



Administración y Gestión de Redes
Lic. Sistemas de información



Laboratorio de Redes,
Recuperación de Información
y Estudios de la Web

Calidad de Servicio

Quality of Services (QoS)

Equipo docente:

Fernando Lorge (florge@unlu.edu.ar)

Santiago Ricci (sricci@unlu.edu.ar)

Alejandro Iglesias (aaiglesias@unlu.edu.ar)

Mauro Meloni (maurom@unlu.edu.ar)

Patricio Torres (ptorres@unlu.edu.ar)

A nivel global

- A nivel global... ¿es diferente un paquete IP que lleva en su carga ... ?
 - un fragmento de película de un servicio de streaming
 - un mensaje de correo electrónico
 - un movimiento en un juego en línea (un first-person shooter)
 - una transacción de banco en línea
 - un movimiento de brazo robótico en una cirugía realizada remotamente
- ¿Se tratan de distinta forma?
- ¿Deben tratarse distinto? ¿Por qué?



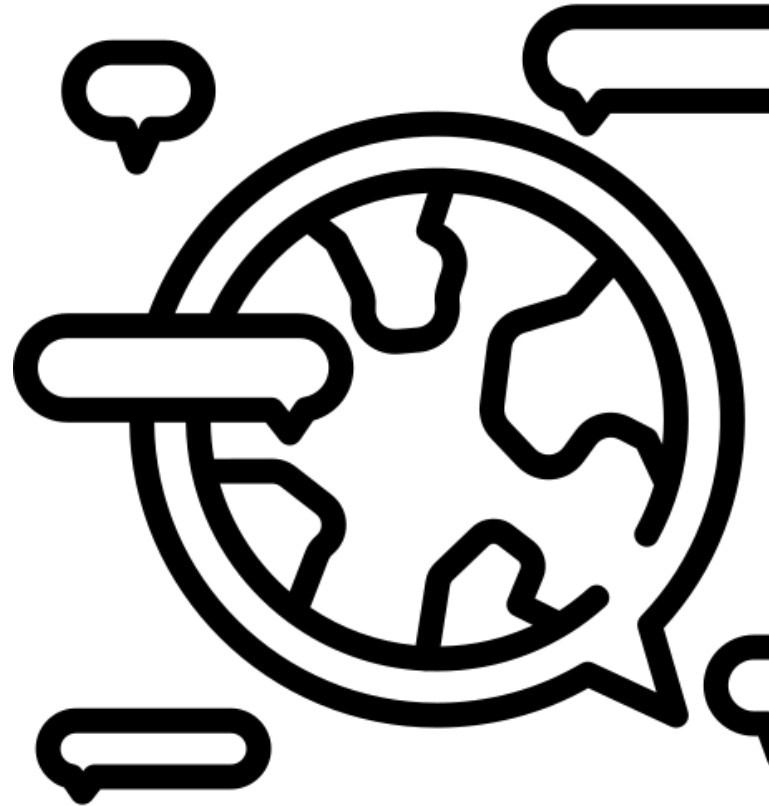
Neutralidad de la red

- A nivel global...
se pretende que Internet se rija por el principio de **Neutralidad de la Red**:
- Esto es: *que todo ISP trate a cualquier comunicación en Internet **de igual manera, sin diferenciar, priorizar o cobrar** de forma distinta al tráfico según quien lo consume, quien lo provee, a qué servicio, protocolo de transporte o aplicación, plataforma, dirección de origen o destino, etc.*
- Esto es similar al uso de la red eléctrica:
¿nos cobran distinto según qué artefacto utilizamos?



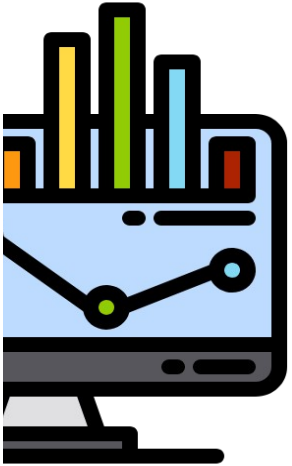
¿Neutralidad de la red?

- ¿Les parece que es así?
 - Los que están a favor, ¿por qué “sí”?
 - Los que están en contra, ¿por qué “no”?
- ¿Y en las redes locales de una organización?
¿cómo debería procederse?



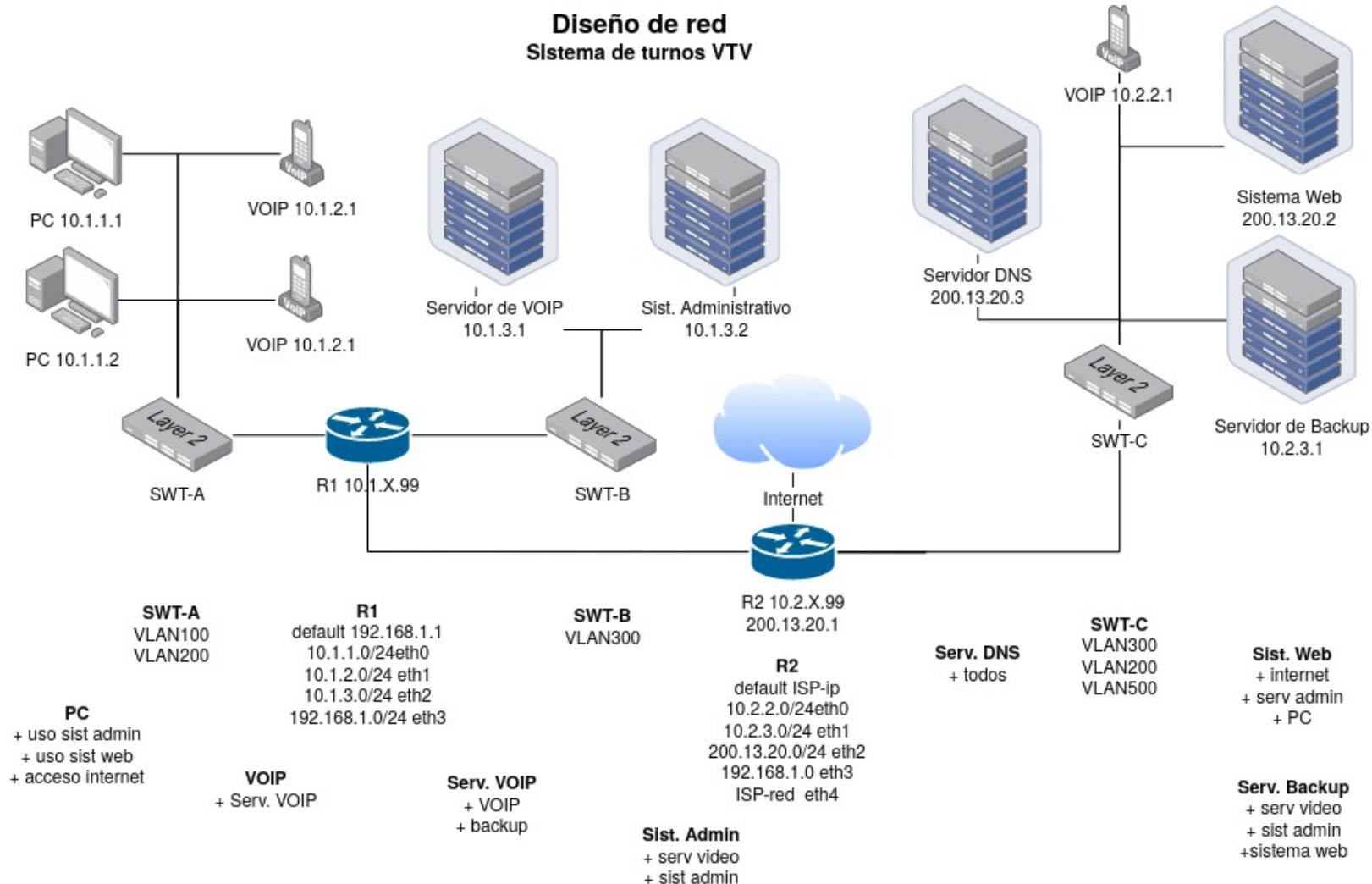
Flow Characteristics	
Performance Requirements	Capacity (e.g., Bandwidth)
	Delay (e.g., Latency)
	Reliability (e.g., Availability)
	Quality of Service Levels
Importance/ Priority Levels	Business/Enterprise/Provider
	Political
Other	Directionality
	Common Sets of Users, Applications, Devices
	Scheduling (e.g., Time-of-Day)
	Protocols Used
	Addresses/Ports
	Security/Privacy Requirements

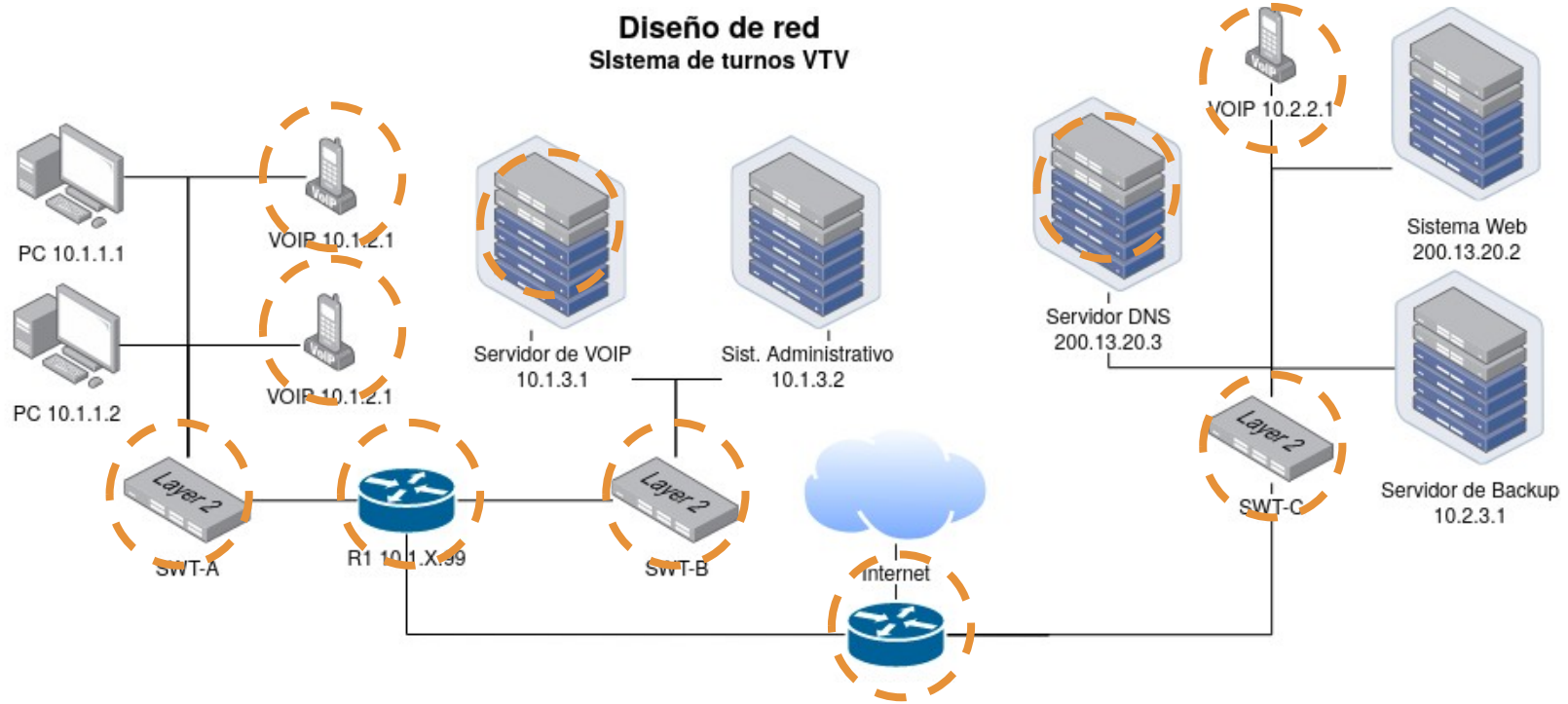
Características del tráfico IP



- **Delay:** retraso de tiempo de entrega del paquete (one-way delay) o de la recepción de la respuesta (RTT).
- **Jitter:** variación del delay entre dos o más paquetes consecutivos.
- **Pérdida de paquetes.**
- **Tasa de transferencia.**
- **Disponibilidad** (de la red o del servicio).
- **Preservación de la secuencia del flujo:** que los paquetes lleguen en orden.

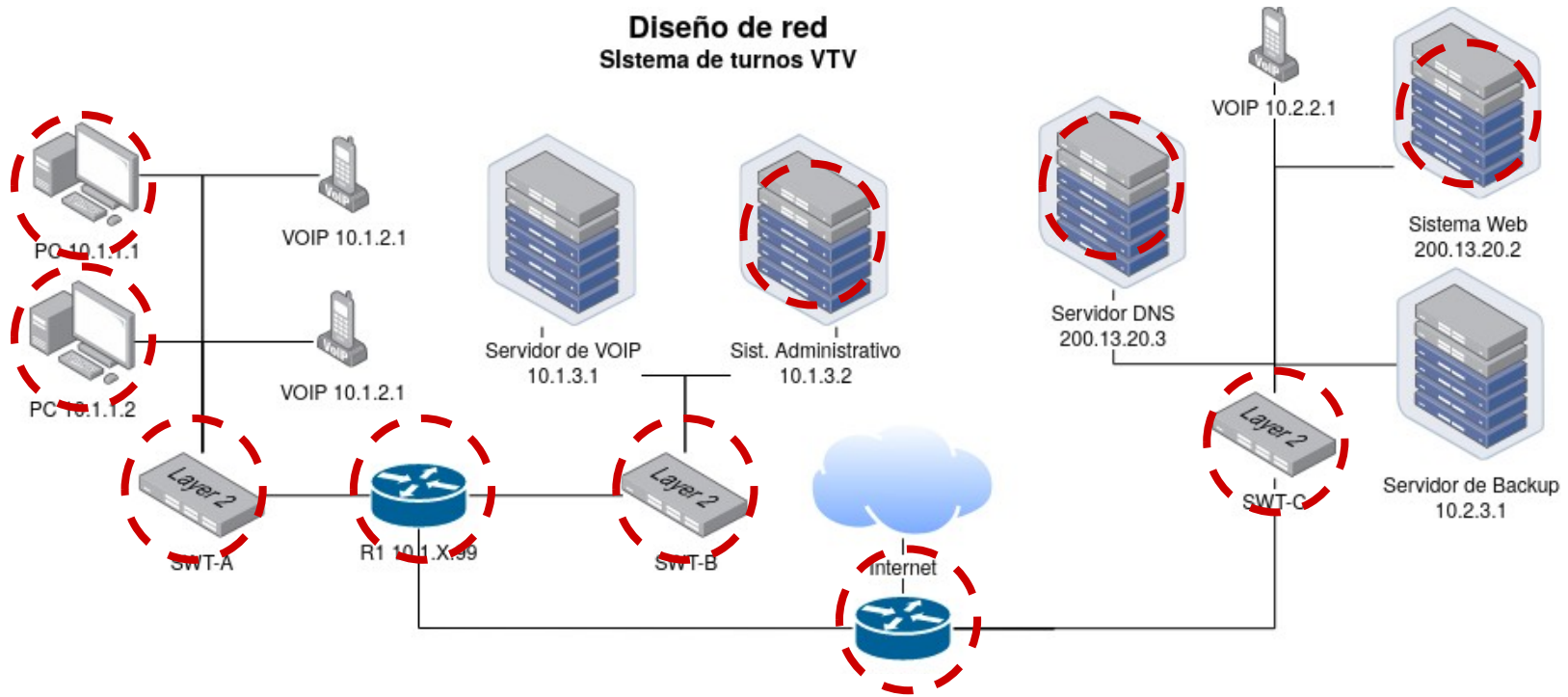
Escenario hipotético



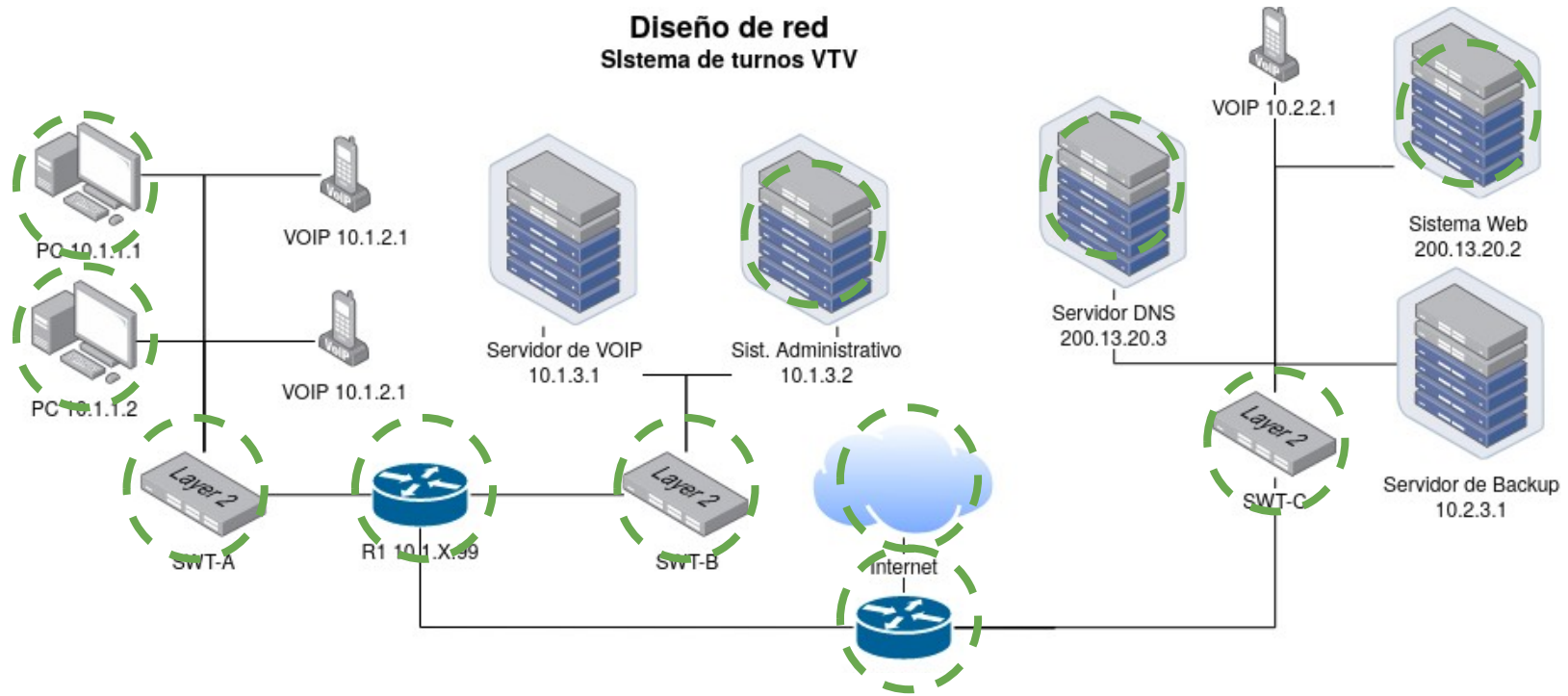


- Requerimientos de tráfico:
 - Bajo delay, Bajo jitter, Baja tasa de transferencia, baja pérdida, bajo reordenamiento.
- Intervienen: VOIP, SWT-A, R1, SWT-B, R2, SWT-C, DNS
- Cantidad de usuarios 3:
 - uso teórico máximo $N*N-1 = 3 * 2 = 6$
- ¿Importancia para la organización?

Sistema Administrativo

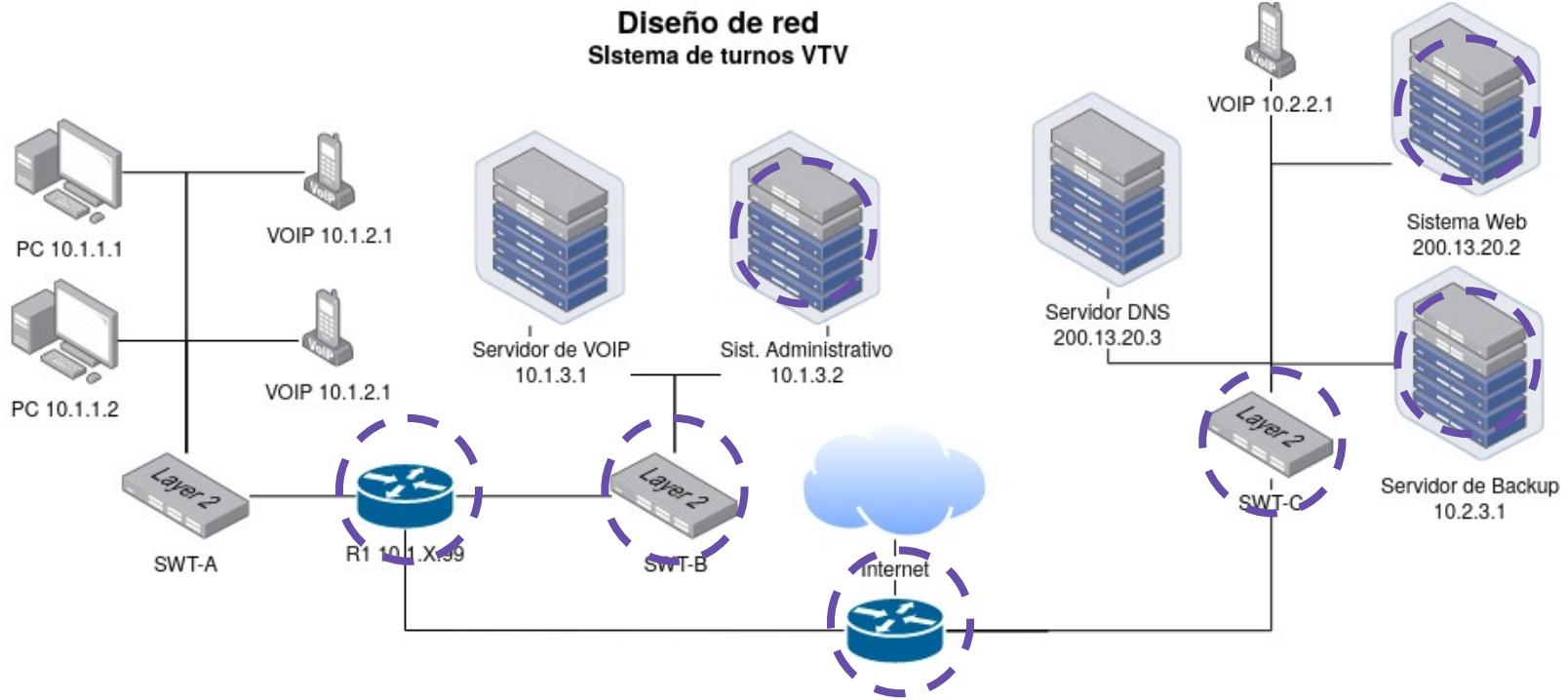


- Tráfico de tipo interactivo:
 - Bajo delay, Bajo jitter, Baja tasa de transferencia
- Intervienen: PC, SWT-A, R1, SWT-B, R2, SWT-C, Admin, DNS, WEB
- No requiere internet. Cantidad de usuarios 2:
- ¿Importancia para la organización?



- Tráfico de tipo interactivo:
 - Bajo delay, Bajo jitter, Baja tasa de transferencia
- Intervienen: PC, SWT-A, R1, SWT-B, R2, SWT-C, Admin, DNS, WEB, Conexión a Internet.
- Cantidad de usuarios: ?
- ¿Importancia para la organización?

Backups



- Tráfico por lotes
 - Alta tasa de transferencia, no importa el delay
- Intervienen: R1, SWT-B, R2, SWT-C, Admin, WEB, Backup
- Cantidad de usuarios: 2
- ¿Importancia para la organización?

¿Falta algún otro flujo de datos? **?**

¿Qué es QoS?

Capacidad de la red para garantizar los recursos necesarios para que un servicio funcione correctamente (respetando las métricas requeridas por el servicio). Por lo tanto se busca:

- **Maximizar satisfacción del usuario final** minimizando los costos.
- Proveer un mejor servicio de red para **tráfico seleccionado**.
- Utilizar **un conjunto de técnicas** para administrar los recursos de la red.

¿Qué dispositivos intervienen?



¿Qué es QoS?

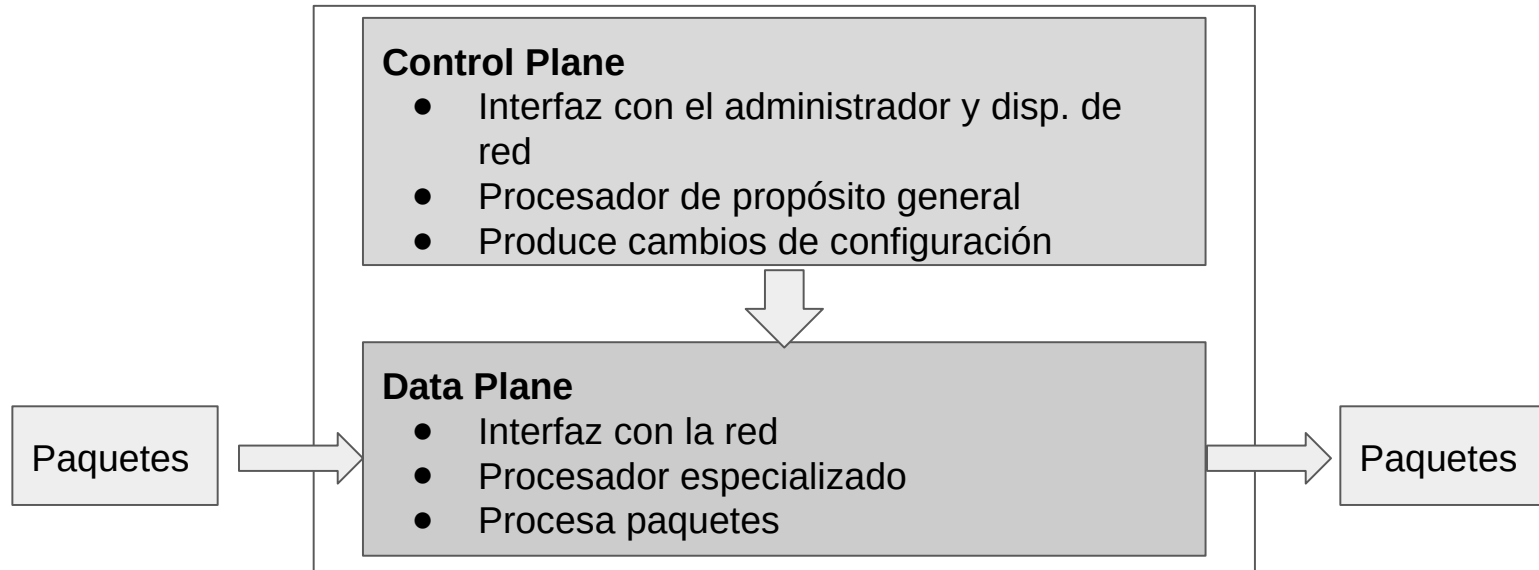
Capacidad de la red para garantizar los recursos necesarios para que un servicio funcione correctamente (respetando las métricas requeridas por el servicio). Por lo tanto se busca:

- **Maximizar satisfacción del usuario final** minimizando los costos.
- Proveer un mejor servicio de red para **tráfico seleccionado**.
- Utilizar **un conjunto de técnicas** para administrar los recursos de la red.

Dispositivos de interconexión (router, switches, etc)

Data plane vs Control Plane

Dispositivo de intercomunicación

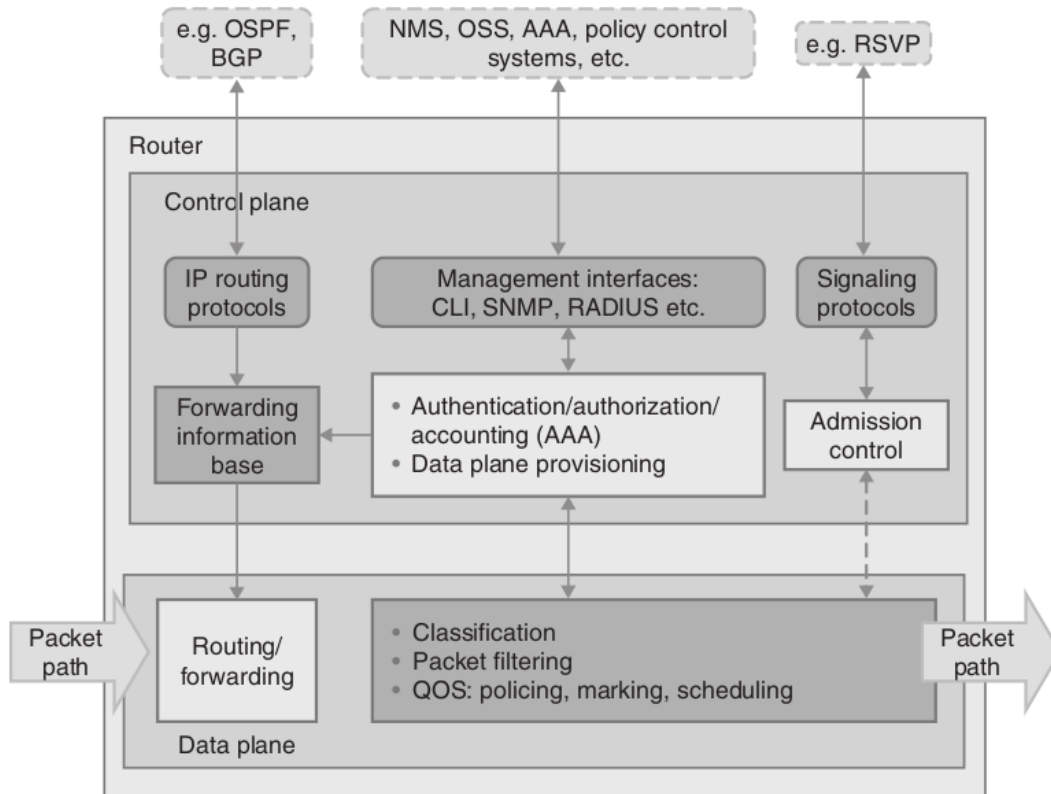


¿Dónde estaría SNMP?



Control plane.

Data plane vs Control Plane



QoS en
Control Plane
RSVP, MPLS

QoS en
Data Plane
Policing, Shaping, Colas etc.

¿Cómo se selecciona tráfico?

Clasificación en flows, streams, traffic classes, de manera:

- **Implícita:** por la interfaz o circuito virtual de origen.
- **Simple:** utilizando campos específicos de los protocolos (ej IPv4 ToS-IPv6 Traffic class → DS, MPLS Exp, Ethernet CoS)
- **Compleja:** utilizando campo/s no específicos, por ejemplo puertos origen o destino.
- **Inspección profunda de paquetes:** analizando datos de capa de aplicación o utilizando datos de conexiones previas relacionadas.

¿Por qué no todos los routers?



¿Cómo se selecciona tráfico?

Clasificación en flows, streams, traffic classes, de manera:

- **Implícita:** por la interfaz o circuito virtual de origen.
- **Simple:** utilizando campos específicos de los protocolos (ej IPv4 ToS-IPv6 Traffic class → DS, MPLS Exp, Ethernet CoS)
- **Compleja:** utilizando campo/s no específicos, por ejemplo puertos origen o destino.
- **Inspección profunda de paquetes:** analizando datos de capa de aplicación o utilizando datos de conexiones previas relacionadas.

Delay: *procesamiento, encolado*, transmission, propagación

Técnicas para administrar los recursos de la red

Para poder proveer QoS se necesita realizar el **acondicionamiento del tráfico**, es decir, aplicar controles y acciones sobre los flujos de datos.

Asegurar que un flujo de datos no supere una tasa determinada:

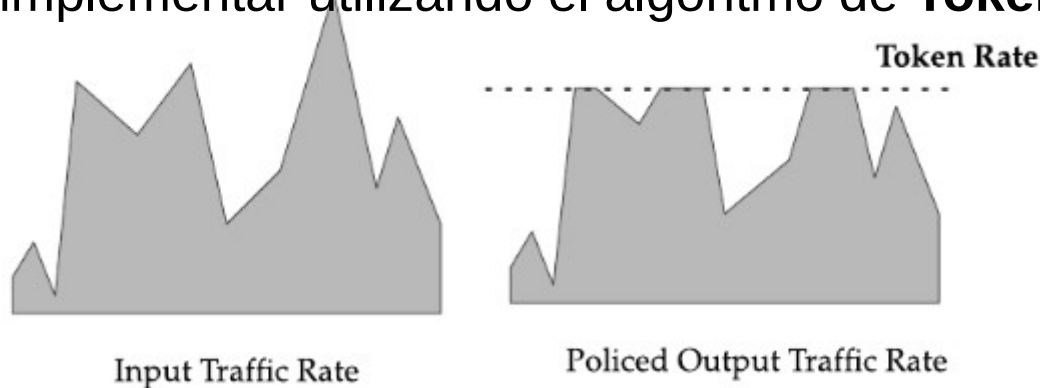
- **Policing:** descartando paquetes.
- **Marking:** marcando paquetes.
- **Shaping:** retrasando paquetes.
- **Metering:** contabilidad para monitorear y controlar.
- **Queueing and Scheduling:** políticas de encolado de paquetes para priorizar.
- **Dropping Techniques:** para controlar el tráfico.

Policing



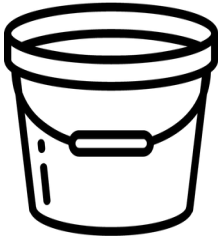
El objetivo del policing es asegurar que determinado tráfico (stream) no supere una tasa máxima definida, dropeando (o marcando) los paquetes que exceden la tasa deseada.

Se puede implementar utilizando el algoritmo de **Token Bucket**.



Token Bucket

Es un mecanismo que sirve para limitar la tasa media de transferencia de un flujo, permitiendo “ráfagas” hasta un tamaño máximo admitido.



CIR: Committed Information Rate - Tasa de Caudal Comprometido

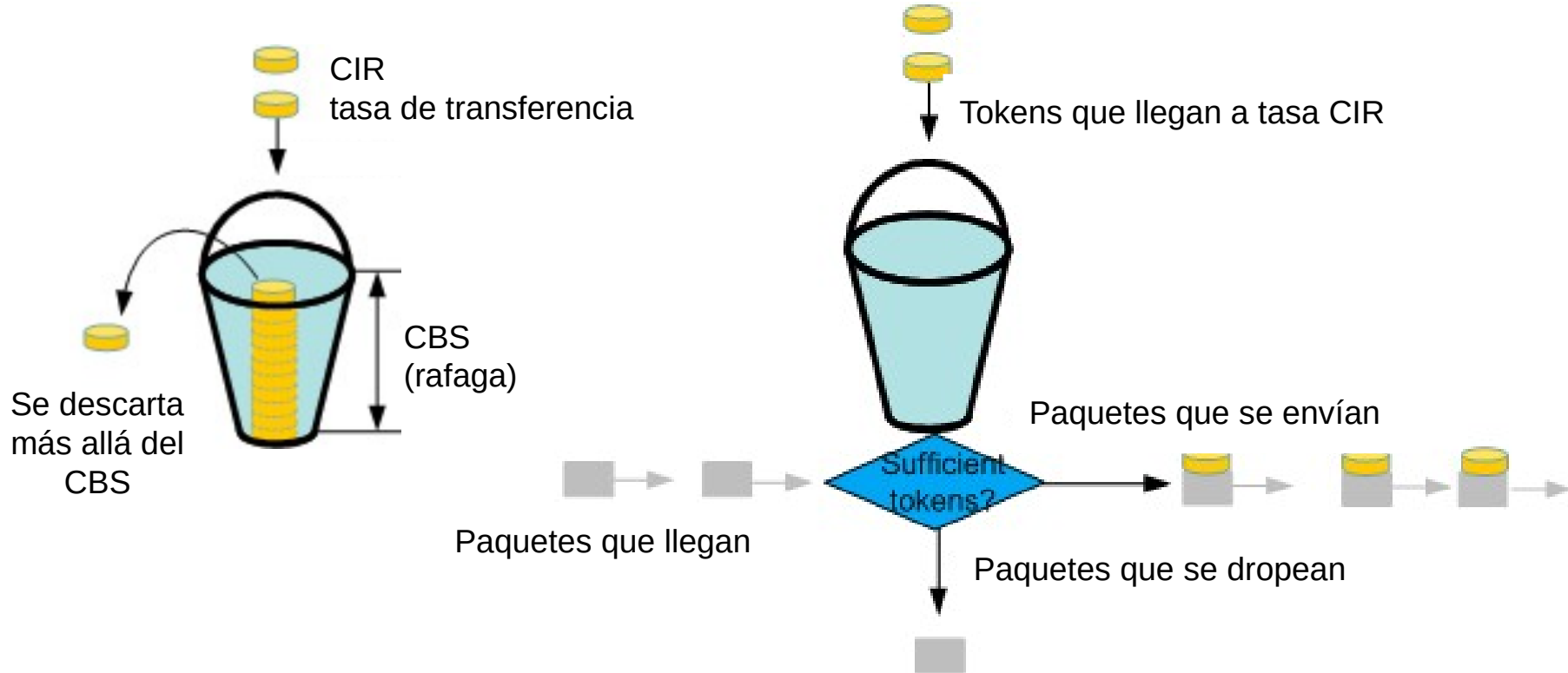
CBS: Committed Burst Size – Tamaño de Ráfaga Comprometido

T: Time Interval - Intervalo

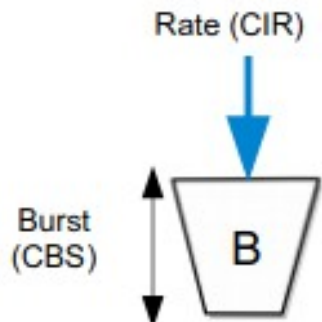
El balde almacena “tokens”.

Existen diferentes variaciones del mecanismo.

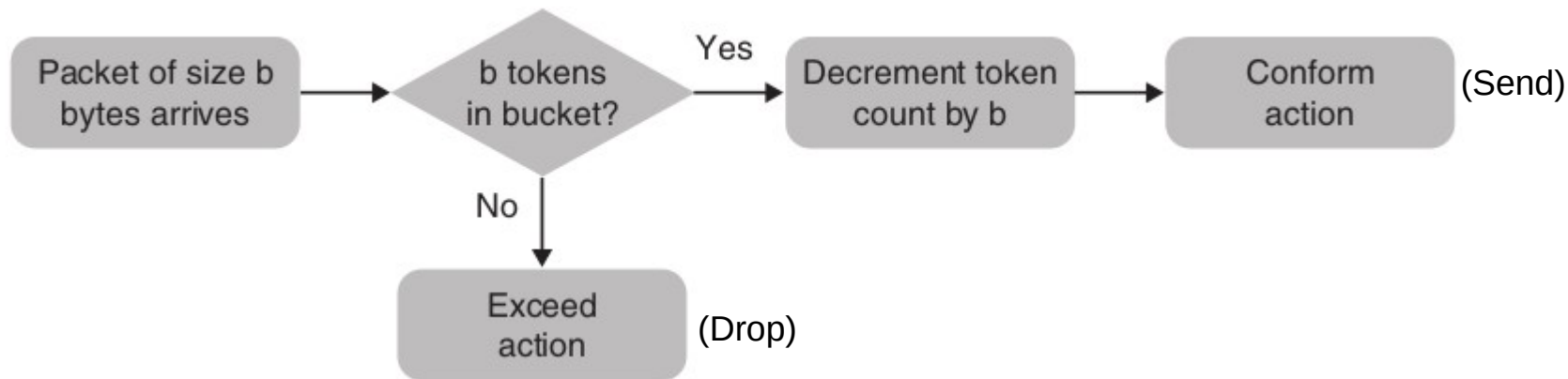
Token Bucket simple



Token Bucket simple

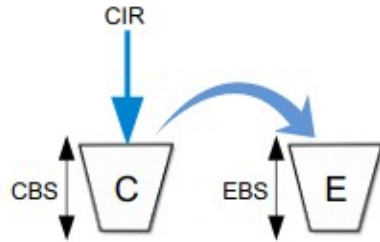


El balde se llena a una tasa “rate” en bps* (o CIR) y se acumula hasta un tamaño Burst en bytes* (CBS).

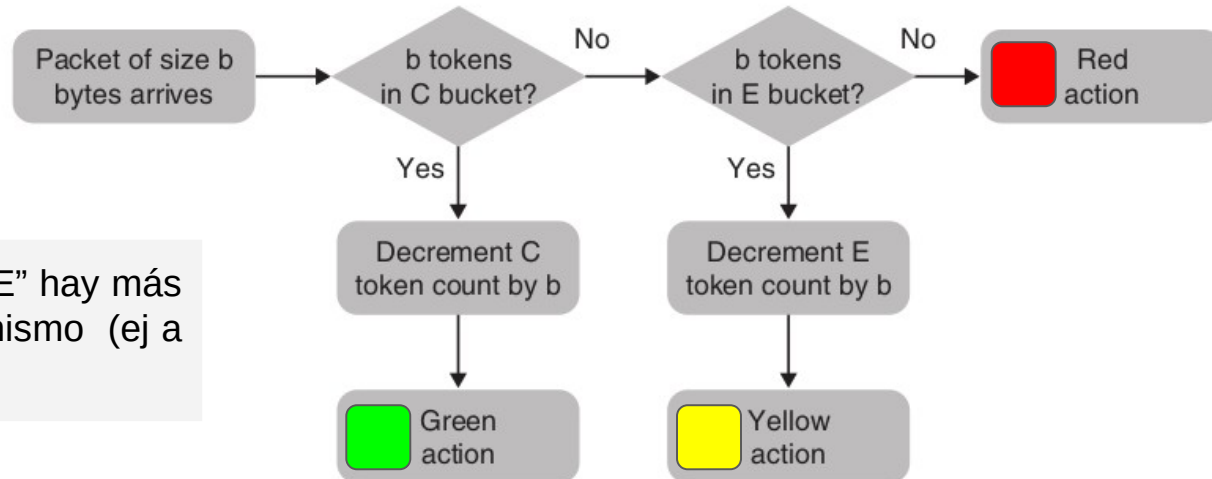


*por ejemplo

Single Rate Three Color Marker



Se agrega un balde nuevo (**E**xceded) que se llena del excedente del primer balde (**C**ommitted) hasta un máximo EBS.



Según como se llena “E” hay más variantes a este mecanismo (ej a tasa propia).

Shaping



Asegurar que determinado tráfico no supere una tasa máxima definida. Shaping retrasa los paquetes que exceden el límite encolándolos.

Se puede implementar utilizando **Token Bucket** o bien **Leaky Bucket**.

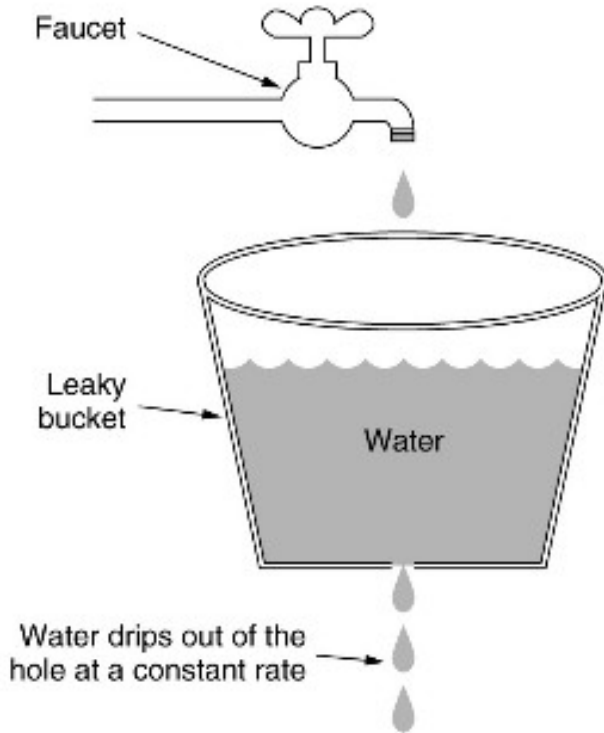


Input Traffic Rate



Shaped Output Traffic Rate

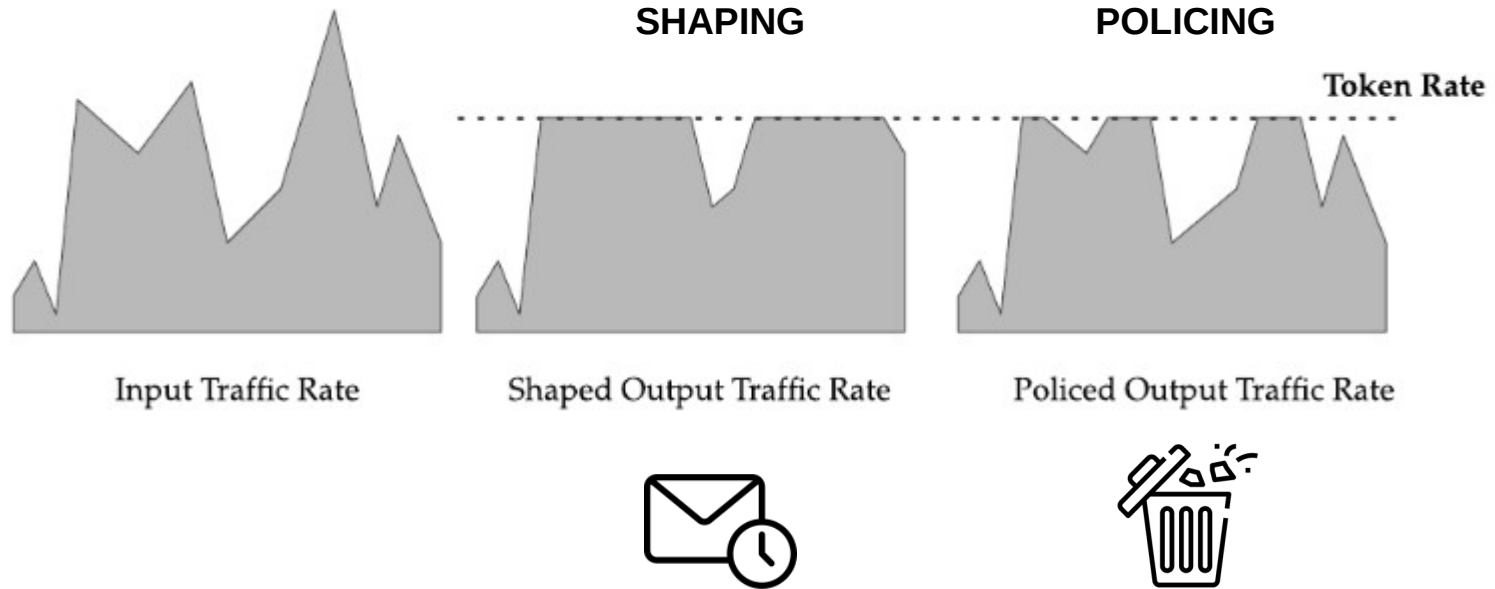
Leaky Bucket



En el balde ahora se acumulan paquetes, que se acumulan hasta el tope (B) que representa la cantidad de paquetes que se pueden encolar.

Los paquetes se entregan o “liberan” del balde a una tasa “ R ”. Si llega un paquete y el balde está lleno se dropea.

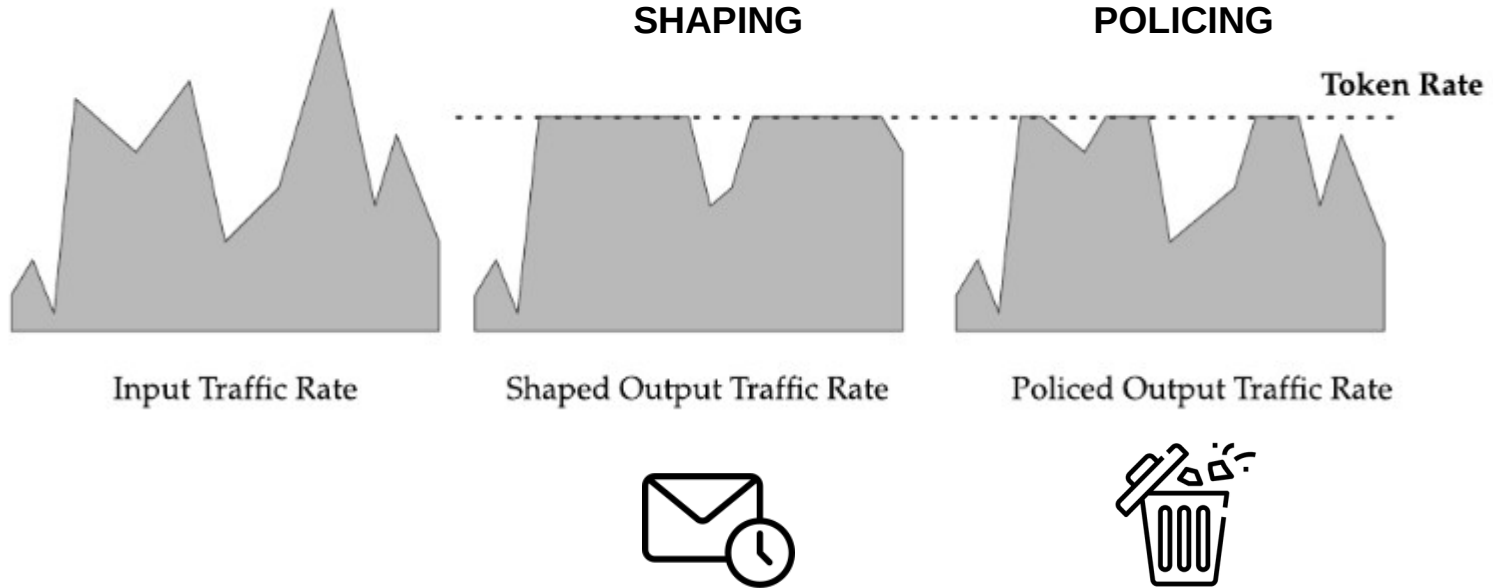
Policing vs Shaping



¿Cuales es mejor y para qué?



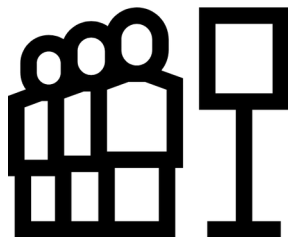
Policing vs Shaping



Policing para tráfico tiempo real

Encolado y planificación

Manejar los paquetes recibidos y planificar su envío en una interfaz de salida. Para ello hay que tener en cuenta:

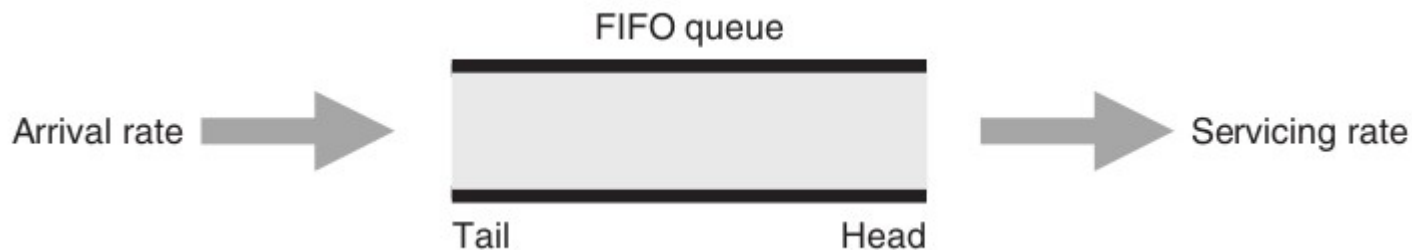


- El tamaño de las colas no puede ser infinito
- Las colas deben tener algún mecanismo para dar prioridad.
- Deben ser eficiente en el uso de recursos.
- No deben morir de “inanición” flujos de datos.
- Las demoras producidas afectan no solo la tasa de transferencia sino también, el delay y el jitter.

FIFO

Primero que entra, primero que sale.

- Fácil de implementar
- Retardo máximo conocido: Tamaño búffer/capacidad del enlace
- No provee OoS

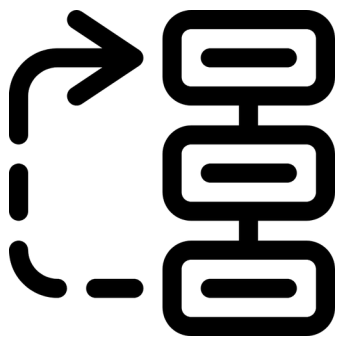


¿Cómo se podría dar QoS?



Encolado, planificación y prioridades

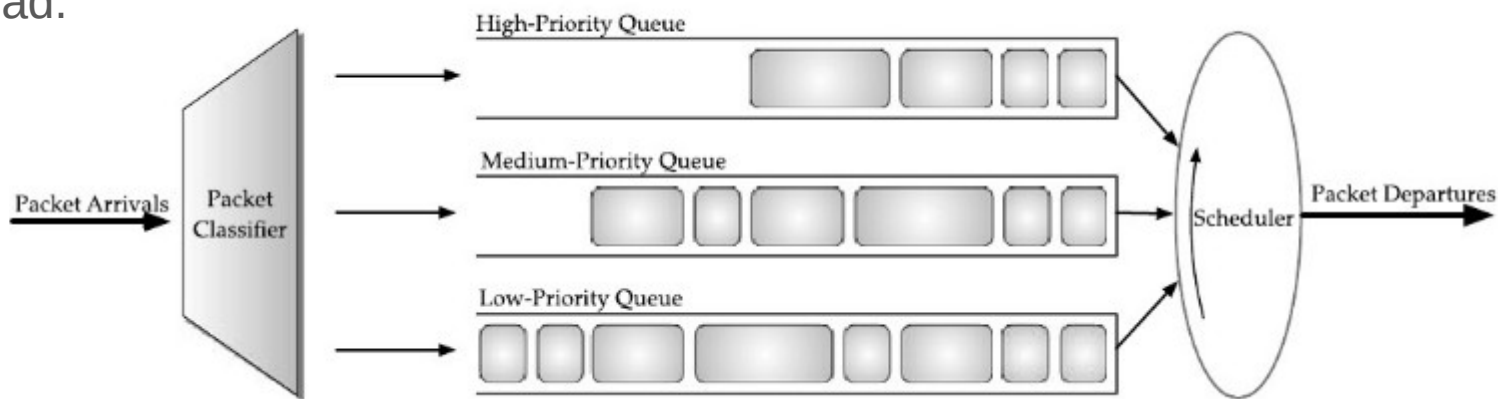
Para soportar QoS se requieren de técnicas más complejas, que permitan un tratamiento distinto a paquetes arbitrarios.



- **Strict Priority Queueing**
- **Weighted Bandwidth Scheduling**
 - Weighted Round Robin
 - Weighted Fair Queuing
 - Deficit Round Robin

Strict Priority Queueing

Existen colas (queues) con diferentes prioridades. Se sirve primero las de mayor prioridad.

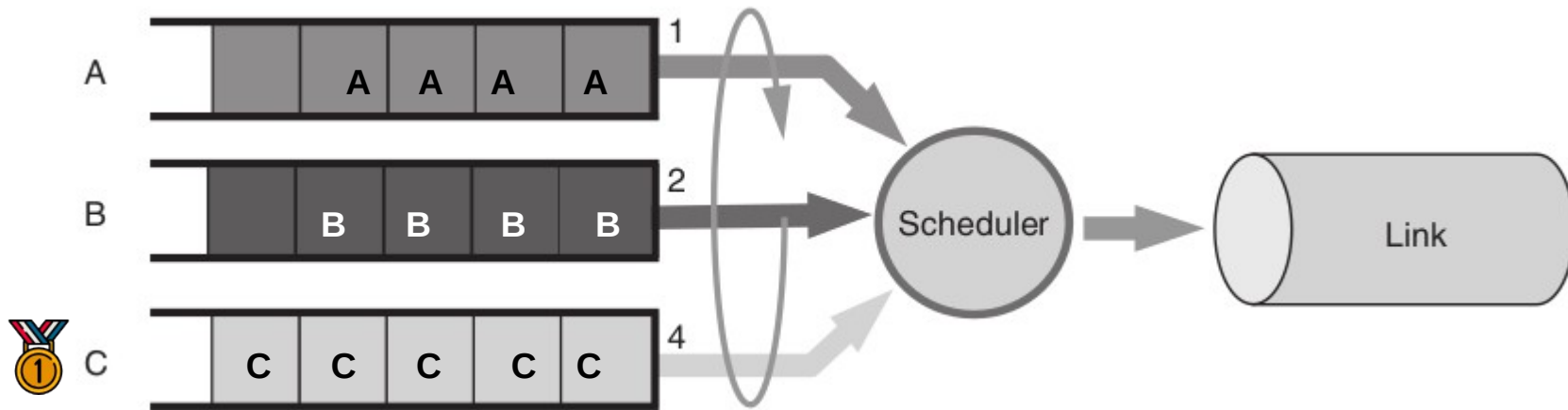


Un sistema de Strict Priority Queuing puede ser pre-emptive

- A nivel de paquete: se interrumpe la emisión de un paquete de menor prioridad
- A nivel de Quantum: se espera a que se termine de enviar el paquete

Weighted Bandwidth Scheduling

Se asigna un peso a cada queue, y se sirven los paquetes según ese peso.

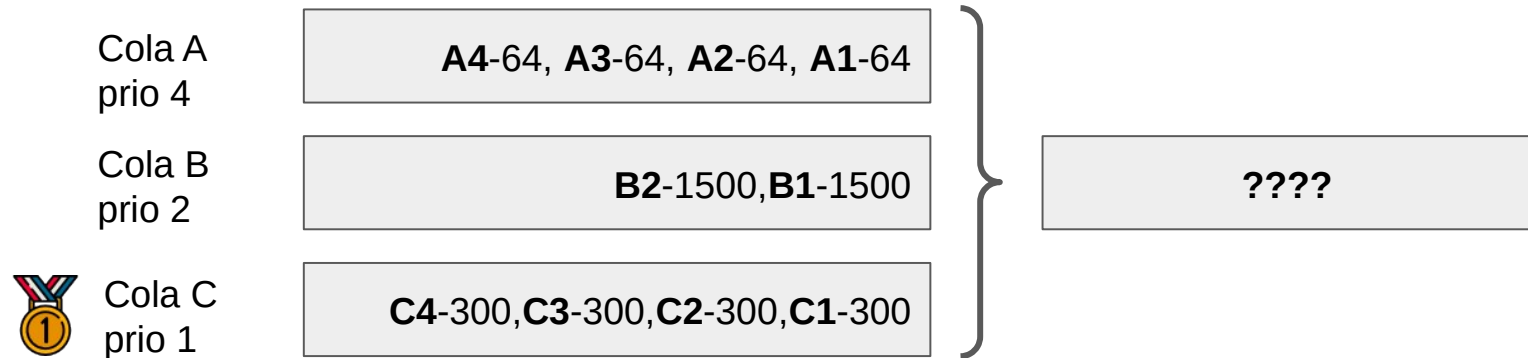


El orden de envío sería: A, B, B, C, C, C, C, A, B, B, C, A, A....

¿Es justo este método? 

Weighted Fair Queueing (WFQ)

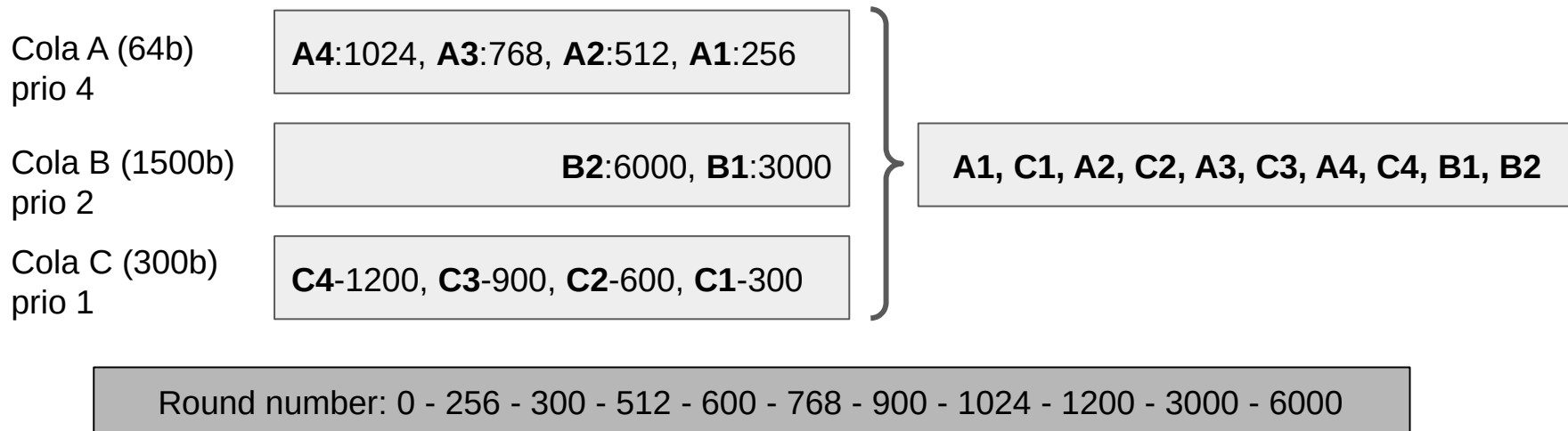
Se asigna un peso a cada queue, y se planifica el envío de los paquetes según su tiempo de “finalización” de envío, teniendo en cuenta su peso y un número que identifica la ronda (un acumulador).



Más peso, significa menos importante.

Weighted Fair Queueing (WFQ)

Se calculan los “Tep” (tiempo de envío para cada paquete) y se incrementa el RN por cada paquete.

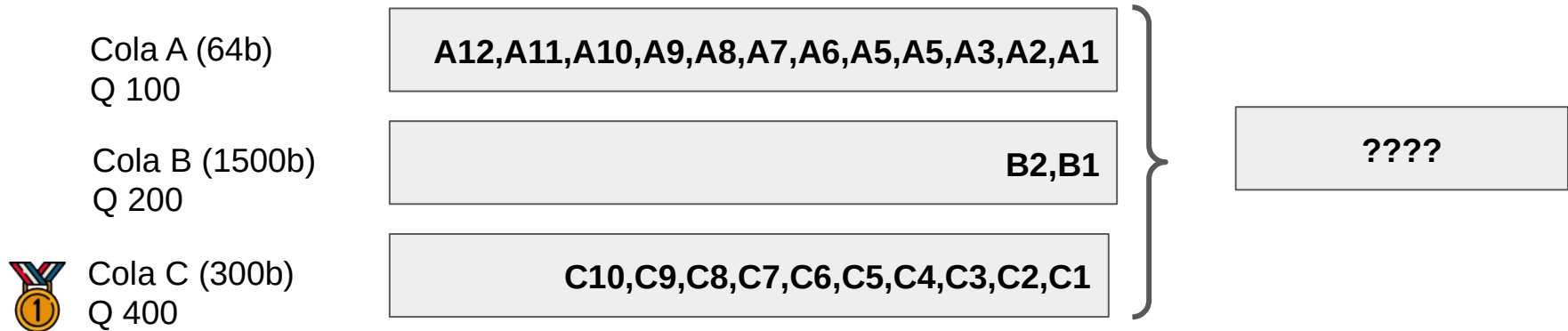


Es justo, pero tiene un costo de procesamiento elevado para muchos elementos que impacta más para conexiones rápidas.

Deficit Round Robin (DRR)

El planificador utiliza, para cada queue:

- Un Quantum de peso o crédito disponible (determina la prioridad)
- El concepto de déficit o deuda.



Deficit Round Robin (DRR)

A4, A3, A2, **A1**

B2,B1

C4, C3 ,C2 ,**C1**

Queue		Round 1
A	Quantum	100
	Pkts sent	1 * 64B {A1}
	Deficit	36
B	Quantum	200
	Pkts sent	0
	Deficit	200
C	Quantum	400
	Pkts sent	1 * 300B {C1}
	Deficit	100

Deficit Round Robin (DRR)

A5, A4, A3, A2, A1

B2,B1

C4,C3,C2,C1

Queue		Round 1	Round 2
A	Quantum	100	136
	Pkts sent	1 * 64B {A1}	2 * 64B {A2, A3}
	Deficit	36	8
B	Quantum	200	400
	Pkts sent	0	0
	Deficit	200	400
C	Quantum	400	500
	Pkts sent	1 * 300B {C1}	1 * 300B {C2}
	Deficit	100	200

Deficit Round Robin (DRR)

A7, A6, A5, A4, A3, A2, A1

B2,B1

C5, C4,C3,C2,C1

Queue		Round 1	Round 2	Round 3
A	Quantum	100	136	108
	Pkts sent	1 * 64B {A1}	2 * 64B {A2, A3}	1 * 64B {A4}
	Deficit	36	8	44
B	Quantum	200	400	600
	Pkts sent	0	0	0
	Deficit	200	400	600
C	Quantum	400	500	600
	Pkts sent	1 * 300B {C1}	1 * 300B {C2}	2 * 300B {C3, C4}
	Deficit	100	200	0

Deficit Round Robin (DRR)

A8, A7, A6, A5, A4, A3

B2 ,B1

C6, C5, C4,C3,C2,C1

Queue		Round 1	Round 2	Round 3	Round 4
A	Quantum	100	136	108	144
	Pkts sent	1 * 64B {A1}	2 * 64B {A2, A3}	1 * 64B {A4}	2 * 64B {A5, A6}
	Deficit	36	8	44	16
B	Quantum	200	400	600	800
	Pkts sent	0	0	0	0
	Deficit	200	400	600	800
C	Quantum	400	500	600	400
	Pkts sent	1 * 300B {C1}	1 * 300B {C2}	2 * 300B {C3, C4}	1 * 300B {C5}
	Deficit	100	200	0	100

Deficit Round Robin (DRR)

B2 ,B1

,C1

Queue		Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Round 7	Round 8
A	Quantum	100	136	108	144	116	152	124	100
	Pkts sent	1 * 64B {A1}	2 * 64B {A2, A3}	1 * 64B {A4}	2 * 64B {A5, A6}	1 * 64B {A7}	2 * 64B {A8, A9}	2 * 64B {A10, A11}	1 * 64B {A12}
	Deficit	36	8	44	16	52	24	0	36
B	Quantum	200	400	600	800	1000	1200	1400	1600
	Pkts sent	0	0	0	0	0	0	0	1 * 1500B {B1}
	Deficit	200	400	600	800	1000	1200	1400	100
C	Quantum	400	500	600	400	500	600	400	500
	Pkts sent	1 * 300B {C1}	1 * 300B {C2}	2 * 300B {C3, C4}	1 * 300B {C5}	1 * 300B {C6}	2 * 300B {C7, C8}	1 * 300B {C9}	1 * 300B {C10}
	Deficit	100	200	0	100	200	0	100	200

Comparando planificadores y queues

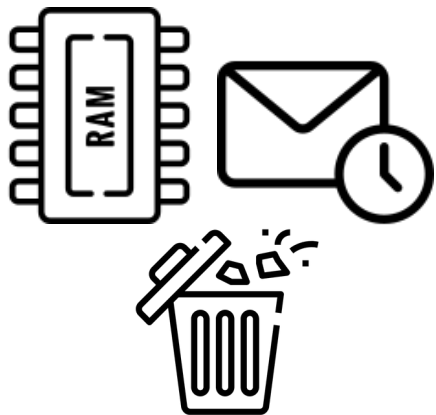
Al momento de elegir un planificador y un tipo de queue, hay que tener en cuenta:

- Fairness
- El delay del peor caso.
- Simplicidad (escalabilidad)

- ¿Por qué no alcanza con Weighted Bandwidth Scheduling (WBS)?
- ¿Porqué es menos costoso computacionalmente DRR que WFQ?
- ¿Cuál mecanismo de queues es más apropiado para clases de tráfico interactivo?
- ¿Cual es es tamaño máximo de las queues?



Técnicas de drop



¿Cuándo hay que dropear paquetes?

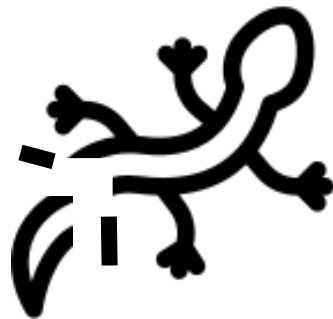
- Buffer lleno.

¿Por qué no buffers más grandes?

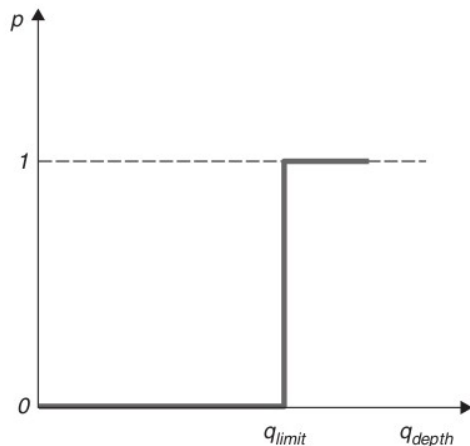
- Crece el tiempo de encolado, mayor delay, SLA/objetivos de latencia dependiendo de la aplicación.

Existen diferentes técnicas:

- Tail Drop
- Weighted Tail Drop
- Random Early Detection
- Weighted Random Early Detection

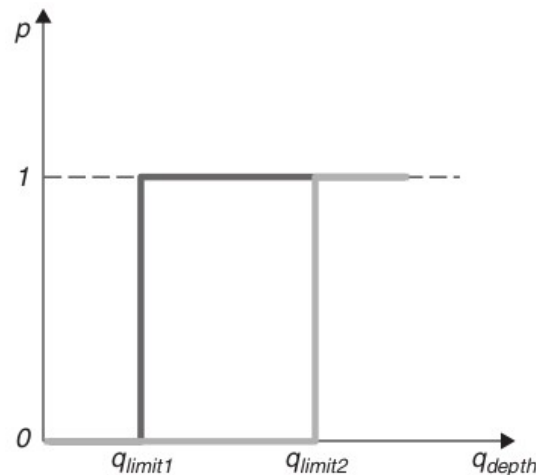


Tail drop & Weighted Tail Drop



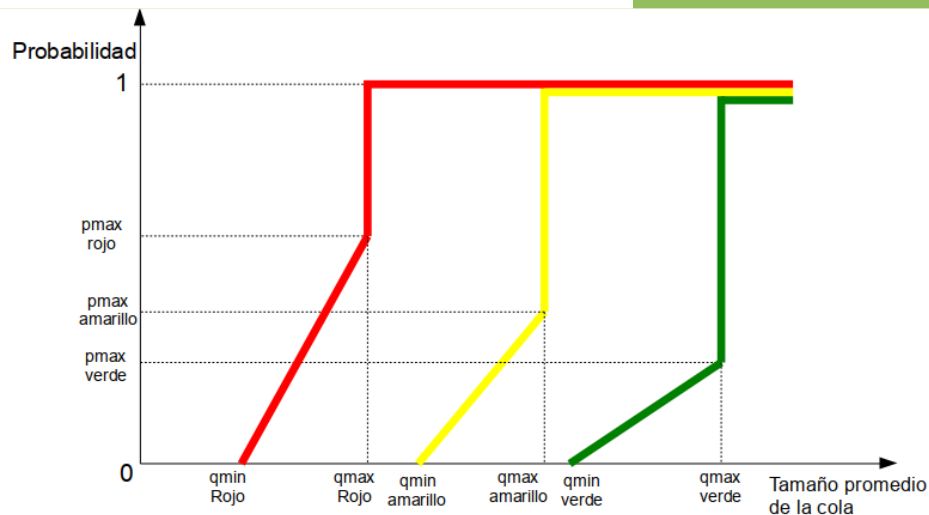
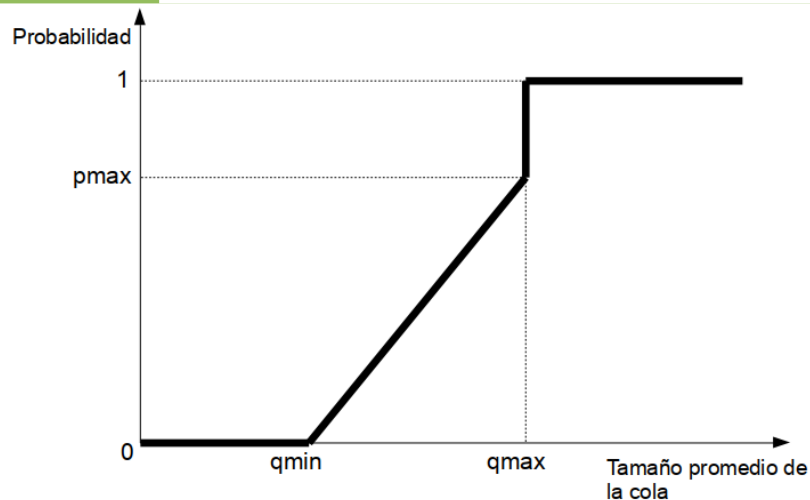
TAIL DROP. Cuando se supera un umbral, se descarta el paquete.

- Máximo delay conocido.
- Alternativa, head drop.



W. TAIL DROP. Existe tráfico con prioridad y distinto umbral para descartar.

Random Early Detection (RED) & Weighted RED



El algoritmo se basa en seguir el tamaño de la queue promedio, y a partir de allí comienza a descartar con cierta probabilidad. Esto lo hace para “avisar” a la capa superior (TCP) tempranamente de la congestión al descartar algunos paquetes para aprovechar mejor la tasa de transferencia.

¿Existe otra forma de anticiparse a la congestión?



Detección de congestión

Implícita: Descarte de paquetes para que el emisor infiera que existe congestión (RED).

Explícita: Notificar explícitamente situación de congestión mediante el marcado de paquetes.

- En IP: ECN - “Explicit Congestion Notification”
- En “colaboración” con protocolos de nivel superior (Transporte/aplicación)

Detección de congestión

ECN Explicit Congestion Notification

RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP

Campo ECN en encabezado IP:

- Bit ECT: ECN capable Transport
- Bit CE: Congestion Experienced

Offsets	Octet	0						1								2								3									
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL			DSCP						ECN		Total Length																
4	32	Identification																Flags		Fragment Offset													
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Flags en encabezado TCP:

- NS: Nonce Sum (RFC 3540 - Exp)
- ECE: ECN-Echo
- CWR: Congestion Window Reduced

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			N	C	E	U	A	P	R	S	F	Window Size															
									S	W	C	R	C	S	S	Y	I																
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															

Detección de congestión

ECN

Explicit Congestion Notification

En UDP propiamente dicho imposible por su naturaleza, pero si en capa superior:

- RFC 6679: Explicit Congestion Notification (ECN) for RTP over UDP
- Debe ser negociada y soportada en la aplicación que lo utilice, (ej. SIP/SDP, WebRTC), u otro transporte (ej. QUIC)

En DCCP Datagram Congestion Control Protocol

- RFC 4340: Conexiones unicast bidireccionales de datagramas no fiables con control de congestión

Implementación de QoS

- En dispositivos intermedios: casi invariablemente en routers IP, switches ATM o dispositivos MPLS.
- En Linux, mediante el módulo del Kernel llamado “tc” (Traffic Control) y una [herramienta de control de línea de comandos](#) homónima.
- Viene “por defecto” en la mayoría de los sistemas.

Bibliografía

- EVANS, J., FILSFILS, C., 2007, Deploying IP and MPLS QoS for Multiservice Networks: Theory & Practice. Morgan Kaufmann.
 - Capítulo 1. “QOS Requirements and Service Level Agreements”
 - Capítulo 2. “Introduction to QOS Mechanics and Architectures”
- MEDHI, D., RAMASAMY, K., 2007, Network Routing Algorithms Protocols and Architectures. Morgan Kaufmann.
 - Capítulo 14. “Router Architectures”
 - Capítulo 22. “Packet Queuing and Scheduling”
 - Capítulo 23. “Traffic Conditioning”