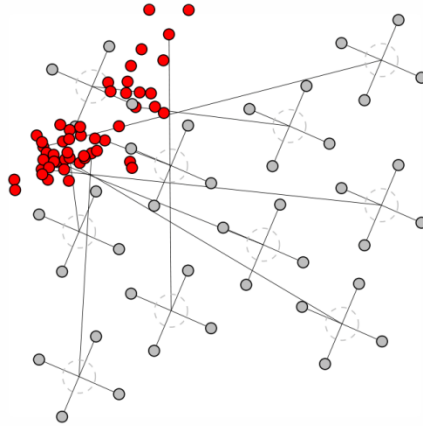
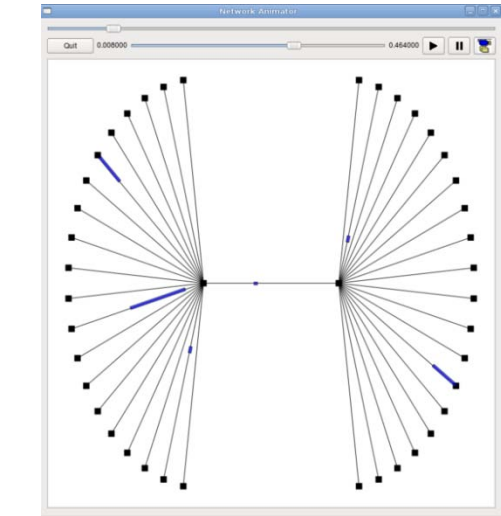


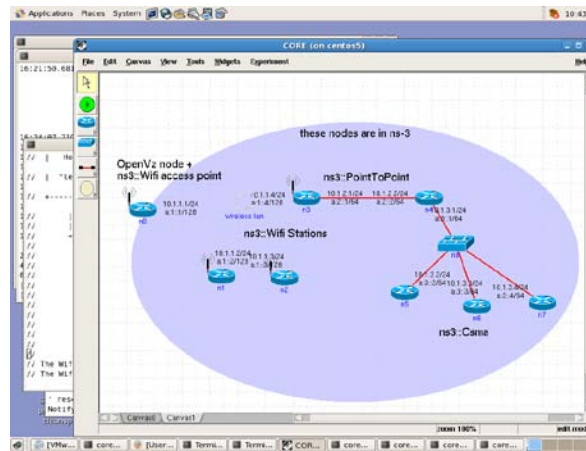
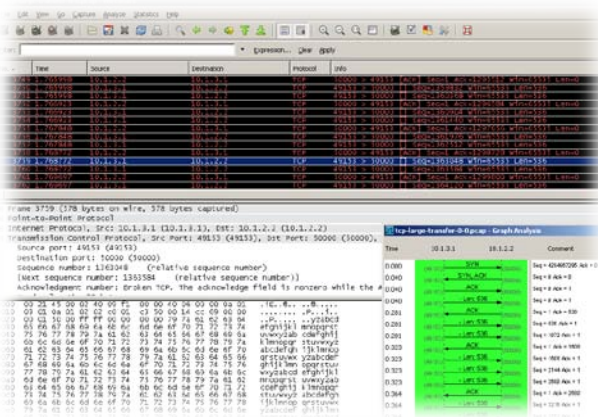
ns-3 Overview



```

1 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader (
2 destination=ff:ff:ff:ff:ff:ff) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (re
3 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) ns3::EthernetTrailer (fcs=0)
4 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader (
5 destination=ff:ff:ff:ff:ff:ff) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (re
6 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) ns3::EthernetTrailer (fcs=0)
7 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
8 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
9 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
10 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
11 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
12 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
13 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
14 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
15 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
16 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
17 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
18 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
19 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
20 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
21 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
22 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
23 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
24 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
25 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
26 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
27 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
28 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
29 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
30 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
31 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
32 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
33 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
34 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
35 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
36 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
37 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
38 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
39 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
40 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
41 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
42 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
43 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
44 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
45 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
46 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
47 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
48 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
49 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
50 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
51 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
52 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
53 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
54 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
55 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
56 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
57 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
58 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
59 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
60 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
61 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
62 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
63 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
64 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
65 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
66 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
67 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
68 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
69 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
70 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
71 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
72 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
73 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
74 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
75 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
76 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
77 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
78 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
79 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
80 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
81 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
82 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
83 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
84 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
85 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
86 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
87 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
88 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
89 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
90 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
91 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
92 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
93 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
94 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
95 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
96 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
97 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
98 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request
99 //NodeList/O/DeviceList/O/Sns3::CsmaNetDevice/Rx ns3::EthernetHeader (length
100 =1024) ns3::LlcSnapHeader (type 0x806) ns3::ArpHeader (request

```



Object Attributes	Attribute Value
ns3::NodeListPriv	
▼ NodeList	
▼ 0	
▼ DeviceList	
▼ 0	
Address	00:00:00:00:00:01
EncapsulationMode	Llc
SendEnable	true
ReceiveEnable	true
DataRate	5000000bps
▼ TxQueue	
▼ 1	
ApplicationList	
ns3::PacketSocketFactory	
▼ ns3::Ipv4L4Demux	
▼ ns3::Tcp	
ns3::Udp	
ns3::Ipv4	
ns3::ArpL3Protocol	
▼ ns3::Ipv4L3Protocol	

<http://www.nsnam.org>

ns-3 overview May 2011

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 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

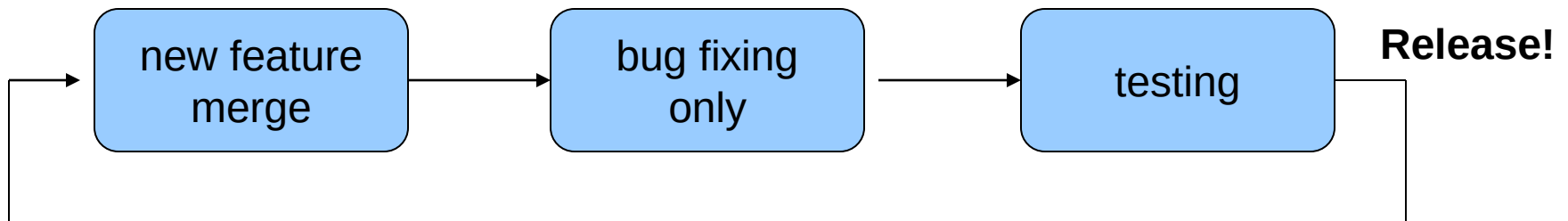
ns-3 development process

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

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

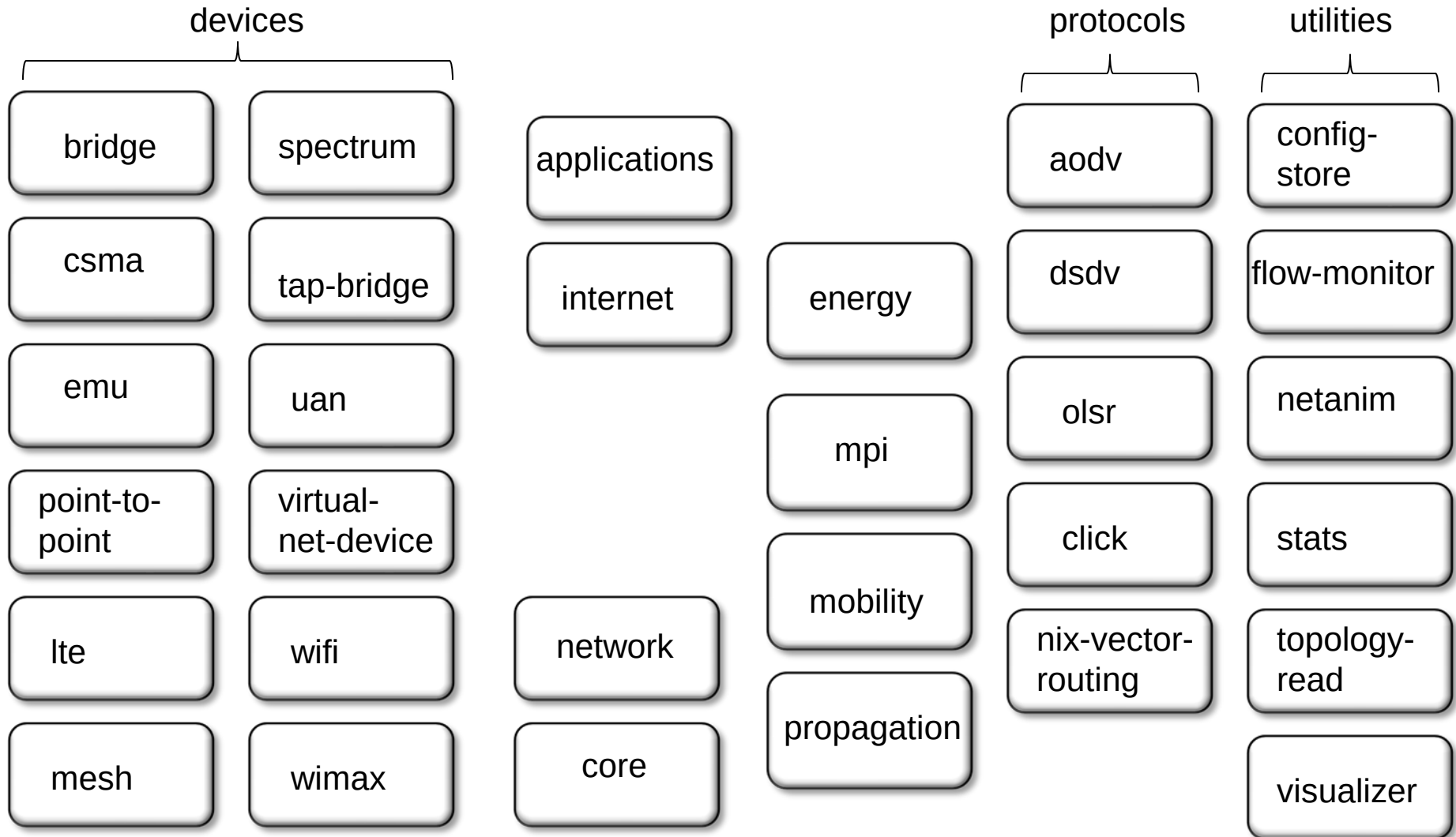
ns-3 development process

- 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)



Analytics

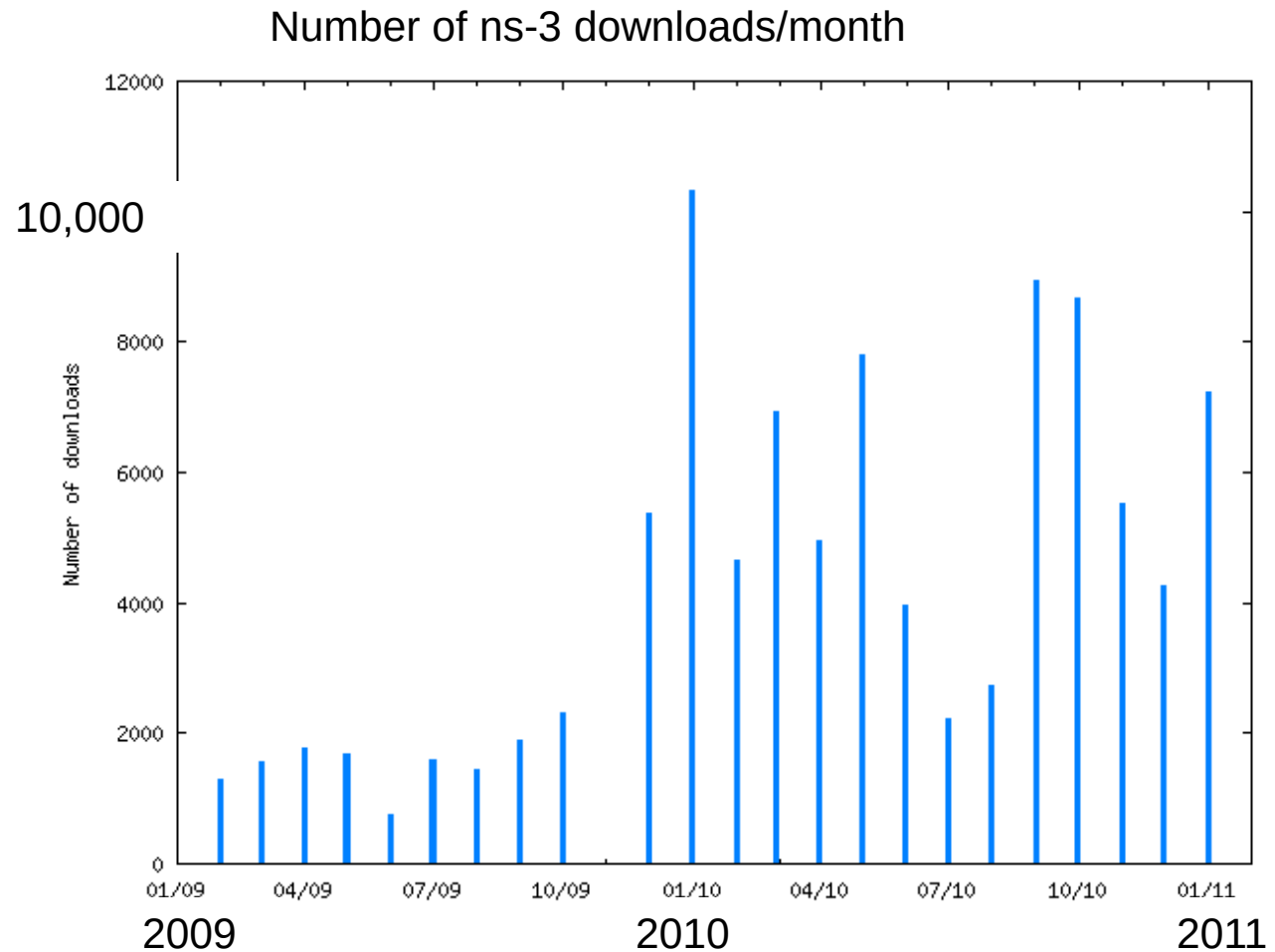
Mailing list

subscriptions:

- ns-3-users: 963
- ns-developers: 1176

Downloads:

- 6000/month in 2010



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):

The screenshot displays the NS-3 Doxygen documentation interface. On the left is a sidebar with a tree view of the documentation structure, including links to 'ns-3 Documentation', 'NS-3 Modules', 'NS-3 Class List', 'NS-3 Class Hierarchy', 'Class Members', 'NS-3 Graphical Class Hierarchy', 'NS-3 Namespace List', 'Namespace Members', and 'NS-3 Related Pages'. The main content area is titled 'ns3::InetSocketAddress Class Reference [Address]'. It includes a navigation bar with tabs for 'Main Page', 'Modules', 'Namespaces', 'Classes', and 'Related Pages', and sub-tabs for 'Class List', 'Class Hierarchy', and 'Class Members'. The text describes 'an Inet address class' and provides a link to 'More...'. It also shows the include directive '#include <inet-socket-address.h>' and a note about a collaboration diagram. The diagram shows 'ns3::Ipv4Address' as a base class for 'ns3::InetSocketAddress', with a member 'm_ipv4' indicated by a dashed arrow. A '[legend]' link is provided for the diagram. Below the diagram is a link to 'List of all members.' and a section header for 'Public Member Functions'.

– see also: <http://www.nsnam.org/documents.html>

Extensible software core (cont.)

- Project maintains software core and key models
 - research community expected to contribute additional 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

ns-3 solution: an object aggregation model

- objects can be aggregated to other objects at run-time
- a “query interface” is provided to ask whether a 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

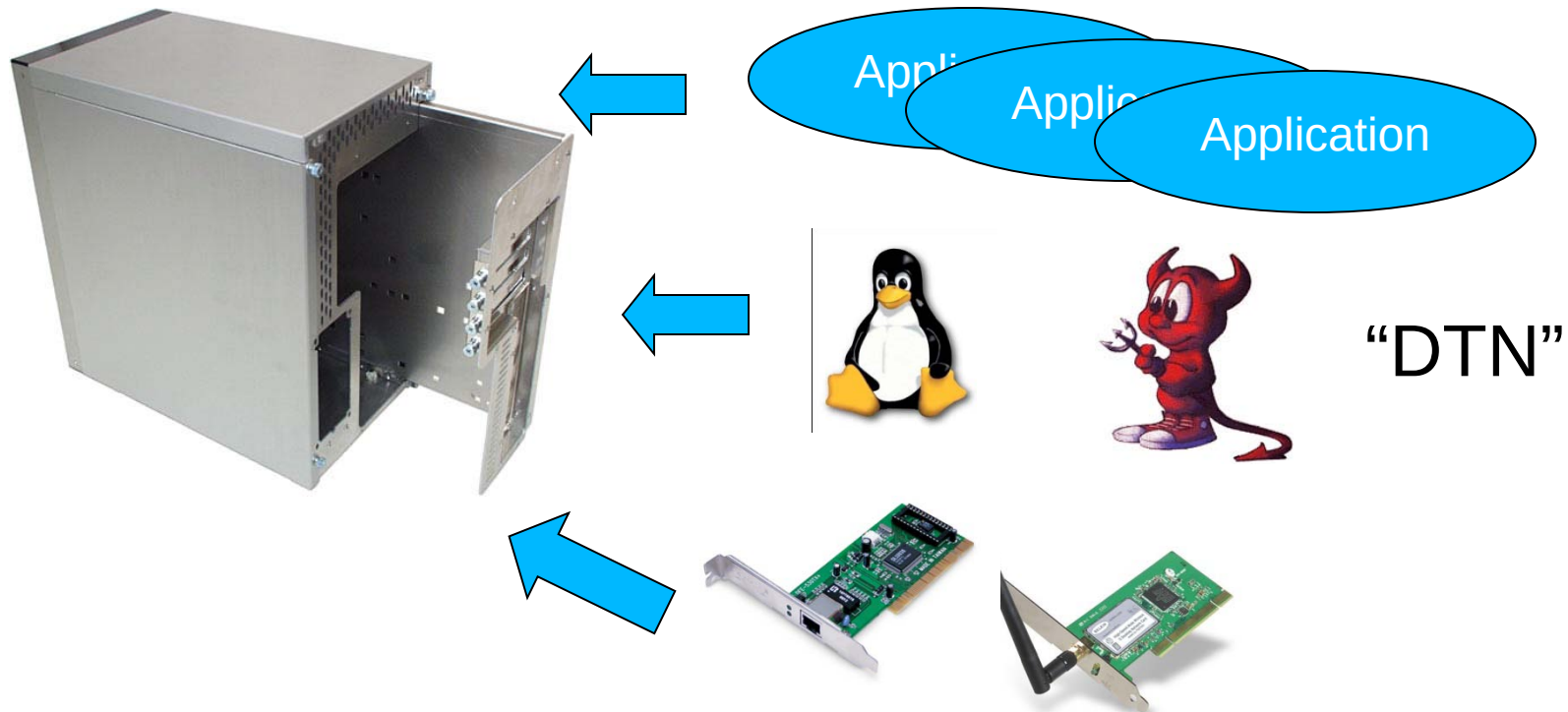
- If the simulator cannot be made to closely model a real system:
 - hard to compare results or validate the model
 - hard to reuse software between the two domains

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

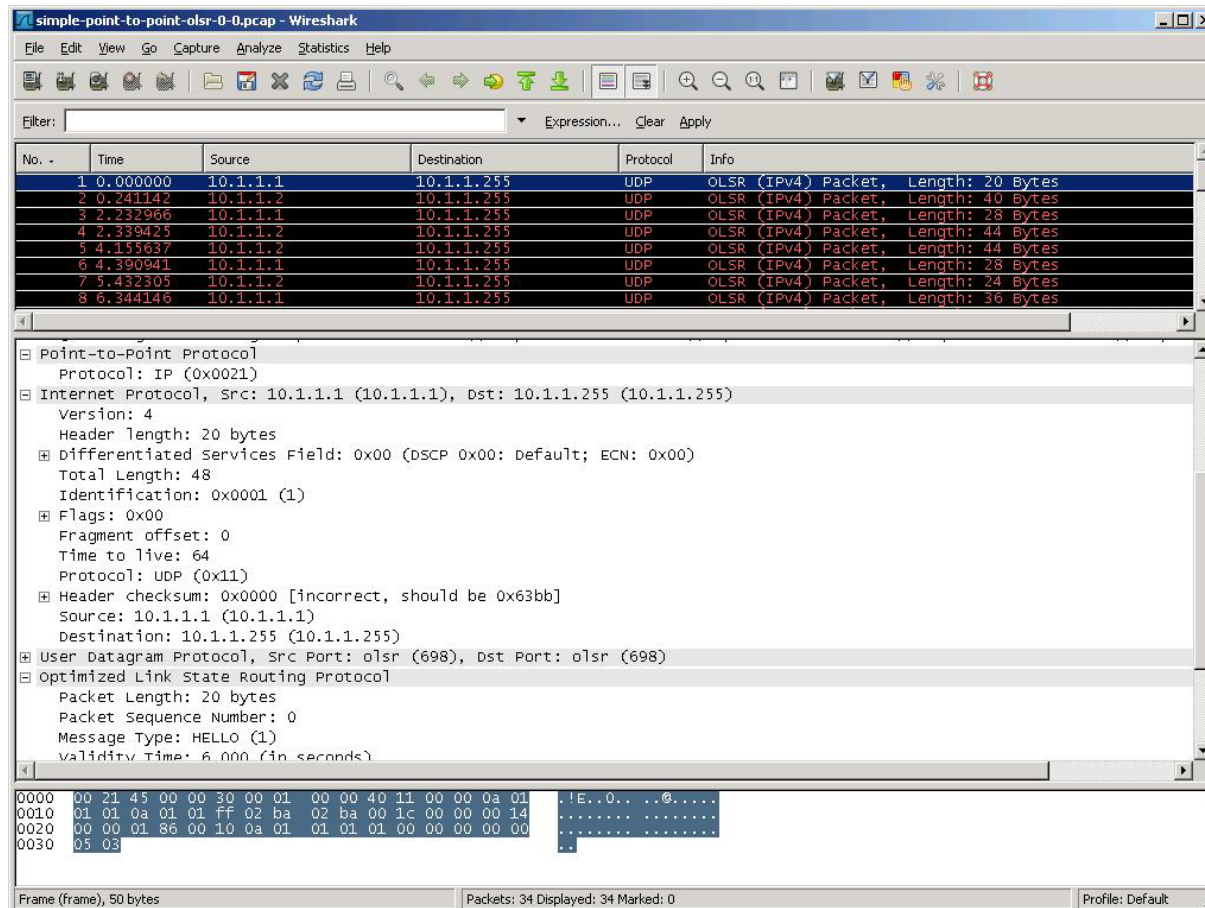
Problem: why reimplement models and tools for which open-source implementations abound?

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:



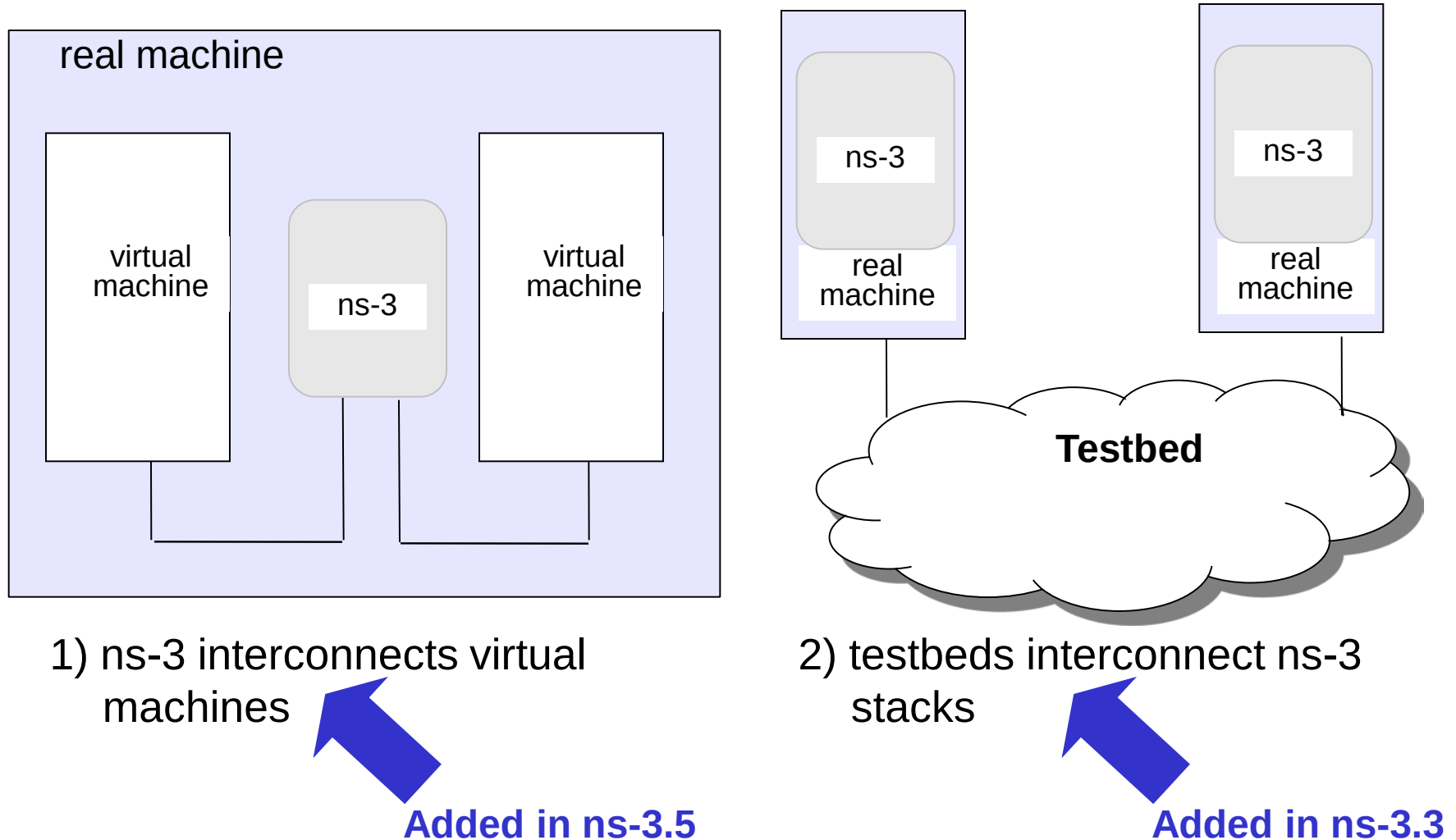
4) support for virtualization and testbeds

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 support for emulation



5) tracing and statistics

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

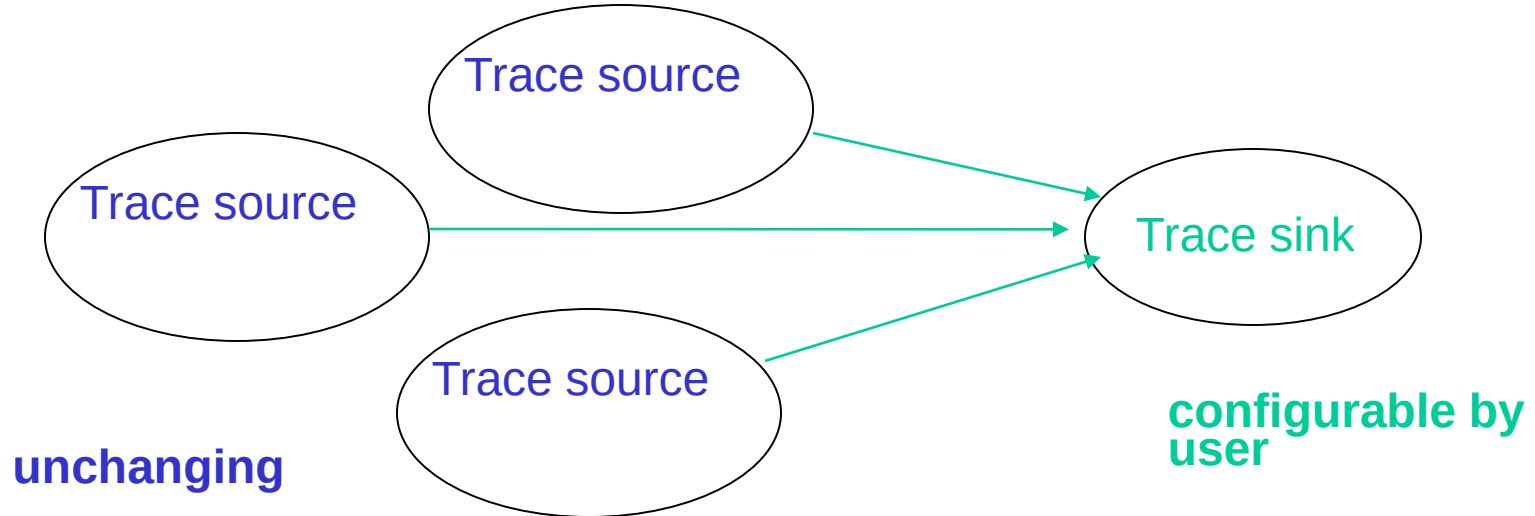
```
+ 1.84375 0 2 cbr 210 ----- 0 0.0 3.1 225 610
- 1.84375 0 2 cbr 210 ----- 0 0.0 3.1 225 610
r 1.84471 2 1 cbr 210 ----- 1 3.0 1.0 195 600
r 1.84566 2 0 ack 40 ----- 2 3.2 0.1 82 602
+ 1.84566 0 2 tcp 1000 ----- 2 0.1 3.2 102 611
```

Problem: 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

ns-3 solution: 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

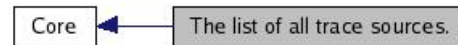
NS-3

- ns-3 Documentation
- NS-3 Modules
- NS-3 Class List
- NS-3 Class Hierarchy
- Class Members
- NS-3 Graphical Class Hierarchy
- NS-3 Namespace List
- Namespace Members
- NS-3 Related Pages

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Related Pages](#)

The list of all trace sources. [Core]

Collaboration diagram for The list of all trace sources.:



ns3::WifiNetDevice

- Rx: Received payload from the MAC layer.
- Tx: Send payload to the MAC layer.

ns3::WifiPhy

- State: The WifiPhy state
- RxOk: A packet has been received successfully.
- RxError: A packet has been received unsuccessfully.
- Tx: Packet transmission is starting.

ns3::MobilityModel

- CourseChange: The value of the position and/or velocity vector changed

ns3::olsr::AgentImpl

- Rx: Receive OLSR packet.
- Tx: Send OLSR packet.
- RoutingTableChanged: The OLSR routing table has changed.

ns3::PacketSink

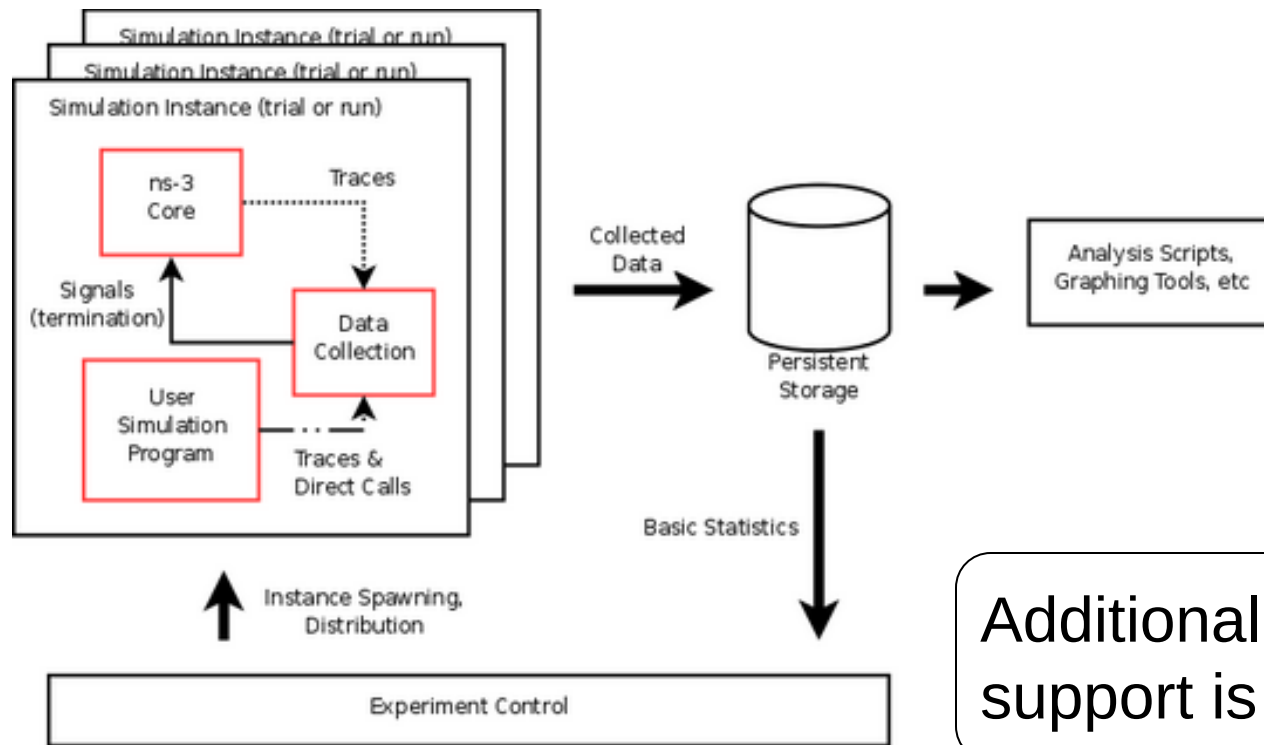
statistics framework

- 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](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

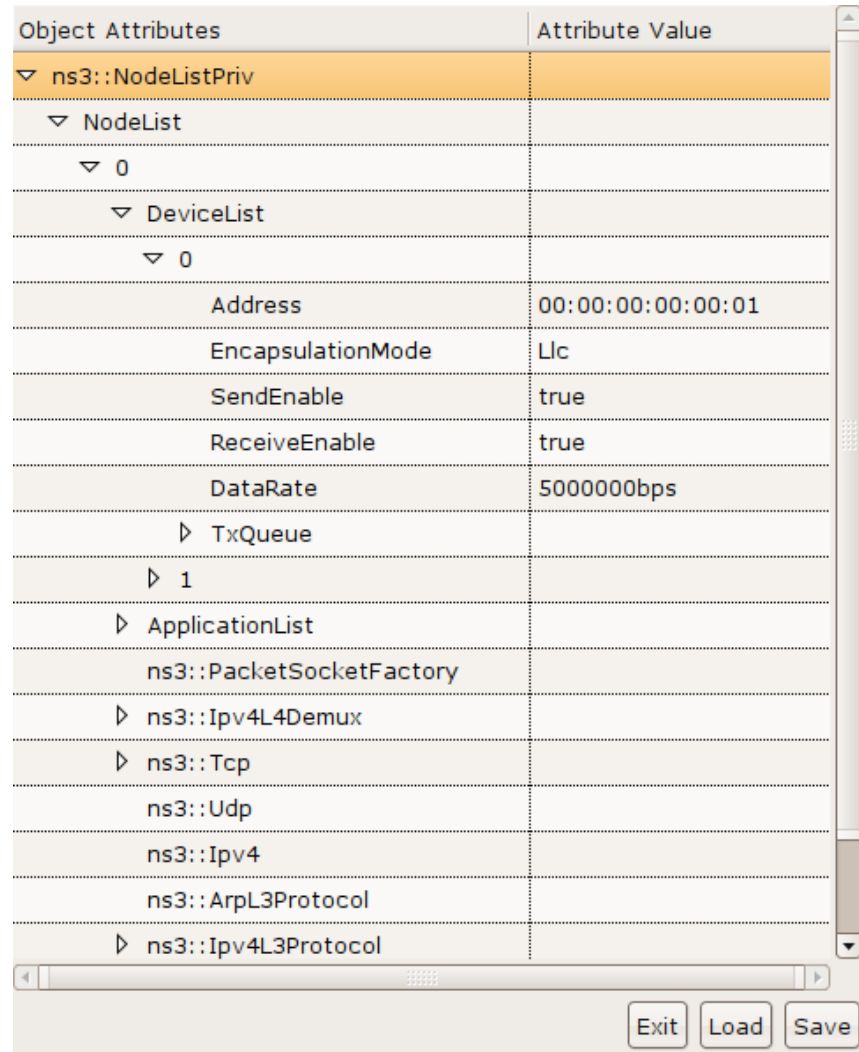
- and configure them easily

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:



The screenshot shows a graphical configuration tool window titled 'Object Attributes'. It displays a hierarchical tree of attributes for the object 'ns3::NodeListPriv'. The tree structure is as follows:

- ns3::NodeListPriv
 - NodeList
 - 0
 - DeviceList
 - 0
 - Address: 00:00:00:00:00:01
 - EncapsulationMode: Llc
 - SendEnable: true
 - ReceiveEnable: true
 - DataRate: 5000000bps
 - TxQueue
 - 1
 - ApplicationList
 - ns3::PacketSocketFactory
 - ns3::Ipv4L4Demux
 - ns3::Tcp
 - ns3::Udp
 - ns3::Ipv4
 - ns3::ArpL3Protocol
 - ns3::Ipv4L3Protocol

The table below represents the data shown in the 'Attribute Value' column for the expanded '0' device list entry.

Attribute	Value
Address	00:00:00:00:00:01
EncapsulationMode	Llc
SendEnable	true
ReceiveEnable	true
DataRate	5000000bps

At the bottom of the window, there are three buttons: 'Exit', 'Load', and 'Save'.

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

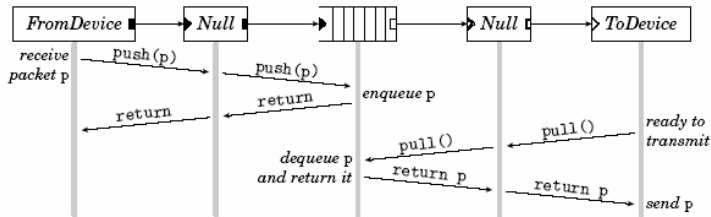
These approaches still need to be developed

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

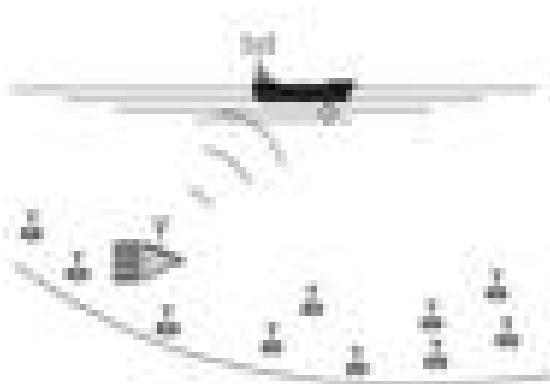
Current and recent ns-3 maintainers

- 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

ns-3 and Google Summer of Code



Click Modular Router
Lalith Suresh

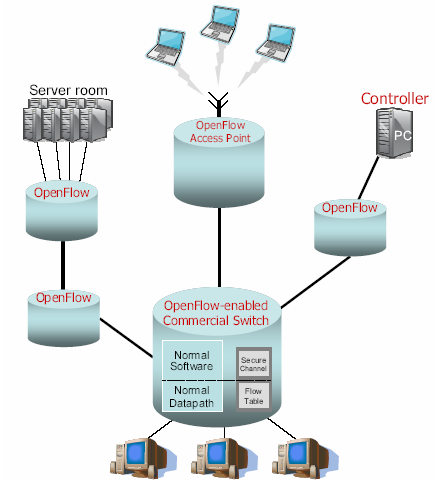


Underwater Acoustic Networking
Andrea Sacco



<http://www.nsnam.org>

ns-3 overview May 2011



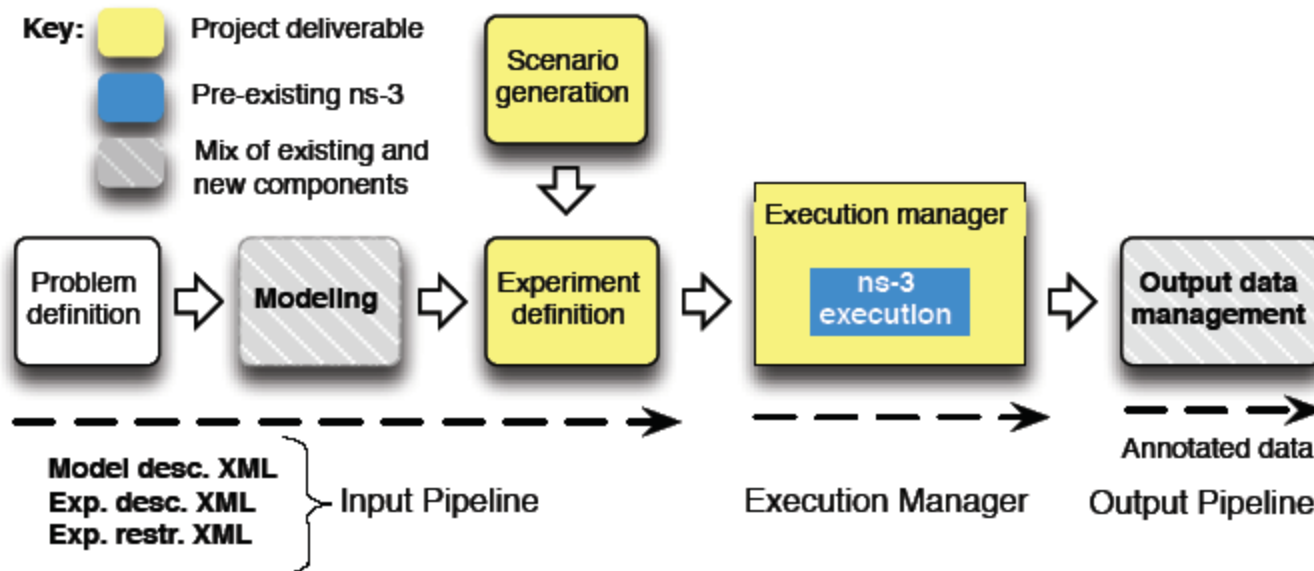
ns-3 OpenFlow
Blake Hurd



ns-3 LTE
Giuseppe Piro

Frameworks for ns-3

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



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

Acknowledgment of support



**Georgia
Tech**



Information Sciences Institute

<http://www.nsnam.org>

ns-3 overview May 2011