

Using NS2 Presentation1

Overview
and
Discrete Event Simulation

Goal

- Explore use of ns2 with students
- Save students' time
- Allow students to concentrate on network research rather than “fighting with” ns2

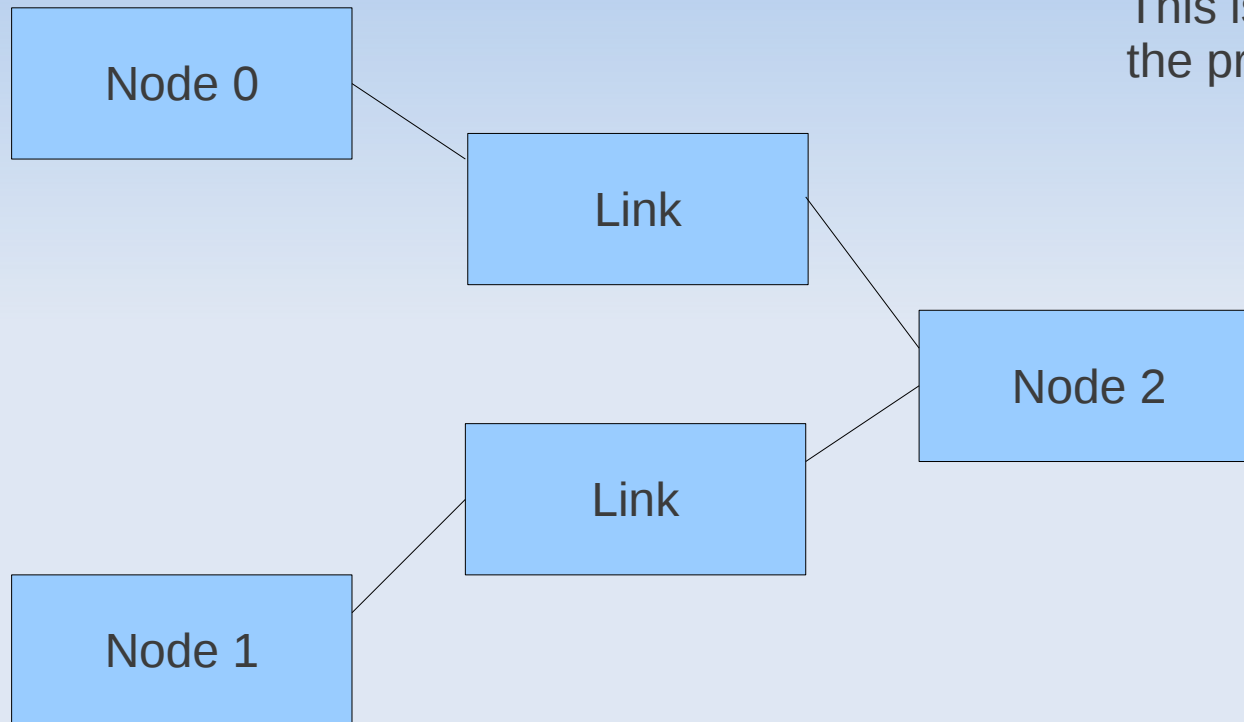
Overview

- Discrete event simulation; how does it work?
- Reading, modifying and writing tcl and otcl
- Mobile nodes
 - Wireless
 - Movement
- The same example (see handout) will be used throughout the first two presentations

Discrete event simulation

- Discrete event simulation as applied to computer communications networks
- Data structure describing the network components and their interconnection
- Event queue
- Event routines

Network model



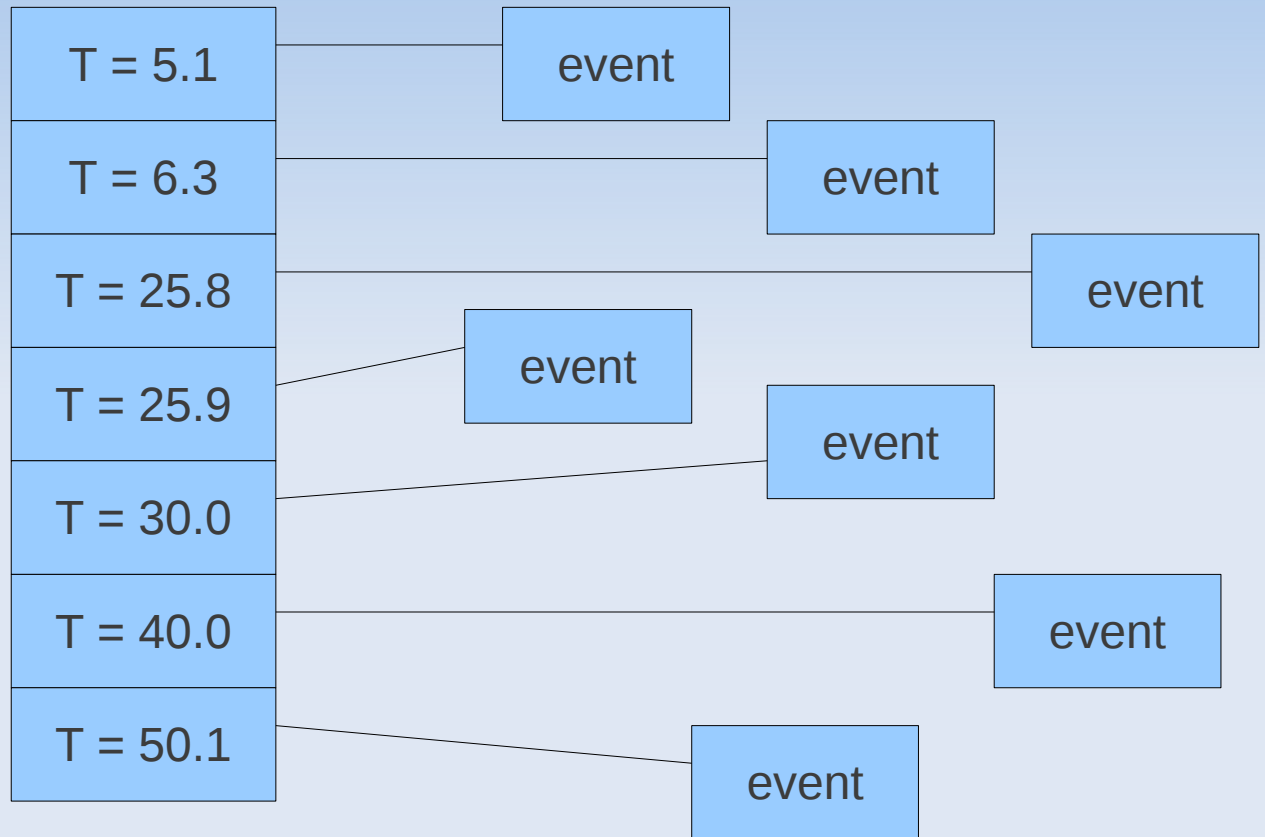
This is not the full diagram for the presentation example

Essentially, a “data structure” of blocks of memory containing information about the various components and their relationships to other components.

Event queue

- A queue of things (“events”) which happen at specific (simulated) times during the simulation
- Conceptually, ordered by increasing values of time

Event queue with event routines



Etc.

Event routines

- Some program code that performs a function during the simulation that can be considered to happen “instantaneously”, that is, takes “zero” time to happen
 - Example: queue a packet to be sent on a link
 - But not: transmit a packet over a link because the transmission must be considered to “take an interval of time” to happen

Types of event routines

- Built in to ns2
 - Scheduled by the simulator itself
 - Example: receive a packet at the destination of a link some time interval after the source starts transmission
- String of otcl code written by the simulation programmer
 - May call an ns2 “method”
 - May call an otcl proc (“procedure”) written by the simulation programmer

Generation of events

- It is quite common for an event routine to generate additional events and insert them into the event queue
- Example: the event routine that starts the sending of a packet on a link adds an event to process receiving of the packet at the destination
 - The time for this generated event will be the current time (which is the time at which sending starts) plus the transmission time (computed) plus the propagation delay of the link (specified)

The simulator's algorithm

- Take the first element (with the lowest time value) from the event queue
- Set the simulator's “current time” to the time of the event
- Execute the associated event routine
- Repeat the above steps until the event queue is empty OR an event routine executes the “exit simulator” command

The simulation programmer's role

- Write a program that does the following
 - Creates the data structures describing the network components and their interconnection
 - Puts some initial events into the event queue
 - **FINALLY**: starts the simulation algorithm running
- **IMPORTANT**: note that if there are no initial events in the event queue, the simulator will do nothing other than simply exit
- **EXAMINE** the example now