International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

A FRAMEWORK FOR INTEGRATED ROUTING, SCHEDULING AND TRAFFIC MANAGEMENT IN MANET

K. Balaji[1]

[1] *Associate Professor, Author Department of EEE, St. Peter's Engineering College, Avadi, Chennai, India*

---***---

Abstract - ***A rapid expansion in the field of mobile computing is due to the proliferation of inexpensive, widely available wireless devices. A mobile ad hoc network is an autonomous collection of mobile devices (laptops, smart phones, sensors etc.,) that communicates with each other over wireless links and cooperates in a distributed manner in order to provide the necessary network functionality in the absence of a fixed infrastructure. In this paper, we present a routing protocol called Scheduling and Traffic Management in flow ordered Routing Meshes for the operation of such mobile ad hoc network. It uses cross layer approach and here each node acts as a router. It provides loop free routes and fast and efficient way of repairing routes, because they contain extra paths that can be used in case of link breaks. Bandwidth and Delay guarantee are achieved on as per hop and end to end basis. The performance analysis shows that compared to AODV for Unicast routing, STORM attains better improvement in case of parameters like Energy consumption, Delay, Packet delivery ratio and Packet lost.***

Key Words: *Scheduling, Traffic management, Ad hoc network*

1. INTRODUCTION

This system introduce a new cross-layer framework for the dissemination of unicast and multicast flows that may be real time(e.g.,voice conversations) or elastic(e.g.,http).For a multi-hop wireless network,STORM provides a method of accessing the wireless channel.The method consists of different time slots for each node in the network.For accessing the time slot,node makes use of distributed elections of available time slots and reservation of time slots.It uses distributed election algorithm based on hashing functions of node identifiers for those nodes that haven't reserved any time slots.
A virtual link has been created for each real time data flow,between source and destination.It is given by Interest-driven routing algorithm where nodes are given number based on their distance and available bandwidth.For providing the virtual link,the routing algorithm gives end-to-end channel access schedule for each data flow.By starting with reservation protocol,the routing algorithm gives the end-to-end channel access schedule for each real time data flow.

1.1 Channel Structure and Traffic Management

Nodes share the same frequency band, and we assume that clock synchronization among the nodes in the network is achieved through a multi hop time synchronization scheme such as the one implemented in Soft-TDMAC which is a TDMA-based MAC protocol that runs over commodity 802.11 hardware. Nodes access the common channel assuming that it is organized using a time-division multiple access structure, which we call STORM frame and is illustrated in Fig.1 Each STORM frame is composed of N time slots (from slot 0 to slot N - 1) and we use the position of a slot within the STORM frame as the identifier of the slot. A STORM frame does not have any particular structure and any time slot can be used to transmit a sequence of packets (signaling or data). There is only one special purpose time slot used to admit new nodes to the network. These admission time slots occur every A time slots, with A >>N, and are used by nodes to transmit their first hello packets on a contention basis.

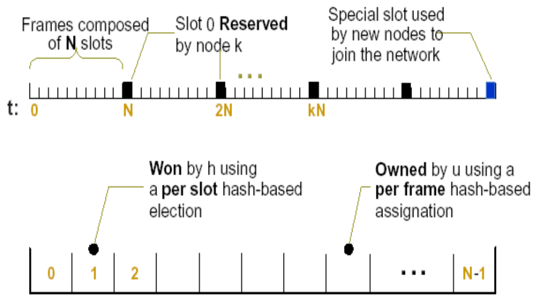

Fig.1 MAC frame

When a node is allowed to transmit over a time slot, it fits as many packets as possible in it. Packets are selected from the local transmission queues, which are FIFO and are served using a priority-based algorithm. Reservation packets have the highest priority (pRsv), because quick consensus is needed on which nodes should have access to which time slots. The next priority is given to network-layer signaling packets (pctr), and data packets waiting in data queues have the lowest priority. Data queues can be either elastic or real-time, and real-time queues are assigned higher priority (pRT) than the priority given to elastic queues (pelastic), given that jitter and latencies are not as important for the latter. The priority of a real-time queue created for flow f is increased from pRT to pRTþ if the current time slot t was reserved on behalf of flow f. Hello packets are transmitted with the lowest priority (pHello_ < pelastic) if more than hello period=2 seconds but fewer than hello period seconds have elapsed since the last time a hello packet was transmitted, because there is no need for the information yet. However, if more than hello period seconds have elapsed, then the neighborhood information must be refreshed and hence the priority of the hello packet is set to pHelloþ > pctr. To summarize, during a time slot allocated to a node, the relationships among traffic priorities are: pHello_ < pelastic < pRT < pRTþ < pctr < pHelloþ < pRsv.

1.2 Neighbor Protocol

Routing, reservations, and transmission scheduling in STORM use distributed algorithms that require each node to know the nodes within its two-hop neighborhood. The neighborhood of a node consists of those nodes whose transmissions the node can decode, which we call one-hop neighbors, and the one-hop neighbors of those nodes are called two-hop neighbors. More formally, let G=(V,E) be an undirected graph with a set of vertices V representing the set of nodes present in a wireless ad hoc network and a set of edges E. Any two nodes u and v share an edge (u,v) ε E if they are one-hop neighbors (i.e., within radio transmission range) of each other. For any node u ε V, we denote IN (u) = {v: (u,v) ε E} as the one-hop neighborhood of u and IN(IN(u)) as the two-hop neighborhood of u. To gather two-hop neighborhood information, each node transmits hello messages periodically every hello period seconds, and each such message contains a list of tuples for the node itself and for each of its one-hop neighbors. Each tuple is composed of a node identifier, a list of the identifiers of the time slots reserved by the node, and the length of the list of reserved slots. Each node stores the last hello message received from each one-hop neighbor (or simply neighbor) in its neighbor list. A neighbor is deleted from the neighbor list if no hello message is received from that neighbor in three consecutive hello periods.

It is worth noting that the neighbor protocol in STORM is very similar to approaches used in traditional routing protocols that also require neighborhood information (e.g., OLSR) in that hello messages are transmitted unreliably but persistently, and convey information about local Neighborhoods. The neighbor protocol is also used to detect when two nodes in a two-hop neighborhood have reserved the same slot. To resolve a conflicting reservation, the node with the larger identifier keeps its reservation over the particular slot, whereas the node with the lower identifier has to give up its current reservation and start a new reservation transaction over a different slot. The main source of these conflicting reservations is node mobility, which changes the neighborhood of nodes. The neighborhood information contained in hello messages allows nodes to detect these collisions before the conflicting nodes become one-hop neighbors.

2. SYSTEM DESIGN AND DEVELOPMENT

2.1 Fact finding

Fact finding is the methods of gathering the information required about the existing system. Some of them are as follows.

- Observation
- Record Searching
- Special purpose Records
- Sampling
- Questionnaires
- Interviewing

Observation of the current work situation will provide clues to problems and atmosphere. Record searching, special purpose records and sampling will give quantitative information about the system which facilitates sizing of the proposed system and may also point the areas of difficulties which are being experienced. Questionnaires can be used to collect the quantifiable data about the system. All of the techniques need to be supplemented by more detailed discussion of the interview situation. The identification of the user requirements, decision areas, objectives, and responsibilities for certain procedures can only be achieved for interviewing. Based on the above fact finding techniques, it is observed the current situation of the existing system. It is very helpful to finding the areas of difficulties, which are being experienced in the existing system. Thus it helps to develop the proposed system with the quantifiable data.

2.2 Input design

Input design is part of overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. The inputs in the system are of three types

- External : Which are prime inputs for the system
- Internal : which are user communication with the system
- Interactive: which are inputs entered during a dialog with the computer.

The above input types enrich the proposed system with numerous facilities that make it more advantageous in comparison with the existing normal system. All the inputs entered are completely raw initially, before being entered into a database, each of them available processing. The input format in this system has been designed with the following objectives in mind.

2.3 Feasibility Analysis

All systems are feasible, given unlimited resources and infinite time. Before going further into the steps of software development, the system analyst has to analyze whether the proposed system will be feasible for the organization and must identify the customer needs. The main purpose of feasibility study is to determine whether the problem is worth solving. The success of the system is also lies in the amount of feasibility study done on it. Many feasibility studies have to be done on any system. But there are three main feasibility tests to be performed. They are

- Operational feasibility
- Technical feasibility
- Economic feasibility

Operational feasibility

During feasibility analysis Operational feasibility study is a must. This is because; according to Software Engineering Principles Operational feasibility or in other words usability should be very high. A thorough analysis is done and found that the system is Operational.

Technical feasibility

The system analyst is to check the technical feasibility of the proposed system. Taking account of the hardware it is used for the system development, data storage, processing and output, makes the technical feasibility assessment. The system analyst has to check whether the company or user who is implementing the system has enough resource available for the smooth running of the application. Actually the requirement for this application is very less and thus it is technically feasible.

Economical feasibility

Before going further into the development of the proposed system, the system analyst has to check the economic feasibility of the proposed system and the cost for running the system is composed with the cost benefit that can achieve by implementing the system. As in the case of Crypto Media development cost is not high, as it does not need any extra hardware and software. Thus the system is economically feasible.

In trace file each line is started with some letter like R, S, D, and N. Each of the letters has meaning. For detailed meaning of the letter refer to the NS Manual page. It will give a lot of information about your simulation result. Cocoa provides you with two basic archiving options:

- ➢ XML based property list
- ➢ Persistent objects

Saving some data to an XML property list is really easy under Cocoa. The simplest way to do it is to save a property list object, which can be any of NSArray, NSDictionary, NSstring or NSData. For most things, an NSArray or NSDictionary will do. A nice feature is that since NSArray and NSDictionary can contain other property list objects, those objects will be written into the property list as well.NSCoding is a protocol that you can implement on your data classes to support encoding and decoding your data into a buffer, which can then be persisted to disk. Implementing NS Coding is actually ridiculously easy.

In this system, we introduce a new cross-layer framework (STORM) for the dissemination of unicast and multicast flows that may be real time or elastic. STORM uses distributed algorithm that provides flow ordered routing meshes consisting of multiple path from source to destinations over which are capable of establishing channel reservation by means of MAC Frame format that meet the end to end requirements of the flows being routed. The Priority based queuing system is used to handle signaling traffic, elastic data flows and real time flows.

2. DISTRIBUTED ALGORITHM

When a process wants to enter a Critical Section (CS), it builds a message containing,

1. The name of the CS it wants to enter,
2. The process number,
3. The current system time.

It then sends message to all other process, including itself. The sending of message is assumed to be reliable (i.e) messages are not lost. When a process receives a request message from another process, the action it takes, depends on its state with respect CS named in the message.

Three cases to be distinguished,

1. If the Rx is not in the CS and does not want to enter it, it then sends back an OK message to sender.
2. If the Rx is already in CS, it does not reply.
3. If the Rx wants to enter the CS but has not yet done, it compares the timestamp in the incoming message with the one contained in the message that it has sent to everyone. The lowest one wins.

Also,

- If the incoming message is lower, the receiver sends back an OK message. If its own message has a lower timestamp, then the Receiver Queues the incoming request and sends nothing.
- After sending out request asking permission to enter a CS, a process sits back and waits until everyone else has given permission.
- As soon as all the permission are in, it may enter the CS,When it exits the CS,it sends OK message to all

process on its QUEUES and deletes them all from the queue.

2.1 SYSTEM MODULES

The basic steps in the project for attaining the comparative analysis are given below:

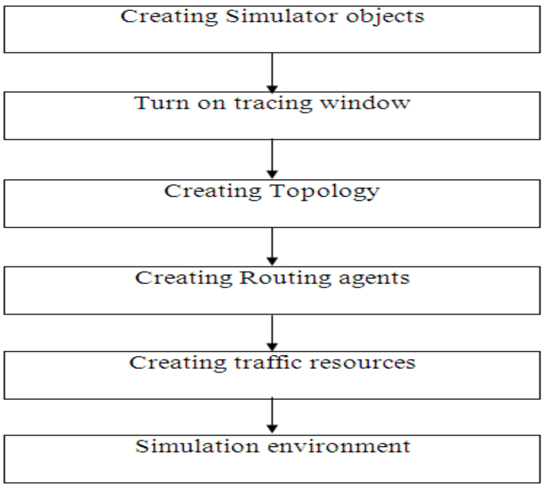

Fig.2 NS2 Generic Script Structure

Creating Simulator Objects

The initial step in a NS-2 script is creating a new simulator object. In that, it gives all the details about the number of nodes, types and parameters of the used network.

Turn on Tracing Window

This window traces the simulation events at each and every seconds of the given simulation period.

Creating Topology

The next step is to give topology for the network. For the MANET, the specified topology is MESH. For any wireless network, it is necessary to give all the necessary parameters like type of channel, type of ad-hoc routing protocol, type pf antenna, etc..

Creating Routing Agents

This section will create the appropriate routing agents for the data flow. In STORM, TCP has been used. It is much more reliable than the other and it is the one which has been supported easily by NS-2.It provides the routing algorithm for the network.

Creating Traffic Resources

Based on the routing agent, it is necessary to give the details about traffic sources. In this project, CBR has been used efficiently. It specifies the type of data flows, rate and the packet size and the time at which flow has to be started.

Simulation Environment

The script might create some output on stdout, it might write a trace file or it might start name to visualize the simulation.

3 RESULT ANALYSIS

We use NS2 as our simulating tool. We assigned a network consisting of 20 nodes from node 0 to node 19.Initially, each node find its neighbor node by transmitting HELLO Messages. The HELLO Messages is transmitted periodically for every HELLO period second. The default transmitting range for HELLO Message is 250m. A node is deleted from its neighbor list if no hello message is received from that neighbor in three consecutive hello periods.

After finding its one hop and two hop neighborhoods, a node start transmitting its packet .The source node sends constant bit rate traffic to destination node. The traffic sources are carried by transport layer protocols User Datagram protocol (UDP) or Transmission control protocol (TCP).In NS2 transmitting object of these two protocols are a UDP agent and a TCP agent, then the receiver must be a Null agent and a TCP sink agent respectively.

At the end of simulation, the trace file is created and the NAM is running (since it is invoked from within the procedure finish{}).Trace file gives the details of packet flow during the simulation.NAM trace is records simulation detail in a text file, and uses the text file the play back the simulation using animation.

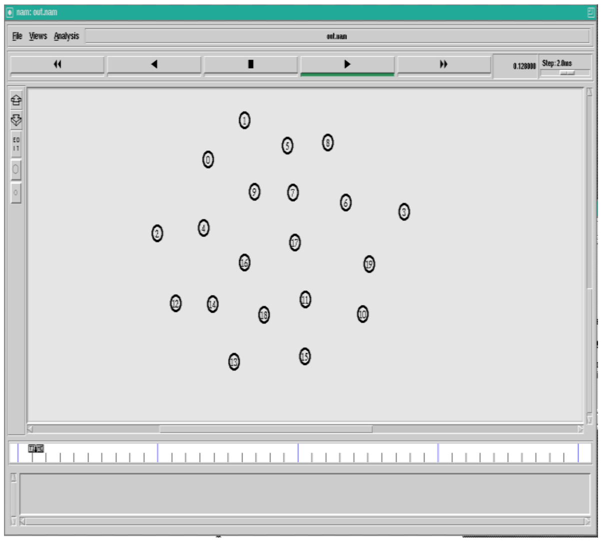

Fig.3 Node Initialization

Here we are assigning 20 nodes from node 0 to node 19 and they are apart from each other.

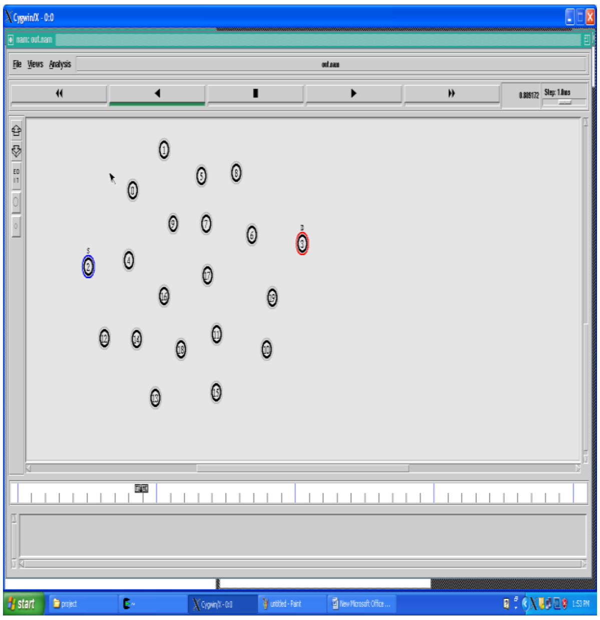

Fig.4 Node Configuration

The source node and destination node are identified separately by colouring source node in blue colour and destination node in red colour.

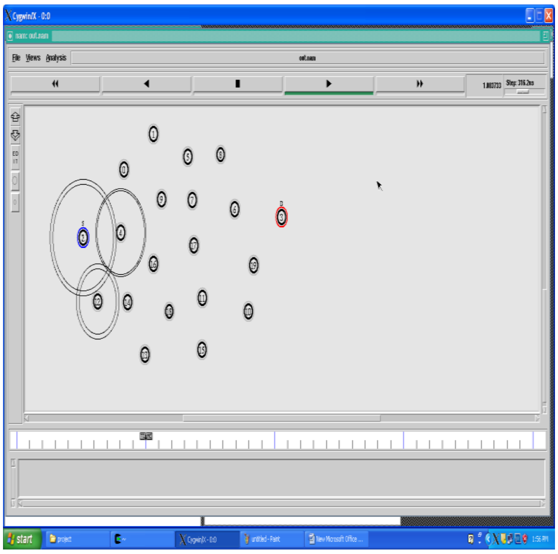

Fig.5 Neighborhood Identification

All the nodes find its neighbor node by sending HELLO Message. They transmit HELLO Messages periodically for every hello period second. A node is deleted from its neighbor list if no hello message is received from that neighbor in three consecutive hello periods.

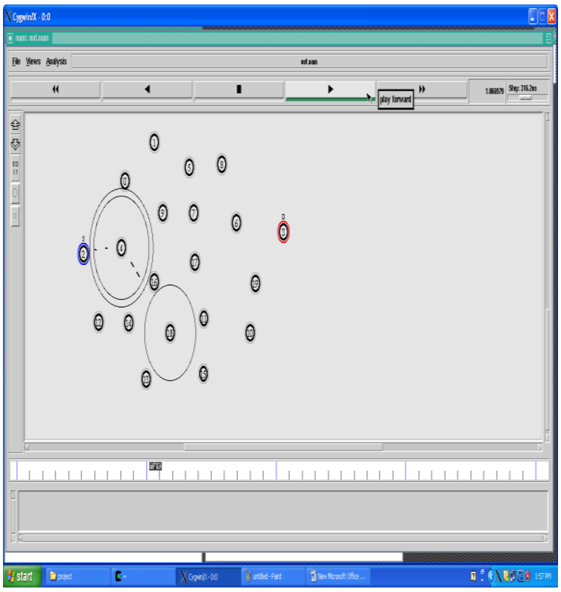

Fig.6 Transmission of data from node2 to node4

The source node that is node 2 start transmitting packet towards destination node 3.The packet flows through node 4 which is one hop neighborhood of node

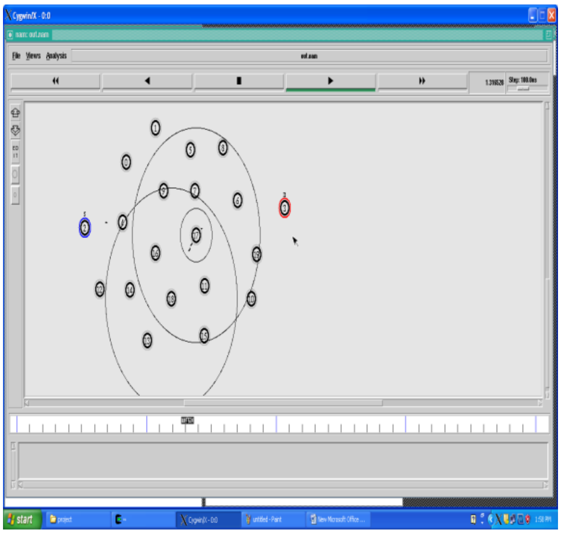

Fig.7 Transmission of data through node17

The packet is forwarded from node 4 to node 16, then from node 16 to node 17.Here the node 16 is two hop neighborhoods of source node .The packet lies within node 16, and then which find its one hop neighborhood and forward the packet to node 17.

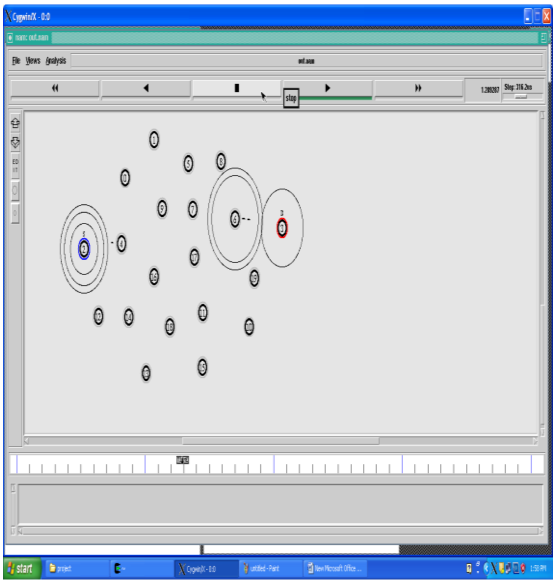

Fig.8 Transmission of data from node6 to node3

Finally the packet reaches the destination that is to node 7 from node 2 through the path (2-4-16-17-6-3)

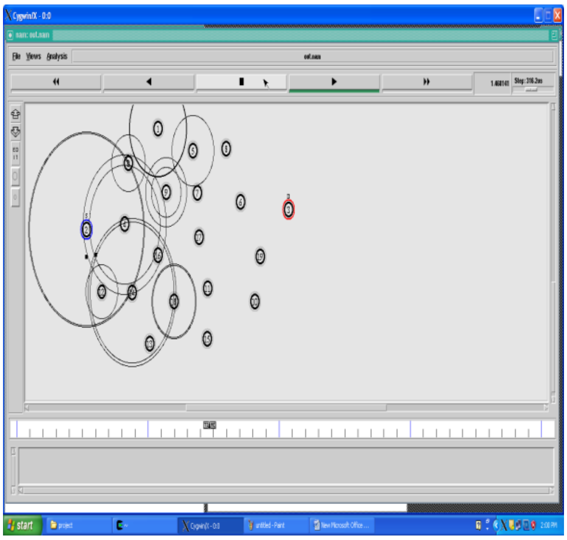

Fig.9 Packet drop from node 2

The above figure indicates that during the transmission of data due to overload some of the packets are dropped out from node 2.Since the queuing type we used here is Drop tail queue, a node which has packet beyond its queue length the packets are dropped out.

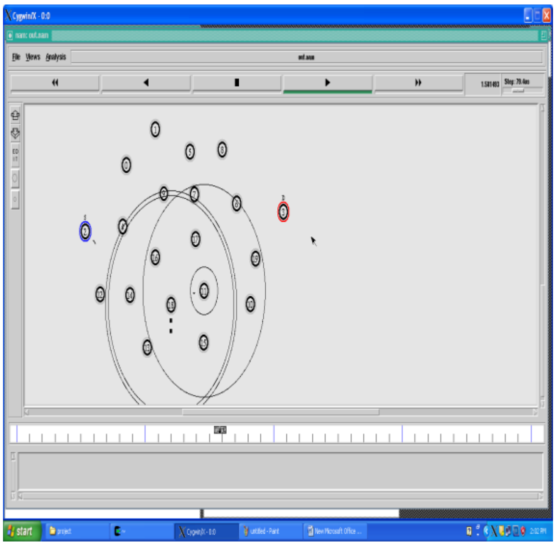

Fig.10 Packet drop from node 13

In this figure, from node 18 the packets are dropped out, hence the source node has to find another route to forward its data.

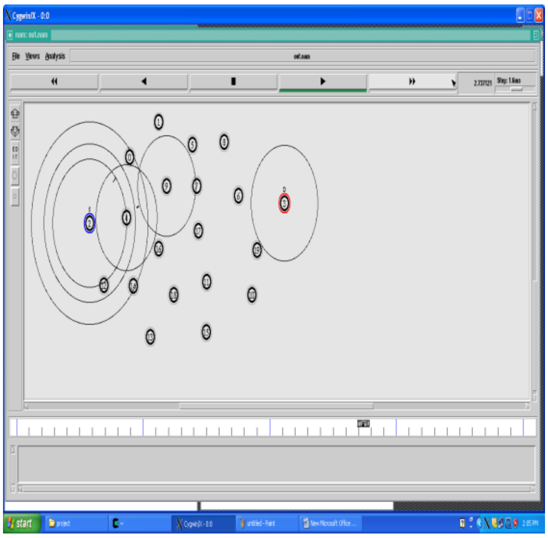

Fig.11 changing its path

Due to packet drop in node 18, the source node has start transmitting data through its one hop neighborhood that is node 0.

XGRAPH FOR PACKET LOST

Packet lost is defined as how many number of packets were lost in overall transmission.It is found by calculating the number of packets that was sent and the number of packets that was received.

Fig.12 XGraph for Packet lost

The above graph shows that the STORM protocol has less packet lost compared to AODV protocol. In case of any link failure or packet lost node switches to alternate path immediately, hence packet lost can be reduced.

XGRAPH FOR PACKET DELIVERY RATIO

Packet Delivery Ratio is defined as the total number of packets that was send by source node and the total number of packets that was received by the destination node.

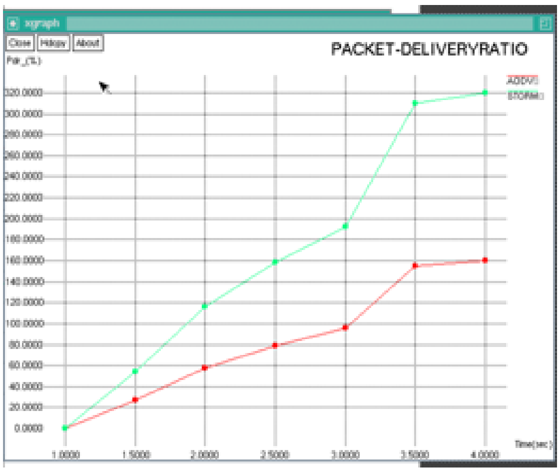

Fig.13 XGraph for Packet Delivery ratio

The Packet delivery ratio is higher for STORM when compared to AODV protocol. Since the routing tables are updated throughout the simulation in case of any link failure or packet lost node switches to alternate path immediately, hence packet delivery ratio can be improved.

XGRAPH FOR END TO END DELAY

Delay is calculated by finding the total number of time taken by the packet to reach the destination from source.

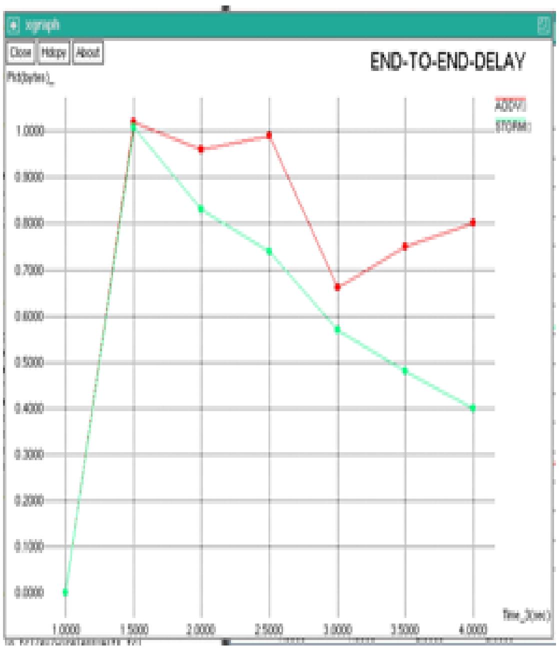

Fig.14 XGraph for End to End Delay

The figure shows that comparing to AODV, STORM has minimum delay. Since multiple paths are available in STORM delay is reduced.

XGRAPH FOR ENERGY CONSUMPTION

Energy consumption is defined as how much amount of energy is consumed by the network for transmission of packets.

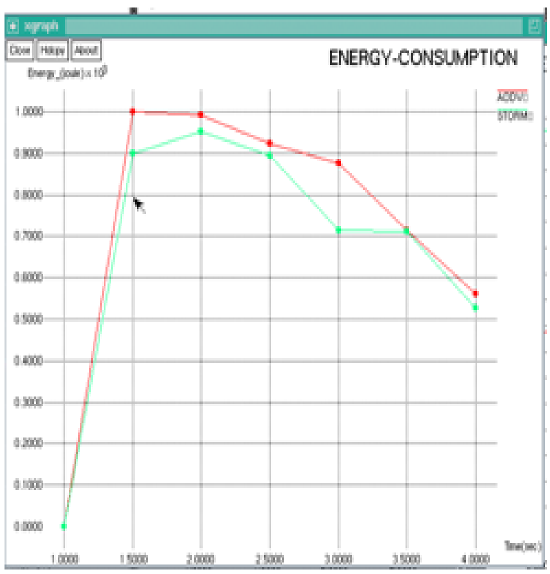

Fig.15 XGraph for Energy Consumption

The above figure shows that compared to AODV, STORM saves more energy. Energy Consumption is reduced to a small extent through reliable transmission of packets.

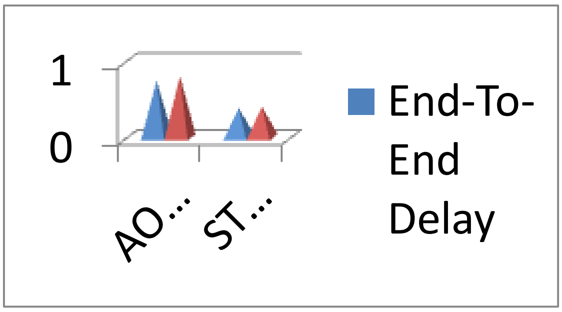

Fig.16 Comparison between AODV and STORM

4 CONCLUSION AND FUTURE WORK

We introduced STORM, a cross layer protocol framework for wireless ad hoc networks that integrate interest driven routing with priority based queuing for traffic management, end to end bandwidth reservations controlled by the routing, and distributed transmission scheduling. All these components work together to provide end to end delay and bandwidth guarantees to real time unicast and multicast data flows in multihop wireless network even when nodes move. The routing meshes established with the STORM are loop free at any time and that the end to end reservations established along routing meshes provide bounded delays to real time data packets. Our simulation results confirm correctness results showing that STORM is very scalable and robust for both unicast and multicast traffic. The results also shows that STORM scales better than the AODV protocol, and the end to end delay attained with the STORM describes the delay characteristics needed to support voice applications.

In future work is to overcome the need for time slotted channel access requiring clock synchronization. Further HELLO Messages, while allowing nodes to learn about neighbor changes in a timely manner, create extra control overhead and increase bandwidth consumption. We chose to include HELO Message in the design of STORM because we did not want STORM to have to rely on an underlying MAC-Sub layer protocol. However we are finding the ways of eliminating the need for HELLO Messages, while still allowing STORM to operate independently from such an underlying protocol.

REFERENCES

[1] C.R.Lin andM.Gerla,(April 2011) "Asynchronous Multimedia multihop Wireless Networks",Proc. IEEE INFOCOM,vol.1,pp. 118-125
[2] C.E.Perkins and P.Bhagawat, (2012) "Highly Dynamic Destination Sequenced Distance Vector Routing for Mobile Computers",Proc. ACM SIGCOMM,pp. 234-244
[3] E.Setton,T.Yoo,X.Zhu,A.Goldsmith and B.Girod, (Aug 2005) "Cross layer Design of Ad hoc Networks for Real time video streaming",IEEE Wireless Comm., vol. 12,no. 4,pp. 59-65
[4] Y.Wu,P.A.Chou,Q.Zhang,K.Jain,W,Zhu and S.Y.Kung, (Jan 2005) "Network Planning in Wireless Ad hoc Networks:A cross layer approach",IEEE J.Selected Areas in Comm.,vol.23,no.1,pp.136-150
[5] J.Yuan,Z.Li,W.Yu and B.Li,(Nov 2006) "A Cross layer optimization Framework for multihop multicast in Wireless Mesh Networks",IEEE J.Selected Areas in Comm., vol.24,no. 11,pp. 2092-2103