An Effective Gateway Discovery Mechanism in an Integrated Internet-MANET (IIM)

Khaleel Ur Rahman Khan*, Ehthesham M*, Kumar M*, Rafi U Zaman*

** CSE Dept. M.J. College of Engg. and Tech. , Hyderabad, India,*

khaleelrkhan@mjcollege.ac.in,
ethesham_mohdabdul@yahoo.com,agarwal_manishkumar1@yahoo.com,
rafi.u.zaman@gmail.com

Abstract

Ad hoc networking allows portable devices to establish communication independent of a central infrastructure. The Integration of the MANETs and infrastructure networks such as Internet extends the network coverage and increases the application domain of the MANET. Communication between a mobile device in an ad hoc network and a fixed device on the Internet is achieved using gateways, which acts as a bridge between them. The ad hoc routing protocol AODV is extended and used to achieve interconnection between a MANET and the Internet. In the existing approaches of gateway discovery either interface queue length or the minimum hop metric criteria or a combination is used for selecting the gateway by the mobile nodes. In the proposed approach, an efficient proactive gateway discovery algorithm is devised that takes into account the length of the routing queue in addition to the minimum hop count metric for selecting an efficient gateway and also selecting the routes to other mobile nodes. This is a novel strategy of gateway discovery. This approach is implemented by calculating the load along a path and updating the routing entry as the route requests are processed from one mobile node to the other. This allows usage of updated routes without waiting for the gateway advertisements for updating the routing entries. It also reduces the delay along the path for the packets traversing within the ad hoc network. The use of this concept increases the throughput by choosing the less congested paths and reduces the routing overhead. In this paper the impact of this new approach is investigated. Simulation results indicate that our protocol outperforms other approaches.

9781-4244-3941-6/09/$25.00 ©2009 IEEE

1. Introduction

The Integration of the MANETs and infrastructure networks such as Internet extends the network coverage and increases the application domain of ad hoc networks.

The difference in the network architectures of the MANET and the Internet imposed various sorts of assumptions on the structure, topology of the underlying networks and on the communication patterns of mobile nodes in both networks. Integrating these two networks into a hybrid network is a challenging problem due to these differences. This interconnection is achieved by using gateways, which act as bridges between a MANET and the Internet. Communication of the mobile devices in an ad hoc network and a fixed device in the Internet requires the modification of the ad hoc routing protocol. Before a mobile node can communicate with the Internet host it needs to find a route to a gateway. Most of the previous approaches use the shortest path for the selection of gateway for onward transmission of data from MANET nodes to the wired hosts. In the proposed approach of Kumar et. al. [8], the length of the routing queue in addition to minimum hop count metric is used for the selection of gateway by the mobile node. This approach has been extended by updating the path to the gateway on the request of mobile node which facilitates efficient handoff from one gateway to another, and thus maintaining continuous connectivity to the fixed host. Another extension is that routing queue length and min hop count metric is not only used to discover the routes to the gateway but also for the routing in the local ad hoc domain among the ad hoc host. Further, the occupancy level of each of the nodes is updated every short interval of time. This updated occupancy is sent to all the neighbors within the radio transmission range within the hello packet. This reduces the delay along the path traversal. The main advantage of this mechanism is the increase in the packet delivery ratio by choosing the less congested paths. The load along a path is calculated and the routing entry is updated as the route request is being processed. This updates the default route i.e. the route to a gateway whenever a request is received by a gateway or a reply is sent by a gateway. It also reduces the delay and allows usage of updated routes without waiting for the gateway advertisements for updating the routing entries. The simulation results indicate that the proposed approach outperforms other strategies.

The remainder of the paper is organized as follows: Section 2, overviews related work in the area of MANET-wired interconnectivity and gateway discovery. In section 3, we presented our proposed algorithm for gateway discovery. Simulation results are presented and discussed in section 4. Finally, section 5 concludes the paper.

2. Related Work

In the literature, many strategies have been proposed for interconnecting MANET with the Internet. Some papers also discuss methods to interconnect MANET and Internet independent of any protocol. But, only few papers have discussed gateway discovery mechanisms. A comprehensive survey on the literature of integration strategies can be found in [1]. Here we briefly list few of the popular strategies.

Jonsson et al. [2] proposed an approach, called MIPMANET based on AODV [3], but it provides Internet access by using tunneling and Mobile IP with foreign agent care-of-addresses. MIPMANET allows a visiting node to switch from its current foreign agent to a new one, a phenomenon known as handoff, only if it is at least two hops closer to the new one

Hamidian et al. [4] proposed a solution which provides Internet connectivity to ad hoc networks by modifying the AODV routing protocol. Three methods of gateway discovery for a mobile node to access the Internet are provided: proactive, reactive and hybrid approach..

Ratanchandani et al. [5] discusses a hybrid gateway discovery approach. AODV and two Mobile IP foreign agents are used to interconnect MANET and the Internet. However, the TTL of the foreign agent's advertisements is limited.

Lee et al. [6] proposed a more sophisticated approach in which advertisements are sent out only when changes in the topology are detected.

Bin et al. [7] proposed an adaptive Gateway discovery scheme. In this scheme, the TTL values of Agent Advertisements (GWADV messages) are dynamically adjusted according to the mobile nodes Internet traffic and their related position from Internet Gateways with which they registered.

Rakesh Kumar et al. [8], proposed an efficient proactive gateway discovery algorithm that takes into account the size of the interface queue in addition to the traditional minimum hop metric to select an efficient gateway.

3. Proposed Mechanism for Gateway Discovery

As stated earlier, in most of the existing strategies for interconnecting MANET and Internet, the shortest path algorithm which takes into account the number of hop counts to the gateways is used, which can be described by the following formula [8].

$$H(s,d) = \begin{cases} \min\left\{H(p) : s \xrightarrow{p} d\right\} & \text{if there is a path from s to d} \\ \infty & \text{otherwise} \end{cases}$$

Where p is a path, H is the shortest-path distance from s to d which is the sum of its links in any path p, s is the source node, and d is the destination node (i.e., Internet gateway). One of the advantages of using the shortest path selection algorithm with the hop count attribute is rapid convergence and thriftiness of resources. This attribute enables a mobile node to reach a wired network using the minimum number of hops. However, if all mobile nodes select the nearest gateway as their serving Internet gateway, then this gateway would become a bottleneck, resulting in high processing latency. This weakness motivates the selection of Internet gateway based on some other metrics.

In this paper, using AODV routing protocol, an additional metric pathload_ is introduced for updating the route to the gateways and the mobile nodes, while processing the requests. This would take into account the packet load present along all the nodes traversed along the route to the gateway. The occupancy_ is used to get the packet load of a particular node. This metric is included by modifying the hdr_cmn field in the structure of the packet. This is used to update the packet load along a routing path in a routing entry and updates the routes to those having lesser load with a combination of minimum hop count. An additional attribute q_occupancy_ is added to the Class mobile node for having the value of the updated packet load for that particular node. An additional metric for selection of a particular gateway called gateway_adv_queue which is computed taking into account the effect of routing queue occupancy level along a route. For this, we propose modification in the gateway advertisement message and also in the routing table maintained at each mobile node. An additional field rt_qlen_metric is added to record the effect of this metric along a route to the Internet gateway in routing table of each mobile node. When a packet is received, the pathload_ metric is compared with the rt_qlen_metric to update the routing entry between the nodes. If the packet is received by a gateway, a reply is sent by setting the RREP_I flag and the routes to the gateways are updated. We include an additional timer node timer that updates the q_occupancy_ attribute of each of the mobile nodes for every short interval of time. This will reduce the end-to-end delay because the occupancy level of the node is already calculated. The modified structure of gateway advertisement message and the hdr_cmn used to implement our approach are depicted in Figures 1 and 2.

The proposed algorithm uses both the minimum number of hops and the least load path to find a route between the mobile nodes and to select the gateway unlike as in the case of [8] wherein the metric is used to select only the gateways.

$$L(s,d,h) = \begin{cases} \min\left\{L(p,h): s \xrightarrow{p} d, h: H(s,d)\right\} & \text{if there is apath from } s \text{ to } d \\ \infty & \text{otherwise} \end{cases}$$

Where L is the packet load along the path, s is the source node, d is the destination node, h is the number of hops between the source and the destination, p is the path between the source and the destination. This will select a path having the minimum packet load and the minimum number of hops.

Internet gateway at discrete intervals broadcasts gateway advertisement message. A mobile node that receives gateway advertisement message, periodically updates its route entry for the gateway. A mobile node creates a new route entry in its routing table for every mobile node as well as for every fixed node that it wishes to communicate with. A routing table maintained at a mobile node MN_A wishing to communicate to fixed node FN_ (0.0.1) and some other mobile nodes in the ad hoc network is shown in Table 1. The organization of routing table is same as is given in [8]. MN_B, MN_C, MN_D and MN_E are mobile nodes. GW_X represents gateway node. The next hop entry for fixed node FN_X (0.0.1) is set to default (-50), so that mobile node should look up this default entry and discover an appropriate gateway for forwarding packets.

<table>
<tr><td>Type</td><td>Reserved</td><td>Prefix Size</td><td>Hop Count</td></tr>
<tr><td colspan="4">Broad_Cast ID</td></tr>
<tr><td colspan="4">Destination IP Address</td></tr>
<tr><td colspan="4">Destination Sequence Number</td></tr>
<tr><td colspan="4">Source IP Address</td></tr>
<tr><td colspan="4">Lifetime</td></tr>
<tr><td colspan="4">Gateway_adv_queue</td></tr>
</table>

Figure1. Structure of modified Advertisement Gateway message

<table>
<tr><td>Ptype_</td><td>Size_</td><td>Uid_</td><td>Error_</td></tr>
<tr><td>Iface_</td><td>Direction_</td><td colspan="2">Ts_</td></tr>
<tr><td colspan="4">Next_hop_</td></tr>
<tr><td colspan="4">Prev_hop_</td></tr>
<tr><td colspan="4">Last_hop_</td></tr>
<tr><td colspan="2">Occupancy_</td><td colspan="2">Pathload_</td></tr>
</table>

Figure2. Structure of modified hdr_cmn in a packet

The second entry indicates the gateway chosen by the mobile node for its Internet connection. The third entry indicates next hop towards the particular Internet gateway. When a request is being processed the load that is present along a path from one mobile node to another is also updated. This load is compared with the newly received request and it is updated if the path is a less congested one.

Table 1. Routing table of a mobile node (MN_A) for Internet connectivity

Entry #	Destination Address	Next hop address	Physical hops	Rt_qlen_metric
1	FN_X(0.0.1)	Default(-50)	∞	∞
2	Default(-20)	GW_X(1.0.0)	3	5.8
3	GW_X(1.0.0)	MN_A(1.0.3)	3	5.8
4	MN_B(1.0.5)	MN_C(1.0.8)	7	1.2
5	MN_D(1.0.7)	MN_E(1.0.10)	5	4.7

3.1 Computing Queue Occupancy

Figure 3 taken from [8], illustrates congestion level accumulation i.e. the packet load along a path. In this, the avg_q_occupancy of mobile node N2 can be calculated by q_occupancy of mobile node N2 itself, and $nb\_q\_occupancy_1$, $nb\_q\_occpancy_2$, $nb\_q\_occupancy_3$ and $nb\_q\_occupancy_4$ of nodes G, N1, N5 and N6. The number of neighbors of node N2, i.e. n is 4. Before initiating a gateway discovery, a gateway will compute its own avg_q_occupancy and fill it in gateway_adv_queue which is an additional field of gateway advertisement message

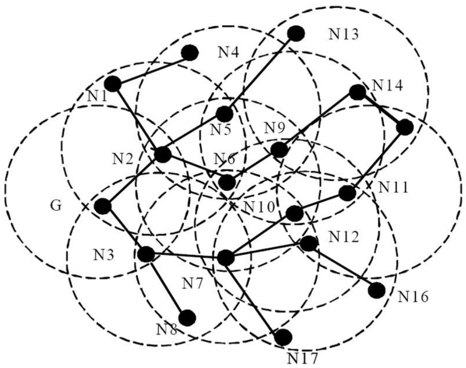

Figure 3. Proactive Gateway Advertisement

. This field contains total load of whole path traversed so far. Here load is referred to as the number of packets in the routing queue of a node. If the packet is being sent or received by the mobile node then its queue occupancy is placed in the pathload_ of the packet. When the next node along the path receives the packet it updates the routing entry to the mobile node that has sent the packet, includes its own load in the pathload_ of the packet and then forwards the packet. The algorithm for computation of routing queue occupancy level is given below which is same as used in [8] with some minor changes.

Algorithm 1: **Computation of Congestion Level**

Step 1: Update the q_occupancy_ of the node for every nodetimer execution

Step 2: Each mobile node computes the average queue occupancy (avg_q_occupancy) using the mobile node's current queue occupancy in its local area.

Step 3: Every mobile node maintains and updates its neighbor's information by exchange of periodic Hello packets which is a one hop packet containing the sender's address and current queue occupancy.

Step 4: **C**ompute level of Congestion at a particular node along a path i.e. avg_q_occupancy using equation (1) as given below.

$$\text{avg\_q\_occupancy} = \frac{\text{q\_occupancy} + \sum_{k=1}^{n} \text{nb\_q\_occupancy}_k}{n+1} \qquad (1)$$

where, q_occupancy $\leftarrow$ node's own queue occupancy
$\text{nb\_q\_occupancy}_k \leftarrow$ node's neighbor's queue occupancy, and
n $\leftarrow$ number of neighbor nodes

Step 5: **C**onvert congestion level equivalent to physical hops by dividing avg_q_occupancy by 1000.
avg_q_occupancy_eq_phy_hop = avg_q_occupancy/1000;

Step 6: Repeat step 1.

3.2 Internet Gateway Selection Algorithm

In scenarios having multiple gateways, a mobile node receives advertisements from multiple gateways. Here, a proactive approach is used wherein every gateway sends its advertisement periodically. Whenever a mobile node receives a non duplicate gateway advertisement, it updates the value of rt_qlen_metric and selects a suitable gateway by comparing the gateway_adv_queue in the gateway advertisement message with the rt_qlen_metric in its routing table. Algorithm 2 describes the process of selecting an efficient gateway. The algorithm is the same as used in [8] with minor changes.

Algorithm 2: **Selecting an Effective Internet Gateway**

Step 1: // Initialization
(i) gateway_adv_queue = 0;
(ii) HopCount = 0;

Step 2: Internet gateway broadcast an advertisement.

Step 3: If the received Internet gateway advertisement is a duplicate node, then drop it and exit.

Step 4: For each unique Internet gateway advertisement arrival at a particular mobile node,
gateway_adv_queue= gateway_adv_queue +avg_q_occupancy_eq_phy_hop + 1;

Step 5: //Check whether an advertisement is from the same gateway If (advertisement from the same gateway) && (ad_dst_seqno > default_rt_seqno) then

Step 6: // Whether an advertisement reached through a less congested path?
If (gateway_adv_queue =< rt_qlen_metric) then

Step 7: // Update routing table of mobile node
// hop_count refers a field in a mobile routing table
(i) rt_qlen_metric = gateway_adv_queue;
(ii) hop_count = HopCount;
(iii) Also update route table next hop address towards this gateway;

Step 8: // In case gateway advertisement is being received from a different
// gateway, make a hand off to another Internet gateway.
else if (advertisement not from the same gateway &&
gateway_adv_queue <= rt_qlen_metric) then

Step 9: // Update routing entries in the mobile node
(i) Make new gateway as default gateway;
(ii) Also update default_rt_seqno, rt_qlen_metric, ad_src, hop_count, default_rt_expiration_time, etc in the mobile node routing table;

Step 10: else keep on using current gateway as the default gateway;

Step 11: Repeat step 2;

In the above gateway selection algorithm, for every fresh gateway advertisement received from the same gateway, the value of rt_qlen_metric in a mobile node routing table is replaced only when gateway_adv_queue field value of gateway advertisement is less than the rt_qlen_metric field value in the routing table. In case, the gateway_adv_queue metric field value in gateway advertisement becomes less than the rt_qlen_metric, and also the gateway advertisement comes from a different gateway, the mobile node need to register with new gateway as the default gateway otherwise default gateway will remain the same. The congestion level of the route is accumulated in the gateway advertisement as it traverses the network.

3.3 Enhancing the Routing Algorithm

This is the extension to the work done in [8]. In MANET, a mobile node uses multiple hops for communicating with the other nodes in the ad hoc domain. This feature helps us in updating the routing entries from one mobile node to another when a request is being processed. We make use of the minimum number of hops and the packet load present between the two mobile nodes to update the routing entry between the nodes. Whenever a mobile node wants to send a request or a reply packet the load of the node is put into the packet. The algorithm for updating the routing algorithm is described in Algorithm 3.

Algorithm 3: **Updating the Routing Algorithm for MANET nodes**

Step 1 : //Initialization
    rt_qlen_metric = INFINITY; Hop_count = 0;
Step 2: //A mobile node sends a request or reply
    Pathload_ = q_occupancy_;
Step 3: //if the request arrives at the mobile node
    Hopcount**++;**
Step 4: // the request is not sent by gateway
    If (RREP_I is not set) If(pathload_ <= q_len_metric) then

Step 5: // Update routing table of mobile node
    // hop_count refers a field in a mobile routing table
        (i) rt_qlen_metric = pathload_;
        (ii) hop_count =Hopcount;
        (iii) Also update route table next hop address towards this mobile node;
    If (thisnode != dst) then
Step 6: //forward the packet along the route
    Pathload_ = pathload_ + q_occupancy_;
    forward packet along the route available to destination Create a reverse routing entry.
Step 7: // the packet is received by a gateway
    Send a reply with RREP_I flag set to the source from which packet received,
    Pathload_ = q_occupancy_; Set RREP_I;
    //this packet traverses along the reverse route entry to the source and the entire node along the route may update the route to the gateway
Step 8: // If the packet was sent by the gateway
    else//when RREP_I is set
    Packet received from the registered gateway
    If(advertisement from the same gateway)
    // Whether a packet reached through a less congested path?
Step 9: If (pathload_ =< rt_qlen_metric) then

Step10: //update the default route of the mobile node receiving the packet
    (i) rt_qlen_metric = pathload_;
    (ii) hop_count = HopCount;
    also update route table next hop address towards this gateway
Step 11: // if received from another gateway
    else if (advertisement not from the same gateway && pathload_ <= rt_qlen_metric) then
Step 12: // Update the following entries in the mobile node
    (i) Make new gateway as default gateway;
    (ii)Also update default_rt_seqno, rt_qlen_metric, src_, hop_count, default_rt_expiration_time, etc in the mobile node routing table;
Step 13: Repeat Step 3

4. Simulation Results Discussion

The proposed strategy is implemented using the network simulator ns-2.31 [9]. The performance of the proposed strategy is compared with the Kumar et al [8] and Hamidian [4] proactive discovery approaches in similar simulation environment.

4.1 Simulation Setup

The scenario consists of 15 mobile nodes, two fixed hosts, two routers and two gateways. The topology is a rectangular area with 800 m length and 500 m width. All the fixed links have a bandwidth of 10Mbps. Each wireless transmitter has a radio range of 250m. All simulations were run for 900 seconds of simulation time. Five of the 15 mobile nodes are constant bit rate (CBR) traffic sources sending data packets with a size of 512 bytes, to one of the two fixed hosts. They are distributed randomly within the mobile ad hoc network. All the five traffic sources start sending data packets after 50 seconds of the start of simulation time. Data rate of mobile nodes from 5 packets/sec (5 * 512 * 8 = 20 Kbps) to 25 packets/sec (25 * 512 * 8 = 100 Kbps). The mobility model is random waypoint model [10].

4.2 Performance Metrics

The following performance metrics were considered for the comparison of the proposed approach and that of Hamidian.

Packet Delivery Ratio : It is defined as the total number of data packets received divided by the total number of data packets sent at all the mobile nodes present in the simulation.

Routing Overhead: It is defined as the ratio of the AODV packets to the data packets sent and received by all the mobile nodes.

End-to-End Delay: It is defined as the delay for sending packets from source node to the fixed host.

4.3 Results Discussion

The performance of the proposed mechanism has been compared with two of the existing approaches [4, 8] as said earlier. The performance comparison has been done between the proposed and existing approaches with respect to the metrics packet delivery ratio, routing overhead, end-end delay by varying the speed of the mobile nodes and the traffic load (packets/sec). Figures 4 to 6 shows the performance of the proposed strategy with respect to varying the ad hoc node speed from 5 to 25 m/s. Whereas Figures 7 to 9 shows the impact of the traffic load measured in packets/second on the packet delivery ratio, routing overhead, end-end delay. It may be observed that the proposed approach outperforms both of the Hamidian [4] and Kumar's [8] approach in terms of the defined metrics. Each data point in the graphs is the average of three simulation runs.

In the proposed approach, the routes from a mobile node to the other node and the gateways are updated whenever a request or a reply is sent from the mobile node to its specified destination. This will update the routes to all the nodes that are present along the path from the source node to the destination node. Since the routes are getting updated every time they get along the path for a request or a reply it will reduce the end-to-end delay for sending a packet, the simple reason being the time to find a route to the destination is being eliminated most of the times. As the end-to-end delay is being reduced, the number of packets that are being sent per unit of time increases, thereby providing better throughput in our case. Since the routes to the other mobile nodes and gateways are being updated for every request or reply, this also reduces the AODV control packets that are being sent to find a route from a source to the destination. Therefore it reduces the control packet overhead. The paths that are updated are being done based on the effect of both the hop counts metric and the packet load metric that is present between the two nodes. This increases the packet delivery ratio.

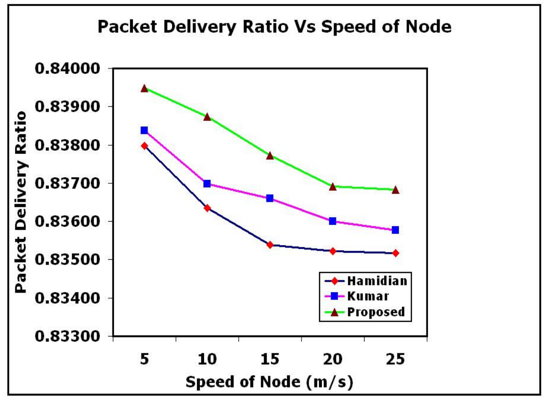

Figure 4. Packet delivery ratio vs Node speed

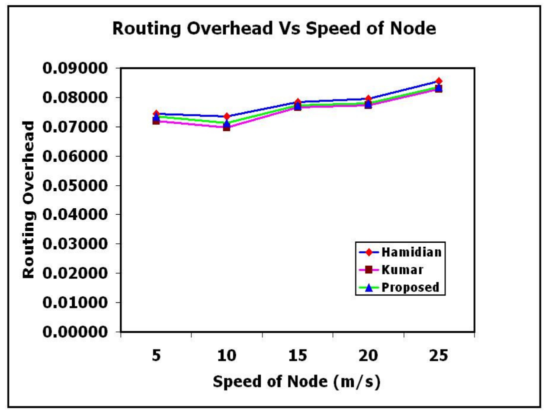

Figure 5. Routing Overhead vs Node Speed

5. Conclusion

In this paper, an effective gateway discovery mechanism has been developed and implemented in heterogeneous network environment of ad hoc network and a fixed wired network. The significant characteristics of the proposed strategy is that, it takes into account both the hop count and the routing queue length in the selection of gateway rather than taking only the hop count metric. The same metric is also used for routing within the ad hoc domain with intent to boost up the performance of the system. This mechanism reduces the congestion in the network and thereby improves the overall performance in terms of packet delivery ratio, end-end delay and the routing protocol overhead. The performance of the proposed scheme has been compared with the Hamidian and Kumar approaches and it is found that the proposed approach outperforms both of the strategies, as exhibited in the simulation results. Presently, in the implementation the proactive approach has been used for gateway discovery. In future, the results can be compared with the hybrid approach of gateway discovery.

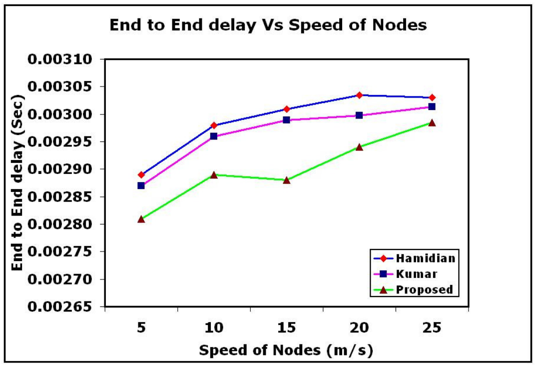

Figure 6. End-End delay vs Node Speed

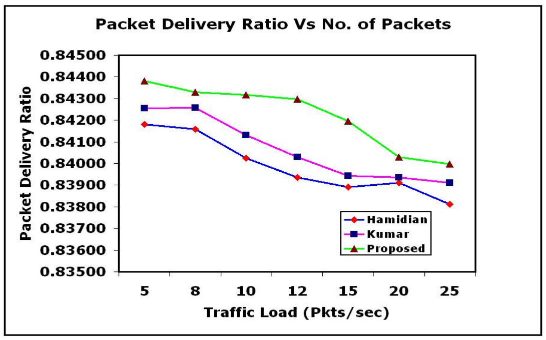

Figure 7. Packet Delivery ratio vs Traffic Load

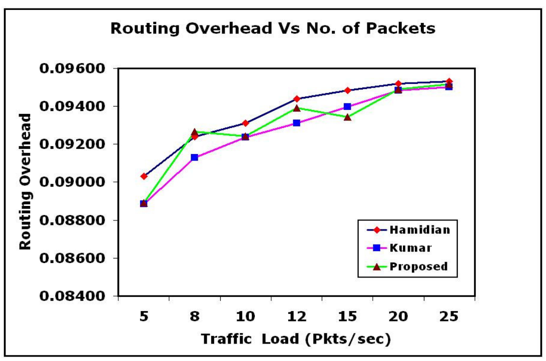

Figure 8. Routing Overhead vs Traffic Load

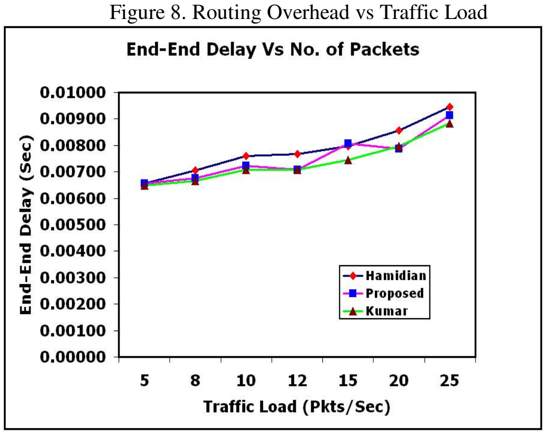

Figure 9. End-End delay vs Traffic load

6. References

[1] Khaleel Ur Rahman Khan, Rafi U Zaman, A.Venugopal Reddy, “Integrating Mobile Ad Hoc Networks and the Internet: challenges and a review of strategies’, Communication Systems Software and Middleware and Workshops, COMSWARE 2008. 3rd International Conference on 6-10 Jan., (IEEE CNF) page(s):536 – 543

[2] U. Jonsson, F. Alriksson, T. Larsson, P. Johansson, and G.M. Maquire, “MIPMANET: Mobile IP for Mobile Ad Hoc Networks,” Proceedings of IEEE/ACM Workshop on Mobile and Ad Hoc Networking and Computing, Boston, MA USA, pp. 75-80, August 1999.

[3] C. Perkins, E. Belding-Royer and S. Das, “Ad Hoc on Demand Distance Vector (AODV) Routing Protocol,” IETF Internet RFC 3561, July 2003.

[4] Hamidian, A., Körner, U. and Plymoth N. A. (2004) “Performance of Internet Access Solutions in Mobile Ad Hoc Networks”, Wireless Systems and Mobility in Next Generation Internet / Lecture Notes in Computer Science, First International Workshop of the EURO-NGI Network of Excellence, Dagstuhl Castle, Germany, June 2004, Vol. 3427, pp. 189-201.

[5] P. Ratanchandani, and R. Kravets, “A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks,” in Proceedings of the IEEE WCNC 2003, New Orleans, USA, vol. 3, pp. 1522-1527, March 2003.

[6] J. Lee, D. Kim, J. J. Garcia-Luna-Aceves, Y. Choi, J. Choi and S. Nam, “Hybrid gateway Advertisement Scheme for Connecting Mobile Ad Hoc Networks to the Internet,” Proceedings of the 57th IEEE VTC 2003, Korea, pp. 1991-195, April 2003.

[7] S. Bin, S. Bingxin, L. Bo, H. Zhonggong, and Z. Li, “Adaptive gateway Discovery Scheme for Connecting Mobile Ad Hoc Networks to the Internet,” Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 795-799, 2005.

[8] Kumar, R., Misra, M. and Sarje, A.K. (2007) “An Efficient Gateway Discovery in Ad Hoc Networks for Internet Connectivity”, Proc. of the International Conference on Computational Intelligence and Multimedia Applications, pp 275-281.

[9] Ns 2 Home page : http://www.isi.edu/nsnam/ns/index.html.

[10] E. Hyytia, H. Koskinen et al., “Random Waypoint Model in Wireless Networks,” Networks and Algorithms: Complexity in physics and Computer Science, Helsinki, June 16-19, 2005.