

ns-3 GSoC wrapup



The ns-3 project participated in the 2019 edition of Google Summer of Code with four student projects. The mentors have written brief summaries below of the projects and student accomplishments.

TCP Testing and Alignment

Student: Apoorva Bhargava

Mentors: Tom Henderson, Vivek Jain, Mohit Tahiliani

One of the major uses of ns-3 is to study transport protocol behavior, and in particular, TCP. ns-3 has several models of TCP, and was refactored in a previous GSoC project (by Natale Patriciello) to adopt an architecture similar to that of Linux TCP. However, the ns-3 TCP models, having been written to follow the Internet RFCs, had not been validated against and aligned with the comparable Linux implementations, which differ in some aspects from the RFCs, and also yield some different performance due to the use of integer arithmetic in the kernel. Some users of ns-3 wish to be able to use TCP models that correspond to Linux very closely. While Direct Code Execution (DCE) is available for some kernel variants (e.g. Linux kernel 4.4), it is easier at the moment for most users to install and use native ns-3 models of TCP.

Apoorva's project focused on creating variants of ns-3 TCP models that were validated against the Linux implementations. She used DCE to provide traces of the Linux implementation, and then created variants of the ns-3 models to match the Linux traces. She started by producing versions of NewReno (LinuxReno) and Proportional Rate Reduction (PRR) recovery, and validating and documenting the results. Along the way, she identified some bugs that were in the native ns-3 models, and proposed patches to fix the bugs, including one with TCP Selective Acknowledgments (SACK). At the conclusion of GSoC, she had begun to validate an earlier ns-3 model (written by Natale Patriciello) of TCP Cubic with hybrid slow start against Linux. Once validated and merged to ns-3, this model, along with other settings such as 'initial window 10', will be considered to become the new default model for ns-3, to match the default in Linux (most consumer operating systems are now moving to Cubic/Hystart as the default).

Integration of the 3GPP TR 38.901 channel model in the ns-3 spectrum modul

Student: Tommaso Zugno

Mentors: Natale Patriciello, Sandra Lagen, Biljana Bojovic

In recent years, ns-3 has become the reference platform for packet-level simulation in the mmWave domain, thanks to the initial release by NYU of the mmWave module, followed by developments done by the University of Padova and CTTC (a telecommunications research center). These developments led to two different modules: one that takes care of the most recent 3GPP standardization (NR), and the other that is the base for the most experimental features in the mmWave frequencies.

One of the community's wishes was that the two modules shared at least the details of the channel and the antenna modeling. However, the job of aligning the channel model to the ns-3 high standards was not easy, required theoretical knowledge, and it was not appealing for the agencies that usually fund research.

Tommaso's project in the context of Google Summer of Code focused on reworking the channel model class to pursue two main objectives. The first was to make it available to the entire ns-3 community, following the strict ns-3 coding and testing rules. The second was a complete refactoring to allow overcoming the limitations that the model had in the beginning. The resulting model is ready to be used, and includes pathloss, shadowing and fast fading, with online updates (so that it does not require pre-computed traces).

Once merged to ns-3, this work will be the base for any module that wants to work in the mmWave bands, as well as sub 6 GHz bands, in line with 3GPP recommendations for channel simulation modeling. We think that also WiFi and WiGig can be adapted to work with such a model, making comparison and coexistence tests more comfortable and fair than ever.

ns-3 GSoC wrapup



Framework for Studying Flow Completion Time Minimization for Data Center Networks

Student: Liangcheng Yu

Mentors: Dizhi Zhou, Mohit Tahiliani

Data center networks (DCN) have been a hot research topic in recent years. The number of services, such as storage, computing, content sharing, etc, offered by data centers is fast growing. Until this project, ns-3 did not have models to support some generic DCN algorithms. Liangcheng's project extended ns-3's modeling support into DCN by implementing three key components.

1) *Scheduling algorithms*: Shortest Job First and Multi-Level Feedback Queue

2) *General DNC topology generators*: BCube, FatTree, leafspine and DCell. Some of these generators extend those from an earlier DCTCP project: (<https://www.nsnam.org/wiki/GSOC2017Tcp>).

3) *Load balancing*: Extending the existing ns-3 option of per packet equal-cost multipath (ECMP) to other two-path selection algorithms; namely, flow-based ECMP and simple flowlet switching. These forwarding approaches allow the packets to be forwarded not only on a per-packet basis, but also at the level of flow entity or flowlet entity

The project's resulting models can help users to evaluate DCN proposals with ns-3 more effectively.

Improving the ns-3 AppStore and linking with bake

Student: Mishal Shah

Mentors: Ankit Deepak, Abhijith Anilkumar

The ns-3 AppStore project was created to support modularization of the ns-3 code base and federation of development activities, and the initial web frontend for the project was created and published as part of a 2017 GSoC project. The next step to the project was to link it with Bake build system, with an ambitious goal to install, update and publish modules from the CLI. However, this is not easy and is a long-term project which requires deeper insight into dependency handling and management of different modules.

Mishal's contributions during GSoC 2019 focused on two main goals:

1) Improve the existing AppStore site (user interface/user experience improvements)

2) Establish a link between the AppStore and Bake build system by opening up REST APIs and creating Bake commands to make use of these APIs

Three new commands were added to Bake:

1) *getconf*: To fetch the bakeconf file of the module

2) *search*: To search for a particular module in the AppStore

3) *install*: Fetch and install a module in the AppStore

Once Mishal's changes are deployed on the server and the Bake changes merge to the main-line repository, any user will be able to interact with the AppStore from the CLI, without switching screens!