

CSC 579 Biweekly Update 2: An Intelligent Transportation System (ITS) Approach to Teletraffic Engineering

Summary: Exploration of SDN to address the gaps in my understanding. Documentation of Mininet installation process (and issues encountered along the way). Encountered issues selecting an appropriate protocol to evaluate. This summary is an in-depth exploration of online resources, rather than academic sources (which were only skimmed).

Overview/lit review

- Recognized my understanding of SDN was limited, and decided to first research further before proceeding with protocol selection.
- I began with basic textbook introduction from Kurose: [5.1 Introduction to the Network-layer Control Plane](#)
- I wanted further information on SDN, so I watched a follow-up video: [Software Defined Networks & OpenFlow - IP Network Layer | Computer Networks Ep. 5.5 | Kurose & Ross](#)
 - I learned from this video that one of the motivating factors of SDN is that surveys of network operators have shown that misconfiguration is likely one of the top reasons for outages across the Internet. So there is a huge demand to solve the underlying issues which make misconfigurations so probable.
- Followed this with a review of *The Road to SDN: An Intellectual History of Programmable Networks* by Feamster et al. (2014) [1]
 - Learned SDN was largely motivated by the OpenFlow [2]
 - It separates the data plane from the control plane in networks.
 - The control plane determines *how* data packets are routed.
 - The data plane carries out this routing, based on the rules defined in the control plane.

- It was spurred on by being seen as having a balance between vision and pragmatism: it was something that could be easily adopted, but also had promising uses for the future.
- The field (of SDN) has been developed based on “use pulls” and “technology pushes,” or rather, spurred by what techniques the technology enables, but limited by what the actual practical uses that excite people.
- Up to the present, a primary (early) use case has been network virtualization.
- My own understanding of SDN *controllers* is that they are for transforming generic SDN switches into other types of network devices. The reason that this control is centralized to the SDN is that if these switches encounter a new type of packet, they first need to communicate with the controller (using a protocol like OpenFlow).

Progress on protocol selection

- Protocol selection has proven extremely difficult. I have yet to reach a resolution.
- The original candidate from my project proposal, *GameTE* [3], has a complexity that I believe extends far beyond the scope of this project.
- I was temporarily interested in seeing whether I could deploy a small part of [a Tailscale network](#) over Mininet. Specifically, I thought I might be able to run my own [DERP \(Designated Encrypted Relay for Packets\) server](#) on Mininet and contrast its performance with [TURN](#), since it is designed as a direct competitor.
 - Unfortunately, both protocols are quite complex and it was unclear to me whether I would be able to successfully deploy them over Mininet with the limited resources that a VM has. Additionally I could find no information on deploying a DERP server in a virtualized environment.
- Although I have not yet explored it, I think *An overview of routing optimization for internet traffic engineering* [4] may offer appropriate options.

Progress on environment setup

- In the past week I explored the installation of Mininet in VirtualBox. This is a challenge because my laptop is an Intel based Mac running macOS Sequoia 15.3.1, and the installation instructions were designed for several versions prior.
- I also had an older version of VirtualBox that was not showing multiple network adapters, so I had to run the `VirtualBox_Uninstall.tool` that comes with a fresh install of VirtualBox.

- The main installation issue I encountered was when trying to add the network adapter required to SSH into Mininet from your host device.
- I resolved this issue through the following steps:

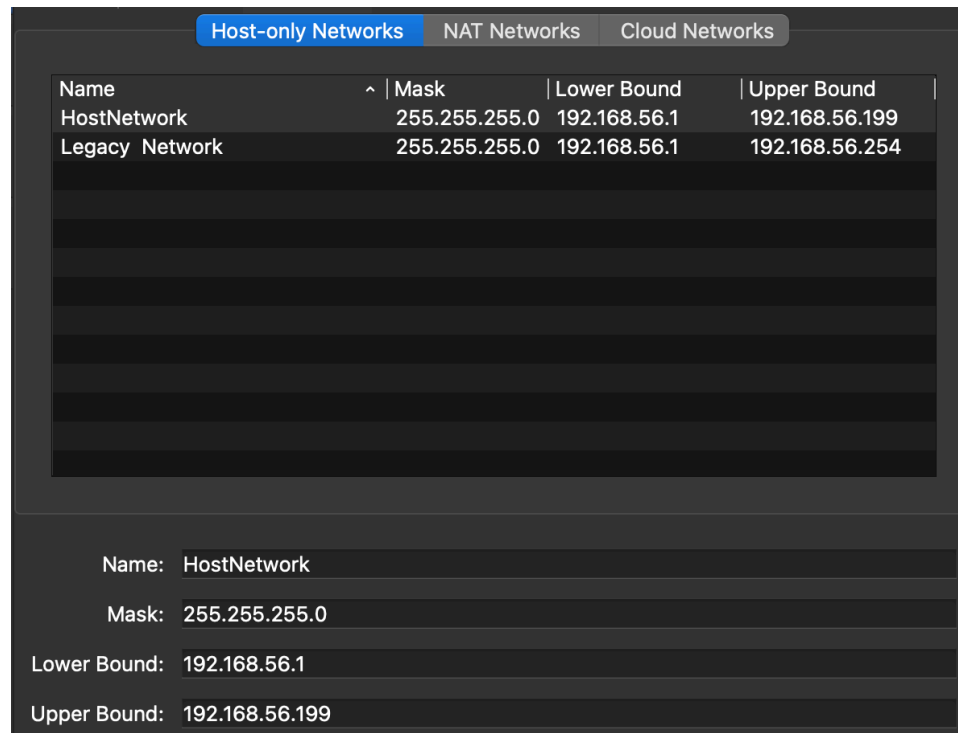


Fig 1. Adding a new Host-only Network in the VirtualBox tools panel.

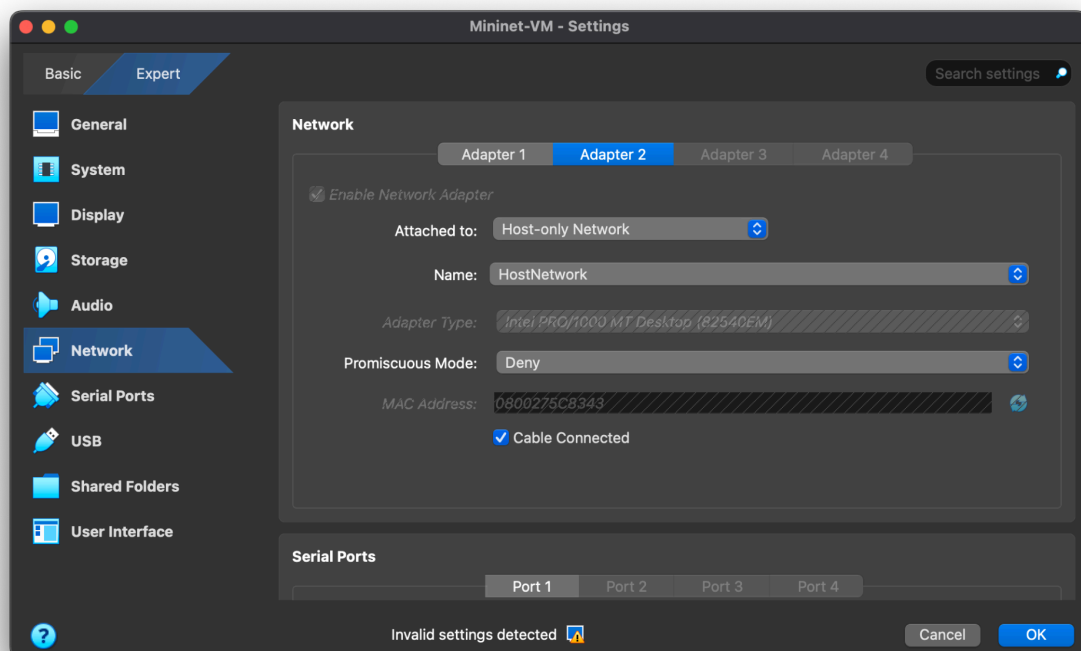


Fig 2. Adding a separate network adapter to access the Mininet VM over SSH.

- I also needed to install XQuartz (which I learned was required from [these instructions](#)).
 - There were some minor installation issues with XQuartz; even after a restart the `$DISPLAY` variable was not available within my VM.
- I later followed Part 1 of the [Mininet Walkthrough](#). This presented the following challenges:
- Wireshark did not start by default. I received the error: “Could not load the Qt platform plugin ‘xcb’” and followed the [corresponding troubleshooting steps](#). In the process of reinstalling Wireshark I discovered that my network adapter for my VM was also misconfigured. By default Adapter 1 was set to an adapter type of Intel PRO/1000 MT Server (82545EM), which meant I was not able to connect to the Internet. Changing this to Intel PRO/1000 MT Desktop (82540EM) resolved my issue.

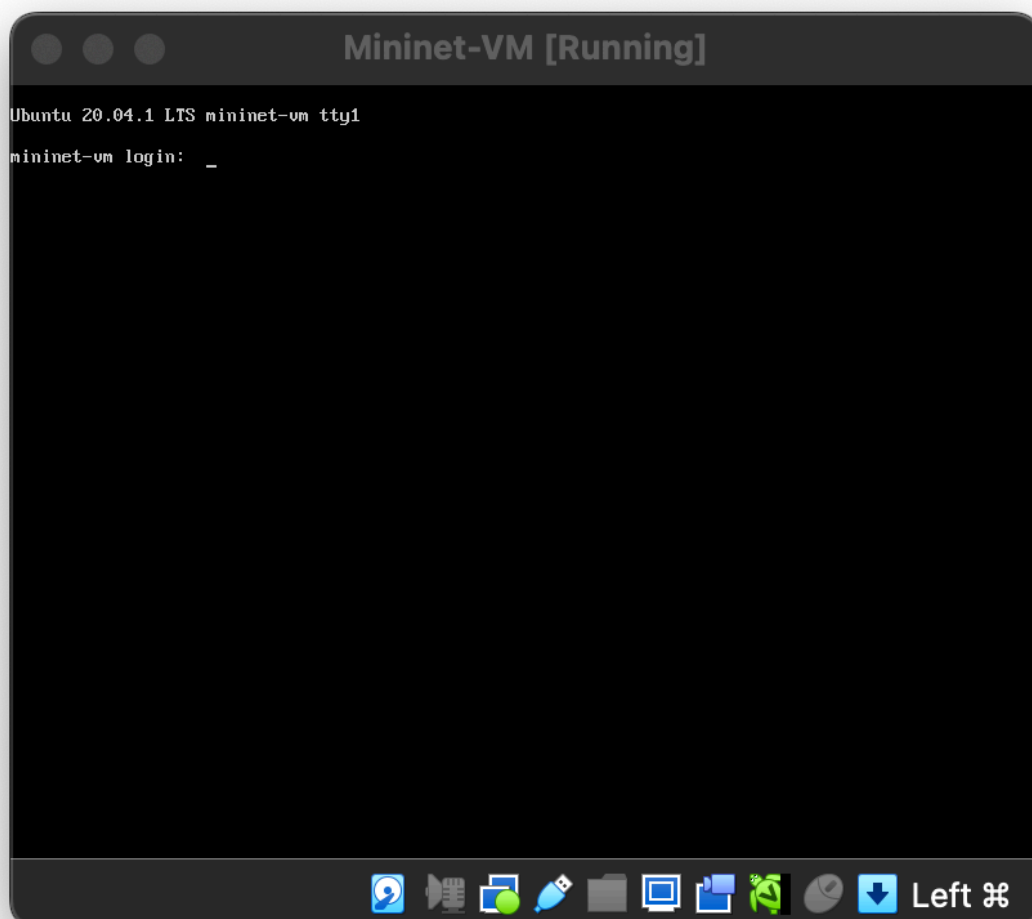


Fig 3. Mininet lives!

- Lastly, I also needed to configure Wireshark to be runnable by regular users:

```
$ sudo dpkg-reconfigure wireshark-common
```

```
$ sudo usermod -a -G wireshark mininet
```

- After installation, I also encountered a useful Medium post by Abdulkader Hajjouz: [Implementing Different Network Topologies Using Mininet](#). I plan to use some of these network topologies if I can figure out how. I have also seen the Internet Topology Zoo [5], which sounds like it will be useful to explore further.

Challenges

- There is a lot more to SDN than I initially expected, and I misunderstood the extent to which SDN is related to the data link layer. This is the area of networks that I am most inexperienced in.
- I am still hunting for how exactly traffic engineering can be applied in my case, because since Mininet is an emulation environment I am worried that it may only be suitable for high fidelity prototypes. Due to limited time and resources, I would prefer to use an already existing prototype, or something that is quick to set up on my own.
- Unclear if I am searching for the correct terminology. Is it really a SDN “protocol” that I want to implement?

Next steps

- Identify how I will use Mininet within my experiments.
- Select protocol for experimentation/evaluation within Mininet.

References

- [1] N. Feamster, J. Rexford, and E. Zegura, “The Road to SDN: An Intellectual History of Programmable Networks,” 2014.
- [2] N. McKeown *et al.*, “OpenFlow: enabling innovation in campus networks,” *SIGCOMM Comput Commun Rev*, vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: 10.1145/1355734.1355746.
- [3] Y. Liu, J. Hua, Y. Zhang, and S. Zhong, “GameTE: A Game-Theoretic Distributed Traffic Engineering in Trustless Multi-Domain SDN,” in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2024, pp. 1248–1259. doi: 10.1109/ICDCS60910.2024.00118.
- [4] N. Wang, K. H. Ho, G. Pavlou, and M. Howarth, “An overview of routing optimization for internet traffic engineering,” *IEEE Commun. Surv. Tutor.*, vol. 10, no. 1, pp. 36–56, 2008, doi: 10.1109/COMST.2008.4483669.
- [5] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The Internet Topology Zoo,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011, doi: 10.1109/JSAC.2011.111002.