DBSENSE: Information Management in Wireless Sensor Networks

Yannis Kotidis
Athens University of Economics and Business
kotidis@aueb.gr

Abstract

Query processing in sensor networks is emerging as a frontier area in data management research. This is exemplified by a flurry of research and vision papers in premium database journals and international conferences. While we are still years away from the smart dust vision, there is a consensus that our future will incorporate a plethora of sensing devices that will participate and help us in our daily activities. In the DBSENSE proposal we seek to understand the fundamental principles in designing a scalable data processing infrastructure in support of emerging applications that utilize wireless sensor node technology. We have identified four major tasks that we pursed in order to accomplish this goal. The first involves exploiting in-network processing techniques in order to leverage the large number of nodes available in such networks and reduce unnecessary data movement. The second objective is to explore distributed compression schemes that will be tailored to the types of multi-valued data streams produced by the sensing nodes. The third objective is to explore new query models that are emerging in applications of sensor networks and differ significantly from those considered in traditional information management systems. Finally, we investigate resilient query processing algorithms that can tolerate the amount of dirty data and failures that are frequent in sensor networks deployments.

1. Introduction

Recent advances in wireless technologies and microelectronics have made feasible, both from a technological as well as an economical point of view, the deployment of densely distributed sensor networks. These networks are increasingly being introduced in diverse applications such as habitat monitoring, article tracking in warehouses, home automation, earthquake disaster area monitoring, etc. Sensor networks typically consist of primitive wireless nodes that are able to "sense" their environment, produce readings, perform simple operations and, if needed, relay their results to other sensors nearby. In the majority of the applications, the sensors are powered by small batteries and replacing them is either too expensive or impossible (i.e., sensors thrown over a disaster area). Thus, designing energy efficient protocols is essential to increase the lifetime of the sensor network. Since radio operation is by far the biggest factor of energy drain in sensor nodes, minimizing the number and volume of radio transmissions is vital in data-centric applications. Even in the case when sensor nodes are attached to larger devices with ample power supply, reducing bandwidth consumption is still important due to the wireless, multi-hop nature of communication and the short-range radios usually installed in the nodes.

In DBSENSE, we seek to understand the fundamental principles in designing viable large-scale data-centric applications using wireless sensor nodes. In what follows we discuss the four major objectives of DBSENSE in more details.

Objective 1: Explore Advanced In-Network Processing Techniques

In densely distributed sensor networks there is an abundance of information that can be collected. Accumulating individual node measurements at a monitoring node close to the user is immensely expensive. Aggregation is an effective means to reduce the data measurements into a much smaller set of comprehensive statistics, like the maximum or average value of all readings. For example, sensors deployed in a metropolitan area can be used to obtain estimates on the number of observed vehicles to assess traffic conditions. Temperature sensors in a warehouse can be used to keep track of the average and maximum temperature for each floor of the building. Often, aggregated readings over a large number of sensors nodes show little variance, providing a great opportunity for reducing the number of (re)transmissions by the nodes when individual measurements change only slightly (as in temperature readings) or changes in measurements of neighbouring nodes effectively cancel out (as in vehicle tracking). Thus, we need to devise techniques that will balance the quality of the aggregate results returned to the user/application against the cost of producing these results. It is desirable that most of the required primitives are implemented inside the network, in order to minimize data movement.

Objective 2: Explore Distributed Data Reduction Schemes

Often applications require detailed historical measurements from the sensor nodes. As an example, consider sensors dispersed in a wild forest, collecting meteorological measurements (such as pressure, humidity, temperature) for the purpose of obtaining a long-term historical record and building models on the observed eco-system. At an abstract level, each sensor generates a multi-valued data feed and often substantial compression can be achieved by exploiting natural correlations among these feeds (such as in case of pressure and humidity measurements). In such cases, sensor nodes are mostly "silent" (thus preserving energy) and periodically process and transmit large batches of their measurements to the monitoring station for further processing and archiving. The new challenge in this setup is to exploit correlations in the obtained measurements within and across multiple nodes in a localized setting. We, thus, need to develop (i) compression algorithms that can be implemented in a distributed manner and are suitable for the type of data produced in sensor network applications and (ii) decentralized algorithms for clustering nodes in a way that will benefit the selected compression scheme by exploiting, for instance, spatial correlations in the obtained measurements.

Objective 3: Explore New Query Types

In our discussion so far we have considered two types of queries that are also found in traditional database management systems: aggregation queries and scan-based (select *) queries. Scan-based queries produce a potentially large detailed data stream while aggregation queries return a result of bounded size containing, for example, simple statistics. Since query processing on sensor networks needs to respect any given limits on resource unitization we would like to explore hybrid query types where parts of the data stream are automatically reduced to a set of statistics when available resources cannot tolerate a full-resolution data stream. The decision of when to perform this reduction and what parts of the data needs to be fused needs to take into account both the available resources as well as the data characteristics.

Objective 4: Handling Outliers and Dirty Data

Sensor readings are inherently dirty. Due to the unattended nature of many applications and the inexpensive hardware used in the construction of the nodes, sensor nodes often provide imprecise individual readings after a failure, i.e., they tend to fail dirty. Thus, we are faced with the daunting challenge to build applications on top of an architecture that is, at times, unreliable and unpredictable. In this proposal we will explore distributed techniques for identifying outlier readings, motivated by real applications of sensor networks. Unlike prior work we will not base our decisions on local individual readings. Instead, our planned framework looks for correlations between sets of readings from multiple sensors in order to properly classify them. Readings from outlier nodes will not be injected in the query result, in order not to distort the outcome, but they still hold valuable information that needs to be conveyed to the user/application. This way, users may investigate outliers further and take appropriate decisions. For instance, if a single sensor is continuously reported as an outlier, the network administrator may investigate it and determine that it has in fact failed and needs to be removed from further consideration during query processing or repaired, if possible.

2. Related Projects

Within the database community, there are a few recent projects, such as COUGAR[1] (Cornell University) and TinyDB[2] (Berkeley), on using embedded database systems in sensor networks. The networking aspects of wireless sensor nodes are a topic that has gained a lot of attention in the networking community. There are some ongoing EU projects related to the deployment of wireless sensor technology such as WISEBED[3], which aims to provide a multi-level infrastructure of interconnected testbeds, ALGODES[4], which focuses on modelling and networking aspects and LOTUS, which looks at algorithms and protocols for establishing secure communications among nodes. The techniques developed in the above works are complementary to ours and provide the underlying primitives that we can base our advanced query processing framework.

3. Description of DBSENSE Outcomes

We now briefly discuss the progress make by the members of our team towards fulfilling the four objectives of DBSENSE.

3.1 RFID Data Aggregation (Objective 1,2)

Radio frequency identification (RFID) technology has gained significant attention in the past few years. In a nutshell, RFIDs allow us to sense and identify objects. RFIDs are by no means a new technology.

---

[1] http://www.cs.cornell.edu/bigreddata/cougar/index.php
[2] http://telegraph.cs.berkeley.edu/tinydb/
[3] http://www.wisebed.eu/

Its origins can be traced back to World War II, where it was deployed in order to distinguish between friendly and enemy war planes [1]. Since then, RFIDs have seamlessly infiltrated our daily activities. In many cities around the word, RFIDs are used for toll collection, in roads, subways and public buses. Airport baggage handling and patient monitoring are more examples denoting the widespread adoption of RFIDs.

With their prices already in the range of a few cents, RFID tags are becoming a viable alternative to bar codes for retail industries. Large department stores like the Metro Group and Wal-Mart are pioneers in deploying RFID tags in their supply chain [2]. Individual products, pallets and containers are increasingly tagged with RFIDs. At the same time, RFID readers are placed at warehouse entrances, rooms and distribution hubs. These readers compute and communicate the list of RFID tags sensed in their vicinity to a central station for further processing and archiving. The ability to automatically identify objects, without contact, through their RFID tags, allows for a much more efficient tracking in the supply chain, thus eliminating the need for human intervention (which for instance is typically required in the case of bar codes). This removal of latency between the appearance of an object at a certain location and its identification allows us to consider new large- or global- scale monitoring infrastructures, enabling a much more efficient planning and management of resources.

Nevertheless, an immediate adoption of RFID technology by existing IT infrastructure, consisting of systems such as enterprise resource planning, manufacturing execution, or supply chain management, is a formidable task. As an example, the typical architecture of a centralized data warehouse, used by decision support applications, assumes a periodic refresh schedule [3] that contradicts the need for currency by a supply chain management solution: when a product arrives at a distribution hub, it needs to be processed as quickly as possible. Moreover, existing systems have not been designed to cope with the voluminous data feeds that can be easily generated through a wide-use of RFID technology. A pallet of a few hundred products tagged with RFIDs generates hundreds of readings every time it is located within the sensing radius of a reader. A container with several hundred pallets throws tens of thousands of such readings. Moreover, these readings are continuous: the RFID reader will continuously report all tags that it senses at every time epoch. Obviously, some form of data reduction is required in order to manage these excessive volumes of data.

In our work, we investigated data reduction methods that can reduce the size of the RFID data streams into a manageable representation that can then be fed into existing data processing and archiving infrastructures such as a data warehouse. Key to our framework is the decision to move much of the processing near the locations where RFID streams are produced. This reduces network congestion and allows for large scale deployment of the monitoring infrastructure. Our methods exploit the inherent temporal redundancy of RFID data streams. While an RFID tag remains at a certain location, its

---

[4] http://ru1.cti.gr/algodes/

presence is recorded multiple times by the readers nearby. Based on this observation we propose algorithms of increased complexity that can aggregate the records indicating the presence of this tag using an application-defined storage upper bound. During this process some information might be lost resulting in *false positive* or *false negative* cases of identification. Our techniques minimize the inaccuracy of the reduced representation for a target space constraint. In addition to temporal, RFID data streams exhibit spatial correlations as well. Packaged products within a pallet are read all together when near an RFID reader. This observation can be exploited by introducing a data representation that groups multiple RFID readings within the same record. While this observation has already been discussed in the literature [5], to our knowledge we are the first to propose a systematic method that can automatically identify and use such spatial correlations.

The contributions of our work are

- We propose a distributed framework for managing voluminous streams of RFID data in a supply-chain management system. Our methods push the logic required for reducing the size of the streams at the so-called *Edgeware*, near the RFID readers, in an attempt to reduce network congestion.
- We present a lossy aggregation scheme that exploits the temporal correlations in RFID data streams. For a given space constraint, our techniques compute the optimal temporal representation of the RFID data stream that reduces the expected error of the approximate representation, compared to the full, un-aggregated data stream. We also consider alternative greedy algorithms that produce a near-optimal representation, at a fraction of the time required by the optimal algorithm.
- We present complementary techniques that further exploit the spatial correlations among RFID tags. Our methods detect multiple tags that are moved as a group and replace them with a surrogate group id, in order to further reduce the size of the representation.
- We provided an experimental evaluation of our techniques and algorithms using real RFID data traces. Our experiments demonstrate the utility and effectiveness of our proposed algorithms, in reducing the volume of the RFID data, by exploiting correlations both at the time and space.

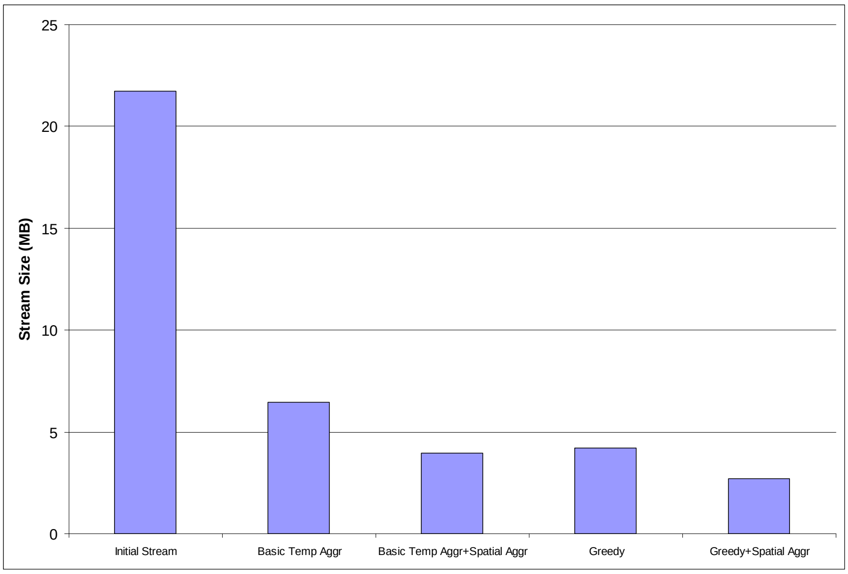

Figure 1: Performance of our techniques using real RFID traces

In Figure 1 we demonstrate the effectiveness of our techniques in reducing the size of a raw RFID data stream using our proposed temporal and spatial aggregation schemes. The first three bars correspond to lossless compression, while the last two bars utilize lossy representation in order to further reduce the size of the dataset. Additional details on our techniques and more results can be found in [6].

3.2 Building Efficient Collection Trees (Objectives 1,3)

Many pervasive applications rely on sensory devices that are able to observe their environment and perform simple computational tasks. Driven by constant advances in microelectronics and the economy of scale it is becoming increasingly clear that our future will incorporate a plethora of such sensing devices that will participate and help us in our daily activities. Even though each sensor node will be rather limited in terms of storage, processing and communication capabilities, they will be able to accomplish complex tasks through intelligent collaboration. Nevertheless, building a viable sensory infrastructure cannot be achieved through mass production and deployment of such devices without addressing first the technical challenges of managing such networks. In our work we focus on developing the necessary data collection infrastructure for supporting data-hungry applications that need to acquire and process readings from a large scale sensor network. While previous work has focused on optimizing specific types of queries such as aggregate [7], join [8], model-based [9], proximity [29] and select-all [10] queries, in DBSENSE we propose a data dissemination framework that can address the needs of multiple, concurrent data acquisition requests in an efficient manner.

It is generally agreed that one cannot simply move the readings necessary for processing an application request out of the network and then perform the required processing in a designated node such as a base station. Most recent proposals rely on building some types of ad-hoc interconnect for answering a query such as the aggregation tree [7, 12, 26, 27]. This is a paradigm of in-network processing that can be applied to non-aggregate queries as well [10, 21, 22]. In this work we explored techniques for building and maintaining efficient *data collection trees* that will provide the conduit to disseminate all data required for processing many concurrent queries in a sensor network, including long-term and ad-hoc type of queries, while minimizing important resources such as the number of messages exchanged among the nodes or the overall energy consumption.

While prior work [13, 14] has also tackled similar problems, previous techniques base their operation on the assumption that the sensor nodes that collect data relevant to the specified query need to include their measurements (and, thus, perform transmissions) in the query result at every query epoch. However, in many monitoring applications such an assumption is not valid. Monitoring nodes are often interested in obtaining either the actual readings, or their aggregate values, from sensor nodes that detect interesting events. The detection of such events can often be identified by the readings of each sensor node. For example, in vehicle tracking and monitoring applications high noise levels may indicate the proximity of a vehicle. In military applications, high levels of detected chemicals can be used to warn nearby troops. In industrial settings, where the sensors monitor the condition of machines, high temperature readings may indicate overheating parts.

In other applications, as in the case of approximate evaluation of queries over the sensor data [15, 16, 20], an event is defined when the current sensor reading deviates by more than a given threshold from the last transmitted value. In all of these scenarios, each sensor node is not forced to include its measurements in the query output at each epoch, but rather such query *participation* is evaluated on a per epoch basis, depending on its readings and the definition of interesting events. In our work we term the monitoring queries where the participation of a node is based on the detection of an event of interest as *event monitoring queries* (EMQs). It is important to note that typical monitoring queries, considered in the bulk of research so far, are a subclass of EMQs, as the former correspond to the case where the participation of sensor nodes in the query result at each epoch is fixed (either true, or not) throughout the query execution.

Our techniques base their operation on collecting simple statistics during the operation of the sensor nodes. The collected statistics involve the number of events (or, equivalently, their frequency) that each sensor detected in the recent past. Our algorithms utilize these statistics as hints for the behaviour of each sensor in the near future and periodically reorganize the collection tree in order to minimize certain metrics of interest, such as the overall number of transmissions or the overall energy consumption in the network. The formation of the collection tree is based on the collection and local transmission of only a small set of values at each node termed as *cost factors* in our framework. Using these cost factors each sensor selects its parent node, through which it will forward its results towards the base station, based on the estimated corresponding *attachment cost*. In a nutshell, the attachment cost of a parent selection is the increase in the objective function (e.g., the number of transmitted messages) resulting from this selection. Given the estimates of attachment costs that our algorithms compute, our work demonstrates that they are able to design significantly better collection trees than existing techniques.

Our contributions are summarized as follows:

- We formally introduce the notion of EMQs in sensor networks. EMQs are a superset of existing monitoring queries, but are handled uniformly in our framework, irrespectively of the minimization metric of interest.
- We present detailed algorithms for minimizing important metrics such as the number of messages exchanged or the energy consumption during the execution of an EMQ. The presented algorithms are based on the collection and transmission of a small, and of constant size, set of statistics. We introduce our algorithms along with a succinct mathematical justification.
- We present alternative techniques that we considered in our work and discuss their intuition and drawbacks.
- We extend our framework for the case of multiple concurrent EMQs of different types.

- We present a detailed experimental evaluation of our algorithms. Our results demonstrate that our techniques can achieve a significant reduction in the number of transmitted messages, or the overall energy consumption, compared to alternative algorithms. These benefit increase with increasing the network size or the number of EMQs.

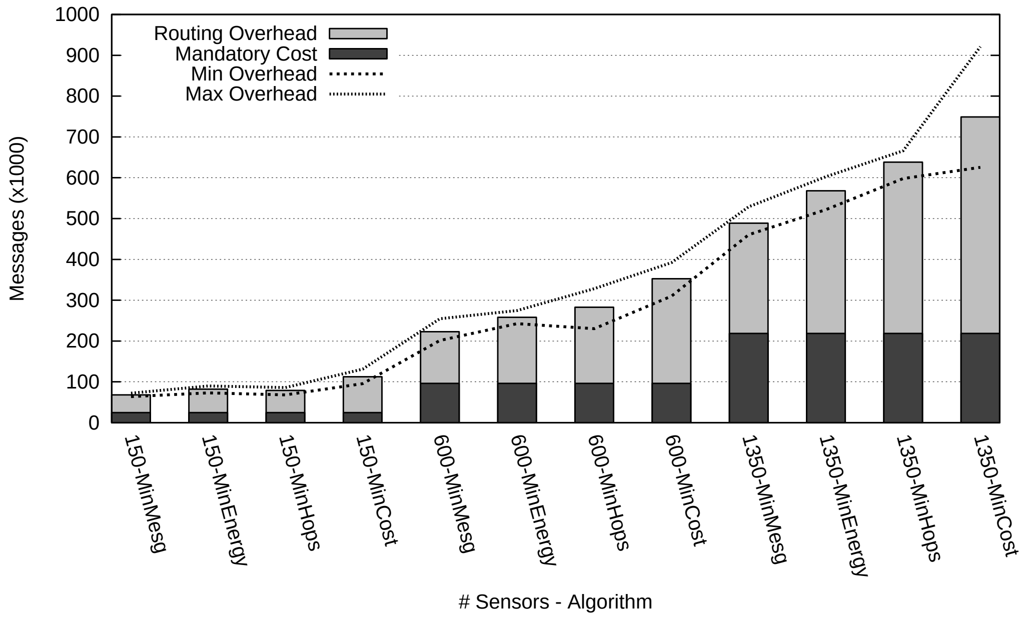

Figure 2: Comparison of Different Algorithms for Collection Tree Construction

In Figure 2 we evaluate the performance of collection trees constructed by our algorithms (MinMesg, MinEnergy) compared to existing techniques (MinHops,MinCost). The y-axis in the graph depicts the total number of messages required for evaluating a continuous aggregate query that counts the number of school buses detecting by a grid of sensors in a metropolitan area. In the x-axis we vary the number of sensors and the algorithm used for construction the collection tree. We can see that in all cases our algorithms result in more efficient collection trees that existing techniques. Additional information on these experiments and the discussed algorithms can be found in [23, 24].

3.3 Computing Outlier-free Aggregates in Sensor Networks (Objectives 1,3,4)

A lot of recent research has focused on the problem of efficiently answering declarative queries in sensor networks. These efforts primarily focus on evaluating aggregate queries, which are of great importance to surveillance applications [7, 12], and on enabling in-network processing. An equally important line of research addresses the issue of data cleaning of sensor readings [4, 17]. A measurement obtained by a node is only an approximation of the physical quantity observed and is constrained in accuracy and precision by the characteristics of the sensing device. Sensors are also often exposed to conditions that adversely affect their sensing devices, yielding readings of low quality. For example the humidity sensor on a MICA mote[5] is very sensitive to rain drops. Moreover, sensor nodes often provide imprecise individual readings after a failure, i.e., they tend to *fail dirty* [18]. Thus, data

[5] Data sheet available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf

processing applications using sensor networks must deal with information that is at times unreliable and unpredictable.

In this work, we present a novel query processing framework for aggregate queries over a network consisting of inexpensive, wireless sensor nodes that are prone to generating dirty data. Our approach computes robust, or ``meaningful'', aggregates by identifying and excluding potentially ``abnormal'' readings. In our query processing model, introduced in [19], the sensor network propagates, in multiple hops towards the base station, the aggregate values, and also recognizes and reports a concise set of readings that are believed to be outliers, along with a set of characteristic values, i.e., *witnesses*, which have been used to derive the requested aggregates. In [25, 28] we build a comprehensive framework for identifying outliers and simultaneously computing in a resilient manner aggregate values in-network. In our proposed framework, users are able to control the minimum amount of support that the readings of each node are required to achieve in order for the node to not be classified as an outlier. This ability is provided through a query-defined parameter, termed as *minimum support*, which regulates the number of tests that measurements have to pass in order to be included in aggregates. This way, our techniques are resilient to environments where spurious readings originate from multiple nodes at the same epoch, due to a multitude of different, and hence unpredictable, reasons. Our framework supports a rich query model that permits grouping, and also allows for semantic constraints on the definition (and detection) of outliers. Respecting the *minimum support* for a query and the enriched query model creates significant challenges for efficient and effective outlier detection, which we successfully address.

A key characteristic of our framework is that we do not use the same, originally constructed, collection tree to gather values throughout the life of an aggregate query, but periodically seek to readjust it based on easy to compute statistics. Using a single, monolithic collection tree constructed in advance, as in [19], does not take into account the existing readings and can lead to suboptimal decisions when computing and communicating outliers in the network. We overhaul the aggregation and outlier detection processes and periodically determine proper routing paths, based on simple statistics collected during query processing. The periodic reorganization of the collection tree based on these statistics provides significant bandwidth and energy savings compared to the monolithic approach without affecting the quality of either the produced aggregate or the detection of outliers.

Our contributions can be summarized as follows

- We propose a formal framework and algorithms for in-network aggregate query processing in the presence of multiple unreliable sensor nodes. Our new query model is based on simultaneous aggregate processing and outlier detection, and results in reporting both outlier and witness nodes in addition to the aggregates, to create increased user confidence in the produced results and to enable further investigation of suspicious readings in an efficient manner. Our framework allows the incorporation of different metrics for similarity testing between measurements of sensor nodes.

- We show that the generation of outliers by sensor nodes renders raw aggregation techniques [7, 24] meaningless and inefficient. We thus develop a novel outlier-aware process for constructing the collection tree that takes into account the nature of our evaluation process. Our algorithms are based on a periodic reorganization of the collection tree using simple statistics of how often the measurements of two sensor nodes are similar. We show that the collection trees constructed by our algorithms result in substantial savings in the number of transmitted bits (up to 43%) and in energy consumption compared to existing methods that are outlier-oblivious when constructing the collection tree. The overhead of communicating the necessary statistics and running our reconstruction algorithm is comparable to the bandwidth consumption of one epoch, and less than 0.4% overall.
- We perform an extensive experimental evaluation of our framework using real traces of sensory data. It demonstrates significant benefits compared to alternative approaches a) in the quality of the reported aggregate computed through our aggregation framework, and b) in energy and bandwidth consumption (up to 6.5 times). We also report comparable performance, in the number of detected and reported outliers, to an out-of-network computation of outliers that uses the full set of node readings per epoch.

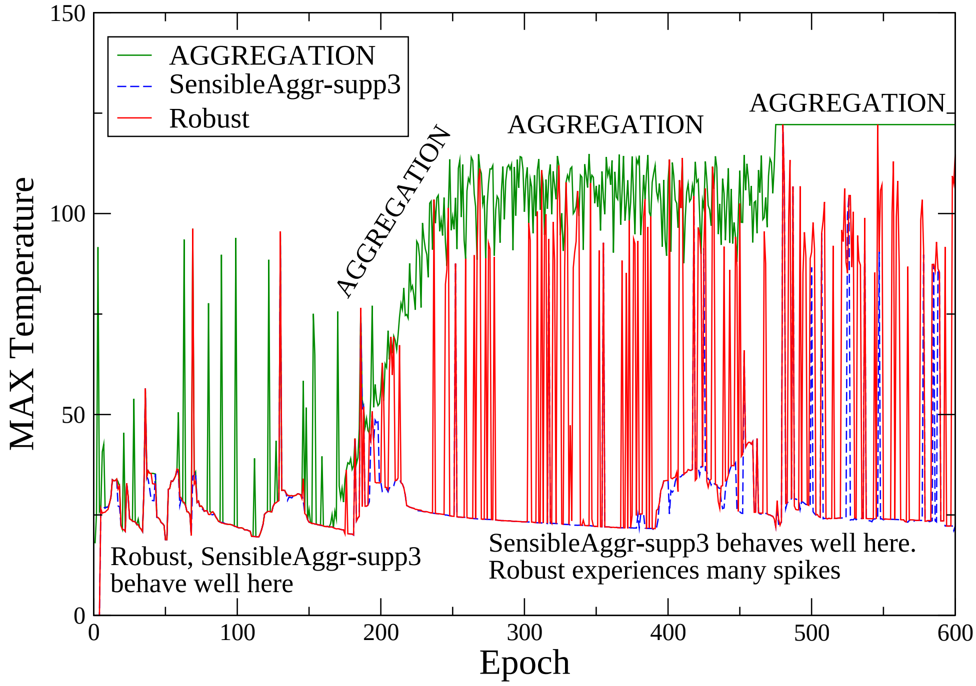

Figure 3: Computing the Maximum Temperature in a Room

In Figure 3 we used a real dataset consisting of temperature measurements of 48 sensors located in a room and attempted to compute the maximum temperature out of their readings. The line denoted as AGGREGATION present the results when using a standard aggregation technique like TAG [24]. Since this data contains many outliers, the reported aggregate value is practically useless, since many nodes report temperature readings exceeding 100^{o}C. The line denoted as Robust corresponds to our preliminary algorithm presented in [19], while SensibleAggr corresponds to our techniques presented in [25] using a minimum support value of 3. More details can be found in [25, 28].

Recently, we have extended our framework for non-hierarchical topologies, such as cluster formations. Our techniques utilize Locality Sensitive Hashing (LSH) in order to perform a distributed evaluation of outliers. A report on these techniques can be found in [30].

4. Conclusions

In this report we provided preliminary results on the outcomes of the DBSENSE project, funded by the Basic Research Funding Program (BRFP) of AUEB. Additional information and related publications can be found at http://pages.cs.aueb.gr/users/kotidis.

5. References

[1] H. Stockman. Communication by Means of Reflected Power. In IRE, Oct, 1948.

[2] S. Chawathe, V. Krishnamurthy, S. Ramachandran and S. Sarma. Managing RFID Data. In Proceedings of VLDB, pages 1189-1195, 2004.

[3] Y. Kotidis and N. Roussopoulos. A Case for Dynamic View Management. ACM Transactions on Database Systems (TODS) 26(4), pages 388-423, 2001.

[4] S. Jeffery, M. Garofalakis and M. Franklin. Adaptive Cleaning for RFID Data Streams. In Proceedings of VLDB, 2006.

[5] H. Gonzalez, J. Han, X. Li and D. Klabjan. Warehousing and Analyzing Massive RFID Data Sets. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), 2006.

[6] D. Bleco, Y. Kotidis. RFID Data Aggregation. In Proceedings of the 3rd International Conference on Geosensor Networks (GSN), Oxford, UK, July 2009.

[7] S.Madden, M. J. Franklin, J.M. Hellerstein, and W. Hong. Tag: A Tiny Aggregation Service for ad-hoc Sensor Networks. In Proceedings of OSDI Conference, 2002.

[8] D. J. Abadi, S. Madden, and W. Lindenr. REED: Robust, Efficient Filtering and Event Detection in Sensor Networks. In Proceedings of VLDB, 2005.

[9] Y. Kotidis. Snapshot Queries: Towards Data-Centric Sensor Networks. In Proceedings of the 21st International Conference on Data Engineering (ICDE), Tokyo, Japan, April 2005.

[10] A. Deligiannakis, Y. Kotidis, N. Roussopoulos. Compressing Historical Information in Sensor Networks. In Proceedings of ACM SIGMOD Int. Conf. on Management of Data, Paris, 2004.

[12] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao. The Cougar Project: A Work In Progress Report. SIGMOD Record, 32(4):53–59, 2003.

[13] J. H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In Proceedings of INFOCOM, 2000.

[14] N. Trigoni, Y. Yao, A.J. Demers, J. Gehrke, and R. Rajaraman. Multiquery Optimization for Sensor Networks. In Proceedings of DCOSS, 2005.

[15] A. Deligiannakis, Y. Kotidis, N. Roussopoulos. Hierarchical in-Network Data Aggregation with Quality Guarantees. In Proceedings of the 9th Int. Conf. on Extending DataBase Technology, 2004.
[16] A. Sharaf, J. Beaver, A. Labrinidis, and P. Chrysanthis. Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks. The VLDB Journal, 2004.
[17] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online Outlier Detection in Sensor Data Using Non-Parametric Models. In Proceedings of VLDB, 2006.
[18] S. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. Declarative Support for Sensor Data Cleaning. In Pervasive, 2006.
[19] Y. Kotidis, A. Deligiannakis, V. Stoumpos, V. Vassalos, A. Delis. Robust Management of Outliers in Sensor Network Aggregate Queries. In Proceedings of MobiDE, Beijing, China, 2007.
[20] A. Deligiannakis, Y. Kotidis, N. Roussopoulos. Bandwidth Constrained Queries in Sensor Networks. The VLDB Journal, Volume 17(3), pages 443-467, May 2008.
[21] A. Deligiannakis, Y. Kotidis, N. Roussopoulos. Dissemination of Compressed Historical Information in Sensor Networks. The VLDB Journal, Volume 16(4), October 2007.
[22] A. Deligiannakis, Y. Kotidis. Data Reduction Techniques in Sensor Networks. IEEE Data Engineering Bulletin, Volume 28(1), March 2005.
[23] V. Stoumpos, A. Deligiannakis, Y. Kotidis, A. Delis. Processing Event Monitoring Queries in Sensor Networks. In Proceedings of the 24th International Conference on Data Engineering (ICDE), Cancun, Mexico, April 2008.
[24] A. Deligiannakis, Y. Kotidis, V. Stoumpos, A. Delis. Building Efficient Aggregation Trees for Sensor Networks' Event-Monitoring Queries. In Proceedings of the 3rd International Conference on Geosensor Networks (GSN), Oxford, UK, July 2009.
[25] A. Deligiannakis, Y. Kotidis, V. Vassalos, V. Stoumpos, A. Delis. Another Outlier Bites the Dust: Computing Meaningful Aggregates in Sensor Networks. In Proceedings of the 25th International Conference on Data Engineering (ICDE), Shanghai, China, March 2009.
[26] A. Deligiannakis, Y. Kotidis, N. Roussopoulos. Processing Approximate Aggregate Queries in Wireless Sensor Networks. Information Systems, Volume 31(8), pages 770-792, December 2006.
[27] D. Zeinalipour-Yazti, P. Andreou, P. K. Chrysanthis, G. Samaras, and A. Pitsillides. The Micropulse Framework for Adaptive Waking Windows in Sensor Networks. In MDM, pages 351–355, 2007.
[28] A. Deligiannakis, V. Stoumpos, Y. Kotidis, V. Vassalos, A. Delis. Outlier-Aware Data Aggregation in Sensor Networks. In Proceedings of the 24th International Conference on Data Engineering (ICDE), Cancun, Mexico, April 2008.
[29] Y. Kotidis. Processing Proximity Queries in Sensor Networks. In Proceedings of the 3rd Int. VLDB Workshop on Data Management for Sensor Networks, Seoul, South Korea, Sept 2006.
[30] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, Y. Theodoridis. TACO: Tunable Approximate Computation of Outliers in wireless sensor networks. In Proceedings of ACM SIGMOD International Conference on Management of Data, Indianapolis, USA, June 2010.