} + t_{p-tr} + t_{tx-eth} + t_{p-eth}$$

$$t_{confirmación} = t_{p-eth} + t_{p-tr} + t_{p-eth}$$

Con lo que:

$$t_{total} = 2t_{tx-eth} + t_{tx-tr} + 4t_{p-eth} + 2t_{p-tr} = 2,1045 \cdot 10^{-3} \text{ s}$$

$$V_{tx} = \frac{P}{t_{total}} = \frac{1.000 \cdot 8}{2,1045 \cdot 10^{-3}} = 3,80 \text{ Mbps}$$

- b) *Peor caso*: Que los puentes acaben de dejar el testigo justo antes de cuando quieran transmitir y por tanto han de esperar a que les llegue de nuevo, lo cual es el tiempo de propagación del testigo a lo largo de todo el anillo (500 m) si ningún nodo más del anillo quiere transmitir.

El tiempo de espera del testigo será:

$$T_{\text{testigo}} = \frac{500}{2 \cdot 10^8}$$

Por lo que el tiempo total quedará:

$$t_{\text{total}} = t_{\text{datos}} + t_{\text{confirmación}}$$

$$t_{\text{datos}} = t_{\text{tx-eth}} + t_{\text{p-eth}} + t_{\text{testigo}} + t_{\text{tx-tr}} + t_{\text{p-tr}} + t_{\text{tx-eth}} + t_{\text{p-eth}}$$

$$t_{\text{confirmación}} = t_{\text{p-eth}} + t_{\text{testigo}} + t_{\text{p-tr}} + t_{\text{p-eth}}$$

$$t_{\text{total}} = 2t_{\text{tx-eth}} + t_{\text{tx-tr}} + 4t_{\text{p-eth}} + 2t_{\text{p-tr}} + 2t_{\text{testigo}} = 2,1095 \cdot 10^{-3} \text{ s}$$

$$V_{\text{tx}} = \frac{P}{t_{\text{total}}} = \frac{1.000 \cdot 8}{2,1095 \cdot 10^{-3}} = 3,79 \text{ Mbps}$$

- c) *Posible mejora del protocolo*: Que se mande el paquete de confirmación no con cada paquete transmitido, sino por cada N paquetes de datos mandados. Cuanto mayor sea N mayor será la eficiencia pero también los paquetes erróneos se recuperarán con más retraso.

12.3.

Las redes inalámbricas 802.11b poseen un modo de funcionamiento llamado infraestructura, donde todas las comunicaciones entre nodos inalámbricos han de pasar necesariamente a través de un punto de acceso que hace las funciones de coordinación central. El estándar habla de velocidades de transmisión de 11 Mbps, pero como se trata de un protocolo con confirmación, cada paquete que un nodo manda al punto de acceso debe confirmarse a nivel MAC.

- a) Estime la máxima y mínima velocidad aprovechable por un protocolo de red que se encapsulase por encima y que se puede obtener con un tamaño máximo de paquete de 1.500 bytes, mínimo de 64 bytes y una distancia máxima entre nodo y estación base de 150 m.
- b) ¿Y si la distancia fuera de 50 Km?

SOLUCIÓN

- a) El tiempo invertido en la transmisión de un paquete de un nodo a otro vendrá dado por la siguiente expresión teniendo en cuenta que forzosamente ha de pasarse a través de la estación base para toda comunicación entre nodos inalámbricos:

$$t_{\text{total}} = t_{\text{datos}} + t_{\text{confirmación}}$$

$$t_{\text{datos}} = t_{\text{tx}} + t_{\text{p}} + t_{\text{tx}} + t_{\text{p}} = 2t_{\text{tx}} + 2t_{\text{p}}$$

$$t_{\text{confirmación}} = t_{\text{p}} + t_{\text{p}} \text{ (tiempo de transmisión de la confirmación despreciable)}$$

$$t_{\text{total}} = 2t_{\text{tx}} + 4t_{\text{p}}$$

$$t_{\text{tx}} = \text{tiempo transmisión paquete de datos}$$

En el mejor caso, los dos nodos se encontrarán muy cerca de la estación base (0 m) y se transmitirán paquetes de tamaño mínimo (64 bytes), con lo que:

$$t_p = 0$$

$$t_{tx} = \frac{64 \cdot 8}{11 \cdot 10^6} = 46,54 \cdot 10^{-6}$$

$$t_{total} = 2t_{tx} = 93,09 \cdot 10^{-6}$$

$$V_{tx} = \frac{P}{t_{total}} = \frac{64 \cdot 8}{93,09 \cdot 10^{-6}} = 5,50 \text{ Mbps}$$

En el peor caso, los dos nodos se encontrarán lo más lejos posible de la estación base (150 m) y se transmitirán paquetes de tamaño máximo (1.500 bytes), con lo que:

$$t_p = \frac{150}{3 \cdot 10^8} = 500 \cdot 10^{-9}$$

$$t_{tx} = \frac{1.500 \cdot 8}{11 \cdot 10^6} = 1,09 \cdot 10^{-3}$$

$$t_{total} = 2t_{tx} + 4t_p = 2,18 \cdot 10^{-3}$$

$$V_{tx} = \frac{P}{t_{total}} = \frac{1.500 \cdot 8}{2,18 \cdot 10^{-3}} = 5,50 \text{ Mbps}$$

Es decir, la limitación del protocolo viene impuesta en mayor medida por la necesidad de tener que hacer una doble transmisión del paquete a través de los saltos intermedios que no de otros factores como distancias o tamaño de paquete.

b) Si la distancia fuera de 50 Km en el peor caso:

$$t_p = \frac{50 \cdot 10^3}{3 \cdot 10^8} = 166,66 \cdot 10^{-6}$$

que es despreciable comparado con t_{tx} con lo que el resultado de V_{tx} será el mismo.

12.4.

¿Por qué el tamaño mínimo de paquete posible en una red Ethernet 10BaseT es de 64 bytes? Demuéstrelo.

SOLUCIÓN

Una Ethernet 10BaseT tiene una velocidad de transmisión de 10 Mbps y la máxima distancia alcanzable son 5 Km según el estándar. Como está basado en un método de acceso al medio CSMA / CD, para detectar las colisiones es necesario que la transmisión de cada paquete dure al menos el doble del tiempo de propagación extremo a extremo de ese paquete.

$$t_p = \frac{2,5 \cdot 10^3}{2 \cdot 10^8} = 12,5 \cdot 10^{-6}$$

$$t_{tx} > 2t_p = 25 \cdot 10^{-6}$$

Longitud de trama mínima:

$$P = t_{tx} \cdot V_{tx} = 25 \cdot 10^{-6} \cdot 10 \cdot 10^6 = 250 \text{ bits} = 32,25 \text{ bytes}$$

Por seguridad se toma el doble, que son los 64 bytes que fija el estándar como mínimo tamaño de paquete.

12.5.

El funcionamiento de las redes LAN se basa en tener sólo una trama a la vez en la red mientras que en redes WAN se hace necesario colocar varias simultáneamente para aprovechar el canal y maximizar la utilización. Comparar la utilización en una red ATM, con celdas de 53 bytes, 155 Mbps y 50 Km de longitud para los dos casos:

- Un sólo paquete simultáneamente en la red.
- Todos los paquetes posibles simultáneamente en la red.

SOLUCIÓN

Tiempo de transmisión:

$$t_{tx} = \frac{P}{R} = \frac{53 \cdot 8}{155 \cdot 10^6} = 2,73 \cdot 10^{-6}$$

Tiempo de propagación:

$$t_p = \frac{L}{c} = \frac{50 \cdot 10^3}{2 \cdot 10^8} = 250 \cdot 10^{-6}$$

Factor a de relación t_{tx} y t_p :

$$a = \frac{t_p}{t_{tx}} = \frac{250 \cdot 10^{-6}}{2,73 \cdot 10^{-6}} = 91,57$$

- Utilización para un sólo paquete simultáneamente en la red:

$$U = \frac{t_{tx}}{t_{tx} + t_p} = \frac{1}{1 + a} = 0,01$$

- Utilización para todos los paquetes posibles simultáneamente en la red:

$$U = \frac{at_{tx}}{at_{tx} + t_p} \approx 1$$

12.6.

Comparar la utilización efectiva de un LAN con protocolo confirmado con otra con protocolo semidúplex sin confirmación.

SOLUCIÓN

Protocolo confirmado: Antes de mandar un segundo paquete hay que esperar que vuelva la confirmación del primero.

$$U_{\text{confirmado}} = \frac{t_{tx}}{t_{tx} + 2t_p} = \frac{1}{1 + 2a}$$

Protocolo semidúplex: Se mandan los paquetes seguidos uno detrás de otros, pero sólo uno en cada momento sobre la red.

$$U_{\text{semidúplex}} = \frac{t_{\text{tx}}}{t_{\text{tx}} + t_{\text{p}}} = \frac{1}{1 + 2a}$$

Relación confirmado-semidúplex: $\frac{1 + a}{1 + 2a}$

Si $a \downarrow \ll 1$: relación = 1 $\rightarrow U_{\text{confirmado}} \approx U_{\text{semidúplex}}$

Si $a \uparrow \gg 1$: relación = 1/2 $\rightarrow U_{\text{confirmado}} \approx U_{\text{semidúplex}}/2$

12.7.

Un inconveniente de la competición por el canal en redes LAN es el desaprovechamiento de la capacidad debido al hecho de que varias estaciones intentan acceder simultáneamente al canal. Supongamos que se divide el tiempo en ranuras distintas, con cada una de las N estaciones intentando transmitir con probabilidad P en cada ranura. ¿Qué fracción de ranuras se desaprovecha debido a colisiones?

SOLUCIÓN

Ranuras desaprovechadas = P (ranuras con tx de más de una estación) = $1 - P$ (tx de una estación).

De N estaciones queremos que una cualquiera transmita, por lo que aplicando combinaciones de N elementos tomados de uno en uno quedan $\binom{N}{1}$ posibilidades.

$$P \text{ (tx de una estación)} = \binom{N}{1} P^1 (1 - P)^{N-1} = NP (1 - P)^{N-1}$$

Por tanto:

$$\text{Ranuras desaprovechadas} = 1 - NP(1 - P)^{N-1}$$

12.8.

Suponga la transmisión de un fichero de un millón de caracteres. Calcular el tiempo total de transmisión del fichero para las siguientes dos redes LAN basadas en una topología en bus con dos estaciones separadas una distancia $D = 1$ Km, velocidad de transmisión V , tamaño de paquete $P = 256$ bits (incluidos los 80 bits de cabecera). Cada paquete se confirma con un paquete de 88 bits antes de enviar el siguiente paquete.

- $V = 1$ Mbps.
- $V = 10$ Mbps.
- Anillo de longitud de circunferencia $2D$, estaciones separadas $D = 1$ Km y $V = 1$ Mbps. La confirmación la produce la estación destino dejando circular el paquete hasta que retorna al origen. El anillo tiene $N = 100$ repetidores repartidos uniformemente y cada uno produce un retraso de un bit.

SOLUCIÓN

El tiempo de propagación tanto del paquete de datos como del de confirmación:

$$t_p = \frac{D}{c} = \frac{10^3}{2 \cdot 10^8} = 5 \cdot 10^{-6}$$

El tiempo de transmisión del paquete de datos:

$$t_{\text{tx-datos}} = \frac{P}{V} = \frac{256}{V}$$

El tiempo de transmisión del paquete de confirmación (no despreciable):

$$t_{\text{tx-conf}} = \frac{P}{V} = \frac{88}{V}$$

El tiempo total por paquete de datos será:

$$t_{\text{paq}} = t_{\text{datos}} + t_{\text{confirmación}}$$

$$t_{\text{datos}} = t_{\text{tx-datos}} + t_p$$

$$t_{\text{confirmación}} = t_{\text{tx-conf}} + t_p$$

$$t_{\text{paq}} = t_{\text{tx-datos}} + t_{\text{tx-conf}} + 2 t_p$$

Resolviendo para los distintos casos tendremos:

a) $V = 1 \text{ Mbps}$

$$t_{\text{paq}} = 354 \cdot 10^{-6}$$

$$t_{\text{fichero}} = t_{\text{bit}} \cdot \text{bits}_{\text{fichero}} = \frac{354 \cdot 10^{-6}}{256 - 88} (10^6 \cdot 8) = 16,09 \text{ s}$$

b) $V = 10 \text{ Mbps}$

$$t_{\text{paq}} = 44,4 \cdot 10^{-6}$$

$$t_{\text{fichero}} = t_{\text{bit}} \cdot \text{bits}_{\text{fichero}} = \frac{44,4 \cdot 10^{-6}}{256 - 88} (10^6 \cdot 8) = 2,01 \text{ s}$$

Cabe destacar que el aumentar la velocidad de transmisión por 10 no significa que se consiga mandar el fichero 10 veces más rápido debido a la penalización del protocolo confirmado.

c) Anillo

$$t_{\text{repetidor}} = \frac{(N/2) \cdot 1}{V} = \frac{50}{10^6} = 50 \cdot 10^{-6}$$

$$t_{\text{paq}} = t_{\text{tx-datos}} + t_{\text{tx-conf}} + 2 t_p + 2 t_{\text{repetidor}} = 454 \cdot 10^{-6}$$

$$t_{\text{fichero}} = t_{\text{bit}} \cdot \text{bits}_{\text{fichero}} = \frac{454 \cdot 10^{-6}}{256 - 88} (10^6 \cdot 8) = 21,61 \text{ s}$$

- a) Si K es la probabilidad de que una estación quiera transmitir en un intervalo 2τ , determinar la probabilidad P de que en dicho intervalo exclusivamente transmita una estación y por tanto no haya colisiones.
- b) Valor de K en función del número de nodos que optimiza la probabilidad P , y el valor P máximo obtenible. ¿Qué valor se obtiene de probabilidad P para un número de estaciones N muy elevado?
- c) Evaluar el número medio de intervalos 2τ necesarios para conseguir una transmisión exitosa.
- d) Considerando paquetes de 127 bytes de longitud media, $V = 10$ Mbps, $N = 50$, $d = 1$ Km, $\tau = 5$ ms/Km. Determinar el tiempo medio necesario para la transmisión de cada paquete que optimiza la probabilidad P .
- e) Eficiencia del protocolo.
- f) Número medio de paquetes M que por unidad de tiempo puede generar cada estación.

SOLUCIÓN

- a) 2τ es el período en el que sólo tiene que transmitir una estación para evitar la colisión y por tanto se transmita con éxito. De N estaciones queremos que una cualquiera transmita en ese período, por lo que aplicando combinaciones de N elementos tomados de uno en uno quedan $\binom{N}{1}$ posibilidades.

$$P = \binom{N}{1} K^1 (1 - K)^{N-1} = NK (1 - K)^{N-1}$$

- b) Para maximizar la expresión anterior derivamos P en función de K , igualamos a 0 y tenemos:

$$\frac{dP}{dK} = 0 = N \frac{d}{dK} [K (1 - K)^{N-1}] = N [(1 - K)^{N-1} - K (N - 1) (1 - K)^{N-2}]$$

Despejando: $K = 1/N$

Y sustituyendo el valor máximo de P será:

$$P_{\text{máx}} \left(1 - \frac{1}{N}\right)^{N-1}$$

$$\text{Si } N \rightarrow \infty: \lim_{N \rightarrow \infty} P_{\text{máx}} = \frac{1}{e} = 0,37$$

- c) El número medio de intervalos 2τ , n , para conseguir transmisión exitosa vendrá de promediar para todas las combinaciones de ocurrencias posibles: transmisión exitosa en el primer intervalo, en el segundo..., según las probabilidades calculadas antes:

$$n = \sum_{i=1}^{\infty} i P (1 - P)^{i-1} = \frac{1}{P}$$

$$\text{Si } N \rightarrow \infty: P_{\text{máx}} = 1/e \text{ y por tanto } n = e = 2,72$$

- d) $t_{\text{paq}} = t_{\text{tx}} + t_{\text{p}} + t_{\text{éxito}}$

$$t_{\text{p}} = \tau = 510^{-6}$$

$$t_{\text{tx}} = \frac{P}{V} = \frac{125 \cdot 8}{10 \cdot 10^6} = 100 \cdot 10^{-6}$$

El tiempo que se tarda de media en conseguir una transmisión fructífera es:

$t_{\text{éxito}} = n \cdot 2\tau$ (número medio de intervalos para transmitir con éxito por la duración de cada intervalo).

Los valores que optimizan la probabilidad de transmisión con éxito:

$$P_{\text{máx}} = \left(1 - \frac{1}{N}\right)^{N-1} = 371,60 \cdot 10^{-3}$$

$$n = \frac{1}{P_{\text{máx}}} = 2,69$$

Con lo que: $t_{\text{éxito}} = 2,69 \cdot 2 \cdot 5 \cdot 10^{-6} = 26,9 \cdot 10^{-6}$

$$t_{\text{paq}} = t_{\text{tx}} + t_{\text{p}} + t_{\text{éxito}} = 131,90 \cdot 10^{-6} \text{ s}$$

$$e) \quad E = \frac{t_{\text{tx}}}{t_{\text{paq}}} = \frac{100 \cdot 10^{-6}}{131,90 \cdot 10^{-6}} = 0,7581 \rightarrow 75,81\%$$

$$f) \quad M = \frac{R \cdot E}{p \cdot N} = \frac{10 \cdot 10^6 \cdot 0,7581}{125 \cdot 83 \cdot 50} = 151,62 \text{ paquetes/s}$$

12.10.

Sea una red en anillo de 400 m de longitud a 14 Mbps. Si posee 30 estaciones conectadas y cada una introduce un retardo en buffer de 1 bit:

- ¿Cuánto tiempo transcurrirá desde que una estación manda un paquete hasta que lo recibe de vuelta?
- Determinar el número de bits que habrá podido mandar el origen antes de que la cabecera del paquete llegue de vuelta al origen.

SOLUCIÓN

a) El tiempo transcurrido será:

$$t = t_{\text{p}} + 30 t_{\text{buffer}}$$

El tiempo de propagación:

$$t_{\text{p}} = \frac{L}{c} = \frac{400}{2 \cdot 10^8} = 2 \cdot 10^{-6}$$

El tiempo consumido en cada buffer:

$$t_{\text{buffer}} = \frac{\text{bits_buffer}}{V} = \frac{1}{14 \cdot 10^6} = 71,42 \cdot 10^{-9}$$

Con lo que queda:

$$t = t_{\text{p}} + 30 t_{\text{buffer}} = 4,14 \mu\text{s}$$

b) Los bits que habrá podido mandar serán:

$$b = t \cdot V = 4,14 \cdot 10^{-6} \cdot 14 \cdot 10^6 = 57,96 \approx 58 \text{ bits}$$

12.11.

Suponga el siguiente diagrama de transmisión de tramas con 24 bytes de cabecera Ethernet+IP y 1.490 bytes de datos cada una de ellas.

- Calcule la utilización de la red si se trata de una red 10BaseT.
- Si se tratase de paquetes con nivel de transporte UDP (8 bytes), ¿cuál será el throughput medido a nivel UDP?

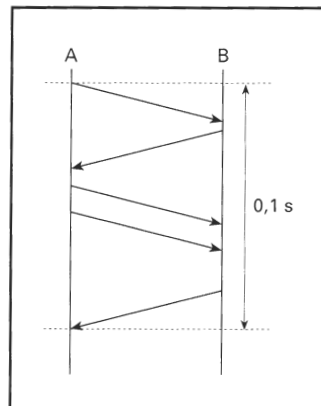


Figura 12.3. Diagrama de transmisión.

SOLUCIÓN

a) A manda 3 tramas y B 2 tramas haciendo un total de 5

$$\text{bytes} = 5 \cdot (24 + 1490) = 7.570 \text{ bytes}$$

$$\text{Utilización} = \frac{\text{bits transmitidos}}{\text{bits a la velocidad de transmisión en el mismo tiempo}} = \frac{7.570 \cdot 8}{10 \cdot 10^6 \cdot 0,1} \rightarrow 6,05\%$$

b) Throughput a nivel UDP:

$$\text{bytes} = 5 (1.490 - 8) = 7.410 \text{ bytes}$$

$$\text{Throughput} = \frac{\text{bits transmitidos}}{\text{tiempo}} = \frac{7.410 \cdot 8}{0,1} = 592,80 \text{ Kbps}$$

12.12.

Suponer una red ALOHA en la que dos máquinas A y B tienen paquetes pendientes de transmisión a otra máquina C. Suponer que A transmite con una probabilidad P_A y que B lo hace con una probabilidad P_B .

- Determinar la probabilidad de que se mande un paquete con éxito (que llamaremos eficiencia del sistema).
- Calcule la máxima eficiencia obtenible en el sistema. ¿Es la red justa con todos los nodos?
- Si $P_A = 0,5$, ¿depende la eficiencia de P_B ?

SOLUCIÓN

a) Eficiencia del sistema:

$$\begin{aligned} E = P(\text{éxito}) &= P((A_{\text{tx}} \text{ y } B_{\text{notx}}) \text{ o } (A_{\text{notx}} \text{ y } B_{\text{tx}})) = P_A(1 - P_B) + (1 - P_A)P_B = \\ &= P_A + P_B - 2P_AP_B \end{aligned}$$

b) Máxima eficiencia:

El máximo se obtiene para $P_A = 1, P_B = 0$ o para $P_A = 0, P_B = 1$, es decir que sólo uno transmite a la vez, lo cual no es una situación deseable (el reparto de ancho de banda no es equitativo).

c) $E(P_A = 0,5) = 0,5$

Es decir, no depende de P_B . Conforme P_B crezca algunas transmisiones de A resultarán en colisiones, pero por contra algunas de B serán exitosas, compensando la eficiencia. En los casos extremos:

$P_B = 0 \rightarrow A$ transmite en exclusiva $\rightarrow E = 50\%$.

$P_B = 1 \rightarrow 50\%$ son colisiones, 100% paquetes de A colisionan y 50% de paquetes de B se transmiten con éxito $\rightarrow E = 50\%$.

12.13.

Considerando que Ethernet se basa en CSMA / CD y que se quiere una red que soporte 4 Mbps efectivos, paquetes de 1.500 bytes y 1.000 m de distancia máxima, ¿será suficiente usar una red Ethernet para un alto número de estaciones?

SOLUCIÓN

Si N es el número de estaciones y $N \rightarrow \infty$, sabemos que: $U = \frac{1}{1 + 3,44a}$

Donde:

$$t_{\text{tx}} = \frac{P}{R} = \frac{1.500 \cdot 8}{10^6} = 1,2 \cdot 10^{-3}$$

$$t_p = \frac{L}{c} = \frac{1.000}{2 \cdot 10^8} = 5 \cdot 10^{-6}$$

$$a = \frac{t_p}{t_{\text{tx}}} = 4,16 \cdot 10^{-3}$$

Por lo que: $U = 98,58\%$, que es mayor del 40% (4 Mbps), por lo que la solución propuesta es válida.

12.14.

¿Qué mejora en el throughput puedo obtener si unimos dos máquinas mediante 2 redes Ethernet 10 Mbps en paralelo, sobre coaxial 2.500 m y 100 bytes de tamaño de paquete, comparado con la conexión mediante una única red?

SOLUCIÓN

Una red Ethernet:

$$t_p = \frac{L}{c} = \frac{2.500}{2 \cdot 10^8} = 12,5 \cdot 10^{-6}$$

$$t_{tx} = \frac{P}{R} = \frac{100 \cdot 8}{10 \cdot 10^6} = 80 \cdot 10^{-6}$$

$$a = \frac{t_p}{t_{tx}} = 156,25 \cdot 10^{-3}$$

$$U_1 = \frac{1}{1 + 3,44a} = 0,6504 \rightarrow (V = 10 \text{ Mbps}) U = 6,50 \text{ Mbps}$$

Dos redes Ethernet en paralelo:

$$t_p = \frac{L}{c} = \frac{2.500}{2 \cdot 10^8} = 12,5 \cdot 10^{-6}$$

$$t_{tx} = \frac{P}{R} = \frac{100 \cdot 8}{20 \cdot 10^6} = 40 \cdot 10^{-6}$$

$$a = \frac{t_p}{t_{tx}} = 312,5 \cdot 10^{-3}$$

$$U_2 = \frac{1}{1 + 3,44a} = 0,4819 \rightarrow (V = 20 \text{ Mbps}) U = 9,63 \text{ Mbps}$$

Es decir, no se consigue el doble de velocidad como cabría suponer a priori.

$$\frac{U_1}{U_2} = 1,48 \rightarrow 48\% \text{ de mejora}$$

12.15.

Compare las velocidades que se pueden obtener para el menor tamaño de paquete en una red Fast Ethernet y tamaño mínimo de paquete de 64 bytes con una Gigabit Ethernet y tamaño mínimo de paquete 512 bytes, considerando en ambos casos 8 bytes más de preámbulo y 12 bytes más de espaciado entre paquetes. ¿En qué medida afectan el preámbulo y espaciado en ambos casos?

SOLUCIÓN

$$\frac{100 \cdot 10^6}{8(64 + 12 + 8)} = 148.809 \text{ paquetes/s} \rightarrow 76 \text{ Mbps} \rightarrow \frac{100 - 76}{100} \cdot 100 = 34\%$$

$$\frac{110^9}{8(512 + 12 + 8)} = 234.962 \text{ paquetes/s} \rightarrow 962,4 \text{ Mbps} \rightarrow \frac{1.000 - 962,4}{1.000} \cdot 100 = 3,8\%$$

En Gigabit, el número de paquetes de tamaño mínimo por segundo apenas es el doble que en Fast Ethernet, pero la velocidad real es mucho mayor. El usar paquetes de tamaño mínimo más grandes en Gigabit hace que el efecto de preámbulos y espaciados sea menor.

12.16.

Considere una red token ring 16 Mbps con un tiempo de propagación de $600 \mu\text{s}$ y 5 estaciones separadas de forma equidistante. Si todas las estaciones tienen tráfico para transmitir consistente en paquetes de 1.000 bytes, ¿cuál es la utilización del anillo con y sin liberación temprana de testigo?

SOLUCIÓN

Mejor caso: Que se envíe entre estaciones más cercanas.

$$t_p = 600 \cdot 10^{-6}$$

$$t_{tx} = \frac{P}{R} = \frac{1.000 \cdot 8}{16 \cdot 10^6} = 500 \cdot 10^{-6}$$

$t_p > t_{tx}$ por lo que el paquete sale del origen antes de que se reciba de vuelta por completo.

Los tiempos de transmisión del testigo se consideran despreciables.

Sin liberación temprana del testigo, el paquete tiene que volver a la estación origen para poder relegar el testigo a la siguiente estación: t_p es el tiempo de vuelta del paquete y t_p/N el tiempo que consume ceder el testigo a la siguiente estación.

$$U_{\sin} = \frac{t_{tx}}{t_p + \frac{t_p}{N}} = 0,6944 \rightarrow 69,44\%$$

Con liberación temprana del testigo, el testigo se manda a la siguiente estación tan pronto como se termina de transmitir el paquete en la estación origen.

$$U_{\sin} = \frac{t_{tx}}{t_p + \frac{t_p}{N}} = 0,8064 \rightarrow 80,64\%$$

12.17.

Una red Ethernet 10 Mbps tiene una distancia máxima alcanzable de 2,5 Km y tamaño mínimo de paquete 64 bytes. En cambio, una Fast Ethernet tiene 100 Mbps, mantiene el tamaño de paquete mínimo pero reduce la máxima distancia alcanzable. ¿Por qué y a qué valor?

SOLUCIÓN

La distancia máxima, y con ello el tamaño de paquete mínimo, vienen determinados por el mecanismo de detección de colisiones utilizado por ambos estándares: CSMA / CD. Si se quiere detectar una colisión, una trama ha de durar al menos dos veces el tiempo de propagación:

$$t_{tx} \geq 2 t_p = 2 \frac{2,5 \cdot 10^3}{2 \cdot 10^8} = 25 \mu\text{s}$$

Este valor, por razones de intervalo de seguridad, se convierte en el doble, 50 ms, a lo que corresponde para una Ethernet 10 Mbps aproximadamente un tamaño de paquete de:

$$P = V t_{tx} = 10 \cdot 10^6 \cdot 50 \cdot 10^{-6} = 500 \text{ bits} = 62,5 \text{ bytes} \approx 64 \text{ bytes}$$

Por tanto, si mantenemos un tamaño de paquete $P = 64$ bytes, pero subimos la velocidad de transmisión a 100 Mbps como en Fast Ethernet, forzosamente el tiempo de propagación tiene que ser menor.

$$\frac{64 \cdot 8}{100 \cdot 10^6} \geq 2t_p$$

$t_p = 2,56 \cdot 10^{-6}$ y por tanto una distancia en el rango de 512 m, que tomando de nuevo el margen de seguridad (del doble) se queda en 200 m, que es la distancia nodo a nodo de una Fast Ethernet (100 m del concentrador/conmutador central a cada nodo).

12.18.

Proponer un modelo de red ALOHA puro y ALOHA ranurado suponiendo que se tienen N fuentes de tráfico conectadas, que generan λ mensajes/s por fuente, la duración de cada mensaje es de τ y la capacidad del canal es C . Suponer que la tasa λ sigue una Distribución de Poisson.

SOLUCIÓN

ALOHA puro:

La utilización del canal por parte de estas fuentes será:

$$U = \frac{N\lambda}{C}$$

Si los mensajes se generan con una función de Distribución de Poisson, la probabilidad de tener n mensajes a transmitir en el mismo intervalo t por una única fuente será:

$$P_n(t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

Como nos interesa la probabilidad de tener n mensajes a transmitir en el mismo intervalo t por todas las fuentes, consideraremos T la tasa total de mensajes/s de todas las fuentes:

$$P_n(t) = \frac{(Tt)^n e^{-Tt}}{n!}$$

Esta probabilidad se corresponde también con la probabilidad de que n fuentes transmitan a la vez, que será el caso en el que habrá colisión, y por tanto la transmisión no será exitosa.

Si los mensajes son de duración τ deberá transmitirse únicamente un paquete en cada período 2τ , tal y como se representa en la figura, para garantizar la no colisión entre mensajes.

$$t_{\text{vulnerable}} = 2\tau = \frac{2}{C}$$

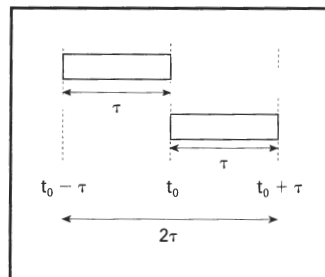


Figura 12.4. Intervalos.

De la expresión de probabilidad, la probabilidad de no colisión será aquella en la que $n = 0$ en un período 2τ , o lo que es lo mismo $t = 2$ si medimos el tiempo en múltiplos de la duración de los mensajes, es decir, una vez que uno transmite nadie transmite nada en ese período, con lo que:

$$P_{n=0}(t=2) = e^{-2T}$$

La probabilidad de que al menos una colisión tenga lugar será:

$$P_{\text{col}} = 1 - P_{n=0}(t=2) = 1 - e^{-2T}$$

La tasa media de retransmisión será:

$$R = TP_{\text{col}} = T(1 - e^{-2T})$$

Y por tanto, la tasa total de mensajes por unidad de tiempo será:

$$T = E + R = E + T(1 - e^{-2T})$$

De los que E son los mensajes transmitidos con éxito por unidad de tiempo:

$$E = Te^{-2T}$$

El máximo se obtendrá de derivar e igualar a 0:

$$\frac{dE}{dT} = 0 = e^{-2T}(1 - 2T)$$

Despejando:

$$T = 0,5 \text{ y por tanto } E = \frac{1}{2e} = 0,1839$$

ALOHA ranurado:

Se procede como con el ALOHA puro pero ahora $t_{\text{vulnerable}} = \tau$ porque cada fuente sólo puede transmitir dentro de una ranura o no transmitir, no puede empezar a transmitir en medio de la ranura, por ejemplo.

Repitiendo los cálculos:

$$E = Te^{-T}$$

Máximo para $T = 1$ y por tanto $E = \frac{1}{e} = 0,3678$.

12.19.

Suponga dos estaciones usando CSMA / CD con espera exponencial (exponential backoff) que acaba justo de transmitir y han colisionado.

a) ¿Cuál es la probabilidad de que colisionen en el siguiente intento?

- b) Calcule la probabilidad de que ambas estaciones transmitan satisfactoriamente en los dos próximos intervalos.
- c) Probabilidad de que no transmitan con éxito dos y tres veces más.

SOLUCIÓN

- a) De acuerdo con el algoritmo de espera exponencial, ambas estaciones reservarán de manera independiente los siguientes slots para transmisión de nuevo con igual probabilidad de elegir uno de los dos, porque acaba de ocurrir una colisión entre ambos.

Hay cuatro posibilidades en los siguientes dos slots: (transmite, vacío), (transmite, transmite), (vacío, transmite) y (vacío, vacío). Una estación entonces elige un slot para transmitir con igual probabilidad, luego cada posibilidad tiene una probabilidad de $1/4$. Por tanto, $1/4$ es la probabilidad de que se colisione (transmite, transmite).

- b) Nos interesan ahora las posibilidades (transmite, vacío) y (vacío, transmite) para ambas estaciones y cada intervalo. Como cada posibilidad tiene una probabilidad de $1/4$: $1/4 + 1/4 = 1/2$
- c) Probabilidad de que colisionen una segunda vez seguida, que ambas estaciones transmitan a la vez en ambos intervalos, el primero de 2 slots y el segundo de 4:

$$P(2 \text{ colisiones}) = P(\text{colisión en intervalo de 2 slots}) \cdot P(\text{colisión en intervalo de 4 slots})$$

$$P(\text{colisión en intervalo de 2 slots}) = 2 \cdot (1/2 \cdot 1/2) = 1/2$$

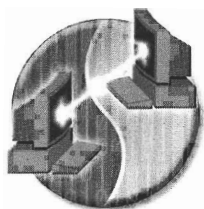
$$P(\text{colisión en intervalo de 4 slots}) = 4 \cdot (1/4 \cdot 1/4) = 1/4$$

$$P(2 \text{ colisiones}) = 1/2 \cdot 1/4 = 1/8$$

$$P(3 \text{ colisiones}) = P(\text{colisión en intervalo de 2 slots}) \cdot P(\text{colisión en intervalo de 4 slots}) \cdot P(\text{colisión en intervalo de 8 slots})$$

$$P(\text{colisión en intervalo de 8 slots}) = 8 \cdot (1/8 \cdot 1/8) = 1/8$$

$$P(3 \text{ colisiones}) = 1/2 \cdot 1/4 \cdot 1/8 = 1/64$$



PROTOCOLOS DE INTERCONEXIÓN DE REDES

13.1.

Suponga que se transmite un fichero de 100 Kbytes directamente sobre un protocolo de transporte UDP. El tamaño de la cabecera IP es constante de 20 bytes y el tamaño de la cabecera UDP es de 8 bytes. Suponga que no hay errores en la transmisión y que se atraviesa una red origen con una MTU de 1.500 bytes, otra red con MTU de 500 y finalmente otra red destino con MTU de 1.500 bytes, todas ellas unidas por encaminadores IP. Determinar el número de bytes que estarán llegando a la red destino. Suponer despreciable el tamaño de las cabeceras de enlace.

SOLUCIÓN

1.ª red: $MTU - IP - UDP = 1.500 - 20 - 8 = 1.472$ bytes de datos en cada paquete (28 bytes de cabeceras)

Paquetes necesarios para transmitir el fichero de 100 Kbytes:

$$\frac{100 \cdot 1.024}{1.472} = 69,56 \rightarrow \begin{cases} 69 \text{ paquetes de } 1.472 \text{ bytes datos} + \text{cabeceras} = 1.500 \text{ bytes} \\ 1 \text{ paquete de } 832 \text{ bytes datos} + \text{cabeceras} = 860 \text{ bytes} \end{cases}$$

Suponemos que la aplicación origen manda los paquetes con estos tamaños sin ser necesario ningún tipo de fragmentación extra. Es decir, todos los paquetes tendrán cabeceras IP y UDP.

2.ª red: $MTU - IP - \{UDP\} = 500 - 20 - \{8\} = 480/\{472\}$ bytes de datos en cada paquete (según incluya 20 de cabecera IP o 28 de cabecera IP + UDP). Sólo el primer paquete de cada segmento llevará cabecera UDP. Por tanto, considero la cabecera UDP dentro de la parte de datos.

Los 69 paquetes de 1.500 bytes de la red anterior se segmentan en:

$$\frac{1.500 - 20}{480} = 3,08 \rightarrow 69 \begin{cases} 3 \text{ paquetes de } 480 \text{ bytes datos} + \text{cabeceras} = 500 \text{ bytes} \\ 1 \text{ paquete de } 400 \text{ bytes datos} + \text{cabeceras} = 60 \text{ bytes} \end{cases}$$

El paquete de 860 bytes de la red anterior se segmenta en:

$$\frac{860 - 20}{480} = 1,75 \rightarrow 1 \begin{cases} 1 \text{ paquetes de } 480 \text{ bytes datos} + \text{cabeceras} = 500 \text{ bytes} \\ 1 \text{ paquete de } 360 \text{ bytes datos} + \text{cabeceras} = 380 \text{ bytes} \end{cases}$$

480 bytes es múltiplo de 64 bits por lo que es válido ese tamaño de datos para definir el offset en los siguientes fragmentos.

3.ª red: No segmenta porque su MTU es mayor que la de la segunda red. Tampoco reensambla como es lo normal en encaminadores IP. Por tanto, se mantiene el número de paquetes y sus tamaños de la red 2 con lo que queda la siguiente suma final de bytes que llegan a esta tercera red:

$$\text{BytesTotales} = 69 \cdot 3 \cdot 500 + 69 \cdot 1 \cdot 60 + 1 \cdot 1 \cdot 500 + 1 \cdot 1 \cdot 380 = 108.520 \text{ bytes}$$

Que está por encima de los 100 Kbytes = 102.400 bytes del fichero, resultando una sobrecarga del 7,6%.

13.2.

Sea una red como la de la figura con un encaminador (router) R1. Determine:

- La red que forman los ordenadores A y B.
- La red del ordenador C.
- Si se colocase un encaminador R2 entre R1 y C, ¿qué nueva subred crearía y qué direcciones IP asignaría a R2 y C?

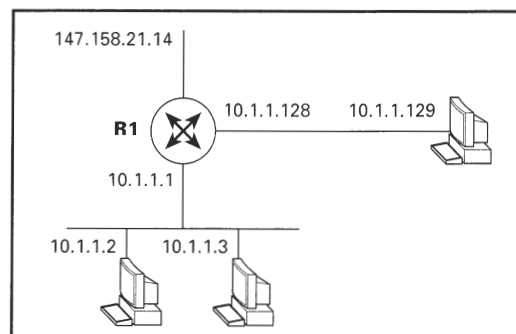


Figura 13.1. Red con encaminador.

SOLUCIÓN

- a) Subred: 10.1.1.0 / 255.255.255.128

Así dispondré de los últimos 7 bits para asignar direcciones dentro de esa subred.

- b) Subred: 10.1.1.128 / 255.255.255.128

De nuevo se dispondrán de 7 bits para asignar direcciones dentro de esa subred.

- c) R1-R2 subred 10.1.1.128 / 255.255.255.192

R1: 10.1.1.128

R2: 10.1.1.129

R2-C subred 10.1.1.192 / 255.255.255.192

R2: 10.1.1.192

C: 10.1.1.193

13.3. ¿Se puede construir un protocolo orientado a conexión sobre una red de datagramas de entrega no fiable?

SOLUCIÓN

Sí, TCP es un ejemplo de ese protocolo.

13.4. Explicar las ventajas e inconvenientes de realizar la fragmentación en el nodo origen como en IPv6 en lugar de en cada encaminador como en IPv4.

SOLUCIÓN

Ventajas: Rapidez en el procesamiento de los paquetes en cada salto intermedio (encaminador) que se despreocupan ahora de las labores de fragmentación.

Inconvenientes: El nodo origen tiene que descubrir *a priori* cuál es la mínima MTU en el camino mediante el correspondiente algoritmo (retraso en el comienzo de la comunicación) o puede usar una MTU de 1.280 bytes que es la mínima MTU exigida en redes IPv6 (peor eficiencia en la utilización de la red si usamos una MTU menor que la que realmente posee la red).

13.5. ¿Sería posible en una red IPv4 realizar el reensamblado en los encaminadores intermedios? ¿Y la fragmentación? ¿Tendría alguna ventaja?

SOLUCIÓN

En principio, no es posible el reensamblado en los encaminadores intermedios porque los paquetes de un flujo pueden seguir rutas diferentes en la red y por tanto no pasar por los mismos encaminadores.

La fragmentación si es posible, de hecho es básica en encaminadores IPv4, porque se realiza por cada paquete, sin tener en cuenta para nada información del resto de paquetes que componen el flujo.

La ventaja del reensamblado intermedio es que minimiza el número de paquetes en la red y por tanto, reduce la sobrecarga del protocolo al haber menos cabeceras y menos tiempo de guarda entre paquetes.

13.6. Un paquete de 5.000 bytes de los que 14 bytes de cabecera de nivel Ethernet, 20 de nivel IP y 8 de nivel UDP es fragmentado para pasar a una red Ethernet con una MTU de 1.514 bytes. Mostrar los campos de longitud total, indicador de más paquetes y desplazamiento de fragmento en cada uno de los fragmentos resultantes.

SOLUCIÓN

La cabecera UDP se encontrará sólo en el primer fragmento por lo que la considero dentro de la parte de datos del paquete IP.

$$\frac{5.000 - 14 - 20}{1.514 - 14 - 20} = 3,35 \rightarrow \begin{cases} 3 \text{ paquetes de } 1.480 \text{ bytes datos} + \text{cabeceras} = 1.514 \text{ bytes} \\ 1 \text{ paquete de } 526 \text{ bytes datos} + \text{cabeceras} = 560 \text{ bytes} \end{cases}$$

1.480 bytes es múltiplo de 64 bits por lo que es válido ese tamaño de datos para definir el offset en los siguientes fragmentos.

Fragmento	1.º	2.º	3.º	4.º
Longitud total	1.500	1.500	1.500	546
Flag más paquetes	1	1	1	1
Desplazamiento fragmento (offset)	0	1.480	2.960	4.440

Tabla 13.1. Valores de campos seleccionados de cada fragmento.

13.7.

Un mensaje de la capa de transporte que contiene 1.500 bits de datos y 160 bits de cabecera se envía a la capa IP, la cual incorpora otros 160 bits de cabecera. Esto se transmite a través de dos redes que utilizan 24 bits de cabecera a nivel de enlace. La red origen tiene un tamaño de paquete máximo de 8.000 bits y la red destino tiene un tamaño de paquete máximo de 800 bits. ¿Cuántos bits incluyendo cabeceras se entregarán al protocolo de la capa de red en el destino?

SOLUCIÓN

La fragmentación ocurrirá en el paso de la primera a la segunda red que tiene una MTU insuficiente para contener el paquete completo generado en la primera red.

$800 - 24 - 160 = 616$ bits no es múltiplo de 64 bits, por lo que no es válido ese tamaño de datos para definir el offset en los siguientes fragmentos. Por tanto, tomamos el múltiplo más cercano que no sobrepase la MTU = 576 de datos:

$$\frac{1.500 + 160}{576} = 2,88 \rightarrow \begin{cases} 2 \text{ paquetes de } 576 \text{ bytes datos} + \text{cabeceras (eth + IP)} = 760 \text{ bytes} \\ 1 \text{ paquete de } 508 \text{ bytes datos} + \text{cabeceras (eth + IP)} = 692 \text{ bytes} \end{cases}$$

A la capa de red del destino llegarán:

$$2 \cdot (576 + 160) + 1 \cdot (508 + 160) = 2.140 \text{ bits}$$

13.8.

Calcule el árbol de expansión para una transmisión multicast multidifusión del nodo A a los nodos C y D suponiendo el coste de los enlaces de la figura. ¿En qué encaminador se duplican paquetes?

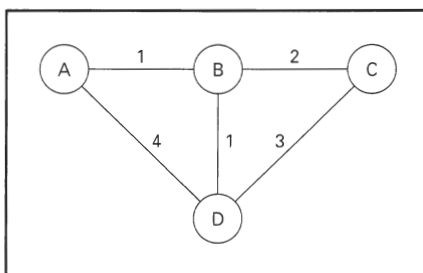


Figura 13.2. Transmisión multidifusión.

SOLUCIÓN

El árbol de expansión será:

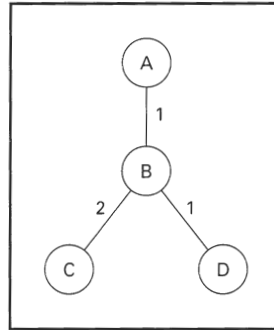


Figura 13.3. Árbol de expansión.

Los paquetes se duplican en el nodo B como queda claro con la figura del árbol de expansión.

13.9.

Suponga la siguiente secuencia de bytes en hexadecimal, suponga que se trate de una cabecera IP a falta del campo de checksum (suma de comprobación de la cabecera) y calcule su checksum. ¿Cómo chequearía la máquina receptora la corrección de esta cabecera? Si esta cabecera es modificada en un salto intermedio, cambiando el 1.421 por 0000, explique una forma de calcular el nuevo checksum sin falta de rehacerlo por completo desde el principio.

03 47 A1 F0 14 21 F5 C0

SOLUCIÓN

Formamos primero palabras de 16 bits: 0347 A1F0 1421 F5C0.

Suponiendo que se encuentran en formato complemento a 2 (habitual forma de representación en ordenadores actualmente), calculamos la suma complemento a 2:

$$0347 + A1F0 + 1421 + F5C0 = 0001 AF18$$

Sumamos el acarreo final (0001) para obtener la suma complemento a 1 de 16 bits:

$$0001 + AF18 = AF19$$

Finalmente calculamos el complemento a 1 de la suma complemento a 1:

$$AF19 = 1010\ 1111\ 0001\ 1001$$

$$C_1(AF19) = 0101\ 0000\ 1110\ 0110 = 50E6$$

El checksum será este valor de 16 bits: 50E6.

El destino se chequearía calculando de nuevo la suma complemento a 1 de todos los bytes (datos + checksum) y será correcto si resulta FFFF:

$$0347 + A1F0 + 1421 + F5C0 + 50E6 = 0001\ FFFE$$

$$0001 + FFFE = FFFF \rightarrow \text{checksum correcto}$$

Si en la cabecera se modifica el 1.421 por 0000, no es necesario recalcular el checksum desde el principio, se puede aplicar un método incremental.

Primero lo vamos a calcular desde cero aplicando el método anterior:

$$0347 + A1F0 + 0000 + F5C0 = 0001\ 9AF7$$

$$\text{SumaC}_1(0001\ 9AF7) = 9AF8$$

$$\text{nuevo_checksum} = C_1(9AF8) = 6507$$

El método incremental consiste en restar la palabra nueva de 16 bits de la original y sumar este valor al checksum anterior resultando el nuevo checksum:

$$0000 - 1.421 = 0000 + C_2(1.421) = 0000 + EBDE = EBDE$$

El complemento a 2:

$$C_2(1.421) = 2^{16} - 1.421 = EBDE$$

$$\text{nuevo_checksum} = C_1(\text{anterior_checksum} + EBDE) =$$

$$= C_1(AF19 + EBDE) = C_1(0001\ 9AF7) = C_1(9AF8) = 6507$$

que coincide con el valor del checksum calculado desde 0.

13.10.

¿Al menos qué campo se modifica siempre en los paquetes al atravesar un encaminador? ¿Es necesario entonces recalcular el checksum en todos los saltos?

SOLUCIÓN

El campo de tiempo de vida (TTL) siempre se modifica en cada encaminador que atraviesa porque debe ser decrementado por cada encaminador en al menos una unidad y evitar así que ciclos en la red hagan que el paquete permanezca en ella indefinidamente (el paquete se descarta cuando este campo llega a 0).

Sí es necesario calcular el checksum en todos los saltos porque como hemos dicho, el campo TTL siempre se modifica en un encaminador. Sin embargo, existe una forma en la que no se recalcula el checksum de la cabecera IP entera sino sólo la contribución del campo modificado en ese encaminador.

13.11.

¿Cuál es la sobrecarga que impone IPv6 con respecto a IPv4 en una red de MTU 1.514 bytes?

SOLUCIÓN

Si consideramos las cabeceras básicas sin opciones, la cabecera de IPv4 es de 20 bytes y la cabecera de IPv6 es de 40 bytes. Por tanto, si comparamos IPv6 con IPv4, la sobrecarga que introduce IPv6 en lo que a cabeceras respecta será:

$$\frac{\text{cabecera}_{\text{IPv6}} - \text{cabecera}_{\text{IPv4}}}{\text{MTU}} = \frac{40 - 20}{1.514} = 0,01319 \rightarrow 1,31\%$$

13.12.

Suponga que la siguiente secuencia de bytes en hexadecimal es una cabecera IP y decodifique cada uno de sus campos:

45 00 00 4E C3 2A 00 00 80 11 17 44 82 CE AA 94 82 CE AF FF

SOLUCIÓN

Bits	Campo	Valor
4	Versión	4
4	Longitud cabecera en palabras de 32 bits	5
8	TOS	0×100
16	Longitud total	$0 \times 004E = 78$
16	Identificación	$0 \times C32A$
3	Flags	0×0
13	Desplazamiento de fragmento	$0 \times 00 = 0$
8	Tiempo de vida	$0 \times 80 = 128$
8	Protocolo	$0 \times 11 = 17 = \text{UDP}$
16	Cheksun	0×1.744 (correcto)
32	Dirección IP origen	$0 \times 82CEAA94 = 130.206.170.148$
32	Dirección IP destino	$0 \times 82CEAFFF = 130.206.175.255$

Tabla 13.2. Campos de la cabecera IP.

13.13.

Suponga dos redes unidas por un encaminador R. Una estación A en la primera red transmite un paquete a una estación B en la segunda red. Suponiendo que se conocen todas las direcciones MAC e IP de todos los interfaces, determine las direcciones que tendrán los paquetes generados en cada red.

SOLUCIÓN

1.ª red: Paquete de A a RA (interfaz del encaminador en la red A).

$$\text{MAC}_{\text{origen}} = \text{MAC}_A$$

$$\text{MAC}_{\text{destino}} = \text{MAC}_{RA}$$

$$\text{IP}_{\text{origen}} = \text{IP}_A$$

$$\text{IP}_{\text{destino}} = \text{IP}_B$$

2.ª red: Paquete de RB (interfaz del encaminador en la red B) a B.

$$\text{MAC}_{\text{origen}} = \text{MAC}_{RB}$$

$$\text{MAC}_{\text{destino}} = \text{MAC}_B$$

$$\text{IP}_{\text{origen}} = \text{IP}_A$$

$$\text{IP}_{\text{destino}} = \text{IP}_B$$

13.14.**¿Qué representa el tiempo obtenido por paquetes ICMP ECHO?****SOLUCIÓN**

Representa el RTT (Round Trip Time), es decir, el tiempo de ida y vuelta de los paquetes entre la máquina origen y destino seleccionada.

13.15.**Sea un encaminador que conecta las siguientes redes:**

Puerto	Red
A	10.1.1.0 / 255.255.255.0
B	10.1.2.0 / 255.255.255.128
C	10.1.2.1 / 255.255.255.128
D	Defecto

Tabla 13.3. Tabla de rutas del encaminador.

¿Por qué puerto sacará los paquetes con destino a las máquinas: 10.1.1.128, 10.2.2.1, 10.1.2.200 y 10.1.2.50?

SOLUCIÓN

10.1.1.128 → A

10.2.2.1 → D

10.1.2.200 → C

10.1.2.50 → B

13.16.

Si se dispone de una red B completa 133.144.0.0 y se quiere repartir ese rango de direcciones en 10 subredes lo mayores posibles, ¿cómo lo haría? ¿Cuántas direcciones dejará libres?

SOLUCIÓN

Para llevar cuenta de 10 subredes necesitaremos al menos 4 bits ($2^4 = 16$) por lo que la máscara común para todos será: X.X.11110000.X = 255.255.240.0

Las subredes resultantes serán las siguientes:

Subred 1: 133.144.16.0

Subred 2: 133.144.32.0

Subred 3: 133.144.48.0

Subred 4: 133.144.64.0

Subred 5: 133.144.80.0

Subred 6: 133.144.96.0

Subred 7: 133.144.112.0

Subred 8: 133.144.128.0

Subred 9: 133.144.144.0

Subred 10: 133.144.160.0

Subredes libres: $2^4 - 10 = 6$ subredes sin asignar con 2^{8+4} direcciones en cada una, por lo que las direcciones no usadas serán: $6 \cdot 2^{16} = 24.576$.

13.17.

Sea una red como la de la figura:

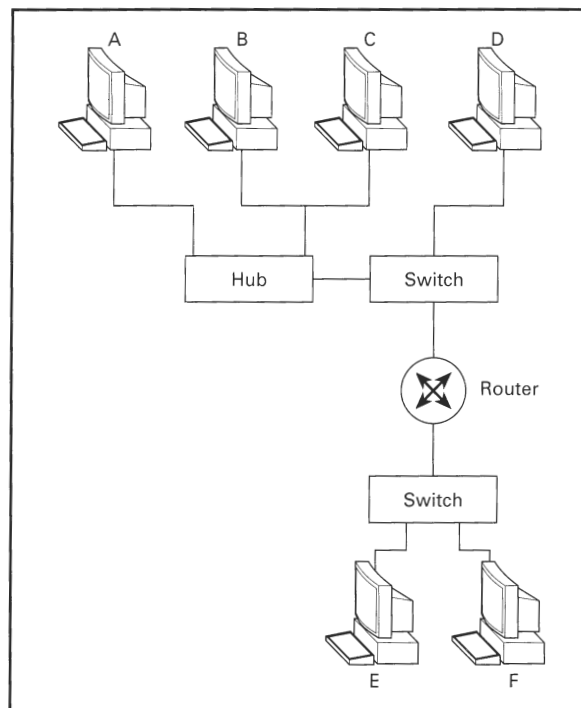


Figura 13.4. Esquema de red.

- Si A manda un paquete a F, ¿qué máquinas podrán ver el paquete?
- Determinar las direcciones MAC e IP que tendrá un paquete dirigido de A a E cuando sea recibido por E.
- ¿Qué máquinas recibirán los paquetes de difusión (broadcast) generados por A?

SOLUCIÓN

- B, C y F: Los que estén en el mismo dominio de colisión de la ruta.
- $$MAC_{\text{origen}} = MAC_{\text{router}}$$

$$MAC_{\text{destino}} = MAC_E$$

$$IP_{\text{origen}} = IP_A$$

$$IP_{\text{destino}} = IP_E$$
- B, C y D: Las que se encuentren dentro de la misma red de área local (subred).

13.18.

En IPv6 el estado asociado con cada flujo en la red se almacena en los dispositivos de encaminamiento y se referencia por medio de las etiquetas de los paquetes. Un requisito del protocolo es eliminar en los encaminadores las etiquetas de flujos que no se van a utilizar más.

- a) Si una fuente siempre envía mensajes de control a los encaminadores que atraviesa determinado flujo suprimiendo su etiqueta cuando el flujo termina, ¿cómo podría persistir una etiqueta de flujo antigua?
- b) Sugiera mecanismos de los encaminadores para resolver el problema de las etiquetas de flujo antiguas.

SOLUCIÓN

- a) Si la finalización del flujo no es correcta (por ejemplo, que se caiga el nodo origen), no se mandarían esos mensajes de control porque no se habría detectado la finalización del flujo y por tanto no se eliminarían nunca las etiquetas correspondientes a ese flujo.
- b) Un mecanismo habitual es el de poner tiempos de vencimiento (timeouts) que se empiezan a contar a partir del último paquete visto correspondiente a ese flujo. Una vez ese contador temporal supera determinado umbral, se puede dar el flujo por finalizado y entonces se pueden eliminar las etiquetas de los encaminadores.

13.19.

Suponga un enlace de 256 Kbps entre dos ciudades separadas 3.000 Km. El tiempo medio obtenido al hacer un ping con paquetes de 64 bytes en ese enlace es de 36 ms. Estime si ese enlace se encuentra sobrecargado.

SOLUCIÓN

Calculamos el tiempo de RTT (Round Trip Time) que tendríamos en un enlace descargado y comparamos con el valor medido con el ping.

$$t_{\text{RTT}} = 2 (t_p + t_{\text{tx}}) = 34 \text{ ms}$$

$$t_p = \frac{3.000 \cdot 10^3}{2 \cdot 10^8} = 15 \cdot 10^{-3}$$

$$t_{\text{tx}} = \frac{64 \cdot 8}{256 \cdot 10^3} = 2 \cdot 10^{-3}$$

El RTT medido de 36 ms es poco mayor que el que tendría un enlace de ese tipo totalmente descargado, seguramente debido a pequeños retardos introducidos por los equipos de red. Por tanto, se puede decir que la red no está sobrecargada porque en caso de estarla este tiempo de RTT sería mucho más elevado.

13.20.

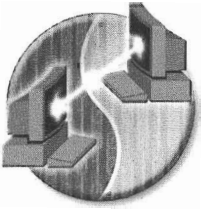
Calcule la máxima tasa con la que un nodo puede generar paquetes IP de 100 bytes para evitar que fragmentos correspondientes a distintos paquetes se puedan reensamblar juntos suponiendo que un paquete puede permanecer en la red hasta 60 segundos antes de ser descartado.

SOLUCIÓN

Los identificadores de fragmento de la cabecera IP son un número finito que eventualmente tiene que ser reutilizado. Para estar seguros de que fragmentos correspondientes a distintos datagramas IP pero con igual identificador no sean generados, debemos tener en cuenta el tamaño de 16 bits de este identificador y evitar que en el mismo intervalo de 60 segundos se puedan llegar a generar dos identificadores iguales.

$$\frac{2^{16} \text{ datagrama}}{60 \text{ s}} \frac{100 \text{ bytes}}{\text{datagrama}} = 3,33 \cdot 10^{16} \text{ bytes/s}$$

Es decir, una velocidad extremadamente elevada no alcanzable por la tecnología actual.



FUNCIONAMIENTO DE LA INTERCONEXIÓN DE REDES

14.1. ¿Qué relación existe entre la transmisión multidifusión y el protocolo RSVP?

SOLUCIÓN

Los computadores transmisores que hacen transmisión multidifusión tienen que llegar a diferentes destinos, y algunos de ellos comparten una parte del camino. Los dispositivos de encaminamiento necesitan estar seguros de las reservas de recursos de cada destino multidifusión.

La transmisión multidifusión es la responsable de gran parte de la complejidad del protocolo RSVP.

Las solicitudes de reserva desde los destinos deben propagarse hacia atrás a través de árboles de encaminamiento hacia cada ordenador origen.

14.2. ¿Cuáles son los tres conceptos que forman la base del funcionamiento de RSVP?

SOLUCIÓN

Los tres conceptos relacionados con los flujos de datos son los siguientes:

- **Sesión:** Flujo identificado por su destino.
- **Especificación de flujo:** Descriptor de flujo que genera una solicitud de reserva del destino (flowspec y filterspec).
- **Especificación de filtro:** Define un conjunto de paquetes para esa reserva.

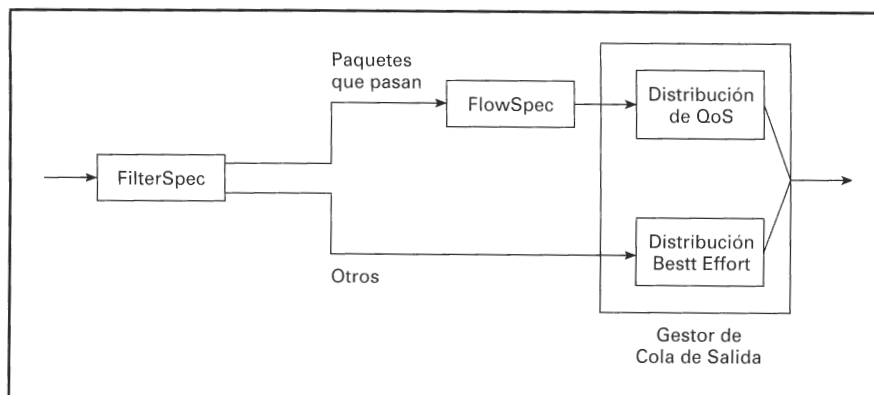


Figura 14.1. Funcionamiento RSVP.

14.3.

Diferencias entre “Información de Encaminamiento” y “Algoritmo de Encaminamiento” al considerar las funciones de encaminamiento de los dispositivos de encaminamiento.

SOLUCIÓN

- **Información de Encaminamiento:** Información sobre la topología y el retardo del conjunto de las redes.
- **Algoritmo de encaminamiento:** Algoritmo utilizado para las decisiones de encaminamiento para un datagrama particular, basado en información actual.

14.4.

Respecto los protocolos de encaminamiento, existen los Sistemas Autónomos, llamados AS. Para estos ASs se definen los protocolos IRP y ERP. ¿Qué diferencia existe entre ambos protocolos?

SOLUCIÓN

IRP por definición es Interior Router Protocol, lo que es lo mismo, Protocolo Interior de Dispositivos de Encaminamiento, que pasa información de encaminamiento en un sistema autónomo. Necesitan construir un modelo de interconexión detallado para el cálculo de camino de menor coste dentro del sistema autónomo.

ERP es por definición Exterior Router Protocol, y lo que significa Protocolo de dispositivo de encaminamiento Exterior. En la unión de diferentes sistemas autónomos que en su interior utilizan IRP, necesitan un ERP para pasar información de encaminamiento entre ambos sistemas. Es más simple que IRP por utilizar información de resumen, menor cantidad de información de alcanzabilidad entre sistemas autónomos.

14.5.

Dibujar el grafo del siguiente sistema autónomo, dónde H_x son los Host, R_x son los dispositivos de encaminamiento y los N_x son las subredes dentro del sistema autónomo.

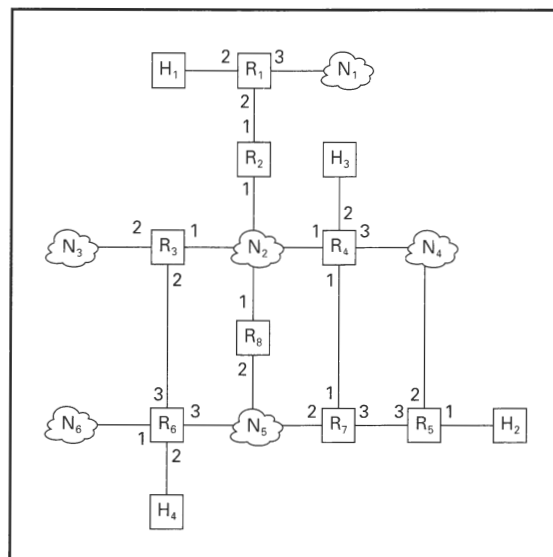


Figura 14.2. Sistema autónomo.

SOLUCIÓN

El grafo resultante será el siguiente:

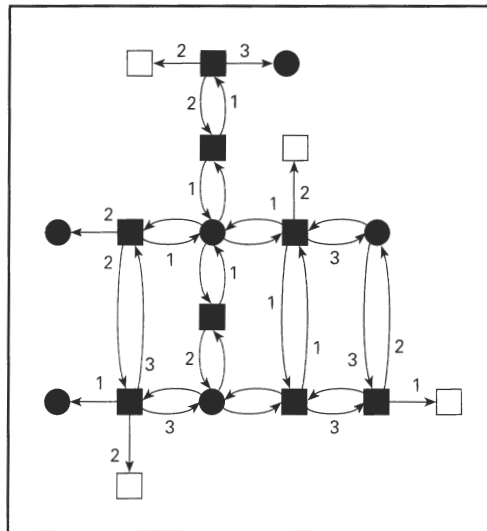


Figura 14.3. Grafo resultante.

14.6. Dibujar el árbol de encaminamiento del dispositivo R_1 .

SOLUCIÓN

El árbol de encaminamiento del dispositivo R_1 en este sistema autónomo será el siguiente:

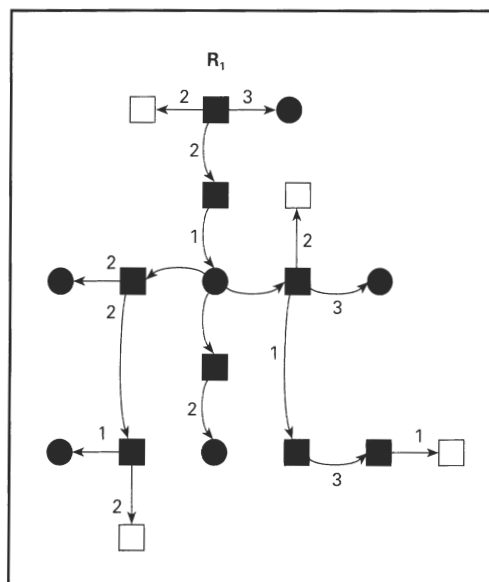


Figura 14.4. Árbol de encaminamiento.

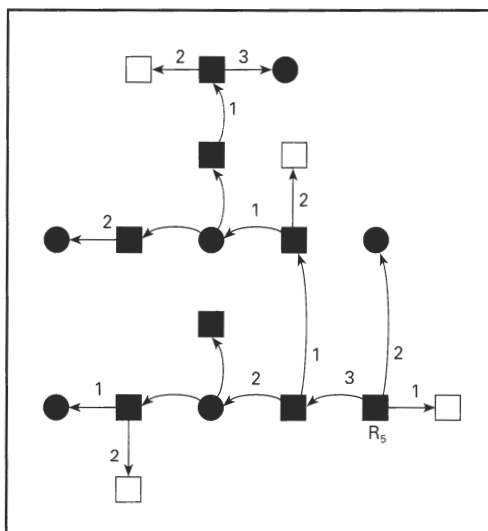
14.7.

Escribir la tabla de encaminamiento resultante del Problema 14.6.

SOLUCIÓNLa tabla de encaminamiento resultante para el dispositivo R_1 será:

Destino	Siguiente salto	Distancia
N_1	—	3
N_2	R_2	3
N_3	R_2	5
N_4	R_2	6
N_5	R_2	5
N_6	R_2	6
H	—	2
H	R_2	8
H	R_2	5
H_4	R_2	7

Tabla 14.1. Tabla resultante.

14.8.Dibujar el árbol de encaminamiento del dispositivo R_5 .**SOLUCIÓN**El árbol de encaminamiento del dispositivo R_5 en este sistema autónomo será el siguiente:

14.9.

Escribir la tabla de encaminamiento resultante del Problema 14.8.

SOLUCIÓNLa tabla de encaminamiento resultante para el dispositivo R_5 será:

Destino	Siguiente salto	Distancia
N_1	R_7	9
N_2	R_7	4
N_3	R_7	7
N	—	2
N	R_7	5
N	R_7	6
H_1	R_7	8
H_2	—	1
H_3	R_7	6
H_4	R_7	7

Tabla 14.2. Tabla resultante.

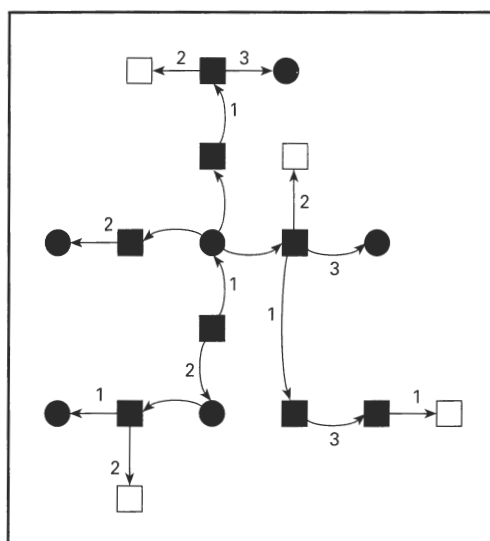
14.10.Dibujar el árbol de encaminamiento del dispositivo R_8 .**SOLUCIÓN**El árbol de encaminamiento del dispositivo R_8 en este sistema autónomo será el siguiente:

Figura 14.6. Árbol de encaminamiento.

14.11.

Escribir la tabla de encaminamiento resultante del Problema 14.10.

SOLUCIÓN

La tabla de encaminamiento resultante para el Dispositivo R_8 será:

Destino	Siguiente salto	Distancia
N_1	R_2	5
N_2	—	1
N_3	R_3	3
N_4	R_4	4
N_5	—	2
N_6	R_6	3
H_1	R_2	4
H_2	R_4	6
H_3	R_4	3
H_4	R_6	4

Tabla 14.3. Tabla resultante de R_8 .

14.12.

Explicar las principales diferencias de RIP y OSPF.

SOLUCIÓN• **RIP:**

- Es un protocolo de enrutamiento basado en el Algoritmo Bellman-Ford.
- Realiza un intercambio de información de retardo de líneas, transmitiendo toda la tabla de enrutamiento.

• **OSPF:**

- Es un protocolo de enrutamiento basado en el Algoritmo Dijkstra.
- Realiza un intercambio de información de retardo de enlace con los demás. Calcula rutas de menor coste, en función de parámetros del usuario.

14.13.

Dada la siguiente red de la figura siguiente, el sistema autónomo AS1 tiene el protocolo de encaminamiento interno OSPF. El dispositivo R_1 pertenece al sistema autónomo AS1, mientras que el dispositivo R_6 pertenece a otro sistema autónomo AS2. El dispositivo R_1 y el R_6 ejecutan el protocolo de encaminamiento externo BGP. Dibujar el flujo de paquetes que se realizaría para que R_1 comunique la información de su sistema autónomo AS1 al dispositivo R_6 del sistema autónomo AS2, recordando que el protocolo de encaminamiento externo BGP se encapsula en el protocolo de transporte TCP.

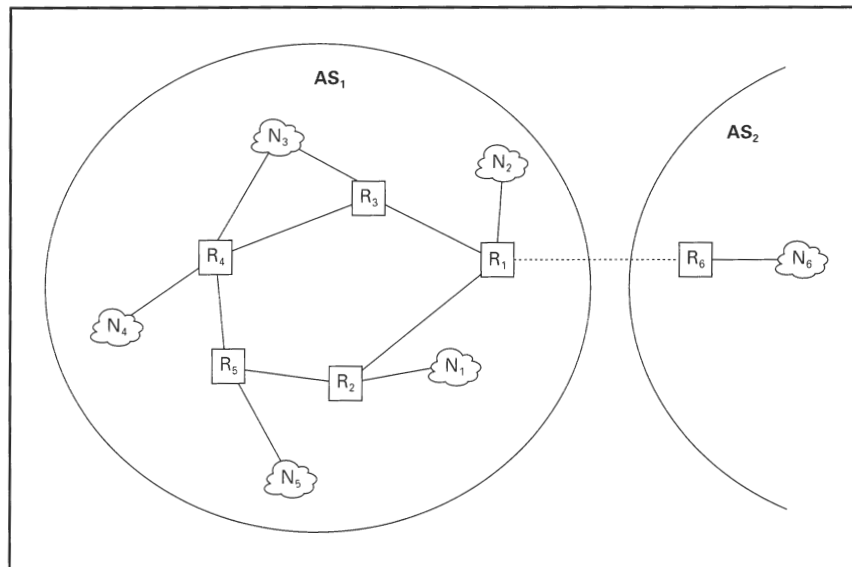


Figura 14.7. Red de sistemas autónomos.

SOLUCIÓN

La respuesta será el siguiente esquema de flujo de paquetes:

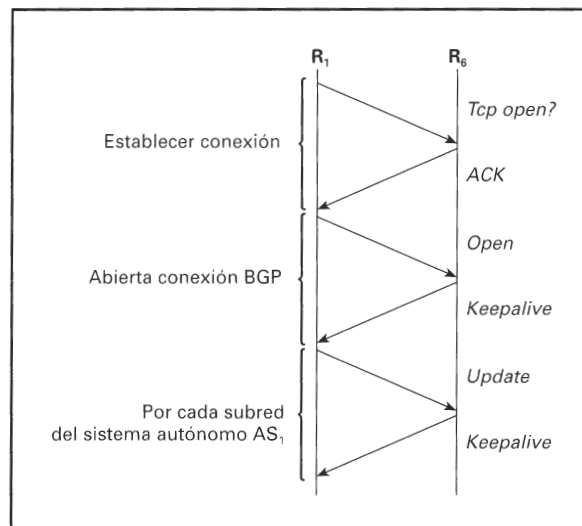


Figura 14.8. Flujo de paquetes.

La transmisión de mensajes BGP de Update y Keepalive se repetirá tantas veces como subredes tenga el sistema autónomo AS₁.

14.14.

Dados los router A, B y C que ejecutan el protocolo de encaminamiento BGP, dibujar los diagramas de flujo que A realizaría en los siguientes casos y razónelo:

- Adquisición de vecino.
- Detección de vecino alcanzable.

- Red alcanzable.
- Error de sintaxis.
- Tiempo de mantenimiento expirado.

SOLUCIÓN

Para el caso de adquisición de vecino, de un dispositivo A a otro dispositivo B tendrá el siguiente flujo de paquetes:

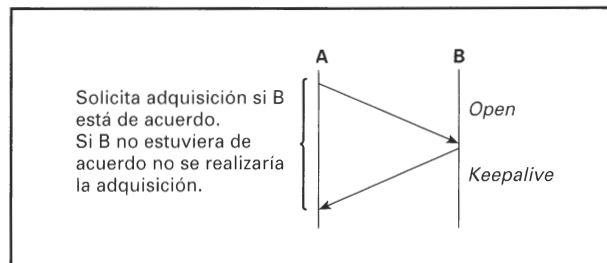


Figura 14.9. Adquisición de vecino.

En el caso de detección de un vecino alcanzable, se realizaría el siguiente flujo:

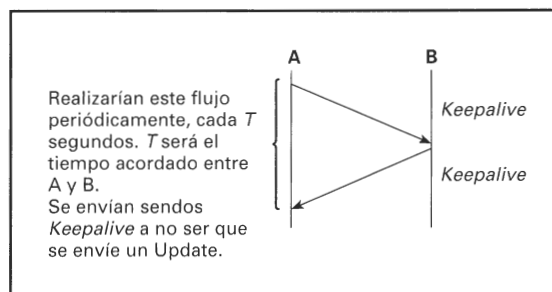


Figura 14.10. Detección de vecino alcanzable.

En el caso de red alcanzable, se realizaría el siguiente flujo:

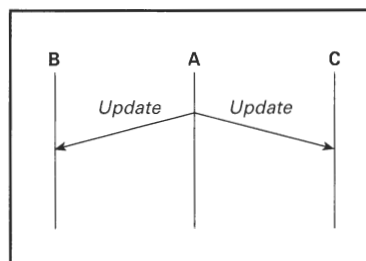


Figura 14.11. Red alcanzable.

El router A informa:

- A todas las redes alcanzables sobre su presencia.
- Y/o si hubiera algún posible cambio en alguna tabla de rutas de sus redes. En este caso lo informaría junto con la ruta preferida para dicho cambio.

En ambos casos, el router A lo informaría mediante el mensaje de difusión *Update*.

Para el flujo debido a un error de sintaxis se mostraría el siguiente flujo:

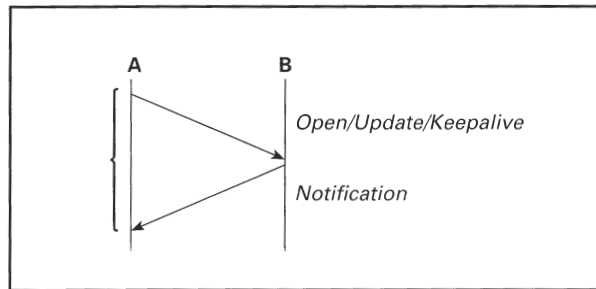


Figura 14.12. Error de sintaxis.

El router A intentará:

- Crear una conexión con el router B (con un mensaje BGP de *Open*).
- O tratará de informarle de algún cambio en alguna tabla de rutas (mediante un mensaje BGP de *Update*).
- O simplemente manteniendo la relación (mediante un mensaje BGP de *Keepalive*).

Pero el Router B le indica que no es posible, por un posible error de sintaxis que se lo notifica en un mensaje *Notification*.

Si el tiempo de mantenimiento ha expirado, el flujo de datos será el siguiente:

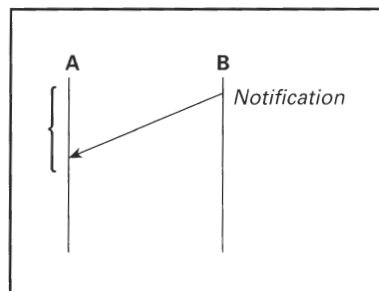


Figura 14.13. Tiempo de mantenimiento expirado.

El router A no ha mandado en el tiempo establecido un mensaje de presencia, mensaje BGP *Keepalive*, o ningún otro de actualización, mensaje BGP *Update*, por lo que el router B supone que el router A no tiene interés en mantener la alcanzabilidad, por lo que se lo notifica mediante un mensaje BGP *Notification*.

salto y NLRI (Network Layer Reachability Information). ¿Cambiará la tabla de rutas del dispositivo R4? ¿Por qué sí, o por qué no?

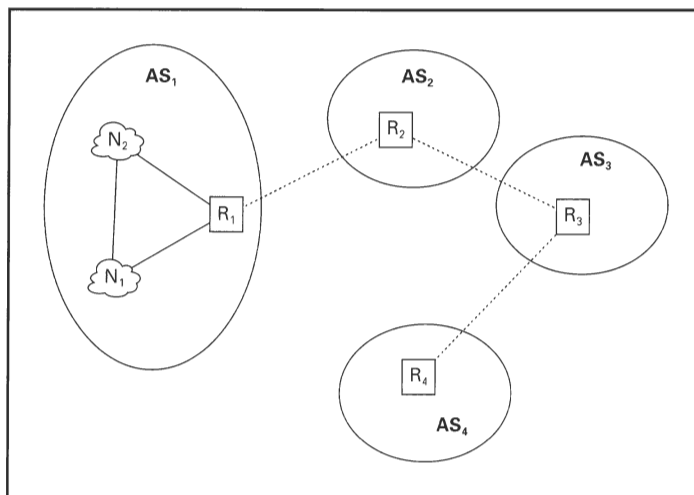


Figura 14.14. Red de sistemas autónomos.

SOLUCIÓN

Primeramente, el dispositivo de encaminamiento R_1 del sistema autónomo AS_1 envía un mensaje de actualización al dispositivo de encaminamiento R_2 del sistema autónomo AS_2 . Dicho mensaje será del tipo *Update* y contendrá los siguientes campos:

Atributos de camino:

- Camino_AS: $\{AS_1\}$
- Origen: OSPF
- Siguiendo salto: Dirección IP_{R_1} del dispositivo de encaminamiento R_1 .
- NLRI: Redes N_1 y N_2 del sistema autónomo AS_1 .

Posteriormente, el dispositivo de encaminamiento R_2 del sistema autónomo AS_2 envía un mensaje de actualización al dispositivo de encaminamiento R_3 del sistema autónomo AS_3 indicándole las modificaciones recibidas del dispositivo de encaminamiento R_1 . Dicho mensaje será del tipo *Update* y contendrá los siguientes campos:

Atributos de camino:

- Camino_AS: $\{AS_1, AS_2\}$
- Origen: BGP
- Siguiendo salto: Dirección IP_{R_1} del dispositivo de encaminamiento R_1 .
- NLRI: Redes N_1 y N_2 del sistema autónomo AS_1 .

Finalmente, el dispositivo de encaminamiento R_3 del sistema autónomo AS_3 envía un mensaje de actualización al dispositivo de encaminamiento R_4 del sistema autónomo AS_4 indicándole las modificaciones recibidas del dispositivo de encaminamiento R_2 para que las tenga en cuenta para cualquier cálculo. Dicho mensaje será del tipo *Update* y contendrá los siguientes campos:

Atributos de camino:

- Camino_AS: { AS_1 , AS_2 , AS_3 }
- Origen: BGP
- Siguiente salto: Dirección IP_{R_3} del dispositivo de encaminamiento R_3 .
- NLRI: Redes N_1 y N_2 del sistema autónomo AS_1 .

Una vez recibida toda la información necesaria, el dispositivo de encaminamiento R_4 decidirá si para acceder a las redes N_1 y N_2 le interesa o no pasar por los sistemas autónomos AS_1 , AS_2 y AS_3 . Puede darse el caso de que el dispositivo de encaminamiento R_4 del sistema autónomo AS_4 tenga un mejor camino para alcanzar las redes N_1 y N_2 del AS_1 . En función de su propia tabla de rutas y costes decidirá si le interesa o no acceder mediante AS_3 , AS_2 y AS_1 , y en dicho caso cambiará o no su tabla de rutas.

14.16.

Dado el sistema autónomo AS_1 con los dispositivos de encaminamiento R_1 , R_2 y R_3 , y el protocolo de encaminamiento RIP, OSPF en el sistema autónomo AS_1 , ¿cómo se informará al dispositivo R_4 (del sistema autónomo AS_2) de las redes N_1 , N_2 y N_3 mediante el protocolo de encaminamiento externo BGP?

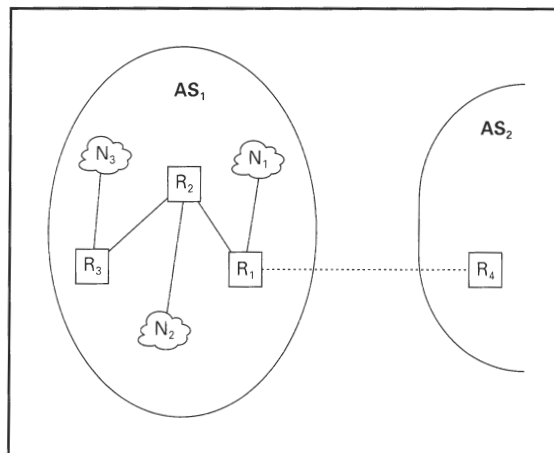


Figura 14.15. Red de sistemas autónomos.

SOLUCIÓN

El dispositivo de encaminamiento R_1 del sistema autónomo AS_1 , envía información de la red N_1 al dispositivo de encaminamiento R_4 , mediante el protocolo de encaminamiento exterior BGP, con un mensaje *Update*.

Posteriormente, el dispositivo de encaminamiento R_2 informa sobre la red N_2 que él observa al dispositivo de encaminamiento R_1 del sistema autónomo AS_1 con el protocolo de encaminamiento interior OSPF.

Una vez que el dispositivo de encaminamiento R_1 tiene la información sobre la red N_2 , le comunica al dispositivo de encaminamiento R_4 del sistema autónomo AS_2 sobre dicha red, con el protocolo de encaminamiento BGP, mediante un mensaje *Update*.

El dispositivo de encaminamiento R_3 del sistema autónomo AS_1 informa al dispositivo de encaminamiento R_2 de la presencia de la red N_3 , mediante el protocolo de encaminamiento interior, OSPF.

El dispositivo de encaminamiento R_2 que ahora tiene información sobre la red N_3 , informa de ésta al dispositivo R_1 que también es del sistema autónomo AS_1 , mediante el protocolo de encaminamiento interior OSPF.

Una vez que el dispositivo de encaminamiento R_1 tiene información de la red N_3 , le informa al dispositivo de encaminamiento R_4 del sistema autónomo AS_2 de dicha red alcanzable mediante el protocolo de encaminamiento exterior, BGP, con un mensaje de este protocolo del tipo *Update*.

14.17.

Considerando el problema anterior, ¿cambia la tabla de rutas de los dispositivos de encaminamiento R_2 y R_3 del sistema autónomo AS_1 , cuando el dispositivo R_1 (del sistema autónomo AS_1) decide la ruta de salida al sistema autónomo AS_2 ?

SOLUCIÓN

El dispositivo de encaminamiento R_1 informa de su ruta decidida para alcanzar el sistema autónomo AS_2 , y lo informa a los dispositivos de encaminamiento R_2 y R_3 mediante el protocolo de encaminamiento interno, OSPF. Si los dispositivos de encaminamiento R_2 y R_3 deciden que es el camino más corto para alcanzar el sistema autónomo AS_2 , lo incluirán en sus bases de datos, sus tablas de rutas.

Si el dispositivo de encaminamiento R_3 decide que el camino indicado no es el camino de menor coste al sistema autónomo AS_2 no cambiará su tabla de rutas. Lo mismo hará el dispositivo de encaminamiento R_2 , en el mismo lugar de no encontrar interesante dicho camino al sistema autónomo AS_2 .

14.18.

Dada la siguiente red de la figura de abajo, dibujar el grafo dirigido.

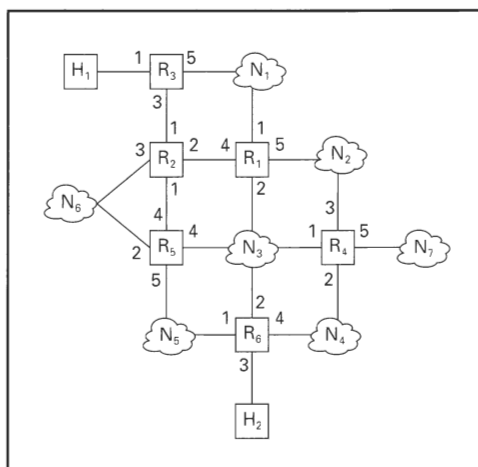


Figura 14.16. Sistema autónomo.

SOLUCIÓN

El grafo resultante será el siguiente:

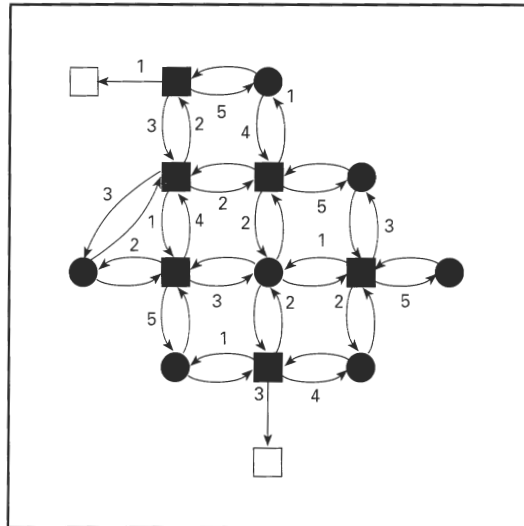


Figura 14.17. Grafo resultante.

14.19.

Dibujar el árbol de encaminamiento del dispositivo R_2 .

SOLUCIÓN

El árbol de encaminamiento del dispositivo R_2 en este sistema autónomo será el siguiente:

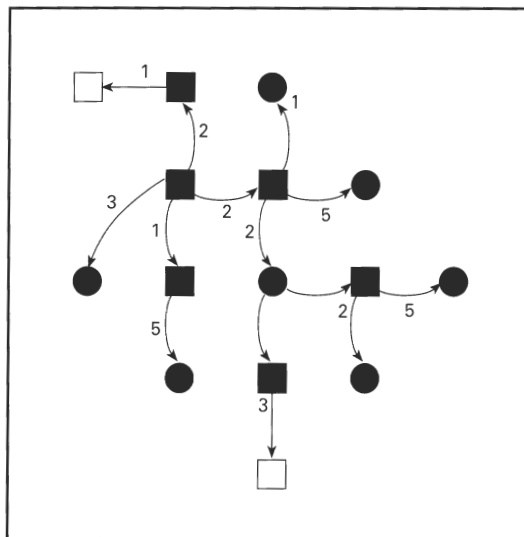


Figura 14.18. Árbol de encaminamiento.

14.20.

Escribir la tabla de encaminamiento resultante del Problema 14.19.

SOLUCIÓN

La tabla de encaminamiento resultante para el dispositivo R_2 será:

Destino	Siguiente salto	Distancia
N_1	R_3	7
N_2	R_1	7
N_3	R_1	4
N_4	R_1	6
N_5	R_5	6
N_6	—	3
N_7	R_1	9
H_1	R_3	3
H_2	R_1	7

Tabla 14.4. Tabla resultante.

14.21.

Los siguientes servicios de Internet se dividen en Tráfico Elástico y Tráfico No Elástico. Indica cuáles de ellos son de un tipo y cuáles otros de otro. Los servicios son: POP, VÍDEO BAJO DEMANDA, TELNET, ICMP, HTTPS, QUAKE, FTP, CHAT, VIDEOCONFERENCIA, IGMP, SMTP, SNMP, WEBCAM, VOZ SOBRE IP.

SOLUCIÓN

Tráfico Elástico	Tráfico No Elástico
POP, TELNET, HTTPS, FTP SMTP, SNMP, ICMP, IGMP	VÍDEO BAJO DEMANDA, QUAKE VIDEOCONFERENCIA, WEBCAM VOZ SOBRE IP

Tabla 14.5. Servicios de Internet.

14.22.

Muchas fuentes de tráfico pueden definir fácilmente un esquema de transmisión de paquetes que proporciona una descripción concisa de la carga que va a imponer un flujo, permitiendo al servicio determinar fácilmente la necesidad en recursos mediante los parámetros de entrada a la función para poder actuar como “policía”. En el siguiente esquema faltan por detallar los parámetros de entrada de dicho esquema. Se listan unos elementos que se deben colocar adecuadamente en su lugar correspondiente. ¿De qué tipo de esquema estamos hablando? ¿Qué son XXX e YYY?

- Datos entrada.
- Datos salida.
- Tamaño YYY: B/octetos.

- Capacidad ocupada actual de la YYY.
- Velocidad de XXX: R octetos IP por segundo.

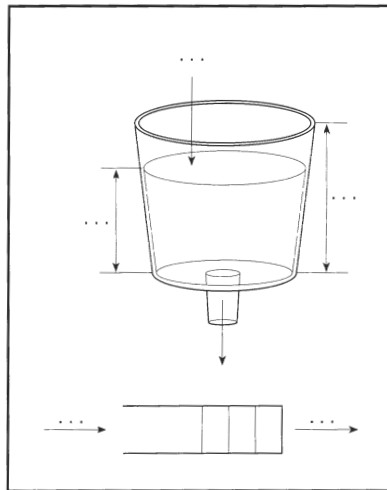


Figura 14.19. Esquema de transmisión de paquetes.

SOLUCIÓN

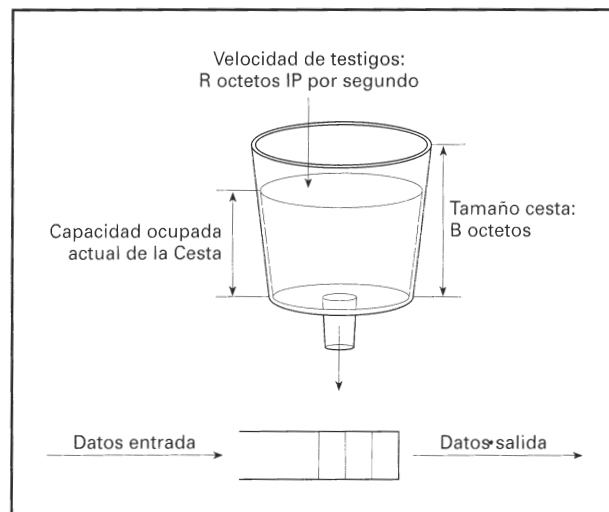


Figura 14.20. Esquema de transmisión de paquetes token bucket.

Esquema de la cesta de testigos.

- XXX es testigos.
- YYY es cesta.

14.23.

De la siguiente lista de definiciones, asociar los términos que se muestran en el recuadro, de modo que la definición se corresponda con el término empleado. ¿De qué tipo de servicio son los términos relacionados?

- A. Medición.
- B. Dominio DS.
- C. Modelado.
- D. Marcado.
- E. Código tipo de servicio.
- F. Nodo frontera DS.
- G. Nodo interior DS.
- H. Descarte.
- I. Clasificador.
- J. Acuerdo de acondicionamiento de tráfico (TCA).
- K. Nodo DS.
- L. Comportamiento en cada salto (PHB).
- M. Acuerdo de nivel de servicio (SLA).
- N. Acondicionamiento de tráfico.
- O. Agregados de comportamiento.

Un nodo DS que no es nodo frontera DS.	
Un acuerdo que especifica reglas de clasificación y reglas de acondicionamiento de tráfico que se aplican a los paquetes seleccionados por el clasificador.	
El proceso de fijar el código DS en un paquete. Los paquetes se pueden marcar en el inicio o pueden ser remarcados por un nodo DS en la ruta.	
El comportamiento de reenvío observable desde fuera, aplicado en un nodo a un agregado de comportamientos.	
Suministra servicios diferenciados. Normalmente, es un dispositivo de encaminamiento. Un ordenador que proporciona servicios diferenciados para sus aplicaciones también lo es.	
Un nodo DS que conecta un dominio DS a otro nodo en otro dominio.	
Un conjunto de paquetes con el mismo código DS cruzando un enlace en una dirección particular.	
Un valor específico de la porción DSCP de 6 bits del campo DS de 8 bits en la cabecera IP.	
Funciones de control implementadas para forzar reglas especificadas en un TCA, que incluye medición, marcado, modelado y descarte.	
Paquetes seleccionados basándose en el campo DS (clasificador BA) o en campos múltiples dentro de la cabecera del paquete (clasificador MF).	
Un contrato de servicio entre un cliente y un proveedor de servicios.	
Un conjunto continuo (conectados) de nodos, capaces de implementar servicios diferenciados, que operan con un conjunto común de políticas de suministro de servicios y definiciones de comportamiento en cada salto.	
El proceso de tirar paquetes basándose en reglas específicas; también llamadas políticas.	
El proceso de retrasar paquetes en un flujo de paquetes para ver si se ajusta a algún perfil de tráfico predefinido.	
El proceso de medir las propiedades temporales (por ejemplo, la velocidad de transferencia) del flujo de paquetes por un medidor. El estado instantáneo de ese proceso podría afectar las funciones de marcado, modelado y de descarte.	

Tabla 14.6. Tabla de definiciones.

SOLUCIÓN

- | | |
|-----------------------------|---|
| A. Medición. | I. Clasificador. |
| B. Dominio DS. | J. Acuerdo de acondicionamiento de tráfico (TCA). |
| C. Modelado. | K. Nodo DS. |
| D. Marcado. | L. Comportamiento en cada salto (PHB). |
| E. Código tipo de servicio. | M. Acuerdo de nivel de servicio (SLA). |
| F. Nodo frontera DS. | N. Acondicionamiento de tráfico. |
| G. Nodo interior DS. | O. Agregados de comportamiento. |
| H. Descarte. | |

Un nodo DS que no es nodo frontera DS.	G
Un acuerdo que especifica reglas de clasificación y reglas de acondicionamiento de tráfico que se aplican a los paquetes seleccionados por el clasificador.	J
El proceso de fijar el código DS en un paquete. Los paquetes se pueden marcar en el inicio o pueden ser remarcados por un nodo DS en la ruta.	D
El comportamiento de reenvío observable desde fuera, aplicado en un nodo a un agregado de comportamientos.	L
Suministra servicios diferenciados. Normalmente, es un dispositivo de encaminamiento. Un ordenador que proporciona servicios diferenciados para sus aplicaciones también lo es.	K
Un nodo DS que conecta un dominio DS a otro nodo en otro dominio.	F
Un conjunto de paquetes con el mismo código DS cruzando un enlace en una dirección particular.	O
Un valor específico de la porción DSCP de 6 bits del campo DS de 8 bits en la cabecera IP.	E
Funciones de control implementadas para forzar reglas especificadas en un TCA, que incluye medición, marcado, modelado y descarte.	N
Paquetes seleccionados basándose en el campo DS (clasificador BA) o en campos múltiples dentro de la cabecera del paquete (clasificador MF).	I
Un contrato de servicio entre un cliente y un proveedor de servicios.	M
Un conjunto continuo (conectados) de nodos, capaces de implementar servicios diferenciados, que operan con un conjunto común de políticas de suministro de servicios y definiciones de comportamiento en cada salto.	B
El proceso de tirar paquetes basándose en reglas específicas; también llamadas políticas.	H
El proceso de retrasar paquetes en un flujo de paquetes para ver si se ajusta a algún perfil de tráfico predefinido.	C
El proceso de medir las propiedades temporales (por ejemplo, la velocidad de transferencia) del flujo de paquetes por un medidor. El estado instantáneo de ese proceso podría afectar las funciones de marcado, modelado y de descarte.	A

Tabla 14.7. Tabla de definiciones (solución).

Se trata del servicio diferenciado (DS, *Differentiated Service*).

14.24.

Colocar en el sitio adecuado los siguientes términos en la siguiente figura. En un mismo lugar pueden coincidir más de un término, y algún término se puede repetir.

- Clasificador.
- Gestor de cola.
- Modelador.
- Dominio DS.
- Marcador.
- Nodo DS frontera.
- Elemento de descarte.
- Medidor.
- Nodo DS interior.

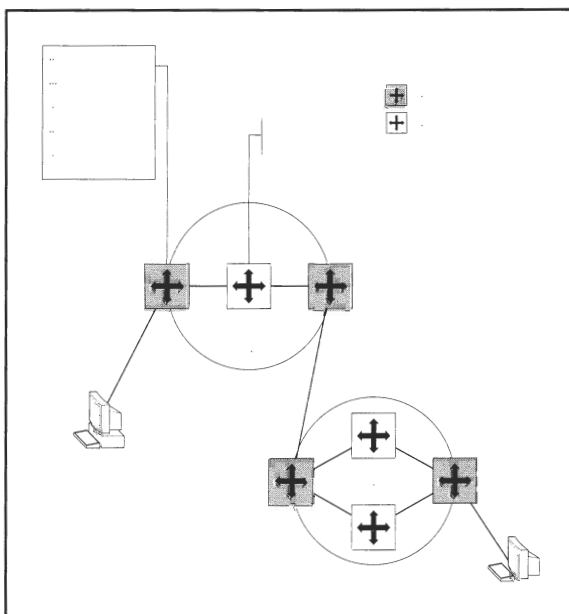


Figura 14.21. Términos de una red de sistemas autónomos.

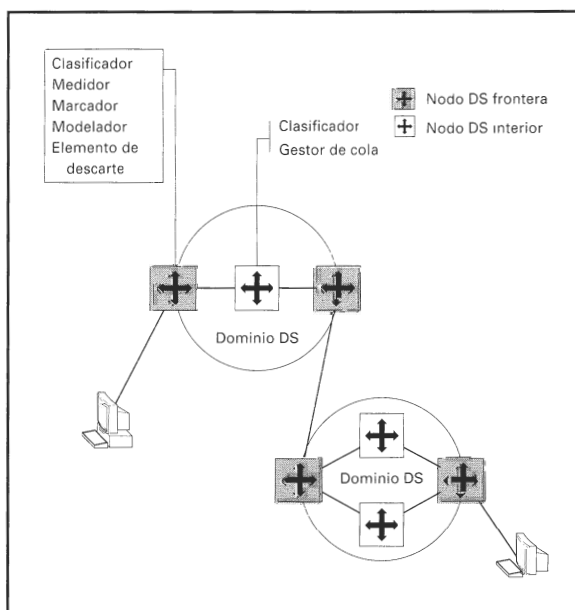
SOLUCIÓN

Figura 14.22. Términos de red (solución).

14.25. Colocar adecuadamente los términos siguientes en la figura.

- Acondicionamiento de tráfico DS.
- Clasificador.
- Marcador.
- Medidor.
- Modelador / elemento de descarte.
- Paquetes.

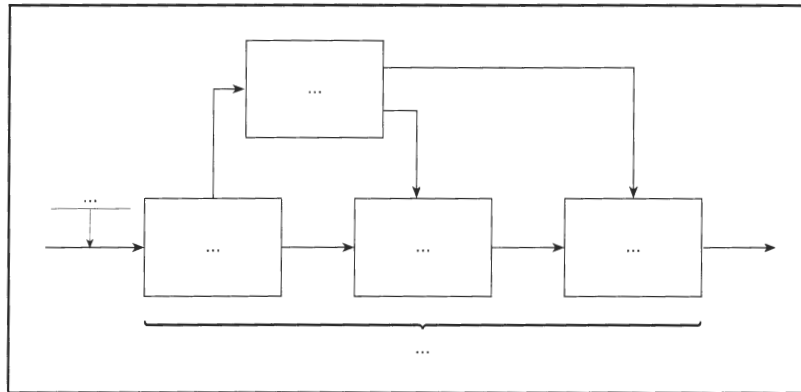


Figura 14.23. Tratamiento del tráfico (términos).

SOLUCIÓN

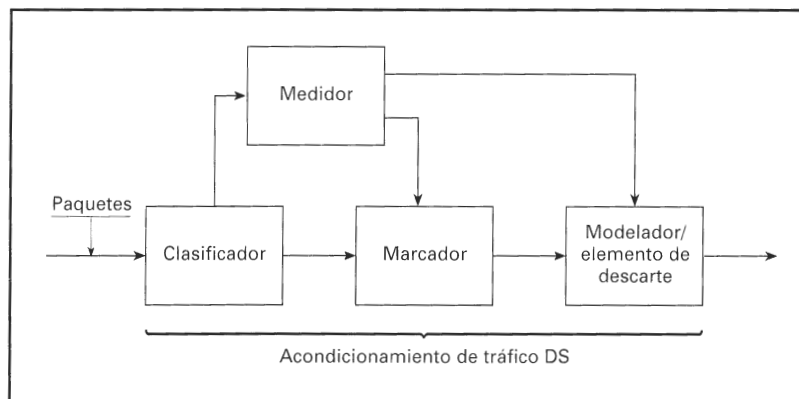


Figura 14.24. Tratamiento del tráfico (solución).

14.26. ¿Debería el encaminamiento en una interconexión de redes preocuparse del encaminamiento interno de la red? ¿Por qué o por qué no?

SOLUCIÓN

No debería preocuparse debido a que existen protocolos de encaminamiento independientes para dentro de una misma red (protocolo interior de dispositivos de encaminamiento, IRP) y para la interconexión de redes (protocolo de dispositivos de encaminamiento exterior, ERP). Por ejemplo, para un encaminamiento interno existe el protocolo OSPF, y para el protocolo de encaminamiento externo existe por ejemplo, el protocolo BGP.

14.27.

Cuando existen múltiples rutas de igual coste a un destino, OSPF puede distribuir el tráfico de igual forma entre las rutas. Esto se llama Balanceamiento de Carga. ¿Qué efectos tiene este Balanceamiento de Carga en el protocolo de la capa de transporte, como es TCP?

SOLUCIÓN

El balanceamiento de Carga reduce la congestión en una ruta concreta, repartiendo la carga con rutas menos congestionadas y de menor coste a un destino. El protocolo de la capa de transporte, como por ejemplo TCP, que es un protocolo orientado a la conexión, tiene un control de congestión de modo que si se pierden paquetes de la conexión (por congestión) o detecta congestión en la ruta, reduce su velocidad de transmisión moldeándose a la carga de red actual. Al realizarse balanceo de carga, detecta que la congestión es menor, por lo que no reduce la velocidad de transmisión y la realización del balanceo de carga favorece la transmisión con el protocolo de la capa de transporte, TCP.

14.28.

El esquema de cesta de testigos establece un límite en la duración del tiempo en el que el tráfico puede transmitirse a la velocidad de transmisión máxima. Si definimos la cesta de testigos por un tamaño de cesta de B octetos y una tasa de llegada de testigos de R octetos por segundo y si la velocidad de transferencia máxima de salida es de M octetos por segundo:

- Obtener una fórmula para S , la longitud de la ráfaga a velocidad máxima. Esto es, ¿durante cuánto tiempo puede un flujo transmitir a la velocidad máxima de salida cuando está gobernado por una cesta de testigos?
- ¿Cuál es el valor de S para $B = 250$ KB, $R = 2$ MBps y $M = 25$ MBps? (Sugerencia: La fórmula de S no es tan fácil como pudiera parecer, ya que llegan más testigos mientras la ráfaga se está enviando.)

SOLUCIÓN

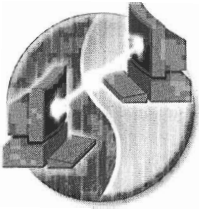
- Con una cesta de testigos se necesitan testigos para poder transmitir. Al transmitir paquetes se utilizan tantos testigos como octetos se transmiten en el paquete. Si se considera que se tiene un tiempo de reposo y la cesta está llena de testigos, se tienen B octetos para transmitir aparte de los R octetos que llegan cada segundo. De esta forma, para cualquier período de tiempo T , se tiene una transmisión, tras un tiempo de reposo, de $B + RT$ octetos. Si no se tiene un tiempo de reposo, es decir, que previamente se han transmitido paquetes de modo que la cesta está vacía, la velocidad máxima para transmitir será de R octetos por segundo, por lo que para un período de tiempo cualquiera T se tendrá RT octetos enviados.

Si S tiene la unidad de tiempo, y es el período de tiempo en el que se quiere que la transmisión sea máxima, es decir M octetos por segundo, se transmitirán MS octetos en ese período. Con la cesta llena de testigos, B octetos, se tendrá para un período S una transmisión de $B + RS$ octetos. Igualando los octetos que se quieren transmitir con los que se transmiten se tiene la siguiente fórmula: $MS = B + RS$.

Se despeja el tiempo a calcular resultando la siguiente fórmula:

$$S = \frac{B \text{ (octetos)}}{M \text{ (octetos / s)} - R \text{ (octetos / s)}} \text{ (s)}$$

- Para unos valores dados, de $B = 250$ KB (Kilo Bytes), $R = 2$ MBps (Mega Bytes por segundo), y $M = 25$ MBps (Mega Bytes por segundo), se obtiene un valor de $S = 0,01087$ s.



PROTOCOLOS DE TRANSPORTE

15.1.

Para los protocolos orientados a la conexión, uno de los grandes problemas que se presenta es el control de flujo. La entidad de transporte receptor puede llegar a saturarse si no modera la tasa de transmisiones de segmentos por saturación de las memorias temporales. Una de las estrategias que se presenta para hacer frente a este requisito es la utilización de un esquema de créditos. Este esquema desliga las confirmaciones del control de flujo, donde cada octeto individual de datos que se transmite se considera que tiene un número de secuencia. Además de datos, cada segmento transmitido incluye en su cabecera tres campos relacionados con el control de flujo: un número de secuencia (*SN*), un número de confirmación (*AN*) y la ventana (*W*). Al enviar un segmento, se incluye el número de secuencia del primer octeto en el campo de datos del segmento.

Se va a suponer un caso por simplicidad de flujo de datos en un solo sentido, de la entidad de transporte *A* a la entidad de transporte *B*. Para mayor sencillez, los segmentos serán de 100 octetos cada uno. Inicialmente a través del procedimiento de establecimiento de la conexión, los números de secuencia de emisión y de recepción están sincronizados. *A* tiene una asignación de crédito inicial de 800 octetos, y comenzando por el octeto de número 2.500.

Se pide dibujar el control de flujo de las entidades de transporte *A* y *B*, junto con el flujo de datos, indicando *SN*, *AN* y *W*. Se mostrarán unos estados en los que se pide los estados de flujo de la entidad de transporte que se presente entre paréntesis.

- Estado inicial (*A* y *B*).
- *A* transmite 4 segmentos (*A*).
- *B* responde los 4 segmentos confirmados y aumenta en 2 segmentos más su crédito inicial (*B*).
- *A* antes de recibir el asentimiento transmite un segmento más (*A*).
- *B* recibe el segmento pero no lo confirma (*B*).
- *A* transmite, tras recibir el asentimiento, 4 segmentos más (*A*).
- *B* transmite el asentimiento de los 5 segmentos y aumenta en un segmento más el crédito (*B*).
- *A* transmite otro segmento antes de la confirmación (*A*).
- *A* transmite otro segmento después de recibir la confirmación (*A*).
- *B* no envía asentimiento (*A* y *B*).
- *B* envía asentimiento de los 2 segmentos recibidos y restablece la ventana en 8 segmentos (*A* y *B*).

SOLUCIÓN

Se muestra tras cada estado, junto con el control de flujo de cada entidad de trabajo.

Estado inicial (*A* y *B*).

Entidades transporte *A* y *B*:

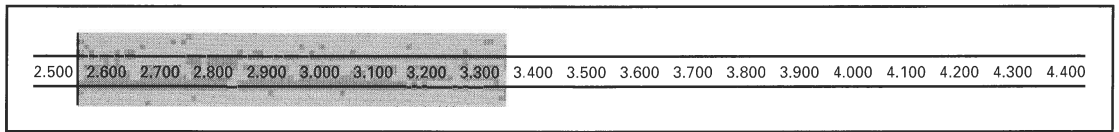


Figura 15.1. Control y flujo de *A* y *B*.

A puede transmitir 800 octetos, desde 2.501 hasta 3.300 y tiene confirmados hasta el número de secuencia 2.500.

B puede recibir 800 octetos, tiene suficiente memoria temporal, y tiene recibidos hasta el número de secuencia 2.500.

- *A* transmite 4 segmentos (*A*).

Entidad de Transporte *A* tras enviar 4 segmentos:

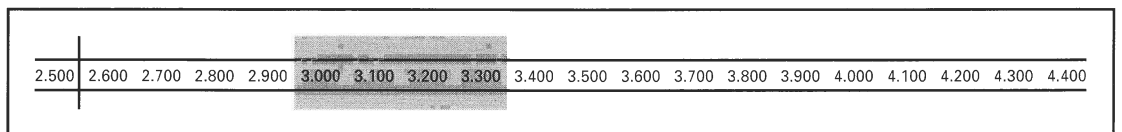


Figura 15.2. Control de flujo de *A*.

A puede transmitir todavía 400 octetos más, desde 2.901 hasta 3.300 y tiene confirmados hasta el número de secuencia 2.500.

Flujo de datos de *A* a *B*:

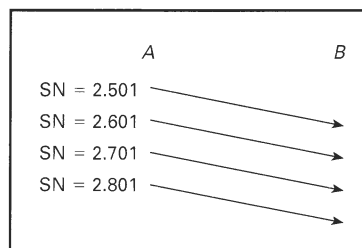


Figura 15.3. Flujo de datos.

- *B* responde los 4 segmentos confirmados y aumenta en 2 segmentos más su crédito inicial (*B*).

Entidad de transporte *B* tras recibir y confirmar 4 segmentos y aumentado en 2 segmentos el crédito:

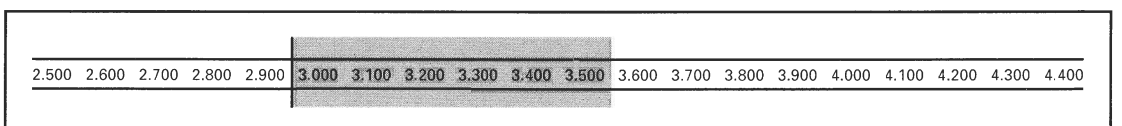


Figura 15.4. Control de flujo de *B*.

B ya ha confirmado hasta el octeto 2.900 y puede recibir todavía 400 octetos, más 200 más, desde 2.901, hasta 3.500 y tiene confirmados hasta el número de secuencia 2.900.

Flujo de datos de *B* a *A*:

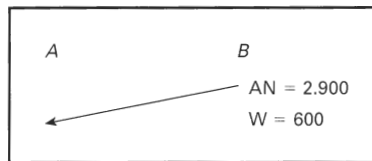


Figura 15.5. Flujo de datos.

- *A* antes de recibir el asentimiento transmite un segmento más (*A*).

Entidad de transporte *A* tras enviar un segmento más:

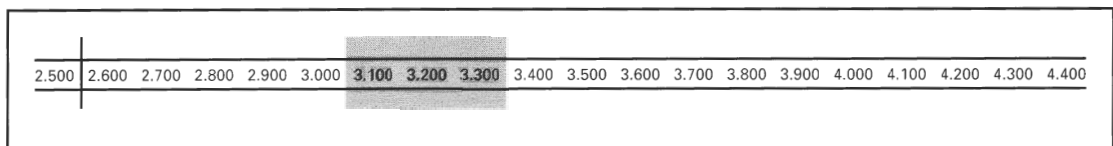


Figura 15.6. Control de Flujo de *A*.

A puede transmitir 300 octetos, desde 3.001, hasta 3.300 y tiene confirmados hasta el número de secuencia 2.500.

Flujo de datos de *A* a *B*:

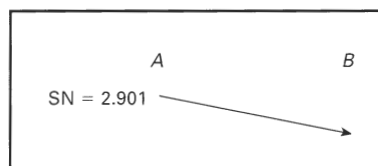


Figura 15.7. Flujo de datos.

- *B* recibe el segmento pero no lo confirma (*B*).

Entidad de transporte *B* tras recibir el segmento:

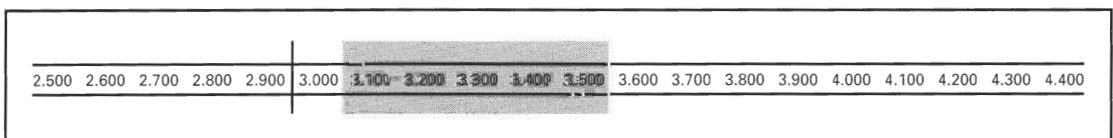


Figura 15.8. Control de flujo de *B*.

B puede recibir 500 octetos más, desde 3.301, hasta 3.500 y tiene confirmados hasta el número de secuencia 2.900.

No hay flujo de datos de *B* a *A*, porque *B* no confirma.

- *A* transmite tras recibir el asentimiento, 4 segmentos más (*A*).

Entidad de transporte *A* tras recibir asentimiento y trasmitir 4 segmentos más:

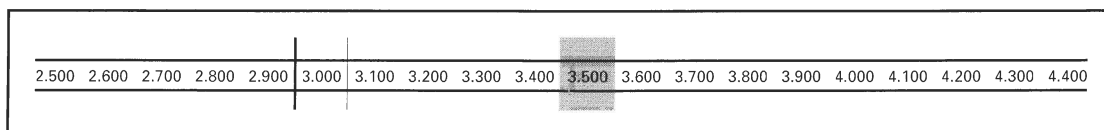


Figura 15.9. Control de flujo de A.

A puede transmitir 100 octetos más, desde 3.401 hasta 3.500 y tiene confirmados hasta el número de secuencia 2.900.

Flujo de datos de A a B:

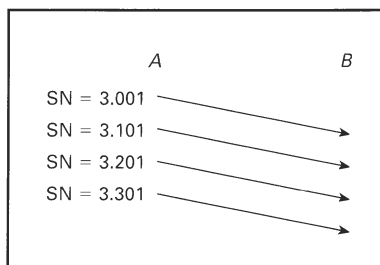


Figura 15.10. Flujo de datos.

- B transmite el asentimiento de los 5 segmentos y aumenta en un segmento más el crédito (B).

Entidad de transporte B:

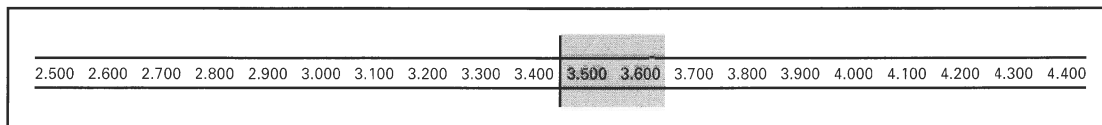


Figura 15.11. Control de flujo de B.

B puede recibir 200 octetos más, desde 3.401 hasta 3.600 y tiene confirmados hasta el número de secuencia 3.400.

Flujo de datos de B a A:

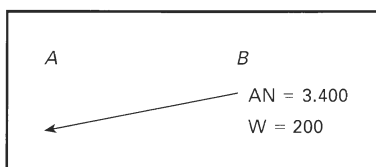


Figura 15.12. Flujo de datos.

- A transmite otro segmento antes de la confirmación (A).

Entidad de transporte A tras enviar otro segmento antes de la confirmación de los anteriores:

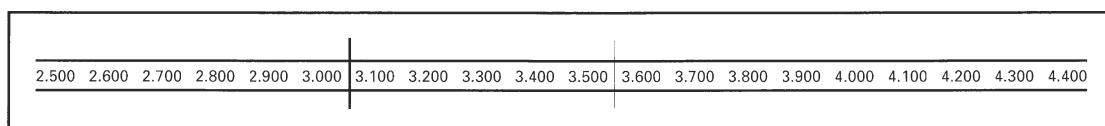


Figura 15.13. Control de flujo de A.

A ya no puede transmitir más, y tiene confirmados hasta el número de secuencia 3.000.

Flujo de datos de A a B:

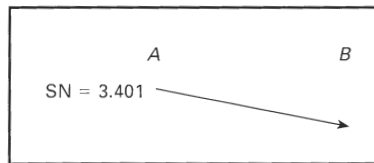


Figura 15.14. Flujo de datos.

- A transmite otro segmento después de recibir la confirmación (A).

Entidad de transporte A antes de enviar y tras recibir el asentimiento:

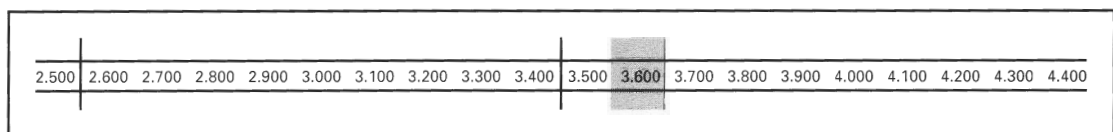


Figura 15.15. Control de flujo de A.

A puede transmitir 100 octetos más (que a continuación se ve el flujo de datos que lo envía), y tiene confirmados hasta el número de secuencia 3.400.

Flujo de datos de A a B:

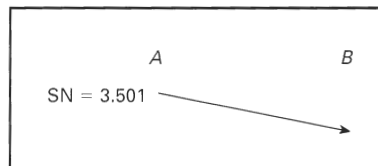


Figura 15.16. Flujo de datos.

- B no envía asentimiento (A y B).

Entidad de transporte A tras enviar el segmento:

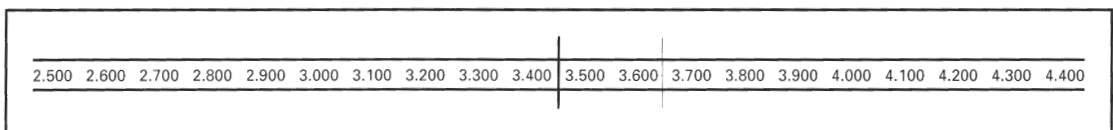


Figura 15.17. Control de flujo de A.

A ya no puede transmitir más, y tiene confirmados hasta el número de secuencia 3.400.

Entidad de transporte B antes de enviar el asentimiento y tras recibir el último segmento de A:

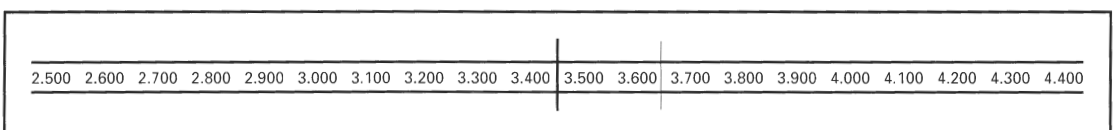


Figura 15.18. Control de flujo de A.

B no puede recibir más, no tiene suficiente memoria temporal, y tiene recibidos hasta el número de secuencia 3.600, pero sólo confirmados hasta el 3.400.

No hay flujo de datos de *B* a *A*.

- *B* envía asentimiento de los 2 segmentos recibidos y restablece la ventana en 8 segmentos (*A* y *B*).

Entidades transporte *A* y *B*:

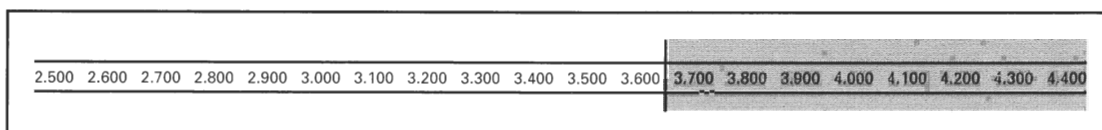


Figura 15.19. Control de flujo de *A* y *B*.

A puede transmitir 800 octetos, desde 3.601, hasta 4.400 y tiene confirmados hasta el número de secuencia 3.600.

B puede recibir 800 octetos más, tiene suficiente memoria temporal, y tiene recibidos hasta el número de secuencia 3.600.

Flujo de datos de *B* a *A*, asentimiento de 2 segmentos y restableciendo la ventana de crédito, los 800 octetos:

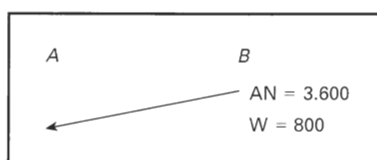


Figura 15.20. Flujo de datos.

15.2

En la capa de transporte cuando el servicio de red es no seguro, se generan unos problemas a tratar. Uno de estos problemas es el duplicado de segmentos, y la detección de éstos. Una de las tácticas que se plantea para la detección de duplicados antes de cerrar la conexión es que cada segmento tenga un número de secuencia que aunque ayuda, no soluciona todos los problemas. Se plantea que el espacio de números de secuencia debe ser lo suficientemente grande para no agotarse en menos tiempo que la vida máxima posible de un segmento.

Se plantea el siguiente caso: Se tiene un espacio del número de secuencia de 1.000 octetos, y cada segmento por simplicidad contendrá 200 octetos. Se transmiten los segmentos con un tasa de transmisión de 53,33 Kbps y el tiempo entre segmentos transmitidos es el 50 ms. Por simplicidad, se supone que se utiliza el protocolo de ventana deslizante.

- Si se tiene un tamaño de ventana de 1.400 octetos, y el tiempo máximo de vida de un segmento es de 0,29 s, ¿el espacio de número de secuencia es lo suficientemente grande? ¿Cuánto debería ser el máximo tiempo de vida de un segmento para el espacio de número de secuencia igual a 1.000?
- Si se tiene un tamaño de ventana de 400 octetos, y tiempo máximo de vida del segmento de 0,26 s, ¿el espacio de número de secuencia es lo suficientemente grande? ¿Cuánto es el máximo tiempo de vida que se puede permitir para un número máximo de secuencia de 1.000?

SOLUCIÓN

Antes de diferenciar los dos casos se realizarán cálculos conjuntos. Cada segmento se transmite cada 0,05 segundos, y el tiempo que se tarda en llegar al destino depende de la tasa de transmisión. Los segmentos de 200 octetos a una tasa de 53,33 Kbps, son 200 bytes, es decir 1.600 bits y resulta un tiempo de 0,03 s aproximadamente.

Los segmentos se transmiten en los tiempos (T) , $(T + 0,05)$, $(T + 2 \cdot 0,05)$, $(T + 3 \cdot 0,05)$, etc.

- a) Con un tamaño de ventana de 1.400 octetos, eso quiere decir que se pueden transmitir sin necesidad de asentimiento (por tener el asentimiento la propiedad de ser acumulativo), todos los segmentos seguidos hasta dar la vuelta al número de secuencia. Supongamos el peor caso, que se transmiten 5 segmentos (con números de secuencia, 1, 201, 401, 601 y 801) pero el primer segmento se retrasa por circunstancias de la red. Empezando en un tiempo $T = 0$, se transmiten los 5 segmentos, por lo que el quinto segmento se recibe en el $T = 0,05 \cdot 4 + 0,03 = 0,23$ s. El siguiente segmento es el segmento 1.001, que por ser el espacio de número de secuencia hasta 1.000, será el número de secuencia número 1. Si entonces llega el número de secuencia 1, correspondiente al segmento 1.001, en $T = 5 \cdot 0,05 + 0,03 = 0,28$ s, el receptor puede pensar que es el segmento de número de secuencia primero que se ha retrasado, por lo que lo confundirá con el segmento del octeto 1.001, y enviará un asentimiento de los 1.000 primeros octetos. Puesto que el tiempo máximo de vida de un segmento es de 0,29 s, puede llegar después el primer segmento retrasado, y en este caso el receptor puede pensar que se trata del octeto 1.001 al que le corresponde un número de secuencia igual a 1. El flujo de datos sería el siguiente.

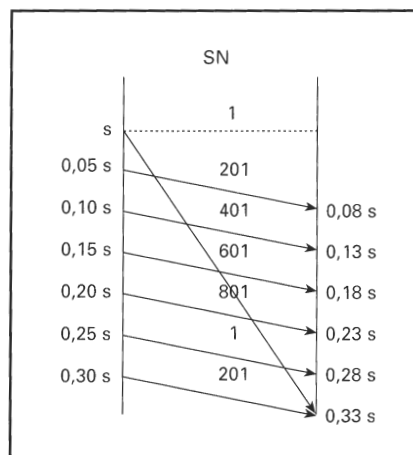


Figura 15.21. Flujo de datos.

Esto demuestra que no es suficientemente grande el número de secuencia. El tiempo máximo de vida debería ser 0,22 s, de forma que no pueda llegar antes que el segmento previo a la vuelta del número de secuencia, es decir antes que el segmento del octeto 801, porque así no se podrá suponer que es posterior a éste.

- b) Con un tamaño de ventana de 400 octetos, éstos deben ser confirmados antes de poder seguir con otros segmentos. El flujo de datos resultante sería el de la Figura 15.22.

Con el peor caso, el expuesto en el apartado anterior, el primer paquete se retarda, pero por ser el tamaño de la ventana de 400 octetos, al recibir los primeros 400 octetos. Y al no recibir asentimiento por parte del receptor, el transmisor vuelve a enviar los primeros segmentos, al recibirse el segmento con el octeto número 1, se asienten los 400 octetos recibidos. Antes que

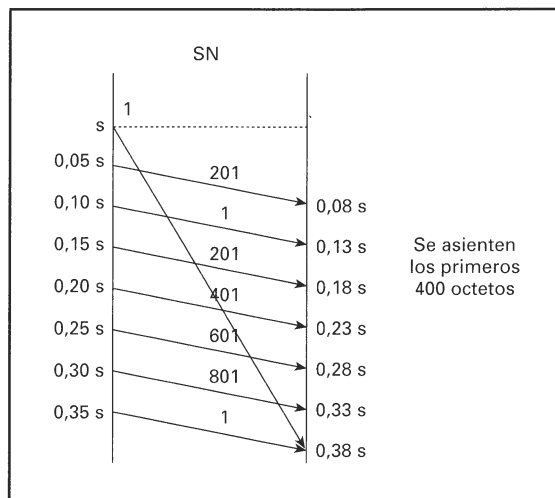


Figura 15.22. Flujo de datos.

el transmisor reciba el asentimiento, ya ha enviado otra vez el segmento del octeto 201, que se duplica, pero el receptor lo descarta. A continuación (por claridad no se han dibujado las respuestas del receptor), el receptor va asentando cada uno de los segmentos que se reciben para que el transmisor pueda transmitir más segmentos. De esta forma se tiene un tiempo en el que se recibe el quinto segmento, que es el tiempo 0,33 segundos. El siguiente segmento que tendrá el número de secuencia otra vez de número 1 llegará en un instante 0,38 segundos. Puesto que el tiempo máximo de vida de un segmento es de 0,29 segundos, la red no dejará que ese segmento llegue tras el quinto segmento y pueda confundirse con el segmento siguiente. Es suficientemente grande el número de secuencia, y el tiempo de vida máximo que se puede permitir es de 0,32 segundos (justo antes que el segmento con el último número de secuencia antes de volver a empezar los números de secuencia desde 1).

15.3.

Para el cálculo del temporizador de retransmisión se utiliza el tiempo de ida y vuelta de los segmentos enviados. El valor estimado puede ser el promedio simple o el promedio exponencial, que depende de una variable α .

Dada la siguiente traza, calcular por el promedio simple y por el promedio exponencial para $\alpha = 0,5$ y $\alpha = 0,875$. Suponer $SRTT(0) = 0$ y $ARTT(1) = RTT(1)$. Los datos son del número de Segmento (S) recibido y en la fila inferior, el tiempo de ida y vuelta Observado (O), el llamado RTT .

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
O	1	2	3	4	5	5	5	5	4	3	2	2	2	3	3	4	5	5	6	7	8	8	8	8	8	8

¿Cuál de las estimaciones se aproxima mejor al tiempo real de ida y vuelta?

SOLUCIÓN

Para el cálculo de los promedios se tienen las siguientes ecuaciones:

a) **Promedio Simple:** $ARTT(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} RTT(i)$ con $ARTT = RTT$ medio

b) **Promedio Exponencial:** $SRTT(K+1) = \alpha \cdot SRTT(K) + (1 - \alpha) \cdot RTT(K+1)$ con $SRTT = RTT$ estimado
Suponer $SRTT(0) = 0$

Tras los cálculos, la gráfica que resulta es la siguiente:

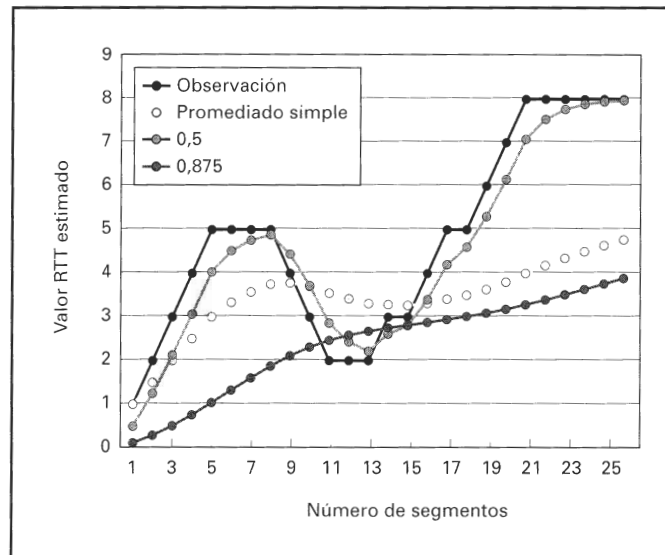


Figura 15.23. Estimación del tiempo de ida y vuelta.

Se aprecia cómo el promedio exponencial de $\alpha = 0,5$ se acerca más rápido en los cambios del *RTT* que el promedio simple. Pero para valores de $\alpha = 0,875$, es mejor el promedio simple. Cuanto más pequeña sea α más rápido cambiará ante cambios del *RTT*, por dar mayor peso a los tiempos de ida y vuelta más recientes, aunque también se modificará con cualquier cambio temporal del tiempo de ida y vuelta.

15.4.

También está el Algoritmo de Jacobson para el cálculo del temporizador de retransmisión (*RTO*) en el que se utiliza el tiempo de ida y vuelta de los segmentos enviados, pero con variables de ponderación de pesos.

Dada la siguiente traza, calcular por el promedio simple de la desviación (*ADEV*), el promedio exponencial de la desviación media (*SDEV*), con $h = 1/4$ y $g = 1/8$ y el tiempo de retransmisión del algoritmo de Jacobson, *RTO* con $f = 2$ y $f = 4$. Suponer $SRTT(0) = 0$, $ARTT(1) = RTT(1)$ y $SDEV(0) = 0$. Los datos son del número de Segmento (*S*) recibido y en la fila inferior, el tiempo de ida y vuelta Observado (*O*), el llamado *RTT*.

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
O	1	2	3	4	5	5	5	5	4	3	2	2	2	3	3	4	5	5	6	7	8	8	8	8	8	8

¿Cuál de las estimaciones se aproxima mejor al tiempo real de ida y vuelta?

SOLUCIÓN

Para el cálculo de los promedios se tienen las siguientes ecuaciones:

a) **Promedio Simple de la desviación:**

$$ADEV(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} |AERR(i)| \text{ con}$$

$AERR(K+1) = RTT(K+1) - ARTT(K)$ Desviación media, siendo

$$RTT(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} RTT(i) \text{ el } RTT \text{ medio}$$

$$RTT(1) = RTT(1)$$

b) Promedio exponencial de Jacobson:

$$SDEV(K+1) = (1-h) \cdot SDEV(K) + h \cdot |SERR(K+1)| \text{ con}$$

$$SERR(K+1) = RTT(K+1) - SRTT(K) \text{ y}$$

$$SRTT(K+1) = (1-g) \cdot SRTT(K) + g \cdot RTT(K+1) \text{ el } RTT \text{ estimado.}$$

$$\text{Suponer } SRTT(0) = 0 \text{ y } SDEV(0) = 0.$$

c) El tiempo de retransmisión de Jacobson:

$$RTO(K+1) = SRTT(K+1) + f \cdot SDEV(K+1)$$

Tras los cálculos, la gráfica que resulta es la siguiente:

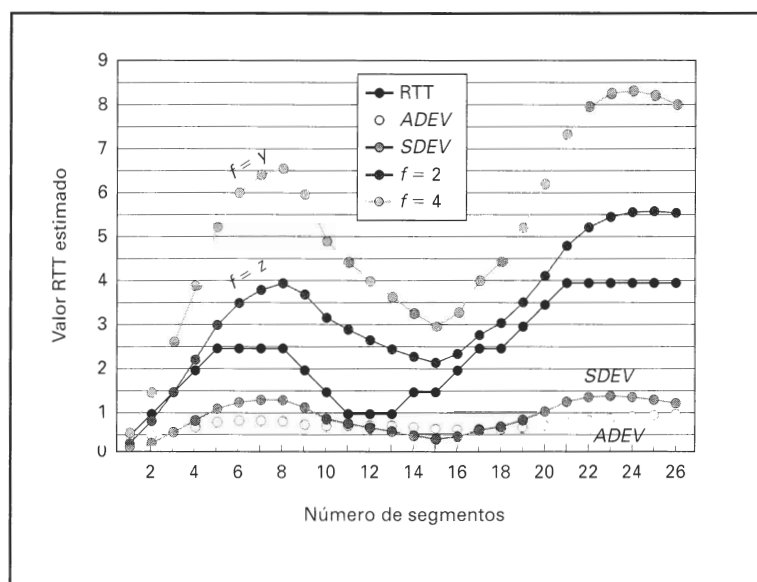


Figura 15.24. Estimación del tiempo de ida y vuelta.

Se aprecia cómo el promedio exponencial de la desviación, *SDEV*, se acerca más rápido en los cambios del *RTT* que el promedio simple de la desviación, *ADEV*. El caso de $\alpha = 0,875$ es el visto en el ejercicio anterior, que es con $g = 0,125$. Respecto el tiempo de retransmisión, *RTO* calculado, se aprecia que con $f = 4$ es mucho mayor el resultado que con $f = 2$. En el caso de $f = 2$, en el momento que se incrementa el tiempo de ida y vuelta llega a tomar valores muy próximos, por lo que se pueden realizar retransmisiones innecesarias.

15.5.

Dibujar un diagrama similar al de la siguiente figura, para las siguientes condiciones, suponiendo un servicio de red seguro y con y sin secuenciamiento:

- Cierre de la conexión: Activo y Pasivo.
- Cierre de la conexión: Activo y Activo.

- c) Rechazo de conexión.
- d) Proceso de cancelar una conexión: un usuario emite un OPEN a un usuario que está escuchando y entonces él mismo emite un CLOSE antes de que se intercambie algún dato.

Ejemplo de establecimiento de conexión Activo y Pasivo:

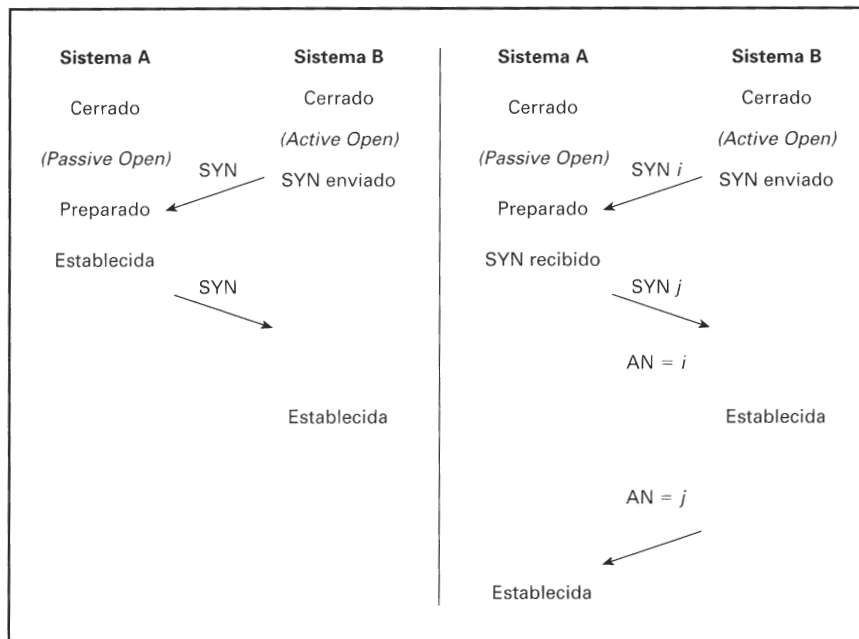


Figura 15.25. Establecimiento de conexión, activo y pasivo.

SOLUCIÓN

- a) Cierre de la conexión: Activo y Pasivo.

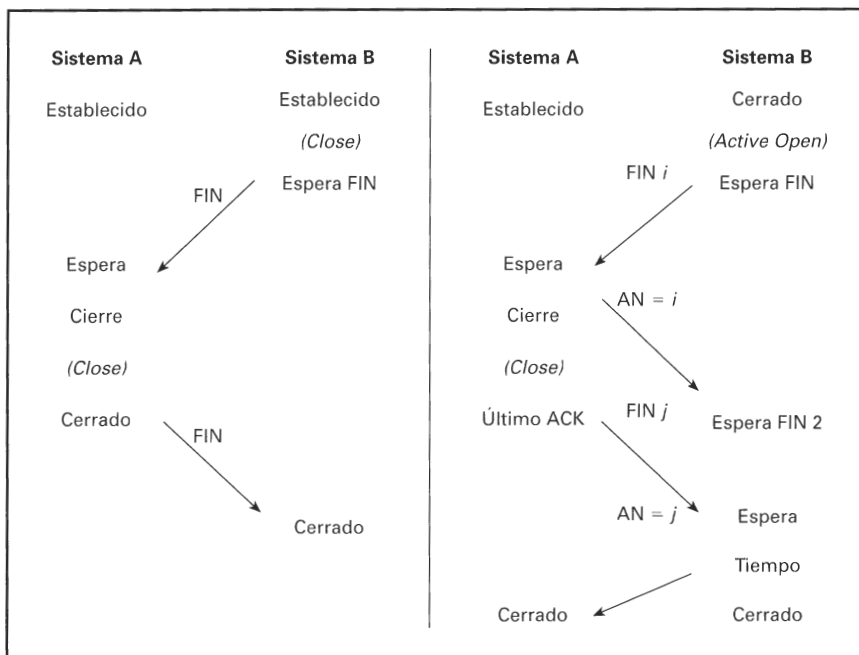


Figura 15.26. Cierre de conexión, activo y pasivo.

b) Cierre de la conexión: Activo y Activo.

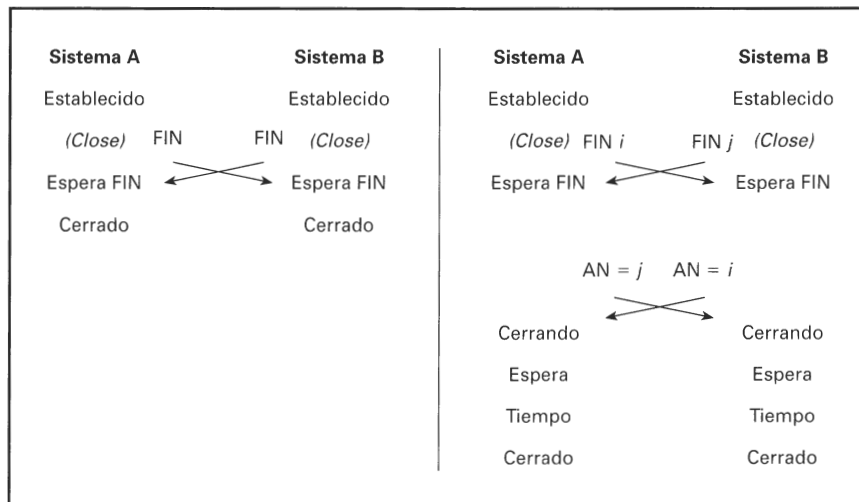


Figura 15.27. Cierre de conexión, activo y activo.

c) Rechazo de conexión.

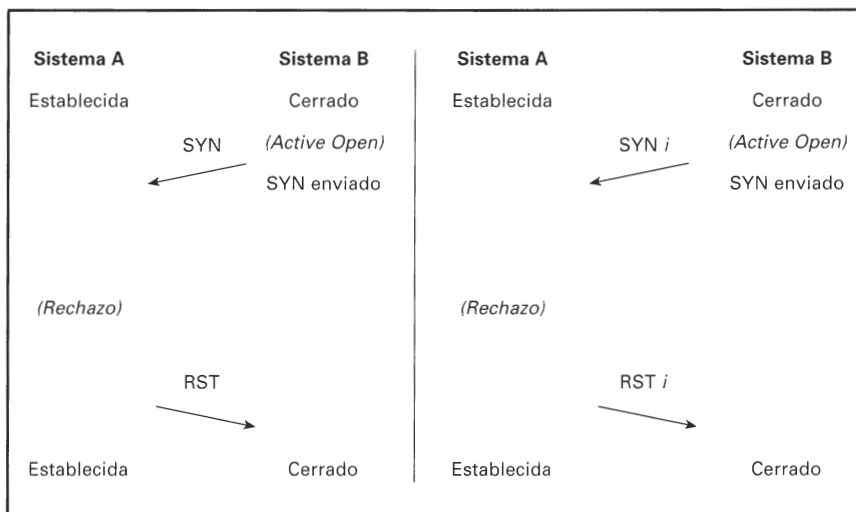


Figura 15.28. Rechazo de conexión.

d) Proceso de cancelar una conexión: un usuario emite un OPEN a un usuario que está escuchando y entonces él mismo emite un CLOSE antes de que se intercambie algún dato.

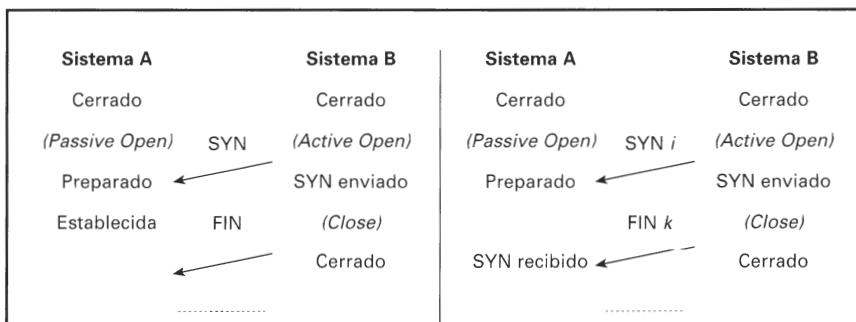


Figura 15.29. Cancelación de conexión.

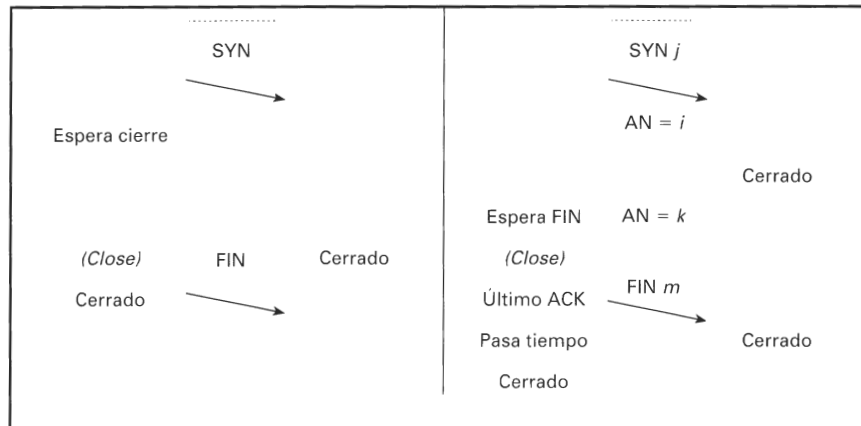


Figura 15.29. Cancelación de Conexión (continuación).

15.6.

En una red que tiene un tamaño máximo de paquete de red de 128 bytes, un tiempo de vida máximo de 20 segundos y un número de secuencia de paquetes de 8 bits, ¿cuál es la máxima tasa de transmisión de datos por conexión?

SOLUCIÓN

En una red con un número de secuencia de paquetes de 8 bits, indica que se pueden enumerar los octetos del 0 al 255 o del 1 al 256, por lo que si el tamaño máximo de paquete es de 128 bytes, indica que con 2 paquetes de tamaño máximo se debe volver a hacer la cuenta desde el inicio. En el peor de los casos, esto resulta cuando la ventana en recepción es lo suficientemente grande, para 2 paquetes máximos, es decir, de 256 bytes sin necesidad de asentimiento.

Puesto que los números de secuencia son de 8 bits, sólo se pueden identificar 256 bytes antes de volver a iniciar el recuento, por lo que no tendría sentido permitir una ventana superior a 256 bytes.

El diagrama de peor caso sería el siguiente (no se dibujan los asentimientos por clarificar el diagrama):

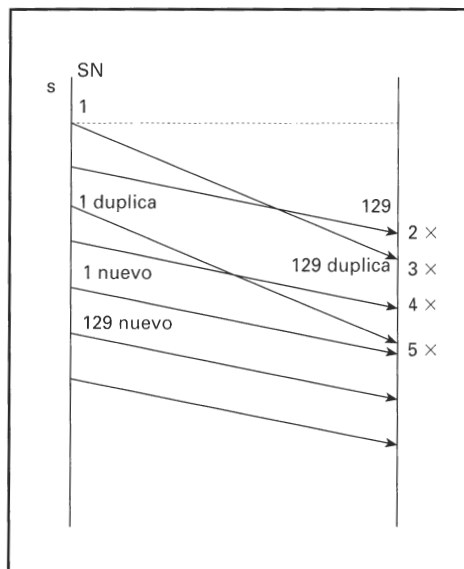


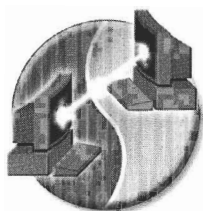
Figura 15.30. Flujo de datos.

Se transmiten paquetes de tamaño máximo, $SN = 1$ y $SN = 129$, pero el primer paquete ($SN = 1$) se retrasa. Llega el segundo paquete ($SN = 129$), pero como falta el primero, no se envía asentimiento por parte del receptor, por lo que el transmisor vuelve a transmitir otra vez los primeros 2 paquetes.

En este caso, al recibir un paquete con número de secuencia (SN) igual a 1, envía asentimiento de que los 2 primeros han sido recibidos satisfactoriamente $AN = 1$ (del receptor al emisor), pero ya está en marcha el segundo paquete $SN = 129$. En realidad, ha llegado el primer paquete de $SN = 1$ que se había retrasado, y ahora se ha retrasado el paquete de $SN = 1$ que se ha retransmitido. El paquete de $SN = 129$ llega duplicado al receptor que también lo notifica ($AN = 1$). Ahora el transmisor ya transmite los siguientes paquetes con el número de secuencia empezando de nuevo, de modo que envía un $SN = 1$ con el tercer paquete de datos. Antes de llegar este paquete ($SN = 1$, pero es el tercer paquete de datos), puede llegar todavía el $SN = 1$ del primer paquete que se ha retransmitido y se ha retrasado en la red. El receptor no lo sabe y no detecta que es duplicado, sino que lo toma como bueno, como el tercer paquete de datos. Si el tiempo de vida máximo de este primer paquete es de 20 segundos, debería de llegar este tercer paquete de $SN = 1$ con un tiempo superior a 20 segundos desde que se retransmitió el anterior paquete de $SN = 1$.

El tiempo máximo de transmisión de datos en esta conexión debería ser aquél que por lo menos un número de paquetes lleguen tras un tiempo superior a 20 segundos. En el diagrama se indica el tiempo de transmisión X por cada paquete. El paquete $SN = 1$ de la retransmisión tardaría X segundos, el paquete de $SN = 129$ llegaría en el tiempo $2X$. El tercer paquete de datos con $SN = 1$ llegaría en $3X$ segundos desde que se retransmitió el paquete de $SN = 1$.

Por lo tanto, $3X$ segundos debe ser superior a 20 segundos para poder detectar duplicados. Así se despeja que $X = 6,666$ segundos por paquete de 128 bytes, que resulta una velocidad de 153,6 bps como velocidad máxima.



SEGURIDAD EN REDES

16.1.

Para la seguridad de redes está el cifrado convencional DES, que consta de clave única, también llamado cifrado simétrico. Esta clave es secreta y sólo debe conocerla el emisor y el receptor. En la parte del cifrado se utilizan bloques de 64 bits del mensaje que se dividen en dos partes de 32 bits. En el proceso de cifrado se expanden a 48 bits y posteriormente se vuelve a reducir (mediante la tabla S) a 32 bits. Para la transformación de 48 bits a 32 bits se utiliza la tabla S que a continuación se expone, y se realiza para bloques de 6 bits que se convierten en bloques de 4 bits, para 8 pasos. Los pasos son $S_j(B_j)$, siendo B_j el paso de 6 bits, y $S_j(B_j)$ de 4 bits. A continuación se muestra un bloque de 48 bits a pasar a un bloque de 32 bits mediante la tabla S.

$B_{1-8} = 011101/110101/011010/001110/001100/111101/010101/000110/$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	S_1															
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	S_2															
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
	S_3															
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	S_4															
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Tabla 16.1. Tabla S.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	S_5															
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	S_6															
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	S_7															
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	S_8															
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Tabla 16.1. Tabla S (continuación).

SOLUCIÓN

Para la transformación se tienen $B_j = b_1b_2b_3b_4b_5b_6$. Si se divide el bloque de 32 bits en pequeños bloques de 6 bits, se tienen los siguientes:

$$B_1 = 011101 = b_1b_2b_3b_4b_5b_6 \rightarrow b_1b_6, b_2b_3b_4b_5 = 01,1110 = 1,14 \rightarrow S_1(B_1) = 0011 \text{ es decir } 3.$$

$$B_2 = 110101$$

$$B_3 = 011010$$

$$B_4 = 001110$$

$$B_5 = 001100$$

$$B_6 = 111101$$

$$B_7 = 010101$$

$$B_8 = 000110$$

Cada bloque B_j se calcula con sus filas de $S_j(B_j)$. El procedimiento es utilizar el valor decimal resultante de b_1b_6 para determinar la fila a elegir de la tabla, y el valor decimal de $b_2b_3b_4b_5$ para determinar la columna. El resultado se pasa a binario y es el bloque de 4 bits esperado.

De esta forma se quedan unos valores de:

$$B_1 = 011101 \rightarrow 1,14 \rightarrow S_1(B_1) = 0011 \text{ (3)}$$

$$B_2 = 110101 \rightarrow 3,10 \rightarrow S_2(B_2) = 0111 \text{ (7)}$$

$$B_3 = 011010 \rightarrow 0,13 \rightarrow S_3(B_3) = 0100 (4)$$

$$B_4 = 001110 \rightarrow 0,7 \rightarrow S_4(B_4) = 1010 (10)$$

$$B_5 = 001100 \rightarrow 0,6 \rightarrow S_5(B_5) = 1011 (11)$$

$$B_6 = 111101 \rightarrow 3,14 \rightarrow S_6(B_6) = 1000 (8)$$

$$B_7 = 010101 \rightarrow 1,10 \rightarrow S_7(B_7) = 0101 (5)$$

$$B_8 = 000110 \rightarrow 0,3 \rightarrow S_8(B_8) = 0100 (4)$$

Resultando el siguiente bloque de 32 bits:

$$S(B) = 0011/0111/0100/1010/1011/1000/0101/0100$$

16.2.

Con la Tabla 16.1: Tabla S calcular el bloque de 32 bits resultantes del siguiente bloque de 48 bits:

$$B_{1-8} = 111011/101010/110100/011100/011001/111010/101010/001101/$$

SOLUCIÓN

Si se divide el bloque de 32 bits en pequeños bloques de 6 bits, se tienen los siguientes y para la transformación se tienen $B_j = b_1b_2b_3b_4b_5b_6$:

$$B_1 = 111011 = b_1b_2b_3b_4b_5b_6 \rightarrow b_1b_6, b_2b_3b_4b_5 = 11,1101 = 3,13 \rightarrow S_1(B_1) = 0000, \text{ es decir } 0$$

$$B_2 = 101010$$

$$B_3 = 110100$$

$$B_4 = 011100$$

$$B_5 = 011001$$

$$B_6 = 111010$$

$$B_7 = 101010$$

$$B_8 = 001101$$

Cada bloque B_j se calcula con sus filas de $S_j(B_j)$. El procedimiento es utilizar el valor decimal resultante de b_1b_6 para determinar la fila a elegir de la tabla, y el valor decimal de $b_2b_3b_4b_5$ para determinar la columna. El resultado se pasa a binario y es el bloque de 4 bits esperado.

De esta forma, se quedan unos valores de:

$$B_1 = 111011 \rightarrow 3,13 \rightarrow S_1(B_1) = 0000 (0)$$

$$B_2 = 101010 \rightarrow 2,5 \rightarrow S_2(B_2) = 0100 (4)$$

$$B_3 = 110100 \rightarrow 2,10 \rightarrow S_3(B_3) = 0010 (2)$$

$$B_4 = 011100 \rightarrow 0,14 \rightarrow S_4(B_4) = 0100 (4)$$

$$B_5 = 011001 \rightarrow 1,12 \rightarrow S_5(B_5) = 0011 (3)$$

$$B_6 = 111010 \rightarrow 2,13 \rightarrow S_6(B_6) = 1101 (13)$$

$$B_7 = 101010 \rightarrow 2,5 \rightarrow S_7(B_7) = 0011(3)$$

$$B_8 = 001101 \rightarrow 1,6 \rightarrow S_8(B_8) = 0111(7)$$

Resultando el siguiente bloque de 32 bits:

$$S(B) = 0000/0100/0010/0100/0011/1101/0011/0111$$

16.3.

Con la Tabla 16-1: Tabla S calcular el bloque de 32 bits resultantes del siguiente bloque de 48 bits.

$$B_{1-8} = 110111/010101/101000/111000/110011/110101/010100/011010/$$

SOLUCIÓN

Si se divide el bloque de 32 bits en pequeños bloques de 6 bits, se tienen los siguientes y para la transformación se tienen $B_j = b_1b_2b_3b_4b_5b_6$:

$$B_1 = 110111 = b_1b_2b_3b_4b_5b_6 \rightarrow b_1b_6, b_2b_3b_4b_5 = 11,1011 = 3,11 \rightarrow S_1(B_1) = 1110, \text{ es decir, } 14$$

$$B_2 = 010101$$

$$B_3 = 101000$$

$$B_4 = 111000$$

$$B_5 = 110011$$

$$B_6 = 110101$$

$$B_7 = 010100$$

$$B_8 = 011010$$

Cada bloque B_j se calcula con sus filas de $S_j(B_j)$. El procedimiento es utilizar el valor decimal resultante de b_1b_6 para determinar la fila a elegir de la tabla, y el valor decimal de $b_2b_3b_4b_5$ para determinar la columna. El resultado se pasa a binario y es el bloque de 4 bits esperado.

De esta forma, se quedan unos valores de:

$$B_1 = 110111 \rightarrow 3,11 \rightarrow S_1(B_1) = 1110 (14)$$

$$B_2 = 010101 \rightarrow 1,10 \rightarrow S_2(B_2) = 0001 (1)$$

$$B_3 = 101000 \rightarrow 2,4 \rightarrow S_3(B_3) = 1000 (8)$$

$$B_4 = 111000 \rightarrow 2,12 \rightarrow S_4(B_4) = 0101 (5)$$

$$B_5 = 110011 \rightarrow 3,9 \rightarrow S_5(B_5) = 1111 (15)$$

$$B_6 = 110101 \rightarrow 3,10 \rightarrow S_6(B_6) = 0001 (1)$$

$$B_7 = 010100 \rightarrow 0,10 \rightarrow S_7(B_7) = 1001 (9)$$

$$B_8 = 011010 \rightarrow 0,13 \rightarrow S_8(B_8) = 0000 (0)$$

Resultando el siguiente bloque de 32 bits:

$$S(B) = 1110/0001/1000/0101/1111/0001/1001/0000$$

16.4.

En la seguridad de redes también está el cifrado asimétrico, conocido como el de clave pública, RSA, que consta de 2 claves, una pública y otra privada. Para el cálculo de estas claves, se utilizan 2 números primos p y q . Posteriormente se calcula el valor n y $\phi(n)$. Se elige una primera clave, la clave pública, el valor de d y se calcula el valor de la clave privada, e . Dados $p = 5$ y $q = 11$, con $d = 23$, hallar el valor de la clave privada e . Comprobar el resultado con un mensaje $M = 17$.

SOLUCIÓN

Con los valores de p y q primos, se tiene el valor $n = p \cdot q = 55$.

Posteriormente, se calcula el valor de $\phi(n) = (p - 1) \cdot (q - 1) = 40$.

Las claves públicas y privadas, sean $\{d, n\}$ y $\{e, n\}$ respectivamente, cumplen una serie de propiedades, entre ellas las siguientes:

- m.c.d. $(d \cdot e) = 1$ (máximo común divisor es 1)
- $e \cdot d = 1 \bmod \phi(n)$
- Dado un mensaje M , se tiene el cifrado $C = M^d \bmod n$.
- Con el cifrado $C = M^d \bmod n$ se tiene el mensaje con $M = C^e \bmod n$
- Se cumple que $M = M^{e \cdot d} \bmod n$

Si ya se tiene $d = 23$, con $n = 55$ y $\phi(n) = 40$, falta calcular e que cumple las dos primeras condiciones, y principalmente $e \cdot 23 = 1 \bmod 40$. De aquí se obtiene que $e = 7$ (el inverso multiplicativo de 23 módulo 40).

Para comprobar que las 2 claves son correspondidas, se cifra un mensaje con la clave pública y luego se descifra con la clave privada y se observa cómo se consigue descifrar el mensaje original.

Dado un mensaje $M = 17$, se puede calcular el mensaje cifrado con $C = 17^{23} \bmod 55$. Para el cálculo de 17^{23} se puede dividir en productos menores siguiendo los números binarios $23 = 10111$. Por lo que $17^{23} = 17^{16} \cdot 17^0 \cdot 17^4 \cdot 17^2 \cdot 17^1$, y según la propiedad del módulo se puede resolver de la siguiente manera:

$$M^{i+j} \bmod n = M^i \cdot M^j \bmod n = M^i \cdot (M^j \bmod n) \bmod n$$

Así se obtiene, $C = M^d \bmod n = 17^{23} \bmod 55 = 18$.

Igualmente se comprueba que $M = C^e \bmod n = 18^7 \bmod 55 = 17$, como se quería demostrar.

16.5.

En el cifrado de RSA, con 2 números primos se calculan las claves pública y privada d y e . Dados $q = 3$, $p = 11$ y $d = 7$, calcular el valor de e .

SOLUCIÓN

Con los valores de p y q primos, se tiene el valor $n = p \cdot q = 33$.

Posteriormente, se calcula el valor de $\phi(n) = (p - 1) \cdot (q - 1) = 20$.

La clave pública y privada, sean $\{d, n\}$ y $\{e, n\}$ respectivamente, cumplen una serie de propiedades, entre ellas las siguientes:

- m.c.d. $(d \cdot e) = 1$ (máximo común divisor es uno)
- $e \cdot d = 1 \bmod \phi(n)$
- Dado un mensaje M , se tiene el cifrado $C = M^d \bmod n$.
- Con el cifrado $C = M^d \bmod n$ se tiene el mensaje con $M = C^e \bmod n$.
- Se cumple que $M = M^{e \cdot d} \bmod n$.

Si ya se tiene $d = 7$, con $n = 33$ y $\phi(n) = 20$, falta calcular e que cumple las dos primeras condiciones, y principalmente $e \cdot 7 = 1 \bmod 20$. De aquí se obtiene que $e = 3$ (el inverso multiplicativo de 7 módulo 20).

16.6.

El cifrado asimétrico de clave pública, *RSA*, necesita de 2 claves, una para cifrar y otra para descifrar. Una de éstas es pública y la otra es privada. Las claves son pares de números enteros. Dada la clave pública $\{17, 77\}$ y la clave privada $\{53, 77\}$, calcular los mensajes cifrados de los siguientes mensajes, y comprobar que al descifrar se consigue el mismo mensaje:

$$M = 20, 21, 17, 18, 15, 3, 10, 8$$

SOLUCIÓN

El cifrado de un mensaje con clave pública, se realiza con la siguiente fórmula:

$$C = M^d \bmod n, \text{ que en este caso } d = 17 \text{ y } n = 77$$

Para $M = 20$, se tiene un cifrado con $d = 17$ y $n = 77$ de $C = 20^{17} \bmod 77 = 48$.

El cálculo se tiene que realizar separando las partes que se multiplican dentro del módulo para que no resulte un número intratable.

17 en número binario es 10001, es decir, $16 + 1$, que si se intenta realizar en múltiplos más pequeños podría quedar así $4 + 4 + 4 + 4 + 1$.

Existe una propiedad en el cálculo del módulo para poder resolver más fácilmente:

$$M^{i+j} \bmod n = M^i \cdot M^j \bmod n = M^i \cdot (M^j \bmod n) \bmod n$$

De esta forma se quedaría para $M = 20$ y $d = 17$ con $n = 77$ de la siguiente forma:

$$\begin{aligned} C &= (((((20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20 \bmod 77 = \\ &= (((71 \cdot 20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20 \bmod 77 = \\ &= ((36 \cdot 20^4 \bmod 77) \cdot 20^4 \bmod 77) \cdot 20 \bmod 77 = \\ &= (15 \cdot 20^4 \bmod 77) \cdot 20 \bmod 77 = 64 \cdot 20 \bmod 77 = 48 \end{aligned}$$

Igualmente, para $M = 21$ se tiene un mensaje cifrado de $C = 21^{17} \bmod 77 = 21$.

Para $M = 17$, se tiene un mensaje cifrado de $C = 17^{17} \bmod 77 = 19$.

Para $M = 18$, se tiene un mensaje cifrado de $C = 18^{17} \bmod 77 = 72$.

Para $M = 15$, se tiene un mensaje cifrado de $C = 15^{17} \bmod 77 = 71$.

Para $M = 3$, se tiene un mensaje cifrado de $C = 3^{17} \bmod 77 = 75$.

Para $M = 10$, se tiene un mensaje cifrado de $C = 10^{17} \bmod 77 = 54$.

Para $M = 8$, se tiene un mensaje cifrado de $C = 8^{17} \bmod 77 = 57$.

Una vez hecho el cifrado, el extremo receptor deberá hacer el descifrado, con la otra clave, de la clave privada $\{53, 77\}$.

La clave 53 en binario es 110101 por lo que sería $1 + 4 + 16 + 32$, que igual que en el caso anterior se podría dividir en $1 + 4 + 8 + 8 + 8 + 8 + 8 + 8$.

Con la propiedad del módulo, se quedaría para $C = 48$ y $e = 53$ con $n = 77$ de la siguiente forma:

$$\begin{aligned}
 M &= (((((((48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (((((((64 \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (((((((15 \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (((((((36 \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (((((((71 \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (((((((1 \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^8 \bmod 77) \cdot 48^4 \bmod 77) \cdot 48 \bmod 77) \cdot 48 \bmod 77 = \\
 &= (64 \cdot 48^4 \bmod 77) \cdot 48 \bmod 77 = 71 \cdot 48 \bmod 77 = 20
 \end{aligned}$$

De esta forma se quedaría el mensaje descifrado, M , para un mensaje cifrado $C = 48$ y una clave pública de $e = 53$ con $n = 77$, $M = 48^{53} \bmod 77 = 20$.

Igualmente, para $C = 21$ se tiene un mensaje descifrado de $M = 21^{53} \bmod 77 = 21$.

Para $C = 19$, se tiene un mensaje descifrado de $M = 19^{53} \bmod 77 = 17$.

Para $C = 72$, se tiene un mensaje descifrado de $M = 72^{53} \bmod 77 = 18$.

Para $C = 71$, se tiene un mensaje descifrado de $M = 71^{53} \bmod 77 = 15$.

Para $C = 75$, se tiene un mensaje descifrado de $M = 75^{53} \bmod 77 = 3$.

Para $C = 54$, se tiene un mensaje descifrado de $M = 54^{53} \bmod 77 = 10$.

Para $C = 57$, se tiene un mensaje descifrado de $M = 57^{53} \bmod 77 = 8$.

16.7.

Para asegurar la red de datos se quiere usar el cifrado *RSA*. El mensaje se envía de A a B . Las claves pública y privada son K_{apub} , K_{apriv} y K_{bpub} , K_{bpriv} respectivamente para A y B . Dado un mensaje M , se quiere asegurar que el mensaje no se lee por ningún otro que el receptor que debe recibirlo, ¿cómo se debe cifrar el mensaje?

SOLUCIÓN

El transmisor del mensaje A deberá cifrar el mensaje M con la clave pública de B , que conoce A por ser una clave pública. Así solamente B será capaz de descifrar y poder recibir el mensaje original M . De modo que el mensaje que se envía es:

$$C = E_{K_{bpub}}[M]$$

Así, B sólo tendrá que descifrar el mensaje cifrado, por ser el único que conoce la clave privada de B para descifrar el mensaje. Éste resulta de la siguiente forma:

$$M = E_{K_{bpriv}}[C] = E_{K_{bpriv}}[E_{K_{bpub}}[M]] = M$$

16.8.

En la seguridad de redes se quiere asegurar un número de conceptos, unos más o menos importantes en función al contenido del mensaje. Por un lado, que el mensaje no sea leído por personas ajenas, por otro la asegurar la identidad del emisor de modo que no es nadie desconocido quien envía el mensaje. También se desea asegurar la integridad del mensaje para que se tenga la seguridad de que nadie ha modificado el mensaje. Con este último punto también se desearía que los mensajes no se reutilicen transcurrido un tiempo, de modo que el mensaje que se recibe corresponde al instante de tiempo correcto. En estos caso, ¿cómo se cifraría un mensaje M , siendo las claves pública y privada K_{apub} , K_{apriv} y K_{bpub} , K_{bpriv} respectivamente para A y B , transmitiendo de A a B ?

SOLUCIÓN

- Para el caso de que sólo el receptor elegido, en este caso B , pueda recibir y sólo él pueda leer adecuadamente el mensaje, se debe cifrar el mensaje M con la clave pública de B , el receptor.

De modo que el mensaje cifrado será $C = E_{K_{bpub}}[M]$ y el descifrado sólo lo puede hacer B que conoce la clave privada de B , de modo que descifre con $M = E_{K_{bpriv}}[C] = E_{K_{bpriv}}[E_{K_{bpub}}[M]] = M$.

- Para asegurar la identidad del emisor se debe utilizar la clave privada del emisor para cifrar el mensaje. Se cifra el mensaje entero o una parte de éste, en función al interés de prevenir escuchas no deseadas, o si no se considera importante la integridad del mensaje.

Considerando que se cifra todo el mensaje, se quedaría un mensaje cifrado de la siguiente forma:

$C = E_{K_{apriv}}[M]$; de esta forma, para descifrar el mensaje sólo hace falta la clave pública del emisor, A en este caso, que por ser pública puede estar en conocimiento de cualquiera. Lo que sí que se garantiza es que quien lo ha enviado es el emisor A , puesto que es el único que conoce la clave privada de A , el emisor.

Para descifrar se calcularía con $M = E_{K_{apub}}[C] = E_{K_{apub}}[E_{K_{apriv}}[M]] = M$.

- Para garantizar la integridad del mensaje, es decir, que al recibir el mensaje se esté seguro de que nadie ha modificado el contenido del mensaje, M , se deberá cifrar con la clave pública del receptor y con una parte del mensaje (una función del contenido), S , con la clave privada del emisor.

Se deberá cifrar el mensaje, M , con la clave pública de B , el receptor, para asegurar que nadie más lo puede leer. Por otro lado, se deberá generar una parte a añadir, S , al mensaje original, que será una función del contenido del mensaje que se cifrará con la clave privada del emisor, A . De esta forma se garantiza que es A quien lo ha enviado puesto que se deberá de descifrar con su clave pública, y por otro lado si alguien realiza algún cambio en el contenido del mensaje, M , deberá de realizar la función del nuevo contenido, M' , generando el nuevo S' y cifrarlo con la clave privada de A , que sólo el emisor, A , conoce. De esta forma se detectará si se ha modificado el mensaje o no, si la parte del mensaje S corresponde con el contenido M o no.

El resultado de la función $S(M)$ puede ser cifrada con la clave pública del receptor, B , o no. Sólo afectaría que únicamente B pudiera acceder a esta parte para descifrarla.

El mensaje cifrado resultará $C = E_{Kbpub}[M + E_{Kpriv}[S(M)]]$.

El receptor será el único que conocerá la clave privada de B , para poder descifrar el mensaje. De modo que se tendrá:

$$E_{Kbpriv}[E_{Kbpub}[M + E_{Kpriv}[S(M)]]] = M + E_{Kpriv}[S(M)]$$

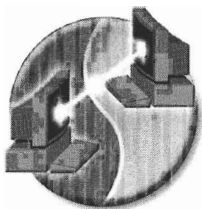
Para comprobar que el mensaje no se ha modificado se deberá usar la clave pública de A , el emisor. Una vez descifrado $S(M)$, se deberá realizar el cálculo para obtener S del mensaje recibido y ver si coincide con el $S(M)$ descifrado.

$$E_{Kpub}[E_{Kpriv}[S(M)]] = S(M)$$

- En el caso de querer asegurar que el mensaje no llega fuera de su tiempo de vida, se debe cifrar el instante de tiempo en el que se transmite el mensaje con la clave privada del emisor, A . De esta forma si se deseara retardar intencionadamente la recepción de un mensaje, se deberá cambiar dicho instante de tiempo y cifrarlo con la clave privada de A , el emisor, que sólo el emisor, A , conoce.

Al cifrar el instante de tiempo en el que se transmite con la clave privada de A , se consigue que cualquiera pueda descifrar dicho instante de tiempo con la clave pública de A , ya que la clave es pública. Pero no se podría modificar dicho valor, porque sería necesario volver a cifrarlo con la clave privada de A , que sólo A conoce, por ser privada la clave. Si no se cifra con la clave privada de A , no se garantiza que sea A , realmente el emisor, quedando detectado que el paquete ha sido modificado, o retardado intencionadamente.

Si el instante de tiempo en el que se transmite es T , el mensaje a enviar sería $M + E_{Kpriv}[T]$. El receptor podría comprobar mediante la clave pública de A el instante de tiempo en el que fue transmitido para tomar las correspondientes medidas. El instante T se descifraría con $E_{Kpub}[E_{Kpriv}[T]] = T$.



APLICACIONES DISTRIBUIDAS

17.1

El ASN.1 es Notación Sintáctica Abstracta número Uno y se creó por el desarrollo de una normalización entre las torres de protocolos, OSI y TCP / IP. Es una representación común para el Intercambio de datos entre entidades diferentes. Con ASN.1 se pueden definir módulos y demás cosas. Considere la siguiente definición:

```
ExpeditedDataAcknowledgement ::= SET {
  desRef      [0]      Reference,
  yr-tu-nr    [1]      TPDUnumber,
  checkSum    [2]      CheckSum OPTIONAL }
NormalEA ::= ExpeditedDataAcknowledgement (WITH COMPONENTS {
  desRef,
  yr-tu-nr (0...127) }
```

Encontrar una definición en ASN.1 equivalente para definir NormalEA.

SOLUCIÓN

Para hacer una definición de *NormalEA*, primeramente hay que entender la estructura que se ha definido.

Se ha definido un conjunto de nombre *ExpeditedDataAcknowledgement*, que contiene tres elementos de tipos diferentes. Los elementos del conjunto son *desRef* del tipo *Reference*, el elemento *yr-tu-nr* de tipo *TPDUnumber* y finalmente el elemento opcional, no siempre obligatorio, *checkSum* del tipo *CheckSum*.

Una vez definido el conjunto *ExpeditedDataAcknowledgement*, se ha definido *NormalEA* como un conjunto que utiliza elementos del anterior conjunto *ExpeditedDataAcknowledgement* para definir que utiliza los elementos *desRef* y *yr-tu-nr*. En el caso del último elemento, le impone una condición en los posibles valores que puede tomar, indica que sólo puede tener valores del 0 al 127 ambos inclusive.

Una definición equivalente para definir el tipo *NormalEA* podría ser la siguiente definición reducida, utilizando únicamente los tipos que interesan:

```

NormalEA ::= SET {
  desRef      [0]    Reference,
  yr-tu-nr    [1]    TPDUnumber (0...127) }

```

De esta forma se define *NormalEA* como un conjunto de dos elementos, *desRef*, de tipo *Reference*, y *yr-tu-nr* de tipo *TPDUnumber* con la limitación de ser un valor entre el 0 y el 127 ambos inclusive.

17.2.

Con las mismas definiciones que en el problema anterior, se añade una nueva definición más:

```

ExtenderEA ::= ExpeditedDataAcknowledgement (WITH COMPONENTS {
  ..., checkSum PRESENT })

```

Se pide una expresión equivalente para definir *ExtenderEA*.

SOLUCIÓN

Por la definición del problema anterior, se parte del conjunto *ExpeditedDataAcknowledgement* que consta de tres elementos:

- *desRef* del tipo *Reference*,
- *yr-tu-nr* del tipo *TPDUnumber*,
- *checkSum* del tipo *Checksum* siendo un elemento opcional y no obligatorio en *ExpeditedDataAcknowledgement*.

Según la definición realizada al elemento *ExtenderEA*, se tiene un conjunto con los mismos elementos que en *ExpeditedDataAcknowledgement* con una característica. Esta característica es que para *ExtenderEA* el elemento *checkSum* es un elemento obligatorio

De modo que se tienen dos conjuntos iguales en el que en uno, *ExpeditedDataAcknowledgement*, el elemento *checkSum* es opcional y en el otro, *ExtenderEA*, el elemento *checkSum* es obligatorio.

Una posible expresión equivalente para definir *ExtenderEA* podría ser la siguiente, de un modo abreviado:

```

ExtenderEA ::= SET {
  desRef      [0]    Reference,
  yr-tu-nr    [1]    TPDUnumber,
  checkSum    [2]    CheckSum }

```

17.3.

Juntando las definiciones de los problemas anteriores, 17.1 y 17.2, se pide definir un nuevo tipo, de nombre *EA*, que esté basado en el conjunto *ExpeditedDataAcknowledgement*, con la restricción de que la suma de comprobación está ausente, y en cuyo caso *yr-tu-nr* debe estar en el rango de 0 a 127 ambos inclusive, o la suma de comprobación está presente, en cuyo caso no existe ninguna restricción adicional en *yr-tu-nr*.

SOLUCIÓN

El conjunto *ExpeditedDataAcknowledgement* consta de tres elementos:

- *desRef* del tipo *Reference*,
- *yr-tu-nr* del tipo *TPDUnumber* en el rango de 0 a 127,
- *checkSum* del tipo *Checksum* siendo un elemento opcional y no obligatorio en *ExpeditedDataAcknowledgement*.

El nuevo tipo debe tener una de las dos posibles estructuras:

- **Estructura número uno:**
 - *desRef* del tipo *Reference*,
 - *yr-tu-nr* del tipo *TPDUnumber*.
- **Estructura número dos:**
 - *desRef* del tipo *Reference*,
 - *yr-tu-nr* del tipo *TPDUnumber*,
 - *checkSum* del tipo *Checksum*.

De modo que primeramente habrá que definir estas estructuras, y luego mediante el tipo **CHOICE** se podrá definir el nuevo tipo, *EA*.

La definición de tipos quedaría como sigue:

```
TipoUno ::= SET {
desRef      [0]    Reference,
yr-tu-nr    [1]    TPDUnumber (0..127) }
TipoDos ::= SET {
desRef      [0]    Reference,
yr-tu-nr    [1]    TPDUnumber,
checkSum    [2]    CheckSum }
EA ::= CHOICE {
uno         TipoUno
dos         TipoDos}
```

Otra forma más compleja pero igualmente válida de definir el tipo *EA* sería una definición sin necesidad de utilizar tipos auxiliares y utilizando el tipo ya existente *ExpeditedDataAcknowledgement*, quedando como sigue:

```
EA ::= CHOICE {
ExpeditedDataAcknowledgement (WITH COMPONENTS {
desRef,
yr-tu-nr (0..127)},
```

Sin checksum pero con restricción para yr-tu-nr

*ExpeditedDataAcknowledgement (WITH COMPONENTS {
..., checksum PRESENT })}*

Con checksum pero sin restricciones adicionales para yr-tu-nr.

17.4.

En la nomenclatura de ASN.1 hay una serie de convenciones a seguir. En la siguiente sintaxis de ASN.1 hay un problema. Corregir e indicar cuál es el problema que se presenta.

```
Protocolo ::= SET {
    tipo          INTEGER,
    longitud      INTEGER,
    texto         IA5String,
    checksum      INTEGER }
```

SOLUCIÓN

El problema que se presenta es la forma de definir el conjunto “Protocolo”, exactamente el tipo de sus elementos. Tres de los cuatro elementos son de tipo INTEGER, por lo que existe una redundancia de etiquetas.

Para corregir este problema habrá que utilizar la marca de IMPLICIT, para distinguir los tipos dentro de un tipo de estructura, de modo que se reemplaza la marca de un tipo antiguo por el nuevo, con la condición de que sólo se pueden usar marcas nuevas.

El problema corregido quedaría como sigue:

```
Protocolo ::= SET {
    tipo          [1]    IMPLICIT    INTEGER,
    longitud      [2]    IMPLICIT    INTEGER,
    texto         IA5String,
    checksum      [3]    IMPLICIT    INTEGER }
```

17.5.

Dada la siguiente sintaxis de ASN.1 indicar si es correcta o no, proponiendo una solución para resolver el problema, si lo hubiera:

```
PDU ::= SEQUENCE {
    Identificador-solicitud  INTEGER,
    NoError                 BOOLEAN,
    WrongType               BOOLEAN OPTIONAL,
    Indice-error            INTEGER (0 ... 8),
    InconsistentName        BOOLEAN OPTIONAL }
```

SOLUCIÓN

En este tipo PDU, se tiene una lista ordenada de tipos y existe una redundancia de etiquetas en los elementos opcionales de igual tipo. La forma de resolver este problema será utilizar IMPLICIT para reemplazar la marca del tipo por otro nuevo. De esta forma se quedaría una definición como la que sigue:

```
PDU ::= SEQUENCE {
    Identificador-solicitud  INTEGER,
    NoError                  BOOLEAN,
    WrongType                [0]   IMPLICIT BOOLEAN OPTIONAL,
    Indice-error             INTEGER (0 ... 8),
    InconsistentName         [1]   IMPLICIT BOOLEAN OPTIONAL}
```

17.6.

Definir con la sintaxis ASN.1 un tipo, de nombre Menor-de-Cien, que sea un tipo formado por dígitos del tipo IA5String, cuyo valor final sea un número del 0 al 99. Sugerencia: Definir algún tipo auxiliar para una mejor claridad.

SOLUCIÓN

Primero se definirá un tipo con los dígitos del tipo *IA5String*, y luego se creará una secuencia de éstos. Finalmente se generará el tipo de nombre *Menor-de-Cien* con la restricción de usar sólo de 1 a 2 dígitos para tener valores de 0 a 99.

La sintaxis resultante será:

```
Digitos ::= IA5String (FROM ( "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ))
```

```
Numero ::= SEQUENCE OF Digitos
```

```
Menor-de-Cien ::= Numero (WITH COMPONENTS (SIZE 1 ... 2))
```

17.7.

Para la gestión de red se tiene un protocolo simple de gestión, SNMP (Simple Network Management Protocol). Se ha ido mejorando el protocolo, pero se ha seguido manteniendo la idea básica de una gran base de información de gestión, la MIB. Suponiendo que el esquema siguiente es una parte de la MIB, reducida para un mejor tratamiento, se pide el código del PDU a utilizar para conseguir los valores mostrados, sean: *sysDescr*, *sysName*, *ipDefaultTTL* y los valores de la tabla partiendo del valor *IpIIndex*.

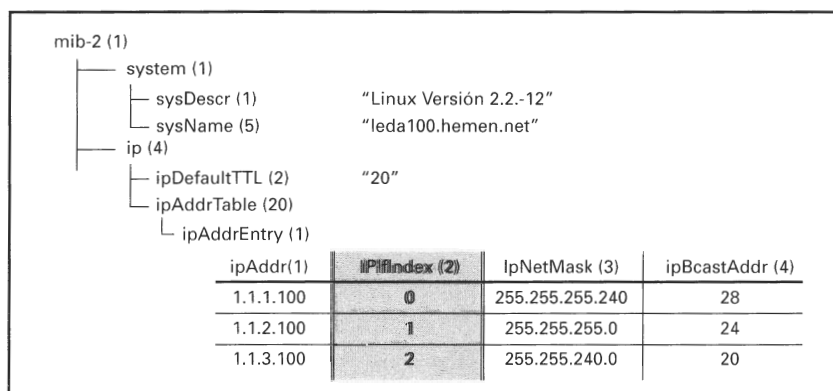


Figura 17.1. Objetos de tabla MIB.

SOLUCIÓN

Para conseguir los valores de la tabla MIB del agente, el gestor deberá enviar una PDU del tipo *GetRequest* con el identificador del objeto del cual quiere el valor. Los identificadores se enumeran en forma de árbol. En el caso de valores escalares, se debe terminar el identificador del objeto con un cero, 0, para indicar el valor del objeto. En el caso de la tabla, hay que conocer el objeto que identifica cada una de las filas de la tabla, de ese modo se debe indicar la columna o el objeto del que se quiere obtener el valor seguido del valor de identificador que marca la fila. En este caso el objeto que marca las filas es el *ipIfIndex*, el índice del Interfaz de IP.

Los identificadores de objeto que se deben pedir son los siguientes, todos ellos precedidos por el tipo de PDU, *Get*:

Get 1.1.1.0	Para conseguir el valor de <i>sysDescr</i> "Linux Versión 2.2.-12"
Get 1.1.5.0	Para conseguir el valor de <i>sysName</i> "leda100.hemen.net"
Get 1.4.2.0	Para conseguir el valor de <i>ipDefaultTTL</i> "20"
Get 1.4.20.1.1.0	Para conseguir el valor de <i>ipAddr</i> del interfaz 0, "1.1.1.100"
Get 1.4.20.1.3.0	Para conseguir el valor de <i>ipNetMask</i> , "255.255.255.240" máscara del interfaz IP número 0
Get 1.4.20.1.4.0	Para conseguir el valor de <i>ipBcastAddr</i> , interfaz 0, "28" bits
Get 1.4.20.1.1.1	Para conseguir el valor de <i>ipAddr</i> del Interfaz 1, "1.1.2.100"
Get 1.4.20.1.3.1	Para conseguir el valor de <i>ipNetMask</i> , interfaz 1, "255.255.255.0"
Get 1.4.20.1.4.1	Para conseguir el valor de <i>ipBcastAddr</i> , interfaz 1, "24" bits
Get 1.4.20.1.1.2	Para conseguir el valor de <i>ipAddr</i> del Interfaz 2, "1.1.3.100"
Get 1.4.20.1.3.2	Para conseguir el valor de <i>ipNetMask</i> , interfaz 2, "255.255.240.0"
Get 1.4.20.1.4.2	Para conseguir el valor de <i>ipBcastAddr</i> , interfaz 2, "20" bits

17.0.

Aplicando la misma tabla que en el ejercicio anterior, ¿qué pasaría si se utilizara la PDU tipo *GetNext*? Empezar con el identificador 1.1.1 y en función de lo que devuelva, indicar qué sucede.

SOLUCIÓN

Al realizar una solicitud con *GetNext*, se recorre el árbol devolviendo el siguiente identificador de objeto de la base de información, con su correspondiente valor.

A continuación se verá para cada identificador de solicitud el identificador que muestra la respuesta con su correspondiente valor.

<u>Solicitud</u>	<u>Respuesta</u>	
<u>ID objeto</u>	<u>ID objeto</u>	<u>Valor</u>
1.1.1	1.1.1.0	"Linux versión 2.2.-12"
1.1.1.0	1.1.1.5.0	"leda100.hemen.net"
1.1.1.5.0	1.4.2.0	20

1.4.2.0	1.4.20.1.1.0	1.1.1.100
1.4.20.1.1.0	1.4.20.1.1.1	1.1.2.100
1.4.20.1.1.1	1.4.20.1.1.2	1.1.3.100
1.4.20.1.1.2	1.4.20.1.2.0	0
1.4.20.1.2.0	1.4.20.1.2.1	1
1.4.20.1.2.1	1.4.20.1.2.2	2
1.4.20.1.2.2	1.4.20.1.3.0	255.255.255.240
1.4.20.1.3.0	1.4.20.1.3.1	255.255.255.0
1.4.20.1.3.1	1.4.20.1.3.2	255.255.240.0
1.4.20.1.3.2	1.4.20.1.4.0	28
1.4.20.1.4.0	1.4.20.1.4.1	24
1.4.20.1.4.1	1.4.20.1.4.2	20

17.9.

Uno de los servicios más utilizados en Internet es el correo electrónico. Este correo tiene un protocolo simple de transferencia de mensajes SMTP. Se pide representar el flujo de información que se crearía con órdenes y respuestas de SMTP para la siguiente situación: el emisor david@elbara.net quiere enviar un mensaje a sus amigos, jon@tux.net, miren@tux.net y arantxa@tux.net. Sin embargo, la dirección de Arantxa ya no es ésta, pero eso David lo desconoce. El mensaje que les quiere transmitir es "Hola,<CR> ¿qué tal si hacemos una cena el próximo viernes?<CR>David".

Solución

El emisor es david@elbara.net y quiere transmitir un mismo mensaje a sus tres amigos Jon, Miren y Arantxa, que están bajo un mismo receptor SMTP, por lo que la comunicación sólo se tiene que hacer una vez con el receptor SMTP.

Primero se crea una conexión TCP entre las dos máquinas. Cuando el receptor le envía un mensaje indicando que está preparado, envía un mensaje "220 Service Ready". Luego empieza el tráfico de SMTP de la siguiente manera (E: Emisor y R: Receptor):

E: HELLO

R: 250 OK

E: MAIL FROM: <david@elbara.net>

R: 250 OK

E: RCPT TO: <jon@tux.net>

R: 250 OK

E: RCPT TO: <miren@tux.net>

R: 250 OK

E: RCPT TO: <arantxa@tux.net>

R: 550 No such user here

E: DATA

R: 354 Start mail input; end with <CRLF>.<CRLF>

E: Hola,

E: ¿qué tal si hacemos una cena el próximo viernes?

E: David.

E: <CRLF>.<CRLF>

R: 250 OK

E: QUIT

R: 250 OK

Y se cerrará la conexión TCP que se ha abierto. Esto no indica que los receptores finales hayan recibido ya el mensaje, sino que el receptor SMTP ha recibido ya el mensaje.

17.10.

Los mensajes enviados con SMTP tienen una cabecera con un formato definido en RFC 822. En el caso del problema anterior, ¿qué forma tiene el mensaje? Si falta algún dato, se deja rellenar opcionalmente.

SOLUCIÓN

El formato RFC 822 tiene unas palabras clave seguidas de (:), para seguir con el argumento de la palabra clave. En este caso, con los datos que se tiene se quedaría de esta forma:

From: <<David>> <david@elbara.net>

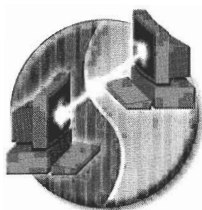
Subject: Cena el viernes

To: jon@tux.net, miren@tux.net, arantxa@tux.net

Hola,

¿qué tal si hacemos una cena el próximo viernes?

David.



ÍNDICE DE FIGURAS Y TABLAS

Figuras

Figura 1.1.	Señal en el dominio del tiempo	1
Figura 1.2.	Señal en el dominio frecuencial	1
Figura 1.3.	Señal hasta segundo fundamental en el dominio frecuencial	2
Figura 1.4.	Señal en el dominio frecuencial	3
Figura 2.1.	Longitud de la antena en función de la frecuencia de la portadora	10
Figura 2.2.	Diagrama de posicionamiento de dos antenas de altura h	10
Figura 2.3.	Diagrama de posicionamiento de dos antenas de altura h_1 y h_2 respectivamente	11
Figura 3.1.	Diagrama de estados de la codificación NRZI	15
Figura 3.2.	Codificador NRZI	16
Figura 3.3.	Probabilidad de error de bit (BER) en función de la relación E_b/N_0	17
Figura 3.4.	Cuantificación lineal: Señal, Error y Espectro del ruido	22
Figura 3.5.	Cuantificación lineal (técnica de dithering): Señal, Error y espectro del ruido	23
Figura 3.6.	Cuantificación lineal	24
Figura 3.7.	Cuantificación no lineal	24
Figura 3.8.	Cuantificación lineal, no lineal y su Distorsión	25
Figura 4.1.	Muestreo de un carácter	29
Figura 5.1.	Eficiencia del enlace en función del tamaño del paquete (BER 5 1024)	36
Figura 5.2.	Cálculo del Resto	37
Figura 5.3.	Cálculo del CRC	38
Figura 5.4.	Error del protocolo ante la pérdida de dos tramas	38
Figura 5.5.	Transmisión de una trama de información	39
Figura 5.6.	Pérdida de una trama de información	39
Figura 5.7.	Pérdida de una trama de información	39
Figura 5.8.	Confirmación de una trama de información	40
Figura 5.9.	Recuperación de la pérdida de una trama de información	40
Figura 5.10.	Recuperación de la pérdida de una trama de información	40
Figura 6.1.	Conmutación temporal	42
Figura 6.2.	Estructura de los conmutadores T-S-T	43
Figura 6.3.	Ocupación de canales en la etapa de entrada (1) y de salida (2) en el peor caso	44
Figura 6.4.	Cálculo de la probabilidad de bloqueo en un conmutador de tres etapas	45
Figura 6.5.	Relación entre n y k en conmutador de tres etapas ($N = 5960$)	47
Figura 6.6.	Complejidad de un conmutador de tres etapas ($N = 960$, P_1 y P_{01})	48
Figura 6.7.	Relación entre n y k para diferentes cargas de los enlaces de entrada ($N = 960$, P_1)	50
Figura 6.8.	Complejidad del conmutador para diferentes cargas de red ($N = 960$, P_1)	50
Figura 6.9.	Ejemplo de segmento de red SONET	51
Figura 6.10.	Capas SONET sobre segmento de red	52
Figura 6.11.	Estructura de trama SONET	52
Figura 6.12.	Estructura de la multiplexación de tramas STS	53
Figura 8.1.	Transmisión de fichero por conmutación de circuitos	76
Figura 8.2.	Transmisión por conmutación de paquetes	77
Figura 8.3.	Arquitectura de un nodo de comunicaciones	78
Figura 8.4.	Tiempos de conmutación interna	79
Figura 8.5.	Diagrama temporal de ocupación de enlaces para la conmutación de paquetes	80
Figura 8.6.	Diagrama temporal de ocupación de enlaces en conmutación de circuitos	81
Figura 8.7.	Diagrama temporal de ocupación de enlaces para conmutación cut-through	81
Figura 8.8.	Estudio del interbloqueo de comunicaciones	82

Figura 8.9.	Ejemplo de configuración de red	84
Figura 8.10.	Árbol de alcanzabilidad resultante	88
Figura 8.11.	Ejemplo de configuración de red	88
Figura 8.12.	Árbol de alcanzabilidad resultante	91
Figura 8.13.	Ejemplo de configuración de red	92
Figura 8.14.	Árbol de alcanzabilidad resultante	93
Figura 8.15.	Ejemplo de configuración de red	94
Figura 8.16.	Resultado paso a paso de la ejecución del Algoritmo de Starobinski	95
Figura 9.1.	Esquema de multiplexor	97
Figura 9.2.	Comparación TDM/Multiplexación estadística	97
Figura 9.3.	Eficiencia de la transmisión con celdas	98
Figura 9.4.	Funcionamiento de un token bucket con buffer infinito	99
Figura 9.5.	Funcionamiento del token bucket con buffer finito (2 celdas)	100
Figura 10.1.	Red con señalización por congestión	102
Figura 10.2.	Tasa instantánea frente al tiempo	104
Figura 10.3.	Gráficas de las expresiones de los apartados a) y b)	107
Figura 11.1.	Red inalámbrica	114
Figura 11.2.	Redes unidas por puentes	115
Figura 12.1.	Esquema de red	121
Figura 12.2.	Esquema de red	122
Figura 12.3.	Diagrama de transmisión	131
Figura 12.4.	Intervalos	135
Figura 13.1.	Red con encaminador	140
Figura 13.2.	Transmisión multidifusión	142
Figura 13.3.	Árbol de expansión	143
Figura 13.4.	Esquema de red	147
Figura 14.1.	Funcionamiento RSVP	151
Figura 14.2.	Sistema autónomo	152
Figura 14.3.	Grafo resultante	153
Figura 14.4.	Árbol de encaminamiento	153
Figura 14.5.	Árbol de encaminamiento	154
Figura 14.6.	Árbol de encaminamiento	155
Figura 14.7.	Red de sistemas autónomos	157
Figura 14.8.	Flujo de paquetes	157
Figura 14.9.	Adquisición de vecino	158
Figura 14.10.	Detección de vecino alcanzable	158
Figura 14.11.	Red alcanzable	158
Figura 14.12.	Error de sintaxis	159
Figura 14.13.	Tiempo de mantenimiento expirado	159
Figura 14.14.	Red de sistemas autónomos	160
Figura 14.15.	Red de sistemas autónomos	161
Figura 14.16.	Sistema autónomo	162
Figura 14.17.	Grafo resultante	163
Figura 14.18.	Árbol de encaminamiento	163
Figura 14.19.	Esquema de transmisión de paquetes	165
Figura 14.20.	Esquema de transmisión de paquetes <i>token bucket</i>	165
Figura 14.21.	Términos de una red de sistemas autónomos	168
Figura 14.22.	Términos de red, solución	168
Figura 14.23.	Tratamiento del tráfico (términos)	169
Figura 14.24.	Tratamiento del tráfico (solución)	169
Figura 15.1.	Control y flujo de A y B	172
Figura 15.2.	Control de flujo de A	172
Figura 15.3.	Flujo de datos	172
Figura 15.4.	Control de flujo de B	172
Figura 15.5.	Flujo de datos	173
Figura 15.6.	Control de flujo de A	173
Figura 15.7.	Flujo de datos	173
Figura 15.8.	Control de flujo de B	173
Figura 15.9.	Control de flujo de A	174
Figura 15.10.	Flujo de datos	174
Figura 15.11.	Control de flujo de B	174
Figura 15.12.	Flujo de datos	174
Figura 15.13.	Control de flujo de A	174
Figura 15.14.	Flujo de datos	175
Figura 15.15.	Control de flujo de A	175
Figura 15.16.	Flujo de datos	175
Figura 15.17.	Control de flujo de A	175
Figura 15.18.	Control de flujo de A	175
Figura 15.19.	Control de flujo de A y B	176
Figura 15.20.	Flujo de datos	176
Figura 15.21.	Flujo de datos	177
Figura 15.22.	Flujo de datos	178

Figura 15.23.	Estimación del tiempo de ida y vuelta	179
Figura 15.24.	Estimación del tiempo de ida y vuelta	180
Figura 15.25.	Establecimiento de conexión, activo y pasivo	181
Figura 15.26.	Cierre de conexión, activo y pasivo	181
Figura 15.27.	Cierre de conexión, activo y activo	182
Figura 15.28.	Rechazo de conexión	182
Figura 15.29.	Cancelación de conexión	182
Figura 15.29.	Cancelación de conexión (continuación)	183
Figura 15.30.	Flujo de datos	183
Figura 17.1.	Objetos de tabla MIB	199

Tablas

Tabla 2.1.	Longitud de la antena en función de la frecuencia de la portadora	9
Tabla 2.2.	Longitud de la antena en función de la frecuencia	9
Tabla 3.1.	Tabla de la verdad de la codificación NRZI	15
Tabla 6.1.	Selección de k para obtener una probabilidad de bloqueo menor que 0,1 y 0,01	46
Tabla 6.1.	(Continuación)	47
Tabla 6.2.	Efecto de la carga del enlace al seleccionar el valor k de un conmutador de tres etapas	48
Tabla 6.2.	(Continuación)	49
Tabla 6.3.	Relación STS, OC y STM	51
Tabla 7.1.	Estudio de tráfico	57
Tabla 7.2.	Valores totales de tráfico y de hora cargada (BH – Busy Hour)	58
Tabla 7.3.	Resumen de valores ABBH para el estudio de tráfico	58
Tabla 7.4.	Valores totales de tráfico y de hora cargada (BH – Busy Hour)	62
Tabla 7.4.	Tabla de Erlang B	63
Tabla 7.6.	Erlang B extendida (50%)	64
Tabla 7.7.	Tabla de H. Jacobsen (70%)	65
Tabla 7.8.	Extracto de la tabla de Erlang C para 2 Erlangs	66
Tabla 7.9.	Extracto de la tabla de Erlang C para 2 Erlangs	67
Tabla 7.10.	Extracto de la tabla de Erlang C para 2 Erlangs	68
Tabla 7.11.	Extracto de la tabla de Erlang C para 3,5 Erlangs	69
Tabla 7.12.	Extracto de la tabla de Erlang C para 2 Erlangs	70
Tabla 7.13.	Extracto de la tabla de Erlang C para cola finita (1 min.) y 2 Erlangs	71
Tabla 7.14.	Extracto de la tabla de Erlang C para cola finita (10 min.) y 2 Erlangs	72
Tabla 8.1.	Estado inicial del sistema	82
Tabla 8.2.	Cada nodo ha transmitido un mensaje a su vecino	83
Tabla 8.3.	Transmisión del mensaje	83
Tabla 8.4.	Estado de interbloqueo	83
Tabla 8.5.	Resumen de resultados al aplicar el Algoritmo de Dijkstra	92
Tabla 8.6.	Resumen de resultados al aplicar el Algoritmo de Bellman-Ford	92
Tabla 11.1.	Tablas de reenvío de los puentes	116
Tabla 13.1.	Valores de campos seleccionados de cada fragmento	142
Tabla 13.2.	Campos de la cabecera IP	145
Tabla 13.3.	Tabla de rutas del encaminador	146
Tabla 14.1.	Tabla resultante	154
Tabla 14.2.	Tabla resultante	155
Tabla 14.3.	Tabla resultante de R8	156
Tabla 14.4.	Tabla resultante	164
Tabla 14.5.	Servicios de Internet	164
Tabla 14.6.	Tabla de definiciones	166
Tabla 14.7.	Tabla de definiciones (solución)	167
Tabla 16.1.	Tabla S	185

Comunicaciones y redes de computadores

Problemas y ejercicios resueltos

La presente obra nace con la intención de ayudar a los estudiantes en el aprendizaje de los conceptos básicos de las redes de comunicaciones. En este sentido, una de las mejores maneras de apoyar al estudiante en el aprendizaje de redes de comunicaciones, es la presentación de ejercicios relacionados con los conceptos teóricos más elementales.

El libro está estructurado de la siguiente manera:

- Múltiples ejercicios acompañados con la solución de los mismos, para que el estudiante pueda trabajar de forma personal su resolución.
- Sigue el esquema del libro de William Stallings *Comunicaciones y redes de computadores*, por lo que éste manual sirve de libro de apoyo para la comprensión de los conceptos introducidos en dicha obra.

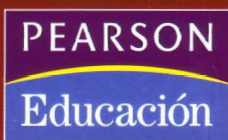


www.prenticepractica.com

PRENTICE PRACTICA es una colección de libros, cuyo texto es eminentemente práctico. La finalidad de esta colección es permitir al alumno comprender y afianzar la asimilación de la teoría a través de diversos ejercicios y ejemplos.

PRENTICE PRACTICA es una colección amena, de carácter muy didáctico y que, de una forma sencilla, consigue que el alumno obtenga un perfecto manejo práctico de la asignatura.

PRENTICE PRACTICA está dirigida al alumno para conseguir su autoaprendizaje en la materia. La colección es una de las más actualizadas del mercado.



www.pearsoneducacion.com

