

# Non-intrusive 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

Since energy is scarce in sensor nodes, wireless sensor networks aim to transmit as few packets as possible. To achieve this goal, sensor protocols often aggregate measured data from multiple sensor nodes into a single packet. In this paper, a survey of aggregation techniques and methods is given. Based on this survey, it is concluded that there are currently several dependencies between the aggregation method and the behavior of the other network layers. As a result, existing aggregation methods can often not be combined with different routing protocols. To remedy this shortcoming, the paper introduces a new ‘non-intrusive’ aggregation approach which is independent of the routing protocol. The proposed aggregation method is evaluated and compared to traditional aggregation approaches using a large-scale sensor testbed of 200 TMoteSky sensor nodes. Our experimental results indicate that existing aggregation approaches are only suited for a limited set of network scenarios. In addition, it is shown both mathematically and experimentally that our approach outperforms existing non-intrusive techniques in a wide range of scenarios.

*Keywords:* Wireless sensor networks, Aggregation, Architecture, Non-intrusive, Experimental evaluation

---

## 1. Introduction

### 1.1. Wireless sensor networks

Wireless sensor networks (WSNs) consist of a large number of sensor nodes that sample their environment. To limit their cost and size, sensor nodes are often resource constrained. A typical sensor node is equipped with

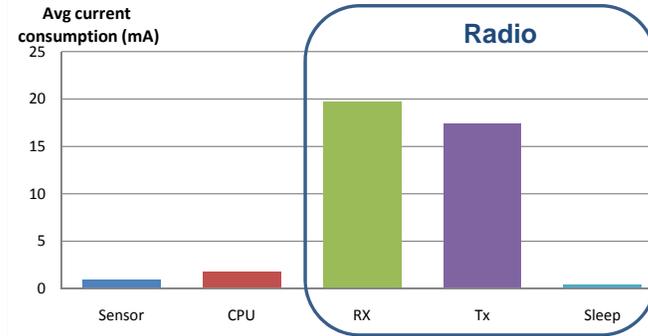


Figure 1: Average energy consumption for the TMoteSky sensor node.

a simple microprocessor with limited processing power, a small battery and a low-power radio chip for wireless communication (250 kbps or less) [1].

Traditionally, sensor networks are used to monitor large - often inaccessible - areas. They are typically used to gather information from homes, offices, nature areas, agricultural zones, oceans, or industry spaces [2, 3]. Due to their low cost, many sensor nodes can be deployed to observe a specific phenomenon. The measured data is sent to one or more remote sink devices to be processed. In large WSN deployments, data packets cannot be sent directly to the sink devices, but are routed over multiple intermediate nodes.

### 1.2. The need for aggregation

Due to the large number of sensor nodes in a single network, developers aim for a functional lifetime of several years before battery replacement. As shown in Figure 1, the radio is one of the main sources of energy consumption in a wireless sensor node [4]. To save energy, and thus obtain a longer network lifetime, MAC protocols of wireless sensor networks use sleep schemes which alternately turn on and off their radio interface [5].

To ensure that the radio can be switched off regularly, the number of packet transmissions must be low. To this end, data-aggregation protocols have been proposed which combine measured information from multiple sources in a single packet at intermediate nodes. Since less packets need to be sent, the radio can be switched more often to an energy saving sleep state. Moreover, because the number of packets is reduced, the total amount of interference and network contention is decreased.

However, as will be shown in Section 2, current data-aggregation approaches often assume that (i) the network does not contain mobile nodes, (ii) the used communication patterns are predictable (all information is sent to a limited number of sink nodes), (iii) information is gathered in predictable intervals, (iv) the overhead of the routing and MAC protocols are negligible compared to the application overhead and (v) only a single application is deployed on the sensor network. In addition, for optimization reasons, existing data-aggregation approaches are *tightly integrated into a specific routing protocol*. As a result, it is generally not possible to combine the aggregation approach with the routing protocol of your choice.

### 1.3. Beyond the state-of-the-art

In contrast to traditional sensor applications, next-generation sensor network applications are neither predictable nor static [6]. As of recently, WSNs have been used for advanced applications, such as wireless building automation [7], e-health applications [8] or voice over sensor networks [9]. The nodes in these scenarios are typically more heterogeneous: nodes with more capabilities can act as ‘actuators’ that interact with their environment [10]. Communication patterns are also more complex: information is no longer gathered by a single sink, but can be sent to any other sensor node. Moreover, in contrast with special purpose networks, multiple services are often deployed on a single network [6]. As a result, the above data-aggregation assumptions are no longer applicable.

Therefore, in this paper, an alternative ‘non-intrusive’ aggregation approach is proposed and evaluated. To support next-generation sensor applications, the proposed aggregation scheme overcomes the issues of existing aggregation schemes. More specifically, our approach:

- also copes with monitored events that occur with a non-predictable frequency;
- can be used with complex traffic patterns such as point-to-point communication;
- combines information exchanges from any network layer (rather than only application data);
- aggregates efficiently when multiple top-level services or applications are deployed on a single WSN;

- is not hindered by networks that contain mobile or failing nodes, which traditionally break fixed aggregation paths; and
- can be used when the network developer lacks time or information to manually fine-tune the aggregation settings.

#### 1.4. Remainder of this paper

In the remainder of this paper, the performance of a new non-intrusive global aggregation scheme is theoretically and experimentally analyzed. Section 1 gives an overview of wireless sensor networks, and argued that current data-aggregation approaches are only applicable in static, predictable networks that use point-to-sink communication. Section 2 supports this statement by giving a survey of existing aggregation techniques and methods. In order to enable aggregation for a wider range of scenarios, Section 3 presents an non-intrusive aggregation architecture that is suitable for next-generation sensor networks. The proposed aggregation approach is theoretically evaluated in Section 4, based on a closed-form ILP formulation. In addition, Section 5 experimentally evaluates the proposed aggregation scheme on a large-scale sensor testbed. A wide range of test scenarios is used to evaluate how our non-intrusive aggregation method performs compared to existing aggregation methods. Finally, after listing future directions in Section 6, the paper is concluded in Section 7.

## 2. Related work

To reduce the number of packets, three main approaches are used.

1. In Wi-Fi based LANs, the most popular approach is the ‘*packet combination*’ paradigm. Since the number of hops is limited in wireless infrastructure LANs, this approach is designed for small-scale networks with mainly single-hop information exchanges.
2. In contrast, wireless sensor networks most often use ‘*data-aggregation*’ approaches. These approaches assume that packet transmissions are mainly used for the sending of measured information to a far-away sink node. As such, these approaches are often optimized for large-scale point-to-sink networks.

3. Finally, to simultaneously reduce the number of packet transmissions from multiple network layers, wireless networks sometimes resort to the *joint design of multiple network layers*.

This section describes the concepts behind these aggregation paradigms and discusses the main limitations of each approach.

### *2.1. Packet combination in Wi-Fi based LANs*

Wi-Fi based LANs are suffering from a large MAC and PHY control overhead. The payload of a IEEE 802.11 Wi-Fi packet is encapsulated in a MAC and PHY header. Regardless of the data rate at which the MAC frame is sent, the corresponding PHY header is transmitted at basic rate only, leading to a sub-optimal channel usage. The RTS/CTS and ACK mechanisms add additional overhead to each packet being sent.

In an attempt to decrease this overhead, several authors have investigated the use of aggregation for wireless LANs. The authors of [11] present an analytical framework for estimating the performance of WLAN aggregation schemes. They list four different types of aggregation techniques. Firstly, there are the IEEE 802.11e [12] 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 IEEE 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 IEEE 802.11 frames at the PHY layer while retaining the original MAC headers. While the first two techniques are specifically designed for use with IEEE 802.11 networks, MAC and PHY aggregation can also be used in sensor networks.

Wi-Fi packets are in general considerably larger than packets sent by sensor networks. As a larger packet size increases the chance of transmission errors and collisions, the benefits of Wi-Fi packet combination largely depend on the channel conditions. Authors such as [13] propose the use of adaptive schemes, which optimize the aggregated packet size depending on the channel conditions. However, since the maximum packet size for wireless sensor networks is generally smaller, WSNs offer less possibilities to alter the packet size.

Packet combination approaches in Wi-Fi networks typically have the following limitations: *(i)* packets are only combined, the information they contain is not inspected nor processed; *(ii)* the concept is limited to communications with neighboring nodes; *(iii)* packet combination relies on the MAC/PHY protocol, and as such cannot be combined with different MAC/PHY layers; *(iv)* several techniques assume broadcast packets are overheard by all neighboring nodes, which is not the case in sensor networks using asynchronous sleeping schemes.

## 2.2. Data-aggregation in WSNs

Traditional WSN applications, such as environmental monitoring applications, consist of a network where a large amount of nodes gather information and send this information to one or more sinks. The measurements from different nodes are typically highly correlated, especially when the nodes are densely deployed. Data-aggregation protocols for WSNs strive to exploit this correlation by processing the *measured data* from different sensor nodes locally, before sending the resulting packet to a remote sink node.

The different data-aggregation techniques are often categorized according to their networking approach [14, 15]: how can the data be processed in intermediate nodes.

(i) *Cluster-based data-aggregation* approaches, such as LEACH [16] and COUGAR [17], select cluster head nodes which collect the measured data from surrounding neighbors. The cluster head performs local aggregation and sends the digest to the sink. To prevent the cluster head from running out of battery power, the role of cluster head is rotated regularly. Implementations differ in the way routing is executed (single-hop or multi-hop clusterheads) and in the way cluster heads are selected and rotated.

(ii) Other aggregation approaches, such as Directed Diffusion [18] and TAG [19], construct *aggregation trees*. For each node, a predetermined path is setup towards the sink. All these paths form a directed tree: the measuring nodes are located at the leaves and the sink is located at the root. Data packets traverse the directed tree towards the root. Aggregation is executed at the location where the different branches merge.

(iii) A third category of aggregation approaches uses *multiple aggregation paths*. Since a tree topology is not robust against node and communication failures, schemes such as Synopsis Diffusion [20] send the aggregated result over multiple paths towards the sink.

(iv) *Hybrid approaches*, such as Tributaries and Deltas [21], combine the advantages of cluster based and tree based data-aggregation. Depending on the network conditions, the aggregation scheme in the different regions of the network is adjusted.

(v) A different approach is taken in PEGASIS [22], which uses *chain based aggregation*. Each data packet is received from and transmitted to the nearest neighbor. By selecting the nearest neighbors, the transmission power (and energy consumption) of these transmissions is very low. Gathered data moves over the ‘chain’ of nodes: the measured data gets fused at every intermediate node and is eventually transmitted to the sink once the end of the chain is reached.

(vi) *Suppression based aggregation techniques* [23] refrain from sending information if the measured data has not changed from the last measured value, or send less data packets by exploiting the spatial correlation of the measured values.

(vii) And finally, *location-aware approaches*, such as [24], use spatial knowledge to optimize aggregation trees.

Based upon this overview, it can be concluded that current *data-aggregation* approaches in sensor networks have the following concepts in common: (i) Aggregation is not implemented as a dedicated aggregation protocol, but is tightly coupled with the routing protocol. As such, the concepts cannot directly be reused with new routing approaches. (ii) Data-aggregation is limited to the measured ‘*data*’. It does not include other exchanged information such as control messages. (iii) Data on the same node that originates from different applications is not aggregated in a single packet. (iv) Data-aggregation schemes for WSNs are designed for monitoring applications with point-to-sink traffic pattern. As such, they cannot be used for fully interactive WSN applications with point-to-point traffic.

### 2.3. Joint design of several layers

A third paradigm to reduce the number of packet transmissions is the joint design of several network layers. By combining the control overhead of different network layers, the number of control packets can be reduced. For example, in [25], a ‘final destination address’ field is added to the RTS and CTS frames, thus combining the routing and MAC algorithms. This way, joint design can be used to define common packets that can be used by more than one layer. Many authors agree that sensor networks can profit strongly from cross-layer design [26].

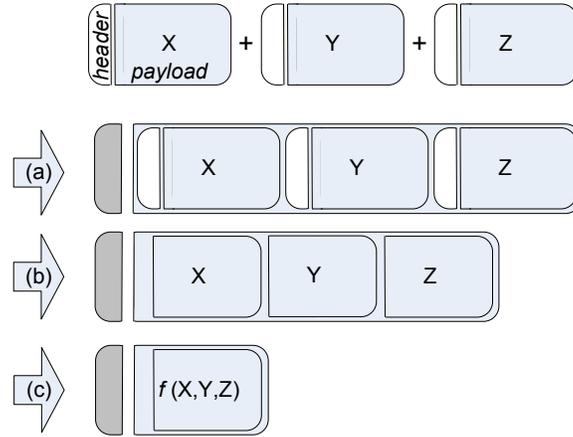


Figure 2: Original packet sequence and new packet structure after applying (a) packet combination, (b) packet fusion, or (c) information merging.

However, joint design also introduces several disadvantages. *(i)* Dependencies between the different layers are introduced, which complicates protocol design. *(ii)* Jointly-optimizing network layers results in large, monolithic code blocks that are difficult to maintain. *(iii)* It is not possible to reuse the developed protocols and aggregation methods in different protocol combinations.

#### 2.4. Packet reduction techniques

From the previous sections, it can be concluded that current aggregation paradigms typically use one of the following options to combine packets.

*Packet combination:* this approach combines entire packets (including their headers) into a new aggregated packet. The aggregated packet (with a new header) is the ‘carrier packet’ for the inner packets (Figure 2a).

*Packet fusion:* this approach combines only the payload from the packets (without any headers), resulting in smaller packets. A new, common header must be created (Figure 2b). To reduce the complexity of this common header, this approach is best suited for networks with multiple traffic flows towards a single destination node.

*Information merging:* an even higher compression ratio can be obtained by not combining 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 (Figure 2c). However, this approach requires that the aggregation technique knows (i) from which application the information originates, (ii) which function should be used to merge the information, and (iii) that the information is similar enough to be merged. The merging function can be a very simple mathematical function, such as *max*, *min* or *average*, or it can be a more complex algorithm which is either *lossy* (not all the original information can be reconstructed) or *lossless* (retaining all original information).

### 3. Non-intrusive aggregation

To overcome these limitations of traditional aggregation approaches, we argue that future aggregation approaches should be non-intrusive. To be considered non-intrusive, an aggregation approach should be (i) independent of the other network protocols, (ii) independent of the information source, and (iii) easy-to-use.

1. In our vision, the aggregation protocol should be **protocol-independent**. The aggregation mechanism should not influence the behavior of the network protocols, nor should it make any assumptions about the traffic patterns or the inner workings of the network protocols. This ensures that the aggregation mechanism can be used in combination with any network protocol.
2. In addition, an efficient aggregation scheme should **aggregate information from any source**. This includes control and application information from all layers of the stack and from any node. When all information exchanges are considered, the number of transmissions can be reduced more strongly than when considering aggregation of measured data only.
3. Finally, non-intrusive aggregation approaches should be **easy-to-use**. Application developers should not be forced to fine-tune aggregation specific parameters, such as the optimal aggregation path refresh rate. Optimizing these parameters should be part of the proposed aggregation scheme. Thus, non-intrusive aggregation should work out of the box for any network scenario.

	Custom network	General purpose network
Traffic pattern	Point-to-sink	Any
Events	Predictable	Unpredictable
# of applications	1	Varying ( $\geq 1$ )
Packet overhead	Mainly data	Any
Development cost	High	Low
Suggested aggregation	Traditional	Non-intrusive

Table 1: Traditional data-aggregation approaches are better suited for custom high-cost, high-performance point-to-sink sensor networks. Non-intrusive aggregation approaches are best suited for dynamic, adaptive or low-cost sensor networks.

In which situations are non-intrusive aggregation approaches more suitable than traditional aggregation approaches? To answer this question, we envisage two types of future wireless sensor networks: custom made sensor networks and general purpose sensor networks (see Table 1).

(i) *Custom sensor networks* are specifically designed to support one or more functions. Thus, these networks are predictable and can be highly optimized in terms of energy consumption or desired QoS. A drawback is that, due to the custom design, these networks are expensive to develop and serve only a single purpose. As such, little value is given to compatibility with other protocols. This type of network will be used for applications with stringent requirements. For these networks, traditional aggregation approaches are a justifiable solution.

(ii) *General purpose networks* are more adaptable to changing network conditions and will support a wide range of applications that are unaware of underlying network conditions. These networks will have to support multiple tasks in one network. For applications such as wireless building automation, nodes with new functionalities can be added after deployment. As such, great care should be taken to allow interoperability with existing protocols and to support changing traffic patterns. To promote the use of off-the-shelf WSNs, both the hardware and the deployment cost of these networks should be as low as possible. For these types of networks, non-intrusive aggregation approaches are better suited, being both more flexible and easier to deploy. In the next section, one such non-intrusive architecture, called ‘global aggregation’ is described.

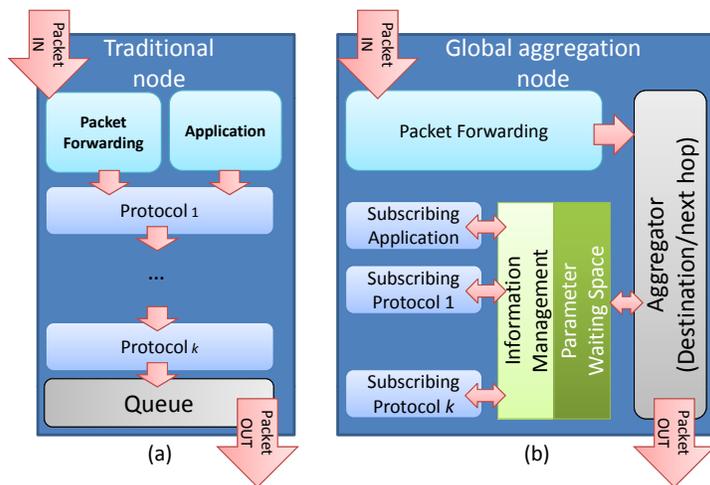


Figure 3: (a) Protocol stack based on the OSI reference model. (b) The ‘global aggregation’ architecture which supports both single-hop and multi-hop non-intrusive aggregation.

### 3.1. An architecture for global aggregation

System architectures based on the OSI reference model, as illustrated in Figure 3a, are based on the assumption that packets 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. Packets which are generated at different protocols are passed down to an output queue and are transmitted as soon as the physical medium is available.

However, in wireless sensor networks, not all packet types need to be forwarded immediately. Control packets generated by protocols (e.g. routing, power management, status information) often have a periodic character. Measurements, such as temperature or remaining battery power, typically do not vary a lot between subsequent status updates. Therefore, it is reasonable to assume that these packets are not very time-sensitive, and can be delayed for a short amount of time before being sent. Similarly, low-priority monitoring packets generated by an application often show a high tolerance for delay.

Using our global aggregation approach, depicted in Figure 3b, protocols and applications subscribe to an information management system. To exchange information, the protocols and applications do not generate packets. Instead, the information management offers a uniform API for exchanging information with a remote node. To send information, the protocol or ap-

```

call InformationExchange.sendParameter (
    SINK_NOTIFICATION, // Parameter type
    sizeof(parameter), // Size of the parameter
    &parameter,        // Location of the parameter
    BROADCAST_ADDR,    // Destination of the parameter
    QoS_level,         // Priority of the parameter
    ACCEPTED_DELAY);  // Maximum acceptable delay (msec)
                      // before packet creation

```

Table 2: Example code: the provided aggregation API is used by a sink node to send a sink notification parameter to all other nodes.

plication hands over a parameter to the information management system. The information management system is responsible for the encapsulation of the parameter: the information management system will either (i) add this parameter to a passing packet or (ii) create a new packet that encapsulates the parameter. Typical examples of parameters are:

- measured data values, such as the local temperature;
- status updates, such as the remaining battery capacity;
- or control information such as a route-request or packet acknowledges.

The example code in Table 2 shows how the API can be used to send a sink notification parameter to remote nodes.

When using our aggregation architecture, each parameter has a unique parameter type. Uniquely identifying network information has greatly aided the development of uniform network management solutions for IP based networks [27]. Similarly, standardization solutions can be used to better integrate diverging applications from different sensor networks. This uniqueness should be enforced network wide, either (i) by manually assigning each parameter a unique type, (ii) through the use of descriptive ontologies, (iii) by using standardization forums, or (iv) by using hashing methods that generate a unique parameter type based on a description.

When handing over a parameter to the information management system, the parameter priority is given, as well as an indication of the *maximum acceptable delay*. The maximum acceptable delay indicates how much delay a specific parameter can tolerate on the initial node before it should be encapsulated in a packet and sent over the network. All parameters are

collected in a central repository, called the *waiting space*. Parameters that do not need to be sent immediately can remain in the waiting space for up to the per-parameter predefined maximum period of time. As the acceptable delay increases, so does the probability that the parameter can be added to a passing packet. To prevent the end-to-end delay from becoming too high, parameters are only delayed in the waiting space of the source node: packets are not further delayed in intermediate nodes.

Algorithm 1 shows the pseudocode for processing incoming packets.<sup>1</sup>

- For each incoming packet, the architecture will first extract all parameters which are destined for this intermediate node. The extracted parameters are distributed through the information management system to all registered protocols and applications.
- If the packet contains no more parameters in the payload, it is dropped by the system. Otherwise, the routing protocol processes the packet and sets the next hop address of the packet.
- The system checks if the destination of any of the parameters from the waiting space corresponds to the destination address or the next hop address of the packet. All matching control parameters are added sequentially to the payload of the relayed packet, starting with the parameter with the nearest deadline. This process continues until the maximum packet size is reached.

A similar algorithm is executed when the maximum delay of any of the parameters from the waiting space is reached. First, a new packet is created which encapsulates the corresponding parameter. This packet is then routed to the correct next hop address, after which phase 3 of algorithm 1 is executed. This way, the system ensures that time-sensitive protocol parameters are delivered timely.

### 3.2. Implementation

For the implementation of these aggregation algorithms, the IDRA framework was used, which was developed in earlier work [28]. The IDRA framework provides a ‘packet facade’ for all interactions with packets. The packet facade is used by the information management system to create a new packet

---

<sup>1</sup>The implementation evaluated in Section 5 also supports broadcast destinations.

---

**Algorithm 1** Pseudocode for processing incoming packets.

---

```
1: // Phase 1: check all encapsulated parameters
2: for all encapsulated parameters do
3:   if parameter destination == local address then
4:     a. Copy the parameter to the parameter space
5:     b. Notify all interested protocols
6:     c. Remove the parameter from the packet
7:   end if
8: end for
9:
10: // Phase 2: drop empty packets
11: if number of remaining parameters == 0 then
12:   Drop the packet
13: else
14:   Route the packet (determine the next hop address)
15:
16:   // Phase 3: add new parameters to the packet
17:   Sort all parameters in the waiting space according to their deadline,
   starting with the most urgent parameter.
18:   for all the parameters in the sorted list do
19:     if the destination of the parameter equals the next hop address of
   the packet or the destination address of the packet then
20:       if the size of the parameter  $\leq$  unused payload size of the packet
   then
21:         Add the parameter to the packet payload
22:       end if
23:     end if
24:   end for
25: end if
26: Forward the packet
```

---

when a parameter deadline has expired, and to retrieve information such as ‘source’, ‘destination’, ‘QoS ID’ or ‘time-to-live’ from passing packets.

The packet facade from IDRA can interpret any packet type, as long as the correct packet descriptor is available [28]. Thus, the packet facade approach ensures that our aggregation architecture can request the next hop and destination address of any packet that passes through the system. As a result, our aggregation approach does not require any specific knowledge about the format of relayed packets (for example: IEEE 802.15.4, 6LoWPAN or propriety packets). This way, our implementation is not only protocol-independent, but also packet-independent. Both IDRA and the global aggregation approach are implemented using the Tinyos [29] operating system and are freely available at <http://idraproject.net>. The proposed algorithms can also be implemented on different frameworks on the condition that the next hop address and destination address of each relayed packet can be located.

### 3.3. Packet fusion versus information merging

Different protocols and applications often exchange different information types. As a result, it is generally not possible to merge different types of parameters from different protocols into a single parameter. Thus, when multiple parameters are added to a packet in our aggregation system, the parameters are stored sequentially in the payload (similar to ‘packet fusion’, see Section 2.4). However, in case parameter fusion is possible, the global aggregation architecture offers three fusion options.

1. The protocols or applications can indicate to the system that the *parameters in the waiting space* may be overwritten so that they contain only the most recent information. This option is useful when several protocols at different layers of a single node wish to send identical information to another node. For example, multiple network protocols might send a parameter that contains information regarding the remaining node energy.
2. Similarly, protocols or applications can indicate to the system that the *parameters in a relayed packet* may be overwritten, if the same parameter type is added to a relayed packet.
3. Finally, some applications can intelligently merge gathered information at intermediate nodes. To realize this, the application can use the packet facade to retrieve, update or remove a specific parameter from

the payload of the packet. Using this retrieved information, the application developer can implement application-level aggregation solutions.

### 3.4. Adapting protocols for use with our architecture

The global aggregation architecture presented in this section is non-intrusive in the sense that it can be combined with any address-based network protocol. However, to use our global aggregation, two changes need to be made to the involved network protocols. (i) Instead of creating packets to exchange information, protocols and applications hand over a parameter to the global aggregation architecture. The architecture will either create a packet to send the parameter, or add the parameter to a passing packet. (ii) Protocols and applications do not inspect the payload of received packets. Instead, the architecture extracts from received packets all the parameters that reached their destination and distributes them to all interested network protocols or applications.

## 4. Mathematical analysis

In this section, a mathematical model is constructed that can be used to quantify the benefits of global aggregation. Section 5 compares the mathematical results with the results of a real-life performance study.

The packet savings will first be mathematically evaluated for single-hop aggregation. For simplicity, in this section, only periodic information sent in discrete time intervals is considered (every  $\Delta T$  seconds).

### 4.1. Definition of variables

Table 3 lists the symbols that are used in the following analysis. Assume that the first information exchange is generated at  $T_1$ . This information is repeated every  $\Delta T_1$  time units. It is assumed that the information 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 parameters 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:

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

Variable	Meaning
$T_j$	Time message type $j$ is generated for the first time
$\Delta T_j$	Time between two generated messages of type $j$
$AD_j$	Maximum <i>Acceptable Delay</i> for message of type $j$

Table 3: List of symbols used for the ILP formulation.

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

Example values of typical periodic message intervals for different protocols are given in Table 4.

#### 4.2. ILP formulation for multi-protocol optimization

Using these variables, it is possible to derive an ILP formulation of the problem which can be applied to any number  $K$  applications.

Consider:

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

$$\Delta T_j, \quad j = 1 \dots K \quad (2)$$

In Equation (1),  $RP = lcm(\Delta T_1, \dots, \Delta T_K)$  is the repetition period for all  $K$  protocols. It is defined as the amount of time between two identical sending patterns of the considered protocol messages, that is, the amount of time units before the initial situation reoccurs. This is illustrated in Figure 4. The repetition period is 20 time units, which is the lowest common multiple of 4 and 5. After this repetition period, the moment of information generation is indistinguishable from the initial situation.

These protocols are each characterized by an inter packet time  $\Delta T_j, j = 1 \dots K$ . The  $x_i$  variables are binary variables, each representing a timeslot in

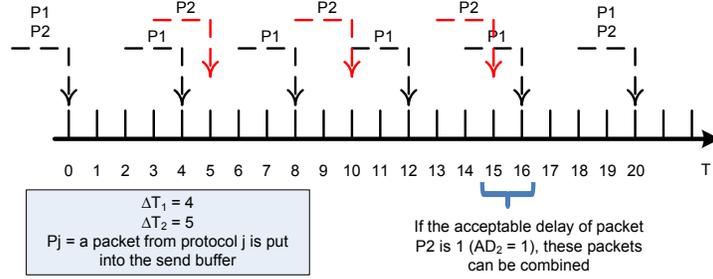


Figure 4: Illustration of the repetition period for 2 network protocols.

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:

$$\sum_{i=0}^{RP-1} x_i \quad (3)$$

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

$$\sum_{S_j^l}^{E_j^l} x_i \geq 1 \quad (4)$$

where

$$\sigma_j = \frac{RP}{\Delta T_j} - 1 \quad (5)$$

$$S_j^l = l(j) \cdot \Delta T_j \quad (6)$$

$$E_j^l = S_j^l + AD_j \quad (7)$$

This formulation can be understood as follows: equation (5) specifies the number of information parameters of protocol  $j$  that are generated in the RP interval. Equation (6) indicates the timeslot when information parameter  $l$  from protocol  $j$  is available (starting to count from 0). Equation (7) indicates the latest timeslot that information should be sent. Equation (4) then assures that during this span of timeslots at least one packet is sent, and this for

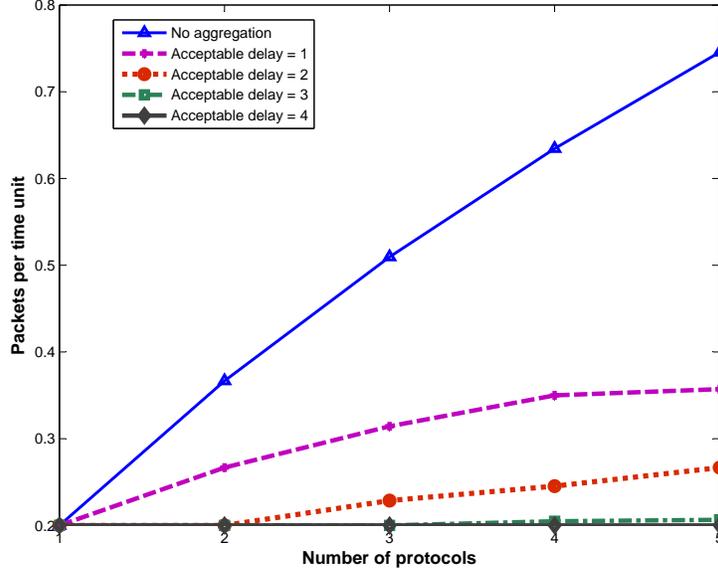


Figure 5: Required number of packets per time unit when using our non-intrusive aggregation scheme with multiple protocols ( $\Delta T_i = i + 4$ ).

every protocol  $j$  and every information parameter to be sent. By minimizing equation (3), the amount of packets sent is minimized. The number of *packets per time unit* (PPT) can be obtained by dividing equation (3) by the RP. This way, the number of required number of packets per time can be obtained for any number of protocols.

A reduction of the number of transmitted packets occurs when (i) the acceptable delay of applications increases, (ii) the information exchange interval of different applications are multiples of each other, or (iii) both these situations happen at the same time. The average number of packets per time unit can decrease up to a minimum of  $\frac{1}{\Delta T_x}$ , with  $\Delta T_x$  the lowest information interval. Or, in other words, the maximum possible transmission savings are limited by the lowest information generation interval. The influence of the acceptable delay is illustrated in Figure 5, where the ILP formulation is solved for a variable number of protocols.

#### 4.3. Applying the formulas to more complex scenarios

Up until now, the mathematical analysis assumed that all applications sent small information exchanges to a single sink node. However, the situation becomes more complex if not all generated information can be combined

in a single packet. There are three main causes for this to occur. *(i)* Due to packet size constraints, not all information might fit in a single packet. *(ii)* If the applications generate information with different QoS requirements which require packet specific processing. And finally, *(iii)* more advanced applications, such as building automation, send information parameters to different destinations and next hop addresses. If this is the case, the mathematical analysis must be applied once for every set of applications that exchanges information with a specific destination.

In addition, the analysis from Section 4.2 does not account for scenarios where packets must be routed over multiple intermediate hops. In this situation, the total number of transmitted packets needs to be modified, depending on the size of the network and the characteristics of the applied routing protocol. The upper limit of the total number of required packets can be obtained by multiplying the results of the ILP formula (3) by the average hop count. This estimate results in a worst case scenario, since intermediate nodes can often add parameters from their waiting space to the relayed packet, resulting in less packet transmissions.

In the next section, the performance of our aggregation scheme will be experimentally evaluated in real-life scenarios. In addition, the performance of our aggregation will be compared the performance of traditional aggregation schemes in varying network conditions.

## 5. Real-life performance evaluation

Network protocols for wireless sensor networks are often evaluated using simulation software (ns2, glomosim, j-sim, matlab, etc.). The use of simulation software has two major benefits: *(i)* experiments are repeatable and *(ii)* large-scale networks are easy to simulate. However, simulation software uses many simplifications that are not found in real-life deployments. As such, simulations often return non-realistic results, which can be very different from the results of real-life deployments. Characteristics like clock drift, failing nodes or fading and reflection of transmissions, are not accounted for. In order to get realistic results, it is necessary to use a real-life testbed [30]. Before giving the results of the performance evaluation, the experimental setup is described.



Figure 6: The W-ilab.t wireless sensor testbed contains about 200 nodes spread over 3 floors. Each floor measures 15 by 90 meter. The sensor nodes are indicated with a dot. Elevator shafts, indicated with a rectangle, provide connectivity between different floors.

### 5.1. Experimental setup

#### *Testbed description*

For our experiments, the W-iLab.t wireless sensor testbed [31, 32] was used, which is located in the IBBT - Ghent University office building in Belgium. The W-iLab.t testbed consists of about 200 TmoteSky sensor nodes, spread out over three floors. Some of the sensor nodes are situated in ventilation shafts between the floors. As such, these ventilation shafts provide a connection corridor through which communication is possible between different floor levels. Figure 6 shows the location of the sensor nodes on the three floors. The sensor nodes are marked with a circle, the ventilator shafts are indicated with a rectangle. Large-scale multi-hop experiments are created by setting the transmissions power of the sensor nodes to an output power of -15 dBm. Using these settings, packets require 4-5 hops to be transmitted from one side of the building to the opposite side. Each test case has a duration of 1 h, and is executed five times. The results are averaged to remove outliers.

#### *Information exchanges*

To evaluate the effectiveness of our aggregation scheme, traffic must be gen-

Information exchange	ex-destination	# of exchanges	$\Delta T$	Acceptable delay (AD)
Sensor data	unicast to the sink	2	0 to 60 sec.	0 to 50% of $\Delta T$
Control traffic	broadcast to the direct neighbors	2	0 to 60 sec.	0 to 50% of $\Delta T$

Table 5: Information exchanges in the monitoring scenario.

erated by the sensor nodes. As stated in [5], most traffic in wireless sensor networks consists of two communication types: information exchanges to a remote destination (measured data) and information exchanges with direct neighbors (to exchange local status information).

Depending on the exact application, multiple types of data are often gathered [3, 6], for example temperature, air pressure, humidity or webcam images. For our evaluation, the number of information types on each node were limited to (a) two types of data traffic and (b) two types of control information between direct neighbors. More types of information exchanges would given an overly positive view of our aggregation approach since global aggregation is able to combine information exchanges coming from different network layers. The following information is exchanged (see also Table 5):

- Each node has two separate applications that send measured information to a sink, which is located on floor 3 at the edge of the building (Figure 6).
- In addition, whilst the network is operational, networking protocols send notification messages to directly neighboring nodes. These information exchanges contain status information, such as the remaining energy or the signal strength. This information is typically used by neighbor discovery, slot assignments, link quality estimations or synchronization protocols.

The applications on the nodes are not synchronized: they start generating information at a random start-up time.

#### *Communication patterns*

Wireless sensor networks typically use point-to-sink communication for monitoring applications and point-to-sink communication for interactive applica-

tions such as wireless building automation. When evaluating point-to-sink scenarios (Sections 5.3-5.5), communication paths are setup in advance by broadcasting a message from the sink to all other nodes. In the case of point-to-point communication (Section 5.6), a random destination node is chosen for each traffic flow during the experiments.

### 5.2. Evaluated aggregation paradigms

In the next sections, the performance of global aggregation is compared to the performance of several traditional aggregation approaches. First, a short description is given of all aggregation approaches which have been evaluated using the real-life testbed.

**No aggregation:** When no aggregation is used, packets are never combined. This situation is used as a reference scenario.

**Packet combination:** This approach is the most commonly used aggregation solution for Wi-Fi networks (see related work Section 2.1). Created packets are delayed for a short time at the MAC or PHY layer before they are sent over the wireless network. If multiple packets to the same next hop address are delayed in this way, they are combined in a single MAC or PHY frame.

**Traditional data-aggregation:** This approach describes the most common non-intrusive aggregation approach for wireless sensor network (see related work Section 2.2). Each application generates information which is encapsulated in a packet. In intermediate hops, the packet is decapsulated and the payload is offered to the application. The application can choose to fuse its own measured data with the received information. Afterwards, the packet is further forwarded to its destination. Data packets coming from different applications are not aggregated.

**Joint-application data-aggregation:** This approach depicts the approach whereby all applications are jointly designed (see related work Section 2.3). The result is similar to traditional data-aggregation, but the resulting application can combine data packets from any application. The evaluated joint-design approach includes only the application levels, thus control packets cannot be combined.

**Global aggregation:** This is our non-intrusive approach which was de-

scribed in section 3. Applications can be maintained by different developers, but their data is combined as if using joint-application data-aggregation. In addition, control messages are also aggregated, without the need for introducing any dependencies between the network layers.

In the following sections, the performance of these different aggregation approaches is evaluated under the following conditions: (i) changing network size, (ii) using different time intervals of control traffic versus data traffic, (iii) using different acceptable delay values, and (iv) using point-to-point traffic patterns. In addition, the average processing overhead, queue occupation and energy savings of the evaluated aggregation approaches are studied.

### 5.3. Influence of network size

According to the vision of ‘the internet of things’, a future office building could exist of several thousands of sensor nodes. As such, it is important to have detailed knowledge about the scalability of different aggregation methods. Therefore, our first analysis investigates which aggregation methods are most scalable for large networks in terms of the number of saved transmissions. For this analysis, the data and control information intervals are fixed, but the network size is varied from half a floor (40 nodes) to three floors (about 200 nodes). The resulting average packet transmissions per minute per node are shown in Figure 7.

*Packet combination* is mainly advantageous in small-scale scenarios, with many single-hop information exchanges. In these networks, many packets are destined to the same next hop address. As the network size increases, packet combination becomes limited to combining control messages to neighboring nodes. As a result, using the number of packet exchanges from Table 5, the average netto savings in our scenario converge towards a fixed reduction of 1 packet transmission per minute.

In contrast, *traditional data-aggregation* becomes more useful in large-scale networks. As the network size increases, more intermediate nodes can add their information to passing data packets, each time resulting in a saved packet transmission.

In small, single-hop networks, the use of *joint-application data-aggregation* results in the same number of packet transmissions as the packet combination approach. After all, when using single-hop networks, aggregation based on next hop addresses results in the same packet combinations as aggregation based on destination addresses. However, when the network size increases,

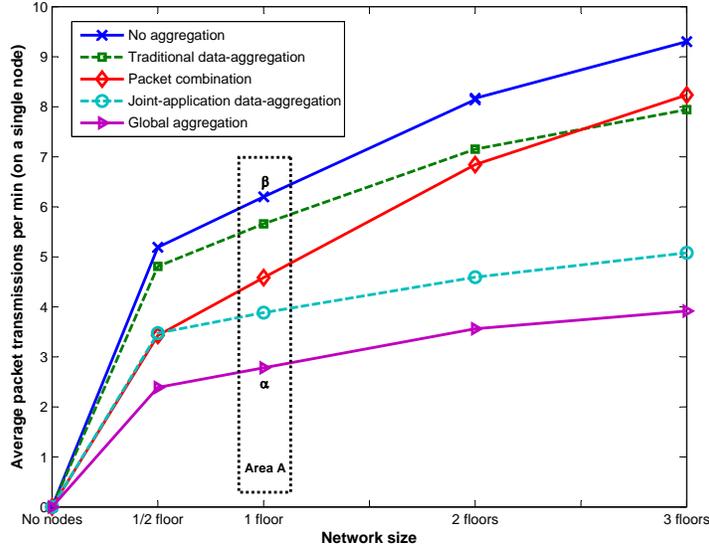


Figure 7: Influence of the network size on aggregation method ( $\Delta T_{data} = \Delta T_{control} = 60$  sec;  $AD_{data} = AD_{control} = 30$  sec).

the efficiency of joint-application data-aggregation becomes much more noticeable. In multi-hop networks, the reduction of a single packet means a packet transmission less for each intermediate hop towards the sink. This shows that (i) aggregating multi-hop packets has a much larger influence than aggregating single-hop packets and (ii) packets should be aggregated as soon as possible, preferably on the node where the data originates.

Finally, *global aggregation* combines the advantages of both approaches: both the destination and the next hop field of each packet are checked for combining packets. Thus, the number of packet reductions of this approach are similar to those of joint-application aggregation, together with the (fixed) netto saving of 1 packet/s from the packet combination approach.

#### 5.4. Ratio of control traffic versus data traffic

Depending on the application, the time interval ( $\Delta T$ ) between sensor measurements can vary widely, from a few seconds for fire-detection applications to several hours for the monitoring of long-term nature phenomena. Similarly, unreliable networks with mobile nodes or frequently occurring node failures require more frequent status updates than static sensor networks. Since existing aggregation protocols are often optimized for data traffic, the

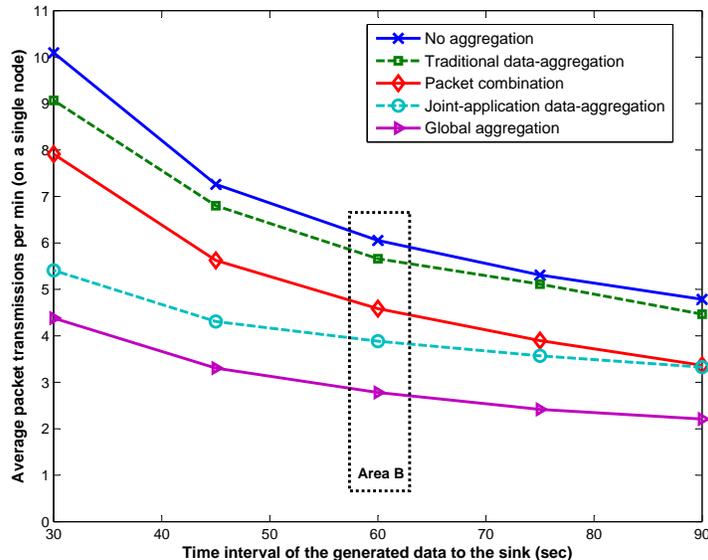


Figure 8: Influence of the traffic frequency on aggregation methods (1 floor;  $\Delta T_{data}$  = variable;  $\Delta T_{control} = 60$  sec;  $AD_{data} = AD_{control} = 30$  sec).

ratio of data traffic versus control traffic has a profound influence on the performance of the aggregation protocol.

In a realistic scenario, the ratio of control traffic versus data traffic depends on many factors, such as:

- protocol design, for example: reactive routing versus proactive routing;
- network stability: a network with many node failures requires regular ‘alive’ messages between neighbors; and
- network mobility: mobility requires the use of regular discovery and registration messages.

In the previous situation, both control information and measured data were sent every 60 seconds. In the next scenario, the number of control messages is kept constant, but the time interval of the data traffic to the sink is varied. The results are shown in Figure 8. In Area B of Figure 8, the data interval is 60 s, which corresponds to the results from Area A of the previous figure.

Section 5.3 demonstrated that traditional data-aggregation does not perform well in small-size networks. Since this experiment uses only a single floor (about 80 nodes), the performance of traditional data-aggregation is

Variable	Meaning
P[pckt_reduction]	Probability that a packet transmission is avoided.
P[info_available]	Probability that information is waiting in the waiting space.
P[pckt_passing]	Probability that a packet passes through the system to which the information can be added, or a local packet is created to encapsulate another information parameter to the same destination.

Table 6: Significance of the symbols used to calculate the number of packet reductions

accordingly also very poor. Traditional data-aggregation will only result in profound transmission savings if the number of data exchanges is several times higher than the number of control exchanges.

For a network with an equal amount of data and single-hop control traffic (Area B of Figure 8), the use of *joint-application data-aggregation* results in more packet savings than *packet combination*, since each data packet requires additional packet transmissions in intermediate hops towards the sink. From this, it can be concluded that for networks with data traffic higher or equal to the amount of control overhead, a destination based aggregation approach is the better choice. In contrast, for networks where the major transmission overhead consists of control messages to direct neighbors (the far right side of Figure 8), the use of a packet combination becomes a better approach.

Finally, *global aggregation* again has the best performance of the compared aggregation methods. The number of packet transmissions is always reduced by over 50% when compared to the reference scenario.

### 5.5. Influence of acceptable delay

For the previous experiments, information could be delayed for up to 30 s. In this experiment, the influence of the acceptable delay is evaluated. Using the notation from Table 6, the probability that information from the waiting space can be combined with another packet is the following:

$$P[pckt\_reduction] = P[info\_available] * P[pckt\_passing]$$

Figure 9 shows the resulting average packet transmissions per minute per node in a network of 1 floor when the acceptable delay is varied. When both applications on each node generate information with the same time interval,

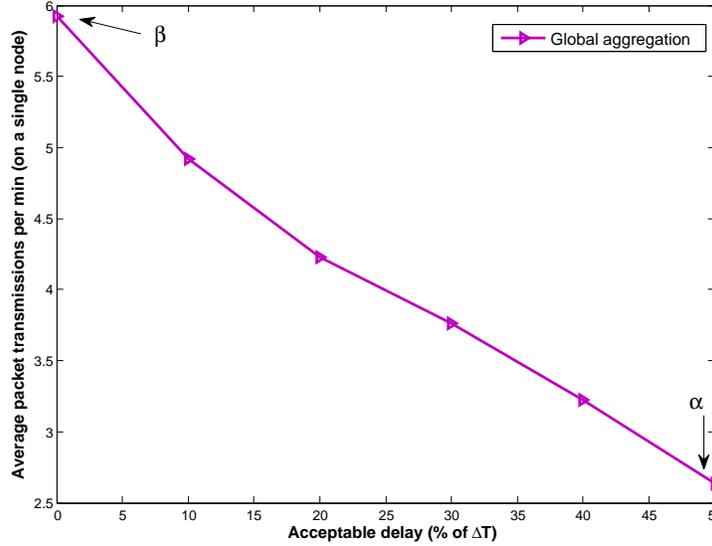


Figure 9: Influence of the acceptable delay on aggregation methods (1 floor;  $\Delta T_{control} = \Delta T_{control} = 60$  sec;  $AD_{data} = AD_{control} = \text{variable}$ ).

an acceptable delay of 50% (Figure 9, point  $\alpha$ ) ensures that the information is delayed long enough to guarantee that the information from the first application can be combined with the data from the second application. Similarly, all control packets of the first network protocol can be combined with the control packets of the second network protocol. This situation corresponds with point  $\alpha$  of the previous Figure 7.

Once the acceptable delay reaches zero, information will no longer be delayed on the first node (point  $\beta$  of figure 9). In this situation, our aggregation scheme does not result in transmission savings, since no parameters can be combined. As such, this situation corresponds to the results from the experiments without aggregation (point  $\beta$  of previous Figure 7).

For the described network, linearly decreasing the acceptable delay results in a linear increase of the packet transmissions. The graphs describing the influence of acceptable delay for the other aggregation schemes can be deduced similarly based on previous figures, by drawing a line from the required number of packet transmissions of a specific aggregation approach, to the required number of packet transmissions in the same scenario without aggregation.

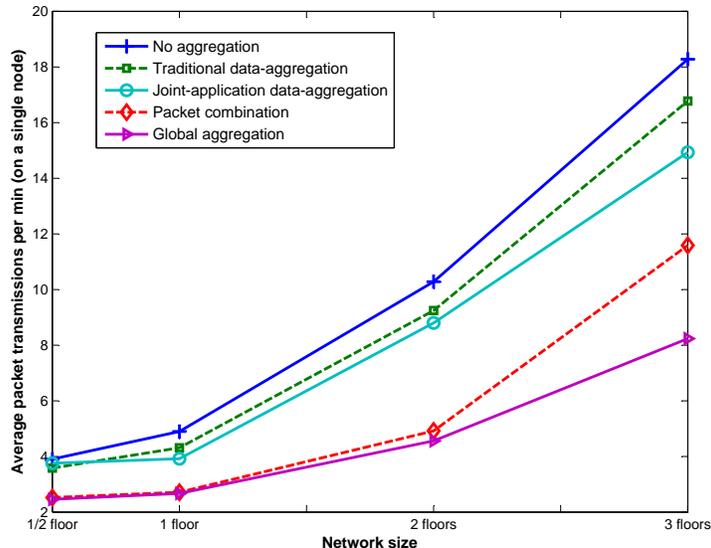


Figure 10: Performance of different aggregation methods in a point-to-point scenario ( $\Delta T_{data} = \Delta T_{control} = 60$  sec;  $AD_{data} = AD_{control} = 30$  sec).

### 5.6. Point-to-point communication patterns

This section evaluates how efficient different aggregation methods support next-generation sensor applications that use complex traffic patterns. In the following scenario, all nodes exchange regular status messages with their neighbors. However, 20% of the nodes are used as a source for point-to-point traffic: they contact a random other node to which they send their measured data. To setup routes, the AODV [33] protocol is used. The AODV implementation generates a limited amount of control overhead: every 10 min the maximum lifetime of an AODV path is reached and a new path setup is executed. In Figure 10, the resulting average number of packet transmissions is shown for different aggregation methods.

The number of packet transmissions when using *no aggregation* increases more quickly than when using the previous point-to-sink scenario. This is mainly due to the additional broadcast messages that are required for route-setups.

*Destination-based* aggregation schemes (traditional and single application data-aggregation) have a positive effective on the number of packet transmissions. However, this effect is rather limited, since the probability of having multiple nodes sending information to the same destination is small.

In comparison, *packet combination* approaches perform better: *(i)* control messages to direct neighbors can be piggybacked on the route-requests that are broadcast and *(ii)* due to the random selection of destination nodes, routes are shorter on average when compared to the single sink scenario which favors the packet combination approach.

*Global aggregation* and packet combination perform similarly for small networks. This demonstrates the fact that in many small networks, the packet reduction from destination based aggregation methods consists mostly of control messages that have both the same next hop and the same destination address. When the number of hops becomes larger, global aggregation scales even better than packet combination in terms of packet transmissions, reducing the total amount of packet transmissions up to 45%.

In conclusion: destination based approaches do not perform well in point-to-point scenarios, due to *(i)* the increase of broadcast control packets (used for path-setups), *(ii)* the decreased probability of having many nodes with the same destination, and *(iii)* the decrease of the average path length (due to the random selection of a destination node). Packet combination or global aggregation still result in a significant reduction of the number of transmitted packets.

### 5.7. Processing overhead

During the experiments, the same simple information merging function was used for all aggregation approaches: all aggregated parameters are copied sequentially to the payload. Thus, the average processing overhead of the individual aggregation method depends mainly on the number of addresses that should be inspected when relaying a packet, e.g.: checking the next hop address of passing packets, checking the destination address or checking both.

Table 7 shows the delay that is required until a received packet is fully processed (e.g. it is ready for sending) using different aggregation methods.<sup>2</sup> It is clear that the use of more complex aggregation methods results in additional processing delay. However, a large processing delay is generally not problematic for WSNs: a MAC protocol for WSNs typically has a sleeping interval that is larger than the processing delays from Table 7. Thus, the delay caused by the MAC protocol will typically become the bottleneck

---

<sup>2</sup>This result is obtained using a microprocessor with a clock speed of 2 MHz.

Aggregation type	Aggregated Addresses	Avg processing delay
No Aggregation	None	5.54 msec
Traditional Data-Aggregation	Destination	6.53 msec
Packet Combination	Next Hop	7.15 msec
Joint-Application Data-Aggregation	Destination	6.52 msec
Global Aggregation	Destination & Next Hop	8.21 msec

Table 7: Typical processing overhead of the different aggregation techniques

with regard to total throughput. In non-WSN deployments, the delay can be lowered by disabling aggregation on a per-packet basis, depending on the required QoS guarantees of the packet.

#### 5.8. Queue occupation

Wireless sensor nodes are usually limited in terms of memory. Lowering the size of the packet queues results in additional memory for other purposes. However, using a small packet queue is only possible when the number of packets in the system (‘the average queue occupation’) is low. The average queue occupation depends on (i) the average number of packets that need to be transmitted; (ii) the duration that a packet remains in the packet queue (i.e. the processing delay from Section 5.7); and (iii) the packet size (which is less relevant for WSNs, since most sensor nodes use queues with a fixed packet size).

In Table 8, the average queue occupation is shown for the results from the experiments depicted in Figure 8, with a data interval of 90 seconds. The column with the first results shows the average queue delay when using a CSMA/CA MAC protocol. This type of MAC protocol sends packets as soon as the wireless medium is free. Note that the average queue occupation when using ‘no aggregation’ is lower than the average queue occupation when using ‘traditional data-aggregation’, even though more packets are transmitted. This can be explained by the higher processing overhead of the latter approach, which is not compensated by a significant decrease of packet transmissions. Thus, when the queue size is limited, it is important to use an aggregation scheme that is efficient in both terms of processing overhead and packet reductions.

Aggregation type	Avg queue occupation (%) (1)	Avg queue occupation (%) (2)
No Aggregation	0.317	13.775
Traditional Data-Aggregation	0.374	13.581
Packet Combination	0.301	13.124
Joint-Application Data-Aggregation	0.102	12.900
Global Aggregation	0.080	11.688

Table 8: Average queue occupation of the different aggregation techniques when using: (1) CSMA/CA and (2) S-MAC [34] with a sleeping period of 200 msec. (max queue occupation = 10 packets).

The column with the second results from Table 8 shows that, when a MAC protocol with sleeping schemes is used [34], the aggregation delay becomes negligible compared to the sleep duration of the radio. Due to the fixed delay of each packet, the queue occupation depends only on the number of packet transmissions and not on the processing complexity of the aggregation protocol.

In summary, for energy-efficient networks that use a MAC protocol with sleeping schemes, the additional processing delay is generally not an issue. The fact that the average queue occupation is lower has two possible positive effects: *(i)* less packets are lost due to packet overflows and *(ii)* memory can be saved by assigning smaller queues for the packets.

### 5.9. Energy savings

By intelligently turning on and off the radio, the amount of time spent in low-power sleep modus is increased. However, the sleep time is limited by the total throughput that the MAC protocol should support. Thus, if less packets need to be sent, the sleep duration can be increased without causing network congestion.

As an example, Table 9 shows the expected energy consumption (estimated from [35]), associated with different throughputs for the popular S-MAC and B-MAC protocols. In the previous sections it was demonstrated that our aggregation protocol can reduce the number of packet transmissions with more than 50%, which in turn increases the network lifetime by 30-50%, depending on the MAC protocol.

Throughput (bits / sec)	S-MAC	B-MAC
25	4 mW	5 mW
50	7.5 mW	7 mW
75	11.5 mW	9 mW
100	15 mW	11 mW
150	23 mW	13 mW
200	31 mW	16 mW

Table 9: Estimated average energy consumption required for a given throughput when using the S-MAC [34] or B-MAC [35] protocols (mW = mJ / second).

### 5.10. Conclusions from the real-life performance evaluation

The results from these real-life benchmarks show that the choice of an optimal aggregation scheme depends on many external factors. Traditional data-aggregation, while often used in current sensor networks, is mainly suited for large multi-hop networks with a single application that generates point-to-sink traffic. Joint-application aggregation performs better than traditional aggregation, both in small and large-scale networks. However, its main use is still limited to scenarios with point-to-sink traffic. In contrast, packet combination performs well in small-size networks, for scenarios that require point-to-point traffic and in networks where control traffic to direct neighbors is dominating the data traffic.

In addition, it was shown that, in order to obtain the best results, aggregation should be executed as soon as possible, preferably on the node where the data originates. Finally, it was shown that our non-intrusive approach combines the advantages of both Wi-Fi packet combination and joint-application data-aggregation. As a result, our proposed aggregation approach is currently the best non-intrusive solution for reducing the number of packet transmissions in a multitude of scenarios.

## 6. Open research directions

During the mathematical and experimental performance evaluation, it was concluded that the optimal (traditional) aggregation approach depends on the network situation. Using this analysis as a basis, new adaptive or hybrid aggregation protocols can be developed that change their behavior depending on the network conditions.

Current state-of-the-art data-aggregation approaches often ignore Quality-of-Service (QoS) constraints [6]. To cope with different QoS classes in our architecture, global aggregation can be configured to:

- disable aggregation for certain QoS classes;
- automatically update the QoS field of outgoing packets to the QoS class of the parameter with highest QoS requirements.

As there is as of yet no agreement on the necessary QoS provisions for next-generation applications for WSNs, it is too early to evaluate if this solution suffices for future networks.

Finally, in current aggregation approaches, it is generally assumed that the most energy-efficiency is obtained by aggregating as much information as possible in a single packet. This assumption is plausible, considering the small size of data in sensor networks. However, 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 depending on the network conditions can be an interesting research topic, similar to research which has been done for Wi-Fi networks in [13].

## 7. Conclusions

Data-aggregation approaches reduce the number of packet transmissions in data-gathering applications. However, many existing aggregation approaches need to be custom fine-tuned for use in specific scenarios. In addition, they are typically tightly integrated with the routing protocol and work only for predictable sensor deployments (in terms of communication patterns and/or event periodicity).

To remedy this situation, the paper presented a non-intrusive aggregation architecture in which information exchanges from all layers are aggregated and combined to reduce the number of wireless transmissions. In contrast to traditional data-aggregation protocols, our architecture can be used for networks that (i) generate regular but unpredictable events, or (ii) have unpredictable communication patterns, or (iii) require rapid deployment without prior information about the network characteristics. In addition, the presented approach is protocol-independent: the aggregation approach can be combined with any other network protocol.

The architecture uses a very simple queuing system without complex calculations or aggregation functions, making it suitable for both Wi-Fi 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  $\Delta T_x$  the lowest information interval.

In addition to the mathematical analysis, the performance of different existing aggregation schemes was compared in various network conditions. The main conclusion is that currently no existing single solution is suited for a wide range applications and network scenarios. However, by combining several existing techniques, our aggregation method proves to be superior in all tested use cases.

Aggregating measured data is currently considered essential for obtaining a long network lifetime for wireless sensor networks. Throughout the paper it was shown that the same will hold true for the extension of data-aggregation towards aggregation of information in general, both control and data information. As such, we strongly believe that aggregation research should be tackled with as few dependencies as possible with existing network protocols.

- [1] FP6 Embedded WiSeNts Project. Report 2.1: Critical evaluation of research platforms for wireless sensor network, [http://www.embedded-wisents.org/studies/survey\\_wp2.html](http://www.embedded-wisents.org/studies/survey_wp2.html).
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, *Communications Magazine, IEEE* Vol. 40 (8) (Aug 2002) pages 102–114. doi:10.1109/MCOM.2002.1024422.
- [3] N. Xu, A survey of sensor network applications, *IEEE communications magazine*, issue 8 Vol. 40 (2002) pp 102.
- [4] Cc2420 radio datasheet, <http://focus.ti.com/docs/prod/folders/print/-cc2420.html> (March 2007).
- [5] G. P. Halkes, T. van Dam, K. G. Langendoen, Comparing energy-saving mac protocols for wireless sensor networks, *Mobile Network Applications* Vol. 10 (5) (2005) pages 783–791. doi:<http://doi.acm.org/10.1145/1160143.1160161>.
- [6] X. Feng, Qos challenges and opportunities in wireless sensor/actuator networks, *Sensors* 2008 (2) (21 February 2008) pp. 1099–1110.

- [7] P. De Mil, T. Allemeersch, I. Moerman, P. Demeester, W. De Kimpe, A scalable low-power wsan solution for large-scale building automation, in: Communications, 2008. ICC '08. IEEE International Conference on, 2008, pp. 3130–3135. doi:10.1109/ICC.2008.589.
- [8] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, J. A. Stankovic, An advanced wireless sensor network for health monitoring, Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), Arlington, VA, April 2-4, 2006.
- [9] R. Mangharam, A. Rowe, R. Rajkumar, R. Suzuki, Voice over sensor networks, RTSS 2006. Proc. of the 27th IEEE International Real-Time Systems Symposium.
- [10] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, S. Singh, Exploiting heterogeneity in sensor networks, In Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2005), Mar. 2005.
- [11] R. R. Choudhury, A. chen, S. Emeott, An analytical view of data aggregation in IEEE 802.11 LANs, in: Proceedings of Globecom06, San Francisco, USA, 2006.
- [12] 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.
- [13] R. Riggio, F. D. Pellegrini, N. Scalabrino, P. Li, Y. Fang, I. Chlamtac, Performance of a novel adaptive traffic aggregation scheme for wireless mesh networks, in: Proceedings of MILCOM07, Orlando, Florida, USA, 2007.
- [14] E. Fasolo, M. Rossi, J. Widmer, M. Zorzi, In-network aggregation techniques for wireless sensor networks: a survey, Wireless Communications, IEEE 14 (2) (April 2007) 70–87. doi:10.1109/MWC.2007.358967.
- [15] R. Rajagopalan, P. Varshney, Data-aggregation techniques in sensor networks: a survey, Communications Surveys & Tutorials, IEEE 8 (4) (Fourth Quarter 2006) 48–63. doi:10.1109/COMST.2006.283821.

- [16] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *Wireless Communications, IEEE Transactions on* 1 (4) (Oct 2002) 660–670. doi:10.1109/TWC.2002.804190.
- [17] J. Gehrke, S. Madden, Query processing in sensor networks, *IEEE Pervasive Computing* 3 (1) (2004) 46–55. doi:http://doi.ieeecomputersociety.org/10.1109/MPRV.2004.1269131.
- [18] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: *Mobile Computing and Networking*, 2000, pp. 56–67.
- [19] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, TAG: a tiny aggregation service for ad-hoc sensor networks, *SIGOPS Oper. Syst. Rev.* 36 (SI) (2002) 131–146. doi:http://doi.acm.org/10.1145/844128.844142.
- [20] S. Nath, P. B. Gibbons, S. S., 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*, ACM, New York, NY, USA, 2004, pp. 250–262. doi:http://doi.acm.org/10.1145/1031495.1031525.
- [21] A. Manjhi, S. Nath, P. B. Gibbons, Tributaries and deltas: Efficient and robust aggregation in sensor network streams, in: *in ACM SIGMOD*, ACM Press, 2005, pp. 287–298.
- [22] S. Lindsey, C. Raghavendra, K. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, *Parallel and Distributed Systems, IEEE Transactions on* 13 (9) (2002) 924–935. doi:10.1109/TPDS.2002.1036066.
- [23] A. Silberstein, R. Braynard, J. Yang, Constraint chaining: on energy-efficient continuous monitoring in sensor networks, in: *In Proceedings of SIGMOD*, 2006, p. pp. 157–168.
- [24] M. Umer, L. Kulik, E. Tanin, A location based aggregation algorithm for selective aggregate queries in sensor networks, *2nd international conference on geosensor networks*.

- [25] C. Suh, Y.-B. Ko, , D.-M. Son, An Energy Efficient Cross-Layer MAC Protocol for Wireless Sensor Networks, H.T. Shen et al. (Eds.): APWeb 2006, LNCS 3842 (2006) 410–419.
- [26] T. Melodia, M. C. Vuran, 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, 2005.
- [27] Distributed management task force, inc., Common Information Model (CIM) Standards, <http://www.dmtf.org/standards/cim/>.
- [28] E. De Poorter, I. Moerman, P. Demeester, An information driven sensor-net architecture, SENSORCOMM 2009 The Third International Conference on Sensor Technologies and Applications, Athens/Glyfada, Greece (June 18-23, 2009) 553–561.
- [29] Tinyos operating system, <http://www.tinyos.net/>.
- [30] C. Sreenan, S. Nawaz, T. Le, S. Jha, On the sensitivity of sensor network simulations, in: Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd, Vol. Volume 3, 7-10 May 2006, pp. Page(s):1043 – 1047, digital Object Identifier 10.1109/VETECS.2006.1682993.
- [31] The IBBT W-iLab.t wireless sensor testbed. <http://ilabt.ibbt.be/>.
- [32] L. Tytgat, B. Jooris, P. D. Mil, B. Latré, I. Moerman, P. Demeester, Demo abstract: Wilab, a real-life wireless sensor testbed with environment emulation, published in European conference on Wireless Sensor Networks, EWSN adjunct poster proceedings (EWSN), Cork, Ireland, 11-13 February 2009.
- [33] Ad hoc on-demand distance vector (aodv) routing. networking group request for comments (rfc): 3561, <http://tools.ietf.org/html/rfc3561> (July 2003).
- [34] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: 21st Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, 2002, pp. 1567–1576. URL <http://dx.doi.org/10.1109/INFCOM.2002.1019408>

- [35] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM, New York, NY, USA, 2004, pp. 95–107. doi:<http://doi.acm.org/10.1145/1031495.1031508>.