Simulation of Mobile IPv6 Using OMNeT++ Simulator

Hayder R. Hamandi[1] **Dr. Emad H. Al-Hemiary[2]**

[1]Department of Information Technology, Technical College of Management/Baghdad,

Foundation of Technical Education, Baghdad

[2]Department of Internet Engineering, College of Information Engineering, Nahrain University, Baghdad

email: hyder_hamandi@yahoo.com **emad@ie-nahrain.org**

Received: 24/05 /2011

Accepted: 22/7 /2013

Abstract **–** As mobile computing is more and more widespread, mobility support for Internet devices becomes very important. Mobile IPv6 (MIPv6) is a promising technology that handles the mobility management and provides the seamless mobile communications. It is expected that MIPv6, as a standard for mobile communication, will open the Mobile Internet Age. In this paper, we will discuss the basic principles of MIPv6 operation by means of handoff process and tunneling mechanism. We will also show how to simulate a proposed MIPv6 network scenario. This work is based on a specific simulator named OMNeT++ which includes simulation packages. Our simulation results include the throughput of each router in the proposed MIPv6 network; it also includes the handoff latency for each mobile node in the proposed MIPv6 network.

Keywords – Mobile IPv6, OMNeT++, INET Framework, Home Agent, Foreign Agent.

.

1. Introduction

Mobile IPv6 (MIPv6) is a protocol developed as a subset of Internet Protocol version 6 (IPv6) to support mobile connections. MIPv6 is an update of the Internet Engineering Task Force (IETF) Mobile IP standard designed to authenticate mobile devices (known as mobile nodes) using IPv6 addresses [1][2].

In traditional IP routing, IP addresses represent a topology. Routing mechanisms rely on the assumption that each network node will always have the same point of attachment to the Internet, and that each node's IP address identifies the network link where it is connected. In this routing scheme, if you disconnect a mobile device from the Internet and want to reconnect through a different network, you have to configure the device with a new IP address, and the appropriate net mask and default router. Otherwise, routing protocols have no means of delivering datagrams (packets), because the device's network address does not contain the necessary information about the node's network point of attachment to the Internet. MIPv6 allows a mobile node (MN) to transparently maintain connections while moving from one subnet to another. Each device is identified by its home address although it may be connected to through another network [3].

This paper is organized as follows: Section 1 gives an introduction to the MIPv6. Section 2 summarizes the OMNeT++ simulation framework along with all the necessary environments and modules. The simulation of an MIPv6 network is discussed in section 3. Section 4 gives the conclusions obtained from the results of simulation.

2. The OMNET++ Simulation Framework

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queuing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects.

OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, alternative programming languages (Java, C#), database integration, System C integration, and several other functions. OMNeT++ is a discrete event simulation environment for modeling communication networks, IT systems, queuing networks, hardware architectures, multiprocessors, distributed or parallel processes and other systems [4][5].

OMNeT++ aspires to be the optimal solution between open-source, research-oriented simulators (like NS-2 [6]) and the high-priced commercial softwares (like OPNET [7]). Therefore OMNeT++ is public source, and under the Academic Public License it is free to use for non-profit aims. OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language called NEtwork Description (NED). Reusability of models comes for free. All the modules and files of the OMNeT++ network simulator can be configured using GUI view or source view

with no restrictions. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

Figure 1 shows a screenshot of OMNeT++ simulation environment.

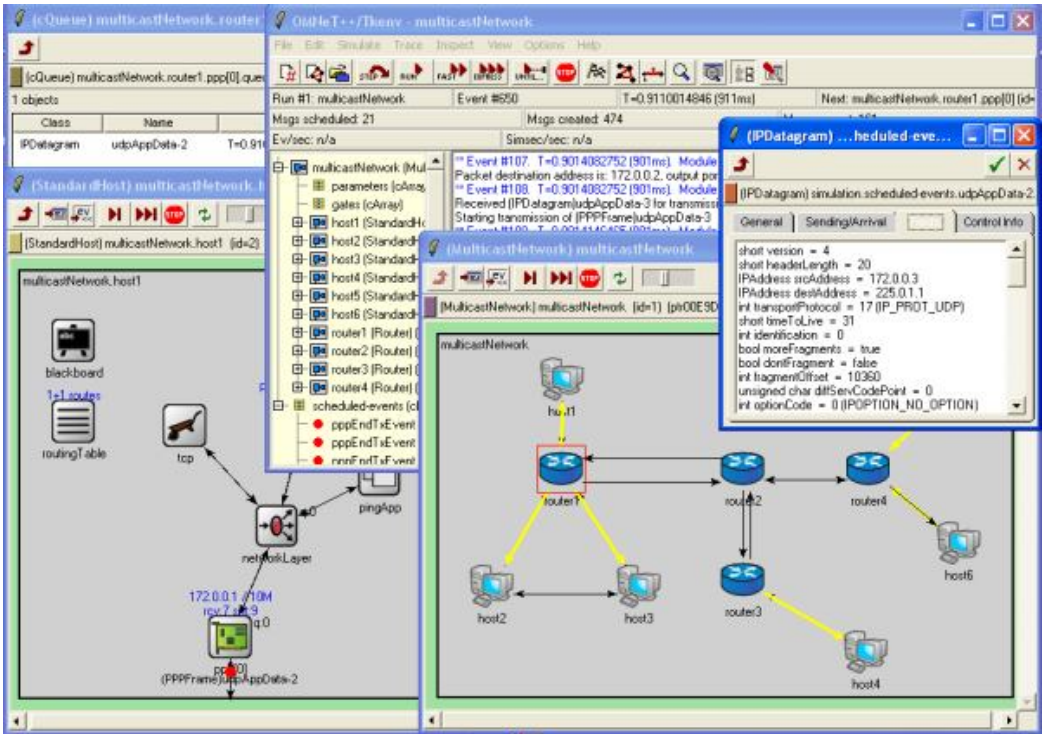

Figure 1. OMNeT++ Simulation Environment

Modules communicate with messages thus message sending and receiving are the most frequent tasks in simple modules, see Figure 2. Messages contain common attributes (like timestamps) and also arbitrary ones (i.e. any other kind of user data). Simple modules typically use gates (input and output interfaces of modules that can be linked with connections) for sending messages, but direct send to destination modules (using an invocation from the OMNeT++ simulation kernel) is possible as well. OMNeT++ messages can be easily defined by specifying the fields and other possible message content in (.msg) files and by letting OMNeT++ to take care of creating the necessary C++ classes from the (.msg) definitions [1].

A special topology description language called NED (NEtwork Description) is applicable for users to define the structure of simulation models (the modules and their interconnection) in OMNeT++. A typical (.ned) description file consists of simple module declarations (i.e. description of the module's interfaces), compound module definitions (i.e. declaration of the module's external interfaces and definition of submodules and their interconnection) and network definitions (i.e. compound modules that are self-containing simulation models). In this way model behavior and model

topology are separated: behavior is defined in C++ code, while topology is determined by the NED language.

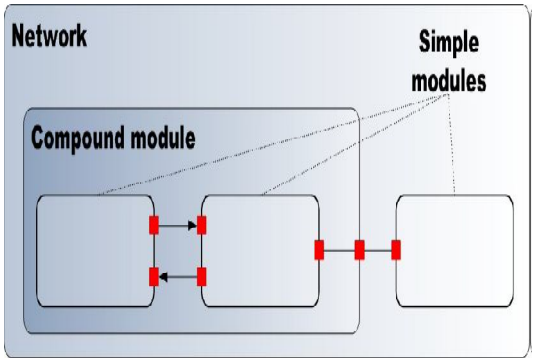

Figure 2. Hierarchy of modules in OMNeT++

Simulation parameters (i.e. initial parameters of simulation runs which are independent both from the C++ and the NED codes) are specified in (.ini) files. Separating initial inputs in this way enable users to run simulations for each one of the interested parameter combination without modifying the existing codes. OMNeT++ uses the configuration (omnetpp.ini) initialization file to start the simulation of the network. The work in this paper is based on OMNeT++ package named as the INET Framework [2].

3. The Proposed MIPv6 Simulation Scenario

When connecting through a foreign network, a mobile device sends its location information to a home agent (HA), which intercepts packets intended for the device and tunnels them to the current location shown in Figure 3. Note that the red line indicates the bidirectional tunnel between the MN and the HA[8][9].

We have designed and simulated an MIPv6 network using the OMNeT++ network simulator. This network consist of three Mobile Nodes (MN[0], MN[1], and MN[2]) moving in random directions, thereby changing their points of attachment to the five Correspondent Nodes (CN[0] through CN[4]) as shown in Figure 4. We have used Ethernet connection to connect the wired nodes, and we have designed the wireless channel of the network to operate with a frequency of 2.4 GHz, and 11 Mbps which is the 802.11 wireless channel standard.

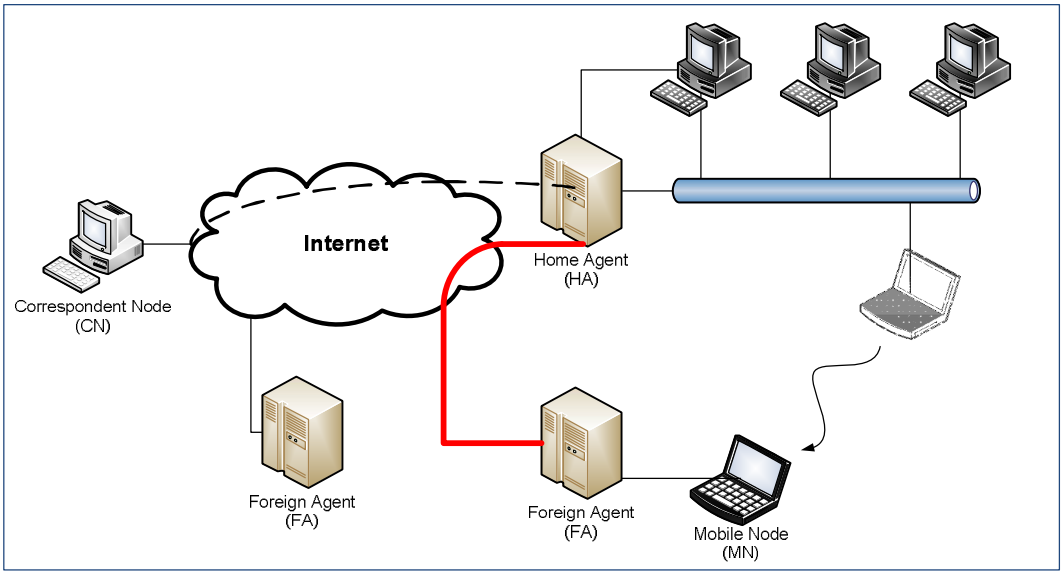

Figure 3. Simple MIPv6 Network Scenario

Each mobile node will perform multiple handoff operations between the 3 APs, AP_Home is the home network of the three mobile nodes. We are assuming that all the mobile nodes were connected to the Home_Agent router, and after that each device changed its position randomly. Therefore AP_1 and AP_2 are acting as foreign agents and are tunneling the packets from the MNs to the home agent via the router R_3.

4. Results and Conclusion

After simulating the network above, we have gathered the results in the analysis file and organized them by creating data sets for vector values of the same type, for example Figure 5 shows the number of Bytes/sec sent from the three routers which is the throughput for each router.

The handoff delays were automatically calculated by the OMNeT++ simulator as shown in Figure 6. This was done by subtracting the association request time from the association confirm time as shown in the following formula:

L2_HO_Delay = Association Confirm Time – Association Request Time

The association request / confirm time are recorded in the scalar result files. Also it is possible to calculate the handoff delay using the authentication request / confirm time, which is the time required for the MN to authenticate the new AP during the handoff operation, these values are also recorded in the scalar file (results.sca).

From the results we conclude that the handoff delay of each mobile node depends mainly on the number of handoff operations and the amount of data bytes to be sent by each node. It also depends on the number of authentication steps for each handoff operation and in our case this number was fixed to 4 steps.

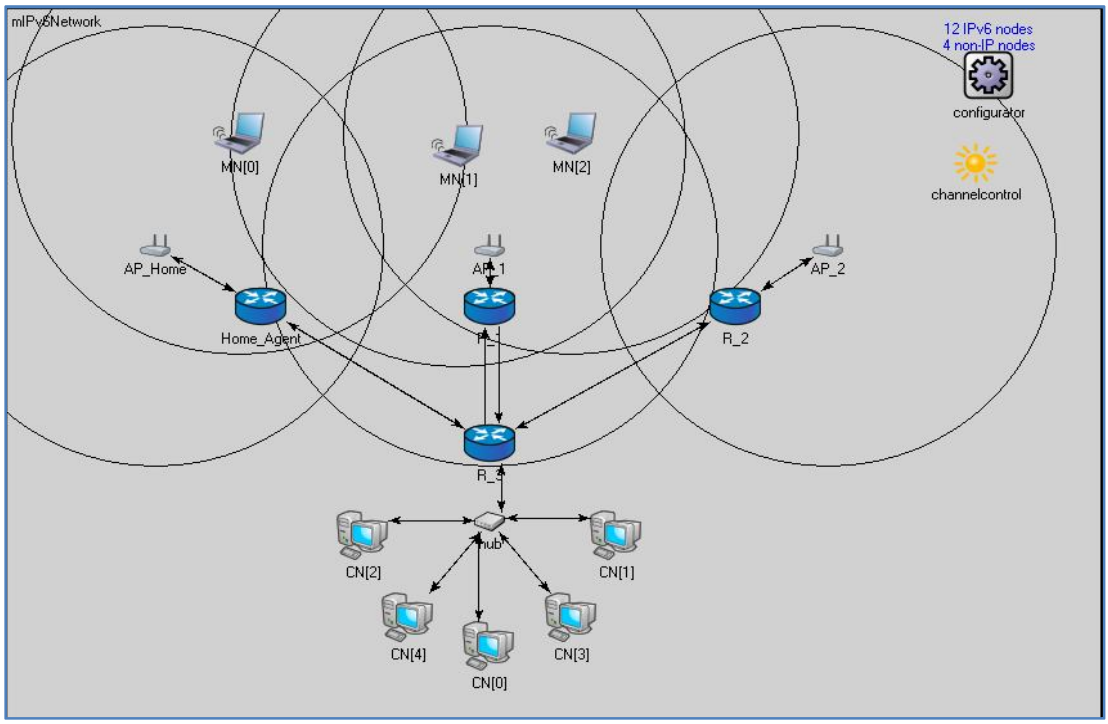

Figure 4. MIPv6 Network Simulation Environment in OMNeT++

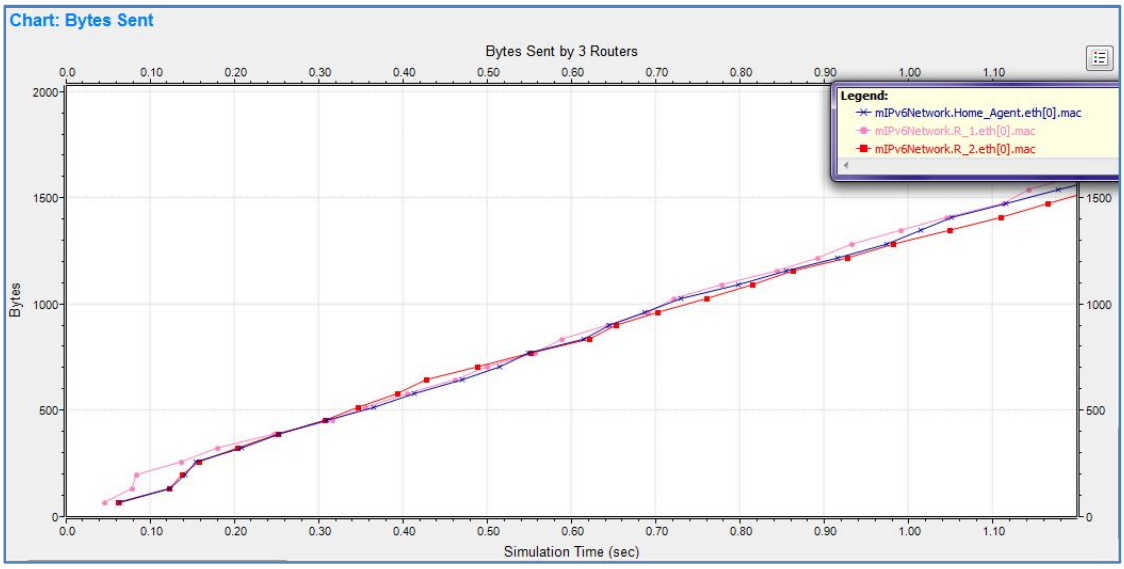

Figure 5. Throughputs of the 3 Routers

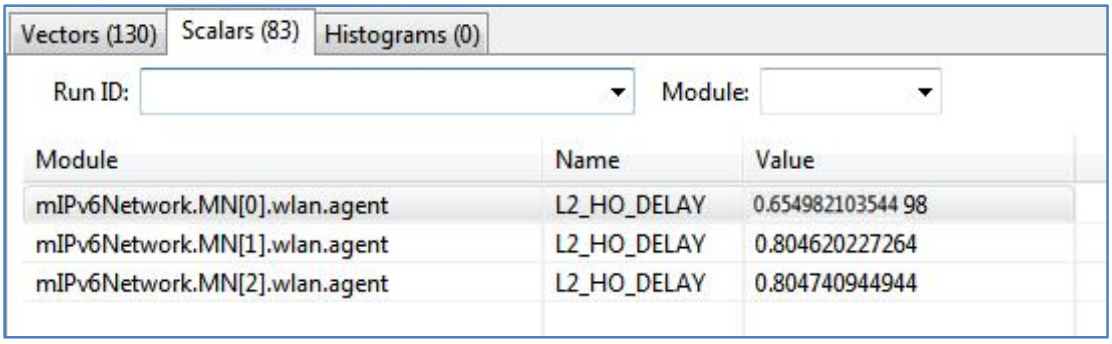

Figure 6. Layer 2 Handoff Delays for each Mobile Node (MN)

References

[1] OMNeT++ Network Simulation Framework, Official homepage: http://omnetpp.org

[2] The Mobility Framework for OMNeT++, Official homepage: http://omnetpp.org/doc/INET/neddoc/index.html

[3] D. Johnson, C. Perkins, J. Arkko: "Mobility Support in IPv6",IETF RFC 3775, (http://www.ietf.org/rfc/rfc3775.txt). June 2004.

[4] J. Arkko, V. Devarapalli, F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", IETF RFC 3776, (http://www.ietf.org/rfcs/rfc3776.txt). May 2005.

[5] The Second International Workshop on Mobile IPv6 and Network Based Localized Mobility Management (MobiWorld2010), Official homepage: http://isyou.hosting.paran.com/mobiworld10/. January 2010.

[6] The Network Simulator ns-2, Official homepage:http://nsnam.isi.edu/nsnam/index.php/main_page

[7] OPNET Technologies Inc., Official homepage:http://www.opnet.com

[8] ZhigangKan, Dongemei Zhang, Runtong Zhang, Jian Ma: "QoS in Mobile IPv6", Nokia China R&D Center, 2005

[9] Guillaume Valadon: "Mobile IPv6: Architectures and Protocols", Ph.D Thesis, June 2008.