

Teaching Statement

Philip Brighten Godfrey

January 17, 2008

Teaching is exciting for me and is critically important for the discipline of computer science: for generating versatile new computer programmers, for generating new research ideas, and for generating new researchers. With this in mind, I want to emphasize several themes that guide my teaching.

- **Challenging the status quo.** Learning an idea is useful; learning how to learn ideas is better. I want students to go a step farther and *learn how to learn ideas that no one else has learned before*—that is, how to innovate. Students should be continually encouraged to skeptically challenge the status quo, to imagine what *might* exist. As an example, for decades Internet communications have almost exclusively used either TCP (a reliable stream-based protocol that controls congestion) or UDP (an unreliable datagram protocol that ignores congestion). I would feel a moment of success if my students ask, why should these particular combinations of features be the only two choices? Why not, say, an unreliable yet congestion-controlling protocol? Actually, there is good reason for such a protocol for real-time applications, and it is the subject of recent work.¹ My teaching will be geared towards provoking these questions. I will ask for nonstandard solutions to real problems, and I will emphasize recent research advances in order to teach innovation by example and keep material fresh.
- **Interactivity.** Interaction among students and between the teacher and students helps all of us learn from each other and stay interested. Lectures should have a strong interactive component. In an undergraduate data structures class, I sometimes had students execute algorithms “live” in recitation, such as executing merge sort with each person performing one recursive call. Interactivity can go beyond the lecture, too. One idea appropriate for a networking course is to have students build a software router and connect the students’ routers together in an overlay network. Evaluation of each router would then be based in part on the number of *other* students’ routers with which it could maintain an end-to-end flow, with live statistics on the course web page. This would encourage students to collaborate on getting their routers to speak with each other successfully.
- **Conveying ideas effectively.** This is probably the most basic task in teaching and depends on two key elements, both of which I especially enjoy. One is figuring out how to explain a concept in multiple ways “on the fly” during a lecture or while working with students one-on-one. The second key element is getting feedback on what explanations worked or didn’t work, why, and for whom. This involves personal sensitivity to students, and is facilitated by having multiple channels through which to receive feedback. One channel which I found very helpful was to accept anonymous questions and comments on note cards after discussion sections. Another channel I have used is more empirical: when presenting my work on “Minimizing Churn in Distributed Systems” in a graduate course at Berkeley, I included three separate explanations for why simply randomizing node selection reduces churn, and polled the audience on which one made the most sense. One of the three came out the clear winner, which showed me which explanation was effective and which needed to be revised or dropped. The more traditional channels of accepting feedback during lectures, office hours, and by email are also critical.

At Berkeley, I had the opportunity to be a teaching assistant for an undergraduate data structures course, including a discussion section and a lab, and for a graduate algorithms course. My teaching evaluations in these courses gave me overall scores of 4.4/5 and 4.3/5, respectively, compared with a mean of 3.9 in both courses for all TAs during the period that records were kept. I have also worked as a private tutor. Over the past 18 months, I have gained experience leading a research project, for which I am coordinating the activities of an undergraduate (who I advise), a first-year graduate student, and a recent graduate who is a postdoc at another institution. My roles outside academia have also helped me learn how to lead a group: for two years at Berkeley, I organized a debate club that presented several events

¹See Kohler, Handley, and Floyd, “Designing DCCP: Congestion Control Without Reliability”, SIGCOMM 2006.

each semester; I am co-leader of an actively gigging seven-piece brass band; and I have previously led several jazz combos.

I am prepared to teach core graduate or undergraduate courses in Networking, Distributed Systems, Algorithms, and Operating Systems, as well as any introductory computer science course. I also plan to develop courses or seminars covering such current research areas as dealing with failures in networked systems, overlay and peer-to-peer networks, and Internet routing. The latter would take a broad view of the topic, including a traditional treatment of Internet routing protocols but ranging from theoretical aspects (such as recent results concerning the complexity of BGP convergence) to operational aspects (such as how ISPs do traffic engineering in the real world). This would set the stage for readings on new Internet architecture proposals.

Given my experiences at Berkeley, I believe I am well prepared to teach, advise students, and lead a research group—and I am certainly excited to get started!