Conmutación de paquetes

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Arquitectura de Redes, Sistemas y Servicios
3º Ingeniería de Telecomunicación

Basadas parcialmente en el material docente de Lawrie Brown sobre el libro de William Stallings (Data and Computer Communications)

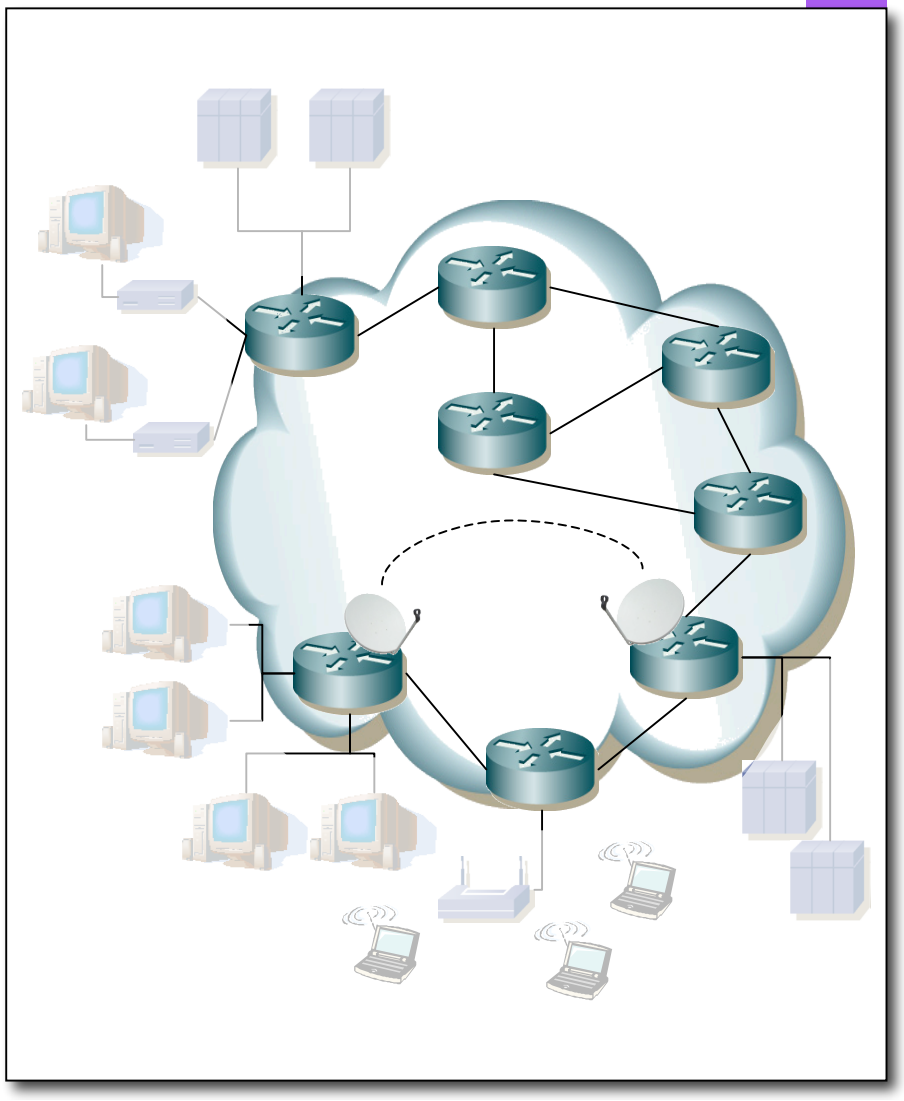
Temario

1. Introducción
2. Arquitecturas, protocolos y estándares
3. **Conmutación de paquetes**
4. Conmutación de circuitos
5. Tecnologías
6. Control de acceso al medio en redes de área local
7. Servicios de Internet

Paradigmas de conmutación

Núcleo de la red

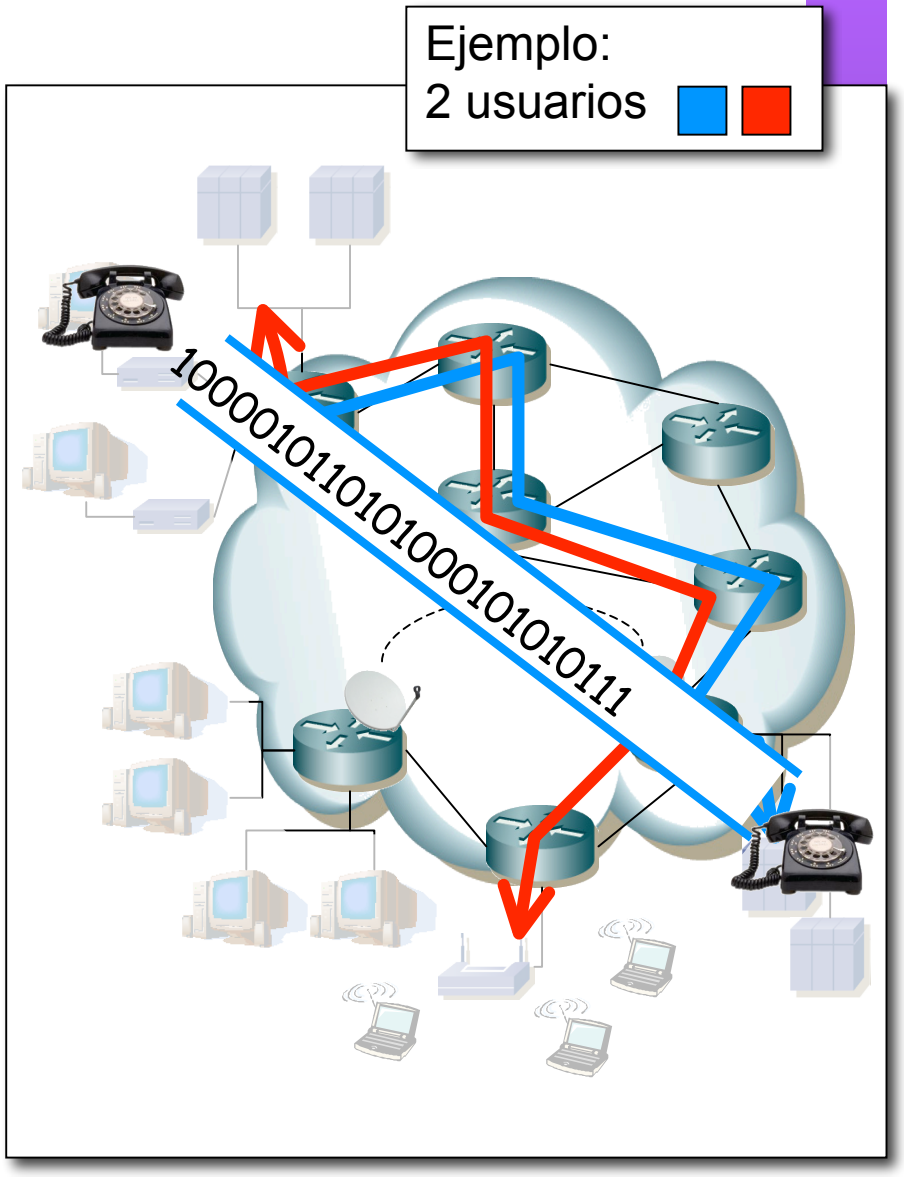
- Interconexión de conmutadores
- *¿Cómo se transfieren los datos por la red?*
  - **Conmutación de circuitos**
  - **Conmutación de paquetes**



Núcleo de la red

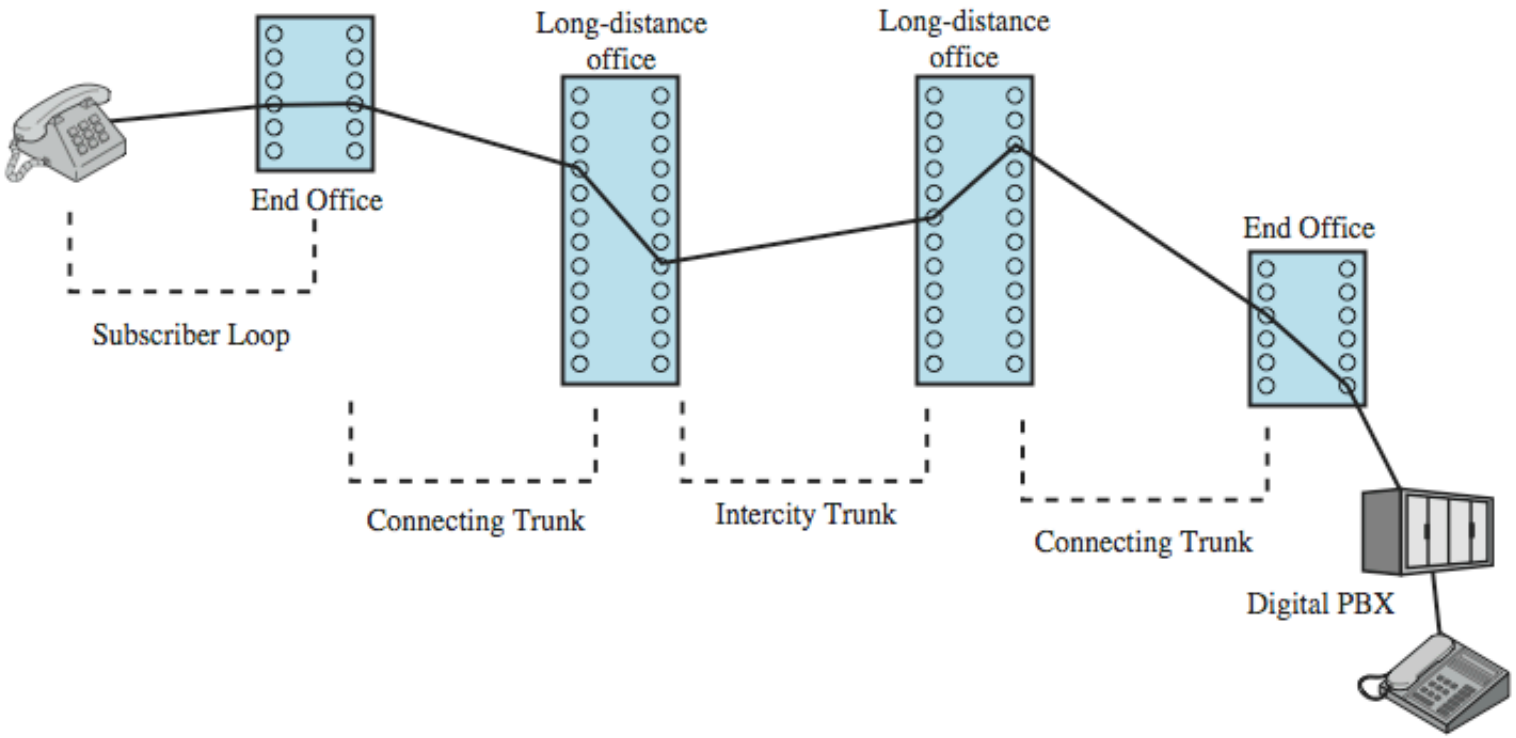
Conmutación de circuitos

- Tres fases: Establecimiento, Transferencia y Desconexión
- RTT en el establecimiento (…)
- Comunicación transparente (...)
- Reserva de recursos:
  - Recursos “extremo-a-extremo”
  - Ancho de banda, capacidad en los conmutadores
  - Recursos (camino) dedicados: no se comparten aunque no se usen
  - Garantías de calidad
- Ineficiente
  - Capacidad del canal dedicada durante la vida del “circuito”
  - Si no se envían datos la capacidad se desperdicia



Conmutación de circuitos

Caso típico: red telefónica conmutada

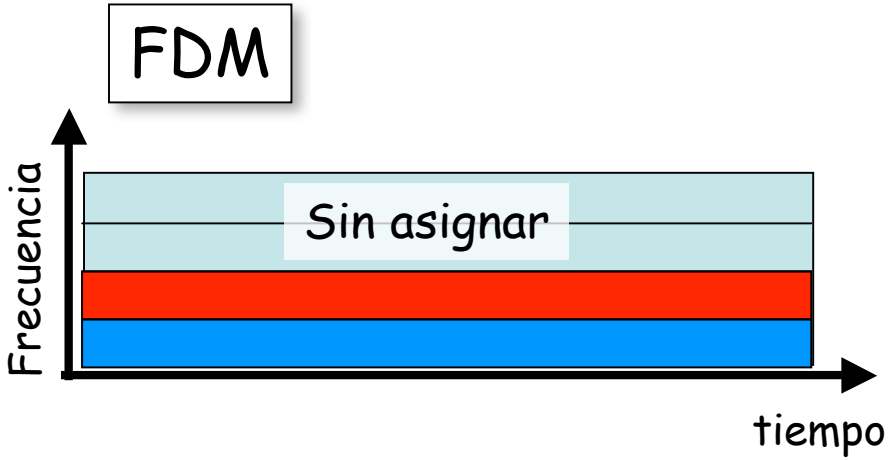


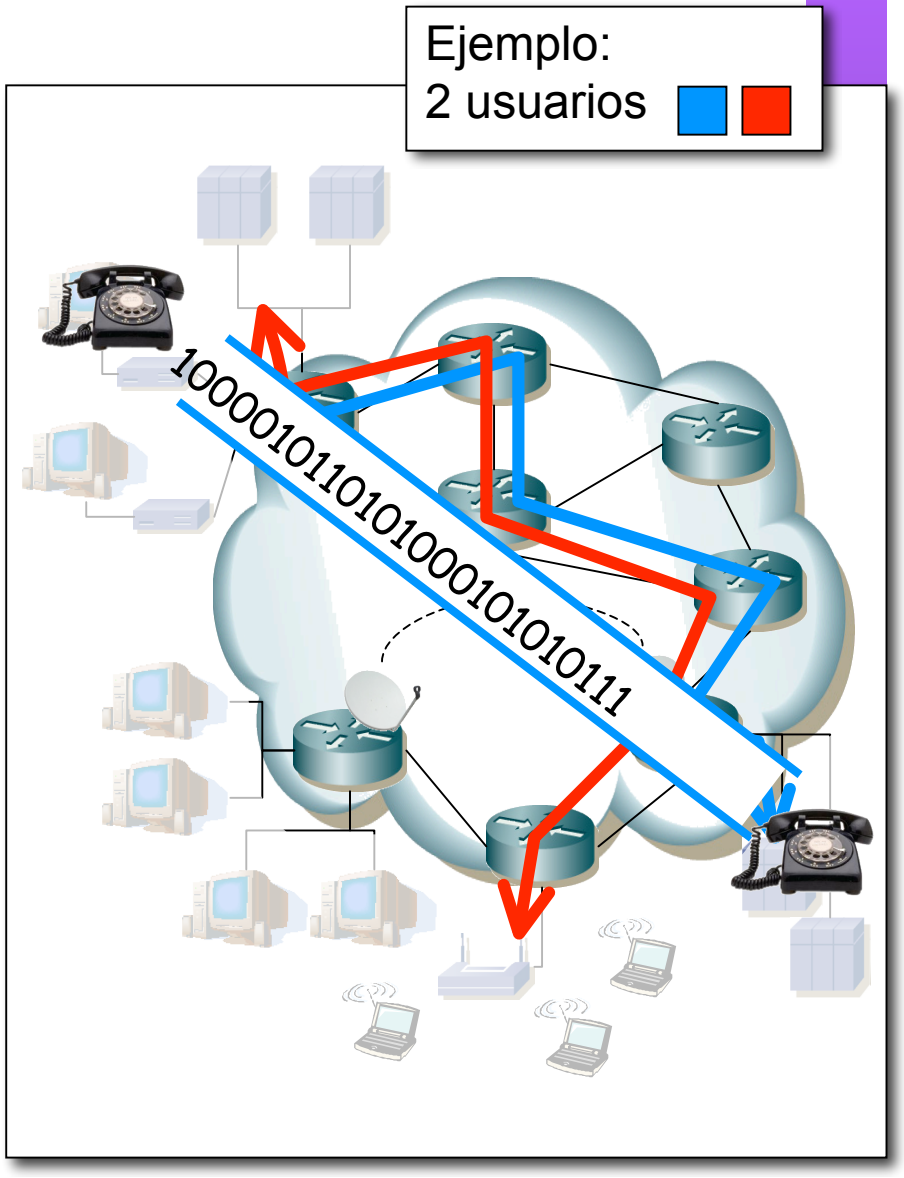
Más en profundidad más adelante

Núcleo de la red

Conmutación de circuitos

- ¿ Cómo emplean el mismo medio de transmisión al mismo tiempo ?
- Técnicas de multiplexación
- Ej:
  - *Frequency Division Multiplexing (FDM) (. . .)*

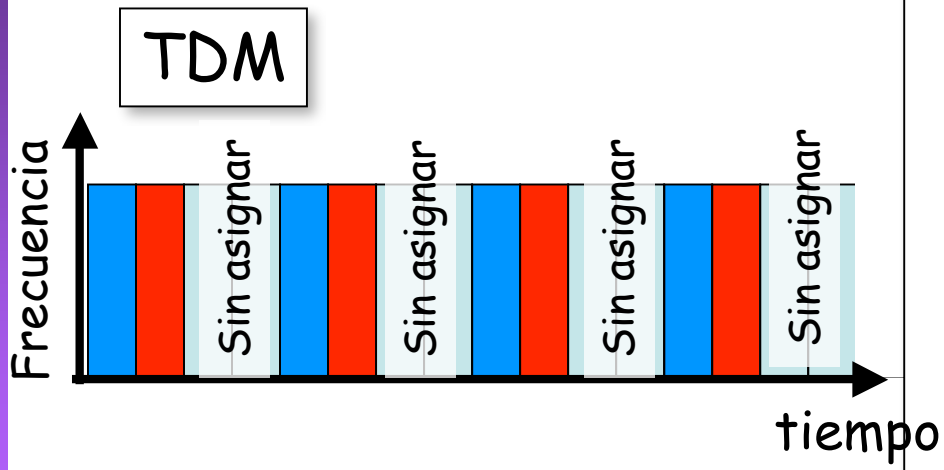


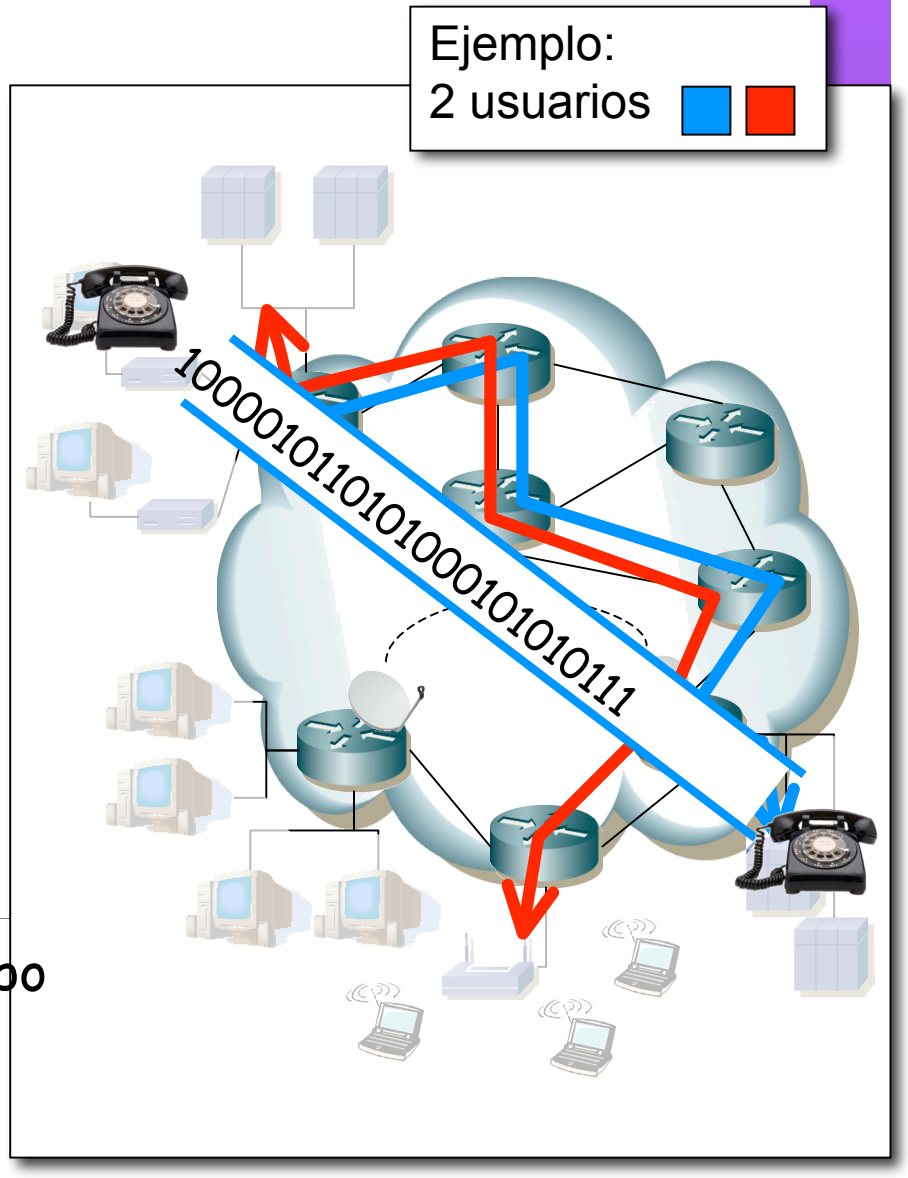


Núcleo de la red

Conmutación de circuitos

- ¿ Cómo emplean el mismo medio de transmisión al mismo tiempo ?
- Técnicas de multiplexación
- Ej:
  - *Time Division Multiplexing (TDM) (. . .)*

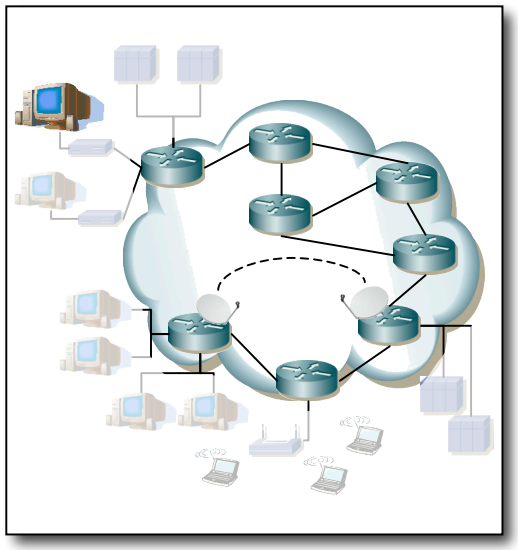


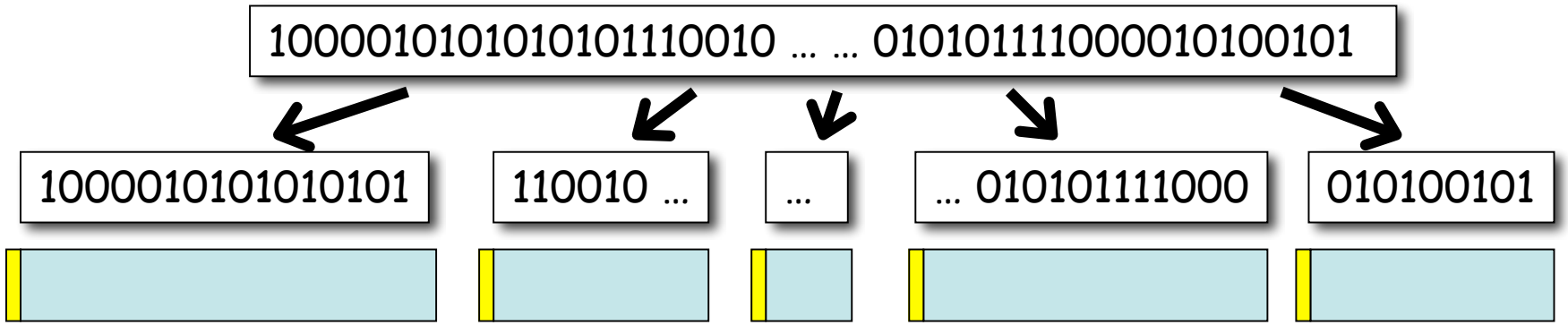


Núcleo de la red

Conmutación de paquetes

- La información se divide en bloques (...)
- Datos + información de control (…)
- Cada paquete contiene información para llegar al destino
- No se reservan recursos

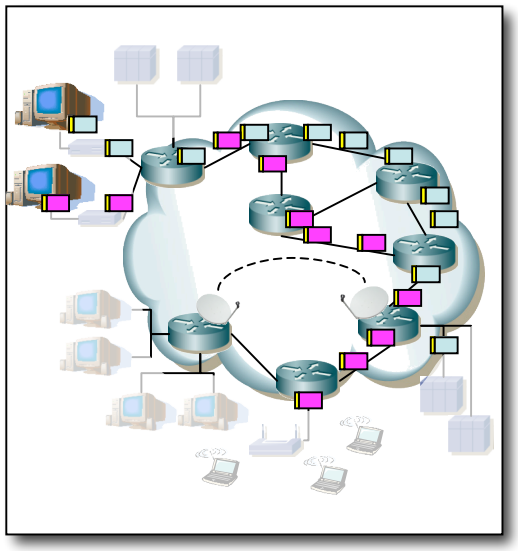


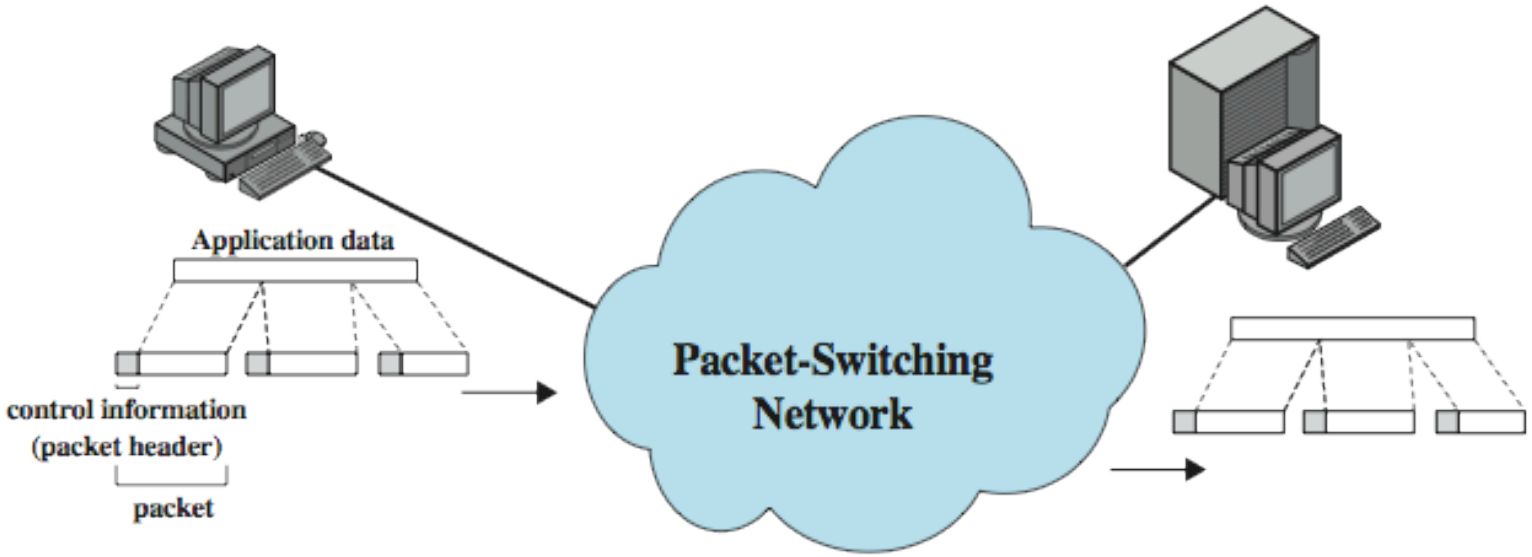


Núcleo de la red

Conmutación de paquetes

- Enlaces compartidos por paquetes de diferentes comunicaciones
- Conversión de velocidad
- *Store-and-forward*
- Cada paquete usa toda la capacidad del enlace…



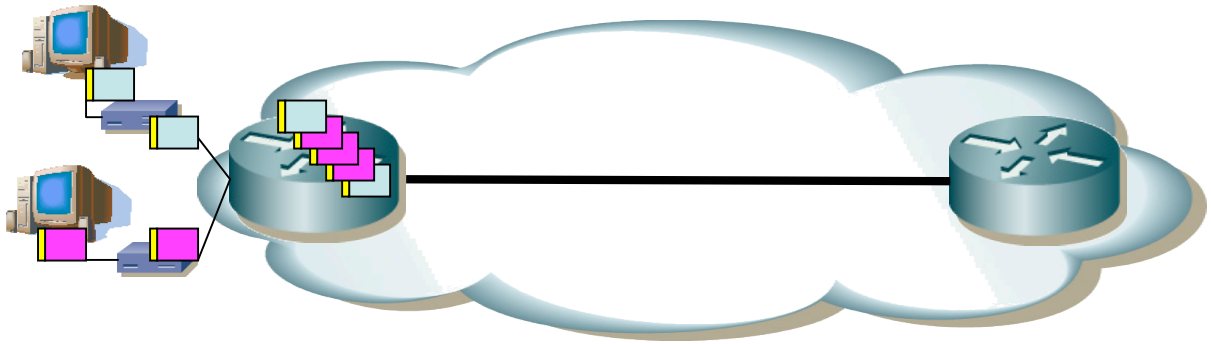


Núcleo de la red

Conmutación de paquetes

- …pero puede tener que esperar a que otros se envíen antes

- Multiplexación estadística
  - Mejor aprovechamiento de recursos
  - Dimensionamiento más complicado



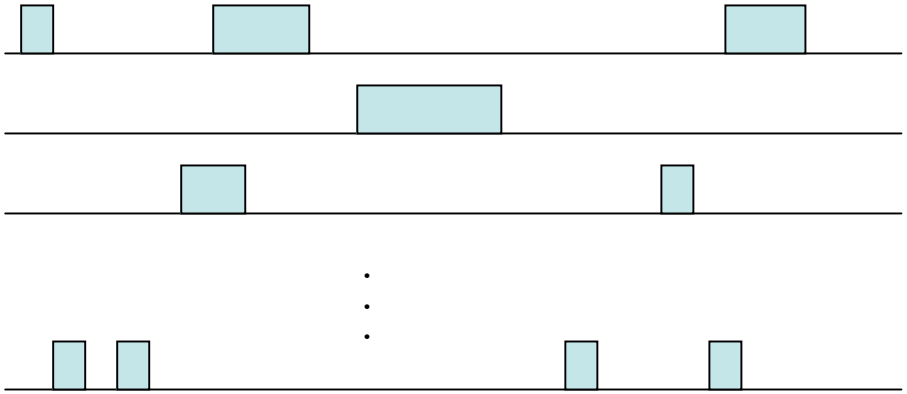
Multiplexación estadística

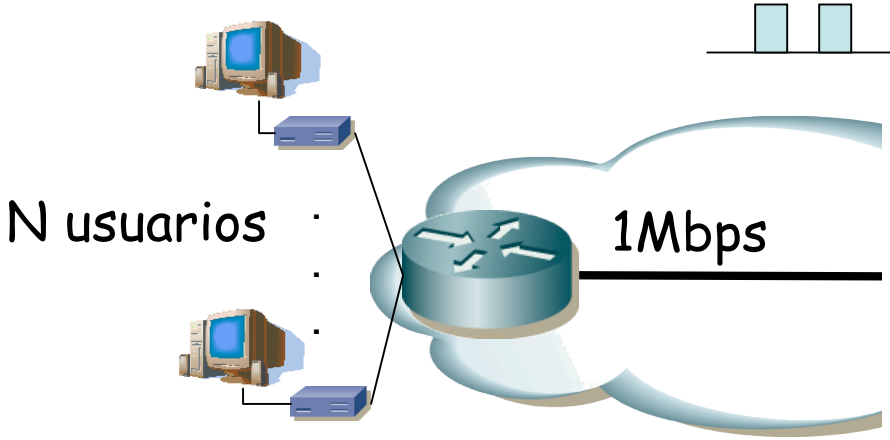
Ejemplo

- Cada usuario:
  - 100 Kbps cuando está activo
  - Activos un 10% del tiempo

- <u>Conm. Circuitos:</u>
  - 10 usuarios

- <u>Conm. Paquetes:</u>
  - Supongamos N=35 usuarios
  - ¿ Cuál es la probabilidad de que más de 10 usuarios transmitan a la vez ? (…)





$$P(>10 activos) < 0.0005$$

Conmutación de paquetes

Circuitos virtuales

- Se establece un camino extremo a extremo
- Los paquetes siguen el camino establecido
- Orientado a conexión

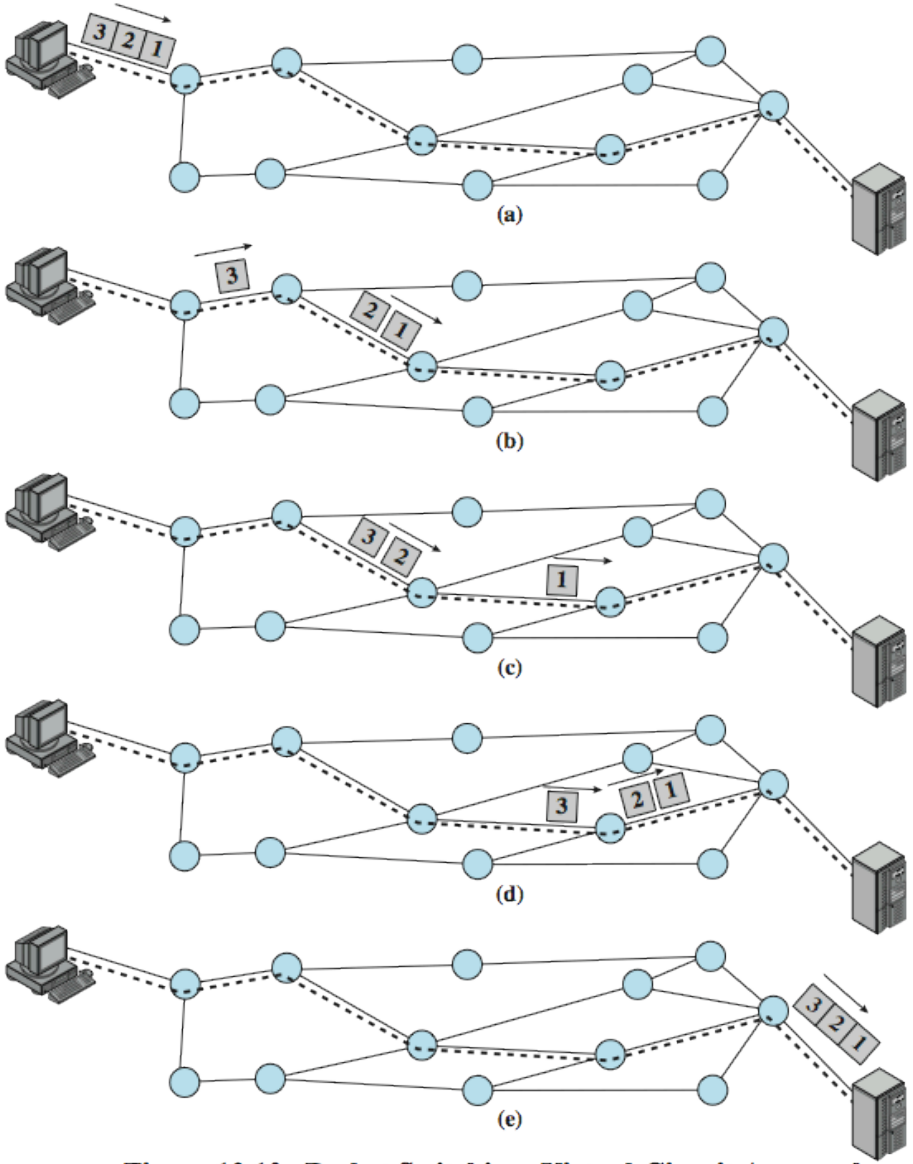


Figure 10.10 Packet Switching: Virtual-Circuit Approach

Conmutación de paquetes

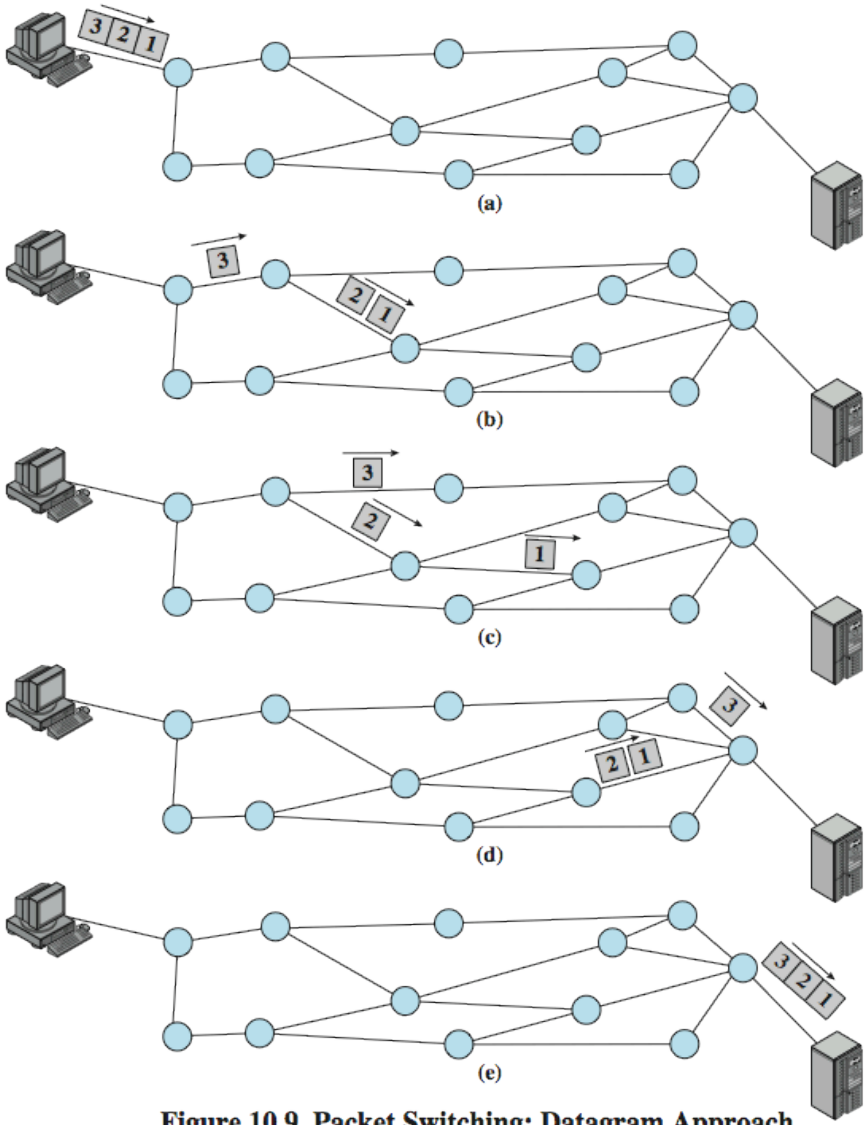


Figure 10.9 Packet Switching: Datagram Approach

Datagramas

- Cada nodo toma la decisión de encaminamiento para cada datagrama
- Sin conexión

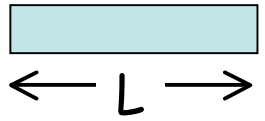
Circuitos virtuales y datagramas

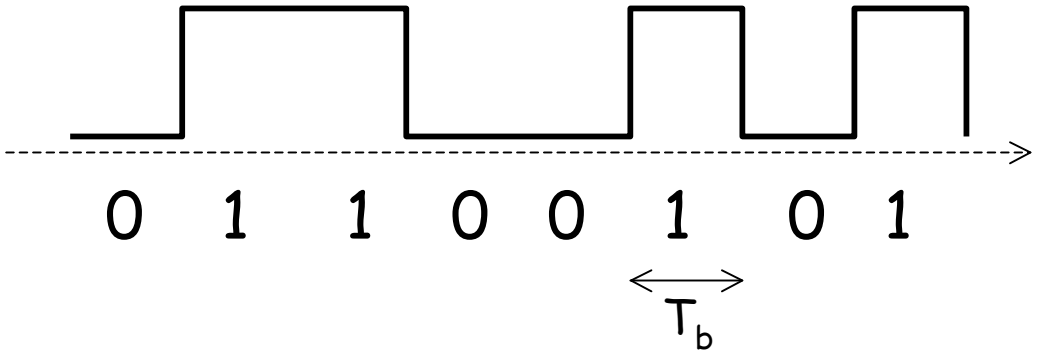
- **Circuitos virtuales**
  - La red puede proporcionar entrega en orden y control de errores
  - Los paquetes se reenvían más rápido (hay que pensar menos por cada paquete)
  - Menos fiabilidad de la red (es más dificil adaptarse a que caiga un enlace)
- **Datagramas**
  - No hay establecimiento de circuito (más rápido)
  - Más flexible
  - Más fiable

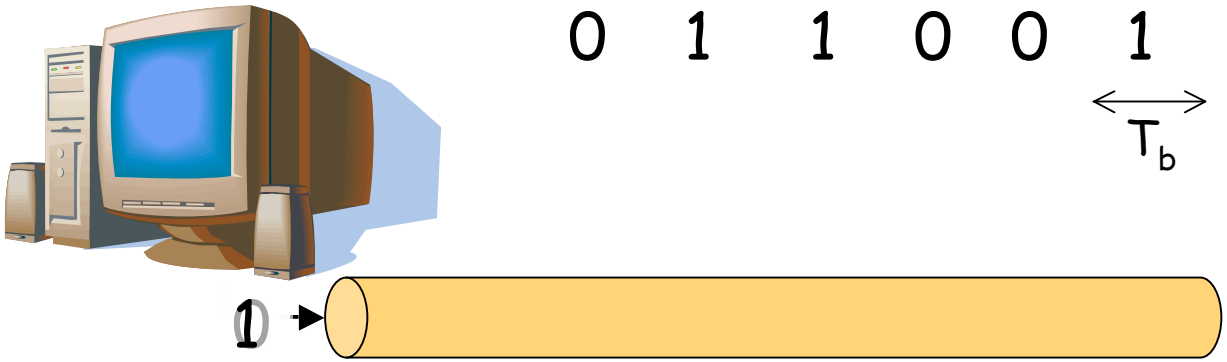
Retardos

Retardo de transmisión

- Tiempo que tarda el transmisor en colocar los bits en el canal
- Bits por segundo (…)
- Ejemplo:
  - Longitud del paquete L = 1.500 Bytes = 12.000 bits
  - Tasa de transmisión R = 57.600bps (T_b=17.36μseg)
  - Tiempo de transmisión = L/R = 12.000 bits / 57.600bps ≈ 208 mseg



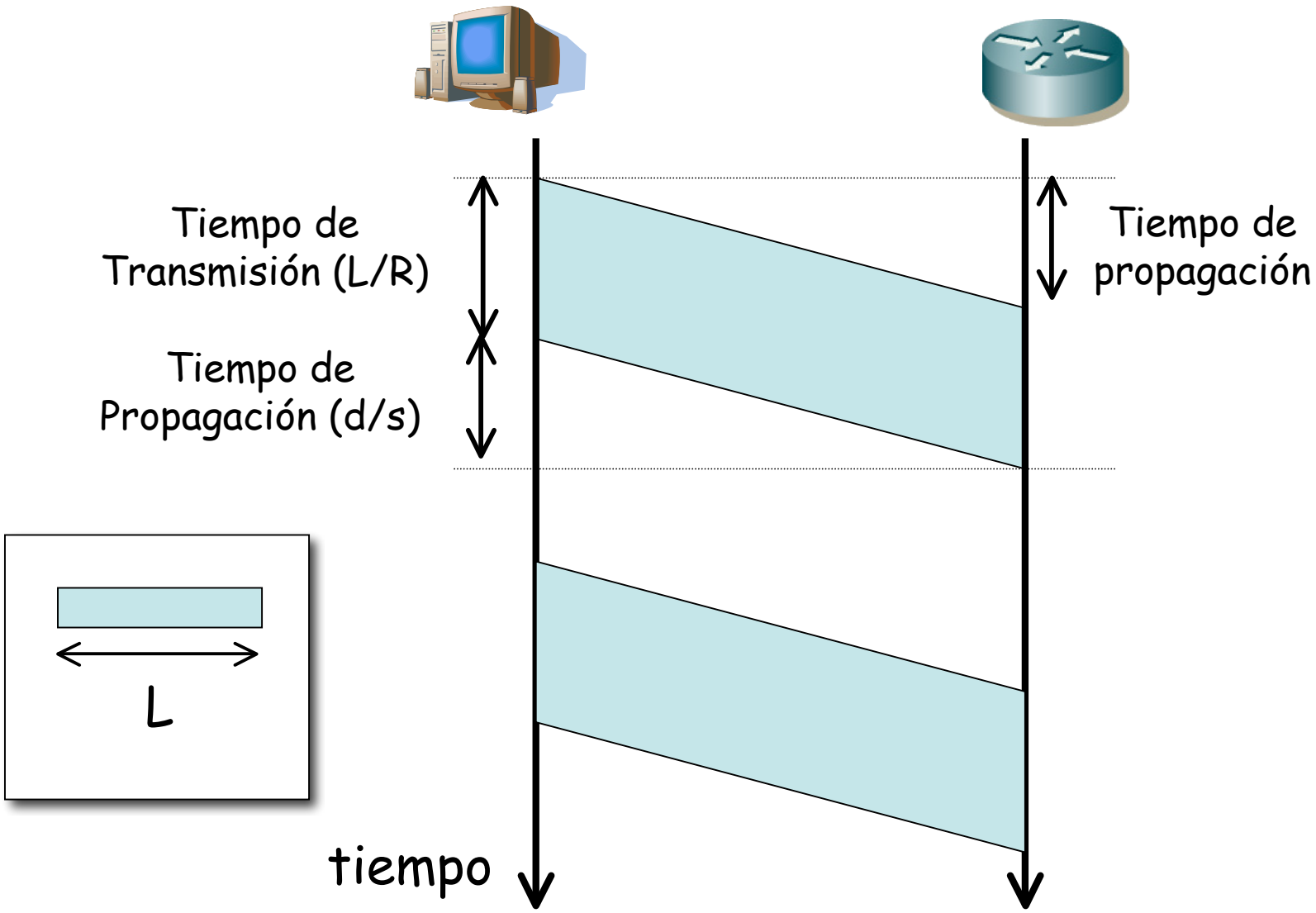




Retardo de propagación

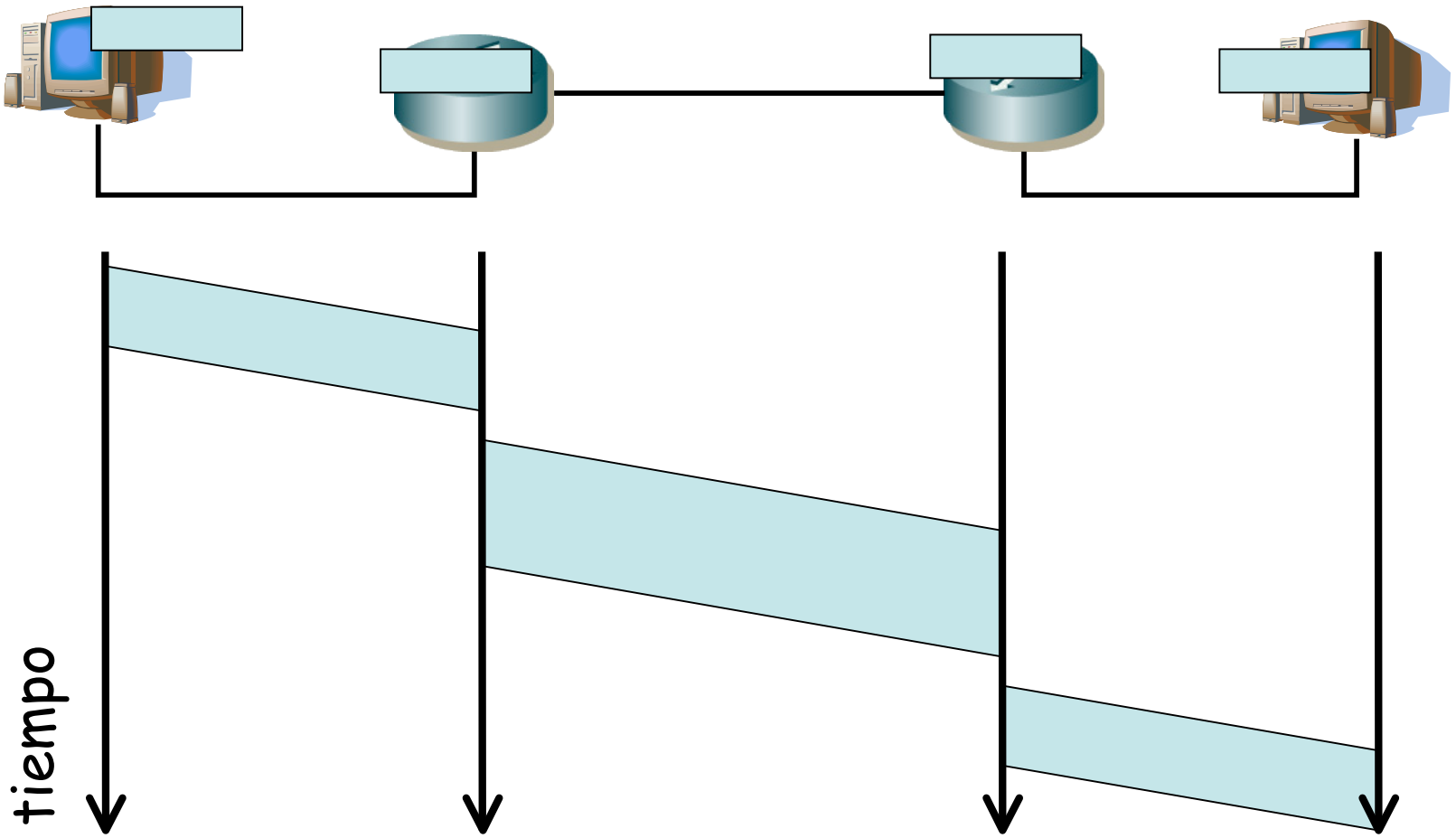
- Tiempo que tarda la señal en llegar al otro extremo del sistema de transmisión (…)
- Ejemplo:
  - Longitud del enlace físico d = 2.000Km
  - Velocidad de propagación en el medio s = 200.000 Km/seg
  - Retardo de propagación = d/s = $2x10^6$ m / ($2x10^8$ m/seg) = 10 mseg
- La velocidad de transmisión y la velocidad de propagación son conceptos muy diferentes

Retardos de transmisión y propagación



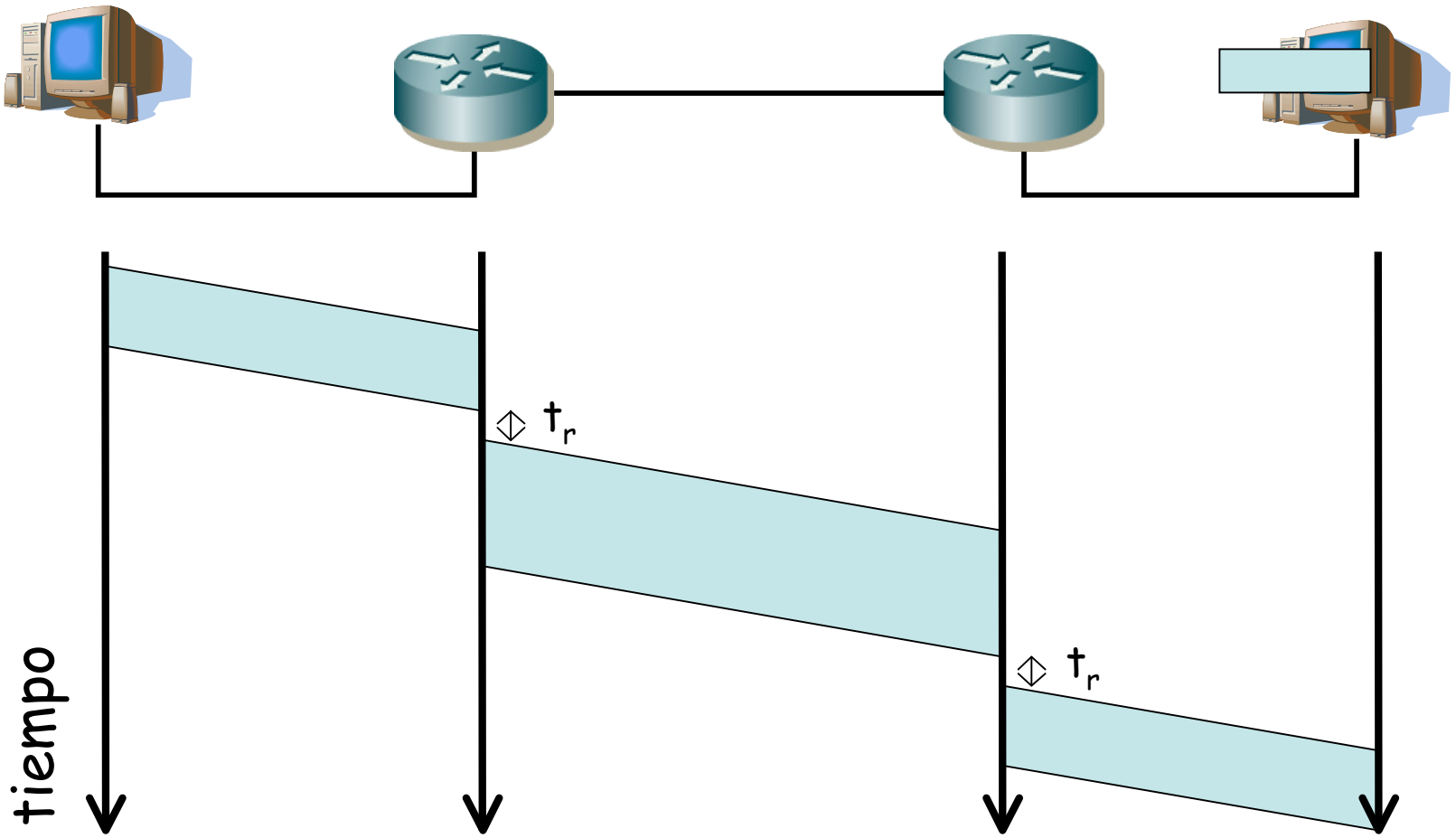
Store-and-forward

- El paquete completo debe llegar al conmutador de paquetes antes de que lo pueda retransmitir (. . .)



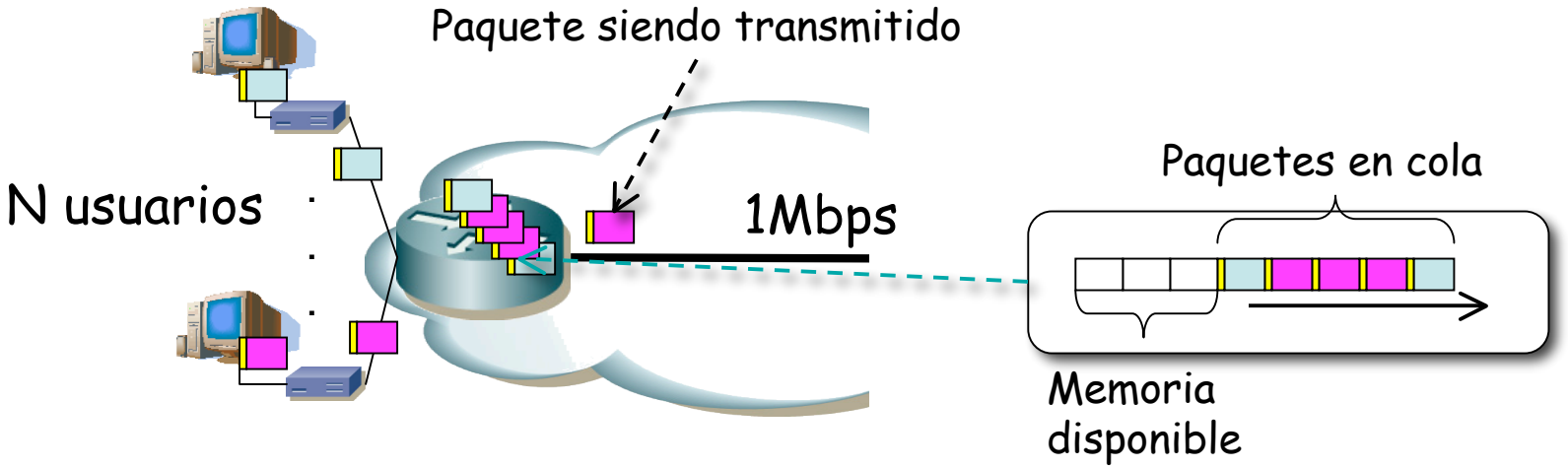
Tiempo de procesado

- El conmutador debe tomar una decisión para cada paquete, la cual lleva tiempo (t_r)



Retardo en cola

- Los paquetes pueden llegar al router a una velocidad mayor que la capacidad del enlace de salida
- El router los almacena en memoria hasta poder enviarlos
- Esperan en una *cola*
- Si no queda espacio en memoria para almacenar un paquete, normalmente éste se pierde (*drop-tail policy*)



Retardo en cola

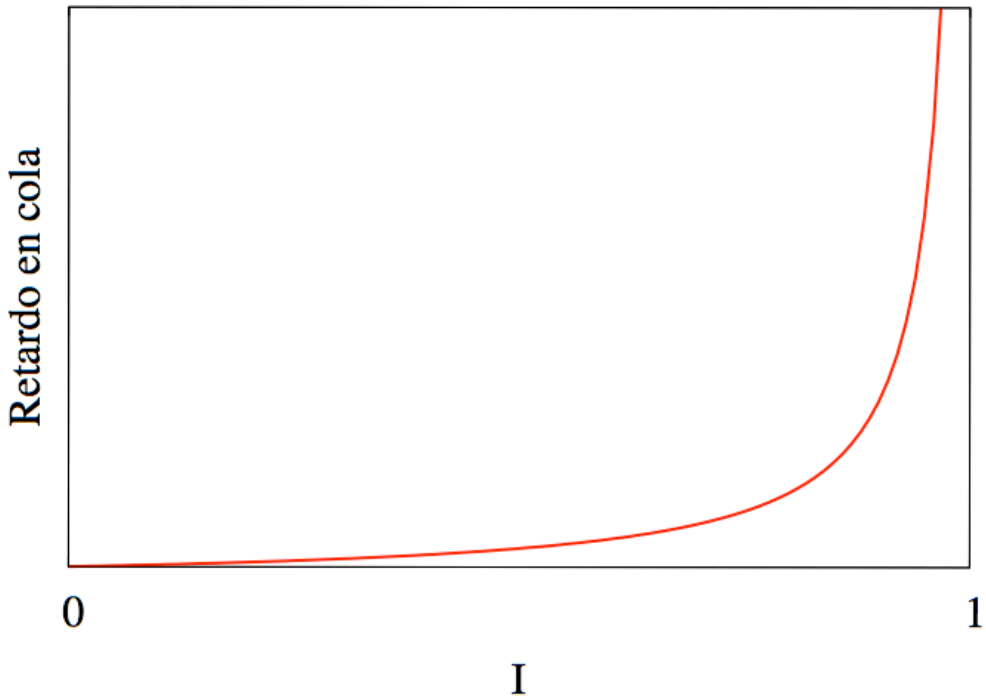
- R = tasa de transmisión
- L = longitud del paquete
- λ = tasa media de llegadas por segundo

- Llegan λ paquetes por segundo
- Llegan λL bps

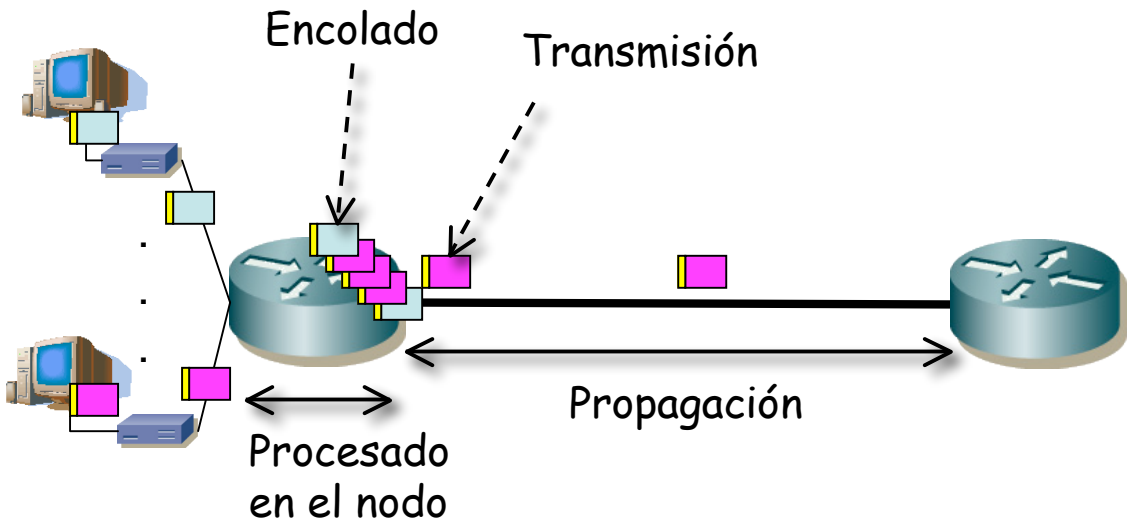
Intensidad del tráfico:

$$I = \frac{\lambda L}{R}$$

- ¿ I > 1 ?
- ¿ Llegadas periódicas ?
- ¿ Llegadas en ráfagas ?



Retardos



$$d_{\text{nodo}} = d_{\text{proc}} + d_{\text{cola}} + d_{\text{trans}} + d_{\text{prop}}$$

d_{proc} = tiempo de procesado
- Unos microsegundos

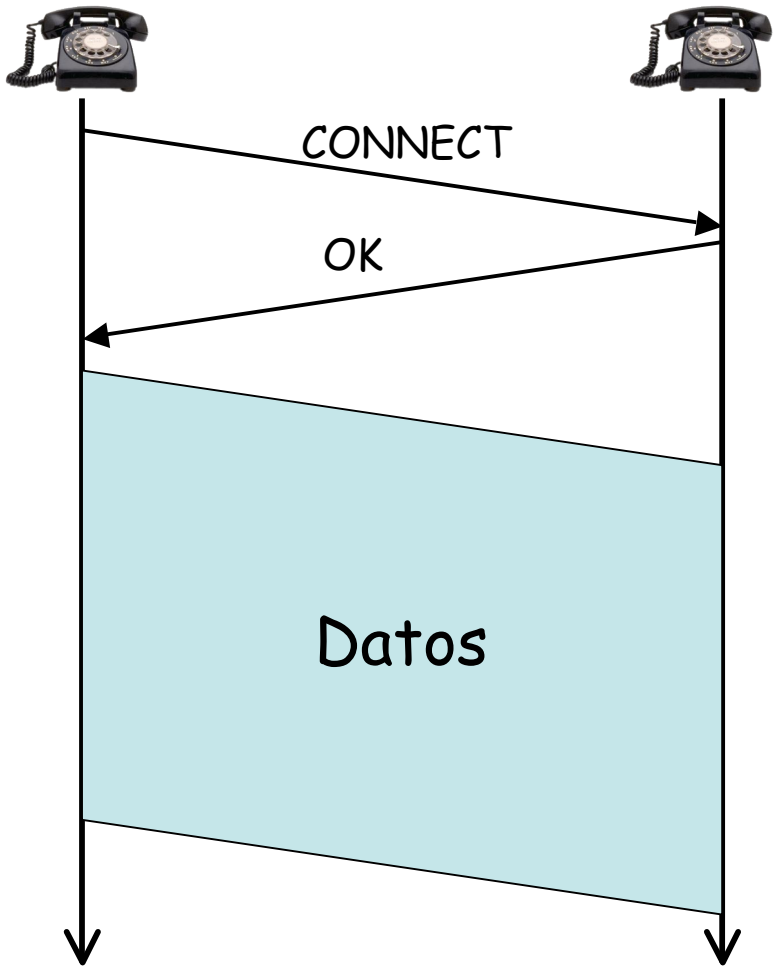
d_{cola} = retardo en cola
- Depende de la congestión

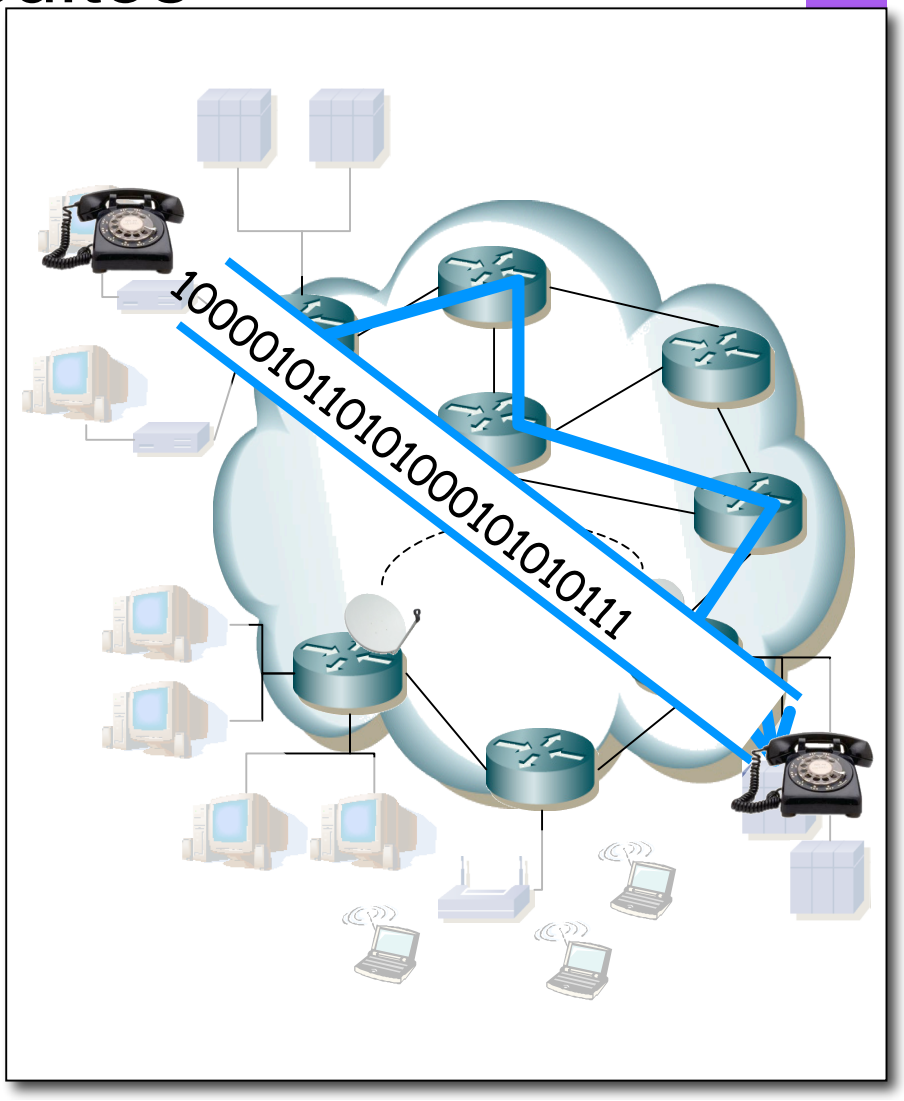
d_{trans} = retardo transmisión
- = L/R, significativo en enlaces de baja velocidad

d_{prop} = retardo propagación
- De unos microseg a centenares de mseg

Ejemplo

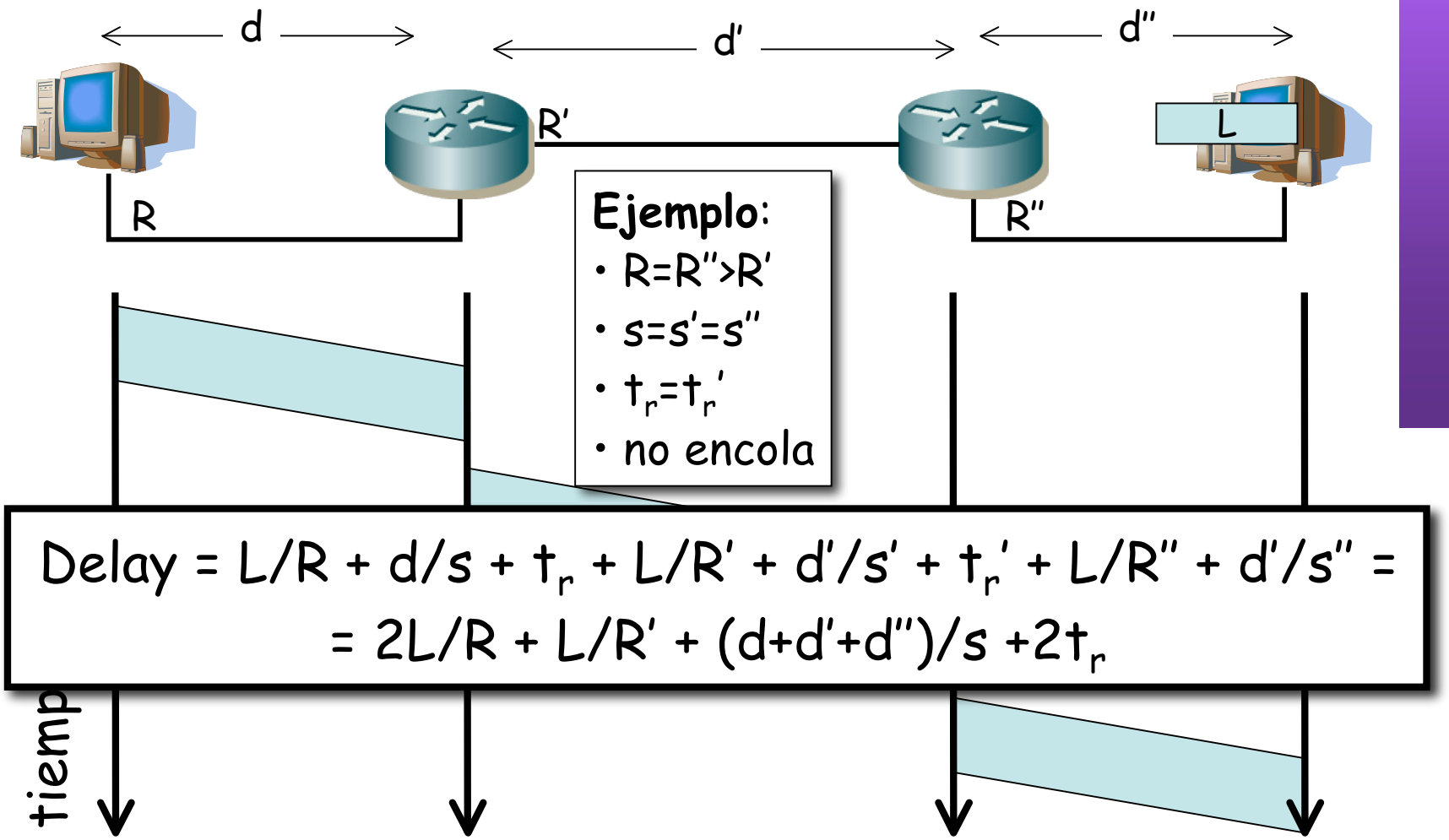
- Conmutación de circuitos





Retardos

- Conmutación de paquetes



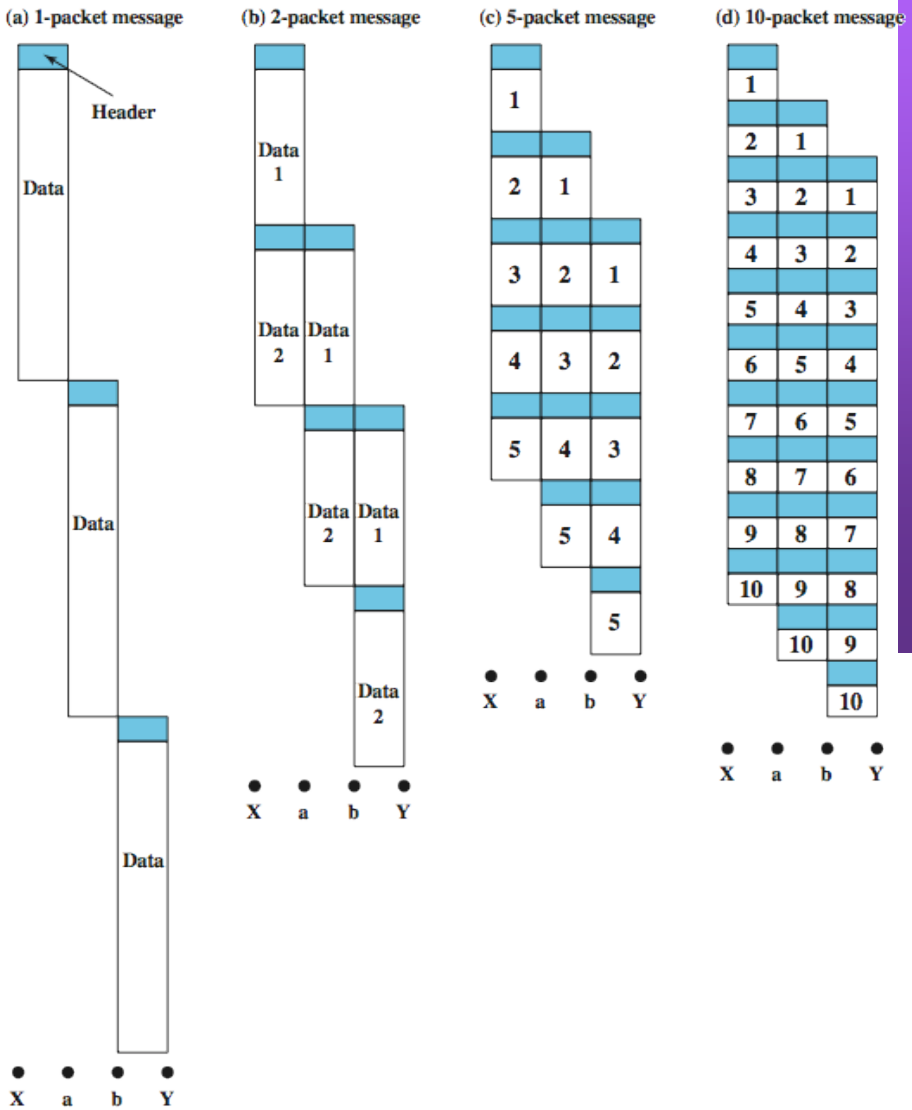
Tamaño del paquete

Mayor tamaño:

- Menos cabeceras, más eficiencia

Menor tamaño:

- Menos tiempo a esperar por *store and forward*



Retardos (Ejemplo)

daniel% traceroute www.berkeley.edu

traceroute to arachne.berkeley.edu (169.229.131.109), 30 hops max, 40 byte packets

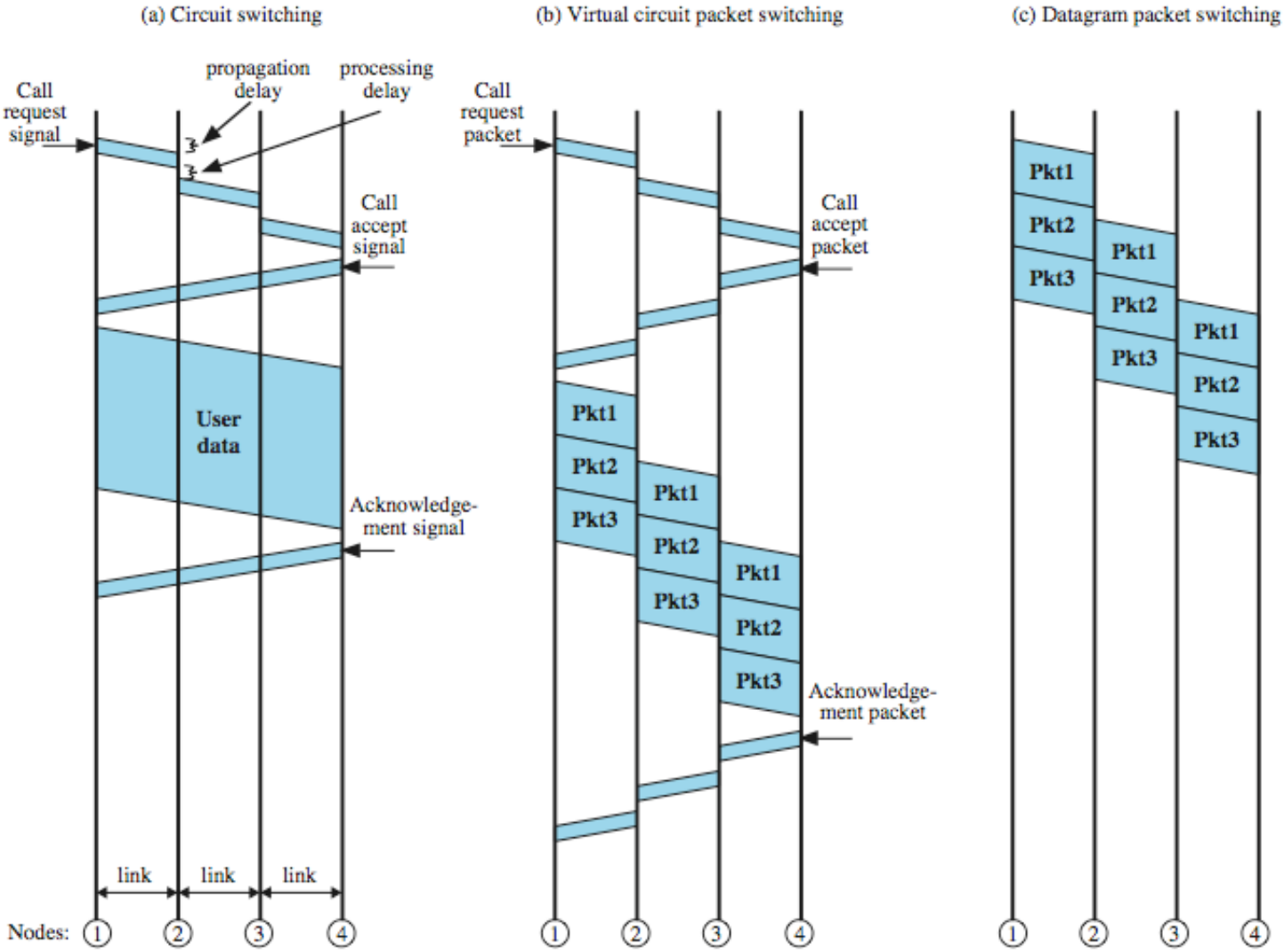
1 arce-un.red.unavarra.es (130.206.160.1) 1.691 ms 0.438 ms 0.417 ms
2 ss-in (130.206.158.25) 1.015 ms 3.091 ms 0.658 ms
3 unavarra-router.red.unavarra.es (130.206.158.1) 1.587 ms 1.87 ms 1.506 ms
4 fe0-1-2.eb-pamplona0.red.rediris.es (130.206.209.13) 1.49 ms 1.741 ms 1.25 ms
5 nav.so2-3-0.eb-bilbao0.red.rediris.es (130.206.240.61) 5.279 ms 4.402 ms 4.398 ms
6 pav.so2-0-0.eb-iris2.red.rediris.es (130.206.240.29) 50.039 ms 16.511 ms 16.35 ms
7 so0-0-0.eb-iris4.red.rediris.es (130.206.240.2) 16.341 ms 17.982 ms 16.405 ms
8 rediris.es1.es.geant.net (62.40.103.61) 118.998 ms 16.741 ms 16.755 ms
9 es.it1.it.geant.net (62.40.96.186) 96.679 ms 39.288 ms 39.513 ms
10 it.de2.de.geant.net (62.40.96.61) 91.118 ms 48.088 ms 49.83 ms
11 abilene-gw.de2.de.geant.net (62.40.103.254) 141.935 ms 141.812 ms 141.716 ms
12 atlang-washng.abilene.ucaid.edu (198.32.8.65) 157.505 ms 157.692 ms 164.648 ms
13 hstnng-atlang.abilene.ucaid.edu (198.32.8.33) 177.182 ms 177.144 ms 177.201 ms
14 losang-hstnng.abilene.ucaid.edu (198.32.8.21) 199.049 ms 198.489 ms *
15 hpr-lax-gsr1--abilene-la-10ge.cenic.net (137.164.25.2) 199.004 ms 198.621 ms 284.873 ms
16 svl-hpr--lax-hpr-10ge.cenic.net (137.164.25.13) 215.55 ms 218.166 ms 206.364 ms
17 hpr-ucb-ge--svl-hpr.cenic.net (137.164.27.134) 210.841 ms 207.409 ms 207.479 ms
18 vlan187.inr-201-eva.berkeley.edu (128.32.0.33) 283.445 ms 207.842 ms 207.318 ms
19 g5-1.inr-210-srb.berkeley.edu (128.32.255.65) 211.052 ms 207.341 ms 207.408 ms
20 arachne.berkeley.edu (169.229.131.109) 207.431 ms 207.451 ms 207.4 ms

¿Circuitos o paquetes?

¿Circuitos o Paquetes?

- Las prestaciones dependen de varios factores
  - Tiempo de propagación
  - Tiempo de transmisión
  - Tiempo de proceso del nodo
- Muchas otras características:
  - Transparencia
  - Cantidad de *overhead*
  - Fiabilidad y robustez
  - Simplicidad de la arquitectura
  - …

Tiempos



Circuitos Circuitos virtuales Paquetes

Implementaciones reales

- Conmutación de circuitos:
  - Red Telefónica Básica (RTB)
  - Red Digital de Servicios Integrados (RDSI)
- Conmutación de paquetes (circuitos virtuales)
  - X.25
  - Frame Relay (conmutación de paquetes asociada a RDSI)
  - ATM
- Conmutación de paquetes (datagramas)
  - IP, IPX, CLNP

Presentación de problemas que veremos

Problemas de redes de circuitos

- **Encaminamiento**
  - Cuando se pide a la red establecer una llamada
  - A partir de la dirección de destino decidir por dónde reservar enlaces desde el origen al destino.
- **Bloqueo**
  - Si en algún punto la llamada necesita recursos no disponibles: no se establecerá y el usuario no recibe servicio.
  - Diseñar las redes de circuitos para que el bloqueo no se produzca o tenga una probabilidad baja

Problemas de redes de paquetes

- **Encaminamiento**
  - Por cada paquete que debe reenviar un nodo debe decidir por qué camino reenviarlo (a qué vecino entregárselo)
- **Bloqueo:** No hay, la red acepta todos los paquetes.

Nuevos problemas:

- **Transporte fiable**
  - ¿Qué pasa si un paquete no se entrega?
- **Control de flujo**
  - ¿Qué pasa si llega un paquete a un destino que está muy ocupado para aceptarlo?
- **Congestión**
  - ¿Qué pasa si la red está aceptando demasiados paquetes y el retardo de entrega crece demasiado?

Transporte fiable

- Los paquetes pueden perderse:
  - Errores físicos de transmisión (checksums y códigos redundantes)
  - Buffers llenos
- ¿Es responsabilidad de la red de paquetes recuperar el paquete? Opciones:
  - Garantizar entrega
  - No garantizar entrega: dejar que recupere las pérdidas otro nivel (extremos de la red)

Transporte fiable

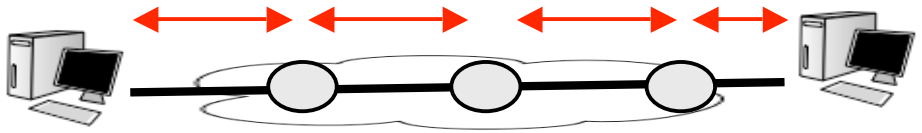
Técnicas para lograr transporte fiable

- Detección de errores:
  - Añadir redundancia para descubrir errores
- Detección de pérdidas:
  - Añadir números de secuencia de los datos para detectar huecos (y de paso desorden)
- Realimentación hacia el emisor
  - Enviar paquetes de confirmación (ACK) o de informe de errores (NACK) para que el origen actúe en consecuencia
- Reenvío
  - El emisor deberá reenviar los paquetes que se hayan perdido o recibido mal
  - Almacenar los paquetes que ha enviado hasta que esté seguro de que han sido recibidos

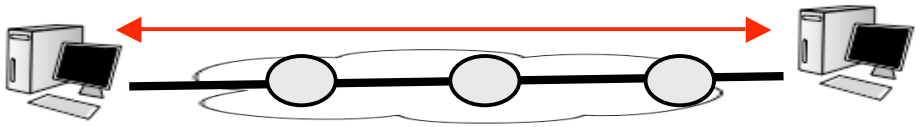
Transporte fiable

¿Quién es el emisor y el receptor?

- En cada salto
  - En cada salto me aseguro de que todo llega hasta el otro lado



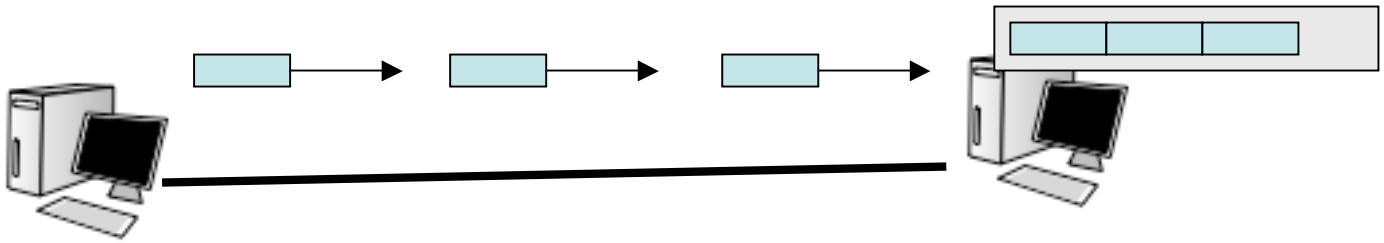
- Extremo a extremo
  - Los extremos inicial y final se aseguran de que todo llega hasta la salida



- Fuera de la red
  - La red no ofrece el servicio de entrega fiable pero los niveles superiores pueden hacerlo
- ¿Cuál es mejor?
- ¿Cuál de estos usa Internet?
- Más adelante en detalle

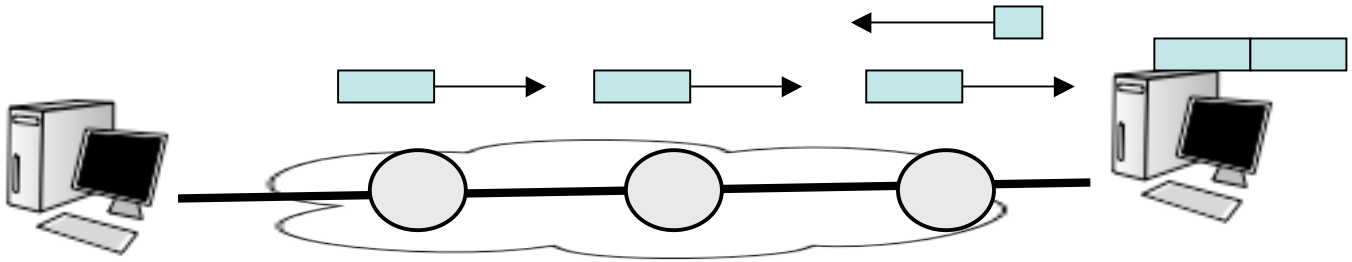
Control de flujo

- En un enlace con un emisor y un receptor
  - El enlace puede ser capaz de aceptar datos a cierta velocidad
  - El receptor puede ser capaz de consumir los datos a menos velocidad
- Control de flujo: mecanismos para que el emisor no sature al receptor
- Realimentación hacia el emisor: listo/nolisto, indicación de espacio, ventanas deslizantes
- Esto es visto en Transmisión de Datos, control de flujo en el nivel de enlace



Control de flujo

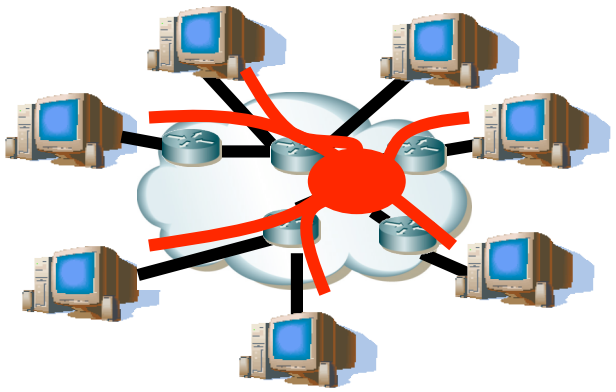
- Y en el nivel de red? Dos aproximaciones
  - Lazo abierto (*Open loop*)
    - Negociación previa con la red de qué es lo que voy a mandar (más típica de circuitos virtuales)
  - Lazo cerrado (*Closed loop*)
    - Realimentación desde el receptor que controle la velocidad de envío del emisor
    - Mismas técnicas que en el nivel de enlace pero hechas extremo a extremo o nodo a nodo.



- Que hace Internet?
  - Tercera opción: no resuelve el problema, que lo resuelva otro nivel

Control de congestión

- Uno de los problemas más difíciles de redes de datos
- La red no es capaz de cursar tanto tráfico
- Sería el equivalente al bloqueo pero como se aceptan todos los paquetes no podemos detectarlo
- En esas condiciones se pierden paquetes
  - Provoca retransmisiones
    - Que generan más tráfico
      - El tiempo de entrega crece y se producen más perdidas lo que provoca más retransmisiones… la eficiencia tiende a cero



- Se resolvería si las fuentes se controlan y no envían demasiado deprisa
- Pero ¿cómo pueden saber que las pérdidas son debidas a congestión?

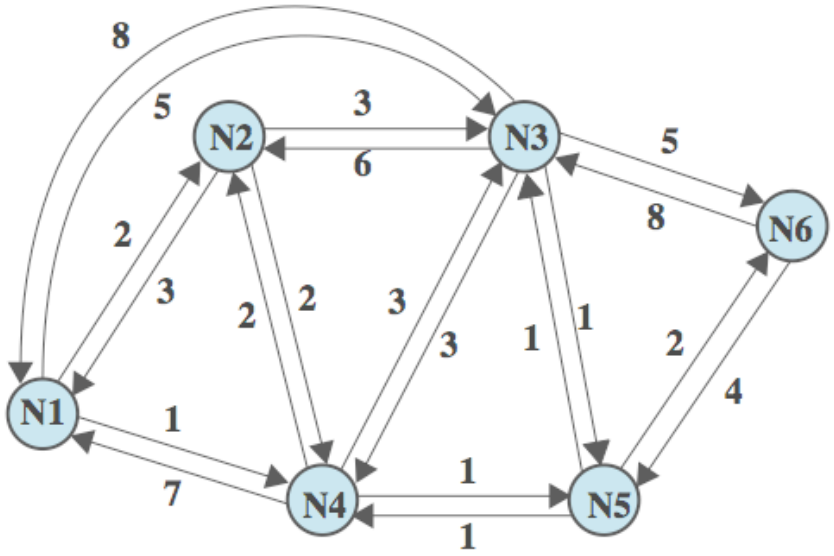
Control de congestión

- Aproximaciones para resolverlo
  - Control de congestión asistido por la red
    - Los nodos que ven congestión envían realimentación hacia los emisores que los están saturando (más propio de circuitos virtuales)
  - Control de congestión extremo a extremo
    - Los emisores intentan estimar si hay congestión y auto regularse para no enviar demasiado

- ¿Qúe hace Internet?
  - Lo deja al nivel de transporte también (TCP trata de auto regularse)

Encaminamiento

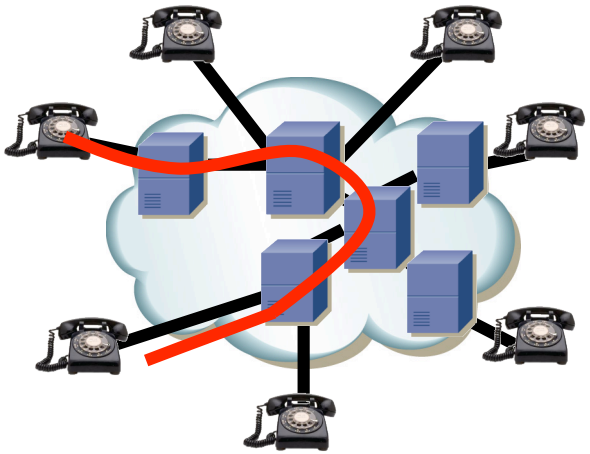
- Encontrar el camino para la información que va al destino D
- Problema matemático de encontrar caminos en un grafo



- ¿Es suficiente con eso? Miremos en más detalle

Encaminamiento

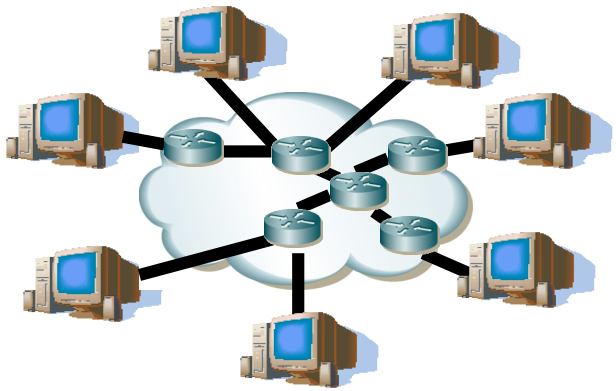
- Llega a la red una llamada (encaminamiento de circuitos)
- Hay que buscar camino para llegar hasta el destino
- Un ente central decide el camino
- Le dice a cada nodo qué recursos debe reservar para la llamada
- El ente central conoce toda la red y puede decidir
- Tiene que tomar tantas decisiones por segundo como llamadas se produzcan
- ¡ No escala !
- Podría ser aceptable para conmutación de circuitos
- El problema es cómo comunicar a cada centralita con el que toma las decisiones
- ¡ Porque es el mismo problema !



Encaminamiento

- En conmutación de datagramas hay una decisión de encaminamiento por cada paquete
- No da tiempo a usar un ente central
- El encaminamiento tiene que decidirse en cada nodo
  - ¿Puedo conseguir que todos los nodos conozcan toda la red?
  - ¿Puedo tomar la decisión de por dónde enviar sin conocer toda la red?

- Más difícil que en conmutación de circuitos
- ¿Y en redes de circuitos virtuales?



Más difícil todavía

- Hasta ahora (enrutamiento, congestión…) consideraban a todos los paquetes iguales
- Se puede tener en cuenta que los paquetes tengan diferentes tratamientos
  - A la hora de descartar paquetes en los nodos si la memoria escasea
  - A la hora de informar con control de congestión a unas u otras fuentes de que no envíen tanto
  - A la hora de decidir si un circuito virtual puede o no puede establecerse
  - …
- Según los paquetes pertenezcan a una clase
  - Por su dirección de origen (¿dar mas prioridad a los que paguen más? ¿Network neutrality?)
  - Por la aplicación de usuario a la que pertenezcan (¿las aplicaciones de voz tengan preferencia sobre las descargas de grandes ficheros?)
  - …
- Añadir a las arquitecturas anteriores calidad de servicio

  En general ya es bastante difícil que funcionen sin calidad de servicio, estos son temas más de investigación