Course Description: 6.852, Distributed Algorithms

1 What this course is about

The field of distributed algorithms has become a well-developed research area over the past 20 years. Its results appear in specialized research conferences such as PODC and DISC, and in general conferences involving distributed computing. Researchers working in this field design new distributed algorithms, articulate the problems that the algorithms are supposed to solve, model the algorithms formally, prove their correctness, and analyze their performance. They also prove impossibility results and lower bounds, which explain why certain tasks cannot be carried out in distributed settings, or cannot be carried out at a certain cost. The algorithms that are studied typically solve problems that arise in practical distributed computing, such as resource allocation, distributed agreement, and management of shared data. Because it is a sub-area of theoretical computer science, the ideal style of the work is mathematically rigorous (though preliminary papers in this area are often less rigorous than the ideal).

6.852 is a graduate-level course that is intended to do two things:

1. Provide a rigorous introduction to the most important results in the area of distributed algorithms.
2. Prepare interested students to carry out independent research in distributed algorithms.

Usually, the students who take the course are a mixture of PhD students and MEng students. These two groups generally have different goals and different levels of preparation. Therefore, the course this year will have two tracks: the "Basic" track will focus on the introductory material, while the "Advanced" track will focus on preparation for research. We expect that, in general, PhD students in EECS will take the Advanced track (and we will require this for students who want to use the course for advanced TQE credit), whereas MEng students will take the Basic track. We will try to ensure that the two tracks have about the same amount of work, but the specific requirements will be a little different: students in the Basic track will be expected to do some extra homework problems on the basic material, whereas students in the Advanced track will be expected to read and understand some recent research results that aren't covered in the regular lectures, and explain them to the rest of the class. The new results will be in the areas of fault-tolerant computing, computing in highly dynamic environments, and computing in mobile systems.

We will ask you to specify which track you want to join, and to provide a preliminary idea of what results you are interested in presenting, when you hand in your solutions to the first problem set, on September 20th.

2 People and places

Instructor: Prof. Nancy Lynch, NE43-365, MIT extension 3-7225, lynch@theory.lcs.mit.edu. Available after class and by appointment.

Teaching assistant: Ziv Bar-Joseph, NE43-441, MIT extension 8-5847, zivbj@mit.edu Office hours: Wednesday 3:00-5:00, in front of room NE43-441.

Course secretary: Joanne Talbot Hanley, NE43-366, extension 3-6054

Class meetings: Tuesdays and Thursdays, 9:30-11:00AM in room 4-163.

Web site and mailing list: We will set up a course mailing list. Please send email right away to Ziv, at zivbj@mit.edu, to join this list.

The course Web site is at URL http://theory.lcs.mit.edu/classes/6.852/01. This site will contain downloadable copies of course handouts and related research papers, and other information about the course.

3 Prerequisites

To take 6.852, you should have:

- Mathematical maturity. In particular, you should be very good at reading and writing mathematical proofs.
- General knowledge about distributed systems. For instance, MITs undergraduate course 6.033, entitled Computer Systems Engineering would be good background.
- Experience with sequential algorithms and their analysis. MITs undergrad course 6.046 would be ideal.
- (Desirable, but not essential) Experience with formal models of computation. MITs course 6.045 on automata theory would be fine for this.

4 Source material

The main source will be the book *Distributed Algorithms* by Nancy Lynch. See the course Web site for some information on finding and purchasing the book. Quantum Books has ordered copies for us.

The latest printing of the book is the fourth printing, so look for a fourth printing copy. This has only a few corrections over the third printing, so using a third printing copy would be OK too. Earlier printings than that have various known errata. These are collected in errata lists, which are accessible from the course Web page.

The book refers to many papers from the research literature on distributed algorithms; you might want to track down and read some of these.

Other books that you might find useful are:

- Hagit Attiya and Jennifer Welch *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. McGraw-Hill, 1998.
  This is another textbook on distributed algorithms, published a little after the Lynch book. The material covered is similar. The style is less formal. We have put two copies of this book on reserve in the LCS Reading Room in NE43.

- Kenneth P. Birman. *Building Secure and Reliable Network Applications*, Manning Publications, 1996.
  This gives a good general introduction to the area of reliable distributed computing. It is written from the perspective of a system designer/developer rather than a theoretician.

- Shlomi Dolev. *Self-Stabilization*, MIT Press, 2000.

In addition, some research papers that are not covered in the textbook will be covered in class and on problem sets. Other research papers will provide the basis for the Advanced students reading and presentations. These papers are listed in a separate handout, Handout 4. We will put as many of these as we can on the course Web site.

5 Course requirements

5.1 Problem sets

These are intended to help you to understand the material being covered in class. Most problems will involve thinking about algorithms and problems already covered in class; some will be designed to get you started thinking about ideas to be discussed in later classes.

Specifically, about six problems will be given out every Thursday. The problems will be batched and due every two weeks, at the beginning of class on alternate Thursdays. There will be a total of six problem set due dates. Of the (approximately) six problems given out each week, (approximately) four will be for everyone to do, while the other (approximately) two will be for the Basic students only. These will generally be more basic than the other problems.

No late homeworks will be accepted.
In case of an emergency, please talk to either Ziv Bar-Joseph or Prof. Lynch.

Homework is an important part of your grade. When grading homework problems, we will try to give full credit to solutions that include all the important logical steps and ideas. We consider it a minus for a writeup to be lengthy and overly detailed. An exception is when we specifically ask for details, for instance, in formal proofs of correctness of algorithms.

Solutions to homework problems will be handed out. Whenever possible, the best student solution will be used. Students who would like their writeups to be handed out can help us by writing elegant and concise solutions and formatting them using LaTeX/LaTeX2$_\epsilon$ (and the LaTeX style files provided by us). When you submit the homework, keep the .tex file since it may need to be edited if your solution is chosen. Problem sets will be graded by teams of students in the class, led by Ziv.

Policy on homework collaboration: You are *strongly encouraged* to discuss possible solutions with other class members. Many students in past instances of this course have formed homework discussion groups. However, *you must always write up the solutions entirely on your own.*

5.2 Problem set grading

For each problem set, a group of about 5 students will be responsible for working with the course staff to grade the solutions. We would like the grading to be completed by the Monday afternoon after the homeworks are handed in, so we can record the grades and hand them back on Tuesday. The number of times you have to grade over the course of the semester will depend on the size of the class. Part of your grade will depend on the quality and promptness of your work on problem set grading.

5.3 Reading and presentations

The last four classes will be devoted to presentations of new papers, not covered in the textbook, by students in the class. The main responsibility for this will rest on the Advanced students, but the Basic students will still be expected to read the papers, participate in class, and perhaps work out some homework problems based on this material.

The Advanced students will work either individually or in small groups (no more than 3) to read and present papers from those listed in Handout 4. Of course, to avoid boring the class, different groups have to choose different papers. You will be expected to read the chosen papers completely, understand them in depth, and present them to the rest of the class. You should not repeat what the authors say literally, but should try to extract the important ideas from the papers and present them in a clear and economical way. This may mean, for example, reading other related papers, restating the authors definitions and results, filling in missing details, or relating the work to other things covered in class.

You should start working on this project as early as possible in the semester, and work on it in parallel with learning the basic material. We will ask you to give a preliminary idea of what papers you might be interested in very early in the term, when you hand in your first problem set on September 20th. Then, after we try to help sort out conflicting choices, we will ask you to finalize choices and groups by the time you hand in your second problem set on October 4th. After that, we expect each group to meet at least twice with Ziv or Prof. Lynch in preparing the presentations. These meetings should discuss (a) what you think the key ideas are, and (b) how to present them to the class. Dont wait too long to meet for (a)—you should do this before Nov. 15.

5.4 Exams

There will be no exams. No midterm, no final. You can go home after the last class, on December 11.

5.5 Course grade calculation

Your course grade will be based on problem set grades, problem set grading grades, class participation, and (for Advanced students) presentation grades. Here is how your grade will be calculated:

- Problem sets: Basic students: 90% (15% for each problem set)
  Advanced students: 60% (10% for each problem set)

- Grading and class participation: 10%

- Presentation: Advanced students only: 30%