Quality of Service Principles

- Tutorial Multimedia Communication
  - Lars Wolf, Carsten Griwodz
  - Transparències:
    - Introduction
    - Quality of Service
    - Networks

Introduction to Quality of Service

From: Engineering the Internet QoS
Sanjay Jha, Mahbub Hassan
University of New South Wales (Sydney)
(with permission of the authors)

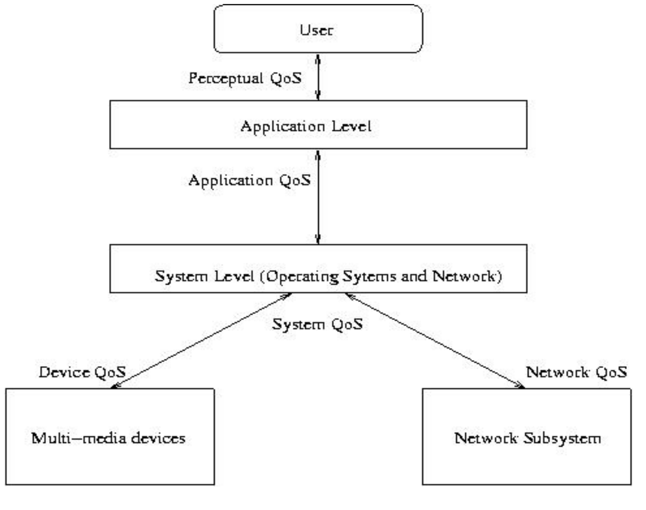
Overview

- What is QoS?
- Why QoS?
- Large Bandwidth vs QoS?
- Networking Trends Leading to QoS
- QoS in Best Effort IP Network

What is QoS?

- Variety of definitions exist in literature
- ATM definition- “*Quality of Service is the performance observed by an end user*”
- QoS is also usually expressed as the combination of network-imposed delay, jitter, bandwidth, loss and reliability
- Internet definition - Still evolving :-)
- Two categories of QoS parameters
  - Technology based
  - User Perception based

QoS Framework



QoS Translation

- QoS parameters need to be mapped between layers
- Application layer QoS Frame rate, size of video
- Network layer QoS bandwidth, Delay

Perceived QoS -> Systems translation

Peceptual Parameter	**System QoS**
Picture detail	Pixel resolution
Picture color accuracy	Maps to color information per pixel
Video Rate	Maps to frame rate
Video smoothness	Maps to frame rate jitter
Audio Quality	Audio Sampling rate and number of bits
Video/audio synchronisation	Video and audio stream synchronised for example lip-sync.

Adapted from: D. Chalmers and M. Sloman, “A Survey of Quality of Service in Mobile Computing Environments”, IEEE Communications Survey, Second Quarter 1999, http://www.comsoc.org/pubs/surveys/

Network QoS Parameters

Category	**Parameter**
Timeliness	Delay, Response Time, Jitter
Bandwidth	Systems Level Data Rate
	Application Level Data Rate
	Transaction Rate
Reliability	Mean time to failure (MTTF)
	Mean time between failures (MTBF)
	Mean time to repair (MTTR)
	Percentage of time available
	Loss or corruption rate

Adapted from: D. Chalmers and M. Sloman, "A Survey of Quality of Service in Mobile Computing Environments", IEEE Communications Survey, Second Quarter 1999, http://www.comsoc.org/pubs/surveys/

A Framework for Adaptive Applications

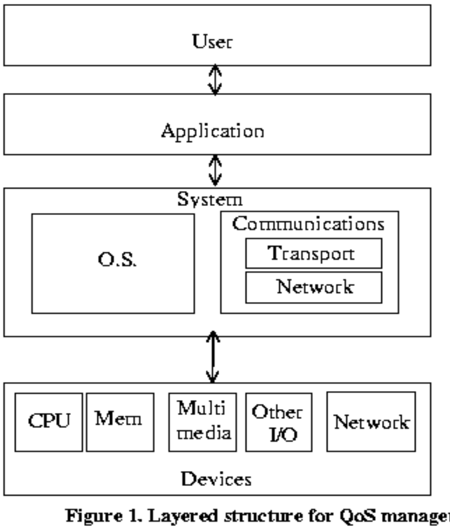


Figure 1. Layered structure for QoS management

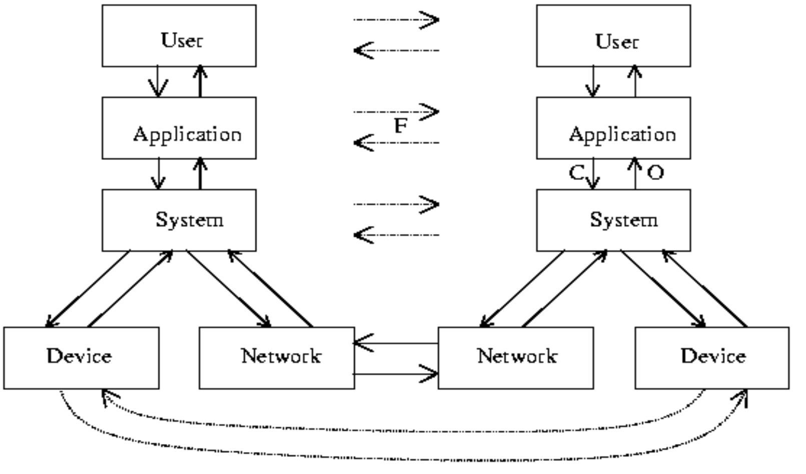
A Framework for Adaptive Applications

Observed parameters	Observed and Controlled parameters
USER	
Response time Media synchronisation Degree of satisfaction Cost	Image size Audio quality Colour depth Screen layout
APPLICATION	
Aspect ratio (video) Round-trip delay (audio) Synchronisation skew Database hit ratio	Video frame rate Frame width Frame height Compression ratio Number of layers
SYSTEM – Operating System	
Task processing time Memory assigned to a process CPU assigned to a process Processing capacity Size of virtual and main memory CPU load Connection processing time CPU assigned per connection	

A Framework for Adaptive Applications

Observed parameters	Observed and Controlled parameters
Communications system	
Goodput Latency Packet interarrival time Packet jitter Set-up time Error rate Recovery time Connection establishment failure prob. Availability	Packet data unit maximum size Packet rate Peak rate
DEVICE – Multimedia Device	
Encoder/decoder buffer size Screen resolution Cache hit ratio Memory hit ratio Disk hit ratio Throughput of the video player/encoder Delay of the video player/encoder	Sample rate Sample size Colour resolution
Network Device	
Bandwidth (capacity) MTU jitter MTU end-to-end delay MTU error rate End-to-end delay Bit error rate	MTU size MTU rate Bit rate

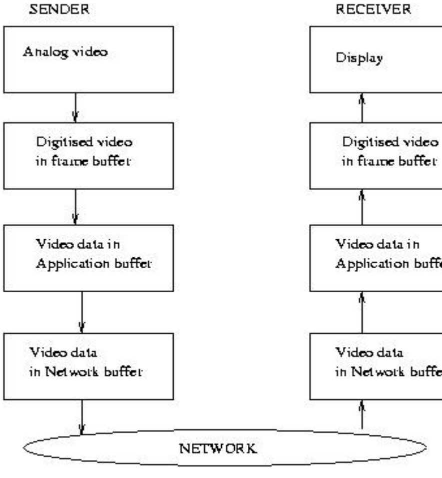
A Framework for Adaptive Applications



Why QoS?

- Quality of Service requirements
  - – Acceptable error rate
  - – Bounds on delay and jitter
  - – Sufficient bandwidth
- These requirements vary from application to application
    - Video on demand (VoD) can tolerate moderate end-to-end delay
    - Internet telephony or conferencing low end-to-end latency

Video Data Flow over Internet



Bandwidth for Audio

Coding Technique	Standard	Data rate
PCM (Pulse Code Modulation)	G.711	64Kbps
4-bit ADPCM (Adaptive Differential PCM)	G.726	32Kbps
2-bit ADPCM	G.726	16Kbps
CELP (Code-Excited Linear-Predictive)	G.728	16Kbps
Adaptive CELP	G.729	8Kbps
Part of H.324	G.723.1	5.3 or 6.3Kbps
Silence suppression		variable

Bandwidth for Video

Encoding technique	Bit rate	Resolution	Broadcast standard
H.261	64 Kbps-2 Mbps	176x144 352x288	QCIF (conference) CIF (VHS quality)
M-JPEG	3-8 Mbps 15-25 Mbps 60-100 Mbps	352x288 720x486 1920x1080	CIF (VHS quality) CCIR601 (PAL) HDTV
MPEG-1	1.2-3 Mbps 5-10 Mbps 20-40 Mbps	352x288 720x486 1920x1080	CIF (VHS quality) CCIR601 (PAL) HDTV
MPEG-2 (H.262)	1-2 Mbps 4-5 Mbps 8-10 Mbps 20-30 Mbps	352x288 720x486 960x576 1920x1080	CIF(VHS quality) CCIR601 (Pal) EDT HDTV

Trend: Faster Media

- One Gbps over 4-pair UTP-5 up to 100 m
  Was 1 Mbps (1Base-5) in 1984.
- Dense Wavelength Division Multiplexing (DWDM) allows 64 wavelengths in a single fiber
  64×OC-192 = 0.6 Tbps
  OC-768 = 40 Gbps demonstrated in 1998.
  Was 100 Mbps (FDDI) in 1993.
- 11 Mbps in-building wireless networks
  Was 1 Mbps (IEEE 802.11) in 1998.

Trend: More Traffic

- Number of Internet hosts is growing exponentially.
- Traffic per host is increasing:
  - Cable modems allow 1 to 10 Mbps access from home
  - 6-27 Mbps over phone lines using ADSL/VDSL
- Bandwidth requirements are doubling rapidly

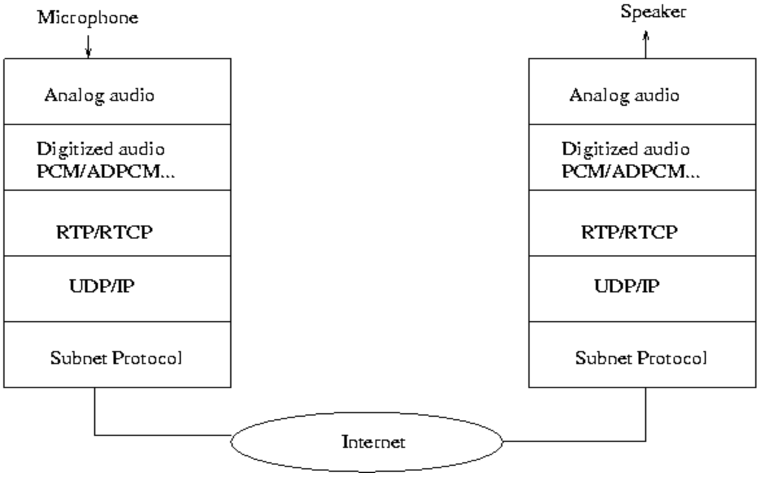
Current Internet Model

- Best effort service
  - Simple interface, robust
- Applications independent of underlying network
- No central administration
  - Autonomous administration of subnets
- Internetworking of heterogenous systems and network

Multimedia Over Best Effort

- Packet loss Recovery Schemes
  - Forward Error Correction
  - Interleaving
  - Repair at receiver
- Adaptation at Receiver to compensate for jitter
- Application Layer protocols RTP and RTCP

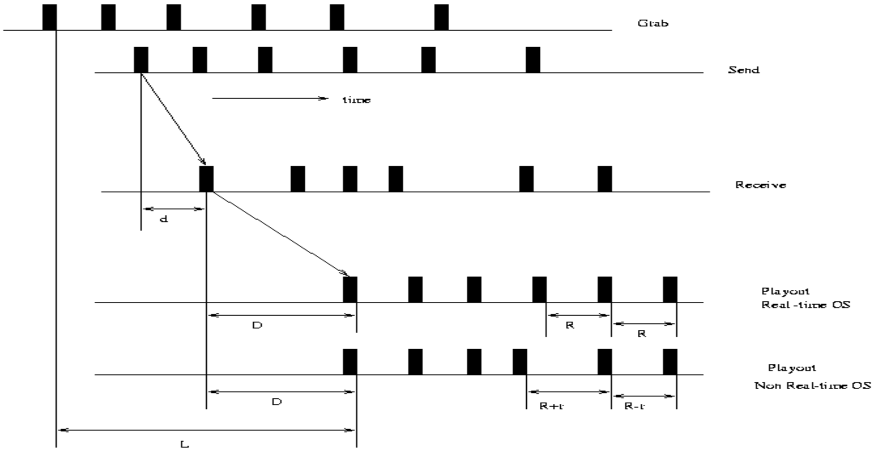
Protocol Stack



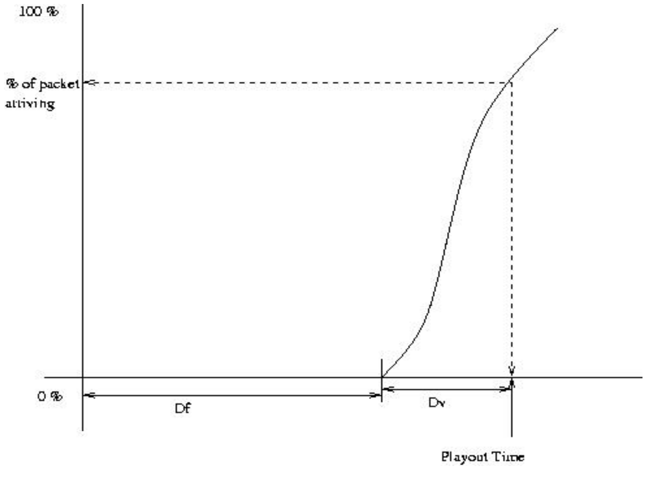
End to End Latency

- Packetization and coding : 10-20 ms
- Application and OS scheduling latencies
  – At both sender and receiver
- Transmission Delay: ~10 µs
- Propagation Delay: 200m/ µs (5µs/km)
- Variable queuing delay at routers
- Delay > 400ms makes communication unintelligible

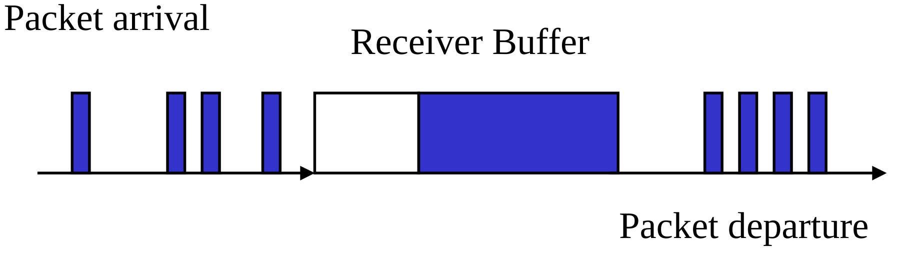
Adaptive Playout



Impact of Destination wait time



Jitter Compensation at Receiver

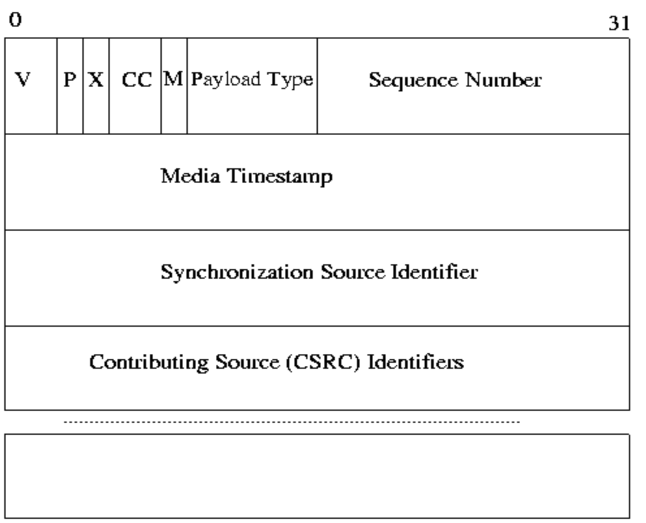


Buffer too many: unacceptable delay
Buffer few: gap in playout

Real-time Transport Protocol (RTP)

- Application layer software protocol
- Fields contain time-stamp and sequence #
  - Reconstruct temporal properties of stream
- Designed to work with a variety of protocols
  - Most of the a/v tools used it over UDP/IP
- Application interoperability facilitated by RTP profile payload formats for variety of a/v encoding

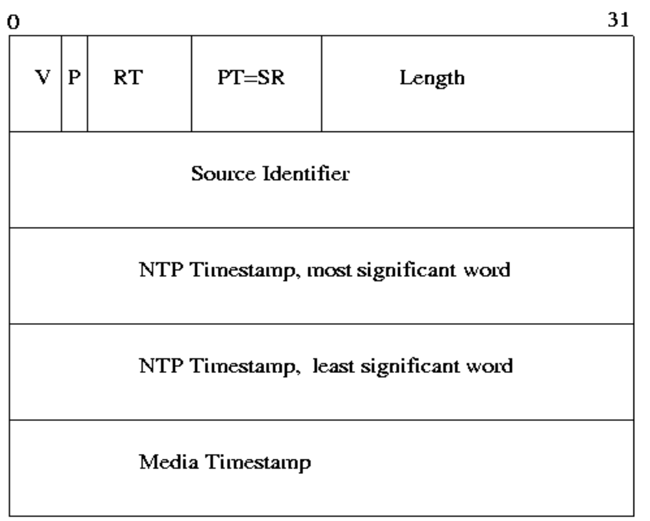
RTP Header



Real-time Transport Control Protocol

- RTCP companion protocol for monitoring and management
  - Feedback to sender periodically on delay, jitter, losses
  - Sender may adjust transmission rates (codec parameters)
- RTCP traffic limited to 5% of total bandwidth

RTCP Header



Deficiencies in Current Model

- No performance guarantee
  - Just one class *best effort service*
- No service level agreement (SLA)
- Most routers based on old packet switching technology
- Routing Protocols support shortest path
  - No load sharing and QoS support
- Need service model with several classes
  - Each meeting needs of set of applications

IETF Efforts

- IETF is standardising extensions to best-effort model
  - Integrated Services Model (IntServ)
  - Resource Reservation Protocol (RSVP)
  - Differentiated Services Model (DiffServ)
  - IntServ over DiffServ
  - Multiprotocol Label Switching (MPLS)

Summary

- What is QoS?
- Application Requiring QoS
- Quality of Service in Best Effort Network
- IETF Effort to Support QoS in the Internet

QoS Fundamentals

From: Engineering the Internet QoS
Sanjay Jha, Mahbub Hassan
University of New South Wales (Sydney)
(with permission of the authors)

QoS Framework

- Static Functions
  - Traffic and QoS specifications (traffic types/parameters)
  - QoS negotiation and signalling
  - Admission control
  - Resource reservation
- Dynamic Functions
  - Traffic shaping and policing
  - Queuing and scheduling (later)
  - Congestion control (later)

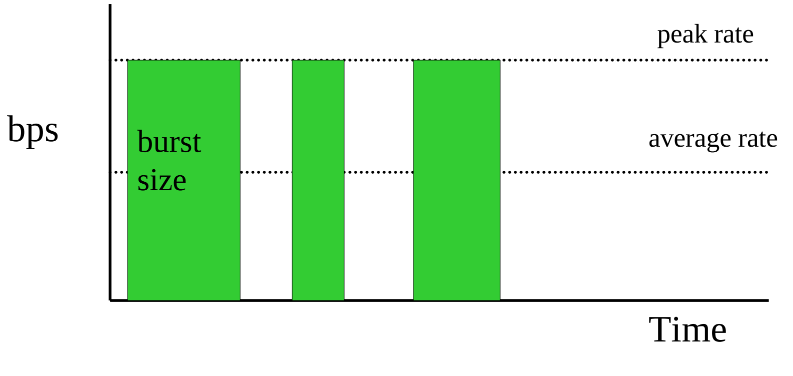
Traffic Source Types

- CBR (Constant Bit Rate) : transmits traffic at a fixed rate, such as 64 Kbps voice
- VBR (Variable Bit Rate) : traffic rate is not fixed; sometimes high, sometimes low, such as MPEG coded video

Traffic Parameters

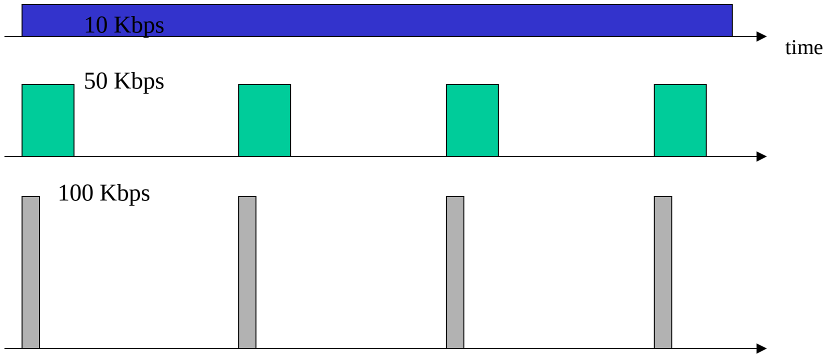
- Different flows have different traffic patterns
- A given traffic pattern can be described using several traffic parameters
- **Peak rate :** maximum rate in any time interval
- **Average rate :** long term average
- **Burst size :** duration of peaks

Traffic Parameters Illustrated



Traffic Patterns

All patterns have the *same average rate (10 Kbps)*, but different peak rate and burst size



QoS Parameters

- Required QoS can be defined by several parameters
- **Delay :** how long it takes for a packet to traverse the network?
- **Jitter :** what is the variance in the delay?
- **Loss :** how often packets get lost in the network and never show up at the destination?

Signalling

- Signalling is a mechanism used by the users to communicate QoS related information to the network
- Using signalling
  – User conveys its traffic parameters and QoS requirements to the network
  – Network conveys any QoS guarantees to the user

Admission Control

- First line of defence against attacks on QoS
- Network should not commit any guarantee if available resources are not enough to maintain requested QoS
- Admission control functions must examine both traffic and QoS parameters carefully before accepting or rejecting a new request for QoS
- Implementation
  – Dynamic : using signalling protocol/software
  – Static : manual process (no signalling required)

Resource Reservation

- To guarantee any QoS, network resources must be reserved in advance
- Types of network resources
  - Bandwidth
  - Buffer space
- Reservation could be dynamic, using signalling, or static (manual)

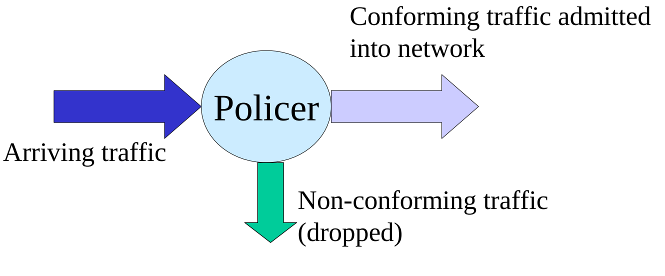
How Much to Reserve?

- Easy for constant bit rate sources
  - Reserve at the peak rate
- Difficult for VBR sources
  - Peak rate reservation wastes bandwidth (no statistical gain)
  - Average rate reservation may cause excessive packet delays

Traffic Policing

- Users violating the traffic contract can jeopardise the QoS of other connections
- The network must protect well behaving users against such traffic violations
- All entering traffic is therefore subject to policing
- Policing functions are deployed at the edge (entry) of the network

Traffic Policing



Requirements for Policing Mechanisms

- Policing functions must operate in real-time
- Easy and simple (not complex) to implement
- For every entering packet, must be capable of detecting whether the packet violates the agreed traffic contract

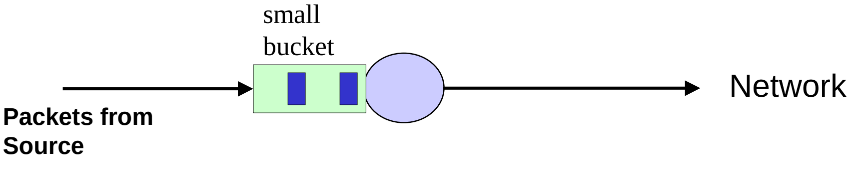
Policing Parameter Combinations

- Peak Rate Only
  – Suitable for CBR sources
- Average Rate and Burst Size
  – For VBR sources without limit on peak rate
- Peak rate, Average Rate and Burst Size
  – VBR sources with peak rate limitation

Policing with Leaky Bucket

- Leaky Bucket is a widely used mechanism to police peak rate, average rate and burst size
- Peak rate policing : simple leaky bucket
- Average rate and burst size : token bucket
- Peak rate, average rate and burst size: leaky bucket and token bucket in tandem

Simple Leaky Bucket



Implementation of Peak Rate Policing with Leaky Bucket

- No buffers needed ! (no queuing)
- Requires only one *counter*
  - *counter* is decremented, to a minimum of zero, at the peak rate
  - *counter* is incremented by one, up to a threshold, for each packet arrival
- An arriving packet is non-conforming if *counter* is at the threshold

Example : Peak Rate Policing

Peak Rate to police : 1000 packets per second

period to decrement the counter : 1 ms

counter threshold (burst allowed) : 2 packets

packet arrivals:

10ms (counter = 1; conforming)

11ms (counter = 1; conforming)

11.2ms (counter = 2; conforming)

11.5ms (counter = 2; nonconforming)

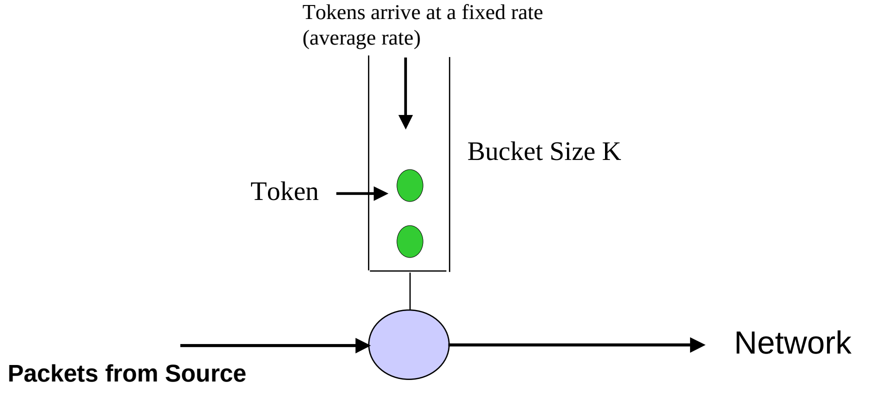
12ms (counter = 2; conforming)

Q. which packets would conform if counter threshold was set to 1?

Policing Variably-Sized Packets

- Transmission rates are expressed in bps
- Fixed-sized packets easily translate bps to packet per second (e.g. ATM cells)
- Internet has variably-sized packets
- *Counter threshold should be set to some bytes, rather than packets*

Token Bucket



Implementing Average Rate and Burst Size Policing with Token Bucket

- No buffer required! (no queuing)
- One counter for token bucket
- Counter is incremented at the average rate up to a threshold (burst size)
- Counter is decremented by one for each packet accepted
- An Arriving packet is considered non-conformant if the counter is zero
- Counter is set to some bytes for variably-sized packets

Example : Average Rate & Burst Size Policing

Average Rate to police : 100 packets per second

period to add a token : 10 ms

counter threshold (burst allowed) : 10 packets

Assume counter=10 [line was idle for a while)

Packet serialisation time = 0.1 ms

packet arrivals:

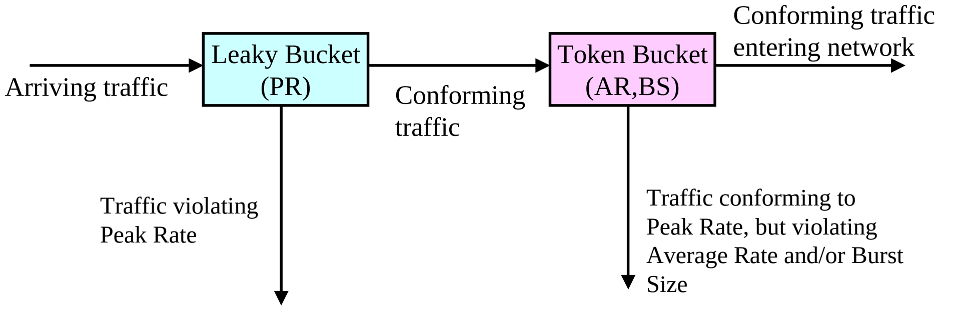
100ms, 100.1ms, 100.2ms, 100.3ms, 100.4ms, 100.5ms (burst of 6 admitted; counter = 10 - 6 = 4)

110ms (admitted; counter=4)

120.1ms, 120.2ms, 120.3ms, 120.4ms, 120.5ms, 120.6 (burst of 5 admitted; last one not admitted because counter became zero)

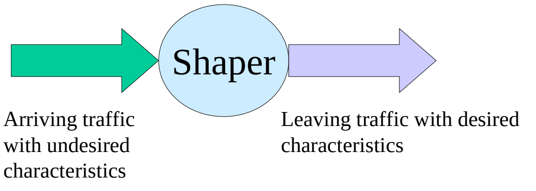
Dual Leaky Bucket Peak Rate, Average Rate, Burst Size

- No buffers
- Two counters are used , one for leaky bucket and one for token bucket



Traffic Shaping

- Altering the traffic characteristics of a given flow is called traffic shaping
- The source must shape its traffic prior to sending it to network so it does not violate traffic contract



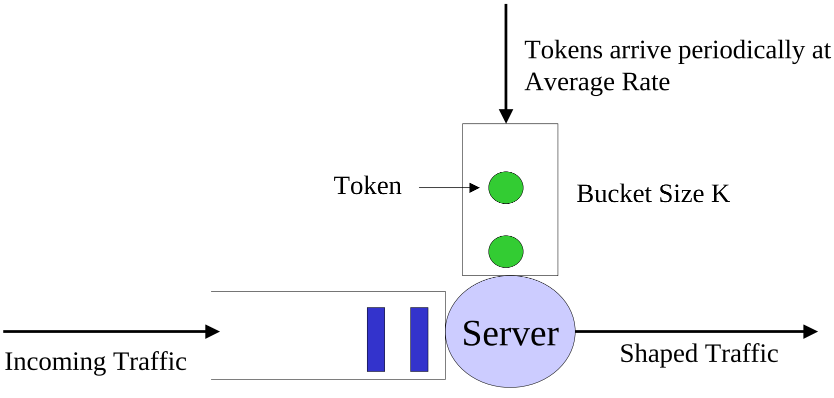
Traffic Shaping vs Traffic Policing

- Shaping *regulates* a flow to make sure it does not violate traffic contract
- Policing *monitors* a flow (*does not regulate*) to detect violation

Shaping Mechanisms

- Similar to policing mechanisms except it *buffers* traffic to smooth it out (policing *does not buffer* traffic as it is not interested in smoothing it)
- Token Bucket : peak rate, average rate and burst size shaping

Token Bucket Shaper (PR,AR,BS)



Scheduling for QoS Management

From: Engineering the Internet QoS
Sanjay Jha, Mahbub Hassan
University of New South Wales (Sydney)
(with permission of the authors)

Outline

- What is Queue Management and Scheduling?
- Goals of scheduling
- Fairness (Conservation Law/Max-min fair share)
- Various scheduling techniques
- Research directions in scheduling

What is scheduling?

- Packets from multiple flows compete for same outgoing link.
- Which packets should be given preference?
- How many packets should be transmitted from a flow?
- Simple solution: First come best served
- Complex solution: Provide QoS guarantees.

Scheduling Goals

- Sharing bandwidth
- Fairness to competing flows
- Meeting bandwidth guarantees (max and min)
- Meeting loss guarantees (multiple level)
- Meeting delay guarantees (multiple level)
- Reducing delay variations

Conservation law

- Sum of the mean queuing delays received by the set of multiplexed connections, weighted by their share of link's load is independent of the scheduling discipline – Kleinrock

$$\rho_i = \lambda_i \quad x_i$$

$$\sum_{i=1}^{N} \rho_i \quad q_i = Const$$

Conservation Law Contd

ρ_i = mean utilization of flow i

λ_i = mean arrival rate of flow i

x_i = mean service time of packets from flow i

q_i = mean wait time of flow i at scheduler

N = number of flows

Max-min fair share

- Allocates the smallest of all demands from all flows
- Distribute remaining resources equally competing of the flows
- Guarantees fairness

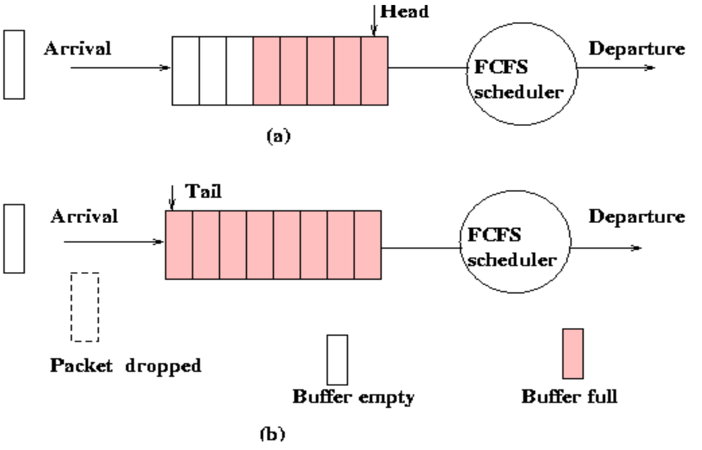
Scheduling Disciplines

- First come first serve (FCFS)
- Priority (PQ)
- Round Robin (RR)/Weighed round robin
- Deficit round robin (DRR)
- Weighted fair queuing (WFQ)
- Class based queuing (CBQ)

First Come First Serve

- Packets queued into a common buffer
- Server serves packet from front of queue
- No fair sharing of bandwidth
- No flow isolation
- No priority or QoS guarantee

FCFS example

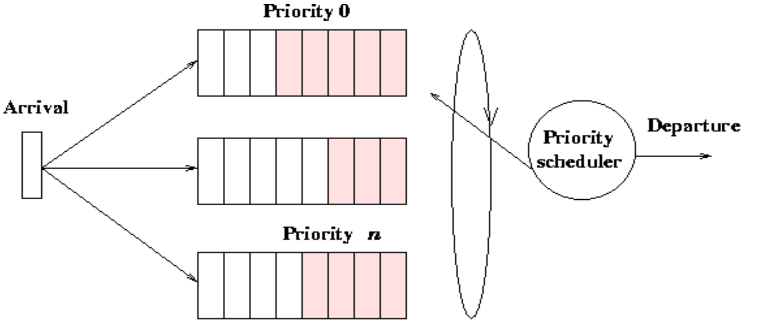


Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

Priority Queuing

- Multiple queues with priority 0 to n-1
- Priority 0 served first
- Priority i served only if 0 to i-1 empty
- Highest priority – lowest delay/loss, highest bandwidth
- Possible starvation of lower class

Priority Queue example



Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

Generalized processor sharing

- Ideal work conserving scheme
- Flows kept in separate queue
- Serve infinitesimal amount of data from each queue
- Serve all active queues in finite time
- Weight can be associated with each queue
- Achieves max-min fair share

GPS Continued

- In GPS terminology, a connection is called backlogged when it has data present in queue.
- Lets assume that there are *K* flows to be served by a server implementing GPS with weights *w(1), .. w(k)* Service rate of i_{th} flow in interval [τ, *t*] is represented as *R(i, τ,t)*. For any backlogged flow *i* in interval [τ,*t*] and for another flow *j*, the following equation holds:

$$R(i,\tau,t)/R(j,\tau,t) \geq w(i)/w(j)$$

Round Robin

- Flows kept in separate queue
- Serve one packet from each active queue
- Fair share but no bandwidth guarantee
- Provides fair allocation if fixed packet size
- What if packet size variable ?

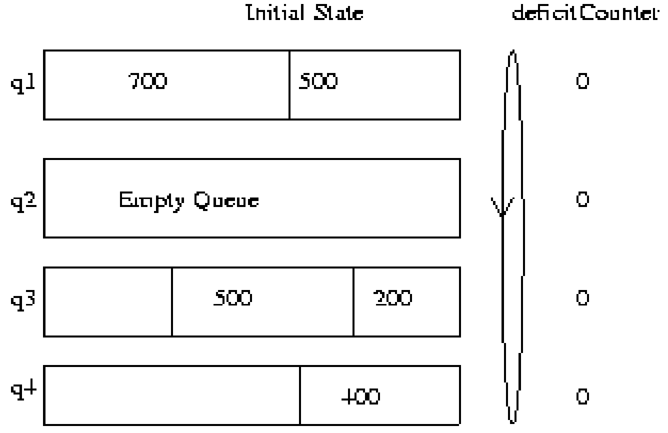
Weighted Round Robin

- Allows variable length packet
- Serves *n* packet from a queue
- *n* adjusted to specific fraction of link share
- Fairness problem at small time scale
- Needs to know packet size *a priori*
- Assume 3 ATM sources (small cell size) with weights 0.75, 1.0 and 1.5. If these weights are normalised to integer values, each source will be served 3, 4 and 6 cells in each round.

Deficit Round Robin

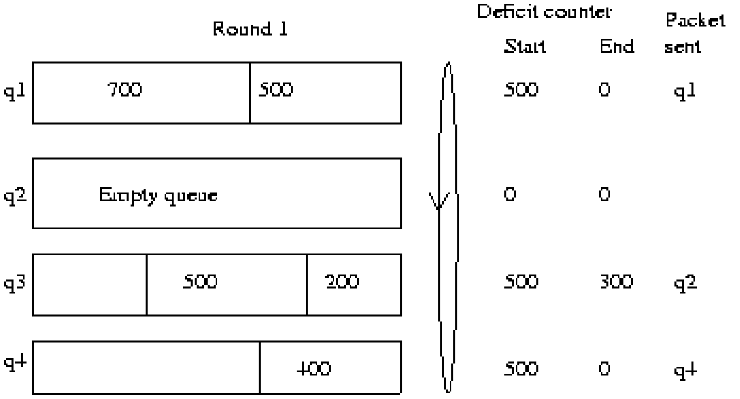
- No need to know packet size *a priori*
- Initially serves each queue *quantum* worth of bits
- If packet less than or equal to *quantum*, serve it
- Else increment *deficit_counter* by *quantum*
- If no more outstanding packet, reset *deficit_counter* (Why?)
- Set *quantum* to minimum MTU of all incoming links
- Fairness problem at smaller time scale (MTU time)

DRR Example 1



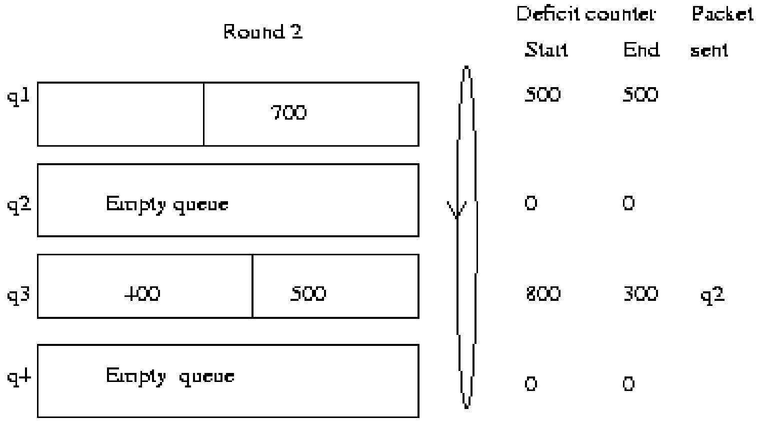
Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

DRR Example 2



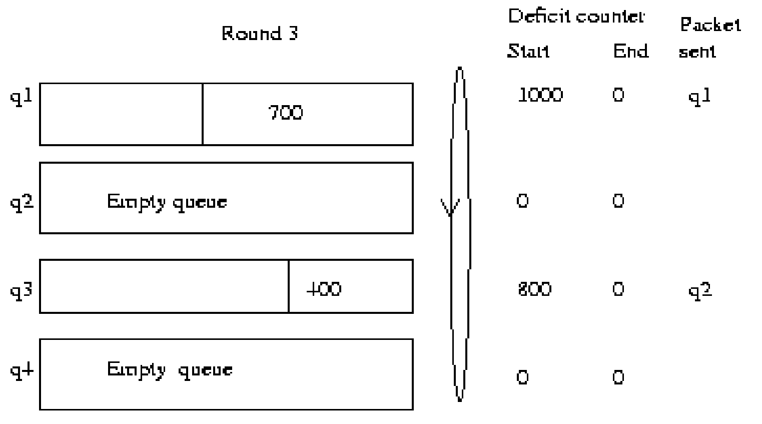
Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

DRR Example 3



Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

DRR Example 4



Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

Weighted Fair Queuing

- Packets tagged with a value identifying the time last bit of packet should be transmitted using GPS simulation
- Packet with lowest tag value transmitted by scheduler
- Uses complex finish time calculation
- Hard to implement with variable packet size

WFQ min Throughput

- QoS guarantees possible (gets bandwidth in proportion of weight)

$$Min\ Throughput = Rw(i) / \sum w(j)$$

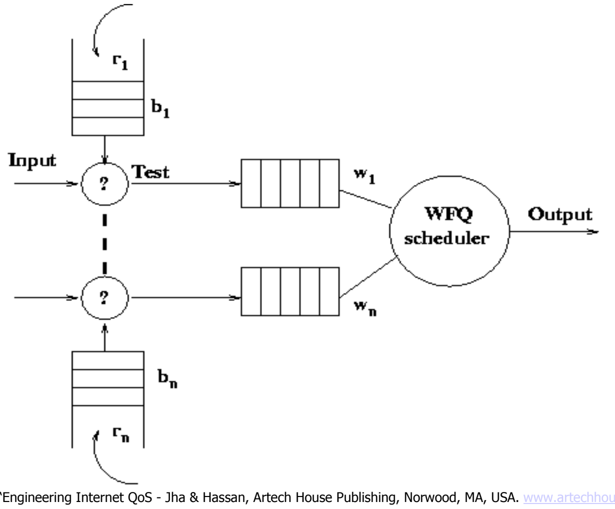
  - R link transmission rate
  - W_i weight for class i
  - j classes that have packets waiting

WFQ Delay bounds

- Delay can be bounded if flows can be policed (token bucket)
- Flows regulated by token bucket are put in different queues
- Each queue has assigned weight
- With token bucket policing, assume that initially the token bucket is full and a burst of b_i packets arrive for a flow of class *i*. Last packet to complete service will suffer a maximum delay of d_{max} given by equation

$$d_{\max} = b_i / (R w(i) / \sum w(j))$$

WFQ Delay with Token bucket



Finish Time Calculation

- Following equation shows the finish time calculation where *R(t)* is called round number. P^c_m is the time required to transmit m_{th} packet from c_{th} connection and *w(c)* is the weight of connection *c*.

$$F^c(m) = \max(F^c_{m-1}, R(t)) + P^c_m / w(c)$$

©UPC DAC JDP

Round Number

- This is the number a bit-by-bit round robin scheduler (in place of GPS's non-implementable infinitesimal data) has completed at a given time. The round number is a variable that depends on number of active queues to be served (inversely proportional to the active queue number). The more queues to serve, the longer a round will take to complete (example and figure in section 3.2.7 of text)

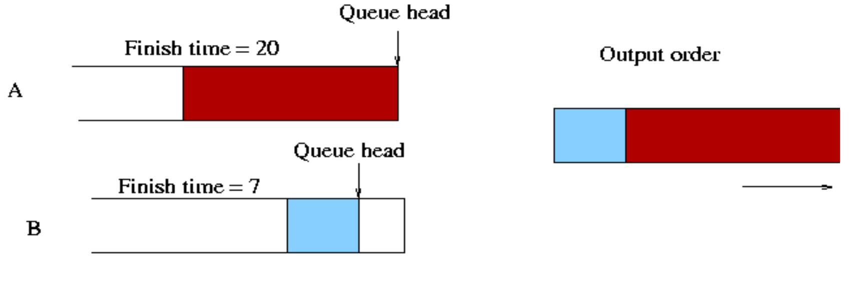
©UPC DAC JDP

Virtual Clock

- Also known as Fair Queuing
- Emulates TDM in place of GPS
- WFQ finish time calculation is very complex
- Virtual clock replaces round time with real time as per the following equation (here, A_m is the real-arrival time of packet *m*):

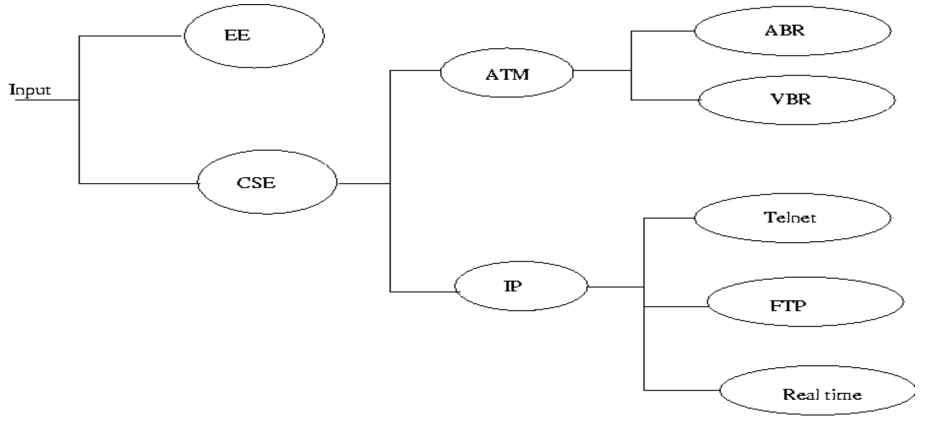
$$F^{c}(m) = \max(F^{c}_{m-1}, A_m) + P^{c}_{m}$$

VC Example



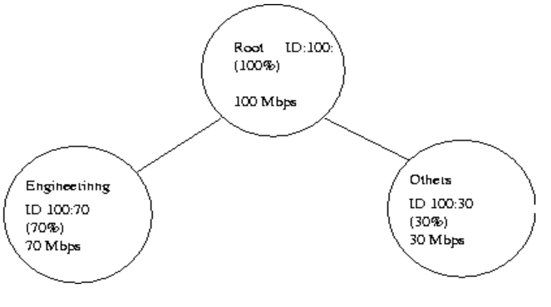
Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

Class Based Queuing



Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

CBQ Contd



Reprinted with Permission from "Engineering Internet QoS - Jha & Hassan, Artech House Publishing, Norwood, MA, USA. www.artechhouse.com

Scheduling Research Directions

- Worst-case fair weighted Fair queuing (WF^2Q)
- Self clocked fair queuing (SCFQ)
- Start time fair queuing (SFQ)
- Core state fair queuing (CSFQ)
- Score others