5G New Radio
Simulations with ns-3

Lorenza Giupponi
Research Director, Mobile Networks Department
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)

Thomas R. Henderson
ns-3 co-founder and lead maintainer
Executive Director, University of Washington ns-3 Consortium

ACMSE Conference
April 17, 2021

Tutorial goals

- Attendees should gain a basic understanding of the scope of the 5G New Radio module in the ns-3 discrete-event network simulator
- Attendees should be able to start running and modifying 5G New Radio example programs

5G -LENA

Outline

1. ns-3 orientation
2. Overview of NR module
3. Review of NR example program

- We will monitor the conference chat for questions
- We will take a 5 minute break at the end of each hour, and answer questions
- Slides are posted on the conference site
- Video will be posted at a later date

Acknowledgments

- Tom Henderson’s work on ns-3 has been supported by multiple NSF and NIST awards*
- A team led by Lorenza Giupponi at CTTC has developed ns-3 cellular models for almost a decade
- ns-3 credits nearly 300 authors, and is built upon the ns-2 and ns-1 tools that originated in the 1990s

*
- NSF award CNS-0551686: (2006-10)
- NSF award CNS-0958139: (2010-15)
- NSF award CNS-1836725 (2018-present)
- NSF award CNS-2016379: (2020-present)
- NIST Cooperative Agreement 70NANB17H170: (2017-2020)
- NIST Cooperative Agreement 70NANB20H179: (2020-present)

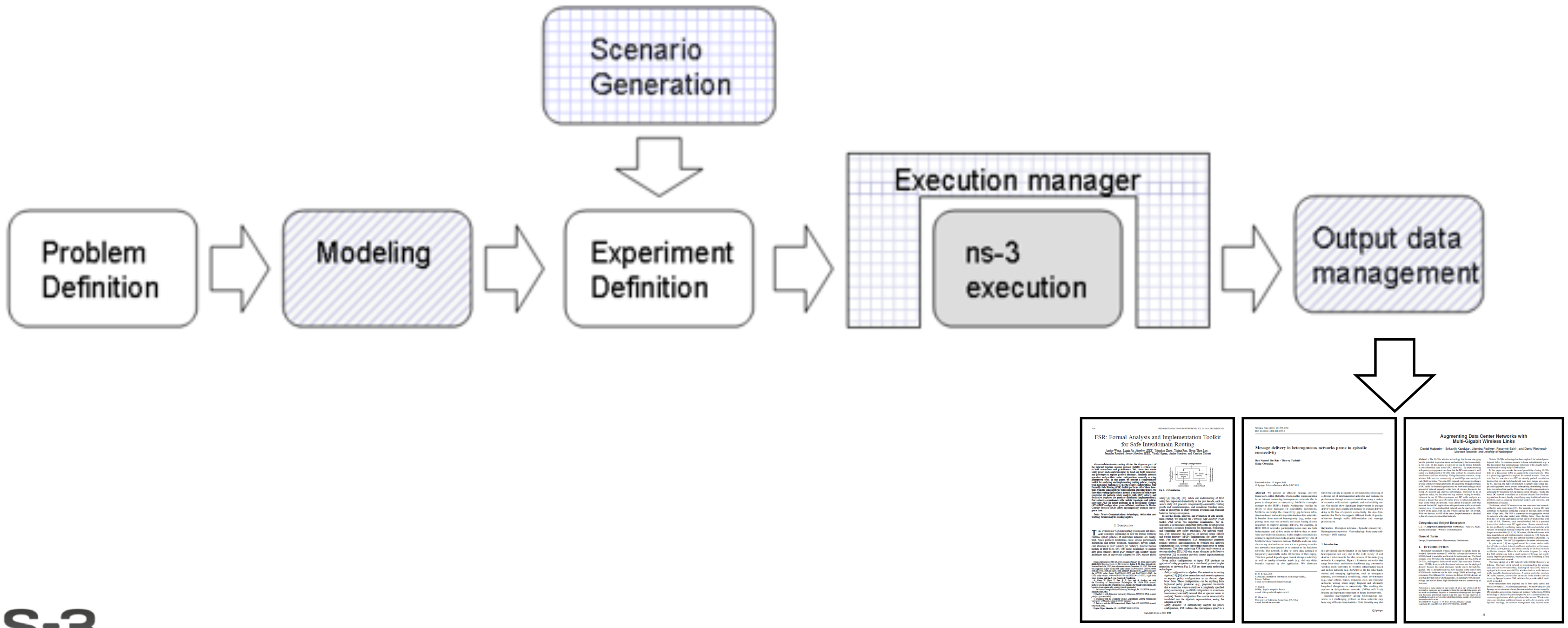
What is 5G NR?

The next generation cellular system

- eMBB (enhanced Mobile Broadband)
- URLLC (Ultra Reliable Low Latency Communications)
- mMTC (massive Machine Type Communications)

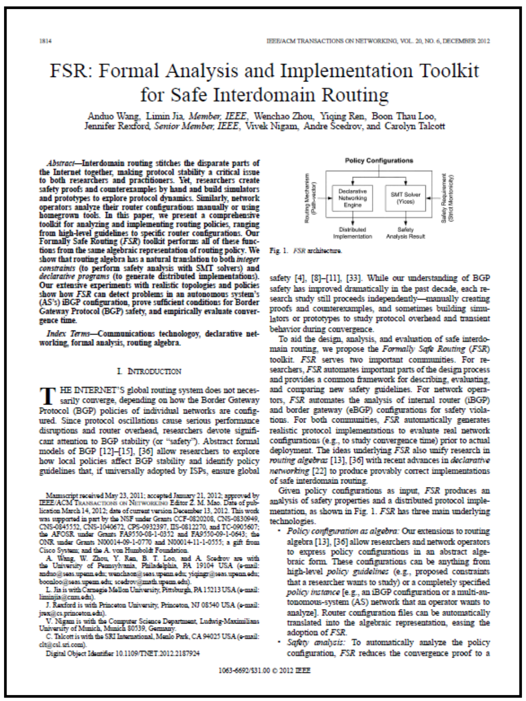
What is ns-3?

- A set of C++ discrete-event simulation libraries
- Community-contributed and maintained modules
- A tool to answer performance questions about computer networks



What has ns-3 been used for?

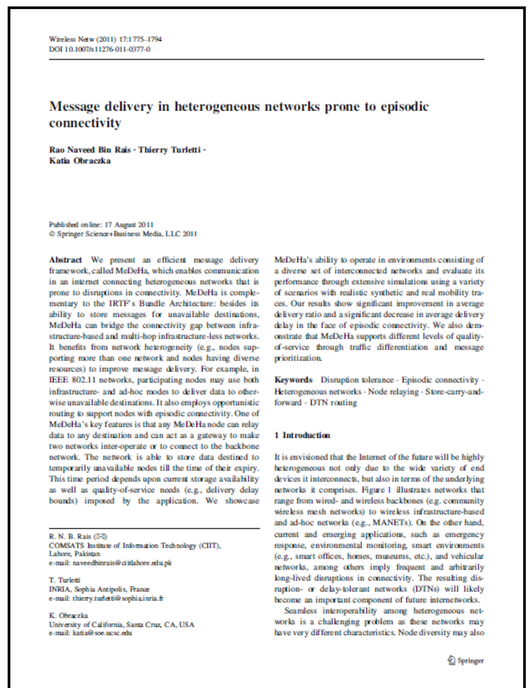
- Thousands of publications to date
  - search of 'ns-3 simulator' on IEEE and ACM digital libraries, or Google Scholar



1814 IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 20, NO. 6, DECEMBER 2012

FSR: Formal Analysis and Implementation Toolkit for Safe Interdomain Routing

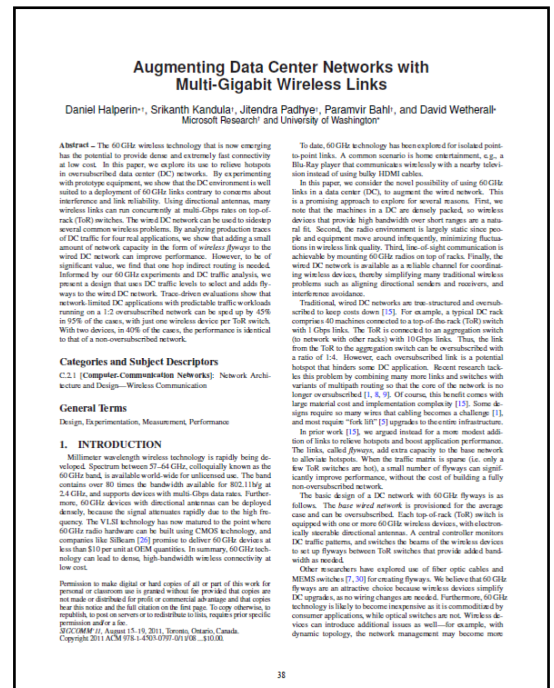
Anduo Wang, Limin Jia, Member, IEEE, Wenchao Zhou, Yiqing Ren, Boon Thau Loo, Jennifer Rexford, Senior Member, IEEE, Vivek Nigam, Andre Scedrov, and Carolyn Talcott



Wireless Netw (2011) 17:1775–1794
DOI 10.1007/s11276-011-0377-0

Message delivery in heterogeneous networks prone to episodic connectivity

Rao Naveed Bin Rais · Thierry Turletti · Katia Obraczka



Augmenting Data Center Networks with Multi-Gigabit Wireless Links

Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall
Microsoft Research and University of Washington

Why simulation for wireless networks?

Simulation is commonly used for wireless research, for the following reasons:

1) experiments can be **replicated exactly**

2) testbeds often do not provide **low-level access** to change how devices work

3) **tests at scale are hard** or expensive to orchestrate

4) there may be **frequency coordination**/interference concerns

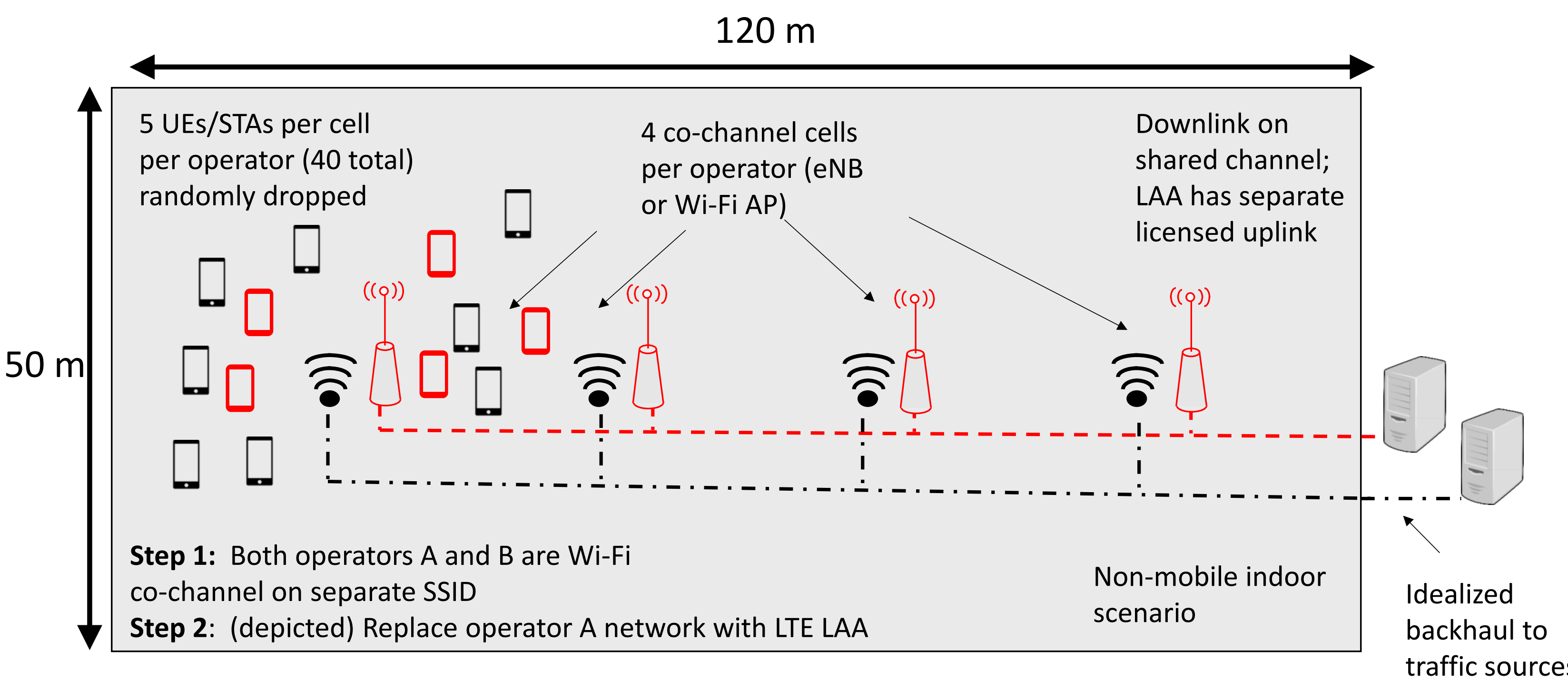
Simulation is also used in the technical standardization process; e.g.

- IEEE 802.11-14/0571r12, 11ax Evaluation Methodology
- 3GPP TR 36.889: Study on Licensed-Assisted Access to Unlicensed Spectrum

Example performance study

LTE Licensed-Assisted Access (LAA) coexistence

- ”Can a specific unlicensed variant of LTE (LAA) operate in the same spectrum as a Wi-Fi network, without impacting Wi-Fi system throughput and latency more than another co-located Wi-Fi network would impact it?”



Configuration details

Unlicensed channel model	3GPP TR 36.889	ns-3 implementation
Network Layout	Indoor scenario	Indoor scenario
System bandwidth	20 MHz	20 MHz
Carrier frequency	5 GHz	5 GHz (channel 36, tunable)
Number of carriers	1, 4 (to be shared between two operators) 1 for evaluations with DL+UL Wi-Fi coexisting with DL-only LAA	1 for evaluations with DL+UL Wi-Fi coexisting with DL-only LAA
Total Base Station (BS) transmission power	18/24 dBm	18/24 dBm Simulations herein consider 18 dBm
Total User equipment (UE) transmission power	18 dBm for unlicensed spectrum	18 dBm
Distance dependent path loss, shadowing and fading	ITU InH	802.11ax indoor model
Antenna pattern	2D Omni-directional	2D Omni-directional
Antenna height	6 m	6 m (LAA, not modelled for Wi-Fi)
UE antenna height	1.5 m	1.5 m (LAA, not modelled for Wi-Fi)
Antenna gain	5 dBi	5 dBi
UE antenna gain	0 dBi	0 dBi
Number of UEs	10 UEs per unlicensed band carrier per operator for DL-only 10 UEs per unlicensed band carrier per operator for DL-only for four unlicensed carriers. 20 UEs per unlicensed band carrier per operator for DL+UL for single unlicensed carrier. 20 UEs per unlicensed band carrier per operator for DL+UL Wi-Fi coexisting with DL-only LAA	Supports all the configurations in TR 36.889. Simulations herein consider the case of 20 UEs per unlicensed band carrier per operator for DL LAA coexistence evaluations for single unlicensed carrier.
UE Dropping	All UEs should be randomly dropped and be within coverage of the small cell in the unlicensed band.	Randomly dropped and within small cell coverage.
Traffic Model	FTP Model 1 and 3 based on TR 36.814 FTP model file size: 0.5 Mbytes. Optional: VoIP model based on TR36.889	FTP Model 1 as in TR36.814. FTP model file size: 0.5 Mbytes Voice model: DL only
UE noise figure	9 dB	9 dB
Cell selection	For LAA UEs, cell selection is based on RSRP (Reference Signal Received Power. For Wi-Fi stations (STAs), cell selection is based on RSS (Received signal power strength) of WiFi Access Points (APs). RSS threshold is -82 dBm.	RSRP for LAA UEs and RSS for Wi-Fi STAs
Network synchronization	For the same operator, the network can be synchronized. Small cells of different operators are not synchronized.	Small cells are synchronized, different operators are not synchronized.

Figure from: [Giupponi et al., Simulating LTE and Wi-Fi Coexistence in Unlicensed Spectrum with ns-3]


3GPP TR 36.889 V13.0.0 (2015-06)

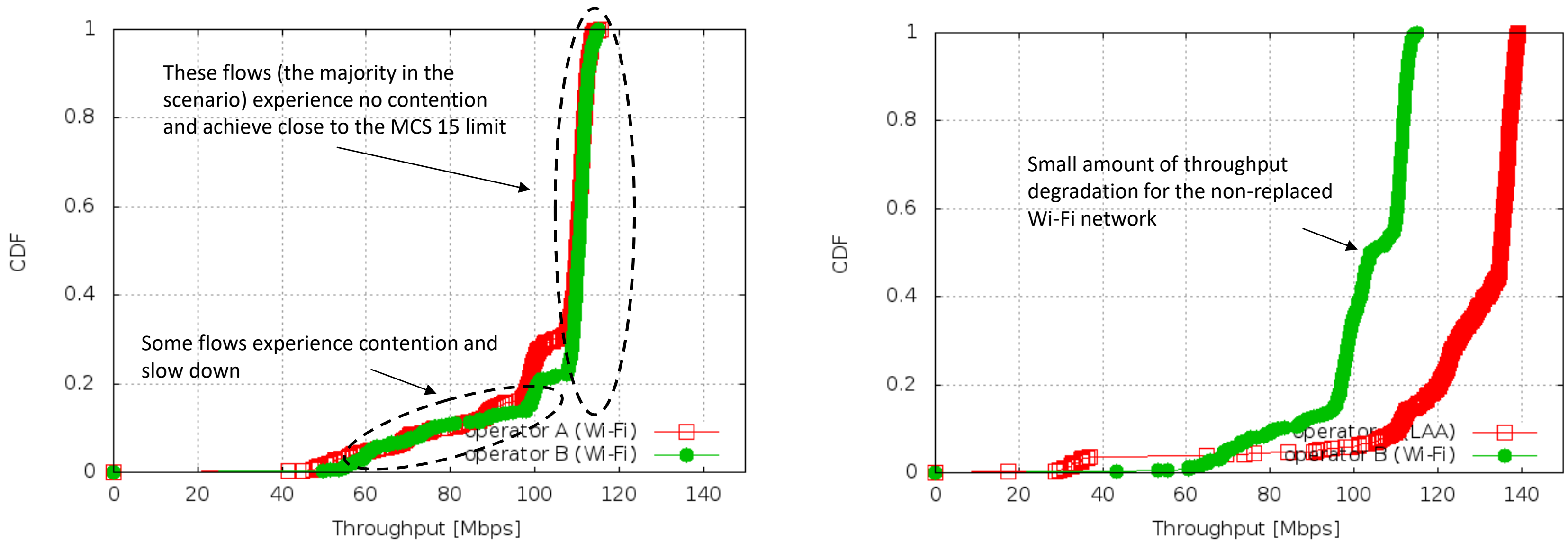
Technical Report

3rd Generation Partnership Project;
Technical Specification Group Radio Access Network;
Study on Licensed-Assisted Access to Unlicensed Spectrum;
(Release 13)

Configuration drawn from TR 36.889

Sample results

- Place two Wi-Fi networks in same region, fully load the system, and plot a CDF of observed throughputs per station. Repeat by replacing one Wi-Fi network with LAA.



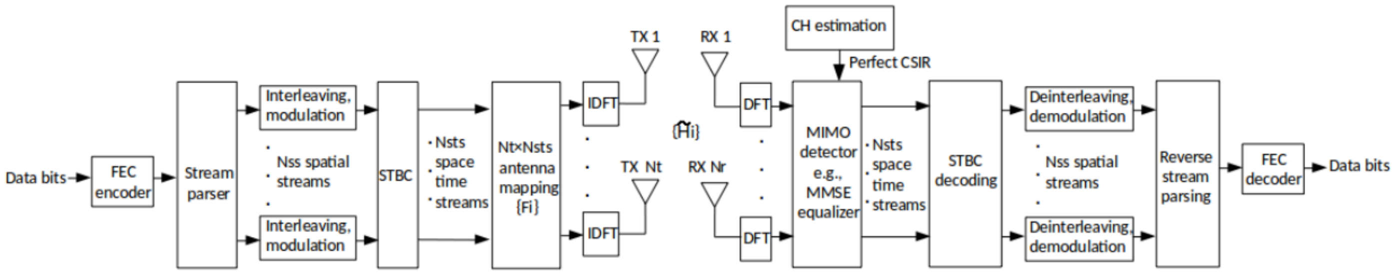
a) Step 1 (Wi-Fi) b) Step 2 (LAA)

Figures from ns-3 LAA Wi-Fi Coexistence simulation project

This is exemplary of an ns-3 performance study

Simulation alternatives

- “Link” or physical-layer simulations
  - MATLAB, Vienna LTE-A Simulator
  - Unit of granularity is the symbol (high-level of PHY realism)



Sian Jin, Sumit Roy, Weihua Jiang, and Thomas R. Henderson. 2020. Efficient Abstractions for Implementing TGn Channel and OFDM-MIMO Links in ns-3. In Proceedings of the 2020 Workshop on ns-3 (WNS3 2020).

modified from T. Paul and T. Ogunfrunmiri. 2008. Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment. IEEE Circuits and Systems Magazine 8, 1 (First 2008), 28–54.

$N_t \times N_r (= N_{ss})$	Bandwidth	MATLAB Full-link
1×1	20MHz	28 min
1×1	40MHz	25 min
2×2	20MHz	37 min
2×2	40MHz	39 min
3×3	20MHz	51 min
3×3	40MHz	60 min

Run-times for 40,000 simulated packets between two nodes

Table 2 from Jin et al (cited above)

Simulation alternatives (cont.)

- "Packet-level" or full-stack simulations
  - The main unit of granularity is the packet
  - ns-2, ns-3, OMNeT++, OPNET, QualNet

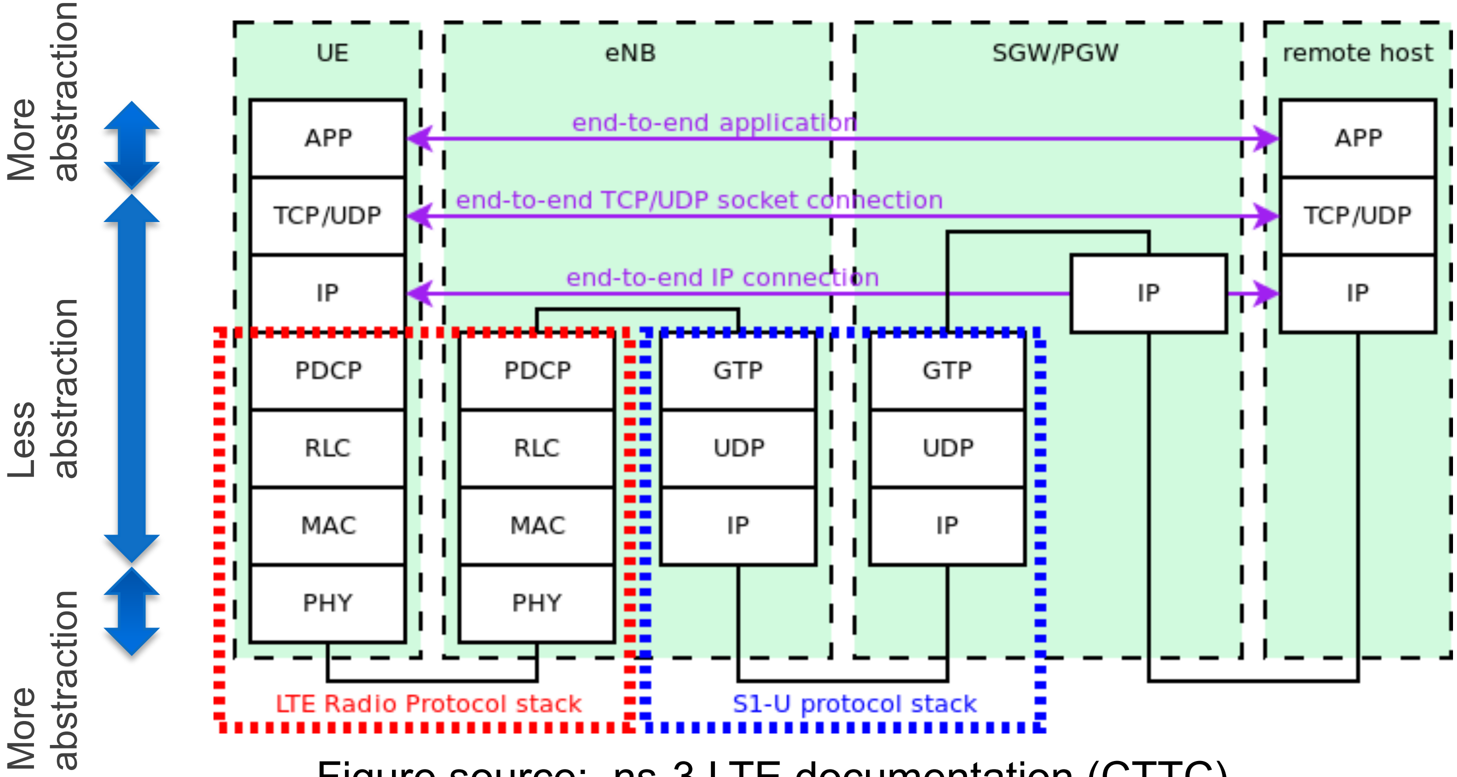


Figure source: ns-3 LTE documentation (CTTC)

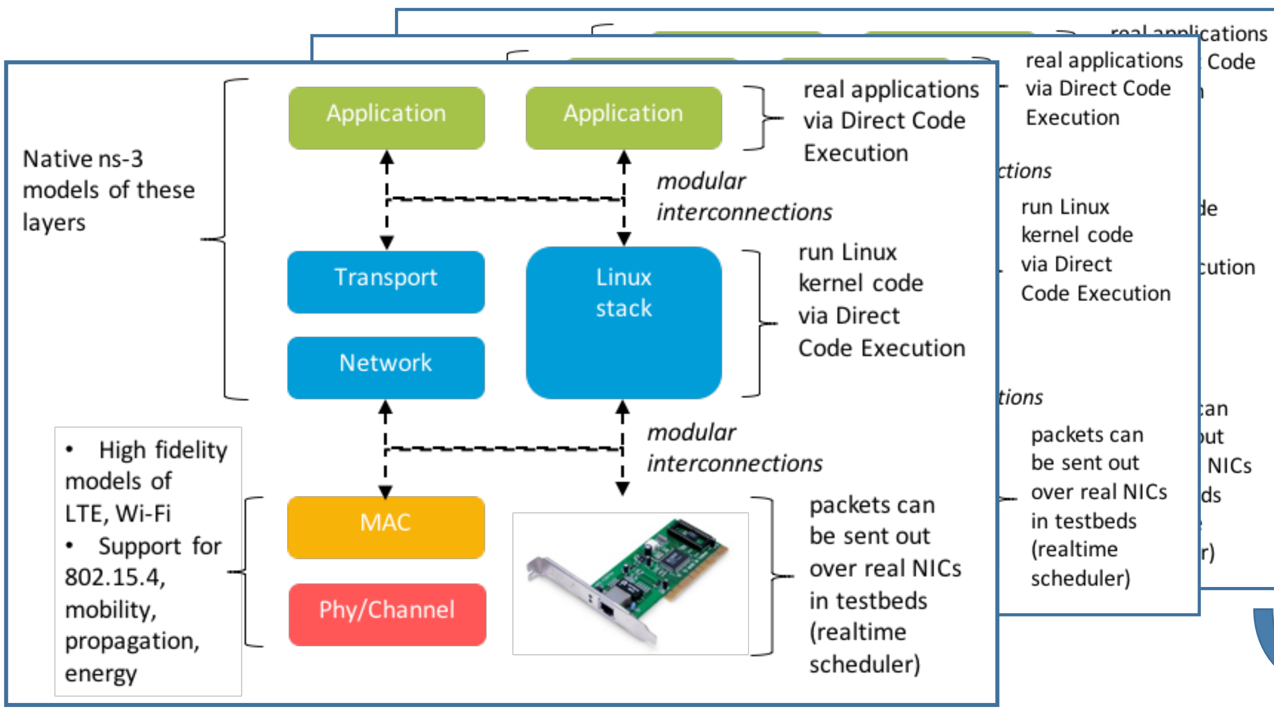
Simulation alternatives (cont.)

- "System-level" or flow simulations
  - Individual packets are not modeled, but the effects of messages exchanged or packet aggregates (flows)
  - examples: ExtendSim, SimGrid

ns-3 architecture

- ns-3 is a leading **open source**, **packet-level network simulator** oriented towards network research, featuring a **high-performance core** enabling **parallelization across a cluster** (for large scenarios), **ability to run real code**, and **interaction with testbeds**

Runs on a single machine



or partitioned across a cluster

Outline

1. ns-3 orientation
   – Obtaining the code
   – Configuring and building ns-3
   – Walkthrough of simple example: udp-client-server.cc
2. Overview of NR module
   – Assumptions and Architecture
   – PHY
   – MAC
   – Validation
   – Extensions (NR-U, NR-V2X)
3. Review of existing NR example
   – cttc-nr-demo.cc
   – pointers to documentation of other examples

Software orientation

ns-3 is written in C++, with Python bindings

- Users need to work at the Linux (or Unix, macOS, etc.) command-line, compile the code, and handle raw output data

Key differences from other network simulators:

1) Command-line, Unix orientation
   - vs. Integrated Development Environment (IDE)

2) Simulations and models written directly in C++ and Python
   - vs. a domain-specific simulation language

ns-3 not written in a high-level language

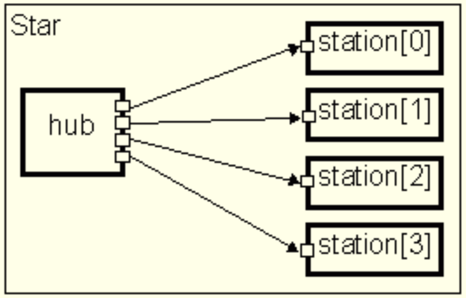
Submodule vectors, gate vectors and multiple connections are illustrated in the following example:

```
simple Hub
  gates:
    out: outport[];
endsimple

simple Station //...

module Star
  submodules:
    hub: Hub
      gatesizes: outport[4];
    station: Station[4];
  connections:
    for i=0..3 do
          hub.outport[i] --> station[i].in;
    endfor
endmodule
```

The result of the above is depicted in Fig.4.



Example of OMNeT++ Network Description (NED) language
Figure excerpted from http://www.ewh.ieee.org/soc/es/Nov1999/18/ned.htm

ns-3 does not have a graphical IDE

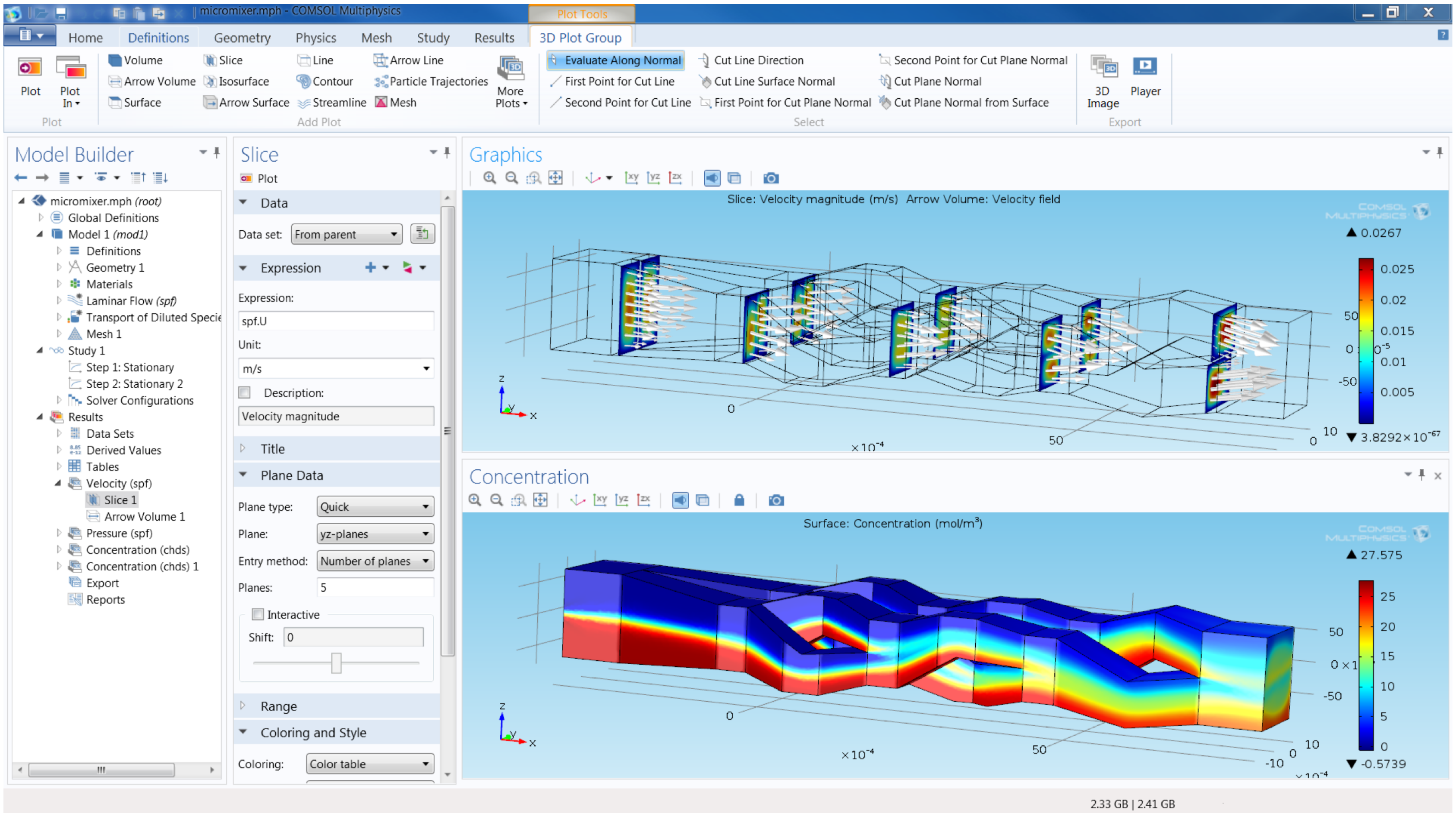
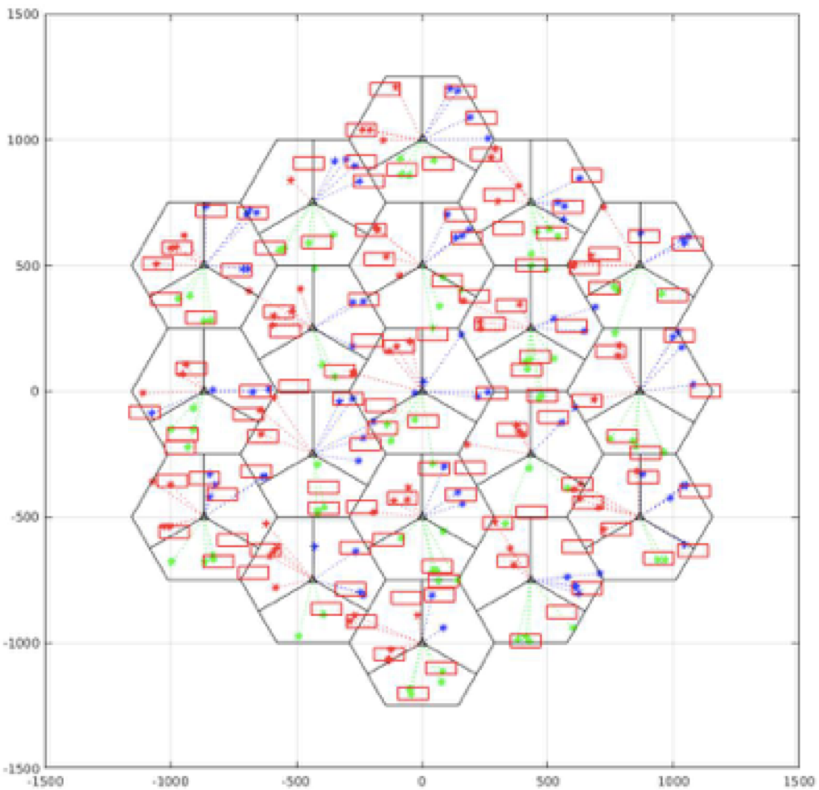


Figure source: https://www.comsol.com/comsol-multiphysics

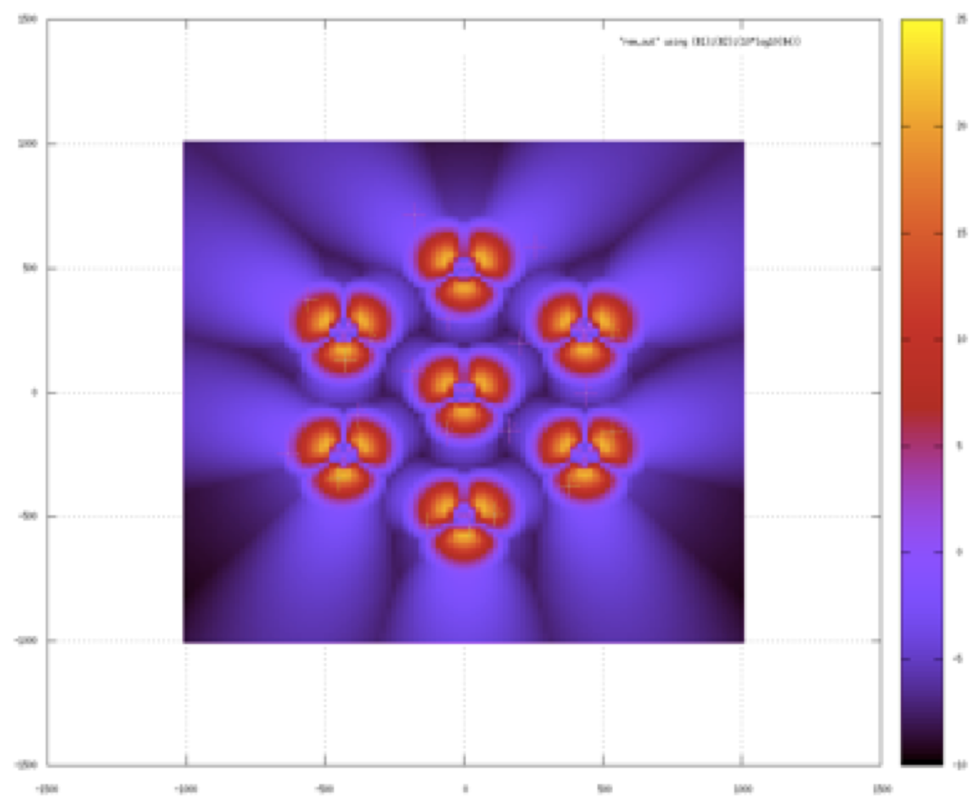
ns-3 uses external programs for graphics

Network cell structure showing eNBs, UEs, and buildings



Matplotlib

Radio environment map showing signal strength from eNBs



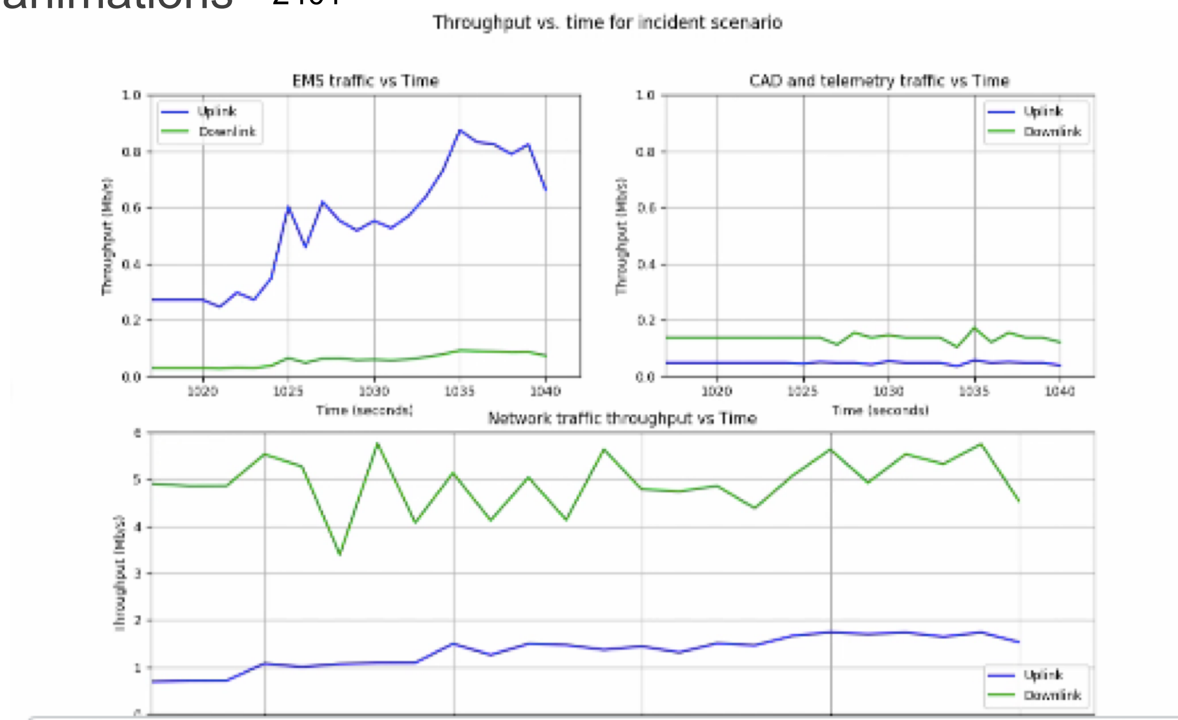
gnuplot

Figures due to Zoraze Ali

ns-3 users write scripts for plotting

Plots or animations

EmsVideo_1_Server 942.36392953 RX 1012 1061 U 2395
EmsVideo_1_Client 942.37317727 TX 1012 1061 U 2398
EmsVideo_1_Client 942.377 RX 64 113 U 2397
WebBrowsingGraphics_0_Server 942.38092876 TX 1024 1073 U 2399
WebBrowsingGraphics_0_Client 942.394 RX 1024 1073 U 2399
AvlAssetPerimeter_1_Server 942.42492988 RX 1408 1457 U 2401



Used to measure KPIs

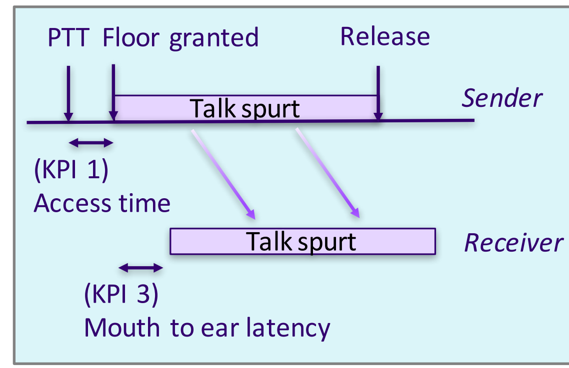
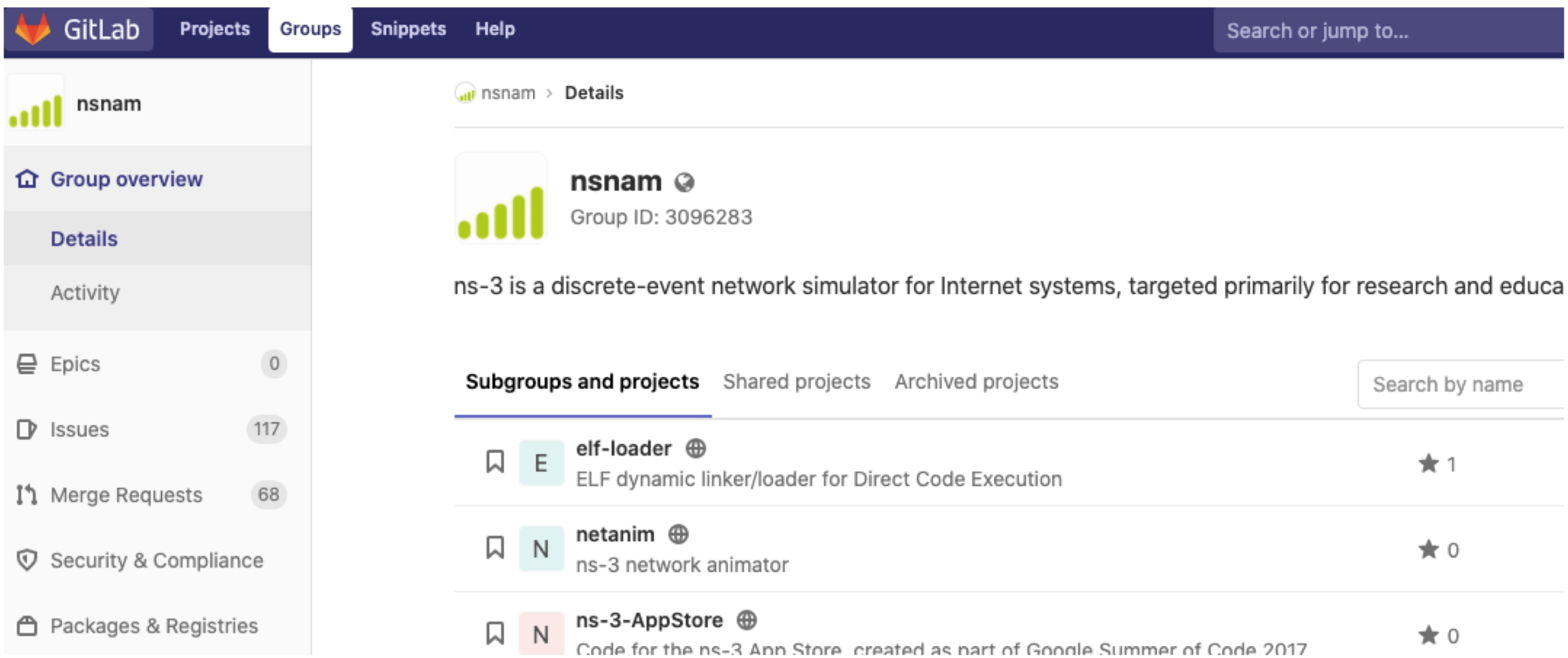


Figure source: 3GPP TS 22.179

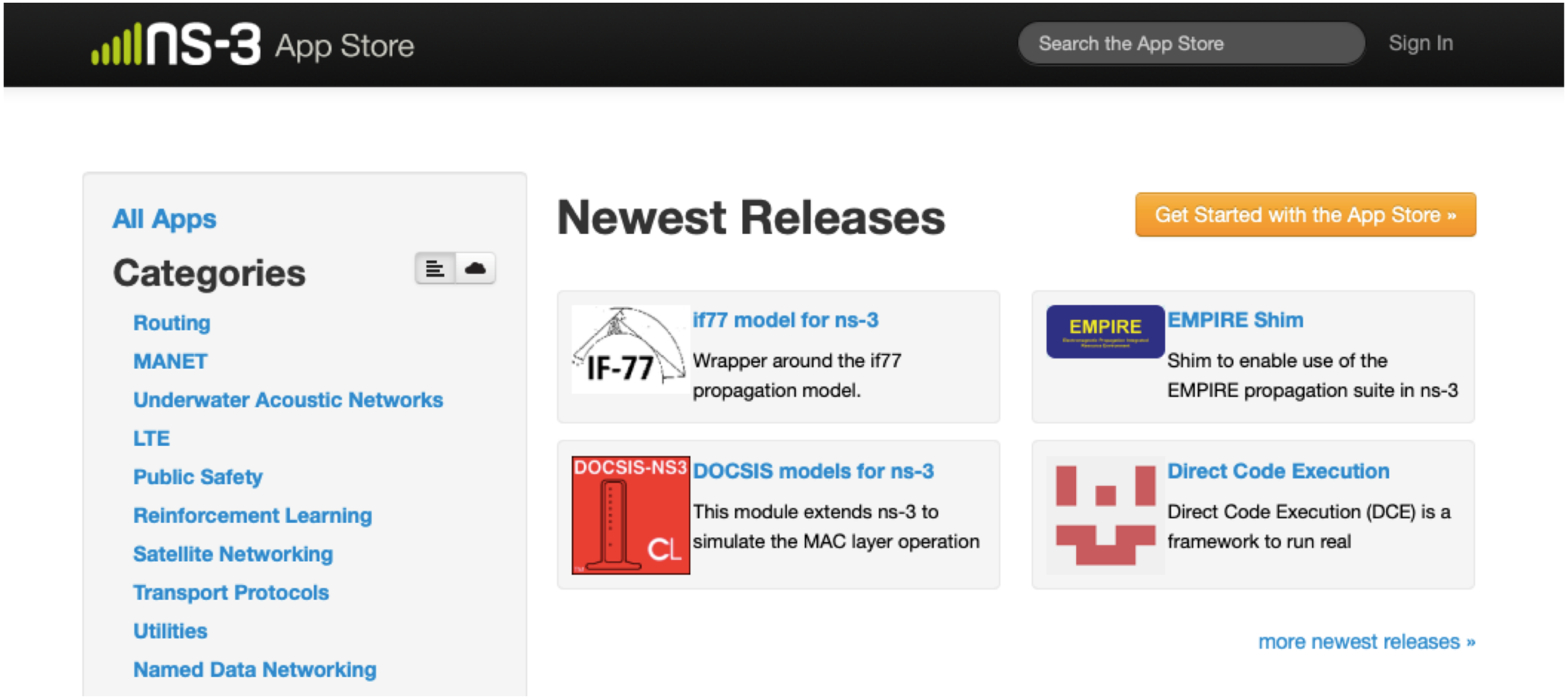
ns-3 software repositories

- The main web site (documentation, wiki) is https://www.nsnam.org
- Software is hosted in Git repositories at https://gitlab.com/nsnam/

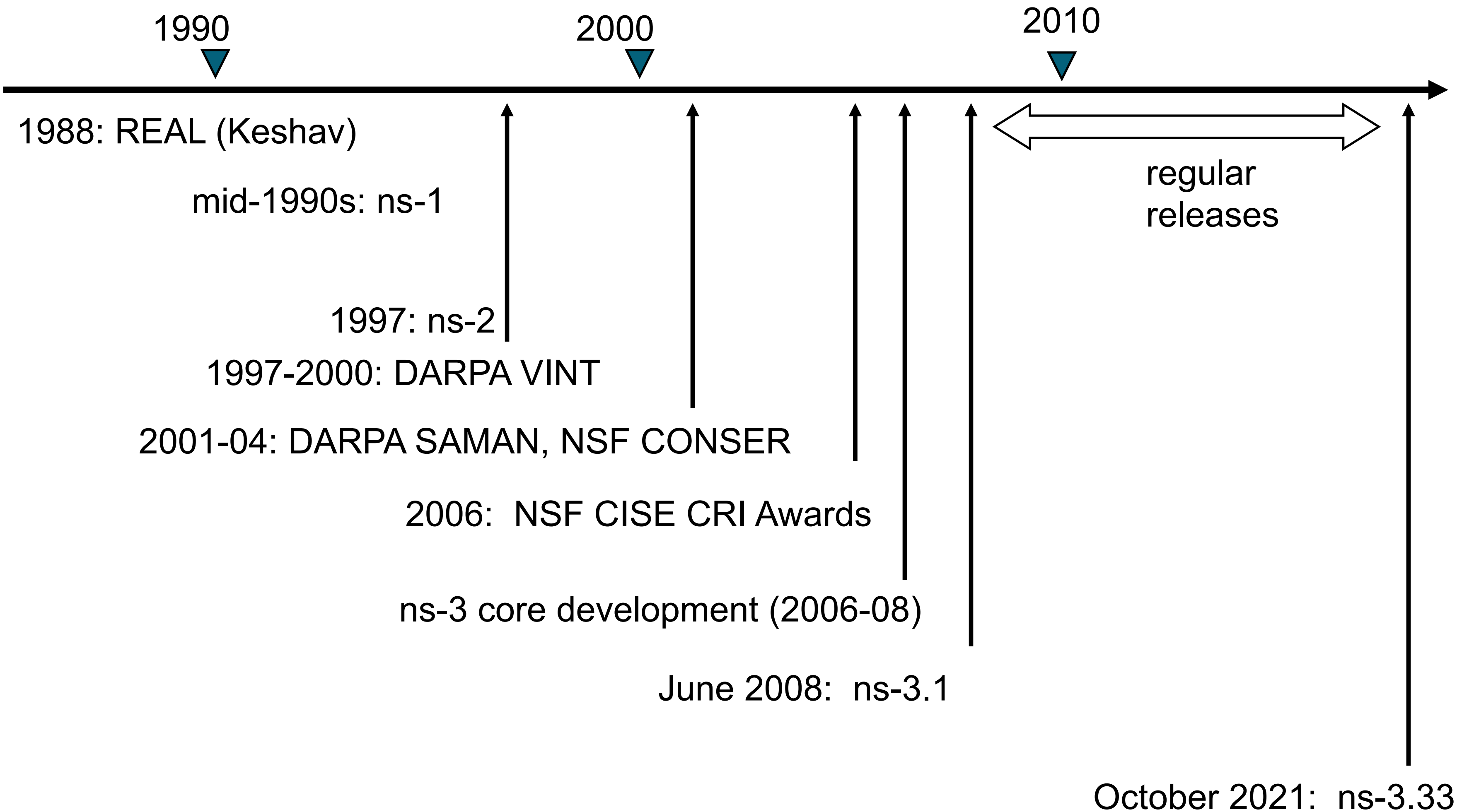


ns-3 App Store

- https://apps.nsnam.org



ns timeline


1990
2000
2010
1988: REAL (Keshav)
mid-1990s: ns-1
1997: ns-2
1997-2000: DARPA VINT
2001-04: DARPA SAMAN, NSF CONSER
2006: NSF CISE CRI Awards
ns-3 core development (2006-08)
June 2008: ns-3.1
regular releases
October 2021: ns-3.33

An open source project

Between mid-2019 and mid-2020:

- **578 commits** by **51 authors** (33 new!)
- **Maintainer commits** from
  – Alexander Krotov, Biljana Bojovic, Manuel Requena, Mohit Tahiliani, Natale Patriciello, Peter Barnes, Getachew Redietab, Sebastien Deronne, Stefano Avallone, Tom Henderson, Tommaso Pecorella, Zoraze Ali
- **72,696 lines** of C++ code added/deleted
  – parsed output of git diff --stat
- **261 Merge Requests** opened
- **160 Issues** filed

Links to videos, tutorials

- To get started using ns-3
  - ns-3 tutorial
  - ns-3 training videos (2019 annual meeting)
    - special sessions for TCP, Wi-Fi, LTE, sensor networks
  - Other videos (available also by searching YouTube)
- More information about ns-3 and wireless
  - 2019 Workshop on Next-Generation Wireless with ns-3 (WNGW)
    - many talks on a variety of wireless technologies, from industry and academia

4G LTE models in ns-3

- LENA: An open source product-oriented LTE/EPC Network Simulator
  - Developed by CTTC, Barcelona from 2012-present
  - Designed around an industrial API: the Small Cell Forum MAC Scheduler Interface Specification
  - Full stack, end-to-end
  - Accurate model of the LTE/EPC protocol stack
  - Specific Channel and PHY layer models for LTE macro and small cells
  - Emphasis on radio-level performance and end-to-end QoE
  - Extended for public safety networks (NIST, UW, CTTC, in the ns-3 App Store)

5G NR module (this tutorial)

- 5G NR (led by CTTC, Barcelona)
  - Leverages mmWave models developed by the University of Padova and NYU Wireless
  - Supports fundamental PHY-MAC NR features aligned with NR Release 15 TS 38.300
  - Additional recent work to support V2X and NR Unlicensed

Outline

1. ns-3 orientation
   – Obtaining the code
   – Configuring and building ns-3
   – Walkthrough of simple example: udp-client-server.cc
2. Overview of NR module
   – Assumptions and Architecture
   – PHY
   – MAC
   – Validation
   – Extensions (NR-U, NR-V2X)
3. Review of existing NR example
   – cttc-nr-demo.cc
   – pointers to documentation of other examples

Obtaining the code

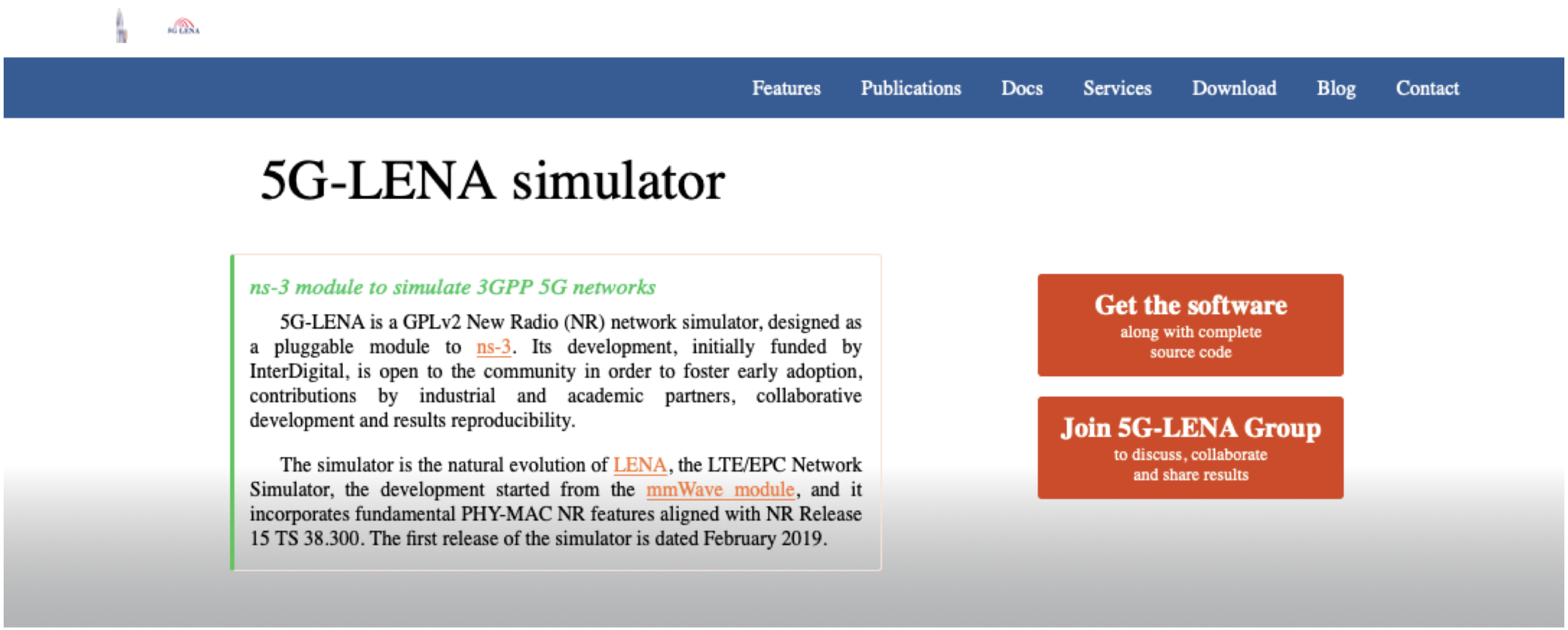
- For working with NR, we recommend using the development version of ns-3 along with the latest release of 5G LENA
  - ns-3-dev: git clone https://gitlab.com/nsnam/ns-3-dev.git
  - 5G NR: git clone https://gitlab.com/cttc-lena/nr.git
- Caveats:
  1. access to 5G NR must be requested from CTTC
  2. the 5G NR module must be downloaded separately as a contributed module
  3. we only support recent versions of Linux or macOS Command-Line Tools

Why a separately downloaded module?

- Useful to track impact
  - Supporting a module such as nr requires significant amount of institutional resources
  - CTTC benefits from this arrangement to better track the dissemination and impact of open sourcing this module for ns-3
- ns-3 in general is moving to a more federated development process through its App Store

Obtaining access to 5G NR module

- Visit https://5g-lena.cttc.es



Quick start

```
$ git clone https://gitlab.com/nsnam/ns-3-dev.git
$ cd ns-3-dev/contrib
$ git clone https://gitlab.com/cttc-lena/nr.git
$ cd nr
$ $ git checkout -b 5g-lena-v1.1.y origin/5g-lena-
v1.1.y
   Branch 5g-lena-v1.1.y set up to track remote
branch 5g-lena-v1.1.y from origin.
   Switched to a new branch '5g-lena-v1.1.y'
$ cd ../../
$ ./waf configure -d optimized --enable-examples --
enable-tests --enable-logs
$ ./waf build
$ ./test.py
$ ./waf --run cttc-nr-demo
```

Documentation

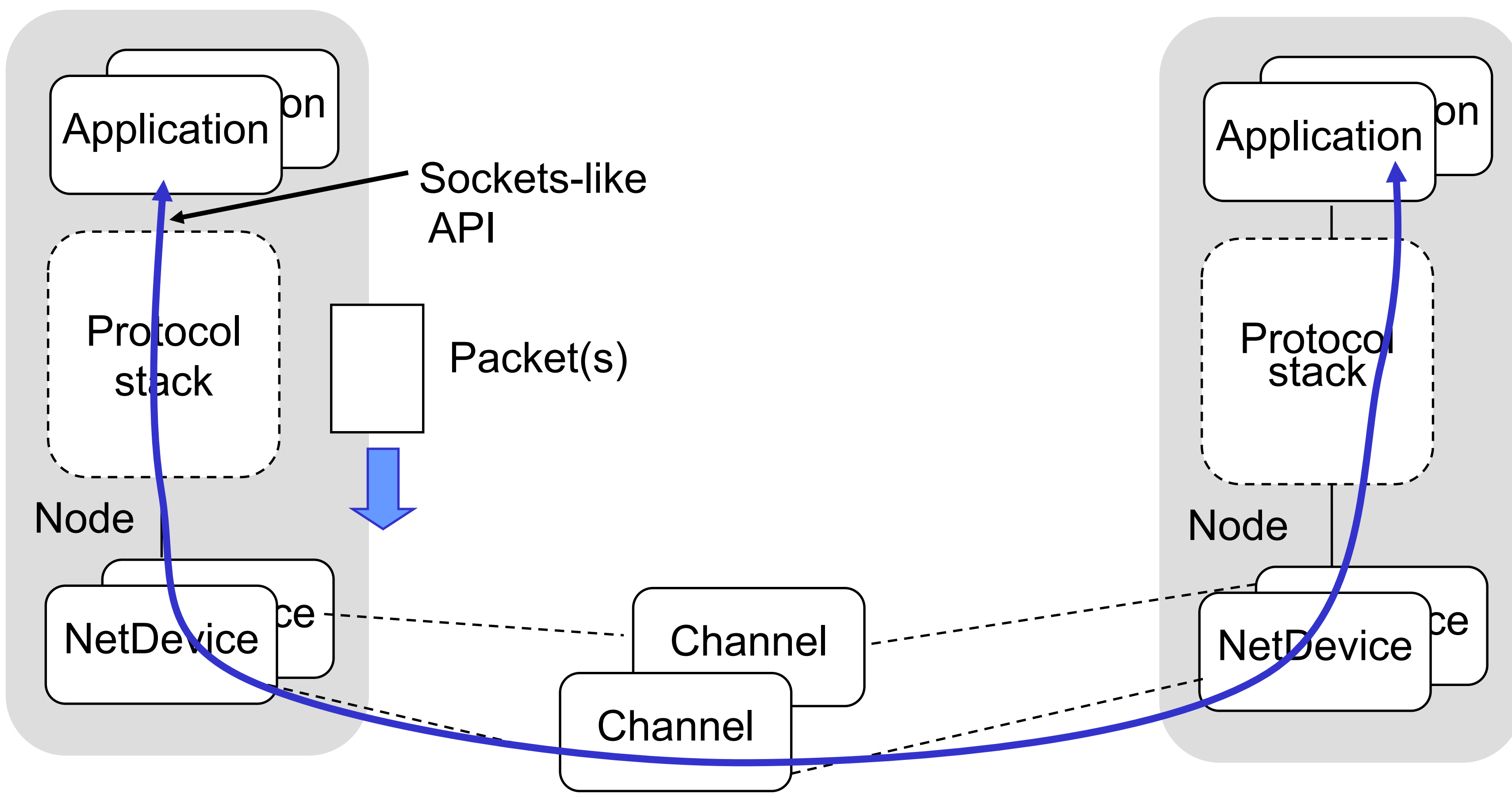
Extensive documentation at both sites

- Main ns-3 website https://www.nsnam.org
  - Tutorial, Manual, Model Library
  - Doxygen
  - Wiki
  - Release Notes
  - Training videos from past Workshops on ns-3
- 5G-LENA site: https://5g-lena.cttc.es/
  - Getting Started, Manual
  - Doxygen
  - Release Notes

Placeholder

review of udp-client-server.cc example

Key classes in ns-3

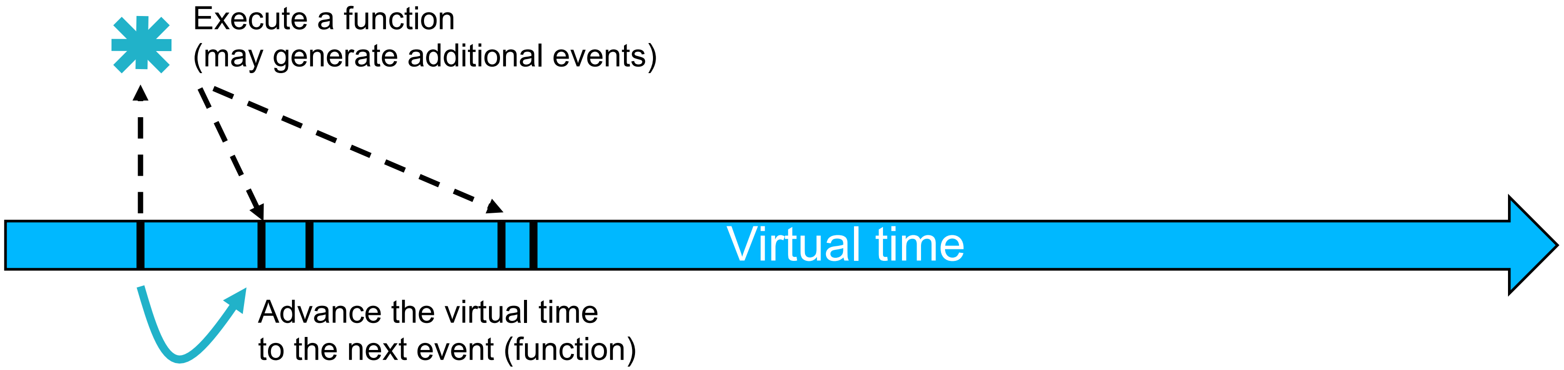


Key concepts

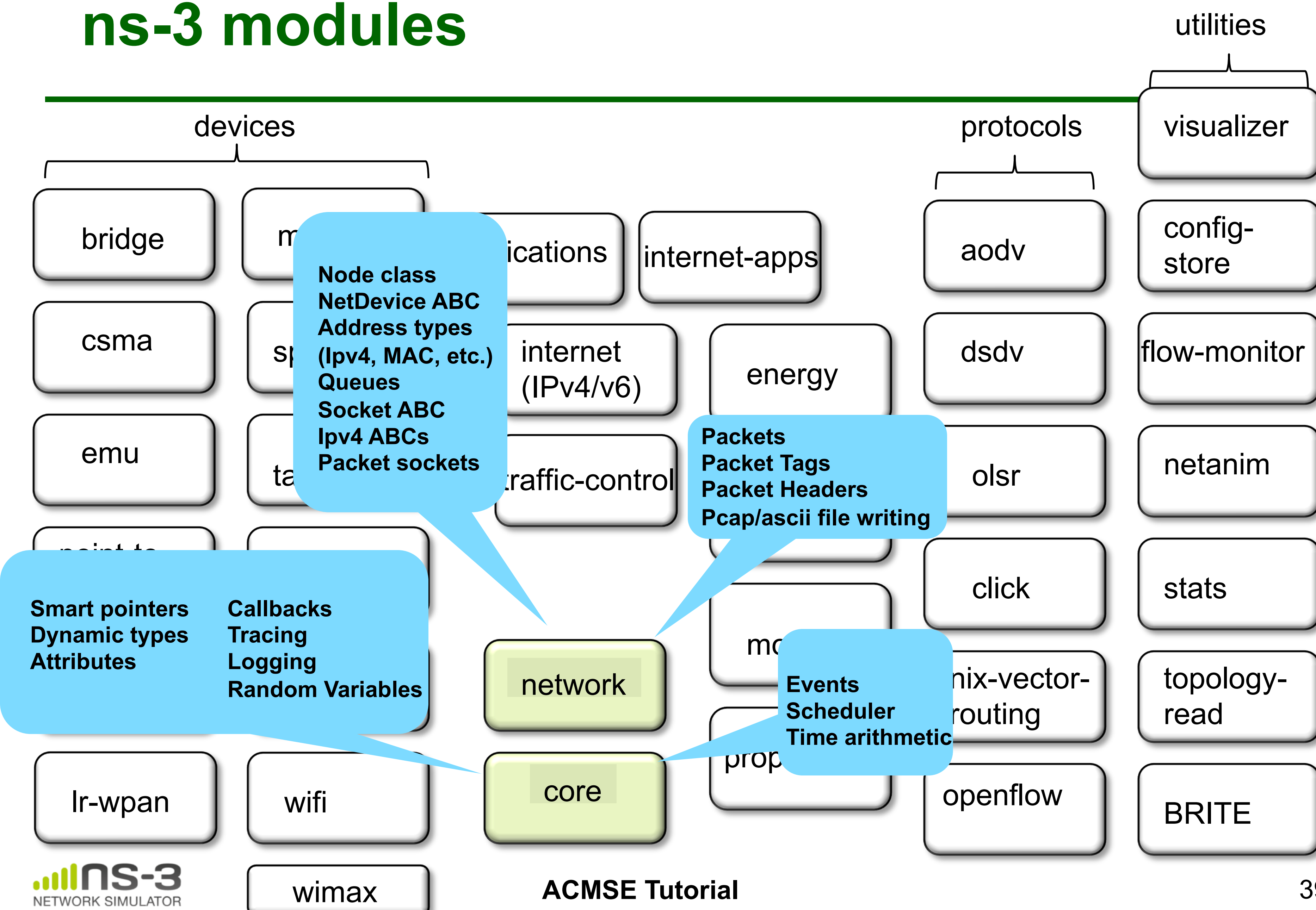
- We are trying to represent the operation of a (wireless) network within a C++ program
- We need a notion of virtual time and of events that occur at specified (virtual) times
- We need a data structure (scheduler) to hold all of these events in temporal order
- We need an object (simulator) to walk the list of events and execute them
- We can choose to ignore things that conceptually occur between our events of interest, focusing only on the (discrete) times with interesting events

Discrete-event simulation

- Simulation time moves in discrete jumps from event to event
- C++ functions schedule events to occur at specific simulation times
- A simulation scheduler orders the event execution
- Simulation::Run() executes a single-threaded event list
- Simulation stops at specific time or when events end



ns-3 modules



utilities
devices
protocols
visualizer
bridge
internet-apps
aodv
config-store
csma
internet (IPv4/v6)
energy
dsdv
flow-monitor
emu
traffic-control
olsr
netanim
click
stats
nix-vector-routing
topology-read
lr-wpan
wifi
openflow
BRITE
wimax
network
core
Node class
NetDevice ABC
Address types (Ipv4, MAC, etc.)
Queues
Socket ABC
Ipv4 ABCs
Packet sockets
Packets
Packet Tags
Packet Headers
Pcap/ascii file writing
Smart pointers
Dynamic types
Attributes
Callbacks
Tracing
Logging
Random Variables
Events
Scheduler
Time arithmetic