Reduced Latency Timing Model for Wireless Sensor Networks using Data Aggregation

Nishith Parmar[a]*,Raxit Jani[b], Jalay Maru[c],Rohit Singh[d]

[a,b,c,d]*Gandhinagar Institute Of Technology, Gandhinar,Gujarat 382721,India*

Abstract

Data aggregation schemes are widely used in wireless sensor networks (WSNs) to avoid redundant transmission of correlated data from tightly distributed sensor nodes. The life of the network will be longer, but it will be seriously affected by the increased data delivery time. The high end-to-end delay experienced by packets is unacceptable for delay-limited applications such as seismic activity monitoring and military field monitoring. This task proposes a new data aggregation timing model for node aggregation timeouts to reduce data delivery time.

Keywords: Wireless Sensor Networks, Data Aggregation, latency minimization, timing model, TOSSIM.

---

1. Introduction

Recent developments in the field of microelectromechanical systems (MEMS) have realized the dream of creating low-cost, small autonomous devices called wireless sensor nodes that can acquire, process, and transmit field data. Due to the low processing power of the node and limited communication capabilities, sensor nodes must be densely located at the monitoring site to cover the entire area and provide fault tolerance for node failures. Densely distributed nodes collect similar data, and there is a high correlation between these data. Communication costs are a major energy consumer in WSN, so it is not worth transferring similar information through many nodes. Much effort has been made to reduce the number of unwanted transmissions in sensor networks.

Data aggregation technology is gaining more attention in achieving power savings in WSNs. Data aggregation is a technique that combines data from various sensor nodes to eliminate redundant information and provide a rich, multidimensional view of the monitoring environment [1]. Many data aggregation protocols have been proposed in Ref. [24] to reduce power consumption. However, the data aggregation algorithm has the problem of longer data delivery times because the aggregator node has to wait for data from the child nodes. The longer the aggregator waits, the more data it collects from its children, which increases the aggregator's profits. As a result, the gain increases with increasing delay and vice versa. Therefore, there is a trade-off between energy and delay [5].

Wireless sensor networks are primarily used to send data wirelessly from the real world, so it is not worth sending information to the base station after the fact. There are many applications that require time-sensitive data delivery, and the challenge is developing data aggregation methods that guarantee delay requirements. If the aggregator node waits for a long time for data from all its children, it will take longer to deliver the data to the sink, causing the data in the current round to interfere with the data in the next round. The timing model defines how long the aggregator node waits for child data. The aggregation timeout should optimize the data aggregation so that the wait time is optimal for the data to be delivered to the sink within the specified time limit. Some of the works to reduce the delay while aggregating data are considered here.

The rest of the paper is organized as follows. In Section 2 the related work in this domain is outlined, in the Section 3 the proposed algorithm is explained, the protocol implementation is given in the section 4, in the Section 5, the performance of the algorithm is presented and the Section 6 gives the final conclusion of the work.

2. Related Work

The aggregation timeout is simply periodic, whether all nodes wait for a certain amount of time, hop by hop, the aggregator waits for a response from all children, or a cascade timeout, the timeout is It depends on the position in the data. Aggregation tree (DAT). Cascade timeout [6] allows the node to schedule a timeout based on its position in the DAT. The node times out after the child times out, allowing the node to collect information from all children. However, all nodes at a particular level are reported with the same timeout, regardless of the number of children. The advantage of this algorithm is that it does not require time

---

* Nishith Parmar
E-mail address: nishith.parmar@git.org.in

synchronization or centralized control. Because the number of children in each tree is not taken into account, nodes with many children lose some of their child data, and nodes at the same level send data at about the same time, causing traffic congestion. In [7], the node timeout is dynamically determined based on the aggregate tree structure and the number of its children. If a node detects a missed deadline, it can increase the node's aggregation timeout. The update process is more complex and the sink's one-hop neighbor, the agent node, is more crowded because the data arrives at the agent node at about the same time.

Adaptive Time Control (ATC) [8] determines the node aggregation timeout based on the level of the sensor node in the data aggregation tree and the number of its children. Nodes with more children get more time, maximizing the opportunity to aggregate data from the children. Therefore, nodes at the same level get different aggregation timeouts. The authors argued that this algorithm offers higher data transfer rates and lower energy costs compared to cascade timeouts. IEEE modified in their simulation.

The 802.11 protocol is used as the media access protocol and does not take into account the node's sleep schedule. This affects energy consumption and delay calculations.[9] considers time-efficient data aggregation in clustered WSNs. The timeout is calculated for each subtree in the cluster based on packet transmission delays and cascading delays. Performance is compared for various modulation techniques that are a key component of packet transmission delay.

The above protocol is simulated by either the network simulator NS2 or a discrete simulator based on C ++. It also uses IEEE 802.11 as the channel access mechanism, with or without some changes. The performance of these protocols may be tested for the real time deployment of wireless sensor nodes. The wireless sensor nodes of our consideration are IRIS motes from Crossbow Technologies [10]. It is good practice to test the performance of any protocol using a simulator before it is implemented in the real world. The simulators like ns2 and other similar simulators do not reflect the real-world scenario properly. Therefore, we have chosen the WSN simulator TOSSIM to test the behavior of our proposed timing model. The TOSSIM simulator can simulate programs written in NesC, the native language of WSN-Mote IRIS, and can merge code into motes with minor changes. [11]

3. Timing Model

The goal of this work is to foster a convention, which conveys the information to the sink with in the cutoff time while adjusting the information collection strategies for diminishing the energy utilization. This convention assesses the break of every hub in the tree in a disseminated way, with the goal that the information created by every hub ought to be conveyed to the sink in short order. This work focuses on the objective stage as the remote sensor bit called IRIS. These bits use TinyOS working framework, which is one of the most broadly adjusted working frameworks for the asset compelled bit organization. The bits send the information to the sink utilizing the assortment tree, which is shaped and kept up with by the Collection Tree Protocol (CTP) [12]. The CTP convention involves the remote connection quality between the bits as the measurement to build the tree and it is a powerful tree. Assuming that the connection quality changes, the tree construction will likewise change. Thus, the collection break ought to be dynamic and refreshed as the tree structure changes.

In falling break, when an aggregator hub gets the solicitation from the sink, it works out its break in light of the level in the tree. The stunned break happens between the levels and time disarticulation between them is only a solitary jump delay. The hubs in a similar level are having same break and henceforth they attempt to get sufficiently close to the channel all the while. In this way, the transmission of the parcels by the hubs in a similar level will be conceded by the MAC and the parent break will happens before the youngster. Consequently, the aggregator hub will miss a portion of the bundles from its kids hubs and the accumulation gain diminishes. To work on the exhibition of the DAT, the aggregator's hub break should be relegated so that it could gather more data productively.

In our proposed calculation, every hub ascertains the underlying opportunity for information accumulation, which depends on the jump distance from the sink and the quantity of youngsters it has in its sub tree established from it. This underlying break will be different for every hub and it increments from the leaf hub to sink hub. The youngsters hubs break first, trailed by its parent and subsequently the information created by the kids hubs are gathered, handled and sent by its parent hub in staggered way. After the underlying staggered break, the hubs follow the proper break of length T, which is information age period. This guarantees that the information created by the hubs arrive at the sink before the following round starts. The Fig. 1 shows the break model of the proposed framework called Delay Efficient Aggregation Timing Model (DEATM).

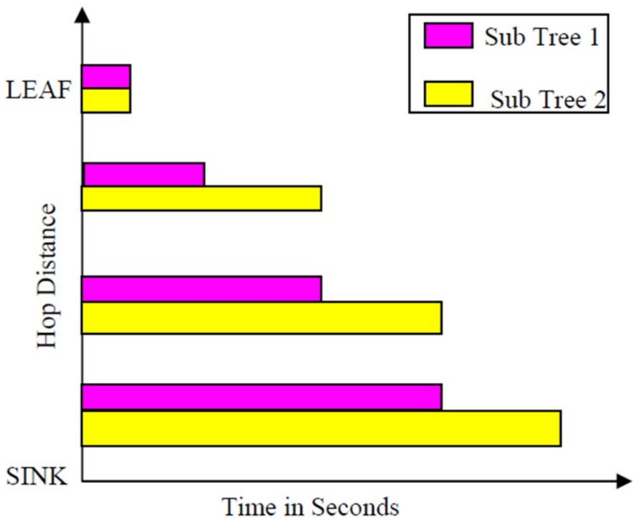

Fig. 1. Timing Model of DEATM

The Fig. 1 shows the initial outing of the nodes at different level from the sink in two different sub trees. Each color the figure shows the aggregation timer for the nodes within the same sub tree and therefore the figure shows two such a sub trees. The nodes within the same level are represented with different color and their initial outing will depend upon the entire number of youngsters within the sub tree during which it resides. The leaf nodes get the smallest amount waiting time and therefore the nodes almost the sink gets more staggering time. All the leaf nodes get an equivalent staggering time but the nodes within the same higher levels will get different stagger time. The leaf node's outing takes place early and therefore the outing of the nodes within the different levels increases because it approaches the sink. The nodes almost the sink get longer out and it's but the deadline T. The nodes within the same level will have different outing and hence the collision within the same level be avoided and it ensures that the info wave will reaches the sink through that sub tree with within the dead line.

3.1 Mathematical Model

The aggregator node I is Ai, and the leaf node j is Lj. For the aggregator node Ai, let Ni be the number of children nodes. i.e. Ni is the number of degrees (Ai). The cost of a path from any leaf node Lj to an aggregator node Ai is calculated as follows:

$$\text{Path\_cost}(L_j, A_i) = \sum_{A \in nk} \deg ree(A) \qquad (1)$$

Where n is the set of aggregator nodes in the path from L_j to A_i. The maximum path cost from any leaf node to an aggregator node is P_i is the maximum path cost from leaf node to the aggregator.

$$P_i = \max \{\text{Path\_cost}(L_j, A_i)\} \ \forall j \qquad (2)$$

The maximum path cost from any leaf node to sink is P_{sink} is the maximum path cost from leaf node to the aggregator.

$$P_{sink} = \max \{\text{Path\_cost}(L_j, sink)\} \ \forall i, j \qquad (3)$$

The initial staggered time out is calculated as follows. Let T_i is the stagger time out for the aggregator node i, which is equal to

$$T_i = T_{ci} + T_{ai} \qquad (4)$$

Where T_{ci} is the cascading time out which depends on the level in which the aggregator is in. This gives the initial timeout as in cascade time out for each node and it is same for all the nodes in the same level. The T_{ai} is the aggregation time out of the node, which depends on the number of children it has.

$$T_{ci} = 2 * [T - (T_{TD} * h)]$$

$$T_{ai} = (P_i / P_{sink}) * (T - T_{TD} * D) \qquad (5)$$

Here, h denotes the hop distance of the node Ai, D is the depth of the tree, T is the data generation period or the dead line and T_{TD} is the one hop delay between the levels. It depends on the queuing delay, MAC delay, processing delay for aggregation function and the transmission delay. It is assumed to be 0.1 seconds as used in [8]. After introducing this initial delay into the aggregation timer, the aggregation timer is triggered every T seconds, allowing the collection of packets generated by all the nodes in the collection tree for that round.

3.2 Update Phase

The aggregation timer is updated whenever the topology changes or the aggregation gain drops due to time synchronization between the nodes. Beacon messages are exchanged regularly between nodes. When the topology changes, the routing engine broadcasts messages to neighbouring nodes so that all nodes in the network update the information. The aggregation timeout Tai is recalculated each time a beacon is received. If the new value and the previous timer value differ by more than the specified threshold Δ, the aggregate timer is reset. Also, the optimal number of responses for each aggregator node per round is Ni, which is the same as the number of child nodes. If the aggregator receives less than Ni, the timer value will increase by T_{TD}, and if it exceeds Ni, the timer value will decrease by T_{TD}.

4. Implementation

The standard CTP routing protocol sets up a data collection tree by exchanging beacon messages containing information about the parent and cumulative link quality to reach the sink. Each node sends a beacon message on a regular basis. The CTP beacon message is modified to carry additional information about the hop distance and Pi. The neighbour table is also modified to record neighbour hop distances and pi. Each aggregator node finds its own child in the parent field of the beacon message sent by its child. When a node receives a beacon message from a neighbouring node with the node ID as the parent, it increments the number of child fields in the neighbouring table. The aggregator node also uses the parent node's beacon information to detect the hop distance. Each aggregator node calculates the path cost for each subtree by adding the pi of the child node and the number of child nodes. The pi of a node is calculated by finding the maximum of all pi. This information is passed to the aggregator function that calculates the value of Ti. The first-time shift is reached and the data wave reaches the sink within the specified time.

Each node uses two timers. One is for data generation and the other is for aggregation. When the construction of the collection tree phase is complete, the node starts both timers. The data timer is triggered every T seconds. The initial timeout of the aggregation timer is calculated from the beacon message and the one-shot aggregation timer starts at the calculated time. When the data timer is triggered, the node reads the default sensor and keeps it in the buffer. When the aggregation timer is triggered for the first time, the aggregation timer restarts in T seconds, which is a periodic timer. When the periodic aggregation timer is triggered, the node aggregates packets from children with unique values and sends the aggregated packets to the top of the tree. The average of simple aggregate operators is used.

5. Simulation and Result

The proposed algorithm is simulated on a Linux platform using the TOSSIM simulator, a simulator for TinyOS 2.x developed by the University of California, Bellekelly, which can execute real TinyOS code without real particles. 100 nodes are evenly distributed over an area of 200-200m2. TOSSIM uses SNR-based simulation, and the simulator parameters are like simulating an indoor environment. The nodes are placed in a uniform topology, dividing the sensor fields into grids of the same size, and randomly placing the nodes on each grid. The standard MAC IEEE 802.15.4 is used for channel access and CTP using 4-bit link estimation as the data acquisition tree. The node generates traffic every 20 seconds.

The performance parameters to consider are aggregate gain, accuracy, and error rate. Aggregation gain is defined as a measure of the reduction in communication traffic due to aggregation related to a node's energy [13]. This is the ratio of traffic reduction by aggregation to total traffic without aggregation.

where t is the number of transmissions for all unaggregated nodes. In the absence of aggregation, the aggregator node must forward all packets from its children. Therefore, the value of t is calculated by adding the total number of transmissions and receptions by all nodes. ta is the total number of transmissions with aggregation and is equal to the number of packets sent by all nodes. Aggregate gain was chosen as one of the metrics for analysing the protocol because it is directly related to the energy consumption of the node.

The information is sent in one packet by an aggregator node, which receives and analyses Nr packets from its offspring. Without aggregation, the aggregator node is responsible for forwarding all packets, regardless of their content. As a result, a node's total number of transmissions is lowered by the factor Nr/ (Nr +1). The aggregation gain is the ratio of the number of transmissions reduced as a result of aggregating to the number of transmissions not reduced. If a node is waiting for an optimum time period, it could collect the information from all of its children, then the aggregation gain will be maximum. If the aggregator timeout is shorter than the optimal timeout, the aggregator node was unable to collect information from some of its child nodes. Therefore, packets arriving after the deadline are discarded because they are not considered when finding the aggregate result. This reduces the accuracy of the aggregation process. Data accuracy is a measure of the amount of data used to extract information. It is defined as the number of readings received for the total number of packets generated on the network. The aggregated package contains the amount of child data used to find the aggregated value. Therefore, the sink can find the accuracy of the aggregation for a particular simulation time. The accuracy of the aggregation process depends on the number of packets received from its children, and then on the deadline requirements of the application and the node density of the network.

Another metric used to assess the protocol's efficiency is the miss ratio. It's calculated by dividing the number of packets that missed the deadline by the number of packets received. The packets of the current round arriving after the aggregation time out will be dropped in each data gathering round. For the duration of the simulation, each node will calculate the total number of packets dropped and the total number of packets received from its offspring. The miss ratio is calculated by adding the total number of packets dropped and received by all nodes. The number of hops, the number of nodes in the sub tree, and the node density in the surveillance field all influence these performance metrics.

The proposed algorithm is compared to simple hop-by-hop and cascade timeouts. For each simple hop, the node waits for a period of time to perform the aggregation, and for a cascading timeout, the staggered timeout is followed by a hop delay. Figures 2 through 7 show the results of the simulation test.

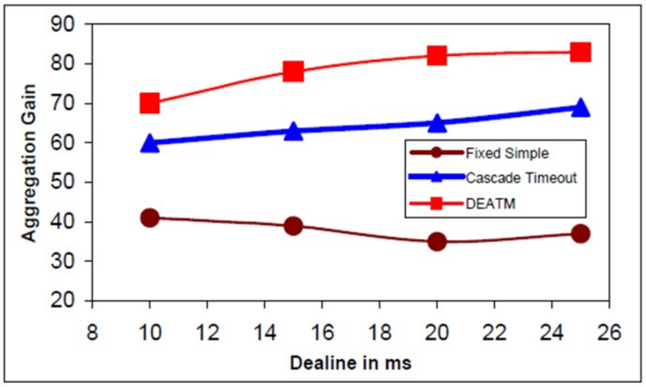

Fig. 2. Impact of Deadline on Aggregation Gain

The Fig. 2 indicates the aggregation benefit for the one-of-a-kind timing fashions as a feature of cut-off date. The nodes pattern the sensor and generate the records for each 20 seconds. This record is aggregated with the records acquired from its kids for the duration of the day trip duration and the aggregated packet is transmitted out while the aggregation timer fires. The records technology duration or cut-off date is numerous from 10 seconds to twenty-five seconds and the aggregation benefit is measured. The aggregation benefit will increase because the cut-off date will increase. If T is small, greater packets will omit the cut-off date because of small ready duration and improved records site visitors. As the cut-off date boom the records site visitors is decreased in addition to the nodes can be given greater possibilities to do the aggregation and therefore the benefit will increase. Our proposed set of rules offers higher aggregation benefit in comparison to cascade day trip seeing that in our method, the aggregator node's timeout consists of the wide variety of baby nodes and therefore it can gather greater records from its kids. In easy and cascade day trip, the nodes, that are very near the sink ought to ship the records with withinside the cut-off date and therefore the benefit is much less in comparison that of our proposed scheme.

The Fig. 3 suggests that the leave out ratio of the proposed set of rules could be very much less in comparison to different schemes due to the fact, every node waits suitable time to accumulate the statistics from all its youngsters and as a result the statistics reaches the sink with withinside the cut-off date. As cut-off date will increase, the leave out ratio decreases.

The fig. 4 suggests the statistics accuracy of the proposed set of rules as a characteristic of cut-off date. The accuracy of the proposed device is usually excessive due to the fact it can accumulate extra statistics from its youngsters in the cut-off date. Accuracy will increase with growth in cut-off date. In order to discover the effect of node density at the overall performance parameters, the simulation is carried out at some point of one hundred seconds and the range of nodes varies from 25 to one hundred. The place of the community is equal for all of the instances and as a result the density of the nodes withinside the sensor area varies.

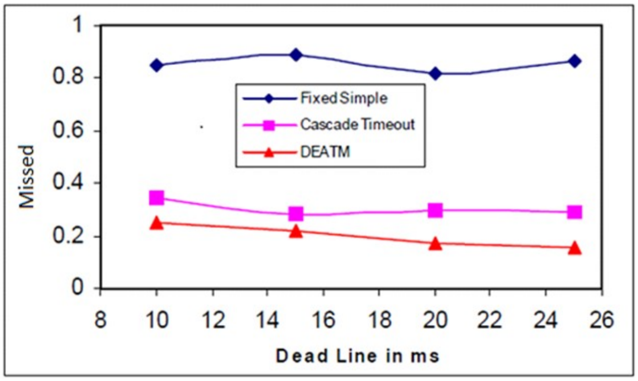

Fig. 3. Impact of Deadline on Miss Ratio

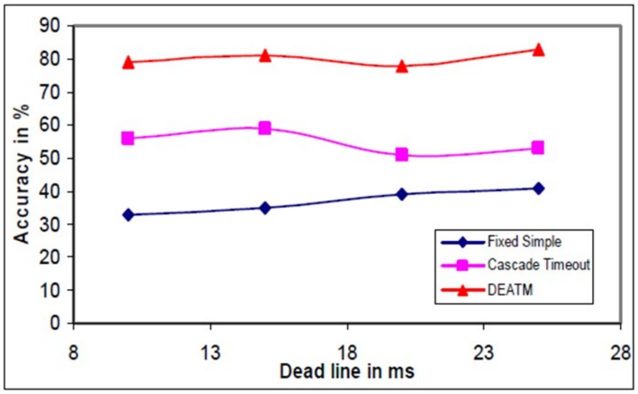

Fig. 4. Impact of Deadline on Accuracy

The Fig. 5 suggests dependency of aggregation advantage at the community length. As the community length will increase, the aggregator nodes should acquire greater packets from its kids and decrease the site visitors with the aid of using aggregation. Hence the advantage will increase because the wide variety of nodes will increase. This isn't linear because of the truth that the improved wide variety of nodes reasons collision and the kids' nodes must wait greater time to get the channel, which ends up in the lower in advantage.

The Fig. 6 suggests the effect of node density at the omit ratio. The omit ratio of the proposed scheme could be very much less as compared to the opposite schemes. Also, for the small node densities, the omit ratio does now no longer extrude a great deal for all of the schemes however the omit ratio will increase because the node density will increase. This is because of the truth that if the node density will increase, the wide variety of competition for a node to get entry to the channel will increase. This will growth the MAC deferring time and as a result the packets will omit the lifeless line and dropped with the aid of using the aggregators.

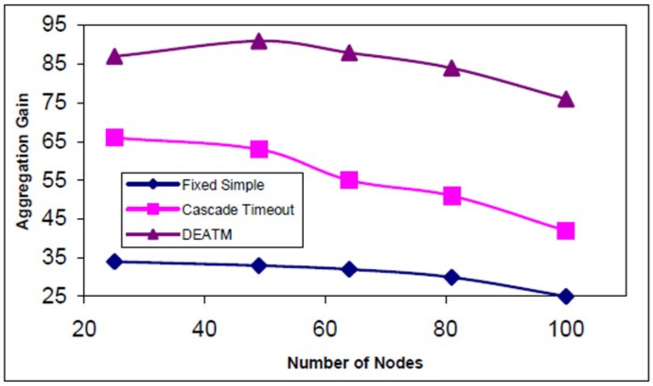

Fig. 5. Effect of Node density on Aggregation Gain

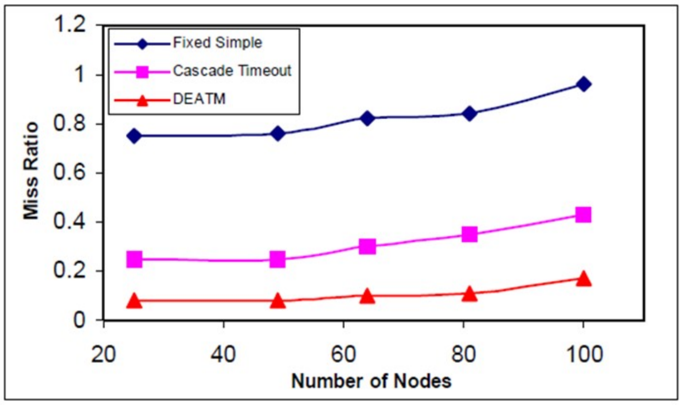

Fig. 6. Effect of Node density on Miss Ratio

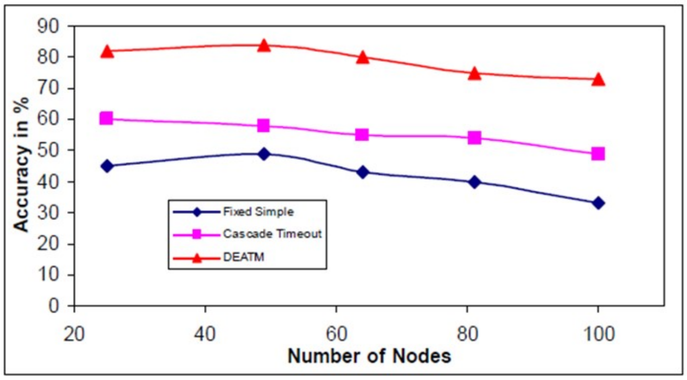

Fig. 7. Effect of Node density on Data Accuracy

The fig. 7 indicates the information accuracy of the proposed set of rules as a feature of community size. The accuracy of the proposed device is usually excessive and will increase with lower in community size. From those figures, our proposed set of rules can do higher than different algorithms even in excessive-density networks.

6. Conclusion and Future work

The proposed protocol offers extra aggregation benefit, which results in much less strength consumption, much less leave out ratio, which offers the statistics with withinside the stipulated time sure and the best statistics accuracy. Thus, our protocol can supply extra correct and clean facts to the sink in a strength green manner.

Also, our proposed set of rules is impartial of time synchronization and it doesn't want any centralized control. It additionally adjusts the time dynamically consistent with the extrade in topology or extrade in synchronization. This proposed set of rules offers extra aggregation benefit in comparison to that of cascading day out scheme and the statistics generated in a spherical is introduced to the sink withinside the equal spherical. Thus, the statistics freshness is maintained.

The benefit, leave out ratio and the accuracy of the protocol depend upon the cut-off date and the scale of the community. From our observations, for a given community size, a minimal cut-off date needs to be constant in order that the statistics may be introduced to the sink with withinside the cut-off date. The proposed paintings can be examined with the actual check mattress includes IRIS motes.

References

1. R Rajagopalan and P. K. Varshney, "Data Aggregation techniques in sensor networks: A Survey", Journal on IEEE Communications, Surveys and Tutorials, Vol. 8, Issue 4, 2006, pp.48-63.
2. Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis and Panos K. Chrysanthis, "TiNA: A Scheme for Temporal Coherency Aware in Network Aggregation", in 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, 2003, pp. 69-76.
3. Tian He, Brian M. Blum, John A. Stankovic and Tarek Abdelzaher, "AIDA: Adaptive Application-independent Data

Aggregation in wireless Sensor Networks", ACM Transactions on Embedded Computing Systems, Vol. 3, No. 2, May 2004, pp. 426–457.

4. Yujie Zhu, Ramanuja Vedantham, Seung-Jong Partk, Raghupathy Sivakumar, "A Scalable correlation aware aggregation strategy for wireless sensor networks", Journal of Information Fusion, Elsevier publishers, Vol. 9, Issue 3, 2008, pp. 354-369.
5. Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks". in IEEE INFOCOMM, 2004.
6. Ignacio Solis and Katia Obraczka, "The impact of timing in data aggregation for sensor networks", in IEEE International Conference on Communication (ICC), 2004.
7. Jae Young Choi, Jong Wook Lee, Kamrok Lee, Sunghyun Choi, Wook Hyun Kwon and Hong Seong Park, **"**Aggregation Time Control Algorithm for Time constrained Data Delivery in Wireless Sensor Networks**",** in IEEE Vehicular Technology Conference, 2006, Vol. 2, pp. 563 – 567.
8. Hong Li, HongYi Yu, BaiWei Yang and Ana Liu, "Timing control for delay-constrained data aggregation in wireless sensor networks", International Journal of communication systems (Wiley), Vol. 20, issue 7, 2007, pp. 875- 887.
9. Shan Guo Quan and Young Yong Kim, "Fast Data Aggregation Algorithm for Minimum Delay in Clustered Ubiquitous Sensor Networks", in International Conference on Convergence and Hybrid Information Technology (ICHIT), 2008, pp. 327-333.
10. Crossbow Technologies http://www.xbow.com
11. Philip Levis, Nelson Lee, Matt Welsh, and David Culler, **"**TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications**",** in the First ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 126-137.
12. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol", in the 7th ACM International Conference on Embedded Networked SensorSystems (SenSys), 2009, pp. 1-14.
13. U. Roedig, A. Barroso, and C. Screenan, "Determination of Aggregation Points in Wireless Sensor Networks", in the 30th EUROMICRO conference, 2004, pp. 503 – 510.