Quality-of-Service

Quality of Service

What is Quality-of-Service?
• QoS refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements, or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates.

Why is this an issue?
• The default service in many packet networks is to give all applications the same service, and not consider any service requirements to the network. This is called a best-effort service.
Quality of Service

Who needs Quality-of-Service?
- Video and audio conferencing → bounded delay and loss rate
- Video and audio streaming → bounded packet loss rate
- Time-critical applications (real-time control) → bounded delays
- “valuable applications” → better service than less valuable applications

How are Quality-of-Service requirements specified?
- QoS parameters are
  - Delay
  - Delay Variation (Jitter)
  - Throughput
  - Error Rate

What is the granularity of QoS?
- Per-flow QoS
  → Guarantees are specified and enforced for single end-to-end data flow
- Class-based QoS
  → Guarantees are specified and enforced for groups of flows
Granularity of QoS

- **Per-flow guarantees**
  - Require per-flow reservations in the network
  - Require per-flow classification at routers

- **Per-class guarantees**
  - Bundle traffic flows with similar service requirements into “classes”
  - No per-flow reservations
  - Per-class guarantees do not immediately translate into per-flow guarantees
### Levels of QoS guarantees

**Absolute QoS guarantees**
- Assumes per-flow guarantees
- The QoS of a flow is independent of QoS of other flows
- Objective of service: “Isolation”
- Guarantees can be deterministic or statistical
- Needs resource reservation mechanism

**Adaptive QoS guarantees**
- Adapts QoS to load conditions in the network; potentially with upper and lower bounds.
- Objective of Service: “Fairness”
  - Does not need resource reservation
  - Needs adaptive feedback mechanism

**No QoS guarantees (Best Effort)**
- Objective of Service: “Sharing”
- Always works
Components of QoS in a Network

Types of QoS guarantees

- **Deterministic QoS**
  - Service guarantees are enforced for all traffic
  - For example, deterministic delay guarantees have the form:
    \[ \text{Delay of a packet from flow } X < D \]
    \( (D \text{ is called a delay bound}) \)

- **Statistical QoS**
  - Allows a certain fraction of traffic to violate the service guarantees
    \[ \text{Prob} \left[ \text{Delay of a packet from flow } X < D \right] > 1 - \varepsilon \]
  - Where \( \varepsilon \) is a small number (e.g., \( \varepsilon = 10^{-6} \))
Quality of Service Network

- Traffic description: 1 Mbps throughput, 100 kB max. burst
- Quality of Service parameters: $10^{-5}$ loss rate, 50 msec delay

Traffic Contract

- An application submits to network a reservation request for a new flow with a traffic specification and desired QoS
- Network performs admission control functions which determines if sufficient resources are available to
  (a) satisfy the desired QoS of new flow
  (b) without violating QoS of existing flows
- If sufficient resources are available, network reserves resources for flow; Otherwise, it rejects the flow
- Network performs policing and shaping at the network entrance to ensure that application adheres to its specification
Why is QoS Hard?

- QoS is easy to achieve if the network over allocates resources for each flow.
- Allocating bandwidth at the peak rate yields deterministic QoS, but it is very inefficient.

Challenge of QoS Networking:
- Build a network that can provide QoS with the least amount of resources.
  Using better algorithms has the same effect as adding bandwidth (or other resources).
- **Problem**: Complexity and scalability of QoS techniques.

Multiplexing Gain

\[
\left( \frac{\text{Resources needed to support guarantees for } N \text{ flows}}{N} \right) < \text{Resources needed to support guarantees for 1 flow}
\]

Sources of multiplexing gain:
- Traffic Conditioning (“Smoothing”)
- Scheduling
- Statistical Multiplexing Gain
Why is it hard to get multiplexing gain?

1. Burstiness of traffic
2. Stringent service guarantees

Traffic Conditioning

- Traffic conditioning is typically done at the network edge
Scheduling

- Scheduling algorithm determines the order in which traffic is transmitted
Designing Networks for Multiplexing Gain

Scheduling
- Earliest-Deadline-First
- GPS
- Static Priority
- FCFS
- Peak Rate Allocation
- Peak Rate
- Token Bucket
- Multiple Buckets

Traffic Characterization/Conditioning

Service / Admission Control
- Deterministic service
- Statistical service
- Average Rate
- Peak Rate Allocation
- Token Bucket
- Multiple Buckets
By now: The design space for deterministic guarantees is well understood.

Still open: Is there an elegant framework to reason about statistical guarantees? → Statistical Network Calculus
Components of a QoS Network
(with absolute per-flow guarantees)

1. At routers: Packet Classification, Packet Scheduling
2. At network entrance: Traffic conditioning
3. At routers or somewhere in the network: Admission Control
4. Between hosts and routers: Signaling

Classification and Scheduling

Routers need to be able to

1. classify arriving packets according to their QoS requirements
   → Packet Classification
2. isolate traffic flows and provide requested QoS
   → Packet Scheduling
Packet Classification

Packet classification is done before routing table lookup

Incoming Packet

Packet Classification

Classifier (Policy Database)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
</tr>
</tbody>
</table>

Forwarding Engine

Action

Packet Scheduling

- Packet scheduling algorithm determines the order in which backlogged packets are transmitted on an output link

Allocating output bandwidth

Controlling packet delay

scheduler
## Scheduling Algorithms

- **First-Come-First-Served** (FCFS or FIFO)
  - Cannot support different QoS

- **Static Priority (a.k.a. Head-of-line priority)**
  - Mechanism: Has one FCFS queue for each priority level; always serves the backlogged FCFS queue with highest priority
  - Can support delay differentiation, but tends to starve low priority traffic

- **Earliest-Deadline-First**
  - Mechanism: Assigns each packet a deadline (deadline = arrival time + delay guarantee of flow)
  - Requires to maintain a sorted queue

---

## Scheduling Algorithms

- **Fair Queueing**
  - Attempts to implement a scheduler that serves all flows with a backlog at the same rate
  - Emulates a bitwise Round Robin scheduling algorithm
  - Not completely trivial to implement Fair Queuing in a packet network

![Diagram of FIFO and Fair Queueing](image)
Suppose we have QoS constraints on delay:
• Flow 1:
• Flow 2:
• Flow 3:

Multiplexing Gain: Example

QoS constraints on delay:
• Flow 1:
• Flow 2:
• Flow 3:

Multiplexing Gain via Scheduling

• Here: fixed priority scheduling
Traffic Conditioning

- Traffic conditioning mechanisms at the network boundary need to enforce that traffic from a flow adheres to its specification
  - Policing
    - Drop traffic that violates the specification
  - Shaping
    - Buffer traffic at network entrance that violates specification
  - Marking
    - Mark packets with a lower priority or as best effort, if the traffic specification is violated

Traffic Conditioning

- The most popular traffic conditioning algorithm is the leaky bucket

A shaper buffers packets until a token becomes available. A policer drops a packet if no token is available. Each packet removes a token from the pool. If the pool is empty, the packet cannot enter.
Traffic Conditioning

- Leaky bucket with peak rate enforcement has two buckets:
  - Size $b$, rate $r$: controls average rate
  - Size 1, rate $P$ (with $P > r$): controls peak rate

This type of leaky bucket is used in ATM QoS and Internet QoS

Traffic Conditioning

- Maximum traffic that is admitted by leaky bucket over any time interval
Traffic Conditioning

- Maximum traffic that is admitted by dual leaky bucket over any time interval

Admission Control

- **Admission Control Conditions** determine if the network can support the QoS requirements for a given set of flows

- **Example**: End-to-end delay must be less than delays at all nodes

\[ D_j = d_{1j} + d_{2j} + d_{3j} \]
Distributed Admission Control

- **Example**: End-to-end delay must be less than a delay bound $D$

  ![Diagram]

  - Calculate smallest possible delay bound at each node: $d^*_1,d^*_2,d^*_3$ and reserve resources
  - At receiver:
    - If $D < d^*_1 + d^*_2 + d^*_3$, reject flow, send reject message to sender and release resources
    - If $D > d^*_1 + d^*_2 + d^*_3$, accept flow, commit resource reservation and notify sender

Centralized Admission Control

- **Example**: End-to-end delay must be less than a delay bound $D$

  ![Diagram]

  - Calculate smallest possible delay bound at each node: $d^*_1,d^*_2,d^*_3$ and reserve resources
  - At receiver:
    - If $D < d^*_1 + d^*_2 + d^*_3$, send reject message to sender
    - If $D > d^*_1 + d^*_2 + d^*_3$, accept flow and reserve resources
Signaling for QoS

- A network with resource reservations needs to have a control protocol which relays the reservation request:
  - ATM Networks: Q.2931
  - IP Networks: Reservation protocol (RSVP)

RSVP Functional Diagram

Source: Gordon Chaffee, UC Berkeley
Resource Reservation

- Senders advertise using PATH message
- Receivers reserve using RESV message
  - Flowspec + filterspec + policy data
  - Travels upstream in reverse direction of Path message
- Merging of reservations
- Sender/receiver notified of changes

Source: Gordon Chaffee, UC Berkeley