A Probabilistic Acknowledgment Mechanism for Wireless Sensor Networks

Miloš Blagojević[1,2], Majid Nabi[1], Marc Geilen[1], Twan Basten[1,2], Teun Hendriks[2], Marcel Steine[1]
[1] *Eindhoven University of Technology, the Netherlands*
[2] *Embedded Systems Institute, Eindhoven, the Netherlands*
`{mblagojevic, m.nabi, m.c.w.geilen, a.a.basten, m.steine}@tue.nl,teun.hendriks@esi.nl`

Abstract—**The inherently unreliable communication infrastructure compel WSN protocols to employ error control mechanisms. Traditionally, error control is achieved by a retransmission scheme using acknowledgment mechanisms. WSN architectures are severely resource constrained and the additional energy expense of transmitting error control messages can seriously degrade network lifetime.**

In this paper, we analyze performance of error control schemes for the case of point-to-multipoint communication. An explicit acknowledgment mechanism may provide for reliable communication, but has two major drawbacks: 1) the overhead is significant for small data messages, and 2) in case of asymmetrical communication links, multi-hop dissemination of acknowledgments is required. As an alternative to such explicit acknowledgment schemes we propose the use of probabilistic acknowledgments. In this probabilistic scheme, a sender estimates the probability that a message has been successfully delivered, based on information about the quality of the radio channel. A message is then retransmitted until the probability of successful delivery reaches a defined threshold value. Network capacity available for error control can be distributed prudently among all information items to be disseminated, possibly taking into account different application requirements. We formulate a retransmission control strategy which results in minimal latency and maximal message delivery ratio.

Keywords- error control; point-to-multipoint; wireless sensor network; retransmission control; gossiping;

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a complex system of spatially distributed sensor nodes with a goal to produce globally meaningful information from locally collected data. The nodes communicate wirelessly, operate autonomously and perform cooperative actions. In order to make good use of the locally collected data, nodes have to collaborate with each other to perform common application(s). An application defines the functionality, requirements, and target environment for the designed WSN. In general, a WSN has to be self-adaptive, resilient to various types of errors and provide efficient mechanisms for multi-hop information dissemination. These goals have to be met in an architecture that is constrained by limited processing capability, unreliable communication resources, and scarce energy resources [5]. These constraints emphasize the need for energy efficient, scalable and robust techniques for data dissemination.

Due to the probabilistic nature of the radio channel, occasionally packets in a WSN are not delivered successfully to its receiver. The success of packet delivery is also affected by various energy optimizations in communication protocols. Typically such optimizations trade off communication reliability for network lifetime [1]. In order to build (semi)reliable packet delivery on top of unreliable communication infrastructures, an error control mechanism is required. The error control can be implemented as an automatic repeat request protocol (ARQ) [22], as a forward error correction protocol (FEC) [20][2] or, as it is often the case, a combination of these two protocols. One contribution of this paper is an analysis of the efficiency of the ARQ protocol in WSN, for the case of point-to-multipoint communication.

The ARQ protocol consists of message (re)transmissions and acknowledgments ([14,19,16]). Each time a node receives a message, it sends an acknowledgment to the sender. The unreliable radio channel affects the acknowledgment delivery as well. If the sender does not receive any acknowledgment in the specified time interval, it retransmits the message. In practice, the sender node makes a bounded number of attempts to successfully deliver a message. Such a protocol is said to be semi-reliable [12]. In order to increase the chance of acknowledgment delivery, some protocols transmit several acknowledgments for each received message. The optimal number of acknowledgments per received message, which provides the minimal number of total transmissions (data and acknowledgments) is calculated in [22] for the case of point-to-point communication.

The error control mechanism with explicit acknowledgments has several disadvantages when applied in a WSN. First, sending an explicit acknowledgment requires communication resources. For small data messages (often the case in WSN), the acknowledgments create significant overhead, which is present even on very good channels [12]. This problem aggravates in the situation when a message has to be delivered to multiple neighbor nodes, e.g. in multicast or gossiping. Radio communication is a broadcast by nature and many communication protocols exploit that fact and use a single transmission to deliver a packet to multiple destinations. However, the ARQ protocol does not have the same benefits. Each of the receiver nodes has to send an acknowledgment, which increases the communication overhead proportionally to the number of receivers. Secondly, the radio communication is typically not symmetrical and often even not bidirectional [13,24] e.g., in the case when nodes use heterogeneous transmission power. In such cases, multihop dissemination of acknowledgments is required. Multihop dissemination of acknowledgments increases method complexity and communication overhead.

The contributions of this paper are as follows. First, we provide an analytical study of an ARQ protocol with explicit acknowledgments in case when a sender node has to deliver messages to multiple receivers. The sizes of messages and acknowledgments are assumed to be the same. One of the main concerns in WSN is energy efficiency. We derive optimal parameters for the explicit acknowledgment mechanism such that the average number of transmissions per message is minimal.

Second, as an alternative to the explicit acknowledgment mechanism we propose a probabilistic acknowledgment mechanism. Greater communication flexibility is achieved if, instead of using explicit acknowledgments, a protocol estimates the probability that data has been delivered successfully. The probability that a packet is delivered successfully is then used as a probabilistic acknowledgment. A sender node (re)transmits a message until the probability of success reaches a certain value, denoted as the retransmission threshold. The probabilistic acknowledgment is estimated based on the channel properties (packet reception ratio) and the number of previous (re)transmissions. Our probabilistic acknowledgment mechanism does not require any acknowledgment packets and thus provides a smaller expected number of total transmissions, for a price of semi-reliable communication. Performance is not guaranteed, but instead we design the WSN for a certain probabilistic performance.

Third, we propose a retransmission control strategy to select between different messages, based on probabilistic acknowledgment values. The sender node does not know the outcome of its transmissions and will continue retransmitting the message until an acknowledgment is obtained (e.g. the retransmission threshold is reached). We assume the case with strictly limited communication resources (typical for WSN). In such a scenario, it often occurs that a node has several messages in the transmission buffer for which it still has not obtained an acknowledgment. In each communication round, a node can transmit only a fixed, small number of the messages from the buffer. The selection of messages to be (re)transmitted has impact on a performance (latency and delivery ratio). In order to improve performance the sender has to retransmit messages with the highest transmission benefits. The transmission benefit of a message is estimated based on the corresponding probabilistic acknowledgment values. The definition of transmission benefit in our method is flexible and can take into account different application requirements. In [21] we use proposed mechanism in such scenario.

Finally, we evaluate our proposed probabilistic acknowledgment mechanism in an environment monitoring application. The probabilistic acknowledgment mechanism reduces energy cost of the error control. The obtained energy savings depends on the number of intended receivers. The proposed method can be successfully applied in heterogeneous networks with asymmetrical links. In addition, the redundancy control strategy provides a significant improvement in the performance, i.e. latency and delivery ratio. The strategy is devised locally, optimizing hop-by-hop behavior. However, improvement in the end-to-end metrics is significant as well.

The paper is structured as follows. In Section II, we introduce a motivating case study and define performance metrics. The details of the ARQ protocol in case of point-to-multipoint communication are analyzed in Section III; mathematical models for explicit and probabilistic acknowledgment mechanisms are given. In Section IV the retransmission control strategy based on the concept of probabilistic acknowledgments is introduced. Section V evaluates the performance of the proposed techniques in the case of gossiping-based message dissemination. Section VI concludes.

II. MOTIVATING CASE STUDY

WSNs are used as a platform for large-scale applications and can perform different tasks such as: continuous monitoring, event-based reporting, tracking, and so forth. A WSN application defines the functionality, requirements, and target environment for the designed WSN system [12]. In this paper we consider an environment monitoring application with the goal to monitor a set of sensors deployed at known positions, i.e., to perform spatial mapping. Furthermore, it can be assumed that nodes are more or less static and that data need to be collected at regular intervals [18]. The application has to provide that data can be extracted from the network at multiple locations, i.e. data has to be delivered to multiple sink nodes (in the extreme case to all nodes in the network). As a link-layer we use a TDMA-based MAC protocol such as [15] or [1]. Each node can transmit only a small and constant number of packets per TDMA frame.

A. *WSN model*

A wireless sensor network is modeled by a connected directed weighted graph $G=(V,E,p)$, where the set V of vertices denotes the nodes in the network, the set $E \subseteq VxV$ of directed edges captures the potential communication links, and p: $E \rightarrow [0,1]$ is a function such that weight $p(i,j)$ specifies the probability that a packet sent by node i is successfully received by node j (assumed spatially and temporally independent and temporally constant). A node might not successfully receive a packet due to various reasons, e.g. due to the inherently unreliable communication infrastructure, power saving operations of a receiver, or due to interference caused by transmissions in the neighborhood [7].

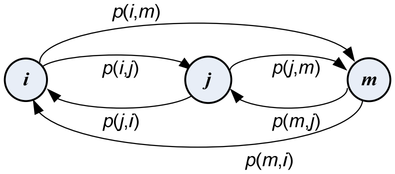


Figure 1. Graph representation of a WSN.

For the sake of simplicity, we assume that the wireless channel between nodes i and j is modeled as a Binary Symmetric Channel (BSC) with constant bit error probability $p_b(i,j)$. This model provides that for a fixed packet length l the outcome of individual packet transmissions is an independent and identically distributed variable according to a Bernoulli distribution with success parameter $p_r(i,j)$:

$$p_r(i,j) = (1 - p_b(i,j))^l \, . \qquad (1)$$

Communication is organized in TDMA periods, rounds. Each node has its own transmission slots that are uniquely assigned in a 2-hop neighborhood, providing a collision free communication schedule. The number of transmission slots per communication round is a MAC parameter. In order to save energy, the communication schedule is made such that in each time frame only a part of the neighbors of a given node turn their receivers on and listen for possible transmission. The MAC success rate $p_{mac}(i,j)$ is defined as the probability that a transmitted message is received successfully, i.e., the probability that when node i transmits a packet, node j has its receiver on and listens for transmission. In the selected TDMA protocol, all nodes have a fixed transmission schedule. In each round a node chooses one of N_L receive schedules. Assuming that the receive schedules partition the set of slots, the probability that a receiver will listen in a sender transmission slot, i.e., the MAC success rate probability is given by:

$$p_{mac}=1/N_L \tag{2}$$

The fraction of packets transmitted at node i that is successfully received at node j depends on both radio and MAC properties. In this case study, the radio and MAC probabilities are per-transmission independent; thus $p(i,j)$ is:

$$p(i,j)=p_{mac}(i,j)\cdot p_r(i,j)=p_{mac}\cdot p_r(i,j) \tag{3}$$

The assumptions made in this section simplify our analysis, but are not crucial for applicability of the proposed solutions. The simplified WSN model allows fast and adequately accurate performance evaluation [4]. In a real deployment, instead of using simplified models, each node can estimate the required probabilities as an average packet reception ratio in some period of time. Each node requires just information about local packet reception ratios, i.e., probabilities corresponding to the edges incident to that node.

B. Environment monitoring application

An environment monitoring application performs measurement of some physical phenomena at a regular sampling period. The measurement results are distributed using data messages. A message consists of three fields: 1) a key number k that uniquely identifies the type of information contained in that message, e.g., identifying the node where the measurement was taken; 2) a measured value m; and 3) a version number v corresponding to the time when the value was measured. Communication is multicast; we assume that each type of information k has to be delivered to a set of intended destination nodes (sinks) ID(k).

For simplicity of presentation, we assume that nodes have unique ids that function as message keys (both nodes and message keys are denoted with the same symbol k). Each node takes sensor measurements at a regular period and generates data messages with its own key. A new version of a message is generated with the regular sampling rate f_{smp}. Furthermore, we assume that sink nodes are interested only in the latest version of the measured values. Thus, as soon as a node receives a newer version the old one is overwritten. This ensures that in the transmit buffer of any node in the network there is at most one version of each measurement. Finally, we assume that the size of the transmission buffer is proportional to the number of message keys, providing that no messages have to be discarded in favor of other messages because of limited buffer size.

C. Performance metrics

In this paper, we analyze properties of an ARQ protocol in a WSN with strict constraints on available resources. The intended goal is to propose a probabilistic ARQ protocol that improves WSN performance metrics: energy consumption, latency and delivery ratio.

Typically radio operations expend the most energy in comparison with other components of a WSN [17,12]. The energy cost of a single transmission consists of two parts: 1) the energy cost of transmitting a message at the sender node and 2) the energy cost of receiving a message at one or more receiver nodes. Radio communication is broadcast in its nature and one message transmission leads to a number of acknowledgment transmissions proportional to the number of receivers. The data messages in WSN are rather small and it can be assumed that the size of the message and the size of an acknowledgment are the same [22]. We define energy cost of the ARQ protocol as the expected number of transmissions required to successfully finish the process of delivering a single message.

WSNs are often characterized with two other essential properties: the expected amount of nodes that ultimately receive a message and the expected time needed to deliver a message. In this work, these two properties are evaluated through two metrics, called delivery ratio and latency. These metrics can be evaluated on a hop-by-hop basis and on an end-to-end basis. Hop-by-hop metrics consider the process of message delivery from the sender node to the set of intended neighboring receivers, while the end-to-end metrics consider message delivery from source to the set of intended destinations (sinks).

The time necessary to successfully transmit a message between two nodes is not fixed, but a random variable that depends on the system properties, e.g., the probabilistic outcome of individual transmissions. Hop-by-hop latency is defined as the expected time necessary to successfully deliver a message from the current node to all indented receivers. Similarly, end-to-end latency is defined as the expected time necessary to successfully deliver a message from source to all sinks. Delivery ratio represents the average ratio of messages delivered successfully. Due to the probabilistic nature of the transmission process, a newer version of the same message may be distributed faster than an old one. The message is not successfully delivered in the case when a node receives the newer message version before the current one is successfully transmitted. In that case, the node discards the old version from the buffer and restarts the retransmission process with the new version. Hop-by-hop delivery ratio is defined as the expected ratio of intended receivers that actually receive a message. Similarly, end-to-end delivery ratio is defined as the expected ratio of sinks that receive a message.

The ARQ procedure proposed in this paper is applied at the hop-by-hop level. Each node has a limited knowledge about the network topology (immediate neighborhood). In Section IV, we propose a retransmission control strategy, where each node, based on the available information, optimizes local hop-by-hop metrics. Simulation results presented in Section V show that such locally-optimal decisions lead to a significant improvement in end-to-end metrics.

III. POINT-TO-MULTIPOINT ACKNOWLEDGMENT MECHANISM

WSNs are characterized by inherently unreliably radio communication, which occasionally results in lost and corrupted data. An error control mechanism has to provide a certain level of reliability without exceeding the available energy budget. This problem aggravates in the case of point-to-multipoint data delivery (Figure 2).

A. *Explicit acknowledgment*

The ARQ protocol with an explicit acknowledgment mechanism is a commonly used error control technique [22]. In the situation when data has to be delivered to multiple receivers, the sender node keeps retransmitting data until acknowledgments from all intended receivers have arrived. The set of intended receivers $R(i,k)$, denotes a subset of the neighbors of a node i that are either the destination for a transmitted message k or that relay that message. The elements of the set of intended receivers are defined by a routing protocol, e.g. multicast, gossiping, or shortest-path routing. In the case of multicast communication, data has to be delivered to a precisely defined set of multicast peer nodes [7]. Similarly, in the case of gossiping based communication, the sender has a goal to deliver data to multiple receivers [10,11]. The difference is that in the case of gossiping, the node does not need to deliver data to all receivers. In order to provide successful gossiping dissemination, it is sufficient to provide that, with sufficient probability, enough receivers obtain the message [11].

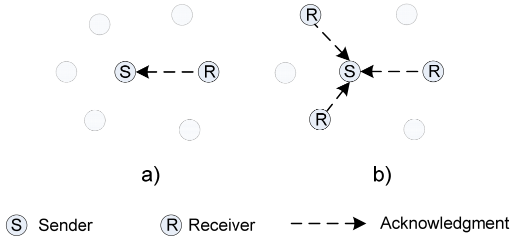


Figure 2. Different communication paradigms

a) point-to-point b) point-to multipoint

As a generalization of the point-to-multipoint error control scenario, we consider the following problem: A sender node transmits a message that has to be delivered to at least N_{min} of N_R receivers. After transmission, the sender waits for acknowledgments. If a sufficient number of acknowledgments do not arrive within a specified timeout interval the sender retransmits the message. Typically, the timeout interval is selected to provide sufficient time for a receiving node to finish its acknowledging procedure. The message might require retransmission for a particular receiver, due to two possible reasons: 1) the receiver does not receive the message; 2) the message is successfully received, but the acknowledgments generated by the receiver do not reach the sender. In order to increase the probability that an acknowledgment is delivered successfully the receiver node can send multiple acknowledgments for each received message (u acknowledgments) [22].

The explicit acknowledgment protocol with multiple acknowledgments per received message is described with Algorithm 1. For each message (*msg*) in the transmission buffer (*Tx_buffer*) the sender maintains a list, *rec_ack*, of neighbors that successfully acknowledged message reception. A message is (re)transmitted until the number of received acknowledgments n_{ack} is at least the minimal number of required acknowledgments N_{min}.

Algorithm 1. Explicit acknowledgment

```
SENDER:
MESSAGE_INIT(msg) {
    $n_{ack}$:=0;
    rec_ack:=Ø;
}
SENDER_TRANSMIT(msg) {
    Transmit msg;
    msg.timeout.init;
}
SENDER_RECEIVE(ack) {
    if(ack is received from node j)
        if($j \notin$ rec_ack)
            rec_ack:=rec_ack $\cup$ {j};
            $n_{ack}$:= $n_{ack}$+1;
        if($n_{ack} \geq N_{min}$)
            Tx_buffer.remove(msg);
}
RECEIVER:
RECEIVER_ACK(msg) {
    Schedule u acknowledgments for transmission;
}
```

According to the system description made in Section II we assume that every individual packet transmission from node i to node j is an independent and identically distributed process with a Bernoulli distribution with success parameter $p(i,j)$. Each time the message is successfully received, the receiver node sends u acknowledgments. The probability that sender i does not receive any of these u acknowledgments is equal to: $(1-p(j,i))^u$. Based on this, it is easy to obtain a success probability of a single transmission $p_s(i,j,u)$, i.e., a probability that the message is delivered and acknowledged successfully after a single transmission. It is calculated as:

$$p_s(i,j,u) = p(i,j) \cdot \left(1 - \left(1 - p(j,i)\right)^u\right). \tag{4}$$

The geometric distribution describes the number of Bernoulli trials needed to get one success. The probability $p_{not}(i,j,u,r)$ that after r (re)transmissions, with $r \geq 1$, sender i still has not received an acknowledgment from the intended receiver j, is the following function (monotonically decreasing in u and r):

$$p_{not}\left(i,j,u,r\right) = \left(1 - p_s\left(i,j,u\right)\right)^r \,. \tag{5}$$

According to the protocol, a transmission is considered successful and a sender node stops retransmitting a message as soon as acknowledgments from at least N_{min} intended receivers are obtained. The probability that acknowledgments from at least N_{min} intended receivers $R(i)$, arrived successfully in the case when sender node i made r retransmissions, is denoted as success probability $p_S(i,u,r,N_{min})$. It is computed as:

$$p_S(i,u,r,N_{\min}) = 1 - \sum_{n=0}^{N_{\min}-1} p_d(i,u,r,n)\,, \tag{6}$$

where $p_d(i,u,r,n)$ denotes the probability that *exactly n* intended receivers have successfully acknowledged the message. That probability is calculated as the sum of probabilities that a specific subset of intended receivers $\{n_1,\dots,n_n\}$ has successfully acknowledged the message, over all possible combinations of subsets of n intended receivers.

$$p_d(i,u,r,n) = \sum_{\{n_1,\dots,n_n\}\in R(i)} \left(\prod_{a=1}^{n} \left(1 - p_{not}(i,n_a,u,r)\right) \cdot \prod_{b\in R(i)/\{n_1,\dots,n_n\}} p_{not}(i,n_b,u,r) \right) \tag{7}$$

The expected number of (re)transmissions at the sender node required to deliver a message and obtain acknowledgments from a sufficient number of receivers is calculated as:

$$N_{ret}(i,u) = \sum_{r=1}^{\infty} r \cdot p_S(i,u,r,N_R)\,. \tag{8}$$

Intended receiver j does not successfully receive all transmissions. The expected number of transmitted message copies that are received at node j is $p(i,j)N_{ret}(i,u)$. When a node receives a message it sends u acknowledgments. Therefore, the expected number of sent acknowledgments from the intended receiver j is:

$$N_{ack}(i,j,u) = u \cdot p(i,j) \cdot N_{ret}(i,u)\,. \tag{9}$$

The number of acknowledgments sent per received message, u, should be selected so that the total number of transmissions required to successfully deliver a message is minimal. The total number of transmissions includes message retransmissions and all acknowledgment transmissions made by the intended receivers.

$$u_{opt} = \arg\min_u \left(N_{ret}(i,u) + \sum_{j=1}^{N_R} N_{ack}(i,j,u) \right). \tag{10}$$

The optimal value u_{opt} is calculated numerically for different topology parameters and experimentally evaluated in Section V.

Taking all the above together, the energy cost of a point-to-multipoint ARQ protocol is estimated as:

$$N_{tot}(i,j) = N_{ret}(i,u_{opt}) + \sum_{j=1}^{N_R} \left(p(i,j) \cdot u_{opt} \cdot N_{ret}(i,u_{opt}) \right). \tag{11}$$

A considerable part of communication resources is spent on the acknowledgment transmissions (second part of Eqn. 11).

B. *Probabilistic acknowledgment*

Greater efficiency can be achieved if instead of depending on explicit acknowledgments, a protocol uses probabilistic acknowledgments. A sender node estimates the probability that a message is delivered successfully to a sufficient number of intended receivers. This probability is considered to be a probabilistic acknowledgment. Success probability is estimated based on the channel properties (packet reception ratio) and the number of previous (re)transmissions. Each retransmission increases the probability that a data item is delivered successfully and therefore decreases the (statistical) need for further retransmission. The better is the link quality, the faster the need for retransmission decreases. A sender node (re)transmits a message until the success probability reaches a certain value, denoted as the retransmission threshold, p_{TH}.

The mathematical model derived for the explicit acknowledgments can be applied with minor changes to probabilistic acknowledgments. A probabilistic acknowledgment is calculated as the probability that after r (re)transmissions at least N_{min} intended receivers have received the message. Eqns. 6 and 7 can be used to calculate this probability. However, the number of acknowledgments per message is not a parameter anymore (since there are no acknowledgments), and $p_{not}(i,j,u,r)$ in Eqn 7 is simplified to the probability that after r (re)transmissions the receiver still has not received the message:

$$p_{not}\left(i,j,r\right) = \left(1 - p\left(i,j\right)\right)^r\,. \tag{12}$$

The generic probabilistic acknowledgment mechanism can be described with Algorithm 2.

Algorithm 2. Probabilistic acknowledgment

```
SENDER:
Message_init(mes) {
    r=0;
}
Sender_transmit(mes) {
    Transmit message;
    r=r+1;
    if($p_S(i,r,N_{min}) \geq p_{TH}$)
        Tx_buffer.remove(msg);
}
```

The number of message retransmissions depends on the retransmission threshold and can be estimated as:

$$r_{th} = \min\left\{ r \mid p_S(i,r,N_R) > p_{TH} \right\}. \tag{13}$$

The retransmission threshold value provides a trade-off between the delivery ratio, and the number of retransmissions that a sender will perform, i.e., the corresponding energy cost. The more times a message is retransmitted, the probability that it is successfully received increases, but the adjoined energy costs increases as well. It is important to notice that, while the improvement in success probability levels off with an increasing number of retransmissions, the corresponding resource usage does not.

The retransmission threshold can be selected based on the desired end-to-end delivery ratio. For example, in the situation when a message has to be propagated over a single path with *m* hops the end-to-end delivery ratio will be: p_{TH}^{m} (assuming that for each hop the same retransmission threshold p_{TH} is applied).

One of the advantages of a probabilistic acknowledgment is that the sender does not need to wait for a timeout interval in order to retransmit a message. The proposed mechanism allows a message to be retransmitted in the next time slot. However, the communication resources are limited and the sender can use only a limited number of transmission slots per communication round. The selection of messages to be (re)transmitted in the next round has a profound effect on performance. This problem is analyzed in the next section.

IV. RETRANSMISSION CONTROL STRATEGY

In the ideal case a node should stop retransmitting a message at the moment when the required number of intended receivers has received the message. However, the sender does not know the outcome of the message transmission. The ARQ protocols considered in this paper (re)transmit the message until the sender receives an acknowledgment or exceeds a retransmission threshold. If a message is retransmitted more often, the probability of successful reception increases; however this increase levels off with the number of retransmissions already made.

In the general case, a sender has multiple messages that are still not acknowledged and waiting for (re)transmission. It can easily happen that the number of messages waiting for retransmission is bigger than the number of messages that can be transmitted in a communication round. Therefore, in each round a node has to decide which messages will be transmitted. In order to provide efficient usage of the available resources a retransmission control strategy is required.

The role of each node in the message dissemination is defined by the applied communication protocol. A node can be a message source, message destination or a relaying node. Each node has a transmission buffer where it stores its own messages and the messages it relays. In the general case, a message is removed from the transmission buffer in two situations: 1) the ARQ protocol has obtained an acknowledgment (explicit or probabilistic threshold); 2) the buffer is full and space for a new message is needed. In our motivating example, we assumed that the buffer is big enough to store exactly one message from each message source. In each communication round a node selects N_S messages from the transmission buffer and broadcasts them. If there are no more than N_S messages in the transmission buffer, then all messages are selected.

Each node has only information about its local neighborhood. In order to provide better end-to-end performance each node uses a retransmission control strategy devised with a goal to optimize local hop-by-hop behavior. Each sender has two goals: 1) to successfully deliver as many messages as possible and 2) to deliver them in the shortest possible time. In the rest of this section, we formulate an efficient retransmission control strategy for two different scenarios. In the situation when sampling rate is high, messages get lost (overwritten) and a sender's first priority is to improve hop-by-hop delivery ratio. On the other hand, in the situation when sampling rate is low, no message is lost and a sender optimizes hop-by-hop latency.

A. Scenario 1. High sampling rate

In case of a high sampling rate, the available communication resources are not sufficient to provide successful delivery of all messages. The message is considered a lost message when a newer message version arrives to the node, while the version currently in the buffer has not been successfully delivered yet. The goal of our retransmission control strategy is to maximize delivery ratio.

Let us assume that in the transmission buffer messages from *N* sources are stored. The number of retransmissions of a message *k* (message from source *k*) prior to round *t* is denoted as $r(k,t)$. Based only on local information, for each message *k* in the transmission buffer, a sender can estimate the probability that a message is successfully delivered to the required number of receivers. That probability is given with Eqn. 6. For the sake of clarity, we simplify the notation to $p_S(k,r(k,t))$. The sender does not know the outcome of individual transmissions, but it can calculate $N_{del}(t)$, the expected number of delivered messages prior to round *t*. Considering just the messages present in the transmission buffer in the round *t*, the expected number of delivered messages prior to that round, is given as:

$$N_{del}(t) = \sum_{k=1}^{N} p_S(k, r(k,t)), \qquad (14)$$

where $p_S(k, r(k,t)) = 0$ for all messages that are not in the buffer.

Let us denote the set of messages selected to be transmitted in round *t* as $Sel(t)$. The goal is to select messages such that the benefit achieved by their transmission is greatest. In other words, the goal is to provide a maximal increase in the expected number of delivered messages, $\Delta N_{del}(t)$:

$$\Delta N_{del}(t) = \sum_{k=1}^{N} S(t)\Delta p_S(k, r(k,t)+1), \qquad (15)$$

where $S(t)$ represents selection function:

$$S(t) = \begin{cases} 0 & , k \notin Sel(t) \\ 1 & , k \in Sel(t) \end{cases}, \qquad (16)$$

and $\Delta p_S(k,r+1)$ denotes improvement in the success probability for message *k* achieved with a new (re)transmission of that message. This improvement is given as:

$$\Delta p_S(k, r(k,t)+1) = p_S(k, r(k,t)+1) - p_S(k, r(k,t)). \qquad (17)$$

Now, it is straightforward to formulate the retransmission control strategy that maximizes the expected number of delivered messages, i.e. expected hop-by-hop delivery ratio (Fig. 3). In each round, for each message, the sender node estimates the achievable retransmission improvement and selects messages that provide the greatest improvement. It should be noticed that retransmission improvement is estimated based on probabilistic acknowledgments (Eqn. 6).

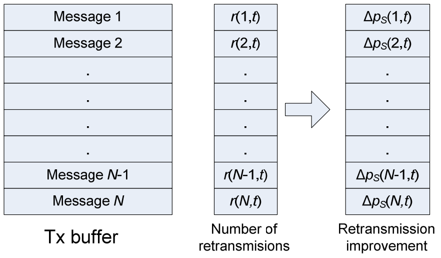


Figure 3. Retransmission control strategy

B. Scenario 2. Low sampling rate

As a second scenario, we consider the situation where the sampling rate is low and the available communication resources are sufficient to provide successful delivery of (almost) all messages. Although all messages get delivered, the sender still benefits from the efficient retransmission control strategy. In this scenario, a sender's goal is to minimize the expected hop-by-hop latency. For example, the hop-by-hop latency will be higher if the sender selects a message that was already retransmitted several times (a message near to the probabilistic threshold), instead of the message that was not retransmitted before.

It is relatively easy to show that the same retransmission control strategy which provides maximal delivery ratio in Scenario 1 provides minimal latency in Scenario 2. Similarly as before, the retransmission procedure should select those messages whose retransmission minimizes expected increase in the latency. For each message that after round *t* remains unsuccessfully delivered, latency is increased by one round. Thus, minimizing the expected average hop-by-hop latency is equivalent to the goal of minimizing the number of messages that remain undelivered after round *t*, i.e., it is equivalent to maximizing the increase in the expected number of delivered messages in the round *t*. This is the same requirement as in Scenario 1.

V. RESULTS AND DISCUSSION

The performance of the proposed probabilistic acknowledgment and retransmission mechanism is evaluated in terms of its energy cost and network performance (delivery ratio and latency).

A. Probabilistic acknowledgment – energy cost

The explicit acknowledgment mechanism in case of point-to-point communication is thoroughly analyzed in [22]. It is shown that in order to provide the minimal number of total transmissions, receiver node *j* should send $u_{opt}=round(1/p(i,j))$ acknowledgments for each message received from sender node *i*. This result is not optimal in the case of point-to-multipoint communication. The optimal number of acknowledgments per receiver, in case of point-to-multipoint communication, is given with Eqn. 10.

In Figure 4, the optimal number of acknowledgments per message is compared for cases of point-to-point and point-to-multipoint ARQ protocols with explicit acknowledgment. As an example, for point-to-multipoint communication, we assume a scenario where the sender has 10 intended receivers and keeps retransmitting a message until it receives acknowledgments from at least 5 intended receivers. For the sake of presentation simplicity, the packet reception ratios of all links connecting the sender with intended receivers are assumed to be equal (x-axis in Figure. 4).

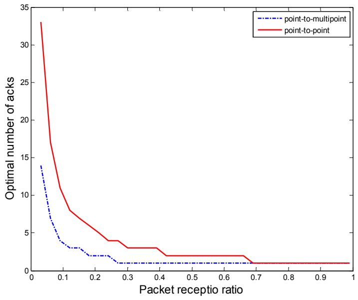


Figure 4. Optimal number of acknowledgments per received packet

The obtained results are as expected. In the point-to-multipoint ARQ protocol the number of acknowledgment transmissions is proportional to the number of intended receivers (Eqn. 11). The more intended receivers a node has, the bigger the imbalance between energy spent on message transmissions and energy spent on acknowledgment transmissions is. Therefore, the benefit of sending multiple acknowledgments per message is smaller. From Fig. 4, it can be observed that the optimal number of acknowledgments sent per received message is noticeably smaller in case of point-to-multipoint communication, than it was in case of point-to-point communication. In a situation when the packet reception ratios are higher than 0.28, for this example, the optimal number of acknowledgments to be sent per message is one.

The energy cost of the ARQ protocol is estimated as the total expected number of transmissions made in attempt to successfully deliver a message (Eqn. 11). In Fig. 5, the number of retransmissions required to deliver a message to a sufficient number of intended receivers is given by the dashed line. This number of retransmissions corresponds to the ideal case when a sender instantaneously receives information about a successful transmission. However, the sender does not know when the message is delivered and continues retransmitting the message until an acknowledgment is received. In the case of explicit acknowledgments, the total number of transmissions consists of two parts: message (re)transmissions and acknowledgment transmissions (Eqn. 11). The imbalance between these two parts in case of point-to-multipoint communication is shown in Fig. 5.

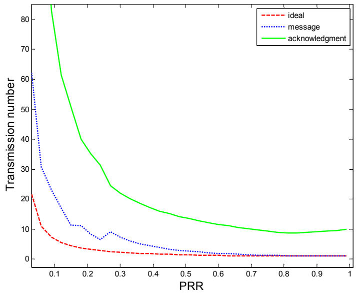


Figure 5. Explicit acknowledgment - average number of transmissions

a) ideal case b) message transmissions c) acknowledgment transmissions

In case of the probabilistic acknowledgment mechanism, receivers do not send acknowledgments. The number of retransmissions required to the reach retransmission threshold is a constant value (Eqn. 13). It should be noted that in case of the explicit acknowledgment mechanism the number of retransmissions per message is variable and the expected value is used for comparison (Eqn. 8).

The energy cost of the explicit and probabilistic acknowledgment is compared in Fig. 6. Results are presented for the given scenario (10 intended receivers where the message has to be delivered to at least 5) and the threshold value of 0.999. The probabilistic acknowledgments provide a smaller energy cost because receivers do not send acknowledgments.

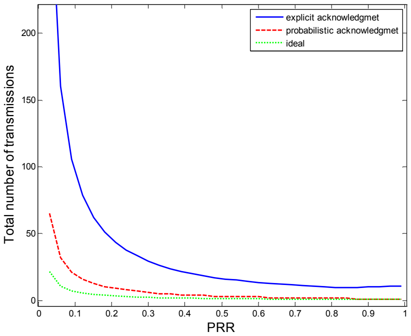


Figure 6. Explicit vs. probabilistic acknowledgment mechanism

The energy reduction achieved by probabilistic acknowledgments depends on several parameters, such as the number of intended receivers, N_R, minimal number of intended receivers that have to receive a message, N_{min}, the selected retransmission threshold, packet reception ratio. The achieved energy reduction, for a different parameter values is presented in Table I. *Eff*(p), for a given packet reception ratio p, denotes the ratio between energy costs of the probabilistic and explicit acknowledgments. The retransmission threshold value was set to 0.999 in all given examples.

TABLE I. ENERGY COST COMPARISON

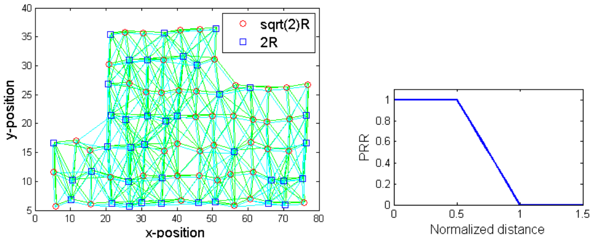
N_R	N_{min}	*Eff*(0.3)	*Eff* (0.6)	*Eff* (0.95)
1	1	2.05	1.73	1.32
2	1	1.22	1.02	0.69
2	2	1.05	1.06	0.80
4	2	0.58	0.49	0.28
4	4	0.54	0.54	0.45
6	3	0.35	0.32	0.15
8	4	0.25	0.22	0.12
8	8	0.25	0.25	0.23
10	5	0.20	0.17	0.09

From the results shown in Table I it can be seen that the probabilistic acknowledgment mechanism has a smaller energy cost in all cases where the number of intended receivers is more than two. The more intended receivers a sender has, the greater the benefit of using probabilistic acknowledgments is. The effect of the link quality is also an important factor. The gain is larger for better quality links.

B. *Retransmission control strategy - network performance*

The goal of the retransmission control strategy is to improve the network performance, i.e., to reduce latency and increase delivery ratio. We apply the proposed hop-by-hop error control mechanism together with the retransmission control strategy and evaluate their effect on the end-to-end system metrics.

The performance evaluation is performed with the use of the PGM WSN discrete event simulator [4]. The simulation parameters are selected according to the selected WSN platform, MyriaNed [23]. The native MAC protocol for this platform is gMac [1]. Each node has up to four transmission slots per communication round. Simulations are performed for an area shape consisting of three rectangular parts (Fig. 7a). Within the area, nodes are positioned in a grid-like structure with a spacing R. Each node was allowed to drift randomly from the grid position by up to $0.4R$ per axis. The network is heterogeneous, consisting of two types of nodes with different transmission power: nodes with a transmission range $\sqrt{2}R$ (that can usually reach their horizontal and vertical direct neighbors and sometimes diagonal neighbors) and nodes with transmission range $2R$ (can typically reach their direct horizontal, vertical or diagonal neighbors and sometimes 2-hop horizontal or vertical neighbors). Simulations are performed for 5 different deployment sizes, containing 83, 121, 155, 200 and 251 nodes. The simulation durations were sufficiently long to provide a 99% confidence interval with relative margins of error less than 5%. For modeling radio behavior, the stochastic radio model of [6] is used. The packet reception ratio for nodes which distance is less than half of the transmission range is assumed to be one, and it linearly goes to zero for the case when distance is equal to the transmission range (Fig. 7b).



(a) (b)

Figure 7. a) Grid-like network deployment and b) radio model

The efficiency of the retransmission control strategy is illustrated through a gossiping case. As a gossiping protocol we use SharedState, which provides a probabilistic mechanism for message dissemination to all nodes in the network [10]. In each communication round a sender node randomly selects messages to be transmitted from its transmission buffer. Our probabilistic acknowledgment mechanism works well for protocols like SharedState. The SharedState random selection procedure is replaced with probabilistic acknowledgments together with our retransmission control strategy.

In Fig. 8, the gossiping performance is compared for the case of plain random selection and the selection procedure defined with the proposed retransmission control strategy. The message sampling rate was selected to be high enough to provide a congested network and thus a noticeable amount of undelivered messages (about 65%). We compare the end-to-end latency and delivery ratio of gossiping with and without a retransmission control strategy. The results show that retransmission control with the probabilistic acknowledgments improves performance significantly. Improvement in the performance increases with the network size; for 83 nodes, latency is reduced 4 times, while for 251 nodes, the reduction is 5.5 times. A similar trend is observed for delivery ratio, but the relative improvements are different, i.e., for 83 nodes, delivery ratio is increased 1.6 times, while for 251 nodes, the increase is 2.2 times. The improvements in end-to-end metrics increase with the number of nodes in the network. This trend is expected; in a case of all-to-all gossiping each node is a message source, and increase in number of sources leads to heavier network congestion. Furthermore, with the heavier congestion the effect of a smart retransmission procedure becomes more visible.

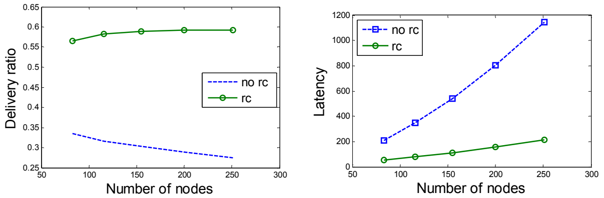


Figure 8. End-to-end delivery ratio and latency with and without retransmission control strategy

VI. CONCLUSIONS

The probabilistic acknowledgment mechanism proposed in this paper provides several advantages compared to the traditional ARQ methods.

The proposed mechanism does not depend on explicit acknowledgments to decide about message retransmissions. Therefore, it can be successfully applied even in heterogeneous networks with an asymmetric communication infrastructure.

The method is especially suitable for point-to-multipoint communication. It benefits from the broadcast nature of the radio communication and provides an energy efficient alternative for traditional ARQ protocols. Gossiping-based message dissemination is one of the communication paradigms that benefits greatly from the proposed techniques.

The number of retransmissions performed by a sender node represents a trade-off between delivery ratio (success probability) and the resource usage (total number of transmissions). The probabilistic acknowledgment mechanism provides an easy and flexible way to control this trade-off of the system performance through the retransmission threshold value. This can be very useful in the design phase of a system, where the communication resources have to be decided, e.g., the number of transmission slots per communication round.

In order to provide efficient use of the available resources, a retransmission control strategy is proposed. In each round, a sender node estimates the benefit achieved by the retransmission of each individual message. The messages with the greatest transmission benefit are retransmitted. The benefit is estimated in terms of expected hop-by-hop delivery ratio and latency. The performance of the proposed retransmission control strategy is evaluated for an environment monitoring application using a gossiping-based dissemination. Simulation-based performance evaluation demonstrates that the retransmission control strategy significantly improves end-to-end dissemination properties (end-to-end delivery ratio and latency).

The probabilistic acknowledgment is estimated based on information about link quality towards neighbors, expressed as packet reception ratios. In static networks, such information can be obtained with relatively small overhead. In our simulation, packet reception ratios are calculated according to the selected radio model and do not change over time. In such idealized situation, the probabilistic acknowledgment can be accurately calculated. In a real deployment, the packet reception ratios towards neighbours have to be estimated, e.g., with a sliding window mechanism. Initial experiments that we performed show that the proposed method has certain robustness to an estimation error. As a part of future work, we plan to analyze the effects of network dynamics in more detail.

The probabilistic acknowledgment can be combined with some error correction techniques. Implicit acknowledgments [16] can be incorporated easily, i.e., in certain situations a sender can obtain acknowledgments by overhearing receiver transmissions (in the case when receivers continue to relay the same message). Also, the method can be easily combined with forward error correction (FEC) protocols. If applied, FEC would reflect in the estimated packet reception ratios. Consequently it would reflect in the energy efficiency as well. The analysis of such aspects is left for future extension of the probabilistic acknowledgment mechanism.

ACKNOWLEDGMENT

This work was supported by the Dutch innovation program Point-One, through project ALwEN, grant PNE07007.

REFERENCES

[1] P.A. Anemaet, Distributed G-MAC: A flexible MAC protocol for servicing gossip algorithms. Master's thesis, Delft University of Technology, The Netherlands, 2008.

[2] J. Ahn, S. Hong, and J. Heidemann, “An adaptive FEC code control algorithm for mobile wireless sensor networks,” Journal of Communications and Networks, vol. 7, no. 4, pp. 489-499, 2005.

[3] P. Antoniou, A. Pitsillides, “Wireless Sensor Networks Control: Drawing Inspiration from Complex Systems,” Med-Hoc-Net Ionian Academy, 2007.

[4] M. Blagojević, M.Nabi, T. Hendriks, T. Basten, M.C.W. Geilen, “Fast simulation methods to predict wireless sensor network performance,” In Proc. PE-WASUN, ACM, pp. 41-48, 2009.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized Gossip Algorithm,” IEEE Trans. on Information Theory, 52(6), 2006

[6] A. Cerpa, J.L. Wong, L. Kuang, M. Potkonjak, D. Estrin, “Statistical model of lossy links in wireless sensor networks,” In Proc. of IPSN 2005, ACM/IEEE. 2005.

[7] P. Chaporkar, S. Sarkar, "Wireless Multicast: Theory and Approaches," *IEEE Trans. Inform. Theory*, vol. 51, pp.1954-1972, 2005.

[8] S. Chen, N. Yang, “Congestion avoidance based on lightweight buffer management in sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol.17, no.9, pp.934–946, 2006.

[9] P. De, S.K. Das, “Epidemic Models, Algorithms and Protocols in Wireless Sensor and Ad-hoc Networks,” Algorithms and Protocols for Wireless Sensor Networks, John Wiley and Sons, 2008

[10] D. Gavidia, M. van Steen, “A probabilistic replication and storage scheme for large wireless networks of small devices,” in *Proc. 2008 IEEE Mobile and Ad Hoc Sensor Systems (MASS)*.

[11] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-Based Ad Hoc Routing”, in *Proc. 2002 IEEE INFOCOM*.

[12] H. Karl, A.Willig, “Protocols and Architectures for Wireless Sensor Networks,” John Wiley & Sons, 2005.

[13] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless network research. Technical Report TR2003-467, Dartmouth College, July 2003.

[14] K. Kulik, W. Rabiner, H. Balakrishnan. “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proc. 1999 ACM/IEEE Mobicom Conference*. 1999.

[15] L. van. Hoesel, P. Havinga, “A lightweight medium access protocol (LMAC) for wireless sensor networks,” Int. Workshop on Networked Sensing Systems, Tokyo, Japan,2004

[16] S. S. Kulkarni, M. Arumugam, “INFUSE: A TDMA based Data Dissemination Protocol for Sensor Networks,” Technical Report MSU-CSE-04-46, Michigan State University, Nov. 2004.

[17] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hämäläinen, M. Hännikäinen, T.D. Hämäläinen, “Ultra-Low Energy Wireless Sensor Networks in Practice,” John Wiley & Sons, 2007.

[18] K.G. Langendoen, “Medium Access Control in Wireless Sensor Networks,” Chapter in "Medium Access Control in Wireless Networks", Nova Science Publishers, 2008.

[19] P. Levis, N. Patel, D. Culler, S. Shenker, “Trickle: a self-regulating algorithm for code maintenance and propagation in wireless sensor networks,” in *Proc. 2004 USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI).*

[20] R. Ma, L. Xing, T. Jin, T. Song, “A Data Transmission Mechanism for Survivable Sensor Networks,” in Proc. IEEE International Conference on Networking, Architecture, and Storage (NAS), 2009.

[21] M. Nabi, M. Blagojevic, M. Geilen, T. Basten, “Dynamic Data Prioritization for Quality-of-Service Differentiation in Heterogeneous Wireless Sensor Networks,” in Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2011, pp.217-225. Salt Lake City, Utah, USA, June, 2011.

[22] I. Stojmenovic, A. Nayak and J.Kuruvila, “Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer, ” IEEE Communications Magazine (Ad Hoc and Sensor Networks Series), 43, 3, March 2005, 101-106.

[23] F. van der Wateren, “The art of developing WSN applications with MyriaNed,” Tech. report, Chess Company. the Netherlands, 2008.

[24] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In MobiSYS, pages 125–138, 2004.