Packet Discarding Policies for In-Network Data Aggregation in Wireless Sensor Networks

Mario Orne Díaz-Anadón
Kin K. Leung
Electrical & Electronic Engineering Department, Imperial College, London, UK
ornediaz@gmail.com, kin.leung@imperial.ac.uk

27 January 2010
The final version of this paper will be presented in the 7th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, 5–8 July 2011

This work is funded by UK EPSRC Research Grant EP/D076838/1, entitled: "Smart Infrastructure: Wireless Sensor Network System for Condition Assessment and Monitoring of Infrastructure".

In wireless sensor networks, measurements from neighboring sensor nodes are typically cross-correlated and can be aggregated and compressed locally. This process, referred to as in-network data aggregation, saves a lot of energy by reducing the amount of data that needs to be transferred to the data sink. We consider the problem of using a TDMA schedule in order to perform data aggregation in a network in which the wireless links are unreliable and heterogeneous. In order to balance the energy consumption of different sensor nodes, the nodes with the weakest links should discard more packets than nodes with the strongest links. The existing packet discarding policies are unsuitable for data aggregation because they fail to consider the dependencies between different packets. We propose three packet discarding policies and show that they are appropriate for different kinds of networks. For large networks, among the policies that we propose, the best policy is discard the oldest packet, and to transmit the oldest packet that has not been discarded.

1 Introduction

A wireless sensor network (WSN) consists of a collection of small battery-powered devices called sensor nodes and a special node called *data sink*. The sensor nodes contain a sensing module that they use to collect information about their environment. The data sink needs to receive all the relevant information collected by the sensor nodes. Since most of the sensor nodes cannot communicate directly with the data sink, their information has to be relayed across multiple hops on its way to the data sink. The sensor nodes communicate with each other by using their wireless transceivers, which dominate their energy consumption [1].

Each sensor node measures and provides information about a certain area, which is called coverage area. From the point of view of minimizing the deployment cost, the number of nodes must be minimized, which means that there should be little overlap between coverage areas of neighboring sensor nodes. However, from the point of view of maximizing reliability of the network, the overlap between coverage areas of neighboring nodes should be big in order to detect faulty nodes and to reduce the noise through averaging.

A dense deployment increases the reliability, but also the amount of data to be transmitted to the data sink. This amount can be reduced inside of the network through a process called *in-network data aggregation* [2]. We assume that a *routing tree* [3] and a *TDMA schedule* [4] are used to organize the aggregation process. We illustrate these concepts based on the network depicted in Figure 1 as follows. Every sensor node is assumed to generate one packet per TDMA frame, and every TDMA frame consists of four timeslots. In the first timeslot, C transmits its packet to B. Then, B aggregates its packet and that of C and transmits the result in timeslot 2. In the third timeslot, node D transmits its packet to A. Then, A aggregates the data from C, B, D and itself into one packet. Finally, A transmits the result to the sink node in timeslot 4.

If there are packet losses, the aggregation process becomes more complicated because acknowledgments are necessary and some packets depend on other packets. For example, in Figure 1, the packet from B depends on the packet from C. If B does not receive packet a packet from C in timeslot 1, what should B do in timeslot 2? On one hand, if B transmits a packet without C's data, B's data reaches the data sink early. However, if B receives C's data later, no data aggregation is performed and thus energy is wasted. On the other hand, if B waits for C's packet, data aggregation is more likely, but B may have to wait for a long time, which increases the latency of the system. In fact, C may die, in which case B would have to wait indefinitely for C. To avoid this problem, B must decide the maximum waiting time for C's packets.

In addition to increasing the delay, packet losses cause unbalanced energy consumption for different sensor nodes because the nodes with weak links have to make more retransmissions and thus consume more energy than other nodes with strong links. In order to balance the energy consumption, the sensor nodes with weak links should transmit fewer packets than the nodes with strong links, or equivalently, discard more packets.

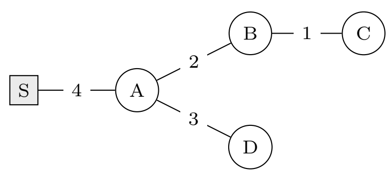


Figure 1: A TDMA schedule for data aggregation in a routing tree rooted at the data sink S.

The packets to transmit and discard are selected according to algorithms called *transmit and discard policies*.

The existing packet discard policies for data aggregation [5, 6, 7] are designed for contention-based MAC and fail to balance the energy consumption among the sensor nodes. In this paper, we propose several energy-balancing packet-discard policies for data aggregation when using a TDMA schedule. We show that the best packet-discard policy depends on the network topology.

This paper is organized as follows. Section 2 presents some related work. Section 3 presents our system model, and Section 4 describes our packet-discard policies. Finally, Section 6 concludes the paper.

2 Related work

Data aggregation is said to be *structured* if the routing tree and the transmission schedule are defined before the data transmission phase starts. The routing tree can be obtained quickly with FAT [3], and the TDMA transmission schedule can be obtained quickly with RandSched [4]. Although RandSched is not designed for networks with packet losses, it can be modified to be used in such networks by assigning the appropriate number of timeslots for retransmissions. RandSched requires an accurate prediction of this number, whereas TBSP [8] can assign new timeslots as needed. None of them balances the energy consumption of the sensor nodes.

Most packet discarding policies are designed for wired networks and cannot tolerate the latency and delay jitter of wireless networks [9]. CAPEL [9], which is designed for wireless networks, seeks to discard early any packet likely to be discarded later. However, not being designed for data aggregation, CAPEL fails to consider that some packets are more valuable than others because they contain the information from more nodes. It also fails to consider that if a packet is aggregated with a packet that was going to be transmitted anyway, its information can travel to the data sink without any increased cost.

TAG [6] and cascading timeouts [7] are two timing schemes for data aggregation. They force a node to discard packets older than a certain threshold that depends on the node's hop distance to the data sink. They assume a contention-based MAC, whereas we assume a schedule-based MAC. Their constraints on packet transmission times cause poor utilization of the channel. Furthermore, they cause uneven energy consumption among the sensor nodes.

NUM-INP [10] formulates a distributed network utility maximization problem related to data aggregation, namely the selection of each node's transmission time and compression degree in order to maximize the utility subject to power constraints. It balances the energy consumption across the sensor nodes, but only under the assumption of very flexible compression functions. For example, it assumes that flows with different data rates can be aggregated and compressed, which is often difficult to implement for some applications in practice. Additionally, it does not consider packet losses.

DPA [11] is a sensor-selection protocol. Under the assumption that some packets are necessary, it restricts the data generation tasks to a small number of sensor nodes. DPA suffers unbalanced energy consumption in networks with unreliable links because it forces the sensor nodes with weak links to consume a lot of energy.

We assume that there is no *temporal correlation*, which is the correlation between the measurements in consecutive reporting intervals. TiNA [12] exploits this correlation to reduce the data volume transmitted to the data sink.

3 System Model

3.1 Wireless Link Model

Every wireless link is assumed to have a constant transmission success probability, which we denote by p^s. Different nodes have different values of p^s. An example of a network where this probabilistic assumption is suitable is a network monitoring the vibrations of a tunnel. The sensor nodes are deployed over the lining of a tunnel. Fast vehicles traverse the tunnel and interrupt the link connectivity among some of the sensor nodes. If the traffic is ergodic and fast, it is reasonable to assume that the success of different transmission attempts are uncorrelated and that their success probability remains constant. Figure 2 shows three routing trees in which each link is annotated with the transmission success probability p^s.

We assume that DATA transmissions may fail, but not ACKs. This can be justified by the fact that ACKs are short and thus protecting them with strong error correction codes incurs little overhead.

3.2 Data Generation and Aggregation Model

Time is divided into *reporting intervals*, denoted by $\{R_1, R_2, \ldots\}$. Every node generates one packet about every reporting interval. Every packet contains information about only one reporting interval, but this information may result from combining data from multiple nodes. Any number of packets associated with the same reporting interval can be aggregated and compressed into a single packet. This aggregation model entails a very high degree of compression. Many useful aggregation functions satisfy this model, including the maximum, the mean, and the histogram [13].

3.3 Timeslot Assignments

In order to balance energy consumption among the sensor nodes, every node is assigned exactly one timeslot per TDMA frame. A sensor node can use its timeslot to transmit new packets or retransmit old packets. The nodes with the highest p^s succeed in most of their packet transmissions and thus manage to transmit information about almost every reporting interval. The nodes with very small p^s manage to transmit very few packets and use most of their timeslots for packet retransmissions. However, since every node transmits the same number of times, namely once per TDMA frame, every node consumes the same amount of energy if the transmission power consumption is much higher than the reception energy consumption. This assumption is reasonable if the distance between sensor nodes is very large [1].

3.4 Node Count and Rate Selection

We define the node count of a *packet* as the number of sensor nodes whose information is contained in the packet through data aggregation. Similarly, we define the node count of a *reporting interval* R_j as the number of sensor nodes whose information about R_j reaches the data sink. We denote this node count by c_j, which clearly depends on the success probability p^s of the links and the TDMA frame period T_f. If T_f is reduced, every sensor node is assigned more timeslots and can perform more packet retransmissions, thereby increasing c_j. We define the normalized frame period Γ as T_f/T_r, where T_r is the duration of every reporting interval. These variables are illustrated in Figure 3. If every node has a link success probability p^s larger than Γ and its buffer is sufficiently large, no packets are discarded and the node count of every reporting interval c_j is one.

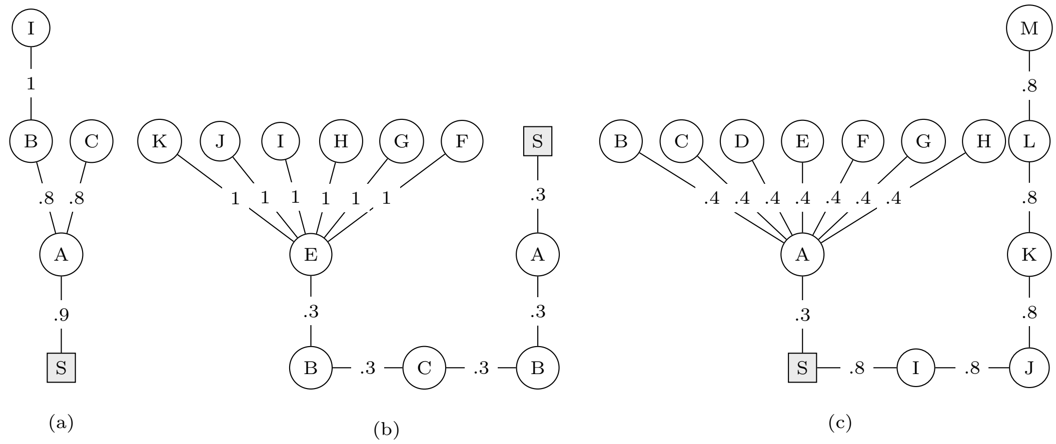


Figure 2: Three simple routing trees rooted at S. Each wireless link is marked with its transmission success probability p^s.

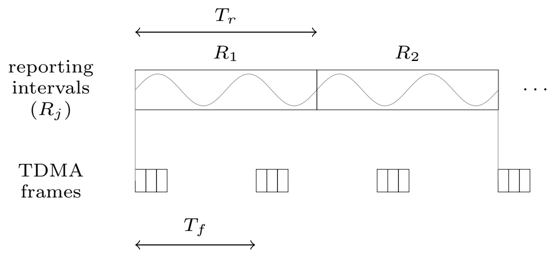


Figure 3: Every sensor node generates one packet with period T_r and is allocated one transmission attempt every T_f. Here, $\Gamma = T_f/T_r = 2/3$.

3.5 Utility Function

The utility derived by the data sink increases with c_j because a high c_j indicates that the information received by the data sink is the product of combining the data from many sensor nodes, and thus is very reliable. We define the utility U by

$$U = \frac{1}{N}\sum_{i=1}^{N} c_i, \tag{1}$$

where N is the number of simulated reporting intervals. If we reduce the normalized frame period Γ, every sensor node is assigned more transmission slots and thus discards fewer packets. However, the number of transmissions and the energy consumption also increase. We define the *utility efficiency* of the schedule by

$$\eta = \frac{U}{N}. \tag{2}$$

4 Packet discarding policies

Every sensor node transmits in its transmission timeslot and listens in its children's slots. Right before its slot, it decides which packet to transmit by executing an algorithm called *selection policy.* Every time that a node's buffer becomes full, it decides which packet to discard by executing an algorithm called *discarding policy.* Note that the packet buffer should be small not because of memory scarcity, but in order to prevent the growth of the latency and to balance the energy consumption of the sensor nodes. If the buffer are large, the nodes with low p^s have many packets in their buffers. Therefore, they transmit packets for longer than their neighbors do.

We propose three packet selection/discarding policies. Policy 1 selects the oldest packet and discards the oldest packet. Policy 2 identifies the packets with the highest node count and transmits the oldest packet among these packets; and it discards the oldest packet. Policy 3 identifies the packets with the highest node count and transmits the oldest packet among these packets; and it identifies all the packets with the lowest node count and discards the oldest packet among these packets.

5 Simulation Evaluation

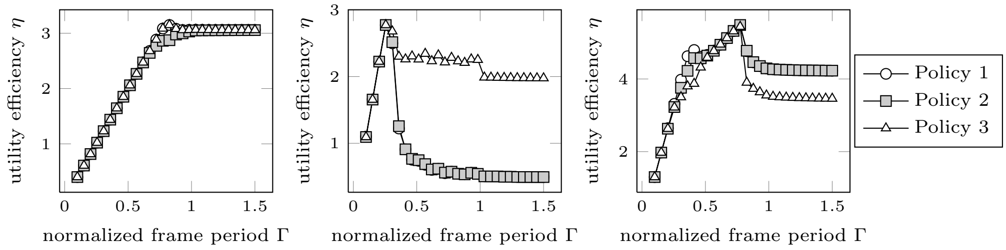
In order to evaluate the three policies, we use a custom C# simulator that is available in [14]. Each network is simulated 100 times, each time for 2000 TDMA frames and multiple values of the normalized frame period Γ between 0.1 and 0.5. The packet buffers of the sensor nodes have a capacity of 30 packets.p

5.1 Results in Three Simple Networks

First, we simulate the three networks depicted in Figure 2. The results are presented in Figure 4, which displays the utility efficiency η of the three networks as a function of the normalized frame period Γ. For small values of the normalized frame period Γ, every sensor node is assigned many timeslots to report each reporting interval, no packets are discarded, and the node count of every reporting interval is equal to the number of nodes in the network. However, energy usage is unbalanced because no packets are discarded, and most timeslots are unused, resulting in a low utility efficiency η. As a result, the sensor nodes waste energy by listening to their children when their children have nothing to transmit.

Figure 4 shows that as the normalized frame period Γ increases, the utility efficiency η increases, but only until Γ reaches a certain threshold, after which η decreases. This is because increasing Γ reduces the number of unnecessary timeslot assignments, and thus each assigned timeslot makes a greater contribution towards the utility U. However, if Γ becomes too large, η decreases because the sensor nodes discard many packets in an uncoordinated way, thereby achieving poor data aggregation.

Figure 4b shows that the selection/discarding policy greatly affects the utility efficiency η. In the network of Figure 2b, node E generates one packet with a node count of 7 in every reporting interval. Whether this packet reaches the data sink depends on the normalized frame period Γ. If $\Gamma < 0.3$, the packet buffer is large, and many TDMA frames are simulated, then every packet reaches the data sink. This is because if Γ is smaller than the minimum p^s in the network, every node has sufficient timeslots to transmit all its packets. Figure 4b also shows that 0.3 is the value of Γ that maximizes η. For $\Gamma > 0.3$, Policy 3



(a) Network in Fig. 2a (b) Network in Fig. 2b (c) Network in Fig. 2c

Figure 4: Utility efficiency η of the three networks.

greatly outperforms the other policies because it makes node B transmit the packet with the largest node count, which greatly contributes towards the utility U because its node count may be as high as 8. In contrast, Policy 1 makes B transmit the oldest packet, which may hardly contribute towards the utility U because its node count may be as small as one.

However, Policy 3 is not always the best, as shown in Figure 4c. This is because transmitting the packet with the largest node count may cause a recent packet to be transmitted, which reduces the period of time during which that packet can be aggregated with other packets. For example, in Figure 2c, suppose that A contains a packet with node count 3 associated with the reporting interval 1 and a packet with node count 4 associated with the reporting interval 2. According to Policy 1, A should transmit the packet associated with the reporting interval 1, which means that when the time comes to transmit the packet associated with the reporting interval 2, this packet's node count may have increased to 6 because A may have received two more packets associated with the reporting interval 2. In contrast, according to Policy 3, A should transmit the packet associated with reporting interval 2 first, which means that if A receives more packets associated with the reporting interval 2 later, those packets will not have been aggregated as much as with Policy 1.

Figure 4c also shows that the utility efficiency η as a function of Γ can have multiple local maxima. In Figure 4c, the local maxima lie at $\Gamma = 0.4$ and $\Gamma = 0.8$. The maximum at $\Gamma = 0.4$ is associated with the branch of node A in Figure 2c, and the maximum at $\Gamma = 0.8$ is associated with the branch of node I in Figure 2c. Therefore, the optimal normalized frame period Γ and the optimal policy may be different for different parts of the routing tree. Choosing different values of Γ for different branches of the tree has the disadvantages of unbalancing the energy consumption and making data aggregation more complex. Choosing different policies for different branches is possible, but maybe complex.

Figure 4 shows that, for the optimal Γ, which we denote by $\Gamma^\star$, the three policies achieve similar utility efficiency η. However, choosing $\Gamma^\star$ may incur some overhead, for example if Γ is obtained by testing different values of Γ with actual transmissions. Therefore, we may not operate always operate at $\Gamma^\star$ and the policies should ideally be insensitive to changes in Γ. Figure 4 shows that the relative sensitivity of the three policies greatly changes between networks, which complicates the policy choice.

5.2 Scalalability with the Network Size

In order to study the scalability with the network size, we perform the following set of simulations. We simulate a monitored area with the shape of a square with side $\bar{x}t$, where $\bar{x}$ is a parameter that we call *normalized network side*, and t is the transmission range, which we assume to be identical for all the sensor nodes. The data sink is located in the middle of one of the sides of the monitored area. For each $\bar{x}$ we simulate 50 random networks. The sensor nodes are deployed randomly within the monitored area with uniform distribution. The number of nodes is selected so that the average number of neighbors per node is 15.

The routing tree is obtained using Dijktra's algorithm using the hop distance as the cost metric. Other algorithms can be found in [2]. Each network is simulated for 30 different values of Γ, and we set $\Gamma^\star$ to the value of Γ that yields the highest utility efficiency η. Figure 5 shows the utility U and the utility efficiency η as a function of the normalized network side $\bar{x}$ for the obtained $\Gamma^\star$. According the figure, Policy 1 performs the best on average, particularly for large networks. A reason for this may be that in our simulated networks the average number of children per node in the routing tree is small.

6 Conclusions and Future Work

We have considered a data-aggregating network in which the wireless links are unreliable and the transmission energy is much larger than the reception energy. The existing data-aggregation protocols cause the sensor nodes with weak links to consume a lot of energy in retransmissions. Therefore, the energy consumption of different sensor nodes becomes unbalanced.

In order to balance the energy consumption of the network, we propose three packet discarding policies that make the sensor nodes with the weakest links discard more packets than the nodes with the strongest links. The choice of the packets to be discarded and packets to be transmitted is important because some packets contain information from multiple nodes.

We have shown that the utility derived by the data sink greatly depends on the TDMA frame period, the routing tree, and the quality of the wireless links. No single packet discarding policy performs the best for all routing trees, and the performance differences between the policies can be significant. Therefore, we have to select the most appropriate policy according to the characteristics of the given network. This observation reveals the need for further research on the subject.

We have also simulated many random networks where the routing tree is obtained with Dijktra's algorithm. Our results show that the first policy outperforms the other two. This policy simply transmits and discards the oldest packets. In our future work, we will propose other policies and evaluate their sensitivity to the normalized frame period Γ and the algorithm used to construct the routing tree.

References

[1] K. G. Langendoen, "Medium access control in wireless sensor networks," in *Medium Access Control in Wireless*

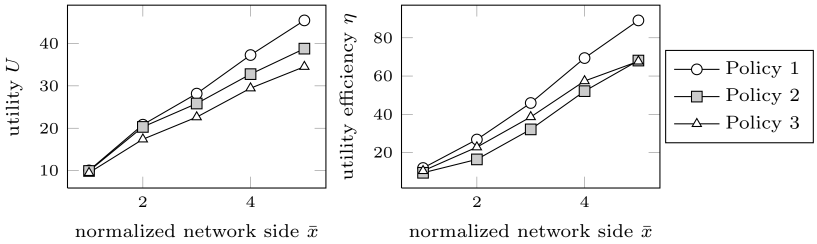


Figure 5: Utility U and utility efficiency η in random networks

Networks, Volume II: Practice and Standards, H. Wu and Y. Pan, Eds. Hauppage, New York: Nova Science Publishers, May 2008, pp. 535–560.

[2] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” *IEEE Wireless Commun. Mag.*, vol. 14, no. 2, pp. 70–87, 2007.

[3] M. O. Díaz-Anadón and K. K. Leung, “Efficient data aggregation and transport in wireless sensor networks,” *Wiley J. Wireless Comm. and Mobile Computing*, 2009.

[4] ——, “Randomized scheduling algorithm for data aggregation in wireless sensor networks,” in *Proc. IEEE European Wireless Conference*, Lucca, Italy, Apr. 2010, pp. 42–48.

[5] T. Abdelzaher, T. He, and J. Stankovic, “Feedback control of data aggregation in sensor networks,” in *Proc. 43rd IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 1490–1495, Vol. 2.

[6] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: A Tiny AGgregation service for ad-hoc sensor networks,” in *Proc. ACM SIGOPS Symp. Operating System Design and Implementation (OSDI)*. Boston, MA, USA: ACM Press, Dec. 2002, pp. 131–146.

[7] I. Solis and K. Obraczka, “The impact of timing in data aggregation for sensor networks,” in *Proc. IEEE Int’l Conf. Comm. (ICC)*, vol. 6, 2004, pp. 3640–3645.

[8] M. O. Díaz-Anadón and K. K. Leung, “A test-based scheduling protocol (TBSP) for periodic data gathering in wireless sensor networks,” in *3rd International Workshop on Multiple Access Communications (MACOM)*, A. Vinel, Ed., Lecture Notes in Computer Science. Barcelona: Springer, Sep. 2010, pp. 25–35.

[9] C.-W. Yuen, W.-C. Lau, and O.-C. Yue, “CAPEL: A packet discard policy for real-time traffic over wireless networks,” *Proc. IEEE Int’l Conf. Comm. (ICC)*, pp. 158–163, Jun. 2007.

[10] S. Eswaran, M. P. Johnson, A. Misra, and T. F. La Porta, “Adaptive in-network processing for bandwidth and energy constrained mission-oriented multi-hop wireless networks,” *Springer Lecture Notes in Computer Science*, Jun. 2009.

[11] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, “Connected sensor cover: self-organization of sensor networks for efficient query execution,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 56–67, Feb. 2006.

[12] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, “Balancing energy efficiency and quality of aggregate data in sensor networks,” *J. Very Large Data Bases (VLDB)*, vol. 13, no. 4, pp. 384–403, Dec. 2004. [Online]. Available: http://dx.doi.org/10.1007/s00778-004-0138-0

[13] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, “Medians and beyond: new aggregation techniques for sensor networks,” in *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys)*, Nov. 2004, pp. 239–249.

[14] M. O. Díaz-Anadón, “C# implementation of lossy data aggregation,” *http://github.com/ornediaz/wsnpy/blob/master/per4/net4.cs*, 2010.