In-Network Data Aggregation and Processing: A Survey

Atish Majumdar
1410110081
Computer Science, B.Tech
Shiv Nadar University
am533@snu.edu.in

Prasanna Natarajan
1410110298
Computer Science, B.Tech
Shiv Nadar University
pn337@snu.edu.in

Vedant Chakravarthy
1410110489
Computer Science, B.Tech
Shiv Nadar University
vc909@snu.edu.in

Abstract — In this paper, we cover various aspects of in-network data aggregation and processing in wireless sensor networks. We survey existing literature about energy efficiency, security, compression and query processing methods, looking at multiple existing research papers across these domains, identifying and highlighting key research objectives, methodologies and finally discuss how they build upon or improve previously existing solutions. Finally, we discuss future research goals in each of the areas.

Keywords—in-network processing; security; aggregation; energy efficiency

I. INTRODUCTION

Advances in technology has led to the production of sensor devices which can sense the physical environment and provide valuable data. These sensors can communicate with each other, but also perform other tasks like signal processing, data aggregation and compression in the network rather than out of the network. There are three common features to all sensor network applications. The first one is the need for simple algorithms as sensor nodes have highly constrained resources. The second one is the need for minimizing communication among sensor nodes, since communication is the primary source of energy usage in these networks. The third is that data throughout a sensor network is bound to have spatial and temporal redundancy.

II. IN-NETWORK PROCESSING: FEATURES AND CLASSIFICATION

A. In-network data aggregation[1]

"In-network aggregation is the global process of gathering and routing information through a multi-hop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime."

- Pipelined Aggregation: In pipelined aggregation, each node combines the data it received from its children in the previous interval with the data it currently produced.
- Interval Based Aggregation: Interval based aggregation is very similar to pipelined aggregation, except that nodes aggregate data sensed for the same time interval. In this approach, each node waits for its children to transmit their data until a certain time assigned by its parent, and then combines the received data with its own data for the same time interval and transmits the aggregate to its parents.

In a different paradigm, data aggregation can be classified by how the data is treated in terms of how data is processes, mutilated and forwarded i.e. aggregation through data compression.

- With size reduction and without size reduction. They refer to the process of data compression and from one node to another and to merging different packets with the objective of reducing the control overhead respectively.

Choice of solution is usually very application dependent because the trade-off between the preservation of accuracy for the energy efficiency of data transfer. Data aggregation broadly

1) Networking protocols (routing protocols)

Classic routing protocols are typically path-centric, trying to create the shortest path to the destination (with respect to some specified metric). However, routing protocols for wireless sensor networks are energy and data-centric and try to route packets based on criteria to promote in-network aggregation.

a) Tree-based Approach

(a) TAG – Tiny Aggregation

(b) Directed Diffusion

(c) PEGASIS – Power-Efficient Gathering in Sensor Information Systems

(d) DB-MAC – Delay Bounded Medium Access Control

b) Cluster-based Approach

(a) LEACH – Low-Energy Adaptive Clustering Hierarchy

(b) Cougar

This technique is elaborated in a future section (Section III.B.)

c) Multi-path Approach

(a) Synopsis Diffusion

d) Hybrid Data Aggregation Approach

(a) Tributaries and Deltas

2) Effective aggregation functions

a) Lossy and lossless

In the first case the original values cannot be recovered after having merged them by means of the aggregation function. In contrast, the second approach (lossless) allows to compress the data by preserving the original information. A choice of style of aggregation function involves a trade-off of precision for efficiency.

b) Duplicate sensitive and duplicate insensitive

Intermediate nodes make the decision of how multiple copies of the same data are handled. Duplicate sensitive nodes aggregate functions do not disregard multiple copies while a duplicate insensitive may choose to discard the duplicates

3) Efficient ways of representing the data

Choosing what information to store, what to discard, what to compress and what to transmit are often heavily dependent on both the data itself and the data structures it uses to store data. Apart from this the manner in which each node is able to represent the data that it stores and whether to keep the complexity at the end of the data transmitter or the receiver and so on.

a) TiNa – Temporal coherency-aware in-Network Aggregation

b) DADMA – Data Aggregation and Dilution by Modulus Addressing

c) Data Aggregation by means of Feedback Control

d) Synopsis Diffusion Framework

e) The Quantile Digest

f) Distributed Source Coding

B. Secured Data Transmission Wireless Sensor Network

In-network processing helps by reducing the network traffic, but also makes the network of sensors vulnerable to different kinds of security threats. These threats are serious because even if a single node is compromised it could skew the whole system of networks. So, securing each and every node becomes necessary. The threats can be classified widely into 2 categories, outsider attacks and insider attacks. Outsider attacks can be further classified into Passive eavesdropping, Denial of service attacks (DoS) and Replay attacks. Insider attacks can be further classified into False data injection and Selective reporting. To avoid or stop all these types threats, protocols have been developed. These protocols are discussed in Section V of this paper.

C. Energy Efficiency of Transmission

As is the case with all wireless sensor networks, energy is by far the most critical resource, as wireless sensors operate on limited power sources. Hence, minimizing energy usage in all stages of application for a sensor becomes a key objective. In-network aggregation techniques reduce the number of messages that need to be sent by any node, reducing overall communication, which ends up saving a significant amount of energy. As the data is aggregated at a designated node, which then forwards the data to the base station, the other nodes need not constantly communicate over the network, which allows greater energy efficiency. Section IV of this paper covers some methods of energy efficient transmission in wireless sensor networks.

III. DATA AGGREGATION IN SENSOR NETWORKS

A. [Weiwei He, Shuang-Hua Yang, Lili Yang, Ping Li] In-network Data Processing Architecture for Energy Efficient Wireless Sensor Networks[2]

This paper proposes an in-network data processing architecture to improve the energy efficiency and the scalability and the accuracy of sensor networks. It also adapts a data mining algorithm to process data for matching a specific event. The paper finds that a reduction in energy consumption and an improvement in data accuracy is seen by adopting these methods.

Data compression: To compress data in the sensor node for energy efficiency in the WSN. This involves compressing data in each sensor node or gateway node.

Semantic data mining: To process the data from multiple sensor nodes, a mechanism is required to represent the collected data for all participants.

1) Techniques/ Methodologies

This paper attempts to develop an in-network data processing system that performs task processing on three levels: Clustering level, Data Compression level, and Data Mining level.

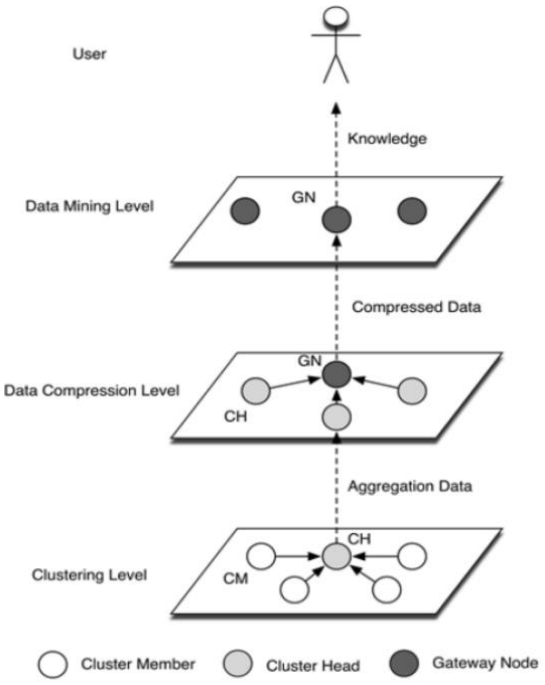

Figure 1 : In-Network Data Processing Architecture[2]

Clustering is done and the nodes with most resources available (or a different criterion based on application) are used as CH. The CHs are used to aggregate data from homogenous sensor nodes. Data from each CH is then aggregated. Data aggregation computes intermediate results at the node level. This method reduces the need for communication considerably. Instead of transmitting the sensor records for each node separately, a node firstly aggregates the incoming reading of the nodes in the communication range and then sends the intermediate results. While hardware on WSN nodes is low, if high accuracy data is required, the data should be compressed to reduce communication costs. Thus the paper suggests compressing data at CHs before sending them to GNs.

At network level, the paper proposes implementing Neighbor-Aware Clustering Aggregation (NMCA). For compression, it suggests using Optimal Clustering Compression (OCC) to reduce aggregated packet size. The CH sends the aggregated data by NMCA to the sink node. A check occurs to see it semantic model employed matches the real-time data to detect the event and send the results to the users. Only signatures of events or abnormalities are received. This is possible because of the lack of semantics of abnormal data. A data pre-processor (DP) is deployed for the outlier detection in the model for improving the data accuracy. The real-time data mining engine (RDME) operates the data mining model and event detector to match the format annotation of the sensed value.

2) Results and Findings

Simulations of this set of schemes was done using Matlab with a real dataset in terms of energy savings for the scenario of fire detection. Sensor used included CO, flame, temperature and flash.

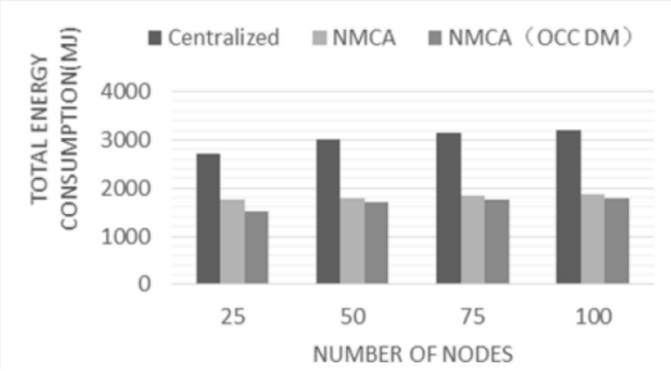

Figure 2 : Energy Consumption for Different Routing Schemes[3]

The tests included using 3 schemes, a centralized system of nodes, clustering though NMCA and clustering through NMCA accompanied by OCC and data mining. It was seen that NMCA, OCC and DM, when applied together, helped create an more energy efficient system. The results in NMCA show energy reduction by 37% energy in 25 nodes and 41% in 100 nodes. The in-network data processing architecture, which combines with NMCA, OCC and DM together, save energy by 11% in 25 nodes and 6% in 100 nodes.

3) Applications

This scheme can be especially useful when the nodes in question are homogeneous nodes, because the aggregation and processing architecture can effectively reduce the amount of data that is being communicated to a very large extent. Specifically, this might be appropriate in settings where the data coming from the network does not necessarily change rapidly – like a forest fire detector. This will mean that a large fraction of the incoming data is simply redundant and can be done away with at a local cluster level.

4) Open Questions/ Future Work

This algorithm has a potentially high setup time and a control overhead for the clustering, resulting in a time overhead. An improvement can be explored where the network can be configured in a combination of a manual configuration to assist its auto-organisation.

5) Improvements

By way of communication costs, it was seen that OCC was very significant.

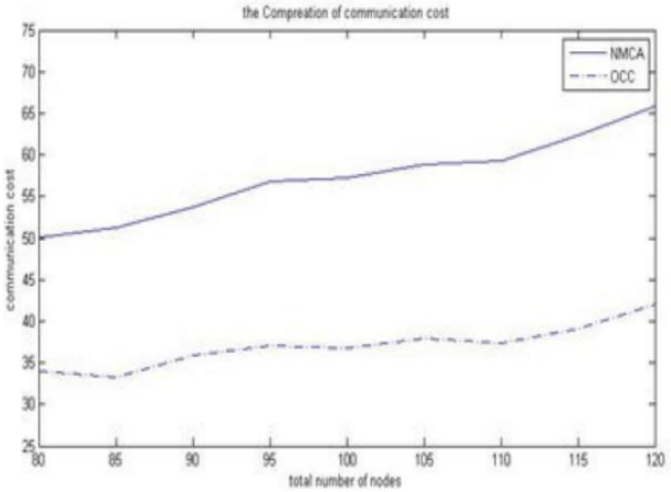

Figure 3 : Communication Cost per Number of Nodes[3]

This potentially improves on the older scheme of centralizing a base station and having devices communicate directly with the base station.

B. [Yong Yao, Johannes Gehrke] The Cougar Approach to In-Network Query Processing in Sensor Networks[3]

Architecturally, on the sensor node, the query proxy lies between the network layer and the application layer, and the query proxy provides higher-level services through queries that can be injected into the network from a specified gateway node.

The paper also argues that because a large constraint of wireless sensor networks is a very limited energy source, this processing can reduce energy consumption and improve sensor network lifetime significantly compared to traditional centralized data extraction and analysis.

1) Techniques/ Methodologies

Local computations are much cheaper than communication, thus, pushing partial computation out into the network could improve energy consumption significantly. The paper thus proposes a loosely-coupled distributed architecture to support both aggregation and in-network computation. A new query proxy layer on each sensor node interacts with both routing layer and application layer. The main addition is a query optimizer is located on the gateway node. It generates distributed query processing plans after receiving queries from the outside.

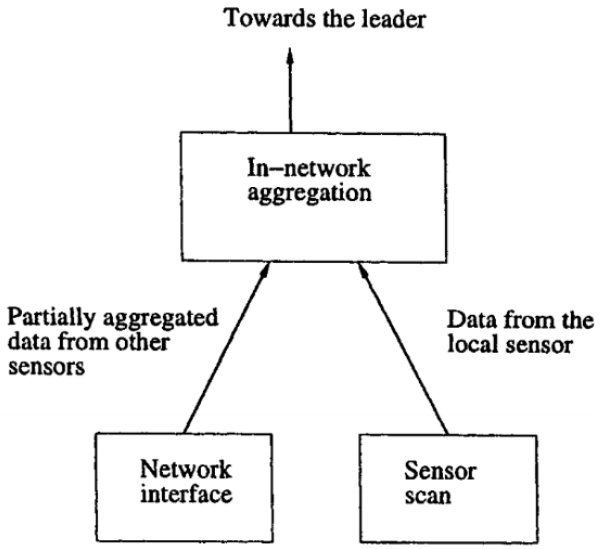

Figure 4 : Query Plan at a Source Sensor[3]

The query plan is created according to catalogue information and the query specification. This query plan specifies both the inter-sensor data flow and a per-sensor exact computation plan (at each sensor). Once generated, the plan is sent out to all relevant sensor nodes. A control structure is put in place to then do synchronization among these sensors followed by the query. At run-time, data records flow back to the gateway node as in-network computation happens on-the-fly.

2) Results and Findings

This paper proposes a new method of introducing a query layer and doing distributed in-network query processing as a way of reducing the amount of energy consumed by the sensor network. The network's life is thus extended.

3) Applications

The idea of distributed query processing and of pre-processing information at intermediate nodes is a method that can potentially be used in many kinds of sensor networks. While it is application dependent, it is true that many types of sensor networks, for example temperature sensing, environmental monitoring and such networks could preprocess data and generate a Query Plan to help conserve energy.

4) Open Questions/ Future Work

In networks with high loss rates, broken links are hard to differentiate from long delays due to high loss rates, making synchronization a non-trivial problem. Apart from this, this particular solution requires a structured query language suitable and specifically dedicated to serve the need of wireless sensor networks. This does not exist and so far, is only implemented as adaptations.

C. [Liu XY, Zhu Y, Kong L, Liu C, Gu Y, Vasilakos AV, Wu MY] CDC: Compressive data collection for wireless sensor networks [4]

This paper proposes a compressive data collection scheme for wireless sensor networks that requires fewer compressed measurements, thus greatly reduces the energy consumption. The proposed scheme allows simple routing strategy without much computation and control overheads, for strong robustness in practical applications. Analytically, the paper proves that it achieves the optimal estimation error bound. It does evaluations on real data sets (from the GreenOrbs, IntelLab and NBDC-CTD projects) and shows that compared with existing approaches, this new scheme prolongs the network lifetime by 1.5× to 2× for estimation error 5% ~ 20%.

1) Techniques/ Methodologies

A. Framework

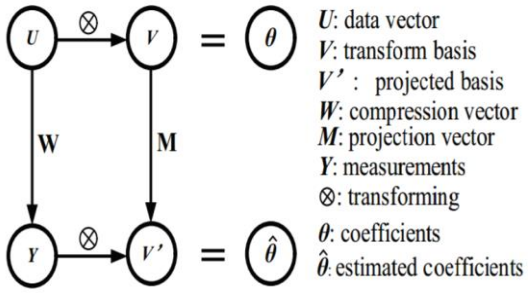

Figure 5 :Frame of CDC [4]

The framework of the paper's scheme is shown in Fig.5. It has two major components: an opportunistic routing and an estimator. The opportunistic routing is responsible for data compression and packet relaying. By modeling it as a Markov chain, the compression probability of each node can be calculated. Then, we prove that nonuniform sparse random projection (NSRP) preserves inner product of two vectors and apply this property to design a simple but quite accurate estimator.

The opportunistic routing has two tasks: packet forwarding and data compression. The data collection path is described, and then the compression process is along this path. Packet Forwarding: For node Si , we define a nearer-to-sink neighbor set as the one-hop neighbors that are closer to the sink than itself. When a packet arrives at node Si , Si compresses its sensory reading into the packet and then sends it out according to the opportunistic routing, i.e., forwarding the packet to a randomly selected one of its nearer-to-sink neighbors. In this way, each packet is guaranteed to be successfully delivered to the sink.

Due to opportunistic routing, the data collection paths are dynamic and random. Two main features are improved: energy balancing and security. These non-deterministic data collection paths will balance energy consumption among nodes, as well as preventing possible attacks.

B. Data Compression

As the packet travels towards the sink, the compression scheme adds or subtracts the sensory reading of a newly encountered node, as shown in Fig.2(a).

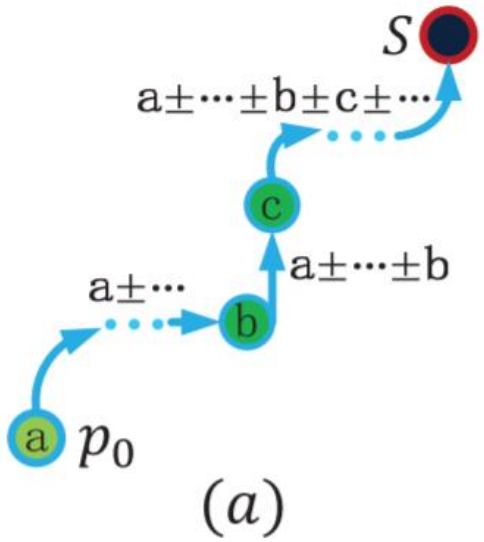

Figure 6 : Compresion methodology[4]

2) Results and Findings

For estimation error within 20%, CDC prolongs the network lifetime by 1.5× ∼ 2×. This is because that CDC requires fewer measurements, even fewer number of sensory readings in each measurement, thus greatly reduces the energy consumption.

3) Applications

4) Open Questions/ Future Work

The paper proposes, In the future, to apply the proposed non-uniform random projection based estimator for adaptive data gathering and considering more attributes.

5) Improvements

The baseline algorithm considered is as follows: Packets are transmitted back to the sink along the shortest path. Then the sink applies the k Nearest Neighbors (KNN) method to estimate the readings, i.e., by averaging the k-nearest neighbors' values. The CDC's routing strategy is quite similar to the baseline scheme and experiences quite similar performance in terms of delay. However, the proposed scheme has a better performance than the baseline. It is seen that the CDC prolongs the network lifetime by 1.5× ∼ 2× as it requires fewer measurements, fewer number of sensory readings in each measurement, thus greatly reducing the energy consumption.

IV. Energy efficiency of transmission through in-network processing

A. *[Tarik Arici, Bugra Gedik, Yucel Altunbasak, Ling Liu] PINCO: a Pipelined In-Network Compression Scheme for Data Collection in Wireless Sensor Networks[5]*

This paper presents a novel in-network data compression scheme for energy constrained, distributed, wireless sensor networks, known as PINCO (Pipelined In-Network COmpression). PINCO reduces redundancy in data collected from sensors, which leads to a decrease in wireless communication among the sensor nodes, saving energy in the process.

1) Contributions/ Improvements

The key contribution of the paper is in the area of compression and energy conservation within a sensor network, as it proposes a new algorithm for in-network compression which saves energy, while also lowering temporal redundancy, even though it is a pipelined compression scheme.

2) Techniques/ Methodologies

PINCO requires the routing protocol to provide the delivery of the query request to all nodes in the network, finding at least one route from sensor nodes back to the base node for data collection and learning the number of hops traversed on the longest route passing through a node. A simple tree- based routing algorithm is capable of providing these three services, and hence is a good option to use.

PINCO uses a tree-based routing algorithm. In this algorithm, one mote is chosen as the root node and that is the node that interfaces between the sensor and the wireline network. Before receiving a tree advertisement message all nodes have their hop-distance to the root set to the maximum possible value. The root node initiates the routing tree formation process by broadcasting a tree advertisement message (TAM) specifying its own node id and hop-distance from the root (i.e. zero in this case). All the motes in the network set the sender of TAMs as their parent while minimizing their hop-distance to the root. The algorithm prioritizes hop minimization over latency minimization in making the parent selection decision for the tree formation process. After selecting or updating its parent, each node rebroadcasts the TAM inserting its own id and hop-distance (i.e. one plus its parent's hop-distance). TAMs disseminate into the network completely in this fashion and all nodes learn their parent and hop-distance from the root.

The paper specified the complete algorithm adopted for this method.

3) Results and Findings

A simulation methodology was followed for the implementation of PINCO. It was implemented in ns-2.26 snapshot of ns-2, and used it's CSMA model as the MAC protocol. Tree based routing and PINCO scheme were implemented as separate ns-2 modules. For simulation of the mote network, packet size was set to mote packet size (30 bytes). Ns-2 connectivity model was used, which was based on the geographic distance between nodes.

The data collection application used in the simulations created a temperature map of the scenario field. Geographical positions of the immobile nodes were cataloged during deployment, and stored in the base node, where requested queries are initiated.

Simulation results showed that for good performance, PINCO parameters need to be changed for specific application characteristics, i.e., parameters are application specific.

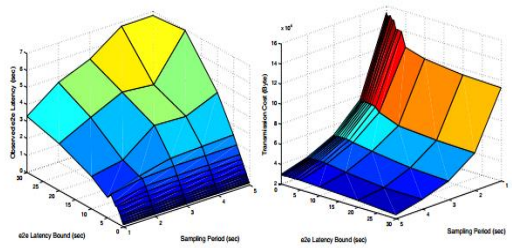

Figure 7 : Impact of e2e Latency Bound and Sampling Period[5]

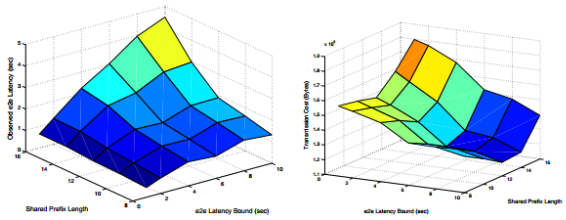

Figure 8 : Impact of e2e Latency Bound and Shared Prefix Length[5]

It was observed that minimum hop based routing-tree created "straggler" nodes in the simulations. Stragglers are defined as nodes that miss the broadcasted TAM (Tree Advertisement Message) because of MAC level contention, and therefore connect to the routing tree using "backward links", in which the recipient of the TAM is closer from the base node than the transmitter.

4) Applications

PINCO is a very useful framework in handling energy-efficient full-data collection applications and helping in discovering tradeoffs when tweaking in-network compression schemes for different degrees of performance requirements. This applies across domains, where in network compression might be required or make performance more optimal, as long as a minor latency does not affect performance.

5) Open Questions/ Future Work

As the authors have stated, they wish to develop PINCO schemes for complex types of sensor data such as audio and video, rather than single valued data like in the current implementation.

Another possible use in the future is the implementation of a different, specialized merge() function to exploit redundancies characteristic to their application or to test their own compression algorithms.

6) Improvements

PINCO buffers sensor data in the network, and combines it through a pipeline compression scheme into groups of data, while satisfying the user specified conditions of end to end latency (bounds). It trades higher latency for reduced energy consumption. Available redundancy in PINCO is reduced by exploiting commonalities of buffered data. As each node buffers its own data, and it is possible to compress data sensed at the same time intervals, PINCO does not require temporal redundancy for optimal performance even though it is a pipelined compression scheme.

B. [Prakash GL, Thesjaswini M, S H Manjula, KR Venugopal, L M Patnaik] Energy Efficient In-Network Data Processing in Sensor Networks[6]

The main objective of the paper is to use temporal coherency tolerances in addition to in-network aggregation to save energy in the network while retaining the user specified quality of data requirement. The scheme is known as Temporal Coherency-Aware in-Network Aggregation, and is independent of the underlying synchronization protocol used for sending and receiving data between the sensor nodes.

1) Contributions/ Improvements

The major contributions of the paper are:

- Proposing materialized views in sensor networks which dynamically replicate query results and enable the sharing of query processing among similar queries.
- Proposing a query processing algorithm that utilizes the materialized views when processing similar queries.
- Formulating the candidate selection problem, i.e., to an optimal subset of candidate sensor nodes to answer a query, as an optimization problem. Design of a greedy algorithm to solve the problem.

- Conducting an extensive simulation study to evaluate the performance of our proposal, comparing it to query processing schemes for sensor networks.

The contribution of this paper is a new scheme for doing temporal in-network aggregation, which balances the trade-off between the quality of results returned to users and energy consumption. in-network aggregation extends current in-network aggregation methods by utilizing temporal coherency tolerance to minimize the size and number of transmitted messages.

2) Techniques/ Methodologies

There are three particular design decisions involved in this paper:

- A query workload model
- Energy cost model
- In-network Data Processing
    - Broadcasting Query Message (BQM)
    - Processing Data Locally
    - Packet Merging

All of these form the basis of the system upon which simulations are run.

3) Results and Findings

Simulation results were performed on a test bed using TOSSIM simulator for TinyOS. Using PowerTOSSIM to estimate the total energy consumption of in-network data processing approaches. To estimate the power consumption of the mica2 sensor node mica2 energy mode is used

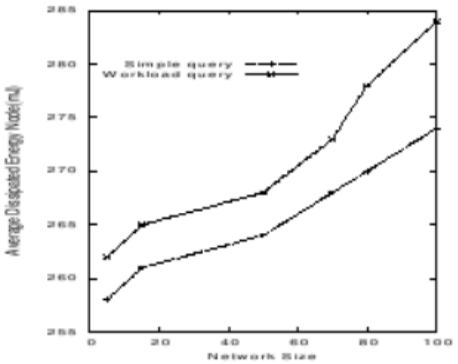

Figure 9 :Average dissipated energy vs Network Size for different query type. [6]

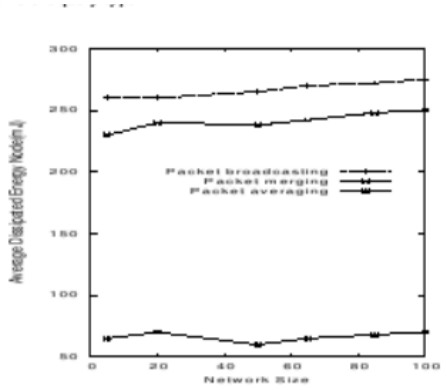

Figure 10 :Average dissipated energy for in-network data processing techniques. [6]

As the results show, without in-network data processing, each node has to send a data packet for each node whose route goes through n number of nodes, so energy consumption increases very fast. Packet broadcasting consists of all raw data, consumes more energy Packet merging consumes less energy than packet broadcasting as it consists of several sensor readings merged in a packet. Packet aggregation in in-network data processing method consumes less energy compared to other methods, it reduces redundancy in sensor readings.

4) Applications

As the results show, there are large savings in power possible by using this method of in-network processing. Hence, it is applicable to all monitoring applications of WSN, as data quality is not compromised. Temperature sensing, humidity, radiation readings, are all possible applications of this.

5) Open Questions/ Future Work

An open research question arising from this paper is the expansion of in-network aggregation to consider spatial and topological redundancy.

6) Improvements

The work improves over existing in-network processing methods, as it reduces the number, as well as the size of the messages that are transmitted over the network, hence the energy required for communication of sensor nodes is greatly decreased, while quality of the data is not greatly affected.

C. [Yingwen Chen, Hong Va Leong, Ming Xu, Jiannong Cao, Keith C.C Chan, Alvin T.S Chan] In-network Data Processing for Wireless Sensor Networks[7]

The key research objective in this paper is maximizing energy efficiency through in-network processing. In wireless sensor networks, energy is the most crucial resource. In-network data processing is a common technique in which an intermediate proxy node is chosen to house a possibly complicated data transformation function to consolidate the

sensor data streams from the source nodes, en-route to the sink node. This paper looks into the placement problem of the proxy. The energy minimization problem is formulated analytically, based on an approach termed Energy Efficient Rate-Governed Yardstick (ENERGY). An optimal solution is derived based on complete network topology information. Taking into account realistic sensor network constraints that only neighboring network connectivity is known to a node, an approximate, but effective solution is arrived at, termed "ENERGY".

1) Contributions/ Improvements

The major contribution of this paper is that the proposed technique, ENERGY, performs well even in low-density networks and for queries requested from data sources which are very distant.

2) Technologies/ Methodologies

The sensor network is modelled as a graph, G = < V, E >, where each vertex represents a sensor node and each edge represents the existence of communication ability between two nodes. A *sink node* is a sensor node that sends out a query message to one or more *source nodes* to gather information in the sensor network. In order to reduce the transmission energy consumption, data processing could be done along the path from the data sources to the sink. A sensor node is called a *proxy node* when it receives two or more raw data streams and aggregates, correlates or filters these data streams into a single stream. Using a node along the route to the sink as a proxy node can reduce the total volume of data transmitted. Consider that the sink node $a \in V$ initiates a query with lifetime T, for inputs from m data sources. In response to the query, each source S_i transmits a data stream at bit rate R_i bps. The sink a would send the transformation function to the proxy p, which will process input streams using f and forward the result stream at a bitrate of r bps to a. Let $h(s,d)$ denote the hop count between node s and d ; ε denote the average energy consumption for transmitting one bit; R_s denote the transmission rate of data stream S; T_s denote the lifetime of data stream S. Then the total energy consumption of data stream S transmitted between s and d is:

$$C(S) = \varepsilon . R_s . T_s . h(s,d)$$

The total cost of the query as a function of the proxy node p, is given as

$$C(p) = \varepsilon \tau h(a,p) + \sum_{i=1}^{m} \varepsilon R_i T h(S_i,p) + \varepsilon r T h(p,a)$$

Where the size of the data processing function is r bits. Hence, the optimal placement of the proxy node p, would be when $h(p,a)$ is the minimal hop count.

The ENERGY** approximation uses Euclidean distance as an estimate for hop count. The query cost for this case is approximated as:

$$C'(z) = \varepsilon \tau d(a,z) + \sum_{i=1}^{m} \varepsilon R_i T d(S_i,z) + \varepsilon r T d(z,a)$$

Where d(a,z) is the Euclidean distance between node a and z , and z is the position of the proxy node.

3) Results and Findings

The simulation evaluates energy consumption with or without in-network processing, and make comparisons to the ENERGY* results. In the simulation, sensor nodes are distributed in a region G, according to a uniform distribution. The communication graph is generated assuming at all nodes have the same transmission range ρ.

It is observed that in-network processing doe reduce energy consumption significantly for all kinds of queries, but ENERGY* works performs much better than its older counterpart (ADOPS), in most cases, except for short-distance queries with constrained data sources.

It is also observed that as the node density increases, the power consumption decreases, due to the fact that where there are more sensor nodes, each node has more neighbors, which helps to shorten the path from the proxy to the other nodes, thus improving energy consumption.

4) Applications

In sparsely populated areas, and harsh terrains, ENERGY can be highly useful to set up energy efficient wireless sensor networks. An example could be wildlife safety monitors across huge landscapes and sanctuaries.

5) Open Questions/ Future Work

Future research work includes the extension of data transformation function placement problem to take into account of dynamic network topology, node mobility, and more extensive considerations of the adaptively and fault-tolerant issues.

6) Improvements

Compared with other available approaches, ENERGY generates fewer control messages and achieves better performance, especially in low-density networks and for long distance queries.

V. SECURED DATA TRANSMISSION THROUGH IN-NETWORK PROCESSING

A. [Jing Deng, Richard Han, Shivakant Mishra] Security Support for In-Network Processing in Wireless Sensor Networks[8]

The focus of this paper is on the challenge of securing in-network processing in WSN. Security mechanisms are proposed for both upstream and downstream data flow. Its main focuses are:

- This paper outlines a method for determining secure membership. (how to authenticate)
- The aggregator can securely and disseminate commands in a scalable fashion.

- A mechanism for aggregators to trust data nodes by authentication.

1) Contributions/ Improvements

To commit authority to an aggregator, a mechanism is presented for distributing one-way sequence numbers, derived from one-way hash chains, to memory constrained aggregators. A lightweight mechanism for establishing pairwise secret keys between sensor nodes and their corresponding aggregators. Finally, the notion of "ripple" keys as an efficient way to achieve secure local broadcast within a small network of sensor nodes in an aggregator's domain.

2) Techniques/ Methodologies

This paper assumes a particular kind of hierarchical flexible wireless sensor network architecture. Based on this network architecture, the paper provides efficient ways to secure the network without trying to affect the in-network capabilities of the network. Securing is done via sharing keys and authentication with the help of MAC (Message authentication code). The paper before proposing a new methodology also provides reason why the existing one is not sufficient or it is not efficient.

3) Results and Findings

The proposed methods are implemented in ns2 for finding out the performance of the protocol.

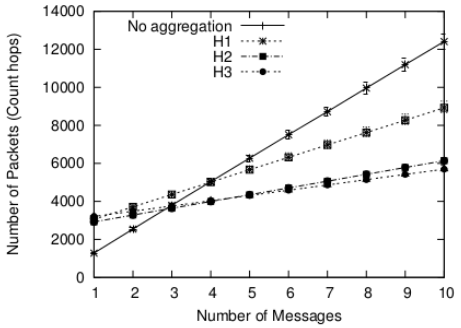

Figure 12 : Performance of In-Network Processing[8]

Here H1, H2 and H3 are levels of cluster heads/aggregators. We can see that when the level of hierarchy increases the number of packets decreases as expected because the aggregators just compute the raw data from a cluster and only send the aggregated value to the base station.

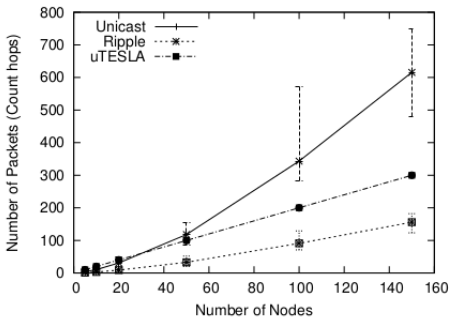

Figure 12 : Performance of Aggregator Command Dissemination[8]

We can see from the graph above that the number of packets is least in the case of Ripple method that this paper proposes as opposed to unicast and uTESLA.

4) Applications

This research could be used in applications which deals with strictly private data. For example, indoor personal monitoring, vitals inside the body etc.

5) Open Questions/ Future Work

Use of random key pre-distribution mechanisms for distributed key management in sensor networks, for being able to operate without the cooperation of the base station. Such schemes could help in building a secure, efficient and Denial of Service (DoS) attack resistant support for sensor nodes to join or leave a subgroup. The computational load on the base station and the communication between the base station and the aggregator nodes could be reduced. Integrate the security support for in-network aggregation with some popular in-network aggregation platform.

6) Improvements

This paper proposes k (5-7) keys for performing the ripple protocol for a particular cluster. Therefore, the total number of keys this paper proposes is a little too high. Section *V.B.* also uses the same idea of authentication but with lesser assumptions on the sensor network architecture and lesser number of keys to be stored. In Section *V.B.*, the use of base station's help in generating and distributing the one-way hash chain is reduced, as the cluster heads themselves will be used to compute those locally for each broadcast. This reduces the number of packets (network traffic) in the network.

B. [Sencun Zhu, Sanjeev Setia, Sushil Jajodia] LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks[9]

The paper describes LEAP+ (Localized Encryption and Authentication Protocol), a key management protocol for sensor networks that is designed to support in-network processing, while providing security properties similar to those provided by pairwise key sharing schemes.

LEAP+ includes support for multiple keying mechanisms, because all the types of data cannot be secured with only one mechanism. Specifically, LEAP+ supports the establishment of four types of keys for each sensor node: an individual key shared with the base station, a pairwise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a global key shared by all the nodes in the network. Moreover, the protocol used for establishing these keys for each node is communication- and energy-efficient, and minimizes the involvement of the base station.

1) Techniques/ Methodologies

The paper at each step proposes a solution and compares its protocol to the previous protocol that already exists and points out the advantage and disadvantage of the present protocol with respect to the other.

A prototype is implemented on TinyOS just for demonstration. This implementation has also been used to

evaluate the memory required and to test the scalability of the protocol.

2) Results and Findings

This protocol proposes the use of 4 different kinds of keys.

- Individual key: Every node has a unique key that is shared with the base station. This is used for secure communication between a node and the base station.
- Global key: This is a secret key shared by all nodes in the network and the base station. Used in broadcast messages.
- Cluster key: This key is shared within a cluster. Used in local broadcast.
- Pairwise shared key: Every node shares a pairwise key with each of its immediate neighbors. This is used for secure communication between any two nodes.

Table II. The Required RAM Space as a Function of the Number of Neighbors d

d	1	5	10	15	20	25	30
RAM (bytes)	600	736	906	1076	1246	1416	1586

As expected, as the number of neighbors increase the ram required on a mica2 mote increases.

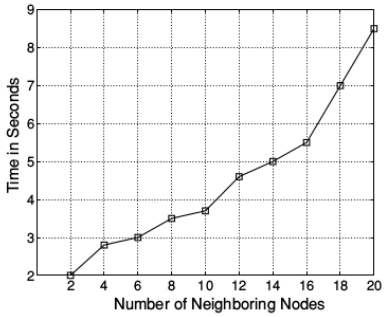

Figure 13 : Time for a sensor node to establish pairwise keys with all of its neighbours[9]

The time taken increases as the number of neighbor nodes increases in this protocol.

3) Applications

This can be used in any application where the data is sensitive. This could be used in military applications, especially because this research was funded by the US military.

4) Open Questions/ Future Work

Implementing full functionality of LEAP+ on TinyOS. Reducing the number of keys that a sensor node must store to save memory, while not increasing the network traffic.

5) Improvements

This paper proposes 4 keys to be compulsorily shared among all the sensor nodes. Section *V.C.* proposes a variable number of keys (2-4) according to where the sensor node is. The basic requirement is just to manage with 2 keys.

C. [Tassos Dimitriou, Ioannis Krontiris] Secure In-Network Processing in Sensor Networks[10]

The key research objective of the paper is to make data secure without affecting the in-network processing capabilities of a wireless sensor network, with minimum number of assumptions about the sensors and topology used in that network.

The paper proposes a new protocol for secure in-network processing in both data aggregation and data dissemination. This protocol has minimum and practically feasible assumptions about the network and the sensor nodes used. The paper describes LEAP+ (Localized Encryption and Authentication).

1) Techniques/ Methodologies

The paper explores protocols used in already existing research papers and marks a clear distinction between those and the ones this paper is proposing. The paper categorizes different types of threats. Then proposes a solution against almost all the them. It uses the idea of securing data while not affecting the in-network processing capabilities of the network via key exchanges from a lot of previous research done in the field. It then provides new efficient way to establish trust between the aggregators and other sensor nodes. The paper provides evidence about the memory required for each sensor, to implement the protocol, by simulating it a 1000 times.

2) Results and Findings

This protocol has 2 keys stored initially. One master key which the base station has and one unique key that each individual sensor nodes have. The protocol has 3 phases:

a) Key establishment for secure aggregation:

Where each of the sensor nodes, share its secure key with the aggregator for it to be able to access the data. The paper proposes that this can be achieved with a reasonable amount of memory per sensor and as the number of aggregators increase the number of keys to be stored by an aggregator decreases.

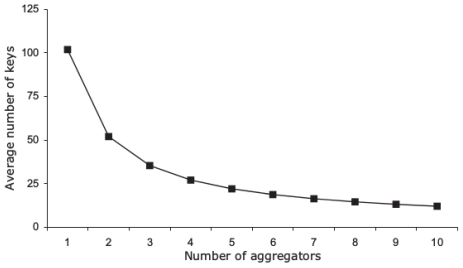

Figure 14 : Average number of keys an aggregator has to store in a random network of 1000 nodes[10]

Memory requirement for the method proposed above is determined by the number of keys the aggregator stores, implies, the number of nodes in a cluster. As expected, as the number of aggregators increases the average number of keys decreases.

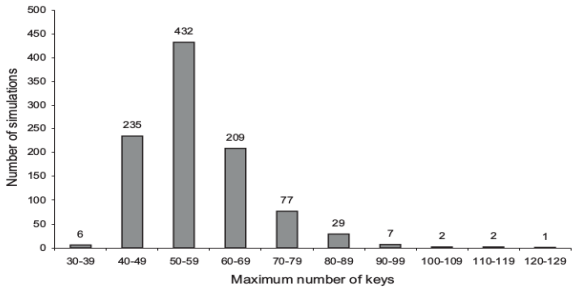

Figure 15 : Maximum number of keys an aggregator has to store in a random network of 1000 nodes. The figure 15 shows that in most cases (432 experiments out of 1000) the maximum number of keys found in any aggregator was between 50-59 [10]

To simulate a more practical situation, since these clusters are selected at random, one could have more than another, they simulated a 1000 random networks and the results are shown below.

b) Key establishment for secure dissemination

In this phase the aggregators provide a group key for a particular cluster, so that the base station, if needed, can send data to the aggregators, which in turn will send it to the whole cluster.

c) Adding new nodes to the network

In this phase, new nodes when added are handled. In this phase the new nodes send their secure keys to the aggregators of a cluster and a new master key is provided to the aggregators by the base station.

3) Applications

This work is useful for sensor networks that carry sensitive information either to the base station or from the base station. Example of applications could be surveillance of enemy and/or borderline.

4) Open Questions/ Future Work

This is a research challenge that remains, where sensor networks are reinforced with an adaptive security architecture that can monitor the network, recognize a security threat and respond either by preventing the intruder or by isolating the damage and restoring the network's normal operation. In a dynamic communication environment of thousands of nodes, such a mechanism must not hinder other network processes but rather co-exists with them and defend them.

5) Improvements

The papers this paper refers to have really strong assumptions about either the network architecture or the network capability to handle huge amount of traffic. This paper does not assume anything non-practical.

CONCLUSION

In this paper, we have presented a review of in-network processing in wireless sensor networks on 3 aspects, namely, Data Aggregation and Compression, Energy and Security.

One of the main design aspects for sensor network architectures is energy efficiency. Therefore, aggregation techniques are essential, as they aim at reducing the number of transmissions required for data collection.

Energy efficiency is critical in wireless sensor networks. The two different schemes discussed in this paper are aimed towards maximizing energy efficiency of a wireless sensor network. Compression is key for saving energy, as the data communication decreases, energy usage decreases.

Security is a major concern in wireless sensor networks, especially without affecting the in-network processing capabilities of the network. Three different protocols using a very similar approach have been discussed. Each one is an improvement from the other in terms of memory required for storing keys and network traffic, that is communication with the base station.

REFERENCES

[1] Fasolo E, Rossi M, Widmer J, Zorzi M. "In-network aggregation techniques for wireless sensor networks: a survey". IEEE Wireless Communications. 2007 Apr;14(2).

[2] He W, Yang SH, Yang L, Li P. "In-network data processing architecture for energy efficient wireless sensor networks". InInternet of Things (WF-IoT), 2015 IEEE 2nd World Forum on 2015 Dec 14 (pp. 299-304). IEEE.

[3] Yao Y, Gehrke J. "The cougar approach to in-network query processing in sensor networks". ACM Sigmod record. 2002 Sep 1;31(3):9-18.

[4] Liu XY, Zhu Y, Kong L, Liu C, Gu Y, Vasilakos AV, Wu MY. "CDC: Compressive data collection for wireless sensor networks". IEEE Transactions on Parallel and Distributed Systems. 2015 Aug 1;26(8):2188-97.

[5] Arici T, Gedik B, Altunbasak Y, Liu L. "PINCO: A pipelined in-network compression scheme for data collection in wireless sensor networks". InComputer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on 2003 Oct 20 (pp. 539-544). IEEE.

[6] Prakash GL, Thejaswini M, Manjula SH, Venugopal KR, Patnaik LM. "Energy efficient in-network data processing in sensor networks". World Academy of Science, Engineering and Technology. 2008 Dec 25;48.

[7] Chen Y, Leong H, Xu M, Cao J. "An energy-efficient framework for multi-rate query in wireless sensor networks". InComputer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on 2006 Sep (pp. 89-89). IEEE.

[8] Deng J, Han R, Mishra S. "Security support for in-network processing in wireless sensor networks". InProceedings of the 1st ACM workshop on Security of ad hoc and sensor networks 2003 Oct 31 (pp. 83-93). ACM.

[9] Zhu S, Setia S, Jajodia S. "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks". ACM Transactions on Sensor Networks, (TOSN). 2006, Nov 1;2(4):500-28.

[10] Dimitriou T, Krontiris I. "Secure in-network processing in sensor networks". Security in Sensor Networks. 2006:275-90.