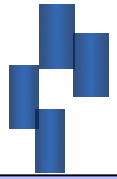
Tema 2
La Capa de Red

Amelia Zafra Gómez
Dpto. Informática y Análisis Numérico



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía

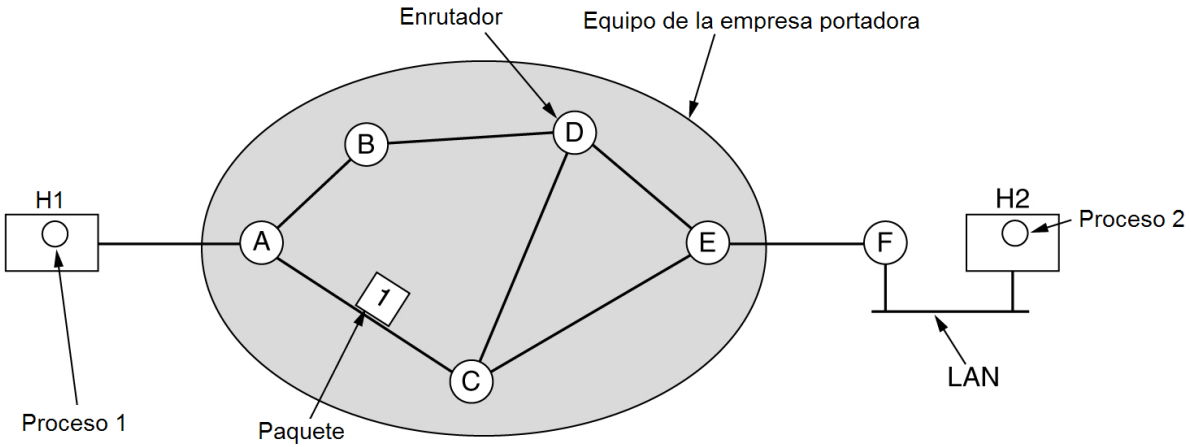
1. Introducción

- La capa de red se encarga de:

1. Encaminamiento: determinación de una ruta a seguir para llevar los paquetes desde el origen hasta el destino.
2. Control de congestión: elegir las rutas adecuadas, teniendo cuidado al escogerlas no sobrecargar algunas de las líneas de comunicación.
3. Interconexión de redes: posibilitar la transmisión de datos entre estaciones finales situadas en diferentes redes.

Componentes del sistema (red)

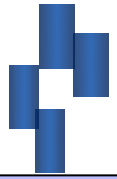
- Equipo de la empresa portadora (enrutadores conectados mediante líneas de transmisión)
- El host (H1) conectado a un enrutador (A) de una empresa portadora mediante una línea alquilada
- El host (H2) en una LAN conectado a un enrutador (F) que a su vez tiene una línea alquilada hasta E



Un host transmite al enrutador más cercano un paquete que tiene por enviar. El paquete se almacena allí hasta que haya llegado por completo, a fin de que la suma de verificación pueda comprobarse. Después se reenvía al siguiente enrutador hasta que lega al host destino, donde se entrega (este proceso es conocido como conmutación de almacenamiento y reenvío).

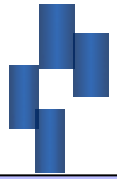
Servicios proporcionados a la capa del transporte

- Deben ser independientes de la tecnología del enrutador.
- La capa de transporte debe estar aislada de la cantidad, tipo y topología de los enrutadores presentes.
- Las direcciones de red disponibles para la capa de transporte deben seguir un plan de numeración uniforme, aún a través de varias LANs y WANs.
- La discusión se centra en determinar si la capa de red debe proporcionar un servicio no orientado a la conexión (SNOC) u orientado a la conexión (SOC).
- Dos ejemplos:
    1. Internet: inherentemente inestable (SNOC). Se inclina por un servicio sin conexión y no confiable.
    2. ATM y compañías telefónicas: calidad de servicio (SOC). Las compañías telefónicas prefieren un servicio confiable orientado a conexión.



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía



2. Conceptos y Técnicas de conmutación

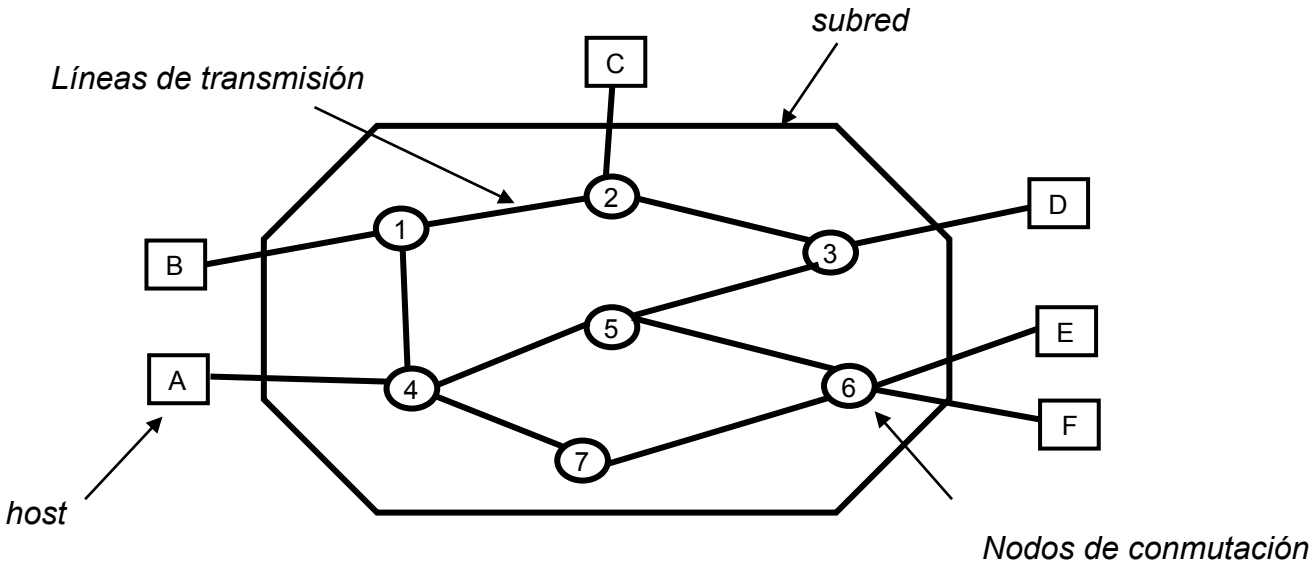
2.1 Componentes de Redes de computadores

2.2 Técnicas de Conmutación

- 2.2.1 Conmutación de circuitos
- 2.2.2 Conmutación de paquetes
  - Conmutación de paquetes mediante circuitos virtuales
  - Conmutación de paquetes mediante datagramas
- 2.2.3 Ejemplos de las técnicas de conmutación

2.3 Comparación de las técnicas de conmutación

- Para simplificar el mantenimiento y ahorrar costes, se introducen elementos específicos en la subred: nodos de conmutación y línea de transmisión.
  - Red no complementamente interconectada: suele haber más de un camino entre cada par de estaciones y diferentes estaciones comparten el mismo camino (necesidad de encaminamiento y control de congestión).



- Retardos de comunicación
  - Retardo de procesamiento de host: tiempo invertido por un host en procesar los datos que se envía (reciben) a (de) la subred.
  - Retardo de transmisión en el enlace: tiempo que tarda el transmisor de cada nodo de la red (host o nodo intermedio) en generar un bloque de datos (ej. paquete) en dicho enlace.
  - Retardo de propagación en el enlace: tiempo que tarda la señal (en realidad un bit de datos) en propagarse por el enlace.
  - Retardo de nodo: tiempo que el bloque de datos (ej. paquete) permanece en un nodo. Es la suma de:
    - Retardo de procesamiento de nodo (ej. decidir por qué enlace de salida se retransmite el paquete).
    - Retardo de espera en cola
      - De entrada: espera de los datos antes de ser procesados por el nodo.
      - De salida: espera de los datos antes de ser reenviados al siguiente nodo intermedio.

Técnicas de conmutación

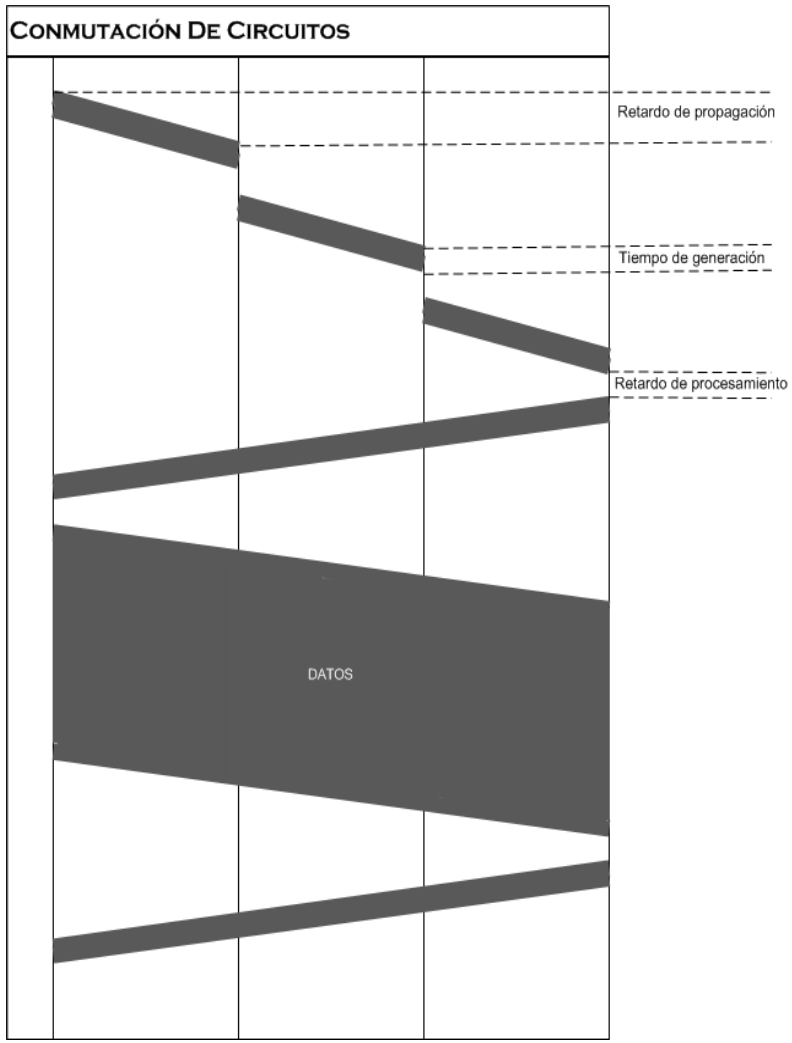
- Las redes que siguen una estructura como la comentada en la sección anterior precisan de una conmutación entre nodos intermedios con el fin de establecer una comunicación origen-destino.
- Se usan dos tipos de técnicas de conmutación:
  - **Conmutación de circuitos**
    - Se conectan físicamente los circuitos desde el emisor hasta el receptor (teléfono).
      - Orientado a conexión.
    - El mensaje se transmite de forma secuencial siguiendo la misma ruta.
  - **Conmutación de paquetes**
    - Se transmite en base a paquetes
      - Son retransmitidos nodo a nodo hasta alcanzar el destino.
    - Dos variantes
      - Datagrama: no se establece conexión
      - Circuitos virtuales: se establece conexión.

Conmutación de circuitos

- Se establece un canal dedicado entre las dos estaciones:
  - Secuencia de enlaces entre nodos
  - Servicio orientado a conexión

- Tres fases
  - ***Establecimiento de la conexión***: se realiza una petición, se evalúan las rutas posibles, se reservan recursos y ancho de banda requeridos.
  - ***Transferencia de datos***: el mensaje se transmite de forma secuencial siguiendo la ruta fijada a una velocidad constante.
  - ***Cierre de conexión***: liberación de las conexiones y recursos.

2.2.1 Conmutación de Circuitos



Evolución temporal

La estación emisora (E) genera un mensaje de establecimiento

- El mensaje se envía al nodo N1, al que se encuentra conectada E

El nodo N1 evalúa el siguiente nodo en la ruta en base a la dirección de destino, N2

- Se reservan los recursos necesarios
- Se repite en todos los saltos necesarios

El destino R responde generando un mensaje de aceptación.

- No hay retardos en los nodos (ruta previamente establecida)

La recepción de la confirmación en E establece la conexión

- Intercambio de información
- Retardo de propagación
- Recepción ordenada

Cierre de la conexión mediante envío de solicitud de cierre por E o R

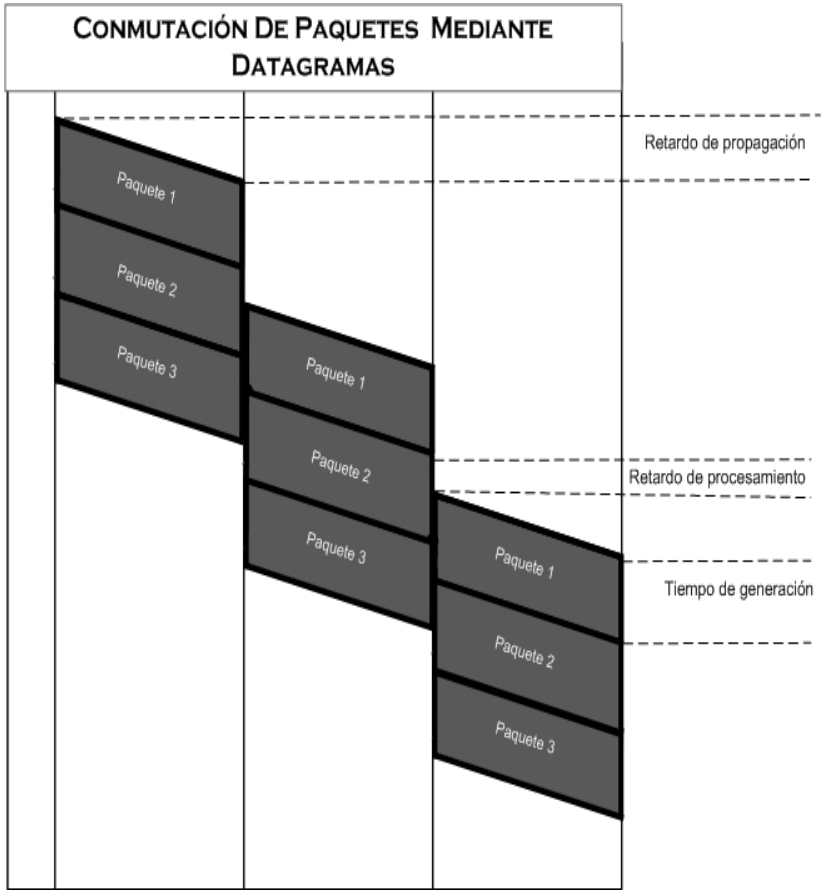
Conmutación de paquetes

- No existe reserva de recursos (al menos implementada en la capa de red!)
- El mensaje se fracciona en paquetes:

  paquete = cabecera + datos.
- Normalmente sigue en funcionamiento de “almacenamiento y reenvío”: un nodo intermedio no retransmite un paquete hasta que no ha sido recibido completamente.

– Dos aproximaciones
- Conmutación de paquetes mediante datagramas.
- Conmutación de paquetes mediante circuitos virtuales.

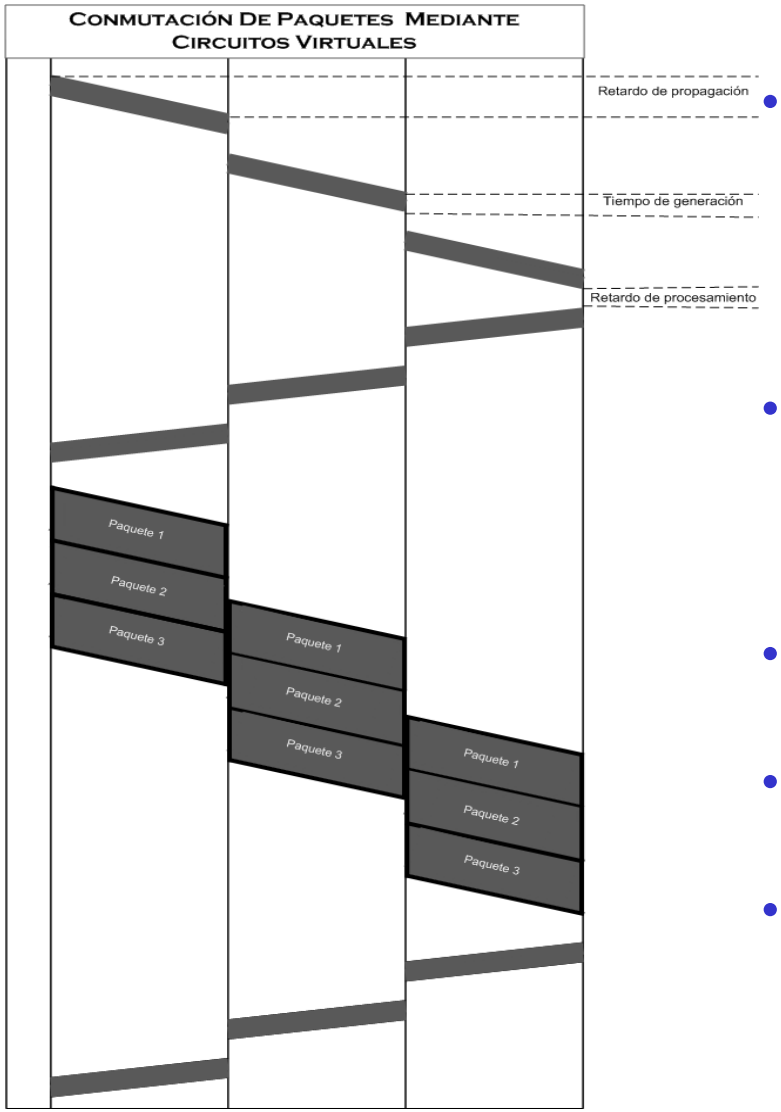
2.2.2 Conmutación de Paquetes



Mediante Datagramas

- No hay establecimiento y cierre de la conexión.
- Las cabeceras de los paquetes contienen la dirección del nodo destino.
- Cada nodo intermedio realiza una decisión de encaminamiento individual para cada paquete
  - cada paquete puede seguir caminos diferentes.
  - posibilidad de recepción de los paquetes sin orden → no hay garantía de recepción ordenada).
- Uso potencial de rutas alternativas en caso de problemas → método muy robusto
- Problemas en servicios interactivos en los que se precisa alta velocidad.

2.2.2 Conmutación de Circuitos



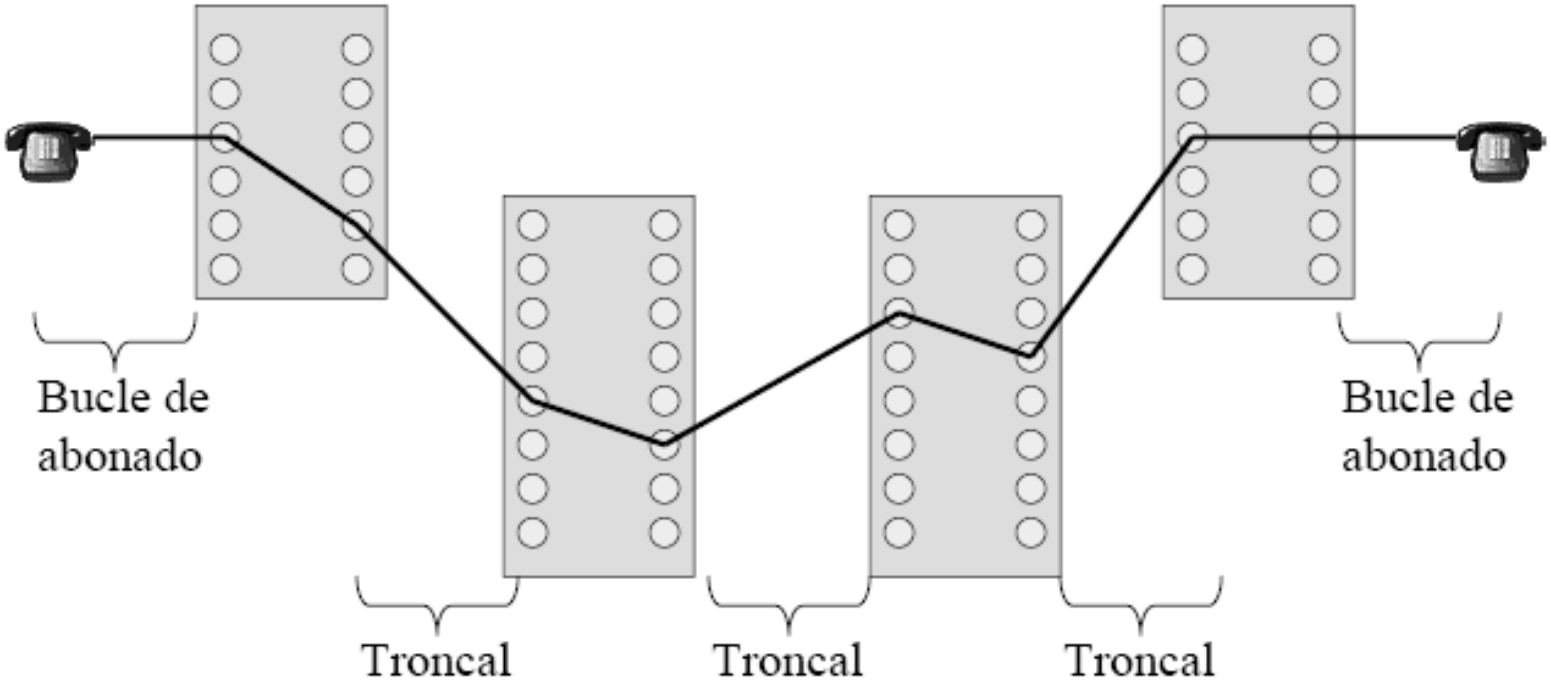
Mediante Circuito virtual

- Se establece un camino fijo antes de enviar paquetes → todos los paquetes del mensaje siguen la misma ruta (circuito virtual)

  Recepción de paquetes ordenada.
- Similar a conmutación de circuitos (se establece un circuito lógico) → pero como diferencia radica que el camino NO es dedicado
- Cada paquete contendrá una indicación del circuito virtual a usar.
- Menor retardo en los nodos (no encaminamiento)
- Problemas frente a desconexión de nodos

2.2.3 Ejemplos de las técnicas

- **Conmutación de circuitos**



Conmutación de paquetes mediante datagramas

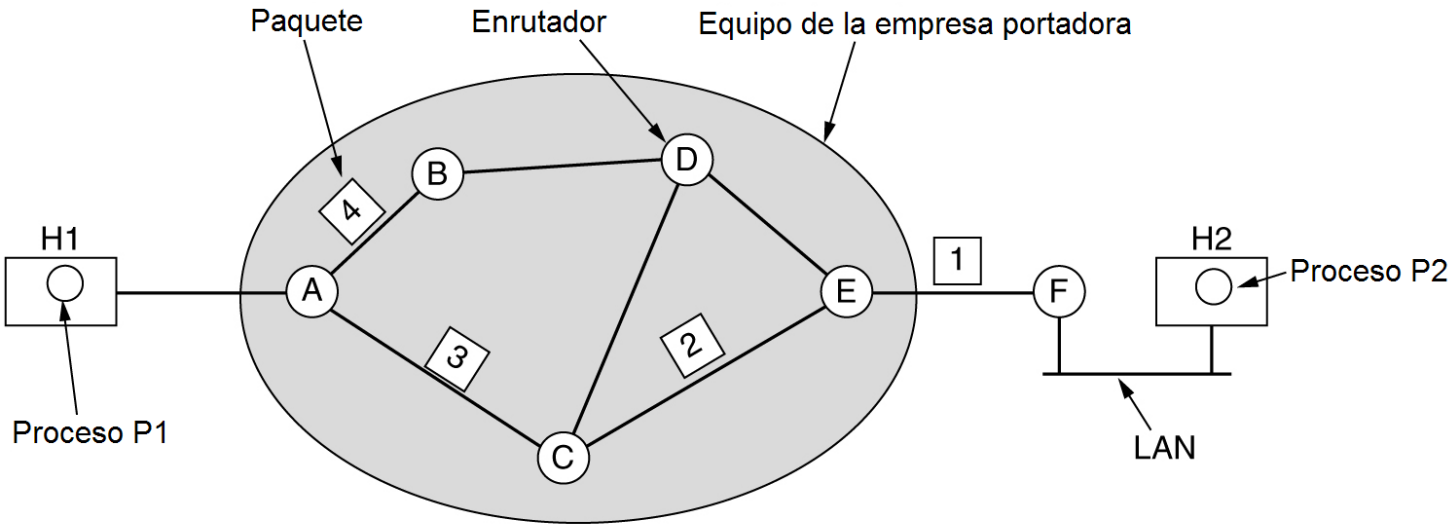


Tabla de A

Dest.	Linea (inicialmente)	Linea (posterior)
A	–	–
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

Tabla de C

Dest.	Linea
A	A
B	A
C	–
D	D
E	E
F	E

Tabla de E

Dest.	Linea
A	C
B	D
C	C
D	D
E	–
F	F

Enrutamiento dentro de una subred de datagramas. Algo pasa con el paquete 4, que se envía por B. Tal vez A se enteró de que había una congestión en la ruta ACE y actualizó su tabla de enrutamiento posteriormente. El algoritmo que maneja las tablas y realiza las decisiones de enrutamiento se llama algoritmo de enrutamiento

Conmutación de paquetes mediante circuitos virtuales

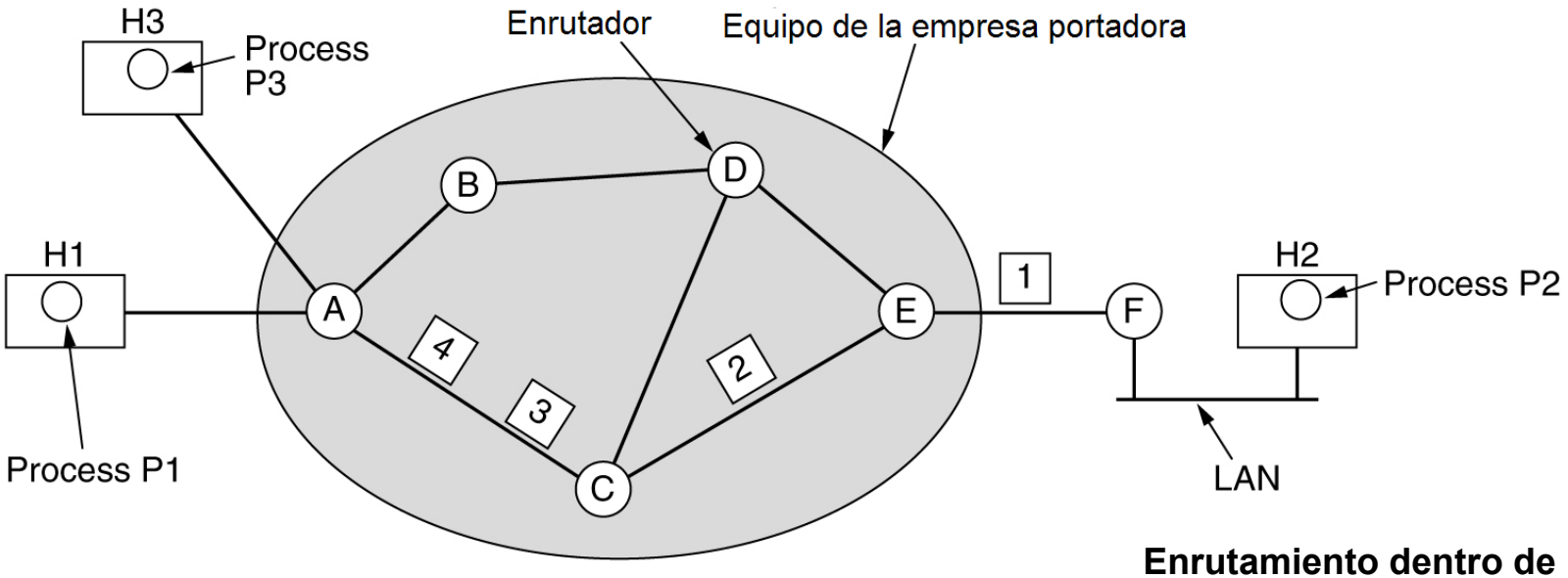


Tabla de A

H1	1
H3	1

Dentro

C	1
C	2

Fuera

Tabla de C

A	1
A	2

E	1
E	2

Tabla de E

C	1
C	2

F	1
F	2

Enrutamiento dentro de una subred de circuitos virtuales. H1 ha establecido una conexión 1 con H2, se recuerda como la primera entrada de cada una de las tablas de enrutamiento. H3 también desea establecer una conexión con H2. Elige el identificador de conexión 1, porque es su primera y única conexión. El router A puede saber que paquetes vienen de H1 o H3, pero C no, por lo que A asigna un identificador de conexión diferente al tráfico saliente para la segunda conexión.

Comparación de las Técnicas de Conmutación

- **Rendimiento:**
  - Tres tipos de retardos: propagación, transmisión y procesamiento en nodo (tiempo en cola más decisión de encaminamiento)
  - Descubrimiento del camino
    - Una sola vez en conmutación de circuitos
    - Ida y vuelta en circuitos virtuales
    - Para cada paquete en datagrama

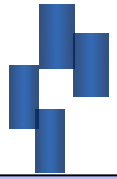
- **Parámetros relevantes**
  - Ancho de banda: cuántos recursos se necesitan y cómo de eficientemente se utilizan.
  - Complejidad en los nodos intermedios: capacidades que deben presentar los nodos
  - Latencia/retardo: tiempo que tarda la transmisión
  - Robustez

2.3 Comparación de las Técnicas

	Circuitos	Paquetes mediante datagramas	Paquetes mediante circuitos virtuales
Utilización de los recursos de la red	-Uso estático: Canal dedicado -No existen bits suplementarios en la transmisión salvo los propios de establecimiento y cierre.	-Uso dinámico: No existe conexión -Existencia de bits suplementarios en cada paquete.	-Uso dinámico: circuito no dedicado -Existencia de bits suplementarios en cada paquete (pero menos que en datagramas) y los propios del establecimiento y cierre de la conexión.
Complejidad de los nodos	Conmutadores simples.	Necesidad de memoria de almacenamiento en los nodos.	-Necesidad de memoria de almacenamiento más números de circuito virtual en los nodos. -Menor retraso por nodo que en datagramas.
Latencia de la red (retardo extremo a extremo)	Adecuada para aplicaciones interactiva y en tiempo real	Poco adecuadas para aplicaciones interactivas y en tiempo real	Poco adecuada para aplicaciones interactivas en tiempo real
Robustez	Nodo o enlace falla -> comunicación finaliza	Nodo o enlace falla -> se busca alternativas	Nodo o enlace falla -> comunicación finaliza

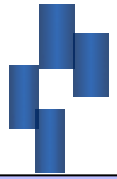
2.3 Comparación de las Técnicas

	Subred de Datagramas	**Subred de circuitos virtuales**
Configuración del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y destino	Cada paquete contiene un número de CV corto
Enrutamiento	Cada paquete se enruta de manera independiente	Ruta escogida cuando se establece el CV, todos los paquetes siguen ruta
Efecto de fallas del enrutador	Ninguno, excepto para paquetes perdidos durante una caída	Terminan todos los CV que pasan a través del enrutador
Calidad del servicio	Difícil	Fácil, si se pueden asignar recursos por adelantado para cada CV
Control de congestión	Difícil	Fácil, si se pueden asignar recursos por adelantado para cada CV



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía



3. Enrutamiento

3.1 Propiedades de los algoritmos de Enrutamiento

3.2 Características que especifican un Algoritmo de

3.3 Enrutamiento

3.4 Principio de optimización

3.5 Algoritmos de Enrutamiento

Algoritmos de Enrutamiento

- El algoritmo de enrutamiento es aquella parte del software de la capa de red encargada de decidir la línea de salida por la que se retransmitirá un paquete de entrada.
  - Si la subred usa datagramas, esta decisión debe tomarse cada vez que llega un paquete de datos, dado que la mejor ruta podría haber cambiado desde la última vez.
  - Si la subred usa circuitos virtuales, las decisiones de enrutamiento se toma sólo al establecerse un circuito virtual nuevo. En lo sucesivo los paquetes siguen la ruta previamente establecida.

Propiedades de los Algoritmos de Enrutamiento

- Independientemente de las características del algoritmo, hay ciertas propiedades que todo algoritmo de enrutamiento debe poseer:
  - Exactitud,
  - Robustez,
  - Estabilidad,
  - Eficiencia y Simplicidad,
  - Equidad,
  - Optimización.

Características de los Algoritmos de Enrutamiento

- Los aspectos involucrados en la especificación de un algoritmo de encaminamiento y a partir de los cuales se define éste, son los siguientes:

1. El criterio de decisión: métrica en base a la cual se toma la decisión de encaminamiento. Ejemplos: distancia, número de saltos, eficiencia, retardo, …
2. El instante de decisión: momento en el que se toma la decisión de encaminamiento. Ejemplos: inicio de sesión (circuito virtual), para cada paquete (datagrama)
3. El lugar de decisión: el punto o puntos en que se toman las decisiones de encaminamiento. Ejemplos: en cada nodo (distribuido), en un nodo central (centralizado), en el nodo origen (fuente).

Características de los Algoritmos de Enrutamiento

4. La fuente de información: procedencia de los datos tenidos en consideración para la toma de decisión. Puede ser local a cada nodo, de lo nodos adyacentes, de los nodos en la ruta, todos los nodos.
5. El tiempo de actualización: establece la adaptación de las tomas de decisión a las condiciones cambiantes de la red. Pueden ser algoritmos estáticos o adaptables.

- A partir de estos aspectos podemos encontrar varias clasificaciones de los diferentes algoritmos de encaminamiento. En nuestro caso, vamos considerar como criterio de clasificación el tiempo de actualización, dividiendo así los algoritmos en estáticos y dinámicos. A su vez, los dinámicos, se clasificarán en función del fuente de información y del criterio de decisión.

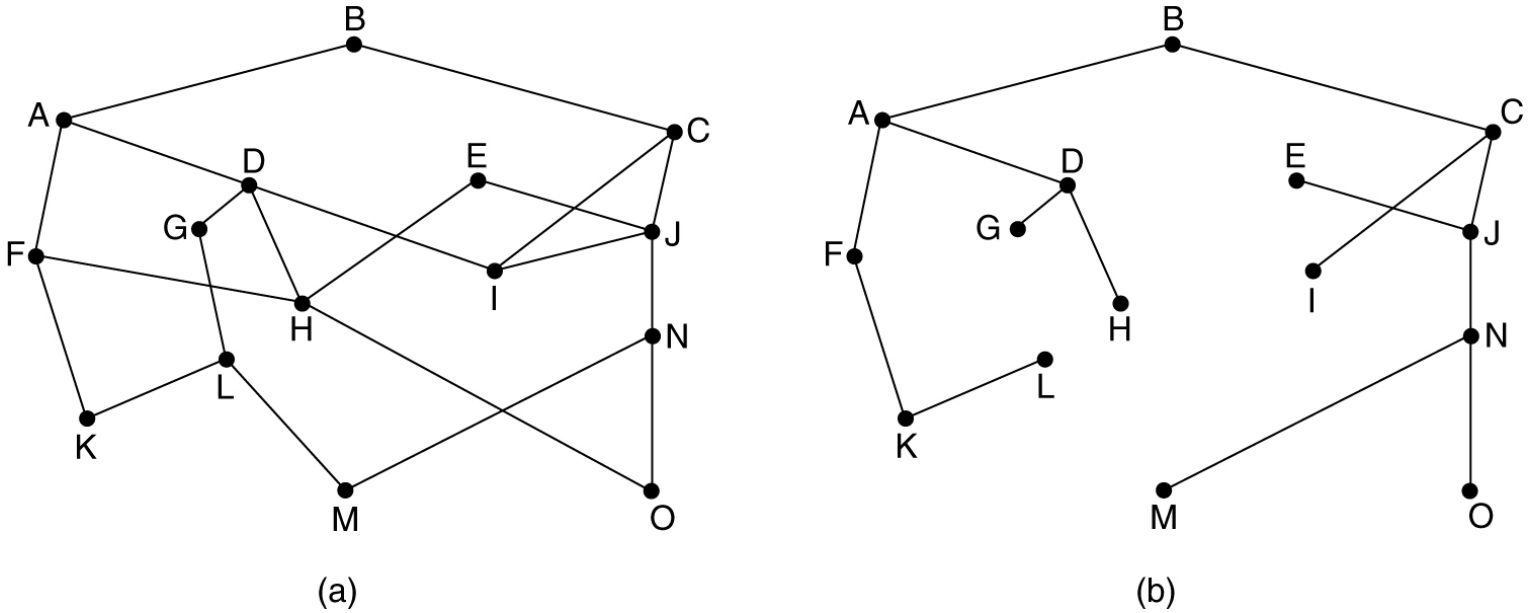
Principio de Optimización

- Establece que si el enrutador J está en ruta óptima del enrutador I al enrutador K, entonces la ruta óptima de J a K, también está en la misma ruta.

- Como consecuencia directa del principio de optimización, el grupo de rutas óptimas de todos los orígenes a un destino dado forman un árbol con raíz en el destino. Este árbol se conoce como **árbol sumidero**.

- La meta de todos los algoritmos de enrutamiento es descubrir y utilizar estos árboles sumideros de los enrutadores

Principio de Optimización



(a) Una subred (b) Arbol sumidero para el enrutador B
La métrica de distancia es el número de saltos

- Los algoritmos de enrutamiento pueden agruparse en dos clases principales:

1. Algoritmos de enrutamiento estáticos: la decisión de qué ruta se usará para llegar de I a J, se toma por adelantado fuera de línea y se carga en los enrutadores al arrancar la red. No toman en cuenta la carga actual de la red.

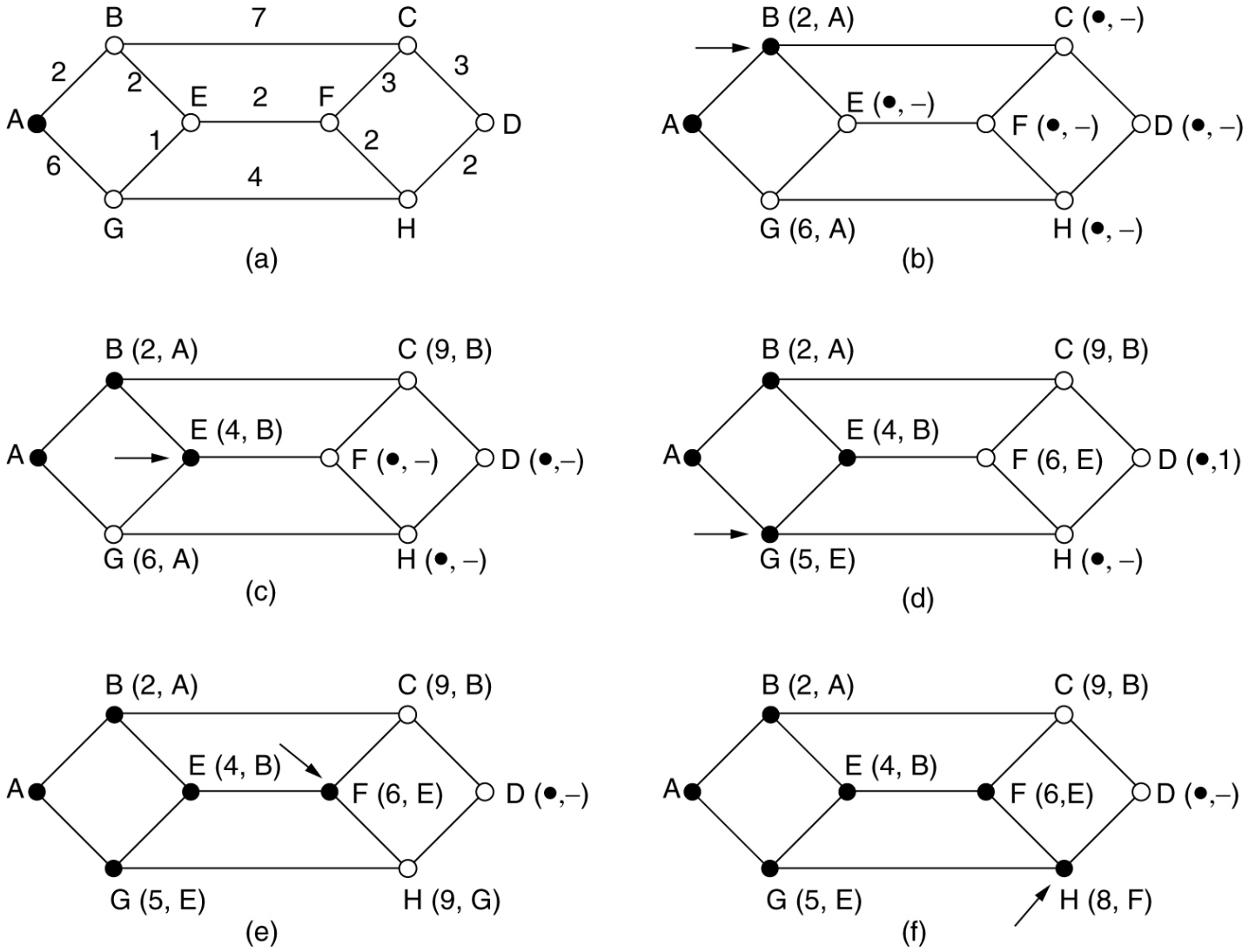
2. Algoritmos de enrutamiento dinámicos: cambian sus decisiones de enrutamiento para reflejar los cambios en la topología y el tráfico, estos difieren en **el lugar de donde obtienen su información** (de los enrutadores adyacentes o de todos) y la **métrica usada para la optimización** (distancia, número de saltos, o tiempo estimado de tráfico).

- Algoritmos de enrutamiento
  1. **Enrutamiento por la ruta más corta o de Dijkstra** (1959)
  2. **Inundación**
  3. **Enrutamiento por vector distancia o de Bellman-Ford (1962).** Se usó en ARPANET hasta 1979
  4. **Enrutamiento por estado del enlace**. Reemplazó en ARPANET al de Bellman-Ford a partir de 1979
  5. **Enrutamiento jerárquico**
  6. **Enrutamiento por difusión y multidifusión**
  7. **Enrutamiento por hosts móviles**
  8. **Enrutamiento en redes ad-hoc**

Enrutamiento por la ruta más corta

- La idea es armar un **grafo de la subred**, en el que cada nodo representa un enrutador y cada arco del grafo una línea de comunicación, llamada enlace
- Para elegir una ruta entre un par dado de enrutadores, el algoritmo simplemente encuentra en el grafo la ruta más corta entre ellos
- La ruta más corta se mide en base a varias métricas:
  1. Cantidad de saltos
  2. Distancia geográfica en Kms
  3. Retardo medio de encolamiento
  4. Costo de comunicación
- En Dijkstra cada nodo se etiqueta (entre paréntesis) con su distancia al nodo de origen a través de la mejor ruta conocida. Las etiquetas pueden ser tentativas o permanentes

3.4.1 Enrutamiento por la ruta más corta



Grafo ponderado no dirigido, donde las ponderaciones representan distancias. Los primeros 5 pasos del cálculo de la ruta más corta de A a D. Nodos tentativos y permanentes.

Enrutamiento por inundación

- Cada paquete de entrada se envía por cada una de las líneas de salida, excepto aquella por la que llegó.

- La inundación genera grandes cantidades de paquetes duplicados, para evitar esto se pueden tomar algunas medidas:
  1. Integrar un contador de saltos en el encabezado de cada paquete, que disminuya con cada salto → el paquete se decarta cuando llega a 0.
  2. Llevar un registro de los paquetes difundidos para evitar enviarlos por segunda vez:
     - Cada enrutador de origen pone un número de secuencia en cada paquete que recibe de sus hosts.
     - Cada enrutador necesita una lista por cada enrutador de origen que indique los números de secuencia originados en ese enrutador que ya ha visto. Si llega un paquete duplicado se descarta y no se difunde.

Enrutamiento por inundación selectiva

- Es una variación un poco más práctica → los enrutadores no envían cada paquete de entrada por todas las líneas de salida, sino sólo por aquellas que van aproximadamente en la dirección correcta.

Características del enrutamiento por inundación

- Algunas aplicaciones dónde es interesante esta técnica:
  - Aplicaciones militares.
  - Aplicaciones distribuidas de base de datos.
  - Redes inalámbricas.
  - Métrica para comparar otros algoritmos de enrutamiento.

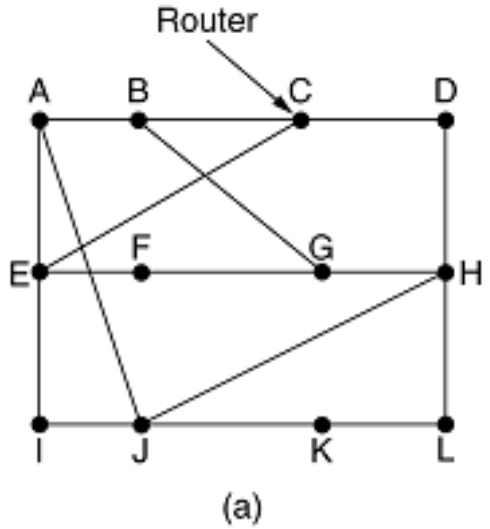
Enrutamiento por vector distancia

- Operan haciendo que cada enrutador mantenga una tabla (es decir un vector) que da la mejor distancia conocida a cada destino y la línea que se puede usar para llegar ahí. Estas tablas se actualizan intercambiando información con los vecinos.

- Cada tabla contiene dos entradas: la línea preferida de salida hacia ese destino y una estimación del tiempo, distancia o cantidad de saltos para llegar a él.

- Se supone que cada enrutador conoce la "distancia" a cada uno de sus vecinos. Una vez cada T mseg, cada enrutador envía a todos sus vecinos una lista de sus distancias estimadas a cada destino. Si la métrica es el retardo puede medirlo el enrutador con paquetes especiales ECO que el receptor simplemente marca con la hora y lo regresa tan rápido como puede.

Enrutamiento por vector distancia



(a)

Nuevo retardo estimado desde J

To	A	I	H	K	Nuevo	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA	JI	JH	JK		
Retardos:	8	10	12	6		

Vectores recibidos de los cuatro vecinos de J

Nueva tabla de enrutamiento para J

(b)

(a) Una subred. (b) Entrada de A, I, H, K y la nueva tabla de enrutamiento de J. Veamos como J calcula su nueva ruta a G. A través de A tarda (18 + 8=26), I (31+10=41), H (6+12=18), K (31+6=37). El mejor de estos valores es 18 por lo que escribe una entrada en la tabla indicando retardo a G=18 mseg vía H.

Enrutamiento por vector distancia

- El enrutamiento por vector distancia se usó en ARPANET hasta 1979, cuando fue reemplazado por el enrutamiento por estado del enlace.

- Dos problemas principales causaron su desaparición:

  1) No tenía en cuenta el ancho de banda de la línea.

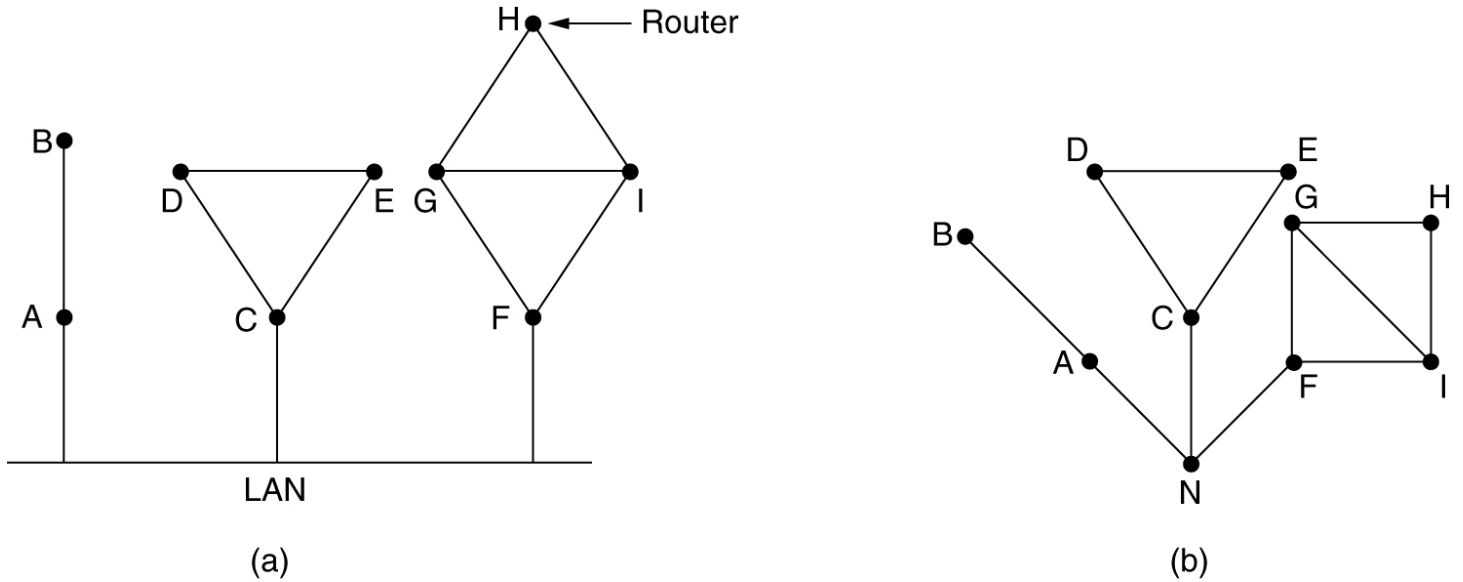
  2) El problema de la cuenta hasta infinito cuando se desconecta un enrutador. Pues cuando X indica a Y que tiene una ruta en algún lugar, Y no tiene forma de saber si él mismo está incluido en la ruta.

Enrutamiento por estado del enlace

- El concepto en que se basa el enrutamiento por estado del enlace puede enunciarse en cinco partes:
  1. Descubrir a sus vecinos y conocer sus direcciones de red.
  2. Medir el retardo o costo para cada uno de sus vecinos.
  3. Construir un paquete que indique todo lo que acaba de aprender.
  4. Enviar este paquete a los demás enrutadores
  5. Calcular la ruta más corta a todos los demás enrutadores.

3.4.4 Enrutamiento por estado del enlace

Paso1. Conocimiento de los vecinos



(a) Nueve enrutadores conectados a una LAN. (b) Modelo de grafo de (a) considerando la LAN como un nodo más.

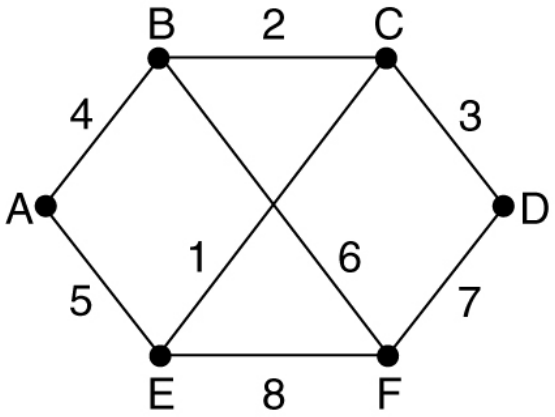
- Cuando el enrutador se pone en funcionamiento su primera tarea es averiguar quiénes son sus vecinos. Para esto envía un paquete HELLO especial a cada línea punto a punto.
- El enrutador del otro extremo regresa una respuesta indicando quién es.
- Estos nombres deben ser globalmente únicos.

Paso 2. Medición del costo de la línea

- El algoritmo de enrutamiento por estado del enlace requiere que cada enrutador sepa del retardo a cada unos de sus vecinos
- Para medirlo se envía un paquete ECO especial a través de la línea y una vez que llegue al otro extremo regresarlo inmediatamente.
- Se mide entonces el tiempo de ida y vuelta y se divide entre dos.
- La prueba se puede llevar a cabo varias veces y calcular el retardo medio.
- En el cálculo de los tiempos de retardo se pueden incluir las cargas de las líneas, con lo cual entre dos líneas con igual ancho de banda, una con una carga continua y otra sin ella, el enrutador considerará como ruta más corta la de la línea sin carga.

3.4.4 Enrutamiento por estado del enlace

Paso 3. Construcción de los paquetes de estado de enlace



(a)

Paquetes de estado de enlace

A	
Sec.	
Edad	
B	4
E	5

B	
Sec.	
Edad	
A	4
C	2
F	6

C	
Sec.	
Edad	
B	2
D	3
E	1

D	
Sec.	
Edad	
C	3
F	7

E	
Seq.	
Edad	
A	5
C	1
F	8

F	
Sec.	
Edad	
B	6
D	7
E	8

(b)

(a)Subred. (b) Paquetes de estado del enlace para esta subred

El siguiente paso es que cada enrutador construya un paquete que contenga todos los datos.

La parte difícil es determinar cuándo construir los paquetes. Una posibilidad es construirlos de manera periódica, a intervalos regulares, otra es hacerlo con la caída o reactivación de una línea.

Paso 4. Enviando los paquetes de estado de enlace

Origen	Sec.	Edad	Banderas envío A	Banderas envío C	Banderas envío F	Banderas ACK A	Banderas ACK C	Banderas ACK F	Datos
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

La tabla muestra el búfer de paquetes para el enrutador B. Cada fila corresponde a un paquete de estado del enlace recién llegado, además hay banderas de envío y confirmación de recepción para cada una de las tres líneas de B (A, C, F).

- La idea fundamental es **utilizar inundación para distribuir los paquetes de etado enlace.**
- **Cada paquete utiliza un número de secuencia**, que se incrementa con cada paquete nuevo enviado. Los enrutadores llevan una tabla para guardar la lista de paquetes ya vistos y comparar.
- Se incluye la **edad** (TTL, time to leave) a cada paquete, con lo que el enrutador especifica que pasado ese tiempo sin recibir lo consideren caído.

Paso 5. Cálculo de las nuevas rutas

- Una vez que un enrutador ha acumulado un grupo completo de paquetes de estado de enlace, puede construir el grafo de la subred completa.

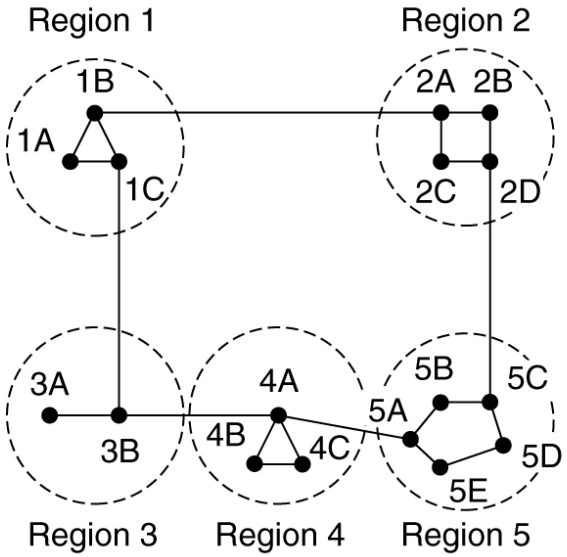
- Después se ejecuta el algoritmo de Dijkstra para construir la ruta más corta a todos los destinos posibles.

- Para una subred con n enrutadores cada uno de los cuales tiene k vecinos, la memoria requerida para almacenar los datos de entrada es proporcional a k*n. Esto puede ser un problema en redes muy grandes, como también puede serlo el tiempo de cómputo → sin embargo, en muchas situaciones prácticas este enrutamiento funciona bien.

Enrutamiento jerárquico

- A medida que crece el tamaño de las redes, también lo hacen de manera proporcional las tablas de enrutamiento del enrutador, consumiendo más memoria y tiempo de CPU para examinarlas.
- La red puede crecer hasta tal punto que no sea factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores, por lo que el enrutamiento debe hacerse de manera jerárquica como en la red telefónica.
- Los enrutadores se dividen en lo que llamaremos **regiones**, donde cada enrutador conoce todos los detalles para enviar paquetes dentro de su propia región, pero no sabe nada sobre la estructura interna de otras regiones.
- En las redes enormes una jerarquía de dos niveles puedes ser insuficiente; tal vez sea necesario agrupar las regiones en clústeres, los clústeres en zonas, las zonas en grupos,…

Enrutamiento jerárquico



(a)

Tabla completa para 1A

Dest.	Linea	Saltos
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Tabla jerárquica para 1A

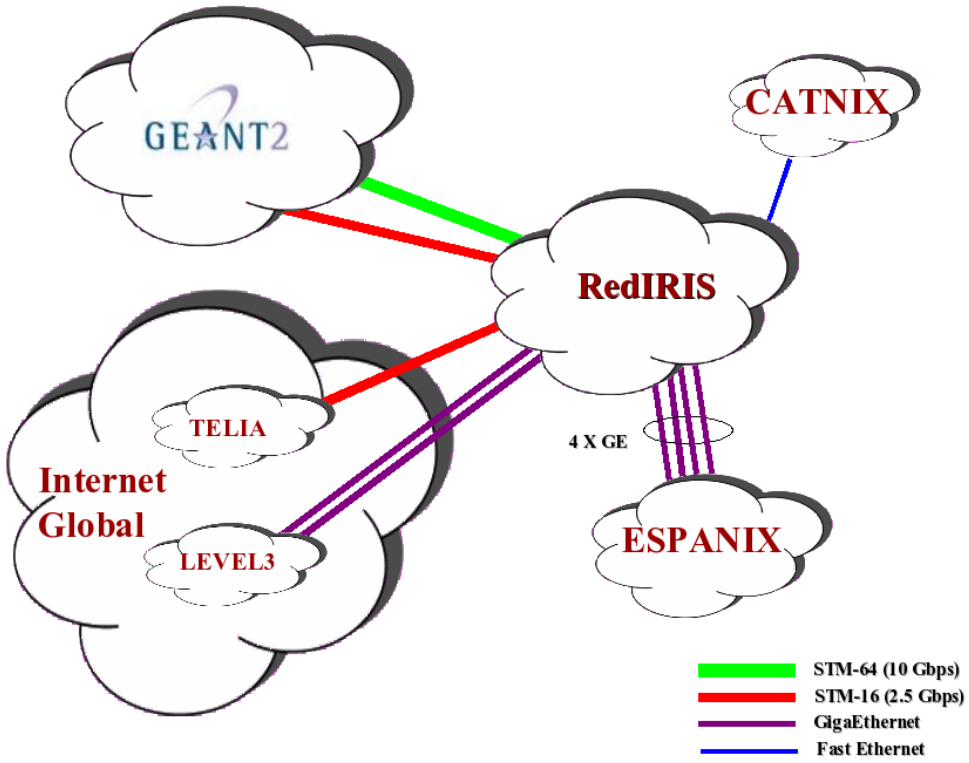
Dest.	Linea	Saltos
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Enrutamiento jerárquico. La tabla completa para 1A se muestra en b). En c) hay entradas para los enrutadores locales y las demás regiones se han condensado en un solo enrutador. Es más reducida pero a costa de obtener una longitud de ruta mayor, en algunos casos (1A-5C) se va mejor por la región 2, pero va por la 3 porque es la mejor para mayoría.

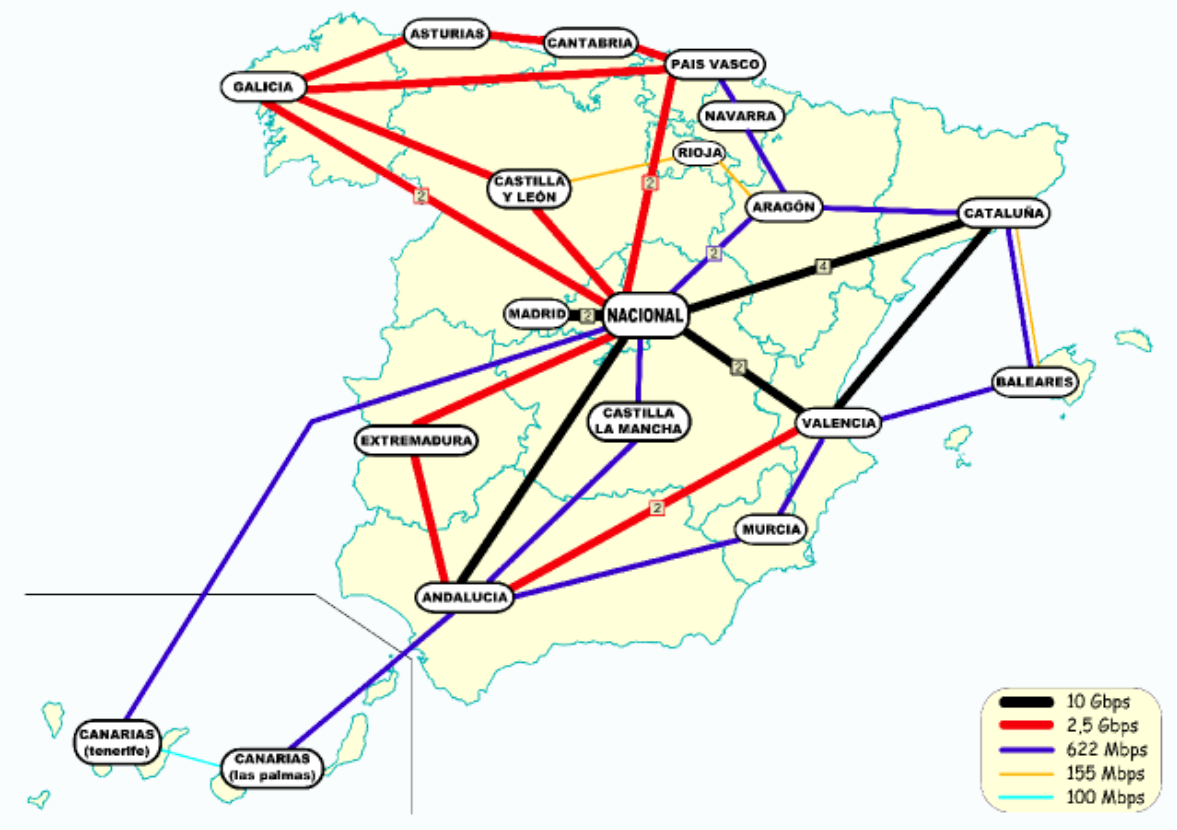
Enrutamiento jerárquico

- Red iris(www.rediris.es). RedIRIS es el proveedor de conexión para las Universidades en España



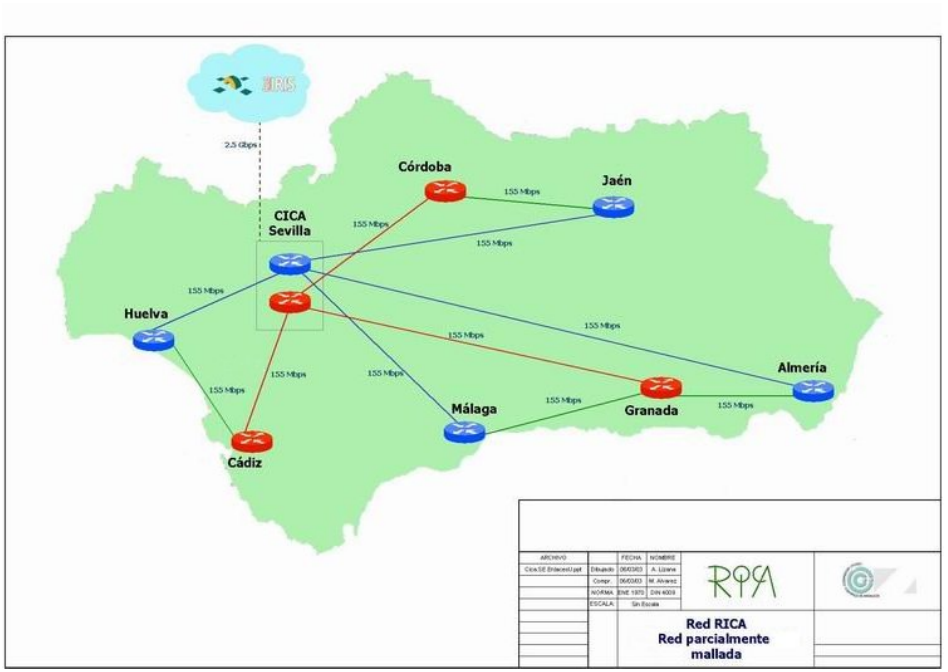
Enrutamiento jerárquico

- Red iris(www.rediris.es). RedIRIS comunica la instituciones universitarias y de investigación españolas entre sí, a través de las redes académicas autonómicas.



Enrutamiento jerárquico

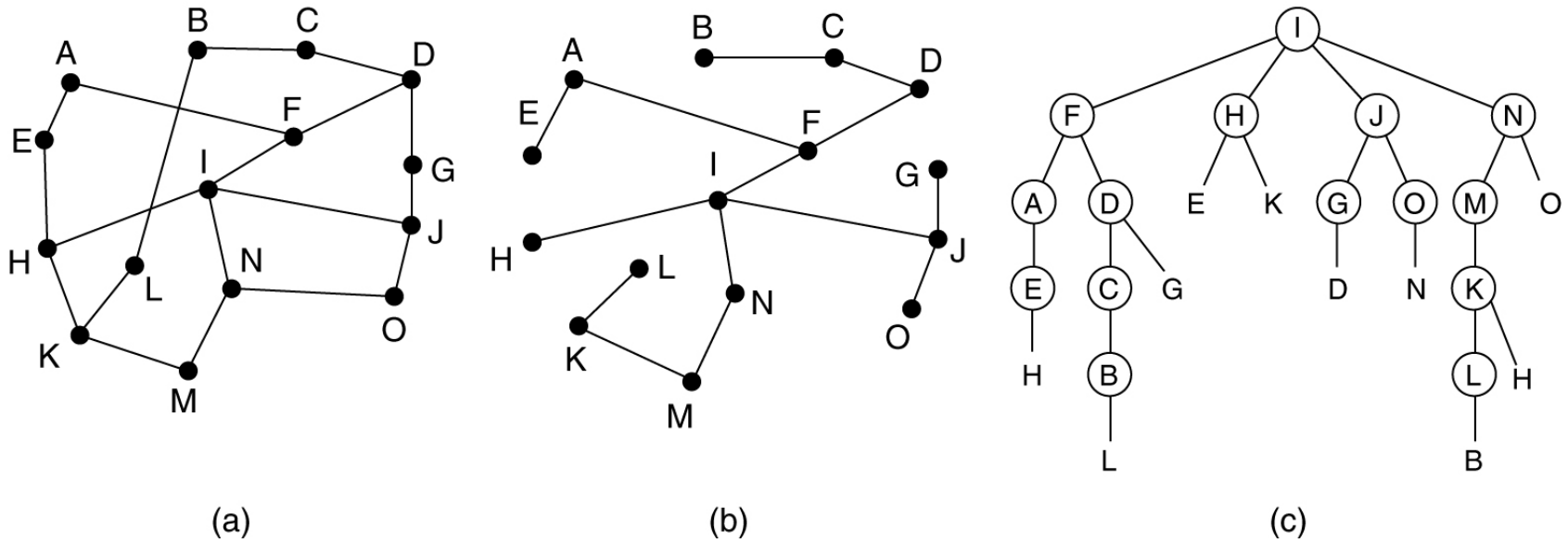
- Red cica (www.cica.es).
  - Los Centros de Investigación de Andalucía están conectados entre sí y con el exterior mediante una red homogénea de comunicaciones de alta tecnología: la red RICA (Red Informática Científica Andaluza).
  - Integrada dentro de la red académica española RedIRIS (Interconexión Recursos Informáticos), RICA forma parte de Internet y ofrece a sus usuarios acceso a servicios distribuidos a nivel global.



Enrutamiento por difusión

- En algunas aplicaciones, los host necesitan enviar mensajes a otros hosts o a todos los demás. Ejemplo el servicio de distribución de informes ambientales, precios de bolsas podrían funcionar mejor difundiéndolas a todas las máquinas y dejando que las que estén interesadas lean los datos.
- Modos de envío, individuales, inundación, enrutamiento multidestino, uso del árbol sumidero o reenvío por ruta invertida.
- El envío simultáneo de un paquete a todos los destinos se llama difusión. Un **método de difusión es el reenvío por ruta invertida**: cuando llega un paquete difundido a un enrutador, éste lo revisa para ver si llegó por la línea normalmente usada para enviar paquetes al origen de la difusión. Si este es el caso reenvía copias del paquete a todas las líneas excepto a aquella por la que llegó, en caso contrario se descarta como posible duplicado.

Enrutamiento por difusión



(a) Subred. (b) Árbol sumidero. (c) Reenvío por ruta invertida utilizando los árboles sumideros.

Enrutamiento por multidifusión

- Algunas aplicaciones requieren que procesos muy separados trabajen en grupo (base de datos distribuida), en estos casos es frecuente que un proceso envíe un mensaje a todos los demás miembros del grupo.

- Si el grupo es pequeño y la subred es grande enviar por difusión a todos es costoso e ineficiente porque la mayoría de los receptores no están interesados en este mensaje.

- El envío de un mensaje a un grupo bien definido de tamaño numéricamente grande, pero pequeño en comparación con la totalidad de la subred se llama **multidifusión**.

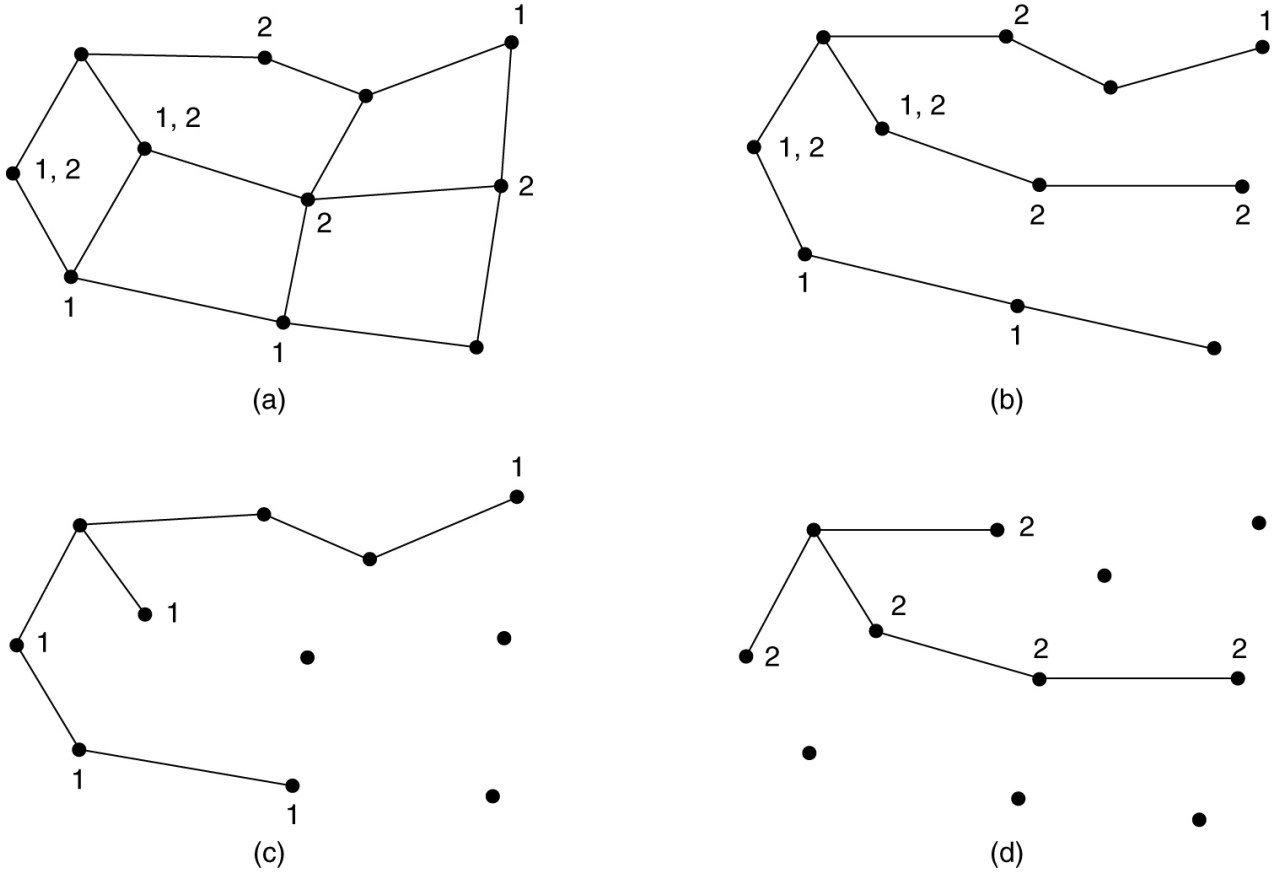
Enrutamiento por multidifusión

- Para multidifusión se requiere administración de grupo. Los hosts deben informar a los enrutadores a qué grupo pertenecen.

- Para realizar enrutamiento de multidifusión, cada enrutador calcula un árbol de expansión que cubre a todos los demás enrutadores de la subred. Algunos enrutadores estarán conectados a hosts que pertenecen a uno o más grupos.

- Cuando un proceso envía un paquete de multidifusión a un grupo, el primer enrutador examina su árbol de expansión y lo recorta, eliminando todas las líneas que conduzcan a hosts que no sean miembros del grupo.

Enrutamiento por multidifusión

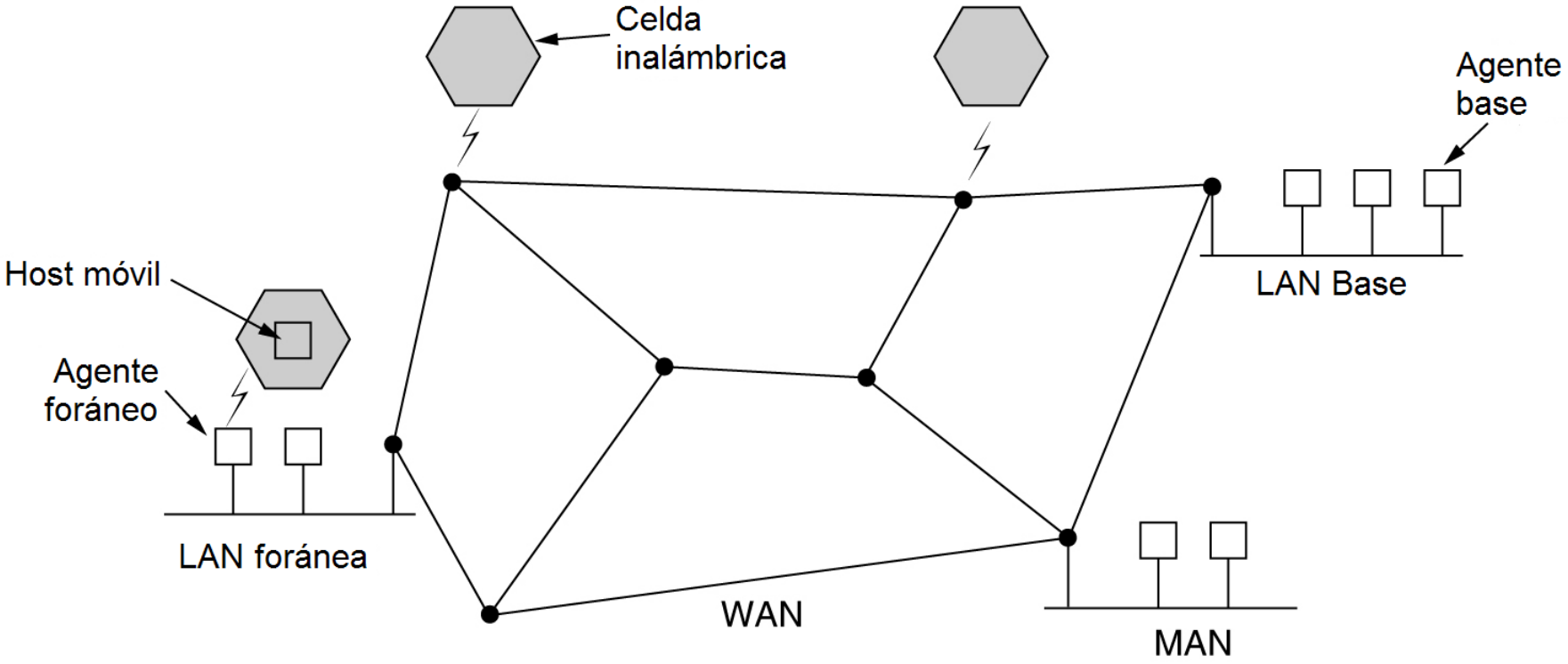


(a) Subred. (b) Árbol de expansión del enrutador extremo izquierdo (c) Árbol de multidifusión del grupo 1. (d) Árbol de multidifusión del grupo 2

Enrutamiento por host móviles

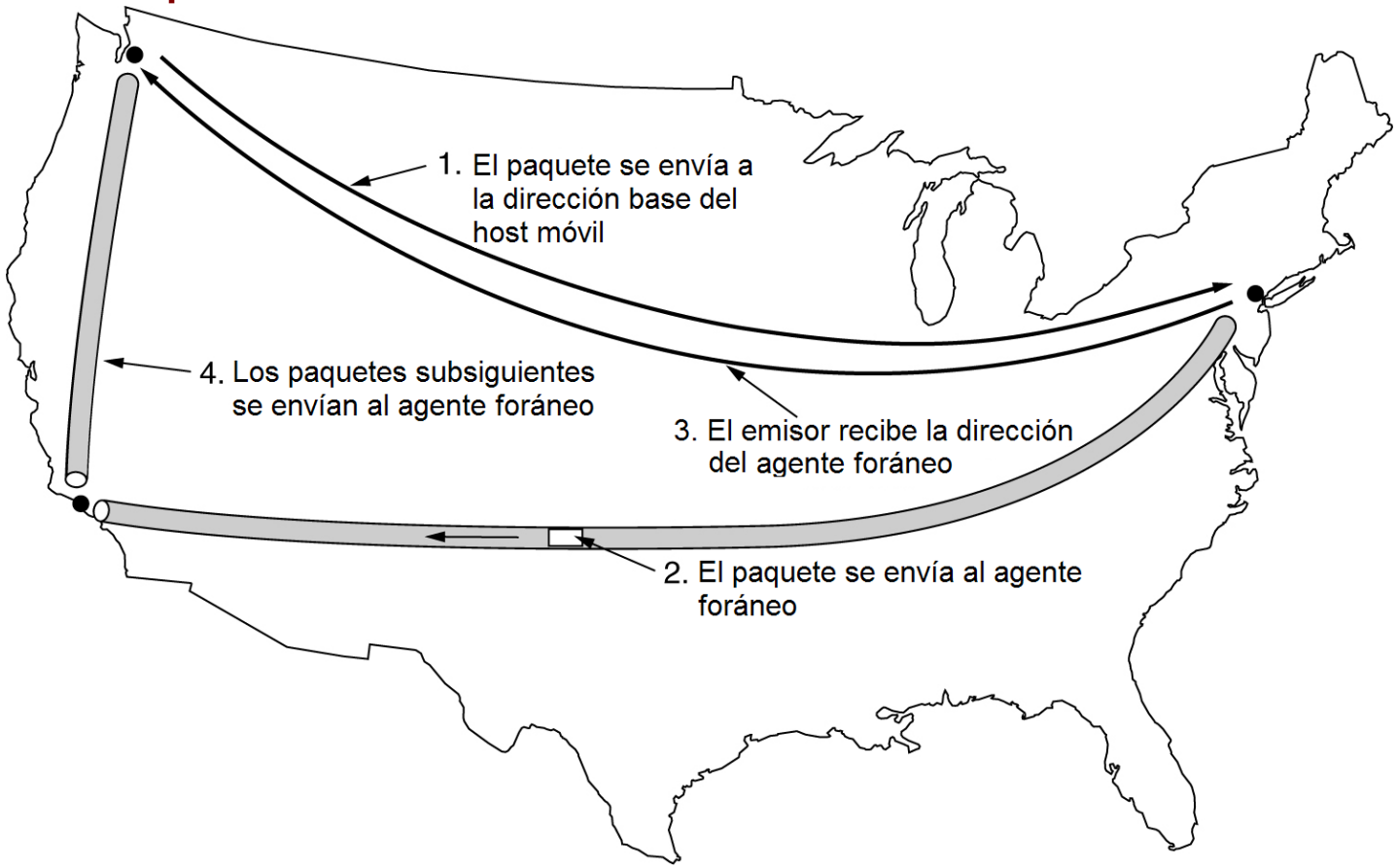
- En la actualidad millones de personas tienen computadoras portátiles y quieren leer sus correos y acceder a sus archivos normales desde cualquier parte del mundo.
- Se dice que los host que nunca se mueven son estacionarios y se conectan a la red por cable o fibra óptica. Los hosts móviles están lejos de casa y necesitan seguir conectados, se pueden distinguir migratorios y ambulantes.
- Enrutamiento en los sistemas con host móviles:
  - Cada host móvil tiene una localidad base que nunca cambia y una dirección base permanente que identifica su localización base.
  - El mundo se divide en áreas, cada área tiene uno o más **agentes foráneos**, que son procesos que llevan registro de todos los hosts móviles que visiten el área.
  - Cada área tiene además un **agente base** que lleva el registro de todos los hosts cuya base está en el área, pero que actualmente está visitando otra área.
  - Cuando un host entra en un área, ya sea al conectarse a ella (LAN) o al entrar en la celda, su computadora debe **registrarse con el agente foráneo de ese lugar.**

Enrutamiento por host móviles



Modelo del mundo: una WAN que consiste en enrutadores y host. Conectada a la WAN hay varias LANs y MANs y celdas inalámbricas

Enrutamiento por host móviles

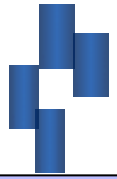


Enrutamiento de paquetes para host móviles.

Enrutamiento en redes ad-hoc

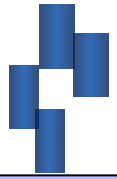
- No sólo los host son móviles sino también los enrutadores.
- Ejemplos:
  1. Vehículos militares en un campo de batalla
     - Sin infraestructura
  2. Una flota de barcos en alta mar.
     - Moviéndose todo el tiempo
- En una red ad hoc, la topología puede cambiar en cualquier momento, por lo que la validez de la rutas cambia a diferencia de las cabledas. Cada nodo consiste en un enrutador y un host en la misma computadora. Los enrutadores pueden ir y venir en cualquier momento.

- Se propone el algoritmo de enrutamiento AODV (Vector de distancia ad-hoc bajo demanda)



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía



4. Algoritmos de Control de Congestión

4.1 Introducción

4.2 Principios generales del control de congestión

4.3 Políticas de prevención de congestión

4.4 Control de congestión en circuitos virtuales

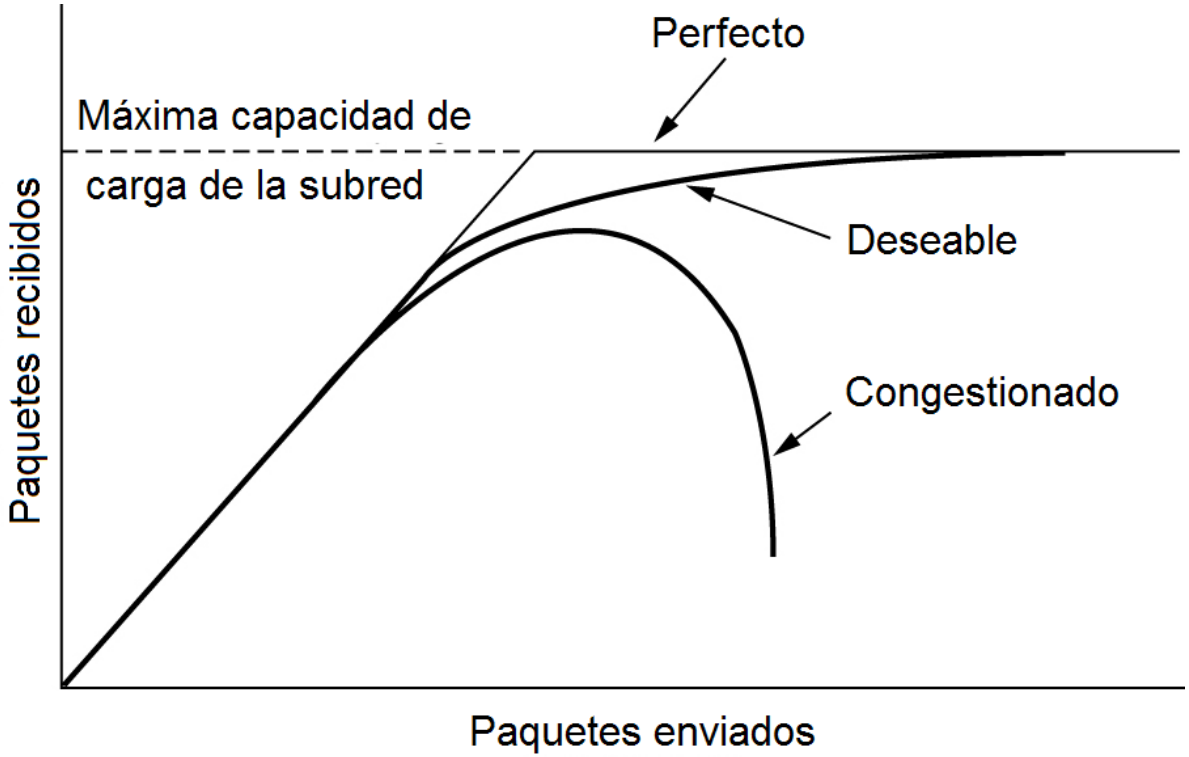
4.5 Control de congestión en subredes de datagramas

4.6 Desprendimiento de carga

4.7 Control de fluctuación

- Cuando hay demasiados paquetes presentes en la subred, hay una degradación del desempeño. Esta situación se llama **congestión**. La carga es (temporalmente) superior (en una parte del sistema) a la que pueden manejar los recursos.

- La congestión puede ocurrir por varias razones:
  1. Si llegan muchos paquetes por varias líneas de entrada para una misma línea de salida
  2. Si las CPU de los enrutadores son lentas las colas pueden alargarse
  3. Líneas de poco ancho de banda



Cuando se genera demasiado tráfico, ocurre congestión y se degrada el desempeño.

- El **control de congestión** se ocupa de asegurar que la subred sea capaz de transportar tráfico ofrecido. Es un asunto global
- El **control de flujo** se relaciona con el tráfico punto a punto entre un emisor dado y un receptor dado

Las soluciones de control de congestión las podemos dividir en dos grupos:

1. Soluciones de ciclo abierto: intentan resolver el problema mediante un buen diseño, para asegurarse de que no ocurran. Una vez que el sistema está en funcionamiento, no se hacen correcciones a medio camino.

   Las herramientas para llevar a cabo el control por ciclo abierto incluyen:

   - Decidir cuándo aceptar tráfico nuevo
   - Decidir cuándo descartar paquetes y cuáles
   - Tomar decisiones de calendarización

2. Soluciones de ciclo cerrado: se basan en el concepto de un ciclo de retroalimentación de forma que constantemente se monitorea el sistema y se actúa en consecuencia.

Las soluciones de ciclo cerrado, comprende tres funciones:

1. Monitorear el sistema para detectar cuándo y dónde ocurren las congestiones
   - Utilizar distintas métricas: pocentaje de paquetes descartados por espacio de búfer, longitud promedio de las colas, retardo medio de los paquetes
2. Pasar esta información a lugares en los que pueden llevarse a cabo acciones.
   - El enrutador que detecta la congestión, envía un paquete (si hay congestión es ineficiente)
   - Se puede reservar un bit en cada paquete de salida para avisar a los vecinos de que está congestionado
   - Que los enrutadores envién paquetes de sondeo para preguntar sobre congestión.
3. Ajustar la operación del sistema para corregir el problema

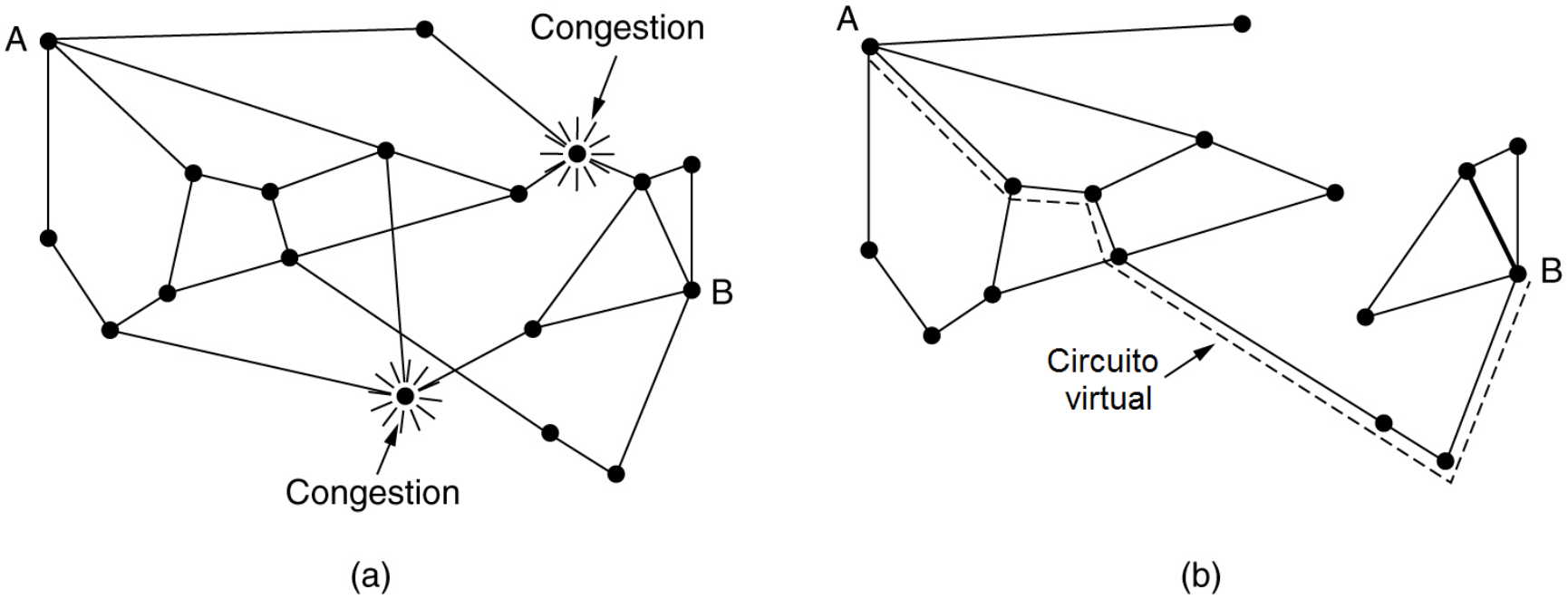
La presencia de congestión significa que la carga es (temporalmente) superior (en una parte del sistema) a la que puedan manejar los recursos. Vienen a la mente dos soluciones:

- Aumentar los recursos
  - La subred emplee otras líneas (p.ej líneas de acceso telefónico).
  - Dividir el tráfico entre varias rutas en lugar de usar siempre la mejor.
  - Emplear enrutadores de repuesto que normalmente sirve sólo como respaldo.
- Disminuir la carga
  - Negar servicio a algunos usuarios
  - Degradar el servicio para algunos o todos los usuarios
  - Obligar a los usuarios a programar sus solicitudes de una manera más predecible

Capa	Políticas
Transporte	• Política de retransmisiones • Política de confirmaciones de recepción • Política de control de flujo • Determinación de terminaciones de temporizador
Red	• Política de encolamiento y servicio de paquetes (líneas de entrada y salida del enrutador) • Política de descartes de paquetes (qué paquetes se descartan cuando no hay espacio) • Algoritmo de enrutamiento (un buen algoritmo evita la congestión) • Administración del tiempo de ida y vuelta del paquete (intervalo de expiración antes de descartar, TTL de los paquetes)
Enlace de datos	• Política de retransmisiones y almacenamiento en caché de paquetes fuera de orden (repetición selectiva mejor que retroceso N) • Política de confirmación de recepción (superposición) • Política de control de flujo (ventanas corredizas)

Métodos para el control dinámico de la congestión en las subredes de circuitos virtuales:

- Control de admisión
  - Una vez que se ha detectado la congestión, no se establecen circuitos virtuales nuevos,
  - Se admiten nuevas conexiones pero enrutando cuidadosamente los CV nuevos por otras rutas que no tengan problemas
- Negociar un acuerdo entre el host y la subred cuando se establece el circuito
  - Se reserva recursos a lo largo de la ruta establecida en el circuito (espacio en tablas y en búfer en los enrutadores y ancho de banda en las líneas)



(a) Subred congestionada. (b) Subred redibujada que elimina la congestión, se muestra un circuito virtual entre A y B

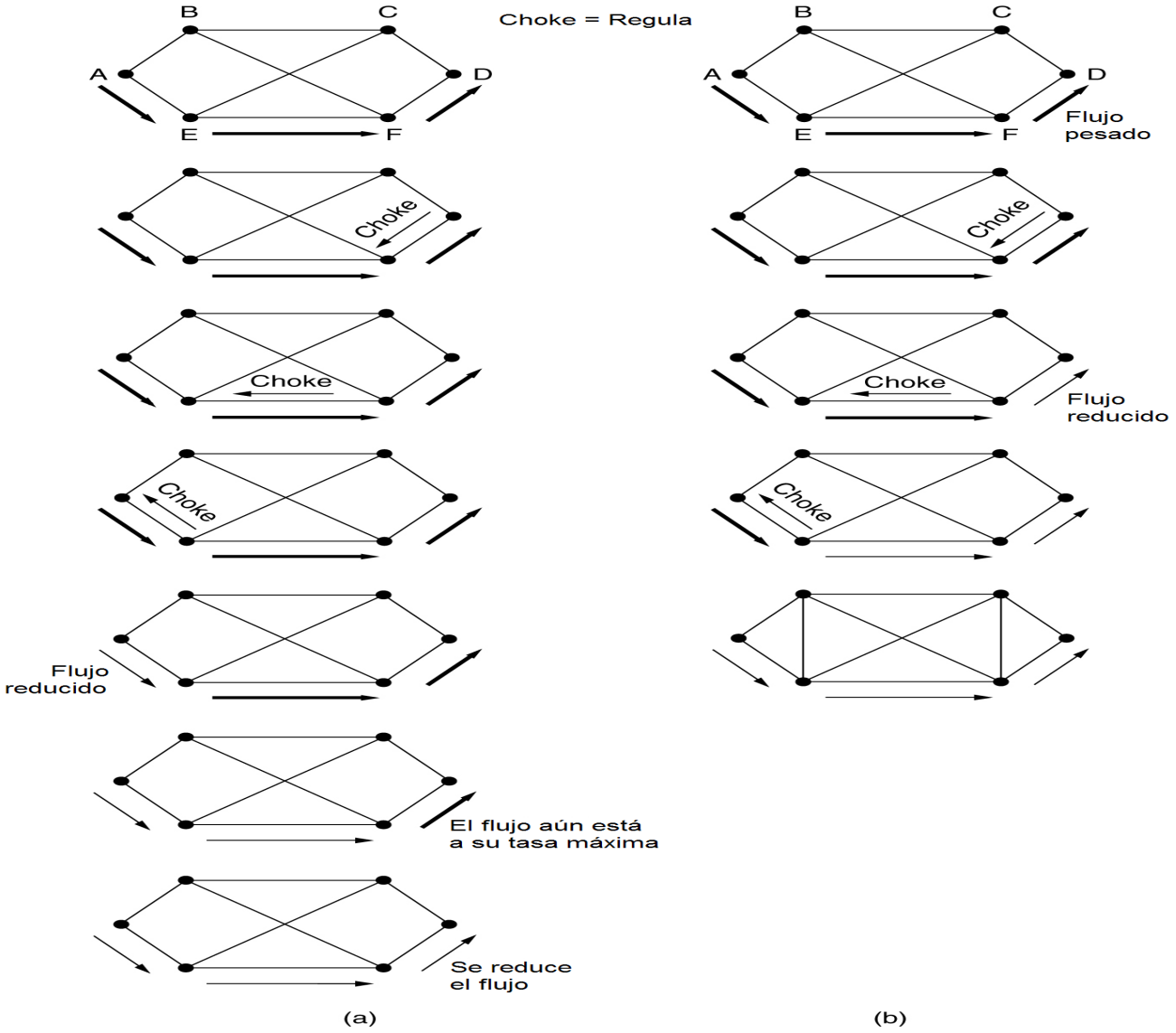
Los métodos aquí establecidos sirven tanto para subredes de datagramas como para subredes de circuitos virtuales.

El proceso consiste en los siguientes pasos:

- Cada enrutador monitorea el uso de sus líneas de salida
- Asocia a cada línea una variable real, u, de 0.0 a 1.0, que refleja el uso reciente. Para tener una buena estimación, la puede tomar periódicamente e ir actualizándola.
- Cuando el valor de u, rebasa un umbral, la línea se establece en un estado de "advertencia".
- Cuando llega un nuevo paquete se revisa para ver si su línea de salida está en estado de advertencia. Si es así, se realiza una de las siguientes acciones:
  - + *El bit de advertencia*
  - + *Paquetes reguladores*
  - + *Paquetes reguladores salto a salto*

1. Bit de advertencia
   - Cuando un enrutador está en estado de advertencia, activa un bit especial en el encabezado del paquete, que cuando llega al destino lo copia en la confirmación de recepción que al recibirlo el origen reduce el tráfico.
2. Paquetes reguladores
   - El enrutador regresa un paquete regulador al origen, proporcionándole el destino encontrado en el paquete. El paquete original se etiqueta para que no se generen más paquetes reguladores en adelante en la ruta.
   - Cuando el host de origen obtiene el paquete regulador, se le pide que reduzca en un porcentaje X el tráfico enviado al destino especificado
3. Paquetes reguladores de salto por salto
   - A altas velocidades o distancias grandes, el envío del paquete regulador no funciona bien porque la reacción es muy lenta
   - Lo ideal es que el paquete regulador ejerza su efecto en cada salto que da

4.5 Control de Congestión en Subredes De Datagramas



(a) Paquetes reguladores. Sólo hasta el séptimo diagrama el enrutador de D no notará el flujo más lento

(b) Paquetes reguladores de salto por salto. El paquete regulador ejerce su efecto en cada salto que da, obligando a los enrutadores intermedios a reducir su flujo a D, con más búferes para los paq. hacia D

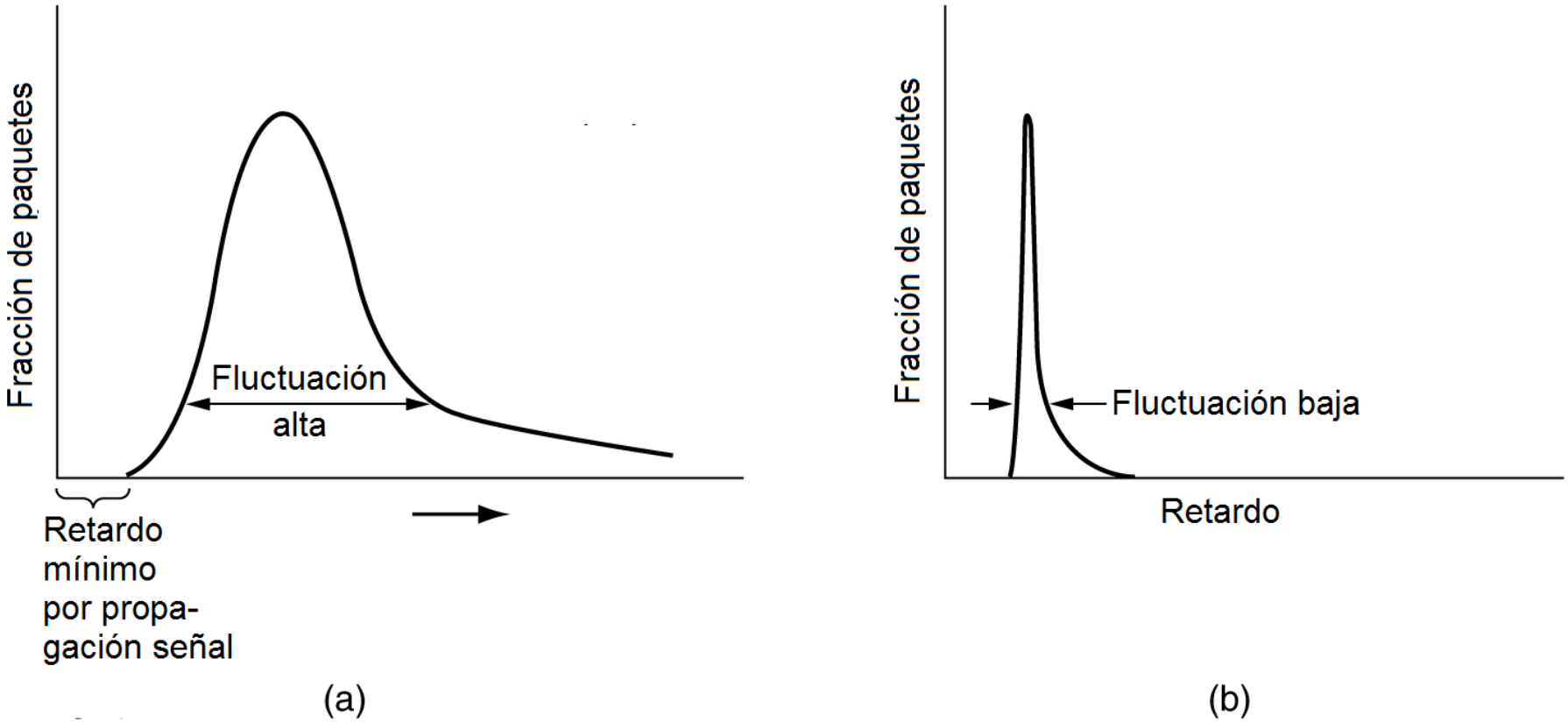
- Cuando ninguno de los métodos anteriores elimina la congestión, los enrutadores pueden sacar la artillería pesada: el **desprendimiento de carga**, o sea que cuando se inundan a los enrutadores con paquetes que no pueden manejar, simplemente los tiran. El origen notará en algún momento la falta de confirmación de recepción y tomará medidas

- El paquete a descartar puede depender de las aplicaciones que se estén ejecutando:
  - Si es una aplicación FTP, se descartan los nuevos, pues si tenemos retroceso N, entonces habría que reenviar más cantidad de paquetes
  - En multimedia, se descartan los viejos. La pérdida de algunos píxeles de una imagen es mucho menos dañina que la pérdida de una línea de texto legible

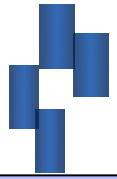
- Para poner en práctica una política inteligente de descarte, las aplicaciones deben marcar sus paquetes con clases de prioridades para indicar su importancia. Así los enrutadores pueden descartar primero los paquetes de clase más baja y así sucesivamente.

- Es mejor tratar con la congestión cuando se detecta por primera vez que dejar que dañe el trabajo y luego tratar de solucionarlo. El **algoritmo de detección temprana aleatoria** se basa en este funcionamiento. Según las métricas, comienza a descartar antes de que sea tarde. Para determinar cuándo comenzar a descartarlos, los enrutadores mantienen un promedio de sus longitudes de cola. Cuando la longitud de cola promedio de algunas líneas de salida sobrepasan un umbral, se dice que la línea está congestionada

- En aplicaciones como la transmisión continua de video y audio no importa gran cosa si los paquetes tardan 20 o 30 mseg en ser entregados siempre y cuando el tránsito (retardo) sea constante
- La variación (desviación estándar) en el retardo de los paquetes se conoce como **fluctuación**
- Una fluctuación alta, por ejemplo cuando unos paquetes tardan en llegar 20 mseg y otros 30 mseg resultará en una calidad desigual del sonido o la imagen
- La fluctuación puede limitarse calculando el tiempo de tránsito esperado para cada salto en la ruta. Cuando un paquete llega a un enrutador, éste lo examina para saber qué adelantado o retrasado está respecto a lo programado. Si está adelantado se retiene un tiempo, si está retrasado, se intenta enviar entre los primeros

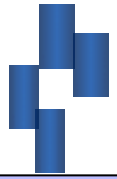


(a) Alta fluctuación. (b) Baja fluctuación



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía



5. Calidad del servicio

5.1 Introducción

5.2 Requerimientos

5.3 Técnicas para alcanzar una buena QoS

5.3.1 Sobreaprovisionamiento

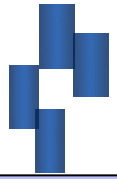
5.3.2 Almacenamiento en búfer

5.3.3 Modelado de tráfico

5.3.4 Algoritmo de cubeta con goteo

5.3.5 Algoritmo de cubeta con tokens

5.3.6 Calendarización de paquetes



5. Calidad del servicio

5.4 Técnicas para alcanzar una buena QoS en servicios diferenciados

5.4.1 Reenvío expedito o acelerado

5.4.2 Reenvío asegurado

- Cada vez más, se requiere proporcionar una calidad de servicio que se ajuste a las necesidades de las aplicaciones.
- Para este fin, no es suficiente con las técnicas de prevención y control de congestión. Las técnicas de prevención y control de congestión ayudan a reducir la congestión y mejorar el rendimiento de las red.
- Se necesitan intentos serios para garantizar la calidad del servicio a través del diseño de redes y protocolos.

- Un **flujo** es un conjunto de paquetes que van de un origen a un destino. La necesidad de cada flujo se puede caracterizar por cuatro parámetros:

  1. Confiabilidad
  2. Retardo
  3. Fluctuación
  4. Ancho de banda

- Estos parámetros en conjunto determinan la **QoS (Calidad del servicio)**

5.2 Requerimientos

Aplicación	Confiabilidad	Retardo	Fluctuación	Ancho de Banda
Correo electrónico	Alta	Bajo	Baja	Bajo
FTP	Alta	Bajo	Baja	Medio
Acceso a Web	Alta	Medio	Baja	Medio
Inicio de sesión remoto	Alta	Medio	Media	Bajo
Audio bajo demanda	Baja	Bajo	Alta	Medio
Video bajo demanda	Baja	Bajo	Alta	Alto
Telefonía	Baja	Alto	Alta	Bajo
Videoconferencia	Baja	Alto	Alta	Alto

Las primeras 4 aplicaciones tienen requerimientos rigurosos en confiabilidad, no es posible enviar bits de manera incorrecta (CRC). Las 4 finales pueden tolerar errores.

FTP, correo y video no son sensibles al retardo, sin embargo las interactivas (telefonía y videoconferencia) si necesitan poco retardo

- Para lograr los requerimientos de QoS, no existe una técnicas que proporciene QoS eficientes y confiables de manera óptima.
- Existen una gran variedad de técnicas con soluciones prácticas que con frecuencia se combinan para obtener los mejores resultados posibles.
- En esta sección examinaremos algunas de estas técnicas que los diseñadores de sistemas utilizan para alcanzar la QoS:
    1. Sobreaprovisionamiento
    2. Almacenamiento en búfer
    3. Modelado de tráfico
    4. Algoritmo de cubeta con goteo
    5. Algoritmo de cubeta con tokens
    6. Calendarización de paquetes

5.3.1 Sobreaprovisionamiento

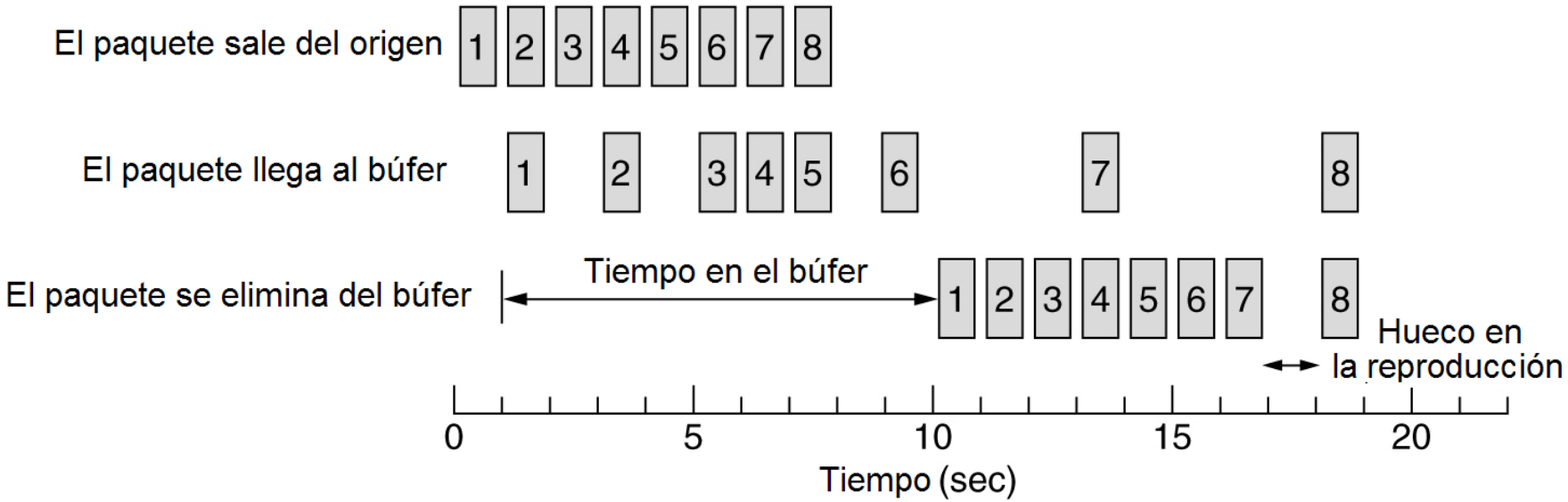
- Consiste en proporcionar la suficiente capacidad de enrutador, espacio en búfer y ancho de banda como para que los paquetes fluyan con facilidad.

- Inconveniente: es muy costosa

- Se vuelve más eficiente conforme pasa el tiempo y los diseñadores tienen una mejor idea de la cantidad de recursos que es suficiente

- El sistema telefónico emplea esta técnica.

5.3.2 Almacenamiento en búfer



Refinamiento del flujo de paquetes almacenándolos en el búfer

Es una técnica útil para reducir la fluctuación. Los flujos pueden almacenarse en el búfer del del receptor antes de ser entregados, lo cual no afecta la confiabilidad o el ancho de banda, pero sí incrementa el retardo. Los sitios Web comerciales que contienen transmisión continua de video o audio utilizan reproductores que almacenan en el búfer aprox 10 seg antes de reproducir

5.3.3 Modelado del tráfico

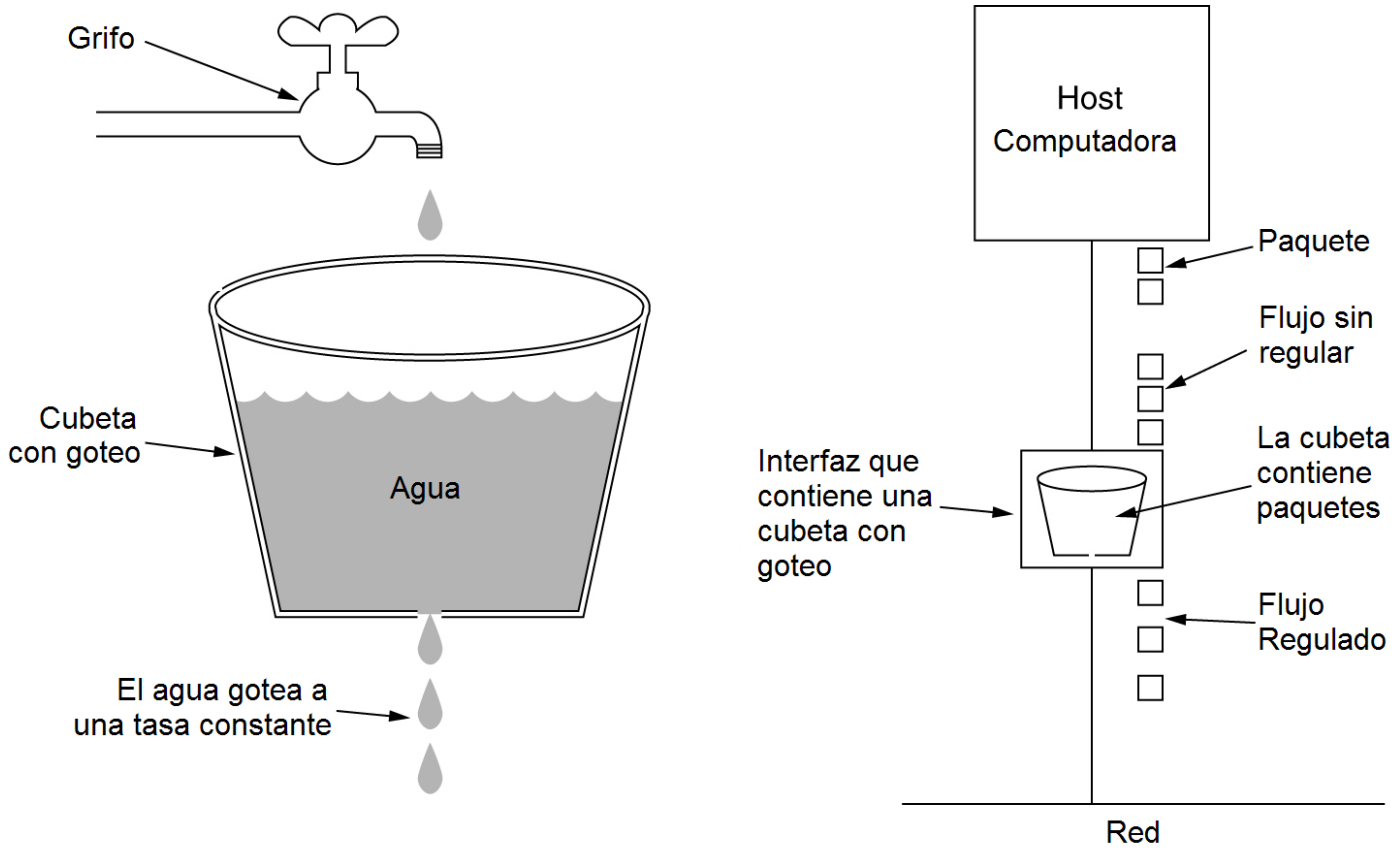
- En el caso que hemos estudiado, el origen envía los datos con un espaciado uniforme entre ellos de una manera regular
- Normalmente el envío es no uniforme y no siempre se puede usar almacenamiento en búfer (caso de aplicaciones de interacción en tiempo real, videoconferencia)
- La solución es que el origen transmita a una tasa uniforme, entonces la calidad del servicio mejoraría
- Esto se llama modelado de tráfico, que consiste en regular la tasa promedio (y las ráfagas) de la transmisión de datos. Para llevarlo a cabo necesitamos:
  - Acuerdo de nivel de servicio
  - Supervisión de tráfico

5.3.4 Algoritmos cubeta con goteo

- Sin importar la rapidez con que entra el agua a la cubeta, el flujo de salida tiene una tasa constante ρ
- Cada host está conectado a la red mediante una interfaz (tarjeta) que contiene una cubeta con goteo. En cada pulso de reloj se transmite un paquete
- Cuando los paquetes no tienen igual tamaño, se habla de número de bytes. De esta forma, cada pulso de reloj se transmiten un determinado número de bytes (que se corresponderá con un número variable de paquetes)

  Si envíamos 1024 bytes por pulso, se corresponderá con un paquete de 1024 bytes, 2 paquetes de 512 bytes, 4 paquetes de 256 bytes, etc.

5.3.4 Algoritmos cubeta con goteo



(a) Una cubeta con goteo llena de agua. (b) Una cubeta de goteo llena de paquetes.

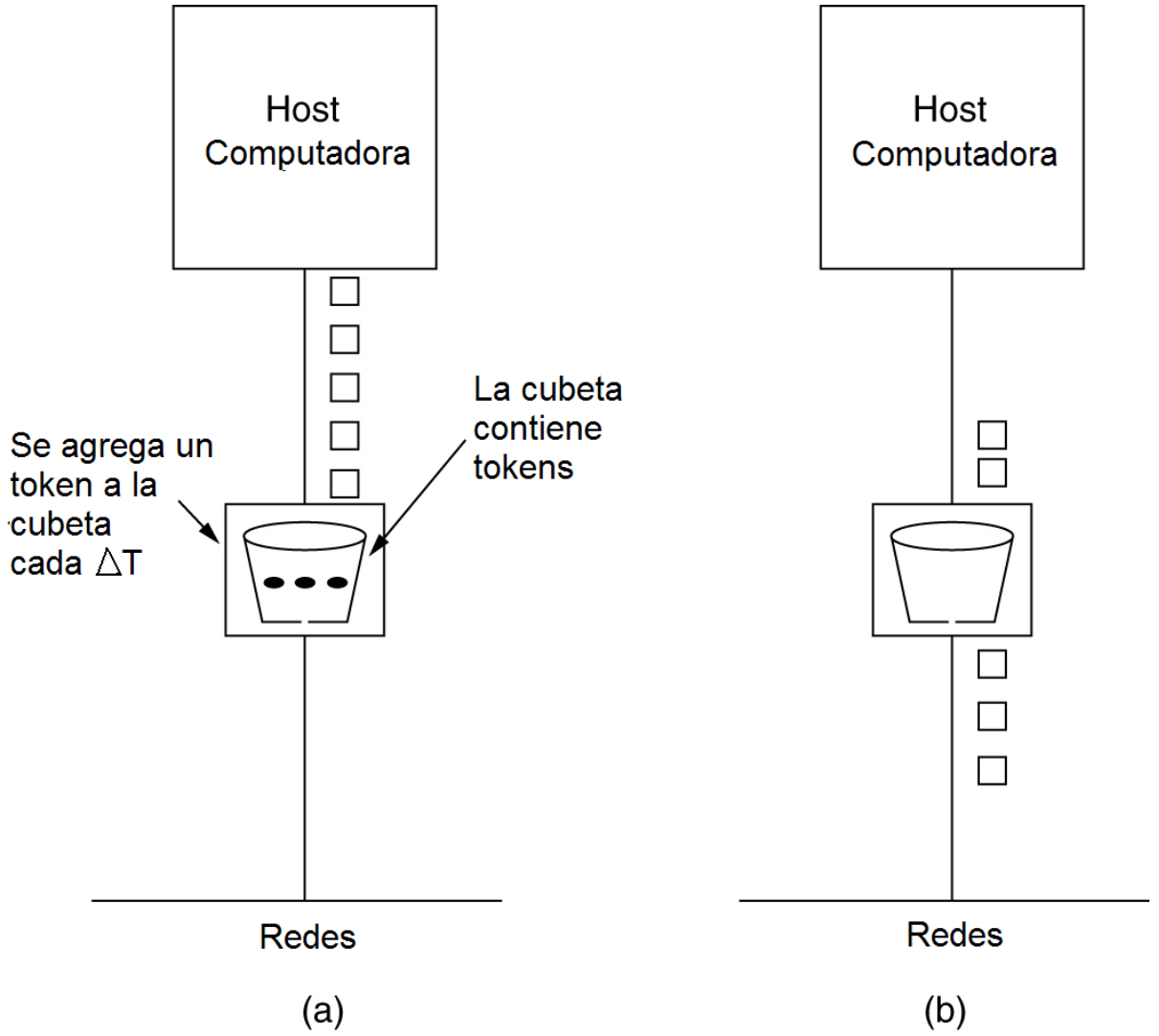
5.3.5 Algoritmos cubeta con tokens

- El algoritmo de cubeta con goteo, impone un patrón de salida rígido a la tasa promedio, sin importar la cantidad de ráfagas que tenga el tráfico

- En muchas aplicaciones es mejor permitir que la salida se acelere un poco, o sea un algoritmo flexible, en dependencia de los tamaños de ráfaga

- Este **algoritmo se denomina de cubeta con tokens**. En este algoritmo la cubeta con goteo contiene tokens, generados por un reloj a razón de un token cada ΔT seg. Para que se transmita un paquete, éste debe capturar y destruir un token

5.3.5 Algoritmos cubeta con tokens



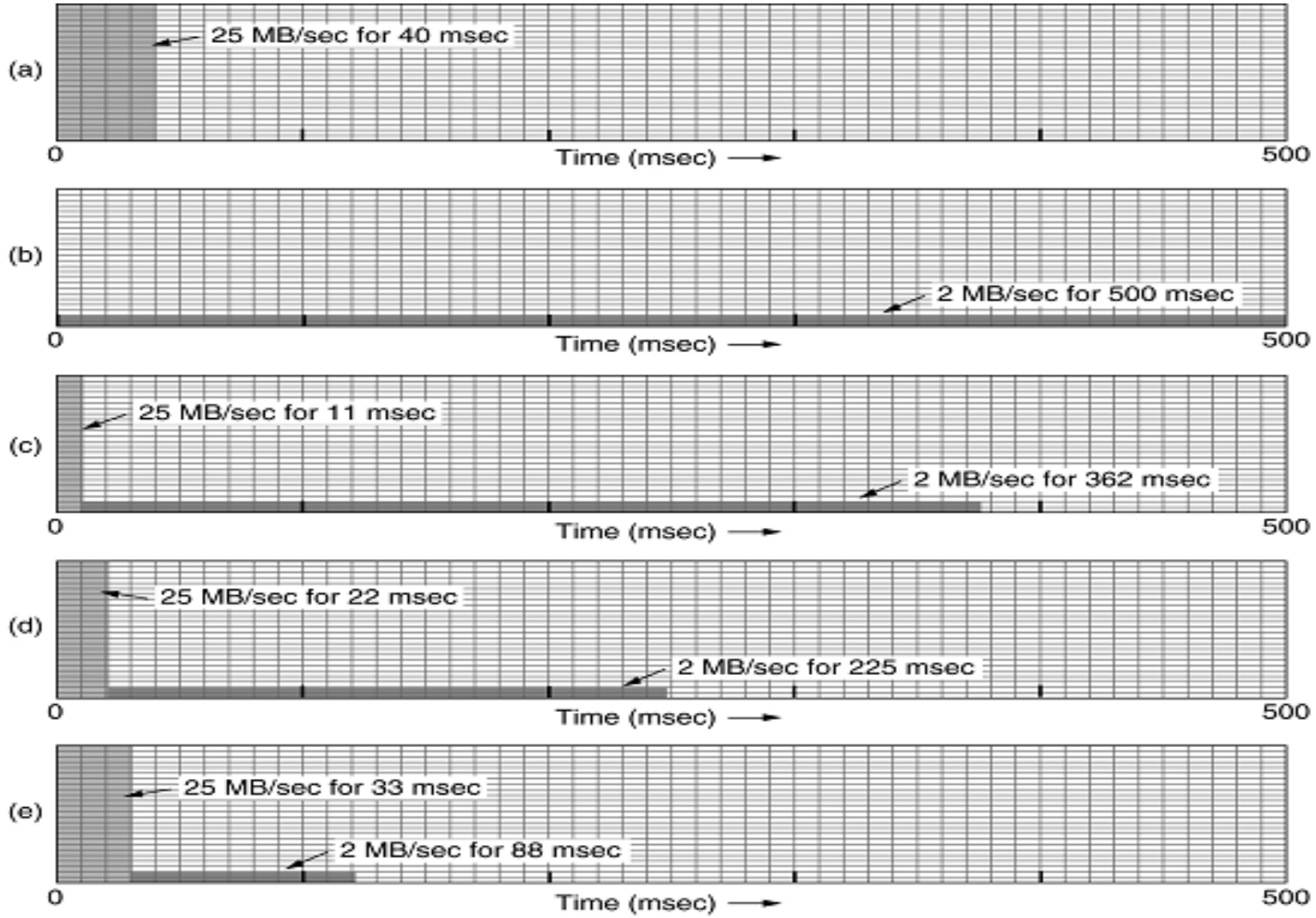
Algoritmo de cubeta con tokens (a) Antes. (b) Después

5.3.5 Algoritmos cubeta con tokens

• La cubeta con tokens permite ráfagas, pero limitadas a una longitud máxima regulada para evitar que se pierdan datos por sobrepasar la capacidad de la cubeta. Cuando se establece una conexión, el usuario y la empresa portadora acuerdan un cierto patrón de tráfico para ese circuito, esto se llama **acuerdo de nivel de servicio**

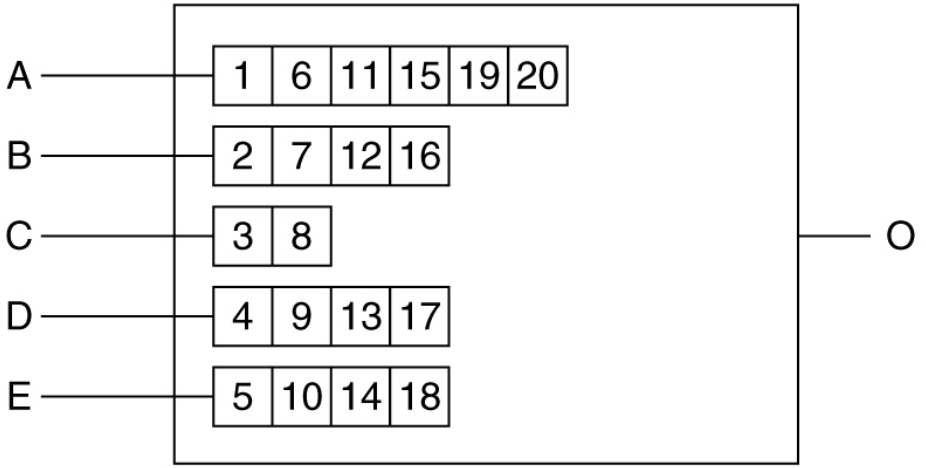
• Se puede regular el desague, o velocidad de drenado, metiendo más o menos tokens por segundo y adaptarse a ráfagas de tráfico.

Ejemplo Algoritmos cubeta con goteo y con tokens



(a) Entrada a una C.G. con ρ=2 MB/seg (b) Salida de cubeta con goteo. Salidas de una cubeta con tokens de (c) 250 KB, (d) 500 KB, (e) 750 KB

5.3.6 Calendarización de paquetes



(a)

Paquete	Tiempo terminación
C	8
B	16
D	17
E	18
A	20

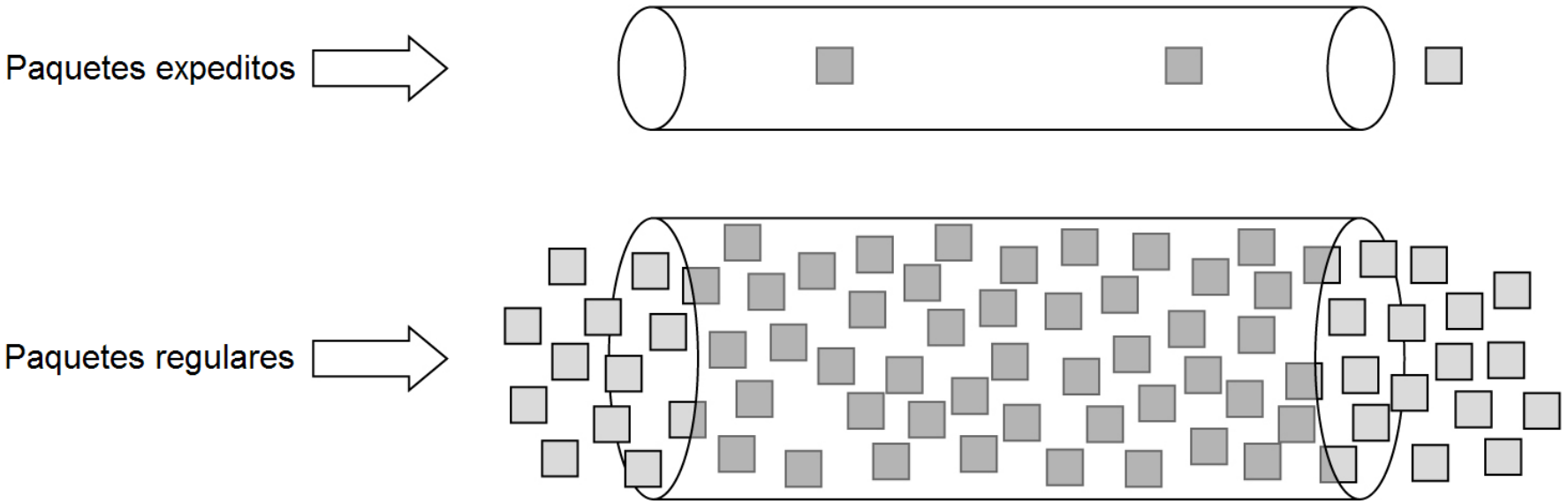
(b)

(a) Un enrutador con cinco paquetes encolados en la línea O. Long. Paquete en bytes
(b) Tiempos de terminación de los cinco paquetes.

Si el enrutador maneja múltiples flujos, existe el peligro de que un flujo acapare mucha de su capacidad y limite a los otros flujos. La esencia del algoritmo es que los enrutadores tienen colas separadas, para cada línea de salida. El enrutador explora circularmente las colas byte a byte, no paquete a paquete hasta que encuentra el instante en que finalizará cada paquete. En la figura el primer paquete en terminar es C después de 8 pulsos (virtuales) de reloj. Se ordenan conforme a su tiempo y se envían

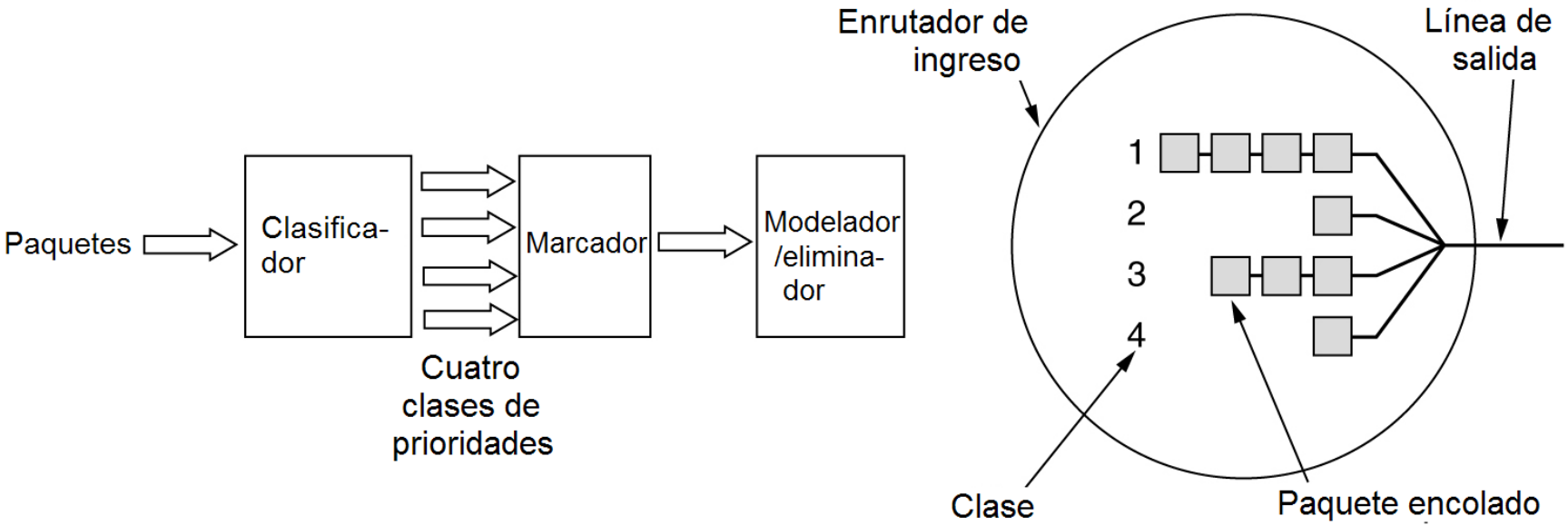
- Es un modelo de servicios de red que clasifica los servicios en un número fijo de clases.
- Paquetes de distintas clases se tratan (almacenamiento y envío) de forma distinta por los routers.
- Servicios Diferenciados surge como una alternativa frente a la dificultad de implementar Servicios Integrados.
- El modelo define las clases de servicios que pueden existir.
  - Servicio Premium: aplicaciones que requieren bajo retardo y bajo fluctuación en la llegada de los paquetes.
  - Servicio Seguro: aplicaciones que requieren mayor confiabilidad que "best effort".
  - Servicio Olímpico: provee tres sub-clases: Oro, plata y bronce, en orden decreciente de calidad.
- Para implementarlo, los paquetes identifican un tipo de servicio. De acuerdo a su contenido los paquetes son clasificados y enviados de manera distinta.
- Los dos técnicas que estudiaremos son:
  - Reenvío expedito o acelerado
  - Reenvío asegurado.

5.4.1 Reenvío expedito o acelerado



En muchas situaciones, es necesario dar a los servidores de video, o algún tipo de servicio premium alguna prioridad de envío. La idea detrás del servicio expedito descrito en el RFP 2598 es muy simple. Dos clases de servicios están disponibles: **regular y expedito**. Los paquetes expedito deben tener la capacidad de transitar la subred como si no hubieran otros paquetes.

5.4.2 Reenvío asegurado

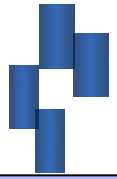


Una posible implementación del flujo de datos para el reenvío asegurado

Un esquema un poco más elaborado se conoce como **reenvío asegurado** y se describe en RFC-2597. Especifica que deberán haber cuatro clases de prioridad y cada una tendrá sus propios recursos.

- El paso 1 es clasificar los paquetes en cuatro clases de prioridades
- El paso 2 es marcar los paquetes de acuerdo con su clase, utilizando un campo llamado Tipo de Servicio del encabezado de un paquete IP.
- El paso 3 es pasar los paquetes a través de un filtro modulador/eliminador que podría retardar o descartar alguno mediante cubeta con goteo o con tokens para dar una forma aceptable a los cuatro flujos.

-Estos 3 pasos se realizan en el emisor y se inyectan en el enrutador



Tema 2: La Capa de Red

1. Introducción
2. Conceptos y Técnicas de conmutación
3. Enrutamiento
4. Control de tráfico y de congestión
5. Calidad del Servicio
6. Bibliografía

- A. S. Tanenbaum. Redes de Computadoras, 4ª Edición. Prentice-Hall, 2003.
- W. R. Stallings. Comunicaciones y Redes de Computadoras, 7ª Edición. Prentice-Hall, 2004.
- W.R. Stevens. TCP/IP Illustrated, Vol.1 The Protocols, Ed. Addison Wesley, 2000.