In-Network Data Aggregation Techniques for WSNs

Mr. Pradeep Patil
M.tech, CSE department
Dr. Ambedkar Institute of Technology
Bangalore-560056

Mr. Shamshekhar S. Patil
Assoc. professor, CSE department
Dr. Ambedkar Institute of Technology
Bangalore-560056

***Abstract*— In this paper we provide a comprehensive review of the existing literature on techniques, approaches and protocols for in-network aggregation in wireless sensor networks. We first define suitable criteria to classify existing solutions, and then describe them by separately addressing the different layers of the protocol stack, which is likely to be needed for optimal performance. To support these techniques and approaches, a discussion regarding efficient data aggregation and researches achievement in Wireless Sensor Networks (WSNs) are presented inclusively in this paper.**

***Keywords*— Aggregation; content; context; routing; WSN.**

I. INTRODUCTION

The Wireless Networks are emerging new topology that will allow users to access the information and services electronically, regardless of their geographical positions. When many sensors cooperatively monitor large physical environment, they form a Wireless Sensor Networks (WSNs).The applications of WSN are broad, like weather monitoring, field police work, temperature, humidity, lightning condition, pressure, soil makeup, noise levels inventory and producing processes etc.

Mobile Wireless Sensor Networks (MWSNs) is also a kind of Wireless Sensor Network where nodes are mobile in nature. In which the position of sensors is varied with respect to time and all.

The recent advances in WSNs are rapidly expanding the range of applications they can support: from "traditional" environmental monitoring, where a number of nodes is scattered across an area collecting data for a single sink, to mobile scenarios involving multiple sinks. This happens when the entities to monitor are animals (e.g., in farming scenarios), people (e.g., in elderly care scenarios), or things moving around (e.g., in logistics), while several mobile devices (e.g., PDAs) are used as sinks, or actuators, also acting as sinks, are involved.

In-network data aggregation [1] is considered an effective technique to reduce communications cost by eliminating the inherent redundancy in raw data collected from the sensors. In-network aggregation based routing protocols are divided to the following approaches: tree-based, cluster-based, multipath, and hybrid.

Synopsis diffusion [2] is a general framework for data aggregation based on multipath approach and uses algorithms to avoid the double counting problem. In synopsis diffusion, sensor nodes are organized as concentric rings around the sink, each ring represents a distance (hops) to the sinks. In addition to this simple ring topology, synopsis diffusion introduced another topology called adaptive rings to increase the robustness of the network due to nodes failure or movement. Synopsis diffusion is suitable for MSNs but it lacks introducing a full routing protocol. In the meantime, it ignores the existence of multiple mobile sinks.

Tributaries and deltas [3] combines the features of both tree-based and multipath approaches. The protocol uses data aggregation tree with the network stable portions (that have low packet loss rate). In order to provide robustness, the protocol uses the multipath approach in the network portions that face high packet loss rate, or the network portions that carry partial aggregation results accumulated from many sensor readings. The major weakness of this hybrid approach is the possible overhead required to maintain the data aggregation structure in addition to the lack of dealing with mobility of sensor nodes. The organization of this paper is as follows: Section II presents a In-network aggregation techniques, Section IIII presents a general framework Synopsis Diffusion, Section IV concludes the paper.

II. IN-NETWORK AGGREGATION TECHNIQUES

E. Fasolo [1] defines the In-network aggregation process as the collecting data and routing data through multihop network, processing that data at intermediate nodes with the aim of reduced resource consumption, thereby increasing span of a network

The In-network aggregation process can be classified into two approaches as:

1. ***In-network aggregation with size reduction:it is the*** process in which data coming from different sources of a network are combined and compressed so that size of the information to be passed through the network has minimal size. Considering for an example, if a node gathers information from two different sources providing same local data then instead of sending both packets its mean value can be forwarded over the network as a single packet which reduces size of information.

2. ***In-network aggregation without size reduction:*** *it's a* process in which instead of

merging information to reduce the size the packets from different sources are merged into a single packet without processing the data. For an example, consider a node receiving two packets carrying different measures (e.g., temperature and humidity). Suh two values cannot be aggregated for a single data, but they can still be transmitted in a single packet and thus overhead can be reduced.

In-network aggregation techniques require three basic ingredients: suitable ***networking protocols,*** effective ***aggregation functions,*** and efficient ways of ***representing the data***. In the remainder of this section we briefly introduce each of these aspects.

Routing Protocols —The most important ingredient for in-network aggregation is a well designed routing protocol [2–10]. Data aggregation requires a new method of routing over traditional methods of network routing, we aim in energy consumption of a node for transmission by aggregating data. Nodes should choose their next hop based on packet content in order to execute in-network aggregation process, and such method is called *data-centric routing*. This method searches for reliable nodes using metrics which will consider the most suitable points for aggregation and also type and priority of the information.

Aggregation Functions —One of the most important functionalities that in-network aggregation techniques should provide the ability to combine data coming from different nodes. There exists many types of aggregation functions [2, 11–18], and most among them are bound to specific sensor application. We can differentiate such methods by few parameters stated below:

• ***Lossy and lossless*****:** There are two approaches of executing aggregation functions to merge or compress the information they are lossy and lossless approaches. In case of lossy approach the original values cannot be recovered after merging the data on using aggregation function. Moreover we may compromise precision in regard of values that are transmitted without compression. Wherein, the second approach (lossless) allows to compress the data by maintaining the original information which will enable us in restoring the readings at the receiver end even after usage of aggregation function.

• ***Duplicate sensitive and duplicate insensitive*****:** An intermediate node often receives multiple copies of the same data from different sources. In such case, we must avoid considering such multiple (same) data for aggregation so that we reduce wastage of efforts in deploying aggregation. And, by using duplicate sensitive method for aggregation it considers frequency of same values in the final result. Otherwise, the aggregation function is said to be duplicate insensitive.

Data Representation — A node owns a buffer whose size is not variable and hence a node faces overflow of data due to its limited storage capability, and hence it may not be able to preserve or store all the information it gathers from surrounding sources in its internal buffer. Therefore it needs to implement a few measures to keep its buffer available by discarding or dispatching data. Hence its needs to represent the information in a suitable way [19–22]. Meanwhile data structure consider to be barrier as it has to be flexible with respect to application and location characteristics. So distributed source coding techniques are defined to deal with data representation. More details on the approach are given later.

A. PROTOCOLS AND APPROACHES FOR IN-NETWORK AGGREGATION:

We briefly discuss classes of routing protocols in the following separately:

A. TREE BASED APPROACHES:

In this approach [4, 6, 9] a spanning tree is constructed with its root towards sink and these nodes will respond to queries from the sink. While data is forwarded towards the root in response to sink queries, the data is aggregated over levels of nodes from leaf to root. Packets might loose their value in the process of forwarding towards sink as data is aggregated level by level due to channel impairment

In spite of the potentially high cost of maintaining a hierarchical structure in dynamic networks and the scarce robustness of the system in case of link/device failures, these approaches are particularly suitable for designing optimal aggregation functions and performing efficient energy management. In fact, there are some studies where the sink organizes routing paths to evenly and optimally distribute the energy consumption while favoring the aggregation of data at the intermediate nodes [23, 24, 25].

Routing approaches based on aggregation trees.

1.TAG: The *Tiny AGgregation* (TAG) The TAG approach is a data oriented protocol. It uses aggregation trees in applications like monitoring. This means that nodes are allowed to produce information periodically. TAG algorithm implementation consists of two phases:

• **The *distribution phase*,** queries from sink are distributed to active sensor nodes.

• **The *collection phase*,** the aggregated data results are forward towards sink in response to its queries.

For the distribution phase, TAG uses a tree-based routing scheme rooted at the sink node. The sink broadcasts a message asking nodes to organize into a routing tree and then sends its queries. In each message there is a field specifying the level, or distance from the root, of the ending node (the level of the root is equal to zero). Initially, when a node gets a query it assumes its level to be eligible to answer to it and considers its source of query as parent. Each sensor then rebroadcasts the received message adding its own identifier (ID) and level. Thus all nodes attain an identifier(ID) and a parent. TAG queries have the following form:

SELECT{agg(expr), attrs} from SENSOR
WHERE{selPreds}
GROUP BY{attrs}
HAVING{havingPreds}
EPOCH DURATION i

All the readings in a aggregate record belong to same interval.

In collection phase, parent has to wait children's' response in order to aggregate data. Epochs are divided into

shorter intervals called *communication slots*. Usually node is put to doze mode soon after it responses in its slot. All intermediate nodes between sink and source get to aggregate the data. However, in order not to limit TAG to the few and very simple aggregation functions defined by the SQL language (e.g., COUNT, MIN, MAX, SUM, and AVERAGE) a more general classification is accounted for by partitioning aggregates according to the *Duplicate Sensitivity*, *Exemplary and Summary*, and *Monotonic* properties [8].

2. DIRECTED DIFFUSION: *Directed Diffusion* [4] is a reactive data-centric protocol. The routing scheme is specifically tailored for those applications where one or few sinks ask some specific information by flooding the network with their queries. Three phases of Directed Diffusion are: (see Fig. 1):

- ***Interest dissemination***
- ***Gradient setup***
- **Data forwarding along the reinforced paths** (*path reinforcement* and *forwarding*)

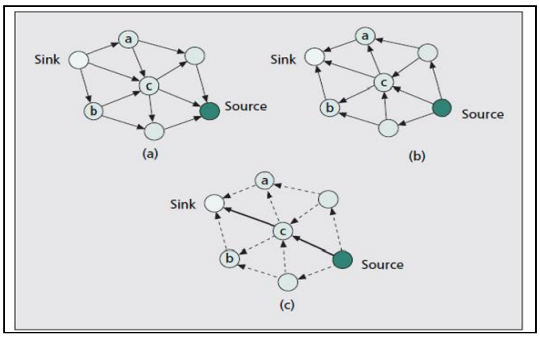

Fig. 1 A simplified scheme for Directed Diffusion: a) interest dissemination; b) gradients setup; c) data delivery along the reinforced path.

When a certain sink is interested in collecting data from the nodes in the network, it propagates an *interest* message (*interest dissemination*), describing the type of data in which the node is interested, and setting a suitable operational mode for its collection. Each node, on receiving the interest, rebroadcasts it to its neighbors. In order to, propagate the results to sink (*gradient setup*) vectors containing next hop to be maid are described. As an illustrative example (Fig. 1), if the sink sends an interest that reaches nodes *a* and *b*, and both forward the interest to node *c*, node *c* sets up two vectors indicating that the data matching that interest should be sent back to *a* and/or *b*. The strength of such a gradient can be adapted, which may result in a different amount of information being redirected to each neighbor. To this end, various metrics such as the node's energy level, communication capability, and position within the network can be used. Each gradient is related to the attribute for which it has been set up. As the gradient setup phase for a certain interest is complete, only a single path for each source is *reinforced* and used to route packets toward the sink (*path reinforcement* and *forwarding*). A valuable feature of Directed Diffusion consists of the *local interaction* among nodes in setting up gradients and reinforcing paths. This allows for increased efficiency as there is no need to spread the complete network topology to all nodes in the network.

B.CLUSTER BASED APPROACHES:

Similar to tree-based algorithms, *cluster-based schemes* [5, 7, 26, 27] also consist of hierarchical organization of the network. In cluster based approach nodes are grouped into clusters. And nodes are chosen as *cluster heads* to aggregate data locally and transmit the aggregation result to the sink.

LEACH — *Low-Energy Adaptive Clustering Hierarchy* (LEACH) [5] is a self-organizing and adaptive clustering protocol using randomization to evenly distribute the energy expenditure among the sensors. Clustered structures are exploited to perform data aggregation where cluster heads act as aggregation points. The protocol works in rounds and defines two main phases:

1. A phase to organize the clusters.
2. A phase to aggregate data and transmit.

In the first phase the nodes organize themselves into clusters. At the initial scenario, each sensor nodes declares itself to be the local cluster head then by using distributive probability and prime node is selected as a cluster head.

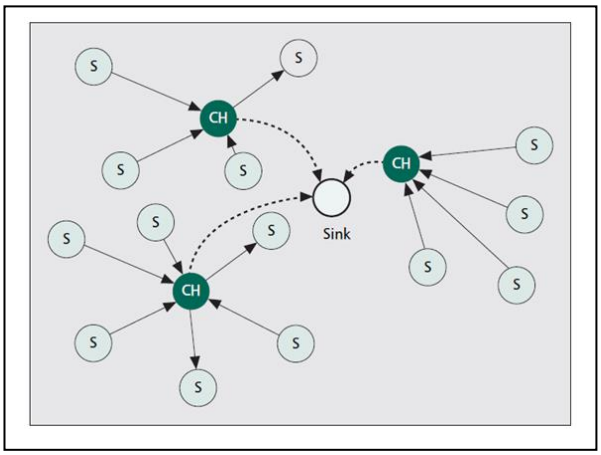

Fig. 2 LEACH clustering approach

In a second phase, source nodes send their data to the cluster head based on TDMA as to avoid collisions within the clusters. Hence it saves time and energy. From Fig. 2 CH (cluster heads) receives data from the source nodes and directly transmits them to sink node. A sleep mode is implemented to save energy when not in use like switching of radios of nodes when node is waiting for its turn in a scheduling method like TDMA. LEACH covers a large area that means there is no need of control message from sink or information from other sensor nodes in deploying it. Unlike traditional clustering algorithm LEACH allows re-electing cluster heads in order to keep them as energy efficient.

C. MULTIPATH BASED APPROACHES:

In order to overcome the robustness problems of aggregation trees, a new approach was recently proposed [2, 3, 28]. Unlike in using aggregation tree approach where data has to be propagated through a single parent, this approach allows to send data over a multiple paths. The idea is that by analyzing broadcast characteristics of the wireless medium the data is forwarded through multiple neighbors, and thus it allows intermediate nodes to data aggregation. As the tree-based schemes discussed above, multipath approaches allow

duplicates of the same information to be propagated. Hence this scheme provides higher robustness (as multiple copies of the same data can be sent along multiple paths) for some extra overhead (due to sending duplicates). This methodology uses aggregation structure such as ring topology. Here sensor nodes are classified into different levels considering number of hops towards sink. As data is propagated through these levels, towards sink the data aggregation is performed (Fig. 3).

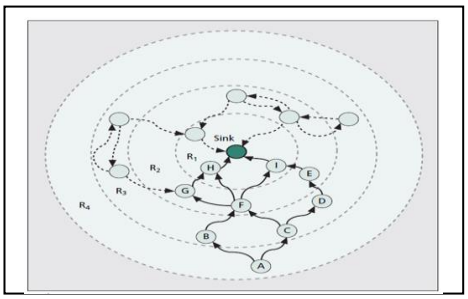

Fig. 3 Example of aggregation paths over a ring structure.

D. HYBRID DATA AGGREGATION APPROACHES:

In order to benefit from the advantages of both tree-based and multipath schemes, it is possible to define *hybrid approaches* that adaptively tune their data aggregation structure for optimal performance. The related protocol is presented next.

1. ***Tributaries and Deltas*** —The *Tributaries and Deltas* protocol [4] uses best features of both schemes tree and multipath structures to overcome their disadvantages. It works on the criteria that if there are less packet loss, data aggregation tree approach is more suitable as it has better efficiency of representing and compressing the data. Multipath approach is more suitable when packet loss rates are higher, as it provides more robustness. In a network nodes are set to use best option among above mentioned approaches and the nodes using tree-based approach are called *T* nodes and those using a multipath scheme are called *M* nodes. The following criterias [13] are used to integrate different data aggregation structures running in different regions:

• ***Edge correctness*:** An arc from *M* node can never be transacted with a *T* node. Hence aggregation result of an *M* node can only be consumed by other *M* nodes (Fig.4).

• ***Path correctness*:** In Fig. 5 there exists arcs from *T* nodes' subgraph to *M* nodes that means aggregation result can be propagated from *T* node to *T* node and also *M* nodes.

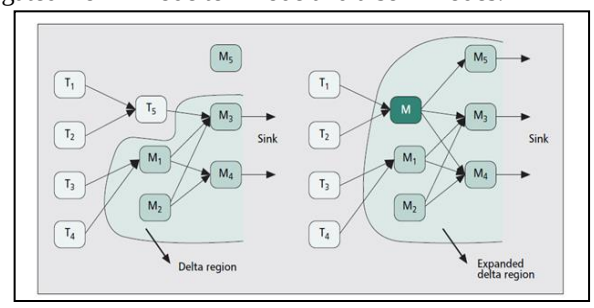

Fig. 4 Example of data gathering regions in Tributary and Delta.

III. GENERALFRAMEWOK: SYNOPSIS DIFFUSION

In this section, we describe synopsis diffusion that enables highly- accurate estimations of duplicate-sensitive aggregates. This section describes the general framework and, to illustrate the framework's use, presents examples of both a routing scheme (called Rings) and an aggregation scheme (for the Count aggregate).

Synopsis diffusion performs in-network aggregation. The partial result at a node is represented as a synopsis [30, 31], a small digest of the data. The aggregate computation is categorized by three functions on the synopses:

- **Synopsis Generation**: A synopsis creating a module SG(.) that takes a sensor reading and delivers a synopsis representing that data.
- **Synopsis Fusion:** A synopsis creating a module SF(. , .) that takes two synopses and generates a new synopsis.
- **Synopsis Evaluation:** A synopsis evaluation function SE(.) translates a synopsis into the final answer.

A synopsis diffusion algorithm consists of two phases: a distribution phase in which the aggregate query is passed through the network and an aggregation topology is defined, and an aggregation phase where the aggregate values are continually routed toward the querying node. Within the aggregation phase, each node periodically uses the function SG() to convert sensor data to a local synopsis and the function SF() to merge two synopses to create a new local synopsis.

A. SYNOPSIS DIFFUSION ON A RINGS OVERLAY:

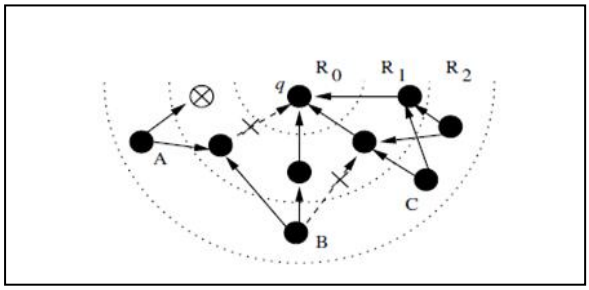

Fig. 5 Synopsis diffusion upon the Ring topology.

During the query distribution phase, nodes defines a set of rings around the node *q* as follows: *q* belongs to ring R_0, and the same belongs to ring R_i and in case it receives the query first from a node in ring R_{i-1}. The query aggregation period is divided into intervals and one aggregate answer is provided at each interval. S. Madden [9], proposes that nodes in different rings are loosely time synchronized and are allotted specific time intervals when they should be awake to receive synopses from other nodes. The duration of the allotted time is determined a priori based on the density of deployment (so that even if the sensors perform carrier sensing, all the sensors get enough time to transmit their messages once).

In this scenario (Fig. 5), node *q* belongs to R_0, the other five nodes in R_1, and four nodes in R_2. At the beginning of each interval, each node in the outermost ring (R_2 in the figure) defines its local synopsis *s = SG(r)*, where r is the sensor reading relevant to the query answer, and broadcasts it. A node in ring R_i wakes up at its allotted time, generates its local synopsis *s := SG(.)*, and acquires synopses from all nodes within transmission range in ring R_{i+1} [1]. After getting a synopsis s^1, it redefines its local synopsis as $s := SF(s, s^1)$. At the end of its allotted time the node broadcasts its reformed synopsis *s*. This synopsis is passed level-by-level towards the node *q*, which at the end of the given interval returns *SE(s)* as the result.

Fig. 5 depicts the scenario in which though in case of link and node fails, nodes *B* and *C* have at least one reliable (failure free) propagation path to the node *q*. Thus, their acquired values are accounted to the results produced in current interval. But, in case of node A all the propagation paths towards node q are failed and its values are not accounted in the result.

IV. CONCLUSION

This paper summaries Mobile Sensor Networks (MSNs) with respect to applications, data aggregation, and challenges, and describes different types of aggregation techniques and approaches with respective routing protocols in MSNs. In-network Aggregation Mechanism aims of collecting data and routing data through multihop network, processing that data at intermediate nodes with the aim of reduced resource consumption ,thereby increasing span of a network. Thus this paper can act as a source for the people working towards Data Aggregation in MSNs.

REFERENCES

[1] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. "In-network aggregation techniques for wireless sensor networks: a survey," Wireless Communications, IEEE, vol. 14, n. 2, April 2007, pp.70-- 87.

[2] Suman Nath , Phillip B. Gibbons , Srinivasan Seshan , Zachary Anderson, ℵ Synopsis diffusion for robust aggregation in sensor networks,ℵ ACM Transactions on Sensor Networks (TOSN), vol. 4 n. 2, March 2008, pp.1-40.

[3] A. Manjhi, S. Nath, and P. B. Gibbosns, ℵTributaries and deltas: efficient and robust aggregation in sensor network stream," ACM SIGMOD 2005, Baltimore, MD, June 2005.

[4] C.Intanagonwiwat *et al.*, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Trans. Net.*, vol. 11, no. 1, Feb. 2002, pp. 2–16.

[5] W.B.Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, Oct. 2002, pp. 660–70.

[6] S.Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks using Energy Metrics," *IEEE Trans. Parallel Distrib. Sys.*, vol. 13, no. 9, Sept. 2002, pp. 924–35.

[7] Y.Yao and J. Gehrke, "Query Processing for Sensor Networks," *ACM CIDR 2003*, Asilomar, CA, Jan. 2003.

[8] S. Madden *et al.*, "TAG: a Tiny AGgregation Service for Ad Hoc Sensor Networks," *OSDI 2002*, Boston, MA, Dec. 2002.

[9] Y. Xu, J. Heidemann, and D. Estrin, "Geographic-Informed Energy Conservation for Ad Hoc Routing," *ACM/SIGMOBILE MobiCom 2001*, Rome, Italy, July 2001.

[10] G. Di Bacco, T. Melodia, and F. Cuomo, "A MAC Protocol for Delay-Bounded Applications in Wireless Sensor Networks," *Med-Hoc-Net 2004*, Bodrum, Turkey, June 2004.

[11] E.Cohen and H. Kaplan, "Spatially-Decaying Aggregation Over a Network: Model and Algorithms," *ACM SIGMOD '04*, Paris, France, June 2004.

[12] A. Sharaf *et al.*, "Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks," *VLDB J.*, vol. 13, no. 4, Dec. 2004, pp. 384–403.

[13] T.He *et al.*, "AIDA: Adaptive Application-Independent Data Aggregation in Wireless Sensor Networks," *ACM Trans. Embedded Computing Systems*, vol. 3, no. 2, May 2004, pp. 426–57.

[14] E. Cayirci, "Data Aggregation and Dilution by Modulus Addressing in Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 7, no. 8, Aug. 2003, pp. 355–57.

[15] D.Petrovic *et al.*, "Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks," *IEEE SNPA '03*, Anchorage, AK, May 2003.

[16] M.Riedewald, D. P. Agrawal, and A. El Abbadi, "pCube: Updateefficient Online Aggregation with Progressive Feedback and Error Bounds," *IEEE SSDBM*
2000, Berlin, Germany, July 2000.

[17] L.Huang *et al.*, "Probabilistic Data Aggregation in Distributed Networks," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-11, Feb. 6, 2006.

[18] X. Wu and Z. Tian, "Optimized Data Fusion in Bandwidth and Energy Constrained Sensor Networks," *IEEE ICASSP '06*, Toulouse, France, May 2006.

[19] N.Shrivastava *et al.*, "Medians and Beyond: New Aggregation Techniques for Sensor Networks," *ACM SenSys '04*, Baltimore, MD, Nov. 2004.

[20]A.Bezenchek, M. Rafanelli, and L. Tininini, "A Data Structure for Representing Aggregate Data," *IEEE SSDBM '96*, Stockholm, Sweden, June 1996.

[21] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed Source Coding for Sensor Networks," *IEEE Signal Processing*, vol. 21, no. 5, Sept. 2004, pp. 80–94.

[22] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Info. Theory*, vol. 19, no. 4, July 1973, pp. 471–80.

[23] M.Ding, X. Cheng, and G. Xue, "Aggregation Tree Construction in Sensor Networks," *IEEE VTC '03*, Orlando, FL, Oct. 2003.

[24]K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks," *IEEE WCNC '03*, New Orleans, LA, Mar. 2003.

[25] H.Albert, R. Kravets, and I. Gupta, "Building Trees Based On Aggregation Efficiency in Sensor Networks," *Med-Hoc-Net 2006*, Lipari, Italy, June 2006.

[26] B. Zhou *et al.*, "A Hierarchical Scheme for Data Aggregation in Sensor Network," *IEEE ICON '04*, Singapore, Nov. 2004.

[27] A.Mahimkar and T. S. Rappaport, "SecureDAV: A Secure Data Aggregation and Verification Protocol for Sensor Networks," *IEEE GLOBECOM 2004*, Dallas, TX, Nov. 2004.

[28] S. Chen and Z. Zhang, "Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks," *ACM/SIGMOBILE MobiCom 2006*, Los Angeles, CA, Sept. 2006.

[29]D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-e_cient multipath routing in wireless sensor networks. Mobile Computing and Communications Review (M2CR), 1(2), 2002.

[30] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In ACM PODS, 2002.

[31] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. DIMACS: Series in Discrete Math. And Theoretical Computer Science: Special Issue on External Memory Algorithms and Visualization, 1999.