# Data Aggregation in Wireless Sensor Networks

The main goal of data aggregation is to decrease the amount of energy used in calculating an aggregate measurement at a sink node (this may be multiple or single sinks, but the focus here is on single sink problems)
Tradeoffs: Energy vs. Latency (negatively correlated), Energy vs. Accuracy (positively correlated)

**Tiny Aggregation (TAG)**[1]

A tree structure is maintained, rooted at the sink
Time is divided into 'epochs', in which each sensor must make a measurement
Total epoch time is divided into intervals based on the depth of the tree:

$$Interval\ Length = \frac{epoch\_length}{tree\_depth}$$

Nodes *i* levels from bottom of the tree transmit during the $i^{th}$ interval, allowing for global coordination
Internal nodes wait for children's transmitting interval to end before aggregating data and sending to parent
Main problem with TAG: Loss of a node results in complete loss of the node's entire subtree

**Synopsis Diffusion (SD)**[2]

Uses multipath routing to overcome the node loss found when using tree aggregation
Problem: This duplicates data which would be counted twice using a standard aggregation algorithm
Each node must use an order and duplicate insensitive (ODI) aggregate algorithm to estimate the true value
Example – Count aggregate:
Define a function CT(x) as: for i < x, CT(x) = i with probability $2^{-i}$
A synopsis is represented as a bit vector of length k = $\lceil \log n \rceil$, where n is the maximum number of sensors
Each node creates a synopsis by creating bit vector of length k with CT(k)$^{th}$ bit set to 1
Synopses are combined at nodes on the way to the sink using the logical OR function
At the sink, the count value can be estimated by $2^{i-1}/0.77351$, where i is lowest order bit that is still 0
Logic: Total nodes is proportional to $2^{i-1}$, based on the functionality of the CT(x) function
For more information, see [2] and [3]
Results found that SD is more robust than TAG, resulting in lower error at loss rates > 0.1
SD has higher error than TAG at low loss rates due to estimation error inherent in ODI aggregation

**Tributaries and Deltas (TD)**[4]

The TD approach aims to combine the benefits of TAG and SD in a single solution
Two types of nodes are defined:
M-Node – uses an SD approach to create estimated aggregate values from all lower nodes it receives from
This is used in high-loss situations to improve the percentage of nodes contributing to the final value
T-Node – uses a TAG approach to calculate exact aggregate (assuming no failures) from a node's children
This is used in low-loss situations to eliminate the estimation error realized when using SD
Nodes change their type within the network dynamically based on the percentage of nodes contributing
Two strategies are proposed for node-type modification:
TD-Coarse: If the percent contributing drops below a threshold, switch all possible T nodes to M nodes
Increases redundancy within the network to increase percent contributing
Adapts quickly to change, but cannot address different failure rates in separate regions of the network
TD: If the percent contributing of a node's subtree drops below a threshold, change that node's children to M nodes
This does not adapt as quickly as TD-Coarse
It can, however, maintain different proportions of M and T nodes in different areas of the network
Results found that TD/TD-Coarse is always better or equal to SD/TAG in performance
In situations where loss rates vary throughout the network, TD outperforms TD-Coarse
The main downside to the TD approach is that the switching of nodes uses extra messages/energy

**OPAG**[5]

Aims to match zero computation error of TAG with loss tolerance of TD
Nodes select a data aggregation node (DAN) several hops away using a list of DAN candidates
Each DAN candidate entry contains $ID_{DAN}$, $Level_{DAN}$, $P_{DAN}$ (probability the DAN can forward to route) and Flist.
Flist is a list of neighbours and the probability with which they will forward the data successful to the DAN
The probability of successful communication is calculated recursively from the root (longer paths = lower probability)
Nodes between a sensor and its DAN simply forward the packet towards the DAN (multipath routing)
This increases data redundancy while still only requiring a single node to aggregate (no estimation error)

Increased overhead is incurred due to periodic management of the DAN list
Results show OPAG with lower mean error than SD and TAG, but extremely high variance cloud the overall result
The energy used by OPAG was also found to be similar to that used by SD and TAG
It is unclear if the small (possible) improvement in performance is worth the added overhead within the network

**Exact Top-k (EXTOK)**[6]

Must find the maximum k values within the sensor network
Previous work either requires full updates every epoch (TAG) or estimated results (FILA)
Initially, EXTOK collects top k as in the TAG approach, but informs all nodes of the min top-k value (α)
    This leads to two types of nodes - TM-Nodes (in the top-k and must update on every change) and F-Nodes (not in the top-k and only update when they surpass α)
This greatly reduces the amount of updates required, while still maintaining exact results (achieving the specified goals)
Results find that, overall, EXTOK has lower data transmission cost than FILA (the previous state of the art)

**Histogram Incremental Update (HIU)**[7]

Using TAG, a histogram aggregate must be updated completely during every epoch
HIU eliminates a large amount of this updating by only passing on difference in bin numbers ($H_{new} - H_{old}$)
    If a sensor value does not move to another bin, no update required
    If children's updates cancel each other out (e.g., [1,0,-2], [-1,0,-1] and [0,0,-1]) no update required
Can also estimate other aggregates from histogram (e.g.,Max = $\frac{Ub_m+Lb_m}{2} \mid Count_m \neq 0, \forall i > m\ Count_i = 0$)
    Decreasing the bin size decreases the error of estimate, but increases amount of updates required
Results found that HIU resulted in much longer network lifetime than TAG approach

**Fast and Simultaneous Multi-Region Aggregation**[8]

Aims to use ideas from database systems research to allow fast and simultaneous queries over multiple regions
Assumes the sensor network is a grid (a poor assumption, but may hold in some scenarios, e.g., warehouse monitoring)
Within a grid, each sensor has location (x,y), value (v(x,y)) and prefix value that summarizes all data above and to the left
Example: prefix-sum value calculated as $pSum(x, y, ) = \sum_{m=0}^{x} \sum_{n=0}^{y} v(m, n)$ where v(x,y) is the value of sensor at (x,y)
A region e:f (rectangle with e as top-left corner and f as bottom-right corner), can have its aggregate sum calculated as:

$$Sum(e{:}f) = pSum(x_f, y_f) - pSum(x_e - 1, y_f) - pSum(x_f, y_e - 1) + pSum(x_e - 1, y_e - 1)$$

Complex (non-rectangular) regions can be calculated using a number of smaller rectangles
While this approach is theoretically fast, it was not compared to any other common aggregation technique

**Bibliography**


[1] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, 2002.

[2] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 250–262.

[3] P. Flajolet and G. Nigel Martin, "Probabilistic counting algorithms for data base applications," *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[4] A. Manjhi, S. Nath, and P. B. Gibbons, "Tributaries and deltas: efficient and robust aggregation in sensor network streams," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 287–298.

[5] Z. Chen and K. G. Shin, "OPAG: Opportunistic Data Aggregation in Wireless Sensor Networks," in *2008 Real-Time Systems Symposium*, 2008, pp. 345-354.

[6] B. Malhotra, M. A. Nascimento, and I. Nikolaidis, "Exact top-k queries in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1513-1525, 2010.

[7] K. Ammar and M. A. Nascimento, "Histogram and other aggregate queries in wireless sensor networks," in *Proc. of SSDBM*, 2011, pp. 1-12.

[8] D. Wu and M. H. Wong, "Fast and simultaneous data aggregation over multiple regions in wireless sensor networks," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 3, pp. 333-343, 2011.