Buffer Aware Routing in Interplanetary Ad Hoc Network

Kamal Mistry, Sanjay Srivastava, R. B. Lenin

DA-IICT, Gandhinagar, Gujarat 382 007
{kamal_mistry, sanjay_srivastava, lenin_bhavanandan}@daiict.ac.in

***Abstract*—Recent research has shown that existing TCP/IP protocol family is likely to exhibit poor performance in space communication networks. To deal with communication challenges in deep space, an architecture called Interplanetary Ad hoc Network (IPAN) is envisioned to establish a communication infrastructure among planets, natural and artificial satellites, and various mission elements such as spacecrafts and rovers. Some of these classes of nodes in such networks may be resource constrained in terms of storage, energy and processing power. This work proposes a probabilistic routing protocol called "Buffer Aware Routing Protocol in Interplanetary Ad hoc Network (BARPIN)" for routing best-effort network traffic based on the store and forward principle of Delay Tolerant Networking (DTN). IPAN is modeled as the network of two kinds of nodes, one with deterministic mobility patterns having greater resources and the others having random movement patterns with limited buffer space. Further we estimate the minimum required buffer size of nodes as a function of network traffic for optimal performance of the protocol.**

Keywords: Mobile ad hoc network, Interplanetary Network, routing, buffer management.

I. Introduction

The routing problem in Mobile Ad hoc Network (MANET) is a well-studied problem. For MANET routing problems, it is assumed that the network is connected or if there is a network partition, then it exists for very short interval of time. Moreover, the objective of such protocol is to route the traffic end-to-end over the best currently available path in terms of delay and bandwidth. In real life, there is a class of networks which does not possess such characteristics. In such networks, due to frequent network partitioning, end-to-end connectivity between source and destination nodes is rare and there is a very high propagation delay because of link unavailability or long distances between nodes. Such networks are called Delay Tolerant Networks (DTNs). In DTN, a complete end-to-end path may remain unavailable for all the times so routing is performed over time to achieve eventual delivery and intermediate nodes must be equipped with sufficiently large and long-term storage.

Interplanetary Ad hoc Network (IPAN) is a network, envisioned to provide communication between regional networks of different planets, natural as well as artificial satellites and spacecrafts. IPAN is a specific case of DTN and communication protocols for IPAN need to have the following two fundamental properties of DTN protocols, (1) it should operate in a "store and forward" mode, very similar to e-mail, where messages are held at routers along the way until a forward path is established and (2) it should avoid the need for a sender to store data until an acknowledgement is received from the other end by operating in a "custodial" mode, which is a hop-by-hop acknowledgement mode.

Many protocols exist in the current literature for routing in DTN. Epidemic routing protocol proposed in [3] is the first such protocol. A probabilistic routing protocol for DTN called PROPHET, proposed in [1] is an extension to the epidemic routing protocol. The PROPHET estimates a probabilistic metric called *delivery predictability (DP)* at every node and the routing is done based on the DP values of the neighbor nodes for the destination of the packet. Protocols proposed in [5], [9] and [16] are also probabilistic routing protocols. While the routing protocols proposed in [4], [13] and [15] are deterministic routing protocol in the sense that the nodes in the network know the movement patterns of other nodes or the topology information is disseminated using the link state or distant vector routing algorithms. Most of these protocols assume that the available buffer is of infinite capacity. The protocols proposed in [12], [6], [10], [11] and [14] handle the buffer when there is a congestion at the node using different buffer management policies. Interrogation based protocols are proposed in [7] and [8] for ad hoc satellite networks, where the satellites interrogate each other to learn more about network topology and nodal capacity to make intelligent routing decisions.

In this paper, we have proposed a protocol called *"Buffer Aware Routing Protocol in Interplanetary Ad hoc Network (BARPIN)"* for IPAN with heterogenous nodes, which takes into account the buffer size of neighboring nodes and the past history of encounters with other nodes while taking routing decision. In this way it tries to reduce the chances of packet drop at the next hop nodes due to buffer overflow. In the next section we discuss the network model we have proposed for IPAN and the assumptions we have made while designing the BARPIN. In section 3, we discuss the protocol BARPIN in detail, in section 4 we show the simulation results for our protocol and section 5 concludes the work.

II. Network Model

We propose a *best effort* communication protocol for Inter-Planetary Ad hoc Networks (IPANs), where communicating

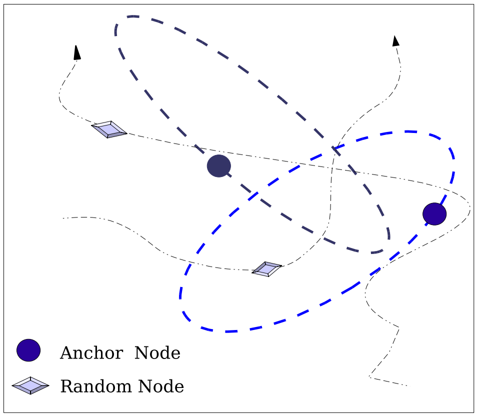

Fig. 1. Network Model for IPAN

entities are of different nature in terms of their implementation architecture, the amount of resources they possess, their trajectory of movements and their locations. These entities transfer different types of traffic among themselves. The traffic such network carries can be classified into two general categories, *real time traffic* and *best effort traffic*. The information which real time traffic carries is very critical and sensitive to delay and information loss. For example the traffic of command, control, and navigation systems of various entities in the space network are of such type. They are given higher priority over other traffic and generally the resources in terms of energy, bandwidth, processing power and memory are reserved for them. While the best effort traffic carries the information which is not sensitive to delay and no special resources are reserved for such traffic. For example, transmission of data of photographs taken by space telescope or various environmental information of other planets and satellites can tolerate delay. BARPIN is targeted to route the best effort traffic using the excess resources which are left after doing the reservation of the real time traffic.

As shown in Figure 1, our network model comprises of two different types of nodes, one which move in deterministic orbits and the other which move randomly in the network. The nodes with deterministic orbits, we call them *anchor* nodes, have fixed orbits of movement and their trajectories are known by all the other anchor nodes in the network. Any anchor node in the network can determine the next future contact time with any other anchor nodes. The second type of nodes are those whose movement patterns are not deterministic, we call them as *random* nodes, and their movement trajectories are unknown to all the random as well as anchor nodes in the network. Further we assume that anchor nodes have greater buffer availability than the random nodes. We have assumed that sufficient memory to store the buffered packets is available at anchor nodes. With this network model, the nodes on planets and natural satellites can be modeled as anchor nodes, since their trajectories are well known and the higher capability resources can be installed on them. Further, the nodes like spacecrafts and rovers on remote planets can

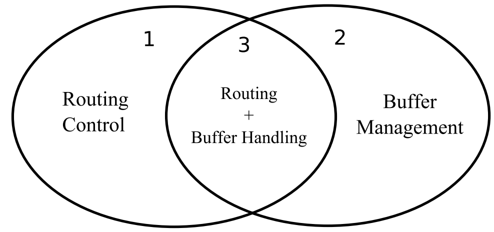

Fig. 2. BARPIN Routing Strategy

be modeled as random nodes. We assume that they have the capability to change their direction of movement at will based on the different event triggers and these changes in direction of movement may not known to other nodes in the network. So the assumption about the random movement pattern to be unknown to all the random as well as anchor nodes in the network is reasonable. Since BARPIN is a unicast routing protocol, we assume that only a single copy of a message exists in the network at any instance of time and all the messages are of same size and of same priority. Further we assume that the network is so sparse that the end-to-end path between the source and the destination of the packet is not available, hence the routing decision to select next hop node for the packet is taken only based on the set of nodes in the neighborhood of the packet forwarding node.

III. The BARPIN Protocol

The aim of BARPIN is to achieve higher delivery ratio (number of packets delivered to the destination successfully over the total number of packets originated by the source) by taking available buffer size of the nodes into consideration, using minimum network knowledge and with minimum protocol overhead.

The overall routing strategy looks like as shown in Figure 2. The functionalities of all the parts are as follows:

1 This part of the protocol handles the generation and dissemination of control information to the neighbor nodes.

2 This part deals with the condition of buffer overflow at the congested node in the network. It handles the buffer overflow either by dropping the buffered packets or by offloading the data to the neighbor node's storage in order to prevent the unnecessary packet loss. The mechanism is independent of routing protocol and can be helpful for any IPAN or DTN routing protocol to reduce the data loss due to congestion.

3 Here the packet forwarding logic is implemented by using the information generated in part-1 (Routing Control). Based on the destination of the packet, forwarding node selects a node from the neighboring nodes as the next hop node of the packet, or may choose not to forward the packet to any of the neighboring node, if the likelihood of neighbor nodes are less then the packet forwarding node.

This design strategy of BARPIN are discussed in following sections as: (A) *Routing control*- deals with generation and dissemination of control information to the neighbor nodes;

(B) *Buffer aware routing*- where the actual packet forwarding logic is implemented and (C) *Buffer management*- deals with the condition of buffer overflow at the congested node.

A. Routing control

In order to make good routing decision, a routing node must have some information about the network namely network connectivity, mobility patterns of the nodes, resource availability etc. In the IPAN, due to frequent network partitions, the link state or distance vector flooding algorithms are very costly, so whenever two nodes encounter, first they exchange the routing control information they have built, based on the past contact with other nodes. After that they update their own control information with the newly received information and make the routing decision using this updated information. In BARPIN, this information is generated and exchanged during the encounter between nodes. Whenever two nodes encounter each other (i.e. they are within the communication range), they exchange the following control information before initiating the actual data transfer: (i) future contact probability (*fcp*) values for all the other nodes in the network, based on their past history of encounters, (ii) their buffer occupancies (B_f) which is the ratio of occupied buffer and the total buffer size of the node and (iii) average encounter rate (δ), which is the average node encounter rate over some past time interval. Additionally when an anchor node meets a random nodes it also sends the expected time duration after which it will encounter other anchor nodes in the network.

Each node in the network maintain the future contact probability (*fcp*) for all the nodes in the network based on the previous contacts. Following is the approach similar to that of delivery likelihood value P_{dl} in MaxProp routing algorithm for disruption-tolerant networks by John Burgess et al [11]. We calculated *fcp* as follows: Let f_j^i be the *fcp* that node i has for node j. Initially for all the nodes i, j

$$f_j^i = \frac{1}{(|N_0| - 1)}, \text{ for } i \neq j, \tag{1}$$

where N_0 is the initial number of nodes in the network. When the node j encounters the node i, then the value of f_j^i is incremented by 1 and then all values for f^i are renormalized so that they all add up to 1 using the following equations:

$$f_j^i = \begin{cases} (f_j^i)_{old}/2, & \text{if the node encountered} \neq j, \\ ((f_j^i)_{old} + 1)/2, & \text{if the node encountered} = j, \end{cases} \tag{2}$$

such that,

$$\sum_{all\ j} f_j^i = 1, \text{ for } i \neq j.$$

Using (1) and (2), each and every node in the network creates its own table of *fcp* values for all the other nodes and exchanges this table with other nodes at the time of encounter during every contact. It should be noted that due to the evolution of f_j^i as given in (2), the choice of N_0 does not continue to affect the f_j^i after a few iterations. There are a few important dissimilarities that exist between MaxProp and BARPIN. MaxProp only exchanges the delivery likelihood values P_{dl} upon encounter with other nodes, while BARPIN exchanges the *fcp* and the buffer occupancy (B_f) values of each other. MaxProp calculates the cost of the path for the destination node up to maximum path length of 10 and forward the packet to the next hop node on minimum cost path. For the selection of next hop node, BARPIN considers only the likelihood of the neighbor nodes of the packet forwarding node. This is important since, in space network due to long distance and large signal propagation delay between nodes, old topology information may not be of much importance at the time when routing decision is being taken. Finally to make the routing decision in BARPIN, the buffer occupancy of the neighbor nodes is considered with their future contact probability to the destination node. This helps in reducing the packet drop at the next hop node due to buffer overflow. For this MaxProp rely only on the delivery likelihood values and based on that the minimum cost path for the destination node. On the other end, in MaxProp when a packet is delivered to its destination, an acknowledgment is sent back from the destination, which is propagated to all peers in the network. This is not done in BARPIN since it aims to route the best effort traffic using the excess resources which are left after doing the reservation of the real time traffic.

TABLE I
POSSIBLE PACKET FORWARDING CASES

No.	Packet Forwarder	Destination
1	Random	Random
2	Random	Anchor
3	Anchor	Random
4	Anchor	Anchor

B. Buffer aware routing

Based on the type of forwarding node and the destination of the packet, we have four possibilities to consider in the design of BARPIN as shown in Table I.

Since trajectory of a random node is unknown, if the destination of the packet is a random node, then forwarding node selects one node from its neighborhood as the next hop node regardless of its type, who has highest likelihood to encounter the destination with sufficient available buffer. If the destination of the packet is an anchor node, then the protocol will use the future contact time between anchor nodes and the delivery likelihood of random nodes to select the next hop node. The proposed routing protocol is probabilistic in nature with consideration of certain information of anchor nodes. We assume that the network is sparse so that the propagation of correct routing control information to all the nodes at correct time is not feasible. Hence, the exact information of the entire topology of the network can not be known to the nodes due to old routing control information and random movement of the random nodes. Hence, for next hop selection, the packet forwarder node looks for the best possible next hop node from all the neighbor nodes at the moment of routing. The outline of the protocol is given in Algorithm 1.

Algorithm 1 Protocol Logic

Let $L = \{l_1, l_2, \ldots, l_{n+m}\}$ be a set of all the nodes in the neighborhood of the packet forwarding node. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of all the anchor nodes and $R = \{r_1, r_2, \ldots, r_m\}$ be a set of all the random nodes in the neighborhood, such that $L = A \cup R$.
if destination of the packet is a random node **then**
  Select a node $l_{fn} \in L$ among all the $l_i \in L$ where $i \in [1, n+m]$, as the next hop node, which has highest likelihood to encounter the destination with sufficient available buffer.
else if destination of the protocol is an anchor node **then**
  if all the nodes in the neighborhood are random nodes **then**
    Select a node $r_{fn} \in R$ among all the $r_i \in R$ where $i \in [1, m]$, as the next hop node, which has highest likelihood to encounter the destination with sufficient available buffer.
  else if all the nodes in the neighborhood are anchor nodes **then**
    Select a node $a_{fn} \in A$ among all the $a_i \in A$ where $i \in [1, n]$, as the next hop node, which takes minimum time to encounter the destination with sufficient available buffer.
  else if both the types of nodes are there in neighborhood **then**
    Select a node $r_{fn} \in R$ among all the $r_i \in R$ where $i \in [1, m]$, which has highest likelihood to encounter the destination and select a node $a_{fn} \in A$ among all the $a_i \in A$ where $i \in [1, n]$, which takes minimum time to encounter the destination.
    Select one node from these two, either r_{fn} or a_{fn} as the next hop node through which the chances of delivering the packet to its destination is higher.
  end if
end if

If the destination of the packet is a random node, then forwarding node selects one of its neighboring node that has the highest likelihood to meet the destination with sufficient available buffer. If the destination of the packet is an anchor node, then the protocol will use the future contact time between anchor nodes and the likelihood of random nodes to select the next hop node. As shown in Figure 3 whenever any node, say F, receives a packet for which the destination node is D, then it has to take a proper routing decision to forward the packet. In that case F selects a node from its current neighborhood, which has higher likelihood to meet the destination than the node itself or if it cannot find such a node, then it keeps the packet with itself. Each of the four forwarding scenario devolves into three cases based on the types of nodes a packet forwarder has in its neighborhood. They are (1) All the neighbors are random nodes, (2) All the neighbors are

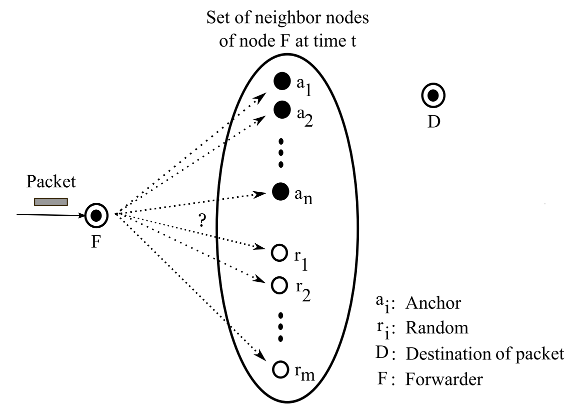

Fig. 3. Selection of next hop node to take routing decision for incoming packet

anchor nodes and (3) Some nodes are random and some are anchor nodes in the neighborhood. If the destination of the packet is a random node then all the nodes, random as well as anchor nodes have similar information, i.e. the fcp and B_f values of the destination. So regardless of the type of the packet forwarding node, all the nodes use the same method of selecting the next hop node. The packet is forwarded to the neighbor node only when it has higher likelihood to deliver the packet to destination than the forwarding node, otherwise the packet stays with the forwarding node.

Every node maintains buffer occupancy information of its neighboring nodes and future contact probabilities for all the nodes in the network. A node uses both of these to select its next hop node so that the chances of a packet getting delivered increases and chances of a packet getting dropped due to buffer overflow decreases. The likelihood of a node to deliver the packet to its destination, P_{LH}, is a function of both the fcp and the B_f. For a node, the value of P_{LH} should increase with increase in fcp and decrease with increase in B_f. We define P_{LH} as,

$$P_{LH} = \frac{fcp}{1 + B_f}. \tag{3}$$

The routing decision for all the three cases are done as listed below:

1) Only random nodes in the neighborhood: In this case a packet forwarding node has to select a most favorable random node r_{fn}, among all the random nodes in its current neighborhood, which is done using Algorithm 2. If the likelihood of

Algorithm 2 Selection of most favorable random node

1: Let R be a set of all the random nodes in the current neighborhood of the packet forwarding node and let $m = |R|$, where $|R|$ is the cardinality of set R. For each $r_i \in R$, where $i = 1, 2, \ \ldots \ , m$; calculate $P_{LH\,i}$ using (3).
2: If the average encounter rate of node r_i is δ_i and if r_{fn} is the most favorable random node in the current neighborhood then

$$r_{fn} = \{r_i | \ (P_{LH\,i}/\delta_i) \text{ is maximum, for } \ i = 1, 2, \ \ldots \ , m\}$$

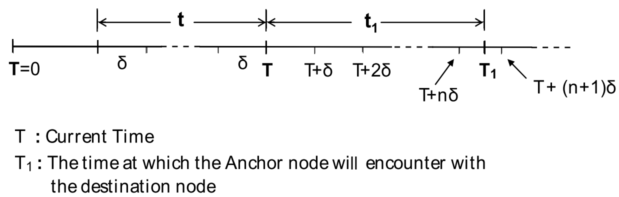

Fig. 4. Comparing anchor and random nodes with respect to time.

r_{fn} is greater than the likelihood of the forwarding node then the packet is forwarded to node r_{fn}, otherwise the forwarding node keeps the packet with itself to route it in future.

2) Only anchor nodes in the neighborhood: If the destination of the node is an anchor node then, the forwarder node knows the time its anchor neighbors will take for the next future contact with the destination. However we have to consider their buffer occupancies also, so that with decrease in delay we can minimize the chances of packet getting dropped due to congestion at the neighbor node. Let the future contact time with destination and the buffer occupancy of neighbor node i be t_i and B_{fi} respectively. To discourage choosing a next hop node with low t_i together with low buffer availability, we define an effective time, $t^{'}$ which takes into account both of the above factors as,

$$t_i^{'} = t_i(1 + B_{fi}). \tag{4}$$

The selection of best anchor node from the current neighborhood of the packet forwarding node can be done using Algorithm 3. If the packet forwarding node is also an anchor

Algorithm 3 Selection of most favorable anchor node

1: Let A be a set of all the anchor nodes in the current neighborhood of the packet forwarding node and let $n = |A|$, where $|A|$ is the cardinality of set A. For each $a_i \in A$, where $i = 1, 2, \ \ldots \ , n$; calculate $t_i^{'}$ using (4).

2: If a_{best} is the most favorable anchor node in the current neighborhood then

$$a_{best} = \{a_i | \ t_i^{'} \text{ is minimum, for } \ i = 1, 2, \ \ldots \ , n\}$$

node and the effective time to meet the destination of node a_{best} is less than that of forwarding node only then a_{best} is selected as the next hop node, otherwise the packet is not forwarded to a_{best} but stored in the buffer of forwarding node.

3) Both anchor and random nodes in the neighborhood: In this case the packet forwarding node has to find out a most favorable random node r_{fn} from all the random nodes in the neighborhood using Algorithm 2 and a best anchor node a_{best} form all the anchor nodes in the neighborhood using Algorithm 3 in its current neighborhood. One node out of these two is selected as the next hop node. Since they are selected based on different parameters there is a need to create a common scale for comparison. As shown in Figure 4, suppose at time T the forwarding node is taking the routing decision. It knows that in future at time T_1, after t_1 time units from T, the best anchor node a_{best} will meet the destination and its current buffer occupancy is, say B_{fA}. Using (4), the effective time the node a_{best} will take to meet the destination is $t_1^{'} = t_1(1 + B_{fA})$.

Let p be the probability that the most favorable random node r_{fn} is going to encounter the destination. On an average r_{fn} has observed one encounter per δ interval. If there is an encounter in first interval $(T, T + \delta]$, then the value of p is simply the value of *fcp* of node r_{fn} for the destination, say p_1. The probability that the encounter happens with the destination in second δ interval $(T + \delta, T + 2\delta]$ can be given as $p_2 = (1 - p)p$. In general, the probability of encounter with destination in n^{th} interval $(T + (n-1)\delta, T + n\delta]$ is $p_n = (1-p)^{n-1}p$. So P_n, the probability of encounter with the destination in total $n\delta$ intervals is the cumulative distribution function given as $P_n = 1 - (1 - p)^n$. If the buffer occupancy of the random node r_{fn} is B_{fR}, then similar to equation (3), the likelihood of the node r_{fn} to deliver the packet to its destination can be given as,

$$P_n^{'} = \frac{1 - (1 - p)^n}{(1 + B_{fR})}. \tag{5}$$

Selection of the next hop should ideally try to:

(C1) maximize the probability of delivery of the packet to its destination and

(C2) deliver the packet in least amount of time, as a consequence it reduce the end to end delay and average buffer occupancy.

These two criteria do not necessarily overlap. For the best effort service, one may argue that criteria (C1) should be sufficient. However, it is evident that when a packet spends more time in the network, then the average buffer occupancy of the network also goes up. In our model, where buffer is a constrained resource, this may potentially lead to dropping of other packets due to buffer overflow. So to find the optimum solution in this case, is analogous to multi-objective optimization problem. In order to have a solution to such a problem, one must decide how much weight should be given to each parameter. For random node with sufficiently high $P^{'}$, if δ is small and relative to that if an effective time $t_1^{'}$ of an anchor node is large, then selection of random node over an anchor node can be a good choice, otherwise the anchor node is a good choice as a next hop node. The relation which reflects such characteristic is given by the following inequality:

$$\left(\frac{P_n^{'}}{\alpha \left(\frac{n\delta}{t_1^{'}} \right)} \right) > \varepsilon \tag{6}$$

If the inequality in equation (6) holds, then the random node is selected as the next hop node, otherwise the anchor node is selected. The values of α and ε depend on how much weight one wants to give to criterion (C1) or criterion (C2). The higher the value of ε more we prefer the anchor node over the random node. In that case one would go for random node only when δ for that node is very low and the value of $P_n^{'}$ is high.

One would go for the random node over an anchor node, only when by selecting the random node more time can be saved in delivering the packet. For example, if $\varepsilon = 1$, then the inequality of equation (6) can be rewritten as,

$$\delta < \frac{P_n' t_1'}{n\alpha}.$$

For the given network, lower the value of δ, the higher the saving in time in delivering the packet to its destination. In above inequality if one increases the value of α, then to hold the inequality for the given values of other parameters, δ must also decrease. It means that if we want to give more importance to saving in time, then we have to keep the value of α high.

Take a case where $\delta \geq t_1'$, which is a case when even a single encounter of the random node takes more time than the effective next encounter time of the anchor node. Set $\alpha = \varepsilon = 1$ for simplicity, then by equation (6), the following must hold.

$$P_n' > \left(\frac{n\delta}{t_1'}\right)$$

As $\delta \geq t_1'$, for all the cases of n, the right side is greater than or equal to 1, but on left side P_n' indicates the probability value, which can never be greater than 1. So such an inequality never holds, and therefore never go for random node in this case.

C. Buffer management

An IPAN is a store-and-forward network in which if an end-to-end path is not available, then instead of dropping the packet, the node stores the packet in its persistence storage (buffer) until it finds destination of the packet or some other node with higher likelihood to meet the destination. By doing this for long time if the storage of the node becomes full then it has to drop the stored packets from the buffer to make room for the incoming packets or just deny to accept the new packets if it does not want to drop the old packets from the buffer. There can be multiple criteria to drop the packet when there is buffer overflow. BARPIN uses the following: when a new packet arrives at the node and if buffer is full, then drop the oldest packet from the buffer, i.e. the packet which has stayed for longest amount of time in the buffer among all the stored packets. The reason for this is, if a node was unable to forward the packet from longer amount of time in the past, then there is very low chance of forwarding the same packet in future. The other possible policy which can reduce the packet loss, can be to offload the packets from the buffer of congested node to any of the neighbor node with lower buffer occupancy.

IV. Simulation Results

We have implemented BARPIN in version 2.29 of the Network Simulator (ns2) [18]. We compare the performance of BARPIN against the probabilistic routing protocol called PROPHET [1]. The reason for selecting PROPHET for comparison is that like BARPIN, it does not flood the control information in the network nor does it run the shortest path algorithm to calculate the minimum cost path for the destination of the packet. Both these protocols make the routing decision based on the current neighborhood of the packet forwarding node using the control information they have exchanged during past encounter with other nodes.

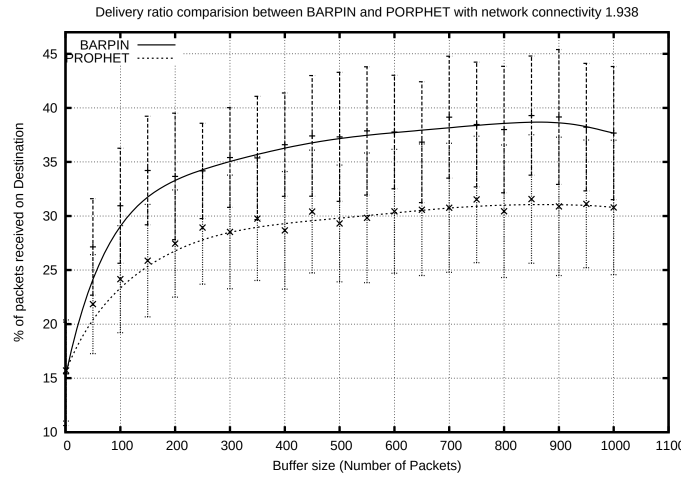

Fig. 5. Delivery ratio vs Buffer size when packet destination is a random node over the field of size 1500×1500 m^2

The default parameter values of our simulation runs are as follows: We have run the simulation on the field size 2000×2000 m^2 and the number of nodes are 15, out of which 10 random nodes and 5 anchor nodes are there. To measure the sparseness in the simulated network, we used the concept of average node degree of the network. For each field size, we have calculated average node degree, a real number, which at any instance of time is the average number of neighbors a node has in the changing topology. In other words, if the average node degree for a field is 1.5 then throughout the simulation period, a node has on an average at least 1.5 nodes in its neighborhood. The lesser the value of average node degree, more sparse the network is. To mimic the varying sparseness of the space network, we have run our simulations on three different fields with sizes 1500×1500 m^2, 2000×2000 m^2 and 2500×2500 m^2 with average node degree 1.938, 0.998 and 0.621 respectively. Each data packet is of 1000 bytes and the application sends the data at the rate of 10 pkt/sec from 5 to 500 second time interval with total background traffic of 20 pkt/sec. The background traffic is generated by two different sources for two different destinations. We have run the simulation for 1000 second. The buffer size of anchor node is 3 times greater than the buffer size of random node. The buffer timeout period is 500 sec, which indicates the maximum amount of time a packet can be stored in any node's buffer. When there is a buffer overflow, the packet which has stayed longer than this period in the buffer, is dropped by assuming that the node is unable to forward the packet in near future. We have fixed the value of α equals to 2.0 and the value of ε be 1.0.

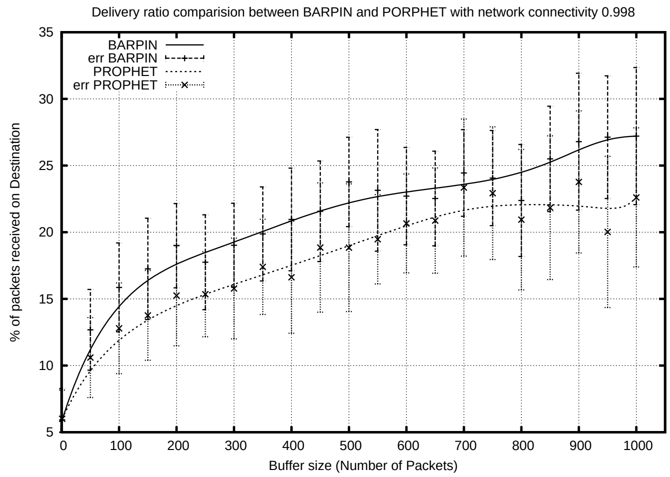

Fig. 6. Delivery ratio vs Buffer size when packet destination is a random node over the field of size 2000×2000 m^2

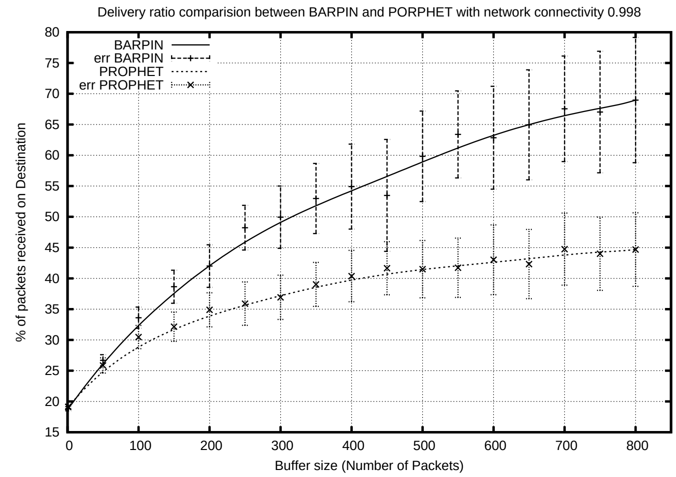

Fig. 8. Delivery ratio vs Buffer size when packet destination is an anchor node over the field of size 2000×2000 m^2

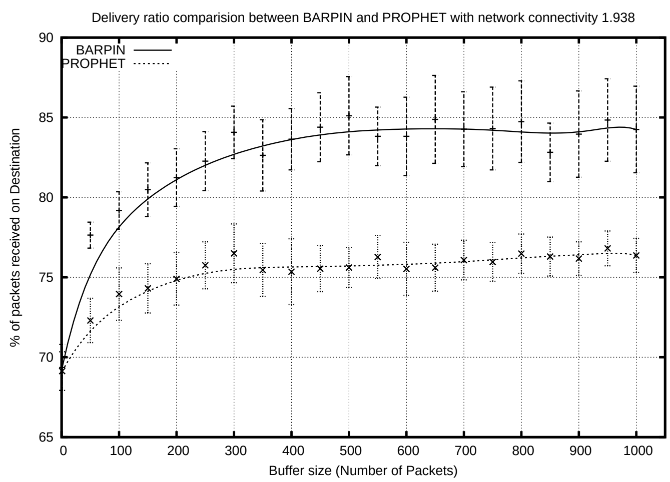

Fig. 7. Delivery ratio vs Buffer size when packet destination is an anchor node over the field of size 1500×1500 m^2

A. Simulation with Different Field Sizes

All the graphs are results of an average of 15 scenario with different initial conditions for the mobility of the nodes. These scaled parameters are chosen to mimic the IPAN parameters. The vertical bars at each data points indicate the standard error in the graph, which we have calculated by $average\ value \pm (standard\ deviation/2)$. The graphs in Figure 5 and Figure 6 show % of packets delivered when destination of the packets is a random node and the graphs in Figure 7 and Figure 8 show the % of packets delivered when destination of the packets is an anchor node over two different filed sizes of 1500×1500 m^2 and 2000×2000 m^2. When the destination of the packet is a random node, then BARPIN treats both types of nodes as same, just like in PROPHET, and forwards the packet based on their fcp and B_f values. To select the next hop node, PROPHET does not consider the buffer occupancy of the neighbor nodes, so it may select the next hop node which has no free buffer available or the buffer occupancy of which is very high, which may cause higher packet loss than BARPIN which results in lower delivery ratio. Because of the use of future contact information of the anchor nodes and an awareness about the buffer occupancies of the neighbor nodes, when destination is an anchor node BARPIN gives much higher performance than the PROPHET as shown in Figure 7 and Figure 8. Since buffer size of anchor node is 3 times greater than the buffer size of random node and destination is an anchor node, for the selection of next hop node BARPIN is biased towards the node which has lower buffer occupancy, which result in the higher delivery ration in case of BARPIN over PROPHET.

B. Varying Anchor Node to Random Node Buffer Size Ratio

The design of our protocol is based on the assumption that anchor nodes have higher buffer size than random nodes. So it is interesting to see what happens to the performance of protocol when we change the buffer size of anchor node. For all the simulation runs, we keep the ratio of 3:1 for ratio of anchor node to random node's buffer size. Here we have shown in Figure 9, 10 and 11, the performance of both the protocols, by varying the ratio of anchor node to random node's buffer sizes for the values 1:1, 3:1 and 5:1, for the field size 2000×2000 and the data rate of 10 pkts/sec.

For the case $B_A = B_R$, i.e. when the buffer sizes of anchor nodes and random nodes are same, then for lower buffer sizes, the delivery ratio of both the protocols are more or less same, but BARPIN starts giving better performance for higher buffer sizes. For the other cases of $B_A = 3 \times B_R$ and $B_A = 5 \times B_R$, BARPIN starts giving better performance from the beginning and the difference keeps increasing with increase in buffer size. The reason for increase in delivery ratio in BARPIN is that, it tries to forward more and more packets to anchor nodes since they have higher buffer size and due to that their buffer

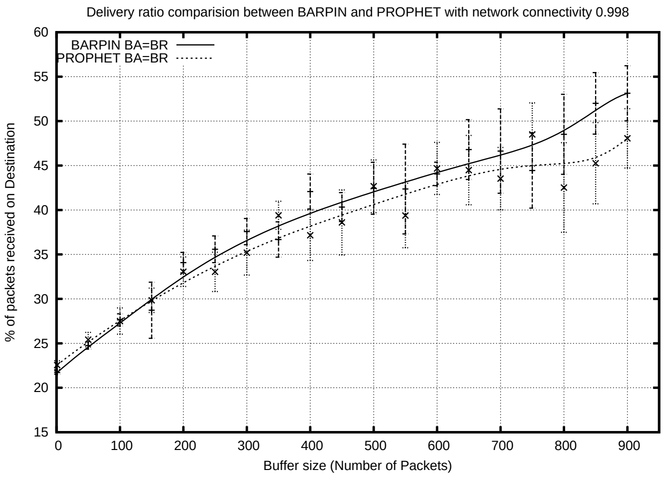

Fig. 9. Delivery ratio vs Buffer size for $B_A = B_R$ when destination is an anchor node

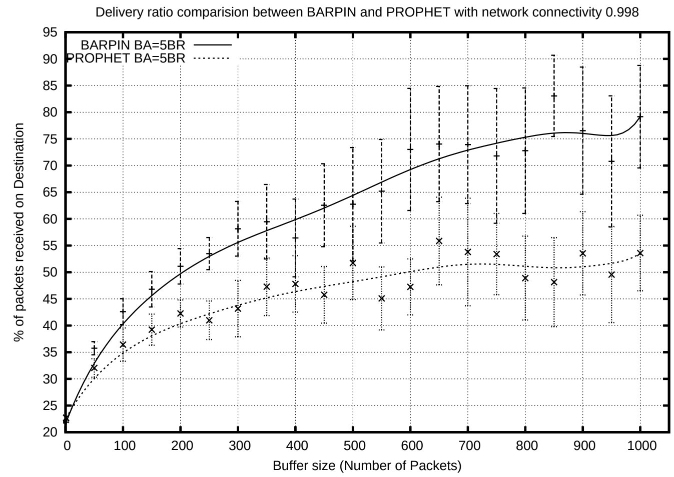

Fig. 11. Delivery ratio vs Buffer size for $B_A = 5 \times B_R$ when destination is an anchor node

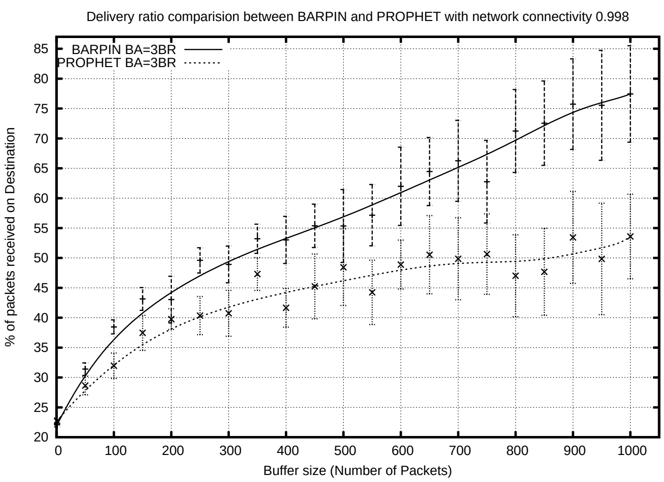

Fig. 10. Delivery ratio vs Buffer size for $B_A = 3 \times B_R$ when destination is an anchor node

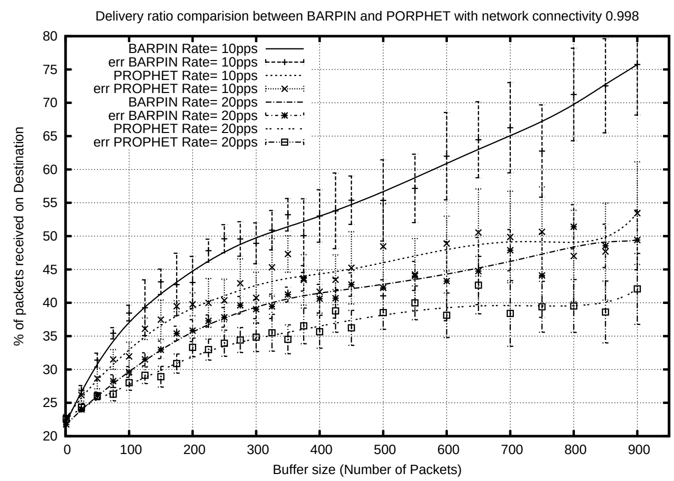

Fig. 12. Delivery ratio vs Buffer size for different data rates

occupancy remains low when compared to random nodes. In this way BARPIN cause lower packet loss when there is a buffer overflow at any node.

C. Simulation with Different Data Rates

We have studied the performance of BARPIN by varying the data rate, in terms of number of packets transmitted by the source in one second. The graph is shown in Figure 12. When the data rate increases from 10 pkt/sec to 20 pkt/sec, then for the given buffer size more number of packets are generated and the buffers at various nodes start filling rapidly. This condition causes congestion at the intermediate nodes and this leads to packet loss which reduces the delivery ratio for higher data rate. In all this cases PROPHET observes higher data loss than compared to BARPIN.

D. Estimation of saturation point for a given traffic rate

It can be seen from the graph in Figure 12 that with increase in data rate the initial slope of the curve reduces. It can be observed from the graph that, the percentage of packets delivered to the destination increase almost linearly with increase in buffer size up to some point for all the data rates and then after, the behavior of the graphs change and the slope of the curves start reducing rapidly. This indicates that the linear increase in delivery ratio in the beginning is only due to increase in buffer size, but after that the other parameters of the network like movement patterns of the nodes, average node degree, inter-encounter time between nodes etc. start affecting the delivery ratio. We call this value of buffer size a *saturation point* of the network for the given traffic rate. Let T_R be a total traffic rate of the network. Let $\bar{\tau}$ be an average inter encounter time between any two nodes in the network. When a node has contact with any other node, then it has an opportunity to

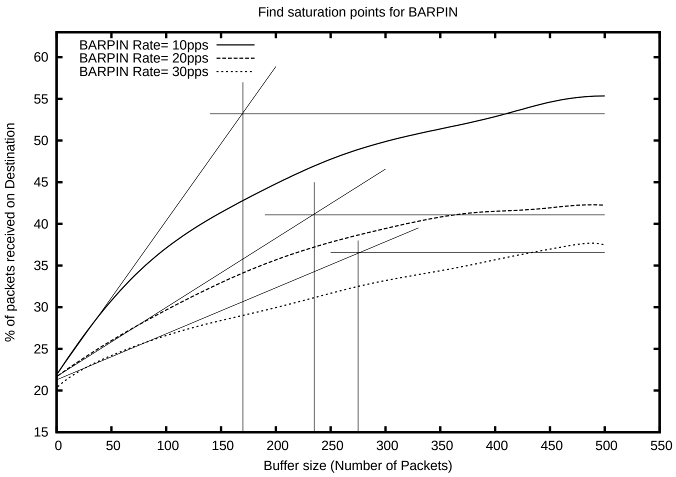

Fig. 13. Determination of the slopes and saturation points for different data rates

free its buffer by transferring packets from its buffer to the newly encountered node. It means that a node has to store the packets for at least $\bar{\tau}$ time interval. In worst case, if all the traffic generated in the network is forwarded to a single node n_w, then the minimum buffer size the node n_w should have, in order to avoid packet drop due to buffer overflow, is $B_w = T_R \times \bar{\tau}$. Such condition rarely happens in practice and due to mobility of the nodes traffic gets distributed among all the nodes in the network. Hence the required buffer size is less than that of the worst case size B_w. Let us assume that the minimum required buffer size at the node for the given traffic rate is B, which is some fraction, say γ, of the worst case buffer size B_w, that is $B = \gamma \times B_w$. Hence the minimum required buffer size or the "saturation point" of the network for the given traffic rate can be given as,

$$B = \gamma \times T_R \times \bar{\tau}. \quad (7)$$

In equation (7), γ is the *"encounter diversity factor"*, which signifies the fraction of total number of nodes encountered by the given node. If total number of nodes in the network are N, then the value of γ lies in the range $\frac{1}{N-1} \leq \gamma \leq 1$ where $N > 1$. To do the detail analysis of graph in Figure 12, we run the simulation for three different data rates of 10, 20 and 30 pkt/sec by taking more intermediate results for the buffer from 1 to 500. The result is shown in graph of Figure 13, where all the plots are of BARPIN, for different data rates.

Table II shows the slopes of all the curves for BARPIN obtained from the graph shown in Figure 13. Higher the data rate, greater the rate of buffer filling at various nodes, hence the $slope \propto 1/(data\_rate)$. For the given network if the data rate increases, then the value of saturation point should increase and the slope must decrease. The value of γ for our network is 0.301 which we have calculated from the simulation results of data rate 10 pkt/sec. In the network we have studied, the value of $\bar{\tau}$ is 18.752 sec and the value of T_R is 30 pkt/sec for the data rate 10 pkt/sec. Taking $B = 170$, the value of γ can

TABLE II
SLOPES AND SATURATION POINTS FOR DIFFERENT DATA RATES

Data Rate (pkt/sec)	Slope	Saturation Point (Buffer size)	
		Simulation results	Estimated results
10	0.1653	170	-
20	0.0783	235	225
30	0.0480	275	281

be calculated using equation (7). In Table II we have shown the estimated saturation points using equation (7), which are closer to the approximate values calculated from the graphs of simulation results.

V. SUMMARY AND CONCLUSIONS

We have proposed BARPIN, an effective buffer aware probabilistic routing protocol for Interplanetary ad hoc Networks, which uses knowledge about the connectivity and the resources consumption of the nodes to make an efficient routing decision. The major contributions of this work are: (1) Modeling of Interplanetary ad hoc network as a network of heterogenous nodes with different capabilities, (2) Developing a probabilistic routing protocol BARPIN, for interplanetary ad hoc networks, that includes the buffer availability state in the routing decision, (3) Formulation of a model to estimate the required optimal average node buffer size for different traffic rates.

We have simulated and compared the performance of BARPIN against the probabilistic routing protocol, PROPHET, for different field sizes, traffic rates, available buffer resources on various nodes and for different source-destination pairs. For all these cases BARPIN performs better than the PROPHET. Even for very sparse network, where the DTN issues are at their most daunting, BARPIN gives better performance than PROPHET in terms of packet delivery for the same buffer size.

REFERENCES

[1] A. Lindgren, A. Doria, O. Schelen, *"Probabilistic Routing in Intermittently Connected Networks"*, In Proceedings of the The First International Workshop on Service Assurance with Partial and Intermittent Resources(SAPIR 2004), August 2004.

[2] Abdelmajid Khelil, *"Mobility-Aware Buffering for Delay-Tolerant Ad Hoc Broadcasting"*, SPECTS, July 2006.

[3] Amin Vahdat and David Becker, *"Epidemic routing for partially-connected ad hoc networks"*, Technical Report CS-2000-06, Duke University, 2000.

[4] Archana Sekhar, Manoj BS, Siva Ram Murthy, *"MARVIN: Movement-Aware Routing oVer Interplanetary Networks"*, IEEE SECON, 2004.

[5] B. Burns, O. Brock, B. Levine, *"MV Routing and Capacity Building in Disruption Tolerant Networks"*, IEEE INFOCOM, 2005.

[6] Brian Gallagher, David Jensen, Brian Levine, *"Explaining Routing Performance in Disruption Tolerant Networks"*, University of Massachusetts Amherst, Technical Report 2005.

[7] Chien-Chung Shen, Girish Borkar, Sundaram Rajagopalan, Chaiporn Jaikaeo, *"Interrogation-Based Relay Routing for Ad hoc Satellite Networks"*, IEEE Globecom 2002.

[8] Chien-Chung Shen, Girish Borkar, Sundaram Rajagopalan, Chaiporn Jaikaeo, *"A flexible routing architecture for ad hoc space networks"*, Computer Networks 46 pages 389−410, 2004.

[9] Darshak Thakore and Subir Biswas, *"Routing With Persistent Link Modeling In Intermittently Connected Wireless Network"*, Military Communications Conference, 2005 (MILCOM-2005, IEEE).

[10] Evan Jones, Lily Li, Paul Warde, *"Practical Routing in Delay-Tolerant Networks"*, Workshop SIGCOMM August 2005.
[11] John Burgess, Brian Gallagher, David Jensen, Brian Neil Levine, *"MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks"*, Proceedings IEEE INFOCOM, April 2006.
[12] Matthew Seligman, Kevin Fall, Padma Mundur, *"Alternative Custodians for Congestion Control in Delay Tolerant Networks"*, SIGCOMM September 2006.
[13] O. Gnawali, M. Polyakovt, P. Bose, R. Govindan, *"Data Centric, Position-Based Routing In Space Networks"*, Aerospace IEEE Conference, 2005.
[14] Padma Mundur, Sookyoung Lee, Matthew Seligman, Seligman, *"Routing for Data Delivery in Dynamic Networks"*, MILCOM 2006.
[15] Sushant Jain, Kevin Fall, Rabin Patra, *"Routing in a Delay Tolerant Network"*, in Proc. ACM SIGCOMM, Aug-2004.
[16] T. Friedman, J. Leguay,V. Conan, *"DTN Routing in a Mobility Pattern Space"*, SIGCOMM05 Workshops, August-2005.
[17] T. Friedman, J. Leguay,V. Conan, *"Evaluating Mobility Pattern Space Routing for DTNs"*, Proc. IEEE Infocom, 2006, Pages: 1-10.
[18] The Network Simulator - ns-2, `http://www.isi.edu/nsnam/ns/`.