Project Tutorial

CS 143 - Communication Networks

Ritvik Mishra

Presentation Outline

- Project Overview
- Requirements
- Network Components
- Design Remarks
- Tools
- Schedule
- Q&A

Project Overview (1)

- What is a network simulator?
  - Takes input
    - Network topology, specifications
    - Flows to occur
  - Simulates realistically
  - Produces performance metrics

Project Overview (2)

- Why?
    - Learn about theoretical results of congestion control, routing in lecture/homework, confirm and visualize in simulation
    - Software engineering project, practice technical skills and making design decisions

Requirements (1)

Components of a network

- Hosts - end points of the network
- Routers - route packets through network
- Links - connect hosts and routers
- Packets - units of information in network
- Flows - generate packets to send on network

Hosts

- End points in the network
- Very simple - in most implementations, the complexity will be in the flows
- Hosts know nothing about other hosts in the network

Routers

- Route packets through the network
- Calculate routing tables with decentralized shortest-path algorithms
  - Given a packet destination, decides where to send the packet next
  - No global knowledge of the network topology

Links

- All links in the project will be half-duplex
  - Data can go both ways, but only one way at a time
- Have a capacity, how fast they can send data (i.e. 10MB/s)
- Have a transmission delay, how long a packet will go from one end to the other
- Have buffer of packets to send on either end
  - Packets dropped when buffer is full

Packets

- Units of data that are sent through the network
- Contain:
    - metadata (source, destination, etc)
    - payload (actual data to be transmitted) OR
    - acknowledgement

Flows

- Represent data being sent from one host to another
- All flows in the project will send a finite amount of data at a specified time
- Do not know about other flows in network
- Generate packets to send from source to destination in accordance with congestion control algorithm

Design Remarks (1)

Everything is completely up to you!

- Any programming language
- You design system architecture
- Make smart decisions here...

Design Remarks (2)

Simulator Design

- Continuous simulation
- Discrete event simulation
- Process based simulation

Design Remarks (3)

Continuous Simulation

- Increment time in very small steps, at each step calculate what happened since the last step
- Continue until everything is done

Design Remarks (3)

Discrete Event Simulation

- Global queue of events and when they occur
- Once finished processing the last event, pick the item on the queue that starts first, process it, and continue
- End when queue is empty

Design Remarks (3)

Process Based Simulation

- Everything in the simulation runs as its own process
- Processes interact with each other
- Complicated to fully implement, but frameworks such as SimPy and C++Sim already exist

Tools

- Required to use source control
  - Github, Bitbucket would be easiest
- Can use issue/track trackers
- Some form of code review
- Get familiar with command line tools
  - Logging state and grep-ing is a good debug tool

Schedule

Week 5: Milestone 1

Week 8 Milestone 2

Week 10: Milestone 3

Week 11: Final Submission

Milestone 1

Architecture Presentation

- Meet with your group to make all high-level decisions
- Give a 10 minute presentation outlining your architecture, including all major decisions you made
- Architecture can (most likely will) change throughout the term

Milestone 2

Preliminary Implementation Presentation

- Should be able to handle Test Cases 0, 1
- Fully implemented hosts, links, packets
- Basic router code (static routing)
- Basic flow code (no congestion control)
- 10 minute presentation on progress
- Short demo

Milestone 3

Final Project Presentation

- Fully spec compliant product, only finishing touches left
- 15-20 minute presentation
- Short demo, overview of simulation results
- Analysis of results compared to theoretical expectations

Final Submission

Code + Final Report

- Turn in source code (link to repo)
- Turn in final report
  - Describe full project, including usage
  - Present and describe results for Test Cases 1, 2
  - Present analysis of Test Case 2 results compared to theoretical expectation
  - Comment on project process

Questions?