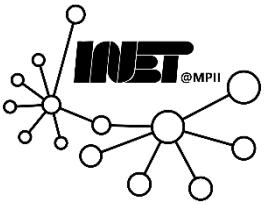
# Quality of Service (QoS)

Prof. Anja Feldmann, Ph.D.

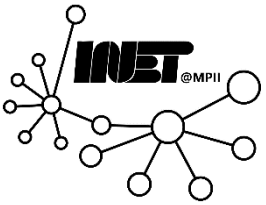
Balakrishnan Chandrasekaran, Ph.D.

(Based on slide deck of Computer Networking, 7th ed., Jim Kurose and Keith Ross.)



# Network support for multimedia

| Approach | Granularity | Guarantee | Mechanisms | Complex | Deployed? |
|---|---|---|---|---|---|
| Making best of best effort service | All traffic treated equally | None or soft | No network support (all at application) | low | everywhere |
| Differentiated service | Traffic "class" | None of soft | Packet market, scheduling, policing. | med | some |
| Per-connection QoS | Per-connection flow | Soft or hard after flow admitted | Packet market, scheduling, policing, call admission | high | little to none |



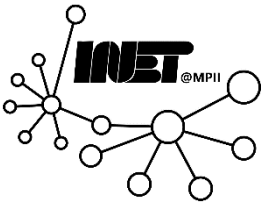
# Dimensioning best-effort networks

***Approach***: Deploy *enough* link capacity so that congestion does not occur, and multimedia traffic flows without delay or loss

- Low complexity of network mechanisms (use current *"best effort"* network)
- High bandwidth costs

## *Challenges*

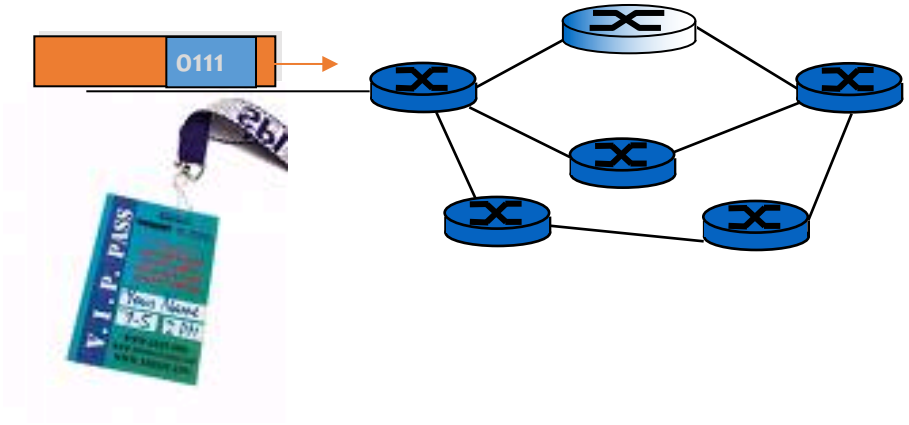
- Network dimensioning: *How much bandwidth is "enough?"*
- Estimating network traffic demand: Required to determine how much bandwidth is "enough" (for that much traffic)



# Providing multiple classes of service

Thus far: *making the best of best effort service*

- *"One-size-fits-all"* service model




***Alternative***: *multiple classes of service*

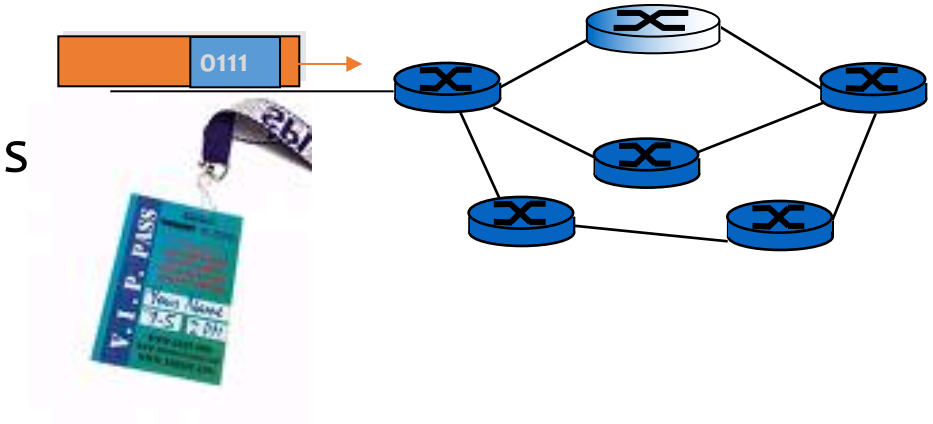
- Partition traffic into *classes*
- *Network treats different classes of traffic differently*
  - *Analogy: VIP service versus regular service*

# Providing multiple classes of service

***Alternative**: multiple classes of service*

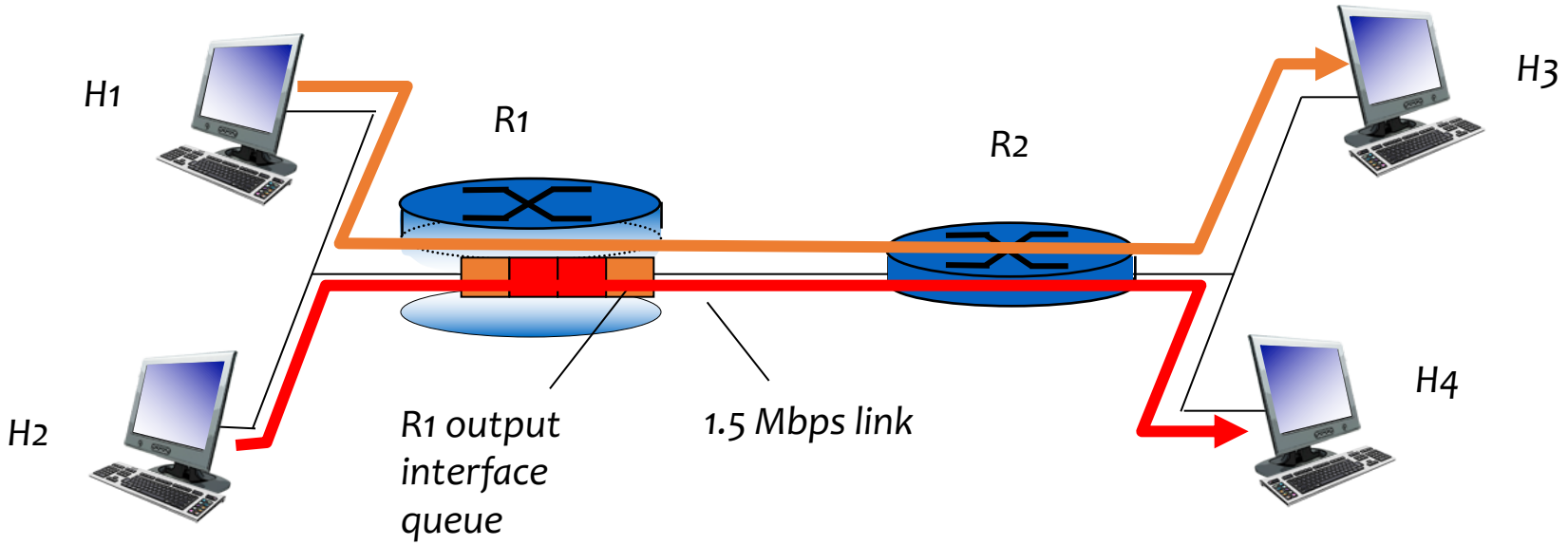
***Granularity***

- Differential service among multiple classes not among individual connections




***History**: **ToS** bits*

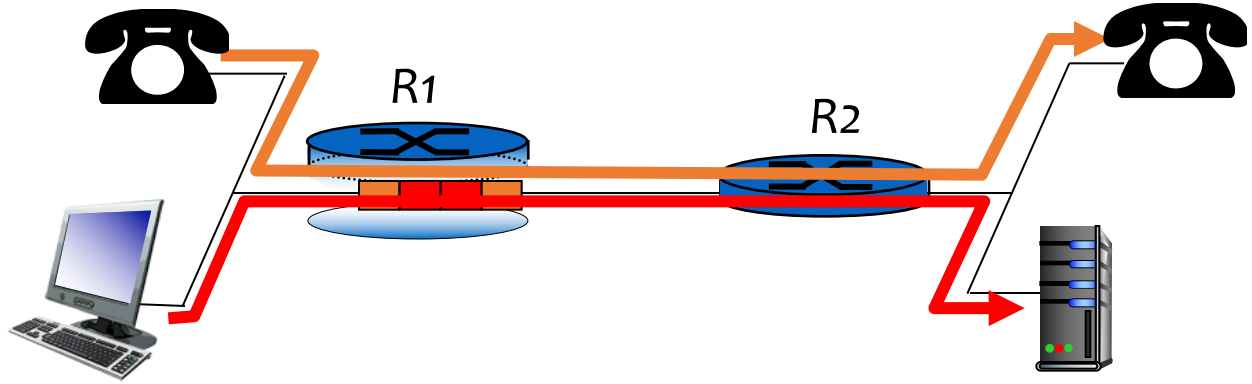
# Multiple service classes: Scenario



# Scenario 1: Mix of HTTP & VoIP

***Example**: 1 Mbps VoIP, HTTP share 1.5 Mbps link*

- HTTP *bursts* can *congest* router, cause audio *loss*
- Ideally: Give *priority* to audio over HTTP




## *Principle 1*

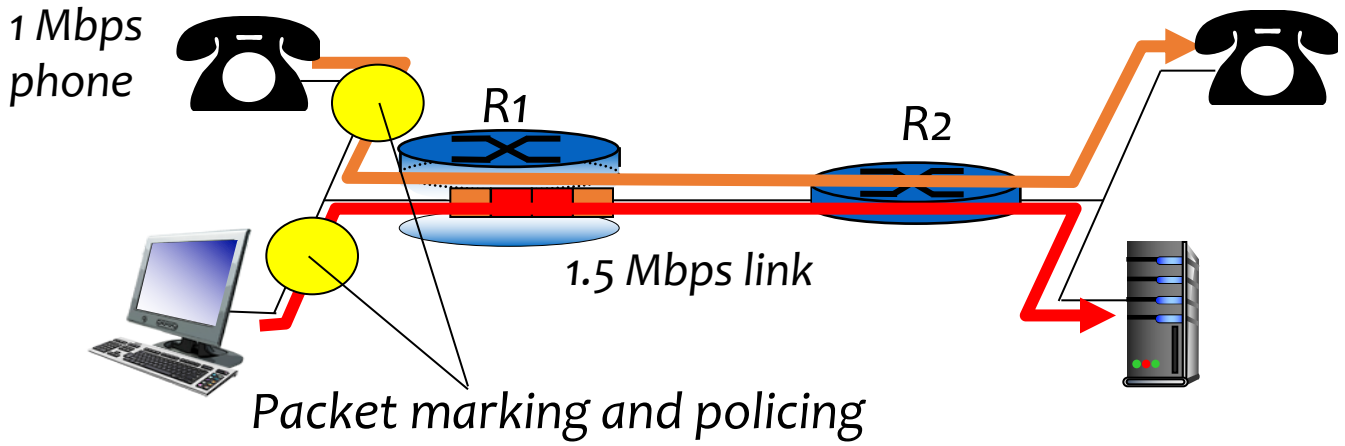
Packet marking needed for router to distinguish between different classes, and new router policy to treat packets accordingly.

# Principles for QoS guarantees

*What if applications misbehave (VoIP sends higher than declared rate)?*

- ***Policing***: Force source adherence to bandwidth allocations

- Marking, policing at network *edge*



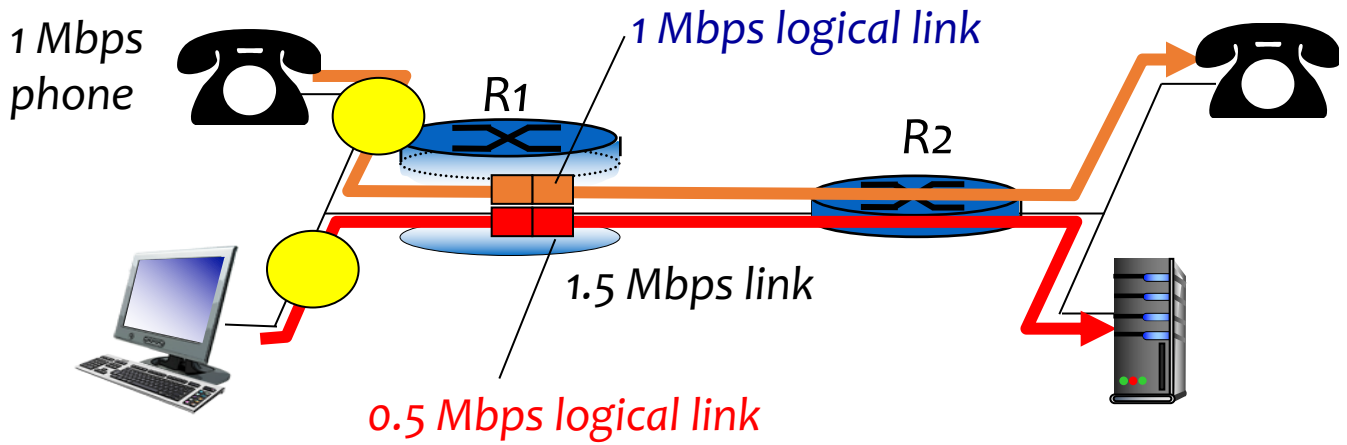

## *Principle 2*

Provide protection (or isolation) for one class from others.

# Principles for QoS guarantees

Allocating *fixed (non-sharable)* bandwidth to flow

- *Inefficient* if flows do not use allocated bandwidth



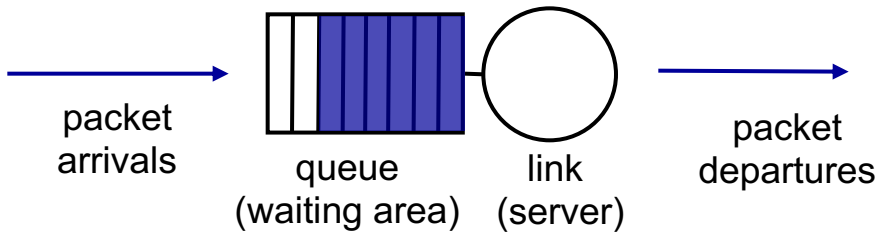

## *Principle 3*

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

# Scheduling Mechanisms

## *Packet scheduling*

- Choose next queued packet to send on outgoing link




- *FCFS: first come first served*
- *Simply multi-class priority*
- *Round robin*
- *Weighted fair queueing (WFQ)*

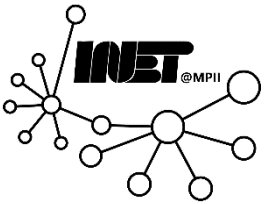
# Policing Mechanisms

## *Goal*

- Limit traffic to not exceed declared parameters

## *Three common-used criteria*

- ***(long term) average rate**: How many packets can be sent per unit time (in the long run)?*
  - *Crucial question: What is the interval length? 100 packets/s or 6000 packets/min have same average!*
- ***Peak rate**: e.g., 6000 packets/min (ppm) avg.; 1500 ppm peak rate*
- ***(max.) burst size**: Max. number of packets sent consecutively (with no intervening idle)*



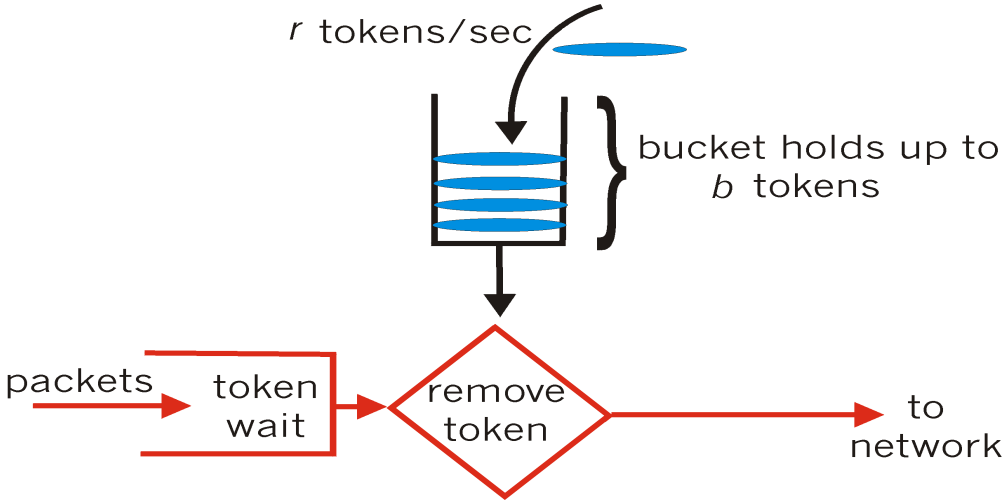
# Policing Mechanisms: Implementation

## *Token bucket*

- *Limit input to specified burst size and average rate*

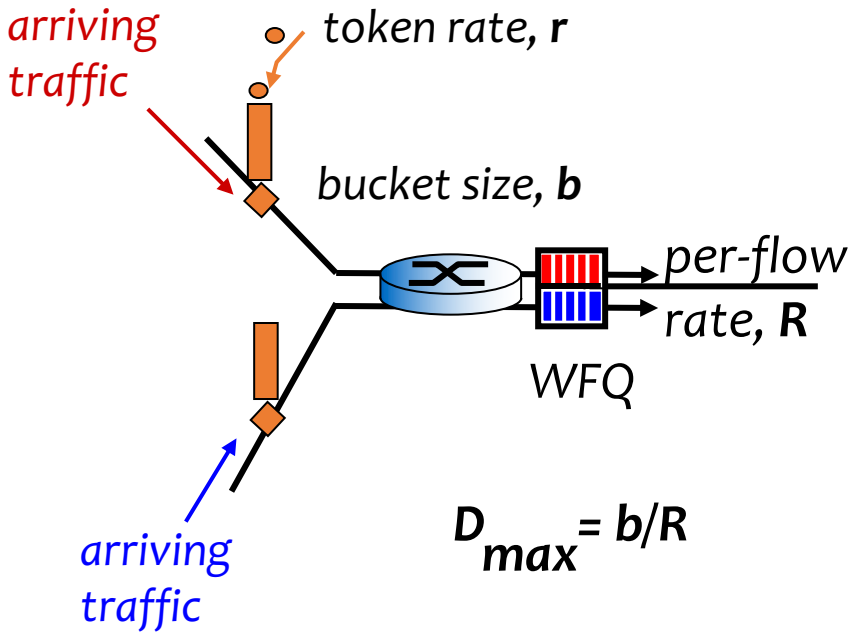
## *How?*

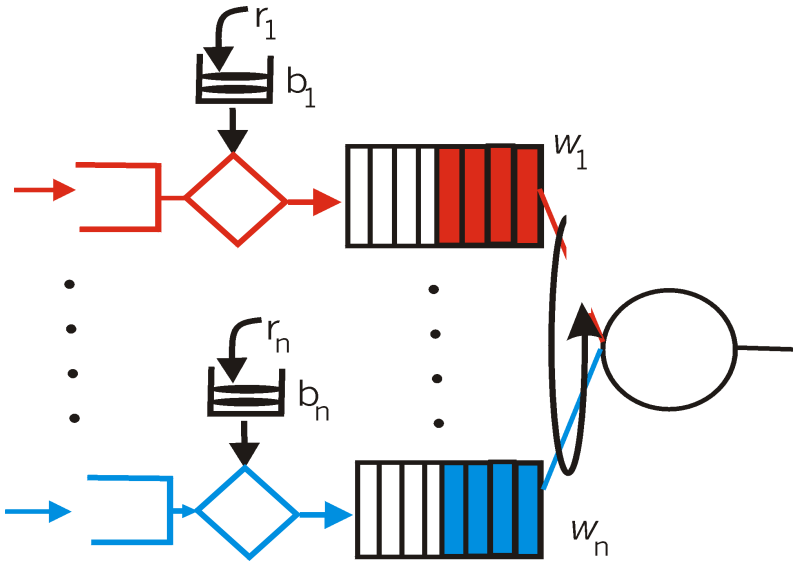
- Bucket can hold ***b*** tokens
- Tokens generated at rate ***r token/s*** unless bucket full
- Over interval of length ***t***: number of packets admitted *less than or equal to* ***(r t + b)***



# Token bucket & WFQ

Token bucket and WFQ combine to provide *guaranteed upper bound* on *delay*, *i.e.*, *QoS guarantee!*



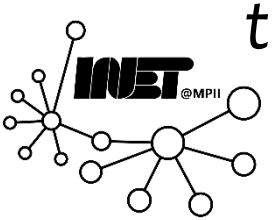
# Differentiated services

- Want *"qualitative"* service classes
  - "Behaves like a wire"
  - Relative service distinction: *Platinum, Gold, Silver*

***Scalability:*** simple functions in network core; relatively complex functions at *edge* routers (or hosts)

- Signaling, maintaining per-flow router state difficult with large number of flows

- Do *not define* define service classes, *provide functional components to build service classes*



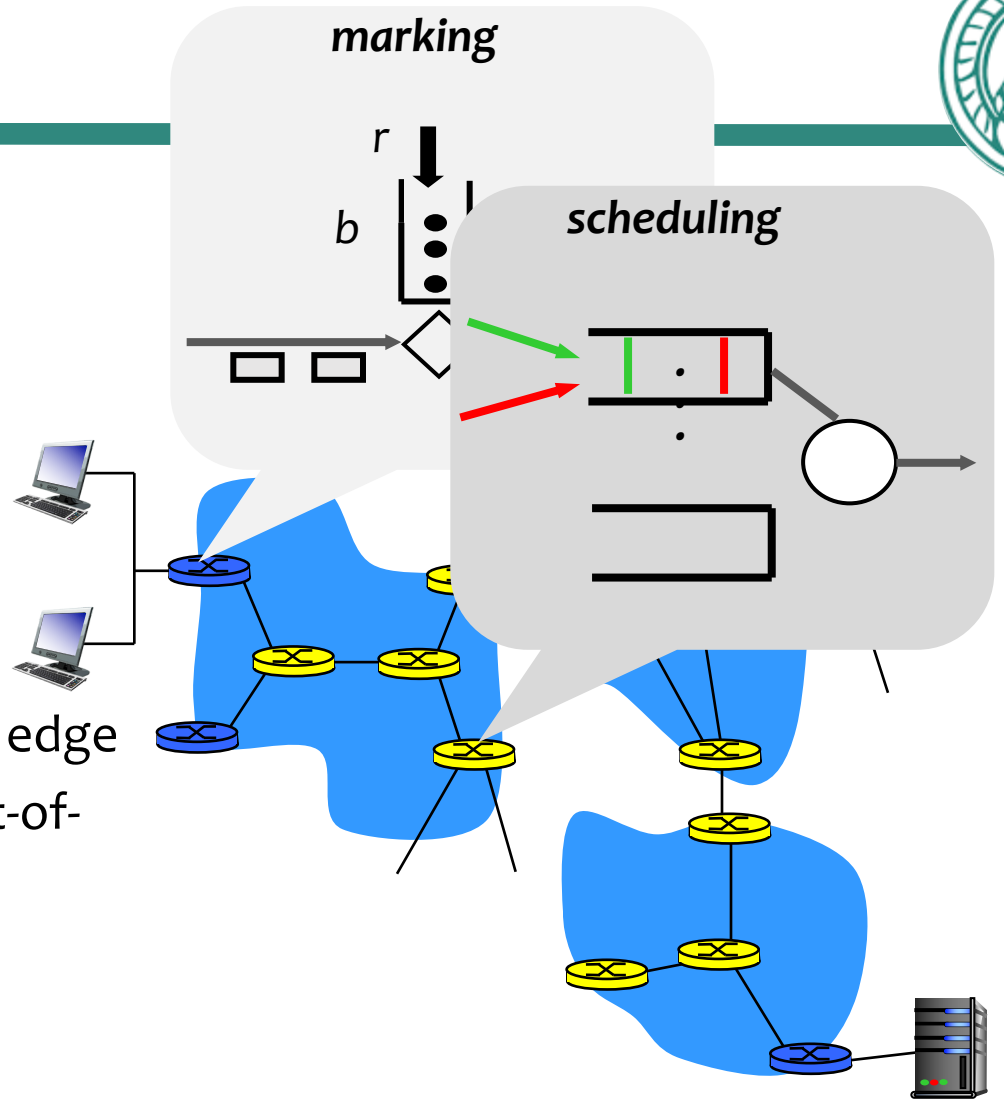
# DiffServ Architecture

### *Edge router*

- *Per-flow* traffic management
- Marks packets as *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 over out-of-profile packets



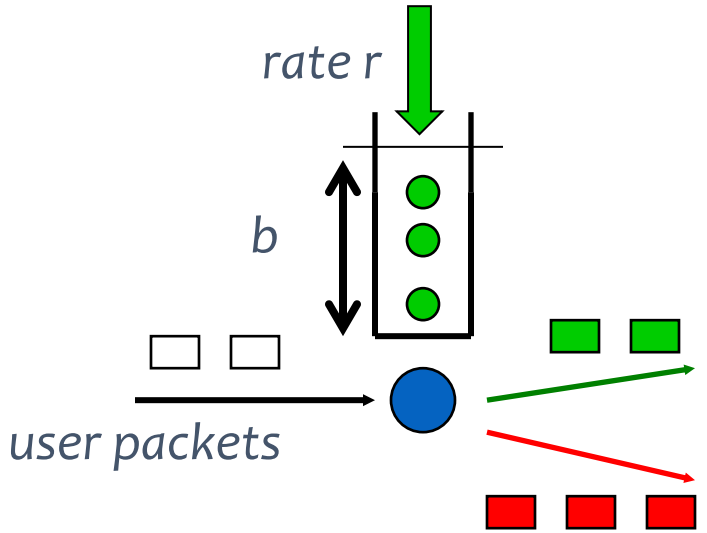
# Edge router packet marking

***Profile:*** pre-negotiated *rate* ***r***, *bucket size* ***b***

- Packet marking at edge based on per-flow profile

*Possible uses of marking*

- ***Class-based marking:*** Packets of different classes marked differently
- ***Intra-class marking:*** Conforming portion of flow marked differently than non-conforming one



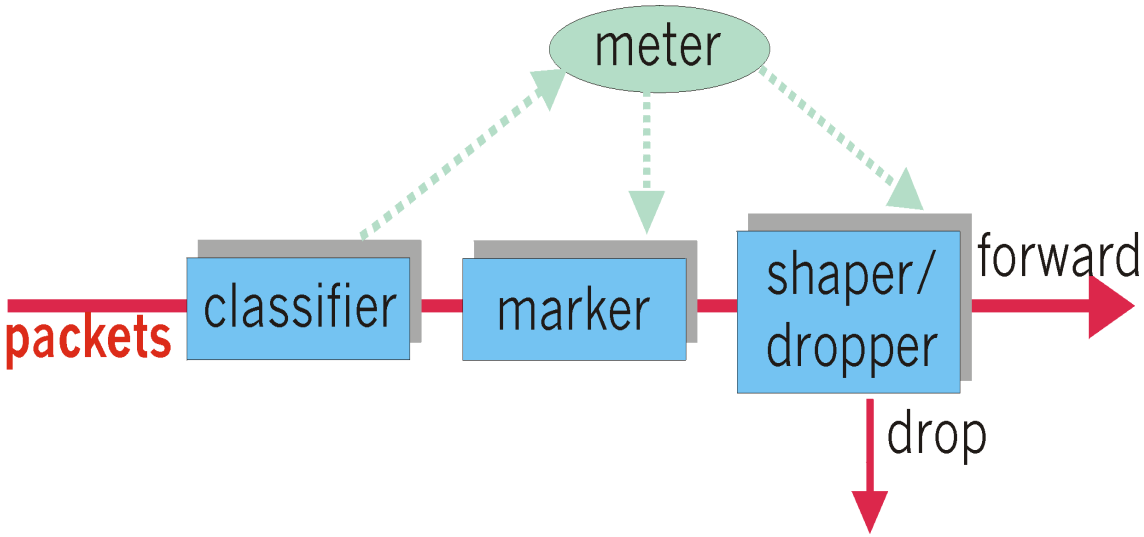
# DiffServ: Marking details


- 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)***
  - Determine *per-hop behavior* that the packet will receive
  - *2 bits* currently unused

# Classification, 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*



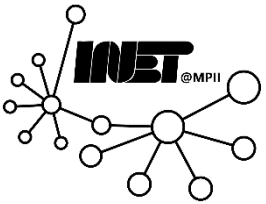
# Forwarding per-hop behavior (PHB)

Per-hop behavior 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



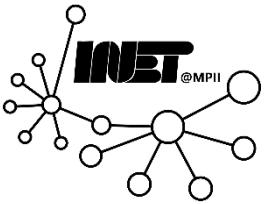
# Forwarding PHB

***Expedited forwarding:*** Packet departure rate of a class equals or exceeds specified rate

- Logical link with a minimum guaranteed rate

***Assured forwarding:*** ***4*** classes of traffic

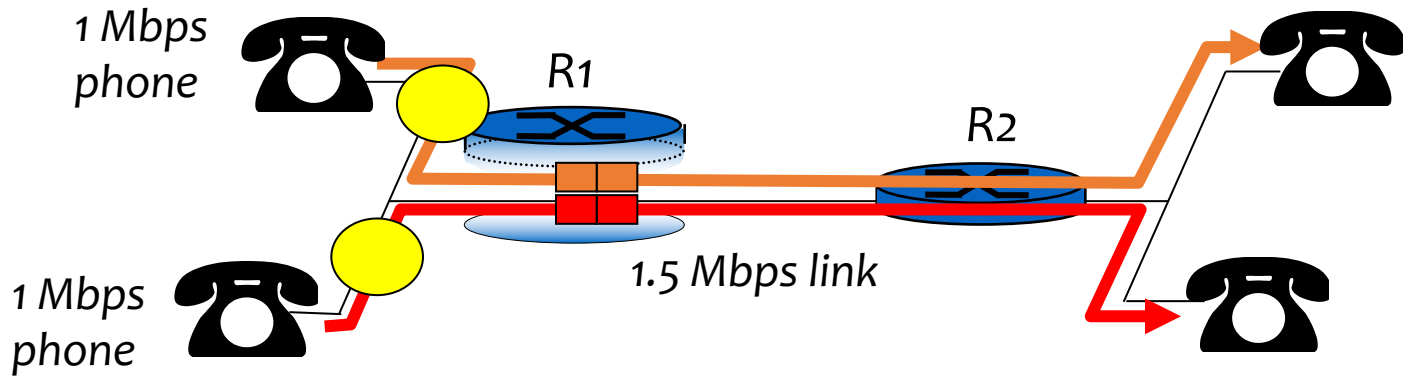
- Each guaranteed minimum amount of bandwidth
- Each with three drop preference partitions



# Per-connection QoS guarantees

## *Basic fact of life*

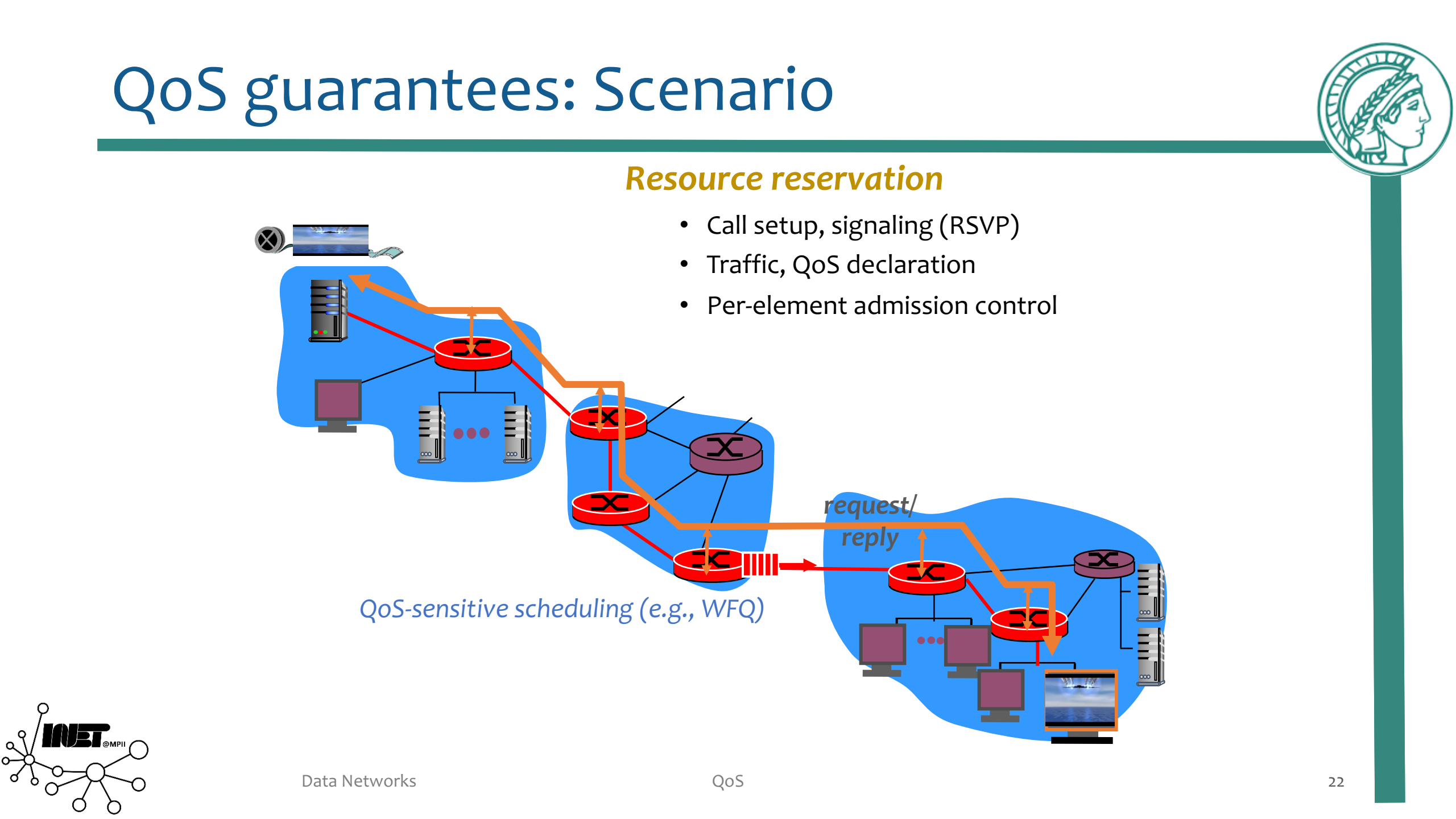
- *Cannot* support traffic demands beyond link capacity!




### *Principle 4*

***Call admission***: Flow declares its needs; network may block call *(e.g., busy signal)* if it cannot meet needs

# QoS guarantees: Scenario



## *Resource reservation*

- Call setup, signaling (RSVP)
- Traffic, QoS declaration
- Per-element admission control




*QoS-sensitive scheduling (e.g., WFQ)*

# Summary

- What is QoS?
- Why is it needed?
- What does it take to support QoS?