“Antidote” for CoVID-19 pandemic induced surge in online learning

Parimal Parag†‡ Preetam Patil† Sharad Shriram† Rajesh Sundaresan†‡§ Himanshu Tyagi†‡§ Nidhin Koshy Vaidhiyan†
†Centre for Networked Intelligence  ‡Department of Electrical Communication Engineering  §Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science
Bengaluru
{parimal,preetampatil,shardrsriram,rajeshs,htyagi,knidhin}@iisc.ac.in

ABSTRACT
The CoVID-19 pandemic has resulted in a paradigm shift in online teaching. Course instructors offering courses in the Spring 2020 term had to improvise and adopt new methodologies and available technology to teach online. Going forward, institutes and universities can consider online learning to make their courses accessible to the student community. In this paper, we explore different software and platform options to develop an online, hands-on based networking lab course. Specifically, we share our experience of developing a networking lab course curriculum on “Antidote”. “Antidote” is an online course development platform that enables course instructors to offer lesson content along with a web-based playground for hands-on exercises, that can help students retain learnt concepts. Traditional teaching tools for networking like ns-3 and Mininet are not supported on web-based teaching platforms (like EdX). “Antidote” provides a means to integrate these tools, in general, any software through a web-based playground environment.

KEYWORDS
online, learning, networking, education, hands-on courses, MOOC

ACM Reference Format:

1 INTRODUCTION
The outbreak of the COVID-19 pandemic has challenged course instructors in adopting new pedagogical approaches to offer their courses online. While offering a course online, the accessibility for students to access the course content, the resource, infrastructure, and the pedagogical style adopted for the course are some factors to consider. In addition to these factors, course instructors need to consider the platform used for offering the online course. The online platform would also impact the teaching tools that would be available for the course. For most networking courses, the teaching tools are desktop applications or software that run on either lab computers or student’s personal computers. While offering a networking course online, there are limitations on how the traditional teaching tools like Mininet is made available in an online environment. In this paper, we explore different software and platforms which offer both the course content and an environment that provides a playground for students to try networking programming exercises online.

“Antidote” is an online platform that provides both lesson content as well as a playground environment for hands-on exercises within a single browser window. The hands-on exercise environment offered by the platform are containers that run as lightweight, virtual resources providing a sandbox for students to complete programming exercises. The platform supports lesson content or lab guides in the form of either a Jupyter notebook or as a markdown file. The markdown file can contain multimedia content like videos, images, and slides besides plain-text and code snippets. “Antidote” makes networking lab courses more accessible and helps instructors offer a playground-based learning experience for students. “Antidote” can be set-up on cloud providers like Google Cloud Platform, AWS and thus, can scale based on the number of users. A kubernetes engine running on the back-end handles the scalability of the platform as well as orchestrates the different container-based resources of the platform.

In section 2, we discuss the Model-View-Controller (MVC) architecture [7] used on online course platforms and their limitations for offering a practical networking lab exercise. In section 3, we discuss the traditional teaching tools used in networking courses like Mininet, ns-3, and highlight their limitations when networking courses are online. In the remaining sections of the paper, we discuss about “Antidote”, and our experience in developing a curriculum for an online networking lab course using “Antidote”.

1.1 Selecting the suitable platform
With a plethora of online platforms available for creating an online course, it becomes a challenge to pick a suitable platform to offer an online networking lab course. For developing our lab course,
we select different platforms and software based on the following criteria:

- The platform is open-source, and is freely available
- The platform can be customized based on the requirements per course
- The platform is able to provide a hands-on learning experience
- The platform can be either used to offer online courses or be used in a hybrid online-and-classroom interaction based courses

Different software and platform options can be used as teaching tools for networking courses. The criterion for the software or platform to be open-source is to make collaborative enhancements and developments on the code. Thus, we identify software and platforms that are open-source, for example instead of representing MOOC providers like edX, Coursera, Udacity, and edX\[7\]. We also consider open-source software like Mininet\[6\] and ns-3 that are used as teaching tools over proprietary software like exata and OPNET. If the course is offered for smaller group of students belonging to an institute, a test-bed based environment can be used as a teaching resource and thus we consider open-source platforms like Emulab \[14\].

### 2 THE MODEL-VIEW-CONTROLLER ARCHITECTURE

In this section, we discuss the strengths of the Model-View-Controller architecture in designing online teaching platforms. We do not consider whether the course is offered as a MOOC or a SPOC. The MVC architecture is a software design pattern that has been widely used in developing online platforms and is also used by popular MOOC providers like Coursera, Udacity, and edX\[7\]. The MVC architecture has three components and each component has specific functionalities. In the context of MOOC providers, the model is responsible for meeting the data needs of the MOOC platform. The model is implemented using database technologies like MySQL or MongoDB\[7\] to store platform assets like curriculum, resources for lessons, instruction materials like videos or slides, along with user information like mapping enrolled courses, recommending related courses and tracking the modules of the courses that were completed by the users. The View component handles user-interaction and provides interfaces of the Model to users (students), course instructors and platform developers. The pedagogical style - video, text or code, hands-on exercise environments, quizzes and forums are some of the different views offered in MOOC platforms. The Controller is responsible for handling user inputs from views and fetching the required or related content from the Model. In some implementations, there is an additional sub-component termed as a “service” inside the Controller that interacts with the Model. Figure 1 describes an overview of the MVC architecture where there is an AJAX controller implemented as a service to interact with the Model.

If the MVC architecture is considered to be used in offering online courses then keeping open-source solutions in mind then open EdX\[1\] can be considered since it is available as a free and open-source

---

**Table 1: Summary of online platforms analyzed in this paper**

<table>
<thead>
<tr>
<th>Platform</th>
<th>openEdX</th>
<th>Mininet</th>
<th>Emulab</th>
<th>Antidote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the platform open-source?</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Does the platform run on the cloud?</td>
<td>Y</td>
<td>Containers or VMs</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Can the platform be used for developing courses for online courses?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Can the platform support collaborated curriculum development?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Does the platform support video lectures?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Can the platform be used to offer online courses</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Does the platform offer an online lab environment?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Can the platform support customized course delivery?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Does the platform support submission and evaluation of assignments?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Can the platform be used a teaching tool for a networking course, both online-based as well as classroom-based?</td>
<td>Y</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
</tr>
<tr>
<td>Does the platform offer support make teaching tools for networking available online?</td>
<td>NA</td>
<td>NA</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

---

1\[https://github.com/edx/edx-platform\]
code. The open EdX platform has many features offered in the EdX platform and can be easy to configure and customize since the codebase is mainly Python. More specifically, it is written in Django - a Python package for developing full-stack web applications.

Most MVC-based platforms support offering reading-based exercises, video-based lectures, and multiple-choice questions-based or code snippet evaluation-based assessment. The support for practical exercises offered on MVC-based platforms is limited to programming exercises that are offered as high-level abstractions of a virtual instance of a resource running on the cloud. For a networking exercise, the commands used require more access to the virtual instance specifically to the virtual os-kernel to run networking-specific commands. To provide more access to the virtual resources, the programming environments offered in MVC-based platforms need to be redesigned which makes MVC-based platforms not suitable for providing online practical exercises. The solution that course instructors offering networking-based online courses use to mitigate the limitations of the online platform is to let students set-up the traditional teaching tools used in networking courses like Mininet on their personal computers or laptops. The students are supported through online forums available on the learning platform to guide and assist them [2]. In the next section, we discuss a few software-based solutions that are currently used as teaching tools for practical networking exercises, followed by a section on “Antidote”, the platform that we chose to develop our online course on networking.

3 SOFTWARE

In this section, we describe about three software or paradigms that are widely used as teaching tools in networking courses and highlight their strengths and limitations as networking courses become online.

3.1 Mininet

Mininet [6] is a prototyping tool for emulating large networks with the resources available on a personal computer, laptop and is largely used for studying networks as well as software-defined networks (SDNs). The ease of set-up, the availability of resources in the form of tutorials, and good documentation of Mininet has encouraged many professors in universities across the world to adopt it as a teaching tool for networking courses [2, 3]. There are different ways through which Mininet can be invoked in addition to the native command-line interaction, there is a desktop GUI application called MiniEdit [5] shown in Figure 2 to visually design and emulate different network topologies.

Under the hood, Mininet uses the Python programming language to create an environment to interact with virtual networking components like controllers and switches using standard linux commands. Thus, it is relatively easy to build applications like MiniEdit or develop wrappers over the existing Mininet framework. However, Mininet is designed to work off-the-shelf in laptops and computer and is not designed to be accessible through a web interface. One approach to make Mininet accessible as a teaching tool for an online course would be to create a container using platforms like Docker and be spawned as an instance running on the cloud [5]. This approach requires time for setting up the infrastructure to host and make the containers accessible to students. Despite being a desktop-based framework, the success of Mininet to be used as a teaching tool in classrooms and MOOCs [2] can be attributed to its versatility in emulating different network behavior and conditions with a relatively easier learning curve.

3.2 ns-3

ns-3 is a network simulation software that is widely-used in networking research and education. The software is a discrete-event simulator and has been widely used to offer a hands-on experience on modifying existing protocols or testing novel protocols and algorithms for the different layers of the OSI stack. ns-3 and its predecessor ns-2 have been used as teaching tools in both courses on computer networking [10, 13] as well as wireless networking. The difference between Mininet and ns-3, apart from one being an emulator and the latter being a simulator, is that ns-3 had comparable or faster run-times and lower memory utilization in implementing simulation on modifying existing protocols or testing novel protocols and algorithms for the different layers of the OSI stack. ns-3 and its predecessor ns-2 have been used as teaching tools in both courses on computer networking [10, 13] as well as wireless networking. The difference between Mininet and ns-3, apart from one being an emulator and the latter being a simulator, is that ns-3 had comparable or faster run-times and lower memory utilization in implementing ns-3 algorithms for the different layers of the OSI stack.

In terms of the learning curve, Mininet is a better choice especially if students enrolling in the course have not done a prior course in C++. In the case of ns-3, there is more steep learning curve to make use of the different features which might be a deterrent in serving the learning objectives associated with a networking lab course. Like Mininet, ns-3 is also a desktop software and will need additional development to make the software accessible, for using in an online networking course.

3.3 From test-beds to Container-Nets

For online hands-on lab courses offered as a SPOC (Small Private Online Courses) experiments can be emulated on virtual networks created on dedicated networking test-beds like PlanetLab [8], and Emulab [14]. Both PlanetLab and Emulab use Linux vservers

![Figure 2: Screenshot of using the MiniEdit interface to configure a network-switch graphically in the emulation of a fat-tree topology](https://github.com/iwaseyusuke/docker-Mininet)
[11] or other related software to provide resources in terms of CPU and network isolation to the users. Emulab is well-documented and the source code to the Emulab test-bed is open-sourced. However, a significant amount of time and infrastructure is required to set-up a test-bed based online learning course. Recently [5] introduced container-support to the Emulab test-bed which indicates a shift towards container-based networking. This indicates that there is growing interest in adopting container-based networking in network emulation, signalling a shift towards software-based networking experiments in the coming years.

3.3.1 What is container-based networking (or) container-nets? Typically large-scale network topology emulations are done by creating one (or) multiple virtual machines whose network interfaces are connected to create the desired topology. In container-based networking, the emulation of network topologies is done by using containers. These are light-weight alternatives that distribute compute, network and memory resources among multiple virtual machines. In terms of performance, container-based networks have comparable throughput and performance when compared with physical networks[12].

3.3.2 Why is container-based networking becoming popular? Currently networking test-beds spawn virtual machines to create virtual networking topologies on-demand. The on-demand resource allocation has limitations, for instance the number of users who can access a particular resource is dependent on the availability of infrastructure. This introduces time-delays in allocating resources and makes the test-bed less accessible for other users. Another limitation for the test-beds to scale is the cost to add additional resources for every x number of new users. Container-based networking (or) container-nets is a light-weight alternative for virtual machines: the performance of container-nets is comparable or better than virtual machines[12]. Container-nets also allow rapid prototyping and testing of networking protocols or software since the time to set-up and develop a container-net is considerably lower than the time to set-up a virtual machine based topology.

3.3.3 Can container-nets be used as teaching tools? Since 2016, there have been several papers that have used container-based networking to study different networking research problems. For instance [9] deployed containers as networking components in their tool called "Containernet-2.0" which allows users to rapidly prototype their study using containers for emulating virtualized network functions. On a related note, [1] developed a platform that uses containers to allow users emulate virtualized network functions. The platform also supports studies where software-defined networks (SDN) coexists in a topology with traditional networking topologies. Experiments to test network function virtualizations or to study routing protocols on hybrid networking topologies with SDNs and traditional networking components is cumbersome and is resource-intensive. Given that container-nets perform similar to physical networking devices, this makes container-nets ready to be adopted as a teaching tool [12]. However, current solutions like 'Containernet-2.0'[9] are also desktop based applications which means, they need to be repackaged to make them available as a teaching tool for online networking courses.

3.4 Summary

Traditional teaching tools for networking courses namely: Mininet and ns-3, are desktop-based software and would need additional development to make them accessible for online courses. Currently online courses on networking asks students to set-up these software on their computers and provide support through forums available online on the platform [2]. There are networking courses that use test-beds set-up within the campus, but while offering online courses providing access to such test-beds poses challenges for security and scalability. The challenges faced by test-beds can be addressed through container-based networking where containers are used as networking components and are made accessible in isolated virtual environment on the computer [12]. However, container-based networking tools like 'Containernet-2.0'[9] currently run as desktop applications and will need to be integrated as a teaching tool for online networking courses. "Antidote" is an course development platform that supports web-based network experiments through the use of containers and container-based networking.

4 ANTIDOTE

Antidote is an online platform that was initially developed to expose the learner to the different tools used in network automation and reliability engineering. The platform is based on a hands-on learning pedagogical philosophy that provides the instructions, i.e., lessons, as well as a browser-based playground to exercise the learnings obtained from the lesson module. The platform relies on containers to provide the resources that would create the environment for the learner to try the hands-on exercises. The instructions for each exercise can be specified either as a Markdown-formatted text or a Jupyter notebook. In this section, we study the design of the platform and explore its use in offering an online, hands-on networking course.

4.1 Design and Deployment

The antidote platform is developed with a component-based architecture with different third-party libraries and packages integrated into a complete online learning platform. In figure 3, we present the different components that will come into play when a user is learning on a lesson on the Antidote platform. The components that are numerically labelled in figure 3 are the fundamental components required for the platform to function. The first component in figure 3 would be the web browser session which the user launches to work on a hands-on exercise. The website has two components namely the lesson instructions on the left-hand side and multiple terminal instance on the right-hand side of the screen. We will discuss the setting up of the lessons instruction and the hands-on environment in detail in section 4.1.1. The lessons are offered through an Apache Tomcat server. The terminal instances for the hands-on exercises are running container-instances on the back-end which are accessed on the browser through a Web-based SSH connection. The lesson instructions can also be in the form of a Jupyter notebook which will be
Figure 3: An user-journey perspective on Antidote’s architecture

Figure 4: Directory structure of an antidote curriculum with the lessons modules

represented in component 3 of figure 3. There are two components associated with creating a new lesson: (i) the lesson curriculum where the course content and the lesson environments are specified and (ii) the docker containers required for the hands-on environment associated with that lesson. A typical antidote lesson has the directory structure as shown in figure 4 where Curriculum is the root directory for all lessons that would be offered through the antidote platform. Each lesson that is offered is a separate directory with multiple modules called stages. Each lesson has a file named lesson.meta.yaml where the meta information about the course, such as the resources namely the required containers, the list of modules that are offered in the course, and also whether the lesson content for the module is a Jupyter notebook or markdown-file, are specified. When the course is selected by a user, the antidote-core spawns the required containers and fetches the lesson modules based on the specifications of the lesson.meta.yaml file. In most cases, the docker containers required are fetched from websites, like docker hub when the lesson starts for the first time, and the lesson content is pulled from a git-repository where the curriculum is maintained. Since, the curriculum is maintained on a git repository like a code-base, the curriculum design process is referred as curriculum-as-code. The lesson.meta.yaml file also allows containers to be connected with each other, which uses the notion of container-based networking discussed in section 3.3.

5 ANTIDOTE AS AN ACADEMIC TEACHING PLATFORM

In our discussion of the Antidote platform, we saw that the platform can be enabled to offer not only lesson instruction but also hands-on exercises for students to get a better understanding of the lesson. In this section, we share our learning on creating a new lesson and on the modifications required to adapt the antidote platform to offer the lessons through two studies on developing lesson modules for our lab course. Through our experiences, we consolidate the strengths, and limitations of the antidote platform when used as an academic teaching tool for offering online courses.
5.1 Creating new lessons

In the discussion below, we assume that the materials and references required for creating the course are already available and focus only on evaluating how the antidote platform can be used for teaching the lab component online. We consider two cases where in the first case, antidote is under-utilized and not a suitable solution and in the second case, antidote is an effective teaching tool but requires some customizations to the platform.

5.1.1 Study I: Lesson on graph representations. The first part of our online networking course gives an introduction to networking. The first hands-on exercise gives students the experience of representing network topologies as graphs. The learning objective of this module is to make students understand why network topologies are modelled as graphs and to introduce the shortest-path algorithm as the engine driving network routing algorithms. To evaluate the use of “Antidote” for this lesson, we create three variants of the lesson: one as a Jupyter notebook on Google Collab and two variants on the antidote platform using both the Jupyter notebook and markdown-file based instruction specification.

For the markdown-file variant of the lesson, we created a Ubuntu-based container where the Python packages used in the lesson are installed. When the lesson is offered on the platform, the container is accessible on the browser through a command-line interface and students can either follow the lesson by typing in the code snippets or could run the code snippets through a button click on the lesson itself. For the Jupyter-notebook variant, we installed the required packages and created a Jupyter-based container. The exercises offered in this lesson on all the variations required students to become familiar creating graph representations with code, understanding the different graph representations and implementing the Dijkstra’s shortest path algorithm. In this lesson, students build a directed or undirected graph, and see how the graph is represented as an adjacency list or matrix. The students then learn about graph algorithms, in this case the shortest path algorithm where they are guided through each step of the algorithm with descriptions and are asked to run a code-snippet to get a better understanding of the algorithm step.

For this introductory lesson, we find that Google Collab is sufficient since it allows us to install the required python packages at run-time and the aspects of scalability and reliability are completely dependent on Google’s infrastructure. On the antidote platform, the lessons take a longer time to load as the platform also starts the containers required for the entire lesson, which can be avoided at least on this lesson where the containers are not required unless and otherwise we use the markdown-file based lesson instructions. While creating this lesson, we had to create a docker image which had all the required python packages on a base ubuntu image. When the Jupyter notebook was used as the instruction file, we had to rebuild the antidote-core since the default Jupyter image on the platform does not have the required packages. Thus, for this lesson module, Google Collab was a better choice over antidote in terms of installing packages, scalability and time to develop and deploy the lessons.

5.1.2 Study II: Lesson on Mininet. In this study, we set-up the antidote platform to give students some hands-on lessons on the different tools and software available for not only emulating but also evaluating a computer communication network. For our study, we consider our lesson module on Mininet where the learning objective was to make students learn to write Mininet scripts, so as to emulate a given topology and study the underlying network link characteristics. For this experiment we created a docker image for Mininet with additional packages and customizations for the image in order to make it easy to send commands to the container through antidote-web. The Mininet container also requires elevated privileges to create network interfaces, thus we rebuild the antidote-core where our Mininet container is added as a privileged container.

Online courses like [2] ask students to set-up Mininet on their personal computers and follow the course to complete exercises. Setting up Mininet is easy but it requires every learner to have access to a computer that has much more than just browser capability. Antidote lessons on the other hand makes students learn Mininet from just within their browsers. However, when a student loads the Mininet lesson, antidote takes time to load the Mininet container. This is to create the hands-on playground. This long delay can make the platform appear to be slow which can negatively affect the learning outcome for the course. Despite this, Antidote is still preferred since hands-on lessons on advanced networking topics can be made more accessible.

5.2 Strengths

The antidote platform gives course instructors many creative options to make lesson content, by supporting media types like Jupyter notebooks, plain text, video, image or a combination of multiple media types. Course instructors are also given the choice to incorporate different learning environments using custom containers. The different choices in deploying the antidote platform enables not only accessibility but also scalability. Using containers for creating different learning playgrounds the platform can be considered as a viable choice to make online courses more accessible to a wider and diverse student community as the students would need to be connected to the platform only through a web browser. The different components of the platform including the development of course material can be done remotely, and collaboratively, thereby making it a viable option in future collaborative curriculum development efforts and crowdsourced curriculum development efforts. The platform offers a new pedagogical approach for computer networking course, especially labs making them more fun and hands-on.

5.3 Limitations

The antidote platform is an open-source project which is currently being actively developed and enhanced. Some changes may not be backward compatible, and may necessitate changes to already designed lessons. While the platform can support different hands-on learning environments, it requires some development to build and test new, custom containers for the lessons. This can be challenging for someone starting anew on containers. There is a learning curve to work on the platform as instructors need to familiarize themselves on the directory structure and the steps for defining a new lesson by specifying the correct parameters in the lesson.meta.yaml
file. Sometimes, the antidote-core would need to be recompiled, for example, when the default Jupyter notebook does not have the required packages installed or when a container, for instance the Mininet container for our lesson, has to run with higher-privileges. This might require more than ordinary coding skill. The platform at present does not support user sign-ins, and persistence of state to let users resume courses from where they left off. From a user’s perspective, the loading time for each lesson makes the platform appear slow and unresponsive, which can negatively impact their learning. As an academic teaching tool, the antidote platform lacks features like resuming lessons, submission and grading assignments.

6 DISCUSSION

In this paper, we discussed different tools and platforms that can be used to offer an online networking lab course. An overview of the platforms studied is summarized in Table 1. From our study of the different platforms, we conclude that MOOC platforms like Open EdX would require a redesign to support hands-on exercises for networking courses. Softwares like Mininet, ns-3 which have been widely used as teaching tools in networking courses around the world, are more suitable for classroom-based courses. They need to be adapted to make them accessible to a wider online student community. In our study, we observe a recent shift towards container-based networking and explore if container-based networking platforms can be used for teaching online, hands-on networking courses which led us to exploring "Antidote".

Antidote is an open-source platform that can be used to create hands-on courses with instructions or course content. The antidote platform provides a hands-on playground where students can get a hands-on experience from within their browser, making lessons more accessible. From our experience in developing a hands-on, online networking course, we believe that antidote can be a viable platform for offering online courses on networking. The platform which is still under active development has numerous limitations but these limitations can be addressed through collaborative development by the community. In terms of curriculum development, the curriculum-as-code approach used in antidote can make it easier for collaborative (or) crowd-sourced development of course curriculum. The developers and users of this platform interact and provide support through discussion forums10, and there are video tutorials available on Youtube11. For assistance and support required for developing curriculum on the platform, GitHub issues12 is used. Currently, we have an internal set-up of the platform where we are developing our networking lab course content, which will be made public soon. While our development efforts continue, we are also studying various means of developing a sandbox created within the antidote platform for students to study container-to-container networking. This type of networking is currently not supported by the antidote platform due to security concerns. The take-away, however, is that in these uncertain times, the "antidote" platform can be the means by which better learning can be enabled.

ACKNOWLEDGMENTS

This work was supported by the Centre for Networked Intelligence (a Cisco CSR initiative) of the Indian Institute of Science, Bengaluru.

REFERENCES