Implementation of Node Transition Probability based Routing Algorithm for MANET and Performance Analysis using Different Mobility Models

Sankararajan Radha and Sethu Shanmugavel

Abstract: **The central challenge in the design of ad-hoc networks is the development of dynamic routing protocol that efficiently finds route between mobile nodes. Several routing protocols such as DSR, AODV and DSDV have been proposed in the literature to facilitate communication in such dynamically changing network topology. In this paper, a Node Transition Probability (NTP) based routing algorithm, which determines stable routes using the received power from all other neighboring nodes is proposed. NTP based routing algorithm is designed and implemented using Global Mobile Simulator (GloMoSim), a scalable network simulator. The performance of this routing algorithm is studied for various mobility models and throughput, control overhead, average end-to-end delay, and percentage of packet dropped are compared with the existing routing protocols. This algorithm shows acceptable performance under all mobility conditions. The results show that this algorithm maximizes the bandwidth utilization during heavy traffic with lesser overhead.**

Index Terms: **NTPA, AODV, DSR, mobility models, GloMoSim, mobile ad hoc network.**

I. INTRODUCTION

Mobile Ad hoc Network [1] is an autonomous system of mobile nodes connected dynamically in an arbitrary manner by wireless links. The idea of ad hoc networking is sometimes also called infrastructureless networking, since the mobile nodes in the network dynamically establish routing among themselves to form their own network "on the fly." In other words, *mobile ad hoc networks are networks of mobile hosts with wireless interfaces that can dynamically form a network without the aid of any pre-existing infrastructure or centralized administration.* There is no static infrastructure such as base station shown in Fig. 1. All nodes of these networks behave not only as hosts but also as routers, forwarding packets to other mobile nodes in the network that may not be within direct wireless transmission range of each other. They take part in discovery and maintenance of routes to other nodes in the network.

In areas in which there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use, wireless mobile users may still be able to communicate through the formation of an ad hoc network. Ad hoc networks are characterized by multi-hop wireless connectivity, frequently changing network topology and the need for efficient adaptive routing protocols. Due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network. When two mobile hosts outside radio range from each other need to communicate, they can do so only via other such nodes which act as routers.

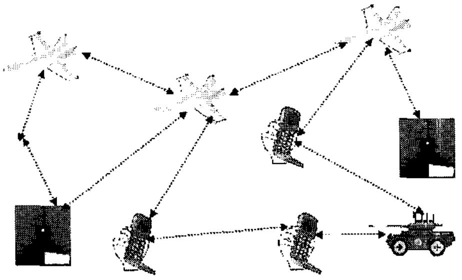


Fig. 1. Mobile ad-hoc network.

The ad hoc wireless network can be modeled as an undirected graph $G = f(V, E)$, where V is a set of N nodes and E is a set of L undirected links connecting nodes in V. Each node has a unique identifier and represents a mobile host with a wireless communication device with transmission range R and an unlimited storage space. Nodes may move around and change their speed and direction independently. An undirected link (i, j) connecting two nodes i and j always has a length less than or equal to R. The link (i, j) is removed from E when nodes i and j move apart and out of their transmission ranges.

Routing algorithms for existing networks have to be designed specifically to provide the kind of dynamic, self-starting and self-organizing behavior needed for ad hoc networks. By their very nature, mobile nodes wander around, changing their network location and link status on a regular basis. Furthermore, new nodes may unexpectedly join the network or existing nodes may leave or be turned off. Ad hoc routing algorithms must minimize the time required to converge after these topology changes. A low convergence time is more critical in ad hoc networks because temporary routing loops can result in packets being transmitted in circles, further consuming valuable bandwidth.

Another problem with wireless network interfaces is that they typically operate at significantly slower bit rates than their wire-based counterparts. Frequent flooding of packets throughout the network can consume significant portions of the available network bandwidth. Ad hoc routing protocols must minimize bandwidth overhead at the same time as they enable proper routing to take place. Also the routing algorithm should be scalable to

Manuscript received July 26, 2002; approved for publication by Sudhir Dixit, Division II Editor, April 18, 2003.

The authors are with the Telematics Laboratory, School of ECE, Anna University, Chennai - 25, email: radha_kumaran@yahoo.com, ssvel@annauniv.edu.

large networks. The overhead has to be minimized to decrease memory requirements of a node and channel bandwidth requirement.

Several routing algorithms [2]–[8] have been proposed to facilitate communication in such dynamically changing network topology and their comparison is shown in Table 1. In the existing protocols such as Dynamic Source Routing Protocol (DSR), Ad hoc On-demand Distance Vector Routing Protocol (AODV), Signal Stability based Adaptive Routing Protocol (SSA) and Route-Lifetime Assessment Based Routing Protocol (RABR), the control packets contribute to the network congestion during times of high load. To overcome this drawback, we have proposed a new algorithm, which minimizes control overhead during high traffic and it is scalable to large networks. The performance of these algorithms is studied under various scenarios.

In Section II, we describe the node transition probability based routing protocols for ad hoc networks. In Section III, we describe the simulation methodology and NTP algorithm implementation. Section IV describes the mobility models for ad hoc network and in section V, the simulation results of these algorithms are compared using GloMoSim.

II. NTP ALGORITHM

In this paper, we propose a new algorithm based on what is called Node Transition probability (NTP) [9], [10], which is computed using the received power at a particular node from all other nodes. Node Transition Probability algorithm (NTP) provides an efficient solution for wireless, mobile ad-hoc networks. We have compared the performance of NTP based routing protocol with existing on-demand routing protocols such as AODV and DSR. When the numbers of communication pairs are increased, a considerable amount of routing overhead will be generated. Simulation results show that the NTP algorithm is more preferable for mobile networks than other algorithms, when the node density, mobility and the traffic levels are high.

A. Description of NTP Algorithm

The basic idea behind NTP based routing is to assess the stability of neighbors by initiating beacons and computing the node transition probability matrix. The following are the steps followed in NTP algorithm to compute the neighbor table and route the packets.

Step 1:
The first sender initiates the first beacon and the receiving neighbors re-initiate the beacons. The source node records the received power level of the beacons in the power table. This is repeated 'n' number of times. The power table has the dimension of N X n, where N is the number of nodes and n is the number of flooding. The power table Sk for the kth node is

$$S_k = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots & p_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{Nj} & \cdots & p_{Nn} \end{bmatrix}_{N\times n}, \qquad (1)$$

where P_{ij} is the power with which node i replies to node k during the jth flooding and $P_{kj} = 0$.

Step 2:
After waiting for a finite interval of time $Tw = 2 * n * t_n$, the elements of the power matrix are arranged in the descending orders of power as $P_{1j} > P_{2j} > \cdots > P_{(N-1)j}$ and $1 \geq j \geq n$. Here n is the number of flooding and t_n is the NODE_TRAVERSAL_TIME and is defined as the worst case time required for the packet to reach the node at the boundary.

Step 3:
The index matrix X_k for kth node is formed as

$$X_k = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ X_{M1} & X_{M2} & \cdots & X_{Mj} & \cdots & X_{Mn} \end{bmatrix}_{M\times n}, \qquad (2)$$

where X_{ij} refers to the id of the node replying with ith power level in jth flooding and is an element of the set $R = 1, 2, 3, \cdots, k, \cdots, N$. If more than one node replies with the same power level, then the node with smaller id is assigned the higher power level and next node id is assigned in the next power level and so on without any loss of generality. M refers to the number of power levels.

Step 4:
The frequency matrix Y_k is formed as

$$Y_k = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1j} & \cdots & n_{1N} \\ n_{21} & n_{22} & \cdots & n_{2j} & \cdots & n_{2N} \\ \vdots & \vdots & & \vdots & & \vdots \\ n_{i1} & n_{i2} & \cdots & n_{ij} & \cdots & n_{iN} \\ \vdots & \vdots & & \vdots & & \vdots \\ n_{M1} & n_{M2} & \cdots & n_{Mj} & \cdots & n_{MN} \end{bmatrix}_{M\times N}, \qquad (3)$$

where n_{ij} refers to the number of times the node j has replied to node k with power level i. The power value received is continuous and usually small value that is expressed in an exponent form. Since the power values are very small (Order of 1×10^{-8}), it will be difficult to differentiate the weight computed for two nodes with two different power values. Hence, we consider power level instead of power values, without any loss of generality. The continuous power can take any value between P_{max} (Transmitting Power) and P_{min} (Threshold Power). This power range is divided into M power zones and each zone is assigned an integer to be used in calculation of the weight as shown in Table 1 for $M = 10$.

Step 5:
The Probability matrix is formed after multiplying the frequency matrix by a weight matrix $W = [w_1, w_2, w_3, \cdots, w_M]$ as

$$P_k = \frac{[W][Y_k]}{\left(\sum_{i=1}^{M} w_i\right) n} = (p_1\ p_2\ p_3\ \cdots\ p_j\ \cdots\ p_N), \qquad (4)$$

where w_i is the ith weighted value and n is the number of flooding. In order to give highest probability for the node replying with highest power level, a weight matrix with weights

Table 1. Power level assignment.

Power level	Power received in Watt
1	>= 3.07645e-07
2	(7.69113e-08, 3.07645e-07)
3	(3.41828e-08, 7.69113e-08)
4	(1.92278e-08, 3.41828e-08)
5	(1.23058e-08, 1.92278e-08)
6	(8.5457e-09, 1.23058e-08)
7	(6.27847e-09, 8.5457e-09)
8	(4.80695e-09, 6.27847e-09)
9	(3.79809e-09, 4.80695e-09)
10	< 3.07645e-09

$w_1 > w_2 > w_3 > \cdots > w_M$ is introduced. In our implementation, we have chosen $w_i = 2(M - i)$, where $i = 1, 2, \cdots, M$. The higher the value of p_i, the larger is the probability that the node i is nearer to node k. In other words, p_i indicates the nearness probability of the each node with respect to the kth node, whereas $q_i = 1 - p_i$ indicates the transition probability of node i. The NTP matrix Q_k is then formed as

$$Q_k = (q_1\, q_2\, q_3\, \cdots\, q_j\, \cdots\, q_N), \tag{5}$$

where $q_j = 1 - p_j$ is the probability of jth node being the next hop node. After computing the NTP matrix Q, each node deletes the matrix S_k, X_k, Y_k and P and preserves only matrix Q_k.

Step 6:

The values q_j in the Q_k matrix indicates the transition probability of the jth node with respect to the kth node. The Q_k matrix is sorted in the ascending order of q_j values and the corresponding index is stored as neighbor table N_k. The first entry in the table denotes the best neighbor of that node and second entry denotes the second best neighbor and so on.

Step 7:

If a node has packets to send, it checks whether the destination is present in the Neighbor Table. If it is so, then it starts transmitting the packets. Otherwise, a search packet is sent to the best neighbor of the node.

Step 8:

If the destination address is found in the neighbor table of the node receiving the search packet, then searching process is terminated. Otherwise, the packet is forwarded to the best neighbor of that intermediate node. This node is entered as the next hop for the particular destination in the Route Table (R_k) and the time at which this hop was found is also recorded. Search packet is not forwarded to the node from which it came. If a node encounters a search packet, which it has already processed, the packet is forwarded to the next best neighbor and the information is entered in the Route Table. If TOE is the time of entry in the route table during the route discovery process, then the source starts sending the data at time $T = TOE + 2t_n$, where t_n is the NODE_TRAVERSAL_TIME.

Step 9:

If there is a link breakage indicated by the MAC layer and the data packet is dropped, then the entries in the route and neighbor table are deleted and the new neighbor table is computed by initiating beacons, and the process is repeated all over again from step 1 through 8.

An example with simulated values of Power matrix S_k for exponential mobility models for node $k = 1$ is given as (6) shown at the bottom of this page.

The index matrix X_k is calculated from the power matrix as,

$$X_k = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 8 & 8 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 7 & 5 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 5 & 7 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 2 & 2 \\ 4 & 4 & 4 & 4 & 9 & 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{7}$$

and the frequency matrix Y_k is derived from X_k as

$$Y_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 8 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 4 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8}$$

where the number in the matrix defines as node j has replied to node k with power level i. The NTP matrix Q for a node 1 is calculated from the frequency matrix as in step 5 of NTP algorithm using the weight matrix $W = [18, 16, 14, 12, 10, 8, 6, 4, 20]$ as

---

$$S_k = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 7.2928e\text{-}9 & 6.3294e\text{-}9 & 5.3675e\text{-}9 & 5.4237e\text{-}9 & 5.4443e\text{-}9 & 7.4489e\text{-}9 & 4.4521e\text{-}9 & 1.4948e\text{-}8 & 1.5058e\text{-}9 & 1.5104e\text{-}9 \\ 7.6911E\text{-}8 & 7.0577E\text{-}8 & 6.9771E\text{-}8 & 6.9554E\text{-}8 & 5.8232E\text{-}8 & 5.5434E\text{-}8 & 4.3953E\text{-}8 & 4.2756E\text{-}8 & 4.1848E\text{-}8 & 3.1411E\text{-}7 \\ 5.2046E\text{-}9 & 5.1766E\text{-}9 & 6.1736E\text{-}9 & 5.1619E\text{-}9 & 4.1501E\text{-}9 & 4.1267E\text{-}9 & 4.1106E\text{-}9 & 3.9035E\text{-}9 & 3.8864E\text{-}9 & 4.0471E\text{-}9 \\ 3.2656E\text{-}8 & 2.5754E\text{-}8 & 2.6193E\text{-}8 & 3.0200E\text{-}8 & 2.7278E\text{-}8 & 3.1294E\text{-}8 & 2.7454E\text{-}8 & 2.4590E\text{-}8 & 1.7593E\text{-}8 & 2.7706E\text{-}8 \\ 6.2279E\text{-}9 & 7.2417E\text{-}9 & 6.2663E\text{-}9 & 7.3104E\text{-}9 & 7.3427E\text{-}9 & 7.3825E\text{-}8 & 6.4222E\text{-}9 & 8.1748E\text{-}9 & 1.5661E\text{-}9 & 1.6386E\text{-}9 \\ 3.1377E\text{-}8 & 4.1056E\text{-}8 & 5.0533E\text{-}8 & 6.0667E\text{-}8 & 4.0374E\text{-}8 & 6.9787E\text{-}8 & 4.8998E\text{-}8 & 6.8720E\text{-}8 & 2.7822E\text{-}8 & 1.7292E\text{-}8 \\ 1.4199E\text{-}8 & 1.4406E\text{-}8 & 1.5002E\text{-}8 & 1.5300E\text{-}8 & 1.5493E\text{-}8 & 1.5861E\text{-}8 & 1.6362E\text{-}8 & 1.7176E\text{-}8 & 5.7874E\text{-}8 & 5.8398E\text{-}8 \\ 4.0844E\text{-}9 & 4.1124E\text{-}9 & 3.9305E\text{-}9 & 4.1525E\text{-}9 & 6.1916E\text{-}9 & 6.2188E\text{-}9 & 5.2275E\text{-}9 & 5.2783E\text{-}9 & 5.3128E\text{-}9 & 5.3432E\text{-}9 \\ 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 & 3.0764E\text{-}7 \end{bmatrix} \tag{6}$$

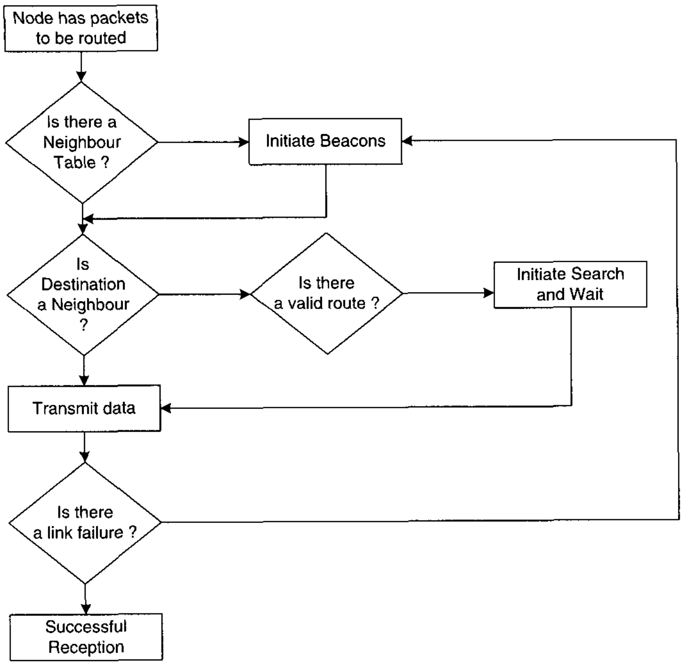


Fig. 2. Flowchart representation of NTP algorithm.

$$P_k = [0.0\ 0.0872\ 0.1636\ 0.04363\ 0.1254\ 0.07630.1400\ 0.1163\ 0.0472\ 0.1818], \tag{9}$$

$$Q_k = [1\ 0.9128\ 0.8364\ 0.9564\ 0.8746\ 0.9246\ 0.86\ 0.8837\ 0.9528\ 0.8182]. \tag{10}$$

The sequence of steps followed by the algorithm to route the packets successfully to the destination is shown in Fig. 2 as a flow chart and the pseudo code of NTPA is given in Fig. 3.

After discussing the proposed on-demand protocol, we have compared the complexity and characteristics of the algorithm with an existing algorithm [8] and are given in Table 2 and Table 3.

III. SIMULATION ENVIRONMENTS AND METHODOLOGY

The routing protocols are simulated within the GloMoSim library [11]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [12], [13]. We simulated a network of mobile nodes placed randomly within a 500 × 500 meter area. Radio propagation range of 250 meters and channel capacity of 2 Mb/s were chosen for each node. There were no network partitions throughout the simulation. Even though there exists network partitions, if the node in the network is moved to other partitions and that being a next hop to any route, then the upstream node knows that there is a link break due to movement of the node. Hence, it computes new route based on neighbor table by erasing the old route table. The node, which joins the new partition, is idle till it receives some request packet from other nodes. Once, it receives the request, it starts computing its neighbor table and then participates in routing the packet. Each simulation is executed for 600 seconds. Multiple runs with different seed values were conducted for each scenario and the collected data was averaged over those runs. Table 4 and 5 list the simulation parameters and environments, which are used as default values unless otherwise specified.

Table 4. Simulation parameter.

Simulation time	10 Min
Bandwidth	2 Mbps
Frequency of operation	2.4 GHz
Source destinaion pair	40
Simulation area	500m X 500 m
Number of flooding	1
Number of nodes	25
Speed	Variable
Offered traffic	4-12 packets/sec
Radio range	250 meters
Received power threshold	-81 dBm
Transmitted power	7.89 dBm
Number of packets generated	10000 packets
Application	CBR, Telnet, FTP
Transport	UDP, TCP
Network	NTP, AODV, DSR
MAC	802.11

Table 5. Simulation environments.

Processor	450 MHz, PIII
Hard disk	10 GB
RAM	128 SDRAM
Operating system	Windows 2000

A free space propagation model was used in our experiments. In this model, signal power attenuates as $1/d^2$ where d is the distance between radios. In addition to the free space channel model, we have also implemented the SIRCIM (Simulation of Indoor Radio Channel Impulse response Models) [14] which considers fading, barriers, foliages, multipath interference, etc. The SIRCIM is more accurate than the free space model, but we have decided against using SIRCIM in our study because: (a) the complexity of the SIRCIM increases simulation time by two orders of magnitude; (b) the accuracy of the channel model does not affect the relative ranking of the routing protocols evaluated in this study; and (c) SIRCIM must be "tuned" to the characteristics of the physical environment (e.g., indoor, outdoor etc.), thus requiring a much more specific scenario than we are assuming in our experiments. In the radio model, capture effects are taken into account. If the capture ratio (the minimum ratio of an arriving packet's signal strength relative to those of other colliding packets) [15] is greater than the predefined threshold value, the arriving packet is received while other interfering packets are dropped. A traffic generator was developed to simulate constant

```
Proc Node(i) ≡
  NodeInit(i);
    if Pkt.Type == data!!pkt received
        foreach Pkt ∈ PktQueue do
          if(Pkt.source != node)
            HandleData(i,Pkt)
          else
            bufer ← buffer U {Pkt};
            if(!Nbrtable(i))
                    for j < FLOODS
                    InitiateBeacon(i);
                    End for
            else
                    if(Packet.Dest ∈ $N_i$)
                    Transmit Packet
                    else
                        if(Packet.Dest ∈ $R_i$)
                        Transmit Packet
                        else
                        InitiateSearch(i,dest)
                        fi
                    fi
            fi
          fi
        od
    else
    Pktprocess(i,Pkt)
    fi

Proc Handledata(i) ≡
  if (Pkt.destination==node)
    Pktreceived!!
  else
    if(!Dataseen(Pkt))
      if(destination ∈ $N_i$ !! destination ∈ $R_i$)
        Transmit data;
      fi
    fi
  fi

Proc NodeInit(i)
  foreach j ∈ V do
    $N_i$(j)= -1;
    $Pwr_i$(j)=0;
    $R_i$(j)=0;
  od

Proc Dataseen(Pkt) ≡
  foreach i < Dataseen.size do
    if(entry[i].id == Pkt.id && entry[i].source == Pkt.source)
        return TRUE;
    else
        return FALSE;
    fi
  od

Proc NbrTable(i,Pkt) ≡
  if($N_i$ ← size > 0)
    return TRUE;
  else
    return FALSE;
  fi

Proc Pktprocess(i,Pkt)
  switch(Ptk.type)
    case BEACON:
        pwr ← pwr U {beacon.pwr}
        if(Pkt.destination != node)
          for j < FLOODS
            initiate beacon(i);
          End for
        fi
        break;
    case SEARCH:
        if(Pkt.dest ∈ $R_i$ && $R_i$(j).Valid == False)
          $R_i$(j).Nexthop = Nbr($R_i$(j).ptr + 1)
          $R_i$(j).ptr = $R_i$(j).ptr+1
          ForwardSearchPkt(i,dest)
        fi
        break;
  end of switch
  if(timer > NODE_TRAVERSAL_TIME)
    Sort(PwrTable(i));
    EvaludateNbr(PwrTable(i));
  fi
```

Fig. 3. Protocol specification of NTP algorithm.

Table 2. Ad hoc routing method.

Method	Distance vector	On-demand	Location stability	Signal strength	Routing overhead	Pkt processing overhead	Route reply process
DSR		✓			Low	High	✓
RABR		✓			Low	Low	✓
SSA		✓	✓ (Portion of the topology)	✓	Low	Low	✓
AODV		✓			High	Low	✓
NTP		✓	✓ **(Node stability)**	✓	**Low**	**Low**	

bit rate sources. Source nodes and destination nodes were chosen randomly with uniform probabilities. A packet is dropped when no acknowledgment is received after retransmitting it a certain number of times. The IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) is used as the MAC layer in our experiments. Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) with acknowledgments is chosen as the access scheme. Optionally, the nodes can make use of

Table 3. Comparisons of the characteristics of source-initiated on-demand ad-hoc routing protocols.

Performance parameters	AODV	DSR	RABR	SSA	NTP
Time complexity (initialization)	O(2d)	O(2d)	-	O(d+z)	O(2d)
Time complexity (postfailure)	O(2d)	O(2d)	-	O(l+z)	O(2d)
Communication complexity (initialization)	O(2N)	O(2N)	-	O(N+y)	O(N)
Communication complexity (postfailure)	O(2N)	O(2N)	-	O(x+y)	O(N)
Routing philosophy	Flat	Flat	Flat	Flat	Flat
Loop free	Yes	Yes	Yes	Yes	Yes
Multicast Capability	Yes	No	Yes	No	Yes
Beaconing requirements	No	No	Yes	Yes	No
Multiple route possibilities	No	Yes	No	No	No
Routes maintained in	Route table	Route cache	Route table	Route table	Route table
Utilizes route cache/table expiration timers	Yes	No	No	No	No
Route reconfiguration methodology	Erase route; notify source	Erase route; notify source	Erase route; notify source	Erase route; notify source	Erase route, define new route based on neighbor table information
Routing metric	Freshest & shortest	Shortest path	Route lifetime based on link affinity	Associativity & stability	Signal strength, stability & shortest path

Abbreviations:
N = Number of nodes in the network, d = Network diameter,
x = Number of nodes affected by a topological change, l = Diameter of the affected network segment
y = Total number of nodes forming the directed path where the REPLY packet transits
z = Diameter of the directed path where the REPLY packet transits

RTS/CTS channel reservation control frames for unicast, virtual carrier sense and fragmentation of packets larger than a given threshold. In our experiment, we employed RTS/CTS and virtual carrier sense. We choose this configuration to minimize the frequency and deleterious effects of collisions over the wireless medium.

A traffic generator was developed to simulate CBR sources. The size of data payload is 512 bytes. Data sessions with randomly selected sources and destinations were simulated. Each source transmits data at a rate of 4-12 pkts/sec. We vary the traffic load by changing the number of data sessions and examine its effect on routing protocols. In this paper, we have used different mobility models [16] for performance study.

A. Route Discovery

In the NTP routing algorithm implementation, for each node i, one list and four tables are maintained. They are: Buffer, Neighbor table Nk, data seen table, power table and route table.

An example set of simulated values of Neighbor Table, route table and propagation of search packet for exponential mobility models for the above simulation parameters are given in Tables 6, 7, and 8. Neighbor Table for 10 nodes of 25 nodes scenario is shown in Table 6. In this table, the column indicates node id and row indicates the position of the best neighbor. For example, the first entry in node 0 of N_k indicates the node 24 is the best neighbor and node 7 is the second neighbor and so on. The size of N_k for each node depends on the transmission range. Table 7 shows an example of the route table R_k computed using the table 6 and maintained at each node.

Table 8 shows the propagation of the search packet for the source node 18 and the destination node 5. The search packet, which is indicated by an arrow in table 8, selects the next hop as the best neighbor to reach the destination. If destination is present in the N_k then the search packet is terminated and R_k is computed. For example, Node 7 has a destination address in its N_k and hence search packet is terminated. Therefore, the route from source node 18 to destination node 5 is 18 → 13 → 7 → 5.

B. Route Maintenance

Route maintenance involves handling link failures between the mobile nodes. Whenever link break occurs in the network, appropriate nodes will be informed. Then, the MAC layer will

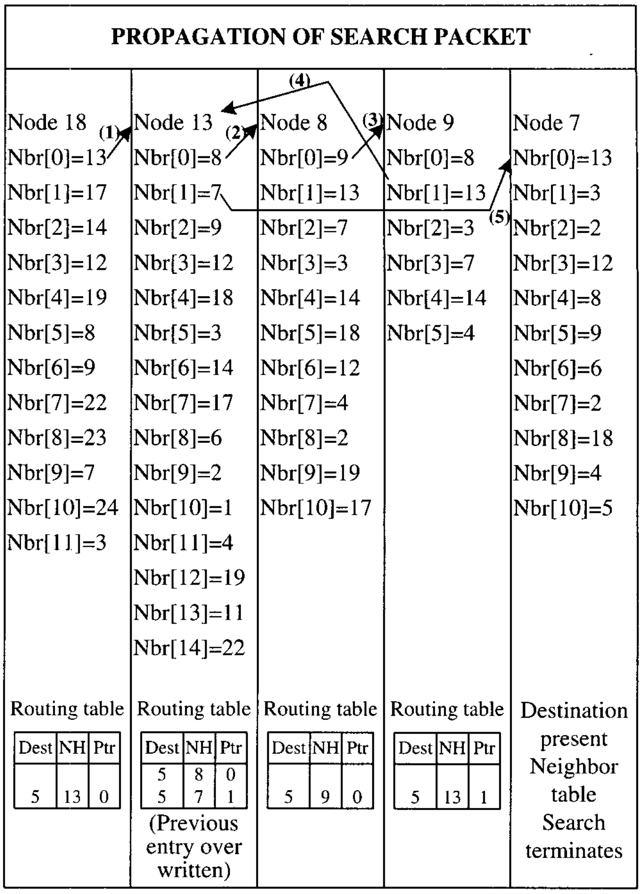
Table 6. Neighbor table.

Best nbr \ Node ID											
nbr [ ]	0	1	2	3	4	5	6	7	8	9	10
0	24	10	15	22	7	12	5	13	9	8	4
1	7	21	0	13	1	7	1	3	13	13	22
2	5	16	6	11	0	0	6	2	7	3	13
3	8	12	8	18	6	18	0	12	3	7	9
4	6	9	12	12	8	13	8	8	14	14	
5	15	20	7	17	18	17	18	9	18	4	
6	3	17	19	9	3	3	12	6	12		
7	19	3	17	20	19	20	3	1	4		
8	14	23	20		24	14	19	18	2		
9			2		14	1	17	4	19		
10			14			5	24	5	17		
11			23			24	14				

Table 7. Route table.

ROUTE TABLE					
Node 0		Node 1		Node 2	
Destination address	Next hop	Destination address	Next hop	Destination address	Next hop
21	24	22	10	8	15

Table 8. Propagation of search packet.



PROPAGATION OF SEARCH PACKET				
Node 18	Node 13	Node 8	Node 9	Node 7
Nbr[0]=13	Nbr[0]=8	Nbr[0]=9	Nbr[0]=8	Nbr[0]=13
Nbr[1]=17	Nbr[1]=7	Nbr[1]=13	Nbr[1]=13	Nbr[1]=3
Nbr[2]=14	Nbr[2]=9	Nbr[2]=7	Nbr[2]=3	Nbr[2]=2
Nbr[3]=12	Nbr[3]=12	Nbr[3]=3	Nbr[3]=7	Nbr[3]=12
Nbr[4]=19	Nbr[4]=18	Nbr[4]=14	Nbr[4]=14	Nbr[4]=8
Nbr[5]=8	Nbr[5]=3	Nbr[5]=18	Nbr[5]=4	Nbr[5]=9
Nbr[6]=9	Nbr[6]=14	Nbr[6]=12		Nbr[6]=6
Nbr[7]=22	Nbr[7]=17	Nbr[7]=4		Nbr[7]=2
Nbr[8]=23	Nbr[8]=6	Nbr[8]=2		Nbr[8]=18
Nbr[9]=7	Nbr[9]=2	Nbr[9]=19		Nbr[9]=4
Nbr[10]=24	Nbr[10]=1	Nbr[10]=17		Nbr[10]=5
Nbr[11]=3	Nbr[11]=4			
	Nbr[12]=19			
	Nbr[13]=11			
	Nbr[14]=22			
Routing table Dest NH Ptr 5 13 0	Routing table Dest NH Ptr 5 8 0 5 7 1 (Previous entry over written)	Routing table Dest NH Ptr 5 9 0	Routing table Dest NH Ptr 5 13 1	Destination present Neighbor table Search terminates

send *MSG_NETWORK_PacketDropped* to network layer. During link break, we choose the new route based on 2nd best neighbor using existing neighbor table without flooding the beacons once again. Due to some implementation problem, in this paper, we have computed new route by repeating step 1 through 8, like AODV, which finds new route once link break is detected. Hence, the computation complexity of the algorithm is $O(N^2)$.

IV. MOBILITY MODELS

A mobility model should attempt to mimic the movements of real mobile nodes. Changes in speed and direction must occur in reasonable time slots. For example, we would not want mobile nodes to travel in straight lines at constant speeds throughout the course of the entire simulation because real mobile nodes will not travel in such a restricted manner. Some of the mobility models for ad hoc networks proposed in the literature [16] and the exponentially distributed random mobility are used for performance study of NTP algorithm.

V. PERFORMANCE EVALUATION

In this section, we analyze the simulation results of NTP algorithm with existing algorithms. The following metrics were used in computing the protocol performance. The metrics were derived from one suggested by the MANET working group for routing protocol evaluation [17], [18].

- **Packet delivery ratio**: Measured as the ratio of the no. of data packets delivered to the destination and the no. of data packets sent by the sender.
- **End-to-end delay**: Measured in ms as the time between the reception of the last and first packet / total no. of packets reaching the application layer. This delay includes processing and queuing delays in each intermediate node.
- **Control overhead**: Measured as the ratio of no. of control packets transmitted during the simulation period by data packet transmitted.
- **Packet dropped**: Data Packets may be dropped en route for two reasons: The next hop link is broken when the data packets is ready to be transmitted or no routing table (cache) entry exists for the intended destination.

The performance results of various algorithms with respect to mobile speed obtained using GloMoSim for different mobility models are presented. The NTP algorithm requires lesser control overhead compared to AODV as shown in Fig. 4. This is due to the fact that only the stable routes are used by the algorithm for routing the packets. Flooding of RREQ and the search for new route contributes to the increased overhead in AODV. On the other hand, the control overhead is less with an increase in packet size in DSR. Both NTP and AODV use equal sized packets.

A slight decrease in throughput is observed in the case of NTP for random walk model when compared to AODV and DSR as

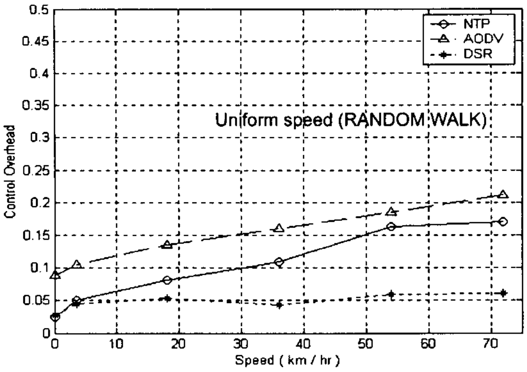


Fig. 4. Control O/H vs. mobility.

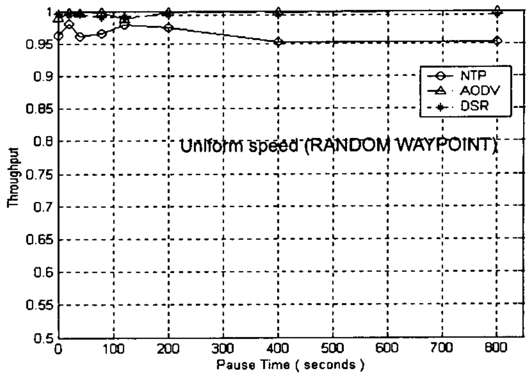


Fig. 7. Throughput vs. pause time.

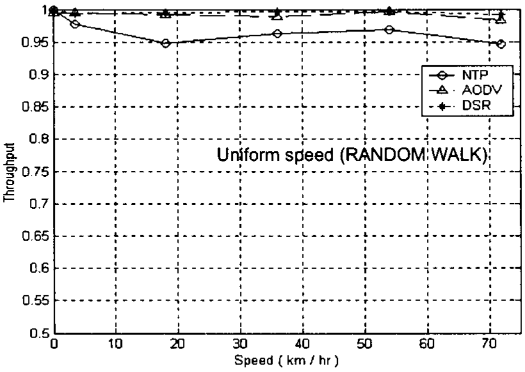


Fig. 5. Throughput vs. mobility.

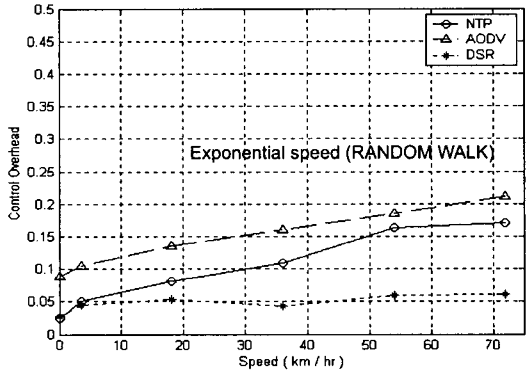


Fig. 8. Control O/H vs. mobility.

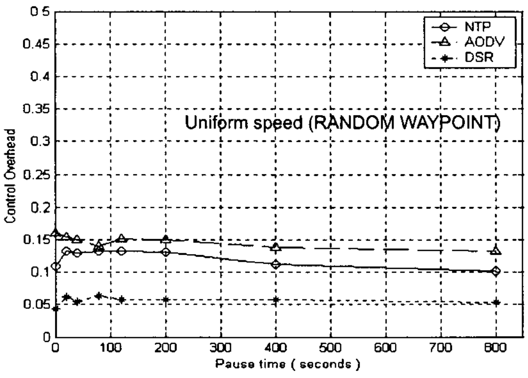


Fig. 6. Control O/H vs. pause time.

shown in Fig. 5. The RREP packet used by AODV helps to maintain the performance. This is an area where there is scope for improvement in NTP algorithm. During high mobility of the participating nodes, all the algorithms show a small degradation in throughput.

Longer is the duration of pause time lower is the mobility. In Fig. 6, the control overhead is found to stabilize for longer pause times for all the three algorithms. NTP algorithm is found to perform in between AODV and DSR in terms of control overhead.

The throughput of all the three algorithms for Random waypoint mobility with uniformly distributed speed is shown in Fig. 7. The performance is found to be better when compared to Random Walk mobility model, which is characterized by abrupt transitions in direction and speed of the mobile nodes. Increasing pause times results in smoother transitions and therefore, the throughput and control overhead remains constant.

In reality, one deserves that lower mobile speeds occur with higher probability and vice versa. Therefore, we have studied the effect of these mobility models with exponential distributed speed. The control overhead and throughput performance is shown for all the three algorithms in Figs. 8 and 9. The Control Overhead for NTP lies between that of AODV and DSR. Random Waypoint with Exponential speed is the most realistic mobility model. The control overhead is found to stabilize for longer duration of pause times for all the three algorithms.

The throughput performance for AODV and DSR is found to improve with exponentially distributed speed as shown in Fig. 10. Fig. 11 shows the amount of control overhead required to obtain the above throughput. It is almost 100% due to the lower mobility of the nodes and therefore, fewer link breakages. As the traffic in the network increases, where the number of transmitting nodes increases, NTP uses lesser control overhead to deliver the packets because of the use of the more stable route. AODV uses more number of control packets since it floods the RREQ packet for every Source-Destination (S-D) pair which is

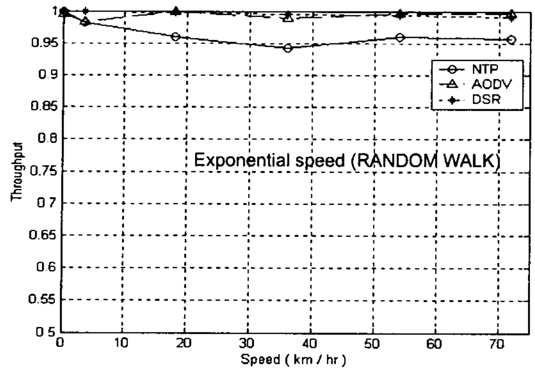


Fig. 9. Throughput vs. mobility.

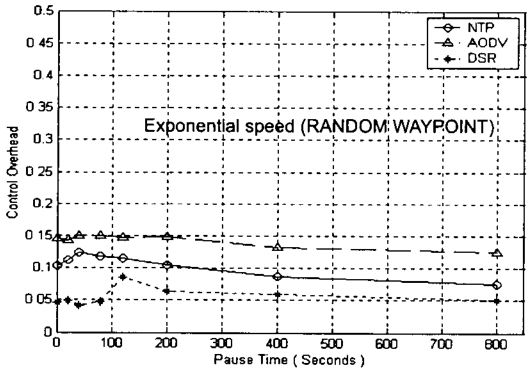


Fig. 10. Control O/H vs. pause time.

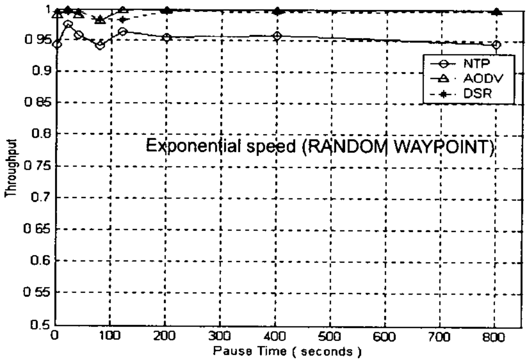


Fig. 11. Throughput vs. pause time.

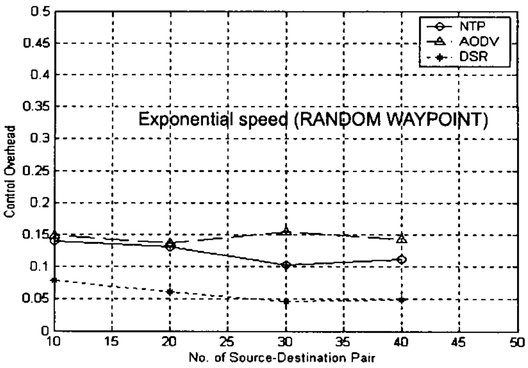


Fig. 12. Control overhead vs. no. of S-D pair.

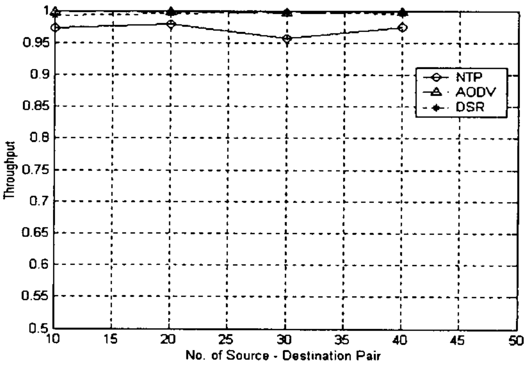


Fig. 13. Throughput vs. no. of S-D pair.

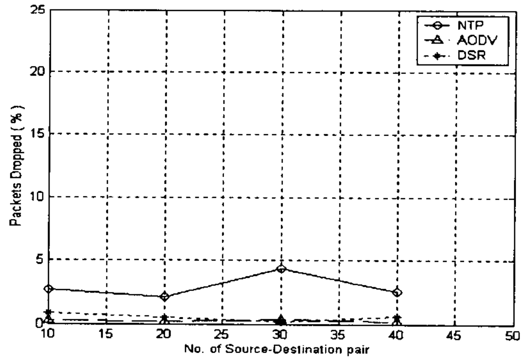


Fig. 14. Percentage dropped vs. no. of S-D pair.

shown in Fig. 12.

The throughput performance of all the three algorithms is shown in Fig. 13 for different traffic conditions. As the traffic increases, the throughput of AODV decreases due to heavy congestion created by the excessive control and data packets. Fig. 14 shows the performance of percentage of packets dropped as the S-D pair increases. DSR and AODV have fewer packets dropped for smaller network whereas NTP based routing protocol has additional drop of 2% to 3% packets and it approaches the same percentage of packets dropped as the traffic increases. For higher traffic under large network, NTP based protocol delivers same throughput as other protocols with lesser control packets, which is discussed later in this paper.

The average end-to-end delay performance of the three protocols is shown in Fig. 15. In NTP, this time is a little more than AODV and DSR. By sending a search reply packet, the end-to-end delay can be decreased as the reply packet determines the route through which the data packets should be transferred. This also helps a more reliable transfer of data.

When the number of nodes in the network increases and the S-D pair increases, the NTP algorithm performs well compared to AODV and almost same performance as DSR, which is shown

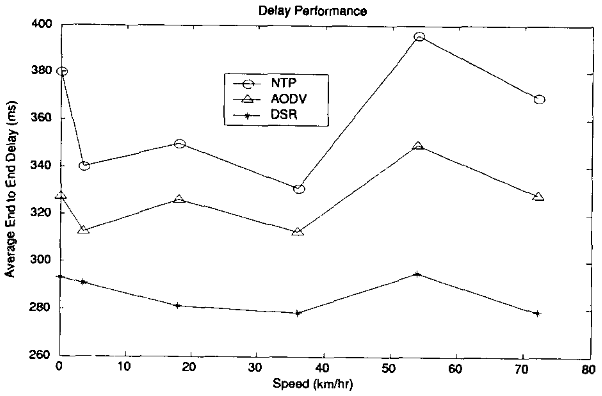


Fig. 15. End-to-end delay performance.

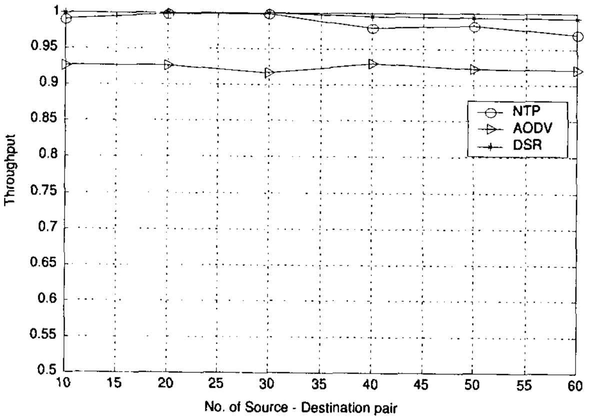


Fig. 16. Throughput vs. S-D pair.

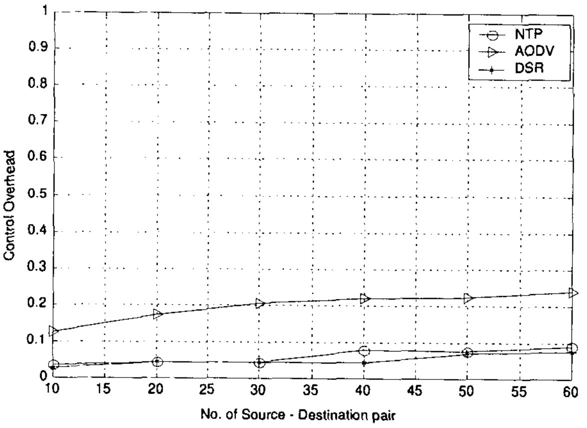


Fig. 17. Control O/H vs. S-D pair.

in Fig. 16. The control overhead of NTP algorithm for 50 nodes with mobile speed of 10 m/s is shown in Fig. 17, which indicates that it produces more or less same overhead as DSR. As the number of S-D pair increases, there is a slight increase in control overhead due to link breaks. The control overhead for AODV is much higher than that of other two algorithms because of separate RREQ and REPLY packets for each S-D pair.

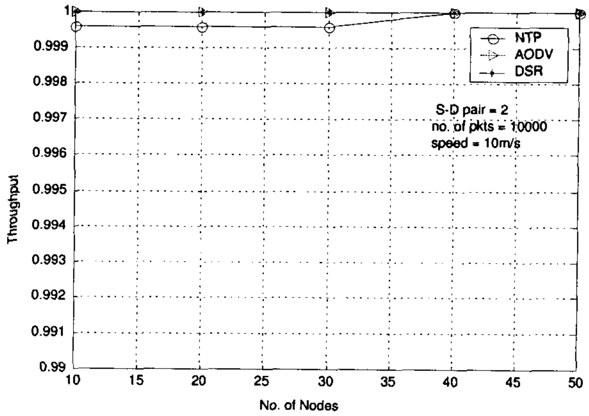


Fig. 18. Throughput vs. no. of nodes.

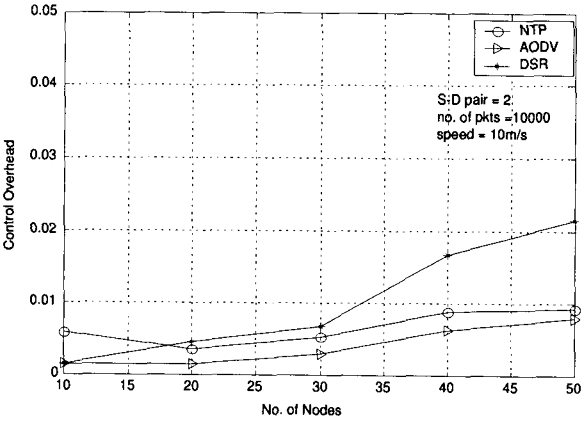


Fig. 19. Control O/H vs. no. of nodes.

The throughput performance of all three algorithms for 50 nodes network size is shown in Fig. 18. As the number of S-D pair increases, both DSR and NTP algorithms perform well, and the AODV algorithm shows that there is a slight degradation in the throughput performance because of heavy traffic. The NTP algorithm has less control overhead and produces 98% of throughput for high traffic networks compared to smaller size network whose performance is discussed with the worst case throughput of 95%. In spite of good throughput, the packet size of DSR increases due to the source route as the network size increases, which prevent DSR being adopted as the standard routing protocol for larger networks. DSR is the most efficient routing protocol when the network size is small which is shown in Figs. 18 and 19. Therefore, NTP algorithm is more suitable for large networks with heavy traffic.

This simulation compares the route valid time for various routing protocols (NTP, DSR, AODV) for different values of pause time with maximum speed of node as 5 m/s. *Route valid time* is the time for which a determined route is valid before a link break occurs due to the mobility of the node.

The graph is obtained by plotting values, which are obtained by averaging the results of 10 different mobility scenarios for the same value of pause times. Fig. 20 shows that the route valid time for NTP is better than that of other routing protocols.

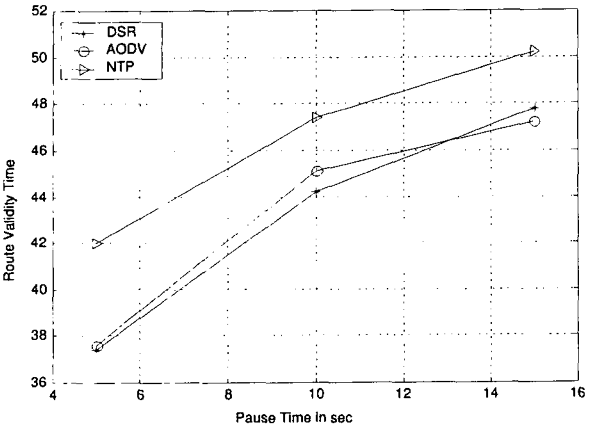


Fig. 20. Route validity time vs. pause time.

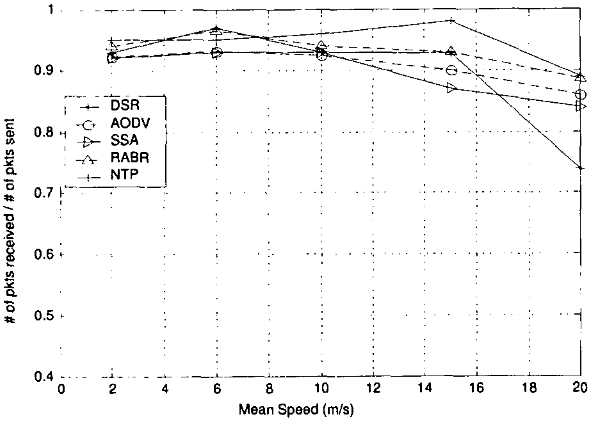


Fig. 21. Number of packets received / number of packets trans. vs. mean speed; pause time=0 sec.

This is due to the fact that the other routing protocols determine the shortest path to the destination to compute the route whereas NTP determines the most reliable path to the destination based on the probability values. Hence the occurrence of a link failure is less in the NTP routing protocol, which implies that the route validity time or stability of the protocol would be longer.

Figs. 21 and 22 compare the performance of the throughput of NTPR for the simulation parameter of 25 hosts moving in a rectangular topology of 1500m x 300m with the existing on-demand protocols, which uses node stability as the criterion. As is expected, the performance of all the protocols degrades with increasing mean speeds and decreasing pause times. *It is observed that the proposed NTP based routing protocol outperforms other protocols at large mean speeds.* Although DSR performs the best at low speeds, the performance of NTP based Routing is comparable to other protocols.

Based on stability of the nodes, the NTP protocol adapts well to increasing network mobility. Other protocols, in which the route enclosed in the first *route request* packet is selected for data transmission, have to face substantial throughput degradation. Even the SSA protocol, in which a route is established over strongly connected channels, does not perform as well as NTP based routing. NTPR and RABR choose the longest lasting route for CBR agents, thus ensuring lesser route failures as compared to other protocols. This feature is even more enhanced at high speeds. Hence NTPR performs well at high speeds, which is shown in Figs. 21 and 22. As compared to RABR protocol, NTPR uses the same concept of route selection based on long-lived routes but it does not depend on the number route requests reached the destination. Also, it defers from RABR, as NTPR does not use route reply packets to the source nodes. Hence, the amount of control overhead of NTPR is naturally less as compared to that of RABR. DSR lacks route cache maintenance strategies to purge the stale routes off the cache. Hence the protocol faces the problem of stale routes and its performance degrades at high speeds when the frequent route failures cause the route cache entries to go stale. In the case of AODV, the concept of sequence numbers and route cache timers alleviates the problem of stale routes. Hence the performance is relatively more consistent in this case.

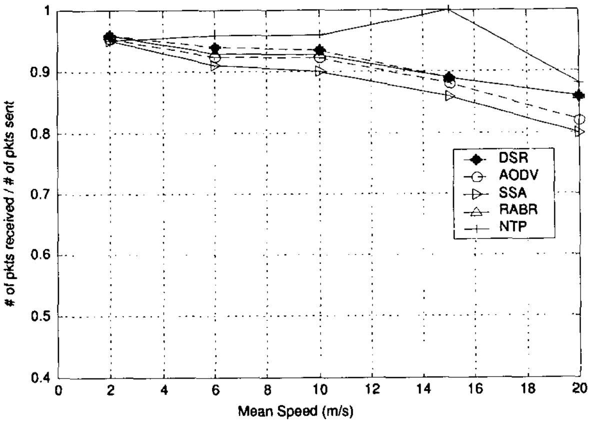


Fig. 22. Number of packets received / number of packets trans. vs. mean speed; pause time=10 sec.

Finally, in this paper we have studied the performance of NTP based routing for scalable networks. The performance of all three algorithms for number of nodes is plotted in Figs. 23 and 24. When the number of nodes in the network increases, the throughput performance for the network also slightly increases due to the availability of number of routes to the destination for the same parameter. Fig. 24 indicates the performance of control overhead increases as the number of nodes in the network increases due to increase in number of route requests and route replies flooded in the network. Among all three algorithms, NTP based routing generates 30% lesser number of control packets when the network size is 250. Also, if we increase the traffic in the same network, the amount of control overhead for NTP based routing will not be increased as much as DSR and AODV. Since it uses the same neighbor table for the route computation of new S-D pair, NTP based routing protocol maximizes the bandwidth utilization for large-scale network and heavy traffic with lesser control overhead.

VI. CONCLUSION

The new Routing algorithm based on Node Transition Probability is found to perform acceptably with considerable reduc-

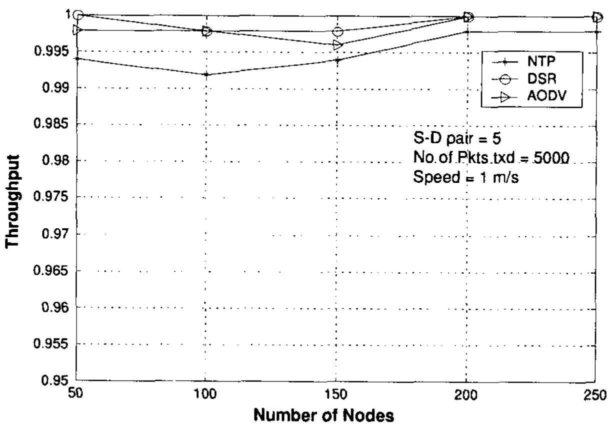


Fig. 23. Throughput vs. number of nodes.

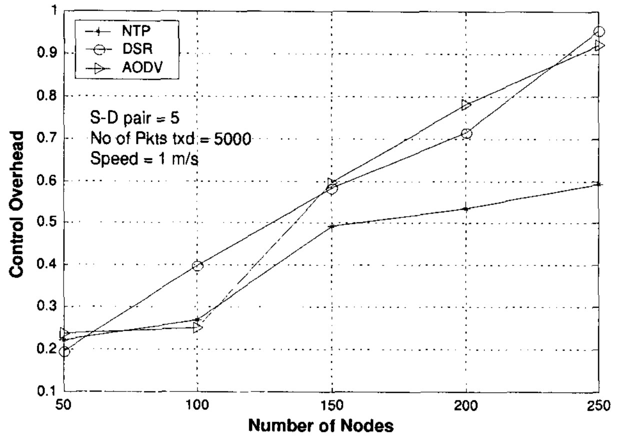


Fig. 24. Control overhead vs. number of nodes.

tion in control overhead. The Control Overhead is found to decrease by as much as 30% with only 3% reduction in throughput when compared to AODV. The performance of NTP is studied using different mobility models proposed for Ad-Hoc networks as shown in Table 9.

DSR is found to perform very efficiently for smaller networks (practically confined to a building). Performance of DSR degrades for larger networks. In AODV when a node suddenly reboots, it forgets all the sequence numbers and the route table. It takes some time for the node to settle down and route the packets efficiently. When a node suddenly reboots or enters the network newly in NTP, it initiates beacons to calculate the neighbor table, which helps not only the source node but also all other nodes in the network to update their neighbor tables.

In addition to link breaks, the finite delay experienced by the packets also contributes to the reduction in throughput. The delay can be checked by making use of the TTL (Time to Live) field. In the current implementation, the TTL field is not used. In AODV, the nodes listen to the on going traffic. Every node processes all the packets it hears whether it is destined to it or not and extracts the last address field to update the route table. This is the reason for the efficient route maintenance. Right now, in NTP, the nodes do not process the packets that are not destined to them. The technique used by AODV can be used in NTP to update the neighbor table. This will strengthen the route maintenance. At this stage, it shows the potential to replace AODV as the standard routing algorithm for mobile ad hoc networks.

Table 9. Comparison of different mobility models.

MOBILITY MODEL	NTP (Node transition probability based routing)		AODV (Ad-hoc on demand distance vector routing)	
	Throughput	Control overhead	Throughput	Control overhead
Random Walk				
i) Uniformly distributed speed	0.9608	0.1149	0.9913	0.1592
ii) Exponentially distributed speed	0.9664	0.0932	0.9934	0.1398
Random Waypoint				
i) Uniformly distributed speed	0.9619	0.1226	0.9954	0.1470
ii) Exponentially distributed speed	0.9541	0.1046	0.9934	0.1422
Gauss Markov	0.9149	0.0849	0.9942	0.1374

REFERENCES

[1] Charles E. Perkins , Introduction to Ad hoc Networking, Addision Wesley, Dec 2001.

[2] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector routing," in *Proc. IEEE WMCSA'99*, New Orleans, LA, Feb. 1999, pp. 90–100.

[3] C. E. Perkins and P. Bhagwat, "Highly dynamic destination- sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM'94*, London, UK, Sept. 1994, pp. 234–244.

[4] J. Broch, D. B. Johnson, and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," *Internet Draft*, draft-ietf-manet-dsr-00.txt, Mar. 1998, Work in Progress.

[5] R. Dube *et al.*, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Pers. Commun.*, pp. 36–45, Feb. 1997.

[6] C.-K. Toh, "Associativity-based routing for ad-hoc mobile networks," *Wireless Pers. Commun. J.*, special issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol. 4, no. 2, pp. 103–139, Mar. 1997.

[7] S. Agarwal *et al.*, "Route-lifetime assessment based routing (RABR) protocol for mobile ad hoc networks," in *Proc. IEEE ICC'2000*, New Orleans, USA.

[8] E. M. Royer and C.-K. Toh, "A review of current protocols for ad hoc mobile wireless networks," *IEEE Pers. Commun.*, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[9] S. Radha and S. Shanmugavel, "Node transition probability based unicast routing algorithm or mobile ad-hoc networks," in *Proc. IEEE Int. Conf. Wireless Lan and Home Networks*, Singapore 2001.

[10] S. Radha and S. Shanmugavel, "Node transition probability based routing," in *Proc. 12th IEEE LANMAN'2002*, Stockhlom, Sweden, Aug. 11-14, 2002.

[11] Glomosim User Manual , http://pcl.cs.ucla.edu/projects/glomosim.

[12] R. Bagrodia *et al.*, PARSEC: A parallel simulation environment for complex systems (IEEE OCT1998).

[13] R. A. Meyer and R. Bagrodia, *PARSEC User Manual Release 1.1*, http://pcl.cs.ucla.edu/, Jan. 1999.

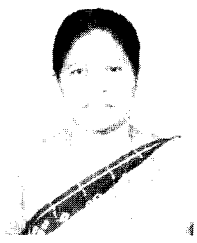
[14] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, Oct. 1995.

[15] T. S. Rappaport, S. Y. Seidel, and K. Takamizawa, "Statistical channel impulse response models for factory and open plan building radio communication system design," *IEEE Trans. Commun.*, vol. COM-39, no. 5, pp. 794–807, May 1991.

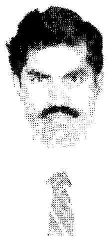
[16] T. Camp, V. Davies, and J. Boleng, "Mobility Models for Ad Hoc Network Simulations," June 6, 2001.

[17] G. Pei, *Scalable Routing Strategies for Large Ad Hoc Wireless Networks*, Ph.D. Thesis, UCLA, 2000.

[18] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proc. IEEE INFOCOM'2000*, Tel Aviv, Israel, Mar. 2000, pp. 3–12.



Sankararajan Radha graduated from Madurai Kamraj University, in Electronics and Communication Engineering during the year 1989. She obtained her Master degree in Applied Electronics from Bharathiyar University. At present she is working as Assistant Professor in the Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Klavakkam, India. She is completed her PhD programme at Anna University, Chennai. She has 13 publications in National and International conferences in the area of Mobile Ad hoc Network.



Sethu Shanmugavel graduated from Madras Institute of Technology in Electronics and Communication Engineering in 1978. He obtained his PhD degree in the area of Coded Communication and Spread Spectrum Techniques from Indian Institute of Technology, Kharagpur, INDIA. During the period November 1978 to March 1989 he was working as Senior Research Assistant at Radar and Communication centre, IIT, Kharagpur, where he worked in various sponsored research projects on Digital Communication and Spread Spectrum Techniques.

He joined the faculty of the Dept. of E and ECE at IIT Kharagpur as Lecturer in 1987 and became Assistant Professor in 1991. Presently he is a Professor at School of Electronics and Communication Engineering at Anna University, Chennai, INDIA. He has published more than 50 research papers in the National and International Conferences and Journals in the area of Mobile Ad hoc Networks, ATM Networks, Spread Spectrum Communication and Error Control Coding. His current areas of interest are Mobile Ad hoc Networks, Broadband ATM networks and CDMA Engineering.

He received IETE - CDIL Award in September 2000 for the Best Paper published in the IETE Journal of Research.