Compressed Data Acquisition from Water Tanks

Sabbas Burdakis
School of Electronic and Computer Engineering
Technical University of Crete
sbourdakis@softnet.tuc.gr

Antonios Inglezakis
School of Electronic and Computer Engineering
Technical University of Crete
aigglezakis@softnet.tuc.gr

Antonios Deligiannakis
School of Electronic and Computer Engineering
Technical University of Crete
adeli@softnet.tuc.gr

ABSTRACT

The Aquanet project builds a low-cost monitoring system for the water resources in the city of Chania in Greece. This monitoring process is performed by installing sensor nodes at distribution tanks and periodically transmitting measurements regarding the water level to a remote base station. In this paper we present the data management aspects of this network, focusing on how the transmitted data is compressed in order to minimize the energy consumption and how information concerning the water level at the distribution tanks can be used to derive a water management policy in the network.

1. INTRODUCTION

Recent advances in microelectronics have enabled the development of large scale sensor networks for a variety of monitoring applications, ranging from wildlife monitoring, health-care, traffic monitoring, agriculture, production monitoring, battlefield surveillance, etc. What many of these applications have in common is the need to use a low-cost system for monitoring, alert and/or decision making. Given the scarcity of water resources in many regions of the world, it is interesting to investigate whether sensor networks can effectively be used to facilitate water management in large cities, in order to optimize water distribution.

In this paper we describe the compressed data acquisition from water tanks in the Aquanet project, a project that involves the design and development of a smart, autonomous, self-powered and low-cost, pilot wireless sensor network (WSN) for drinking water management, appropriate for city-wide scale and a system for water-leakage detection. Aquanet aims at augmenting the existing monitoring and management functionality for water management of the DEYAX Water Company at Chania city in Greece. In particular, given limitations in the physical design of the distribution network, water can only be moved from large, remote tanks, usually residing in high altitude, to many different distribution tanks that lie closer to the consumer. In high demand periods, such as the summer, when there is significant shortage of water, it is essential that water is distributed accurately where and when needed. This intuitively requires that water is kept as much as possible in the remote tanks. On the other hand, one should not waste valuable resources, such as the water available from springs, which cannot be used (thus, not exploiting the water that they provide, since this water ends up in lakes or in the sea) when the remote tanks are full. Thus, a monitoring system that maintains a continuous, accurate view of the network is required.

In order to maintain an accurate, up-to-date view of the network, in the Aquanet project we are currently placing sensors that measure the water level at different water tanks. There are several water tanks at the municipality of Chania and within the scope of the project sensors will be placed in about 20 points, mainly at the core of the distribution network. The readings from the monitored tanks are transmitted through a wire to sensor nodes that are placed at the rooftops of the tanks and, from the later sensors, data can then be transmitted, using multihop communication through wireless links towards a central gateway and stored in a database. Users can remotely access this data remotely through the internet, but a GIS application is also under development by other project partners for the visualization of the network state. Given the known layout of each water tank and differences in its water level, one can deduce the amount of water that is consumed. The sensor nodes placed at the rooftops are battery-powered and the frequent replacement of their batteries is not desirable, as it would require not only unounting them, but also require then ensuring that their protecting boxes are properly sealed to protect them from rain, humidity, dust etc.

In this work we demonstrate how we can explore the temporal correlation of sensor readings in order to reduce the transmitted data in the network. By appropriately transforming our problem, we can use standard techniques for the compression of data, such as Huffman Encoding, Golomb-Rice and Exponential Golomb-Rice codes. We evaluate these techniques on real data, provided by the DEYAX water company, based on their compression ratio (i.e., how much they compress the transmitted data). We need to note that another critical limitation in designing low cost sensors relates to the memory required by the sensor nodes. While some tested algorithms have slightly worse compression ratio, they may be preferred since they have minimum memory requirements and extremely simple implementations.

CySWater'15, April 13-16, 2015, Seattle, WA, USA

http://dx.doi.org/10.1145/2738935.2738943.

Our paper proceeds as follows. In Section 2 we briefly present related work, while in Section 3 we present the network setup and briefly mention some guidelines for water management. In Section 4 we present our algorithms for reducing the amount of transmitted data, along with some optimizations. Section 5 contains our experimental evaluation, while Section 6 contains concluding remarks.

2. RELATED WORK

While a vast amount of past work (i.e., [7, 5, 4]) has looked into ways of prolonging the lifetime of sensor networks by reducing data communication, many research findings are not applicable in our application scenario. First of all, the fluctuating water demand renders as inapplicable the use of models [4, 5] that infer/predict the readings of one sensor from the corresponding readings of other nodes. While in-network processing [7] is very efficient for aggregate functions such as MAX, MIN, SUM etc, they are not efficient when the readings of all sensor nodes need to be collected at frequent intervals. Please note that compression techniques that seek to collect large amounts of historical data [1] are also inapplicable, as they can transmit data at infrequent intervals and, thus, fail to provide an up-to-date view of the network. Because accurate water level readings are required in our application, since a future goal is to also detect water leaks in the network, prior approaches for approximate aggregate queries [2, 11] are inapplicable, while standard data collection techniques [7] do not achieve any data reduction.

In [8] the authors design and implement a query based processor to collect each measurement collected by sensor nodes at each epoch at a central site, called TinyDB. The work in [3] describes a lossy compression schema, by identify linear correlations among historical readings of the sensor node measurements. Their framework compresses and transmits the data only when enough data is collected.

Several data compression techniques have been implemented in wireless sensor networks [13]. Due to the limited resources available in our sensor nodes, we opted for lossless compression methods with low memory and processing requirements. These methods are based on Huffman, Golomb-Rice, Exponential techniques (see [9], [10] and [14]). The work in [14] describes modified static and adaptive Huffman algorithms, on the residue values, suited for wireless sensor networks. The authors in [9] describe a version of an exponential-Golomb code that can compress negative and non-negative differences based on their statistical characteristics.

In our work we exploit the statistical features of the data, in order to compress the collected measurements either without a coding lexicon, or with a low memory space dictionary, as in the case in [6, 12, 10]. The work in [10] calculates the optimal number of low order bits to represent the group index in the Golomb-Rice encoding technique. A statistical compression technique, that uses a predictive coding scheme to compress differences between current observed values and predictive values, for WSN is described in [6]. Residues/Differences which fall inside a small error range are represented with a compressed code, while the remaining values are sent in a raw, uncompressed form. Similarly, the work in [12] presents results for the probability distribution of the residual values, by identifying the optimal parameters from the statistical distribution of the collected measurements, in order to maximize the energy savings.

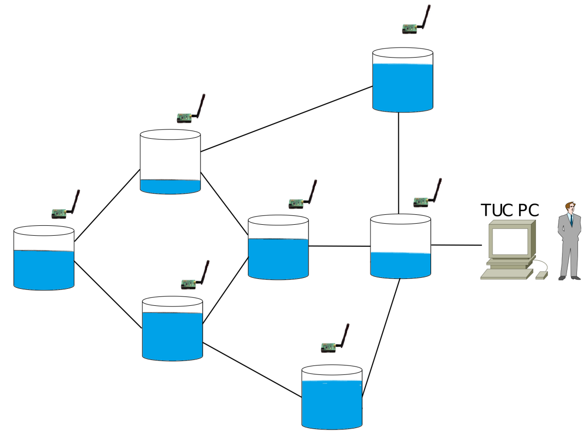

Figure 1: Aquanet - Water Distribution Network

3. PROBLEM SETUP

In this work we describe the important aspects of data acquisition and data management for a low power, water management system based on a wireless sensor network for a city distribution network. An illustration of the water management network topology is depicted in Figure 1. The network contains some large, remote tanks, usually residing in high altitude, that supply water to many different distribution tanks that lie closer to the consumer. Sensors that measure the water level are placed in water tanks. Their readings are transmitted through a wire to sensor nodes that are placed at the rooftops of the tanks and, from the later sensors, data can then be transmitted, using multihop communication through wireless links towards a central gateway and stored in a database. The gateway and database are placed at the Technical University of Crete. Prior to their transmission, the collected water level readings are encoded and compressed, in order to reduce the bandwidth consumption and to prolong the network lifetime.

Given the known layout of each water tank and differences in its water level, one can deduce the amount of water that is consumed. This estimated water consumption can then be used in order to derive a simple water management algorithm. In high demand periods, such as the summer, when there is significant shortage of water, it is essential that water is distributed accurately where and when needed. This intuitively requires that water is kept as much as possible in the high-altitude remote tanks. On the other hand, one should not waste valuable resources, such as the water available from springs, which cannot be used (thus, not exploiting the water that they provide, since this water ends up in lakes or in the sea) when the remote tanks are full. Thus, besides estimating the current water consumption in the network, another important input parameter for decision making is the amount of water that can be collected by springs and other "free" resources. If the latter amount cannot cover the current water consumption, then we operate in a mode of shortage of water, by trying to accumulate water in the large, remote tanks, while operating the distribution tanks at lower levels. In prolonged high demand periods the distribution tanks will operate close to their minimum allowed level. At the same time, water pumping from springs is set to its maximum value, water is pumped from water wells and, as a last resort, water may be purchased by a differ-

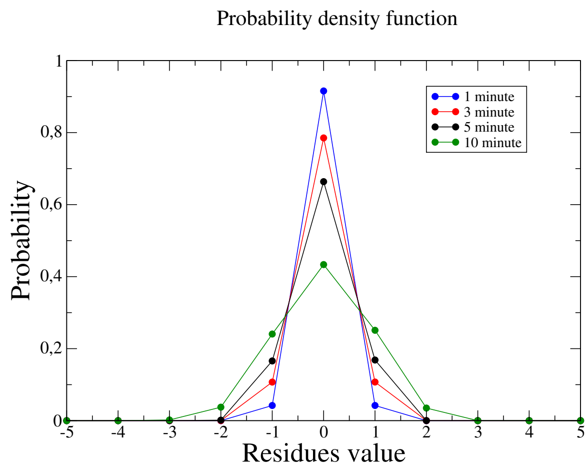

Figure 2: Residue values distribution, by varying the consecutive sensor measurements sample rate

ent provider. On the other hand, if the water supplied by springs is enough, we start filling the distribution tanks and keep operating them close to their maximum level, while pumping only the required amount of water fron springs.

4. COMPRESSING WSN MEASUREMENTS

4.1 Data Collection - Main Ideas

In this section, we present the in-network processing scheme that we used in our system, for the dissemination of compressed and approximate measurements. In particular, we demonstrate how we can explore the temporal correlation of sensor readings, in order to reduce the bandwidth of the transmitted measurements and to maximize the network lifetime. We adopt statistical compression models, where each discrete tank level value is represented with a variable-length code, based on the probability distribution of the source. An higher frequency alphabet value corresponds to a higher appearance probability of occurrence and is preferable to be encoded using a shorter binary representation. In contrast, an infrequent level tank measurement is encoded using a larger binary encoding. We want to generate binary sequences that can be uniquely decoded and with the minimum average coding size. Coding algorithms based on the probability estimation produce binary sequences, containing a lower number of bits, while the coding process is either performed entirely without a coding lexicon, or using a low memory space dictionary.

Due to the limited memory capabilities of our low cost sensors nodes, we focused on encoding techniques that can efficiently compress the transmitted data, are simple to implement and have low memory requirements. We, thus, focused on techniques such as Huffman Encoding, Golomb-Rice and Exponential Golomb-Rice codes. Such techniques are efficient when some data values are significantly more frequent from others, by encoding frequent values with fewer bits. In the real data that we had available, this was not the case, thus seemingly rendering them as ineffective.

However, one can naturally expect consecutive measurements from any given water tank to have similar values. We exploit this data characteristic by using Residual-Delta Coding compression techniques to compress the differences

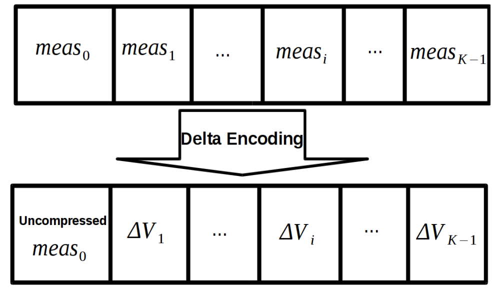

Figure 3: Delta Coding packet transformation, while sending the first measurement as is. Deltas are then compressed using standard encoding techniques.

(deltas) of consecutive sensor measurements, rather than the measurements themselves. In Figure 2 we depict the distribution of such differences in our real data when the interval between consecutive measurements is varied. The distribution resembles a Laplacian distribution.

Similar to prior work, for the Golomb-Rice and Exponential Golomb-Rice encoder, negative residues values can be represented by mapping (during the encoding process) ΔV_i to the value $\Delta V_i{}' = 2|\Delta V_i| - sign(\Delta V_i)$. Respectively, during the decoding process we map an even received difference value $\Delta V_i{}'$ to $\Delta V_i{}'/2$, while an odd $\Delta V_i{}'$ value is mapped to $-(\Delta V_i{}'/2)+1$. This process is not needed for the Huffman encoder, since it can easily store symbols corresponding to negative values in its tree dictionary building process.

One may incorrectly assume that it suffices to transmit at each timestamp the delta of the current reading from the previous one. In sensor network applications messages are often lost due to collisions. Thus, such an approach would incur a reconstruction error that may accumulate over time. Instead, we propose delaying the transmission of data until K successive measurements have been collected and to compress them as a group. The value of K depends on the maximum latency tolerated by our application for receiving water tank level measurements. To avoid the same problem with lost packets, the first measurement is sent as is, while each of the remaining $K-1$ measurements $meas_i$ is encoded via successive differences $\Delta V_i = meas_i - meas_{i-1}$. This is depicted in Figure 3. The receiver can recalculate the original measurements packets by adding the difference on the first uncompressed data and then, in the same manner, for the rest of the residue values. In case of lost packets, we exploit the temporal correlation of the measurements and interpolate the missing measurements based on the measurements prior and after the lost packet. As a note, due to the acquisition of measurements at specific intervals and the little time required for the transmission of data, there is no need to transmit timestamps for the encoded measurements. Although this technique fails to reduce the transmitted values by a factor greater than K (since 1 out of K values is not compressed), it is in practice very efficient at reducing the number of discrete values that can occur, thus substantially limiting the size of the coding dictionary.

4.2 Additional Optimizations

We now present two additional optimizations for further reducing the amount of transmitted data. The first optimization exploits the Laplacian distribution of the residual values and the high frequency of the most common difference in order to further compress the data. The second optimization targets a lossy compression of the collected measurements.

Trailing Zero Deltas. Until now we have only exploited the distribution of the residues values. In the same manner, we can exploit K consecutive differences in the sent packet. As we have mentioned, consecutive measurements are correlated abd do not significantly change over a small time window. Moreover, given Figure 2, the most common residual ΔV value is by far the value of zero. We now describe an additional compression ratio optimization that can be employed. When sending a packet with K encoded measurements, we can omit the binary code of the trailing zero deltas (i.e., the most common difference between consecutive readings) in the encoding sequence, instead of including the binary representation for all differences. The receiver decodes the coding sequence and, if he receives fewer than K values, fills the remaining trailing positions with zero delta values. We term this optimization as *ZeroLast* in our experiments.

Allowing for Lossy Compression. An equally significant aspect is to implement a lossy compression approach for comparative performance testing with the lossless techniques. Generally, lossy compression techniques ensure higher compression ratios, but incur loss of accuracy since the decompressed data does not represent the real measured values with 100% accuracy. Techniques they would introduce additional complexity in our implementation, or have more memory requirements, are rejected.

We now explain how our techniques can better compress the transmitted data when we allow a maximum absolute *error factor* ϵ at the reconstructed data. For each of the last $K-1$ measurements $meas_i$ of the packet, instead of transmitting the encoding of the delta ΔV_i from the previous measurement, we select and encode a different $\Delta V_i^{'}$ value, such that: (1) when the received reconstructs the approximate value of $meas_i$, the maximum absolute error of the reconstruction is ϵ, and (2) $\Delta V_i^{'}$ has the smallest binary encoding from all permissible values that satisfy the first condition. In case of a tie between different $\Delta V_i^{'}$ values with encodings of the same length, we select the one that results at the lowest reconstruction error. We demontrate in our experiments that even a small value of ϵ can significantly reduce the size of the transmitted data, especially when combined with our ZeroLast optimization.

5. EXPERIMENTS

We now evaluate our techniques for compressing WSN measurements from water tanks. For our metric we use the commonly used *compression ratio*, defined as: $(1-\frac{output\ size}{input\ size})\times 100\%$. All data sizes are expressed in bytes. For the described lossy-approximate compression techniques, we define the maximum absolute error factor ε which ensure us to provide, user pre-defined error guarantees. In our experiments we utilized a real world data set with tank level measures provided to us by the Municipal Enterprise of Water and Sewage of Chania (DEYAX). We used 10% of the data for training (i.e., to extract the residue distribution). Our validation data consists of water level measurements, taken every minute for an entire year. Our simulator was written in C language. All experiments were run on a Alix

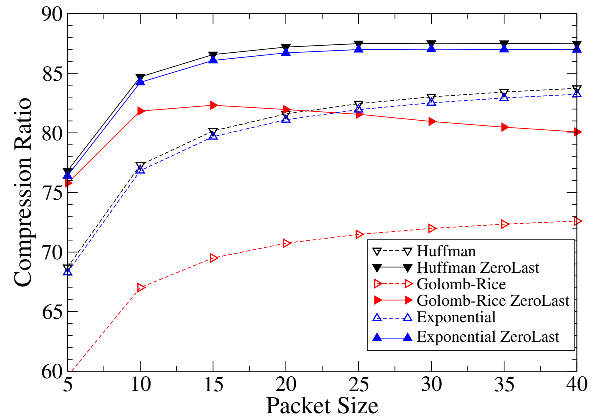

Figure 4: Compression Ratio vs Packet Size

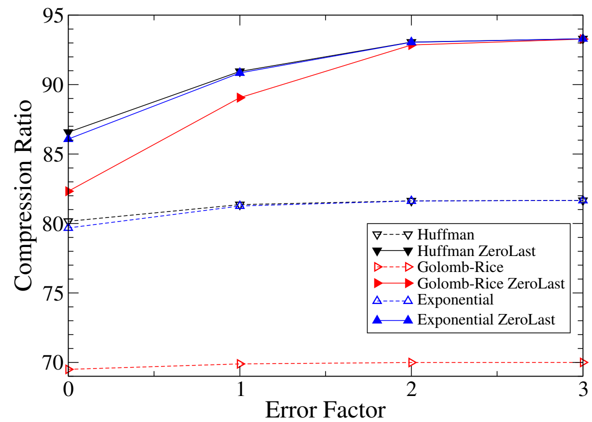

Figure 5: Compression Ratio vs Error Factor, Packet Size=15

system board, running Ubuntu, with processor clocked at 500 MHz and 256 MB RAM.

In Figure 4 we depict the compression ratio as we vary the number of measurements encoded within a single packet. Please recall that the first measurement is always sent uncompressed. As the packet size increases, the overhead of the uncompressed measurement is amortized over more measurements, thus resulting in an increasing compression ratio. Huffman Encoding performs the best, while Exponential Golomb-Rice closely follows. Please note that the latter method has no memory requirements and, thus, could be preferable in cases of sensors with very limited memory capabilities.

The ZeroLast optimization offers significant benefits in all cases. The benefits are larger for smaller packet sizes since, in that case, a larger percent of the measurements (i.e., those with zero delta values at the end of the packet) is encoded without requiring any space. Please note that the reduction for ZeroLast is always twice in Golomb-Rice, compared to Huffman and to Exponential Golomb-Rice. The reason is that the value of 0 was encoded using two bits in Golomb-Rice, while it required just one bit in Exponential-Golomb Rice and in Huffman. Thus, a given number of trailing zero delta values in a packet saves twice as much space (compared to not using ZeroLast for each particular technique) in Golomb-Rice when using ZeroLast. Please note that the number of encoded values in a packet is often dictated by the application, as increasing this value leads to a larger latency for receiving the measurements from each tank (i.e., the required number of time periods must expire in order to collect the desired number of measurements).

In Figure 5 we depict the compression ratio for all encoding algorithms when we vary the maximum allowed error at the reconstruction of the measurements, when each packet includes 15 measurements. While this allowed data approximation provides some benefits for all techniques, these benefits are more profound when this is combined with our Zero-Last optimization. This is because many delta differences at the end of the packet can often be encoded, for all encoding algorithms, as a zero difference, thus significantly reducing the transmitted information.

6. CONCLUSIONS

In this paper we presented our techniques for effectively reducing the amount of transmitted data from sensors placed at water tanks in the city of Chania in Greece. Our algorithms employ residual-delta compression techniques, coupled with encoding schemes like Huffman Encoding, or Golomb-Rice and Exponential Golomb-Rice codes. Our experiments demonstrate that our techniques can achieve high compression ratio, especially when coupled with additional optimizations that we propose.

ACKNOWLEDGENTS

This work was supported by the SYN11-6-925 AquaNet project, which is executed within the framework of the "Cooperation 2011" program of the Greek General Secretariat for Research & Technology (GSRT), funded through European Union and national funds. The authors thank the research personnel of the Telecommunication Systems Research Institute (Technical University of Crete, Greece), of the Water Municipality of Chania (DEYAX) and of the Geospatial Enabling Technologies (GET, Greece) company for their assistance and for providing helpful suggestions during this research.

7. REFERENCES

[1] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Compressing Historical Information in Sensor Networks. In *ACM SIGMOD*, 2004.

[2] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Hierarchical In-Network Data Aggregation with Quality Guarantees. In *EDBT*, 2004.

[3] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Dissemination of compressed historical information in sensor networks. *The VLDB JournalâĂŤThe International Journal on Very Large Data Bases*, 16(4):439–461, 2007.

[4] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-Driven Data Acquisition in Sensor Networks. In *VLDB*, 2004.

[5] Y. Kotidis. Snapshot Queries: Towards Data-Centric Sensor Networks. In *ICDE*, 2005.

[6] Y. Liang and W. Peng. Minimizing energy consumptions in wireless sensor networks via two-modal transmission. *ACM SIGCOMM Computer Communication Review*, 40(1):12–18, 2010.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny Aggregation Service for ad hoc Sensor Networks. In *OSDI*, 2002.

[8] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2003.

[9] F. Marcelloni and M. Vecchio. A simple algorithm for data compression in wireless sensor networks. *Communications Letters, IEEE*, 12(6):411–413, 2008.

[10] T. Robinson. Simple lossless and near-lossless waveform compression. *Cambridge Univ. Eng. Dept., Tech. Rep. TR*, 156, 1994.

[11] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. TiNA: A Scheme for Temporal Coherency-Aware in-Network Aggregation. In *MobiDE*, 2003.

[12] T. Srisooksai, K. Kaemarungsi, P. Lamsrichan, and K. Araki. Energy-efficient data compression in clustered wireless sensor networks using adaptive arithmetic coding with low updating cost. *International Journal of Information and Electronics Engineering*, 1(1):85–93, 2011.

[13] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki. Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 35(1):37–59, 2012.

[14] C. Tharini and P. V. Ranjan. Design of modified adaptive huffman data compression algorithm for wireless sensor network. *Journal of Computer Science*, 5(6):466, 2009.