15-744: Computer Networking

L-18 QOS - IntServ

QOS & IntServ

- QOS
- IntServ Architecture
- Assigned reading
  - [She95] Fundamental Design Issues for the Future Internet
  - [CSZ92] Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms

Overview

- Why QOS?
- How do we build QOS?

Motivation

- Internet currently provides one single class of **"best-effort" service**
  - No assurances about delivery
- Existing applications are ***elastic***
  - Tolerate delays and losses
  - Can adapt to congestion
- Future "real-time" applications may be ***inelastic***

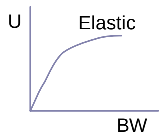
Inelastic Applications

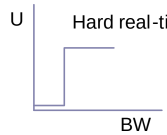
- Continuous media applications
  - **Lower and upper limit** on acceptable performance.
  - BW below which video and audio are not intelligible
  - Internet telephones, teleconferencing with high delay (200 - 300ms) impair human interaction
- Hard real-time applications
  - Require **hard limits on performance**
  - E.g. control applications

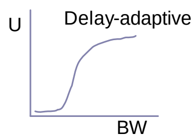
Why a New Service Model?

- What is the **basic objective** of network design?
  - Maximize total bandwidth? Minimize latency?
  - **Maximize user satisfaction** – the total **utility** given to users
- What does utility vs. bandwidth look like?
  - Must be non-decreasing function
  - Shape depends on application

Utility Curve Shapes

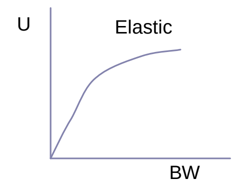






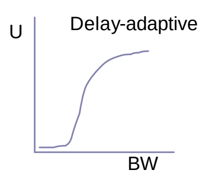
Stay to the right and you are fine for all curves

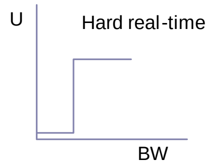
Utility curve – Elastic traffic



Does equal allocation of bandwidth maximize total utility?

Utility Curves – Inelastic traffic





Does equal allocation of bandwidth maximize total utility?

Why a New Service Model?

- Given the shape of different utility curves – clearly equal allocation of bandwidth does not maximize total utility
- In fact, desirable rate for some flow may be 0.

Admission Control

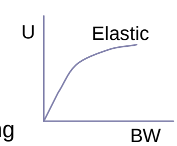
Principle 1 for QOS guarantees:

Admission control → deciding when the addition of new people would result in reduction of utility

- Basically avoids overload

Admission Control

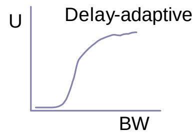
- If U(bandwidth) is concave → elastic applications



  - Incremental utility is decreasing with increasing bandwidth
  - Is always advantageous to have more flows with lower bandwidth
  - => No need of admission control; This is how the Internet works!

Admission Control

- If U is convex → inelastic applications
  - U(number of flows) is no longer monotonically increasing
  - Need admission control to maximize total utility



Admission Control

- Caveats
  - Admission control can only turn away new requests → sometimes it may be have been better to terminate an existing flow
  - U(0) != 0 → users tend to be very unhappy with no service – perhaps U should be discontinuous here
- Alternative → overprovision the network
  - Problem: high variability in usage patterns
  - "Leading-edge" users make it costly to overprovision
  - Having admission control seems to be a better alternative

Other QOS principles

1. Admission Control
2. Marking of packets is needed to distinguish between different classes.
3. Protection (isolation) for one class from another.
4. While providing isolation, it is desirable to use resources as efficiently as possible

   => sharing.

How to Choose Service – Implicit

Network could examine packets and **implicitly** determine service class

- No changes to end hosts/applications
- Fixed set of applications supported at any time
- Can't support applications in different uses/modes easily
- Violates layering/modularity

How to Choose Service – Explicit

Applications could **explicitly** request service level

- Why would an application request lower service?
  - Pricing
  - Informal social conventions
  - Problem exists in best-effort as well → congestion control
- Applications must know network service choices
  - Difficult to change over time
  - All parts of network must support this → places greater burden on portability of IP

Overview

- Why QOS?
- **How do we build QOS?**
  - Todays lecture**: IntServ**
  - Next lecture: DiffServ

Components of Integrated Services

1. Type of commitment
   What does the network promise?
2. Packet scheduling
   How does the network meet promises?
3. Service interface
   How does the application describe what it wants?
4. Establishing the guarantee
   How is the promise communicated to/from the network
   How is admission of new applications controlled?

Components of Integrated Services

1. **Type of commitment**
   What does the network promise?
2. Packet scheduling
   How does the network meet promises?
3. Service interface
   How does the application describe what it wants?
4. Establishing the guarantee
   How is the promise communicated to/from the network
   How is admission of new applications controlled?

1. Type of commitment

What kind of promises/services should network offer?

➡ Depends on the **characteristics of the applications** that will use the network ….

Playback Applications

- Sample signal → packetize → transmit → buffer → playback
  - Fits most multimedia applications
- Performance concern:
  - Jitter – variation in end-to-end delay
    - Delay = fixed + variable = (propagation + packetization) + queuing
- Solution:
  - Playback point – delay introduced by buffer to hide network jitter

Characteristics of Playback Applications

- In general lower delay is preferable.
- Doesn't matter when packet arrives as long as it is before playback point
- Network guarantees (e.g. bound on jitter) would make it easier to set playback point
- Applications can tolerate some loss

Applications Variations

- Rigid & adaptive applications
  - Rigid – set fixed playback point (*a priori* bound)
  - Adaptive – adapt playback point (*de facto* bound)

Adaptive Applications

- Gamble that network conditions will be the same now as in the past
- Are prepared to deal with errors in their estimate
- Will in general have an earlier playback point than rigid applications
  - *A priori* bound > *de facto* bound
- Experience has shown that they can be built (e.g., vat, various adaptive video apps)

Applications Variations

- Rigid & adaptive applications
  - Rigid – set fixed playback point (*a priori* bound)
  - Adaptive – adapt playback point (*de facto* bound)
- Tolerant & intolerant applications
  - Tolerance to brief interruptions in service
- 4 combinations

Applications Variations

Only two classes of applications

1) **Intolerant and rigid**
2) **Tolerant and adaptive**

Other combinations make little sense

3) Intolerant and adaptive
   - Cannot adapt without interruption
4) Tolerant and rigid
   - Missed opportunity to improve delay

So what service classes should the network offer?

Type of Commitments

- **Guaranteed** service
  - For **intolerant and rigid** applications
  - Fixed guarantee, network meets commitment as long as clients send at match traffic agreement
- **Predicted** service
  - For **tolerant and adaptive** applications
  - Two components
    - If conditions do not change, commit to current service
    - If conditions change, take steps to deliver consistent performance (help apps minimize playback delay)
    - Implicit assumption – network does not change much over time
- **Datagram/best effort service**

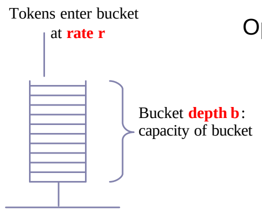
Components of Integrated Services

1. Type of commitment
   What does the network promise?
2. **Packet scheduling**
   How does the network meet promises?
3. Service interface
   How does the application describe what it wants?
4. Establishing the guarantee
   How is the promise communicated to/from the network
   How is admission of new applications controlled?

Scheduling for Guaranteed Traffic

- Use **token bucket filter** to characterize traffic
  - Described by rate r and bucket depth b
- Use **WFQ** at the routers
- Parekh's bound for worst case queuing delay = b/r

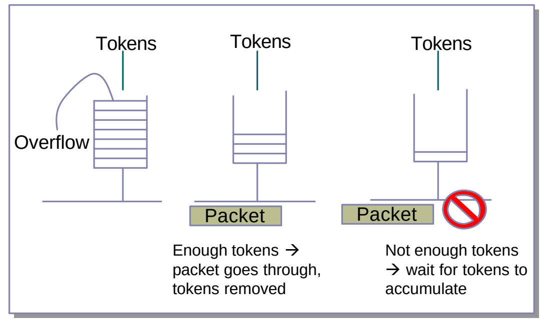
Token Bucket Filter



Operation:

- If bucket fills, tokens are discarded
- Sending a packet of size P uses P tokens
- If bucket has P tokens, packet sent at max rate, else must wait for tokens to accumulate

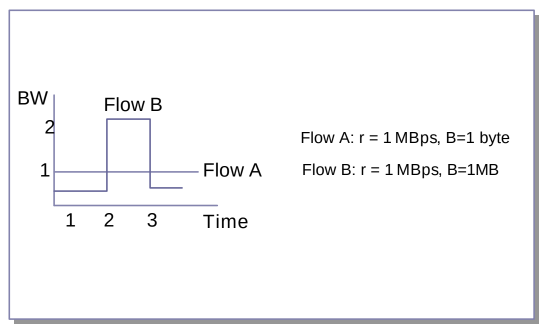
Token Bucket Operation



Token Bucket Characteristics

- On the long run, rate is limited to r
- On the short run, a burst of size b can be sent
- Amount of traffic entering at interval T is bounded by:
  - Traffic = b + r*T
- Information useful to admission algorithm

Token Bucket Specs



Possible Token Bucket Uses

- Shaping, policing, marking
  - Delay pkts from entering net (shaping)
  - Drop pkts that arrive without tokens (policing)
  - Let all pkts pass through, mark ones without tokens
    - Network drops pkts without tokens in time of congestion

Guarantee Proven by Parekh

- Given:
  - Flow *i* shaped with token bucket and leaky bucket rate control (depth *b* and rate *r*)
  - Network nodes do WFQ
- Cumulative queuing delay D_i suffered by flow *i* has upper bound
  - **D_i < b/r,** (where r may be much larger than average rate)
  - Assumes that Σr < link speed at any router
  - All sources limiting themselves to *r* will result in no network queuing

Predicted Service

Goals:

- Isolation
  - Isolates well-behaved from misbehaving sources
- Sharing
  - Mixing of different sources in a way beneficial to all

Mechanisms:

- WFQ
  - Great isolation but no sharing
- FIFO
  - Great sharing but no isolation

Predicted Service

- FIFO jitter increases with the number of hops
  - Use opportunity for sharing across hops
- FIFO+
  - At each hop: measure average delay for class at that router
  - For each packet: compute difference of average delay and delay of that packet in queue
  - Add/subtract difference in packet header
  - Packet inserted into queue based on order of average delay not actual delay

FIFO+ Simulation

- Simulation shows:
  - Slight increase in delay and jitter for short paths
  - Slight decrease in mean delay
  - Significant decrease in jitter
- However, more complex queue management
  - Packets are now inserted in sorted order instead of at tail of queue

Unified Scheduling

- Assume 3 types of traffic: guaranteed, predictive, best-effort
- Scheduling: use WFQ in routers
- Each guaranteed flow gets its own queue
- All predicted service flows and best effort aggregates in single separate queue
  - Predictive traffic classes
    - Multiple FIFO+ queues
    - Worst case delay for classes separated by order of magnitude
    - When high priority needs extra bandwidth – steals it from lower class
  - Best effort traffic acts as lowest priority class

Components of Integrated Services

1. Type of commitment
   What does the network promise?
2. Packet scheduling
   How does the network meet promises?
3. **Service interface**
   How does the application describe what it wants?
4. Establishing the guarantee
   How is the promise communicated to/from the network
   How is admission of new applications controlled?

Service Interface: Guaranteed Traffic

- Service interface
  - Specifies rate to network
    - Why not bucket size b?
  - If delay not good, ask for higher rate

Service Interface: Predicted Traffic

- Service interface
  - Specifies (r, b) token bucket parameters
  - Specifies delay D and loss rate L
  - Network assigns priority class
  - Policing at edges to drop or tag packets
    - Needed to provide isolation – why is this not done for guaranteed traffic?

Components of Integrated Services

1. Type of commitment
   What does the network promise?
2. Packet scheduling
   How does the network meet promises?
3. Service interface
   How does the application describe what it wants?
4. **Establishing the guarantee**
   How is the promise communicated
   How is admission of new applications controlled?

Establishing the guarantee

- Admission control
  - Don't give all bandwidth to real-time traffic
    - 90% real-time, 10% best effort
  - Very much dependent on how large fluctuations in network traffic and delay are
    - Should measure this dynamically instead of having built-in assumptions

IETF Internet Service Classes

- Guaranteed service
  - Firm bounds on e2e delays and bandwidth
- Controlled load
  - "A QoS closely approximating the QoS that same flow would receive from an unloaded network element, but uses capacity (admission) control to assure that this service is received even when the network element is overloaded"
- Best effort

Next Lecture: RSVP & DiffServ

- RSVP
- DiffServ architecture
- Assigned reading
  - [CF98] Explicit Allocation of Best-Effort Packet Delivery Service
  - [Z+93] RSVP: A New Resource Reservation Protocol

Parekh Bound on Delay Across Net

D_i = (bucket size/weighted rate allocated) + [(nhops - 1) * MaxPacketLen / weighted rate allocation] + Σ m=1 to hop_i (max packet length / outbound bw at hop)

- 1st term: delay when running at full speed
- 2nd term: packetization effects
- 3rd term: added delay due to packet approx of FQ (goes away as data rate increases)