TEACHING STATEMENT

Md Ashikul Haque

In building an academic career, research and teaching are complementary to each other. I consider teaching a professional commitment and one of the tasks that I enjoy most. My teaching philosophy is shaped by my teaching experiences, the experiences that I accumulated as a student from my teachers, my work on production software at Samsung, and my own views. Most of all, I want my students to learn to reason about systems, because a student who can explain why a network behaves as it does can learn any protocol or platform later.

1 Teaching Experience

My experiences include teaching at the University of Texas at Dallas (UTD) and at Millennium University in Bangladesh. Since Spring 2026, I have been a teaching assistant at UTD for two graduate courses, *Computer Networks* (CS 5390) and *Performance of Computer Systems and Networks* (CS 6352). I grade assignments, examinations, and projects. I also hold office hours and help students debug their implementations. I have seen that students struggle less with a protocol itself than with the model behind it, such as a queue, a Markov chain, or a timing diagram. I now explain the model first and the protocol as an instance of it. In 2025, I also gave several guest lectures on LoRa networking and the NS-3 network simulator in a graduate Internet of Things course.

Before graduate school, I was a lecturer at Millennium University from 2018 to 2019 and the instructor of record for *Structured Programming* and *Data Communication*. I designed the lectures, the laboratory assignments, and the examinations for both courses. *Structured Programming* was the first programming course for most of its students. Teaching *Data Communication* to new programmers taught me to build every concept from a worked example before the general formula.

2 Mentoring and Service

Since January 2025, I have mentored Nasif Ahmed, a junior doctoral student in our laboratory. Our work together covers identifying a new problem, running experiments on Universal Software Radio Peripheral (USRP) radios and commercial off-the-shelf LoRa devices, and writing a systems paper. I have also collaborated with Aakriti Jain, a fellow doctoral student, on two papers, one led by each of us (IEEE RTSS 2024 and ACM/IEEE SenSys 2026). Since 2022, I have served as a referee, sub-reviewer, or reviewer for 7 conferences and 3 journals, among them SenSys, RTSS, RTAS, and ACM Transactions on Cyber-Physical Systems.

3 Teaching Interest and Course Design

I am open to teaching any fundamental course, such as *Data Structures*, *Algorithms*, and *Computer Organization*. At the undergraduate level, I am interested in *Computer Networks*, *Operating Systems*, *Data Communication*, *Artificial Intelligence*, and programming in C and C++. At the graduate level, I am interested in *Computer Networks*, *Wireless Networks*, *Internet of Things*, *Real-Time and Embedded Systems*, and *Performance Evaluation of Computer Systems*.

I am also planning to design three new courses that grow from my research. The first is a graduate course, *Dependable Wireless Systems*, which covers jamming defenses, spectrum sharing, real-time medium access, and cross-technology communication. Each unit pairs a research paper with a laboratory on software-defined radios. In one laboratory, students jam a cabled LoRa link with a USRP, watch the packet reception rate fall below 1%, and then implement a gateway-side decoder that recovers the packets. The second is an undergraduate laboratory course, *Software-Defined Radio and Wireless Systems*, in which students build a receiver from samples up on USRP

radios with GNU Radio. The third is a graduate seminar, *Real-Time Physical Artificial Intelligence (AI)*, on scheduling and admission control for vehicles and robots that act on large AI models.

In addition to the fundamental and essential materials, my course syllabi will cover the topics that make each course competitive with similar courses offered in highly reputed departments. I will also update course materials based on market demands. For example, in a *Computer Networks* course, I will emphasize network security as a concluding topic, including jamming.

4 Teaching Philosophy

Practicing an Active Learning Classroom. A key part of my teaching philosophy is to practice an active learning classroom, where students take part in learning rather than receive knowledge passively. I will break every class into short cycles of prediction, experiment, and explanation, because students learn more when they predict an outcome and test it than when they only listen. In a unit on contention, students will predict how packet delivery and delay change as more senders share a channel, test it in a small simulator, and explain where their model failed.

Handling Mixed Backgrounds. I have studied in Bangladesh and in the United States, and I have taught students for whom English was a second language. A graduate networking course seats students who have written a kernel driver next to students who have never opened a socket. To address the needs of all of them, I will explain every complex topic in more than one way, such as a worked example and a figure, so that each student can follow at least one of them. I will connect each topic to a real system. For instance, I will teach priority inversion through the Mars Pathfinder lander, whose computer kept resetting itself on Mars because of a priority inversion.

Feedback and Assessing Student Progress. To engage students, I will frequently reassess their learning needs. A short recap of the previous lecture at the beginning of each class helps students get on track, because not all of them have the same level of preparation. I will also seek feedback every 3 or 4 weeks and adjust my teaching style if needed. Each assessment will align with a stated learning objective, and I will publish its rubric before students begin. I will return feedback within one week, and my comments will name the misconception rather than the symptom.

Eliciting and Exercising Student Thought-Process. Critical thinking is an important skill that students can acquire through course activities. Therefore, I will start teaching an algorithm, a network protocol, or any other solution by stating the problem and its constraint, and I will ask the students to propose their own solution first. The proposals are often the designs that earlier work tried, and the discussion of why each one fails prepares the students for the one that works. My exams will test reasoning rather than memorization. For example, I will not ask students to describe a medium access protocol. I will describe a deployment, such as 1,000 sensors that each report once an hour, and ask which protocol fits it and why.

Promoting Collaboration in the Classroom. Promoting collaboration in the classroom is one of my key principles of teaching. I will often form small groups in class for peer discussion, each with a problem to solve. A semester-long project will give students a better chance to acquire collaboration skills by working with classmates who bring different strengths. I will form teams of 3 or 4 students myself and rotate the roles, so that every student writes code, runs an experiment, and presents. To ensure that every member contributes, I will ask each student to assess their own contribution and that of each teammate, and I will count these assessments in the project grades.

In summary, my teaching philosophy is grounded in my view that computer science courses should prepare students for research, for entrepreneurship, or for careers as computer scientists and engineers who can face tomorrow's challenges. I want to equip my students with the skills that can make them knowledgeable and independent thinkers, researchers, and entrepreneurs.