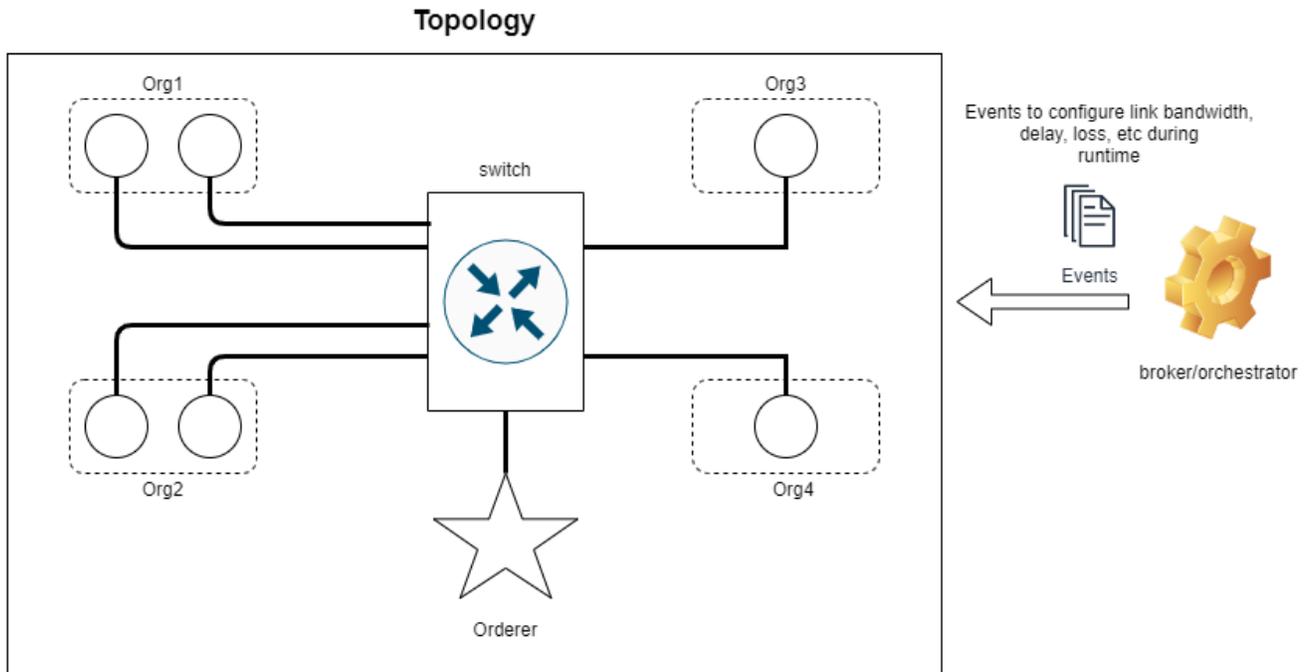


# Project Plan - Adding Network Fuzzing Capabilities to Hyperledger Umbra

## Goal

Hyperledger Umbra (<https://github.com/hyperledger-labs/umbra>) is a project created during the 2019 mentorship period as a network simulation tool to run the blockchain environment. The plan for this year is to add network fuzzing capabilities such as packet drops, delay, loss, etc.



## Proposed Timeline

### Week 1 - 3 (June 1 - June 19)

- Understand blockchain, docker, containernet, SDN, etc concepts
- Setup and get familiar with the development environment - run tests, read logs, etc
- Get high level understanding of the current umbra code - advanced Python ramp-up
- From above learnings, improve umbra docs, setup scripts, etc if needed
- **Outcome:** setup [dev environment](#) and run the project. Familiarized with the code, run the sample fabric configs, understand the output.

### Week 4 - 6 (June 22 - July 10)

- Pathfinding: ability to save and replay packet. E.g. each peer saves each packet sends so umbra-broker can instruct all the nodes to replay the packet
  - **Outcome:** current maintainer has the sample implementation (umbra/umbra-agent, umbra/umbra-monitor) that will be integrated to umbra-broker. Need to study and understand how it works
    - umbra/umbra-agent: Uses third-party tools like Linux `tcpdump` and `tcpdump` to save and replay packet
    - umbra/umbra-monitor: Monitor host platform (via `platform` Python standard lib), Linux processes (via `psutil` third-party library), and Docker container (via Docker Python API)
    - Learned a lot about python asyncio which is used extensively to schedule commands in both umbra-agent/monitor to simulate the environment
    - Found issue with scheduler.py where it cannot run command repeatedly. Raphael Rosa fixed it with this [commit](#)
- Implement the ability to remove container(s), remove/update link behavior (bandwidth, delay, loss, etc), and change node resources (cpu/mem) in umbra-scenario component
  - Test feature - refer [umbra/examples/fabric/build\\_configs.py](#)
  - **Outcome:** pending, moved to next quarter. But learned a lot about the internals design and how the umbra-scenario <--> umbra-broker works together.

### Week 7 - 9 (July 13 - July 31)

- Continue working on packet save/replay feature.
- Add events to umbra-broker based on previous quarter link/node behavior changes in umbra-scenario
- Review implementation details with Umbra maintainer
- **Outcome:** rough implementation of event that can update environments like stopping container, updating cpu/mem resources, and link interface status (UP or DOWN)

## Week 10 - 12 (Aug 3 - Aug 21)

- Add the ability in broker to parse events from config (e.g., event with “category” field set to “scenario”), containing the proper params to call the scenario modifications (e.g., node: name, operation: delete, etc)
- Generate a report for this fuzzing activity - list out all the events, status, etc
- Document and wrap-up work (scripts to automate stuff, etc)
- Prepare for project presentation
- **Outcome:** Two PRs for code changes [PR#9](#) (integrate umbra-agent + monitor + environment plugin), [PR#10](#) (Refactors + Simulation report + update docs)
  - Ongoing work on how to use Umbra as university course. Verdict: might be too ambitious to use Umbra as teaching material at this current state. Lots more efforts required to make it nice and easier to use by students.

## Process

- Weekly meeting with mentor and other mentees that use Umbra. Other meetings scheduled as required
- Chat via Signal

## Project presentation



Hyperledger Men...ork Fuzzing.pdf