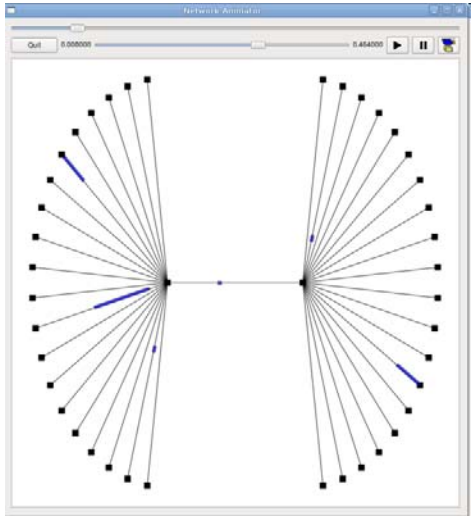
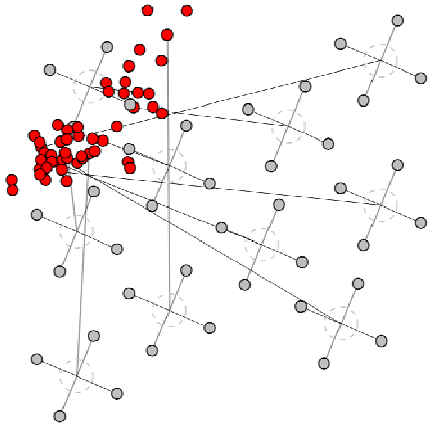
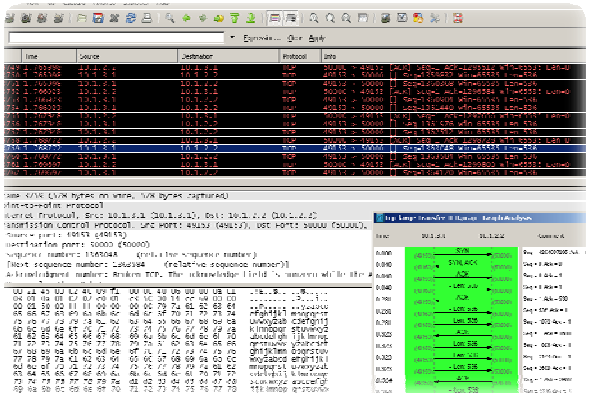
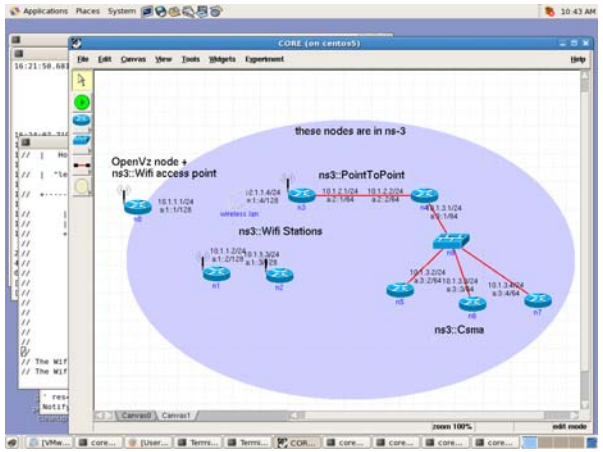
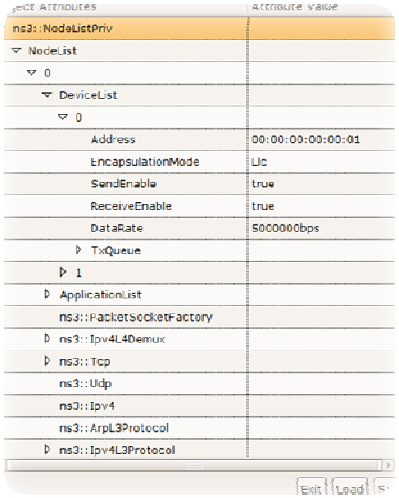
ns-3 Overview











http://www.nsnam.org

What is *ns-3*?

- *ns-3* is a discrete-event network simulator for Internet systems
  - ns-3 allows researchers to study Internet protocols and large-scale systems in a controlled environment
  - ns-3 is a new simulator (not backwards-compatible with ns-2)
- ns-3 is a free, open source software project organized around research community development and maintenance
  - the target user community is networking researchers and educators

ns-3 project goal

Develop a preferred, open simulation environment for networking research

1) a tool aligned with the simulation needs of modern networking research

2) an open-source project that encourages community contribution, peer review, and validation of the software

Overview

This presentation is organized around the goals stated above:

- "aligned with the needs of simulation research"
- "community participation"

ns-3 timeline and roadmap

- Project makes quarterly releases

ns-3.4: Apr 2009:
- Tap Device
- Object names
- new Wifi models
- calendar queue scheduler
- allinone build system

ns-3.5: July 2009:
- 802.11e MAC EDCA
- 802.11n A-MSDU frame aggregation
- 802.11b PHY
- Nakagami loss
- Gamma, Erlang, Zipf random variables

ns-3.6: Oct 2009:
- Minstrel rate control
- WiFi Athstats and 5/10MHz channels
- IPv6 radvd, ICMP
- 802.11s mesh
- Nix-vector routing
- Flow Monitor

ns-3.7: Jan 2010:
- 802.11p PHY
- AODV
- Waypoint mobility
- NetAnim
- IPv6 Extension and Option headers

ns-3.8: May 2010:
- MPI-based sims
- WiMAX
- 802.11n Block Ack
- Gauss-Markov and steady state random waypoint mobility models
- Matrix prop. loss mode
- Two-way ray prop model

Google Summer of Code
Three student projects

Google Summer of Code 2010

http://www.nsnam.org

Part 1: Overview of ns-3 features

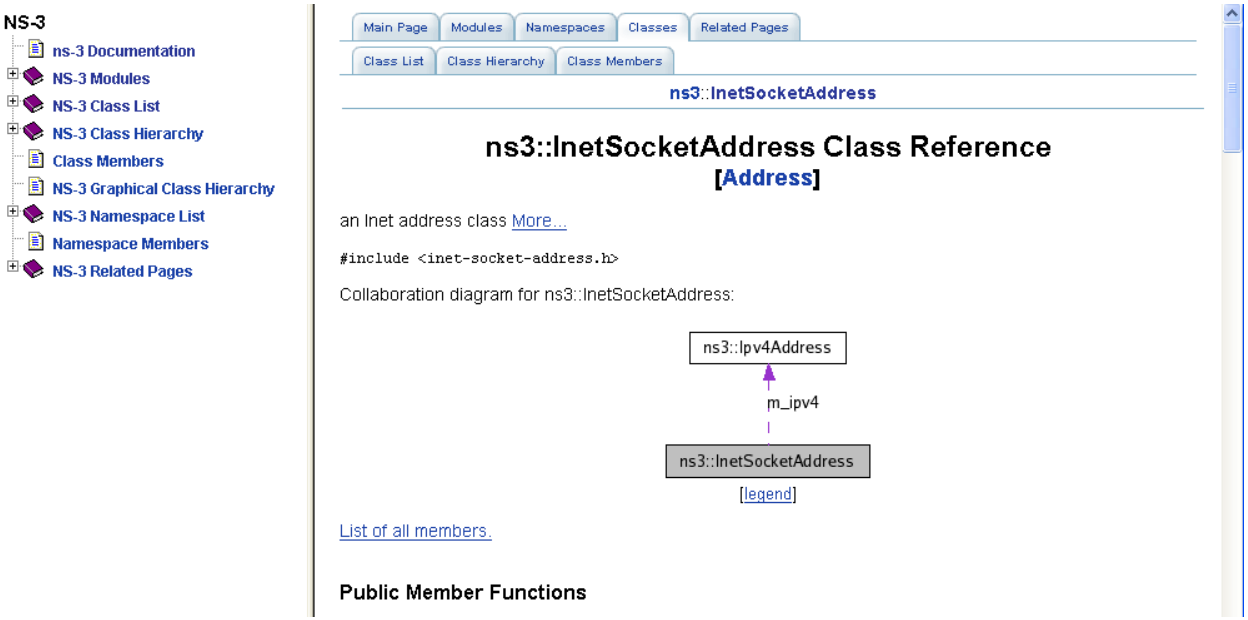
Requirements

ns-3 responds to trends in how Internet research is being conducted

1. extensible software core
2. attention to realism
3. software integration
4. support for virtualization and testbeds
5. flexible tracing and statistics
6. attribute system
7. new models

1) Extensible software core

- Written in C++ with optional Python interface
  - extensively documented API (doxygen):



  - see also: http://www.nsnam.org/documents.html

Extensible software core (cont.)

- Project focus to date has been on setting the long-term architecture
  - rather than developing new models

- Trying to avoid some problems with ns-2, such as
  - interoperability and coupling between models
  - lack of memory management
  - debugging of split language objects

Example: ns-3 object aggregation

Problem: coupling between models hinders software reuse in different configurations

- must intrusively edit the base class for this
- or, leads to C++ downcasting, e.g.:

```
// Channels deal with Node pointers, but here I really want a
// MobileNode pointer, to access the MobileNode API
double
WirelessChannel::get_pdelay(Node* tnode, Node* rnode)
{
        // Scheduler    &s = Scheduler::instance();
        MobileNode* tmnode = (MobileNode*)tnode;
        MobileNode* rmnode = (MobileNode*)rnode;
```

- known as the C++ “weak base class” problem

Example: ns-3 object aggregation

<u>ns-3 solution:</u> an object aggregation model

- objects can be aggregated to other objects at run-time
- a “query interface” is provided to ask whether an particular object is aggregated
- similar in spirit to COM or Bonobo objects

```
// aggregate an optional mobility object to a node:
node->AggregateObject (mobility);
...
// later, other users of node can query for the optional object:
 senderMobility = i->first->GetNode ()->GetObject<MobilityModel> ();
// we did not have to edit class Node (base class), or downcast!
```

2) attention to realism

Problem: Research often involves a mix of simulations and testbed or live experiments

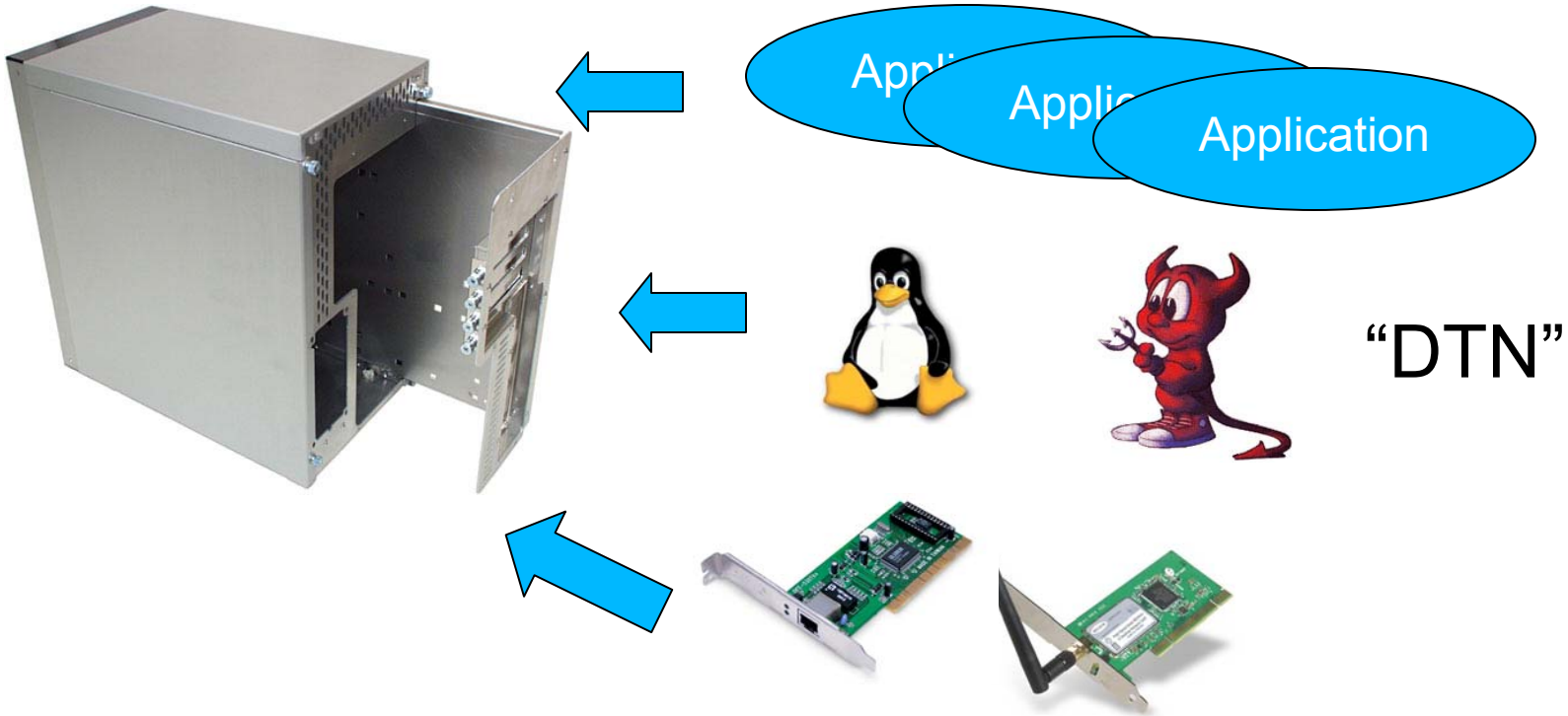
- If the simulator cannot be made to closely model a real system:
  - hard to compare results or validate the model
  - hard to reuse software between the two domains

ns-3 solution:

- model nodes more like a real computer
- support key interfaces such as sockets API and IP/device driver interface (in Linux)
- reuse of kernel and application code

attention to realism (example)

An ns-3 Node is a husk of a computer to which applications, stacks, and NICs are added



3) software integration

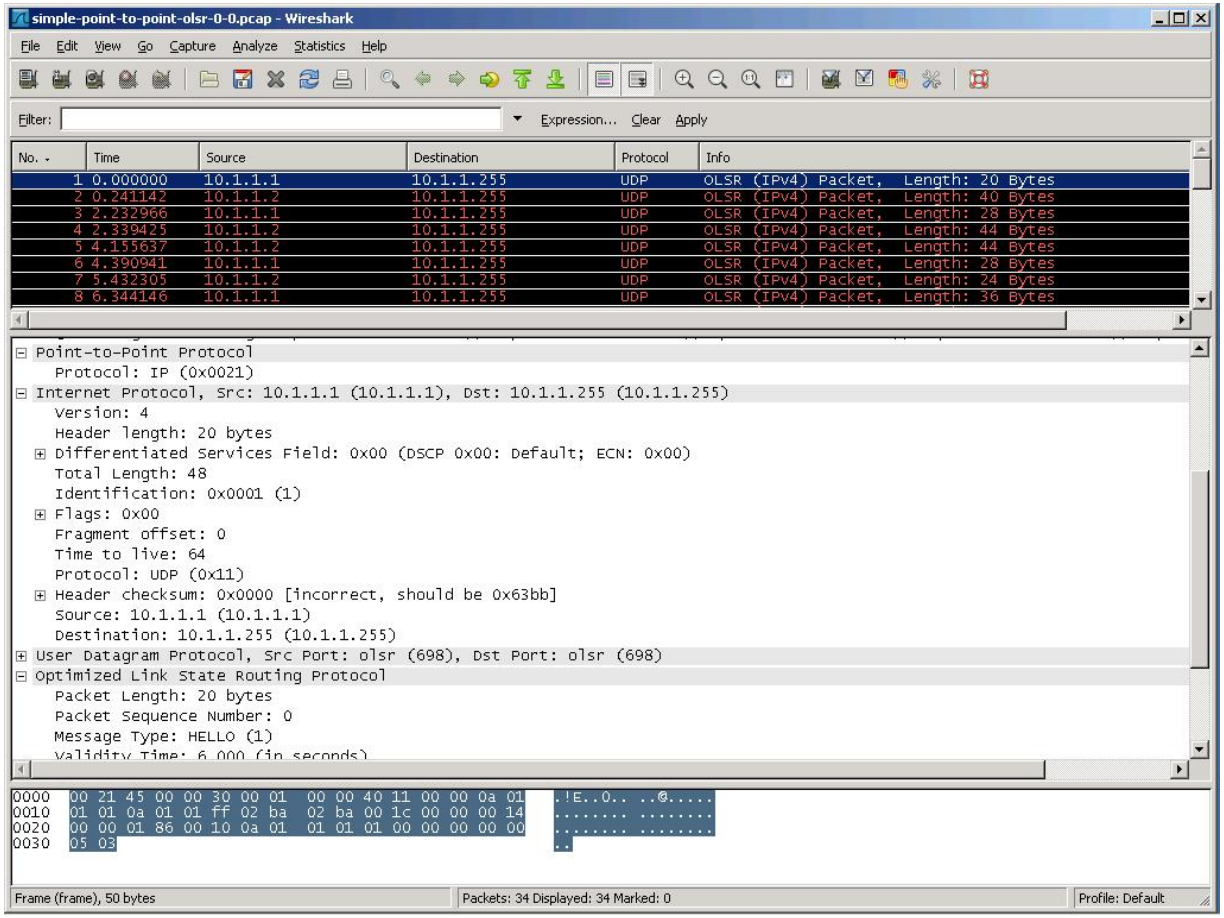
Problem: why reimplement models and tools for which open-source implementations abound?

ns-3 solution:

- ns-3 conforms to standard input/output formats so that other tools can be reused.
  - e.g., pcap trace output, ns-2 mobility scripts
- ns-3 is adding support for running implementation code
  - Network Simulation Cradle (Jansen) integration has met with success: Linux TCP code
  - ns-3 "process" API

software integration (cont.)

- Example: ns-3 trace viewed with Wireshark:



4) support for virtualization and testbeds

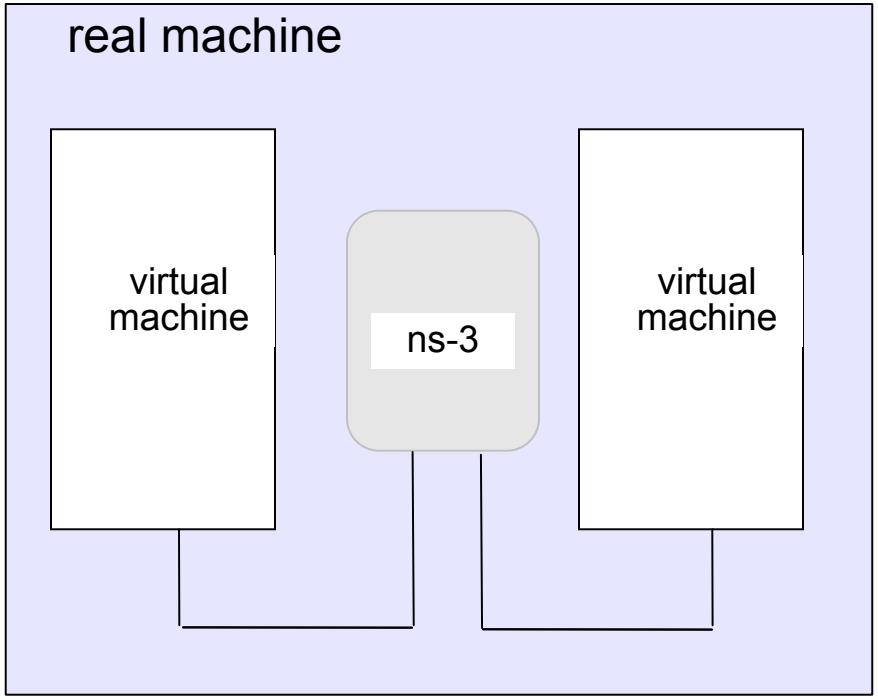
Problem: need better support for the researcher moving between simulation and testbeds or live systems

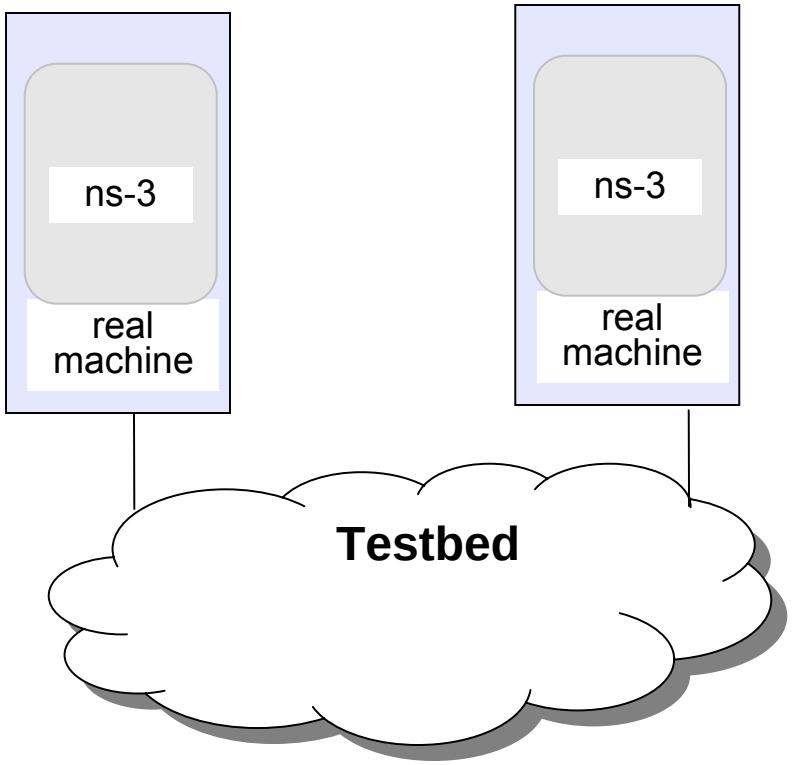
ns-3 solution: Developing two modes of integration with real systems:

1) virtual machines run on top of ns-3 devices and channels

2) ns-3 stacks run in emulation mode and emit/consume packets over real devices

ns-3 goals for emulation





1) ns-3 interconnects virtual machines

2) testbeds interconnect ns-3 stacks

Added in ns-3.5

Added in ns-3.3

http://www.nsnam.org

5) tracing and statistics

- Tracing is a structured form of simulation output
- Example (from ns-2):

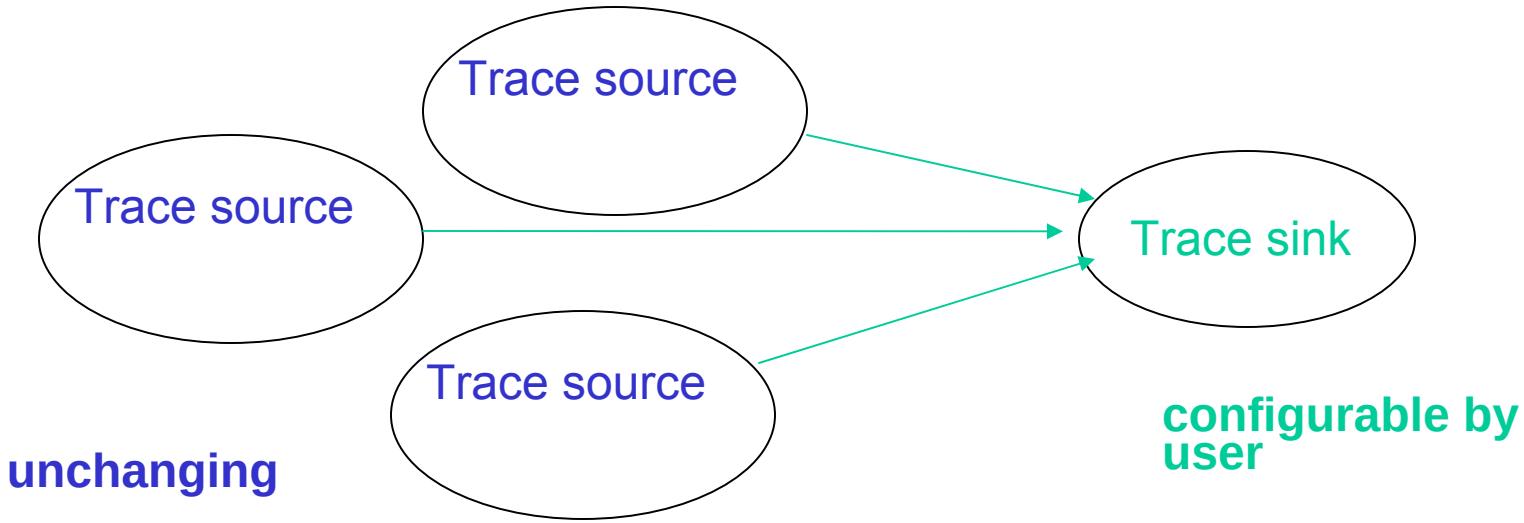
```
+ 1.84375 0 2 cbr 210 ------- 0 0.0 3.1 225 610
- 1.84375 0 2 cbr 210 ------- 0 0.0 3.1 225 610
r 1.84471 2 1 cbr 210 ------- 1 3.0 1.0 195 600
r 1.84566 2 0 ack 40 ------- 2 3.2 0.1 82 602
+ 1.84566 0 2 tcp 1000 ------- 2 0.1 3.2 102 611
```

<u>Problem:</u> Tracing needs vary widely

- would like to change tracing output without editing the core
- would like to support multiple outputs

ns-3 has a new tracing model

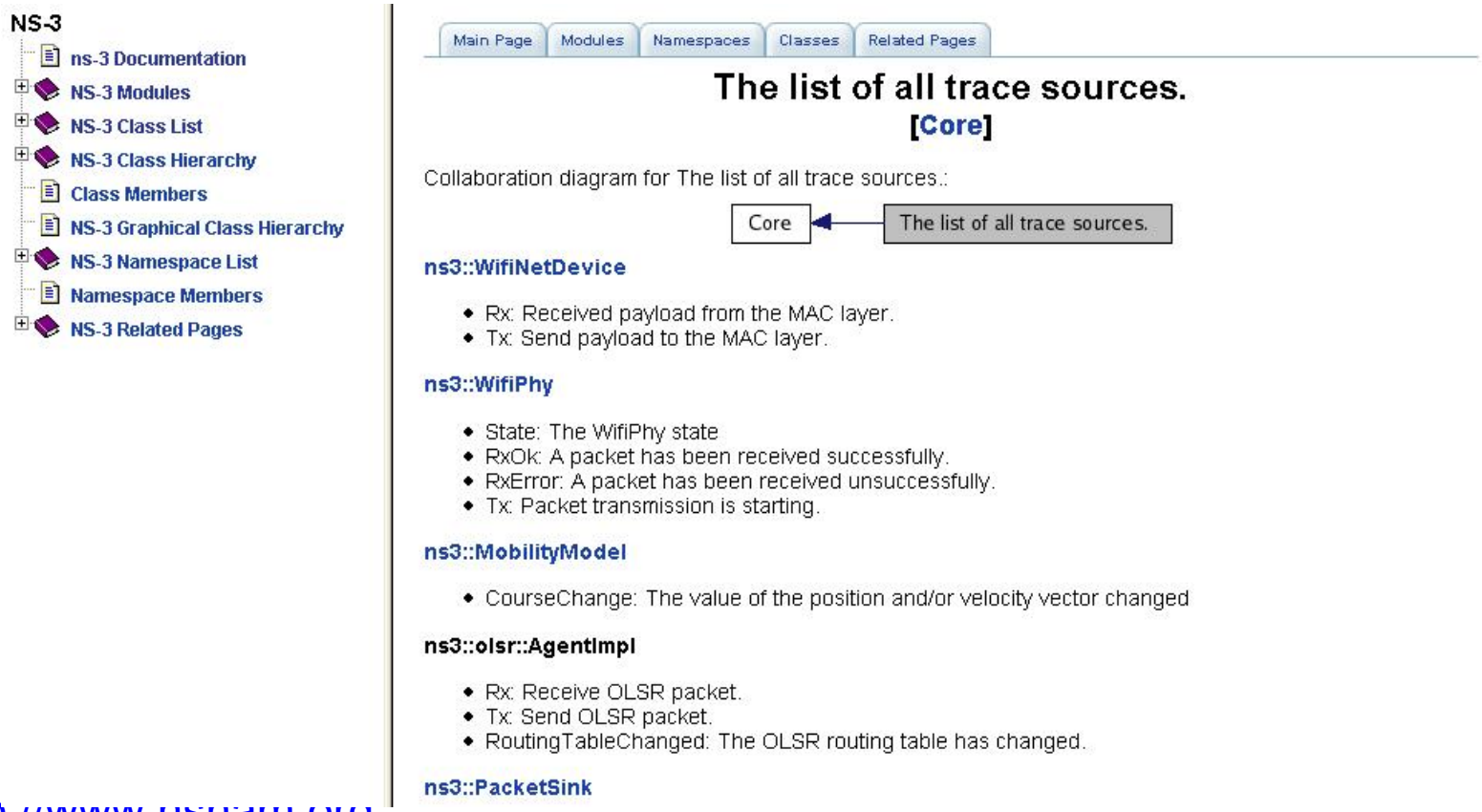
<u>ns-3 solution:</u> decouple trace sources from trace sinks



Benefit: Customizable trace sinks

ns-3 tracing

- various trace sources (e.g., packet receptions, state machine transitions) are plumbed through the system

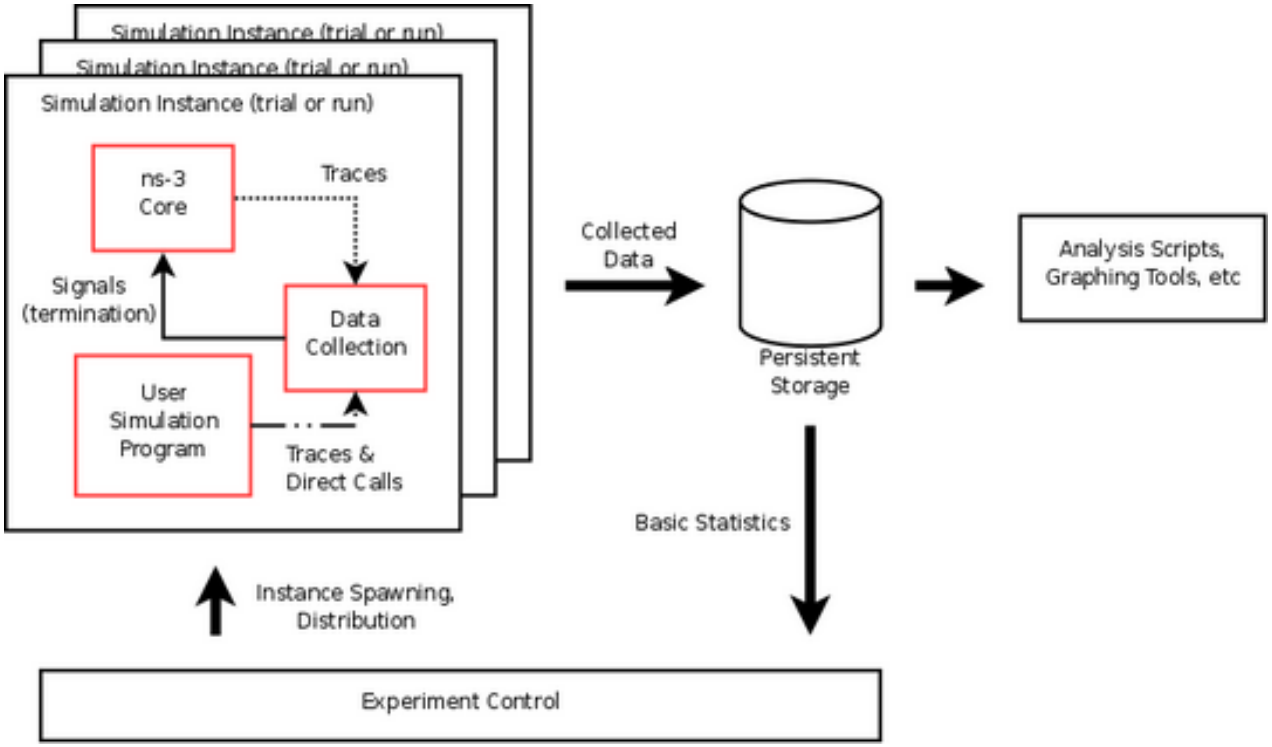


statistics framework

- Tracing system supports a statistical and data management framework
  - (currently under development)
- Features:
  - manage multiple independent runs of a scenario
  - marshal data into several output formats
    - including databases, with per-run metadata
  - hook into ns-3 trace sources
  - statistics objects can interact with simulator at run-time
    - e.g. stop simulation when counter reaches a value

statistics framework (cont.)

- Details at:
  - http://www.nsnam.org/wiki/index.php/Statistical_Framework_for_Network_Simulation



6) ns-3 attribute system

Problem: Researchers want to know all of the values in effect in their simulations

- and configure them easily

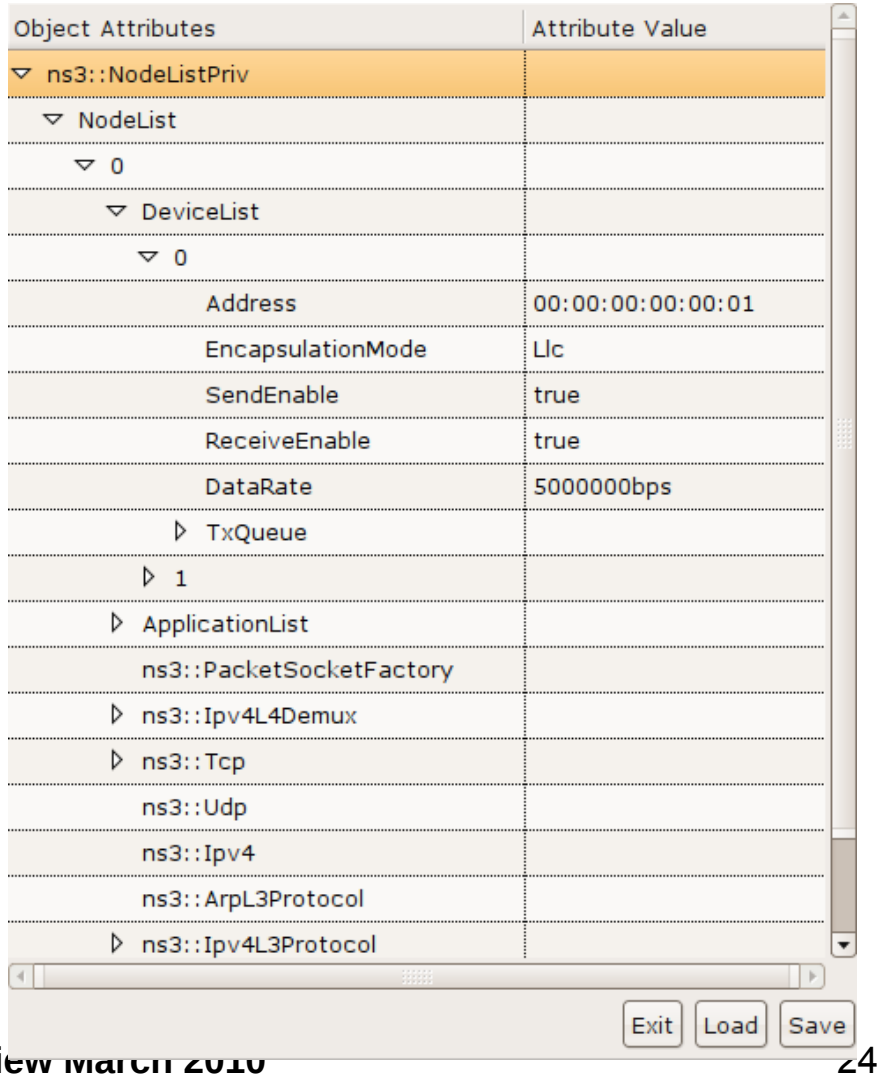
ns-3 solution: Each ns-3 object has a set of attributes:

- A name, help text
- A type
- An initial value

- Control all simulation parameters for static objects
- Dump and read them all in configuration files
- Visualize them in a GUI
- Makes it easy to verify the parameters of a simulation

ns-3 attribute system

- Object attributes are organized and documented in the Doxygen
- Enables the construction of graphical configuration tools:



http://www.nsnam.org

7) ns-3 models

	Existing core ns-2 capability	Existing ns-3
Applications	ping, vat, telnet, FTP, multicast FTP, HTTP, probabilistic and trace-driven traffic generators, webcache	OnOffApplication, asynchronous sockets API, packet sockets
Transport layer	TCP (many variants), UDP, SCTP, XCP, TFRC, RAP, RTP Multicast: PGM, SRM, RLM, PLM	UDP, TCP
Network layer	Unicast: IP, MobileIP, generic dist. vector and link state, IPinIP, source routing, Nixvector Multicast: SRM, generic centralized MANET: AODV, DSR, DSDV, TORA, IMEP	Unicast: IPv4, global static routing Multicast: static routing MANET: OLSR
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	PointToPoint, CSMA, 802.11 MAC low and high and rate control algorithms
Physical layer	TwoWay, Shadowing, OmniAntennas, EnergyModel, Satellite Repeater	802.11a, Friis propagation loss model, log distance propagation loss model, basic wired (loss, delay)
Support	Random number generators, tracing, monitors, mathematical support, test suite, animation (nam), error models	Random number generators, tracing, unit tests, logging, callbacks, mobility visualizer, error models

ns-3 includes a mix of new and ported models

summary of ns-3 features

- modular, documented core
- C++ programs or (optionally) Python scripting
- alignment with real systems (sockets, device driver interfaces)
- emphasis on software integration
- virtualization and testbed integration are a priority (emulation modes)
- well-documented attribute system
- updated models

Part 2: Community participation

ns-3 is an open-source project

- software is GNU GPLv2 licensed, or otherwise GPL-compatible
- open mailing lists, development lists, tracker, wiki
- open-source development model

Challenges for ns-3

- ns-3 needs participation from the research community
  1) improving simulation credibility
  2) contributed and supported models
  3) maintainers

1) improve simulation credibility

Problem: simulations, in general, often suffer from lack of credibility

ns-3 solutions:

1) we will host ns-3 code and scripts for published work (improve reproducibility)
2) flexible means to configure and record values (the ns-3 attribute system)
3) tutorials on how to do things right
4) support for ported code should make model validation easier and more credible

1) improve simulation credibility

ns-3 needs ways to certify models too

- capture level of community acceptance
- publication lists, cross-reference
- need to identify maintainers, or state the absence of a maintainer
- validation techniques

These approaches still need to be developed

Incumbent upon users, ultimately, to produce credible simulations-- we can only try to help

2) contributed code

- one of ns-2's strongest assets

	Existing core ns-2 capability	ns-2 contributed code
Applications	ping, vat, telnet, FTP, multicast FTP, HTTP, probabilistic and trace-driven traffic generators, webcache	NSWEB, Video traffic generator, MPEG generator, BonnTraffic, ProtoLib, AgentJ, SIP, NSIS, ns2voip, Agent/Plant
Transport layer	TCP (many variants), UDP, SCTP, XCP, TFRC, RAP, RTP Multicast: PGM, SRM, RLM, PLM	TCP PEP, SCPS-TP SNACK, TCP Pacing, DCCP, Simulation Cradle, TCP Westwood, SIMD, TCP-RH, MFTP, OTERS, TCP Eifel
Network layer	Unicast: IP, MobileIP, generic dist. vector and link state, IPinIP, source routing, Nixvector Multicast: SRM, generic centralized MANET: AODV, DSR, DSDV, TORA, IMEP	AODV+, AODV-UU, AOMDV, ns-click, ZRP, IS-IS, CDS, Dynamic Linkstate, DYMO, OLSR, ATM, AntNet, Mobile IPv6, IP micro-mobility, MobileIP, GPSR, RSVP, PGM, PLM, SSM, PUMA, ActiveNetworks
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	802.16, 802.11e HCCA, 802.11e EDCA, 802.11a multirate, UWB DCC-MAC, TDMA DAMA, EURANE, UMTS, GPRS, BlueTooth, 802.11 PCF,, 802.11 PSM, MPLS, WFQ schedulers, Bandwidth Broker, CSFQ, BLUE
Physical layer	TwoWay, Shadowing, OmniAntennas, EnergyModel, Satellite Repeater	ET/SNRT/BER-based Phy, IR-UWB
Support	Random number generators, tracing, monitors, mathematical support, test suite, animation (nam), error models	Emulation, CANU mobility, BonnMotion mobility, SGB Topology Generators, NSG2, simd, ns2measure, ns-2/akaroa-2, yavista, tracegraph, huginn, multistate error model, RPI graphing package, jTrana, GEA,

http://www.nsr

Contributed code (cont.)

- we want to replicate this success with ns-3
- we will host your code/scripts in a number of possible ways
  - Contribute your code to the ns-3 main tree
  - Contribute code to a src/contrib directory
    - policy is to accept all maintained code
  - Contribute unmaintained code or user scripts to our repository
  - Contributed Code page (wiki)

3) maintainers

- we need active maintainers for key models and parts of the system
  - respond to user questions and bug reports
  - help with testing and validation

Current ns-3 maintainers:

Module	Person	Email
src/simulator	Mathieu Lacage	mathieu.lacage@sophia.inria.fr
src/core	Mathieu Lacage	mathieu.lacage@sophia.inria.fr
(random number generator in src/core)	Michele Weigle	mweigle@cs.odu.edu
src/common	Mathieu Lacage	mathieu.lacage@sophia.inria.fr
src/node	Craig Dowell	craigdo@ee.washington.edu
src/devices/csma	Craig Dowell	craigdo@ee.washington.edu
src/devices/point-to-point	Craig Dowell	craigdo@ee.washington.edu
src/devices/wifi	Mathieu Lacage	mathieu.lacage@sophia.inria.fr
src/devices/bridge	Gustavo Carneiro	gjc@inescporto.pt
src/mobility	Mathieu Lacage	mathieu.lacage@sophia.inria.fr
src/helper	Craig Dowell	craigdo@ee.washington.edu
src/internet-stack (IPv4, TCP)	George Riley	riley@ece.gatech.edu
src/applications	George Riley	riley@ece.gatech.edu
src/routing/global-routing	Craig Dowell	craigdo@ee.washington.edu
src/routing/olsr	Gustavo Carneiro	gjc@inescporto.pt
src/bindings	Gustavo Carneiro	gjc@inescporto.pt
regression tests	Craig Dowell	craigdo@ee.washington.edu
src/doc/tutorial	Craig Dowell	craigdo@ee.washington.edu
waf build system	Gustavo Carneiro	gjc@inescporto.pt
wiki, ns-developers list, web site content	Tom Henderson	tomh@tomh.org
code and web server sysadmin issues	Josh Pelkey	jpelkey@gatech.edu

- if you are a domain expert on a particular model, and a user of ns-3, consider to help with model maintenance

ns-3 project financial support

- U.S. National Science Foundation
  - CNS-0551686, CNS-0551378, CNS-0551706, CNS-0924385, CNS-0958139, CNS-0958142, and CNS-0958015
- Support from the French government (INRIA) via Planete research team (Walid Dabbous)
- Google Summer of Code (2008-10)
- Georgia Institute of Technology
- University of Washington
- Wireless Networks research group at INESC Porto, University of Porto
- U.S. Naval Research Laboratory

Summary

ns-3 is an active open-source project

- several simulator features designed to aid current Internet research
- community-based development and maintenance model

ns-3 needs you!

Resources

Web site:

http://www.nsnam.org

Mailing list:

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

IRC: #ns-3 at freenode.net

Tutorial:

http://www.nsnam.org/docs/tutorial/tutorial.html

Code server:

http://code.nsnam.org

Wiki:

http://www.nsnam.org/wiki/index.php/Main_Page