Volume 2, No. 10, October 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

PERFORMANCE ANALYSIS OF DYNAMIC ROUTING PROTOCOL IN MOBILE AD HOC NETWORK

Nikhil Patearia
Department of C.S.E
Samrat Ashok Technological Institute
Vidisha, M.P., India
nik.sati29@gmail.com

Abstract: Mobile ad hoc network is a collection of mobile nodes communicating through wireless channels without any existing network infrastructure or centralized administration. Due of the limited transmission range of wireless network, multiple "hops" are needed to exchange data across the network. Routing protocols used in ad hoc networks must automatically adjust to environments that can vary between the extremes of high mobility with low bandwidth, and low mobility with high bandwidth. This paper argues that such protocols must operate in an on-demand fashion and that they must carefully limit the number of nodes required to react to a given topology change in the network. I have embodied these two principles in a routing protocol called Dynamic Source Routing. As a result of its unique design, the protocol adapts quickly to routing changes when node movement is frequent, yet requires little or no overhead during periods in which nodes move less frequently. This paper generalizes the lessons learned from Dynamic Source Routing, so that we can be applied to the new routing protocols that have adopted the basic Dynamic Source Routing framework. The paper proves practicality of the Dynamic Source Routing protocol through performance results, and it demonstrates several methodologies for experimenting with protocols and applications in an ad hoc network environment.

Keywords- MANET, DSR, Multicast Routing Protocol.

INTRODUCTION

Mobile adhoc network (MANET) is a “on the fly” network of mobile nodes. Packets are routed through mobile nodes instead of any fixed base station. In a typical ad hoc network, mobile nodes come together for a period of time to exchange information. While exchanging information, the nodes may continue to move, and so the network must be prepared to adapt continually. In the applications we are interested in, networking infrastructure such as repeaters or base stations will frequently be either undesirable or not directly reachable, so the nodes must be prepared to organize themselves into a network and establish routes among themselves without any outside support. The idea of ad hoc networking is sometimes also called infrastructure-less networking, since the mobile nodes in the network dynamically establish routing among themselves to form their own network “on the fly” [1] and [3].

The basic routing problem is that of finding an ordered series of intermediate nodes that can transport a packet across a network from its source to its destination by forwarding the packet along this series of intermediate nodes. In traditional hop-by-hop solutions to the routing problem, each node in the network maintains a routing table: for each known destination, the routing table lists the next node to which a packet for that destination should be sent. The routing table at each node can be thought of as a view into part of a distributed data structure that, When taken together, describes the topology of the network. The goal of the routing protocol is to ensure that the overall data structure contains a consistent and correct view of the actual network topology. If the routing tables at some nodes were to become inconsistent, then packets can loop in the network. If the routing tables were to contain incorrect information, then packets can be dropped. The problem of maintaining a consistent and correct view becomes harder as there is an increase in the number of nodes whose information must be consistent, and as the rate of change in the actual topology increases.

The challenge in creating a routing protocol for ad hoc networks is to design a single protocol that can adapt to the wide variety of conditions that can be present in any ad hoc network over time. The routing protocol must perform efficiently in environments in which nodes are stationary and bandwidth is not a limiting factor. Yet, the same protocol must still function efficiently when the bandwidth available between nodes is low and the level of mobility and topology change is high. Because it is often impossible to know *a priori* what environment the protocol will find itself in, and because the environment can change unpredictably, the routing protocol must be able to adapt automatically. Most routing protocols include at least some *periodic* behaviors, meaning that there are protocol operations that are performed regularly at some interval regardless of outside events. These periodic behaviors typically limit the ability of the protocols to adapt to changing environments. If the periodic interval is set too short, the protocol will be inefficient as it performs its activities more often than required to react to changes in the network topology. If the periodic interval is set too long, the protocol will not react sufficiently quickly to changes in the network topology, and packets will be lost [1] and [2] and [3] and [4].
This paper concentrates on achieving high-performance multicast routing in multi-hop wireless ad hoc networks.

BACKGROUND

The Dynamic Source Routing protocol (DSR) is based on source routing, which means that the originator of each packet determines an ordered list of nodes through which the packet must pass while traveling to the destination. The key advantage of a source routing design is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets that they forward, since the packet's source has already made all of the routing decisions. This fact, coupled with the entirely on-demand nature of the protocol, eliminates the need for any type of periodic route advertisement or neighbor detection packets. The DSR protocol consists of two basic mechanisms: Route Discovery and Route Maintenance. Route Discovery is the mechanism by which a node **S** wishing to send a packet to a destination **D** obtains a source route to **D**. To reduce the cost of Route Discovery, each node maintains a Route Cache of source routes it has learned or overheard. Route Maintenance is the mechanism by which a packet's originator **S** detects if the network topology has changed such that it can no longer use its route to the destination **D** because some of the nodes listed on the route have moved out of range of each other.

SOURCE ROUTING

The routes that DSR discovers and uses are *source routes.* That is, the sender learns the complete, ordered sequence of network hops necessary to reach the destination, and, at a conceptual level, each packet to be routed carries this list of hops in its header. The key advantage of a source routing design is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets that they forward, since the packets themselves already contain all the routing decisions. Aggregating information about the network topology at the source of each packet allows the node that cares most about the packet, namely its source, to expend the appropriate amount of effort to deliver the packet. It also enables the explicit management of the resources in the ad hoc network. In some sense, DSR has an even stronger "end-to-end philosophy" than the Internet itself. Intermediate nodes in a DSR network maintain even less state than nodes in the core of the Internet, which must maintain up-to-date routing tables for all destinations in the network. Basing the routing protocol on source routes also has two additional benefits. First, the protocol can be trivially proved to be loop-free, since the source route used to control the routing of a packet is, by definition, of finite length, and it can be trivially checked for loops. Second, each source route is a statement that a particular path is believed to exist through the network. As source routes travel through the network riding on control packets, such as ROUTE REQUESTs or ROUTE REPLYs, or the data packets whose forwarding they control, any node overhearing a source route can incorporate the information it contains into its Route Cache. At the cost of no overhead above that used to carry out the normal operation of the protocol, the protocol itself spreads topology information among the nodes in the network. The information carried by a source route on a data packet also has the useful property that the more frequently heard routes and the most recently heard routes are the most likely to contain accurate information, since those routes are currently being tested by the packets flowing along them [3] and [4].

Although DSR uses source routes, and each packet is routed based on a discovered source route, recent improvements to DSR have made it so that most packets do not need to incur the overhead of carrying an explicit source route header.

ROUTE DISCOVERY

Route Discovery works by flooding a request through the network in a controlled manner, seeking a route to some target destination. In its simplest form, a source node **A** attempting to discover a route to a destination node **D** broadcasts a ROUTE REQUEST packet that is re-broadcast by intermediate nodes until it reaches **D**, which then answers by returning a ROUTE REPLY packet to **A**. Many optimizations to this basic mechanism are used to limit the frequency and spread of Route Discovery attempts. The controlled flood approach used by DSR works well in wired networks, but it is particularly well-suited to the nature of many wireless networks, where the communication channel between nodes is often inherently broadcast. A single transmission of a ROUTE REQUEST is all that is needed to re-propagate the REQUEST to all of a node's neighbors.

Route Maintenance-

When sending or forwarding a packet to some destination **D**, Route Maintenance is used to detect if the network topology has changed such that the route used by this packet has broken. Each node along the route, when transmitting the packet to the next hop, is responsible for detecting if its link to the next hop has broken. In many wireless MAC protocols, such as IEEE 802.11, the MAC protocol retransmits each packet until a link-layer acknowledgment is received, or until a maximum number of transmission attempts have been made. Alternatively, DSR may make use of a *passive acknowledgment* or may request an explicit network-layer acknowledgment. When the retransmission and acknowledgment mechanism detects that the next link is broken, the detecting node returns a ROUTE ERROR packet to the original sender **A** of the packet. The sender **A** can then attempt to use any other route to **D** that is already in its route cache, or can invoke Route Discovery again to find a new route for subsequent packets.

ROUTE CACHE

All the routing information needed by a node participating in an ad hoc network using DSR is stored in a Route Cache. Each node in the network maintains its own Route Cache, to which it adds information as it learns of new links between nodes in the ad hoc network, for example through packets carrying either a ROUTE REPLY or a source route. Likewise, the node removes information from the cache as it learns previously existing links in the ad hoc network have broken, for example through packets carrying a ROUTE ERROR or through the link-layer retransmission mechanism reporting a failure in forwarding a packet to its next-hop destination.

There is tremendous room for innovation inside the interface defined for the Route Cache, and this is intentional. An implementation of DSR may choose for its Route Cache whatever cache replacement and cache search strategy are most appropriate for its particular network environment. For example, some environments may choose to return the

shortest route to a node (the shortest sequence of hops), while others may select an alternate metric for the Get() operation.

I have experimented with many different types of Route Cache, and found that several general principles are helpful.

→The Route Cache should support storing more than one source route for each destination.

→If a node **S** is using a source route to some destination **D** that includes intermediate node **N**, **S** should shorten the route to destination **D** when it learns of a shorter route to node **N** than the one that is listed as the prefix of its current route to **D**. However, the cache should still retain the ability to revert to the older, longer route to **N** if the shorter one does not work.

→The Route Cache replacement policy should allow routes to be categorized based upon "preference", where routes with higher preferences are less likely to be removed from the cache. For example, a node could prefer routes for which it initiated a Route Discovery over routes that it learned as the result of promiscuous snooping on other packets. In particular, a node should prefer routes that it is presently using over those that it is not.

DESTINATION SEQUENCED DISTANCE VECTOR (DSDV)

DSDV is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom.

BASIC MECHANISMS

Each DSDV node maintains a routing table listing the "next hop" for each reachable destination. DSDV tags each route with a sequence number and considers a route R more favorable than R if R has a greater sequence number, or if the two routes have equal sequence numbers but R has a lower metric. Each node in the network advertises a monotonically increasing even sequence number for itself. When a node **B** decides that its route to a destination **D** has broken, it advertises the route to **D** with an infinite metric and a sequence number one greater than its sequence number for the route that has broken (making an odd sequence number). This causes any node **A** routing packets through **B** to incorporate the infinite-metric route into its routing table until node **A** hears a route to **D** with a higher sequence number [3] and [10].

AD HOC ON-DEMAND DISTANCE VECTOR (AODV)

AODV is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV.

TEMPORARILY ORDERED ROUTING ALGORITHM (TORA)

TORA is a distributed routing protocol based on a "link reversal" algorithm that finds and maintains routes via local relaxation of link direction. It is designed to discover routes on demand, provide multiple routes to a destination, establish routes quickly, and minimize communication overhead by localizing algorithmic reaction to topological changes when possible. Route optimality (shortest-path routing) is considered of secondary importance, and longer routes are often used to avoid the overhead of discovering newer routes. The actions taken by TORA can be described in terms of water flowing downhill towards a destination node through a network of tubes that models the routing state of the real network. The tubes represent links between nodes in the network, the junctions of tubes represent the nodes, and the water in the tubes represents the packets flowing towards the destination. Each node has a height with respect to the destination that is computed by the routing protocol. If a tube between nodes **A** and **B** becomes blocked such that water can no longer flow through it, the height of **A** is set to a height greater than that of any of its remaining neighbors, such that water will now flow back out of **A** (and towards the other nodes that had been routing packets to the destination via **A**).

BASIC MECHANISMS

At each node in the network, a logically separate copy of TORA is run for each destination. When a node needs a route to a particular destination, it broadcasts a QUERY packet containing the address of the destination for which it requires a route. This packet propagates through the network until it reaches either the destination, or an intermediate node having a route to the destination. The recipient of the QUERY then broadcasts an UPDATE packet listing its height with respect to the destination. As this packet propagates through the network, each node that receives the UPDATE sets its height to a value greater than the height of the neighbor from which the UPDATE was received. This has the effect of creating a series of directed links from the original sender of the QUERY to the node that initially generated the UPDATE. When a node discovers that a route to a destination is no longer valid, it adjusts its height so that it is a local maximum with respect to its neighbors and transmits an UPDATE packet. If the node has no neighbors of finite height with respect to this destination, then the node instead attempts to discover a new route as described above. When a node detects a network partition, it generates a CLEAR packet that resets routing state and removes invalid routes from the network [2], [9], [10].

PROPOSED TECHNIQUES FOR AD-HOC NETWORKS

In evaluating DSR and the other protocols propose in this paper, I use several sets of metrics. To characterize the basic performance of the protocols, I use a set of high-level summary metrics that are of interest to network users. To understand the internal functioning of the protocols, I used other sets of metrics: some of which are protocol specific and described as needed in the text, and some of which are general to all on-demand routing protocols and described below.

THE METRICS

The following three metrics capture the most basic overall performance of DSR and the other protocols implement in this paper:

→**Packet delivery ratio**: The ratio between the number of packets originated by the "application layer" sources and the number of packets received by the sinks at the final destination.

→**Routing overhead**: The total number of routing packets transmitted during the simulation. For
packets sent over multiple hops, *each* transmission of the packet (each hop) counts as one transmission. Routing packets are those that are originated by the routing protocol and do not also include user data. For protocols like DSR, which include both routing data and user data in the same packet, all the bytes of routing data in the packets are counted as routing overhead.

→**Path optimality**: The difference between the number of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when the packet is originated. Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support. This metric characterizes both the completeness and correctness of the routing protocol. Routing overhead is an important metric for comparing these protocols, as it measures the scalability of a protocol, the degree to which it will function in congested or low-bandwidth environments, and its efficiency in terms of consuming node battery power. Protocols that send large numbers of routing packets can also increase the probability of packet collisions and may delay data packets in network interface transmission queues. I did not include the number of IEEE 802.11 MAC packets or ARP packets in routing overhead, since the routing protocols I studied could be run over a variety of different medium access or address resolution protocols, each of which would have different overhead.
I also evaluated routing overhead in terms of total number of bytes of routing information transmitted. If a packet contains only routing information, the entire size of the packet, including IP headers, is counted as byte overhead. For packets that carry both user data and routing data, such as the source routes used by DSR, we counted the overhead as only the number of bytes in the source route and its associated sub header. In the absence of congestion or other "noise," path optimality measures the ability of the routing protocol to efficiently use network resources by selecting the shortest path from a source to a destination. Path optimality is calculated as the difference between the shortest path found internally by the simulator when the packet was originated, and the number of hops the packet actually took to reach its destination. Because packets can be stored in the buffers of nodes while the nodes move, it is possible for the length of
the optimal path for a packet to take through the network to change between the time the packet is originated and when it is received by the destination. As a result, packets are occasionally received by the destination after traveling fewer hops than the optimal path computed when the packet was originated. The effect is negligible in most experiments reported in this paper, but does become more pronounced as the packet buffer time, and hence the packet delay, increases.

MEASURING ROUTE DISCOVERY

Two additional metrics, *containment* and *discovery cost*, proved useful to evaluate the cost of on-demand Route Discovery.

→**Containment:** Containment is defined as the percentage of nodes that do not receive a particular ROUTE REQUEST. For a non-propagating ROUTE REQUEST, containment is equivalent to measuring the percentage of nodes in the network which are not neighbors (within transmission range) of the node originating the request. For a propagating ROUTE REQUEST, containment measures how far out the request propagates before running into either the edge of the network or a band of nodes with cached information about the target that is wide enough to stop further propagation. Values of containment approaching 1 indicate that a ROUTE REQUEST was well contained and interrupted very few nodes, whereas containment values approaching 0 indicate that most of the nodes in the network had to process the request.

→**Discovery cost:** The cost of a single Route Discovery is defined as

$$1+ \text{FwReq} + \text{OgRep} + \text{FwRep}$$

Where 1 represents the transmission of the original request, FwReq is the number of ROUTE REQUEST forwards, OgRep is the number of ROUTE REPLY originations, and FwRep is the number of ROUTE REPLY forwards. For each Route Discovery, this metric measures the number of routing packets (requests and replies) that were transmitted to complete the discovery. The average discovery cost is calculated as:

$$(\text{OgReq} + \sum \text{FwReq} + \sum \text{OgRep} + \sum \text{FwRep}) / \text{OgReq}$$

Where OgReq is the number of ROUTE REQUEST originations, and FwReq, OgRep, and FwRep are summed over all Route Discoveries.

➔**Proposed algorithm used at every node to obtain reliability, bandwidth and delay characteristics of preferred route-**

```
pkt_size = data_pkt_length;
 pkt_tr_time = pkt_size/BW;
  min_rel = 100;
 for(i=0;i<20;i++)
        {
        for(j = 0;j<10;j++)
              {
              if(node_rel[i][j] < min_rel)
                    {
                    min_rel[i] = node_rel[i];
```

```
                    }
                }
        }
         min_BW = 100;
for(i=0;i<20;i++)
        {
        for(j = 0;j<10;j++)
                {
                if(node_rel[i][j] < min_rel)
                        {
                min_BW[i] = node_BW[i];
                        }
                }
        }
best_rte = 0;
for(i=0;i<20;i++)
        {
        if(min_BW[i]*min_rel[j] > best_rte)
        best_rte_num = i;
        best_rte = min_BW[i]*min_rel[j];
        }
```

Mathematical Derivation of above algorithm:

Following are the considerations for mathematical derivation:

Average route length = **L**

Total Number of Nodes = **N**

Probability of Non-Reliable Node = $\mathbf{P_m}$

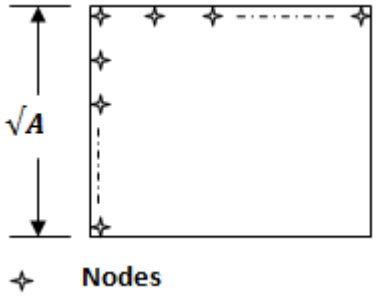
Area of Network = **A**

Distribution of Nodes Considered being Uniform

Hence Node density per square km.:-

$$D_n = \frac{N}{A} \dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

Maximum Range of Transmission = ***R***



As assumed above that nodes are uniformly distributed in square area then the distance between two nodes can be given as

$$d_n = 1/\sqrt{D_n} \dots\dots\dots\dots\dots\dots (2)$$

Hence to make the communication possible the eq^n. (3) must be true

$$R > d_n \dots\dots\dots\dots\dots\dots\dots\dots (3)$$

Calculation for route selection Possibilities

Let us start with first node, the first node have option to select any one of the surrounding nodes within its transmission range ***R,*** if we designate it by N_1 we can calculate its value as follows

$$N_1 = D_n R^2 \dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

And for second & other Nodes

$$N_2 = K_i D_n R^2 \dots\dots\dots\dots\dots\dots\dots (5)$$

Where K_i is the selection ratio for $\mathbf{i_{th}}$ node with respect to first node for considered square system K_i = **3/4** for all values of **i**.

Now maximum number of routes that can be established in assumed network.

$$Rt_{max} = D_n R^2 \prod_{i=1}^{L} (D_n R^2 K_i) \dots\dots (6)$$

$$Rt_{max} = (D_n R^2)^{L+1} \prod_{i=1}^{L} (K_i) \dots\dots\dots (7)$$

For K_i = **3/4**

$$Rt_{max} = (D_n R^2)^{L+1} \left(\frac{3}{4}\right)^{L} \dots\dots\dots\dots (8)$$

Now the probability of selection of Non-Reliable Node

$$Rt_{faulty} = P_m L \ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

Probability of packet loss

$$Pkt_{loss} = P_m L \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Now maximum number of Non-Reliable routes that can be established in assumed network

$$Rt_{fmax} = P_m L (D_n R^2)^{L+1} \left(\frac{3}{4}\right)^L \ldots (11)$$

Time & Overhead Calculations for proposed algorithm

Let we assume that only a few packets travels and dropped by Non-Reliable nodes to reduce traffic overhead

Acknowledgement Packet transmission Probability

$= P_a$

Hence probability of acknowledgement packet to catch a Non-Reliable route

$$P_{af} = P_a P_m L (D_n R^2)^{L+1} \left(\frac{3}{4}\right)^L \ldots\ldots (12)$$

Hence probability of acknowledgement packet to catch a correct route

$$P_{ac} = P_a (D_n R^2)^{L+1} \left(\frac{3}{4}\right)^L \ldots\ldots\ldots\ldots (13)$$

For detection of one faulty node we need to transmit $1/(P_{af} + P_{ac})$ acknowledgement packet.

If the network having N_f numbers of Non-Reliable nodes then we need.

$$Pkt_{ack} = \frac{N_f}{(P_{af} + P_{ac})} \ldots\ldots\ldots\ldots\ldots (14)$$

Both the containment and discovery cost metrics of Route Discovery are sensitive to a third parameter that characterizes the topology over which the discovery is running: the average *degree* of the nodes in the network. The degree of a node is the number of direct neighbors the node has, and the average degree measures how tightly interconnected the network is. As the degree of interconnectivity goes up, it is harder to contain a ROUTE REQUEST to one part of the network. In addition, the "branching factor" of a propagating ROUTE REQUEST increases, which causes more nodes to receive and process it. Thus, we would expect containment to decrease and discovery cost to increase in environments where the average node degree increases.

RESULTS

In our experiment is carried out with the simulator by performing several experiments that illustrate the performance of the system. The simulation parameters like number of nodes, terrain range etc. as given in table 1 along with their respective values are used to examine the performance of the network. The values can be adjusted according to requirements. After adjusting the values in this file, this file is executed. An output file is used to check the various parameters to analyze the performance of network.

Table 1: Simulation Parameters

Parameter	Value	Description
Simulation time	120 Sec	Maximum execution time
Terrain Dimensions	1000 X 1000 Mt	Physical area in which the nodes are placed in meters
Number of Nodes	10-300	Nodes participating in the network
Traffic Model	CBR	Constant Bit Rate link used
Node Placement	Uniform	Node placement policy
Mobility	0-10 (m/s)	Speed of node
Routing Protocol	DSR	Routing protocol used

NETWORK DELAY

Network delay is the total latency experienced by a packet to traverse the network from the source to the destination. At the network layer, the end-to-end packet latency is the sum of processing delay, packet, transmission delay, queuing delay and propagation delay. The end-to-end delay of a path is the sum of the node delay at each node plus the link delay at each link on the path. A higher value of end to end delay means that the network is congested and hence the routing protocol doesn't perform well. Figure 1 shows the network delay comparisons of proposed DSR scheme (red line) and existing scheme (blue line). In our mechanism network delay is decreases corresponding to simulation time increases. The figure 1 described that when simulation time less than 2m (minutes) network delay is maximum. After that simulation times are increases and the network delay is linearly decreases. On the other hand existing works networks delay is linearly decreases corresponding to simulation time are increases.

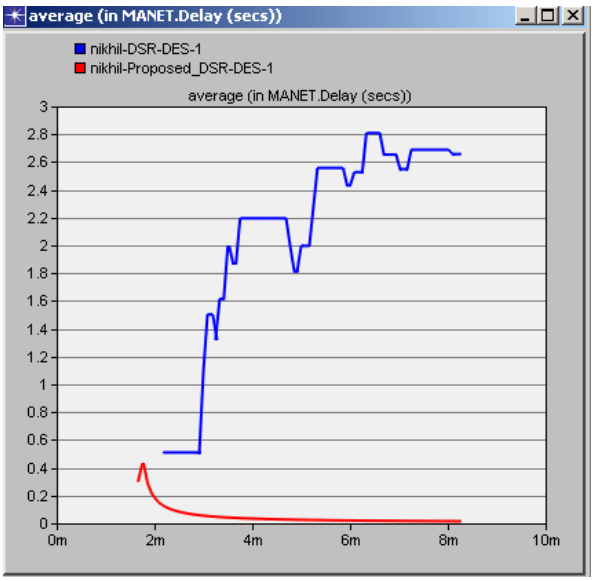


Figure 1: Network Delay

NETWORK TRAFFIC

Packet Delivery Ratio (PDR) is number of successfully delivered legitimate packets to number of generated legitimate packets. A higher value of PDR indicates that most of the packets are being delivered to the higher layers and is a good indicator of the protocol performance. Figure 2 shows the network traffic comparisons of proposed DSR scheme (red line) and existing scheme (blue line). In our mechanism network traffic is decreases corresponding to simulation time increases. The figure 2 described that when simulation time are increases and the network traffic is linearly decreases, because the number of packets receive linearly increases. On the other hand existing works networks traffic is linearly decreases corresponding to simulation time are increases.

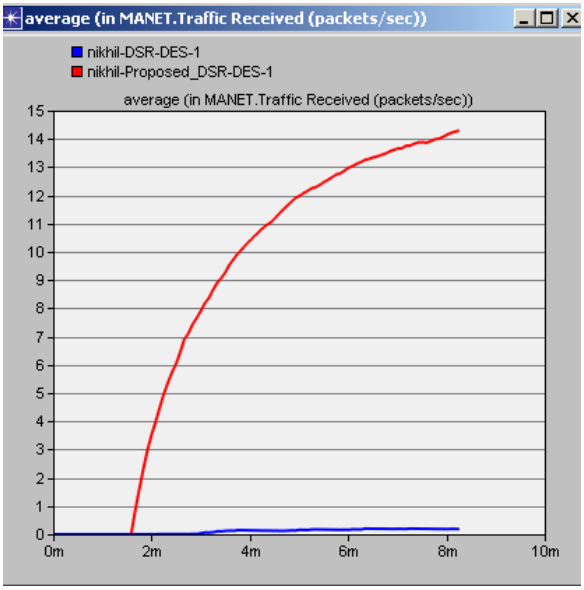


Figure 2: Network Traffic

CONCLUSION

The ability for nodes to form ad hoc networks in the absence of communication infrastructure is a critical area of current research. End to End delay for proposed DSR is less than existing DSR with the varying number of nodes and mobility. Finally from the above comparison it is concluded that multicasting protocol proposed DSR for ad-hoc networks perform well as compared to existing works in terms of end to end delay and network traffic.

There are existing communication needs which ad hoc networks can meet, such as military and commercial applications, and the development of ad hoc network technology will enable new classes of applications. With the potential for low cost deployment and high availability, coupled with the dropping costs of wireless transceivers, ad hoc networks are becoming economically and technologically feasible right now.

REFERENCES

[1] G.Narsimha Dr. A.Venugopal Reddy Prof. S .S.V.N.Sarma, "The Effective Multicasting Routing Protocol in Wireless Mobile Adhoc Network", IEEE Proceedings of the Sixth International Conference on Networking (ICN'07).

[2] Yogesh Chaba, Yudhvir Singh, and Manish Joon, "Performance Evaluation and Analysis of Cluster Based Routing Protocols in MANETs", IEEE 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp 64-66.

[3] Yudhvir Singh, Yogesh Chaba, Monika Jain, and Prabha Rani, "Performance Evaluation of On-Demand Multicasting Routing Protocols in Mobile Adhoc Networks", IEEE 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, pp 298-301.

[4] Vikram Bali, Rajkumar Singh Rathore, Amit Sirohi, and Prateek Verma, "A Framework to Provide a Bidirectional Abstraction of the Asymmetric Network to Routing Protocols", IEEE Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09, pp 1143-1150.

[5] Geetam S. Tomar, Manish Dixit & Shekhar Verma, "AODV Routing Protocol with Selective Flooding", IEEE 2009 International Conference of Soft Computing and Pattern Recognition, pp 682-686.

[6] J.Premalatha and P.Balasubramanie, "Enhancing Quality of Service in MANETS by Effective Routing", IEEE 2010.

[7] B.Malarkodi,P.Gopal and B.Venkataramani, "PERFORMANCE EVALUATION OF ADHOC NETWORKS WITH DIFFERENT MULTICAST ROUTING PROTOCOLS AND MOBILITY MODELS", IEEE 2009 International Conference on Advances in Recent Technologies in Communication and Computing, pp 81-84.

[8] A K Daniel, R Singh and Zubair Khan, "Position Based Multicast Routing Protocol for AD-hoc Wireless Network Using Backpressure Restoration", IEEE 2010 2nd International Conference on Computer Engineering and Technology, pp 458-462.

[9] Afsaneh fathi and Dr hasan taheri, "Enhance Topology Control Protocol(ECEC) to Conserve Energy based clustering in Wireless Ad Hoc Networks", IEEE 2010, pp 356-360.

[10] Pratibha Tomer, and Munesh Chandra, "An Application of Routing Protocols for Vehicular Ad-hoc Networks",

IEEE 2010 International Conference on Networking and Information Technology, pp 157-160.

