Network Simulator NS-2

N

Mubashir Husain Rehmani
COMSATS Institute of Information Technology, Wah Cantt, Pakistan

Yasir Saleem
Sunway University, Malaysia

INTRODUCTION

Simulation is an imitation of a real world process and simulators are used to provide such an imitation to model the real world. Like other fields of sciences and technology, simulators are also used in the field of telecommunications and computer networking. Researchers and scientists validate their algorithms and protocols using simulators, designed to simulate different kind of networks before their actual deployment. Because verification by actual implementation is too much expensive, so it is appropriate to first verify and fine tune algorithms and protocols using simulators, which provide imitation of real world environment, and after verification, they can be implemented in actual. Moreover, these simulators are also used in educational purposes for the better understanding of underlying network behavior. Many simulators are available on which algorithms and protocols can be developed and their performance can be evaluated by using different techniques and metrics. Some well-known simulators are NS-2 (Network Simulator, 2014), OMNet++ (OMNET++, 2014), QualNet (QualNet, 2014), Opnet (Opnet, 2014), Glomosim (Glomosim, 2014). Among them, NS-2, OMNet++ and Glomosim are open source simulators, while QualNet and Opnet are commercial simulators. The most widely accepted and used simulator among researchers is NS-2, therefore, we focused NS-2 simulator and shall discuss about it in rest of the article. Thus accordingly, the main objective of this article is to provide an overview of Network Simulator NS-2, which has been widely used to simulate vast variety of wireless, telecom, and computer networks. This overview includes background on NS-2, fundamentals and simulation process in NS-2 together with some advance topics as well as differences between NS-2 and NS-3.

BACKGROUND

To model a real world process or system, simulators have been used. In telecommunication networks and computer networking, researchers use simulators to validate their protocols and algorithms. The most widely used simulator in the networking research community is Network Simulator NS-2. The network simulator NS-2 is an open source discrete event simulator which was incepted in 1989, in which the core backend libraries are developed in C++ programming language. NS-2 was further supported by Defense Advanced Research Project Agency (DARPA), USA under the VINT project. It is also used for education purposes for better understanding of the underlying network behavior. Additionally, it is also used for validating new and existing network related algorithms and protocols. These networks range from ZigBee, LANs, IEEE 802.11 to Vehicular Ad-Hoc Networks. Also, it is worth noted that new simulators or support in existing simulators are required whenever new technology is introduced. For example, when Cognitive Radio Network (CRN) was introduced, NS-2 existed at that time. Thus in order to simulate CRN, either a new simulator or CRN support in existing simulators is required. Since NS-2 is an open source simulator, therefore the community has contributed a lot for providing support of new technologies in NS-2 including CRN. While on the other hand, in commercialized simulators, only the company can provide support for new technologies and most of the time, they take extra charges for such support. Thus, this is one of the foremost benefits of NS-2 as being open source besides its widely adoption among the researchers.

For instance, Fall et al. (2005) provide a comprehensive NS-2 manual under VINT project, which contains complete documentation of NS-2. Other than

DOI: 10.4018/978-1-4666-5888-2.ch615

this, Issariyakul et al., (2009; 2012) wrote two books on NS-2. These books contain information on NS-2 installation, running simple examples, modifying existing and creating new modules in a comprehensive manner. There is also a tutorial on NS-2 provided by Chung et al. (2002). This tutorial is written in a very simpler way and is specially designed for novice users. Liu (2002) worked under VINT project and provided a document for understanding the implementation of IEEE 802.11 MAC protocol in NS-2. This document covers technical aspects and is very useful for the users working on IEEE 802.11 MAC protocol in order to understand its technical flow. Rehmani et al., (2010) provided another tutorial for understanding implementation of AODV (Ad hoc On-Demand Distance Vector) routing protocol for wireless networks in NS-2. This AODV routing protocol is by default implemented in NS-2 and most of the routing protocols take motivation from AODV. Thus, in order to design a new routing protocol, it is imperative to first understand existing routing protocols in NS-2, and for this purpose, this tutorial is very useful and beneficial. Kubinidze et al. (2006) provide basic concepts and evolution of NS-2 by discussing shortcomings, potential development and enhancement schemes for NS-2.

The network simulator NS-2 provides support for the simulation of routing, MAC protocols, and transport layer protocols over wired and wireless networks. Additionally, it also provides support to satellite networks. Since NS-2 is an open source simulator, therefore there is lot of contribution from research community to provide additional functionalities in NS-2 which are not supported by default. In particular, (Baldo et al., 2002; Baldo et al., 2010) provide NS-2, the support of embedded engine for handling cross-layer messages and also provide support for multiple interfaces within each layer of protocol stack. Calvo et al. (2007) provide mechanism for adding multiple interfaces support in NS-2 by claiming that existing information is either not complete or is very specific, while they provided a very generic solution by giving complete flexibility to users during scenarios configuration. Betancur et al. (2006) propose PHY layer and propagation model for IEEE802.16 standard (WiMAX) and also describe implementation details in NS-2 which is a good contribution to NS-2 for providing PHY and propagation layer support for WiMAX networks. Khan et al., (2010) provide a tutorial for broadcasting packets over multiple channels in multi-interface network in NS-2 by using CRCN patch. The discussion on CRCN patch will be provided in advance topics in NS-2 in the section entitled, *Advanced Topics in NS-2*.

In this article, our goal is to discuss in detail about the fundamentals of network simulator NS-2 and its basic functionality.

NETWORK SIMULATOR NS-2 FUNDAMENTALS

In this section of the article, we discuss the fundamentals of Network Simulator NS-2 which include:

Basic Architecture

The basic architecture of NS-2 is simple. It consists of TCL language which provides us to write scenario file where we describe all the simulations parameters. This TCL script then links to the C++ core code of NS-2. It has main files of protocols and it is composed of several hundreds of classes of different objects. These C++ objects are linked with OTcl language objects. It means that the core of NS-2 is composed of two languages, i.e., C++ and OTcl. While TCL language is used to write the simulation scenario and acts as an interface between the user and the core NS-2 code. After the execution of simulation, NS-2 provides the trace file. This trace file can be analyzed by using AWK and Perl languages. Moreover, the results can also be visualized through Trace Graph, Network Animator (NAM) and GNU plot.

Installation

NS-2 is an open source program freely available on the Internet. We can run NS-2 on Windows and Linux platforms. To run NS-2 on Windows, we need to install Cygwin software. Cygwin is a virtual machine through which we can get the Linux environment over Windows Operating System. NS-2 can be installed by pieces, however, for the beginners, the all-in-one package of NS-2 will be better. This all-in-one package of NS-2