

# Design and Development of Cut Node Based Routing Protocol for Delay Tolerant Networks

Guru H. G.  
M.Tech 4th Semester,  
B.I.T, Bangalore.

Jyothi D.G.  
Associate Professor,  
Dept. Computer Science & Eng,  
B.I.T, Bangalore.

Shobha Y.  
Associate Professor,  
Dept. Computer Science & Eng,  
B.I.T, Bangalore.

**Abstract:** Routing in delay tolerant networks (DTNs) is a challenging problem in networking research. Existing DTN routing solutions have used many approaches to increase the success rate of message delivery, such as meeting probabilities between nodes, packet replication and flooding. One important feature of these protocols is using local connection information to find the "best" path with high likelihood to deliver a packet. From a global view, a general disconnected network can have many small instantaneously clustered mobile nodes. Mobility allows nodes carrying messages to deliver them to other clusters. Selecting appropriate nodes to carry and deliver messages becomes important in order to reduce message delay and overhead. The proposed protocol tackles this issue by utilizing cut nodes among a local sub-graph formed by including all neighbors of two "meeting" nodes. Cut nodes are the cut vertices of this local sub-graph, and by definition are the nodes, whose removal will disconnect the graph. Thus, these cut nodes are more likely to be able to deliver messages outside the local cluster. Packets will be buffered in these nodes and forwarded to other cut nodes when they meet. The process repeats until messages reach their destinations. The simulation results show that the proposed algorithm performs better than related protocols in terms of delivery rate and efficiency.

**Keywords**—DTN, MANET, Routing Protocol, Connectivity

## I INTRODUCTION

Mobile network techniques allow users to communicate in situations that were not present in traditional wired Internet. Specifically, networking opportunities extend to scenarios where connections to other nodes or Internet are intermittent. Delay tolerant networking architecture (DTN) has been studied for these scenarios. In addition, a mobile ad hoc network (MANET) can turn to bearing such links when the number of nodes in the network becomes sparse. These connection outages create several disconnected partitions of the connected network. Thus, a hybrid network with DTN and MANET can

be built to assist message delivery. Many applications and projects are built on this hybrid network approach such as ocean sensor networks [1], and vehicular networks [2, 3].

Traditional MANET routing protocols do not apply when the entire network is not fully connected. Instead, many DTN routing protocols are proposed recently. The main

challenge of routing in the hybrid delay tolerant MANET is the uncertainty about network conditions. In the rest of the paper, we will refer to this hybrid network simply as DTN.

Researchers have proposed several approaches on how to find a high probability path to deliver the message with extremely limited local information. Those approaches include estimating the likelihood of nodes meeting by using different mechanisms and packet replication. The probabilistic approach faces the high failure rate of delivering messages since it only keeps one copy of a message along the network, while packet replication suffers the high overhead of storing and forwarding multiple copies of messages. In this paper, we propose a mixture model of routing in the hybrid network.

An observation on the character of the network in question is that it can be separated to several subnets which have high local connectivity. In those subnets, some nodes contact with other subnets intermittently. Based on this observation, we call such nodes as cut nodes in DTN. The cut nodes have higher probability to deliver the message outside the local strongly connected sub-graph. Unlike the computation of cut vertices in graph theory, the cut nodes are locally computed without the knowledge of the entire graph. The cut nodes are discovered when two nodes meet each other and exchange the routing information stored in their routing table. They may not be actual cut vertices because their removal does not necessarily disconnect the entire graph, but instead their removal disconnects the graph in a local sense.

The major contributions of the paper are: (1) We propose a new routing protocol combining forwarding and replication routing. (2) Our protocol requires relatively little overhead from computing and storing information: since each node only computes the connectivity of a local sub-graph and forward packets to the nodes which are identified as "cut nodes". (3) Our protocol provides high message delivery rates: most possible paths from source to destination are covered by this protocol.

## II RELATED WORK

**A. Routing Protocols in DTN:** Routing in DTN has drawn a lot of interest because traditional routing approaches (e.g., routing in MANETs) cannot apply directly. People have recognized that though a path from a source to a destination does not exist at the instantaneous moment, such a path may occur over a time period. However, packets are not guaranteed to be delivered if such a path does not occur. DTN routing protocols have targeted at many application scenarios. Among them, a few have utilized real world contact patterns. Here we briefly discuss routing protocols that are close to our approach.

The simplest DTN routing protocol is epidemic flooding [4]. In this case, packets are replicated to every node which can possibly deliver the packets. Obviously it will waste resources and degrade performance dramatically and cannot be used in practice. Several researchers try to reduce the replicated packets along the network by using historic node meeting information [2, 5-7]. ProPHET [7] is using past node meeting information to compute the probability of meeting a node again. Nodes that are encountered frequently have higher probability to meet again and older contacts degrade over time. Messages are only replicated when the probability exceeds a threshold.

SimBet [8] is similar to our approach in that it also uses neighbor information to construct local graph to calculate graph properties of the encounter events. The authors use the concept of similarity and betweenness to develop SimBetUtil. In SimBet, when two nodes meet, they compute the SimBetUtil scores, and the packet is forwarded to the node with higher score. Here, the social networking feature helps to identify nodes that are more capable in meeting other nodes for message delivery. A potential drawback could occur when stable social relationships (so stable encounter patterns) do not occur in a particular DTN application. That is to say, the analysis of similarity and betweenness is sensitive to the mobility of a network. In SimBet, when the node with the highest score may have the highest probability to contact every nodes, so to deliver the message eventually. It is also possible that it could be hard for a message to be delivered outside a node with high SimBet score. For example, if there exists a node with high SimBet score near the source node and several lower SimBet score nodes close to the destination, it's difficult for the message to be delivered. If replying on the node with higher score to deliver message, SimBet will also require large buffer size.

**B. Related Graph Theory:** DTN can be treated as an undirected graph in which nodes represent the mobile stations and edges represent the communications between mobile stations. In this paper, we make use of the cut vertex concept from graph theory in order to find the nodes utilized to carry and deliver the messages. A cut vertex is defined as: a vertex in the graph such that removal of the vertex disconnects a connected graph. The concept of cut

vertex can be applied to both directed and undirected graphs.

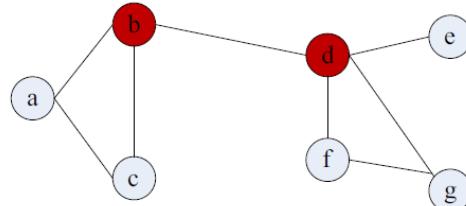


Figure 1. An undirected graph with two cut vertices b and d.

The nodes b and d in Fig. 1 are the cut vertices of an undirected graph. The removal of either node b or d will disconnect the graph. However, knowledge of the whole graph is required to compute cut vertices. In DTN, we cannot rely on the knowledge of the global network. We modify this concept and propose cut nodes which are cut vertices of a local sub-graph.

## III THE PROTOCOL

**A. Cut Node Computation:** As we mentioned earlier, we cannot assume global knowledge of a DTN. Instead, every node in the network maintains the routing information to the nodes which are its neighbors.

Our definition of a cut node: when two nodes meet each other, they will exchange a list of their directly connected neighbors along with any known connections among those neighbors. Thus, they can construct a connected sub-graph  $G'$  ( $V'$ ,  $E'$ ) where  $V'$  is the vertex set which includes the two nodes and all nodes with direct connection to them and  $E'$  is an undirected edge set which represents the known connections among those nodes. We then compute the cut vertices of  $G'$  by using depth first search (DFS). We call a cut vertex found in  $G'$  a cut node of  $G$ . For example, node b and node h will construct a local graph  $G$ , depicted in Fig. 2, when they meet. From Fig. 2 we can see nodes b and h are cut vertices in that local graph and also become cut nodes according to our definition.

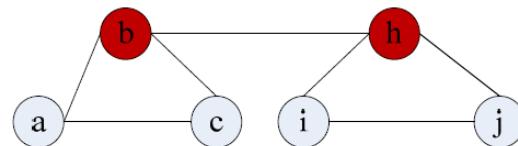


Figure 2. A local graph  $G'$  is constructed when node b and h meet.

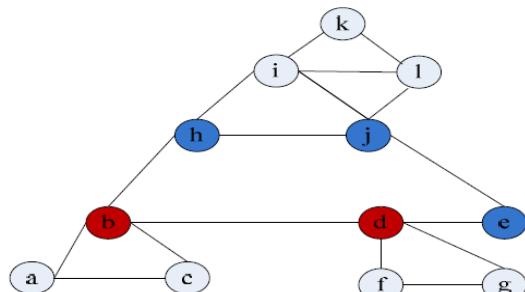


Figure 3. An undirected graph contains cut vertices b and d.

Fig. 3 shows some characteristics of cut nodes. Notice that nodes b and d, colored red are cut vertices of the whole graph and will be cut nodes, while nodes h, j, and e, colored blue, can also be cut nodes of the graph. The cut nodes can play important roles in delivering messages. For example, in Fig. 3, when nodes i, k, or l want to send a message to any of nodes a, c, f, or g, the message must go through one or more cut nodes. It is also important to note that although computing the cut vertices of a graph requires global knowledge of the graph, computing the cut nodes is performed with only a local sub-graph.

Here we also give a concrete example of the disadvantage of the SimBet algorithm where messages can be trapped at the node with the highest betweenness. Suppose in Fig. 3, node g wants to send a message to node k, it will forward the message to d first since node d has the highest betweenness and the similarity score is 0. Once node d has the message, there is no chance that it can further forward the message since both its neighbors b and e have lower SimBet scores. The message will stay at node d forever. However if we use cut nodes to forward the message, then it will be forwarded either to node h or j, and thus the message will move closer to its destination.

**B. The Protocol:** Every node in our protocol has a message vector, neighbor vector and cut vector. The definitions of these vectors are described as follow:

**Message Vector:** This vector stores messages which need to be forwarded. When the vector overflows, the oldest message will be replaced by newest message.

**Neighbor Vector:** This vector stores directly connected neighbors along with their neighbors.

**Cut Vector:** This vector stores all the cut nodes the node has met.

In our protocol, when a node wants to send a message to a destination node, it will check if the destination node is in its neighbor vector, if so it will try to deliver the message to the destination node directly. Otherwise, it will store a copy of the message in its message vector and forward the message to all of its directly connected cut nodes. Those cut nodes will then attempt to deliver the message in the same manner as the source node.

Algorithm 1 describes the communication between nodes m and n when they met each other. When node n receives a hello message from node m, it will look in its neighbor vector to determine if m is a new neighbor. If this is the case, it will also search the destinations of the messages in its message vector, and any message with destination m is delivered. At this point, node n also asks node m for its neighbor vector. Node m will send the neighbor vector back to node n.

**Algorithm 1.** Pseudo-code of node n

upon receive Hello message from node m do

```
if newNeighbor(m) == true
    if messageVector.containDestinationOf (m) == true
        deliverMessages(m)
        requestNeighbor (m)
```

```
upon receive neighbor vector nv from node m do
    updateNeighbor()
    calculateCutVertex(bv)
    propogateCutNodes()
    if isCutNode(n) == true
        || isCutNodeNeighbor (n) ==true
        requestMessage (m)
```

```
upon receive message vector mv from node m do
    updateMessageVector (mv)
    confirmReceiving()
```

Node n will construct a local sub-graph after receiving the neighbor vector. Then it will compute the cut vertices of this local graph. If node n is a cut node, node m will forward all current messages it carries to node n. Node n then becomes a carrier of those messages and vice versa. If neither node is a cut node, then messages will be forwarded from n to any neighbor of m that is not a neighbor of n and is a cut node. If none of the cases exists, then both nodes will keep the message.

#### IV EXPERIMENTAL RESULTS

We have implemented our Cut Node Based Routing Protocol and also the Epidemic flooding protocol using ONE (Opportunistic Network Emulator) simulator. ONE simulator allows users to create scenarios based upon different synthetic movement models and real-world traces and offers a framework for implementing routing and application protocols.

The focus of the simulator is on modeling the behavior of store carry-forward networking, and hence we deliberately refrain from detailed modeling of the lower layer mechanisms such as signal attenuation and congestion of the physical medium. Instead, the radio link is abstracted to a communication range and bit-rate. These are statically configured and typically assumed to remain constant over the simulation. However, the context awareness and dynamic link configuration mechanisms can be used to adjust both range and bitrate depending on the surroundings and the distance between peers.

The performance metric chosen for comparison is Delivery Probability. Delivery probability is defined as the ratio between the number of messages successfully delivered to the destination and the total number of messages generated. Its value ranges between 0 and 1. If all the messages are successfully delivered to their corresponding destinations, then the delivery probability is 1. So, delivery probability is an important metric to compare the routing protocols for Delay Tolerant Networks.

**A. Message delivery statistics:** Fig. 4 shows the comparison between Epidemic and Cut Node based Routing protocols in terms of message delivery probability. The Epidemic Routing Protocol suffers from buffer overflow problem when the buffer size is less. But, our Cut node based Routing algorithm delivers more number of messages than Epidemic Flooding. The message delivery probability increases with increase in buffer size.

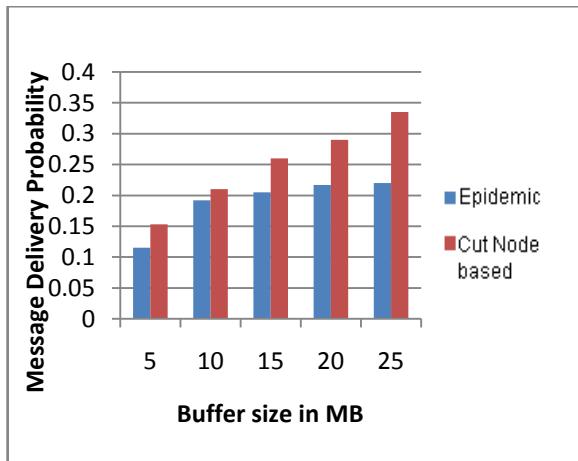


Figure 4. Message delivery statistics

**B. Message Overhead Ratio:** This metric is used to estimate the extra number of packets needed by the routing protocol for actual delivery of the data packets. It is defined as  $(\text{Number of Packets Relayed} - \text{Number of Packets Delivered}) / (\text{Number of Packets Delivered})$ . Epidemic routing creates replicas of messages for each time it forwards messages to the connected neighbors.

So, Epidemic routing puts lot of overhead on the network by the creation of large number of message copies. But, our cut-node based algorithm forwards messages only to the cut nodes. So, the overhead ratio is very less compared to Epidemic algorithm. The statistics are shown in the Fig. 5. We can also observe that the overhead ratio increases with the increase in buffer size as it prevents some of the messages to be dropped due to insufficient buffer space and hence those messages will increase the overhead ratio of the network.

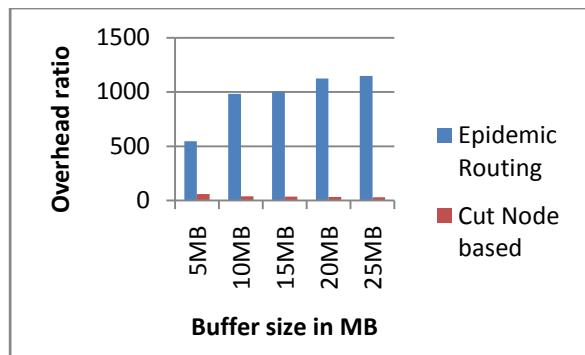


Figure 5. Statistics for Message Overhead Ratio

## V CONCLUSION

In this paper, we propose an algorithm which utilizes graph theory to solve the routing problem in DTN. In our algorithm, when two nodes meet, they will exchange their neighbor information and construct a local sub-graph to compute Cut Vertices. We name those Cut Vertices as Cut Nodes. Those nodes have high probability to deliver messages outside the local cluster and hence also have high probability to forward messages to their destination. We simulate our algorithm using ONE simulator and compare our algorithm to Epidemic flooding. Our simulation results show that Cut Node Based Routing has high delivery rate.

## VI REFERENCES

- [1] J. Partan, J. kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in Proc. ACM WUWNet, 2006, pp. 17-24.
- [2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine "MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks," in Proc. IEEE Infocom, 2006, pp. 1-11.
- [3] J. Ott, D. Kutscher, "A Disconnection-Tolerant Transport for Drive-thru Internet Environments," in Proc. IEEE INFOCOM, Miami, 2005, pp. 1849-1862.
- [4] A. Vahdat, D. Becker "Epidemic routing for partially-connected ad hoc networks," in Proc. Conference Name, Conference Location, 2000, pp.
- [5] J. A. Davis, A. H. Fagg, and B. N. Levine "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in Proc. IEEE ISWC, 2001, pp. 141-148.
- [6] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in Proc. IEEE Infocom, 2005, pp. 398-408.
- [7] A. Lindgren, A. Doria, and O. Schel'en, "Probabilistic routing in intermittently connected networks," Lecture Notes in Computer Science, 2004, vol. 3126, pp. 239-254.
- [8] E. Daly, M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in Proc. MobiHoc 07, Montreal, Quebec, Canada, 2007, pp.32-4