ns-3 Project Plan

Tom Henderson and **Sumit Roy**,
University of Washington
Sally Floyd, ICSI Center for Internet Research
George Riley, Georgia Institute of Technology

June 24, 2006

2006 NSF CRI-PI Meeting

What is *ns*?

- *ns* is a discrete-event network simulator for Internet systems
  - protocol design, large scale systems studies, prototyping, education
- *ns* has a companion network animator called *nam*
  - hence, has been called the *nsnam* project

What is *ns* (cont)?

- *INSERT nam animation movie here*

ns-2 Impact

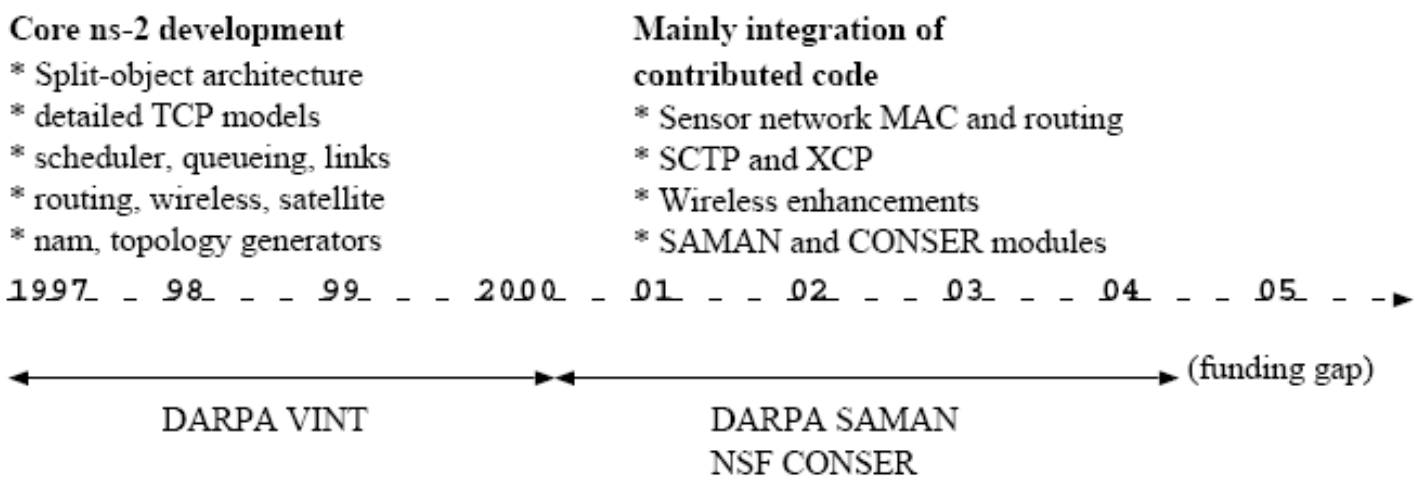
ns is a research community resource

Simulators	ns-2	OPNET	QualNet/GloMoSim
Transport layer and above	123(75%)	30(18%)	11(7%)
Network layer	186(70%)	48(18%)	31(12%)
MAC & PHY layers	114(43%)	96(36%)	55(21%)

Source: Search of ACM Digital Library papers citing simulation, 2001-04

- Other statistics:
  - Over 50% of ACM and IEEE network simulation papers from 2000-2004 cite the use of *ns-2*
    - Source: ACM Digital Library and IEEExplore searches
  - 8000 downloads/month, 450 messages/month on ns-users mailing list
    - November 2005-May 2006

ns-2 Funding History



- Funding on the simulator itself finished in 2000
- Current software is out-of-date
  - models, GUI, overall software design, scripting interface

Technical Goals of the ns-3 Project

Develop a redesigned network simulation tool for Internet research and education

- **Core:** Redesign the core of the simulator
- **Integration:** Better software encapsulation and integration
- **Models:** Updated protocol models
- **Education:** New educational support
- **Maintenance:** Testing, documentation, support

ns-3 Program Details

- Four-year, multi-institution collaboration
- External collaborations also desired
  - INRIA Sophia-Antipolis, Planete research group
  - Industry (TBD)
  - Existing ns-2 user/research community
- CRI funding supports 1+ staff programmers, 1+ students, PIs
- Software developed using freely available tools on commodity hardware
- Open source licensing and development model

Core: Refactor the ns core

Current limitations:

- Scalability, scripting interface, emulation support

Design themes:

- Features: C++ core, new scripting interface (TBD), improved emulation support, new animation
- Techniques: modern object-oriented design patterns, support for parallel execution and staged computations, better tracing and statistics computation,

Leveraging:

- Georgia Tech Network Simulator (GTNetS)
- yans (INRIA)
- Parallel, Distributed ns (PDNS)
- staging techniques such as SNS (Cornell)
- others

Integration: Reuse more code

Current limitations:

- protocol implementations need to be specially written for simulation environment or abstraction library
- trace files and simulation outputs are non-standard formats

Design themes:

- APIs and software support for process-driven implementations in an event-driven simulation framework
- standard (e.g., pcap) simulation outputs

Leveraging:

- Network Simulation Cradle (Jansen)-- methodology for porting kernel code into ns-2
- New techniques for linking existing application code
- Experience with porting quagga routing to ns-2 and GTNetS

Integration: Interact with real-world

Current limitations:

- emulation code is out-of-date
- difficult to transition between simulations and PlanetLab (or real) experiments

Design themes:

- revised emulation support
- interfaces for PlanetLab
- continued support of Utah's Emulab and other testbeds

Leveraging:

- University of Magdeburg (Mahrenholz) third-party emulation extensions
- planned collaboration with PlanetLab and Emulab projects

Models: Update available models

Current limitations:

- little support for peer-to-peer applications, IEEE 802.11 variants, IPv6 protocols, modern routing protocols, new network architectures (e.g., DTN)

Design themes:

- Emphasis on wireless, new traffic models, emerging protocols (e.g., high-speed TCP) and applications (e.g., BitTorrent), other models of important research/educational interest

Leveraging:

- Software from other open-source projects
- Contributed ns-2 code where possible

Community contribution of models has been outstanding for ns-2

Models: Update available models (cont.)

	Existing core ns-2 capability	Planned additions for ns-3
Applications	ping, vat, telnet, FTP, multicast FTP, HTTP, probabilistic and trace-driven traffic generators, webcache	Sockets-like API (to allow porting of existing applications to ns environment), peer-to-peer (e.g. BitTorrent)
Transport layer	TCP (many variants), UDP, SCTP, XCP, TFRC, RAP, RTP Multicast: PGM, SRM, RLM, PLM	TCP stack emulation (Linux, BSD), DCCP, additional high-speed TCP variants
Network layer	Unicast: IP, Mobile IP, generic dist. vector and link state, IPinIP, source routing, Nixvector Multicast: SRM, generic centralized MANET: AODV, DSR, DSDV, TORA, IMEP	full IPv4 support, full IPv6 support, NAT XORP/Click Routing support: BGP, OSPF, RIP, IS-IS, PIM-SM, IGMP/MLD
Link layer	ARP, HDLC, GAF, MPLS, LDP, Diffserv Queueing: DropTail, RED, RIO, WFQ, SRR, Semantic Packet Queue, REM, Priority, VQ MACs: CSMA, 802.11b, 802.15.4 (WPAN), satellite Aloha	new 802.11 model, 802.11 variants (mesh, QoS), 802.16 (WiMax), TDMA, CDMA, GPRS
Physical layer	TwoWay, Shadowing, OmniAntennas, EnergyModel, Satellite Repeater	IEEE 802 physical layers, Rayleigh and Rician fading channels, GSM

Education: More impact in courses

Current limitations:

- students find current scripting syntax arcane
- protocol models are sometimes too abstracted
- paucity of educational scripts

Design themes:

- more implementation-oriented architecture and software
- revised user interface
- integration with courseware and texts,

Leveraging:

- efforts to integrate more real-world code (above)
- simulation framework that better mirrors implementations
- use in PIs' courses

Maintenance

Funding for staff programmers to:

- Maintain ns-2 while we transition
- Implement the selected architecture
- Reuse and clean up existing ns-2 and GTNetS models
- Model validation and debugging
- Documentation
- Regression testing
- Software packaging and releases
- Educational script generation

Broader impacts

CRI funding intended to seed the larger *nsnam* project

- We'll continue to solicit inputs and participation from the broader networking community
- Project will use established open source development practices
  - ns-3 will use a free software licensing structure encouraging academic and commercial participation
- Our intent is to make the simulator a self-sustaining project driven by research community inputs and industry funding

Leverage and grow the "network effect" of ns-2's user base

Criteria for success

In four years, ns-3 will be a success if it:

- continues to be preferred simulation environment for network research
  - performance, scalability, openness
  - comprehensive and current model support
- allows easy integration of implementation code
- allows researchers to more easily move between simulation and live experiments
- contains current wireless and application models
- is used for undergraduate/graduate courseware
- project is self-sustaining beyond CRI funding

Questions?

Web site:

http://www.nsnam.org

Mailing list:

http://mailman.isi.edu/mailman/listinfo/ns-developers