Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 6, June 2013, pg.398 – 406

RESEARCH ARTICLE

A New Cluster Based Broadcast Protocol for Delay Tolerant Mobile Networks

Mr. Harish Kumar T.R[1], Dr. Dinesh K. Anvekar[2], Mr. K. Sundeep Kumar[3], Mr. Manoj Challa[4]

[1]Computer Science and Engineering, CMR Institute of Technology, India
[2]Computer Science and Engineering, CMR Institute of Technology, India
[3]Computer Science and Engineering, CMR Institute of Technology, India
[4]Computer Science and Engineering, CMR Institute of Technology, India

[1] harishkumartr21@gmail.com; [2] dinesh.anvekar@gmail.com; [3] sundeepkk@yahoo.com; [4] manoj.c@cmrit.ac.in

Abstract— Broadcast makes sensor nodes to evenly distribute messages to the whole network serving higher level operations. Some nodes switch between active and inactive states in Wireless Sensor Networks (WSNs) to reduce the energy consumption and to increase life of network. Broadcast in Low Duty-Cycle Wireless Sensor Networks is a major problem to be addressed. In this paper, the broadcast problem is revisited with active/inactive states. Efficient and scalable distributed implementation for broadcast problem is considered. Similarly Delay-Tolerant Mobile Networks (DTMNs) are considered. In DTMNs, there is lack of continuous end to end connectivity among nodes. A cluster-based broadcast protocol for DTMNs is proposed. Clusters are formed based the contact probability of nodes. An exponentially weighted moving average (EWMA) scheme is employed here for on-line updating nodal contact probability. The simulation results show that proposed protocol achieves higher delivery ratio and significantly lower overhead and end-to-end delay compared to WSNs.

Key Terms: - Broadcast; Reliability; Low Duty-Cycle Networks; Wireless Sensor Networks; Delay Tolerant Mobile Networks

I. INTRODUCTION

Broadcast is one of the most basic fundamental services in wireless sensor networks. It properly distributes messages to entire network serving higher level operations. A control message may be broadcast from the sink node to all other nodes in a network for data collection. Broadcasting is needed for scattering sensed data, also for instructions and code updates. Two important approaches for Broadcast are Flooding and Gossiping [7]. Practically, if every node in a network is active (i.e. *all-node-active assumption*) every node in the network will only need to receive and forward broadcast message at most once. To save energy duty cycle is required and *all-node-active-assumption* is only valid for conventional multi hop networks and wired networks in which saving energy. To save energy, some techniques have to be used for broadcast in WSNs.

© 2013, IJCSMC All Rights Reserved

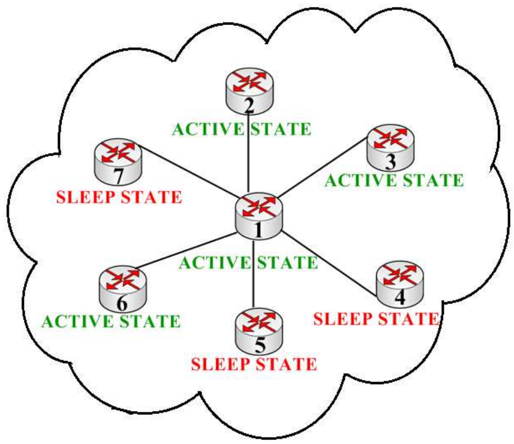

Fig. 1: Active/sleep states in Low Duty-Cycle WSNs

Fig.1 shows a schematic of low duty-cycle WSNs with sleeping (dormant) nodes. Suppose node 1 is required to forward a message. After it forwards a message to its neighbor, the nodes 4, 5 and 7 may not be able to receive the message because they are not active. As the nodes switch to active state, only during a sensing task and after actively performing a sensing task for communication they go for sleep, battery power consumption is minimized. Duty-cycle is defined as the ratio between active period and full total of active and dormant periods. For longer lifetime of operation, low duty-cycle is required but this contradicts *all-node-active assumption*. In low duty-cycle operation it is possible to wait until all nodes in a network become active for broadcast if the number of nodes in a network is very small. However if the number of nodes is very large, synchronization becomes a significant problem.

Delay Tolerant Mobile Networks [1], [2], are formed in an unplanned way, with no proper network connectivity among nodes, and it is difficult to connect end to end. To preserve end–to-end connection in mobile ad hoc networks and WSNs, one has to consider the environmental factors and energy constraints, and this makes network preservation challenging. These are a class of emerging networks that experience frequent and long duration partitions. In such networks, broadcast or routing is largely based on nodal contact probabilities (or more sophisticated parameters based on nodal contact probabilities).

Clustering [3] 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 and formed into groups satisfying some conditions and other nodes are excluded which do not satisfy the conditions. Many clustering algorithms have been investigated in the context of mobile ad hoc networks. But as DTMNs are not well connected networks, the common network algorithms cannot be applied directly as it requires information to be shared among nodes with respect to time.

The scalable and robust implementation for broadcast problem considered in this paper relies on local operations which calculate optimal forwarding sequence through covering-set and receiving-set lists in each node to enhance broadcast in low duty-cycle networks. Here distributed clustering scheme is investigated and a cluster-based broadcast protocol is proposed for DTMNs. To this end, an exponentially weighted moving average (EWMA) scheme is employed for on-line updating of nodal contact probability. Based on the contact probabilities of nodes, a set of functions is executed for cluster formation and gateway selection. Here, the gateway nodes are responsible for exchange of information between clusters.

The rest of the paper is organized as follows. Section II deals with the literature survey and related work done to set a background for the concepts of this paper. Section III explains an existing system and its drawbacks. Section IV explains the proposed protocol and architecture in detail. Section V presents the simulation using JProwler for WSN and DTMNs to evaluate their performance. Section VI gives the conclusion and future scope.

II. BACKGROUND AND RELATED WORK

Among the different proposals for Broadcast in wireless ad hoc networks, flooding and gossiping [7] are the basic proposals for WSNs. An example is the smart gossip [8] which addresses the problem of minimization of forwarding overhead. To know forwarding probability, the algorithm maintains and keeps track of earlier broadcast information and adjusts probabilities to fix the topological properties among the sensor nodes to exact match. Another recent work is on Robust Broadcast Propagation (RBP) [9] which works for reliable broadcast and extends the flooding-based approach. The concept in RBP is every node floods the received broadcast message only once, and then by receiving exact ACKs, it retransmits messages for locality repairs. The node

© 2013, IJCSMC All Rights Reserved

density and topology information collected from earlier rounds of broadcast have an impact on both retransmission and number of retransmissions.

This paper considers scenarios from those in earlier papers and also assumes more realistic scenarios. Earlier approaches suffer from poor performance or fail due to invalidation of the *all-node-active assumption*. Low duty-cycle WSNs have been explored recently, one approach is the probability-based broadcast forwarding (PBBF) [12] which shows how to implement a MAC layer solution for flooding in low duty-cycle WSNs. Here, a systematic way for implementation considering flooding reliability, delay and limited usage of energy is discussed.

In [10], delay tolerant network architecture is proposed. Here an application interface structured around optionally-reliable asynchronous message forwarding with very less resources and low end-to-end connectivity and node resources is presented. Another work [11] describes the main structural elements of DTN architecture. It focuses on a new end-to-end overlay network and proposes a protocol called Bundling. It also examines the internet infrastructure adaptations that might yield comparable performance, and concludes that the simplicity of the DTN architecture promises easier deployment and extension.

A recent work [12] proposes a DTN Hierarchical Routing (DHR) protocol to boost routing scalability. DHR is predicated on a deterministic mobility model, wherever all nodes move as per strict, repetitive patterns, that are noted by the routing and cluster algorithms. It cannot be generalized to such networks with unknown mobility as DTN-based peer-to-peer mobile ad hoc networks.

III. A DISTRIBUTED SCALABLE LOW-DUTY CYCLE WSNS

In this section, a distributed scalable solution, with high quality of broadcast and well resistant to wireless losses is presented.

A. Creating Forwarding Sequence

First thing is to calculate the best forwarding sequence covering nodes with in two hops. To come up with forwarding sequence, consider a node k that has to decide forward message. Here, Covering set or CovSet is defined as the set that consists of 1- and 2-hop neighbors covered by at least one forwarding. A CovSet is freshly created once a brand new broadcast message is received, and is updated once node k forwards a broadcast message or a broadcast message is received or overheard. The node k finds presently active neighbors based on active or inactive states once it broadcasts a message and when it reaches it will add them to its CovSet. In the same way, it adds and verifies the presently active neighbors of sender of the message sender and adds them to its list once a broadcast message is received or overheard.

B. Accommodating Wireless Losses

Sometime the precise coverage will not be achieved even when inserted in to CovSet. This is often because of wireless channels suffer from transmission errors. In case of transmission errors, to obtain actual coverage we introduce a receiving set, or RcvSet. The RcvSet is defined as for each node k within the network, as the set of 1-and 2-hop neighbors that familiar (by node k) having already received the message. Specifically, once a broadcast message is received by node k, it checks whether received message is new and received for first time, and if so, it creates associate degree RcvSet for this message and adds the sender of this message into the set. Later it adds the neighbor into its RcvSet if the node k once more receives or overhears constant message from its neighbor if it is not within the set.

© 2013, IJCSMC All Rights Reserved

C. Distributed solution

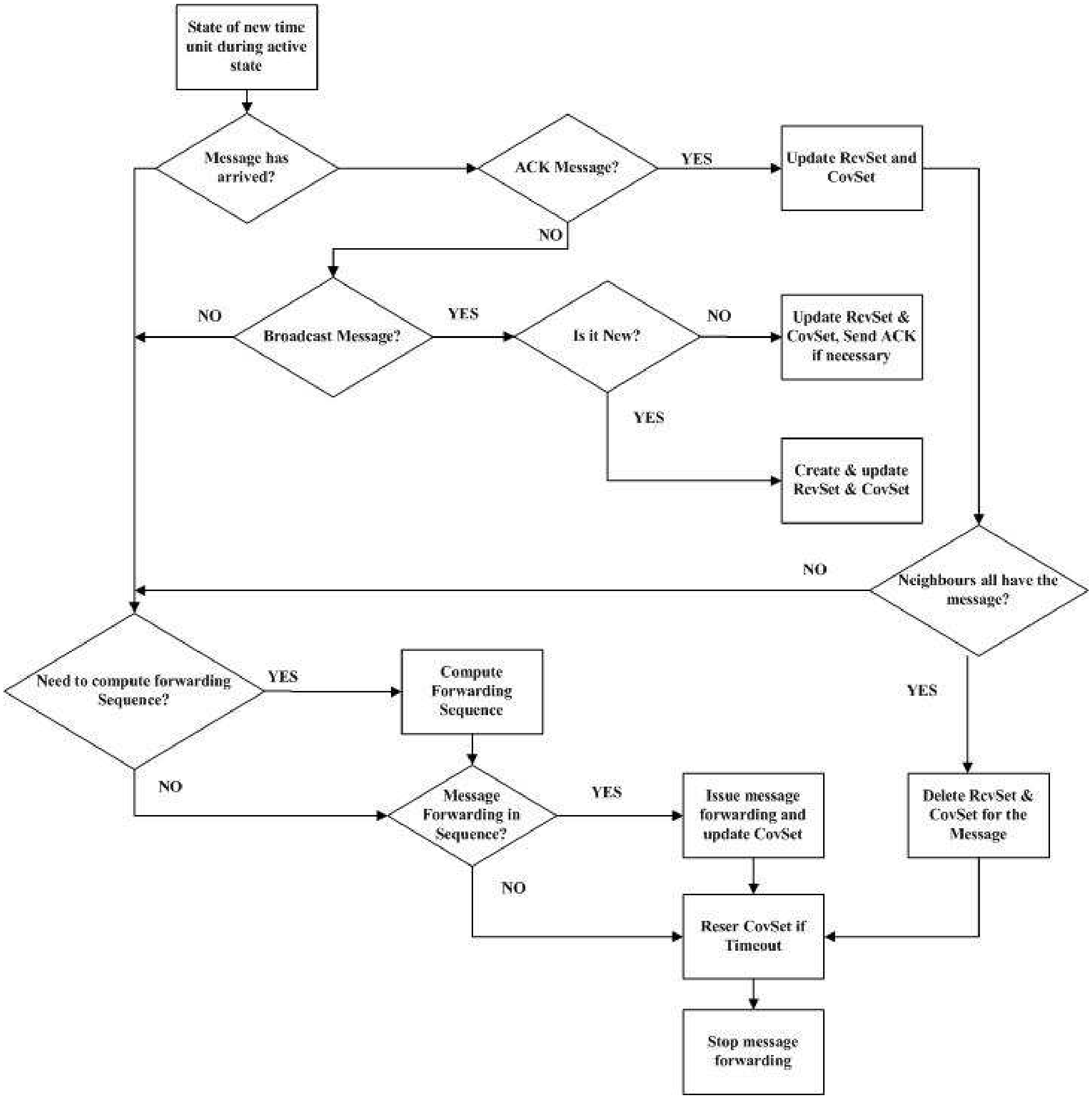

Figure 2: Sequence flow for Distributed solution (when a node is in active state)

Figure 2 shows the main core operations of distributed solution. When node (k) is in active state, it sporadically checks for any message arrival (which may be either received or overheard). Based on the type of the message received the node k can further process the received message. If the received message is new broadcast message, then node k can create RcvSet and CovSet for this message and it will add the sender of this message and therefore the neighbors within the RcvSet piggy backed with this message into its own RcvSet and CovSet. Additionally, node k also adds its neighbors that are presently active and covered by this message into its own CovSet. The first received message could be a broadcast message that has been already received before then node k directly updates corresponding RcvSet and CovSet. It additionally schedules to send associate ACK if the received message is merely targeting on itself. On the other hand, if the received message is an ACK, node k can update its RcvSet and CovSet by adding the sender of this message. After processing the arrived messages, node w then checks its RcvSet whether or not all its neighbors have received the broadcast message. If so, node k will safely stop forwarding the broadcast message and unuse the memory not to store the corresponding RcvSet and CovSet. Otherwise, node w checks whether or not its CovSet change follows its current forwarding sequence. If not, node k will further recompute its new forwarding sequence. Additionally, if there is any message has been scheduled to forward, node k can retransmit the message. Finally, the CovSet is reset to the RcvSet if there is a timeout. This approach will fail in case of node movements; hence this approach is considered as a drawback of the existing system.

© 2013, IJCSMC All Rights Reserved

IV. PROPOSED CLUSTER BASED BROADCAST APPROACH

Here, Clusters are formed by the contact probability of nodes. When a node meets another node its contact probability increases and when it not met its contact probability decreases. A threshold value is given to form the clusters. If a node crosses the threshold value, proposed cluster based broadcast algorithm is executed to decide whether to make new cluster or exit the current cluster or to join other cluster. After the formation of clusters broadcast process is carried out.

A. Clustering

This paper focuses on distributed clustering and cluster-based routing protocols for DTMNs. The fundamental plan is to autonomously learn unknown and presumably random mobility parameters and to cluster mobile nodes with similar mobility pattern into an equivalent cluster. The nodes in every cluster will then interchangeably share their resources for overhead reduction and load equalization, reaching to succeed economical and scalable routing in DTMN. In our protocol, an EWMA scheme is used for on-line change of nodal contact probability, with its mean tested to converge to actual contact probability. Then a collection of functions such as *synchronize( )* ,*Leave( )*, and *join( )* are executed to form clusters and choose gateway nodes having highest contact probability. Finally, the entranceway nodes exchange network info and perform routing. Simulations show the benefit of cluster-based broadcast protocol. The results show that it achieves higher delivery ratio and considerably lowers overhead and also reduces end-to-end delay which is very less than the non-clustering counterpart.

Intermittent end-to-end connections leads to challenges for clustering and routing as detailed below:

On-line Estimation of Contact Probabilities*:* Pair-wise contact probability has been widely used as a routing parameter in expedient networks. However, one amongst the foremost issues in DTN is how to get this parameter by distributive method. A better approach is to safely keep all native contact meeting information (i.e. contact probability) as history, whereas providing strength and robustness is expensive in storage and lacks gracefulness to adapt to changes in node movements. Because of this, a straightforward and effective approach, called exponentially weighted moving average (EWMA) has been adopted. Additionally the node p will keep the information about contact probability βpq for every other node q which is already met by node p. βpq is updated in each time slot, in step with the subsequent rule:

$$\beta pq = \begin{cases} (1-\Delta)[\beta pq] + \Delta, & p\ meets\ q \\ (1-\Delta)[\beta pq], & otherwise \end{cases}$$

where, Δ - constant parameter between 0 and 1, β_{pq} is the old contact probability.

B. Cluster Based Broadcast Algorithm.

The algorithm shown below is the main clustering protocol in DTMN. First it takes number of nodes as input and creates as many numbers of nodes to makes the node move with pattern. A node p, maintains its ID (i.e., p), its cluster ID (denoted by Ω(p)), a cluster table, and a gateway table as its native information. The cluster table consists of 4 fields, namely, Node ID, Contact probability, Cluster ID, and Time Stamp (last meet time). Each entry in the table is for a node entry met by Node p. As an example, the entry for Node k contains its contact probability with node p (denoted by βkp) and its cluster ID (denoted by Ω_p^k). Additionally, the node p inserts and maintains its gate way native information as a table entry. It consists of four Fields Cluster ID, Gateway, Timestamp (last meet time) and Contact Probability. The Cluster ID has a list of nodes known by p. As an example consider the scenario for every cluster c the id of the gateway is inserted as G^c_p and contact Probability as $£^c_p$ which denotes the Highest contact between G^c_p and $£^c_p$. The Timestamp field contains the most recent time in which the entry got updated. For each new time, it moves the node to next position and when it meets nodes the main functions are executed to decide whether to remain in the same cluster *synchronize( )*or to join the other cluster through *join( )* or to leave the cluster through *leave( )* based on contact probability as it crosses or uncrosses the threshold value. Periodically EWMA also calculated for the updating of contact probability of nodes.

Cluster based broadcast algorithm is shown below. The number of nodes is given as the input. Then nodes are created and made to move with pattern. When a node meets another node its contact probability increases and when it not met its contact probability decreases. A threshold value is given to form the clusters. If a node crosses the threshold value, proposed cluster based broadcast algorithm is executed to decide whether to make new cluster or exit the current cluster or to join another cluster. After the formation of clusters broadcast process is carried out.

```
Input  : no. of nodes
Boolean  nodesmet;
Begin  algorithm
Creates node with node id, cluster id, clustertable, gatewaytable;
For(i=0;i<no.of nodes;i++)
    Move node to next position( );
    If(nodesmet == true)
            meetnodeevent( );
            synchronize( );
            join( );
            leave( )
    else
            time out;
end for
EWMA( );
Gatewayupdate( );
End algorithm
```

C. System Architecture

The system architecture shown in Figure 3 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 Broadcast Engine.

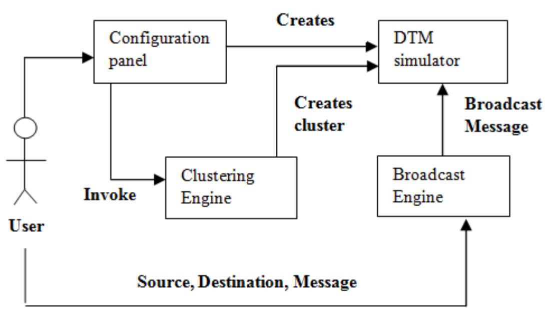

Fig. 3: System Architecture of cluster based broadcast in DTMN

a) Configuration Panel-Through this module, user can configure the simulator with the 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 Figure 4.

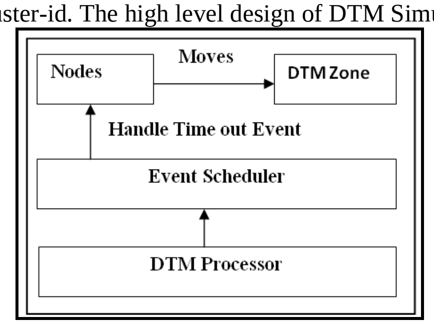

Fig. 4: High Level design of DTM Simulator

c) Clustering Engine – It executes the cluster algorithm for DTMNs that undergoes the subsequent steps.

- First, every node learns about direct contact chances to different nodes. It is not necessary that a node stores contact information of all different nodes in network.
- Second, a node decides to affix or leave a cluster supported by its contact possibilities to different members of that cluster. Since our objective is to cluster all nodes with high pair-wise contact probabilities along, a node joins a cluster given that its pair-wise contact possibilities to any or all

existing members are greater than a threshold γ. A node leaves the present cluster if its contact possibilities to some cluster members drop below γ.

- Finally once clusters are shaped, entryway nodes are known for inter-cluster communications.

d) Broadcast engine and its types in DTMNs

Broadcast Engine executes broadcast 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 ($\geq \gamma$), a pair of gateway nodes is identified to bridge them. If, for example, Node *i*, intends to send a data

message to Node *j, then* Node *i* looks up its cluster table to find the cluster ID of Node j, i.e... Ω_i^j. According to Ω_i^j, three scenarios are considered: intra-cluster broadcasting, one-hop inter-cluster broadcasting, and multi-hop inter-cluster broadcasting.

I. ***Intra-cluster broadcasting***

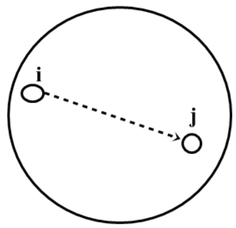

Figure 5.1: Intra-Cluster Broadcasting

In this case, $\Omega_i^j = \Omega(i)$, Nodes *i* and *j* are in the same cluster (see Fig. 5.1). Since all nodes in a cluster have high contact probability, direct transmission is employed. In other words, node *i* transmits the data message only when it meets node *j*. No relay node is involved in such intra-cluster routing.

II. ***One-hop inter-cluster broadcasting***

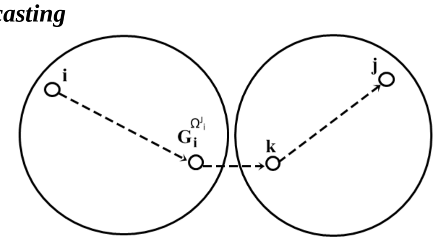

Figure 5.2: One hop Inter-Cluster Broadcasting.

Here, $\Omega_i^j \neq \Omega(i)$,(see Figure 5.2). Node i looks up its gateway table. If an entry for Ω_i^j is found, there exists a gateway (path), i.e. $G_i^{\Omega ij}$, to the cluster of node j. In this case, node i sends data message to its gateway $G_i^{\Omega ij}$. Upon receiving the data message, the gateway looks up its gateway table to find cluster ID of node j. Whenever, it meets any node, e.g., node k, in node j's cluster, it forwards the message to node k, which in turn delivers the data message to Node j through intra-cluster routing as discussed above. Since Node $G_i^{\Omega ij}$ is the gateway, it has high probability to meet at least one node in node j's cluster. Note that, node k in Figure 5.2 need not necessary be the gateway node.

III. Multi-hop Inter-cluster broadcasting

In this case, $\Omega_i^j \neq \Omega(i)$, and Node i doesn't find Ω_i^j in its gateway table. The above two approaches fail to deliver the data message, because the destination (node j) is not in any cluster that is reachable by node i gateways. As a result, the data transmission from node i to node j need to be devised for multi-cluster routing. In this type of routing, whenever a node i meets another node say node k, it checks whether node k is the destination node. If it is not a destination, it forwards the message to node k. When node k meets another node, it checks whether the met node is the destination. If not, same procedure is repeated for a particular period. After the time period, if destination node is not met, then message is deleted from the node.

© 2013, IJCSMC All Rights Reserved

V. SIMULATION RESULTS

Figure 6.1: Data delivery ratio in DTMNs and low duty-cycle WSNs.

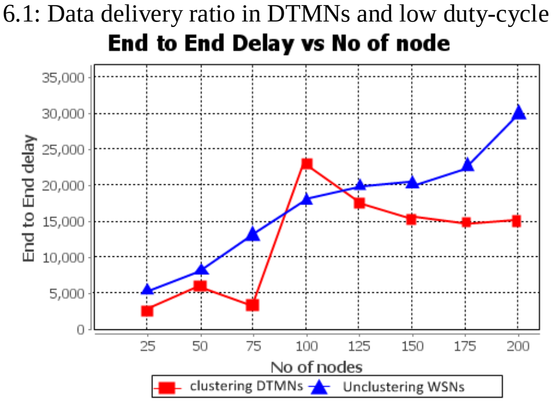

Figure 6.2: End to end Delay in DTMNs and low duty-cycle WSNs.

In this section, we examine the performance of the proposed solution through simulations. The simulation of the proposed scheme is done with the JProwler simulator. The sensing field may be a square of 200 m by 200 m, and also the wireless communication can go up to 10 m. The numbers of nodes within the network vary from 10 to 200. We take into account the data delivery ratio and end-to-end delay as parameters. The simulation graphs shows the performance of clustering based broadcast protocol for delay tolerant mobile networks in comparison with Low duty-cycle WSNs. Figure 6.1 shows performance of packet delivery ratio versus number of nodes in DTMNs, as the number of nodes in network increases that ends up in loss of packets at some cases as seen within the graph, packets drops once the range of nodes reaches 60 and 70. This is because when the number of nodes increases, the time taken to create cluster and communicate with members also increases. But, compared to duty cycle networks (see Fig. 6.1) this protocol works better for data deliver ratio. Additionally, Figure 6.2 shows end-to-end delay versus number of nodes in DTMNs. Time taken to deliver the message increases when the numbers of nodes increases because, when the number of nodes is more the number of clusters are also more. It takes longer to find destination node in other clusters and also possibilities of meeting nodes are less. Once the probability of meeting the nodes is high delay decreases. However to end-to-end delay is reduced when compared to duty cycle networks (see Fig. 6.2).

.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, the broadcast problem has been is revisited with active/inactive states. Here, efficient and scalable distributed implementation for broadcast problem is considered as the drawback, which depends on local information with reliable broadcast. In low duty-cycle WSNs neighbor nodes are not available for broadcast it is similar to the condition lack of network connectivity in DTMNs. Here, end-to-end connection for DTMNs is preserved by forming the clusters of nodes. Cluster based broadcast protocol for DTMN is propped to connect end-to-end nodes and to enhance broadcast. The simulation results show that proposed protocol achieves higher delivery ratio and significantly lower overhead and end-to-end delay compared to low duty-cycle WSNs

Improvement of contact probability based broadcast clustering and addressing the results of proposed protocol is a future direction for future work.

ACKNOWLEDGEMENT

The authors thank the management of CMR Institute of Technology for support for carrying out this work.

REFERENCES

[1] http://searchnetworking.techtarget.com/definition/delay-tolerant-network
[2] Khaled A. Harras, Kevin C. Almeroth and Elizabeth M. Belding-Royer "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks" Khaled Harras et al, University of California 2010.
[3] Balabhaskar Balasundaram, Sergiy Butenko, "Network Clustering" Texas A&M University, 2006
[4] Ha Dang and Hongyi Wu, "Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Networks" IEEE Transactions on Wireless Communications, vol. 9, no. 6, june 2010 .
[5] Feng Wang, Jiangchuan Liu, "On Reliable Broadcast in Low duty-Cycle Wireless Sensor Networks" IEEE Transaction on Mobile Computing, Vol. 11, No. 5, May 2012
[6] Feng Weng and Jiangchuan Liu, " Duty-cycle-Aware Broadcast in Wireless Sensor Networks" NSERC
[7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.
[8] P. Kyasanur, R.R. Choudhury, and I. Gupta, "Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks," Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS), 2006.
[9] F. Stann, J. Heidemann, R. Shroff, and M.Z. Murtaza, "RBP: Robust Broadcast Propagation in Wireless Networks," Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys), 2006.
[10] K. Fall, ."A delay-tolerant network architecture for challenged Internets,."in Proc. ACM SIGCOMM, pp. 27.–34, 2003.
[11] 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.
[12] C. Liu and J. Wu,."Scalable routing in delay tolerant networks". in Proc.v ACM MobiHoc,2010.

© 2013, IJCSMC All Rights Reserved