



Tema 3: QoS en redes IP

calidad de servicio QoS

- Conjunto de tecnologías q permiten a **aplicaciones** y **usuarios** recibir niveles de servicio predecibles
 - en términos de **capacidad transmisión datos (bandwidth)**, **retardo** y **variación retardo (jitter)**.
 - Pérdidas de paquetes
- Aplicaciones elásticas (tradicionales):
 - adapitan variación estos paráms
 - Modelo Best effort
- Aplicaciones inelásticas (ej: multimedia)
 - muy sensibles → variación jitter

Objetivo: satisfacer las necesidades diferenciadas de los usuarios

Principios de Calidad de Servicio

Principio 1 Tratamiento diferenciado

clase tráfico > **distintos** en red (**WAN**, **LAN**)
distintas **clases**

Principio 2 Vigilancia: policing / shaping

proporcionar grado distanciamiento entre clases
una clase → flujo no se ve afectada más por otro con comportamiento indebido

Principio 3 No se reservan recursos para uso exclusivo

mientras proporciona **deseable** distanciamiento → uso eficiente de **bandwidth** recursos

Principio 4 control de admisión de llamada [CAC]

si red no hay recursos necesarios prestar servicio solicitado
la conexión no será admitida

Retardo de tránsito

componentes que contribuyen retardo extra → extr

- Retardo de **transmisión**: $\Theta(4T_{\text{lat}})$ a través del medio transm.
lat = $\frac{\text{distancia}}{\text{vel. enlace}}$
- Retardo de **Propagación**:
 - en medios metálicos:
- Retardo de **Procesamiento**:
 - en cada nodo → depende $\frac{\text{capacidad nodo}}{\text{tasa de consumo/nodo}}$ → longitud tabla enlace
- Retardo de esperas en colas de E/S:
 - depende nivel de carga del nodo
 - ↑ congestión (agotamiento buffer) → descarte de **paquetes**
- otros: conformación de tráfico, recuperación sincronismo, ...

Gestión de Colas - Descartes

mechanismo básico para ofrecer QoS

- colas **separadas** → interfaces (E/S)
- objetivo: poder atender antes el tráfico más prioritario
- colas → hardware (pequeñas, FIFO) → tamano ajustable
- software (prioritarios) → priority queuing (PQ), weighted fair queuing (WFQ), low-latency queuing (LLQ), ...
- colas de software solo se activan → situación congestión
- Descartes: resolver SATURACIÓN
 - Mecanismo básico (tail drop)
 - Mecanismo Avanzado (RED / WRED)
 - [Weighted] Random Early Detection

Congestión

- los colas de los nodos (**E/S**) congestión → tráfico excede su capacidad
- razones:
 - ↑ vel. interface entrante > interface salida
 - combinación tráfico intenso entrante > interface/s salida

- La QoS no resuelve congestión: alivia cierto tráfico → empeora otro
- los descartes → pueden forzar la regularización (conexiones TCP)

Modelos de Calidad de Servicio

La IETF (Internet Engineering Task Force) → 2 modelos para ofrecer QoS

• servicios diferenciados DiffServ

- modelo QoS basado en clases
- cada elem red → clasifica y trata de forma de acuerdo → toda el tráfico q no da causa de congestión → dependiendo → close a lo q es
- modelo simple y escalable

• servicios Integrados IntServ

- modelo QoS basado en flujos
- cada flujo de info → tratamiento adecuado por parte de todos los elem de la red

- modelo complejo → requiere coordinación elem. red

Tema 3: QoS en redes IP

Servicios Diferenciados

- Orientado a clases de tráfico → todos datagramas = clase → reciben tratamiento homogéneo
- Clasificación tráfico debe **seguirse** → **antes** posible
- Muy escalable → pocas clases y sin sincronización (tratamiento indep.)
- Puede interoperar con dispositivos no Diff Serv
- Hay q establecer dominio de confianza a.
- (se apunta a marcado IP rebeldes q seuelve reasignar)

FASES

- Identificar el tráfico presente en la red
- Dividir el tráfico en redes
 - Voz y video → requiere más prioridad
 - Misión-Critical → Transaccional
 - Best-effort → Tráfico elástico: correo, FPP, tráfico web,...
 - tráfico "basura" (menos q best-effort) ej: aplicaciones P2P

3) Definir la Política QoS a aplicar:

- Bandwidth → Reservar para cada clase tráfico
- tráfico a priorizar y tratamiento
- Mecanismos para evitar congestión

CLASIFICACIÓN / MARCADO

- Clasificación → proceso de identificación
 - ↓ clasificación tráfico en clases
- Criterio: Aplicación, interfaz E, direcc. origen o dst,...

• Métodos de Marcado

- Class Of Service (CoS, 802.1Q) L2
- MPLS experimental (EXP) bits L2
- Tipo de Servicio (TOS) en cabecera IP L3
 - IP Precedence (IPOP)
 - Differentiated Service Code Point (DSCP)

Modelo F(x)al de DiffServ

2 unidades principales → 2 nodos

F(x)s Frontera (nodos límite/tráfico)

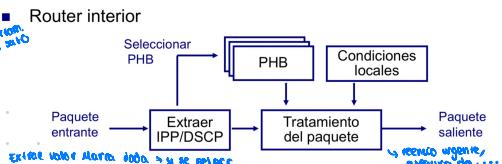
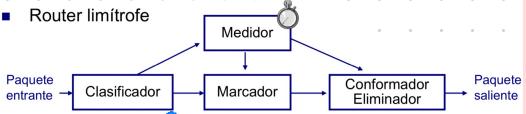
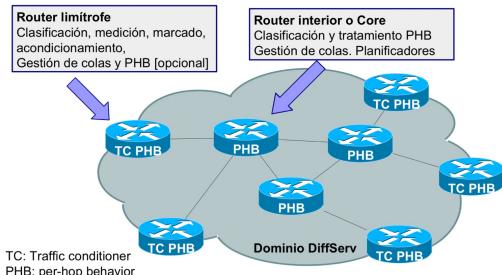
↳ Clasificación/Marcado: rellenar campo **CoS** o **IP**

↳ Acondicionamiento tráfico: conformación/descarte tráfico en exceso

F(x)s de núcleo (nodos internos)

↳ Tratamiento fijo de los **IP** en **f(x)** clase

↳ PHB (Per-Hop Behaviour, comportamiento por salto)



Acondicionamiento de tráfico

- Token-Bucket**: limita B/W q sera consumido por clase

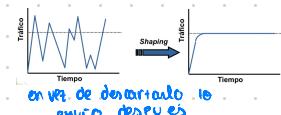
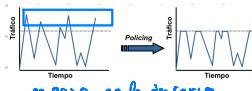
- Cuando los tokens no están disponibles:

+ **IP** está en el buffer **shaping**

+ **IP** se descarta **policing** (no hay token)

+ Opcional: **IP** marca (tráfico excesivo) → **IP** prob descarte mayor

Policing vs Shaping



Planificadores

- PHB** → **IP** tratamientos fijos en **f(x)** clase ∈ [Per-Hop Behaviour]

↳ #s colas de espera

↳ Disciplina de planificación para extraer **IP**

Planificadores

↳ FIFO (best-effort)

↳ Colas con Prioridad (PQ, Priority Queuing)

↳ Colas equitativas (round-robin)

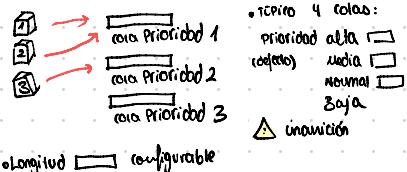
↳ Colas personalizadas (CQ, Custom Queuing)

↳ Colas equitativas ponderadas (WFQ, Weighted Fair Queuing)

Tema 3: QoS en redes IP

(Prioridades)

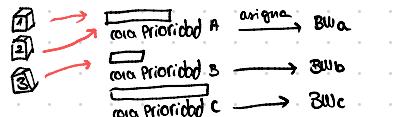
Priority Queuing



Custom Queuing

(redes tener)
cola se procesa en orden → cada cola propia prioridad o peso!!

Término cola ⇒ Procesa contenido proporcional a la longibilidad de la cola
[Antes de pasar siguiente]



Por ejem P10: Priorizar tráfico voz (más BW)

• Config Num colas [10x16] turnos (T) e bytes
obsoleto

Weighted Fair Queuing Wfq

• Objetivo: Priorizar tráfico ligero → peso + clasifica clases

↳ Crear 2 colas dinámicamente para cada tipo de tráfico. [cada cola = peso]

• Flujos: todos datagramas con = valores ↓ IP protocolo → Prioridad

• Config: num y turnos colas

Class-based Weighted Fair Queuing

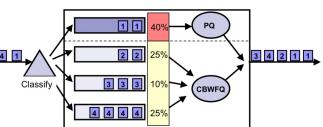
CBWFQ

- Objetivo: repartir BW de forma ponderada.
↳ cada clase tráfico ↔ 1cola ↔ 1peso
- Ausencia: SWI alguna clase no reparte clases → con ponderación relativa
- PESO: BW (típico) 0 < BW

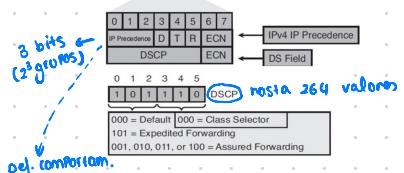
Low-latency Queuing LLQ

- CBWFQ + PQ
- Tráfico prioritario → Prioridad estricta (Aunque se limite BW asignado)

Resto tráfico: BW proporcional al peso



Campo ToS



en función de DSQ → Prioridad

DSCP

PDiffer → 4 colas de tráfico
una vez clasificados
recibirán tratamiento PIBB
(comportamiento por servicios)

- Expedited Forwarding: 101 → urgente
- Assured Forwarding: 4 clases → garantizada
- Class Selector PIBB: usado para dispositivos no DSCP
- Defect PIBB: servicio best-effort sin prioridad

Expedited Forwarding PIBB (EFP)

- Garantiza retardo transito mínimo (jitter) → perdidas
- Garantiza BW reservado → Policing
- Tratamiento: Clases de Prioridad estricta (CQoS)
- Recomendación: USO CAC
- Codificación: 1 0 1 →

Evitar la congestión: RED

Alternativa: tail drop: hay q descartar!!
↓↓↓↓↓

- Random Early Detection: El aleatorio

↑ tom. → ↑ Prob descartar

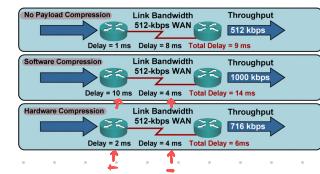
- Conexiones TCP se comportan mejor con
 - ↳ afectan poco
 - ↳ aprovecha mejor BW

WRED (CB-RED)

- Weighted RED → eliminación zanamora de forma ponderada
 - ↳ eliminación afecta más a ID menores Prioritarios
 - ↳ si se anula o claves → CB-WRED

Uso eficiente del enlace: compresión

- Utilizado ? → uso intensivo CPU



Assured Forwarding PIBB

- BW garantizado para cada clase AF

AF1	00 1	-	0	+	cada clase tiene su propio
AF2	0 10	-	0		tratamiento independiente
AF3	0 11	-	0		
AF4	1 00	-	0		

prioridad de descarte (congestión)

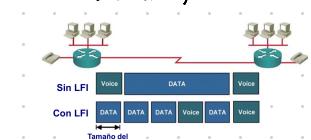
alta (1) Media (10) Baja (0)

- Clases AF tendrían acceso a más BW
- ↳ esto es deseable

Link fragmentation and Interleaving (LFI)

segmentación e intercalado

- ↳ evita tráfico datos intensive (ID grande)
- ↳ retardo Paquetes tributarios



Tema 3: QoS en redes IP

Servicios Integrados (IntServ) RSVP

Cada **Flujo** de usuario **puede recibir tratamientos** *

Paquetes = valores
Protocolo → dirección origen, destino
Puerto origen, destino

Características Principales

- + Serialización extra a extra
- + CAC, control de Admisión (en cada elem. red)
- + Reserva de recursos
- + Mantenimiento de estado (cada \rightarrow flujo)
- + Admite tráfico unicast y multicast

Utiliza RSVP (ReSerVation Protocol)

Solicita de QoS $\xrightarrow{\text{req}}$ nuevo **flujo**
 $\xrightarrow{\text{esta info a lo largo ruta del}}$

Modelo RSVP



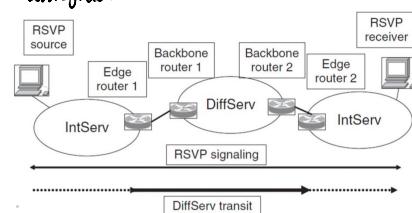
Especificación de servicio [Tspec y Rspec]

- Al Solicitar servicio garantizado mediante RSVP debe indicar correct. tráfico a usar: Tspec [Traffic Specification]
y Requerimientos QoS solicitados: Rspec [Request Specification]

Parámetros Tspec (usados en Token Bucket)

- r: Bucket rate BPS \rightarrow vel. generación tokens
- b: Bucket depth Bytes \rightarrow tam. cibeta
- p: Peak rate BPS \rightarrow vel. pico máxima
- M: Maximum packet size B \rightarrow tam. máximo aceptado
- m: minimum police unit B \rightarrow Cualquier < m Bytes
 \hookrightarrow considerado m Bytes

integración IntServ - DiffServ



Categorías de servicio RSVP

- + Servicio garantizado \rightarrow Tasa de transmisión acordada está garantizada.
- + Servicio controlado \rightarrow en ausencia congestión posibles pérdidas
- + Servicio Best-effort \rightarrow sin garantías ?? pérdidas