www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 3 Issue 5 may, 2014 Page No. 6027-6032*

# Fast Greedy Algorithm for Routing in Delay Tolerant Network

***Aditya Pancholi, Sapna Grover***
aditya.cs.du@gmail.com, sapna.grover5@gmail.com
Assistant Professor
University of Delhi, India

**Abstract:** *Delay-tolerant networking (DTN) is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks which may lack continuous network connectivity. Traditional routing algorithms try to establish a complete route from source to destination and then forward actual data. Due to lack of end to end connectivity, this is not possible in DTN. Also, security guarantees are difficult to establish in a network without persistent connectivity. This paper gives a fast greedy algorithm that intelligently selects next carrier node(s), optimizing the chances of successful delivery.*

## 1. Introduction

The process of deciding how to transmit incoming packets over a network is called routing and if the subnet uses datagrams internally, this decision must be made anew for every arriving data packet. Routing techniques can be broadly classified as Adaptive and Non-Adaptive routing. In Non-Adaptive routing, an administrator manually records the information about nodes' connectivity and embeds a non-changeable route for every destination. On the other hand, in case of Adaptive routing, this information is automatically configured at regular intervals with the help of routing protocols like distance vector routing protocol  and link state routing protocol.

Routing schemes described above are applicable in networks with fixed infrastructure. However, in case of infrastructure less networks such as Mobile Ad-hoc Networks (MANETs) [7], above mentioned routing techniques do not suffice. MANETs lack the presence of a centralized control mechanism. A message is never sent in MANETs until an end-to-end path is established between the sender and receiver. On the other hand, in MANETs, if no end-to-end path exists between the source and destination and there is no guarantee that it will exist ever in future then the packet needs to be delivered on an opportunistic basis. Such a network is called a Delay Tolerant Network (DTN) [8] where a node saves a packet and waits for some other (intermediate) node to come in contact and forward it further. However, in both, MANETs and DTN, the nodes are highly mobile making routing a challenging task.

Since there does not exists any end-to-end path to a destination in DTN, popular routing protocols for Ad-hoc networks such as AODV (Ad hoc on-demand distance vector routing ) [9] and DSR (Dynamic Source Routing) [10] are not applicable. Routing protocols for DTN need to follow store and forward strategy which implies that a packet is stored in the buffer of an intermediate node unless a connection to next hop (or, destination) is discovered.

In this paper, we present a greedy algorithm that addresses the routing issues in a DTN. The algorithm selects the next carrier node(s) in a greedy fashion, optimizing the chances of successful delivery. The algorithm also adapts the dynamic nature of the network.

The rest of the paper is organized as follows: Section 2 describes the related work. Section 3 describes the network assumptions and the model. Section 4 and 5 describes the routing algorithm and conclusion respectively.

## 2. Related Work

In DTN, a message is stored in the buffer of an intermediate node until it reaches its destination. One of the techniques to increase chances of successful delivery of message in such a scenario is replicating many copies of the message, called flooding the message. Vahdat et. al [1] follow such an approach.

Lindgren et. al [2] devised another protocol that utilizes pure non-random nature of mobility of nodes. The protocol attempts to forward messages to nodes with high delivery predictability, which is calculated on the basis of history of nodes' encounters and transitive relation between connection of two nodes. By incorporating transitivity, a node A, not directly connected to node C, will still be able to deliver messages to it via some other node B if it has a direct contact with C.

Musolesi et. al [4] gave Context-aware adaptive routing protocol which tries to maximize the chances of delivery of message by intelligently selecting a node which promises higher delivery probability, almost similar to the way Lindgren et. al have done. However, a carrier node is chosen using a function defined on the context attributes of a node and then applying Kalman filter prediction technique[6] on it.

Dini et. al [5] introduce the concept of reputation of a node, which is a local notion of a node in contrast to the concept mentioned in [4]. The protocol given in [5] is claimed to be an extension of the work of Musolesi et. al and provide security against black-hole attack also.

All the work mentioned above is based on choosing intermediate node by mathematically measuring just the capability of the node to forward a message. However in MaxProp protocol, discussed by Burgess et. al [3], the cost of all the paths to the destination is first calculated and then the smallest one is chosen as a route. Also, the packets are transmitted according to a pre-defined priority.

## 3. Preliminaries

In this section we describe certain terms and concepts that will be used extensively by our algorithm.

### 3.1 Notations

Due to lack of continuous network connectivity in DTN, the routing protocol needs to follow "store and forward" approach for routing. The protocol assumes infinite buffer capacity of each node. As the movement pattern of the nodes in the network is assumed to be completely random; replicating many copies of the packet maximizes the probability of the message being successfully delivered. But at the same time, it might result in flooding of the packets across the network. Thus, our algorithm intelligently replicates a finite number of copies such that the chances of delivery is maximized. Though, acknowledgement messages are still flooded all over the network.

As we need to select outgoing nodes for packet delivery, every node is evaluated on two main parameters; the chances the node will contribute towards successful delivery of the packet and the delay caused by this node due to the lack of contact.

Let the set of nodes in the network be $\Pi$. Each node $v \in \Pi$, estimates how well other nodes behave regarding forwarding its messages. We call this estimate as Delivery Fitness. A node *u* is said to be more fit if it has higher chances of successful delivery. For every node $u \in \Pi$, node *v* in the network keeps a track of the expected time after which *u* will come in contact with it. Nodes with lesser expected time to meet are considered more preferable. This expected time to meet is captured in Contact Function, which is estimated by every node for every other node. For brevity, we formally define Delivery Fitness and Contact Function.

**Definition 1** (Delivery Fitness): For every node $v \in \Pi$ (where $\Pi$ is the set of all the nodes in the network), we define Delivery Fitness *df(v,u)* for every other node *u*. Delivery fitness $df(v,u) \in [0,1]$ defines the probability that the packet will be delivered to the destination if *u* is selected as the next hop.

**Definition 2** (Contact Function): For every node $v \in \Pi$, we define Contact Function *cf(v,u)* for every other node *u*. Contact Function $cf(v,u) \in [0,1]$ defines the probability that *u* will be the next node to come in contact with *v*.

These two parameters will together decide the node to which a packet will be forwarded. Over time, the values of these parameters will change adapting the networks' non-static behavior. Also, every packet maintains a node-list that helps to retrace the path taken by that packet from source to destination.

### 3.2 Node-List

Let us suppose that a sender *s* sends a message to the destination *d*. Each packet carries an extra information in its header called node-list (denoted by *nl*). Initially, this list is empty. Whenever an intermediate node *v* receives this packet, it appends its digital signature in the node-list of that packet. A node whose digital signature is already present in the packet is never selected for further delivery of the packet. This ensures that the packet doesn't get lost in some cycle. Also, this node-list will be embedded in the acknowledgement and sent back once the packet reaches *d*. This will be used to increase the Delivery Fitness values of these nodes.

### 3.3 Delivery Fitness and Aging

A node *v* estimates Delivery Fitness *df(v,u)* for every other node *u* in the network. When the packet reaches the destination successfully, it sends an acknowledgment using standard flooding technique. Upon receiving the acknowledgement, the Delivery Fitness value of all the nodes *u* in the path to the destination is increased. But if the acknowledgement doesn't arrive, it cannot know which node(s) along the path(s) misbehaved. To cope up with such node(s), a mechanism based called aging is deployed. Aging states that a node periodically decreases the Delivery Fitness value of all other nodes. This choice is made to have a conservative policy because we cannot identify the misbehaving nodes.

Whenever a node *v* arrives in the network, it needs to initialize the Delivery Fitness values for every other node. Due to lack of prior knowledge, it initializes *df(v,u) to 0.5* for every node *u*. This estimation can be very vague and hence, a node demands the Delivery Fitness table from every other node that comes in contact with it within some duration called the settling time $\tau$. This exchange is repeated multiple times over a total time-span of $\tau$ and the Fitness table gets stabilized with initial values reflecting network characteristics.

Whenever a positive acknowledgement arrives, node *v* increases the Delivery Fitness of all the nodes who participated in the transmission process. Let, if the *node-list* of a packet looks like *{<s>, <$u_1$>, ... , <$u_t$>, <v>, <$u_{t+1}$>, <$u_{t+2}$>, ... , <d>}* (where <$u_k$> is the digital signature of the node $u_k$); the node *v* increases the Delivery Fitness of its immediate carrier node (i.e. $u_{t+1}$) by a value $\alpha$. For node $u_{t+2}$, it is increased by $\alpha+\delta$ and for $u_{t+3}$, by $\alpha+2\delta$ and so on. These values are then normalized in the interval [0,1]. This approach favors nodes with shorter path to the destination.

Finally, aging is used to reduce the Delivery Fitness of every node at periodic intervals. Aging technique has been studied well in the past and Lindergren et. al [2] describe a very simplistic approach for the same. Let $\gamma \in [0,1]$ be the aging constant and *k* be the number of time units that have elapsed since the last time the metric was aged, aging can be described by the equation (1).

$$df(v,u) = df(v,u) \times \gamma^k. \quad (1)$$

### 3.3 Contact Function

Contact Function takes into consideration that a node with a high Delivery Fitness but very rare to come in contact should not be banked upon for the delivery. Not only the Delivery Fitness, but also the expected time of contact should be considered for finding the next carrier node(s).

Each node $v \in \Pi$ keeps track of probability of meeting peer $u \in \Pi$ . This probability can be estimated as the likelihood that the next node *v* will come in contact of will be *u*. Contact Function *cf(v,u)* is initially set to 1/(|$\Pi$|-1) for all nodes. Whenever a node *u'* is encountered, the value of *cf(v,u')* is incremented by 1 and then all the values of *cf(v,u)* are re-normalized. This technique is often called incremental averaging and is described by Burgess et. al [3]. This is demonstrated with an example given in Fig 1.

Assuming that nodes A, B and C are already existing in the network and a new node D arrives. Due to lack of prior information, it initializes Contact Function value to 1/3 for every node. This can be seen in Fig 1(a). After some time A comes in contact with D and C comes in contact with B. Contact Function values are updated as shown in Fig 1(b). Similarly, Fig 1(c) and Fig 1(d) shows subsequent connections and updation of Contact Functions.

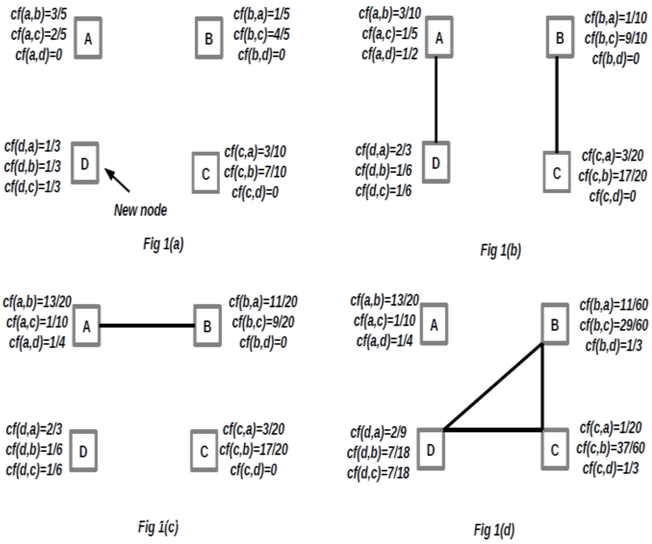

Figure 1: Contact function updation

## 4. Our Algorithm

The main result of this section is a greedy algorithm for routing in a Delay Tolerant Network. The algorithm is based on selective flooding of data packets. The idea is to intelligently select a set of nodes which maximizes the chances of successful delivery of the data packet to the final destination.

The core idea is to identify a set of nodes that will be used for the transmission of the packet. Assuming that we have infinite buffers, a node keep the packets in its buffer till the acknowledgement arrives. Once the acknowledgement arrives, packet is first removed from the buffer and the Delivery Fitness is updated for every node that helped in successful delivery of the packet to the destination, as mentioned in the earlier section. If the packet is not present in the buffer, it indicates that the Delivery Fitness was already updated in the past. As there are multiple copies of the packet, the destination node ensures that it sends acknowledgement for only the first copy of the packet received.

The next carrier node is selected on the basis of Contact Function as well as Delivery Fitness. Whenever a new packet arrives at a node *v*, it adds it's digital signature in the packets' *node-list*. In order to find out the next possible set of carrier nodes, it calculates the *node-fitness* for every other node *u* as given by equation (2).

$$node\text{-}fitness(u)= df(v,u)\times cf(v,u). \qquad (2)$$

Note that, higher *node-fitness(u)* value indicates that at-least one of the Contact Function or Delivery Fitness for that node is high and hence a node with higher *node-fitness* value has higher chances of successful delivery of the packet to the destination.

In order to avoid flooding of data packets, only top *ceiling(ρ%)* nodes according to *node-fitness* are chosen as next carrier node(s) and the packet is forwarded to all of these nodes. By choosing a suitable value of *ρ*, we can control flooding and at the same time increase the chances of successful delivery.

## 5. Conclusion

In this paper, we have presented a greedy algorithm, which forwards messages to intermediate nodes on the basis of its Delivery Fitness and Contact Function. The algorithm intelligently replicates data messages by selecting nodes with high *node-fitness* value and a specified parameter *ρ*.

## References

[1] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected AdHoc Networks, Technical Report, Department of Computer Science, Duke University, 2000.

[2] A. Lindgren, A. Doria, and O. Scheln. Probabilistic Routing in Intermittently Connected Networks. In Proc. Workshop on Service Assurance with Partial and Intermittent Resources, August 2004.

[3] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In Proc. IEEE INFOCOM, April 2006.

[4] M. Musolesi, C. Mascolo, Car: context-aware adaptive routing for delay-tolerant mobile networks, IEEE Transactions on Mobile Computing 8 (2009) 246–260.

[5] G. Dini, A.L. Duca, Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network, http://www.journals.elsevier.com /ad-hoc-networks, March 2012.

[6] R. E. Kalman, “A new approach to linear filtering and prediction problems,” Transactions of the ASME Journal of Basic Engineering, March 1960.

[7] Abdul Shabbir, Anasuri Sunil Kumar (January 2012). "An Efficient Authentication Protocol for Security in MANETs". *IJCCT* 3 (1): 71–74.

[8] Artemios G. Voyiatzis, A Survey of Delay- and Disruption-Tolerant Networking Applications, Journal of Internet Engineering, Vol. 5, No. 1, June 2012, 331-344.

[9] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In The Second IEEE Workshop on Mobile Computing Systems and Applications, February 1999.

[10] D. B. Johnson and D. A. Maltz. Mobile Computing, chapter Dynamic source routing in ad hoc wireless networks, pages 153–181. Kluwer Academic Publishers, February 1996.