A user’s view of ns

VINT ns retreat

Lloyd Wood (L.Wood@surrey.ac.uk)
http://www.ee.surrey.ac.uk/showstaff?L.Wood

Centre for Communication Systems Research,
University of Surrey
http://www.ee.surrey.ac.uk/CCSR/
What I’m using *ns* for:

Playing with multicast across large mesh networks (satellite constellations). Trying to keep it simple; my understanding of *ns* is still pretty minimal.

Want dynamic topology (but static at present) with costs and shared core-based trees, so it’s CtrMcast or PIM-SM...

So, what’s up with PIM-SM? Just throwing protocol work away?
Education: watching my masters student...

He’s started building a TCP/IP over satellite simulator.

“Why does this work in Tk, but not in ns?”

ns is using an OTcl shell; shells are interactive.

parameter-passing isn’t. Leverage Tk! Then we can build educational tools (and gentler introductions to ns).

object-oriented ns -> filing system -> object-oriented nam? Overkill for educational purposes. Roll them together...
Stability and splintered worldviews

More than one of *anything* gives the implementer a choice of what to use (and what must be tested with) when adding other functionality. Will they use and test with everything? Apparently not.

Other oddities, e.g. multiPath_ only works with DV; multicast vs lans, stuff that only works with (or without?) mobile nodes, fun with addressing. I worry about derived classes complicating testing.

Validate doesn’t go anywhere near far enough; can it be made to run through every script?

Now documentation has expanded, example scripts are the weakest point. An annotated guide - 300-word summary explaining features of each script? *Good* example code needed.
Visible improvements in the last year

Seemingly trivial things do count for a lot with users:

**Visible CVS tree and class hierarchy.**
(but stuff checked in isn’t necessarily maintained... snoop? topology generators?) Easier for users to provide immediately useful feedback; easier for developers to shrug ‘well, it’s all visible’ and go and do interesting stuff instead.

**Online HTML documentation.**
Docs can never be too accessible.

**Marc Greis’ tutorial.**
Without proof that an outsider could demonstrate useful things with ns (then teach them to others) I’d have given up early.
Future additions

diffserv; being able to easily assign queueing behaviours to groups of nodes, a la PHBs. Research in QoS calculus?

Questioning MAC-layer tradeoffs. MAC support is on LANs. It’s in wireless. But many things (e.g. multicast) don’t work across lans; MAC can be significant for overall latencies; allows support for more realistic error models.

Errors - an error rate expressed as a proportion of variable-length packets just isn’t that useful...

Fragmentation.
Obligatory audience participation section

just checking you’re awake
Three futures for *ns* - California dreams
(with apologies to Kim Stanley Robinson)

**drastic triage.** throw away everything that is known not to work under all conditions; cut back and refine to produce a limited tool with a solid featureset. One way of getting out of beta, but not very research.

**unconstrained expansion** without control; wild additions without moderation for interactions; *everything* gets into the CVS tree, even if it’s not being actively maintained to stay current.

**controlled expansion.** Lever educational and commercial synergy to enlarge the *ns* userbase and pool of developers; grow *ns* carefully, without letting it get *too* wild and uncontrolled.
**Broadband integrated satellite network traffic evaluation**

might also be: Binary executable satellite-focused ns to end-users

Simulating future connectivity for ISPs and characterising networks; interested in simulating:
broadband satellite using IP over ATM QoS
GPRS (packet radio extensions to GSM)
but using existing protocols: telnet, http, etc.
Tool choice came down to ns or Opnet; ns won out.

Participants in Austria, Germany, France, UK.
Several years’ funding; early days.

http://www.bisante.org/
and finally.... since UCL is hosting IWQoS...

I asked jon crowcroft if there was anything he wanted me to bring up, and he replied:

> 1/ documentation :-)
> 2/ stability
> 3/ coding standards

...so there you have it.