Network QoS
371-2-0213

Lecture 1

Gabriel Scalosub

Outline

Who, Where, When?

- LECTURER: Dr. Gabriel Scalosub
- EMAIL: sgabriel@bgu.ac.il
  Please include "network qos" in subject line
- LECTURES: Here...
- RECEPTION HOURS: coordinate in advance by e-mail.
- WEBSITE: accessible via BGU's Moodle:
  `http://moodle2.bgu.ac.il/`
- SYLLABUS: available on course's site
- SLIDES: available the day before each lecture (hopefully...)

Course Overview

- Design and analysis of protocols for providing *Quality-of-Service* (QoS):
  - Currently used protocols, and
  - New designs and models
- Focus on
  - Routing
  - Scheduling
  - Buffer management
  - Admission control
- A *research* oriented course
  - Mathematical maturity
  - Interest in research
  - Reading (books, and recent research papers)
  - Writing (English, LaTeX)
  - Thinking...

Course Requirements

- Homework:
  - 30%-50% of final grade
  - 2 assignments
  - Partly based on book chapters / research papers, aiming at:
    - preparing for the next lecture
    - filling-up missing proofs/concepts from the previous lecture
    - understanding basic definitions and models
    - experimenting with models via simulations
  - Submission in pairs
  - No late submissions

Course Requirements

- Final project:
  - 50%-70% of final grade
  - Assigned a research paper on a topic related to QoS
  - Write an 5-6 pages critical report (English, pref. in LaTeX):
    - present model/problems/solutions addressed in the papers
    - write a critical review of the model/solutions
    - propose ways to further address some of the problems
    - present preliminary steps in one of these paths
    - well-documented(!) implementation of one of the paper's solutions, and your extension, with comparison and discussion.
  - These are not meant to be paper summaries!!
  - Project goal: be a starting point for doing research (mainly targeted at MSc/PhD candidates)
  - Project topic:
    - each student will pick 3 possible papers related to QoS
    - one of the papers will be the assigned paper
    - details and deadlines: mid-semester

Course Requirements

Final grade:

$$FinalGrade = 0.3 \cdot HW + 0.5 \cdot FinProj + 0.2 \cdot \max\{HW, FinProj\}$$

- Fill out the online form:
  - `http://goo.gl/oZcf7`
  - Name, ID, Email, phone
  - Undergraduate/graduate, semester, advisor and subject (project/thesis)
  - Various questions:
    - what interests me in the course?
    - what related topics did I study?
    - am I taking the course for sure?, etc.
- "The lecture in retrospect"
  - `https://forms.gle/scwKw1CGbsH2qYnS6`
  - please try and fill after each lecture (1min...)
  - allows me to know where you stand

Syllabus

(order might be somewhat different)

- Introduction to QoS
- Models for QoS: IntServ, DiffServ
- Protocols (e.g., RSVP)
- Network Calculus
- Scheduling algorithms (e.g., FIFO, WFQ, etc.)
- Adversarial queuing theory
- Buffer management algorithms

Time permitting we will address additional related topics, e.g.: caching, QoS in wireless networks, etc.

Bibliography

- Kurose and Ross, "*Computer Networking: A Top-Down Approach*", 5th ed., Addison-Wesley, 2010.
- Peterson and Davie, "*Computer Networks: A Systems Approach*", 4th ed., Morgan Kaufmann, 2007.
- Wang, "*Internet QoS: Architectures and Mechanisms for Quality of Service*", Morgan Kaufmann, 2001.
- Kumar, Manjunath, and Kuri, "*Communication Networking: An Analytical Approach*", Elsevier, 2004.
- Le Boudec and Thiran, "*Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*", Springer, 2001.
- Research papers (will be listed in slides / on website)

Motivation for QoS

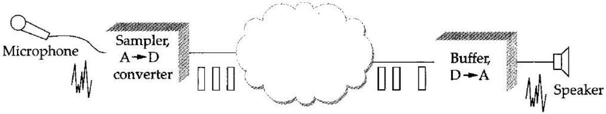
- At the airport...
    - How long should I arrive before my flight?
    - What if I fly business?
    - What if I'm a foreign minister?
- In a communication network:
    - The need emerges at the *user*
        - I want to make a phone call over IP
    - Translated to a technical requirement by the *application*
        - I need 64Kbps bandwidth, and
        - I need $<$300 ms end-to-end delay, and ...
    - Also benefits the network as a whole

What's QoS all about?

Satisfying application needs, and enabling new services

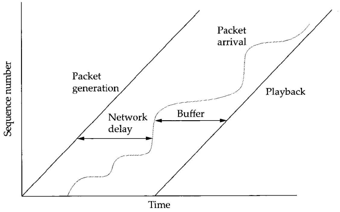
What Do Applications Want?

- Consider a real-time audio application (e.g., VOIP)
  - Audio sampled at rate of one per 125μs
    - 8000 samples/sec, 8 bit/sample $\Rightarrow$ 64Kbps
  - Should be played back at the same rate
  - Every sample has a *playback deadline*
  - Samples that arrive late / dropped are useless



What Do Applications Want?

- What can we do?
  - Ensure samples take *exactly* the same time to traverse the network
    - samples actually *arrive* at their injection rate
    - virtually impossible, due to, e.g., variable queuing delays at routers
  - Playback buffer
    - force some extra delay by buffering: a playback offset
    - insurance against variable network delay
    - $<$300 ms offset is still acceptable for real-time audio



Measures of Network QoS

- main performance metrics:

Metric	high priority	low priority
Delay	Audio (real-time)	FTP (elastic)
Loss	Control Systems	Multimedia
BW	Video (adaptive codecs)	Email
Jitter	Control Systems	HTTP

QoS means:

- Performance assurance
- Service differentiation

- Means for providing QoS
  - Resource allocation
  - Performance optimization

Resource Allocation Architectures

- Main problems in networks arise from drops/delays, e.g.:
  - Applications: bumpy audio/video
  - Network: increased congestion due to TCP retransmissions
- Network resources (BW, buffers, ...) shared by all the users
- QoS support must be active about resource allocation
- Most current networks: best-effort
  - "All packets were created equal"
  - FIFO scheduling
  - No admission control
    - rate regulation based on AIMD (e.g., TCP)
    - "abuse-friendly": multiple TCP-connections, tweaking TCP-stack
- Big question: how should resources be allocated?

Resource Allocation Architectures

- Integrated Services (IntServ)
  - Early 1990s
    - Internet Engineering Task Force (IETF) IntServ workgroup
    - Goal: standardize new resource allocation architecture for QoS
    - Main motivation: real-time / multimedia
  - Per-flow resource reservation
    - flow estimates its requirements
    - routing protocol finds end-to-end path
    - per-hop admission control verifies sufficient resources are available
    - Protocol: RSVP (Resource reSerVation Protocol)
  - Two service models
    - Guaranteed service: deterministic worst-case delay guarantees
    - Controlled load service: similar to lightly-loaded best-effort
  - Handicaps:
    - accounting (e.g., inter-AS)
    - scalability (per-flow state)

Resource Allocation Architectures

- Differentiated Services (DiffServ)
  - Late 1990s
    - Internet Engineering Task Force (IETF) DiffServ workgroup
    - Goal: service classes, better than best-effort
    - address handicaps of IntServ
  - Based on:
    - dividing traffic into small number of forwarding classes
    - not “per-flow”
    - edge policing: per-class limit on traffic injected (edge)
    - provisioning: providers adjust resources allocated
    - traffic prioritization: scheduling/buffer management at routers
    - packets marked at the edge, subject to customer-provider Service Level Agreement (SLA)
    - not trivial, but rates are much lower at the edge
  - Handicaps:
    - no reservation
    - difficult and expensive to provide deterministic QoS guarantees

Performance Optimization

- Multiprotocol Label Switching (MPLS) (late 1990s)
  - Goal: Enhance IP networks
    - establishing virtual circuits (like ATM)
    - enables reservation and marking (IntServ/DiffServ-like)
    - facilitates VPNs
    - allows traffic engineering
  - Enables explicit control of paths of flows (via header labels)
    - not restricted by IP destination-based prefix lookup
- Traffic engineering
  - Goal: optimize usage of network resources
    - via provisioning and control of network flows
  - Metrics include:
    - maximize utilization, minimize congestion, load balancing
  - Sophisticated routing
    - constraint-based (not destination-based)
    - needs "global" information: topology, traffic (mostly intra-AS)
    - uses MPLS capabilities

Competitive Analysis - Definition

- Given an instance I of an optimization problem $\mathcal{P}$, denote by $\mathrm{OPT}(I)$ the value of an optimal feasible solution for I.
- In the online setting, the input to the problem is made available in parts.
- An online algorithm A is said to be c-competitive for problem $\mathcal{P}$ if for every instance I of $\mathcal{P}$, $A(I)$ satisfies:
  - $A(I) \leq c \cdot \mathrm{OPT}(I) + \alpha$ (minimization problem $\mathcal{P}$)
  - $A(I) \geq \frac{1}{c} \cdot \mathrm{OPT}(I) - \alpha$ (maximization problem $\mathcal{P}$)

  where $\alpha \geq 0$ is some additive term independent of I.
- Common to assume that I is generated by an *adversary*
- The online problem is then viewed as a *game*:
  - Adversary (produces I and an optimal solution to I), vs.
  - Algorithm

Example: Ski Rental

- You're going on your first skiing vacation
  - Should I buy skis? (cost a lot!!, say, B)
  - Should I rent skis? (cost less, say, R/day)
- Problem:
  - All the snow might melt in the middle of the vacation.
    - it's the cheap Pesach package, remember?
  - You don't know if/when.
- If I end up renting, and it never melts, I pay a lot.
  - Would have been cheaper to buy.
- If I buy up front, and it melts on the second day:
  - Would have been better to rent.
- Online input: good days, and then it melts
  - Adversary controls the number of good days

Example: Ski Rental

- Online algorithm PIS (Play-It-Safe):
  - Rent for k days, until $kR = B$ (assuming snow didn't melt).
  - If on day $k+1$ the snow didn't melt, buy.

Theorem

Algorithm PIS is 2-competitive

Proof.

- If snow melts before you buy - you're optimal.
- Otherwise:
  - You paid $2B$,
  - OPT would have paid only B (bought up front)
- In any case, you didn't pay more than twice what OPT paid.

$\square$

QoS Admission Control Problem

- Input:
  - Directed graph $G = (V, E)$, with edge capacities $u : E \mapsto \mathbb{R}$.
  - Set of connection requests arriving online $C = \{R_i = (s_i, t_i, b_i, a_i, d_i, p_i) \mid 1 \leq i \leq m\}$ s.t.
    - s_i: source node
    - t_i: target node
    - b_i: required BW
    - a_i: arrival time
    - d_i: duration
    - p_i: profit
- Goal:
  - Accept/reject (and route!) requests s.t. profit of accepted requests is maximized
  - Feasibility: always satisfy capacity constraints,

$$\sum_{i \mid \text{ using } e} b_i \leq u(e)$$

  - Non preemptive, non re-routable

AAP$_\varepsilon$ Algorithm

Awerbuch-Azar-Plotkin, 1993

- Offline problem is NP-hard
  - All parameters are 1: disjoint paths
- Simplifying assumptions:
  - Infinite duration, homogeneous BW and profits
  - WLOG, $p_i = D$ ("diameter"), $b_i = 1$
- Notation:
  - A_i: admitted requests before getting R_{i+1}
  - π_i: path of an admitted request R_i
  - $L_i(e)$: *load* on edge e before getting R_{i+1},

$$L_i(e) = \frac{1}{u(e)} \sum_{j \in A_i | e \in \pi_j} 1$$

Feasibility: $L_i(e) \leq 1$

  - $C_i(e)$: *cost* of edge e before getting R_{i+1},

$$C_i(e) = u(e) \left[\mu^{L_i(e)} - 1\right]$$

where (for some $\varepsilon > 0$)

$$\mu = 2^{1+1/\varepsilon} \cdot D$$

AAP_ε Algorithm

Awerbuch-Azar-Plotkin, 1993

- ε will be a "tradeoff" parameter
- Assume capacities are relatively big, i.e.

$$u(e) \geq 1 + \varepsilon + \varepsilon \log D$$

Algorithm 1 AAP_ε: Upon receiving request R_i

1: find a minimal-length s_i-t_i path π_i w.r.t. edge length $\frac{C_{i-1}(e)}{u(e)}$ $\uparrow$ $\boxed{\mu^{L_i(e)} - 1}$
2: **if** π_i's length is less than D **then**
3: admit R_i on path π_i
4: **else**
5: reject R_i
6: **end if**

AAP$_\varepsilon$ - Feasibility

Theorem (Feasibility)

AAP$_\varepsilon$ produces a feasible solution.

Proof. (by induction on request index i)

- Trivial for $i = 0$.
- Assume $L_{i-1}(e) \leq 1$ for all $e \in E$.
- If R_i is rejected, nothing changes.
- Assume R_i is admitted on path π_i.
- For every $e \notin \pi_i$, nothing changes.
- Consider $e \in \pi_i$. By the admission criteria

$$\frac{C_{i-1}(e)}{u(e)} \leq \sum_{e' \in \pi_i} \frac{C_{i-1}(e')}{u(e')} \leq D$$

AAP$_\varepsilon$ - Feasibility

Theorem (Feasibility)

AAP$_\varepsilon$ produces a feasible solution.

Proof. (by induction on request index i)

- By the assumption that capacities are big $\boxed{u(e) \geq 1 + \varepsilon + \varepsilon \log D}$

$$\frac{C_{i-1}(e)}{u(e)} = \mu^{L_{i-1}(e)} - 1$$
$$= \mu^{L_i(e) - 1/u(e)} - 1$$
$$\geq \mu^{L_i(e) - 1/[1+\varepsilon+\varepsilon \log D]} - 1$$
$$= \mu^{L_i(e) - 1/\varepsilon \log \mu} - 1$$
$$= \frac{\mu^{L_i(e)}}{2^{1/\varepsilon}} - 1$$

$\boxed{\mu = 2^{1+1/\varepsilon} D}$

$\boxed{\log \mu = 1 + 1/\varepsilon + \log D}$

$\boxed{\mu^{1/\varepsilon \log \mu} = 2^{1/\varepsilon}}$

take log...

AAP$_\varepsilon$ - Feasibility

Theorem (Feasibility)

AAP$_\varepsilon$ produces a feasible solution.

Proof. (by induction on request index i)

- Combining the above with the previous inequality we get

$$\frac{\mu^{L_i(e)}}{2^{1/\varepsilon}} - 1 \leq D$$

- It follows that

$$\mu^{L_i(e)} \leq (D+1)2^{1/\varepsilon} \leq 2^{1+1/\varepsilon} D = \mu$$

which implies $L_i(e) \leq 1$. □

AAP$_\varepsilon$ - Competitive Ratio

Theorem (Competitive Ratio)

AAP$_\varepsilon$ is $(1 + 2^{1+1/\varepsilon} \log \mu)$-competitive.

Some additional notation:

- A^*: requests admitted by some optimal solution OPT.
- $\tilde{A} = A^* \setminus A_m$: requests admitted by OPT but not by AAP$_\varepsilon$.
- $C_i = \sum_{e \in E} C_i(e)$: potential function
  - (a.k.a. *Lyapunov function* in stochastic analysis)

We will use the following lemmas:

- **Lemma 1.** $C_m \geq D \left|\tilde{A}\right|$
- **Lemma 2.** $C_i \leq D\ 2^{1+1/\varepsilon}\ \log \mu\ \ |A_i|$ (for all i)

AAP$_\varepsilon$ - Competitive Ratio

Theorem (Competitive Ratio)

AAP_ε is $(1 + 2^{1+1/\varepsilon} \log \mu)$-competitive.

Proof (theorem).

Since $|A^*| \leq |A_m| + \left|\tilde{A}\right|$, we get

$$\begin{aligned} D\,|A^*| &\leq D\,|A_m| + D\left|\tilde{A}\right| \\ &\leq D\,|A_m| + D\;2^{1+1/\varepsilon}\;\log\mu\;\;|A_m| \\ &= D(1 + 2^{1+1/\varepsilon}\;\log\mu)\,|A_m| \end{aligned}$$

$\square$

AAP$_\varepsilon$ - Competitive Ratio

Lemma (1)

$C_m \geq D \left|\tilde{A}\right|$

Proof.

- Consider some $R_i \in \tilde{A}$, routed on some path π_i^* by OPT.
- R_i is rejected by AAP$_\varepsilon$: $\sum_{e \in \pi_i^*} \frac{C_{i-1}(e)}{u(e)} > D$
- $$\begin{aligned} \left|\tilde{A}\right| D &< \sum_{i \in \tilde{A}} \sum_{e \in \pi_i^*} \frac{C_{i-1}(e)}{u(e)} \\ &\leq \sum_{e \in E} C_m(e) \sum_{i \in \tilde{A} | e \in \pi_i^*} \frac{1}{u(e)} \\ &\leq \sum_{e \in E} C_m(e) L_m^*(e) \\ &\leq \sum_{e \in E} C_m(e) = C_m \end{aligned}$$

$\square$

AAP_ε - Competitive Ratio

Lemma (2)

$C_i \leq D\ 2^{1+1/\varepsilon}\ \log\mu\ |A_i|$ *(for all i)*

Proof (by induction on request index i):

- For $i = 0$, LHS=RHS=0.
- Assume $C_{i-1} \leq D\ 2^{1+1/\varepsilon}\ \log\mu\ |A_{i-1}|$.
- If R_i is rejected, nothing changes.
- Assume R_i is admitted on path π_i.
- It suffices to show:

$$C_i - C_{i-1} = \sum_{e \in E} (C_i(e) - C_{i-1}(e)) \leq D\ 2^{1+1/\varepsilon}\ \log\mu$$

AAP$_\varepsilon$ - Competitive Ratio

Lemma (2)

$C_i \leq D\ 2^{1+1/\varepsilon}\ \log\mu\ |A_i|$ *(for all i)*

Proof (by induction on request index i):

- For every $e \notin \pi_i$, $C_i(e) = C_{i-1}(e)$.
- Consider any $e \in \pi_i$.

$\boxed{C_i(e) = u(e)[\mu^{L_i(e)} - 1]}$

$$\begin{aligned} C_i(e) - C_{i-1}(e) &= u(e)\left(\mu^{L_{i-1}(e)+1/u(e)} - \mu^{L_{i-1}(e)}\right) \\ &= u(e)\ \mu^{L_{i-1}(e)}\ \left(\mu^{1/u(e)} - 1\right) \\ &= u(e)\ \mu^{L_{i-1}(e)}(2^{\frac{\log\mu}{u(e)}} - 1) \\ &\leq u(e)\ \mu^{L_{i-1}(e)}\ \frac{\log\mu}{u(e)}\ 2^{1/\varepsilon} \\ &= 2^{1/\varepsilon}\log\mu\left(\frac{C_{i-1}(e)}{u(e)} + 1\right) \end{aligned}$$

$\boxed{\frac{\log\mu}{u(e)} \leq \frac{1+1/\varepsilon+\log D}{\varepsilon+1+\varepsilon\log D} = \frac{1}{\varepsilon}}$

$\boxed{2^y - 1 \leq y2^{1/\varepsilon} \quad \forall y \in [0, 1/\varepsilon]}$

AAP_ε - Competitive Ratio

Lemma (2)

$C_i \leq D\ 2^{1+1/\varepsilon}\ \log\mu\ \ |A_i|$ *(for all i)*

Proof (by induction on request index i):

- Summing,

$$\begin{aligned}\textstyle\sum_{e\in\pi_i}(C_i(e) - C_{i-1}(e)) &\leq 2^{1/\varepsilon}\log\mu \textstyle\sum_{e\in\pi_i}\left(\frac{C_{i-1}(e)}{u(e)} + 1\right)\\ &= 2^{1/\varepsilon}\log\mu\left[\textstyle\sum_{e\in\pi_i}\frac{C_{i-1}(e)}{u(e)} + \sum_{e\in\pi_i} 1\right]\\ &\leq 2^{1/\varepsilon}\log\mu\,[D + D]\\ &= D\ 2^{1+1/\varepsilon}\log\mu\end{aligned}$$

$\square$

Summary and Related problems

- Choosing constant ε yields $O(\log |V|)$-competitive algorithm.
- Related problems:
  - Min-Congestion:
    - accept *all* requests, but try to minimize maximum load

$$\max_{e \in E} L_m(e)$$

    - network provisioning for supporting QoS
    - $O(\log |V|)$-competitive algorithms exist
  - Oblivious Routing (initially due to Räcke 2002)
    - precompute routing table
    - independent(!!) of actual traffic
    - $O(\text{poly} \log |V|)$-competitive
- References:
  - Awerbuch, Azar and Plotkin, Throughput-competitive online routing. In FOCS 1993, pp. 32-40.