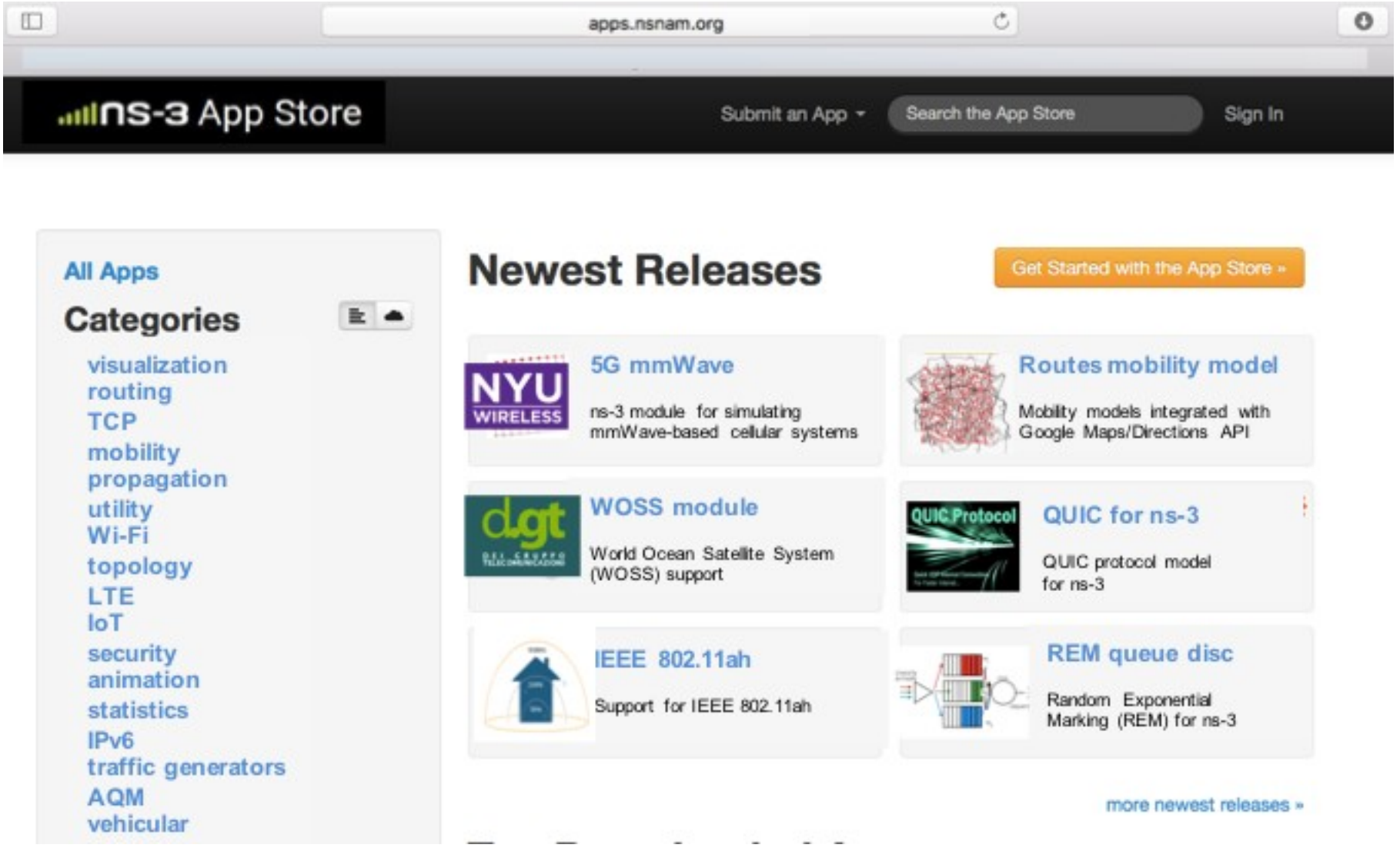
ns-3 app store overview

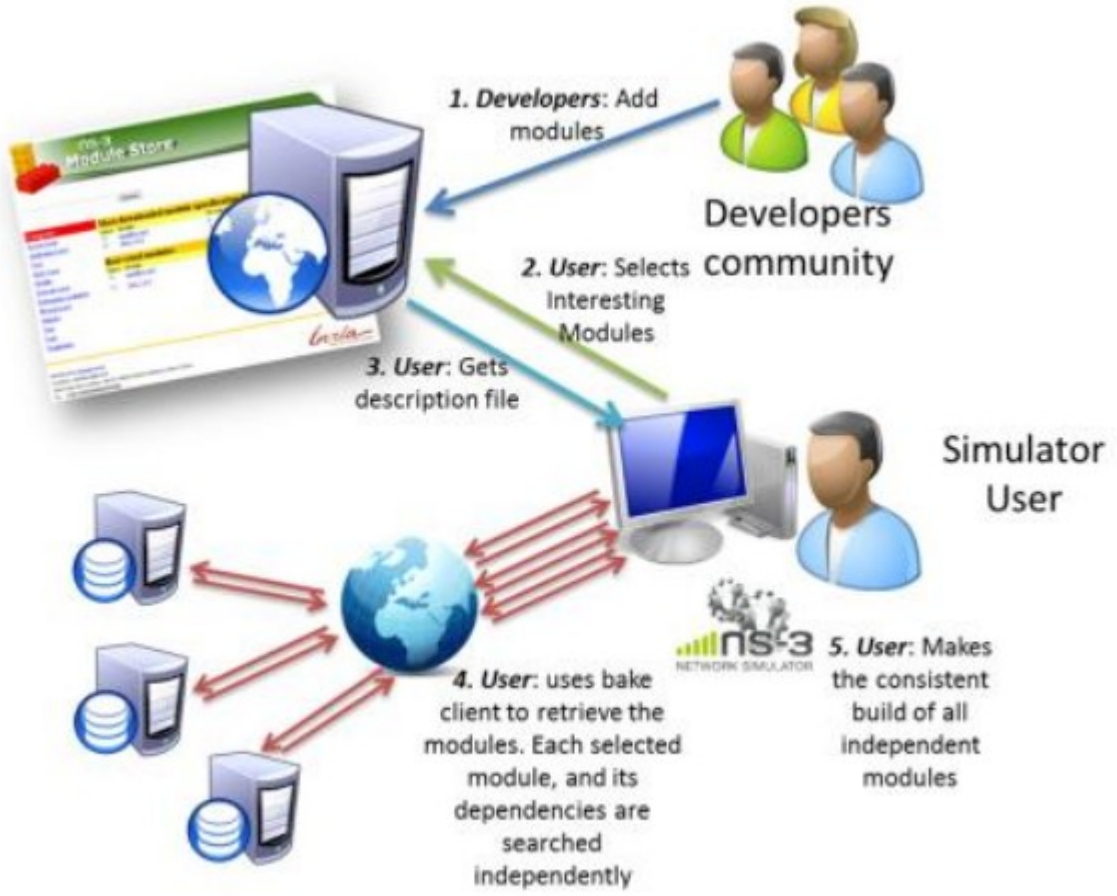
August 11, 2017



Terminology

- An ns-3 '**module**' is a collection of models built as a shared library and linked with ns-3 main programs
  - In ns-3, these are organized as directories under the 'src/' directory; e.g. 'olsr' and 'wifi'
- Starting with the ns-3.27 release, contributed, third-party modules may be provided and published to the ns-3 community
  - The '**app store**' is the front-end Web site that enables third-party module providers to publish and advertise their modules
  - The 'bake' build tool provides a command-line tool to download and build the module, and any dependencies, based on an XML module description called a '**modulespec**'

Conceptual overview



Benefits of a federated development environment:

1) Allow users to tailor their ns-3 installation (reduce build and linking of unnecessary modules)

2) Allow developers to make releases of their modules independently of the ns-3 release cycle

- Figure source: Daniel Camara, INRIA

Source code releases

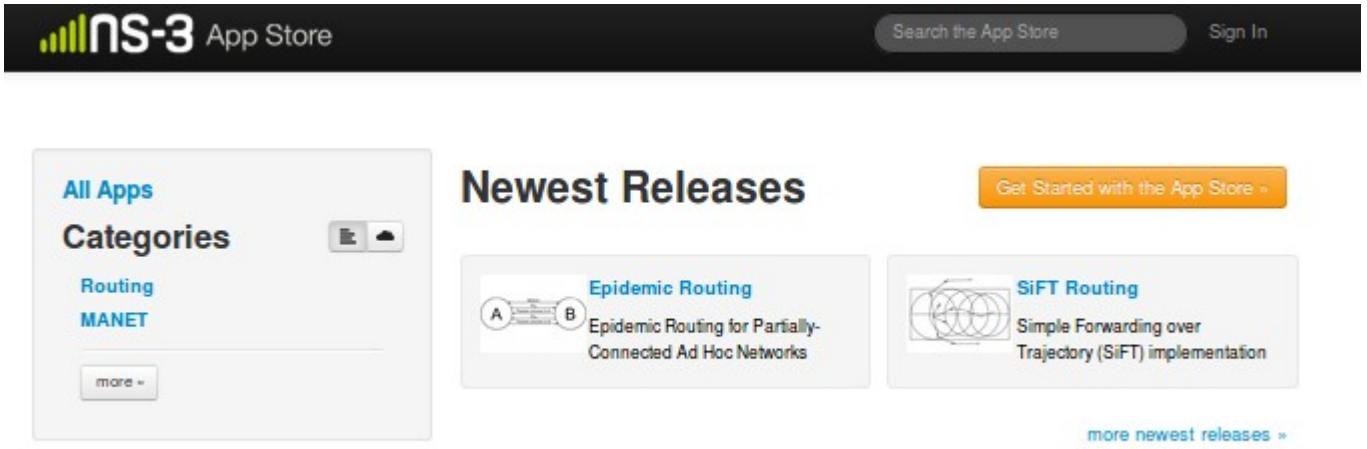
- ns-3 still relies on source archive releases
  - May move also to binary package releases in the future, but that is outside of this scope of this presentation
- Future 'ns-allinone-3.xx.tar.bz2' releases will contain the source code of fewer modules
  - However, the 'bake' tool will be populated with modulespec files tested to work with the release
  - bake will also be able to update its modulespecs
- We may create additional larger ns-allinone releases analogous to 'spins' of a desktop distro (outside the scope of this presentation)
  - e.g. 'ns-allinone-3.xx-wireless' or 'ns-allinone-3.xx-lte'

User workflow

1) User downloads an ns-3 release
2) User learns about additional modules of interest
   - by perusing the app store, or via the 'bake' CLI
3) User visits the app store to learn more about the module
   - prerequisites or installation instructions
   - how to report bugs/issues
   - where to find module documentation
   - etc.

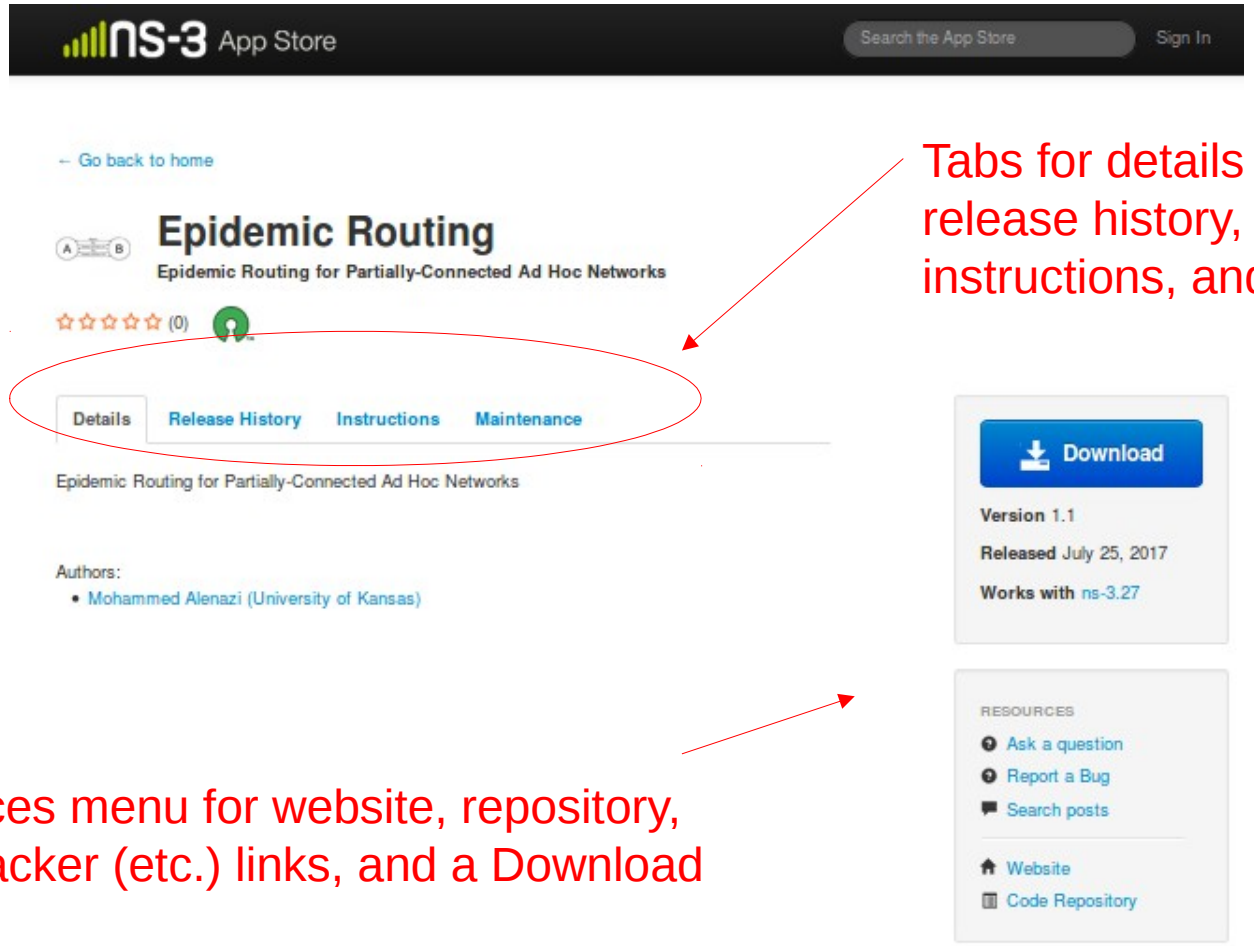
Example

- Front page lists available modules in order of newest to oldest (other search criteria available including by name, by tag (e.g. MANET), etc.)
- Visit http://ns-apps.ee.washington.edu for our prototype (work in progress)



Example (cont.)

- User interested in 'Epidemic Routing' can find additional information on the module page:



Tabs for details (screenshots), release history, installation instructions, and maintenance status

Resources menu for website, repository, issue tracker (etc.) links, and a Download button

Download options

Depending on the module, many options may exist

1) Use 'bake' to configure, download, and build the module
2) The author may have made available a downloadable archive, to be unpacked in the ns-3 contrib/ directory
3) The user may 'git clone' the source code into the ns-3 contrib/ directory

The module author will need to decide what he or she wants to be linked as the primary Download option

Release options

- Module authors may make releases, based on their own numbering system
  - e.g. epidemic-routing 1.0, 1.1, etc.
- A new modulespec should be generated and uploaded to the app store for each release
- bake should fetch and build the module version compatible with the user's ns-3 release

Module code organization

- Module authors need to provide their module in a form that is downloadable
- Module authors should use a publicly available service (github, gitlab, bitbucket) or web site (for a compressed archive)
- the repository should have the directory structure of an existing ns-3 module
  - e.g. 'src/', 'test/', 'doc/' etc. directories
- Module authors must provide a modulespec file for each release
- ns-3's 'create-module.py' will be modified to help guide prospective authors with a skeleton

modulespec example

key attributes are 'name', 'type' of 'ns-contrib', and min_version for minimum ns-3 version

```
<configuration>
  <modules>
    <module name="epidemic-routing" type="ns-contrib" min_version="ns-3.27">
      <source type="git">
        <attribute name="url" value="https://github.com/tomhenderson/epidemic-
routing.git"/>
        <attribute name="module_directory" value="epidemic-routing"/>
      </source>
      <build type="none">
      </build>
    </module>
  </modules>
</configuration>
```

module_directory attribute declares how the module will be named under the contrib/ directory

Installation options

- **Manual:** Based on information learned from app store Installation tab, user downloads or clones source code directly into ns-3 contrib/ or src/ directory, installs necessary prerequisites, etc.
- **Semi-automatic:** User downloads the modulespec XML file and places it in his or her bake/contrib directory, and uses bake to install it to ns-3
- **Automatic:** User uses bake at the command line to discover changes
  - e.g. 'bake update' will pull modulespecs from the app store, and then 'bake list available' will display available packages
  - This requires bake extensions and is outside the scope of the GSoC project

module version

- Modules should have a version number according to arbitrary convention
  - e.g. '0.1', '3.27.1', '0.1.rc', 'dev', 'devel'
  - each version should have its own modulespec file named 'modulename-version.xml'
  - each modulespec should be self-contained with all non-ns-3 prerequisites listed
- Module versions can be tied to ns-3 versions
  - e.g. '0.1' is compatible with ns-3.27
  - the 'min_version' attribute declares this