Technical University Berlin

Telecommunication Networks Group

A Data Aggregation Framework for Wireless Sensor Networks

Holger Karl, Marc Löbbers,
Tim Nieberg

karl,loebbers@ee.tu-berlin.de; T.Nieberg@utwente.nl

Berlin, September 2003

TKN Technical Report TKN-03-016

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

A Data Aggregation Framework for Wireless Sensor Networks

Holger Karl, Marc Löbbers
Technical University of Berlin, TKN
Einsteinufer 25
D-10587 Berlin
{Karl|Loebbers}@ft.ee.tu-berlin.de

Tim Nieberg
University of Twente, EWI
Postbus 217
NL-7500 AE Enschede
T.Nieberg@utwente.nl

Abstract—**In wireless sensor networks, a possibility to reduce the amount of data to be transmitted, and therefore to conserve energy, is to combine several sensor readings in intermediate nodes along the way towards the requester. This process is known as data aggregation.**

However, due to the dynamics of a wireless, ad hoc network, transmissions are error-prone. Messages may not be received correctly and thus all the combined information is lost. We are investigating strategies to adaptively employ different link-local error control mechanisms (forward error correction codes and automatic resend requests) depending on the amount of information of a message in order to increase the overall aggregated information available for the requester. The question of different criteria to grade the informational content, taking into account some characteristics of the wireless network, is investigated.

Keywords—**Wireless Sensor Networks, Data Aggregation, FEC, ARQ**

I. Introduction

Wireless sensor networks (WSNs) are an emerging field of research which combines many challenges in distributed computing and embedded systems [1]. The improvements in digital circuitry technology allow for the integration of sensing, processing, and wireless communication capabilities on a single chip in the near future. Small, battery-operated sensor nodes can be cheaply deployed virtually everywhere, and the resulting distributed sensor network offers a great variety of applications, usually composed around the monitoring facilities of the sensors. The limited transmission ranges of the sensor nodes themselves create a multi-hop, ad hoc communication topology [2].

A typical task of a wireless sensor network is the monitoring of a larger area with respect to some given physical quantity, e.g., temperature. Usually, the end user wants to extract information from the sensor field: this information is gathered by the sensors, and reported to a point which we refer to as *data sink*. For this process, a sink node requests readings from the entire network by flooding the network with appropriate request messages. There exist several mechanisms to control the flooding process and construct e.g. a convergecast tree along which the answers are reported back to the sink.

This paper deals with the reverse process of each sensor reporting back its reading for the request. The energy efficiency of this operation has to be balanced with the quality of the result that is obtained from the operation: making sure that every single sensor's reading reaches the monitoring node gives the best possible result, but can require a lot of energy to combat wireless transmissions errors and to forward data via intermediate nodes. On the other hand, this effort might be ill-spent: because of the underlying physical process that is observed, sensor readings of neighboring nodes are typically related to each other and hence requiring really *all* readings to arrive can be exaggerated. Also, the envisioned spatial density of the sensors in the field can be exploited this way: a sensor reading from a region already covered (partially) by another sensor has less informational content.

The amount of new information that a given message carries should thus be put into perspective with the effort that is required to transmit this message over an additional hop. This is particularly important if *aggregation* in the network is used: instead of sending every message to the data sinks, intermediate nodes delay messages until they have received (all or some) messages from their children nodes (in the convergecast tree), compute an aggregated value of all these values (e.g., an average temperature or a maximum), and then forward only a single message with the aggregated value. Such a convergecast tree with the number of aggregated sensor readings per link is illustrated in Figure 1.

While this is in principle straightforward, problems appear, e.g., because of the error-prone nature of wireless links. Hence, a concept is required that decides what to do with lost messages, and intuitively, such a concept should take into account the relative importance of messages that

This work has been partially supported by the European research project EYES (IST 2001-34734, http://eyes.eu.org).

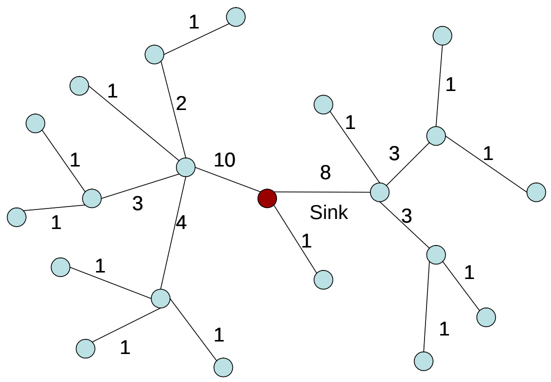

Fig. 1. Convergecast tree with number of aggregated sensor readings communicated over each link

carry different amounts of aggregated readings. Figure 2 illustrates that lost messages with only a few messages need not incite big reactions, but that lost messages with a lot of aggregated information inside warrant some additional expenditure of energy to repair such losses, since otherwise a lot of information into which already a lot of energy has been invested would have been lost. A concept like this one has not been considered in the literature so far; the most closely related papers are [5] [4] [3] .

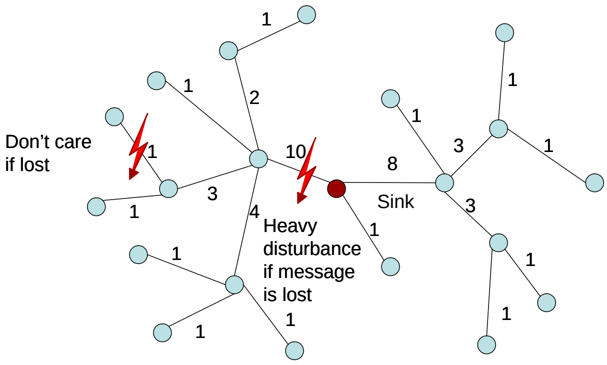

Fig. 2. Convergecast tree with lost messages

Following this intuition, a protocol hence has to combine end-to-end characteristics – how many sensors are there, how important is a given piece of information, does it come from an area that is not otherwise covered, etc. – with link-local decisions – how should lost packets be treated, should energy be invested to avoid packet losses in the first place, etc. The optimization goal is to maximize the relationship of obtained information in the monitoring node with the total energy required to provide this information.

In the following section, we shortly discuss approaches to ensure a correct transmission of a data packet. The aggregation and possible ways to determine the informational content, and thus the importance, of a message are presented thereafter. We performed some initial simulations with a simple aggregation-scheme, these are presented in Section IV. This paper concludes with an outlook on the future work which is currently part of our research.

II. Link Layer Issues

This section briefly describes the link-layer mechanisms and procedures that are required to support our concept.

A. Forward Error Correction (FEC)

FEC allows to add some redundancy to a packet so that the packet can be correctly reconstructed from the received message even if it encountered a (limited) number of bit errors during the transmission. Different forward error control codes exist so that proper choice for packet size, channel condition and intended protection are feasible. The trade-off is between reduced packet error rate and longer packet length, i.e. message size.

B. Automatic Repeat Request (ARQ)

ARQ protocols enable retransmissions of failed packets by sending explicit acknowledgements upon reception and detection of missing acknowledgements at the sending end of a single transmission. The trade-off here is the required overhead for acknowledgements in the correct case against the shorter packet length (when compared to FEC).

C. Transmission Power

Higher transmission power reduces the packet error rate by improving the signal-to-noise ratio, but increases the energy consumption. Also, interference with other nodes that come within reach upon increasing the transmission power poses additional problems, and this mechanism has to be supported by the radio.

D. Data Rate Adaption

Controlling the data rate comes at the trade-off between reducing the time necessary to transmit a packet at a cost of increased energy consumption. Data rate adaption is closely related to the FEC approach and has to be supported by the radio present in the nodes.

This paper focuses on the first two mechanisms as these are independent of the radio front end used in the sensor nodes. The goal of the resulting *aggregation-aware link layer protocol* is to choose these mechanisms (typically in combination) depending on the *importance* of a message that is to be forwarded. How to determine this importance by the informational content is presented in the following section.

III. Aggregation Mechanism

The question here is how to handle the aggregation of data in sensor messages to enable an aggregation-aware link/transport layer protocol. The process starts at a leaf node (a node without children in the convergecast tree), which intends to transmit a single reading to its parent node. Such a transmission is hence annotated with the following values:

- *Value* The actual value of the reading.
- *Number* The number of sensors that have contributed to this value (in the leaf case, 1).
- *Area* The (approximated) area that is covered by this reading (in the leaf case, the area that is supervised by the given sensor). This area is required to make sure that sensor readings from all parts of a supervised region are taken into account.

This information forms the basis of a recursion (and also the basis form which the link layer mechanisms described in the previous section decide how to transmit the packet). The recursion step happens in an intermediate node: After collecting the relevant readings from the child nodes, this intermediate node computes

- the aggregation function from all the received values as well as its own sensor reading (if applicable),
- determines a new, approximative description of the covered area,
- sums up all the numbers of contributing sensors, and
- constructs a new packet with these values, to be passed down to the link layer.

The aggregation function will typically be fairly simple, e.g., minimum, maximum, or average, and is trivial to compute. More complex aggregation functions can also be accommodated, if necessary. Summing up the number of contributing sensors is also straightforward.

The calculation of the covered area is both conceptually and computationally the most challenging aspect. Several possibilities exist:

- A simple list of circles (or polygons), concatenating and describing the list of the individual sensor nodes' covered areas.
- A single circle, describing the smallest circle that includes all the areas of the contributing sensors.
- A polygon, describing and aggregating the area of coverage of the sensors in more detail.
- As a trivial alternative, the area description can also be dropped if the geographic spread and balance of the obtained values is not important.

Evidently, there is a trade-off between precision and overhead which has to be characterized in future performance evaluations. Currently, the preferred solution is to either not use an area description at all if that is not relevant to the application or to use a single circle (from which, together with the number of contributing sensors, the average density of nodes can be derived).

Once this information triplet (value, number, and area) has been computed, the packet is passed on to the link layer, where a redundancy control module decides how to transmit the packet.

IV. Preliminary Results

In order to get a first impression, we did not use the area to judge the informational value of a packet, but simply used the number of individual sensor readings contributing to an aggregated message. We simulated the convergecast over error-prone links with independent bit errors. For this assumption on the error behavior, using FEC is the best strategy to increase the reliability of the transmission.

In our simulations, we implemented several strategies in the redundancy control module:

- *Strategy 1*: Do not use an error-correction code.
- *Strategy 2*: When the packet contains two entities, add an FEC that can correct a single bit error. For each additional information entity, use an FEC that can correct one additional bit error, but use at most an FEC that corrects four bit errors.
- *Strategy 3*: Use an FEC that corrects the same number of bit errors as information entities are presented in the packet. Never correct more than four bit errors.
- *Strategy 4*: Always use an FEC that corrects four bit errors.

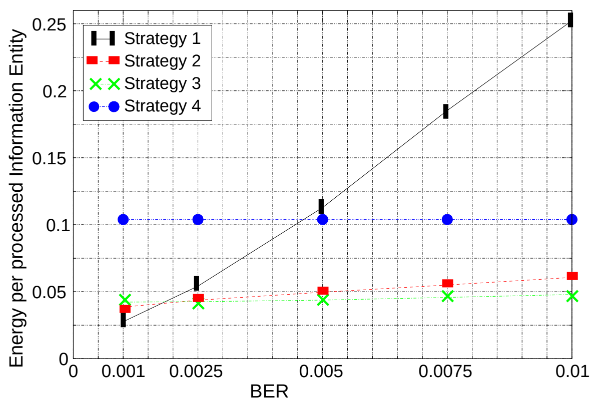

Fig. 3. Energy consumption of proposed strategies

The first strategy has the smallest overhead, but is not very reliable. The first metric – overhead – is covered using the energy spent per processed information entity. Processed information entities are entities that were sent by the sources and became part of the final value. Entities that were lost on the way to the sink are not counted. The

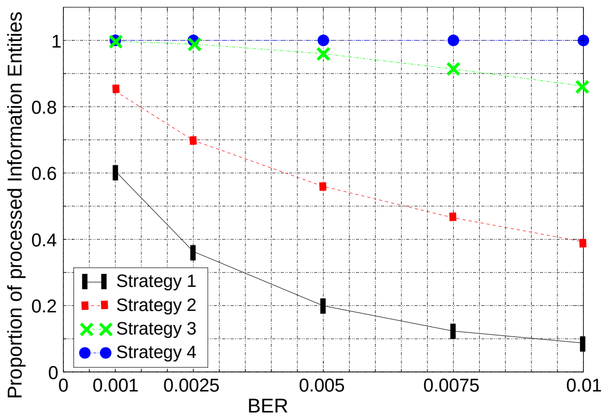

Fig. 4. Proportion of processed information entities

energy spent in the transmission of data entities is compared for the various strategies. The second metric – reliability – is compared with the ratio of information entities that were present at the sources and the number of entities that became part of the final estimate.

Figures 3 and 4 give an impression of the performance of different strategies for BER (Bit Error Rate). For the cases represented here, the third strategy seems to be a good compromise between energy efficiency and accuracy of the final estimate. However, the assumption that bit errors appear independent is not always justified. The development of strategies that work under a wide range of channel assumptions is part of the future work.

V. Future Work

In contrast to other, context-free networks, WSNs have a close connection with the surrounding area where the physical entities are sensed. The resulting spatial (or, sometimes, temporal) density has to be exploited for an energy-efficient overall network. Future work puts a focus on how to add the resulting redundancy to the process, and how to combine the mechanisms described in this paper to yield an overall optimized, i.e. energy-efficient, data gathering process. Such a process is obviously the main task of a wireless sensor network.

The preliminary results suggest that the approaches presented here can significantly reduce the amount of energy invested to provide the data sink with requested, aggregated information using an aggregation- and redundancy-aware protocol.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci. Wireless sensor networks: A survey. In *Computer Networks*, volume 38(4), pages 393–422, 2002.

[2] T. Nieberg, S. Dulman, P. Havinga, L.v. Hoesel, and J. Wu. Collaborative algorithms for communication in wireless sensor networks. In T. Basten, M. Geilen, and H. de Groot, editors, *Ambient Intelligence: Impact on Embedded Systems Design*, pages 271–294. Kluwer Acad. Publishers, 2003.

[3] S.-J. Park and R. Sivakumar. Sink-to-sensors reliability in sensor networks. In *Proc. ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Annapolis, MD, June 2003.

[4] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A reliable transport protocol for wireless sensor networks. In *Proc. 1st Workshop on Sensor Networks and Applications*, Atlanta, GA, September 2002. http://www.cse.nd.edu/ surendar/conferences/wsna02/papers/p1-wan.pdf.

[5] A. Willig. Mechanisms for improving single packet delivery probability in wireless sensor networks. In H. Karl, editor, *Proc. 1. GI/ITG Fachgespräch "Sensornetze" (Technical Report TKN-03-012 of the Telecommunications Networks Group, Technische Universität Berlin*, pages 79–82, Berlin, July 2003.