RESEARCH ARTICLE OPEN ACCESS

Review on Packet Forwarding using AOMDV and LEACH Algorithm for Wireless Networks

Mrs. P. V. Meghare[1], Prof. P. A. Deshmukh[2]
[1]Department of Computer Science, Nagpur University, Nagpur
Email: megharepoonam@gmail.com
[2]Department of Computer Science, Nagpur University, Nagpur
Email: preeti.deshmukh8@gmail.com

ABSTRACT
This paper focuses on reducing the energy consumption and to avoid packet loss of wireless sensor networks. Therefore, a communication protocol LEACH (Low-Energy Adaptive Clustering Hierarchy) is used. LEACH, a clustering-based protocol that utilizes randomized rotation of local cluster base stations (cluster heads) to evenly distribute the energy load among the sensors in the network. The clustering routing technology is the most widely influential. It is a classical clustering routing in wireless sensor networks. A Mobile Ad hoc Network is highly dynamic wireless network that can formed without the need for any pre-existing infrastructure in which each node can act as a router. In this paper we focus on the AOMDV (Ad-hoc 0n-demand Multipath Distance Vector) Routing protocol. AOMDV protocol is an extension to the AODV (Ad-hoc On-demand Distance Vector) Routing protocol for computing multiple loop free and link disjoints paths. AOMDV was designed primarily for highly dynamic ad-hoc networks where link failures and route breaks occur frequently. It incurs more routing overheads and packet delay than AODV but it had a better efficiency when it comes to number of packet dropped and packet delivery. AOMDV reduces routing overhead by reducing the frequency of route discovery operation.
Keywords-AOMDV, LEACH, Mobile Ad-hoc Network, and Multi hop WirelessNetworks.

I. Introduction

Wireless sensor network (WSN) is a self-organized network composed by a large number of micro sensors that are randomly deployed in monitoring regional through wireless communication. Sensors nodes rely on battery power supply, their communication capability and energy storage capacity are very limited, so to utilize the energy of Nodes efficiently, balance the network energy consumption and extend the network lifetime by using the clustering protocol LEACH [2]. LEACH collects the data from the distributed micro sensors and transmits it to a base station. It uses clustering model where nodes elect themselves as cluster-heads, which collects sensors data from other nodes in the vicinity and transfer the aggregated data to the base station [3]. The nodes chosen as the cluster head drain out more energy as compared to the other nodes as it is required to send data to the base station which may be far located. Hence LEACH usesrandom rotation of the nodes required to be the cluster-heads to evenly distribute energy consumption in the network [7].After the clustering process by LEACH algorithm, routing can be done by using the AOMDV routing protocol. AOMDV is based on a prominent and well-studied on-demand single path protocol known as ad hoc on-demand distance vector (AODV). AOMDV extends the AODV protocol to discover multiple paths between the source and the destination in every route discovery. Multiple paths so computed are guaranteed to be loop free and link disjoints [4],[5],[6].

AOMDV also finds routes on-demand using a route discovery procedure. AOMDV relies as much on the routing information already available in the underlying AODV protocol, thereby limiting the overhead incurred in discovering multiple paths. It does not require any special control packets. Extra RREPs and RERRs for multipath discovery and maintenance along with a few extra fields in routing control packets (i.e. RREQs, RREPs and RERRs) constitute the only additional overhead in AOMDV relative to AODV [5].

II. Protocol Overview

2.1. LEACH Protocol

LEACH, a clustering –based protocol that minimizes energy dissipation in sensor networks. The key features of LEACH are:

- Localized coordination and control for cluster set-up and operation

- Randomized rotation of the cluster "base stations" or "cluster head" and the corresponding clusters.
- Local compression to reduce global communication.

2.1.1. LEACH Operation

The operation of LEACH is dividing into two phases. Each of these phases consists of a set-up and a steady –state phase. During the set-up phase cluster heads are determined and the clusters are organized. During the steady-state phase data transfer to the base station occur. The steady phase is longer than the setup phase. This is done in order to minimize the overhead cost.

2.1.1.1 Setup Phase

During the setup phase, a predetermined fraction of nodes, p, choose themselves as cluster-heads. This is done according to a threshold value, T(n) which is depends upon the desired percentage to become a cluster –head-p, the current round r, and the set of nodes that have not become the cluster-head in the last 1/p rounds, which is denoted by G. The formula is as follows:

$$T(n) = \frac{p}{1 - p \times (r \times mod \frac{1}{p})} \forall n$$

$T(n)$: otherwise

Every node wanting to be the cluster-head chooses a value, between 0 and 1. If random number is less than threshold value, T(n), then the node becomes the cluster-head for the current round. Then elected cluster-head floods an advertisement message to the rest of the nodes in the network to invite them to join their clusters. Depending on the strength of the advertisement signal, the non-cluster head nodes decide to join the clusters. The non-cluster head node then informs their respective cluster-heads by sending an acknowledgement message. After receiving the acknowledgement message, depending upon the number of nodes under their cluster and the type of information required by the system, the cluster-heads creates a TDMA schedule and assigns each node a time slot in which it can transmit the sensed data. The TDMA schedule is broadcasted to all the cluster-members. If the size of any cluster becomes too large, the cluster-head may choose another cluster-head for its cluster. The cluster-head chosen for the current round cannot again become the cluster-head until all the other nodes in the network haven't become the cluster-head.

2.1.1.2. Steady phase

During the steady phase, the sensor node starts sensing data and sends it to their cluster-head according to the TDMA schedule. The cluster-head node, after receiving data from all the member nodes, aggregates it and then sends it to the base-station

After a certain time, which is determined a priority, the network again goes into the setup phase and new cluster-head are chosen. Each cluster communicates using different CDMA codes in order to reduce interference from nodes belonging to other clusters.

2.2. Data Transmission

Once the clusters are created and TDMA schedule is fixed, data transmission can begin. Assuming nodes have to send data to their allocated transmission time to the cluster head. Transmission uses a minimal output and not corrupts the transmission of nodes in the cluster. Using CDMA codes, solves the problem of multiple access in a distributed manner[2].

2.3. Attacks

LEACH protocol is difficult to attack as compared to the more conventional multi-hop protocols.

In LEACH, cluster-heads are only nodes that directly communicate with the base station and network irrespective of base station and cluster-heads are periodically and randomly changed and spotting these cluster-heads is very difficult for the adversary.

LEACH is a cluster based protocol, relying on cluster-heads for data aggregation and routing, attacks involving cluster-heads are the most damaging. If any adversary nodes become a cluster-heads, then it can facilitate attacks like Sybil attack, Hello flood attack and selective forwarding. The intruder can broadcast a powerful advertisement to all the nodes in the network and hence, every node is likely to choose the adversary as the cluster-head. The adversary can then selectively forward information to the base station or modify or dump it.

2.4. Security

Key management is an effective method to improve network security, where node interacts with a quite static set of neighbors. But, LEACH is formed dynamically and periodically changes interactions among the nodes and requires that any node needs to be ready to join any cluster-head at any time. So the key distribution schemes are not suited for wireless sensor network, then because of lot of processing, its vulnerability and requires huge memory and since wireless sensor networks consists of sensors with small computation power and negligible memory they are unable to incorporate these security mechanism [3],[7].

III. Ad-hoc On-demand Multipath Distance Vector Routing Protocol

A new class of on-demand routing protocols for mobile ad hoc networks have been developed with the goal of minimizing the routing overhead. AOMDV has three novel aspect compared to other on-demand multipath protocols. First, it does not have high inter-nodal coordination overheads. Second, it ensures disjointness of alternate routes via distributed computation without the use of source routing. Third, AOMDV computes alternate paths with minimal additional overhead over AODV; it does this by exploiting already available alternate path routing information as much a possible [6].

a. Protocol Overview

AOMDV shares several characteristics with AODV. It is based on the distance vector concept and uses hop-by-hop routing approach. The key characteristic of an on-demand protocol is the source initiated route discovery procedure. In AOMDV, RREQ propagation from the source towards the destination establishes multiple reverse paths both at intermediate nodes as well as the destination. Multiple RREPs traverse these reverse paths back to form multiple forward paths to the destination at the source and intermediate nodes. AOMDV also provides intermediate nodes with alternate paths as they are found to be useful in reducing route discovery frequency.

The core of the AOMDV protocol lies in ensuring that multiple paths discovered are loop-free and disjoint and in efficiently finding such paths using a flood based route discovery. AOMDV route update rules, applied locally at each node, play a key role in maintaining loop-freedom and disjointness properties.

i. Loop Freedom

Loop freedom is assured for a node by accepting alternate paths to destination if it has a less hop count than the advertised hop count for that destination. Because the maximum hop count is used, the advertised hop count therefore does not change for the same sequence number. When a route advertisement is received for a destination with a greater sequence number, the next-hop list and the advertised hop count are reinitialized [4],[5],[6]. Two issues arise when computing multiple loop-free paths at a node for a destination. First, one of the multiple paths should a node offer or advertise to other? Since each of these paths may have different hop counts, an arbitrary choice can result in loops. Second, which of the advertised paths should a node accept? Again, accepting all paths naively may cause loops. Based on this, a set of sufficient conditions for loop-freedom. These conditions allow multiple paths to be maintained at a node for a destination[6].

Sufficient Conditions

1. Sequence number rule: Maintains routes only for the highest known destination sequence number. For each destination, we restrict that multiple paths maintained by a node have the same destination sequence number. With this restriction, we can maintain a loop freedom invariant similar to AODV. Once a route advertisement containing a higher destination sequence number is received, all routes corresponding to the older sequence number are discarded.
2. For the same destination sequence number,

a. Route advertisement rule: Never advertise a route shorter than one already advertised.
b. Route acceptance rule: Never accept a route longer than one already advertised.

To maintain multiple paths for the same sequence number, AOMDV uses the notion of an 'advertised hop count'. Every node maintains a variable called advertised hop count for each destination. This variable is set to the length of the 'longest' available path for the destination at the time of first advertisement for a particular destination sequence number. The advertised hop count remains unchanged until the sequence number changes[6].

ii. Disjoint Paths

AOMDV can be used to find node-disjoint or link-disjoint routes. To find node-disjoint routes, each node does not immediately reject duplicate RREQs. Each RREQs arrive in via a different neighbor of the source defines a node –disjoint path. This is because nodes cannot be broadcast duplicate RREQs, so any two RREQs arriving at an intermediate node vie a different neighbor of source could not have traversed the same node. In an attempt to get multiple link-disjoint routes, the destination replies to duplicate RREQs, the destination only replies to RREQs arriving via unique neighbors. After the first hop, the RREPs follow the reverse paths, which are node-disjoint and thus link-disjoint. The trajectories of each RREP may intersect at an intermediate nod, but each takes a different reverse path to the source to ensure link-disjointness [4],[5],[6].

b. Protocol Description

AOMDV protocol describe in four components: routing table, route discovery, route maintenance and data packet forwarding.

i. Routing Table

AOMDV route table entry has a new field for the advertised hop count. Besides a route list is used in AOMDV to store additional on formation for each alternate path including: next hop, last hop, hop count and expiration timeout. Last hop information is useful in checking the disjointness of alternate paths[6].

ii. Rout Discovery

Like AODV, when a traffic source needs a route discovery process by generating a RREQs. Since the RREQs are flooded network-wide, a node may receive several copies of the same RREQ. All duplicates copies are examined in AOMDV for potential alternate reverse path, but reverse paths are formed only using those copies that preserve loop-freedom and disjointness among the resulting set of paths to the source

When an intermediate node obtains a reverse path via a RREQ copy, it checks whether there are one or more valid forward paths to the destination. If so, node generates a RREP and sends it back to the source along the reverse path; the RREP includes a forward path that was not used in any previous RREPs for this route discovery. The intermediate node does not propagate the RREQ further. Otherwise, the node re-broadcasts the RREQ copy if it has not previously forwarded any other copy of this RREQ and this copy resulted in the formation/updation of a reverse path.

When destination receives RREQ copies, it also forms reverse paths in the same way as intermediate nodes. The destination generates a RREP in response to every RREQ copy that arrives via a loop-free path to the source even though it forms reverse paths using only RREQ copies that arrive via loop-free and disjoint alternate paths to the source [6].

iii. Route Maintenance

Route maintenance in AOMDV uses RERR (Route Error) packets. When link breaks it then creates a RERR message, in which it lists each of these lost destinations. The node sends the RERR upstream towards the source node. If there are multiple previous hops that were utilizing this link, the node broadcasts the RERR; otherwise, it is unicast.

When a node receives a RERR, it first checks whether the node that sent the RERR is its next hop to any of the destination listed in the RERR. If the sending node is the next hop to any of this destination, the node invalidates this route table and then propagates the RERR back towards the source. The RERR continues to be forwarded in this manner until it is received by the source. Once the source receives the RERR, it can re-initiate route discovery if it still requires the route [4].

iv. Data Packet Forwarding

For data packet forwarding at a node having multiple paths to a destination, we adopt a simple approach of using a path until it fails and then switch to an alternate path; we use paths in order of their creation.

In other alternative , alternate paths are used simultaneously for load balancing where data packets are distributed over the available paths, thereby improving the network utilization and end-to-end delay [6].

IV. Conclusion

In this paper, we described that AOMDV gives better throughput and end-to-end delay as compared to DSR and AODV. By using the AOMDV implementing the LEACH algorithm where AOMDV extends the single path AODV protocol to compute multiple paths. AOMDV ensures that the set of multiple paths are loop-free and alternate paths at every node are disjoint. The performance of AOMDV using ns-2.We concluded that AOMDV is better than AODV. AOMDV reduces the packet loss. AOMDV also gives better performance with increasing pause time and better PDF value.

By using the LEACH minimizes global energy by distributing the load to all the nodes at different points in time. LEACH is completely distributed, requiring no control information from the base station and the nodes do not require knowledge of the global network in order for LEACH to operate. Distributing the energy among the nodes in the network is effective in reducing energy dissipation from a global perspective and enhancing system lifetime

References

[1] P. Santhiya and J. Bhavithra, *"An Optimal Selective Forwarding Algorithm for Reducing Packet Loss in Mobile Sampling of Sensor Field," Bonfring International Journal of Research in Communication Engineering, Vol. 2, Special Issue 1, Part 3, February 2012.*

[2] Qian Liao and Hao Zhu, *"An Energy Balanced Clustering Algorithm Based on LEACH," Proceeding of the 2nd International Conference on System Engineering and Modeling(ICSEM-13).*

[3] M. J. Handy, M. Haase and D. Timmermann, *"Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection,"*

[4] Manveen Singh Chadha, Rambir Joon and Sandeep, *"Simulation and Comparision of*

AODV, DSR and AOMDV Routing Protocols in MANETs," International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, Issue 3, July 2012.

[5] S. R. Biradar, Koushik Majumder, Subir umar Sarkar and Puttamadappa C,*" Performance Evaluation and Comparision of AODV and AOMDV," International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, No. 02, 2010, 373-377.*

[6] Mahesh K. Marina and Samir Das, *"Ad hoc on-demand multipath distance vector routing"Published in Wiley InterScience 2006.*

[7] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, *"Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000.*