AN-Sim : Active Network Simulator

Samrat Bhattacharjee Ken Calvert Ellen Zegura

Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA.
`http://www.cc.gatech.edu/projects/canes`

An active network simulator

AN-Sim is:

- (Yet) A(nother) general purpose discrete event simulator
- Written mostly in C, parts in C++
  - Yet configurable (at runtime)
- Designed to work with *large* topologies

AN-Sim : Design Goals

- Gracefully incorporate arbitrary node designs
  - able to experiment with different *active* functions
  - and combination of active and non-active nodes
- Be able to
  - support wide-area topologies — $O(10^3)$ nodes
  - simulate total number events — $O(10^7)$ events
  - support several (text/graphical) front ends
- ... and be *reasonably* quick

AN-Sim : Assumptions

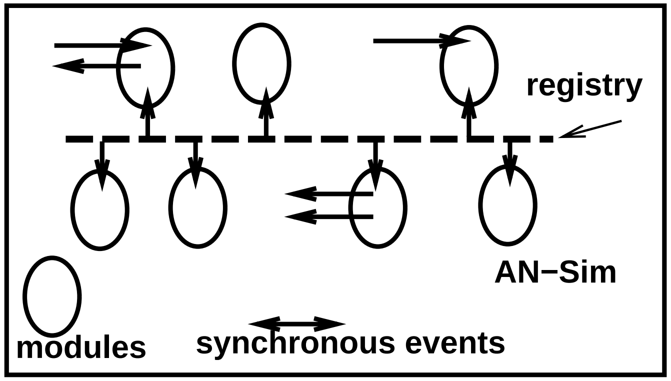
- Interested in topology, network design properties
  - Correlation of property with topology
  - e.g. How does cache hit ratio change with increase in degree of backbone nodes?
  - e.g. How well does the protocol work when the number of active nodes is doubled/halved?
- Not for detailed simulation of small systems
  ns, **opnet** already do that, well.
- Users not averse to writing in non-scripting languages

AN-Sim : Topologies

- Uses the **gt-itm** – Georgia Tech-Internet Topology modeling toolkit, `http://www.cc.gatech.edu/projects/gtitm`
- **gt-itm** provides:
  - Models of network geography, i.e., structure that goes beyond simple topology to include policy and other considerations, including known scaling properties
  - Compositional techniques for abstracting large internets as aggregates of smaller geographical components

AN-Sim : Design

- Designed with autonomous code modules that use *Synchronous Events* to communicate.
- Graph models*, Module Registry, Event Registry, and Event Invocation are the only "core" parts of the simulator



- Rest are configured at run time

Modules and Events

- Named code blocks
- Implements *specific* functionality
- May **bind** to and **raise** events
- Exports **handlers** for synchronous events
- Thus, events provide an anonymous publish-subscribe interface
- In general, modules not aware of other modules installed/active

Currently available modules

`event-gen`	`rand`	`dist-zipf`
`unicast`	*multicast*	`node-arch-0`
`node-arch-1`	`cache`	`text-log`
`socket`		

- In general, active functions should be written as modules.

Synchronous Events (Syncents)

- Mechanism for composition and communication
- Event handlers bind to specific events
- Generic event Interface
- Example:

```
bind_to_event (AN_NULL_EVENT_LIST,
generate_events, 0x0);
```

- Event Registry — dynamic event list
- Arbitrary number of handlers for each event

minor Detail: `Generic Functions`

- Some *events* will only have one function bound
  e.g. Distribution and generation functions – generate integer uniformly at random between $0..n$
- Instead of invoking event for these functions, use a generic function pointer

A Complete Example

Module	Event	Handle
`rand`	an-rand-long	`long-rand`
`dist-zipf`	an-source	`gen-query-source`
`dist-zipf`	an-dest	`gen-pop-dest`
`dist-zipf`	an-object-id	`gen-object-id`
`event-gen`	AN-NULL-EVENT-LIST	`generate-events`
`log`	AN-EVENT-COMPLETE	`log-event`
`node-arch-1`	AN-ACTIVATE-BEGIN	`an-fn-evaluator-1`
`unicast`	UNICAST-DG-FORWARD	`unicast-fwd`
`log`	UNICAST-DG-ARRIVAL	`u-dg-log`
`unicast`	UNICAST-DG-ARRIVAL	`unicast-reply`

Case Study: Caching (v0.0)

- Evaluation of Active Caching
- Several different caching policies
- Topologies : 700-2000 nodes
- Cache Sizes — $O(10^3)$ items per interface
- Several different access polices
- Caching as a module in v0.1

Representative Caching Result

- 1500 nodes, 150 servers, 60–1500 caches, 3.71 avg. degree
- 2×10^9 objects, 10^6 Queries-Response pairs
- Output:
  - Round trip latencies, Average Lifetimes,
  - Hits, Misses, Flushes, Occupancy
- Routing setup: approx. 30-40 seconds
- In general simulation : 4–20 minutes (Ultra-1, 167 MHz)
- v0.1 : 10^6 Queries — 130 seconds (33 seconds routing setup)

AN-Sim : Revisions, availability

- Version 0.0 implemented in Summer 1996
- Version 0.1 implemented in Summer–Fall 1997
- modules for Version 0.1 being implemented now
- Useful version available *early next year*