Project Tutorial

Daniel Guo
Fengyu Zhou

Caltech

Oct 2018

Based on previous versions edited by Ruijia, Zilong and Ritvik

Outline

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

Project Overview

What is a network simulator

- Takes input (Network topology; Flows)
- Simulates realistically
- Produces performance metrics

Project Overview

Objective

- 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

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

Host

End points in the network

- Has one interface connecting to a router
- Hosts know nothing about other hosts in the network
- Implementation should be simple
  - the complexity will be in the flows

Router

Route packets through the network

- Has multiple interfaces (with different IP addresses)
- Calculate routing tables with decentralized shortest-path algorithms
  - Given a packet destination, decides where to send the packet next
  - Exchange routing tables

Link

Connect hosts and routers

- All links in the project will be full-duplex
  - Data can go both ways independently
- Capacity: how fast they can send data
- Propagation delay: how long a packet will go from one end to the other
- Transmission delay: how long the link takes to transmit a packet
- Have finite buffer on either end

Packet

Units of data that are sent through the network

- Contain:
  1. metadata (source, destination etc.)
  2. payload (actual data) or ack
- Types: Payload, Control (from routers)

Flow

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
- Each flow has its own congestion control algorithms (Reno, Vegas, FAST …)

Design Remarks

Everything is completely up to you!

- Any programming language
- Any system architecture
- Make smart decisions

Design Remarks

Simulator design

- Discrete event simulation
- Continuous simulation
- Process based simulation

Design Remarks

- 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

- 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

- 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, GitBucket, etc.)
- Recommended
  - Some forms of code review
  - Get familiar with command line tools

Timeline

- Week 6-7: Mid-term presentation
- Week 8: Demonstration to TAs
- Week 10 (or 11): Final presentation
- Week 11: Final Submission

Mid-term Presentation

Architecture

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

Presentation to TAs

Preliminary Implementation

- 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)
- Short demo

Final Presentation

Final Result

- Fully spec compliant product (handle Test Case 0, 1, 2)
- 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 labor division
  - Present and describe results for Test Cases 1, 2
  - Present theoretical analysis for Test Case 2
  - Comment on project process

Weekly Meeting Schedules

- Every team is required to make a brief progress report with either of TAs every week (Week 4 – 10)
- Meetings are scheduled by email
  - Daniel Guo: lguo@caltech.edu
    - Tuesday: 3-5 pm
    - Thursday: 10 am-12 pm, 7-9 pm
  - Fengyu Zhou: f.zhou@caltech.edu
    - Tuesday: 6-9 pm
    - Wednesday: 6-9 pm

Q&A