Research Article

Well-Suited Similarity Functions for Data Aggregation in Cluster-Based Underwater Wireless Sensor Networks

Khoa Thi-Minh Tran, Seung-Hyun Oh, and Jeong-Yong Byun

Department of Computer Science, Dongguk University, Gyeongju 78-714, Republic of Korea

Correspondence should be addressed to Seung-Hyun Oh; shoh@dongguk.ac.kr

Received 22 May 2013; Accepted 10 July 2013

Academic Editor: Tai-hoon Kim

Copyright © 2013 Khoa Thi-Minh Tran et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an efficient data aggregation approach for cluster-based underwater wireless sensor networks in order to prolong network lifetime. In data aggregation, an aggregator collects sensed data from surrounding nodes and transmits the aggregated data to a base station. The major goal of data aggregation is to minimize data redundancy, ensuring high data accuracy and reducing the aggregator’s energy consumption. Hence, similarity functions could be useful as a part of the data aggregation process for resolving inconsistencies in collected data. Our approach is to determine and apply well-suited similarity functions for cluster-based underwater wireless sensor networks. In this paper, we show the effectiveness of similarity functions, especially the Euclidean distance and cosine distance, in reducing the packet size and minimizing the data redundancy of cluster-based underwater wireless sensor networks. Our results show that the Euclidean distance and cosine distance increase the efficiency of the network both in theory and simulation.

1. Introduction

Underwater wireless sensor networks (UWSNs) are composed of sensor nodes that are deployed in an underwater environment and are capable of monitoring nearby surroundings. Sensor nodes are small devices with constrained energy and little memory, and in underwater environments, they are costly and difficult to replace. In addition, the sensor nodes are scattered over large areas with the purpose of periodically sensing nearby surroundings and transmitting data to a sink or base station. Hence, a crucial issue for the efficient deployment of a UWSN is to maximize the network lifetime.

Clustering is a valuable technique to extend network lifetime [1, 2], especially in UWSNs with sensor nodes that are densely deployed over a large area. Each cluster consists of a cluster head (CH) and several cluster members (CMs). After forming a cluster, a CH is responsible for collecting data from its CMs and for transmitting the data to the sink or base station (BS). Hence, CHs and aggregators have very similar behaviors, and data aggregation is also an important technique for cluster-based UWSNs. There are many techniques that can be used for network clustering, such as CH selection [3], network communications (intrannetwork and internetwork) [4, 5], cluster joining [6], and data aggregation [7–11]. In this work, we assume that clusters are already formed and focus only on using the similarity function for data aggregation.

Data aggregation has been researched as an essential technique for reducing the energy consumption in wireless sensor networks by minimizing redundancy from the raw data sensed by the multiple sensor nodes as well as the number of transmissions to the sink or BS [7]. Not only can the data aggregation process help to enhance the accuracy of information which is obtained by entire networks but it can also reduce the traffic load and prolong the network lifetime [8]. However, a drawback of data aggregation is the reduction of the inherent redundancy and total number of messages. Hence, the accuracy of the final results could also be reduced [10]. In addition, sensor nodes may be positioned randomly in the environment that they are meant to periodically monitor. Neighboring sensor nodes may collect very similar
error in a highly energy-efficient manner was presented in functions that aggregates partial results with no computation and data collection. Their results reveal that the proposed finding proper data aggregation nodes, creating signatures, collect data from sources: creating a routing tree structure, layer and the data aggregation layer. It uses four phases to sink. redundancy and sending only necessary information to the their approach reduces data size by eliminating in-network while these economic data are performed at the aggregator level and is called local aggregation in sensor networks. The algorithm is performed in a threshold to decide how similar two compared data are, or dissimilar sets will not appear as a candidate pair with a high probability. The authors also compared their algorithms to other algorithms and demonstrated the effectiveness of their algorithms through experimental evaluation on real and synthetic data sets.

To achieve a long-term monitoring network, a UWSN is deployed on the basis of cluster structure. Each cluster consists of a cluster head (CH) and several cluster members (CMs). After forming a cluster, a CH is responsible for collecting data from its CMs and transmitting the data to the sink/BS. Because CHs and aggregators have very similar behaviors, data aggregation is an important technique in cluster-based UWSNs. Focusing on data aggregation in cluster-based UWSN, our previously published paper [7] evaluated four similarity functions (Euclidean’s distance, cosine distance, Jaccard’s distance, and Hamming distance). In the paper, we found normalized thresholds for each function in order to determine which similarity function is the best choice for data integration in cluster-based wireless sensor networks. This evaluation concluded that the Euclidean and cosine distances can help increase the effectiveness of cluster-based UWSNs by reducing the packet size and minimizing data redundancy sent to the sink/BS.

2. Related Research

Researchers in [16] presented work on data aggregation for periodic sensor networks using similarity functions between sets. The authors stated that they were the first to use these functions for data aggregation in sensor networks, and they provided a new prefix filtering method to study set similarity in sensor networks. Their method has two phases: the first one is performed at the node level and is called local aggregation while the second one is performed at the aggregator level using the Jaccard similarity function. Their results show that their approach reduces data size by eliminating in-network redundancy and sending only necessary information to the sink.

An overlay protocol for duplicate sensitive aggregation functions that aggregates partial results with no computation error in a highly energy-efficient manner was presented in [10]. This protocol aggregates data in two layers, the routing layer and the data aggregation layer. It uses four phases to collect data from sources: creating a routing tree structure, finding proper data aggregation nodes, creating signatures, and data collection. Their results reveal that the proposed protocol outperforms other existing ones (OPAG and TAG) in terms of energy consumption and data accuracy.

Researchers in [12] proposed new algorithms for evaluating set-similarity joins, namely, PARTENUM and WTENUM. Their algorithms handle a large subclass of set-similarity predicates allowed by the definition of the set-similarity join operator. The subclass involves standard set-similarity measures such as the Jaccard and Hamming distances. Thus, the algorithms guarantee that two highly dissimilar sets will not appear as a candidate pair with a high probability. The authors also compared their algorithms to other algorithms and demonstrated the effectiveness of their algorithms through experimental evaluation on real and synthetic data sets.

3. Similarity Functions in Cluster-Based UWSN Data Aggregation

In UWSNs, a node’s position is critical. At every single position, a node periodically captures phenomena and sends the captured data to the aggregator. Data aggregation is one of the key processes of the aggregator. By eliminating data redundancy, it decreases not only the energy consumption of the overall network but also reduces the packet size being transmitted to the sink/BS.

In data aggregation, each aggregator collects and stores a set of measured data as a vector at a certain time. Then the aggregator identifies pairs of sets whose similarities are above a given threshold. Hence, applying a similarity function is a promising approach for the aggregator. A similarity function uses a threshold to decide how similar two compared data are. To further the goal of minimizing network consumption and the size of data packets, we apply similarity functions to aggregators. If the compared data are found to be similar to each other, the aggregator does not need to transmit all sets of data to the sink/BS.

The main responsibility of an aggregator is to collect sensed data from neighbor nodes, store collected data, compare between two sets of data (the current and the new data sets) using similarity functions, and transmit data to
the sink/BS. In order to do the comparison, an aggregator stores the collected data as a vector in order of the neighbor nodes. An aggregator applies a similarity function to compare the similarity between two data sets. If the two data sets are concluded to be very similar, the aggregator transmits only one data set instead of both to the sink/BS. Otherwise, it forwards all data to the sink/BS.

Figure 1 shows the behavior of an aggregator. Let \( u, v \) be two sets of collected data where \( u = \{u_1, u_2, \ldots, u_n\} \) is a set of previously collected data, \( v = \{v_1, v_2, \ldots, v_n\} \) is a new data set, and \( n \) is the number of neighbor nodes. The function \( SF \) can be any similarity function; we briefly discuss here the Euclidean distance and cosine distance. There are other similarity functions that can be used for set comparison, such as edit distance, Euclidean's distance, cosine distance, Jaccard's similarity, and generalized edit distance. However, researchers in [7] showed that the Euclidean distance and cosine distance are two appropriate similarity functions for UWSNs.

In this section, we describe in detail how the Euclidean distance and cosine distance work and how each of them affects the network. The Euclidean distance measures the dissimilarity between each pair of data in the data set and is calculated by

\[
E_d = \sum_{i=1}^{n} (u_i - v_i)^2.
\]  

(1)

Thus, \( u \) and \( v \) are said to be similar if \( E_d \leq t_d \).

The cosine of the angle between two vectors is one kind of similarity. The cosine distance equals one minus the cosine of the angle between two vectors and is represented by

\[
C_d = 1 - \frac{\sum_{i=1}^{n} (u_i \times v_i)}{\sum_{i=1}^{n} (u_i)^2 \times \sum_{i=1}^{n} (v_i)^2}.
\]  

(2)

Thus, \( u \) and \( v \) are said to be similar if \( C_d \leq t_d \).

The cosine and the Euclidean distances use the collected values directly to compute the dissimilarity between pairs of data. However, the cosine distance computes the distance based mainly on the angle between two vectors, whereas the Euclidean distance calculates the straight-line distance between two vectors. This generates different values that must be scaled for comparison, and normalization is the basic method for doing this. Some research has been conducted on vector normalization for particular applications and domains [18, 19]. It is very important to understand the effect of normalization on the distance data. All vectors are scaled to have the same variation so that we can perform exact comparisons among those vectors.

The normalization formula is given by

\[
t_d' = \frac{t_d - m}{6\sigma} + \frac{1}{2},
\]  

(3)

where the variable \( t_d \) is the original distance threshold value, \( t_d' \) is the normalized distance value, \( m \) denotes the mean value, and \( \sigma \) represents the standard deviation of the pairwise distances of the data, given by

\[
\sigma = \sqrt{\frac{\sum_{i=1}^{n} (t_{di} - m)^2}{n}}.
\]  

(4)

4. Simulation Results and Analysis

In this section, we simulate the two similarity functions, the Euclidean distance and cosine distance, which have proven effective for in-network data aggregation in real UWSNs. Then we compare the results between theory and simulation in order to show again the effectiveness of the similarity functions in cluster-based UWSNs.

4.1. Simulation Environment. Our simulations were conducted in the QualNet5 simulator, and the implementations were based on underwater wireless sensor networks. Unless otherwise specified, all general parameters were set to simulate a shallow water environment, such as a network deployed 200 m below the sea, where the channel frequency is 35 KHz and the propagation speed is 1500 m/s. The low channel frequency and propagation speed are set in order to facsimile the real shallow underwater environment. Also, energy consumption parameters were set according to the special model of underwater acoustic modem, LinkQuest UWM1000 [20].

We simulated three different scenarios with different numbers of neighbor nodes surrounding an aggregator; that is, the scenarios consisted of (i) 5 neighbor nodes with 5 constant bit rate (CBR) applications, (ii) 10 neighbor nodes with 10 CBR applications, and (iii) 20 neighbor nodes with 20 CBR applications. Sensors are small devices with limited battery, so the packet size is set at small size, 64 bytes, for all CBR applications. All the sensor nodes operated at a...
fixed data rate and maximum transmission range power of 9.6 Kbps and 50 dBm, respectively. Scenario dimensions were set to 5000 × 5000 m, and the time for each simulation run was 600 s.

Table 1 lists the simulation parameters of the tests, and Figure 2 shows the general scenarios in our simulation.

In Figure 2, the two cloud shapes indicate the two wireless networks, and the dotted lines indicate the wireless links. The red sensor node is the aggregator which connects the two networks (5 nodes in (a), 10 nodes in (b), and 20 nodes in (c)). In all scenarios, the node labeled 1 indicates the sink/BS node. The direction of green arrows indicates which sensor nodes are communicating with others.

The aggregator uses two communication channels, one for communicating with neighbor nodes and another one for communicating with the sink/BS node. Assuming all sensor nodes transmit data to the sink node through the aggregator node, each neighbor sensor node has its own routing tables.

Figure 3 shows the distance thresholds suggested by [7] for aggregators. The distance threshold of the aggregator is set to 1.00 for the cosine distance, and the distance threshold is set to 0.85 and 0.80 for the Euclidean distances in scenarios (a) and (b), respectively. In scenario (c), the distance thresholds were set to 0.85 and 0.75 for cosine and the Euclidean distances, respectively. It was shown in [7] that the cosine and Euclidean distances directly use the collected data to compute the dissimilarity between pairs of data. Thus, for the denser neighbor nodes surrounding the aggregator, a smaller distance threshold is needed.

**Table 1: Simulation parameters.**

| Parameter               | Value                  |
|-------------------------|------------------------|
| Dimensions (m)          | 5000 m × 5000 m        |
| Number of neighbor nodes| 5/10/20 w/random       |
| Traffic                 | 64 bytes CBR           |
| Number of traffics      | 5/10/20                |
| Transmission power (dBm)| 50                     |
| Simulation time (s)     | 600                    |
4.2. Simulation Results. In the simulation results, we evaluate four metrics in order to show the effect of similarity functions on data packet size reduction and data redundancy minimization. Similarity functions also enhance network performance by lessening the energy consumption at the aggregator nodes. The four metrics are the following: the percentage of data sets sent to the sink/BS, the percentage of deleted data sets, the percentage of lost data, and the energy consumption of the aggregator.

The graph in Figure 4 shows the percentage of data sent to the sink node throughout the aggregator. The two columns indicate the analysis results of the cosine and Euclidean distances and the two lines indicate the simulation results. The blue indicates the results for the cosine distance and the red indicates the results for the Euclidean distance. Because the aggregator applies a high distance threshold to determine redundant data, almost all collected data from the neighbor nodes are considered similar. Thus, the aggregator transmits only one new data instead of two to the sink/BS node. This helps eliminate the redundancy as well as to reduce the packet size at the sink/BS node. It also can be seen that the results of simulations are lower than those of the analysis. The reason for this can be explained as follows: in the real underwater sensor network, there are many factors that affect the communication among sensor nodes and can cause packet loss, high propagation delay, unpredictable wireless links, in-network collisions, and so forth. However, collisions can be avoided by using underwater transmission protocols.

Figure 5 shows the percentage of data deleted before it is sent to the sink/BS node. When two data sets are concluded to be similar, the CH deletes one and sends the other to the sink/BS. Otherwise, both data sets are sent to the sink/BS. The deleted data sets consist of the duplicated data as well. Thus, the more duplicated data is deleted, the more data redundancy is eliminated. In addition, the packet size sent to the sink is reduced. The results of analysis and simulation are quite similar. This means that the Euclidean and cosine distances are well suited for the purpose of reducing redundant data that would otherwise be transmitted to the sink/BS.
The graph in Figure 6 shows the percentage data sent to the sink/BS that was lost. These percentages are computed at the end of every simulation run to get the average value. Note that we only consider a similarity function-based data aggregation process without any underwater protocol. The lost data consists of the collected data that did not arrive at the sink/BS. We did not compute the packets lost because of collisions or unreliable wireless links. These simulation results show that our approach conserves data integrity. Therefore, we conclude that our approach is, overall, a lossless process.

Figure 7 shows the results of energy consumption at the aggregator, or cluster head, with and without similarity functions. The blue line indicates the energy consumption of the aggregator without applying any similarity function. The red and green lines indicate the energy results of aggregators using the cosine and Euclidean distances, respectively. As shown in the graph, the energy consumption is reduced dramatically when the aggregator uses a similarity function. The graph also shows that the effects of the cosine and Euclidean distances on energy consumption are quite similar. This again proves that the cosine and Euclidean distances are well-suited similarity functions for data aggregation in cluster-based wireless sensor networks.

5. Conclusion

In this paper, we evaluated the percentage of data sent to the sink and the percentage of deleted data sets in both analysis as well as simulation. From the simulation, we also showed that our approach performs an overall lossless data process as well as reduces energy consumption. These metrics prove the effectiveness of the Euclidean and cosine distances on reducing the packet size and minimizing the data redundancy sent to the sink/BS. The energy at the aggregator is also reduced, resulting in the better energy consumption of the overall network. In this work, we only took into consideration data aggregation based on similarity functions without any underwater protocol. In practice, packet loss caused by collision or unreliable wireless links may affect the results of the percentage of data sent.

In our future work, we will work on the combination of similarity functions and underwater protocols (such as routing and MAC protocols) in order to implement a high performance underwater sensor network.

Acknowledgment

This work was supported by the (MKE) Ministry of Knowledge Economy (A004700008), Development of realistic sense transmission system with media gateway supporting multimedia and multidevice.

References

[1] S. Razieh, J. Sam, and K.-Z. Ahmad, “Comparison of energy efficient clustering protocols in heterogeneous wireless sensor networks,” International Journal of Advanced Science and Technology, vol. 7, pp. 27–40, 2011.

[2] K. Hyunsook, “An efficient clustering scheme for data aggregation considering mobility in mobile wireless sensor networks,” International Journal of Control and Automation, vol. 6, no. 1, 2013.

[3] G. Yang, M. Xiao, E. Cheng, and J. Zhang, “A cluster-head selection scheme for underwater acoustic sensor networks,” in Proceedings of the International Conference on Communications and Mobile Computing (CMC ’10), vol. 3, pp. 188–191, Shenzhen, China, April 2010.

[4] F. Salvá-Garau and M. Stojanovic, “Multi-cluster protocol for ad hoc mobile underwater acoustic networks,” in Proceedings of the OCEANS 2003, vol. 1, pp. 91–98, San Diego, CA, USA, September 2003.

[5] P. Wang, C. Li, J. Zheng, and H. T. Mouftah, “A dependable clustering protocol for survivable underwater sensor networks,” in IEEE International Conference on Communications (ICC ’08), pp. 3263–3268, Beijing, China, May 2008.

[6] H.-S. Kim, J.-S. Han, and Y.-H. Lee, “Scalable network joining mechanism in wireless sensor networks,” in Proceedings of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet ’12), pp. 45–48, Santa Clara, CA, USA, January 2012.

[7] T. T. M. Khoa and O. S. Hyun, “A comparative analysis of similarity functions of data aggregation for underwater wireless sensor networks,” International Journal of Digital Content Technology and Its Applications, vol. 7, no. 2, pp. 830–837, 2013.

[8] K. Maraiya, K. Kant, and N. Gupta, “Wireless sensor network: a review on data aggregation,” International Journal of Scientific and Engineering Research, vol. 2, no. 4, pp. 269–274, 2011.

[9] R. Rajagopalan and P. K. Varshney, “Data aggregation techniques in sensor networks: a survey,” IEEE Communications Surveys & Tutorials, vol. 8, no. 4, pp. 48–63, 2006.

[10] M. Ashouri, H. Yousefi, A. M. A. Hemmatyar, and A. Movaghar, “FOMA: flexible overlay multi-path data aggregation in wireless sensor networks,” in Proceedings of the IEEE Symposium on Computers and Communications (ISCC ’12), pp. 508–511, July 2012.

[11] D. Virmani, T. Sharma, and R. Sharma, “Adaptive energy aware data aggregation tree for wireless sensor networks,” International Journal of Hybrid Information Technology, vol. 6, no. 1, pp. 25–36, 2013.
[12] A. Arvind, G. Venkatesh, and K. Raghav, “Efficient exact set-similarity joins,” in Proceedings of the 32nd International Conference on Very large databases (VLDB ’06), pp. 918–929, 2006.

[13] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, “Robust and efficient fuzzy match for online data cleaning,” in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 313–324, June 2003.

[14] M. Henzinger, “Finding near-duplicate web pages: a large-scale evaluation of algorithms,” in Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 284–291, August 2006.

[15] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, “Syntactic clustering of the web,” Computer Networks, vol. 29, no. 8–13, pp. 1157–1166, 1997.

[16] J. M. Bahi, A. Makhoul, and M. Medlej, “Data aggregation for periodic sensor networks using sets similarity functions,” in Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC ’11), pp. 559–564, Istanbul, Turkey, July 2011.

[17] E. Deza and M. M. Deza, Encyclopedia of Distances, Springer, New York, NY, USA, 2nd edition, 2009.

[18] G. Qian, S. Sural, Y. Gu, and S. Pramanik, “Similarity between euclidean and cosine angle distance for nearest neighbor queries,” in Proceedings of the ACM Symposium on Applied Computing, pp. 1232–1237, March 2004.

[19] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang, “Supporting similarity queries in MARS,” in Proceedings of the 5th ACM International Multimedia Conference, pp. 403–413, November 1997.

[20] LinkQuest, http://link-quest.com/html/uwm1000.htm.