

QoS

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

Redes
4º Ingeniería Informática

Temario

1. Introducción a las redes
2. Encaminamiento
3. Transporte extremo a extremo
4. **Arquitectura de conmutadores de paquetes**
5. Tecnologías para redes de área local
6. Tecnologías para redes de área extensa y última milla
7. Conmutación de circuitos

Objetivos

- Conocer qué es la QoS
- Comprender dónde encaja la planificación dentro de los esquemas de QoS

¿ Qué es esto de la calidad ?

Para el usuario final

- Para un usuario experimentado es normal que una llamada con un ordenador tenga diferente calidad que una por teléfono fijo o que una por móvil
- ¡ Aunque todas se cursen por la misma red !
- Es simplemente aquello a lo que está acostumbrado
- Si nunca ha usado un móvil esperará una calidad similar a la PSTN y se quejará
- Lo mismo si nunca ha usado VoIP
- La calidad es relativa a las expectativas
- Lo mismo con el precio, si está acostumbrado a una tarifa plana o gratis le extrañará pagar

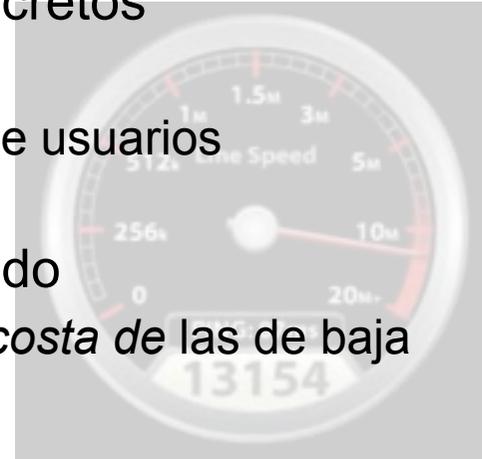
(...)



¿ Qué es esto de la calidad ?

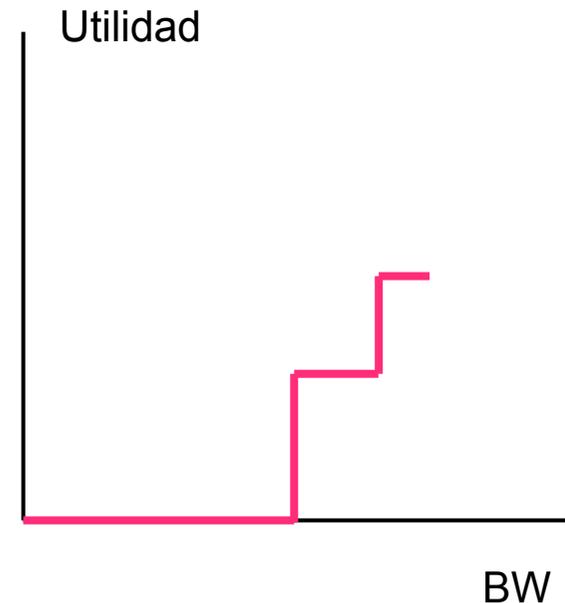
Para el técnico

- Habilidad de la red de *diferenciar* a unos determinados tipos de tráfico, probablemente de unos servicios concretos
- Controlar ciertos parámetros estadísticos:
 - Bandwidth, pérdidas, retardo, jitter... quejas de usuarios
 - Más absolutos y medibles
- Se basa en un reparto “injusto” pero controlado
 - Ofrecer recursos a clases de alta prioridad *a costa de* las de baja
- Formalizados en SLAs
 - Dentro varios SLSs (*Service Level Specifications*)
 - Acuerdo entre proveedor de servicio (la red) y el suscriptor (el cliente)
 - Especifica la calidad de servicio que garantizará el proveedor
 - La red mantendrá su promesa mientras los flujos de usuario se mantengan dentro de su especificación de tráfico
 - Especifica las medidas que se tomarán si se incumple
 - Gran cantidad de parámetros posibles según el servicio



Usuario: Utilidad

- Aplicaciones son sensibles a:
 - Retardos
 - Pérdidas
- Por debajo de un umbral puede no ser útil el tráfico
- Ofrecer garantías de prestaciones para
 - Que el usuario esté satisfecho
 - Que los recursos se usen de forma óptima



¿ Quién necesita QoS ?

- Dos tipos de aplicaciones/tráfico:
 - Elástico
 - Se ajusta ante grandes cambios en retardo y throughput
 - Sigue manteniendo la funcionalidad de la aplicación
 - Inelástico
 - Si no se cumplen unos requisitos de calidad la utilidad se vuelve 0



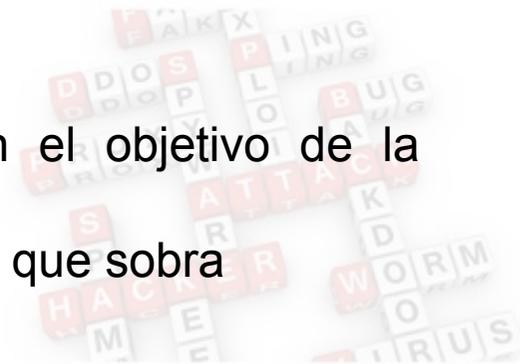
Requisitos de QoS de las aplicaciones

Aplicación	Fiabilidad	Retardo	Jitter	Ancho de Banda
Correo electrónico	Alta (*)	Alto	Alto	Bajo
Transferencia de ficheros	Alta (*)	Alto	Alto	Medio
Acceso Web	Alta (*)	Medio	Alto	Medio
Login remoto	Alta (*)	Medio	Medio	Bajo
Audio bajo demanda	Media	Alto	Medio	Medio
Vídeo bajo demanda	Media	Alto	Medio	Alto
Telefonía	Media	Bajo	Bajo	Bajo
Vídeoconferencia	Media	Bajo	Bajo	Alto

- Transferencia de ficheros: si es interactiva el usuario espera que tarde proporcionalmente al tamaño, luego depende del BW
- (*) La fiabilidad alta en estas aplicaciones se consigue automáticamente al utilizar el protocolo de transporte TCP

¿ Quién necesita QoS ?

- Voz (IP telephony, radio?)
- Vídeo (streaming, videoconferencia)
- Ciertas aplicaciones de datos (generalmente elásticas)
 - *Transactional Data/Interactive Data* (SAP, Oracle...)
 - *Bulk Data* (backups, replicación en redes de contenidos...)
 - *Locally Defined Mission-Critical Data* (mayor que *transactional*)
- Resto:
 - *Best Effort*
 - Dejar BW para él
 - Gran cantidad de aplicaciones en una empresa (centenares)
 - Probablemente no se puedan clasificar todas, ¡no ahogarlas!
- ¿Queda algo?: *Scavenger Service*
 - *Less than BE*
 - Tráfico no deseado: DoS, Worms, etc
 - Web surfing a destinos no relacionados con el objetivo de la empresa
 - Si no se descarta se cursa solo en la capacidad que sobra

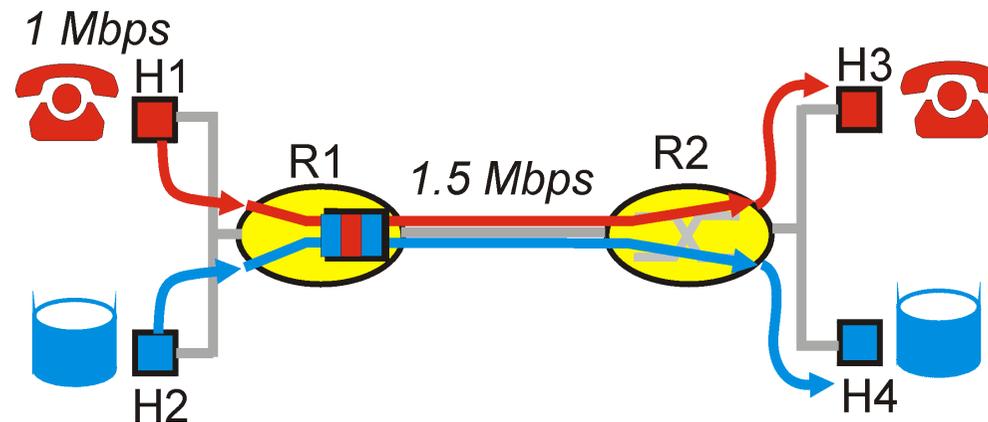


Elementos para garantizar QoS

Elementos

- **Clasificación / Marcado**

- ¿Cómo distinguir entre flujos?
- Ejemplo: Teléfono IP a 1Mbps, comparte enlace de 1.5Mbps con FTP
 - Ráfagas de FTP pueden congestionar el router y causar fallos en el audio
 - Queremos dar prioridad al audio sobre el FTP

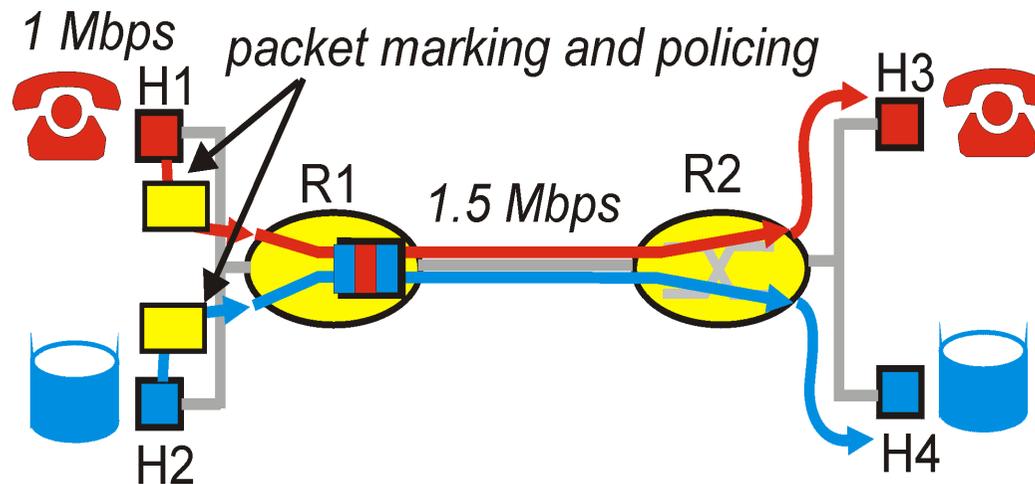


Principio 1

Los routers necesitan distinguir el tráfico de diferentes clases y aplicarles diferentes políticas: *packet marking* (generalmente a la entrada a la red)

Elementos

- **Traffic shaping y policing**
 - Marcar, descartar o retrasar el tráfico en exceso
 - ¿Qué sucede si las aplicaciones no se comportan como deben?
 - Por ejemplo la aplicación de audio envía más de lo previsto
 - Necesitamos forzar que las fuentes se comporten como se ha acordado

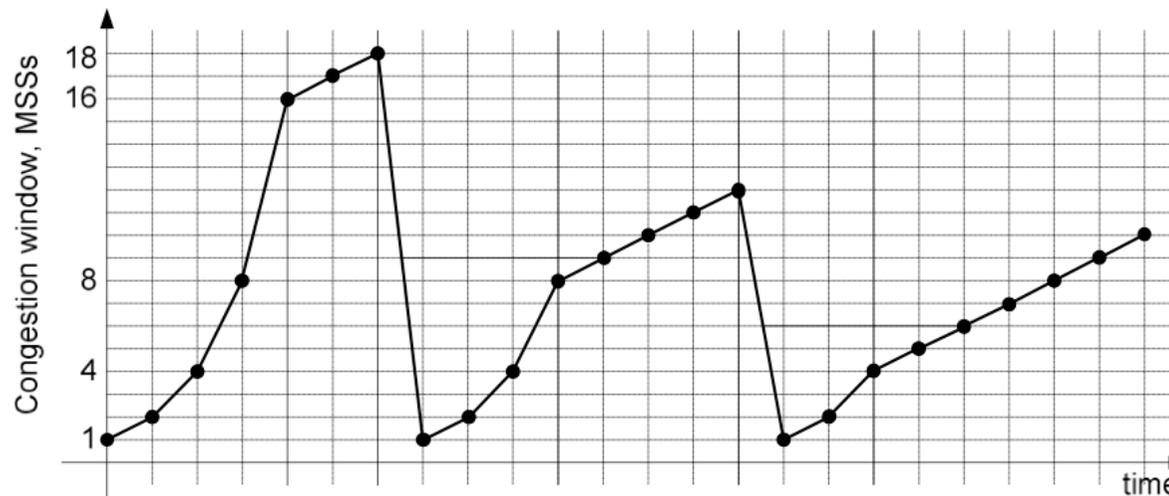


Principio 2

Forzar que una clase de tráfico se comporte dentro de lo contratado:
policing (típicamente a la entrada)

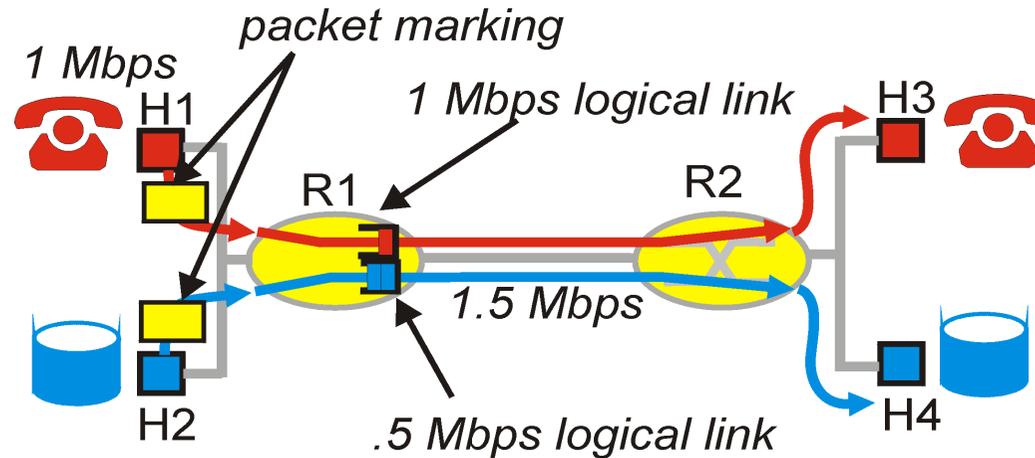
Elementos

- **Gestión de cola**
 - ¿Qué paquetes tirar si se llena?
 - Gestión pasiva como por ejemplo *drop-tail* o *head-drop*
 - Gestión activa (*AQM Active Queue Management*):
 - Ante una llegada descarta paquetes de la cola aunque no esté llena
 - TCP reacciona ante pérdidas reconociéndolas como congestión y bajando la velocidad de envío
 - *RED : Random Early Detection*



Elementos

- **Planificación de recursos (*scheduling*)**
 - El recurso normalmente es el enlace
 - ¿Cómo organizar a los paquete que deben enviarse?
 - ¿Dar prioridades? ¿Repartir la capacidad?

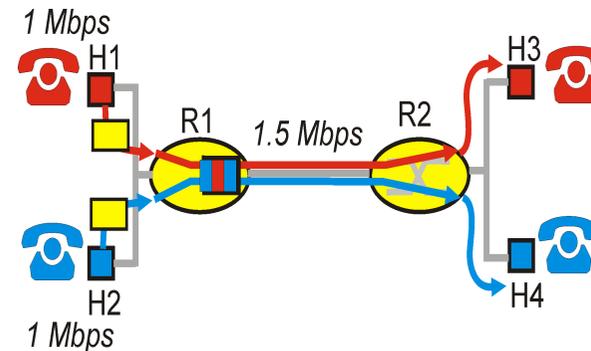


Principio 3

Mientras se ofrece aislamiento es deseable emplear los recursos de forma eficiente (*work conserving*): *scheduling* (en todos los routers del camino)

Elementos

- **Connection Admission Control (CAC)**
 - ¿Puede la red cursar el nuevo flujo de tráfico manteniendo los parámetros de QoS ofrecidos a todos los usuarios?
 - Aceptarlo en la red o rechazarlo
 - No se pueden satisfacer las demandas más allá de la capacidad del enlace
 - Es algo básico desde siempre en redes de conmutación de circuitos porque hay reserva de recursos
 - Con flujos de tasa constante el cálculo es relativamente simple
 - ¿Y con flujos de tasa variable ?

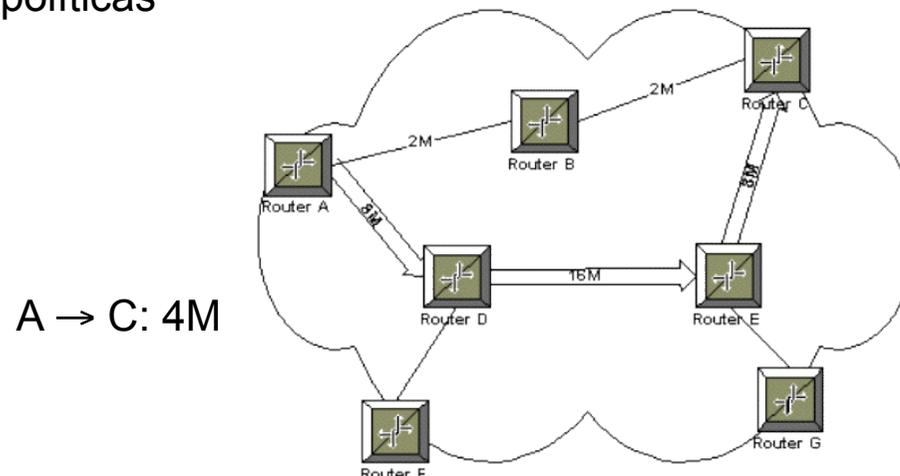


Principio 4

El flujo declara sus necesidades pero la red puede *bloquear* al flujo si no puede satisfacerlas: *call admission*

Elementos

- **QoS routing**
 - Encontrar caminos “buenos” para flujos con requisitos específicos de QoS
 - Usar la red de forma eficiente: aumentar la probabilidad de aceptar peticiones futuras
 - Es complicado:
 - Información precisa sobre el estado de la red es difícil de mantener
 - Calcular caminos que cumplan requisitos de QoS es costoso (computacionalmente hablando)
 - *Constraint-based Routing*
 - Calcular caminos teniendo en cuenta no solo QoS sino también políticas



Arquitecturas

Propuestas del IETF

- **IntServ** (Integrated Services)
 - Filosofía: reserva de recursos
 - Cada router del trayecto ha de tomar nota y efectuar la reserva solicitada
- **DiffServ** (Differentiated Services)
 - Filosofía: priorización de tráfico
 - El usuario o un primer equipo de red marca los paquetes con un determinado nivel de prioridad
 - Los routers van agregando las demandas de los usuarios y propagándolas por el trayecto
 - Esto le da al usuario una confianza razonable de conseguir la QoS solicitada
- Pueden coexistir

IntServ: Características

- RFC 1633
- Para cada flujo (puede ser agregado) reserva recursos en todo el camino
- Orientado a conexión
- Requiere un protocolo de señalización que soporten todos los routers
- No requiere modificar los protocolos existentes
- RSVP no hace la reserva, solo la señala

IntServ: Servicios

- *Best Effort*
- *Controlled load service*
 - RFC 2211
 - “commitment ... to provide ... with service closely equivalent to unloaded best-effort”
 - Prácticamente sin pérdidas
 - No da garantías estrictas
- *Guaranteed service*
 - RFC 2212
 - “provides firm (mathematically provable) bounds on end-to-end datagram queueing delays.”
 - Garantías de BW
 - Retardo acotado
 - Sin pérdidas en buffers
 - Garantías estrictas

DiffServ

- IntServ no escala bien
- RFC 2475, 2638
- Clasificar el tráfico en pocas clases
- Clasifican los *ingress routers* (complejidad en la frontera) con un *codepoint* en la cabecera IP
- DiffServ mapea en cada nodo el *codepoint* en el paquete a un PHB en concreto
- PHB = *Per Hop Behavior*
 - El tratamiento que se le da al paquete en cuestión de scheduling y gestión de cola en ese nodo
 - El mapeo *codepoint* \leftrightarrow PHB debe ser configurable
- No es sensible a los requisitos de un flujo individual

PHBs

PHBs

- *Best-Effort* (BE, codepoint 0)
- *Expedited Forwarding* (EF)
 - Alta prioridad: pocas pérdidas, baja latencia, bajo jitter, bw garantizado
 - “...the departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate.”
 - “The EF traffic SHOULD receive this rate independent of the intensity of any other traffic attempting to transit the node. “
 - Este PHB se puede implementar con PQ, WRR, LLQ, etc.
- *Assured Forwarding* (AF_{xy})
 - En realidad son 4 grupos de PHBs (AF1x, AF2x, AF3 y AF4x) independientes
 - Cada uno tiene una reserva en cada nodo (BW, buffer)
 - Cada uno con 3 probabilidades de descarte (*drop*) (en total 12 codepoints)
 - Drop precedence alta implica mayor probabilidad de descarte
 - Para cada drop precedence, empezando desde la más baja, la probabilidad de descarte debe ser menor o igual que para la siguiente
 - No hay relación entre las probabilidades de descarte de clases diferentes
 - Debe emplear AQM (RED, WRED, ...)

Resumen

- Requisitos de QoS en las aplicaciones
- Elementos en la red para garantizarla:
 - Clasificación y marcado
 - Policing y shaping
 - Gestión de cola y scheduling
 - Routing
- Arquitecturas propuestas por el IETF
 - IntServ
 - DiffServ