MULTI-RADIO MOBILE NODE SIMULATION MODEL IN MANET USING AODV PROTOCOL

[1]Mr. G. Ramesh, [2]T. G. Asmitha Chandini

[1]Professor, [2]PG Final Year Student,
Department of Information Technology,
K.L.N.College of Engineering, Sivagangai, Tamil Nadu, India.

***Abstract*: Due to the rapid growth of the wireless communication, it is now believed that wireless devices with multiple interface support will highly increase in the near future. In this project, we present a solution to design a simulation model for mobile nodes with multiple radio interfaces in mobile ad-hoc wireless networks (MANET). This solution extends the mobile node architecture in the NS2 network simulator to allow analysis and evaluation of the performance of multi-channel mobile ad-hoc wireless networks. Simulation results show the effectiveness of the multi-radio mobile node model. Network performance with new mobile node architecture is greatly improved as the number of wireless interfaces increases.**

***Index Terms*: Multi-Radio Interface AODV Protocol; NS2 simulation; MANET; AD-HOC Wireless Network;**

INTRODUCTION

Wireless communication has seen a sharp increase in popularity and growth worldwide in recent years. This is mainly due to recent advances in wireless technology and mobile computing, along with their flexibility and decreased costs. Mobile wireless devices such as mobile phones, laptops, wireless sensors, Personal Digital Assistants (PDAs), and satellite receivers have become smaller, lighter and sufficiently portable to be carried by mobile users. In wireless networking, the information is exchanged among nodes by the transmission of electromagnetic waves through the air (radio frequency signals).

MANETs are self-configured networks and nodes having a dynamic topology. Hence, mobile nodes (routers) can freely move and arrange themselves on the fly, meaning that the topology of MANET may be subject to frequent unpredictable changes. MANETs can adapt to take a different structure depending on the circumstances and applications. They can operate in mobile, standalone and networked structures without any help from the base station (infrastructure) or central administration. Nodes in an ad hoc network are mobile and move spontaneously at different directions and speeds, and may leave/join the network as they wish. Therefore, MANETs need self -organization and self-healing mechanisms to ensure that the network can still function, even if some nodes move out of the transmission range of others. As mobile nodes have a limited transmission range, nodes may need intermediate nodes to relay a packet through several hops (multi-hop communication) until it reaches its destination.

The Network simulator (NS2) is widely used as a generic and powerful network simulator with open source code and simulation results from NS2 are generally considered credible and acceptable. NS2 has the capability of simulating not only wired networks but also wireless networks with a single frequency channel. Many researchers used the NS2 simulator to analyze the performance of MNET networks, but NS2 only supports mobile node with single radio interface, so it is not possible to analyze the performance of multi-channel multi-interface MANET. This paper presented a multi-radio mobile node model in NS2 and is used to evaluate the performance of multi-channel MANET.

PROBLEM STATEMENT

In traditional wireless ad hoc networks, all nodes are assumed to use a single shared channel for communication. In such networks, all nodes are contending to use the same shared medium for communication and only one node at a time can transmit or receive in any communication range. A MAC protocol is used to control and regulate access to the medium and reduce the probability of contention and collisions among transmitting nodes while accessing the medium.

We present a solution to design a simulation model for mobile node with multiple radio interfaces in mobile ad-hoc wireless networks (MANET). This solution extends the mobile node architecture in the NS2 network simulator to allow analysis and evaluation of the performance of multi-channel mobile ad-hoc wireless networks. NS2 does not support multi-radio mobile nodes. In order to extend the architecture of the mobile node with more radio interfaces, the old architecture needed to be improved by adding more copies of the wireless interface model.

LITERATURE SURVEY

1. A Generic Model of Multi-Interface and Multi-Channel for Multi-hop Wireless Networks

Authors : Xinhua Liu, YangFan Li, HaiLan Kuang, Fangmin Li

In this paper, a generic model of MIMC (abbreviated as GMM) for multi-hop wireless networks is proposed for NS2. The architecture of GMM consists of an MIMC control sub-layer, separated interfaces, and dynamic channels. The MIMC control sub-layer manages all interfaces and channels provides MIMC services to upper layers. Each separated interface has its own physical layer, MAC layer, link layer, and propagation model so that it can support heterogeneous interfaces. The dynamic channels are shared by all the interfaces, and each interface can be assigned to one or more channels by a present channel assignment algorithm. Compared with other existing MIMC models, the proposed GMM is more generic and more flexible.[1]

2. Multi-path Routing Improved Protocol in AODV Based on Nodes Energy

Authors : Xiaoxia Qi, Qijin Wang and Fan Jiang

Aiming at energy-constrained of Ad Hoc network, this paper proposes a multi-path routing protocol (EM-AODV) In AODV that based on the nodes energy. EM-AODV designs methods of obtaining nodes energy by upgrading the route discovery and route maintenance process of AODV, calculates the path of comprehensive energy derived path priority by routes total hops and nodes energy to format the multi-path routing mechanism. The energy as the metric prerequisite during the routing process, by setting nodes energy bound and balancing nodes data forwarding to postpone network lifetime. Simulation results show that EM-AODV has a lower average end-to-end delay, well improve the energy consumption. [2]

3. A multi-interface multi-channel routing (MMCR) protocol for wireless ad hoc networks

Authors : Reghu Anguswamy, Maciej Zawodniok and Sarangapani Jagannathan

Multiple non-interfering channels are available in 802.11 and 802.15.4- based wireless networks. The capacity of such channels can be combined to achieve a better performance thus providing a higher quality of service (QoS) than for a single channel network. However, existing routing protocols often are not suited to fully take advantage of these channels. The proposed multi-interface multi-channel routing (MMCR) protocol considers various QoS parameters such as throughput, end-to-end delay, and energy utilization as a single unified cost metric and identifies the route that optimizes the cost metric and balances the traffic among the channels on a per-flow basis. Multipoint relay nodes (MPRs) are first selected using available energy and bandwidth and utilized in routing. A novel load balancing scheme is introduced and analytical performance guarantees are demonstrated. Simulation results using the Ns2 show superior performance of the MMCR over the multi-channel optimal link state routing protocol (m-OLSR) in terms of throughput end-to-end delay, and energy efficiency. [3]

4. Multi Interface Multi Channel and Improved AODV Routing Protocol

Author : Aditi Mandape and Deepti Theng

The multi-interface multi-channel (MIMC) technique is used for the appropriate utilization of WMNs as backbone networks. Switching delays and distribution of routes plays an important factor in multi-channel networks. The shortest route in a multi-channel network might include multiple channel-switching which would cause further delay in transmission than a longer route which requires much lesser number of channel switches This paper presents the multi-interface multi-channel network and improved AODV routing protocol. When multiple channels are available, having more than one interface is helpful. IM-AODV (Improved MIMC AODV) includes the multi-target PREQ mechanism (M-PREQ), the predictive PREQ algorithm (P-PREQ), and the PREQ sender assignment algorithm for IEEE 802.11a. [4]

PROPOSED SYSTEM

- The number of channels in a particular scenario should be modifiable.
- The number of interfaces per node is variable and does not need to be the same for all nodes within a single scenario.
- Each node within the same scenario could connect to a different number of channels (of the ones that had been previously defined).
- Routing agents may take advantage of the modified model, but the legacy operation of the simulator must be preserved, to ensure backward compatibility.

MODULES

A. MOBILE NODE ARCHITECTURE IN NS2

The mobile node model of the MANET network has only one wireless interface built into the NS2 simulator. The mobile node is designed to simulate the structure of the actual mobile network nodes (Figure 1), in which the wireless interface includes the following objects: Link Layer (LL), Address Resolution Protocol (ARP), Interface Priority Queue (Ifq), Medium Access Control Layer (MAC), Propagation Model, Network Interface (netIF), all connected to the Channel object. Each mobile node uses the routing agent, to calculate the route to other nodes in the MANET network. The data packets are sent from the application and received by the routing agent. This module will determine the path of the packet to its destination. The packet is then sent down to the Link Layer. The link layer uses the ARP address resolution protocol to determine the physical address of the next node and map the IP address to the corresponding network interface. The packet is sent to the Interface Queue and waiting for the signal from the MAC protocol. When the MAC layer determines that the packet is sent to the channel, the packet is retrieved from the queue for delivering to the Network Interface and then sent to the radio channel. The packet is copied to all interfaces at the time at which the first bit of the packet would begin arriving at the interface in a real physical system. Each network interface stamps the packet with its own properties, and invokes the propagation model. The propagation model is invoked at the received part. It uses to transmit and receive stamps to determine the power that the interface receives the packet. The receiving network interface is left to decide whether the packet is received successfully or not. If it is successful, the packet is passed to the MAC layer. If the MAC layer receives this packet as error-free and collision-free, it passes the packet to the node's entry point. The packet then reaches a de-multiplexer, which decides whether the packet should be forwarded again or if it has reached its destination node. If the received node is the destination, the packet is sent to a port de-multiplexer which decides the application for which the packet should be delivered. If the packet should be forwarded again, the routing agent is called and the operation is repeated.

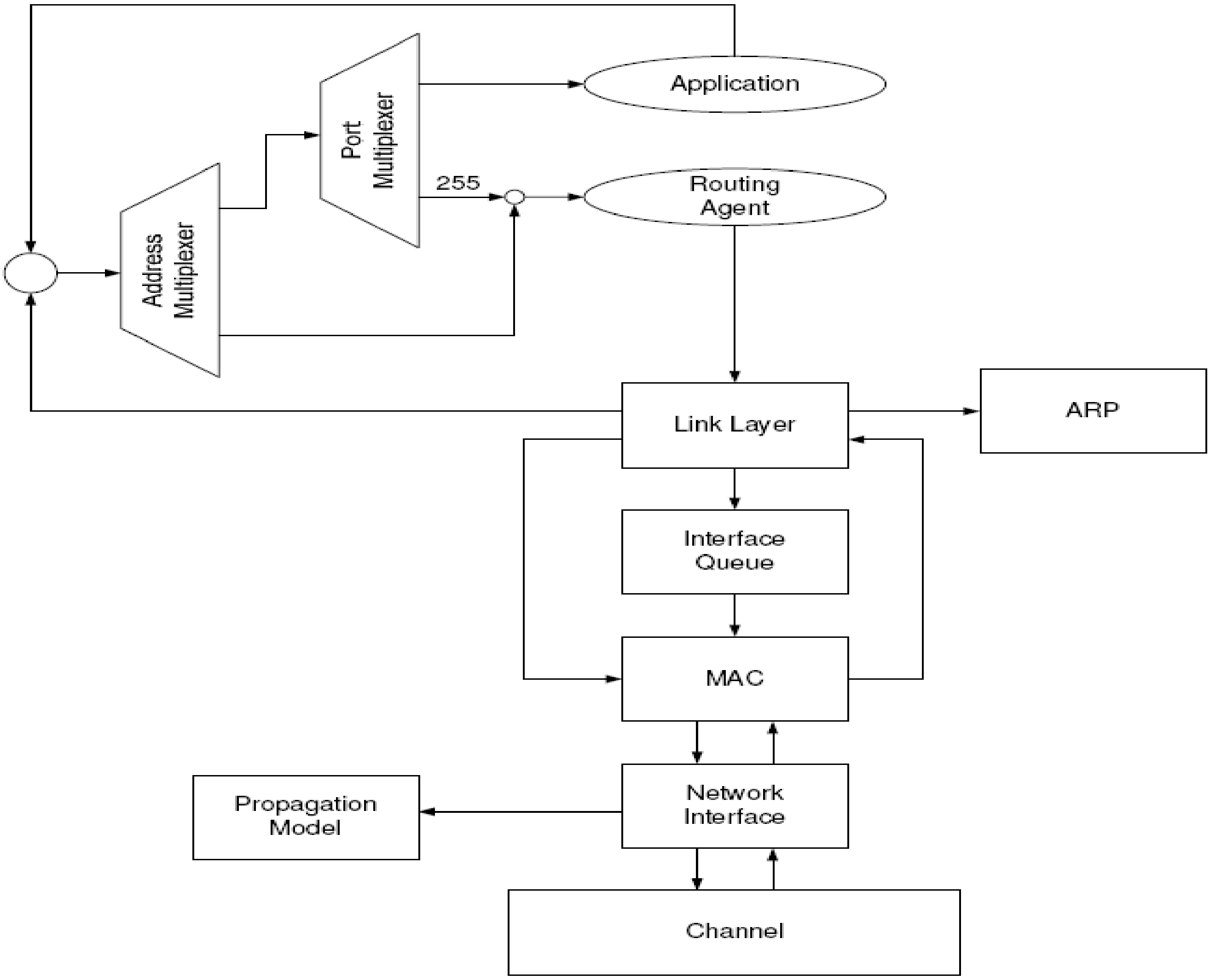

Figure 1 Mobile node architecture in NS2

B. EXTENDING NS2 WITH MULTI-RADIO INTERFACE SUPPORT

NS2 does not support multi-radio mobile nodes. In order to extend the architecture of the mobile node with more radio interfaces, old architecture needed to be improved by adding more copies of the wireless interface model. Figure 2 presents a high-level architecture of the modified Mobile Node object. Each node can have as many instances of the link layer, ARP, interface queue, MAC, network interface and channel entities as the number of interfaces it has. One can imagine that each instance actually represents a wireless network interface. Thus, this design scheme emulates the fact that our multi-radio mobile node ad-hoc network implementation will not require any modification to existing IEEE 802.11 hardware. As can be observed, most legacy operations of ns-2 are still preserved. Incoming traffic arrives through the corresponding channel and travel through the different components in ascending order then eventually merges to a single point at the address multiplexer. For outgoing traffic, the determination of selecting which interface to pass the data packets is to be handled by the routing agent. In other words, modifications will be required in implementing the routing agent to add the intelligence of selecting the appropriate interface.

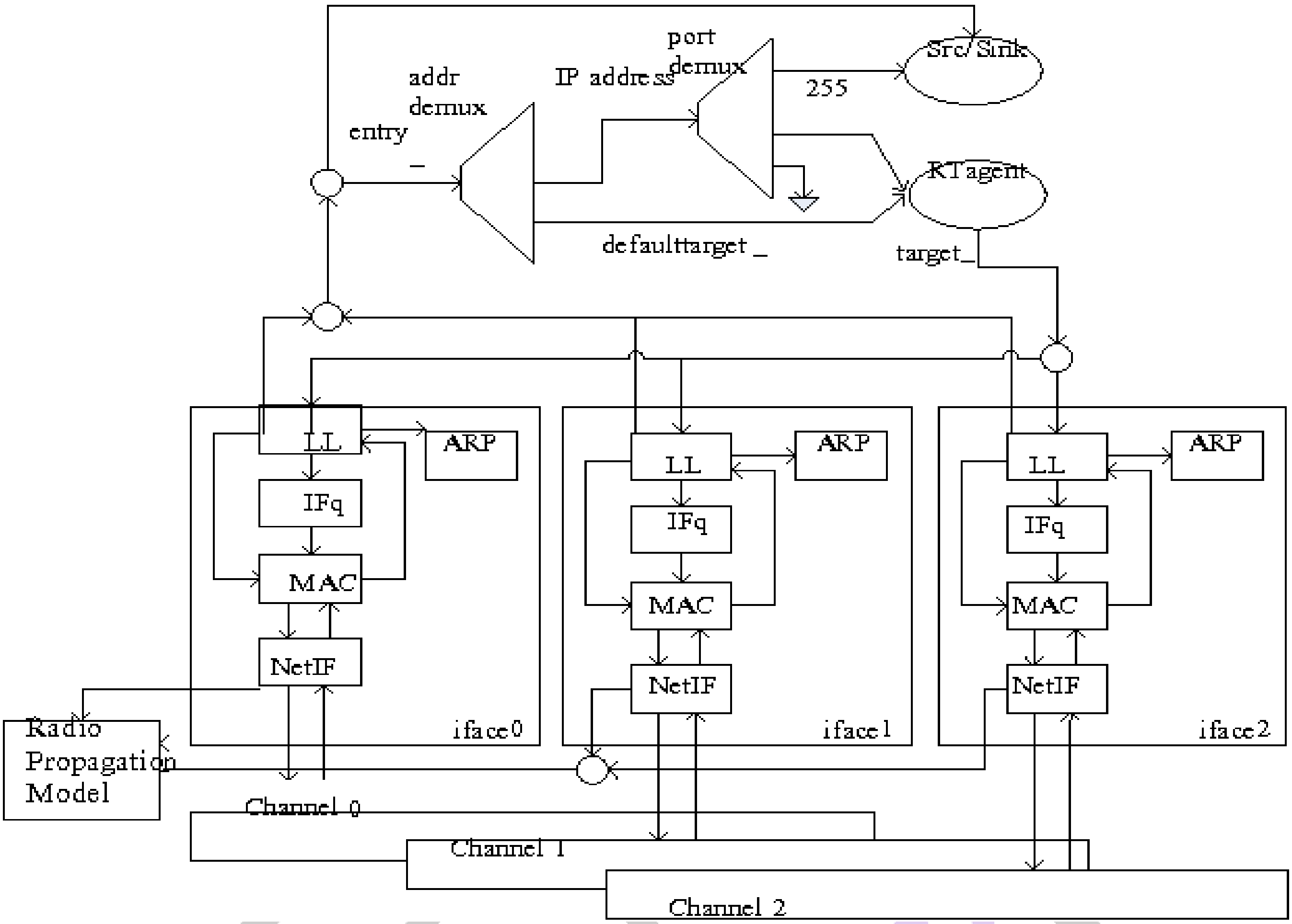

Figure 2 modified mobile node architecture with the support of multiple interface

This section describes of the various changes that the guideline in proposes to add the multiple interface support in ns-2. As can be deduced from the ns-2 architecture mentioned in earlier sections, the modifications involve both the OTCL and C++ realms.

Changes in ns-lib.tcl file

- **Change-numifs {***newnumifs***}** Allows users to specify *newnumifs* interfaces per node. This procedure is to be called prior to creating a wireless node in the scenario script. Once called, it will affect all subsequent nodes until another invocation of the procedure is issued.
- **add-channel {***indexch ch***}**Adds an interface (channel) to a node. This procedure takes two arguments: *indexch* is the index of the channel to be added; *ch* references to the channel object previously created.
- **get-numifs {}** Retrieves the number of interfaces currently defined.
- **ifNum {val}** Adds multiple interfaces as an argument to *node-config*, which is an existing ns-2 command used to configure a MobileNode object, by setting a value for the local variable, *numifs_*.
- **node-config args {}** Adds the support for multiple channels. For backward compatibility, initializes *chan* as a single variable if normal operation is used or as an array if multiple interfaces are defined. Adds the *numifs_* variable as a new member in the argument list *args* that is passed to the procedure.
- **create-wireless-node {}** Takes in the number of interfaces specified in *numifs_* and iteratively calls *add-interface* as many times as the number of interfaces that the node has. *add-interface* is an existing ns-2 procedure that adds an interface to a previously created MobileNode object.

Changes in ns-mobilenode.tcl file

- **add-target {**agent port**}** This procedure attaches a routing *agent* to a MobileNode object, picks a *port* and binds the agent to the port number. Gets the number of interfaces via calling *get-numifs*; doing so allows the procedure to determine whether multiple interfaces are present. If the number of interfaces is non-zero (implying that the multiple interface extension is used), the procedure iteratively calls the *if-queue* command as many times as the value returned by *get-numifs.*
- **add-target-rtagent {**agent port**}** This procedure, called by *add-target*, adds a target routing agent. It first gets the number of interfaces that a node has from *get-numifs*. If the number of interfaces is non-zero, the procedure associates the routing agent with the corresponding link-layer target entity as many times as the number of interfaces for both of the sending and receiving targets.
- **add-interface {}** This procedure adds an interface to a Mobile-Node object. Originally, it creates one ARP table (for

address resolution) per node. We modify it to be one ARP table per interface although having one ARP table per node resembles the real-life case. The reason for such a walk-around is that, if a node is using one interface to communicate with another one, the current design of Mobile-Node in ns-2 will not allow the node to use another interface since the request to the ARP entity will still be serving the previous interface.

- **init args {} and reset {}** Due to the above change of assigning the ARP table, these two procedures are modified to initialize and reset the ARP table of a Mobile-Node object per the number of interfaces defined.

Changes in common/mobilenode.h file

After adding the multiple interface support in the OTCL realm, relevant changes have to be made in the C++ code for the Mobile-Node object, the channel entity, and the MAC layer model. NS-2 controls each instance of the Mobile-Node objects which are associated with a channel by means of a linked list. Two lists are managed; one references the previous node, *prevX_*, while the other references the next node, *nextX_*. The original format of the list is simply a pointer to a node. In order to support multiple channels, the list is modified to be an array of pointers with the size of the array being the maximum of number of channels: *nextX_[MAX_CHANNELS] , prevX_[MAX_CHANNELS]*. The index for referring to a node is the channel number.

We remove the inline declaration of the *getLoc*() function. Due to the above changes on the Mobile-Node lists, the original declaration has been found to always return a zero distance, which leads to wrong packet receptions.

Changes in mac/channel.cc file

Due to the changes on the Mobile-Node lists, we modify accessing each node entry when attaching, removing, and updating a new node to a channel to refer to the corresponding channel number. This number can be accessed by *this->index ()*, where this is the current instance of the channel class object. In other words, whenever *nextX_* and *prevX_* appear in **channel.cc**, they need to be replaced by*: nextX_[this->index()]***,** *prevX_[this->index()]* In the *affectedNodes* () function, we add a new condition to check which of the interfaces of the destination node is connected to the same channel before transmitting the packet to another interface. The original design of ns-2 does not consider the case with multiple channels; it simply sends the packet to all of the destination interfaces. Accessing the channel of the destination interface is carried out by*: rifp->channel ()* where *rifp* is a pointer to the receiving interface. The *channel* () member returns a channel object that has the same type as this.

Changes in mac/mac-802_11.cc file

For the correct handling of multiple interfaces by the routing agent, we modify the *recv* () method in the MAC 802.11 class to register the correct MAC receiving interface in the MAC header. The hardware address of the interface can be accessed by: *hdr->iface () = addr ()*

C. EXTENDING AODV ROUTING PROTOCOL FOR MULTI-RADIO MOBILE NODES

The original AODV protocol is only suitable for mobile nodes with single radio interfaces. Therefore, in order to adapt the protocol for multiple radio interface, we must understand how routing is performed in ns-2's AODV agent. As an example, we examine how AODV establishes a route from the first to the last node in a chain topology with four nodes. AODV builds routes using a Route Request (RREQ)/Route Reply (RREP) query cycle, as illustrated in the figure 3. When the nodes in an ad-hoc wireless network are created, function *command* () in ns-2's AODV library is executed by each node to initialize itself in the AODV routing agent. When a source desires a route to a destination node, it broadcasts a RREQ packet across the network by calling *sendRequest* (). Intermediate nodes receiving this packet update their information for the source node and set up a backward pointer to the source node in their routing tables. The packet is also forwarded to the next node via *forward* (). Once a node receiving the RREQ is the desired destination, it unicasts a RREP packet back to the source by calling *sendReply* (). As the RREP propagates back to the source node, the intermediate nodes receiving the packet via *recvReply* () set up a forward pointer to the destination and forward it to the previous node by *forward* (). As soon as the source node receives the RREP, an active route has been established; it may start to forward data packets to the destination by calling *forward* (). The aforementioned steps are known as the Route Discovery state.

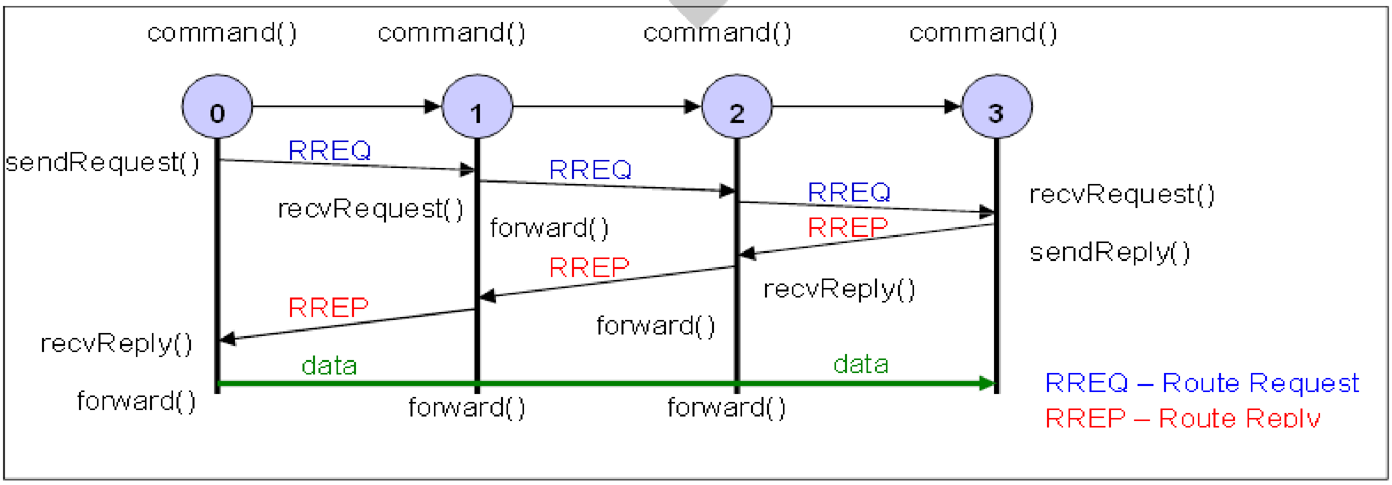

Figure 3 AODV route establishment for a chain topology with 4 nodes

Next, the Route Maintenance state is for updating the route previously established. Occasionally, a node of an active route may offer connectivity information by calling *sendHello* () to broadcast local Hello messages. Whenever a node receives a Hello message from a neighbor via *recvHello*(), the node makes sure that it has an active route to the neighbor, and creates one if necessary. The current node can now begin using this route to forward data packets. Furthermore, when a link breakage is detected by a node, the node will call *sendError* () to notify the source node with a Route Error (RERR) packet. The source node will initialize a new Route Discovery state as mentioned before.

SIMULATION SCENARIO

To demonstrate the performance of our implemented interface switching algorithm in ns-2 and the benefits of using a Multi-radio mobile node, ad-hoc wireless networks in simple chain topologies are used for simplicity. The reasons are that only one single route between the source and the destination is involved and that direct communication is only possible between adjacent nodes on the chain, as illustrated in the following block diagram. (figure 4)

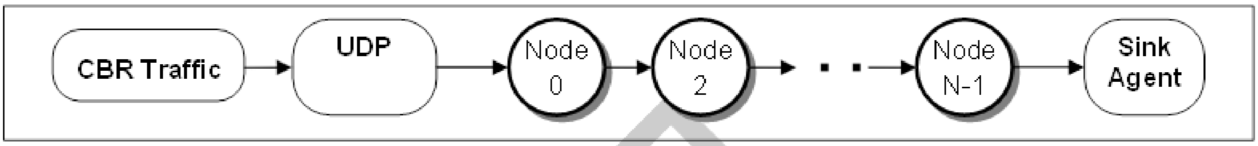

Figure 4 Block diagram of the NS-2 simulation scenario in chain topologies

The following list summarizes the various scenario parameters for our simulations:

- The length of the chain topologies is varied from 2 to 11 nodes, denoted by N.
- The nodes are stationary within a confined area of 1000 m x 1000 m.
- The number of orthogonal channels is varied from 2 to 4, denoted by C.
- Each node is always equipped with two interfaces (1 fixed, 1 switchable); thus, I = 2.
- The duration of each simulation is 60 s; however, the actual simulation time increases as the length of the chain topology as well as the number of channels increases.
- The data rate or the channel capacity is set to 5.4 Mbps.
- Constant Bit-Rate (CBR) traffic is passed from Node 0 to Node N-1, as denoted by the ns-2 CBR Traffic application and the Sink Agent in Figure 4.
  - ✓ The data packet size is set to 1000 bytes.
  - ✓ The CBR data (packetization) rate is set to 1.4ms, which is chosen to be large enough to saturate the network according to the following calculation: $\frac{(1000\ byte)/(8bit/byte)}{1.4ms} = 5.7Mbps > 5.4\ Mbps$ where 5.4 Mbps is our specified channel capacity.
  - ✓ Data packets are transported using the User Datagram Protocol (UDP), as indicated by the UDP Agent in Figure 10. Therefore, no flow and congestion controls will take place that may impair the throughput performance. As well, the packets can be transmitted as fast as the packetization rate.

- The performance of the above multi-radio mobile node scenarios is compared with the performance of AODV when using a single channel single interface ad-hoc wireless network. The following highlights some key configurations for the case of 4 nodes in the chain topology, 3 channels and 2 interfaces:

Mac/802_11 set dataRate_ 5.4e6 set val(nn) 4 set val(nc) 3 set val(ni) 2 set pktsize 1000 set pktrate 0.0014	Set up the channel capacity, the number of nodes, channels and interfaces, packet size and packetization rate
Set udp0 {new Agent/UDP} $ns_ attach-agent $n(0) $udp0	Define the UDP Agent and attach it to Node 0
Set sink0 [new Agent/Null] Set last_node_id [expr $val(nn)-1] $ns_ attach-agent $n($last_node_id) $sink0 $ns_ connect $udp0 $sink0	Define the Sink Agent and attach it to Node N-1. Finally, connect the UDP Agent with the sink agent to specify the data flow path.
Set cbr0 [new Application/Traffic/CBR] $cbr0 attach-agent $udp0 $cbr0 set packetSize_ $pktsize	Define the CBR Traffic application, attach it to the UDP Agent and specify the CBR packet size and data rate

SIMULATION RESULT

Throughput is the number of successful delivery of data within a unit of time. Figure 5 shows that with the start of a simulation, the throughput of the single radio interface and multi-radio interface AODV protocols are all starting to rise, and the rise of the extended multi-radio interface AODV protocol is significantly faster. This indicates that the throughput of the extended OLSR protocol is better than the single radio interface AODV.

End-to-end delay is the delay time of the CBR packet to the end. From Figure 6 it can be seen that after 10 seconds the end-to- end delay is basically stable, and the 01027-p.4 extended AODV protocol is lower than the original AODV protocol.

Jitter means the delay variance. Because the state of the network changes and the traffic is always changing. So the data may be queued in the queue, waiting for transmission. The time of each packet from the source node to the destination node is not necessarily the same. This time difference is Jitter. As can be seen from Figure 7, the jitter of the single radio interface AODV protocol is much bigger than the multi-radio interface AODV, so it is shown that the multi-radio interface AODV protocol is more stable than the original AODV protocol.

Packet loss represents the packet loss rate. Which is calculated by the ratio of the number of packets sent by the total sending node to the difference of the total receiving packet received from the total sending packets to be transmitted. From figure 8 we can see that the packet loss rate decreases with time. When the packet loss rate is the same, the single radio interface AODV protocol is earlier to reach the moment. And multi-radio interface needed more time than a single radio interface. Thus, in the packet loss rate comparison experiment, the loss rate of the multi radio interface packet is smaller than that of the single radio interface, and the performance of the multi-radio interface is better than that of the single radio interface.

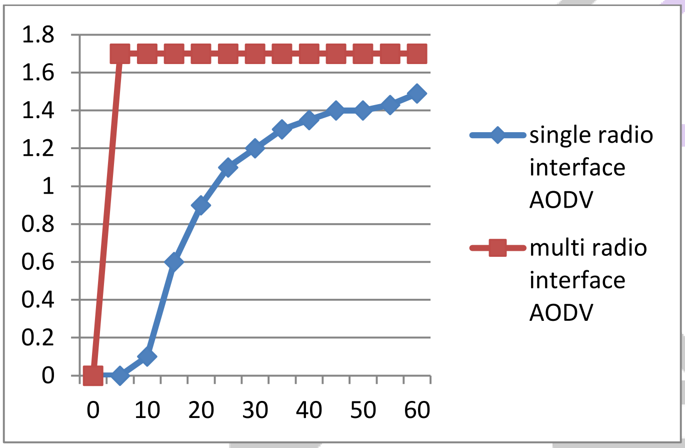

Figure 5 : Throughput

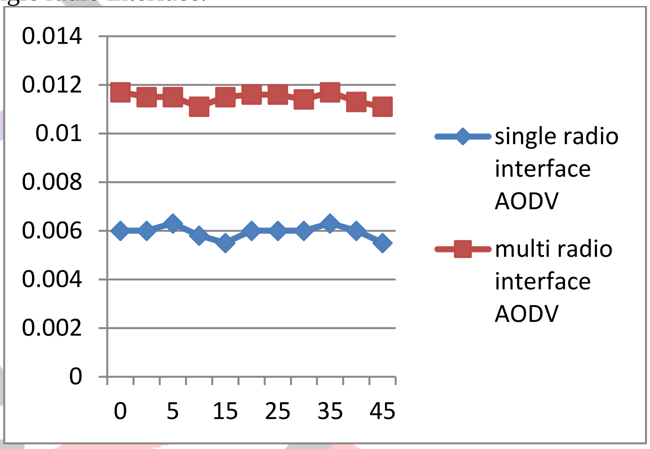

Figure 6 : End-to-End delay

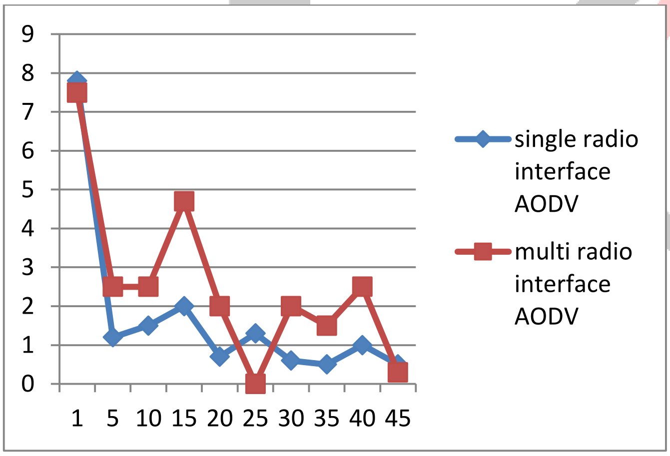

Figure 7 : Jitter

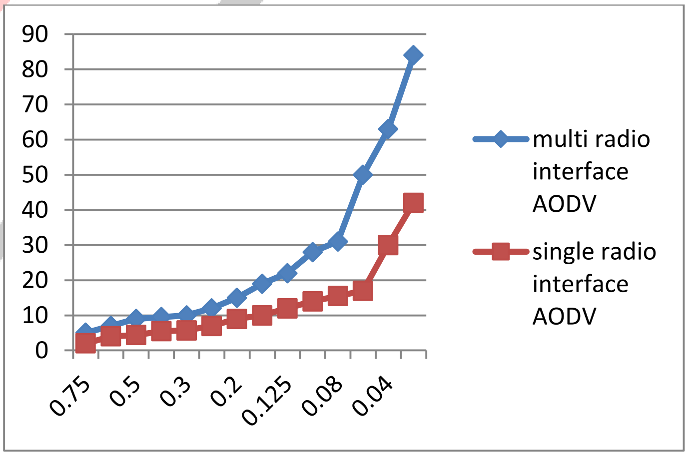

Figure 8 : Packet loss

CONCLUSION

Using the modified ns-2 with the multi-radio interface and interface switching capabilities, we have simulated a multi-radio mobile node ad-hoc wireless network in simple chain topologies. By varying the number of available channels and the number of nodes in the chain while keeping each node to have two interfaces (one fixed and one switchable), we have generated various interface switching interactions and end-to-end throughputs using the modified ns-2. Simulation results validate the effectiveness of our implemented interface switching algorithm in ns-2's AODV routing protocol. More importantly, the results demonstrate that network throughput can significantly improves in ad-hoc wireless networks with multiple channels and multiple interfaces in each wireless node.

REFERENCES

[1] Xinhua Liu, YangFan Li, HaiLan Kuang, Fangmin Li, "A Generic Model of Multi-Interface and MultiChannel for Multi-hop Wireless Networks," 2015 4th International Conference on Computer Science and Network Technology (ICCSNT 2015).

[2] Xiaoxia Qi, Qijin Wang and Fan Jiang , ''Multi-path Routing Improved Protocol in AODV Based on Nodes Energy,'' International Journal of Future Generation Communication and Networking Vol. 8, No. 1 (2015)

[3] Reghu Anguswamy, Maciej Jan Zawodniok, and Jagannathan Sarangapani "A Multi-interface Multi-Channel Routing (MMCR) Protocol for Wireless Ad Hoc Networks," IEEE Communications Society subject matter experts for publication in the WCNC 2009.

[4] Aditi Mandape, and Deepti Theng, "Multi Interface Multi Channel and Improved AODV Routing Protocol," ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India

[5] Sachin Dattatraya Patil and Varshapriya Jyotinagar, "Implementation of Multiple Heterogeneous Interface Mobile Node in NS2" International Journal of Engineering Innovation & Research Volume 1, Issue 2, 2012.

[6] Carine TOHAM , François JAN, and Andrzej DUDA" A Multi-Channel and Power Controlled Based AODV for Throughput Improvement in Multi-hop Ad Hoc Networks" The 4th International Conference on Mobile Ad-hoc and Sensor Networks 2008.

[7] The Network Simulator NS2. Available: http://www.isi.edu/nsnam/ns

[8] K. Fall and K. Varadhan, Eds., The ns Manual, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Apr. 2002. available: http://www.isi.edu/nsnam/ns/.