15..08.2018

NS-3 Installation and Development Tutorial

Erdem Tuna, GT ARC GmbH,Ernst Reuter Platz 7, 10587 Berlin, Germany

1.Installation (without MmWave-Module)

- The recommended OS for NS-3 environment is Linux/Unix distributed systems.
- First, all of the prerequisites must be installed as guided in *https://www.nsnam.org/wiki/Installation#Prerequisites* .
- Then, navigate to the directory that you want to install NS-3. The complete version of the NS-3 can be installed easily with the following commands:
    - `hg clone http://code.nsnam.org/ns-3-allinone`
    - `cd ns-3-allinone && ./download.py`

2.Configuration of NS-3 Environment

- Compilation environment of the NS-3 is *WAF* it must be configured. Navigate to *~/ns-3-allinone/ns-3-dev.* Then paste the following commands:
    - `./waf configure --build-profile=debug --enable-examples --enable-tests`
    - `./waf build`
- To see if everything is installed and configured correctly, paste following command:
    - `./waf --run hello-simulator`
- The output must be *"Hello Simulator"*.

3.Working with NS-3 MMWave-Module

- To work with MMWave-Module the following repository must be downloaded to the desired directory *https://github.com/nyuwireless-unipd/ns3-mmwave.git* .
- Then the procedures written in Section-2 must be followed by navigating to *~/ns3-mmwave-new-handover.*

4.Configuration of NetAnim

- Go to directory of */netanim.* Then paste the following codes in the given order:
    - `make clean`
    - `qmake NetAnim.pro`
    - `make`
- After the installation is done, animator can be launched in the /netanim directory with the following command:
    - `./NetAnim`
- Generate an *.xml file to observe mobility of the created topology.*

4.Printing Custom Traces to .txt File

- To print the custom traces, first the related variable must be found in respective ".cc" class file.
- Once the variable is found, find the method that the variable is in.
- Go to the ".h" file of the same class.
- If the method is private and constant, declare the following variables in the private section of the header file.
    - mutable std::ofstream m_mmWaveOutFile;
    - mutable std::string m_mmWaveOutputFilename;
    - If the method is not constant," mutable" declaration could be removed.
- Then in the ".cc" file find the ClassName::ClassName() method. Enter the following line inside the method.
    - m_mmWaveOutputFilename = "UlDlBfGain.txt";
- Find the ClassName::~ClassName() method and enter the folowing lines inside the method.
    - if(m_mmWaveOutFile.is_open())
      {
        m_mmWaveOutFile.close();
      }
- Then find again the related method in ".cc" file and type
    - if (!m_mmWaveOutFile.is_open ())
      {
        m_mmWaveOutFile.open(m_mmWaveOutputFilename.c_str());
      }
- Lastly, below the above codes type,
    - m_mmWaveOutFile << respectiveVariable << std::endl;