Introduction to ns2 - *Network Simulator*

Fall 2014

Varsha Apte (some slides from Kameswari Chebrolu)

Ns2

- A powerful tool for *discrete event simulation* of packet networking protocols

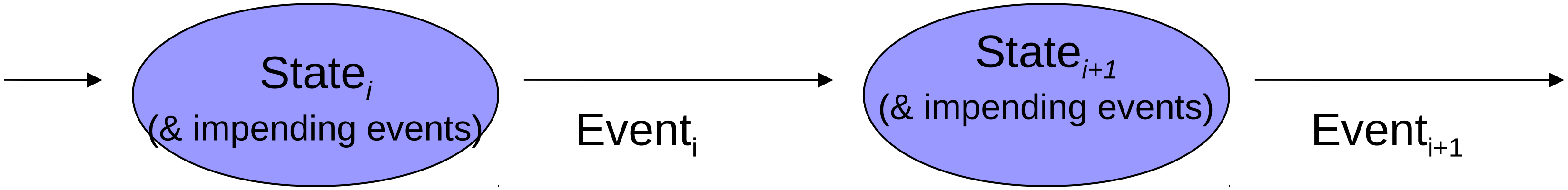
- But what is a “discrete event simulation”?

Discrete Event Simulation

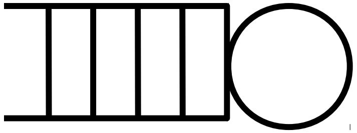
- A general methodology for studying behaviour of dynamic systems & for calculating system performance metrics
- A dynamic system (e.g. a network) has
  - *a state* (e.g. which packets are where, and at what stage of tranfer)
  - *A number of impending events* (e.g. finishing of a packet transmission)
    - When an event happens, the state of the system changes
    - When state of the system changes, new events become possible

Discrete Event Simulation

- System can be visualized as being in a perpetual “loop” of changing state and events
- The “discrete” means that system state (is assumed to) change at *discrete* points in time, not continuously
- These discrete points in time are when “events” happen
- Thus Discrete Event Simulation



Example: network link



- State?
  - link busy or not
  - packets waiting in the queue, their sizes etc
- Impending events?
  - Arrival of a new packet
    - State change?
      - (buffer state, link state if it was idle)
    - New events?
      - (next arrival)
  - Finish transmission of packet whose transmission was in progress
    - State change?
      - (link may become idle)
    - New events?
      - (if new packet transmission started then, its finish is now impending)

Why simulation?

Cheap -- does not require costly equipment

Complex scenarios can be easily tested

Results can be quickly obtained – more ideas can be tested in a smaller timeframe

The real thing isn't yet available

Controlled experimental conditions

Repeatability helps aid debugging

Disadvantages: Real systems too complex to model

ns2

- Ns2 applies the basic discrete event simulation methodology to simulate a complex network
- You do not have to write a simulation program from scratch – it offers rich a high level language related to networking, which you can use to write a script
- Script is run and produces a trace
- A lot of analysis of performance metrics can be done if a time-stamped event trace is written out

Features of ns2

Protocols: TCP, UDP, HTTP, Routing algorithms, MAC etc

Traffic Models: CBR, VBR, Web etc

(CBR: Constant Bit Rate flow, for voice packet traffic
VBR: Variable Bit Rate flow, for video packet traffic)

Error Models: Uniform, bursty etc

Misc: Radio propagation, Mobility models , Energy Models

Topology Generation tools

Visualization tools (NAM), Tracing

ns2 basics

NS is an object oriented discrete-event simulator

- Simulator maintains list of events and executes one event after another
  - Single thread of control: no locking or race conditions
    - (All events are ordered chronologically and processed)

Back end is C++ event scheduler

- Protocols mostly
- Fast to run, more control

Front end is oTCL

- Creating scenarios, extensions to C++ protocols
- fast to write and change

ns2 script structure

Create the event scheduler

Turn on tracing

Create network topology

Create connections

Generate traffic

Insert errors etc

Now build a script

- Go back to lab02 and follow instructions to create and run your first ns2 script.