Norman Anderson April 7, 2025

CSC 579 Biweekly Update 4: An Intelligent Transportation System (ITS) Approach to Teletraffic Engineering

Summary: Clarified what information I need to learn and what goals I need to set in order to complete my project. Worked on implementation of S and F switches in Mininet with Ryu. Created and recorded final presentation for project.

Guiding questions

1. How can I implement policy enforcement with Ryu?
    - (I think I can do basically what is done in rest_firewall.py, or perhaps use the firewall itself)
2. More specifically, how do I implement the F and S switches from my selected paper?
    - And how do I make this implementation simply “an extension” of default forwarding behaviour?
3. What underlying topology should be used in my experiment?

Goals

1. Implement example security policy with Open vFlow/Ryu
2. Generate SSH and HTTPS traffic on hosts
3. Create basic demo with implemented security policy and generated traffic
4. Design presentation based on (wip) experimental results and previous work

Brief summary

- Explored mixing different types of switches in Mininet
    - Didn’t pan out – I am currently just using one Ryu controller which differentiates switches based on datapath ID (dpid)
- Learned I probably want to select a topology similar to what is described in Wang et al. [1], which is referenced by Liu et al. [2]
- Sketched diagram for simplified topology I will be using for experimentation

- Learned how to set/get dpid in Ryu
- Reread my reference implementation paper [2] (several times) for more clarity
    - Spent some time understanding why the proposed solution is supposedly better than simpler approaches
- Implemented sketched Mininet topology
- Started work on coding S-switch and F-switch behaviour, distinguishing types of switches using dpid
    - Current implementation is not in working state
- Drafted and recorded presentation (did not manage to add experimental results)

Next steps

- Fix S and F switch implementation
- Implement “chain-of-switches” topology approach for policy enforcement
- Run experiment on both topologies to see if I can measure difference in performance metrics (eg. latency) during policy enforcement
- If time: Perform queueing analysis to also compare and contrast both approaches

References

[1] R. Wang, D. Butnariu, and J. Rexford, “OpenFlow-Based Server Load Balancing Gone Wild,” presented at the Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 11), 2011. Accessed: Mar. 31, 2025. [Online]. Available: https://www.usenix.org/conference/hot-ice11/openflow-based-server-load-balancing-gone-wild

[2] J. Liu *et al.*, “Leveraging software-defined networking for security policy enforcement,” *Inf. Sci.*, vol. 327, pp. 288–299, Jan. 2016, doi: 10.1016/j.ins.2015.08.019.