



Distributed Clustering and Routing for Delay Tolerant Mobile Networks

Veda D*

vedadgowda@gmail.com

*Assist Prof, Dept of Information Science
and Engineering, Rajarajeshwari College of
Engineering, Bangalore, India.*

Swetha K H

swetha_kh@rediffmail.com

*Lecturer, Dept of Computer Science and
Engineering, Shridevi Institute of
Technology, Tumkur, India.*

Abstract

Delay-Tolerant Network (DTN) has been introduced to mobile wireless technology because there is a major concern on the intermittent connectivity among the mobile nodes. There are many technologies like Delay Tolerant Sensor Networks, Mobile network and peer to peer mobile ad-hoc [3][4][5] networks. DTN is basically a communication system which will exist temporarily so cannot be used to establish the reliable connections for data delivery. In such networks, we can route based on the node capabilities. So the key idea will be to effectively maintain, update and utilize the nodal contact probabilities. In this architecture every nodes will play the similar roles in the route.

Keywords: Cluster, Mobility, Routing, Contact probability, Synchronization.

1. Introduction

With the increased mobile technologies, wireless communication has been the most widely used communication channel now. Most of the latest architectures and technologies may not work efficiently in the poor conditions where there is a frequent network disruption. The Delay Tolerant Network (DTN) [1][2] is introduced for the mobile network due to the continuous disruption in the mobile communication, usually because of low nodal density and poor signaling. From many studies, clustering has long been analyzed to be the most effective approach to reduce the overhead in the network which will also help scalability. These cannot be directly applied on the DTNs where the node doesn't follow any patterns which are available in the many clustering algorithms. The objective of this study is to analyze the cluster based routing algorithms which can adapted to the Delay Tolerant Mobile Networks (DTMN)[6]. The attempt is made to learn any unknown or possible parameters to group the nodes with same pattern in a cluster in Delay-Tolerant Mobile Networks (DTMN). The nodes under the same cluster can share their resources which will reduce the overhead and also can balance the load. This will realize an efficient and scalable routing in DTMN.

From the various studies, clustering has been considered one of the best approaches to manage the network overhead and also to improve the scalability. At the same time, various clustering algorithms have been analyzed for the mobile networks. However, none of them can be applied directly for the DTNs. All these algorithms are designed well connected mobile networks and share enough data among the nodes. Usually with well-connected networks, the nodes move according to a strict and similar patterns known by these clustering algorithms [7][8]. Since DTNs will have unknown mobility patterns, we cannot adopt these algorithms directly. The community based mobility model is introduced where each node will have their own home location that it visits very often, along with some more frequent locations. So, if there any two nodes from the same location, then the probability of the meeting are higher. With this paper, we investigate on modeling the cluster based routing algorithm for DTMNs.

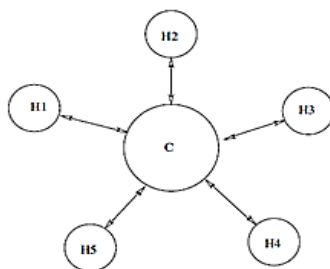


Figure 1: Community Based Clustering

The basic idea with this study is to learn any unknown or any random mobility parameters and to group those nodes into the same cluster. All the nodes in the same cluster can share their resources among themselves to reduce the network overhead and maintain the load balance. This will help to achieve efficient and more scalable routing in DTMN. Grouping the nodes in DTMS will be very unique and non-trivial as the network is not well connected. Due to frequent disruption to the network, the mobile nodes may not have sufficient information and therefore reply differently. Because of this, it is very challenging to gather necessary information from the nodes to group and also to ensure the stability. Although clustering is considered one of the best approaches to improve the network scalability, there is no much study done for the unique networks like DTMN. With this study, we are putting efforts to investigate any clustering problems and routing in such intermittent environments.

2. Distributed Clustering

With the increased mobile technologies, wireless communication has been the most widely used communication channel now. Most of the latest architectures and technologies may not work efficiently in the poor conditions where there is a frequent network disruption. The Delay Tolerant Network (DTN) [1][2] is introduced for the mobile network due to the continuous disruption in the mobile communication, usually because of low nodal density and poor signaling. From many studies, clustering has long been analyzed to be the most effective approach to reduce the overhead in the network which.

For inconsistent connectivity between the mobile terminals, especially under poor signaling and the nodal density, DTN is introduced in the mobile communications. In DTN, the mobile communication links exists only temporarily. This will make impossible to establish the end to end channels for the data pass through. In such networks, the routing is usually based on the probabilities of the nodal contacts for a particular end point or any other parameters based on these probabilities.

A. Challenges and Actions

With continuous changes in the mobility patter among the nodes and also the possible errors in the calculation of the contact probabilities, union and the stability will be major challenges in clustering in DTMN[9]. With this paper, we propose a process for the DTMN, which under goes the following phases.

I. Contact Probability Clustering

With this phase, the patterns are established with the probabilities of direct contacts of the node with the other nodes. We may not store the information of all the nodes but just the probabilities. So based on the probabilities, the node will decided to join or leave the cluster of that member. A threshold on probabilities will be set in order to decide whether the node can join or leave the cluster. When the probability drops below the threshold, the node will be dropped from the cluster.

II. Estimation of Probabilities

In order to estimate the probabilities we will a routing parameter in the network. For this purpose, pair wise contacts can be used for the checking the contact probabilities. One feasible approach will be to store the entire meeting history. But it will be very costly in storage and adapt to changes in mobility patterns. So, we can adopt a much simpler and more effective approach where

each node maintains the list of probabilities for other nodes which it has contacted. The list will be updated in timely fashion.

More specifically, Node i maintains the contact probability ξ_{ij} for every node j . ξ_{ij} is updated in every time slot according to the following rule.

$$\xi_{ij} = \begin{cases} (1 - \alpha)[\xi_{ij}] + \alpha, & i \text{ meets } j \\ (1 - \alpha)[\xi_{ij}], & \text{otherwise,} \end{cases}$$

III. Fractional Clusters

Any errors in the estimation of the probability list or any unusual sequences in the list of meeting, we may end up with many small sized clusters. To handle this scenario we can provide an algorithm so that the node can join the cluster which has better stability.

IV. Inconsistent Membership and GW Selection

We might also arrive in situation where the inconsistency occurs while node selects the gateway. For examples, if there are two nodes in a cluster which has two different gateways, the node might lose the gateway to the next cluster since the node has left. In order to address this, we need to employ a synchronization mechanism between the nodes so that only latest information is available.

V. Cluster Member with Low Contact Probability

In a real time scenario, there might be situation where the mobility pattern changes and the list are still not updated. This will lead to a member with low contact probability to still appear in the contact list of the other node. In order to avoid this condition, a time out can be used for the membership. Once the timeout, the contact info in the list will be updated.

B. Consolidate Meta-Data

In order to consolidate the information based on the contact probability, a mobile node maintains the table as shown in the figure below. It maintains its ID, its cluster ID, cluster table and the gateway table as its local information.

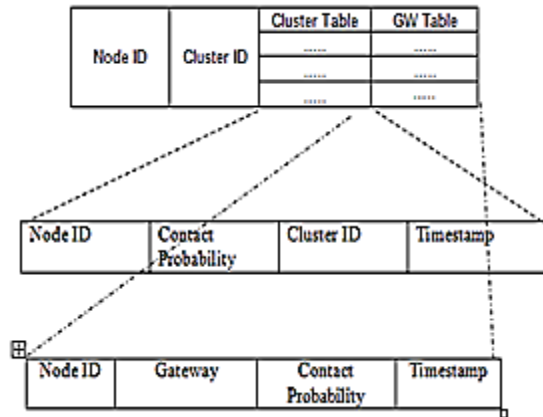


Figure 2: Meta-data

The cluster table in turn will consist of four fields Node ID, Contact Probability, Cluster ID and the Timestamp. Each entry correspond to the node which ever met by the Node A. For example, the entry for Node B contains its contact probability with Node A (denoted by C_{ab}) and its cluster ID(denoted by I_{ab}). Note that both C_{ab} and I_{ab} are updated based on the best knowledge of Node A. Also, generally C_{ab} may not exactly equal to the actual contact probability, although they are expected to be close. The timestamp will have the latest time when Node A and Node B met each other. This field can be used for synchronization.

Also, the node will maintain its gateway details in its Gateway table. This table will consists of four fields namely, Node ID, Gateway, Contact Probability, and the Timestamp. The Cluster ID field will consists of the clusters know by Node A. The Gateway field will consists of the gateway ID of the

cluster, whereas Contact Probability field indicates the highest contact probability between the gateway and any node in the cluster. The Timestamp field will contain the last time when the entry was updated.

C. Distributed Clustering Algorithm

The main part of the algorithm lies on the event where the nodes meet each other. A node will join the cluster when it qualified to be a member and leaves the cluster when it joins some other cluster or it is no longer eligible to be part of that cluster. Whenever the member nodes meet each other, they will undergo synchronization in order to update their information. So we can define three main functions namely, Join, Leave and Sync for the algorithm.

During the initialization, the node creates an entry with its Node ID and two empty tables. The Cluster ID will be set to its Node ID, appended with a sequence number. The sequence number will be incremented by one whenever the node creates a cluster. This will ensure the Cluster ID is unique. The algorithm is purely Event based. Hence, the node can wait for three possible events called, Slot Timeout, Meet A Node and Gateway Outdate.

I. Slot Timeout Event

This event will ensure to update the contact probability. This event will be generated at the end of every time slot, triggering the process of updating the contact probabilities. Once the contact probabilities are updated, Gateway Update proc is invoked to update the gateway table. Since node has updated the contact probabilities, it can choose to better gateways.

II. Meet A Node Event

This event is generated when an HELLO message is received by the node. The HELLO message is exchanged between the nodes when they meet. When this message is received there will be a sequence of actions taken at both sides as explained below.

If node A and B are in the same cluster C, the membership check procedures are invoked to see if they still qualify for the same cluster. If they choose to stay in the same cluster, they sync their cluster information. The synchronization will include two steps, Synchronization of Members and Gateways respectively.

III. Gateway Outdate Event

The timestamp field in the table will be used to refresh the gateway entries in the node. A threshold or timeout is set for the same. Once the timestamp of any entry in the gateway table crosses the threshold, a Gateway Refresh event is generated for that particular entry. If the time difference since last update is more than threshold, it's more likely that the gateway is down or lost. Thus the gateway table entry for the cluster is reset to NIL.

Based on the above methods, the Contact Probabilities, Cluster and Gateways are updated. The outcome will be the more stable clusters that comprise the nodes with similar mobility patterns.

3. Routing Based On Clusters

Once the above clustering procedures are finished, a Node will be associated with a cluster. For any two clusters whose members have enough contact probabilities, a pair of gateways is identified in order to bridge them. In this section of this paper, we will discuss on how the data are routed efficiently in Delay Tolerant Networks [10] by making use of these clusters and gateways.

Consider Node A which intends to send some message to Node B. Node A looks up the cluster table to look for the Cluster ID of the Node B i.e. I_{ab} . According to I_{ab} , we can consider three scenarios: intra-cluster, one hop inter-cluster and multi hop inter-cluster routing. Message forwarding from one node to another will be determined upon meeting events between the nodes.

A. Intra-Cluster Routing

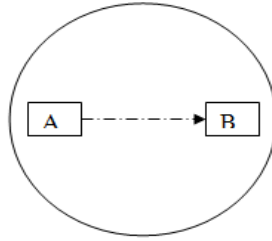


Figure 3: Intra Cluster Routing

If $I_{ab} = I_{(a)}$, then the Node A and Node B are in the same Cluster. Since all the nodes in a cluster will usually have higher contact probabilities, the messages will be directly transmitted. Meaning, the Node A can transmit the data message only when it meets Node B. There will be no relay node is involved in intra-cluster routing.

B. One Hop Inter-Clustering Routing

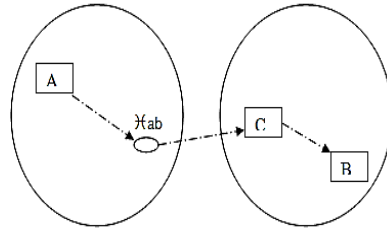


Figure 4: One Hop Inter Cluster Routing

If $I_{ab} \neq I_{(a)}$, then the Node A looks up its gateway table. If an entry for I_{ab} is found, there exists a gateway C_{ab} to Node B's cluster. When the data message is received, the gateway looks up its gateway table to find Node B's Cluster ID. Whenever it meets any node ex . Node C in Node B's cluster, which in turn delivers the message to Node B. through intra-cluster routing as discussed above. Since C_{ab} is the gateway, it has highest probability to meet minimum one node in Node B's cluster. Note that Node C is not necessary to be the gateway. It pairs with C_{ab} for these two clusters.

C. Intra-Cluster Routing

If $I_{ab} \neq I_{(a)}$, then the Node A looks up its gateway table. If Node A doesn't find I_{ab} in its gateway table, the above discussed methods will fail since the destination is not in any cluster that is reachable by its gateway. So, the data transmission needs to be revised for multi-hop routing.

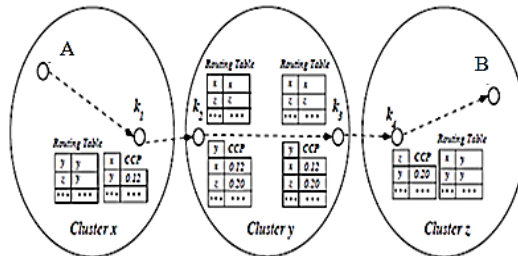


Figure 5: Multi-Hop Inter Cluster Routing

With the low connectivity in DTMNs, on-demand routing algorithms will not work efficiently. This is because the flooding based on-demand routing will lead to high packet loss. On the other hand any table-driven algorithms can be used for multi-hop routing. In the protocol, every gateway builds and shares the Cluster Connectivity Packet (CCP) to other gateways in the network. The CCP will

contain its Cluster ID, and the list of clusters it serves. Such information can be readily read from the gateway table. Fig.5 will show the CCP. Unique sequence numbers will be used in order to avoid any duplication.

Once the gateway accumulates the CCPs, it will construct a network graph. A link will connect if there are gateways between these two clusters. Based on the network graph, the shortest path algorithm is used to find the routing paths.

4. Conclusions

As part of this study, we investigate the distributed cluster-based routing protocols for Delay-Tolerant Mobile Networks (DTMNs). The idea with this approach is to try for any probable similarities of mobile parameters and use them to group the nodes falling under those probabilities. The nodes under the same cluster and dynamically balance the load by sharing their resources. Usually with well-connected networks, the nodes move according to a strict and similar patterns known by these clustering algorithms. Since DTNs will have unknown mobility patterns, we cannot adopt these algorithms directly. The community based mobility model is introduced where each node will have their own home location that it visits very often, along with some more frequent locations. So, if there any two nodes from the same location, then the probability of their meeting are higher.

5. Future Enhancement

This work can be extended to provide the reliable communication channels for the emergency calls. This will ensure the emergency calls will not be impacted due to the low signaling.

6. References

1. K. Fall, "A delay-tolerant network architecture for challenged Internets,," in Proc. ACM SIGCOMM, pp. 27.-34, 2003.
2. S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking.—an approach to interplanetary Internet,," IEEE Commun Mag., vol. 41, no. 6, pp. 128.- 136, 2003.
3. <http://www.princeton.edu/mrm/zebranet.html>.
4. T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad ho networking paradigm (or where there is a whale, there is a way),," in Proc. MobiHOC, pp. 233.-244, 2003.
5. T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay tolerant wireless networks,," in Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking and Related Topics, pp. 260.-267, 2005.
6. Y. Wang and H. Wu, "DFT-MSN: the delay fault tolerant mobile sensor network for pervasive information gathering,," in Proc. 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM.'07), pp. 1235.-1243, 2006.
7. Y. Wang and H. Wu, "Delay/fault-tolerant mobile sensor network (DFTMSN): a new paradigm for pervasive information gathering,," IEE Trans.Mobile Computing, vol. 6, no. 9, pp. 1021.-1034, 2007.
8. Y. Wang, H. Wu, F. Lin, and N.-F. Tzeng, "Cross-layer protocol design and optimization for delay/fault-tolerant mobile sensor networks,," IEEE J. Sel. Areas Commun, vol. 26, no. 5, pp. 809.-819, 2008. (A preliminary version was presented at IEEE ICDCS.'07.)
9. H. Wu, Y. Wang, H. Dang, and F. Lin, "Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks,," IEEE Trans. Wireless Commun., vol. 6, no. 9, pp. 3287.-3296, 2007.