Cluster Based Approach for Overhead Reduction and Load Balancing in Delay Tolerant Mobile Applications

[1] Shankar B Sajjan, [2] Amit Kumar, [3] Masrath Begum, [4] Ambrish Kaneri
[1234] *Assistant Professor, CSE Dept, GNDEC Bidar.*

Abstract

Delay Tolerant Mobile Networks (DTMN) is fundamentally an opportunistic communication system, where communication links only exist temporarily, where it is impossible to establish end-to-end connections for data delivery. In such networks, routing is largely based on nodal contact probabilities (or more sophisticated parameters based on nodal contact probabilities). The key design issue is how to maintain efficiency, update, and utilize such probabilities? This paper deals with an approach of distributively group mobile nodes with similar mobility pattern into a cluster, which can then interchangeably share their resources (such as buffer space) for overhead reduction and load balancing, aiming to achieve efficient and scalable routing in DTMN. Based on nodal contact probabilities, a set of functions including Synchronize (), Leave (), and Join () are devised for cluster formation and gateway selection. Finally, the gateway nodes exchange network information and perform routing.

1. Introduction

Wired and wireless networks have enabled a wide range of devices to be interconnected over vast distances. For example, today it is possible to connect from a cell phone to millions of powerful servers around the world. As successful as these networks have been, they still cannot reach everywhere, and for some applications their cost is prohibitive. The reason for these limitations is that current networking technology relies on a set of fundamental assumptions that are not true in all environments. The first and most important assumption is that an end-to-end connection exists from the source to the destination, possibly via multiple intermediaries. This assumption can be easily violated due to mobility, power saving, or unreliable networks. For example, if a wireless device is out of range of the network (e.g. the nearest cell tower, 802.11 base station, etc.), it cannot use any application that requires network communication. Delay-tolerant networking is an attempt to extend the reach of networks.

As a natural consequence of intermittent connectivity among mobile nodes, especially under low nodal density and/or short radio transmission range, the Delay-Tolerant Network (DTN) technology has been introduced to mobile wireless communications, such as ZebraNet , Shared Wireless Info-Station (SWIM), Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN), and mobile Internet and peer-to-peer mobile ad hoc networks.

Delay and Disruption Tolerant Networking is a new networking paradigm that deals with the establishment of new communication protocols to improve the network communication in case the connectivity is intermittent and/or subject to disruptions. Delay means the end-to-end latency of the data transmission. Disruption refers to factors that are in the origin of connections to break down or even of not being established. Delay Tolerant Networks are networks in which no stable infrastructure exists that can guarantee permanent link connectivity. Most existing DTN protocols are "flat", where every node plays a similar role in routing. The flat architecture is simple and effective in small networks, but not scalable to large size DTNs.

Meanwhile, clustering has long been considered as an effective approach to reduce network overhead and improve scalability. In a clustering scheme the mobile nodes in a network are divided into different virtual groups, and they are allocated geographically adjacent into the same cluster according to some rules with different behaviors for nodes included in a cluster from those excluded from the cluster. Various clustering algorithms have been investigated in the context of mobile ad hoc networks. However, none of them can be applied directly to DTN, because they are designed for well-connected networks and require timely information sharing among nodes.

This paper investigates the distributed clustering and cluster-based routing protocols for Delay-Tolerant Mobile Networks (DTMNs). The basic idea is to autonomously learn unknown and possibly random mobility parameters and to group mobile nodes with similar mobility pattern into the same cluster. The nodes in a cluster can then interchangeably share their resources for overhead

reduction and load balancing, aiming to achieve efficient and scalable routing in DTMN.

2. Literature Survey

Kelvin Fall [1] in paper delay-tolerant network architecture for challenged Internets" propose a network architecture and application interface structured around optionally-reliable asynchronous message forwarding, with limited expectations of end-to-end connectivity and node resources. The architecture operates as an overlay above the transport layers of the networks it interconnects, and provides key services such as in-network data storage and retransmission, interoperable naming, authenticated forwarding and a coarse-grained class of service.

S. Burleigh, A. Hooke, L. Torgerson[2] in their paper "Delay-tolerant networking.—an approach to interplanetary Internet" describe the main structural elements of Delay Tolerant Network architecture, centred on a new end-to-end overlay network protocol called Bundling. They also examine Internet infrastructure adaptations that might yield comparable performance but conclude that the simplicity of the DTN architecture promises easier deployment and extension.

In [3] Lindgren et al. propose a probabilistic routing approach to enable asynchronous communication among intermittently connected clouds of hosts. Their approach is based on the fact that if a node has visited a location several times before, it is likely that it will visit that location again.

T. Small and Z. J. Haas [4] in their paper "Resource and performance trade-offs in delay tolerant wireless network" examined the storage-delay and energy-delay trade-offs in delay-tolerant wireless networks, and they proposed a number of approaches to control the trade-offs. The use of anti packets, small headers that are retained after a packet is offloaded to its destination, assists the network in removing obsolete, already offloaded packets. This reduces the utilized storage in the network, not only without adversely impacting the packets' delays, but in fact causing some reduction in delays.

Y. Wang, H. Wu, F. Lin, and N.-F. Tzeng [5] in paper naming "Protocol Design and Optimization for Delay/Fault-Tolerant Mobile Sensor Networks" studies efficient data delivery in Delay/Fault-Tolerant Mobile Sensor Networks (DFT-MSN's). DFT-MSNis fundamentally an opportunistic network, where the communication links exist only with certain probabilities, and thus are the most crucial resource. Without end-to-end connections, routing in DFT-MSN becomes localized and ties closely to the medium access control, naturally calling for merging layer-3 and layer-2 protocols in order to reduce overhead and improve network efficiency. To this end, authors proposed a cross-layer data delivery protocol, which consists of two phases, i.e., the asynchronous phase and the synchronous phase. In the first phase, the sender contacts its neighbours to identify a set of appropriate receivers. Since no central control exists, the communication in the first phase is contention-based. In the second phase, the sender gains channel control and multicasts its data messages to the receivers. Furthermore, we have identified several optimization issues, with solutions provided to reduce the collision probability, and to balance between link utilization and energy efficiency. Extensive simulations have been carried out for performance evaluation. Their results have demonstrated that the proposed cross-layer data delivery protocol for DFT-MSN achieves a high message delivery ratio with low energy consumption and an acceptable delay.

H. Wu, Y. Wang, H. Dang, and F. Lin [6] in their paper "Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks", they focused on the performance evaluation of the Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN) proposed for pervasive information gathering. DFT-MSN has several unique characteristics such as sensor mobility, loose connectivity, fault tolerability, delay tolerability, and buffer limit. They have established a queuing model for DFT-MSN by using Jackson network theory.

3. Clustering

The process of dividing the network into interconnected substructure is called clustering and the interconnected substructures are called clusters. The cluster head (CH) of each cluster act as a coordinator within the substructure. Each .CH acts as a temporary base station within its zone or cluster. It also communicates with other CHs. The Cluster based routing provides an answer to address nodes heterogeneity, and to limit the amount of routing information that propagates inside the network. The grouping of network nodes into a number of overlapping clusters is the main idea behind clustering. A hierarchical routing is possible by clustering in which paths are recorded between clusters instead of between nodes. It increases the routes lifetime, thus decreasing the amount of routing control overhead. The cluster head coordinates the cluster activities inside the cluster. The ordinary nodes in cluster have direct access only to cluster head and gateways. The nodes that can hear two or more cluster heads are called gateways.

With introduced for the election of cluster heads in mobile networks include the Highest-Degree, the Lowest-Identifier, and Distributed Clustering Algorithm.

1) Highest-Degree (HD) algorithm: It uses location information for cluster formation. It elects the cluster head from the highest degree node in a neighbourhood.

2) The Lowest-Identifier algorithm: The node with the minimum identifier (ID) is elected as a cluster head. This causes battery drainage resulting in short lifetime span of the system.

3) The Distributed Clustering Algorithm: It is a modified Version of the Lowest-Identifier algorithm. Each cluster selects its cluster head from its neighbouring nodes having the lowest ID. In this algorithm every node can determine its cluster and only one cluster, and transmits only one message.

4. Distributed Clustering Algorithm

The key part of the algorithm lies in the meeting event between any pair of nodes. A node then decided its action subsequently, a node will join a new cluster if it is qualifies to be a member. Similarly, a node leaves its current cluster if it joins a new cluster, or it is no longer qualified to be in the current cluster. When two member nodes meet, they trigger the synchronization process to update their information. To this end, we define three main functions namely, join, leave and sync for the algorithm. During initialization, node *i* creates a cluster that consists of itself only and two empty tables. Its cluster ID is set to be its node ID appended with a sequence number, each node maintains its own sequence number, which increases by one whenever the node creates new cluster, to avoid duplication. The algorithm is event-driven.

a. Nodal Contact Probability

The delivery probability indicates the likelihood that r can deliver data message to sink. The delivery probability of a power I, is updated as follows,

$$\xi_i = \begin{cases} (1-\alpha)[\xi_i] + \alpha\xi_k, & Transmission \\ (1-\alpha)[\xi_i], & Timeout, \end{cases}$$

Where is the delivery probability of power *i* before it is updated, is the delivery probability of node *k* (a neighbour of node *i*), and is a constant employed to keep partial memory of historic status.

i. Synchronize

The *Synchronize ()* process is invoked when two cluster members meet and both pass the membership check. It is designed to exchange and synchronize two local tables. The synchronization process is necessary because each node separately learns network parameters, which may differ from nodes to nodes. The Time Stamp field is used for the "better" knowledge of the network to deal with any conflict.

ii. Leave

The node with lower stability must leave the cluster. The stability of a node is defined to be its minimum contact probability with cluster members. It indicates the likelihood that the node will be excluded from the cluster due to low contact probability. The leaving node then empties its gateway table and reset its Cluster ID.

iii. Join

The *Join ()* procedure is employed for a node to join a "better" cluster or to merge two separate clusters. A node will join the other's cluster if, it passes membership check of all current members. Its stability is going to be improved with the new cluster. By joining new cluster, it will copy the gateway table from the other node and update its cluster ID accordingly. Thus the distributed clustering algorithm is used to form a cluster in DTMN.

5. Methodology

The System architecture shown in Figure 1 provides a high-level overview of the functionality and the responsibilities of the system. At its core, the architecture consists of four different modules – Configuration Panel, DTM Simulator, Clustering Engine, and Routing Engine.

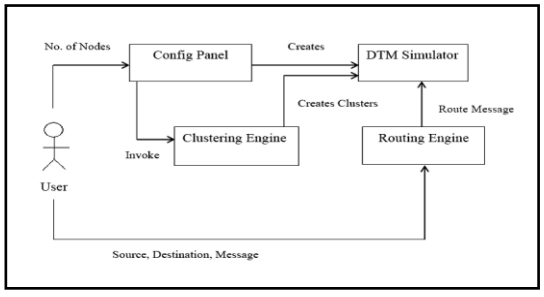

Fig 1 System Architecture

a. Configuration Panel

Through this module, user can configure the simulator with number of nodes. It is the main user interface for the user to interact with the system. User starts the system using this interface.

b. DTM Simulator

Through this module, user can see the node movement at each time instance. Each node is represented by its node-id and cluster-id. The high level design of DTM Simulator is shown in fig 2.

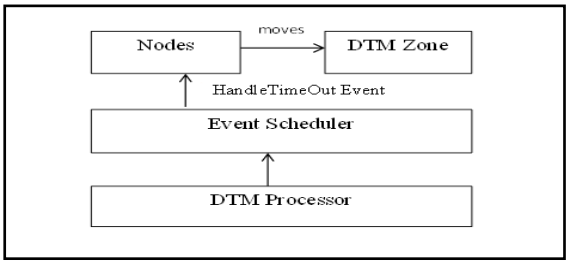

Fig 2 High Level design of DTM Simulator

c. Clustering Engine

Clustering Engine executes the clustering algorithm for DTMN, which undergoes the following steps. First, each node learns direct contact probabilities to other nodes. It is not necessary that a node stores contact information of all other nodes in network. Second, a node decides to join or leave a cluster based on its contact probabilities to other members of that cluster. Since our objective is to group all nodes with high pair-wise contact probabilities together, a node joins a cluster only if it's pair-wise contact probabilities to all existing members are greater than a threshold α. A node leaves the current cluster if its contact probabilities to some cluster members drop below α. Finally once clusters are formed, gateway nodes are identified for inter-cluster communications.

d. Routing Engine

Routing Engine executes Routing algorithm. Once the clustering procedure is finished, each node in the network is associated with a cluster. For any two clusters whose members have high enough contact probability (>=α), pair of gateway nodes are identified to bridge them.

e. Routing mechanisms and algorithms of clustering:

- **Head and Gateway selection**

```
If (Neigh! 0)
        (Leaders==0) then
             ROLE=leader
        Else (there exist atleast one leader)
             If (ROLE=leader) then
                  Solveconflict ();
        Else (my role is not leader)
                  If (leaders ==1)
                  ROLE=member
                  Verify_ consistency ();
        Else (there exists more than one leader)
                  ROLE=gateway;
         ENDIF
    ENDIF
    ENDIF
ELSE
          ROLE = any node _member
ENDIF
```

i. Intra-cluster Routing

If Nodes are in the same cluster since all nodes in a cluster have high contact probability, direct transmission is employed here. In other words, Node transmits the data message only when it meets nodes. No relay node is involved in such intracluster routing.

```
MAKE_WEIGHT_LIST ()
     SORT (WEIGHT_ LIST)
FOR (All nodes contacting with Cluster head)
     IF (CONTACT PROBABILITY OF CLUSTER < THRESHHOLD PROBABILITY OF CLUSTER)
          THEN
    DELETE those nodes from WEIGHT_LIST

FIND_AVERAGE_WEIGHTED_PROBABILY
FOR (all nodes in Weighted List)
   IF (AVERAGE_WEIGHTED_PROBABILITY > CONTACT PROBABILITY OF NODE)
          THEN
DELETE those nodes from WEIGHT_LIST
          REPLICATE ()
Transfer all the Data messages to Node Available in WEIGHT_LIST
```

ii. One-hop Inter-cluster Routing

If Node looks up its gateway table. If an entry for is found, there exists a gateway, Node sends the data message to gateway. Upon receiving the data message, the gateway looks up its gateway table to and Node cluster ID. Whenever, it meets any node, cluster forwards the message, which in turn delivers the data message to Node through intra-cluster routing.

```
IF (TRAFFIC is not reduced after applying INTRA_LOAD_BALANCING)
               THEN
START_INTER_CLUSTER_LOAD_BALANCIN G ()
CHOOSE_ADJUCENT_CLUSTER ()
IF (AVERAGE_WEIGHTED PROBABILITY OF Present cluster < AVERAGE_WEIGHTED PROBABILITY NODE in Adjacent Cluster && NUMBER OF NODE IN Adjacent Cluster > NUMBER OF NODE in Present Cluster)
               THEN
Transfers data to those cluster and
Apply INTRA_CLUSTER_LOAD_BALANCING () in Adjacent Cluster.
```

iii. Multi-hop Inter-cluster Routing

If and Node and gateway table, will fail to deliver the data message, because the destination (Node) is

not in any cluster that is reachable by Node "s gateways. As a result, the data transmission from Node to Node needs to be devised for multi-cluster routing. Given the low connectivity in delay-tolerant mobile net works, on-demand routing protocols do not work effectively here, because the flooding-based on-demand route discovery leads to extremely high packet dropping probability. On the other hand, any table-driven routing algorithms may be employed for multi-hop inter-cluster routing. For simplicity, a link state-like routing scheme is used. In this the protocol, every gateway node builds and distributes a Cluster Connectivity Packet (CCP) to other gateways in the network.

1. The CCP of Gateway comprises its cluster ID, and a list of clusters to which it serves as the gateway and the actual implementation of CCP also includes a sequence number to eliminate outdated information.
2. Once a gateway node accumulates a sufficient set of CCPs, it constructs a network graph. Each vertex in the graph stands for a cluster. A link connects two vertices if there are gateways between these two clusters.
3. The weight of the link is the contact probability of the corresponding gateway nodes. Based on the network graph, a shortest path algorithm is employed to and routing paths and establish the routing table. Each entry in the routing table consists of the ID of a destination cluster and the ID of the next hop cluster, in order to reach the destination If Node " is not a gateway, it doesn't maintain the routing table and thus has no clue about routing.
4. As a result, it asks the first gateway node it meets for routing information. One-hop inter-cluster routing is employed to send the data message to any node. The above procedure repeats until the data messages are delivered to the destination.

Load Balancing:

Load balancing is an effective enhancement to the proposed routing protocol. The basic idea is to share traffic load among cluster members in order to reduce the dropping probability due to queue overflow at some nodes. Sharing traffic inside a cluster is reasonable, because nodes in the same cluster have similar mobility pattern, and thus similar ability to deliver data messages. Whenever the queue length of a node exceeds a threshold, denoted by 1, it starts to perform load balancing.

6. Performance Analysis

The performance metric used in this paper are throughput packet delivery ratio bandwidth end to end delay, energy construction and routing overhead. The measures and details of the various parameters are given below.

A. Packet delivery ratio:

It is defined to be the percentage of the ratio of number of packets received to the number of packets sent.
PDR= Number of packets received / No. Of packets sent X 100%

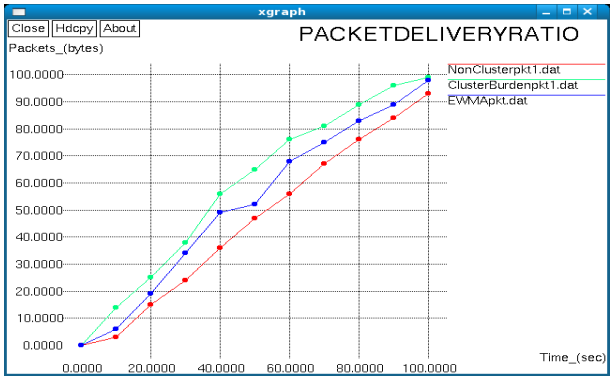

Fig3. Performance comparison using Packet delivery ratio

The X graph shows the variation of Packets (bytes) received based on the time when three different routing schemes are implemented.

B. End-end to delay:

The time interval between the first packet and second packet. Here the total delay takes 1.3 in non-cluster method and 0.9 in EWMA and power balanced communication have 0.4.

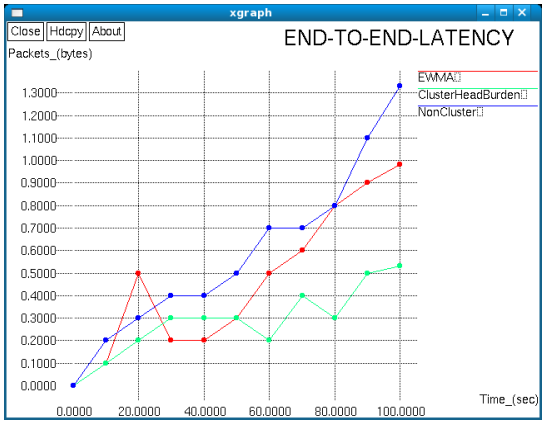

Fig4. Performance comparison using end to end latency

From the X graph shows the proposed power balance communication system achieves low end to end delay.

C. Routing overhead:

Total number of route request and the route reply at the time

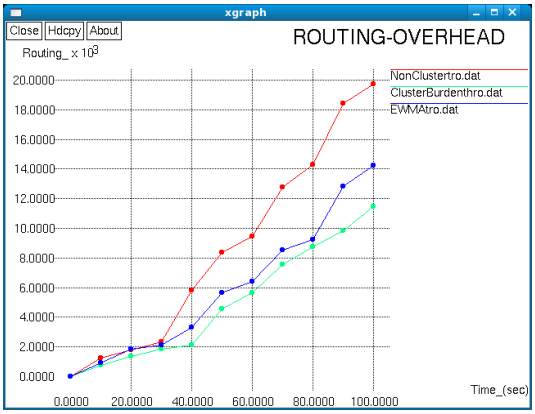

Fig5.Performance using routing overhead

From the X graph shows the routing over head has low in power balanced communication when compare to other existing methods.

D. Throughput:

Throughput is the ratio of number of packets received to the time seconds.

Throughput = Number of packets received / Time (sec)

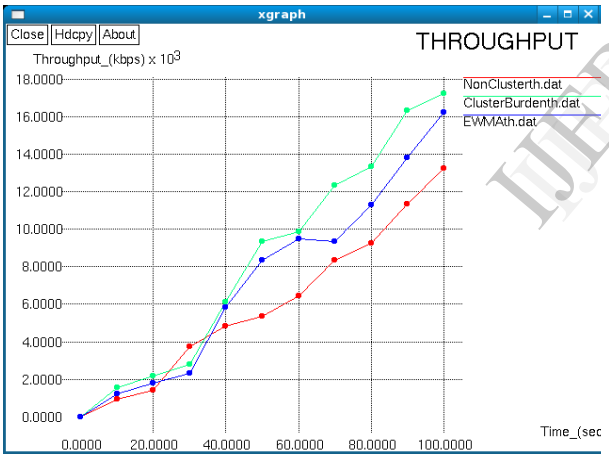

Fig6. Performance comparison using throughput

From the X graph shows the throughput high value to the Power balanced communication when compare to other existing methods.

Advantages:

- Reduces the Network Overhead.
- Load Balancing is achieved.
- Reduction of end-to-end delay takes place.
- Achieves high delivery ratio.

Disadvantages:

- Mobile nodes may have inconsistent information and therefore respond differently.

7. Conclusion

We have investigated clustering and cluster-based routing in DTMN. The basic idea is to let each mobile node to learn unknown and possibly random mobility parameters and join together with other mobile nodes that have similar mobility pattern into a cluster. The nodes in a cluster can then interchangeably share their resources for overhead reduction and load balancing in order to improve overall network performance. Due to the lack of continuous communications among mobile nodes and possible errors in the estimated nodal contact probability, convergence and stability become major challenges in distributed clustering in DTMN. Based on contact probabilities, a set of functions including *Sync ()*, *Leave ()*, and *Join ()* has been devised for cluster formation and gateway selection. Finally, the gateway nodes exchange network information and perform routing. An efficient routing protocol has been provided for the delay tolerant networks through which the stability of the network is maintained. Another important consideration taken into account is load balancing which is implemented using grouping of mobile nodes with similar mobility pattern techniques either retransmission or replication data to their neighbour nodes. Nodal contact probability ratio or threshold is maintained in each group head to achieve better stability and increase scalability among mobile nodes.

8. References

[1] K. Fall, “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] A. Lindgren, A. Doria, and O. Scheln, “Probabilistic routing in intermittently connected networks,” in *Proc. First International Workshop on Service Assurance with Partial and Intermittent Resources*, pp. 239–254, 2004.

[4] T. Small and Z. J. Haas, “The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way),” in *Proc. MobiHOC*, pp. 233–244, 2003.

[5] 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.)

[6] 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.