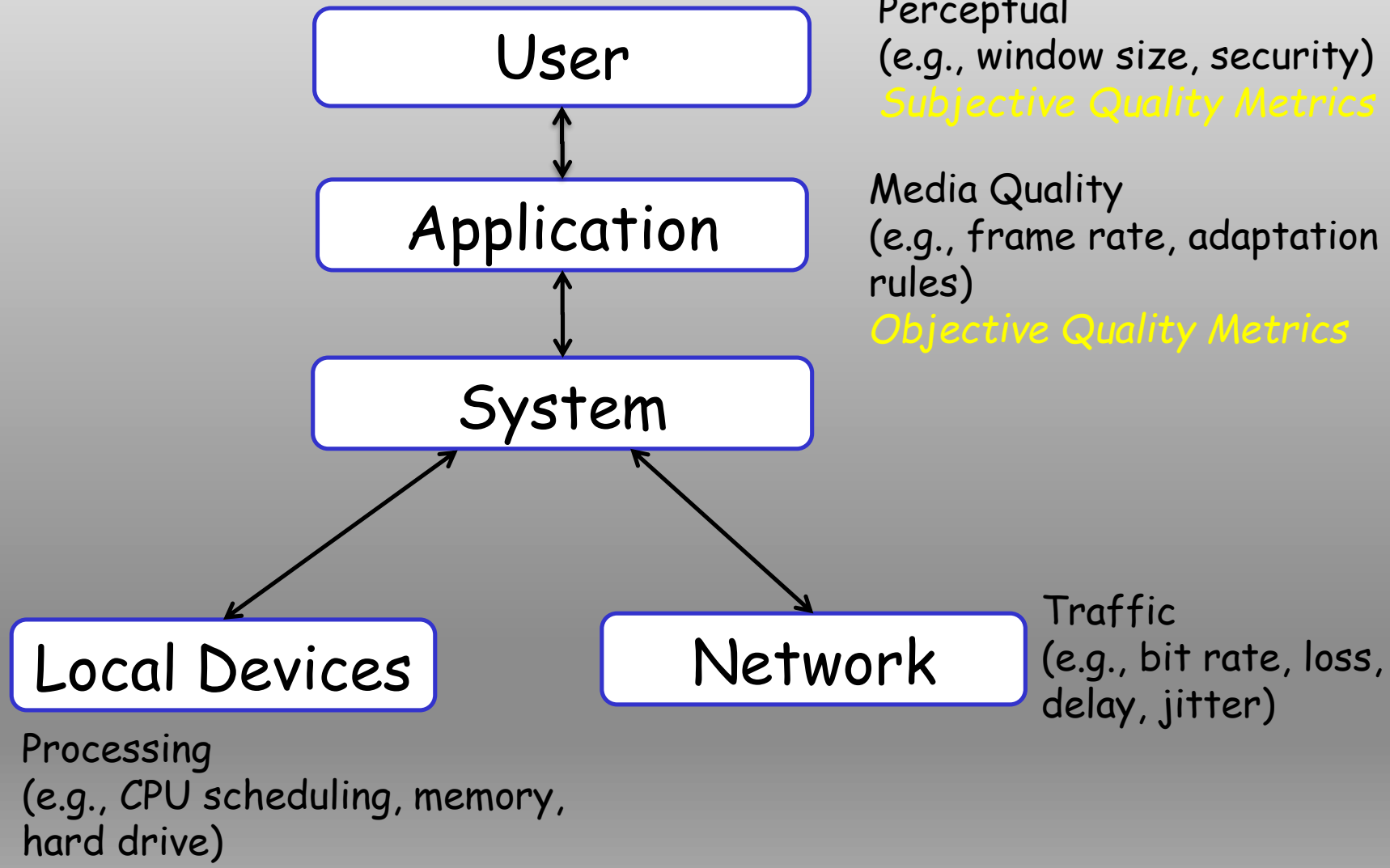
Chapter 2

Quality of Service (QoS)

QoS Support in IP Networks

- Need for QoS
- Principles
- IntServ
- DiffServ

QoS Layers


User
Application
System
Local Devices
Network
Perceptual
(e.g., window size, security)
Subjective Quality Metrics
Media Quality
(e.g., frame rate, adaptation rules)
Objective Quality Metrics
Traffic
(e.g., bit rate, loss, delay, jitter)
Processing
(e.g., CPU scheduling, memory, hard drive)

QoS Support in IP Networks

- Need for QoS
- Principles
- IntServ
- DiffServ

The Need for QoS

- Assigning different levels of priority to different IP traffic flows and provides special treatment to higher-priority IP traffic flows.
- For higher-priority IP traffic flows:
  - Reduces packet loss during times of network congestion
  - Helps control delay (latency) and delay variation (jitter)
- For low-priority IP traffic flows:
  - Provides a best-effort delivery service.
- Used mechanisms:
  - Classification and marking, policing and shaping, congestion management and avoidance.

Causes and Results of Quality Issues

When packets are delivered using a best-effort delivery model, they may not arrive in order or in a timely manner, and they may be dropped.

- For video, this can result in pixelization of the image, pausing, choppy video, audio and video being out of sync, or no video at all.
- For audio, it could cause echo, talker overlap (a walkie-talkie effect where only one person can speak at a time), unintelligible and distorted speech, voice breakups, long silence gaps, and call drops.

The following are the leading causes of quality issues:

- Lack of bandwidth
- Latency and jitter
- Packet loss

Lack of Bandwidth

The available bandwidth on the data path from a source to a destination equals the capacity of the lowest-bandwidth link.

When the maximum capacity of the lowest-bandwidth link is surpassed, link congestion takes place, resulting in traffic drops.

The solution to this type of problem:

- Increase the link bandwidth capacity
- Implement QoS mechanisms such as policing and queueing to prioritize traffic according to level of importance.
  - Voice, video, and business-critical traffic should get prioritized forwarding and sufficient bandwidth to support their application requirements.
  - The least important traffic should be allocated the remaining bandwidth.

Latency and Jitter

One-way end-to-end delay, also known as network latency, is the time it takes for packets to travel across a network from a source to a destination.

Regardless of the application type, ITU Recommendation G.114 recommends:

- A network latency of 400 ms should not be exceeded,
- For real-time traffic, network latency should be less than 150 ms;
- ITU demonstrated that real-time traffic quality does not begin to significantly degrade until network latency exceeds 200 ms.

Network latency can be broken down into fixed and variable latency:

- Propagation delay (fixed)
- Serialization delay (fixed)
- Processing delay (fixed)
- Delay variation (variable)

Serialization Delay/Processing Delay

Serialization delay is the time it takes to place all the bits of a packet onto a link.

- It is a fixed value that depends on the link speed; the higher the link speed, the lower the delay.
- The serialization delay s is equal to the packet size in bits divided by the line speed in bits per second.

Processing delay is the fixed amount of time it takes for a networking device to take the packet from an input interface and place the packet onto the output queue of the output interface.

- The processing delay depends on factors such as the following:
  - CPU speed (for software-based platforms)
  - Router architecture (centralized or distributed)
  - Configured features on both input and output interfaces

Delay Variation/Packet Loss

Delay variation / jitter is the difference in the latency between packets in a single flow.

Experienced due to the queueing delay experienced during periods of network congestion.

Packet loss is usually a result of congestion on an interface, and can be prevented by implementing one of the following approaches:

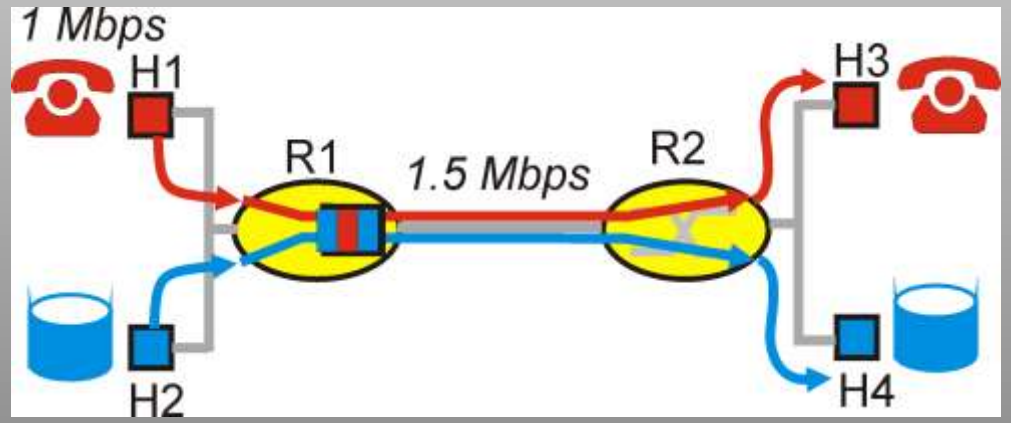
- Increase link speed.
- Implement QoS congestion-avoidance and congestion-management mechanism.
- Implement traffic policing to drop low-priority packets and allow high-priority traffic through.
- Implement traffic shaping to delay packets instead of dropping them since traffic may burst and exceed the capacity of an interface buffer.
  - ✓ Traffic shaping is not recommended for real-time traffic because it relies on queuing that can cause jitter.

QoS Support in IP Networks

- Need for QoS
- Principles
- IntServ
- DiffServ

Principles for QoS Guarantees

- Let us explore these functions using a simple example
  - 1Mbps IP phone, FTP share 1.5 Mbps link.
  - bursts of FTP can congest router, cause audio loss
  - want to give priority to audio over FTP

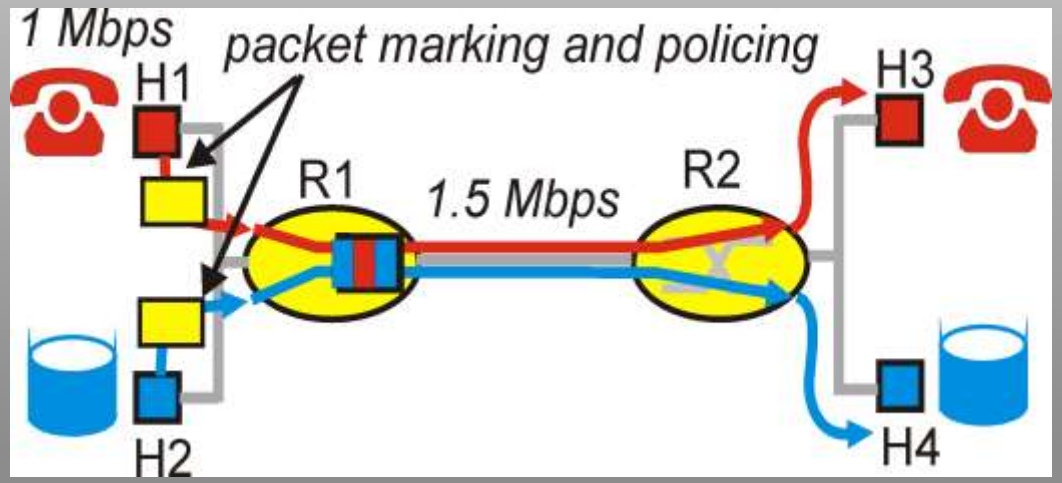


Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

Principles for QoS Guarantees (more)

- what if applications misbehave (audio sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- marking and policing at network edge:

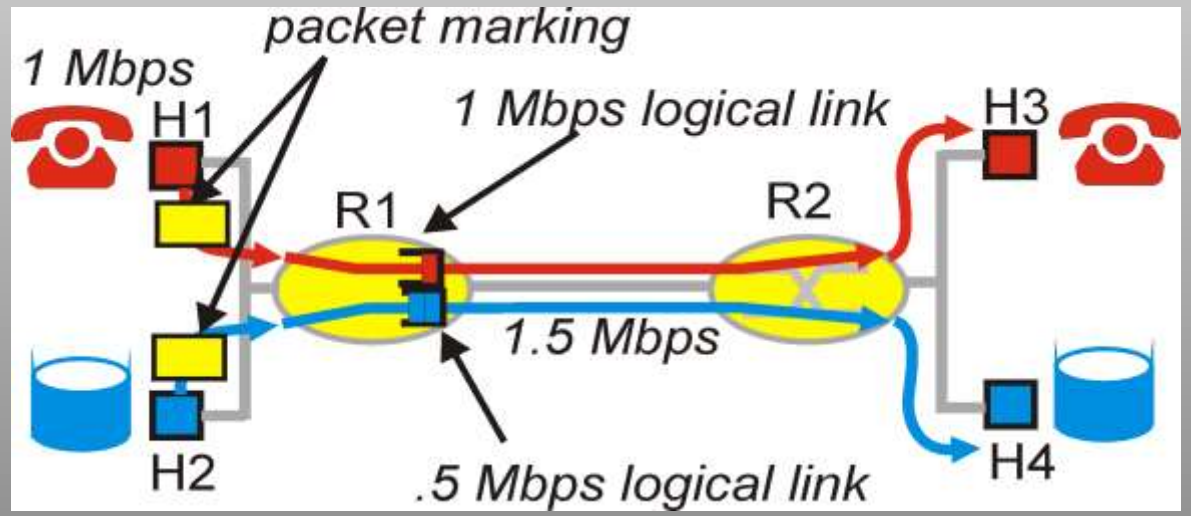


Principle 2

provide protection (*isolation*) for one class from others

Principles for QoS Guarantees (more)

- Allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flows doesn't use its allocation

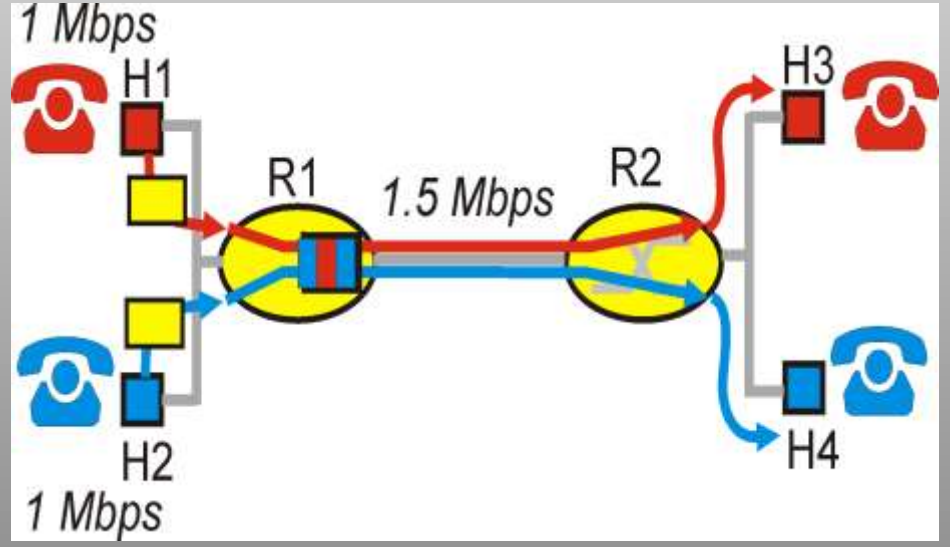


Principle 3

While providing isolation, it is desirable to use resources as efficiently as possible

Principles for QoS Guarantees (more)

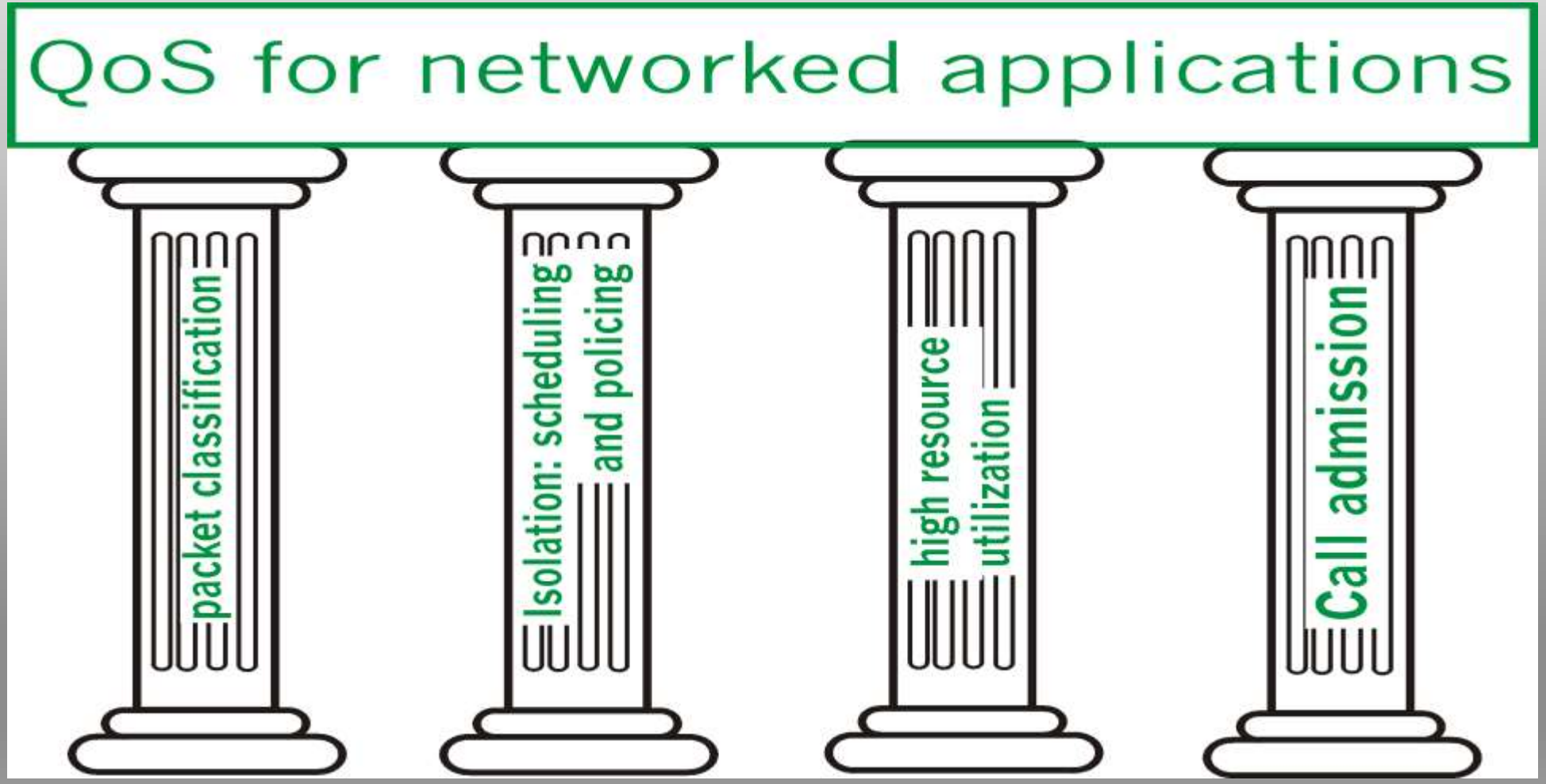
- *Basic fact of life:* can not support traffic demands beyond link capacity



Principle 4

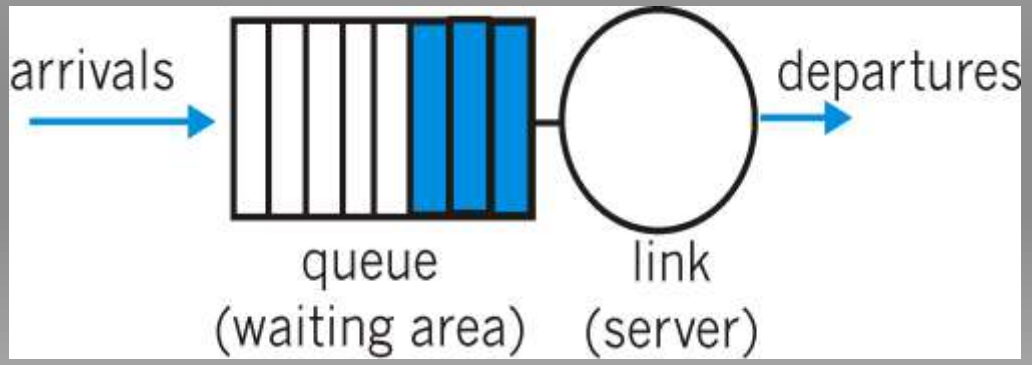
Call Admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

Summary of QoS Principles



Scheduling And Policing Mechanisms

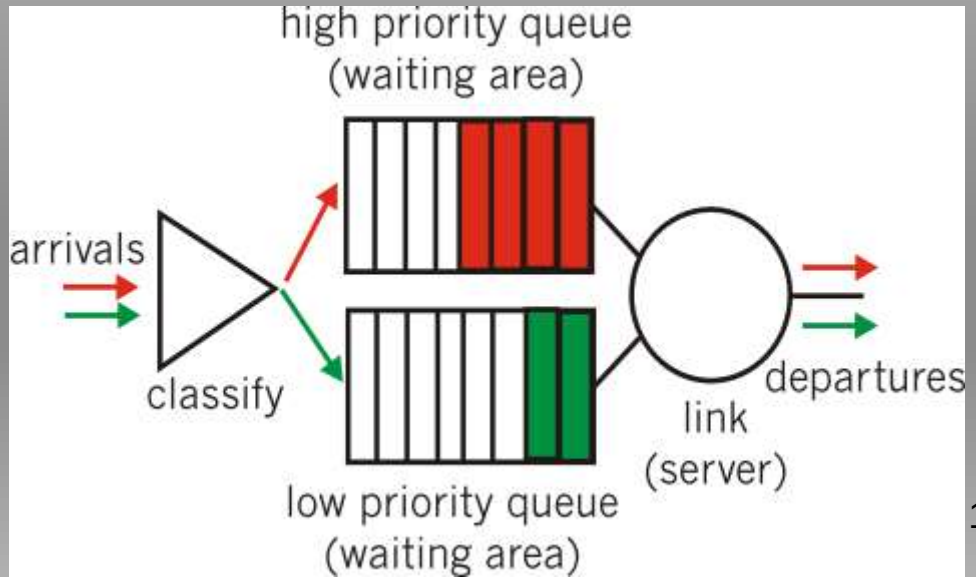
- Scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
  - Discard policy: if packet arrives to full queue: who to discard?
    - Tail drop: drop arriving packet
    - Priority: drop/remove on priority basis
    - Random: drop/remove randomly



Scheduling Policies

Priority scheduling: transmit highest-priority queued packet

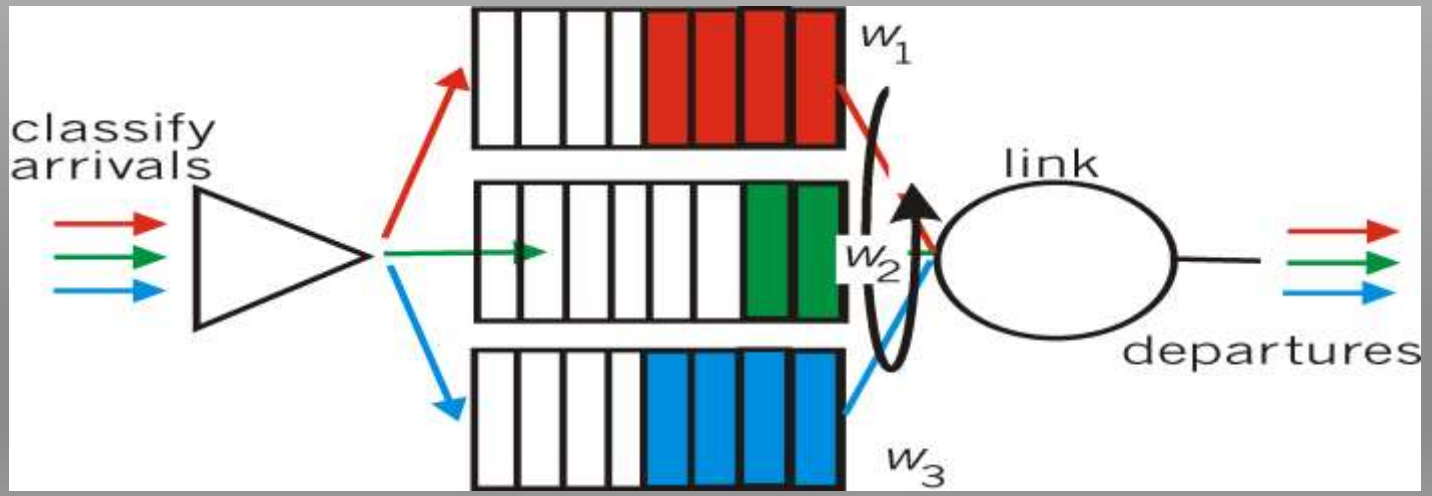
- Multiple *classes*, with different priorities
  - Class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..



Scheduling Policies

Weighted Fair Queuing:

- Generalized Round Robin
- Each class gets weighted amount of service in each cycle



Policing Mechanisms

Goal: limit traffic to not exceed declared parameters

Three common-used criteria:

- *(Long term) Average Rate:* how many pkts can be sent per unit time
  - Crucial question: what is the interval length: 100 packets per sec and 6000 packets per min (ppm) have same average!
- *Peak Rate:* e.g.,
  - Avg rate: 6000 ppm
  - Peak rate: 25 pps (= 1500 ppm)
- *(Max.) Burst Size:* max. number of pkts sent consecutively (with no intervening idle)

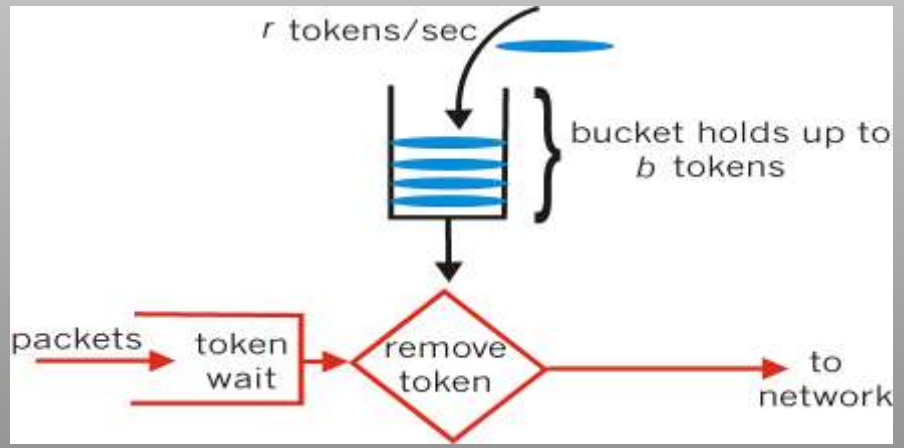
Putting it all together

- Average Rate = long-term traffic load (e.g., 6000 ppm).
- Peak Rate = maximum instantaneous sending speed (e.g., 25 pps).
- Burst Size = how many packets can be sent consecutively before you must slow down.

Policing Mechanisms

Leaky Bucket:

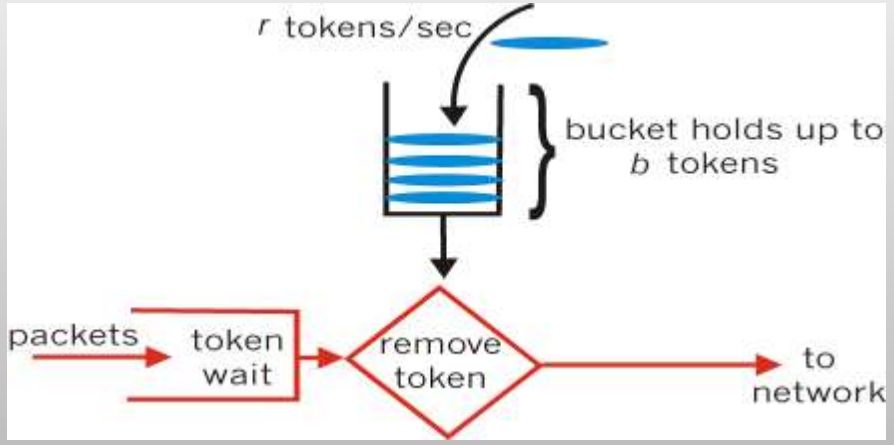
- traffic shaping / policing mechanism
- limit input to specified Burst Size and Average Rate.



If no tokens are available, the packet must ***wait*** (shaping) or is ***dropped*** (policing).

- Bucket can hold b tokens (Represents the burst size)
- Tokens generated at rate *r token/sec* unless bucket full (represents the average rate)
- Over interval of length t: number of packets admitted less than or equal to (r t + b).

Policing Mechanisms



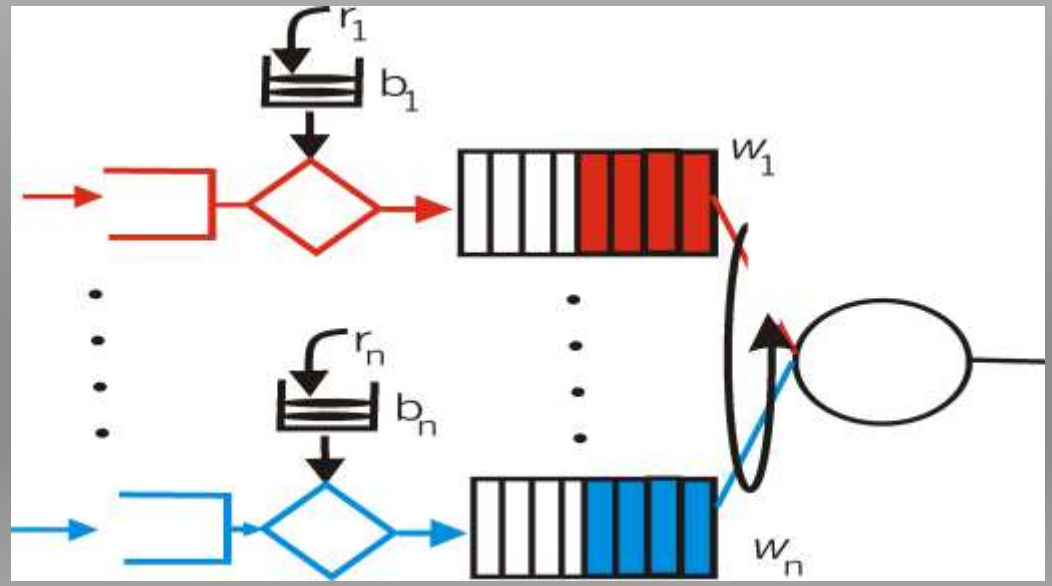
Leaky Bucket / Example:

- Suppose:
  - Bucket size b=10 tokens.
  - Token generation rate r=5 tokens/sec.
- Then:
  - At time t=0, bucket is full → *10 packets can be sent immediately* (burst).
  - After that, only *5 packets per second* can be sent (because only 5 new tokens arrive each second).
  - In any interval of t seconds, you can't send more than:

$$r \cdot t + b = 5t + 10$$

Policing Mechanisms

- Leaky bucket + WFQ ➔ provide guaranteed upper bound on delay, i.e., *QoS guarantee!* How?
  - WFQ: guaranteed share of bandwidth
  - Leaky bucket: limit max number of packets in queue (burst)



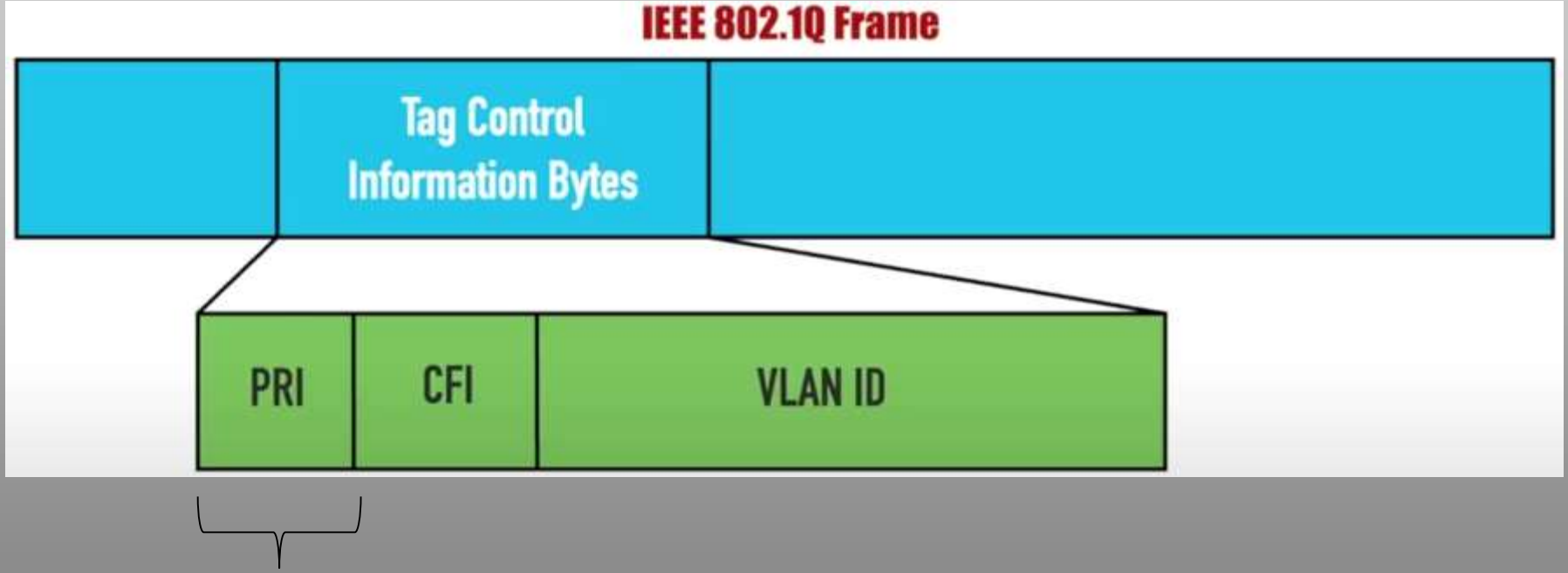
QoS Models

QoS is not enabled for this model. It is used for traffic that does not require any special treatment

The network identifies classes that require special QoS treatment.

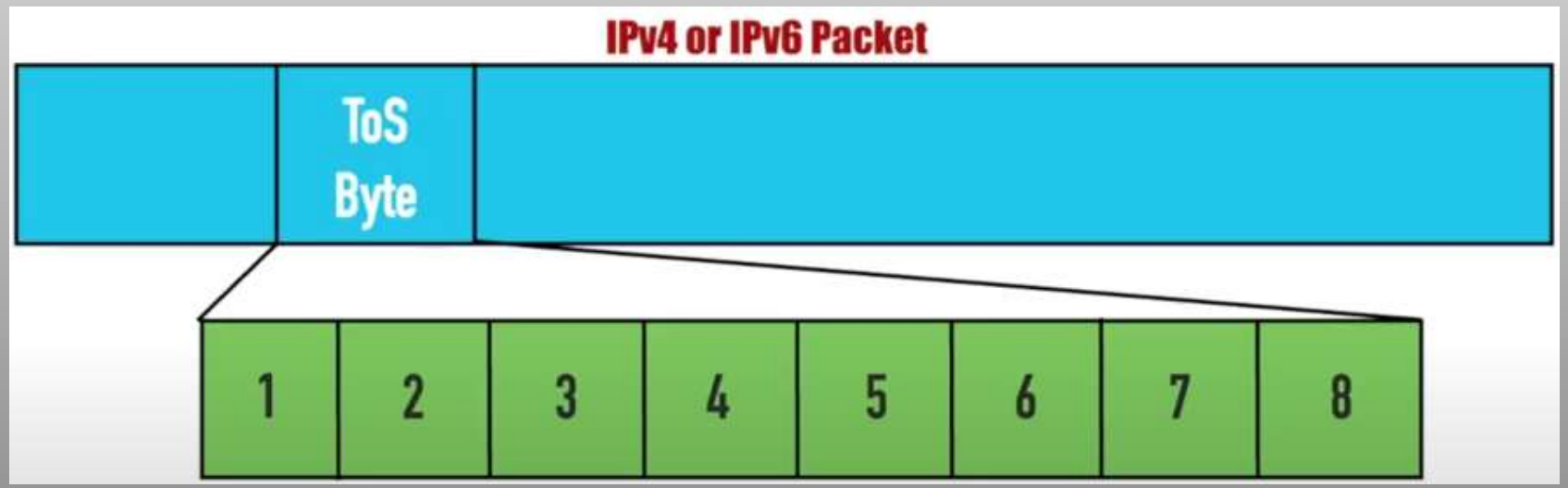
Applications signal the network to make a bandwidth reservation and to indicate that they require special QoS treatment.

Class of Service (CoS)



Type of Service (ToS) byte

Traffic Class in IPv6



IP precedence = priority

0-5 for user applications
6-7 for network control traffic

DSCP

QoS Support in IP Networks

- Need for QoS
- Principles
- IntServ
- DiffServ

IETF Integrated Services (IntServ)

- architecture for providing QoS guarantees in IP networks for individual application sessions (flow)
- resource reservation: routers maintain per flow state info of allocated resources, QoS req's
- admit/deny new call setup requests

Components of IntServ

- Type of Service Model
  - What does the network promise?
- Service Interface
  - How does the application describe what it wants?
- Packet Scheduling
  - How does the network meet promises?
- Establishing the guarantee
  - How is the promise communicated to/from the network?
  - How is admission of new applications controlled?

Service Models Provided by IntServ

Service Types	Guaranteed	Control Load	Best Effort
Provide QoS	• Guaranteed BW • End-to-end Delay Bound	• Emulate a lightly loaded network	None
RFC	RFC 2212	RFC 2211	None

IntServ – Guaranteed Service

- Targets hard real-time applications
- User specifies traffic characteristics and a service requirement.
- The network should guarantee that the delay experienced < specified maximum value
- Requires admission control at each of the routers.
- Can mathematically guarantee bandwidth, delay and jitter.

IntServ – Controlled Load

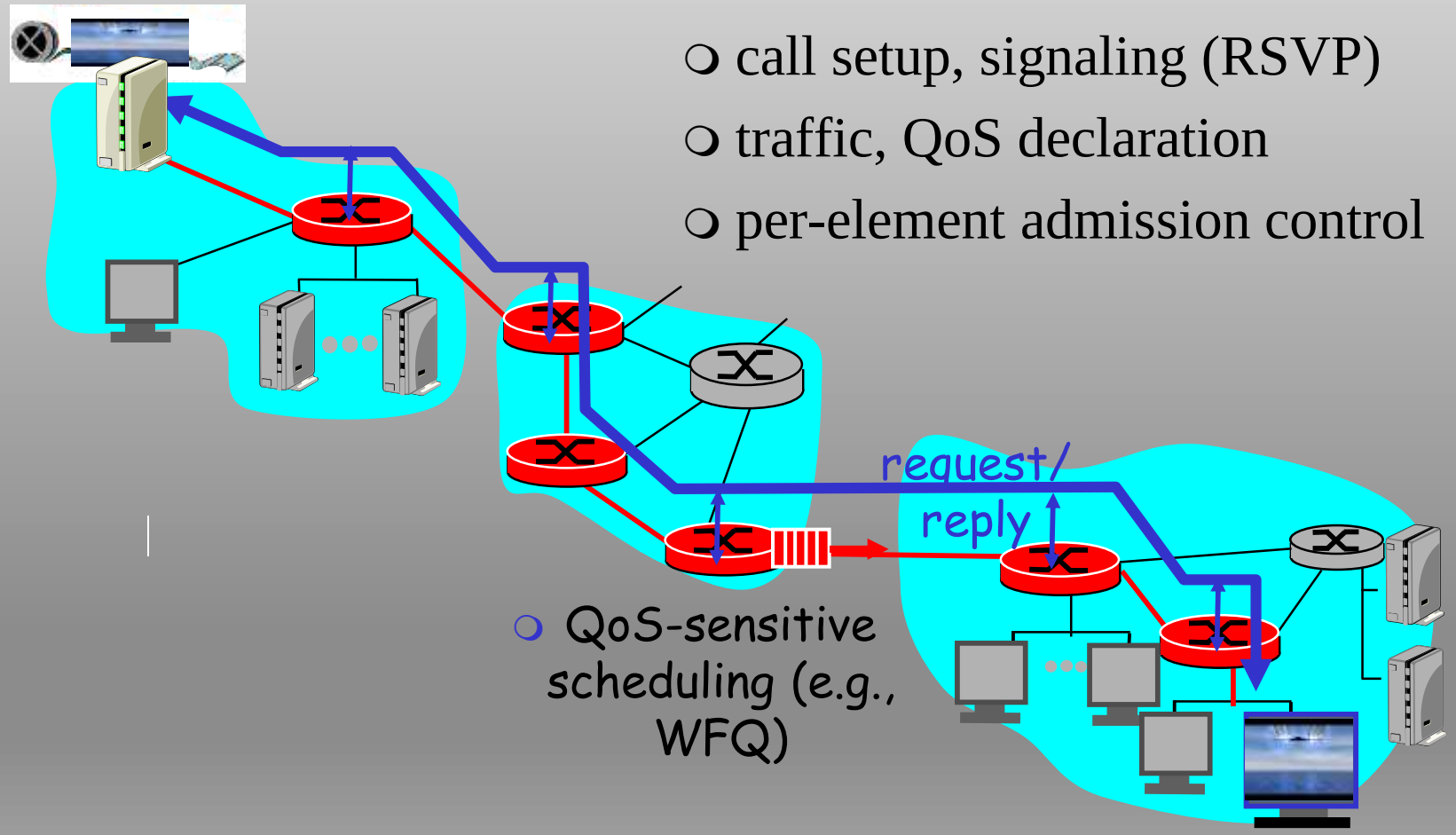
- Emulate a lightly loaded network even though the network as a whole may in fact be heavily loaded.
- It does not guarantee strict delay bounds (like the Guaranteed Service model), but it tries to ensure that packets experience minimal delay and loss.
- Targets applications that can adapt to network conditions within a certain performance window.
- Use specifies traffic characteristics and bandwidth.
- Requires admission control at each of the routers.

Controlled Load vs Guaranteed Service

Feature	Controlled Load	Guaranteed Service
Type of guarantee	Soft (best-effort under low load)	Hard (strict delay bound)
Delay sensitivity	Moderate	Very high
Applications	Video, VoIP, streaming	Industrial control, medical telemetry
Tolerance to loss	Somewhat tolerant	Very low tolerance

IntServ: QoS guarantee scenario

- Resource reservation
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control



Call Admission

Arriving session must:

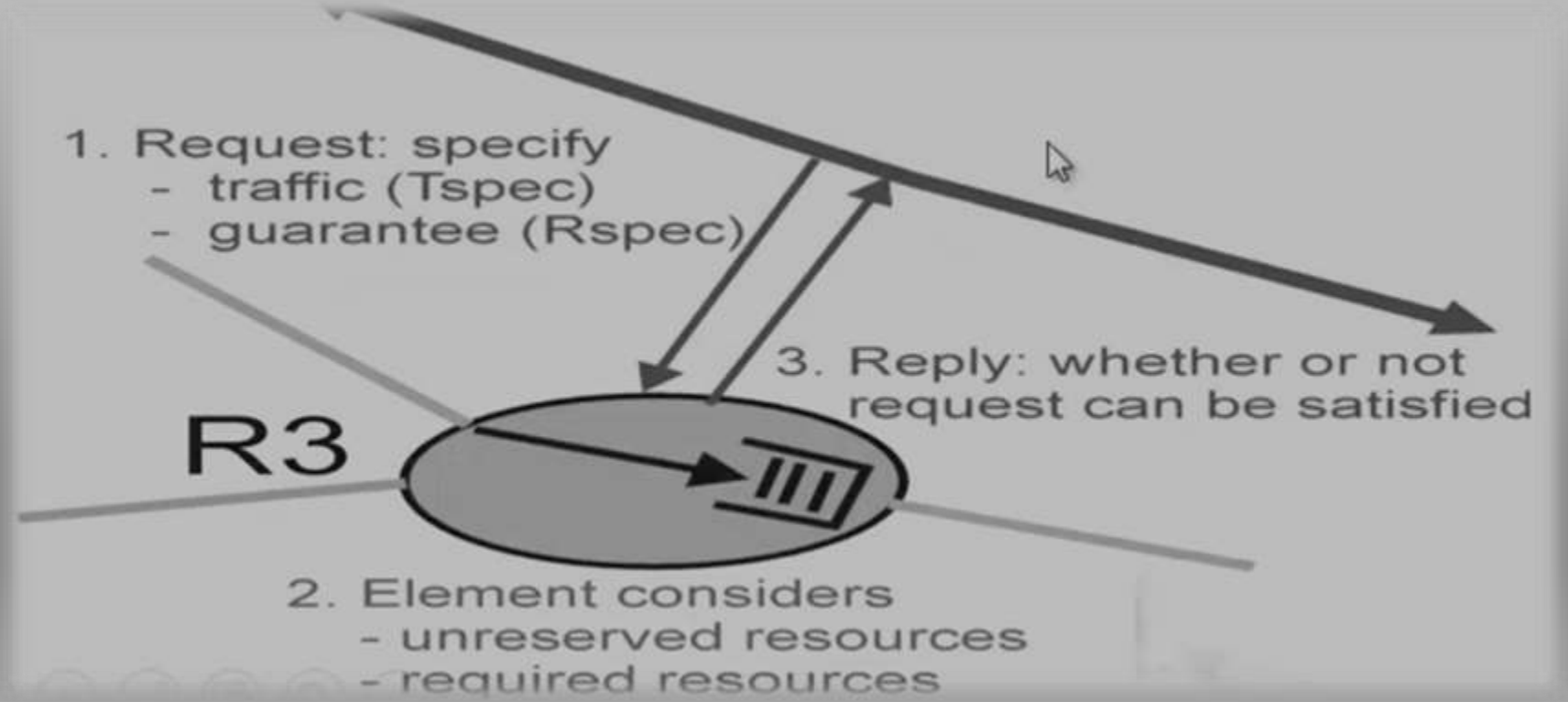
- Declare its QoS requirement
  - Reservation Specification (Rspec): defines the QoS being requested, e.g., bandwidth needed (*rate r*)
- Characterize traffic it will send into network
  - Traffic Specification (Tspec): defines traffic characteristics, e.g., average bit rate and maximum burst size (*leaky bucket with rate r and buffer size b*)
- Signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - RSVP

Packet Scheduling

- Guarantee Service
  - Use token bucket filter to characterize traffic
    - ✓Described by rate r and bucket depth b
- Use WFQ at the routers

Call Admission

- CA: routers admit call based on their R-Spec and T-Spec and the current resource allocated at the routers to other call.



Intserv Mechanisms

- Flowspecs
  - Set of information provided to the network
- Resource Reservation (RSVP)
  - Used to exchange information such as
    - ✓Requests for service,
    - ✓flowspecs,
    - ✓and admission control decisions.

Intserv - Flowspecs

- Tspec: describes the flow's traffic characteristics
- The TSpec is often modeled using a token bucket:

$$b(t) \leq b + r \times t$$

where:

- b(t) = number of bits sent in time tt
- b = bucket size (burst)
- r = token rate (average rate)

RSVP-style TSpec

```
TSpec {
  token_rate = 1 Mbps,
  bucket_size = 5000 bytes,
  peak_rate = 2 Mbps,
  min_policed_unit = 64 bytes,
  max_packet_size = 1500 bytes
}
```

Intserv - Flowspecs

- Rspec: describe the service required from the network
  - Is service specific & easy to describe
  - Contolled load service -> no parameters
  - Guaranteed service -> delay target or bound

Parameter	Description	Symbol
Reserved Rate	The rate (bandwidth) the network must reserve for this flow.	**R**
Slack Term	Extra delay allowance (margin) beyond the minimum delay — used in Guaranteed Service.	**S**
Service Type	Indicates the type of IntServ service requested (e.g., *Guaranteed* or *Controlled Load*).	—

Intserv - Flowspecs

Guaranteed Service flow

```
RSpec {
  service_type = Guaranteed,
  reserved_rate = 1 Mbps,
  slack_term = 10 ms
}
```

Controlled Load service

```
RSpec {
  service_type = Controlled Load,
  reserved_rate = 800 Kbps
}
```

Intserv – Flowspecs - RSVP

- Requirements: Connection must be connectionless (end-to-end connectivity should be maintained)
- Operates *at the Transport Layer* but interacts closely with the *Network Layer.*
  - **Full form:** Resource ReSerVation Protocol
  - **RFC:** Defined in **RFC 2205**
  - **Type:** Signaling protocol (not a routing protocol)
  - **Purpose:** To request and maintain QoS for IP data flows.

Intserv – Flowspecs - RSVP

In the IntServ model, RSVP is used to:

- Carry TSpec (Traffic Specification) from sender to receiver.
- Carry RSpec (Reservation Specification) from receiver to routers.
- Communicate with routers along the path to reserve bandwidth and buffer space.
- Maintain the reservation state as long as the application is active.

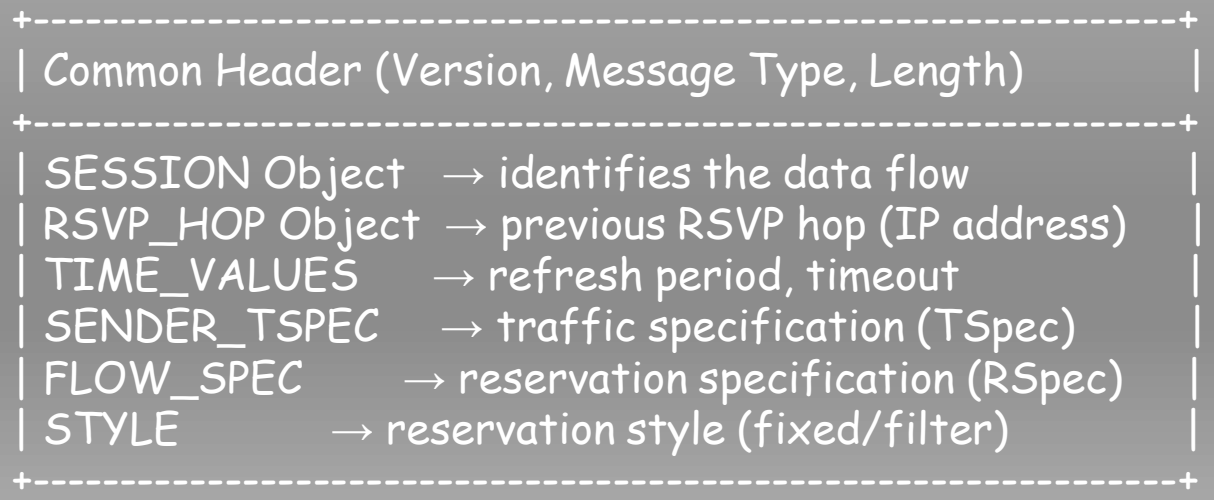
```
FlowSpec {
  TSpec {
    token_rate = 1 Mbps,
    bucket_size = 5000 bytes,
    peak_rate = 2 Mbps,
    min_policed_unit = 64 bytes,
    max_packet_size = 1500 bytes
  }
  RSpec {
    service_type = Guaranteed,
    reserved_rate = 1 Mbps,
    slack_term = 10 ms
  }
}
```

Intserv – Flowspecs – RSVP: how it works?

- Step 1 — PATH Message (from Sender → Receiver)
  - The sender sends an RSVP PATH message downstream.
  - The PATH message carries the TSpec (traffic characteristics).
  - Each router along the path stores path information for reverse communication.
- Step 2 — RESV Message (from Receiver → Sender)
  - The receiver, after receiving the PATH message, decides what QoS is needed.
  - It sends a RESV (Reservation) message upstream.
  - The RESV message carries the RSpec (resource requirements).
  - Each router checks if it has enough resources to satisfy the request:
    - If yes, it reserves them.
    - If no, it rejects the request and sends an error message.

Intserv – Flowspecs – RSVP: how it works?

- Step 3 — Data Flow
  - Once the reservation is successful, data packets can flow through the path with guaranteed QoS.
- Step 4 — Refresh and Tear Down
  - RSVP is soft state — reservations expire if not refreshed.
  - Periodic PATH and RESV messages refresh the reservation.
  - PATH TEAR and RESV TEAR messages are used to remove reservations.



OSI Layer	Role in IntServ	Involvement of TSpec / RSpec
Layer 7 – Application	Application requests QoS for a specific data flow (e.g., video streaming, VoIP).	**Applications define TSpec and RSpec**parameters and pass them to RSVP.
Layer 6 – Presentation	No direct involvement.	—
Layer 5 – Session	Manages the session and may request resource reservation for that session.	RSVP operates here conceptually to **establish and maintain reservations**.
Layer 4 – Transport	Works with RSVP to map QoS needs to transport flows (TCP/UDP ports).	RSVP messages carry TSpec/RSpec between endpoints.
Layer 3 – Network	**Main layer for IntServ and RSVP.** Routers use TSpec/RSpec to allocate bandwidth, schedule packets, and enforce QoS.	**Core layer where TSpec/RSpec are interpreted and enforced.**
Layer 2 – Data Link	Provides lower-level QoS mechanisms (e.g., priority queues, MAC scheduling).	May support the network layer's QoS but doesn't process TSpec/RSpec directly.
Layer 1 – Physical	Provides transmission capacity (bit rate, delay, etc.).	Only indirectly affects achievable QoS.

QoS Support in IP Networks

- Need for QoS
- Principles
- IntServ
- DiffServ

IETF Differentiated Services

Concerns with IntServ:

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows
- Flexible Service Models: Intserv has only two classes. Also want “qualitative” service classes
  - relative service distinction: Platinum, Gold, Silver

DiffServ approach:

- simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don’t define service classes, provide functional components to build service classes

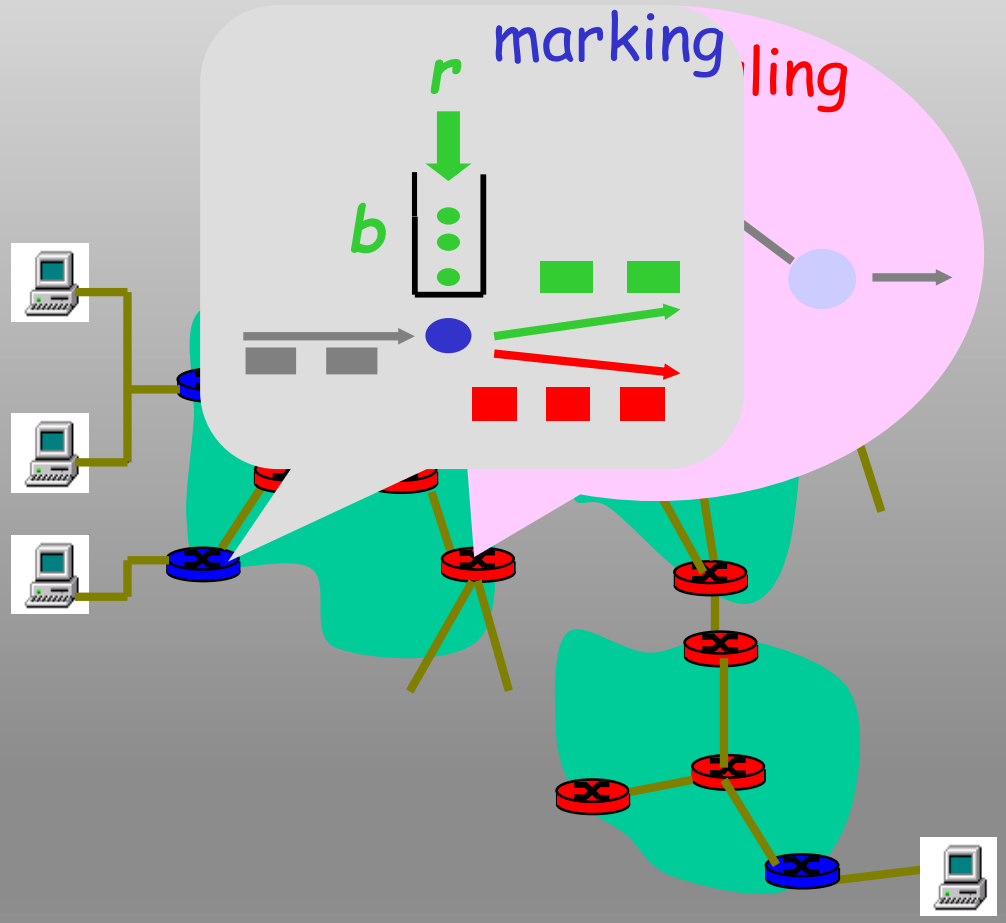
DiffServ Architecture

Edge router:

- per-flow traffic ma nagement
- Classifies (marks) pkts
  - different classes
  - within a class: in-profile and out-profile

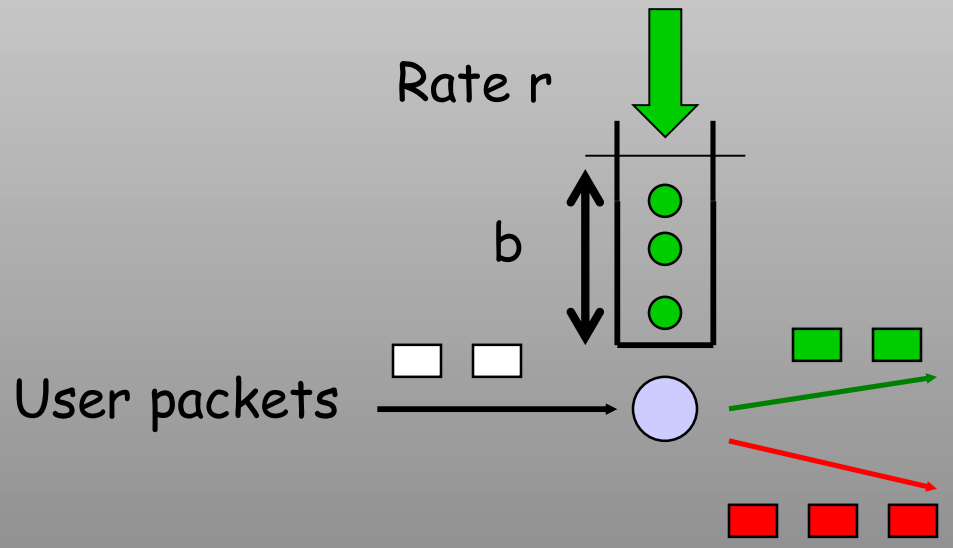
Core router:

- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets



Edge-router Packet Marking

- profile: pre-negotiated rate r, bucket size b
- packet marking at edge based on per-flow profile



Possible usage of marking:

- class-based marking: packets of different classes marked differently
- intra-class marking: conforming portion of flow marked differently than non-conforming one

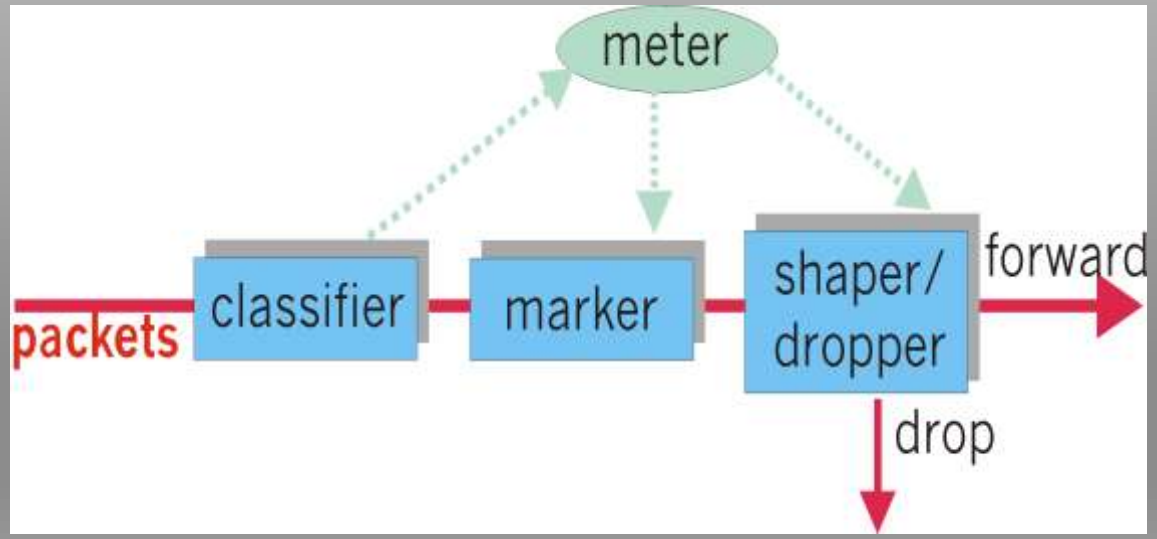
Edge-router: Classification and Conditioning

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine Per-Hop Behavior (PHB) that the packet will receive
- 2 bits are currently unused

Edge-router: Classification and Conditioning

may be desirable to limit traffic injection rate of some class:

- user declares traffic profile (e.g., rate, burst size)
- traffic metered, shaped if non-conforming



Core-router: Forwarding (PHB)

- PHB result in a different observable (measurable) forwarding performance *behavior*
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B

Core-router: Forwarding (PHB)

PHBs being developed:

- Expedited Forwarding (EF): pkt departure rate of a class equals or exceeds specified rate
  - Logical link with a minimum guaranteed rate
  - May require edge routers to limit EF traffic rate
  - Could be implemented using strict priority scheduling or WFQ with higher weight for EF traffic
- Assured Forwarding (AF): multiple traffic classes, treated differently
  - Amount of bandwidth allocated, or drop priorities
  - Can be implemented using WFQ + leaky bucket or RED (Random Early Detection) with different threshold values.
    - See Sections 6.4.2 and 6.5.3 in [Peterson and Davie 07]

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000
Expedited Forwarding (EF)	46	101110

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000
Expedited Forwarding (EF)	46	101110

Same as IP precedence bits

101 = 5
The highest precedence value in IPv4

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000
Expedited Forwarding (EF)	46	101110
Class Selector 1 (CS1)	8	001000

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000
Expedited Forwarding (EF)	46	101110
Class Selector 1 (CS1)	8	001000
Class Selector 2 (CS2)	16	010000

Core-router: Forwarding (PHB)

NAME (PHB)	Decimal Value	Binary Value
Default	0	000000
Expedited Forwarding (EF)	46	101110
Class Selector 1 (CS1)	8	001000
Class Selector 2 (CS2)	16	010000
Class Selector 3 (CS3)	24	011000
Class Selector 4 (CS4)	32	100000
Class Selector 5 (CS5)	40	101000
Class Selector 6 (CS6)	48	110000
Class Selector 7 (CS7)	56	111000

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 001100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

DSCP Assured Forwarding

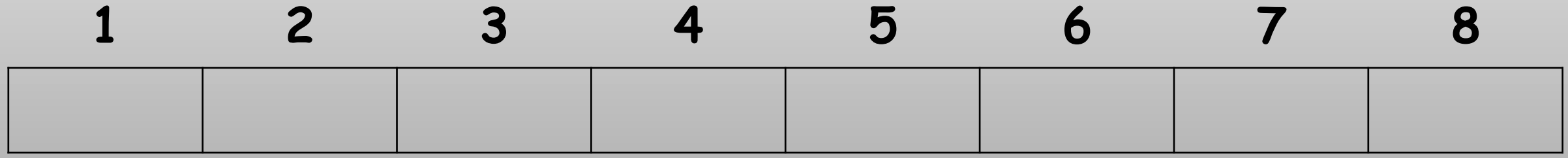
	Low Drop Probability	Medium Drop Probability	High Drop Probability
Class 1	AF11 (10) 001010	AF12 (12) 00100	AF13 (14) 001110
Class 2	AF21 (18) 010010	AF22 (20) 010100	AF23 (22) 010110
Class 3	AF31 (26) 011010	AF32 (28) 011100	AF33 (30) 011110
Class 4	AF41 (34) 100010	AF42 (36) 100100	AF43 (38) 100110

Question

A DSCP PHB name of CS4 has what equivalent decimal value?

Answer

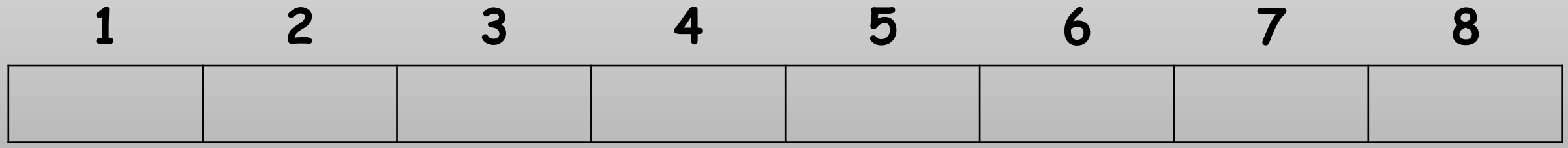
Type of Service (ToS) Byte



CS4

Answer

Type of Service (ToS) Byte



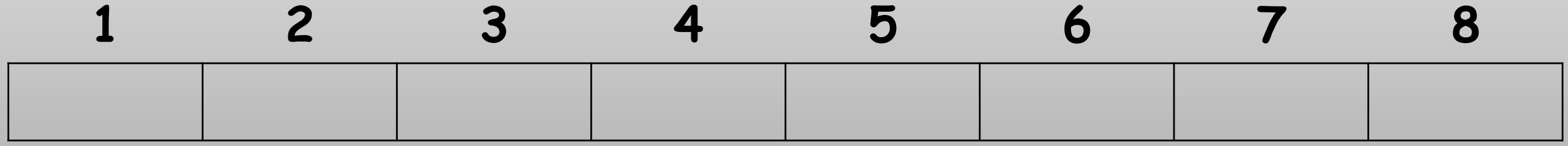
IP Precedence

4	2	1
1	0	0

CS4

Answer

Type of Service (ToS) Byte



IP Precedence

4	2	1
1	0	0

CS4

DSCP

32	16	8	4	2	1
1	0	0	0	0	0

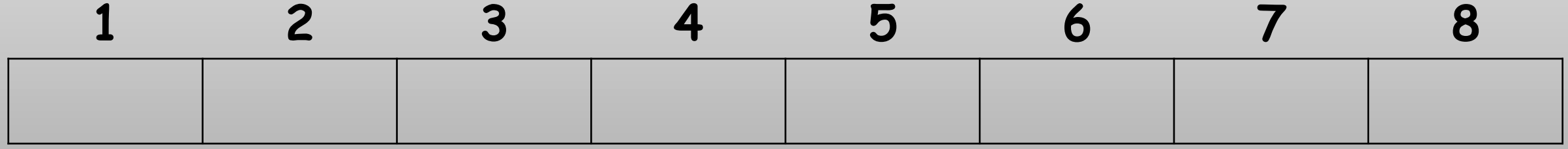
= 32

Question

A DSCP decimal value of 20 has what equivalent PHB name?

Answer

Type of Service (ToS) Byte



Answer

Type of Service (ToS) Byte

1	2	3	4	5	6	7	8

DSCP (Decimal Value = 20)

32	16	8	4	2	1
0	1	0	1	0	0

Answer

Type of Service (ToS) Byte

1	2	3	4	5	6	7	8

DSCP (Decimal Value = 20)

32	16	8	4	2	1
0	1	0	1	0	0

4	2	1	2	1	Always zero
					0

Answer

Type of Service (ToS) Byte

1	2	3	4	5	6	7	8

DSCP (Decimal Value = 20)

32	16	8	4	2	1
0	1	0	1	0	0

4	2	1	2	1	Always zero
0	1	0			0

Answer

Type of Service (ToS) Byte

1	2	3	4	5	6	7	8

DSCP (Decimal Value = 20)

32	16	8	4	2	1
0	1	0	1	0	0

4	2	1	2	1	Always zero
0	1	0	1	0	0

A Decimal Value of 20 has a DSCP Value of AF22