*International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue*
*1st International Conference on Advent Trends in Engineering, Science and Technology*
*"ICATEST 2015", 08 March 2015*

# Network Monitoring and Performance Analysis Model

Mr. Parag S. Naik[1], Mr. Manish K. Hadap[2]
[1] *M.Tech 4th Sem (I.T), Plot no 292 Ashok Nagar Sindhi Colony Nagpur, paragspnaik@gmail.com , 8855816838*
[2] *Asst. Professor,YCCE Hingna Wanadongri, manishhadap@yahoo.co.in , 9158888949*

**Abstract:** In mobile Ad-hoc networks, during the transmission of data, there is a chance of route failure, so that the route rediscovery should be done. Reactive routing protocol determines a route to a specific destination when a particular packet is going to send. The nodes in the network acts as routers, which transmits data packets from one neighboring node to another. We focus on monitoring network performance, packet tracker, route cost, and packet loss detection.The proposed approach improves network performance better than the other reactive routing protocol (AODV). The projected route recovery management technique can prevent the repeated collision and degradation in the network performance. Reactive routing protocol is an on-demand routing protocol for mobile ad-hoc networks, which uses routing tables to store routing information. During packets transmission, every intermediate node in the discovery route create routing table to store the information regarding neighbor node and the destination node information. The routing table information updated for every packet transmission during the message transmission. Neighbour node discovery problem is nothing but detecting the mobile nodes within one node's communication range. The location information of nodes over time has to be updated accordingly. Also multiple numbers of nodes should not be allowed to access the same destination at the same time there by avoiding packet loss and errors. Packet capturing (or packet sniffing) is the process of collecting all packets of data that pass through a given network interface. Capturing network packets in our applications is a powerful capability which lets us write network monitoring, packet analyzers and security tools. In this approach new model are used to monitor and analyze the network performance

**Keywords –** Mobile Ad-Hoc Networks, NS2, Reactive Routing Protocol, Route Discovery

## 1. INTRODUCTION

The nodes in the network acts as routers, which transmits data packets from one neighbouring node to another.We focus on monitoring network performance, packet tracker, route cost, and packet loss detection. In this project we are designing the network model which is used to find the shortest path among the nodes and maintain the routing table. The source node sends the packets to destination node by finding the shortest path among them. During packets transmission, every intermediate node in the discovery route create routing table to store the information regarding neighbor node and the destination node information. The routing table information updated for every packet transmission during the message transmission. Neighbour node discovery problem is nothing but detecting the mobile nodes within one node's communication range. The location information of nodes over time has to be updated accordingly. Also multiple numbers of nodes should not be allowed to access the same destination at the same time there by avoiding packet loss and errors.

The routing protocols for ad hoc wireless network should be capable to handle a very large number of hosts with limited resources, such as bandwidth and energy. The main challenge for the routing protocol is that they must also deal with host mobility, meaning that hosts can appear and disappear in various locations. Thus, all hosts of the ad hoc network act as routers and must participate in the route discovery and maintenance of the routers to the other hosts.

To monitor the network performance which include packet capture, packet loss, bandwidth detection, and analysis of various protocol which are used in the network. The node are connected with each other via link and they transfer the packets from node to node using the different bandwidth of the link.

## 2. REACTIVE ROUTING PROTOCOL

Reactive Routing Protocol Reactive routing protocol is an on-demand routing protocol for mobile ad-hoc networks, which uses routing tables to store routing information. The protocol comprises of three main functions like route discovery, route establishment and route maintenance. Reactive routing protocols, such as the AODv nodes have four types of message to communicate between each other. These are Route Request, Route Reply, Route Error and Hello messages with a key feature that doesn't require any distribution routing information and then, keep the routing information about the failure links. Reactive routing protocol, consists of three steps to find the optimal path. Initially, calculate the shortest path to the source node and created reverse route table. Then, filtered these paths to obtain optimal path for communication in the mobile adhoc network by calculating distance to the destination node. Then in the third step, a comparative analysis conducted in between three different protocols in terms of packet delivery ratio, routing overhead, throughput and average end to end delay, by using NS2 simulator

## 3. OPTICAL PATH FINDING APPROACH

We study the problem of selecting an optimal route in terms of transition probability and link available time. Finally we calculate optimal path between source and destination node by three steps, which execute and forwarding RREQ (route request) packets, RREP (route reply) packet and RRER (route error) packets. Experiments have been carried out using NS2 as network simulator ware and results encouraging.

i. Reverse Route Computation in RREQ

In mobile ad-hoc network each node will create a reverse route table when it receives a RREQ (route request), the RREQ is discards if it has already been processed. It records and indicates the route to the source node; otherwise the source address and the broadcast ID from RREQ resolve is there buffered to prevent it from being processed again. Furthermore, each node will calculate the distance every time, and most importantly, this distance is the key reason to choose the shortest path from the source node. Initially, when a node receives RREQ, it will create a reverse route entry which indicates the next hop (forwarding the RREQ) of the source node and calculate the distance between the next hop node and the source node. Second, each node will also make the similar decision when it receives RREQ and update reverse route table or discard RREQ.

ii. Reverse Route Computation in RREP

We have used the similar calculation mechanism to find the optimal path in forwarding RREP. The simply difference is that the distance we calculate in RREP is from the node forwarding RREP to the destination node.

iii. Reverse Route Computation in RERR

We have used the similar calculation mechanism to find the optimal path in forwarding RERR. The simple difference is that, When a node detects a link-break (i.e. receives a link layer feedback signal from the MAC protocol, neither receive passive acknowledgments nor receives hello packets for a certain period of time), and it performs a one hop data broadcast to its immediate neighbours.

## 4. RELATED WORK

Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Some frameworks have been proposed in mobile Ad-hoc network for performance-based routing protocol. Few of them are simulated, but we have used the concept of reverse reactive routing to find an optimal path between source and destination.

i. Simplified User's View of NS

NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup (plumbing) module libraries (actually, plumbing modules are implemented as member functions of the base simulator object).

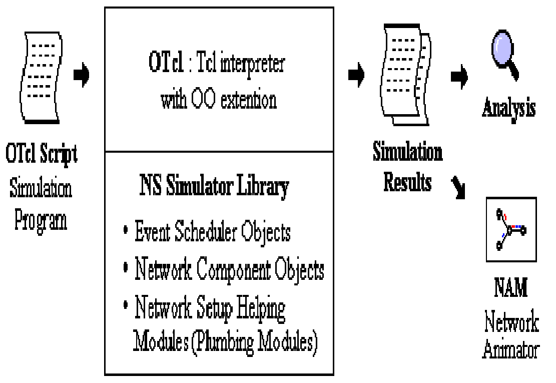

Fig 1. Simplified User's View of NS

In other words, to use NS, you program in OTcl script language. To setup and run a simulation network, a user should write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library, and tells traffic sources when to start and stop transmitting packets through the event scheduler. The term "plumbing" is used for a network setup, because setting up a network is plumbing possible data paths among network objects by setting the "neighbor" pointer of an object to the address of an appropriate object. When a user wants to make a new network object, he or she can easily make an object either by writing a new object or by making a compound object from the object library, and plumb the data path through the object. The power of NS comes from this plumbing. The data can be used for simulation analysis or as an input to a graphical simulation display tool called Network Animator (NAM). It can graphically present information such as throughput and number of packet drops at each link, although the graphical information cannot be used for accurate simulation analysis.

*ii.* Architectural View of NS

In this figure a general user (not an NS developer) can be thought of standing at the left bottom corner, designing and running simulations in Tcl using the simulator objects in the OTcl library.

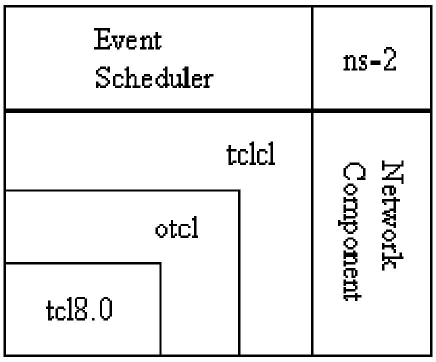

Fig.2. Architectural View of NS

The event schedulers and most of the network components are implemented in C++ and available to OTcl through an OTcl linkage that is implemented using tclcl. The whole thing together makes NS, which is an OO extended Tcl interpreter with network simulator libraries.

## 5. SYSTEM DESIGN AND IMPLEMENTATION

NS2 Programming contains the following steps

1. Create event scheduler
2. Turn on tracing
3. Creating Network
4. Monitoring

**Every ns2 script starts with creating simulator object**

```
set ns [new simulator]
```

**How to create node**

```
set n0 [$ns node]
set n1 [$ns node]
```

**Creating Link**

N0
1Mb, 10ms
N1

```
$ns duplex-link $no $n1 1Mb 10ms DropTail
```

**How to use Trace**

1. Generic trace
   for use with xgraph, and other things
2. Nam trace
   for use with visualization

```
# open trace file
 set tracefile [open out.tr w]
 $ns trace-all $tracefile

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
```

**Finish Procedure**

```
 proc finish {}
{
    global ns tracefile nf
    $ns flush-trace
    close $nf
    close $tracefile
    exec nam out.nam &
    exit 0
}
```

Finish procedure is forced to be called at the end with the line

```
$ns at 5.0 "finish"
```

Every tcl script must contain following statement

```
$ns run
```

## 6. CONCLUSION

We analyzed the neighbour node discovery problems in static and dynamic multihop Sensor Networks. The study was conducted to propose a reactive routing protocol, consists of three steps to find the optimal path. Initially, we calculated the shortest path to the source node and created reverse route table. Then, we filtered these paths to obtain optimal path for communication in the mobile adhoc network by calculating distance to the destination node. Then in the third step, a comparative analysis conducted in between three different protocols in terms of packet delivery ratio, routing overhead, throughput and average end to end delay, by using NS2 simulator. Finally, according to the average end to end delay, we have shown that DSR is lower than AODV, where the number of nodes we have used in our experiment. The "nam" window display the network node, packet flow and it capture the packet in the "tr" file. Using Wireshark tool I have analyzed the packet and tries to display that packet data

## REFERENCES

[1]. Bhabani Sankar Gouda, Chandan Kumar Behera, "A Route Discovery Approach to Find an Optimal Path in MANET Using Reverse Reactive Routing Protocol" 2012 National Conference on Computing and Communication Systems (NCCCS).

[2]. V. Karthikeyan, A.Vinod, P. Jeyakumar, "An Energy Efficient Neighbour Node Discovery Method for Wireless Sensor Networks".

[3]. R. Madan and S.Lall, "An energy-optimal algorithm for neighbor discovery in wireless sensor networks," In proc of Mobile Networks and Applications, ACM, pp. 317–326, 2006.

[4]. S. R. Biradar, KoushikMajumder, Subir Kumar Sarka and Puttamadappa C," Performance Evaluation and Comparison of AODV and AOMDV", In International Journal on Computer Science and Engineering Vol. 02, pp. 373-377, 2010.

[5]. M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," SIGMOBILE Mob. Comput.Commun. Rev., vol. 6, no. 3, pp. 92–93, Jun. 2002.

[6]. C.Perkins, E.Belding-Royer, S.Das "Ad hoc On-Demand Distance Vector (AODV) Routing", Feb.2003.

[7]. S. Kaushik, P.R.Deshmukh, "Comparison of effectiveness of AODV, DSDV AND DSR routing protocols in mobile ad hoc networks", International Journal of Information Technology and Knowledge Management July-December 2009, Volume 2, No. 2, pp.499-502.

[8]. R. Al-Ani, "Simulation and performance analysis evaluation for variant MANET routing protocols", International Journal of Advancements in Computing Technology, Volume 3, Number 1, February 2011.

[9]. A. Boukerche, B. Turgut, N. Aydin, M. Ahmad, L. Boloni, and D. Turgut, "Routing protocols in ad hoc networks: a survey," Computer Networks, vol. 55, no. 13, pp. 3032–3080, September 2011.