The NAM Network Animator

Mark Handley
USC/ISI

What is a Network Animator?

Nam takes packet trace data (either from ns or from real network traces) and presents it as a graphical animation of packets flowing on a network.

This allows you to examine interesting events, and to get an intuitive feel for what your protocol is actually doing, the effects of queuing, topology, etc, that is sometimes difficult to get from traditional static graphical summaries of the same information.

Nam history

- **Nam was first written by Steve McCanne in 1990, and functionality was enhanced slowly over time as required.**
- **Additional changes were added by Marylou Orayani, and more recently Nam has started to be supported by the VINT project, and significant additional functionality has started to be added.**
- **Nam is still in alpha-release form!**

Nam's basic model of a network

- **A network consists of nodes and links**
  - an interface is simply a link leaving a node
- **Links can optionally have queues associated.**
- **Packets:**
  - can be inserted into and taken out of queues
  - transmitted and received on links
  - dropped from queues or links
- **Agents can be associated with nodes or node-link interfaces.**
  - agents can contain named "features" such as variables, timers, lists, etc.
- **Routing information can also be associated with interfaces.**

Nam's basic model of a network

- **So long as the simulation you wish to animate conforms to Nam's model of a network, animation is a simple process.**
- **Extending the Nam model is currently quite difficult, and it is one of our goals to make certain aspects of extension much simpler.**

Nam versions

- **Nam 0.8a has been available from Berkeley for some time.**
  - animates packets, queues, nodes, links, and plays forwards
- **Nam 1.0 is our current experimental version with enhanced functionality. Support for Nam 1.0 narrowly missed being in the release cut for NS 2.0 last week, although it will be in the next beta release.**
  - adds routing animation, link drops, link and node failures, agent and protocol state animation, plays forwards and backwards, and is more interactive.

Nam 1.0 tracefile generation

- **NS 2.0 can generate tracefiles directly to be used by Nam 1.0. Some additional commands can be added to the NS TCL script to specify link directions, node shapes and colours of packets, links, etc.**
- **NS then uses this information along with information about propogation delay, link speed, etc to write a tracefile header followed by all the events you ask it to trace.**
- **Beware: asking NS to dump all trace information about long complex simulations is slow and takes a large amount of disc space!**

Nam tracefile format

```
+ -t0.0366 -s2 -d7 -pack -e40 -i3 -a10
- -t0.0366 -s2 -d7 -pack -e40 -i3 -a10
h -t0.0366 -s2 -d7 -pack -e40 -i3 -a10
r -t0.0379653 -s4 -d9 -pack -e40 -i2 -a20
+ -t0.0379653 -s9 -d8 -pack -e40 -i2 -a20
- -t0.0379653 -s9 -d8 -pack -e40 -i2 -a20
h -t0.0379653 -s9 -d8 -pack -e40 -i2 -a20
R -t0.038 -s9 -d8 -g24 -miif -p* -T2.0
a -t0.1 -s3 -d8 -n"TCP 21"
f -t0.1 -s3 -a"TCP 21" -ncwin -Tv -v1000
+ -t0.140632 -s7 -d6 -pack -e40 -i23 -a11
- -t0.140632 -s7 -d6 -pack -e40 -i23 -a11
h -t0.140632 -s7 -d6 -pack -e40 -i23 -a11
- -t0.142221 -s8 -d9 -ptcp -e1000 -i17 -a20
h -t0.142221 -s8 -d9 -ptcp -e1000 -i17 -a20
l -t0.145293 -s8 -d9 -SDOWN
n -t0.145293 -s8 -SDOWN
R -t0.145293 -s8 -d9 -g24 -moif -p* -T1.892707 -x
```

A Still Frame from a Nam Animation

Getting NS to dump a Nam tracefile

```
set ns [new MultiSim]
$ns trace-nam [open nam.tr w]
# set up colour table
$ns color 0 red
$ns color 1 blue

# create node with given shape and color
set n(0) [$ns node circle red]
set n(1) [$ns node circle blue]
...
# create links and layout
$ns duplex-link $n(0) $n(1) 1.5Mb 10ms DropTail right
$ns duplex-link $n(1) $n(2) 1.5Mb 10ms DropTail right-down
...
# set up simulation
.....
$ns at 3.5 "$n(0) change-color red"
....
$ns at 10.0 "finish"
$ns run
```

Interactivity

- **All animation objects can be clicked on to display further information, and monitors can be set to continuously monitor the status of an object.**

Future plans

- **These are our current ideas for how to make Nam more useful and effective in evaluating simulations. Please let us know what you think would be useful new functionality.**

Near Term Plans:

Animation Features

- **Currently screen distance is proportional to delay.**
  - this causes a problem when animating LANs or satelite hops in the same animation as terestrial WANs.
  - we're adding a mechanism to selectively break this proportionality when required.
- **Need a way to visualise LANs**
  - current thoughts are to do this as a virtual node so LANs will tend to look like stars.
  - other topologies make it more difficult to indicate who the sender is and add odd false delay implications.

Near Term Plans:

Animation Features: Interactivity

- **need ways to turn on or off the display of certain agents and routes.**
- **want to be able to interactively control the coloring and labelling of packets.**
- **want to be able to lay out simulation topologies more easily using nam in interactive mode, simulate them in ns, and view them again in nam.**
- **auto-layout would be nice for animations that originate in ns**
  - ns has enough information to give the delays and bandwidths to nam directly. It doesn't know about directions though.

Near Term Plans:

Animation Features: Queues

- **nam queues can now go forwards and backwards, but they still model FIFO queues.**
  - although we can drop other than the tail, but only when playing forwards! Need a queue position in the drop event when playing backwards: TBD
- **how to model non-FIFO queues?**
  1. model a class-based or priority queue as muliple queues?
  2. have tracefile primitives for queue re-ordering?

Near Term Plans:

Finding Interesting Events

This can be hard in nam

- **Addition of annotations and being able to provide an annotation index so you can jump to an annotation would help significantly.**
- **Being able to display the state of an object across the animation timeline would also help**
  - should be possible to graph all sorts of properties from within nam through the entire timeline to help you identify when interesting events occur.

Longer Term Plans:

Animation Features

Support interactively switching between many different views of the same animation:

- Time-sequence plots of individual connections
- load/loss shown on the network topology instead of packets
- zooming into parts of complex animations
- Tim Shepards TCP plots
- ???

Allow scriptable object animation

- TCL fragments to define how to display an object (particularly an agent) in various plots.
- Eg: display graphically SRM agents' request and response timers.

Summary

- **Nam is a useful tool for providing a new way of looking at various aspects of protocol behaviour.**
- **It doesn't replace traditional views such as sequence-number against time, which provide a summary of some aspects of behaviour.**
- **As we make Nam more interactive and add new features, feedback from you, the user community, is essential to ensuring that useful functionality is added. Let us know what you think!**