NEIL SPRING
Teaching Statement

I have taken advantage of many opportunities to teach and supervise graduate and undergraduate students. I have taught the undergraduate computer networks course twice. I have also been the teaching assistant for graduate computer networks and transaction processing courses, tutored undergraduate students, informally supervised several graduate research projects, and led the development of the "Fishnet" network emulator for undergraduate programming assignments. Even when not formally teaching, I am involved in undergraduate and graduate networking courses because students often build upon my software. In the future, I hope to lead classes not just in networking, but in all aspects of systems, from operating systems to databases and programming languages. I enjoy helping students make connections from networking to these related areas, and look forward to teaching these subjects. Below, I describe some of my successes and mistakes that shape how I will teach classes and advise students.

At first, I volunteered to team-teach the undergraduate networks course because I knew from my experience as a teaching assistant how much I enjoyed working with students and presenting material. I wanted to know that an academic career was right for me so that I could stay motivated through the hard times older graduate students warned me about. I was also happy that teaching the course allowed me to contribute to the department. Although I was enthusiastic, I was also unprepared: it was only my second year of graduate school and I had not even taken an undergraduate computer networks class. Surprisingly, my lack of experience made teaching the class more fun and personal — I was able to decide what to present in terms of which high-level concepts students would find valuable and which details should be skipped.

In designing our course, we made choices that matched our expectations for the best possible networks class, but our decisions produced a course that was too difficult for some. We chose a dense but complete text, Keshav's *An Engineering Approach to Computer Networking*, that we imagined students could use as a reference after the course. We gave programming assignments that challenged students to push their knowledge of data structures (a Patricia trie for longest prefix matching), algorithms (Bellman-Ford routing), and systems programming (sliding window transport). Although most students had already taken data structures, and some had even taken operating systems, these were not prerequisites, so many students had a difficult time. As a dedicated teacher with only the responsibilities of a graduate student, I enjoyed helping less prepared students with extensive one-on-one interaction. However, I knew that this approach would be too time consuming for a faculty member and I wanted more students to experience the satisfaction of completing programming assignments on their own, so I adapted.

I volunteered to teach the networks course a second time because I wanted to teach it on my own and to do a better job. My goal was to allow students to demonstrate their mastery of course material in different ways, with less emphasis on programming. In particular, I developed new assignments that focused on analysis instead of implementation. A common, difficult programming assignment in a networks class is to implement a reliable, sliding window transport protocol. I drew upon my research experience with network measurement to construct a new assignment in which students analyze the behavior of the real transport protocol used in the Internet, TCP. Students wrote some code to analyze a packet trace to estimate the state variables, such as the congestion window size and round trip time, at both ends of the connection. Students then graphed how these variables change over time and turned in a written description of what factors limited performance. This approach worked well: the new assignments were a little less challenging in implementation, but the students enjoyed their experience studying a real protocol on a real network and were able to focus on concepts instead of implementation details.

My experience with and enthusiasm for the networks course has kept me involved. On my recommendation, the course was changed to add a data-structures prerequisite and a credit-hour for a discussion section. Because these changes ensured that students were better prepared and had more support from teaching assistants, it became possible to expect students to build an integrated network stack, including a routing protocol, a transport protocol, and an application on top. To support these programming assignments, I led the development of a simple network emulation library named Fishnet. Students have particularly enjoyed the capstone assignment: using their Fishnet-based network protocol stacks to turn 802.11-enabled iPAQ handhelds into walkie-talkies. In this assignment, students were able to implement walkie-talkies that interoperated not just with the solution key but with each other — a challenge unique to networks.

I have mentored several groups of students in our graduate networking course, especially those who have made use of tools from my Scriptroute and Rocketfuel research projects. The project I most enjoyed working with tried to solve a difficult problem in network mapping: to accurately and efficiently determine which IP addresses belong to the same router. My students were creative and developed exciting new techniques that we believed would work well. Initial measurements seemed promising, but as we refined the heuristics further, we found that their accuracy was disappointing. The lesson was obvious in retrospect. In research that I am currently supervising, I focus first on ensuring that the validation strategy is sound, and then allow students to be creative in devising a working technique.

Interacting with students excites me, and I look forward to teaching different subjects and sharing my enthusiasm for teaching and research with students.