

A Multi Tree based Approach for Performance Analysis in Hierarchical Wireless Sensor Networks

Itu Snigdha¹, Swarn Saurabh²

^{1,2}Dept. of Computer Science & Engg., B.I.T Mesra, India

¹ itusnigdha@bitmesra.ac.in, ² swarn3107@gmail.com

ABSTRACT

A wireless sensor network consisting of heterogeneous sensors is capable of dealing with more than one type of information about any region at a time. Sensor applications that cater to disaster relief require faster data delivery and effective connectivity than the other monitoring applications. Moreover, usually, sensors collect a single type of data. Our approach focuses on generating a tree based communication architecture that is made up of three different types of sensors and ensures that at least one sensor of each type is connected to the communication backbone; thereby forming a MULTI-TREE. By responding to query initiated by the sink for one type of data at a time, the heterogeneous sensors ensure reduced amount of carried load and latency in data transmission. Our strategy implemented for structural health monitoring application shows that when compared to a minimum spanning tree under the same simulation constraints, performs better as it reduces the carried load and delay to one third under the existing set of ad hoc protocols.

KEYWORDS

Wireless sensor network, Minimum Spanning Tree, Multi tree, Carried load, Energy consumption, Delay.

1 INTRODUCTION

The main aim of a wireless sensor network is to be able to deliver the required functionality with unattended operation for the longest possible time without sacrificing the major constraints. Irrespective of the type of application scenario, the main requirement for nodes is to be self-powered. This implies they will either have to contain enough stored energy to last for years, or they will have to be able to scavenge energy from the environment. Till date, a wireless sensor network has been subject to a vast number of application

domains. These vary from Type I applications (eg. real time applications like intrusion detection systems, Alarm or event detection, structural health monitoring etc.) to Type II applications (eg. Continuous monitoring like environmental monitoring) or Type III (eg. hybrid applications like structural health monitoring that requires continuous monitoring as well as immediate alarm system in case of an anomaly or disaster). The basic areas of importance in a WSN can be summarized as follows in figure 1.



Figure 1. Wireless sensor network features

As obvious, the communication quality of a network can vary according to the varied spatial displacements in addition to the associated

connections. This are true to a greater degree for ad-hoc (or infrastructure-less) communication. Once sensors have been deployed, they need to form some kind of network for communication. Since the sensors have limited energy, the network used should be efficient enough to minimize the energy usage.

In a clustered network, many sensors connect to a single sensor, known as cluster head, forming a cluster, which joins to the sink to complete the communication backbone network. All sensors send their data to their cluster head, from where these data are transferred to the sink. The communication between the member and cluster heads and the corresponding inter cluster head communication usually happens in multi tiers. So, this architecture resembles a hierarchy similar to a tree which we analyze in our approach. Network energy consumption is reduced as the number of nodes involved in long distance transmission gets reduced. But in this type of architecture, when cluster head goes out of order, the whole cluster becomes useless rendering reconfiguration. This situation is analogous to a tree, where, to use the sensors of such cluster, tree restructuring can be done for that part of the network. Hence, we advocate the use of a tree-based network for the aforementioned purpose. In a tree based network, we free the node connections from any loop or cycle, which ensures the non redundancy and transmission of data in one direction only. Here, each sensor node is connected to sink directly or via other nodes depending on their distance from the sink and all the sensors can send their data to the sink via single or multi-hops. The issue that arises is to minimize these transmission distances by using different available tree routing algorithm and protocol. Some of the available tree based architecture in wireless sensor networks are: clustered tree network [1], Minimum Steiner tree [2], Minimum spanning tree, Minimum-hot-spot query trees [3], Query routing tree [4]. Minimum-Hot-Spot Query Trees are query routing trees with three properties; decreased collisions during data transmission, decreased query response time, and increased system lifetime and coverage. [4] uses a query routing tree, which is formed in two phases 1) node discovery phase and 2) node balancing phase. It tries to make a near_balanced tree on the

basis of work-load. Similar research as in [1] deals with a Clustered hierarchical tree, in which sensors join to cluster middle head, which further join to cluster head, and finally joins the sink. It uses Simulated Annealing method to optimize the energy usage. [5] uses a Greedy incremental tree, in which first trunk is formed adding the branches latter. It focuses on data aggregation techniques for minimizing energy consumption by using some threshold value. [6] Deals with multi-tree formation in four steps 1) Initialization phase, 2) Tree selection phase 3) Normal phase 4) Recovery phase. This procedure first forms the multi-trees and uses the minimum cost tree for communication. When minimum cost tree fails, next minimum cost tree is used.

The above researches focus either on the energy conservation or on managing the workload, delay or coverage of the network. Moreover they deal with homogeneous sensors, which can be used for only one type of data. If we need to monitor some area for more than one type of data at the same time, we need different sensors. There is no literature available as far as our knowledge that takes into account collection of multiple data types via the same communication backbone. This problem motivates us to search for a method where a single backbone is utilized for collecting different types of data.

Ongoing current research focuses on in-network data processing or data aggregation. There are ample works that advocate the use of tree based structure for achieving the quality of service metrics for wireless sensor networks. In most cases we can infer that use of single paths to connect each node to the base station has been preferred as opposed to using the multiple paths at the same time due to the duplication of the same packet [12]. Moreover, on analysis of the performance of cluster-based and tree-based topologies, it is found that cluster-based topology is more energy efficient for aggregation than tree-based topology; but in the case of acquisition, tree-based topology is more energy efficient than cluster-based topology. Other similar works like Tree-based routing protocols are aimed to construct the best route from a node to base station [13]. Protocols like HTECRP [14] claim to manage congestion and perform fairness on the

network by assigning privileges to the traffic. ViTAMin [15] offers a hierarchical backbone tree algorithm for energy efficiency and sufficient network lifetime. While Localized area spanning tree (LAST) protocols for wireless short range sensor networks optimizes the energy cost and the interference imposed by the structure [16], a BFS based tree rooted at the base station offers shortest path traversal for each data message which utilizes the sensor resources efficiently by employing a local repairing approach for the crashing nodes thereby increasing the lifetime. [17]. CTP is a routing protocol implemented in TinyOS-2. x, offers 90-99.9% packet delivery in highly dynamic environments while sending up to 73% fewer control packets than existing approaches. [18], [19], [20]. Tree based strategies reduce the burden of retransmissions and hence can be used for congestion management. [14]. Thus, it can be believed that a tree structure is very popular as in wireless sensor network structure, in most of applications when we have one sink and too many sender nodes.

To address this problem we discuss a query routing multi-tree approach on a set of heterogeneous sensors. By using different sensors for different purpose we aim to reduce the burden of the nodes in terms of packets generated, forwarded and aggregated by one third. This also affects the overall delay and the final throughput of the network. We implement our algorithm for structural health monitoring applications specifically as it does not require a large number of sensor nodes or a dense deployment scheme and needs three different sensors to aptly monitor the structure. The Requirements of Structural Health Monitoring are enlisted as follows:-

- SHM is based on monitoring buildings and bridges which requires different classes of data to be collected. Our multi-tree approach tries to facilitate this requirement.
- Data redundancy is acceptable.
- Data security is not that important.
- Doesn't require dense deployment.
- Saving energy is not the main constraint as redeployment is possible.

These conditions make it ideal for analyzing the performance of our algorithm.

The remainder of this paper is structured as follows. Section 2 describes the MULTI-TREE algorithm. The results are analysed in section 3 where we compare the performance of the proposed algorithm with respect to a Minimum spanning tree under a given set of protocols. Concluding remarks and directions for future research are provided in Section 4.

2 SYSTEM MODEL

We assume that the initial deployment of the sensors in the region of interest is an undirected graph $G = (V, E)$, where $V = V_A \cup V_S \cup V_T$ and V_A , V_S and V_T are the set of nodes for Accelerometer, Stress/Strain and Temperature type sensors respectively and E is the set of edges connecting these vertices. For the necessary connectivity a sub-graph ' $T = (V_t, E_t)$ ' is constructed which is a spanning tree of ' G ' and $E_T = (u, v)$ be the set of edges such that $E_T \in T$, $u \in V_t$ and $v \in (V - V_t)$ which is selected according to the following condition.

Sink calculates its distance from V_A , V_S and V_T and connects to $\min(E)$ for each set of V_A , V_S and V_T putting the connected vertex in V_t and edge in E_t . Next V_A , V_S and V_T from the set V_t calculates its distance from nodes in the set $(V - V_t)$ and connects to $\min(E)$ for each set of V_A , V_S and V_T from $(V - V_t)$ putting the connected vertex in V_t and edge in E_t . This process repeats $\forall (V_A, V_S \text{ and } V_T) \in V$.

2.1 Assumptions

Once the sensors are connected, they start communicating in order to receive and send data. We assume that the sensor network has the following properties:

- Sensor nodes are heterogeneous (three types of sensors) and location aware; and Sensor nodes have limited and irreplaceable battery power.
- The tree is formed under a constraint that each sensor will connect to only one sensor

of each type within their communication range.

- After tree construction, sink sends query of desired type to receive data from the network.
- Data aggregation and fusion takes place to minimize the total energy usage.
- Sensor nodes are randomly deployed.
- Sensor nodes and sink are stationary.

2.2 Procedure

Step 1: Choose any element 'r' and a set 'S' = {'r'} where S and E = ϕ (Taking 'r' as the root node of the spanning tree.)

Step 2: Find the lightest edge (here the weight is chosen on the basis of the shortest distance and the associated sensor class type) such that one end point is in 'S' and the other point is in $V \setminus S$. Add the edge to 'E' and the other point to 'S'.

Step 3: If $V \setminus S = \phi$ then stop and output the spanning tree (S, E) otherwise go to step 1.

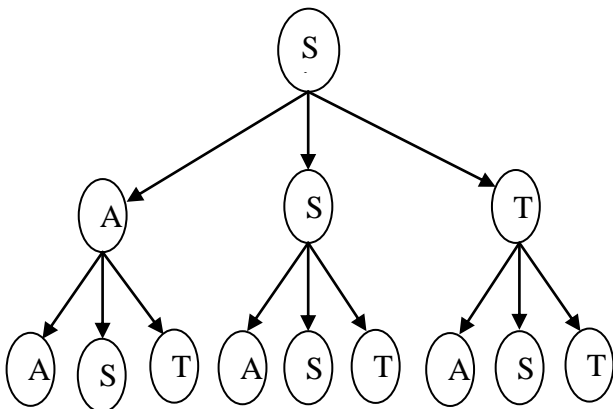


Figure 2. Multi-Tree

A: Accelerometer
 S: Stress/Strain,
 T: Temperature/Pressure

In our architecture, sink links to one sensor of each type, as shown in the figure 2, restricting its children to be a maximum of three. This results in a balanced tree construction, in which none of the nodes are overburdened. When the sink requests a data type, only those sensor nodes that belong to its class respond to the query, this reduces the

number of receptions in case of the parent node and also the cost of data fusion or aggregation . Since the majority of the energy consumption is in transmitting and receiving data, this architecture ensures that none of the sensors will deplete its energy sooner than the other.

Table 1. NOTATIONS

Notations	Description
N	total number of nodes deployed of each type
Node_List[]	Array to store all deployed nodes
Array_T	Array to store Temperature/Pressure sensor nodes
Array_A	Array to store Accelerometer sensor nodes
Array_S	Array to store Stress/Strain sensor nodes
Dist[]	Array to store Euclidean distance between i and j
ED _{i,j}	Euclidean distance between i and j

2.3 Algorithm

/*Tree Generation*/

1. Begin
2. Node_List[0] = Sink;
3. K=1;
4. For(i=0 to N-1)
5. {
6. For(j=i to N-1)
7. {
8. Dist[j] = Calculate ED_{i,j};
9. }
10. Sort Dist[i,n];
11. Connect Node_List[i] and Array_T[i];
12. Node_List[k]=Array_T[i];
13. K++;
14. Repeat STEP 5 to STEP 13 for Array_A[];
15. Repeat STEP 5 to STEP 13 for Array_S[];

- 16. }
- 17. End

For query processing, whenever sink needs to get any type of data, it sends a query to all its children, which is further forwarded to rest of sensor nodes by next level nodes in the tree till it reaches to all the sensor nodes. Sensors after receiving the query responds by transmitting its collected data to the upper level nodes in the network, which is further forward till it reaches sink. All the sensors respond to the query of their type only and discard other queries, which results in maximum of one-third sensors of the entire network to send its data. This reduces the load on the entire network by significant amount, thus increasing the lifetime of the network. As the sink node starts the tree construction, for which it calculates the distances of all the neighboring nodes the complexity is $O(n)$. However, for sorting the distances the complexity is $O(n^2)$. So, the overall complexity of Multi-Tree is $O(n^3)$. The performance has been evaluated for Zigbee application between the nodes in Qualnet simulator as shown in figures 3 and 4. Figure 3 depicts the snapshot of backbone using a multi tree with three different node types (represented by different colors) while figure 4 depicts the MST backbone with homogeneous nodes. The results presented are obtained by Qualnet and stand alone ‘C’ packages as per the following parameters:

Table 2. Simulation Parameters

S.No	Parameters	Values
1.	Radio type	802.15.4
2.	Transmission power	3.0 dbm
3.	Node classes	Stress/strain, pressure and temperature, Accelerometer type
4.	Number of nodes	13 nodes of each type
5.	Packet reception model	PHY 802.15.4 Reception model
6.	Modulation scheme	O-QPSK
7.	CCA Mode	Carrier sense
8.	Noise factor	10.0

9.	Energy model	Linear gradient model
10.	Node Type	MICAZ motes
11.	Items to Send	100
12.	Item Size	127
13.	Application Type	Zigbee

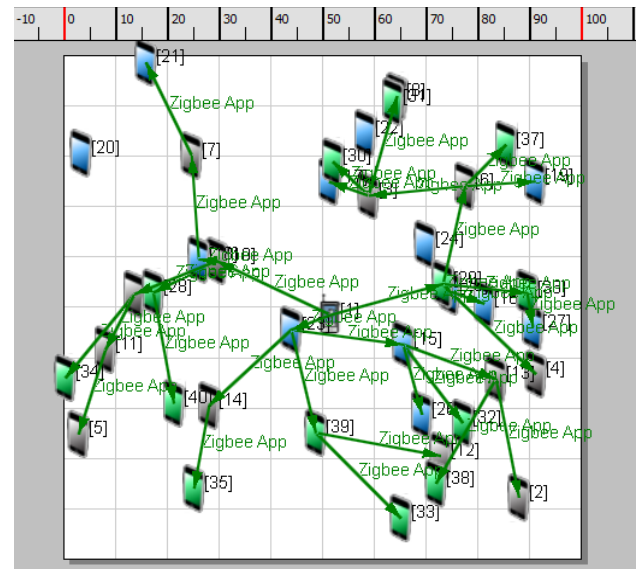


Figure 3. Multi tree using Heterogeneous sensors

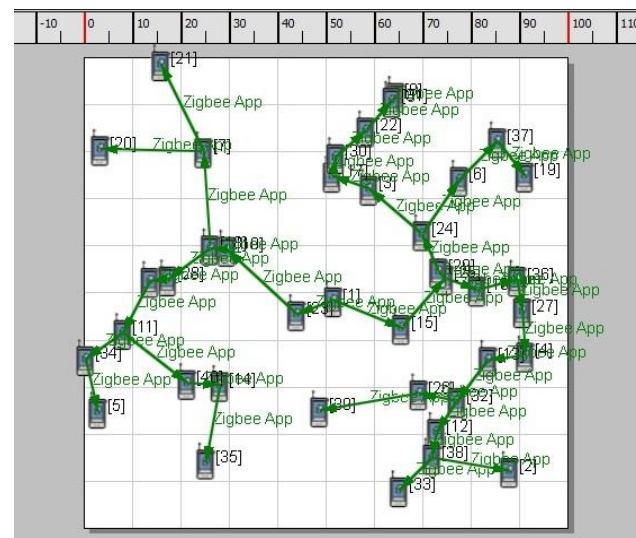


Figure 4. Minimum Spanning Tree

3 RESULTS and DISCUSSION

The multi tree is compared and analyzed with respect to a corresponding minimum spanning tree

for the same set of simulation parameters and for four routing protocols namely; AODV, Bellman Ford, Dymo and Fisheye. The following figures show the throughput of the proposed algorithm, the offered load at the transport layer, the average carried load, the number of packet drops, the average energy consumed in transmitting and receiving, the delay and the jitter. The results are analyzed for the Network and MAC Layer as the Mac Layer protocols that traditionally manage power saving are designed to be application aware to some degree, for example they provide service differentiation for data, query and management packets. The factors considered for validating the performance of our approach over the minimum spanning tree is as follows:

- a) **Throughput at the application layer:** Throughput can be calculated as: Total bytes sent or received*8 / (Simulation time - time first packet is sent or received).
- b) The **Carried load:** Indicates the total workload that an individual node needs to process in terms of data communicated.
- c) The **Average energy** consumed by sensor nodes in transmitting and receiving as any wireless sensor network has to take energy constraints into consideration. Energy consumed is calculated using radio energy model [7]. The energy consumed to send a k -bit message over distance d is:

$$E_s(k, d) = k.E_{elec} + k.\epsilon_{fs}.d^2 \text{ --- } d < d_0 \quad (1)$$

$$E_s(k, d) = k.E_{elec} + k.\epsilon_{mp}.d^4 \text{ --- } d \geq d_0 \quad (2)$$

Where E_{elec} denotes the electronics energy; ϵ_{fs} and ϵ_{mp} are the amplifier energy; $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ is a constant.

To receive a k -bit message, the consumed energy is:

$$E_r(k) = k.E_{elec} \quad (3)$$

To fuse m messages with k -bit, the consumed energy is:

$$E_{fs}(m, k) = m.k.E_{fusion} \quad (4)$$

To gather and transmit a frame of data, the total energy dissipation of a node is:

$$E_n = m.k.E_{elec} + m.k.E_{fusion} + k.E_{elec} + k.\epsilon_{fs}.d^2 \quad (5)$$

d) The **Average delay** at the network layer which is crucial in case of real time applications, the average delay is computed as follows:

Avg. End to end delay = Total transmission delays of all the received packets/ No of packets received
 Where,

The transmission delay of a packet = Total time when a packet is received at the server - Time when a packet is transmitted at the client.

e) **Average Jitter** is calculated as the total packet jitter for all received packets divided by number of packets received minus one where Packet jitter is the Transmission delay of the current packet minus Transmission delay of the previous packet [8, 9, 10, 11].

f) **Packet drops:** The number of packets dropped at the Mac layer is due to link failure or congestion or collisions. The throughput is directly affected by the number of packets dropped in the MAC layer and hence those statistics are also included.

Figure 5 shows the throughput of the entire network, which is better for multi-tree than MST for all the protocols. We know that in case of a tree based communication the chances of contention and collision increases due to the single available path to transmit. Hence we see that in an MST the throughput at the application layer is lesser due to non availability of the channel. However, in case of multi-tree we have different routes for different types of data thereby decreasing the contention for single route to the base station or sink. Figure 6 shows load at transport layer due to unicast, which is less for multi-tree than MST for all the used protocols, indicating that our approach observes less traffic load. We include this graph to confirm that our multi-tree reduces the individual traffic load that needs to be processed by each node in contrast to the total load processed in case of a uniform MST.

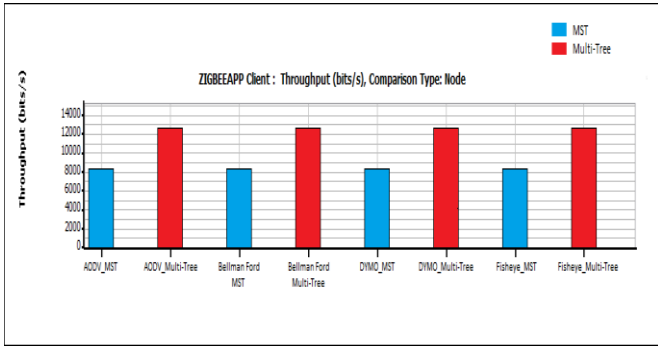


Figure 5. Throughput (bits/second)

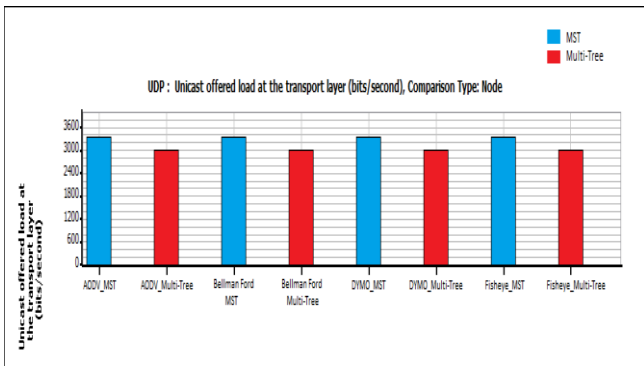


Figure 6. Unicast Offered load at the transport layer (bits/second)

Figure 7 shows that average carried load per node is lesser in case of multi-tree for AODV, DYMO and Fisheye protocols for the reasons already mentioned. However, for Bellman Ford it is more for multi-tree owing to the static nature of protocol while assignment of the routes. This implies that the total energy consumption per packet in case of multi-tree is lesser in comparison to the similar MST backbone.

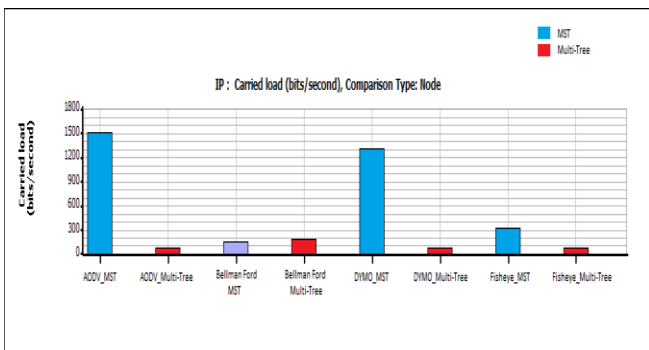


Figure 7. Average Carried load per node (bits/second)

Figure 8 and 9 shows that multi-tree consumes less energy as compared to the corresponding MST in transmission as well as in receiving for AODV, DYMO and Fisheye while for Bellman Ford protocol both tree consumes almost same energy.

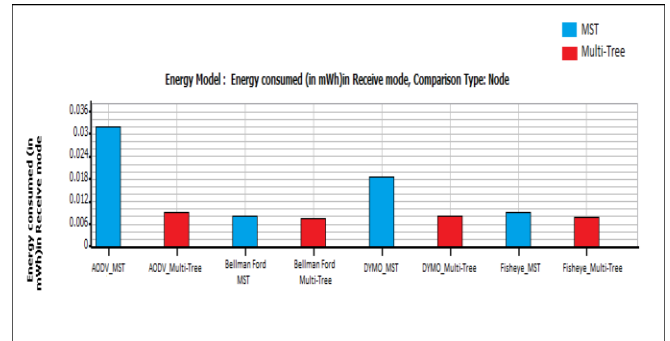


Figure 8. Energy consumed (in mWh) in Receive mode

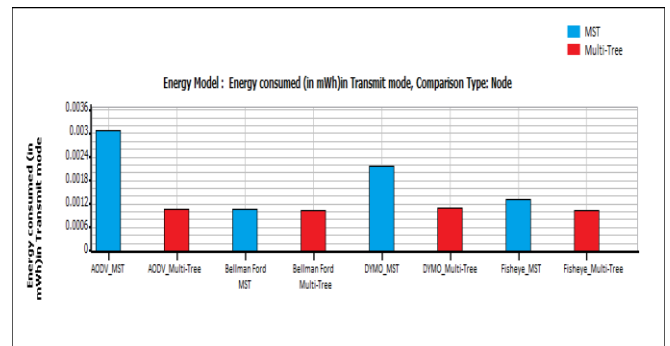


Figure 9. Energy consumed (in mWh) in Transmit mode

Figure 10 and 11 show that average delay and jitter is lesser in case of multi-tree for AODV, DYMO and negligible for Bellman Ford and Fisheye for both the trees. The reduction in delay is noticed due to the fact that multi tree reduces the contention for single routes and hence the packets are sent without prolonged wait times. Since we use the ZIGBEE protocol the packets sent in CAP is more hence a lesser delay and corresponding jitter is observed.

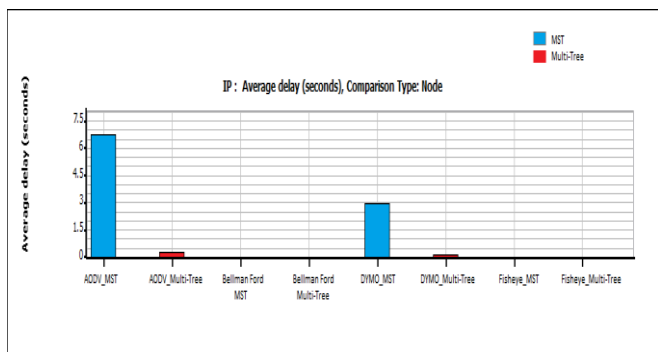


Figure 10. Average Delay (in seconds) per node for the protocols

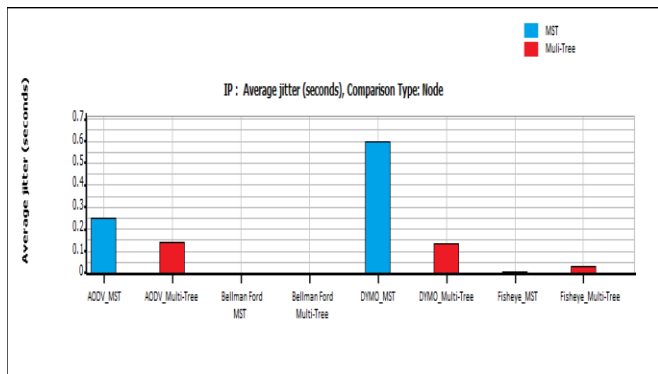


Figure 11. Average Jitter (in seconds) per node for the protocols

Figure 12 shows that the total number of packets dropped at MAC layer are lesser in case of multi-tree for AODV, DYMO and Fisheye and almost same for Bellman Ford for both trees.

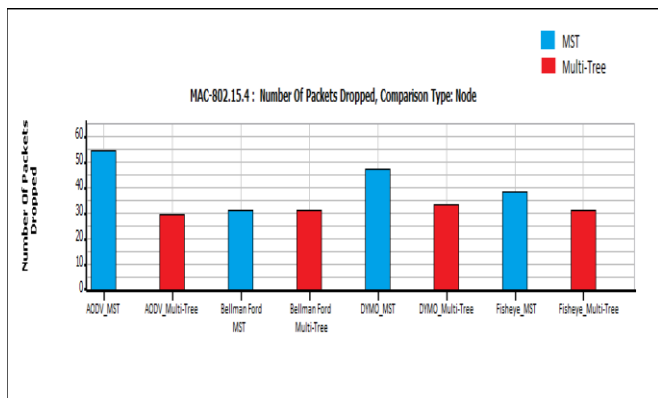


Figure 12. Number of packets dropped at MAC layer

4 CONCLUSIONS

From the simulation results we validate our claim as to the reduction in the carried load which affects the delay and the corresponding energy consumption. The load criteria is not a very important parameter for analyzing the performability of a sensor network as the data communicated in structural health monitoring applications is usually redundant. But we see that since the congestion and collisions are lesser due to reduced traffic, we observe lesser number of packet losses at the MAC layer. The results show the performance of the multi-tree approach in all the layers of the communication namely; throughput at the application layer, total unicast load at network layer in IP UDP, delay and jitter. We also include the results of the MAC layer in terms of the number of packet losses. Also the energy comparison depicts that the multi tree shows significant saving under the same network scenario in comparison to the MST backbone based communication architecture. Hence it can be used for real time sensor network applications that require faster communication. The following results depict the prospects of using multiple sensors for reduced data latency. It can further be addressed to improve the reliability of networks.

7 REFERENCES

1. Hung-Chin, J., Lee, H., Huang, J.: Optimal Energy Consumption for Wireless Sensor Networks, JCIS, (2006).
2. Dengpan, Z., Gao, J.: Maintaining approximate minimum steiner tree and k-center for mobile agents in a sensor network, In INFOCOM Proceedings IEEE, 1-5, (2010).
3. Georgios, C., Zeinalipour-Yazti, D., Gunopulos, D.: Minimum-hot-spot query trees for wireless sensor networks, Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access, 33-40, (2010).
4. Yingchi, M., Xiaofang, L., Yi, L.: Workload-based Query Routing Tree Algorithm in Wireless Sensor Networks, International Conference on Computational Intelligence and Software Engineering (CiSE), 1-4, (2010).
5. Kai, L., Zhao, H., Yin, Z., Luo, D.: An Adaptive Classified Data Aggregation Arithmetic for Wireless Sensor Networks, International Conference on Wireless

- Communications, Networking and Mobile Computing, WiCom., IEEE, 2739-2742, (2007).
6. Fariborzi, H., Moghavvemi, M.: EAMTR: energy aware multi-tree routing for wireless sensor networks, IET communications 3, no. 5, 733-739, (2009).
 7. Bencan, G., Jiang, T.: A tree-based routing protocol in wireless sensor networks, International Conference on Electrical and Control Engineering (ICECE), IEEE, 5729-5732, (2011).
 8. Qiang, T. L., Yu, Q.F.: A Distributed Slot Assignment Algorithm with Minimum Jitter and Delay Guarantee for Real Time Applications for Wireless Sensor Networks, 12th IEEE International Conference on High Performance Computing and Communications (HPCC), 383-390, (2010).
 9. Kan, Y., Gidlund, M., Akerberg, J., Bjorkman, M.: Low Jitter Scheduling for Industrial Wireless Sensor and Actuator Networks, 39th Annual Conference of Industrial Electronics Society, IECON, IEEE, 5594-5599, (2013).
 10. Nghia, L. H., Zalyubovskiy, V., Choo, H.: Delay-minimized Energy-efficient Data Aggregation in Wireless Sensor Networks, International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, 401-407, (2012).
 11. JooHwan, K., Lin, X., Shroff, N. B., Sinha, P.: Minimizing delay and maximizing lifetime for wireless sensor networks with anycast, IEEE/ACM Transactions on Networking (TON) 18, no. 2, 515-528 (2010).
 12. Daniele, M., Ortolani, M., Re, G. L.: A Network Protocol to Enhance Robustness in Tree-Based WSNs Using Data Aggregation, International Conference on Mobile Adhoc and Sensor Systems, (MASS 2007), IEEE, 1-4, (2007).
 13. Zusheng, Z., Yu, F.: Performance analysis of cluster-based and tree-based routing protocols for wireless sensor networks, International Conference on Communications and Mobile Computing (CMC), vol. 1, IEEE, 418-422, (2010).
 14. Mohajerzadeh, A. H., Mohammad, H. Y., Zahra E.: Tree based energy efficient and congestion aware routing protocol for wireless sensor networks, 11th IEEE Singapore International Conference on Communication Systems, (ICCS), 1707-1711, (2008).
 15. Jaekwang, K., Lee, J.: ViTAMin: A Virtual Backbone Tree Algorithm for Minimal energy consumption in wireless sensor network routing, International Conference on Information Networking (ICOIN) IEEE, 144-149, (2012).
 16. Johansson, T., Evgeny O., Lenka C.: Interference aware construction of multi-and convergecast trees in wireless sensor networks, Next Generation Teletraffic and Wired/Wireless Advanced Networking, Springer Berlin Heidelberg, 72-87, (2008).
 17. Chakraborty, S., Nandi, S., Karmakar, S.: A Tree-Based Local Repairing Approach for Increasing Lifetime of Query Driven WSN, 14th International Conference on Computational Science and Engineering (CSE), IEEE, 475-482, (2011).
 18. Patel, D., Chawla, B., Parekh, C.: Energy Aware and Link Quality Based Routing in Wireless Sensor Networks under TinyOS-2. x., International Journal of Modern Engineering Research Vol.3, Issue.3, 1357-1365, (2013).
 19. Colesanti, U., Santini, S., The Collection Tree Protocol for the Castalia wireless sensor networks simulator, ETH Zurich, Zurich, Switzerland (2011).
 20. Gnawali, O., Rodrigo F., Kyle J., David M., Philip L.: Collection tree protocol, Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, 1-14, (2009).