

ns-3 developers meeting, March 6-8, 2013

The meeting was hosted by the Inria Diana team at Inria's Sophia Antipolis Research Center.

Attendees

George Riley
Daniel Camara
Tommaso Pecorella
Vedran Miletic
Hajime Tazaki
Giovanni Rigazzi
Alina Quereilhac
Nicola Baldo
Manuel Ricardo
Brian Swenson
Takai Eddine
Tom Henderson
Damian Vicino

Future workshops on ns-3

The meeting discussed possible alternative venues, possibly North America, Asia. Some views expressed:

- ns-3 is open source, should be able to publish openly (ACM may now allow this)
- would be nice for wns3 presentation to be recorded (possibly streaming), put on the web
- is there a benefit colocating workshop with other conference? not clear (different views expressed, there is a tradeoff in cost vs. travel justification)
- remote access for developers meeting would be nice
- affiliation with digital libraries (hard requirement? DOI, or specific organization?), organization is important
- avoid summer, spring is better alternative

Actions:

- look into alternative between standalone ACM publish vs. ACM-affiliated workshop (co-located) (George)
- look into North American venues for 2014 (either Atlanta standalone or associated with some conference) considering May in Atlanta standalone
- open access alternatives (Tommaso, Vedran); need to compare with ACM alternative

ns-3 Consortium

The meeting discussed the consortium event from Monday and, in general, how to make it succeed, and the kinds of events needed.

- unclear why an academic should join the consortium? Prestige? (ns-3 is not large enough for this now)? We discussed one possible incentive being the promotion of members' funded projects; to make European/NSF projects more connected
- Could consortium join an FP7 project? directly, probably not.
- The consortium could potentially join the Google Summer of Code; there is an issue taking mentor payments, however
- what is benefit to industry? Need to work on these kind of benefits
- another expressed goal: increase user and contributor base

ns-3 Windows

We discussed possibly making a native Windows version one of the products of Consortium work, and trying to raise money from industry to maintain Windows code.

- could post next version release for Windows, and gauge interest (# of downloads)
 - o announce that this is `_unsupported_` officially, and point people to the page where the consortium information is provided
- could try to segregate costs of maintenance and ask for a consortium donor to support it
- discussion of waf vs Visual Studio; waf requires python (unless distributed as binary), also visual studio offers debugger
- Discussion veered to the various packages for making ns-3 available in general; Alina recommended to look at OpenSuse obs (used to compile ns-3 for PlanetLab) to create packages for different architectures; need to write all the files to create packages
- will take some time (many months) to get feedback on this

Action:

- ask John Abraham to update Windows version and create ns-3.16 or 3.17 for Windows Visual Studio; also post the current patch to codereview
- create separate web page to gauge interest and solicit contributions

ns-3 educational

What models should we focus on for educational use?

- models or animations?
- port fair queuing to ns-3
- CoDel (were copyrights an issue?)
- promote more sharing of scripts or lab assignments through wiki
- need a maintained "zip" archive of videos and useful materials

- provide per-language course materials
- Tommaso suggests creative commons copyright/license, and also ns-3 consortium
- Coursera? a lot of work, but broad visibility. Could be Consortium activity to develop this?
- look at the "Moodle" platform
- YouTube and iTunes Univ. dissemination (need to check copyright status)
- Cisco uses canvas (cloud based); Coursera-like service
- (aside: in the tutorial, the fifth.cc connect to socket code is too complicated, and should be rewritten)

Action:

- Tom will ask again the status on CoDel; need to clarify copyright.
- Tommaso would like to seed this with 6LowPAN models
- Add separate tab on wiki for videos and multimedia (Tom)
- ns-3 channel on YouTube (? consortium)
- look into iTunes U. (Tommaso)
- provide helper for fifth.cc patch (Vedran)

ns-3 sensor development, status

Discussion about how to progress the 802.15.4 code and related models.

- Tom described state and history of lr-wpan code
- Tommaso described 6LOWpan status, advocated for getting this module on the path to merge
- Manuel, working on 802.15.4 and 4G, will be working with Contiki, and integration of 6LowPAN
- Takai and Daniel are supporting Contiki, porting in the platform
- Manuel suggested 802.15.4g (larger frames, OFDM)
- Tommaso described his 6LOWpan status, including UDP issue (possible refactoring needed) to enable header compression, ndisc cache accessibility (bug on IP ARP/ndisc accessibility)

Action:

- Tom to send Manuel the current lr-wpan documentation
- Tom and Tommaso to look at merging Stephane's LR-WPAN MAC code, review documentation, try to get that, 6LoWPAN (need routing solution) to a code review (ns-3.18 August)

ns-3 Data Collection Review

In general, people expressed desire to get this capability into the codebase.

Issue 1) Collector/Probe interface is constrained to type double

People agreed that this is suboptimal for the long term, and the consensus was that we ought to extend the TypeId system to provide type information that could be queried by a helper to make the right type selection. Regarding the difficulty of dynamically changing the function pointer in the callback (to the one corresponding to the right type), Vedran suggested making this function pointer instead a pointer to a function pointer, and changing the function pointed to (e.g. the TraceSink) to the right type based on user input.

Alina has some other reasons she wants to extend the typeId system, so perhaps this can be worked together.

Issue 2) Should aggregators synchronously poll collectors, or should we preserve the asynchronous nature of the trace sources and deal with the synchronization complexity in the aggregators?

Nicola supported keeping the collectors publishing data and make the aggregators deal with synchronizing it. Daniel suggested that if we have to resort to polling the collectors to get current values, the polling operation should have no side effects (e.g. of resetting the counters).

Issue 3) 'data-collection' module dependency should be included in most modules above 'core'.

People seemed OK with this; some suggested just to make this part of 'core' itself.

Nicola walked through some of how the LTE code does data collection:

<http://www.nsnam.org/docs/release/3.16/models/html/lte-user.html#simulation-output>

basically, LTE code has complex trace sources that are unpacked and read into space-delimited columns into a file or a sqlite database.

Also, Nicola reminded us of the athstats helper he wrote for wifi:

http://www.nsnam.org/docs/release/3.16/doxygen/athstats-helper_8h_source.html

We should study how to make these traces (being written to file) fit into our framework for a file-based aggregator. (Tom, Felipe, Mitch)

ns-3 Start/Stop

Discussion of the latest work by Vedran on this topic.

1) Object::Start() is conflated between Initialization and logical "Start"; can we clean this up?

? Rename existing Start() to Initialize() and pair it with Dispose()

2) Object::Start()/Stop() should be implemented in a special base class e.g.

StoppableObject : public Object

for those objects that need this API (this compromise to appease those people who prefer not to put it into the Object base class)

? Vedran to try this approach

put it into 'src/core'; something like

```
class StoppableObject : public Object
{
public:
    static TypeId... (put trace source here);
    void Start () { DoStart ();}
    void Stop () { DoStop ();}
    bool IsStarted () const;

protected:
    virtual void DoStart () = 0;
    virtual void DoStop () { NS_FATAL_ERROR ("Implement"); }
private:
    bool m_started;
    TracedCallback ...;
}
```

do not, by default, start/stop aggregates

alternative would be to enforce via policy/documentations, but the belief is that this will be used enough to justify moving some of the implementation into a common base class (for interface consistency as well).

If the class gets more complicated than this (e.g. multi-state rather than boolean), or with lots of "listener" callbacks, alternative suggested is to move this out of the inheritance chain and into some kind of object that is pointed to.

If we want to support a more complicated start/stop/failure model and interface, such classes can aggregate some other special failure handling object.

DCE discussion

Hajime reviewed the current status of DCE, aiming for a release schedule aligned with ns-3. dce-1.0 will be requested for review on roughly March 15. Hajime answered several questions about how DCE presently works and is built.

Waf

Alina mentioned cleanup/documentation of src/wscript will help future maintenance, and documentation of how to extend waf for ns-3 would be helpful.

Actions:

- look at bindings generation failure, whether can be disabled by disabling new gcc-4.6 warnings (?)
- finish off --pyrun and --enable-sudo (perhaps Debian issue) for waf upgrade (Alina)
 - ? maybe ask Gustavo to look at --pyrun fix
 - ? patch replaces our custom pkgconfig generation for waf's built-in feature
 - ? Alex to document his boost fix
 - ? document and possibly refactor (or ask for refactoring) of our customizations (? no takers on this item)
 - ? fix create-modules script (different template wscript) (Vedran will fix)

bake discussion

We reviewed current status of bake with Daniel.

Vedran: regarding making more build scripts available, there is a makefile wrapper for waf on the wiki that might be useful to put into ns-3-dev.

Agreement on long-term direction of flattening ns-3 into component libraries; need to figure out how to get there in a least painful way.

For ns-3.17, we need to

- 1) preserve ./download.py ./build.py experience, and ns-3 waf operation as much as possible
 - ? minor changes to ns-3 waf. look for ../../build/liblinux.2.6.26.so instead of ../nsc/liblinux.2.6.26
- 2) DCE helper will also integrate with ns-3 waf; usable from ns-3 scratch directory?
 - ? long-term direction for "scratch/" directories is not settled; could have one bake-level scratch or multiple, or we could provide example wscripts and Makefiles for users to clone and modify

For ns-3.18, experiment with implications of moving a "large" module outside of ns-3-dev? e.g. wifi or lte, or wimax

Issues:

- ? how to control modularity of the build? Does .ns3rc and --enable-modules go away?
- ? version number for separate module?
- ? how to gracefully remove/migrate from ns-3-dev
- ? long term, do we manage multiple bakeconf.xml files?

Testing on (selected) virgin virtual machines for bake is necessary, in addition to bake building ns-3 on all of our buildslaves. FreeBSD, Linux, OS X

Target architectures (ns-3.17)?

We discussed what platforms to target for the next release.

targets with Jenkins buildslaves:

- Latest two Fedora, Ubuntu, 32/64 bit machines
- Ubuntu 10.04 LTS 64 bit
- FreeBSD 8.2, 9.1 64-bit
- Mac OS X 10.8 64-bit

targets without buildslaves (maintainer who will check release candidates):

- PowerPC (Linux) (Tom)
- Mac OS X 10.7 64-bit (Tom)
- Cygwin on Windows 7 (Tom)
- Debian 6 64-bit and Debian 7 64-bit (Vedran)
- CentOS?? (this is up to Vedran)
- Raspberry PI (Tommaso)
- Linux Mint (Tommaso)

Note: Latest Ubuntu has different libc version (2.15 or 2.16) than Debian (2.13), has some binary incompatibility issues.

Note: DCE tested with Ubuntu 10.04 to 12.04 (32 and 64), and recent Fedora (14-16)

SunOS bugs, mark low priority or patch needed, or wontfix if persists for > 1year

Testing

How can we improve the amount of test code that is submitted? How can we get more performance tests going?

Q. how to get access to test infrastructure before merging?

A. contact Tom to get access; Tom needs to write documentation on how to do this

performance testing

- need to create or find framework to look at performance regressions over time
- need to ask people what to measure

simulations: time to execute, memory consumed
real-time:

Mathieu created ns-3-benchmark, run many times, perhaps look at reviving this.
Alina suggested to run multiple times, look at averages instead of one shot runs.

Vedran: looks for order of magnitude performance issue How to calibrate? There is a Firefox example that looks for order of magnitude performance problems.

Tommaso, need to limit the effort here; we have 200 bugs to fix

For real-time, need to create test scripts

- Emu
- Tap
- EmuFdNetDevice

Alina-- need to do the testing seriously. Need to automate the analysis of pcap files.

Tom to take action to get something started... Possibly run UnixBench on a UW server, then write specially designed simulation scripts to scale in terms of size and number of packets. GlobalRouting vs. Nix. DCE vs. non-DCE, etc. Build on Alina's tests for real-time.

Nicola to upload the LTE scripts

Bug 1563 (valgrind and test granularity):

Need to consider whether "unit", "system", "performance" are enough testing categories.

Categories on test suites
Scopes on test cases (light, normal, full)

new switches on test.py

- c constrain
- ? scope
- g valgrind

Extend categorization to test case. You may have some test suites where you may want to run only a couple of test cases for short tests, and all for long tests.

(Tom/Mitch to work this)

Git discussion

Should the project move to git?

Benefits: Host our own repo; mirror on GitHub. Can increase # of forks, improve interaction with developers. Could do this when we split repository for bake (i.e. at the same time we modularize ns-3, we also move to git).

Concerns: loss of history, dealing with lots of forks, our workflow may suffer

Issues: do not lose history in some way

Actions:

Vedran to post ns-3.16 or ns-3.17 on github to test whether we get interesting results. Point out that we are not using git as a working repository, that we are experimenting with github.

User feedback

Tom reviewed some feedback from people (John Abraham, Konstantinos) who are responding to a lot of ns-3-users mail. This discussion led into other suggestions for improvement by the attendees.

- debugging-- all compiled libraries go to same build library; cannot use LD_LIBRARY_PATH
- better help to get IDE experience up and running would be welcomed
- tracing: hard to master, documentation may be buggy (available traces are somewhat 'hidden'), not available at Python API at callback level (in Python, cannot overload for scalar tracing), debugging a mismatch between function signature and callback signature, parameters of the trace source are not in the documentation

Action: need to audit and improve documentation

Progress counter would be nice to have: time report, % of simulation remaining (number of events)

GSOC 2013

Discussion about improving outcomes for Google Summer of Code.

- discussion about improving student selection to ensure higher probability of project success and higher probability to stay with the project. We will do more in the following two areas: 1) encourage more code submission before applications are due (submit patches), 2) set up projects in a way that are hard to fail to achieve code merge; do not try to get feature additions out of GSoC. Do not want to make code patches a hard requirement; we make the statement: "If you submit patches or interact with our maintainers, your probability of selection will be improved." Also warn: "If you interact well but then have a bad application, you will not be selected."

FdNetDevice review

Reviewed Alinas longstanding proposal to deprecate existing EmuNetDevice in favor of the FdNetDevice that provides Emu helper plus a new Tap and PlanetLab helper.

- does not presently affect TapBridge; need to document clearly difference between Tap and TapBridge
- EmuFdNetDeviceHelper -> FdNetDeviceEmuHelper ?

- may be useful for existing TapBridge to be deprecated in favor of new FdBridge with a Tap helper that makes it more general (not dependent on the Tap, but preserving across the module the behavior that it is bridged). TapBridge has lots of options to create the tap. This could be decoupled into helper, and do similar things to FdNetDevice. Since TapBridge already using FdReader, should not be so much work. file descriptor could also be changed during a simulation

Action: Tom to document better that create-module.py spits out an outline of desired documentation

Action: Nicola to ask Jose how he achieves backpressure in his emulation work

Time::SetResolution update

Peter Barnes called in to walk through his suggested revisions of the code to address the Time::SetResolution bug (bug 954). The meeting agreed with his current approach, and the next step is for Peter to prepare a new code review.