HOMOGENEOUS MULTI-INTERFACE MOBILE NODE SUPPORT IN NS2

SUHAS J. PATIL[1] & B. R. CHANDAVARKAR[2]

[1,2]Department of Computer Science and Engg., National Institute of Technology Karnataka, Surathkal, Mangalore, India
E-mail : mailtosuhas@gmail.com & brcnitk@gmail.com

Abstract – NS2 is a widely used, open source tool for network simulation. A Mobile Node (MN) in NS2 by default provides only a single Wi-Fi interface. It makes difficult for users to simulate the scenario where a mobile node is connected to multiple networks through different interfaces at the same time. Some projects have been done to implement multiple Wi-Fi interfaces but according to our view they have some limitations. This paper presents the implementation of mobile nodes in NS2 with multiple Wi-Fi interfaces and multiple WiMAX interfaces trying to overcome those limitations.

Keywords- *NS2; Multiple Interfaces*

I. INTRODUCTION

The 4th Generation (4G)[1] is one of the emerging technologies, which incorporates different networks such as Wi-Fi, WiMAX, UMTS etc. In the next generation of wireless networks a Mobile Node (MN) equipped with multimedia-enabled wireless devices will be expected to use real-time and non-real time applications at any time anywhere from diverse networks. Furthermore MNs are expected to conduct multiple communications sessions at the same time (e.g. voice, video or downloading). A MN should be enabled with multiple interfaces in order to connect to different networks, roam freely across the different networks and to be always best connected (ABC).

Network Simulator 2 (NS2)[2] is one of the widely used simulators by the researchers in computer communication networks to simulate and study network performance of new technologies for communication. A MN in NS2 by default does not have multiple interface support which makes difficult to simulate 4G scenarios. In order to help simulate such scenarios in NS2 multiple interfaces support for a MN must be provided.

The objective is to provide multiple interface support to a MN in NS2. NS2 is chosen as it is an open-source, widely used simulator designed specifically for research in computer communication networks. To investigate network performance, researchers can simply use an easy-to-use scripting language to configure a network, and observe results generated. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e. a frontend).

II. RELATED WORK

Some projects have been implemented for providing multiple interface support to a MN in NS2. But according to our view they have some drawbacks or some limitations. All the projects targeted only Wi-Fi interfaces as support for Wi-Fi is by default in the NS2. We are going to discuss some important projects done so far in this area.

A. TENS

TENS [3] project was done at the Indian Institute of Technology of Kanpur, India. It aimed to improve the ns-2.1b9a implementation of the WLAN (IEEE 802.11) and to add multiple interface support. This model is based on multiplexing the physical layer. Most of the implementation was done in C++. They used a loop for add-interface method in ns-mobilenode.tcl.

B. HYACINTH

Hyacinth [4] project was done at the State University of New York in ns-2.1b9a. Its main drawback is that it has quite a static configuration so that all the nodes within the scenario have 5 interfaces. Another drawback is that, a static (manual configurable) routing agent was implemented to use this multi-interface capability. The existing routing agents don't work with multi-interface MN.

C. NS-2 Multiple Interface Support

This project [5] at the University of Cantabria implemented the multi-interface MN in NS 2.33. In this project they multiplexed link layers and physical layers of a MN. They also modified routing agent to use the multi-interface facility. MN can have variable number of interfaces and routing algorithms are not static. In this user has to call add-channel procedure as many times as the number of interfaces, instead of specifying the number of interfaces and adding those many channels automatically. Its patch can be found at[6]

All the projects explained above implemented only Wi-Fi interfaces the as support for Wi-Fi is by default in the NS2. Some projects have implemented fixed number of interfaces or have manual configurable routing agent. Here we have implemented multiple interfaces for Wi-Fi as well as WiMAX so that a MN can have variable umber of Wi-Fi or WiMAX interfaces. User just needs to specify the number of interfaces and the channels to be attached to each interface.

III. ARCHITECTURE

A. Single Interface MN Architecture in NS2

Figure 1 shows the original architecture of the MN [7] in NS2. It consists of a chain of different modules which emulates the different entities of protocol stack that any mobile host would have in the real life. These entities are Link Layer, MAC Protocol, ARP, Interface Queue, and Network Interface. All the entities are connected to the same shared wireless channel. In addition, the Propagation Model is used which simulates the effect of the real wireless channels on the transmitted signal.

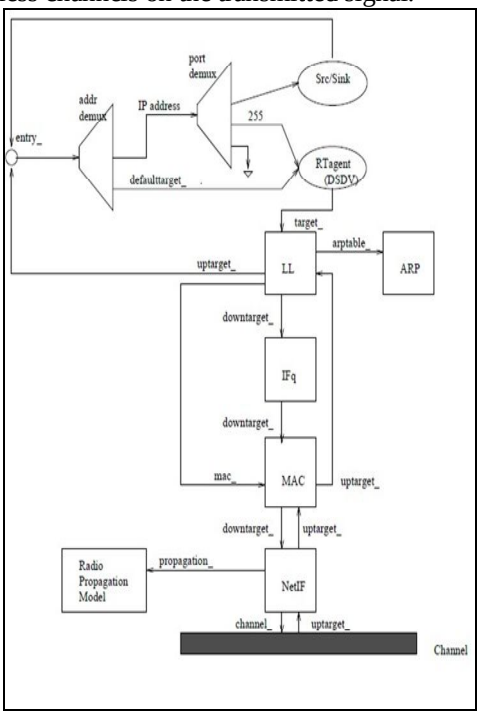

Fig. 1 : Single Interface Mobile Node Architecture

B. Multiple Homogeneous Interface MN Architecture in NS2

Figure 2 presents the architecture of the modified MN with multiple interface support. After routing agent, chain of all the modules of link layer and physical layer are multiplexed as many times as the number of interfaces a node has. Incoming packets arrive through the corresponding channel and travel through the different entities in ascending order to the Link Layer, which is connected to the same common point, i.e. routing agent. For outgoing traffic, routing agent is modified such that it can select the appropriate interface through which packets can be transmitted. This architecture is inspired from the architecture in [5] and we have extended it for multiple WiMAX interface as well.

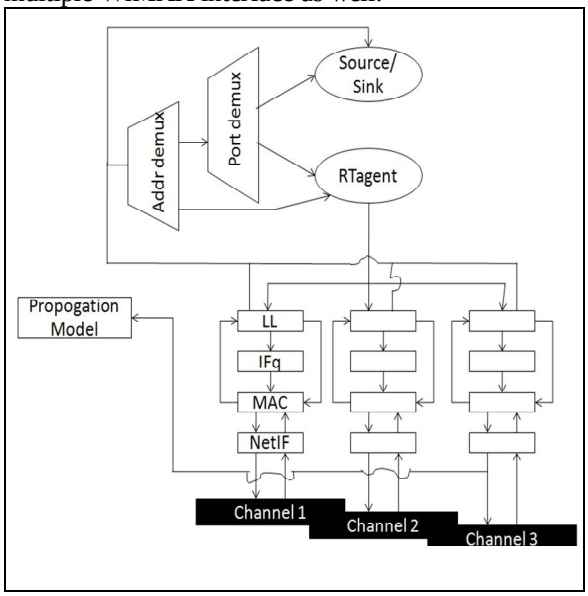

Fig. 2 : Multiple Interface Mobile Node Architecture

IV. IMPLEMENTATION

Here we have implemented multiple interfaces MN in NS 2.30. The NS version 2.30 is chosen because support for WiMAX interface is provided for this version by NDSL[8]. Support for WiMAX is not by default in NS2. To simulate WiMAX network scenario, supporting module for WiMAX provided by any third party must be added to NS2. Before starting implementation we added the WiMAX module in NS 2.30 provided by NDSL. While implementing care is taken so that the product is backward compatible. As NS2 has two languages tcl and C++ modifications are done at both levels.

A. Tcl Implementation

Most of the modifications are done in ns-lib.tcl and ns-mobilenode.tcl. First a variable numif is added in ns-lib.tcl to store the number of interfaces a MN has. A new procedure multiIFnode is introduced which creates the MN with specified number of interfaces. The parameters are the number of interfaces and channels to be assigned to those interfaces. Changes are made to node-config procedure so that array of channels is created to store the channels assigned to each interfaces. The channels are stored in this array when multiIFnode procedure is called from scenario script. Procedure create-wireless-node is modified so that add-interface procedure is called as many times as the number of interfaces.

Changes to the add-target procedure of ns-mobilenode.tcl are made to create as many interface queues as the number of interfaces. Procedure add-target-rtagent is also modified so that it can attach routing agent with the corresponding link layer

entities. Procedure add-interface is modified so that an array of ARP tables is created where one ARP table is dedicated for one interface. The above changes executes only when number of interfaces are more than one. For a MN with one interface above changes will not execute so that the backward compatibility is maintained.

B. C++ Implementation

Mobilenode.cc is modified such that it can correctly associate each interface with appropriate channel. The simulator controls the nodes which are connected to a channel by means of a list which is managed using two arrays of pointers. Channel.cc is modified such that it can manage the nodes attached to the channel. Finally changes are made to mac-802_11.cc and mac-802_16.cc files. The recv procedure of Mac802_11 and Mac802_16 class is modified to identify the interface through which a packet is received. This information is required for routing agent.

Now multiple interface support is added but to use that feature routing agent must be modified. So routingAgent.cc is modified to include array of targetlist and ifqueuelist. To broadcast whenever required, a packet is sent through all the interfaces. To send unicast packets an index, which is stored in routing table entry by routing agent is used to forward the packet to appropriate target.

Routing protocols (AODV, DSDV) are modified to make use of multiple interface support. Loop is used to broadcast the packets in sendRequest, sendError and sendHello methods of the aodv.cc and dsdv.cc files. Unicast packets are sent through the appropriate interface by using index in sendReply and forward methods. Finally routing table is modified to include index in the routing table entry. Procedure rt_update is also modified to include interface index as an extra parameter in the routing table entry.

V. SIMULATION SCRIPT

Before creating a MN with multiple interfaces as many channels must be created as the interfaces a mobile node is supposed to have. After that a node is created by calling multiIFnode procedure. The parameters to this method are the number of interfaces and the channels to be associated with each interface. Before creating the multiple interface nodes, it must be configured by calling node-config command.

To create a mobile node with Wi-Fi interfaces mac value must be mac/802_11 and for a MN with WiMAX interfaces mac value must be mac/802_16. So to create a MN with 2 interfaces following tcl script is used.

```
set ns [new Simulator]
for {set i 0} {$i < 2} {incr i} {
   set chan_($i) [new $val(chan)]
}
set mifnode [$ns multiIFnode 2 chan_(0) chan_(1)]
```

VI. RESULTS

To see the working of the MN with multiple interfaces we simulated scenarios where a MN has 2 Wi-Fi interfaces and a TCP agent is attached to it. It is observed that the MN with 2 interfaces works fine. Same thing is repeated for WiMAX interfaces and it worked fine too.

Table 1. shows the total number of packets processed during the given duration for Wi-Fi as well as WiMAX interfaces. It shows the number of packets sent by sender and number of packets received by receiver. It also shows number of packets dropped during the communication.

TABLE I. NUMBER OF SENT, RECEIVED AND DROPPED PACKETS

Time (sec)	Wi-Fi			WiMAX		
	Sent	***Received***	***Dropped***	***Sent***	***Received***	***Dropped***
5	150	140	10	90	80	10
10	334	324	10	151	137	14
15	539	529	10	191	169	22
20	742	730	12	221	194	27
25	945	934	11	261	228	33
30	1148	1141	7	289	253	36
35	1352	1344	8	316	277	39
40	1559	1546	13	355	310	45
45	1762	1752	10	395	343	52
50	1967	1954	13	426	370	56
55	2211	2100	111	460	400	60
60	2375	2363	12	504	436	68

Fig. 3 shows the graph of throughput (in kbps) against the duration of simulation (in seconds). It is observed that throughput gained in case of Wi-Fi is more than that in case of WiMAX as practically Wi-Fi offers more bandwidth than WiMAX.

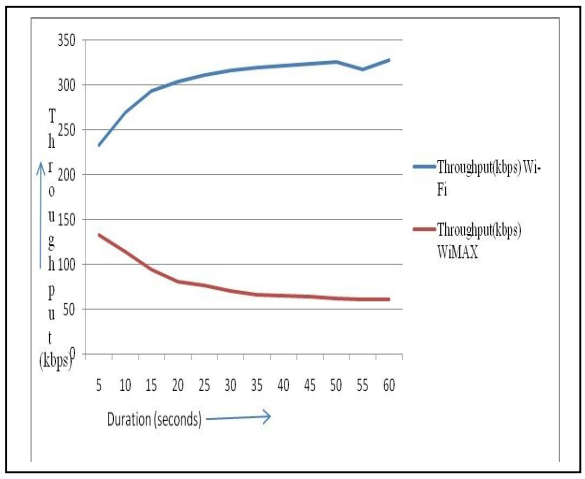

Fig. 3 : Throughput vs Duration of simulation

VII. FUTURE WORK

As the outcome of this work it is possible to create a MN with multiple Wi-Fi or multiple WiMAX interfaces using simple tcl scripts. So we were able to create MN with multiple homogeneous interfaces. This model can be extended to include multiple technologies and not the same technologies so that a MN can have Wi-Fi as well as WiMAX interfaces. Also many other technologies like UMTS, Bluetooth can be added to this model. So we will be able to create MN with multiple heterogeneous interfaces.

A. Proposed Architecture of Multiple Heterogeneous Interface MN

The same architecture can be extended for multiple heterogeneous interface support. Figure 4 shows the proposed architecture of a MN with multiple heterogeneous (Wi-Fi and WiMAX). Here the network layer and physical layer entities are grouped according to the type of interfaces. Each group is attached to a different propagation model since effects of different type of channels may be different. All the groups are attached to the single routing agent which will decide which interface to use for communication.

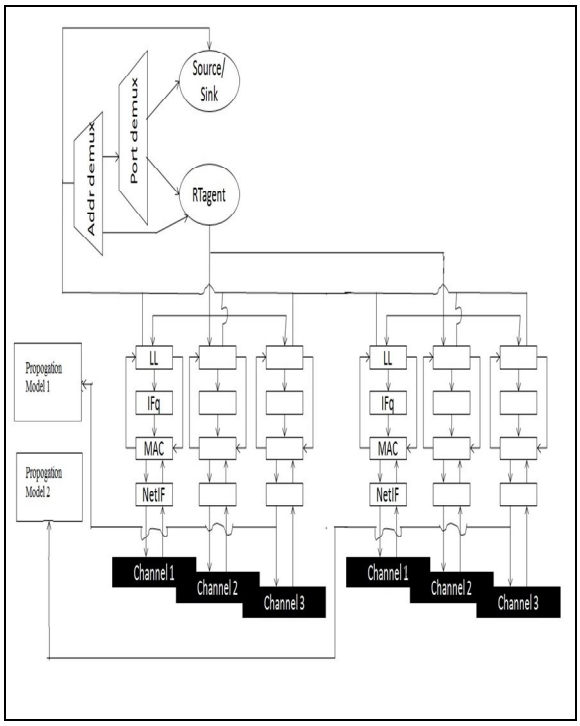

Fig. 4 : Proposed Architecture of Multiple Heterogeneous Interface MN

VIII. CONCLUSION

It is now possible to simulate a scenario in NS2 where a mobile node has multiple homogeneous i.e. Wi-Fi or WiMAX interfaces. This will help researchers to simulate the next generation wireless scenario where a MN is connected to multiple wireless networks through multiple interfaces at the same time.

REFERENCES

[1] V. Gazis, N. Housos, A. Alonistioti, L. Merakos, “Evolving perspectives of 4G mobile communication systems”, Personal Indoor Mobile Radio Communications (PIMRC) 2002, 15 – 18 September, Lisbon, Portugal.

[2] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[3] Siddharth Saha, Sabyasachi Roy, Ashwinias, "The Enhanced Network Simulator". http://www.cse.iitk.ac.in/users/braman/tens.

[4] Tzi cker Chiueh, Ashish Raniwala, Rupa Krishnan, and Kartik Gopalan. "Hyacinth: An IEEE 802.11-based Multi-channel Wireless Mesh Network." http://www.ecsl.cs.sunysb.edu/multichannel, October 2005.

[5] Ramon Calvo and Jesus Campo, "Adding Multiple Interface Support in NS-2", University of Cantabria, January 2007.

[6] http://mohittahiliani.blogspot.in/2010/03/adding-multiple-interface-support-in-ns.html

[7] Kevin Fall, Editor, Kannan Varadhan, Editor, “The *ns* Manual (formerly *ns* Notes and Documentation)”, The VINT Project, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, May 9, 2010.

[8] NDSL WiMAX Module for ns-2: http://ndsl.csie.cgu.edu.tw/wimax_ns2.php