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

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

Network QoS Parameters

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeliness</td>
<td>Delay, Response Time, Jitter</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Systems Level Data Rate</td>
</tr>
<tr>
<td></td>
<td>Application Level Data Rate</td>
</tr>
<tr>
<td></td>
<td>Transaction Rate</td>
</tr>
<tr>
<td>Reliability</td>
<td>Mean time to failure (MTTF)</td>
</tr>
<tr>
<td></td>
<td>Mean time between failures (MTBF)</td>
</tr>
<tr>
<td></td>
<td>Mean time to repair (MTTR)</td>
</tr>
<tr>
<td></td>
<td>Percentage of time available</td>
</tr>
<tr>
<td></td>
<td>Loss or corruption rate</td>
</tr>
</tbody>
</table>

### A Framework for Adaptive Applications

<table>
<thead>
<tr>
<th>USER</th>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Response time</td>
<td>Image size</td>
</tr>
<tr>
<td></td>
<td>Media synchronisation</td>
<td>Audio quality</td>
</tr>
<tr>
<td></td>
<td>Degree of satisfaction</td>
<td>Color depth</td>
</tr>
<tr>
<td></td>
<td>Cost</td>
<td>Screen layout</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aspect ratio (video)</td>
<td>Video frame rate</td>
</tr>
<tr>
<td></td>
<td>Round trip delay (audio)</td>
<td>Frame width</td>
</tr>
<tr>
<td></td>
<td>Synchronization skew</td>
<td>Frame height</td>
</tr>
<tr>
<td></td>
<td>Database link ratio</td>
<td>Compression ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of layers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Task processing time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Memory assigned to a process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU assigned to a process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Processing capacity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size of virtual and main memory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU load</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Connection processing time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU assigned per connection</td>
<td></td>
</tr>
</tbody>
</table>

### Communications System

<table>
<thead>
<tr>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goodput</td>
<td></td>
</tr>
<tr>
<td>Latency</td>
<td></td>
</tr>
<tr>
<td>Packet interarrival time</td>
<td></td>
</tr>
<tr>
<td>Packet jitter</td>
<td></td>
</tr>
<tr>
<td>Setup time</td>
<td></td>
</tr>
<tr>
<td>Error rate</td>
<td></td>
</tr>
<tr>
<td>Recovery time</td>
<td></td>
</tr>
<tr>
<td>Connection establishment failure probability</td>
<td></td>
</tr>
</tbody>
</table>

### MULTIMEDIA DEVICE

<table>
<thead>
<tr>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder/Decoder buffer size</td>
<td></td>
</tr>
<tr>
<td>Screen resolution</td>
<td></td>
</tr>
<tr>
<td>Cache hit ratio</td>
<td></td>
</tr>
<tr>
<td>Memory hit ratio</td>
<td></td>
</tr>
<tr>
<td>Disk hit ratio</td>
<td></td>
</tr>
<tr>
<td>Throughput of the video player/encoder</td>
<td></td>
</tr>
<tr>
<td>Delay of the video player/encoder</td>
<td></td>
</tr>
</tbody>
</table>

### NETWORK DEVICE

<table>
<thead>
<tr>
<th>Observed parameters</th>
<th>Observed and Controlled parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth (capacity)</td>
<td></td>
</tr>
<tr>
<td>MTU</td>
<td></td>
</tr>
<tr>
<td>MTU-end-to-end delay</td>
<td></td>
</tr>
<tr>
<td>MTU error rate</td>
<td></td>
</tr>
<tr>
<td>Error rate</td>
<td></td>
</tr>
</tbody>
</table>
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

Sender
- Analog video
- Digitized video in frame buffer
- Video data in application buffer
- Video data in network buffer

Receiver
- Display
- Digitized video in frame buffer
- Video data in application buffer
- Video data in network buffer

Bandwidth for Audio

<table>
<thead>
<tr>
<th>Coding Technique</th>
<th>Standard Data rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM (Pulse Code Modulation)</td>
<td>G.711 64Kbps</td>
</tr>
<tr>
<td>4-bit ADPCM (Adaptive Differential PCM)</td>
<td>G.726 32Kbps</td>
</tr>
<tr>
<td>2-bit ADPCM</td>
<td>G.726 16Kbps</td>
</tr>
<tr>
<td>CELP (Code-Excited Linear-Predictive)</td>
<td>G.728 16Kbps</td>
</tr>
<tr>
<td>Adaptive CELP</td>
<td>G.729 8Kbps</td>
</tr>
<tr>
<td>Part of H.324</td>
<td>G.723.1 5.3 or 6.3Kbps</td>
</tr>
<tr>
<td>Silence suppression</td>
<td>variable</td>
</tr>
</tbody>
</table>
Bandwidth for Video

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

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 $64 \times OC-192 = 0.6$ Tbps
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
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

Packet arrival
Receiver Buffer
Packet departure

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**

0 | 31
---|---
V | P | X | CC | M | Payload Type | Sequence Number

- Media Timestamp
- Synchronization Source Identifier
- Contributing Source (CSRC) Identifiers

---

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

<table>
<thead>
<tr>
<th>0</th>
<th>V</th>
<th>P</th>
<th>RT</th>
<th>PT=SR</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Source Identifier</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTP Timestamp, most significant word</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NTP Timestamp, least significant word</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media Timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 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

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:
10 ms (counter = 1; conforming)
11 ms (counter = 1; conforming)
11.2 ms (counter = 2; conforming)
11.5 ms (counter = 2; nonconforming)
12 ms (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

Tokens arrive at a fixed rate (average rate)

Bucket Size K

Network
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

![Diagram of Dual Leaky Bucket](image)

**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

![Diagram of Traffic Shaping](image)
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)

Tokens arrive periodically at Average Rate

Token

Bucket Size K

Server

Incoming Traffic

Shaped Traffic

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

\[ \rho_i = \lambda_i \cdot x_i \]

\[ \sum_{i=1}^{N} \rho_i \cdot q_i = Const \]
Conservation Law Contd

\[ \rho_i = \text{mean utilization of flow } i \]
\[ \lambda_i = \text{mean arrival rate of flow } i \]
\[ x_i = \text{mean service time of packets from flow } i \]
\[ q_i = \text{mean wait time of flow } i \text{ at scheduler} \]
\[ N = \text{number of flows} \]

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
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

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 $[\tau, t]$ is represented as $R(i, \tau, t)$. For any backlogged flow $i$ in interval $[\tau, t]$ and for another flow $j$, the following equation holds:

$$\frac{R(i, \tau, t)}{R(j, \tau, t)} \geq \frac{w(i)}{w(j)}$$

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

```
Initial State

q1  700  500
q2  Empty Queue
q3  500  300
q4  400  0
```

deficit Counter: 0 0 0 0

---

DRR Example 2

Round 1

```
q1  700  500
q2  Empty queue
q3  500  300
q4  400  0
```

Deficit counter:

- Start
- End
- Backet sent

```
q1  500  0
q2  0  0
q3  500  300
q4  500  0
```

---

DRR Example 3

Round 2

<table>
<thead>
<tr>
<th>Queue</th>
<th>Deficit</th>
<th>Packet sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>700</td>
<td>500</td>
</tr>
<tr>
<td>q2</td>
<td>Empty queue</td>
<td>0</td>
</tr>
<tr>
<td>q3</td>
<td>400</td>
<td>300</td>
</tr>
<tr>
<td>q4</td>
<td>Empty queue</td>
<td>0</td>
</tr>
</tbody>
</table>

DRR Example 4

Round 3

<table>
<thead>
<tr>
<th>Queue</th>
<th>Deficit</th>
<th>Packet sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>700</td>
<td>1000 q1</td>
</tr>
<tr>
<td>q2</td>
<td>Empty queue</td>
<td>0</td>
</tr>
<tr>
<td>q3</td>
<td>100</td>
<td>0 q2</td>
</tr>
<tr>
<td>q4</td>
<td>Empty queue</td>
<td>0</td>
</tr>
</tbody>
</table>

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)

\[ \text{Min Throughput} = \frac{R_w(i)}{\sum_w j} \]

- \( R \) 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 $d_{\text{max}}$ given by equation

$$d_{\text{max}} = b_i / (Rw(i) / \sum w(j))$$
Finish Time Calculation

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

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

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)
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 $m$):

$$F^c_m = \max(F^c_{m-1}, A_m) + P_m$$

VC Example

Scheduling Research Directions

- Worst-case fair weighted Fair queuing (WF²Q)
- Self clocked fair queuing (SCFQ)
- Start time fair queuing (SFQ)
- Core state fair queuing (CSFQ)
- Score others