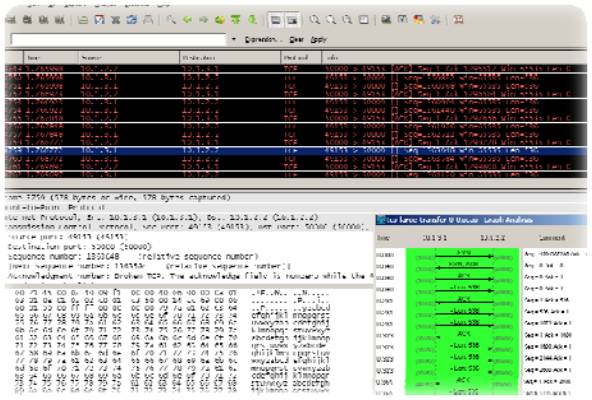
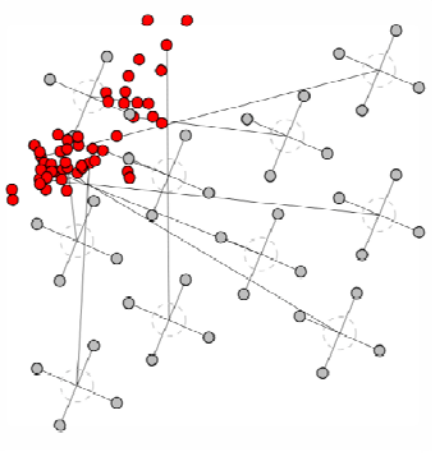
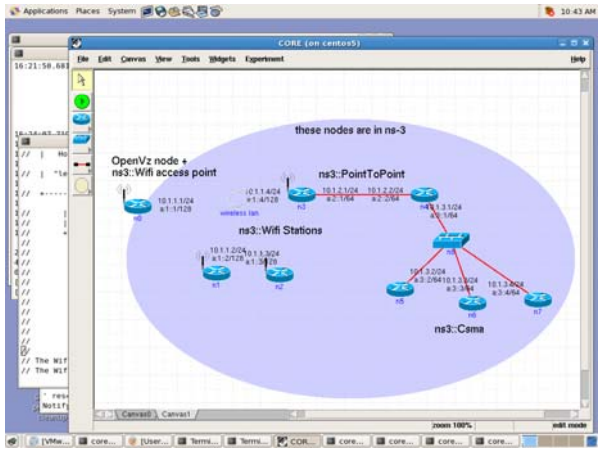
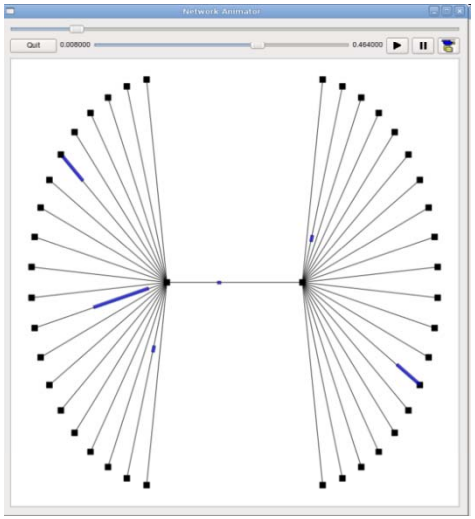
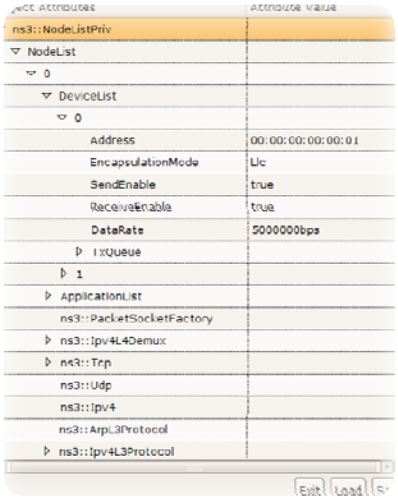
ns-3 project update











Tom Henderson

March 25, 2011

Talk outline

- ns-3 project goals
- progress since last WNS3
- new and forthcoming features
- developer meeting, and loose ends
- Q&A, feedback

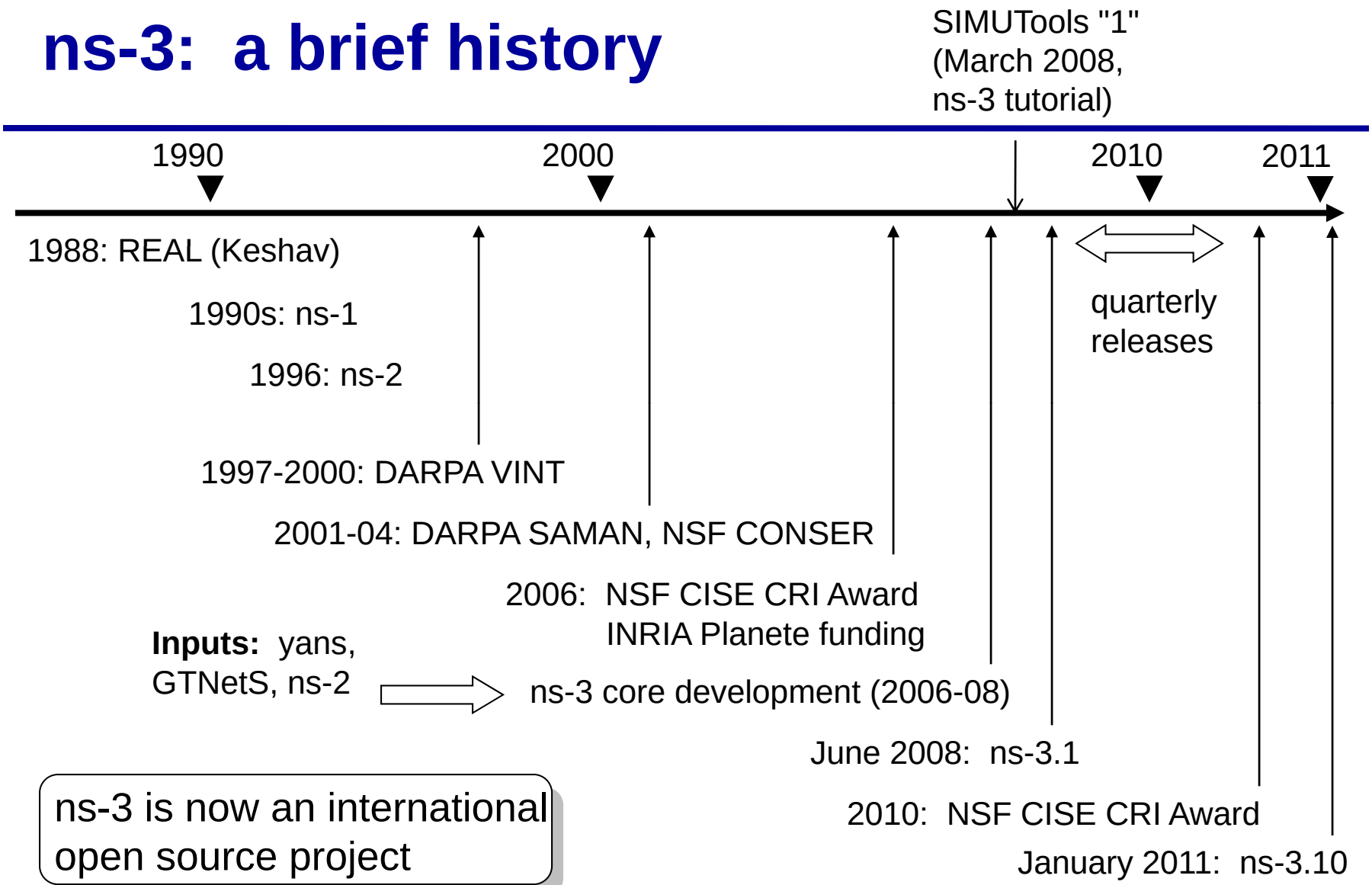
What is ns-3?

ns-3 is a free, open source software project building and maintaining a discrete-event network simulator for research and education

Technical goals:

- Build and maintain a simulation core aligned with the needs of the research community
- Help to improve the technical rigor of network simulation practice

ns-3: a brief history



ns-3 is now an international open source project

ns-3 themes

- Research and education focus
  - Build and maintain simulation core, integrate models developed by other researchers
  - Support research-driven workflows
- Open source development model
  - Research community maintains the models
- Leverage available tools and models
  - Write programs to work together
- Enforce core coding/testing standards

ns-3 software overview

- ns-3 is written in C++, with bindings available for Python
  - simulation programs are C++ executables or Python programs
  - Python is often a glue language, in practice
- ns-3 is a GNU GPLv2-licensed project
- ns-3 lacks an integrated development/visualization environment (IDE)
- ns-3 is not backwards-compatible with ns-2

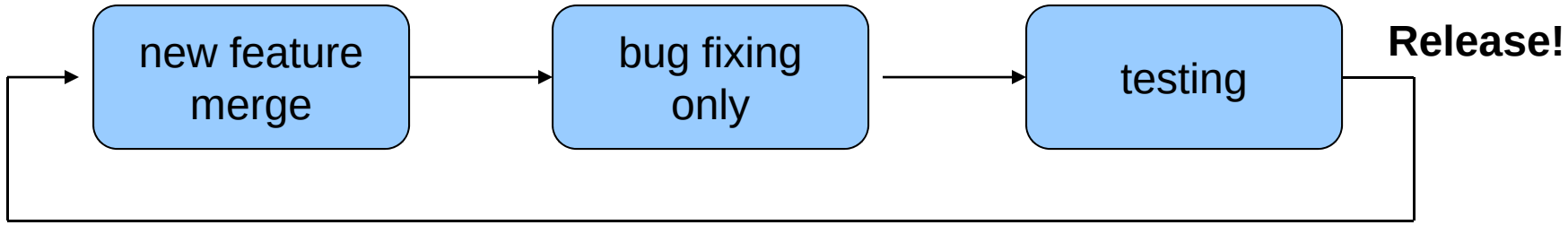
ns-3 development process

ns-3 is run as an open source project backed by research funding

- GPLv2 licensing stance
- open mailing lists
- uses standard tools (Mercurial, Bugzilla, Mediawiki, GNU/Linux development)
- ~20 maintainers worldwide

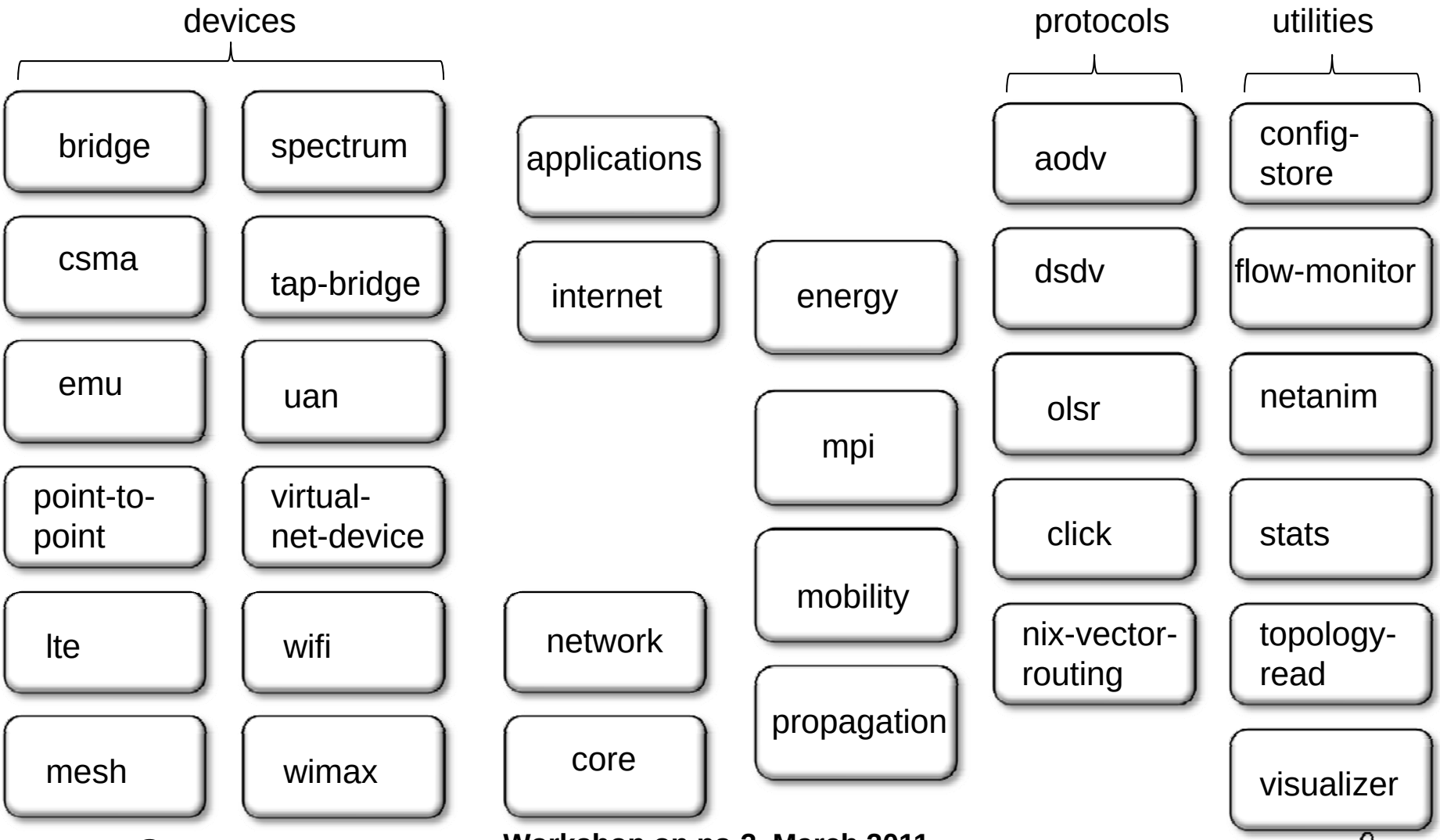
ns-3 development process

- date-driven quarterly releases



- All code for merge to ns-3 is openly reviewed by maintainers
  - Syntactic (style) reviews
  - Design reviews
  - Documentation and tests

Available modules (ns-3.11 May 2011)



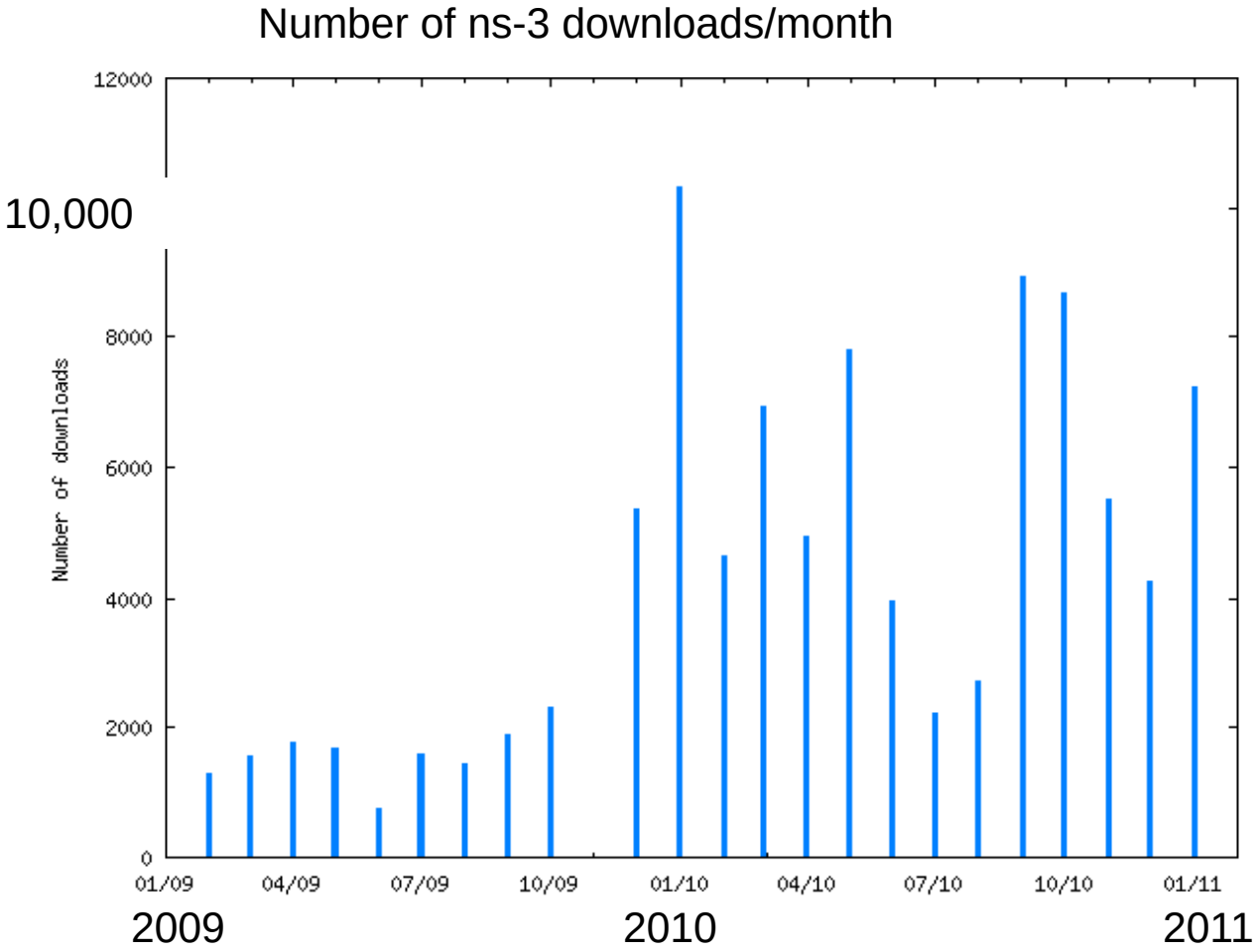
Current and recent ns-3 maintainers

- Kirill Andreev
- Nicola Baldo
- Elena Buchatskaya
- Pavel Boyko
- Gustavo Carneiro
- Craig Dowell
- Joe Kopena
- Flavio Kubota
- Tom Goff
- Tom Henderson
- Blake Hurd
- Mathieu Lacage
- Hemanth Narra
- Tommaso Pecorella
- Josh Pelkey
- George Riley
- Lalith Suresh
- Adrian Tam
- Leonard Tracy
- Sebastien Vincent
- Mitch Watrous
- Florian Westphal
- Michele Weigle
- Tony Wu

Analytics

Mailing list subscriptions:
- ns-3-users: 963
- ns-developers: 1176

Downloads:
- 6000/month in 2010



Talk outline

- ns-3 project goals
- progress since last WNS3
- new and forthcoming features
- developer meeting, and loose ends
- Q&A, feedback

ns-3.8 (April 2010)

- **WiMAX Net Device:** Simulate IEEE 802.16 point to multi-point based networks
- **Distributed simulation:** Support for point-to-point-based simulations for decomposition to multiple machines using Message Passing Interface (MPI)
- **Topology reader:** Allows quick and easy creation of large topologies by reading Inet or Orbis files.
- **Two-ray ground propagation loss model:** Calculates the crossover distance under which the Friis model is used.
- **Tracing support:** Unified tracing across all NetDevice models.

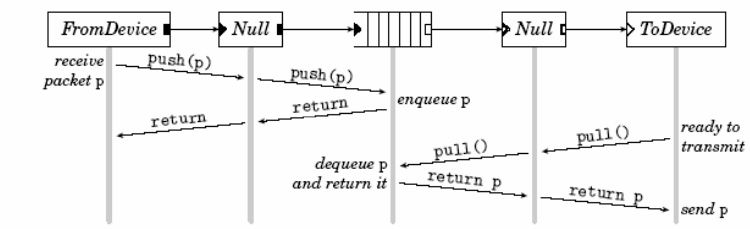
ns-3.9 (August 2010)

- **Spectrum Framework:** Allow modeling of spectrum-aware devices, and allow cross-system interference
- **OFDM error rate models for WiFi:** 802.11a/g
- **ns-2 mobility trace reader:** For BonnMotion, SUMO, TraNS, etc.
- **Energy model framework:** Energy sources and devices, and initial models
- **Rocketfuel topology dataset support**
- **Underwater Acoustic Network (UAN) models**
- **WiFi mode/rate reorganization**

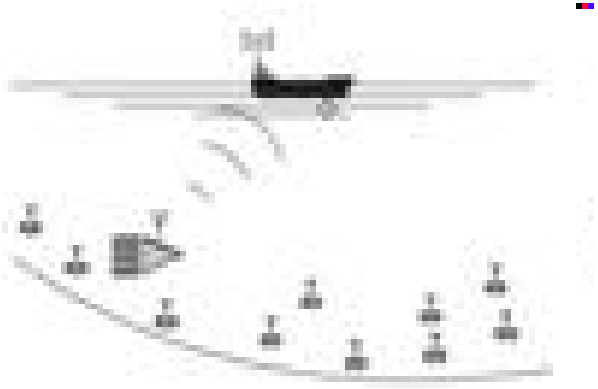
ns-3.10 (January 2011)

- **Pyviz:** Python-based visualizer with interactive python console (runtime introspection)
- **3GPP LTE:** Initial support for 3GPP LTE devices
- **new TCP:** Modular TCP congestion control
- **Energy model extensions:** R-V battery model, support for WiFi
- **Destination Sequenced Distance Vector (DSDV) routing**
- **Documentation moved to Sphinx format**

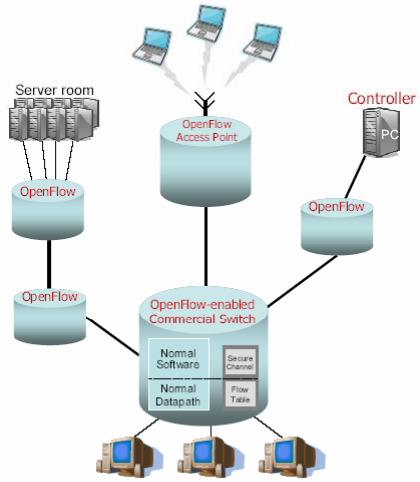
ns-3 and Google Summer of Code



Click Modular Router
Lalith Suresh



Underwater Acoustic Networking
Andrea Sacco



ns-3 OpenFlow
Blake Hurd

ns-3 LTE
Giuseppe Piro

Additional third-party project releases

- **PhySim:** high-fidelity physical layer model for 5 GHz OFDM
  - http://dsn.tm.uni-karlsruhe.de/english/ns3-physim.php
- **SliceTime:** time synchronization between Xen PVMs and ns-3
  - http://www.comsys.rwth-aachen.de/research/projects/slicetime/
- **Multipath TCP:** IETF TCP extensions
  - http://code.google.com/p/mptcp-ns3/
- **LENA project:** LTE/EPC network simulator
  - http://www.ubiquisys.com/femtocell-media-press-releases-id-203.htm
- **ns-3 DCE:** Direct Code Execution environment
  - http://www-sop.inria.fr/members/Mathieu.Lacage/dce.html

Additional third-party project releases

- **LTE:** Univ. of Padua, HARQ and MOBILITY modules
  - https://sourceforge.net/projects/ns3-lte/
- **NEMO/Mobile IP:** based on ns-3 direct code execution, zebra routing, and umip mobile IP
  - http://mailman.isi.edu/pipermail/ns-developers/2010-November/008455.html
- **Content Addressable Networks DHT**
  - http://mailman.isi.edu/pipermail/ns-developers/2010-November/008460.html

Course usage

Three courses taught using ns-3

- Georgia Tech (George Riley)
  - ECE 6110; http://users.ece.gatech.edu/~riley/ece6110/
- University of Kansas (James Sterbenz)
  - EECS 882; http://www.ittc.ku.edu/~jpgs/courses/mwnets/
- University of Pennsylvania (Boon Thau Loo)
  - CIS553; http://netdb.cis.upenn.edu/cis553projects/

Talk outline

- ns-3 project goals
- progress since last WNS3
- new and forthcoming features
- developer meeting, and loose ends
- Q&A, feedback

ns-3.11 (May 2011)

- modular build system
- Click Modular Routing
- OpenFlow switch support

Modular build goals

- Phase 1: make existing ns-3 modular
  - remove circular build dependencies between modules
  - consistent module layout
  - ability to disable modules from build
  - decouple tests from models
- Phase 2: "app store" model
  - modules may be maintained elsewhere
  - support multiple build systems

Consistent module layout

- Flattened module layout under `src/`
- (Mostly) uniform layout

```
bindings/
doc/
examples/
helper/
model/
test/
wscript
```

Disable modules from build

- Default: all modules
- ./waf configure --enable-modules=a,b,c
  - implicit dependencies are picked up
  - not persistent across configurations
- .ns3rc file in src/utils
  - top-level ns-3 directory
  - home directory ~/

Modular build, user experience

- Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

ns-3.10	ns-3-dev (full)	wifi + internet
• default build		
• `./waf configure && time ./waf build`		
• 119 seconds	• 95 seconds	• 12 seconds
• touch model file		
• `time ./waf build`		
• 72 seconds	• 43 seconds	• 2.5 seconds
• touch test suite		
• `time ./test.py -s object-name-service`		
• 74 seconds	• 3 seconds	• 1.2 seconds

Python bindings

- There is one separate Python extension module for each ns-3 module
- Scanning API definitions (apidefs) is done on a per ns- module basis
- Each module's apidefs files are stored in a 'bindings' subdirectory of the module directory
- Monolithic (ns3 namespace):

```
from ns3 import Node
```

- modular (ns namespace):

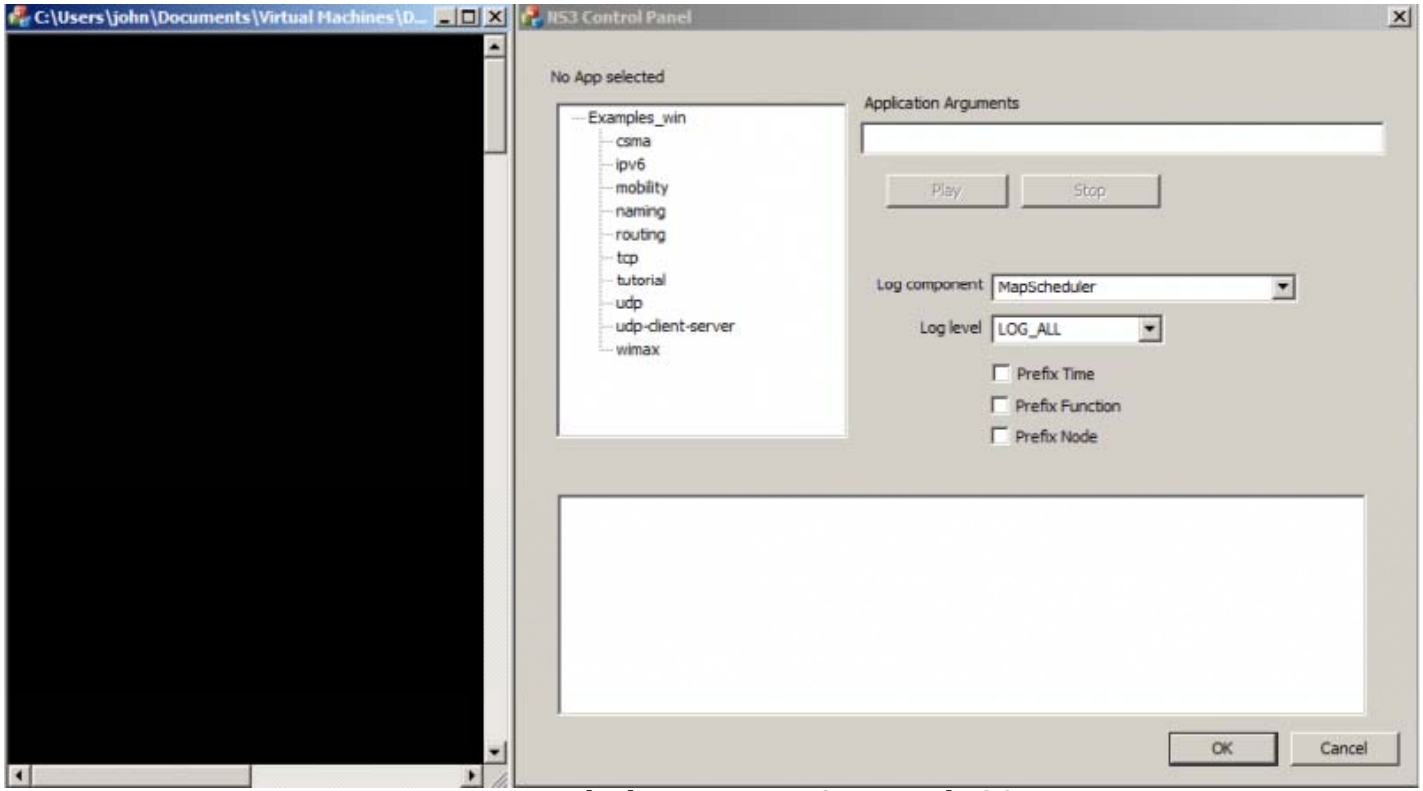
```
from ns.network import Node
```

Open issues

- What to do about certain examples and samples?
- Should we clean up coding style during this release cycle?
- What should the default module list be for ns-3.11 (if not all_modules)?

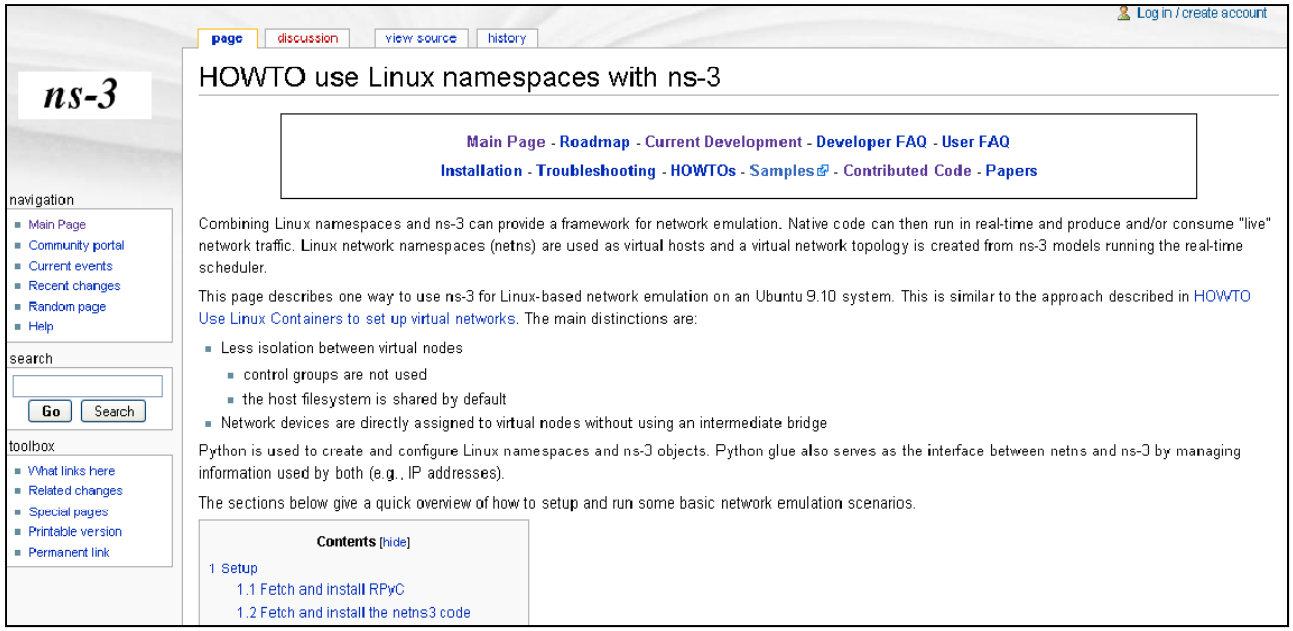
ns-3.10 on Windows (experimental)

- Use MS Visual Studio 2010
- Provide execution GUI
- Developed by John Abraham (Georgia Tech.)



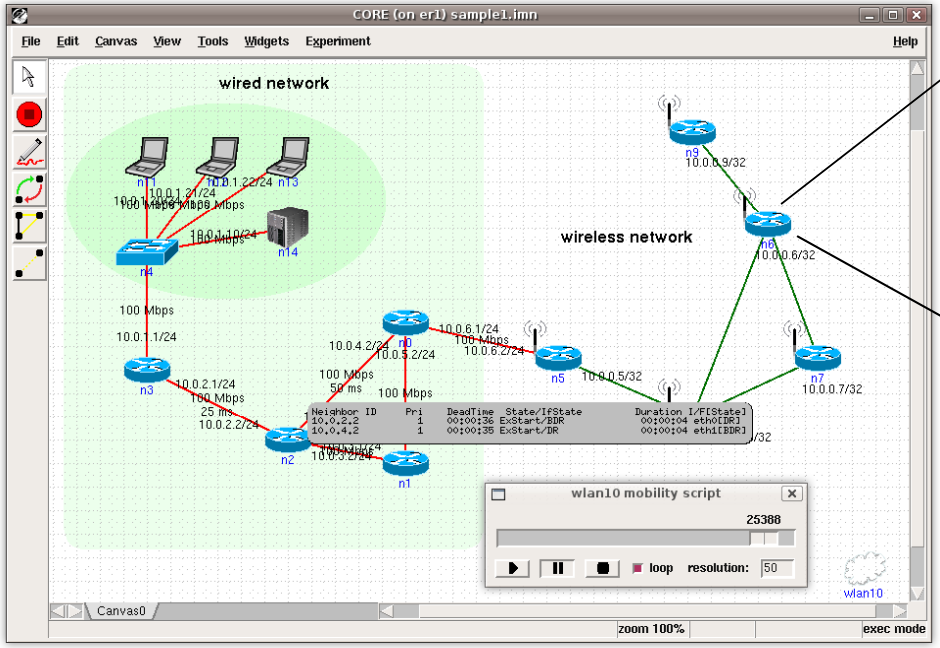
netns3 (experimental)

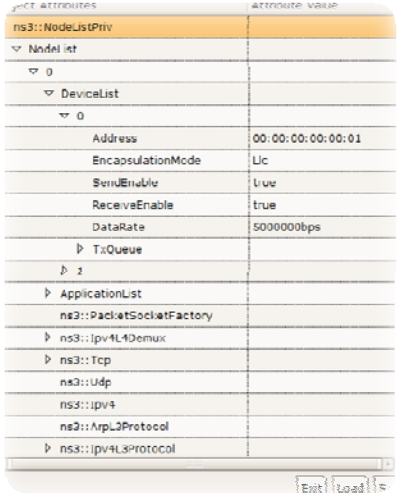
- Written by Tom Goff (Boeing)
  - Documentation and prototype posted on wiki
- Basic Python-based framework using ns-3 Python bindings, RPyC distributed computing library, and ns-3 tap bridge framework



Integrating ns-3 and CORE emulator

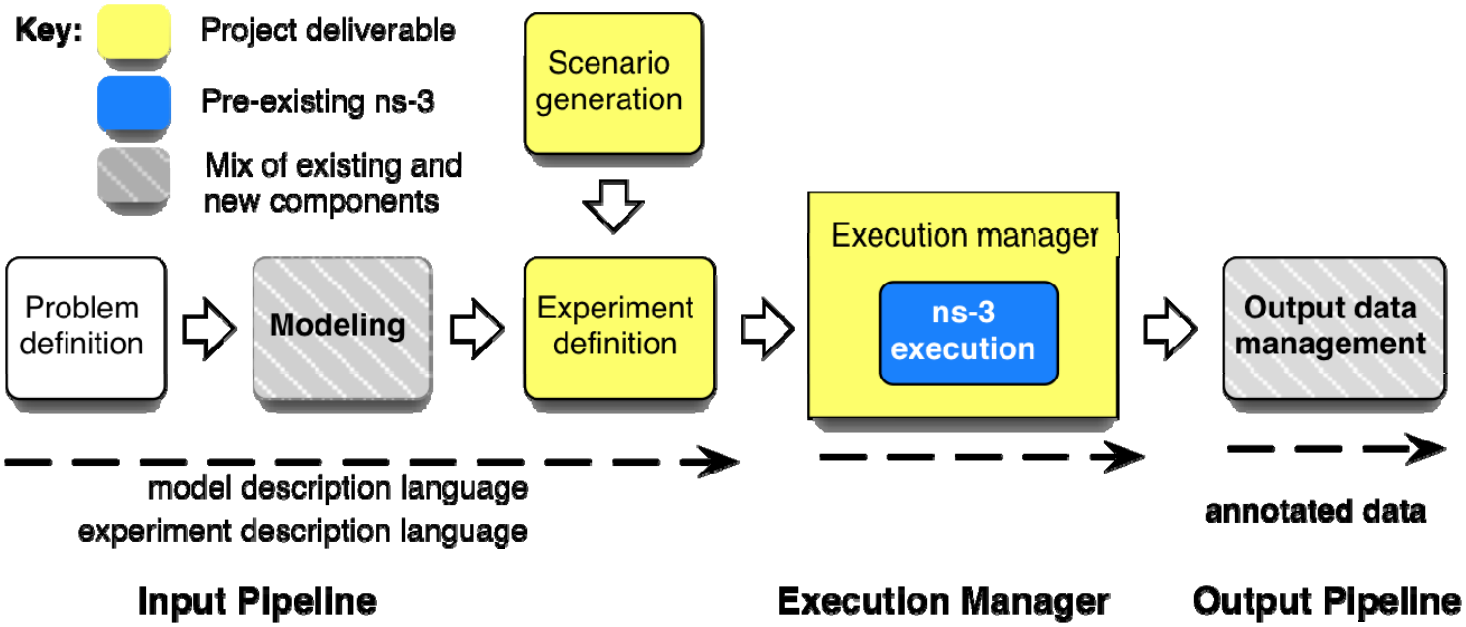
- Prototyped by Craig Dowell in 2010
  - ad hoc WiFi ns-3 with quagga routing in Linux containers





Frameworks for ns-3

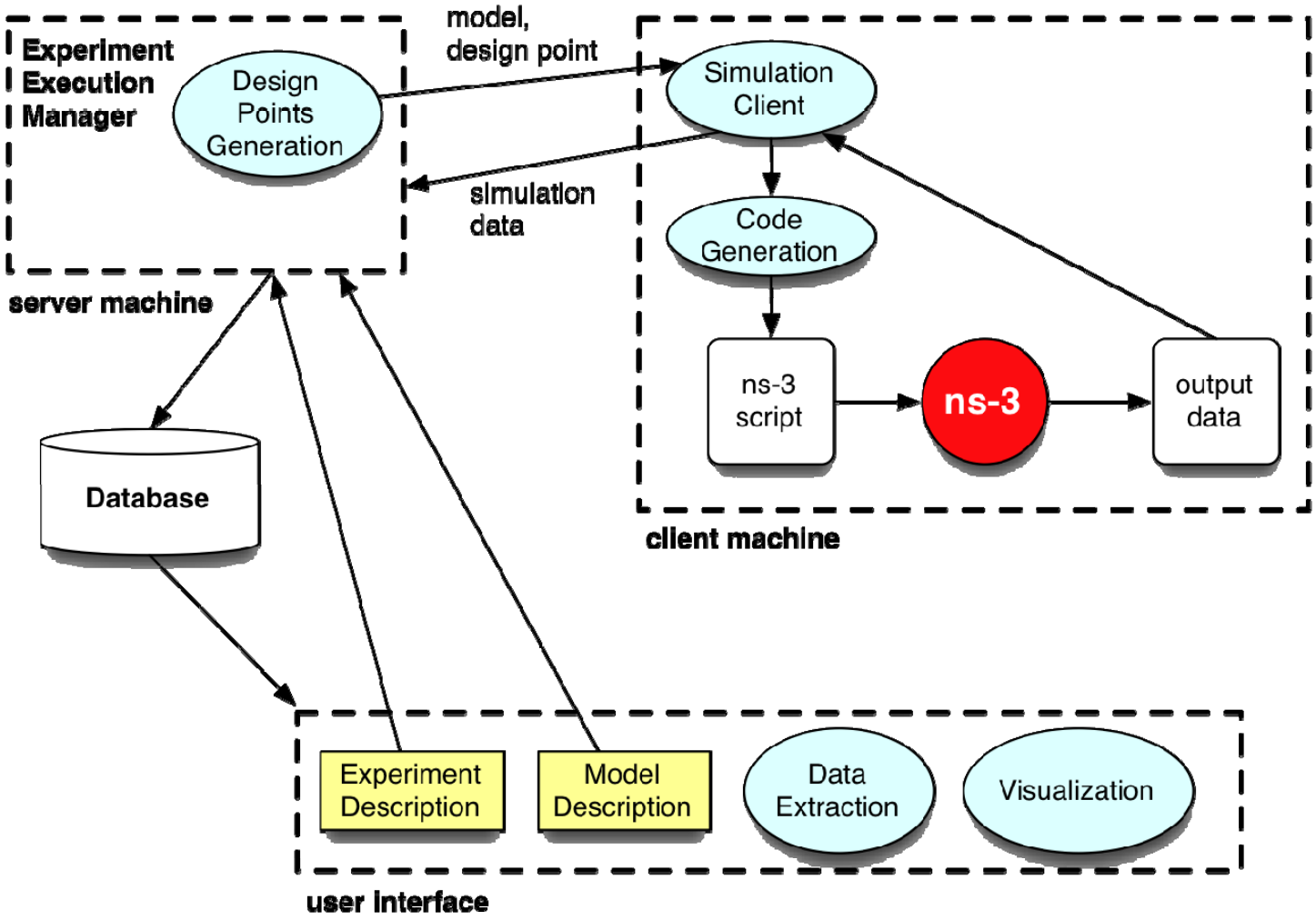
- NSF CISE Community Research Infrastructure
  - University of Washington (Tom Henderson), Georgia Tech (George Riley), Bucknell Univ. (Felipe Perrone)
  - Project timeline: 2010-14



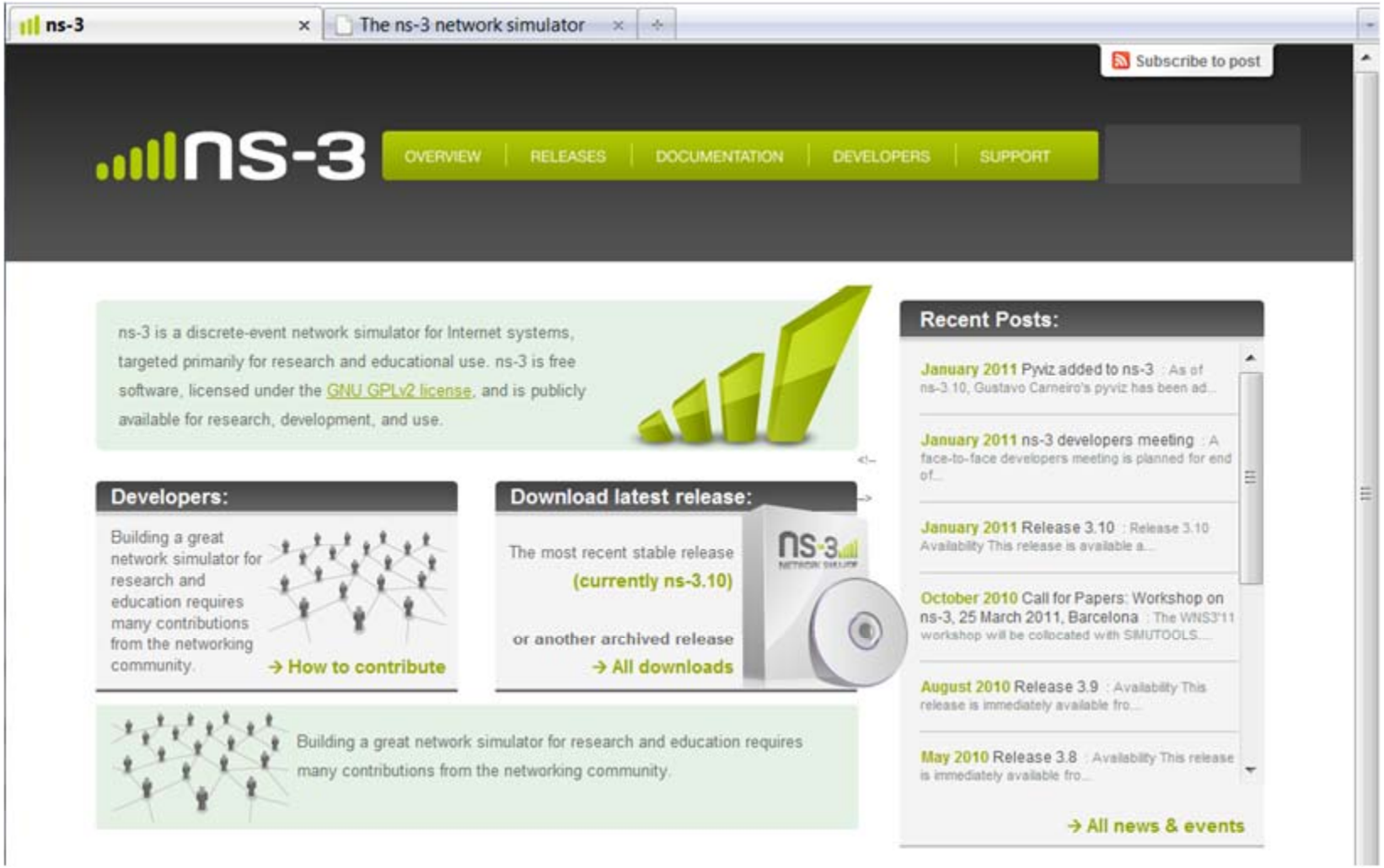
Status

- **Data Collection Framework** (Felipe Perrone-- short talk later)
- **High-level model description**; XML schemas and parsers (Andrew Hallagan, Bucknell)
- **Experiment Execution Manager**; experiment description, execution across clusters, transient and run-length detection plugins, managing combinatorial design points, web interface (Bryan Ward, Bucknell)
- **BRITE** (Boston Univ. topology generator) (George Riley and Josh Pelkey)

General architecture



New Wordpress-based site preview



Other announced projects

- Wireless jamming model
- MPLS
- VANET mobility model
- TDMA
- TCP Vegas
- DSR routing
- SimpleWireless model
- 6LOWPAN
- Chord/DHash DHT
- 802.11n
- TMix and DelayBox
- multi-core parallelization

Talk outline

- ns-3 project goals
- progress since last WNS3
- new and forthcoming features
- developer meeting, and loose ends
- Q&A, feedback

Meeting goals

- review/feedback on data collection frameworks
- review/feedback on modular builds
- revisit open technical issues
  - processing delays
  - starting and stopping of objects
- review new website and logo
- documentation review
- usability review
- mentored projects, unofficial SoC

Mentoring

- ns-3 *not* selected to GSoC 2011, ending a 3-year run
  - 175/417 accepted, 50 new organizations
- Discussing options for unofficial SoC and mentoring projects at developers meeting

Talk outline

- ns-3 project goals
- progress since last WNS3
- new and forthcoming features
- developer meeting, and loose ends
- Q&A, feedback

Acknowledgment of support