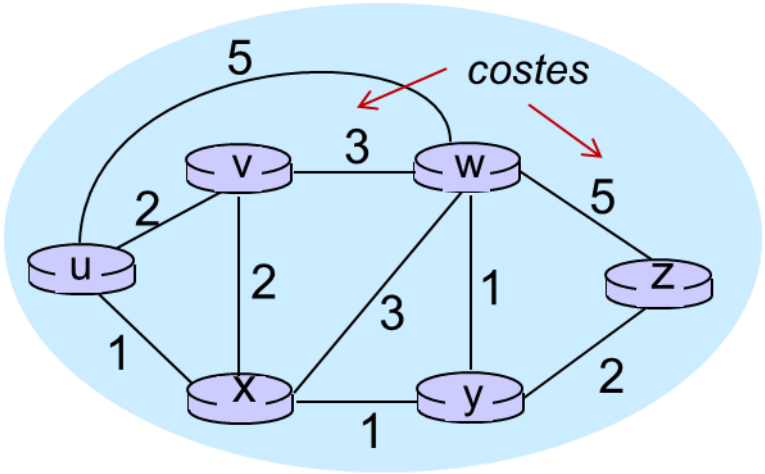
4.- ALGORITMOS DE ENCAMINAMIENTO

- Encaminamiento: Encontrar la ruta óptima para interconectar diferentes redes
- Los diferentes nodos toman decisiones basándose en su conocimiento de la red
  - Información de nodos vecinos
  - Conocimiento a priori de la topología
- Se utilizan protocolos de encaminamiento para obtener la mejor ruta posible
  - Estáticos
  - Dinámicos

Algoritmos de encaminamiento

- **Encaminamiento estático**
  - Se configuran las rutas de forma permanente para cada par de nodos origen-destino
  - Las rutas son fijas
  - Sólo cambian cuando hay un cambio en la topología
  - Los costes de enlace no pueden estar basados en datos dinámicos, pero pueden estar basados en volúmenes estimados de tráfico o en la capacidad de cada enlace



Algoritmos de encaminamiento

- **Encaminamiento estático**
  - Ventajas
    - Carga de procesamiento mínima
    - Fácil de configurar
  - Desventajas
    - Configuración y mantenimiento prolongados y laboriosos
    - Configuración propensa a errores
    - Se requiere la intervención de un administrador para el mantenimiento de las tablas de rutas
    - No se adapta bien a las redes en crecimiento
    - Requiere un conocimiento completo de la red

Algoritmos de encaminamiento

- **Encaminamiento dinámico**
  - Las rutas pueden cambiar para adaptarse a las variaciones de las condiciones en el conjunto de redes
  - Principales condiciones que afectan a las decisiones de encaminamiento
    - Fallo de un encaminador: no se utilizará como parte de la ruta
    - Congestión: encaminar evitando la zona congestionada
    - Añadir nuevos nodos: comprobar si los nuevos nodos, permiten rutas más óptimas que las existentes

Algoritmos de encaminamiento

- **Encaminamiento dinámico**
  - Ventajas:
    - No es necesario reconfigurar el sistema cuando se añaden redes
    - Los protocolos se adaptan de forma automática cuando se produce un cambio en los nodos
    - Es flexible ante fallos o caídas en la red
    - Pueden ayudar a controlar la congestión del tráfico
    - Si está bien diseñado, es menos propenso a errores
    - Escala con mucha más facilidad

Algoritmos de encaminamiento

- **Encaminamiento dinámico**
  - Inconvenientes:
    - Los nodos necesitan un mayor tiempo de procesamiento por paquete
    - Es necesario intercambiar información sobre el estado de la red entre routers: Mayor carga de tráfico para la red
    - Difícil equilibrio en la velocidad de adaptación a los cambios

Algoritmos de encaminamiento

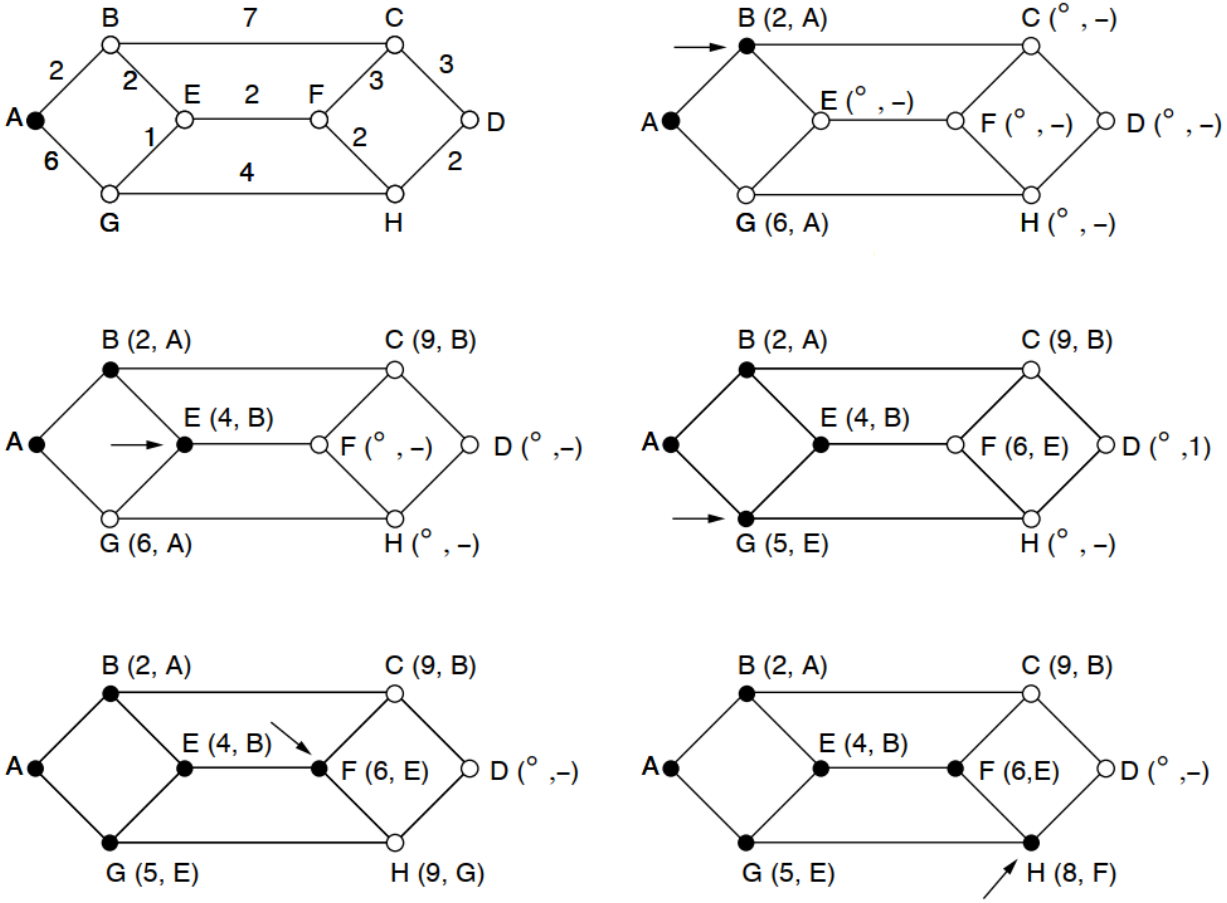
- **Clasificación basada en la fuente de información**
  - Local:
    - Los nodos toman las decisiones basándose únicamente en la información que ellos mismos posean
    - Algoritmos de la patata caliente, aprendizaje hacia atrás e inundación
  - Descentralizado:
    - Los nodos utilizan la información recibida de los nodos adyacentes
    - Algoritmos de vector distancia (DS)
  - Global:
    - Los nodos utilizan la información recibida de todos los nodos
    - Algoritmos de estado del enlace (LS)

Algoritmos de encaminamiento

- **Algoritmo de camino más corto**
  - Busca encontrar la distancia más corta entre dos puntos
  - Los diferentes nodos almacenan información de por dónde tienen que enrutar el tráfico
  - Las métricas de la distancia pueden ser variables- Retardo, velocidad, distancia…
  - Algoritmo de Dijkstra

Algoritmos de encaminamiento

- **Algoritmo de camino más corto**



[1]

Algoritmos de encaminamiento

- **Algoritmo de la patata caliente**
  - Enviar los datagrama por la cola de menor longitud
    - Equilibra la carga entre las redes
  - Combinar la carga de la línea con la dirección preferida de envío
  - Es un algoritmo dinámico
- **Algoritmo de aprendizaje hacia atrás**
  - Cada paquete tiene un contador que se incrementa cada vez que da un salto
  - Si un nodo recibe un paquete por la línea k de H y tiene un 4 en el contador, sabe que enviando por esa línea H estará como mucho a 4 saltos
  - Es un algoritmo dinámico

Algoritmos de encaminamiento

- **Algoritmo de inundación**
  - Cada nodo envía el paquete por todas las líneas excepto por la que le llegó
  - Los paquetes enviados cuentan con un contador que se decrementa por cada salto que da
  - Se pueden implementar mejoras para evitar reenviar paquetes repetidos
  - El emisor necesita conocer la distancia al receptor, o al menos el tamaño máximo de la red

Algoritmos de encaminamiento

- **Algoritmo de inundación**
  - Ventajas
    - Es extremadamente robusto
    - Al menos una copia ha llegado por el camino más corto posible – Puede ser útil para establecer un circuito virtual
    - Se recorren todos los nodos de la red
  - Inconvenientes
    - Se genera una gran cantidad de tráfico

Algoritmos de encaminamiento

- **Algoritmo de vector de distancias**
  - Todos los enrutadores de la red, mantienen una tabla con el resto de nodos de la red, el siguiente nodo de envío y una métrica que indica cuánto tardan en alcanzarlos
  - Se pretende encontrar el camino más corto entre todos los nodos de la red
  - Al inicio del algoritmo, los nodos solo conocen la métrica con sus vecinos, mientras que la métrica con el resto se considera infinita
  - Los nodos vecinos intercambian información de forma periódica
  - Después de varias iteraciones, todos los nodos completan una tabla con información de toda la red

Algoritmos de encaminamiento

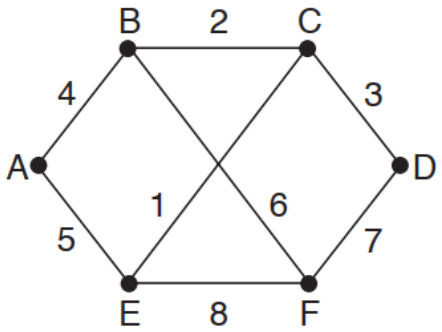
- **Algoritmo de vector de distancias**
  - Es necesario ajustar la frecuencia de los intercambios
  - El algoritmo se adapta muy bien a mejoras en las redes
  - Tiene una convergencia muy lenta en el caso de caída de algún nodo – Cuenta a infinito

Algoritmos de encaminamiento

- **Algoritmo de estado de enlace**
  - Es el más utilizado en Internet a día de hoy en sus diferentes variantes
  - Sigue cinco pasos:
    - Descubrir a los nodos vecinos y conocer su dirección
    - Establecer el coste hasta cada uno de ellos
    - Crear un paquete para transmitir esa información
    - Difundir ese paquete por todas las interfaces
    - Calcular la ruta más corta utilizando la información recibida

Algoritmos de encaminamiento

- **Algoritmo de estado de enlace**
  - Todos los nodos reciben todos los paquetes
  - Es un elemento clave decidir cuando se reenvía



(a)

Enlace

A	
Sec.	
Edad	
B	4
E	5

B	
Sec.	
Edad	
A	4
C	2
F	6

Estado

C	
Sec.	
Edad	
B	2
D	3
E	1

D	
Sec.	
Edad	
C	3
F	7

Paquetes

E	
Sec.	
Edad	
A	5
C	1
F	8

F	
Sec.	
Edad	
B	6
D	7
E	8

(b)

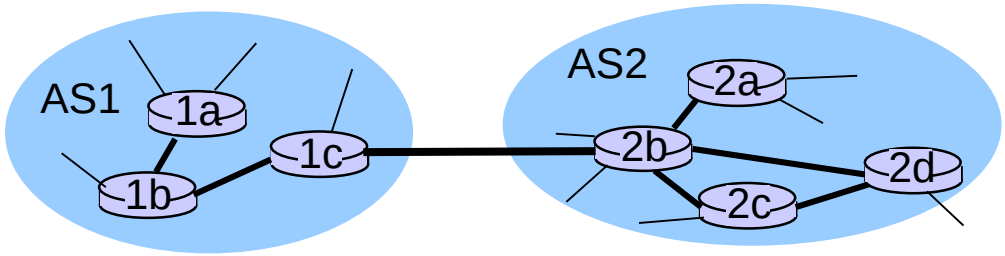
[1]

Algoritmos de encaminamiento

- **Encaminamiento jerárquico**
  - Para redes de gran tamaño, no es viable que todos los nodos conozcan a todos
    - Tablas de encaminamiento enormes
    - Sobrecarga de la red por intercambios de vectores
    - Estructuras de red privadas
  - Encaminamiento mediante sistemas jerárquicos

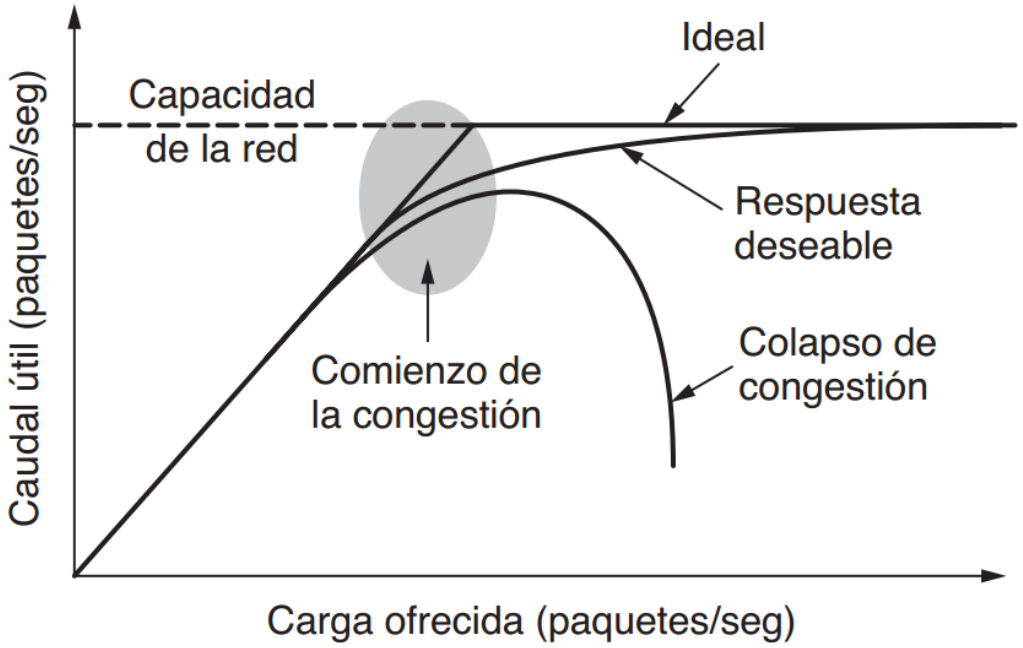
Algoritmos de encaminamiento

- **Encaminamiento jerárquico**
  - Existen una serie de nodos, que conectan redes entre sí
  - Puede poseer múltiples niveles y cada uno de ellos utilizar un mecanismo de encaminamiento diferente:
    - Interior Routing Protocol (IGP)
    - Exterior Routing Procol (EGP)



5.- CONTROL DE CONGESTIÓN

- **Congestión:** Se produce cuando el tráfico enviado a la red, se aproxima a su capacidad máxima
  - Elevados tiempos de entrega de paquetes
  - Paquetes perdidos o descartados
  - Se produce en un nodo y se propaga hacia atrás



[1]

Control de congestión

- **Causas de la congestión**
  - Los diferentes nodos no tienen capacidad de procesar todo el tráfico
  - Las líneas no tienen capacidad para enviar todos los paquetes
  - Las colas de los nodos poseen memoria limitada
    - ¿Se solucionaría con colas infinitas?
  - Es necesario realizar un control que no sature la red y provoque elevadas pérdidas de rendimiento
  - Diferente a control de flujo
    - No afecta solo a los extremos, sino a toda la red
    - Controlar el flujo puede ayudar a controlar la congestión

Control de congestión

- **Métodos de control de congestión**
  - De ciclo abierto o pasivos
    - Realizar un buen diseño de la red
    - Seleccionar a priori qué trafico aceptar y descartar
    - Regular el tráfico para que sea predecible
  - De ciclo cerrado o activos
    - Las decisiones se toman cuando aparece la congestión
    - Consta de 3 fases:
      - Monitorización
      - Envío de información
      - Ajuste del sistema

Control de congestión

- **Métodos de control de congestión: Activos**
  - Monitorización: Controlar diferentes parámetros de la red
    - Longitud promedio de las colas
    - Nº de paquetes para los que vencen los temporizadores
    - Retardo promedio de los paquetes
    - Porcentaje de paquetes descartados por falta de memoria o capacidad de la red

Control de congestión

- **Métodos de control de congestión: Activos**
  - Envío de información: Informar a todos los nodos afectados que se produce congestión
    - Nodo que detecta la congestión envía un paquete especial notificando el problema al origen del tráfico
    - Utilizar un campo (o un bit) del paquete para que los encaminadores avisen de la congestión a sus vecinos
    - Host o encaminadores envían periódicamente paquetes de sondeo preguntando por el estado de la congestión
    - Es muy importante controlar el tiempo de reacción
      - Demasiado pronto: el sistema oscilará y nunca convergerá
      - Demasiado tarde: el aviso ya no será útil

Control de congestión

- **Métodos de control de congestión: Activos**
  - Ajuste del sistema: Introducir diferentes cambios en el sistema para reducir la congestión
    - Balancear el tráfico entre rutas
    - Utilizar encaminadores de respaldo, utilizados para la tolerancia a fallos, para encaminar el tráfico
    - Reducir la inyección de paquetes en la red
    - Negación de servicio a usuarios
    - Descartar paquetes

Control de congestión

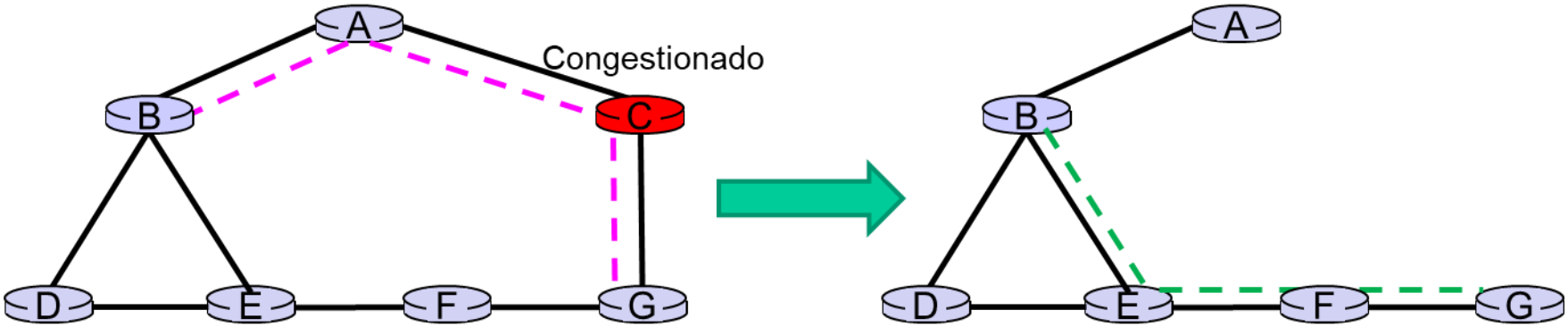
- **Detección temprana aleatoria**
  - *Random Early Detection* (RED)
  - Se aprovecha que para TCP, cuando se pierde un paquete, se reduce el flujo de datos
  - Cuando se reciben señales de congestión en el propio nodo, se empiezan a descartar paquetes de forma aleatoria
  - Al reducirse el tamaño de la ventana de transmisión, se reduce el flujo
  - Suponen una mejora respecto a descartar paquetes cuando se llena el buffer, aunque requieren un mayor ajuste

Control de congestión

- **Regulación de tráfico**
  - Mantener una tasa de encolamiento baja
    - Es dependiente del tipo de tráfico que se tenga
    - Si se pasa un determinado umbral, se toman medidas
  - Envío de paquetes reguladores, que controlen la cantidad de tráfico que se genera – Puede ser hasta el nodo origen o salto a salto
  - Notificación explícita de congestión (Bit *ECN* cabecera IP)

Control de congestión

- **Control de admisión – Circuitos virtuales**
  - No se establecen nuevas rutas hasta que la red pueda trabajar con el tráfico que ya tiene
  - Puede ser complejo estimar la cantidad de tráfico que la red puede manejar, especialmente en el tráfico a ráfagas
  - Se pueden buscar rutas alternativas no congestionadas



Control de congestión

- **Desprendimiento de carga**
  - Es la técnica más agresiva, ya que descarta paquetes completos
  - Se aprovecha de la señalización de los paquetes para saber qué tráfico descartar
    - En ciertos casos, es interesante descartar los paquetes más nuevos y en otros, los más viejos
  - Los paquetes suelen llevar diferentes categorías de transmisión: Normal, urgente, no descartar...

6.- CALIDAD DE SERVICIO (QoS)

- *Quality of Service (QoS)*
  - No todas las aplicaciones pueden utilizar un servicio sin garantía alguna
  - La mejor forma de proporcionar una QoS alta es mediante overprovisionamiento – Demasiado caro
  - La capa de red puede proporcionar algunas mejoras extra que aumenten la QoS
  - Parámetros principales:
    - Ancho de banda
    - Retardo
    - Variación del retardo (*jitter)*
    - Pérdidas

Calidad de servicio

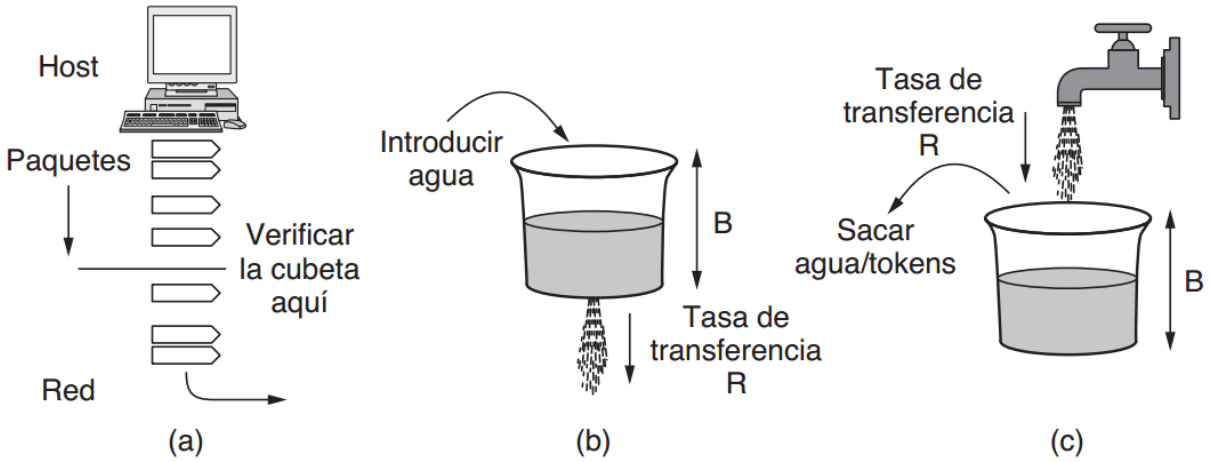
- Requerimientos de QoS de distintas aplicaciones

Aplicación	Ancho de banda	Retardo	Jitter	Pérdida
Correo Electrónico	Bajo	Bajo	Baja	Media
Compartir archivos	Alto	Bajo	Baja	Media
Acceso Web	Medio	Medio	Baja	Media
Inicio sesión remota	Bajo	Medio	Media	Media
Audio bajo demanda	Bajo	Bajo	Alta	Baja
Vídeo bajo demanda	Alto	Bajo	Alta	Baja
Telefonía	Bajo	Alto	Alta	Baja
Videoconferencia	Alto	Alto	Alta	Baja

[1]

Calidad de servicio

- **Modelado de tráfico**
  - Garantizar un tráfico estable pese a las variaciones en la red
  - *Leaky Bucket* y *Token Bucket*



[1]

Calidad de servicio

- **Gestión de paquetes en cola**
  - Método FIFO (*First Input – First Output*)
    - Muy sencillo de implementar
    - Utilizada en el descarte RED
    - No es muy útil para proporcionar una buena QoS
  - Método LIFO (*Last Input – First Output*)
    - Casi tan simple como FIFO y poco óptimo
    - Permite priorizar el tráfico más reciente – Interesante para aplicaciones de tiempo real

Calidad de servicio

- **Gestión de paquetes en cola**
  - Sistemas con prioridad
    - Cada nodo posee varias colas con diferentes prioridades
    - Los paquetes van a cada una de las diferentes colas, en función de la prioridad que tenga
    - Las colas de mayor prioridad se vacían primero
    - Problema: Es posible que las colas con menos prioridad no sean atendidas nunca

Calidad de servicio

- **Gestión de paquetes en cola**
  - Encolamiento circular (*Round Robin*)
    - Todas las tareas van recibiendo el mismo tiempo de procesamiento
    - Una de las implementaciones más comunes
    - Pueden utilizarse variaciones con prioridades, con apropiación, etc.
    - Es el que tiene una implementación más compleja

Calidad de servicio

- **Garantía de QoS**
  - Es necesario combinar todas las técnicas estudiadas previamente
  - Los algoritmos de encaminamiento, basarán sus decisiones según los parámetros que posean los diferentes nodos
    - Teoría de colas – Proporciona los retardos promedio en cada nodo en función de la carga
  - Si la red considera que puede gestionar el tráfico y proporcionar la QoS pedida, aceptará la conexión, sino, podrá rechazarla

Referencias

– [1] Redes de ordenadores, 5ª Ed., Andrew S. Tanenbaum, Prentice Hall