Simulating the Networks

Kai Shen
Dept. of Computer Science, University of Rochester

Outline
- Network topology
- Discrete event simulator (my framework and a sample)
- How to run your own simulation?

Network Topology
- backbone topology: 
  /u/kshen/networks/netmodel/inet-3.0/inet.3037
  Format:
  nodes links
  ... id x y
  ... id1 id2 distance
  end hosts are randomly attached to backbone nodes
- latencies:
  - 8-12 ms for backbone links
  - 1-4 ms for edge links

A Project on Network Topology
- analyzing the raw data dumped from backbone routers to generate backbone topology graphs
- Two sources:
  - University of Oregon Route Views Archive Project
    http://archive.routeviews.org/
  - National Lab for Applied Network Research
    http://most.nlanr.net/Routing/rawdata
- Your results could be used other project groups!
Discrete Event Simulator

An example - modeling the working of a server:
- For each incoming request, the server processes it if the server is idle. The request is rejected if the server is busy processing a previous request. (No concurrency/no queuing)
- Request int-arrival time follows even distribution between 0 and 2*average.
- Request processing time follows even distribution between 0 and 2*average.
- Two types of events: request arrival, request complete.

How to run your network simulation?
- Keep the event chain management kernel and all utilities; replace my sample main program for simulating a concurrent server.
- Simulation of a network protocol:
  - declaring all possible events, data structures (e.g., a pointer to a node/server) can be added into the event structure
  - reading in the backbone topology from a file, adding end hosts, assigning link latency; initializing all data structures
  - schedule initial events into the event chain
  - walk the simulation by repeatedly taking out the next-in-line event and process it; (the processing of an event may result in more future events injected into the event chain)
  - the whole thing ends at a simulated time of your choosing; not too short to have a stable result

Overlay Network Simulation
- For simplicity and simulation speed, you might only want to simulate the end hosts, not backbone nodes (routers).
- For end hosts, you will need to simulate events (e.g., arrival of a message); maintain data structure (e.g., local routing table)
- What happens when an end host sends a message to another end host?
  - simulate internet routing?
  - go through the shortest path?
- Realism vs efficiency

My Framework
- My discrete-event simulation framework: 
  /u/kshen/networks/simulator
- Components:
  - Event chain management kernel: events.h/events.cc
    - schedule()
    - next_event()
  - Distribution utilities: distribution.h/distribution.cc
    - Even, normal, exponential, etc.
  - Supporting utilities: ran1.h/ran1.cc, hrtimer_x86.h/hrtimer_x86.cc
  - A sample simulation program: sample_sim.cc
    - Declare simulation-specific events and their handling
    - Maintain simulation-specific data structures