2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing

Robust and Dynamic Data Aggregation in Wireless Sensor Networks: a Cross-layer Approach

Weigang Wu[1,3], Jiannong Cao[2], Hejun Wu[1,3] Jingjing Li[2]
1. Department of Computer Science, Sun Yat-sen University, Guangzhou, China
2. Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
3. The Key Laboratory of Machine Intelligence and Sensor Networks, Ministry of Education, China
{wuweig, wuhejun}@mail.sysu.edu.cn, {csjcao, csjli}@comp.polyu.edu.hk

Abstract--**In-network data aggregation is an effective technique to reduce communication cost in wireless sensor networks. Recent works have focused on two issues individually: dynamic aggregation to handle irregular traffic of events and robust aggregation to tolerate packet losses. However, how to achieve both the objectives simultaneously is still not touched. In this paper, we propose a cross-layer approach to robust and dynamic data aggregation by making use of direct support from MAC layer. A new MAC protocol, DA-MAC is delicately designed to serve such purpose. Based on the channel contention information obtained from DA-MAC, a node can dynamically determine where and when to do aggregation. To cope with packet losses, a virtual overlay, Rings is adopted to forward one packet to multiple nodes. We have implemented our design in TinyOS based sensor networks. Performance evaluations though simulations and experiments show that, compared with existing algorithms, our proposed solution is more efficient in terms of both time and energy cost.**

Keywords--Data aggregation; event detection; cross-layer; MAC; WSN

I. INTRODUCTION

Data aggregation is a key technique to extend the lifetime of Wireless Sensor Networks (WSN) by reducing the number of bits sent and received. With data aggregation, the partial results of sensed data are combined at intermediate nodes during message routing so as to significantly reduce the amount of communication and hence the energy consumed. Simple data aggregations include the maximum or average value of the data collected by all the nodes. More advanced aggregations can be frequency counting and quantile estimation [22].

Early works on data aggregation [4][8][9] mostly target at continuous sensing (with regular data traffic) and usually rely on a static tree-like structure. Those aggregation algorithms suffer a lot from packet losses and cannot handle dynamic traffic in event detection applications. Recent researches on data aggregation have focused on these two issues, i.e. robustness and dynamics in data aggregation.

Due to the weakness of embedded device and wireless link, WSNs are prone to various failures, from node crash to transient link breakage. With aggregation, packet loss becomes more destructive because the loss of one packet may cause the loss of information aggregated from a number of nodes. Therefore, robustness is a crucial issue for data aggregation.

A recent and effective approach to achieve robustness is sketch (also called synopsis) based approximate aggregation [5][10]. A sketch is a small-size digest of the original value [5][20]. The final result can be estimated based on the sketches. With delicately designed sketches, any particular sensor reading is accounted for only once even if the sketch is delivered more than once through difference paths. Therefore, one piece of sketch can be sent to multiple parents to cope with packet losses.

However, all existing sketch based aggregation algorithms target at continuous sensing applications, where the sensor nodes transmit a stream of readings in a predefined schedule. The traffic pattern in the network is static, at least predictable, and can be easily handled in data aggregation. For event detection, another typical application of WSNs [1], there is no robust aggregation approached proposed.

The major challenges of data aggregation for event detection lie in the dynamics of data traffic. Since the event only occurs at some time instance in some specific location, data aggregation is necessary only when the event occurs and only at the nodes involved in sensing and routing the results. Moreover, packets losses due to various failures make the data traffic more dynamic. Therefore, dynamic data aggregation according to the event occurrence is desirable. However, due to the dynamic event occurrence and packet loss, when and where to do aggregation is really a challenging issue to be addressed.

Applying existing data aggregation methods for continuous sensing is obviously not acceptable. With the predefined schedule, sensor nodes will waste time and energy to listen even if there is no event at all. Additional mechanism is necessary to coordinate the aggregation operations.

Although a few solutions have been proposed for event-based dynamic data aggregation [9][12][28], they rely on tree structures and consequently cannot tolerate failures. Also, these methods have various constraints and limitations as discussed in the next section.

Motivated by these observations, we propose the first data aggregation algorithm that can achieve robustness and dynamics simultaneously by a cross-layer approach. We propose a new MAC protocol, named DA-MAC (MAC for Data Aggregation), which provides direct support for data

978-0-7695-4843-2/12 $26.00 © 2012 IEEE
DOI 10.1109/UIC-ATC.2012.33

aggregation. With the channel contention information obtained by DA-MAC, a node can dynamically determine where and when to do aggregation, so that dynamic sensing and aggregation with respect to event occurrence can be realized. Robustness is achieved by forwarding one packet to multiple nodes in MAC layer according to a virtual overlay, called Rings.

Similar to many existing MAC protocols for WSNs [2][7][21], CSMA is adopted as the basic media control mechanism and asynchronous duty cycling is used to reduce energy consumption. Each sensor node switches between sleep and wake according to its own schedule. Preamble packets are used to coordinate the sending and receiving operations among sensor nodes. Different from other MAC protocols, DA-MAC uses our delicately designed preamble handling mechanism to control data aggregation operations.

Both simulations and experiments have been conducted to evaluate the performance of our proposed approach. The evaluation focuses on the effect of the MAC protocol. We run sketch-based aggregation on top of DA-MAC and B-MAC [21] for comparison purpose. The results show that our approach can save both communication cost and time cost.

The rest of the paper is organized as follows. In Section 2, we review existing work on data aggregation and MAC protocols for sensor networks. Section 3 presents the system model and preliminary knowledge of our work. The detailed design of DA-MAC and the corresponding data aggregation algorithm is described in Section 4. Section 5 presents optimality of key parameters, which is followed by performance evaluation results in Section 6. Finally, Section 7 concludes the paper.

II. RELATED WORK

In this section, we first review existing works on dynamic aggregation and robust aggregation, and then briefly summarize existing MAC protocols for WSNs.

The first dynamic data aggregation algorithm is proposed by Zhang and Cao [28] for object tracking applications. The focus of this work is how to modify the tree over the nodes in event area, with the movement of the target. Such an approach is not suitable for event detection, where maintaining a tree is too costly. Fan et al. [9] proposed a dynamic aggregation algorithm based on the structure of a tree on directed acyclic graph (DAG). The algorithm requires that the event size cannot exceed the size of one grid, which limits the application of the algorithm. Gao et al. [12] proposed another tree based algorithm, which dynamically construct more than one tree based on the location of sensor nodes. However, this algorithm requires that each node in the boundary of the target area is connected to the sink with additional special high speed connection, which is usually unavailable.

Besides different constraints and limitations described above, a common problem of existing dynamic aggregation algorithms is the vulnerability to faults, due to the tree based aggregation structure.

Robustness has been considered in data aggregation for periodic data traffic. However, existing robust data aggregation algorithms cannot handle the dynamics of event detection applications. Robustness is usually achieved by employing multipath routing explicitly or implicitly to forward partial results. According to the aggregation approaches used, these algorithms can be classified into two categories: duplicate-sensitive and duplicate-insensitive.

In the duplicate-sensitive algorithm [18], a sensor node sends partial result to more than one parent, along a DAG. To avoid duplicates, the partial value is decomposed into fractions and each fraction is then sent to a distinct parent. However, this approach can not reduce the inaccuracy caused by packet losses [5].

Duplicate-insensitive algorithms adopt approximate aggregation functions to cope with duplicates. Sketch-based aggregations [5][10][20] make use of sketch originally proposed by Flajolet and Martin [11] for the purpose of quickly estimating the number of distinct items in a database. A sketch is small-size digests of the original value. With well designed sketches, any particular sensor reading is accounted for only once even if the same piece of sketch is delivered multiple times through different paths. The algorithms in [5][10][20] differ in the sketch design. Manjhi et al. [19] proposed an approach to combine the duplicate-sensitive (tree based) and duplicate-insensitive (multipath based) aggregations. CountTorrent [14] proposed a different duplicate-insensitive approach. Labeled aggregates ensure all values are counted only once during communication. Obviously, passing labels certainly causes additional communication cost and storage cost.

In event detection applications, due to the dynamics brought by the event occurrence and failures, existing robust aggregation algorithms will waste much time and energy when there is not data to send. To address this problem, we propose to determine when and where to do aggregation based on the channel contention status of wireless links, i.e. a cross-layer design involving MAC protocol. However, existing MAC protocols for WSNs cannot be used for our purpose.

Most existing MAC protocols are design for general node to node traffic pattern, where any node may be the sender and/or receiver. S-MAC [27] and T-MAC [6] are synchronous protocols, which negotiate a common schedule of sleep period. Asynchronous protocols such as B-MAC [21], WiseMAC [7], X-MAC [2], RI-MAC [23] and DW-MAC [24] rely on low power listening (LPL) [13], also called preamble sampling, to link together a sender to a receiver.

Our work, however, requires a MAC protocol for aggregation traffic pattern, where the source nodes send packets to sink node. Existing works for aggregation pattern [3] [15] [16] [17] focus on static all-to-one scenarios, where all the sensor nodes need to send a packet to the sink. Also, they consider delivery of individual data packets. In our work, we target at dynamic data aggregations and the packets must be sent in a preferable order, so that the data can be aggregated as much as possible. Therefore, we propose a novel MAC protocol, based on which sketch-based data aggregation can be conducted dynamically upon the occurrence of an event.

III. SYSTEM MODEL AND THE RINGS OVERLAY

Sensor network. We consider a wireless sensor network deployed in a two-dimension area to detect events specified by users. The target events occur at unknown time and in unknown regions of the covered area. Upon the occurrence of the event, a subset of the nodes, $n_1 \ldots n_k$, covering the event region can detect it and get reading values $\{v_i | 1 \leq i \leq k\}$. Such nodes are called source nodes. The readings are aggregated and delivered to the sink node, which is connected to the sensor nodes via multi-hop paths.

The sensor nodes communicate using CSMA technique. Due to the weakness of wireless links and sensor nodes, the communication channel may lose data packets from time to time. Also, the sensor nodes may fail due to exhaustion of battery power or various damages.

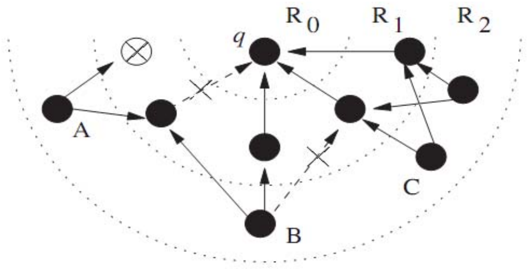


Fig. 1 The Rings Overlay

Asynchronous duty cycle. Like in all existing MAC protocols, the communication module of a sensor node works according to duty cycling so as to save energy. That is, a node switches between two states: *sleep* and *awake*. The duty cycling is asynchronous and each node has its own schedule. In the sleep period, a node monitors the channel state by using Low Power Listening [21]. It periodically does Clear Channel Assessment (CCA) to check whether the channel is busy. Fig. 2 shows a visual representation of the duty cycling of three nodes in our system.

The Rings overlay. We borrow the idea of Rings, a virtual overlay, from sketch aggregation [5][20]. The Rings is constructed by dividing sensor nodes into different rings according to hop count from the sink node, as shown in Fig. 1. The sink node q is in the ring R_0; a node is in the ring R_i if it is i hops away from q.

IV. DA-MAC BASED DATA AGGREGATION

In this section, we describe the details of our design. As mentioned before, to achieve robustness and dynamics simultaneously, we propose a cross-layer design. Data aggregation is conducted in the application level and implemented as an application in a WSN. To address the dynamics of event occurrence, we let the MAC layer determine when and where to send/receive data by designing a new MAC protocol. With the Rings overlay, a node sends one packet to multiple neighboring nodes so as to cope with packet losses.

In the following, we first present the design of DA-MAC protocol, and then describe the data aggregation algorithm based on DA-MAC. Please notice that our work focuses on the design of the aggregation algorithm and MAC protocol, the mechanism to realize interactions between the two layers is an implementation issue and not our focus.

A. Overview

Fig. 2 shows an example transmission procedure of DA-MAC. When a node s_i in the ring R_i has data to send, it first transmits a specific number of preamble packets with fixed pause between preambles.

When a node s_j in R_{i-1} wakes up and detects (by CCA) that the channel is busy, it receives the preambles from s_i. s_j will then find that the preamble is sent by a node in the outer ring and it will keep awake until the data packet is received or the preamble stops.

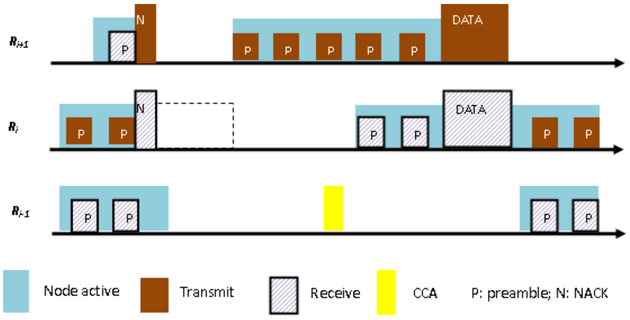


Fig. 2 Example operations of DA-MAC

Similarly, a node s_l in R_{i+1} can also detect the busy channel and receive the preamble from s_i. However, s_l do not need to wait for the data sent by s_i. Moreover, if s_l itself also has some data pending, it will send an NACK to s_i by making use of the pause between two preambles. Otherwise, s_l returns to sleep.

For s_i, if it does not receive any NACK during the transmission of preambles, it will transmit the data following the last preamble. Otherwise, s_i stops transmitting preambles turns to sleep, because some node in outer ring has data to send and s_i needs to wait for such data.

B. Detailed Operations

A sensor node may take the roles of sender or receiver at different time. Please notice that our protocol operates on top of CSMA, which acts as the basic media control mechanism, and we do not show the operations of CSMA itself for clarity purpose. Our protocol operates as follows.

Transmission of Preamble. Like X-MAC, DA-MAC also uses short preamble series. Each short preamble packet contains the ring ID of the sender. After the transmission of each preamble, there is a pause, during which the sender will listen to the medium. The pauses between preambles enable the nodes in the outer ring to send NACK packets.

The number of preambles in one series is determined by the sleep period of s_i's neighbor nodes in R_{i-1} and R_{i+1}. Same as in other preamble based MAC protocols [2][21], the preamble series should last longer than the sleep period of all neighbor nodes, so that each neighbor node can receive at least one preamble packet.

Upon the reception of a preamble packet, the neighbor nodes in R_{i-1} will remain awake for the remainder of the preamble series, which will be followed by the data packet.

Transmission of NACK. When a node in R_{i+1} received the preamble packet, it will check whether itself also has data to send. If the answer is yes, the node will send a NACK packet during the pause between two preamble packets. Otherwise, it discards the preamble received and turns to sleep. Similar to preamble packets, the ring ID of the transmitter will be included in NACK packets. Please notice that if two or more nodes simultaneously want to send NACK, with the underlying CSMA technique, at most one of them will obtain the channel at any moment, that is, collision is avoided by CSMA.

When the sender of preamble s_i receives a NACK from some node in the outer ring, it will stop sending preambles because s_i needs to receive data form outer ring before it sends the current data packet. Therefore, it interrupts the attempt to send data and turns to sleep.

Transmission of Data. If no NACK is sent to s_i during the transmitting of a series of preambles, there is no pending data at nodes in outer ring. Therefore, s_i can transmit the data packet after the last preamble. Please notice it is not necessary to insert pause between preamble and data because the receiver must have been waiting for the data. If the attempt of sending is interrupted by some outer ring node (i.e. NACK is received), s_i needs to wait and retry after receiving the data from outer ring.

One concern may arise due to the hidden terminal problem. Since no acknowledgement to the preambles is sent, it is possible that two hidden terminal nodes transmit data packets simultaneously. Then, a node in the transmission range of the both transmitter cannot successfully receive any data due to interference. However, we argue that, in our design, the effect of hidden terminal is minor because of the fault tolerance feature of DA-MAC. All the data packets are broadcasted, i.e. there are multiple receivers for each data packet. Even if only one node receives the data successfully, the information in the packet can still be delivered eventually. Obviously, the scenario that all the target nodes are affected by hidden terminal simultaneously may occur with a very low probability.

C. *Data Aggregation based on DA-MAC*

With the DA-MAC protocol, data aggregation becomes quite simple. Upon the occurrence of an event, the sensor nodes in the event area, i.e. the source nodes, will detect the event and collect the raw data. Then, each source node has data to be delivered to sink. Following the operations of DA-MAC, a source node s will firstly construct and broadcast preambles.

If no nodes in outer ring have data to send, s will be allowed to send out the data packet. If the sending attempt of s is interrupted by a NACK from outer ring, s will wait for the data from outer ring. Please notice that such waiting will delay the sending of the inner data but the final aggregation result will not be delayed because it can only be obtained after the sink gets all the data sensed, including those from the outermost ring.

Data aggregation is done when a data packet is received from an outer ring node, and the receiving node has pending data. The pending data may be a raw data packet or middle data of earlier aggregations.

After the aggregation, the node will try to send the data to inner ring by following the operations of DA-MAC. Since a node cannot know how many data packets will be sent from outer ring, aggregation and re-sending will be triggered upon the reception of any data packet from outer ring.

Obviously, with DA-MAC, the in-network aggregation and data sending is roughly ordered along the Rings overlay towards the sink node, although there is no predefined schedule. Moreover, the channel contention is limited among only nodes involved in the sensing and aggregation for the event. This is exactly what we want to achieve in this work.

V. PERFORMANCE EVALUATION

We conduct both simulations using ns2 and experiments using 18 (excluding the sink) TelosB motes to evaluate the performance of our proposed algorithm. Since the advantages of robust (duplicate-insensitive) aggregation over traditional (duplicate-sensitive) aggregation have been fully examined and validated in [5][20] and our major contribution lies in the cross-layer design, our evaluation focuses on the effect of DA-MAC on robust aggregation for handling dynamics in event detection. Without loss of generality, we use MAX as the aggregation function. The MAX aggregation is run on top of DA-MAC and B-MAC [21] respectively, for comparison purpose. B-MAC basically uses a preamble long enough to ensure the receiver will detect the preamble. We choose B-MAC as the comparison point because it is the only asynchronous MAC protocol suitable for broadcasting, which is the basic communication operation in duplicate-insensitive aggregation.

A. *Simulation results*

The simulation is conducted using the popular network simulator ns-2. The major parameters involved are listed in Table 1.

The event is generated with a fixed interval. The event area is set to be one fourth of the whole target area while the location of the event is randomly distributed in the target area. To tolerate packet losses, we set a timeout for packets from outer rings (i.e. the packet from lower levels to aggregate). For the settings of MAC protocols, we basically follow the settings used the X-MAC in [23].

Table 1 Parameters in the Simulation

Parameters	Values
Target area size (m^2)	1000x1000
Number of Nodes	50
Transmission/Sensing range(m)	250
Event interval (s)	50, 100, 150, 200
Event area size (m^2)	1/4 of the target area
Data packet size (Bytes)	28
Timeout for outer ring packets(s)	1, 3, 5
Timeout for aggregation (s)	50
Sleep period(s)	1
Preamble packet size (Bytes)	6
Gap between preambles	(N)ACK transmission time + SIFS + propagation delay
Duration of one series of preambles	Sleep period + preamble gap
Simulation time (s)	20 times of event interval

Especially, sleep time is fixed to one second, same as in [23]. From optimality in the previous section, the optimal sleep time is determined by the event behavior and device constants. However, obtaining such values is difficult and adopting the optimal sleep time calculated through our optimality in performance evaluation is part of our future work.

We adopt three different performance metrics, i.e. *average duty cycle, average aggregation delay* and *average relative error*. We report the simulation results according to the metrics.

1) Average Duty Cycle

Average Duty Cycle (*ADC*) is defined as the percentage of awake time over the whole simulation time. This is the metric of energy cost as used in prior works [2][24], because directly measuring the energy consumed is not feasible and not meaningful (the energy consumption of different radios varies significantly even in the same radio state).

Fig. 3 shows the results of ADC. The interval of event occurrence affects ADC significantly. The more frequently the event occurs, the more large duty cycle is needed to do data aggregation. Our proposed algorithm can always reduce duty cycle at various timeout settings. This is obviously achieved by the support of DA-MAC protocol, which can help address the dynamics of event occurrence, so that more readings can be aggregated and fewer transmissions are necessary.

Moreover, with a short event interval, our proposed algorithm can save more duty cycle, compared with the aggregation with B-MAC. This is because that, our algorithm can aggregate more readings at intermediate nodes, and consequently the duty cycle does not increase as fast as the algorithm based on B-MAC.

2) Average Aggregation Delay

Average Aggregation Delay (AAD) refers to the time between the event occurrence and the finalization of the aggregation result at the sink node. This is the metric to measure the latency of event detection. Since we set a timeout for the aggregation operation, the finalization of the aggregation result for one event occurrence is indicated by the arrival of the last valid data packet at the sink node.

The results of AAD are plotted in Fig. 4. Obviously, with DA-MAC, our proposed aggregation algorithm can reduce aggregation delay significantly, especially at the timeout value of 3s. Comparing AAD at different timeout values, we can see that our proposed algorithm can complete data aggregation with a quite stable latency but the AAD of the B-MAC based aggregation is affected significantly by the timeout value. This can be explained as follows. For B-MAC based aggregation, each node has to wait until the timeout occurs even if there is no data packet to be sent by the nodes in outer rings. With DA-MAC, the aggregation can be determined by the MAC protocol, so the timeout for packet waiting occurs only if an expected data packet is lost. This well reflects that our DA-MAC address the dynamics of event occurrence.

3) Average Relative Error

Average Relative Error (ARE) is defined as the error of the aggregation results compared with the exact value. This metric reflects the accuracy of the aggregation algorithm.

The results of ARE are shown in Fig. 5. Our algorithm performs not so good as the one based on B-MAC. This is the price to pay for saving energy and time. Both the aggregation algorithms are duplicate-insensitive and the difference comes from the underlying MAC protocol. With DA-MAC, more aggregations can be done at intermediate nodes. Then, with one packet loss, more information may be lost compared with one with less aggregation. Consequently, ARE of our algorithm is higher. However, it is important to notice that ARE for both the algorithms is quite small, ranging from 1% to 9%. Therefore, ARE of our algorithm is acceptable.

B. *Experimental results*

We implemented DA-MAC based on X-MAC in TinyOS on a network of TelosB sensor motes. The implementation was in nesC language on top of TinyOS 2.0 [25]. The code size is about 20KB and used 3KB RAM. Hence, the space for upper level complex aggregation applications is still large, since a TelosB mote has 48KB flash memory and 10KB RAM memory. Same as in simulations, we also implemented B-MAC for comparison purpose.

In the experiments, we deployed 18 sensor motes in our office building corridor and one sink mote connected to a computer. The diameter of the network is three in hop count. The sensor motes were assigned to detect whether its sensory data exceeds a given threshold. For calculate the exact results for measurement purpose, the sensory data of a node is given by a random data set generated in advance. Since it is difficult to measure aggregation latency in experiments, we adopt extra sleeping time and aggregation accuracy as metrics.

The extra sleeping time refers to the sleeping time that is especially introduced by our aggregation algorithm based on DA-MAC during the aggregation of one event occurrence, compared with the aggregation based on B-MAC. This metric directly indicates energy saving of our algorithm. Each sensor mote logs the extra sleeping time and sends this information to the sink mote together with its event detection results to avoid the interference caused by metric data collection.

The results of extra sleeping time are illustrated in Fig. 6. The event interval of each sensor mote in the experiments is two seconds. All the sensor nodes, except leaf nodes (with ID greater than 12) and the sink (Node 1) obtain extra sleeping time of about 0.5 second, which can reduce about 25% energy consumption. This obviously shows the benefit of our algorithm. DA-MAC enables the internal nodes to sleep when waiting for the larger hop nodes to report their results.

The aggregating accuracy is shown in Fig. 7. Aggregation accuracy is the total number of errors in 1000 even occurrences (epochs). An error is defined as that the difference between the aggregated result and the exact result exceeds the system defined threshold. In the 1000 epoch results, our algorithm accumulated about 40 errors and the algorithm based on B-MAC has about 57 errors.

It is interesting to see that, DA-MAC performs better than B-MAC in terms of accuracy/error, which seems contradict with the results in simulations. We think this is caused by two reasons. First, simulation is not so accurate to model the real environment system and different parameters can get much different results, especially on sensory data values, which depend on the specific values. Second, timeout values for data receiving affect the results significantly. In simulation the timeout value is much shorter than the event interval, but in the experiments the timeout is set the same as the event interval (two seconds) to cope with the clock asynchrony.

VI. CONCLUSIONS

In this paper, we have proposed the first dynamic and robust data aggregation algorithm for wireless sensor networks of event detection applications. The major objective is to carry out robust data aggregation fast with low communication cost. To tolerate packet losses, we adopt duplicate-insensitive aggregation with multi-path routing. The major challenges in our work come from the dynamics of traffic pattern caused by event occurrence and packet loss. Different from existing works, we adopt a cross-layer design. A specially design MAC protocol is proposed to provide support for upper layer data aggregation operations. Based on the channel content status, the MAC protocol help a sensor node determine when and where to aggregate and send out the data. We have constructed analysis to optimize the key parameters in our design. Performance evaluation through both simulations and experiments show that our proposed algorithm can aggregate and deliver the results faster than the duplicate-insensitive aggregation without cross-layer design and at the same time save energy consumed.

ACKNOWLEDGEMENTS

This research is partially supported by Pearl River Nova Program of Guangzhou (No. 2011J2200088), National Natural Science Foundation of China (No. 60903215), Hong Kong Research Grant Council under GRF (No. PolyU 5106/10E), Program for Doctor Degree Foundation of Ministry of Education of China (No. 200805581008), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] I. Akyildiz, S. Weilian, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, Comput. Net. 38 (4) (2002), pp. 393–422.

[2] M. Buettner , G. V. Yee , E. Anderson, R. Han, X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in Proc. of Sensys'06, October, 2006.

[3] Qing Cao, Tarek Abdelzaher, Tian He, and John Stankovic, Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection, in Proc. of IPSN'05, pp. 20–27, Apr. 2005

[4] L. Chitnis , A. Dobra , S. Ranka, Aggregation methods for large-scale sensor networks, ACM Transactions on Sensor Networks (TOSN), v.4 n.2, pp.1-36, March 2008

[5] J. Considine, M. Hadjieleftheriou, F. Li, J. Byers, G. Kollios, Robust approximate aggregation in sensor data management systems, ACM Trans. on Database Systems (TODS), v.34 n.1, p.1-35, April 2009

[6] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proc. of SenSys'03, pp. 171–180, 2003.

[7] A. El-Hoiydi and J. Decotignie. Low power downlink mac protocols for infrastructure wireless sensor networks, ACM Mobile Networks and Applications, 10(5):675–690, 2005.

[8] K.W. Fan, S. Liu, and P. Sinha, "Structure-Free Data Aggregation in Sensor Networks," vol. 6, no. 8, IEEE Trans. on Mobile Computing, Aug 2007.

[9] K. W. Fan, S. Liu and P. Sinha, Dynamic Forwarding over Tree-on-DAG for Scalable Data Aggregation in Sensor Networks, vol. 7, no. 10, IEEE Trans. on Mobile Computing, Oct 2008.

[10] Y. C. Fan, A. L. P. Chen, Efficient and Robust Schemes for Sensor Data Aggregation Based on Linear Counting, IEEE Tran. on Parallel and Distributed Systems, Jan. 2010.

[11] P. Flajolet, G. N. Martin. Probabilistic counting algorithms for database applications. Journal of Computer and System Science, pages 31, 1985.

[12] J. Gao, L. J. Guibas, N. Milosavljevic, John Hershberger, Sparse data aggregation in sensor networks, IPSN 2007: 430-439.

[13] G. Halkes, T. V. Dam, and K. Langendoen. Comparing energy-saving mac protocols for wireless sensor networks. ACM Mobile Networks and Applications, 10(5):783–791, 2005.

[14] A. Kamra , V. Misra, D. Rubenstein, CountTorrent: ubiquitous access to query aggregates in dynamic and mobile sensor networks, in Proc. of Sensys'07, Nov., 2007

[15] Abtin Keshavarzian, Huang Lee, and Lakshmi Venkatraman, Wakeup Scheduling in Wireless Sensor Networks. In Proc. of MobiHoc'06, pp. 322–333, May 2006.

[16] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data

Gathering in Wireless Sensor Networks. In Proc. of IPDPS'04, Apr. 2004.

[17] Gang Lu, Narayanan Sadagopan, Bhaskar Krishnamachari, and Ashish Goel. Delay Efficient Sleep Scheduling in Wireless Sensor Networks. In Proc. of Infocom'05, pp. 2470–2481, Mar. 2005.

[18] S. Madden, M.J. Franklin, J. M. Hellerstein , Wei Hong, TAG: a Tiny AGgregation service for ad-hoc sensor networks, ACM SIGOPS Operating Systems Review, v.36 n.SI, 2002

[19] A. Manjhi, S. Nath, P. B. Gibbons, Tributaries and deltas: efficient and robust aggregation in sensor network streams, in Proc. of SIGMOD'05, June, 2005

[20] S. Nath , P. B. Gibbons, S. Seshan, Z. Anderson, Synopsis diffusion for robust aggregation in sensor networks, ACM Transactions on Sensor Networks (TOSN), v.4 n.2, p.1-40, March 2008

[21] J. Polastre, J. Hill, and D. Culler, Versatile low power media access for wireless sensor networks, In Proc. of SenSys'04, pp. 95–107, November 2004.

[22] R. Rajagopalan and P. Varshney, Data aggregation techniques in sensor networks: a survey, IEEE Commun. Survey Tutorial 8 (4) (2006), pp. 48–63.

[23] Yanjun Sun, Omer Gurewitz , David B. Johnson, RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, in Proc. of Sensys'08, November 05-07, 2008

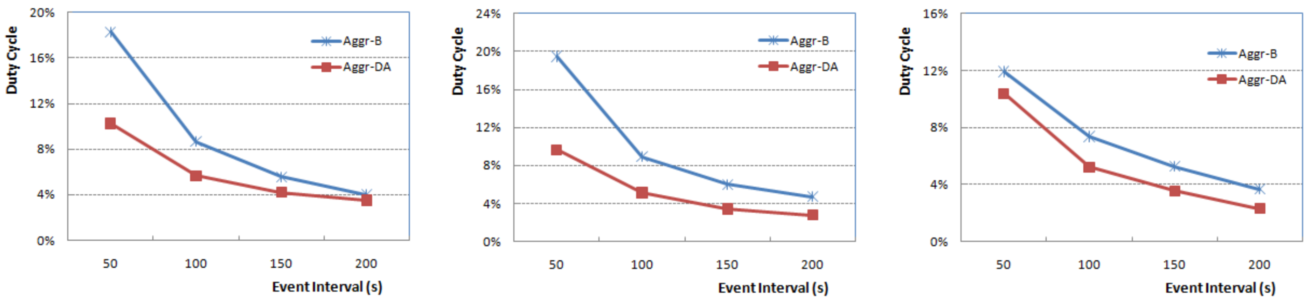
[24] Yanjun Sun, Shu Du, Omer Gurewitz, David B. Johnson, DW-MAC: A Low Latency, Energy Efficient Demand Wakeup MAC Protocol for Wireless Sensor Networks", in Proc. of MobiHoc'08, pp. 53-62, May, 2008.

[25] TinyOS2.0. http://www.tinyos.net, 2010.

[26] K. Y. Whang, B. T. Vander-Zanden, and H. M. Taylor. A linear time probabilistic counting algorithm for database applications. ACM Trans. on Database Systems, vol. 15, issue 2, pages 208-229,1990.

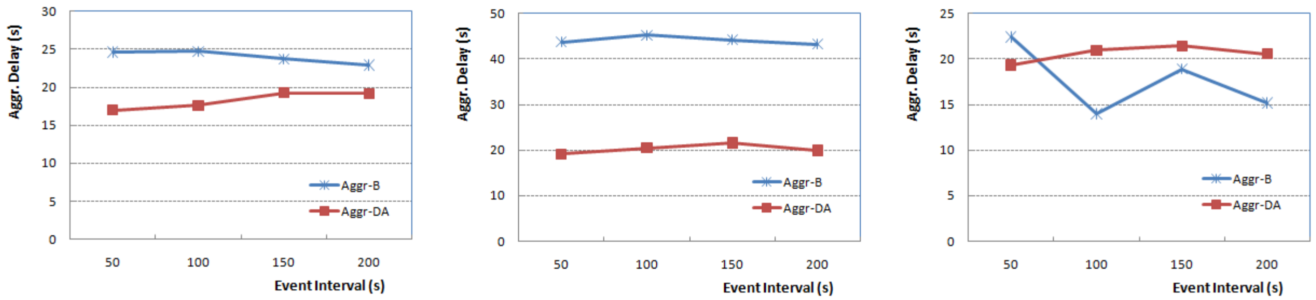
[27] W. Ye, J. Heidemann, and D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in Proc. of Infocom'02, 2002.

[28] W. Zhang, G. Cao, DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks, IEEE Trans. On Wireless Comm., vol. 3, pp. 1689-1701, Sep. 2004.



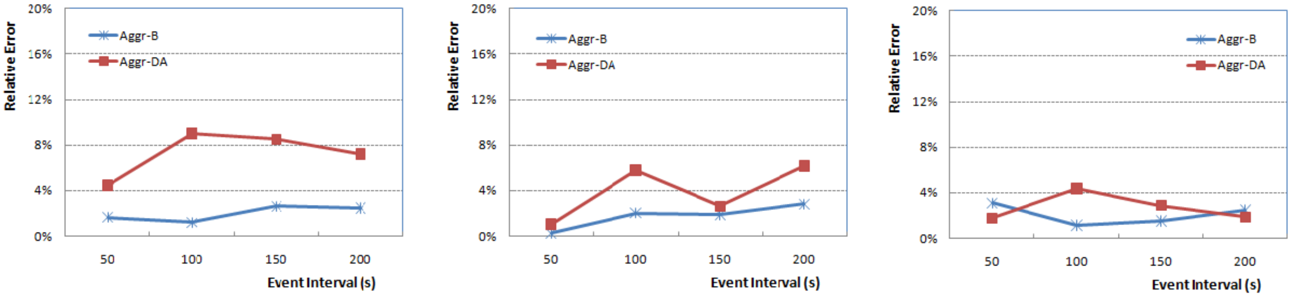
Timeout = 1s Timeout = 3s Timeout = 5s

Fig. 3 Average Duty Cycle with Different Timeout Values



Timeout = 1s Timeout = 3s Timeout = 5s

Fig. 4 Average Aggregation Delay with Different Timeout Values



Timeout = 1s Timeout = 3s Timeout = 5s

Fig. 5 Average Relative Error with Different Timeout Values

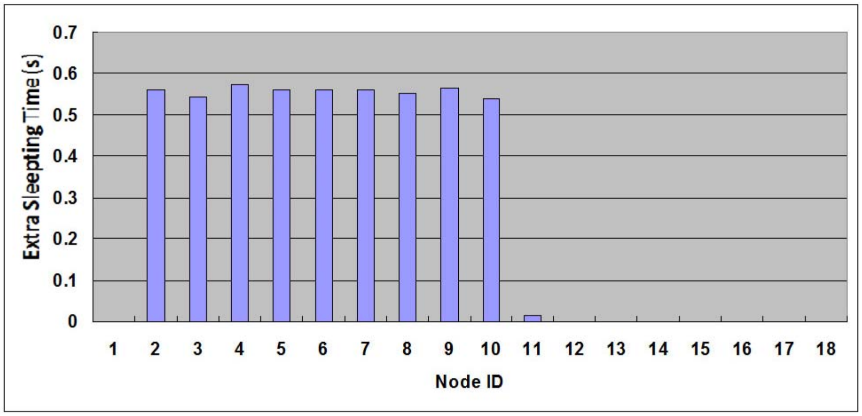


Fig. 6 Extra Sleeping Time per Epoch

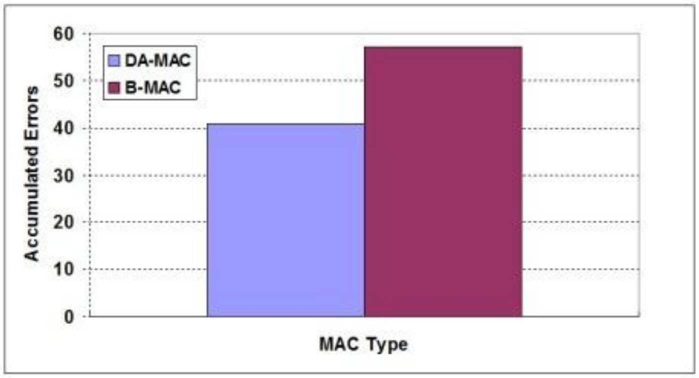


Fig. 7 Aggregation Accuracy