Broadening the Concept of Aggregation in Wireless Sensor Networks

Eli De Poorter Stefan Bouckaert Ingrid Moerman Piet Demeester
Ghent University - IBBT, Department of Information Technology (INTEC)
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium
Email: eli.depoorter@intec.ugent.be

Abstract

Energy is scarce in sensor nodes. Therefore wireless sensor networks aim to transmit as few packets as possible. An often used technique is to aggregate data coming from multiple sensor nodes into a single packet. In this paper, we broaden this approach so that control information can also be sent together with other (control) information. The benefits of this broadened aggregation approach are discussed, and we give several examples of this generalized aggregation view. We also provide a classification for different aggregation techniques. Furthermore, we define an architecture that supports some of the techniques that were proposed and prove mathematically that even simple techniques can significantly reduce the number of sent packets. Finally, several open issues are identified, and future research directives are given.

1 Introduction

Wireless sensor networks (WSNs) are used to monitor or automate large - often inaccessible - areas [3]. Since sensor nodes are mostly battery powered, many research efforts have focused on increasing the network lifetime by reducing the energy usage at protocol level. The transmission of packets accounts for a very large part of the required energy [2], therefore several efforts have been made to reduce the number of needed transmissions through aggregation.

The aggregation techniques for sensor networks focus on reducing the amount of data traffic [6, 13]. They use intermediate nodes to collect data from the application layers of different nodes. There, the application data is processed and aggregated into a single packet.

In this paper, we broaden the concept of aggregation towards 'global aggregation'. Limiting the concept of aggregation to data ('data aggregation') does not result in optimal energy savings. We expand the concept so that control packets from different layers can also be aggregated.

As an example, consider the situation where a routing

protocol wants to send its remaining energy to its neighbors, a clustering protocol wants to advertise itself as clusterhead and a MAC protocol wants to advertise its sleeping scheme. Sending this information in a single transmission reduces the energy usage. Furthermore, reducing the number of transmissions also helps to avoid interference, resulting in better network performance.

This approach is especially suited for wireless sensor networks, due to the importance of energy conservation. Moreover, now is an opportune time to suggest conceptual improvements for WSNs, since no standard architectures for wireless sensor networks are yet agreed upon.

In Section 2, we give a definition of global aggregation and highlight its most important concepts. Next, in Section 3, we discuss related work and show how our work differs from existing work. Section 4 gives a classification method for global aggregation, which is meant to provide ideas for future aggregation techniques, and as a way to classify current and future aggregation methods. Next, in Section 5, we illustrate the additional value of using global aggregation, by presenting a conceptual architecture that implements some of the simpler techniques that were proposed. The mathematical analysis in Section 6 proves that even simple techniques result in a performance gain, reducing the number of saved transmissions further when more protocols are combined. Finally, after listing future directions in Section 7, the paper is concluded.

2 Definition

We define global aggregation as follows:

"Global aggregation covers all methodologies that offer architectural support for the intelligent gathering, combining and processing of information from any information source in a node or network. The main objective of global aggregation is to reduce the number of transmissions, thereby increasing the network lifetime."

This definition contains three key concepts, indicated in bold, which are discussed below in greater detail.

2.1 ...'from any information source'...

The main innovation of global aggregation is that any information source is considered for aggregation. This includes control and application information from all layers of the stack and from any node. Since all information exchanges are considered, the number of transmission can be strongly reduced.

2.2 ...'gathering, combining and processing'...

There are various ways to reduce the number of transmissions (see also Section 4). **Gathering** refers to methods for collecting the exchanged information. This includes all collection methods, regardless whether the information comes from different information sources within a node, or whether information coming from other nodes is collected in an intermediate node for routing purposes. **Combining** refers to the act of joining packets together from different information sources. **Processing** refers to the adaptation of the exchanged information.

2.3 ...'architectural support'...

In our vision, global aggregation should be implemented transparently and protocol independent. The system should provide a uniform interface towards the protocols, but the protocols should not require any knowledge of the exact global aggregation implementation. This ensures that no dependencies exist between the protocols and the global aggregation mechanism. Thus, implementations of the different protocols can easily be changed, similar to the independence of the different layers in the OSI-reference model. For this reason, we claim that global aggregation should not be implemented as an extra network protocol, but should be a part of the system architecture.

3 Related work

Currently, two related paradigms are in use: packet combination in Wireless LANs (to decrease the control overhead) and data aggregation in wireless sensor networks (to decrease the number of data packets).

3.1 Packet combination in Wireless LANs

Wireless LANs are suffering from a large MAC and PHY control overhead. In an attempt to decrease this overhead, several authors have investigated the use of aggregation for wireless LANs. The authors of [5] present an analytical

framework for estimating the performance of WLAN aggregation schemes. They also list four different types of aggregation techniques. Firstly, there are the IEEE802.11e [1] and similar block ACK schemes, in which an ACK is only sent after a group of data frames is received correctly instead of on a per packet basis. Secondly, some techniques expand the block ACK scheme by reducing the 802.11 short inter-frame spacing (SIFS), allowing frames to be aggregated over less time. A third aggregation technique combines MAC frames by separating the packet payload from its MAC headers, creating a new, larger packet with a single compressed MAC header. This technique is suitable for MAC frames destined to a single receiver. A fourth technique combines IEEE802.11 frames at the PHY layer while retaining the original MAC headers.

While the first two techniques are specifically designed for use with IEEE802.11, MAC and PHY aggregation can also be used in sensor networks.

3.2 Data aggregation in WSNs

In many wireless sensor networks, **measured data** is highly correlated. This data can be aggregated in order to reduce the number of packet transmissions. The different techniques are often categorized according to their networking approach [6, 13]. *Cluster based data aggregation* uses cluster heads which collect the data from surrounding neighbors. The cluster head performs local aggregation and sends the digest to the sink. Typical examples are LEACH [8] and COUGAR [7]. A second approach uses *aggregation trees* where data is aggregated while it is being routed over the aggregation path. Typical examples are Directed Diffusion [9] and TAG [10]. Finally, some aggregation schemes such as Synopsis Diffusion [12] use *multiple aggregation paths* to send the aggregated result.

3.3 Joint design of several layers

Most authors agree that sensor networks can profit strongly from cross-layer design [11]. An example of the cross-layer approach is the joint design of several layers at once. Sometimes, this joint design is used to define common packets that can be used by more than one layer. For example, in [15], a 'final destination address' field is added to the RTS and CTS frames, thus combining the routing and MAC algorithms.

The jointly designed packets contain information from several layers, resulting in a reduction of the number of required transmissions. However, joint design introduces several major disadvantages: (i) several dependencies between the different layers are introduced, (ii) the jointly-optimized protocols are no longer compatible with the original protocols and (iii) the approach is not scalable towards a large number of protocols.

Therefore, we claim that a general form of aggregation should be an architectural design issue, which is preferably transparently applicable to all layers without introducing incompatibilities.

3.4 Beyond the state of the art

The concept of global aggregation aims to combine and broaden the existing concepts listed above, thus overcoming their limitations. In particular, the following conceptual limitations are inherent to packet combination and data aggregation:

Limitations of **packet combination** in WiFi networks:

- Packets are only combined, the information they contain is not inspected nor processed.
- The concept is limited to communications with neighboring nodes.
- Packet combination is not implemented in the system architecture but relies on the MAC/PHY protocol.
- Techniques are often based on broadcast packets, and are thus not applicable in sensor networks with asynchronous sleeping schemes.

Limitations of **data aggregation** in sensor networks:

- Data aggregation is not implemented in the system architecture but in a dedicated protocol.
- Aggregation is linked to the routing protocol.
- Data aggregation is limited to the measured 'data'. It does not include other exchanged information such as control messages.
- Data on the same node that originates from different applications is not aggregated in one packet.

In conclusion, data aggregation and packet combination can still be considered a part of global aggregation. However, the concept of global aggregation aims to combine the advantages of both approaches and at the same time apply them to all layers of the network. The concept of global aggregation is thus much broader in scope.

4 Classification

Since the concept of aggregating information from all layers is new, we propose a classification of future concepts and algorithms. Three aspects are discussed: the *aggregation technique*, the *networking type* and the *representation of the information*.

4.1 Aggregation techniques

The main purpose of aggregation is to reduce the amount of data that is sent over the network.

4.1.1 Packet combination vs packet fusion. When combining several information packets in one packet, two main approaches are possible.

(i) '*Packet combination*' combines entire packets (including their headers) into a new aggregated packet. This approach is very suited for broadcast protocols with over-hearing, since each neighboring node can extract the relevant packets from the aggregated packet.

(ii) '*Packet fusion*' combines only the payload from the packets (without their headers), resulting in smaller packets. However, a new, shared header must be created. This approach is best suited for end-to-end aggregation of information towards a single destination.

4.1.2 Information merging. Packet fusion combines the *payload* from different packets and adds a new common header. However, a higher compression ratio can be obtained by combining not the payload but only the *information* it contains. This way, similar information coming from different protocols or nodes can be processed and merged together, resulting in a higher compression ratio. This technique is being used for measured data in sensor networks, but can (and should) be extended towards general information exchanges.

For example, consider a typical management use case where sensor nodes send reporting messages to a central terminal. Information such as the remaining energy or the packet loss ratio from each node can be merged together using a merging function. This merging function can be a very simple mathematical function, such as *max*, *min* or *average*, or it can be a very complex algorithm which is either *lossy* (not all the original information can be reconstructed) or *lossless* (retaining all original information).

4.1.3 Duplicate sensitive vs. insensitive. There will be situations where a node receives the same information several times. For example, multiple protocols on a single node may exchange the same information with neighbors. Or, an intermediate node may receive multiple copies of the same information from different nodes. If the aggregation method filters duplicates and only forwards the most recent information, it is called '*duplicate sensitive*'. If not, it is called '*duplicate insensitive*'.

4.1.4 Deterministic vs. stochastic aggregation. This distinction relates to whether the system can aggregate information which is generated at times which are well known in advance ('*deterministic*'), or information which is generated at arbitrary times ('*stochastic*'). A typical example of the deterministic information is periodic traffic such as routing info or keep-alive messages.

Figure 1. Predetermined (end-to-end) vs dynamic (hop-per-hop) aggregation

4.2 Networking type

An important design issue is whether to consider only information which is exchanged with neighboring nodes, called 'single-hop aggregation', or also information which is destined to remote nodes, called 'multi-hop aggregation'.

Single-hop aggregation. Since most information packets are sent to neighboring nodes, combining these results in profound energy savings. Since no routes to remote nodes must be considered, this results in a simple aggregation system.

Multi-hop aggregation. More energy savings can be expected when information packets that are destined for remote nodes are also aggregated. However, this requires the set up of aggregation paths, resulting in more complex algorithms and additional transmissions. As such, there is a trade-off between the energy saved and the additional energy required. This approach is most beneficial if communication frequently happens between a small set of nodes.

An additional classification into 2 subcategories can be made (cf. Figure 1). When using (i) *predetermined multi-hop aggregation*, only information destined to the same end node is combined. When using (ii) *dynamic multi-hop aggregation*, information is combined per next-hop destination. As such, information for different end-destinations can temporary be sent together over several links before their paths split again.

4.3 Information representation

The use of global aggregation can lead to new standardisation efforts. We strongly encourage the system designer to standardise the format which is used to represent the exchanged information. This ensures that information from different protocols and sources can be kept up-to-date, processed, combined and reused by participating nodes.

5 Example architecture

In this section, an architecture which supports the aggregation of parameters from different protocols will be described. It is important to note that this architecture does not implement all proposed techniques for global aggregation. In the aforementioned taxonomy, the proposed architecture can be categorized as an single-hop, deterministic, information fusing, duplicate sensitive aggregation scheme. The mathematical analysis of this architecture follows in Section 6.

The packets which need to be sent from a node, are (i) data packets received from neighboring nodes which need to be relayed to a different neighbor, (ii) control packets generated by protocols (e.g. routing, synchronization, power management, status) and (iii) data packets generated within the node by an application.

The arrival of the first type of packets cannot be predicted, although to a certain extent, long-term traffic profiling of static networks can lead to statistical arrival probabilities. In general, packets which need to be forwarded should be sent as fast as possible, as every delay in the send buffer results in an unwanted increase of the end-to-end delay of the packet. The second type of packets generally has a more periodic nature. Due to their periodic character, it is reasonable to assume that some of these packets can be delayed for a short amount of time before being sent. The third type of packets can be considered equivalent to packets of either the first or the second category as far as periodicity and tolerance for delay is concerned.

Figure 2. Node Architectures. (a) Traditional architecture. (b) Architecture with support for single-hop global aggregation.

Consider the traditional node architectures from Figure 2a. Traditionally, packets which are generated at different protocols are passed down to an output queue, and are sent to the MAC or physical layer as soon as the phys-

- 1) An empty packet is created.
- 2) While the packet is not full:
 - a) Add parameters with the same destination address
 - b) Add parameters with the same next-hop address.
- 3) Forward the packet.

Table 1. Algorithm for sending a packet when using multi-hop global aggregation

ical medium is assumed to be available. In a traditional architecture, if several packets are in the output queue, the physical layer will try to send them as soon as possible.

Figure 2b shows our proposed architecture which supports single-hop global aggregation. Instead of protocols sending their control information in a data packet directly to the output queue, the protocols subscribe to a management system which collects the control parameters in a central repository, called the *parameter space*. Parameters which need to be sent can remain in the waiting space for up to a per-parameter predefined period of time. They can then be added to data packets which are relayed through the node to the same destination. Or, if no data is sent, a new packet can be generated which combines the parameters from different protocols. Note that if several protocols at different layers make use of identical parameters, e.g. remaining energy of neighboring nodes, the same parameter is accessible by all protocols. Thus, when multiple protocols wish to send information regarding the remaining energy of the node, the parameter space sends only the most up-to-date parameter.

In short, protocols should subscribe to the parameter space and provide information regarding the acceptable delay with which parameters should be delivered. The system ensures that time-sensitive protocol parameters are delivered in a timely manner.

Since all information is transmitted to the same neighbor, the 'next-hop' field is required only once per packet. However, by adding a destination field per parameter, the algorithms can easily be updated to support multi-hop global aggregation (see Tables 1 & 2).

6 Mathematical analysis

In order to quantify the potential benefits of the presented architecture, a mathematical model is described in this section. Below, the case of single-hop aggregation is considered, where **periodic information destined for neighbors is combined**. Using a discrete time system, formulas are constructed which calculate the number of transmissions. In what follows, we will consider only information which is sent on a regular basis, i.e., every ΔT seconds. Adding non-predictable (stochastic) packets would lead to an even

- 1) Packet arrives.
- 2) For all associated parameters:
 - a) If the parameter is destined for this node:
 - i) Extract the parameter
 - ii) Write the parameter to the parameter space
 - iii) Notify interested protocols
- 3) If the packet requires further forwarding:
 - a) While the packet is not full:
 - i) Add parameters with the same destination address
 - ii) Add parameters with the same next-hop address
 - b) Forward the packet

Table 2. Algorithm for receiving a packet when using multi-hop global aggregation

Variable	Meaning
T_x	Time message type x is generated for the first time
ΔT_x	Time between two generated messages of type x
AD_x	Maximum <i>Acceptable Delay</i> for message of type x

Table 3. List of symbols.

greater improvement of the amount of saved packets, since there will be more opportunities to combine packets.

First, several important variables are defined in Table 3. Table 4 shows some typical values of these variables. Next, a simple formula calculating the number of packets sent when no aggregation is given in Section 6.1. Then, in Section 6.2, a formula calculating the required number of packets per time unit is derived, in the special case where only two protocols are combined. Section 6.3 does the same analysis in a situation where the starting time of these two protocol messages can differ. The formulas are analyzed in Section 6.4 and ultimately lead to an ILP formulation of the general problem in Section 6.5.

Assume that a first information packet is created at T_1 . This packet is repeated every ΔT_1 time units. The packet must not be sent immediately: it is allowed to remain in the data buffer until its acceptable delay AD_1 is reached. However, once this moment is reached, a new packet must be sent, and all other stored information packets from other protocols are added and sent together with it. This ensures that the information is combined in an optimal way, so that the least amount of packets needs to be sent.

Assumptions:

$$\forall \Delta T_i : \begin{cases} \Delta T_i \leq \Delta T_{i+1} \\ AD_i < \Delta T_i \end{cases}$$

Protocol	Message	Frequency	Size
MAC	Synchronization	15 sec	10 bytes
Routing	Route reinforce	3 min	15 bytes
Routing	Location information	30 sec	25 bytes
Clustering	Clusterhead signaling	60 sec	10 bytes
Clustering	Energy update	10 min	4 bytes
Application	Data measurements	Variable	Variable

Table 4. Typical messages resulting in periodical exchanges between neighbors

6.1 Original situation: no aggregation is used

In the following, we will study a simplified situation in which the information of only two protocols is combined. When no aggregation is used, the average number of packets sent per time interval, $ppt_{reference}$, is:

$$ppt_{reference} = \frac{1}{\Delta T_1} + \frac{1}{\Delta T_2} \quad (1)$$

6.2 Combining two protocols

Figure 3. Packets are being generated with a time interval 4 for the first protocol and a time interval 5 for the second protocol.

Consider Figures 3 and 4 as two examples. In Figure 3, the greatest common divisor (gcd) of the two time intervals between the generated control packets is 1, unlike in Figure 4. Mathematically, both situations are equivalent. The major difference exists in the so called *repetition period* (RP) of the sending pattern. It is defined as the amount of time between two identical sending patterns of the considered protocol messages, i.e., the amount of time units before the initial situation reoccurs. While the RP in the first situation is $\prod_{x=1}^2 \Delta T_x = \Delta T_1 * \Delta T_2$ or 20 time units, the RP in second situation is the lowest common multiple (lcm) of the cycle times ΔT_x . In the example of Figure 4, the repetition period is 12 time units: after this period two packets arrive at the same time just like they did at $T=0$.

Figure 4. Packets are being generated with a time interval 4 for the first protocol and a time interval 6 for the second protocol.

In general, the repetition period can be calculated as:

$$RP = lcm(\Delta T_1, \Delta T_2) \quad (2)$$

First, suppose that neither of the types of control packets can be delayed. The number of packets which are generated during the repetition period, in a first approximation, is equal to the repetition period divided by the period of the generated messages. However, in the observed model, the packets at $T=0$ can always be combined, resulting in a single saved packet:

$$PacketsReq = \sum_{x=1}^2 \frac{lcm(\Delta T_1, \Delta T_2)}{\Delta T_x} - 1 \quad (3)$$

Then, we will consider how many packets are saved when the control packets are allowed to stay in the waiting space for a certain period of time, called the *acceptable delay period* (AD). Applying the Chinese Remainder Theorem, one can prove (cf. addendum 9) that the arrival times will always at some point be 'skewed' compared to each other by a number of time units equal to $x * gcd(\Delta T_1, \Delta T_2)$. As an example, in the situation from Figure 3, if protocol 1 can accept a delay of 1 time unit, no message must be generated at time 4. Instead the outgoing packet is generated at time 5 where the control packet P_2 can be sent together with it. Similarly, if the acceptable delay of protocol 2 is 1 ($AD_2 = 1$), a combined packet can be sent at $T = 16$. Together with the packet combination at $T=0$, this results in 6 outgoing packets instead of 9. Similarly, it can be observed in Figure 4 that the acceptable delay of packet P_1 has to be 2 to send a combined packet at $T = 6$.

Thus, the AD needs to be at least $gcd(\Delta T_1, \Delta T_2)$ in order to result in a single saved transmission (see addendum 9). From this follows, that the number of packets saved during the repetition period equals:

AD_1	AD_2	$Period_{rep}$	PacketsReq	PPT
No aggregation		20	9	$9/20 = 0.45$
0	0	20	8	$8/20 = 0.40$
0	0	20	8	$8/20 = 0.40$
0	1	20	7	$7/20 = 0.35$
1	0	20	7	$7/20 = 0.35$
1	1	20	6	$6/20 = 0.30$
0	2	20	6	$6/20 = 0.30$
1	2	20	5	$5/20 = 0.25$

Table 5. Packets per time unit (PPT) for different acceptable delays. $\Delta T_1 = 4$, $\Delta T_2 = 5$

Figure 5. Situation 3: different starting times for the protocol cycles.

$$PacketsSaved = \sum_{x=1}^2 \left\lfloor \frac{AD_x}{\gcd(\Delta T_1, \Delta T_2)} \right\rfloor \quad (4)$$

Note that the minimum number of transmissions is limited by $\frac{1}{\Delta T_1}$, with ΔT_1 the lowest time interval of the information packets. An overview with results for the case depicted in Figure 3 is shown in Table 5.

When combining these formulas, the number of packets per time unit thus becomes:

$$\begin{aligned} ppt_{aggr} &= \max \left(\frac{PacketsReq(3) - PacketsSaved(4)}{RP(2)}, \frac{1}{\Delta T_1} \right) \\ &= \max \left(\frac{\sum_{x=1}^2 \frac{lcm(\Delta T_1, \Delta T_2)}{\Delta T_x} - 1}{lcm(\Delta T_1, \Delta T_2)}, \frac{1}{\Delta T_1} \right) \end{aligned} \quad (5)$$

6.3 Two packet cycles with different starting times

When the cycles of the control packets do not start at the same time, some adaptations have to be made to the formula. In Figure 5, again a situation is depicted where $\Delta T_1 = 4$ and $\Delta T_2 = 6$, but this time cycle 2 starts at $T=1$.

It can be verified that P1 should accept a delay of at least 1 time unit for saving a packet. A second packet is saved when the accepted delay increases with $\gcd(\Delta T_1, \Delta T_2)$. Thus, formula (4) (calculating the number of packets saved by accepting extra delay) from the previous section is no longer adequate. Instead, it should become (see addendum 9):

$$PacketsSaved = \sum_{x=1}^2 \left\lfloor \frac{AD_x + Y * (T_x - T_{(3-x)} + X * \gcd(\Delta T_1, \Delta T_2))}{\gcd(\Delta T_1, \Delta T_2)} \right\rfloor \quad (6)$$

with

$$\begin{aligned} X &= \begin{cases} 1 & \text{if } T_x < T_{(3-x)} \\ 0 & \text{if } T_x \geq T_{(3-x)} \end{cases} \\ Y &= \begin{cases} 1 & \text{if } \gcd(\Delta T_1, \Delta T_2) \neq 1 \\ 0 & \text{if } \gcd(\Delta T_1, \Delta T_2) = 1 \end{cases} \end{aligned} \quad (7)$$

When substituting this result in (5), the most general formula for 2 packet cycles is obtained.

6.4 Analysis

We derived mathematical expressions for the required number of transmissions per time unit with and without the use of aggregation. When transmitting the packets from 2 cycles without aggregation, the number of packets required per time unit is given by formula (1).

When using aggregation, this number is reduced to:

$$\begin{aligned} ppt_{aggr} &= \max \left(\frac{\sum_{x=1}^2 \frac{lcm(\Delta T_1, \Delta T_2)}{\Delta T_x} - 1}{lcm(\Delta T_1, \Delta T_2)}, \frac{1}{\Delta T_1} \right) \\ &= \max \left(\frac{\sum_{x=1}^2 \left\lfloor \frac{AD_x + Y * (T_x - T_{(3-x)} + X * \gcd(\Delta T_1, \Delta T_2))}{\gcd(\Delta T_1, \Delta T_2)} \right\rfloor}{lcm(\Delta T_1, \Delta T_2)}, \frac{1}{\Delta T_1} \right) \end{aligned} \quad (8)$$

with X, Y as in equation (7).

When comparing these formulas, we can derive several conclusions.

When the acceptable delay increases, so does the number of packets which can be sent together. Thus, when the acceptable delays increases, the average number of packets per time unit decreases, down to a minimum of 1 transmission every ΔT_1 time units.

In order to verify the above formula, we calculate the maximum number of packets saved when aggregating packets of two protocol cycles. It is expected that the use of aggregation can result in a reduction in the number of required transmissions of a factor two, i.e., when the packets can always be combined.

To obtain a maximal reduction ($\frac{ppt_{reference}}{ppt_{aggr}}$). The numerator $ppt_{reference}$ needs to be maximized and the denominator ppt_{aggr} minimized. Assuming a fixed ΔT_1 , the former occurs when $\Delta T_1 = \Delta T_2$ (since $\Delta T_i \leq \Delta T_{i+1}$). The latter occurs when the acceptable delays are high enough to obtain the minimum number of transmissions.

$$reduction_{max} = \frac{ppt_{reference}}{ppt_{aggr}} = \frac{\frac{1}{\Delta T_1} + \frac{1}{\Delta T_2}}{\frac{1}{\Delta T_1}} = 2$$

6.5 ILP formulation for multi-protocol optimization

It is possible to present exact formulas of the required number of transmissions when more than two subscribing protocols are used. However, these formulas are long, complex and do not provide any additional insights into the aggregation technique. Instead, we present an ILP formulation of the problem which can be applied to any number K protocols. Consider:

$$x_i, \quad i = 0..RP - 1 \quad (9)$$

$$\Delta T_j, \quad j = 1..K \quad (10)$$

In equation (9), $RP = lcm(\Delta T_1, \dots, \Delta T_K)$ is the repetition period for all K protocols. These protocols are each characterized by an inter packet time $\Delta T_j, j = 1..K$. The x_i variables are binaries, each representing a timeslot in which a packet can be sent. They are defined as follows:

$$x_i = \begin{cases} 1 & \text{if a packet is sent during timeslot } i; \\ 0 & \text{if no packet is sent during timeslot } i. \end{cases}$$

Then minimize:

$$\sum_{i=0}^{RP-1} x_i \quad (11)$$

Satisfying, $\forall j = 1..K, \forall l(j) = 0..\sigma_j$

$$\sum_{S_j^l} x_i \geq 1 \quad (12)$$

Minimize $\sum_{i=0}^{11} x_i$ (Formula (11))

With regards to Formula (12):

The following equations ensure that the packets from the first protocol are sent in time:

$$\begin{cases} x_0 + x_1 \geq 1; \\ x_2 + x_3 \geq 1; \\ x_4 + x_5 \geq 1; \end{cases}$$

The following equations ensure that the packets from the second protocol are sent in time:

$$\begin{cases} x_0 + x_1 \geq 1; \\ x_2 + x_3 \geq 1; \end{cases}$$

Table 6. Example ILP formulation for $\Delta T_1 = 2$ and $\Delta T_2 = 3$. $AD_1 = AD_2 = 1$

Where

$$\sigma_j = \frac{RP}{\Delta T_j} - 1 \quad (13)$$

$$S_j^l = l(j) \cdot \Delta T_j \quad (14)$$

$$E_j^l = S_j^l + AD_j \quad (15)$$

This formulation can be understood as follows: equation (14) indicates the timeslot when packet l from protocol j is available (starting to count from 0). Equation (15) indicates the latest timeslot that packet should be sent. Equation (12) then assures that during this span of timeslots at least one packet is sent, and this for every protocol j and every packet which needs to be sent. This is illustrated with an example in Table 6.

By minimizing equation (11), the amount of packets sent is minimized. The number of *packets per time unit* ('PPT') can be obtained by dividing equation (11) by the RP.

Thus, we can obtain the number of required number of packets per time for using any number of protocols. Conclusions are very similar to the case with two packet cycles. By increasing the acceptable delay, the average number of packets per time unit decreases, up to a minimum of $\frac{1}{\Delta T_x}$, with ΔT_x the lowest information interval. The maximum reduction of transmissions is equal to the number of control packet types. To summarize, global-aggregation results in more profit when more parameters need to be exchanged and when these parameters have no strict deadlines.

This result is illustrated by showing the solution to the ILP formulation for a variable number of protocols in Figure 6. The first protocol has a protocol cycle (ΔT) of 5, each additional protocol has its ΔT increased by one. As can be seen clearly in Figure 6, when using global aggregation, adding new protocols does not significantly increase the required number of packets. Once the acceptable delay

reaches 4 (one less than the lowest protocol cycle), the minimal number of packets per time unit is obtained, provided that the packet size is big enough to contain the information from all the protocols.

Figure 6. Required number of packets per time unit when using global aggregation with multiple protocols ($\Delta T_i = i + 4$).

7 Future directions

We explored the concept of inter-node aggregation between neighbors, giving both an example architecture and mathematical formulas calculating the reduction of required packets. However, our classification (see section 4) hints at several other aggregation approaches and many more can be contrived. As such, there is still ample room for *future work regarding different aggregation approaches and which technique is best suited for which situation*. In a later phase, adaptive or hybrid aggregation protocols can be developed, combining the advantages of these different approaches.

In our mathematical analysis, we neglected the influence of the size of the packets. We assumed that all available information can be aggregated in one packet. This assumption is plausible, considering the small size of data in sensor networks. However, the maximum packet size is often limited. Moreover, using big packets may not be most optimal in terms of reliability and delay. As such, the *definition of an optimal packet size for WSNs* can be an interesting research topic, similar to research which has been done for WiFi networks in [14].

The use of aggregation influences the Quality-of-Service (QoS) parameters of the aggregated information. Extra delay is introduced and the influence of packet loss increases. However, since less packets are required, the probability

of collisions may decrease. Currently, very little work exists [4] that focuses on *aggregation methods with QoS support*.

It is well known that the transmission of packets is one of the main sources of energy waste in a wireless sensor network [2]. The main objective of aggregation is to increase the network lifetime. However, much work remains to be done to *analyze how the amount of aggregated packets translates into an increase of the network lifetime*. Therefore, local energy savings should be calculated and translated into the savings on a network scale.

The above research topics tackled the design and analysis of aggregation protocols. However, using aggregation for control information influences the behavior of networking protocols: additional delay is introduced and packet losses may result in the loss of several control packets at once. Therefore, as a final open issue, we suggest to investigate the *influence of aggregation on existing network protocols*. The results of this research might even result in different design criteria for future networking protocols.

8 Conclusions

In this paper, we presented a concept in which exchanged information from all layers is aggregated and combined to reduce the number of wireless transmissions. This approach extends and combines two existing techniques: packet combination in WiFi networks and data aggregation in wireless sensor networks.

A detailed definition of global aggregation was given, and detailed classification criteria for aggregation protocols were listed. This classification serves both as an idea base for the design of future aggregation techniques and as a way to organize future protocols.

Furthermore, we described an example architecture which can be used for aggregating information which is destined for neighboring nodes. This architecture uses a very simple queuing system without complex calculations or aggregation functions, making it suitable for both WiFi and sensor networks. Nevertheless, even a simple architecture such as the one described, results in a profound reduction of the wireless transmissions. This was demonstrated through a thorough mathematical analysis, which showed that the number of packets per time unit can decrease up to a minimum of $\frac{1}{\Delta T_x}$, with ΔT_x the lowest information interval.

Finally, we realize that we have covered only the tip of the iceberg regarding this topic. Therefore, we gave a range of interesting topics which can be investigated more deeply.

Aggregating measured data is currently considered essential for obtaining a long network lifetime for wireless sensor networks. We are convinced that the same will hold true for the extension of data aggregation towards aggregation of information in general, both control and data infor-

mation. As such, we strongly encourage further research into this promising research area.

9 Addendum: Chinese remainder theorem

The Chinese remainder theorem answers the question of when a system of linear congruences can be solved. If $n_1, n_2, \dots, n_k$ are integers which are pairwise coprime. Then, for any given integers $a_1, a_2, \dots, a_k$ there exists an integer x solving the system of simultaneous congruences

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \vdots \\ x \equiv a_k \pmod{n_k} \end{cases}$$

Furthermore, all solutions x to this system are congruent modulo the product $N = n_1 * n_2 * \dots * n_k$. Hence $x \equiv y \pmod{N}$ for all $1 \leq i \leq k$, if and only if $x \equiv y \pmod{N}$.

The simultaneous congruences can be solved even if the n_i 's are not pairwise coprime. A solution x exists if and only if:

$$a_i \equiv a_j \pmod{\gcd(n_i, n_j)} \quad \text{for all } i \text{ and } j.$$

All solutions x are then congruent modulo the least common multiple of the n_i . Applied to section 6.2 this results in the following. The number of congruences is 2, since we are considering only 2 packet cycles. Since ΔT_1 and ΔT_2 are not coprime ($\gcd(\Delta T_1, \Delta T_2) \neq 1$), the Chinese remainder theory says that a solution exists only if $a_1 \equiv a_2$. Luckily, to result in a saved packet, a_1 should be equal to a_2 so that the system can be written as:

$$\begin{cases} x \equiv a_1 \pmod{\Delta T_1} \\ x \equiv a_1 \pmod{\Delta T_2} \end{cases}$$

There exists one solution x in interval $lcm(\Delta T_1, \Delta T_2)$ if and only if a_1 is divisible by the greatest common divisor of ΔT_1 and ΔT_2 . Thus, a reduction in the number of packets is obtained every time the acceptable delay is at least equal to the greatest common divisor d of ΔT_1 and ΔT_2 .

Section 6.3 is proven similarly by replacing a_1 by $a_1 + T_1$ and a_2 by $a_2 + T_2$. A solution then exists if

$$\begin{aligned} & a_1 + T_1 \equiv a_2 + T_2 \pmod{\gcd(\Delta T_1, \Delta T_2)} \\ \Leftrightarrow & \begin{cases} a_1 \equiv a_2 + T_2 - T_1 \pmod{\gcd(\Delta T_1, \Delta T_2)} \\ \text{or} \\ a_2 \equiv a_1 + T_1 - T_2 \pmod{\gcd(\Delta T_1, \Delta T_2)} \end{cases} \end{aligned}$$

Since the starting times have no influence if $\gcd(\Delta T_1, \Delta T_2) = 1$, formula (6) is made generally applicable by introducing variable Y . Finally, note that variable X in formula (6) ensures that $T_1 - T_2$ and $T_2 - T_1$ are positive.

References

- [1] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 8: Medium access control (MAC) quality of service enhancements, 802.11E-2005.
- [2] Cc2420 radio datasheet. <http://focus.ti.com/docs/prod/folders/print/cc2420.html>, Mar 2007.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, Aug 2002.
- [4] G. D. Bacco, T. Melodia, and F. Cuomo. A MAC protocol for delay-bounded applications in wireless sensor networks. In *Med-Hoc-Net*, Bodrum, Turkey, Jun. 2004.
- [5] R. R. Choudhury, A. chen, and S. Eneott. An analytical view of data aggregation in IEEE 802.11 LANs. In *Proceedings of Globecom06*, San Francisco, USA, November 2006.
- [6] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2):70–87, April 2007.
- [7] J. Gehrke and S. Madden. Query processing in sensor networks. *IEEE Pervasive Computing*, 3(1):46–55, 2004.
- [8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, Oct 2002.
- [9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.
- [10] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.
- [11] T. Melodia, M. C. Vuran, and D. Pompili. The state of the art in cross-layer design for wireless sensor networks. In *Proceedings of EuroNGI Workshops on Wireless and Mobility*. Springer Lecture Notes in Computer Science 3883, Como, Italy, July 2005.
- [12] S. Nath, P. B. Gibbons, S. S., and Z. R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *Sensys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 250–262, New York, NY, USA, 2004. ACM.
- [13] R. Rajagopalan and P. Varshney. Data-aggregation techniques in sensor networks: a survey. *Communications Surveys & Tutorials, IEEE*, 8(4):48–63, Fourth Quarter 2006.
- [14] R. Riggio, F. D. Pellegrini, N. Scalabrino, P. Li, Y. Fang, and I. Chlamtac. Performance of a novel adaptive traffic aggregation scheme for wireless mesh networks. In *Proceedings of MILCOM07*, Orlando, Florida, USA, October 2007.
- [15] C. Suh, Y.-B. Ko, and D.-M. Son. An energy efficient cross-layer MAC protocol for wireless sensor networks. In *Advanced Web and Network Technologies, and Applications. International Workshop on Sensor Networks (IWSN 2006)*, pages 410–419, 2006.

The Second International Conference on Sensor Technologies and Applications

SENSORCOMM 2008

MESH 2008: International Conference on Advances in Mesh Networks

ENOPT 2008: The Second International Workshop on Energy Optimization in Wireless Sensors Networks

UNWAT 2008: The Second International Workshop on Under Water Sensors and Systems

25-31 August 2008

Cap Esterel, France

Editors

Mathilde Benveniste

Reinhardt Karnapke

Bart Braem

Jaime Lloret Mauri

Cosmin Dini

M. Sohrab H. Monsi

Giancarlo Fortino

Sponsored by

Published by

IEEE computer society

SENSORCOMM 2008 Information

Search

Papers by Session

Papers by Author

Getting Started

Trademarks

**The Second International
Conference on Sensor
Technologies and Applications
(SENSORCOMM 2008)**

IEEE
computer
society

MESH 2008: International
Conference on Advances
in Mesh Networks
ENOPT 2008: The Second
International Workshop
on Energy Optimization
in Wireless Sensors Networks
UHWAT 2008: The Second
International Workshop on Under
Water Sensors and Systems

**25-31 August 2008
Cap Esterel, France**

Editors

Mathilde Benveniste
Bart Braem
Cosmin Dini
Giancarlo Fortino
Reinhardt Karpapka
Jaime Lloret Mauri
M. Schrab H. Monai

Product Number E3330

DMS Part Number CFP0008C-COR

ISBN 978-0-7695-3330-8

Library of Congress Number 2008928440

Copyright 2008 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.