Using Tools and Testbeds for Teaching and Learning Computer Networking During the Pandemic and Beyond

Michael Zink, Department of Electrical and Computer Engineering, University of Massachusetts Amherst

I have been teaching the subject of computer networks to undergraduate and graduate students in the Department of Electrical and Computer Engineering for more than 10 years. During that period, I have also been involved in creating tutorials and labs that allow educators to use GENI and CloudLab testbed infrastructures for educational purposes. I have used these in the classes I teach and collaborated with other instructors who have been using them for their own courses.

In my teaching approaches, I strongly believe in the process of “muscle memory”. What I mean by that is that combining theoretical exercises combined with “hands-on” exercises have a more long-lasting effect. Teaching students the theoretical foundations of computer networks and have them apply this knowledge in lab assignments seems to lead to a much more improved learning experience for the students. Course evaluations have indicated that students liked the courses I taught significantly better if they included such “hands-on” labs.

Based on my past experiences and the use of labs that are independent of local hardware, I was able to switch my graduate course from in-person to online without significant disruption when all in-person classes at the UMass Amherst campus were halted. All labs and the final project assignment had already been designed such that they did not require on campus presence but could be performed from any location that provides reasonable Internet access.

In hindsight, the experience over the past couple of months have shown me that certain course components can be designed independent of the form the course is taught (in-person, online, or blend). I believe that we should take this lesson into account for future courses, which will give us significant flexibility in the form in which we teach them. Therefore, the following observations and statements focus more on certain components (mainly assignments) and not the actual course lectures.

Obviously, there is a large set of tools that can be used to allow students hands-on experience in computer networks. I believe that all of these tools have their place in the computer networks curriculum depending on the objectives of a specific course. I share my experiences with some of these tools in the following:

- **Simulations** have shown to be an important tool for computer networks research, and it is important to educate graduate students to allow them to perform meaningful and appropriate simulations on their own. For the undergrad level, I believe it is not essential to teach students how to perform them. There is a steep learning curve and being able to perform simulations is not a highly sought-after skill for B.S. entry-level positions in industry.
• **Wireshark** is a great tool for undergrads to learn more about the inner workings of Internet protocols. It allows the instructor to create a whole set of assignments from just observing network traffic to much more sophisticated analyses of protocol behavior. The downside is that students can only measure the traffic where they “tap” into the network. I would be interested in discussing the idea of a repository of wireshark traces (wireless, from a core router, etc.) that could be shared with the community and used by students to analyze more complex traffic scenarios in addition to the ones that can be captured from a desktop computer or a laptop.

• **Mininet**, is a tool to create realistic networks on a single machine (e.g., a student’s laptop). It has been heavily used for education on the topic of computer networks. Mininet has the benefit that experiments can be executed locally without the need for additional hardware. I have observed students being able to quickly learn how to use it for lab assignment and their research. In addition, it allows students to run experiments in the mininet environment (potentially clearing bugs and errors) and port them to testbeds like GENI and CloudLab.

• Testbeds (GENI, CloudLab, Chameleon, ORBIT, etc.) give students the opportunity to perform labs on actual (although in some cases virtualized) hardware. There is a much steeper learning curve to use testbeds for labs but tutorials that have been made publicly available can be used to teach students the basics they need to know to use these testbeds. I have observed that using such testbeds can be an “eye opener” for students, since they run into challenges they do not face when they use simulations or mininet. While the major use of these testbeds is most like by graduate students, I have made the experience that undergrads can also benefit from labs performed on testbeds, if they conduct well-guided labs on these testbeds.

• On-site labs give students the opportunity to “touch and feel” networking equipment. Such an approach is definitely in favor of students who benefit from a more “tangible” learning approach and will teach them practical skills for professional positions (e.g., network administrator). While there clearly are benefits of such on-site labs, they require a significant amount of local administration and care. In addition, the number of students that can use such labs are very limited. Most smaller colleges or institutions might not have the human and financial resources that are required to acquire and maintain the infrastructure for such a lab.

In the past, I have been using some of these tools for both a 3rd year undergraduate course (ECE 374) and a graduate course (ECE 671). For both courses, my RAs and I have created a set of labs that (http://emmy10.casa.umass.edu/) can be used for both classes. In the following, I am presenting some examples of how I have been using these labs for ECE 374 and ECE 671. For both classes I am using “Computer Networking A Top-Down Approach by Jim Kurose and Keith Ross as the textbook. The first lab, called “Three Node Routing” is designed to have the students gain their own experiences with routing and it accompanies Chapter 4 of the book. It

---

1 There is already a large repository of existing wireshark trace (https://wiki.wireshark.org/SampleCaptures) but it is quite unorganized and additional information must be added to make them usable by students.
is a very simple assignment that allows students to implement basic routing in an actual testbed. The second lab, called “OpenFlow Learning Switch” is designed to give the students the opportunity to learn how to implement an Ethernet Learning Switch. In this lab, the students use Open vSwitch\(^2\) controlled by OpenFlow to implement the Learning Switch. This lab accompanies Chapter 5 of the book. An example for a lab that I have been using for the graduate course is the “Network Address Translation” (NAT) lab. In this particular instance the lab is divided into two parts. In the first part, the students are tasked to implement a NAT router in mininet. In the second step, the task is to implement the NAT router in an actual testbed (in this case GENI or CloudLab\(^3\)). We have decided to take this two-step approach since it allows students to initially implement the NAT router in a simpler environment (mininet) and once it is working correctly, the students can transfer it to an actual testbed where setting up the actual slice to carry out the lab has its own challenges. Experiences in class have shown that this two-step approach is more manageable for the students. This lab accompanies the material on NAT that is covered in Chapter 4 of the book.

As with any lab, no matter on which of the tools mentioned above it is based on, grading will most likely require more effort. We have been experiencing this first-hand and tried to provide approaches that help ease the pain for the graders. As we all know from experience, assignments that are correct are always much easier to grade\(^4\). This is especially true in the case of labs. For example, in the case of the NAT lab mentioned above, the grader can test the correctness of the implementation with tools like ping, iperf, or tcpdump. Grading becomes more complicated when the lab is not correctly implemented and the grader has to evaluate if partial credit should be given. In the following, we describe a couple steps we have taken to reduce as much load on the grader as possible. First, we try to provide very precise instructions (some of them as short video clips) to avoid any misunderstanding from the students' side. Second, where applicable, we provide templates where certain parts (e.g., an OF controller or an Rspec to describe a slice topology) are already given and only certain parts or functions have to be implemented by the student. While we have not implemented this idea so far, there might be labs (or portions of a lab like an OF controller implementation), where the grading could potentially be automated by using tools like Gradescope\(^5\). We have also experimented with some automated testing. For example, using predefined pings to test if the slice topology has been set up correctly. This functionality is currently not implemented in our labs.

In summary, the computer networks community has a set of tools and testbeds that can be readily used for teaching purposes. Luckily, instructors can combine these tools to the needs of their curriculum and the needs of their students. In addition, many of these tools are accompanied by material (e.g., tutorials) that aid students in teaching them how to use these

---

2 \(\text{https://www.openvswitch.org/}\)
3 All labs are designed such that they can be performed on either GENI or CloudLab.
4 The benefit of GENI and CloudLab is that everyone who is part of a project can log into the nodes that belong to a certain slice, independent of who initiated the slice. Thus, a grader who is part of the project can access the slice to examine if the lab was implemented correctly.
5 \(\text{https://www.gradescope.com/}\)
tools. My long-term experiences in teaching computer networks to undergraduate and graduate students have shown that these tools can be use independent from the modality of the class (in-person, online, or mixed). I also believe that the four tools on the top of the list above should be heavily supported by the community (and potentially funding agencies), since they contribute significantly to the democratization of computer networks education. These tools can be used by any instructor worldwide and ownership of actual networking equipment is not required.

I would like to finish by clearly stating that I see significant benefits in in-person teaching. Interestingly, I believe the benefit does not come from in-person lectures, discussions, or labs but from the one-to-one interactions at office hours, and before and after class. These conversations have often helped me tremendously in understanding where students have challenges and how they could benefit from me adjusting my lectures to their needs.