Network Optimization by Randomization

Summer Semester 2011

TU Berlin

Organization

- Instructor: Florin Ciucu
  - Senior Research Scientist, INET Chair (Prof. A. Feldmann), Deutsche Telekom Laboratories / TU Berlin
  - Ph.D. in Computer Science, University of Virginia, 2007
  - M.Sc. in Computer Science, George Mason University, 2002
  - B.Sc. in Informatics, University of Bucharest, 1998
  - Research interests: stochastic models for network analysis, resource allocation, randomized algorithms

- Additional Instructor: Stefan Schmid
  - Senior Research Scientist, INET Chair (Prof. A. Feldmann), Deutsche Telekom Laboratories / TU Berlin
  - Dr. in Science, ETH Zurich, 2008
  - Diploma / M.Sc. in Computer Science, ETH Zurich, 2004
  - Research interests: Networks & Distributed Systems, Algorithms, Game Theory & Incentives, Virtualization

Organization (contd.)

Time and Place

- Lecture: Thursdays 12:00 – 14:00
- Tutorial: Thursdays 14:00 – 16:00
- Room: FR 1067

- Prerequisites (desirable): algorithms, data structures, basic probability
  - Yes, the course involves math (proving that certain things work, and explaining why certain things are the way they are)
  - … but the level of math is supposed to be for an audience with the primary background in Computer Science / Engineering

- Lectures
  - Focus on concepts, theoretical aspects, and examples
  - Material (algorithms, analysis) to be mostly presented on the black-board; some supporting slides to be handed-in before the lecture
  - Students must take notes

- Tutorials
  - Focus on additional examples, solutions for homeworks

Organization (contd.)

- Textbooks (cover partial material)
  - M. Mitzenmacher and E. Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis
  - M.J. Neely. Stochastic Network Optimization with Application to Communication and Queueing Systems

- Homeworks
  - Assigned every week
  - To be handed in one week later during class
  - Late returns are possible (subject to “good” excuses)
  - To be graded by the instructor(s)
  - 50% of the points needed to pass the tutorial

- Final Exam
  - Written (2-3 hrs)
  - Contains problems closely related to HWs

- Updated Info at

  http://www.net.t-labs.tu-berlin.de/teaching/ss11/NOR_lecture/

Why Should You Take This Course?

- The course it's about Computer and Communication Networks
  - Perhaps one of the greatest discoveries of the last century
  - Definitely part of the future

- The course it's about algorithms, i.e., what Computer Scientists like to do anyway

- The course it's about the use of probability in Computer Science
  - Not exactly what Computer Scientists like to do
  - … But extremely useful to understand modern systems and design efficient algorithms
- The course will improve your ability to think abstractly
  - And to solve hard problems efficiently

What's The Course About?

- Efficient Algorithms for Network Protocols
  - What does "efficient" mean? Fast? Reliable?
  - Why care? Improve networks' performance (i.e., optimize)

- We look at a specific class of algorithms including a random (or probabilistic) component
  - More precisely: it makes random choices during execution
  - What does "random" mean? Unpredictable??? Hopefully not entirely… Let's leave this concept vague for now …

What is an Algorithm Anyway?

- Intuitive but informal definition: a collection of simple instructions, to be executed in some order, for carrying out some specific task

- There is a problem with this definition
  - Cannot be used to prove that some tasks are algorithmically unsolvable (i.e., no algorithms exist)
  - E.g., Hilbert 10^{th} problem: testing the existence of an integral root for a polynomial; and many others…

- Church-Turing thesis

  Intuitive notion of algorithms = Turing machine algorithms

- Computability Theory – classify problems in solvable / not solvable
- Complexity Theory – classify problems in easy and hard

Why Randomness?

- What is it? Does it exist? ("The randomness is the unknown, and that the nature is determined in its fundamentals.", Democritos)

- Hard time along history
  - Newton's law: the universe is deterministic (a "big" computer would roughly predict the future, subject to some "exact" initial conditions)
  - Belief in determinism had emotional roots, as people connected randomness with chaos, uncertainty, and unpredictability, all related to fear ("There are only two possibilities, either a big chaos conquers the world, or order and law.", M. Aurelius)

- (Scientific) acceptance as late as in the 20th century
  - Invention of modern Quantum Mechanics: the universe evolves according to laws, but the backbone of the laws is random
  - Confirmation of the ancient view of Epikurus ("The randomness is objective, it is the proper nature of events.")

- Randomness is commonly regarded as an essential component to model the universe/nature, and for what we care computer and communication systems

Why Randomized Algorithms?

- **Efficiency**: execution time (or space requirement) can be much smaller than that of the best deterministic algorithm for some specific problems

- **Simplicity**: Much easier to understand and implement

- These gains come at a price
  - Correctness may be uncertain
  - Efficiency may be also uncertain
  - However, these uncertainties can be quantified mathematically (e.g., the probability of error is 2%)

- If the probability of error is “small enough” (or tolerable) then it’s worth paying the price

Network Areas Covered in This Class. Tentative

- Multiple Access Protocols
- Scheduling Algorithms (e.g., Backpressure)
- Flow and Congestion Control, Routing
- Resource Allocation
- Peer-to-Peer and Social Networks
- Online Algorithms
- Network Security
- Network Coding
- Network Reliability

- First few lectures focus on classical algorithms (e.g., sorting, selection) and topics (e.g., random number generators)

Back to (Efficient) Algorithms

- Recall Growth of Function (big-Oh, little-oh)
- Problem: Multiplication of two big numbers using only a piece of paper and a pencil
  - Naive
  - Classic
  - à la russe
  - Karatsuba
- Example: A randomized algorithm more efficient than any deterministic algorithm