Latency constrained aggregation in sensor networks

Document status and date:
Published: 01/01/2006

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Download date: 16. Aug. 2022

Latency Constrained Aggregation in Sensor Networks *

Luca Becchetti[1], Peter Korteweg[2], Alberto Marchetti-Spaccamela[1], Martin Skutella[3], Leen Stougie[2,4], and Andrea Vitaletti[1]

[1] University of Rome "La Sapienza"
[2] TU Eindhoven
[3] University of Dortmund
[4] CWI Amsterdam

Abstract. A sensor network consists of sensing devices which may exchange data through wireless communication. A particular feature of sensor networks is that they are highly energy constrained due to their use of batteries. Data aggregation is a possible way to save energy consumption: nodes may delay data in order to aggregate them into a single packet before forwarding them towards some central node (sink). Since transmission is highly energy consuming, aggregation can contribute to increasing network lifetime. However, many applications impose constraints on the maximum delay of data; this translates into latency constraints for data arriving at the sink. Data aggregation, latency constraints and energy preservation give rise to a wide variety of combinatorial optimization problems.

In this paper we study the problem of data aggregation to minimize maximum energy consumption under latency constraints on sensed data delivery. In the problem we study, transmission energy and time depend only on the pair of nodes involved in the transmission and we assume unique transmission paths that form a tree rooted at the sink. Most of our results also hold when minimizing the total energy consumption of all nodes.

We prove that the off-line version of this problem is strongly **NP**-hard and we design a 2-approximation algorithm. The latter uses a novel rounding technique which has potentially wider applicability.

Almost all real life sensor networks are managed on-line by simple distributed algorithms in the nodes. In this context we consider both the case in which sensor nodes are synchronized or not. We consider distributed on-line algorithms and we use competitive analysis to assess their performance. We also provide lower bounds for the models we consider, in some cases showing the optimality of the algorithms we propose. To evaluate the strength of our distributed algorithms we also present competitive results for the centralized model.

1 Introduction

Advancements in wireless and sensor technologies have paved the way for the development of tiny and cheap devices equipped with sensors and wireless transceivers. Such devices, named sensor nodes, are able to monitor events (e.g. seismic activity, animals moving in a forest, enemies or intruders entering a monitored area), to process the sensed information and to communicate the sensed data. A sensor network consists of sensor nodes and one or more central nodes or *sinks*. Sinks are powerful base stations which gather data sensed in the network; sinks either process this data or act as gateways to other networks. The sensed data are sent through a sequence of sensor nodes to the sinks.

A particular feature of sensor nodes is that they are battery powered, making sensor networks highly energy constrained. Replacing batteries on hundreds of nodes, often deployed in inaccessible environments, is infeasible or too costly. Therefore, a key challenge in a sensor network is the reduction of energy consumption and the most natural objective is to minimize the maximum energy consumption over all nodes.

Energy consumption can be divided into three domains: sensing, communication and data processing [1]. Communication is most expensive because a sensor node spends most of its energy in data transmission and reception [11]. This motivates the study of techniques to reduce overall data communication, possibly exploiting processing capabilities available at each node. Data aggregation is one such technique. It consists of aggregating redundant or correlated data in order to reduce the overall size of sent data, thus decreasing the network traffic and energy consumption.

Most literature on sensor networks assumes *total aggregation*, i.e. data packets are assumed to have the same size and aggregation of two or more incoming packets at a node results in a single outgoing packet. Observe that even if this might be considered a simplistic assumption, it allows us to provide an upper bound on the expected benefits of data aggregation in terms of power consumption. We refer here to a selection of papers, focused on the algorithmic side of the problem [2, 10, 13, 15, 14, 16]. However, these papers mainly focus on empirical and technical aspects of the problem.

We concentrate on data aggregation in sensor networks under constraints on the *latency* of sensed events; this means that data should be communicated to the sinks within a specified time after being sensed. Preliminary results are given in [12, 19]. In [19] the authors provide empirical rules to set the aggregation timers at each aggregation point, in order to meet the latency constraints. The basic observation here is that packets should wait longer at nodes closer to the sink to possibly receive information from a large number of nodes. In [12] the authors presented a protocol to dynamically change the data aggregation period according to the aggregation quality. In both cases formal proofs of the performance are not provided.

A sensor network is naturally represented by a graph whose nodes represent the sensors and the arcs the wireless communication links. Data aggregation, latency constraints and energy preservation, give rise to a large variety of graph optimization problems depending on the following issues.
- Transmission energy and time can be seen as functions of the size of the packet and the transmission arc. Typically, these are concave functions exhibiting economies of scale in the size of the packets sent.
- The latency may depend on the (types of) sensor data or on the sensor nodes.
- Sensor networks can be modeled as synchronous or asynchronous systems.
- Data is delivered to one or more sinks.
- The overlay routing paths connecting nodes to the sinks can be fixed a priori, (e.g. a tree or a chain) or may also be chosen as part of the optimization process.
- There might be several objective functions; the most natural ones are to minimize a function of the energy consumption of the nodes or to maximize the amount of sensed data arriving at the sinks with a given energy constraint.
- Sensor networks are usually managed in a distributed on-line way thus reflecting most sensor networks in practice. Studying off-line and/or centralized versions may give valuable knowledge about the structure of problems and their solutions.

By considering the above issues, we formulate the sensor problem in a combinatorial optimization setting, which allows us to derive, what we believe to be, the first results on worst case analysis for on-line algorithms on wireless sensor networks, as opposed to mainly empirical current results. We foresee sensor networks becoming a new class of interesting combinatorial problems.

We concentrate here on a basic subclass of latency constrained data aggregation problems. We assume that transit times and transit costs, in terms of energy consumption, are functions of the arcs only, modeling the situation of total aggregation, while the objective is to minimize the maximum transit cost per node over all nodes. There is only one sink and the transmission paths from the nodes to the sink are unique, forming an intree with the sink as the root. The tree is a typical routing topology in sensor networks; see [3, 10, 14, 17, 18].

The problems we considered are formalized and specified in Section 2. For a thorough understanding of the problem we have studied both the off-line and the on-line version of the problem, although the latter version is the relevant one in practice.

We consider both *centralized* and *distributed* models. The centralized model assumes perfect communication over the network or complete information by the base station about the presence

(not the contents) of information at the nodes. We consider two distributed models based on their use of time synchronization. Time synchronization, in the sense of the existence of a common clock for the nodes, is a crucial component in wireless sensor networks. In typical applications a time stamp forms a crucial part of the sensed data. Time synchronization in wireless sensor networks has been studied in [5, 7, 9, 6]. Time synchronization introduces overhead and in some scenarios, a synchronous model, in which all nodes share the same clock, may not be a requirement. We consider both the *synchronous model* and the *asynchronous model*. We restrict ourselves to the analysis of deterministic algorithms.

In Section 3 we show that the off-line problem is **NP**-hard and we give a 2-approximate algorithm for its solution. We remark here that our approximate solution is based on a new rounding technique of the LP-relaxation of an Integer Linear Programming formulation of the problem, which might be useful for other applications. For the specific case where the graph is a chain we give a dynamic programming formulation which solves the off-line problem in $O(n^3)$ when minimizing the total energy. In this case the problem is equivalent to a batch processing problem. For this problem Finke et al. [8] have developed independently of us exactly the same dynamic programming formulation.

In Section 4 we describe the on-line problem. Our main results are the following:
(a) *Centralized model*. We give a lower bound of 2 on the competitive ratio of any deterministic algorithm. If the routing network is a chain and message latencies are constant, we show a 2-competitive algorithm and a $\sqrt{2}$ lower bound on the competitive ratio.
(b) *Distributed synchronous model*. We present an $\Theta(\log U)$-competitive algorithm, where U is the ratio between the maximum and the minimum time that a packet can wait in its route toward the sink. We also show an $\Omega(\log U)$ lower bound on the competitive ratio of any deterministic distributed algorithm, whence the proposed algorithm is best possible up to a multiplicative constant.
(c) *Distributed asynchronous model*. We give an $O(\delta \log U)$-competitive algorithm in the specific case of unit transit times on the arcs. Here, δ is the depth of the tree. We also show that this algorithm belongs to a broad class of algorithms for which we can prove a lower bound of $\Omega(\delta^{1-\epsilon})$ on the competitive ratio, for any $\epsilon > 0$.

Finally, in Section 5 we suggest possibilities for future research in this rich research area.

In spirit [3] comes closest to our paper. In [3] the authors consider optimization of TCP acknowledgement (ACK) in a multicasting tree. The problem their work addresses is a data aggregation problem. However, energy consumption is not an issue in this problem and latency is considered as a cost instead of a constraint, resulting in an objective of minimizing the sum of the total number of transmissions and the total latency of the messages.

In [4] the authors studied the optimal aggregation policy in a single-hop scenario (i.e. the graph is a star). Namely an *aggregator* performs a request and starts waiting for answers from a set of sources. The time for each source to return its data to the aggregator is independent and identically distributed according to a known distribution F. At some point, the aggregator stops waiting for data and returns an answer depending only on the data received so far and obtains a reward which grows with the number of collected data and decreases with time. They prove that for certain broad families of distributions and broad classes of reward functions, the optimal plan for the aggregator has a simple form. The main differences with our paper are that they assume that the distribution F is known, and they focus on a single-hop scenario.

2 The sensor problem formalized

We study sensor networks $D = (V, A)$, which are *intrees* rooted at a *sink node* $s \in V$. Nodes represent sensors and arcs represent the possibility of transmission between two sensors. Given an arc $a \in A$ we denote its head and tail nodes by head(a) and tail(a), respectively.

Over time, n messages, $N := \{1, \ldots, n\}$, arrive at nodes and have to be sent to the sink. Message j arrives at its *release node* v_j at its *release date* r_j and must arrive at the sink via the unique $v_j - s$-path at or before its *due date* d_j. Thus, each message is completely defined by the

triple (v_j, r_j, d_j). Unless otherwise stated we assume that messages are indexed by increasing due date, i.e., $d_1 \leq d_2 \leq \cdots \leq d_n$. We refer to $L_j := d_j - r_j$ as the *latency* of message j.

A *packet* is a set of messages which are sent simultaneously along an arc. More precisely, each initial message is sent as one packet. Recursively, two packets j and ℓ can be aggregated at a node v. The resulting packet has due date $d = \min\{d_j, d_\ell\}$. This definition naturally extends to the case of more packets aggregated together.

Transition of a message along an arc takes time and energy (cost). In this paper we assume that the transit time $\tau : A \to \mathbb{R}_{\geqslant 0}$ and transit cost $c : A \to \mathbb{R}_{\geqslant 0}$ are independent of packet size. We often refer to the *transit cost* of a node as the transit cost of its unique outgoing arc. This models the situation in which all messages have more or less the same size and where *total aggregation* is possible, as discussed in the introduction. For $v \in V$, let τ_v and c_v be, respectively, the total transit time and total transit cost on the path from v to s. For message j and node u on the path from v_j to s, we define *transit interval* $I_j(u)$ as the time interval during which message j can transit at node u: $I_j(u) := [r_j + \tau_{v_j} - \tau_u, d_j - \tau_u]$. In particular, $I_j(s) = [r'_j, d_j]$, where $r'_j := r_j + \tau_{v_j}$ is the *earliest possible arrival time* of j at s. We abbreviate I_j for $I_j(s)$ and call it the *arrival interval* of message j. We also write $|I|$ for the length of interval I; note that $|I_j(u)| = |I_j|$ for all j and for all u on the path from v_j to u.

Finally, we define $\delta := \max_v \tau_v$ as the depth of the network in terms of the transit time.

The objective of the sensor problem is to send all messages to the sink in such a way as to minimize the maximum transit cost per node (MIN MAX ENERGY) while satisfying the latency restrictions. Given that transit costs are independent of the size of packets sent, but linear in the number of packets sent, it is clearly advantageous to aggregate messages into packets at tail nodes of arcs.

The objective to minimize the total transit cost over all nodes (MIN SUM ENERGY) is closely related to the MIN MAX ENERGY objective and we will show that most of our results also hold for this objective function.

3   The off-line problem

The sensor problem is intrinsically on-line. However, the study of the off-line case is relevant for a thorough understanding of the problem.

In this paragraph we give some positive and negative results on the off-line MIN MAX ENERGY sensor problem. We prove that the problem is strongly **NP**-hard. We formulate the problem as an ILP and we design a novel rounding technique for its LP-relaxation, yielding a 2-approximation. The rounding technique might prove useful in other combinatorial optimization problems, as well. We also prove that these results hold for MIN SUM ENERGY; furthermore we prove for this objective that the problem is polynomially time solvable on a chain.

We start by proving some properties of optimal off-line solutions of both MIN MAX ENERGY and MIN SUM ENERGY.

Lemma 1. *There exists a minimum cost solution such that:*

  (i) *whenever two messages are present together at the same node, they stay together until they reach the sink;*
 (ii) *a message never waits at an intermediate node, i.e., a node different from its release node and the sink;*
(iii) *the time when a packet of messages arrives at the sink is the earliest due date of any message in that packet.*

Proof. (i): Repeatedly apply the argument that whenever two messages are together at the same node but split up afterwards, keeping the one arriving later at the sink with the other message does not increase cost.

(ii): Use (i) and repeatedly apply the following argument. Whenever a packet of messages arrives at an intermediate node and waits there, changing the solution by shifting this waiting time to the tail node of the incoming arc does not increase cost.

(iii): Follows similarly as (ii) by interpreting the time between the arrival of a packet at the sink and earliest due date as waiting time. □

3.1 NP-hardness

Theorem 1. *The off-line* MIN MAX ENERGY *sensor problem is strongly* **NP***-hard.*

Proof. We prove the theorem using a reduction from the Satisfiability Problem. Given an instance of SAT with n boolean variables $X_1, \ldots, X_n$ and m clauses $C_1, \ldots, C_m$, we construct the intree on $n + 2$ nodes depicted in Figure 1.
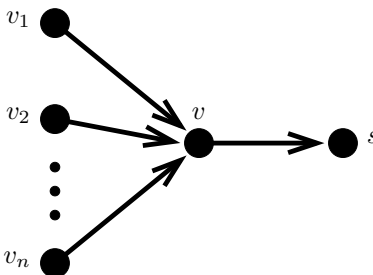

Fig. 1.

The nodes $v_1, \ldots, v_n$ on the left correspond to variables $X_1, \ldots, X_n$. There is one intermediate node v and the sink s on the right. The transit costs of the arcs are determined later. The transit times of all arcs are zero, whence the earliest arrival times of the messages coincide with their release dates.

For clause C_i we define a time interval $T(C_i) = [3i(n+1), 3(i+1)(n+1) - 1]$ and a message $z_i = (v, r_i, d_i) := (v, (3i+1)(n+1), (3i+2)(n+1))$, $i = 1, \ldots, m$. Notice that the arrival interval $I_{z_i} = [r_i, d_i] \subset T(C_i)$. We also define two dummy messages $z_0 := (v, 0, 0)$ and $z_{m+1} := (v, 3(m+1)(n+1), 3(m+1)(n+1))$. Notice the crucial fact $|I_{z_0}| = |I_{z_{m+1}}| = 0$, leaving no choice in sending z_0 and z_{m+1}.

If variable X_j occurs unnegated in clause C_i, we create a message $x_i^j = (v_j, r_i^j, d_i^j) := (v_j, (3i+1)(n+1) + j, (3i+2)(n+1) + j)$. If X_j occurs negated in clause C_i, we create message $x_i^j := (v_j, 3i(n+1) + j, (3i+1)(n+1) + j)$. If X_j does not occur in C_i no message x_i^j is created. Notice that in both cases the arrival time interval $I_i^j \subset T(C_i)$. If X_j does not occur in C_i no message x_i^j is created. An illustration is given in Figure 2.

The rough idea behind the reduction is the following: in an optimal solution, message x_i^j is either sent at its release or at its due date (the reason for this will become clear later). Moreover, sending x_i^j at its release (due) date means setting X_j to true (false). Thus, message z_i can join message x_i^j at node v if and only if the value of variable X_j makes clause C_i true.

We continue with the description of the instance. Let $i_1^j < \cdots < i_{k_j}^j$ denote the indices of the clauses in which variable X_j occurs. We create $k_j + 1$ additional messages released at node v_j. The release and due dates of these messages are chosen such that the $2k_j + 1$ arrival time intervals formed by the release and due dates of all messages released at node v_j form a partition of the interval $[0, 3(m+1)(n+1)]$; see Figure 3.

We will demonstrate that for appropriately chosen cost functions any SAT problem reduces to the MIN MAX ENERGY sensor problem. We define the cost function by $c(v_j, v) = (\max_l k_l + 1)/(k_j + 1)$ for $j = 1, \ldots, n$ and let $c(v, s) = (\max_l k_l + 1)/(\sum_{l=1}^n k_l + 2)$.
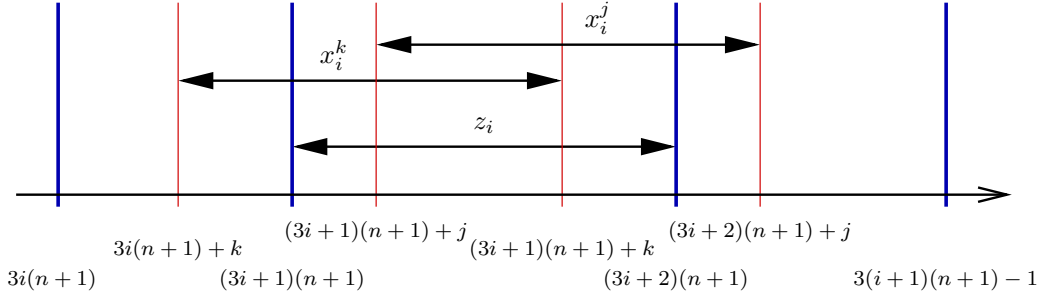
5

Fig. 2. Arrival intervals corresponding to clause C_i. In the depicted example, the clause has the form $C_i = (X_j \vee \neg X_k)$.
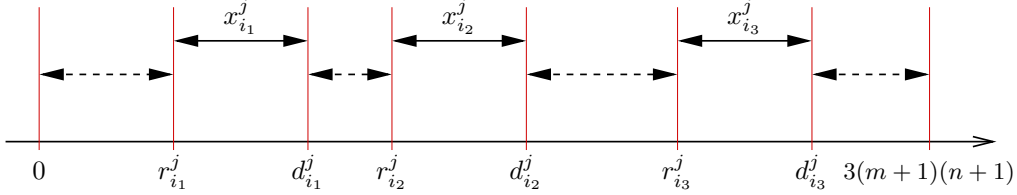

Fig. 3. Arrival intervals of messages with release node v_j. In the depicted example, variable X_j occurs in three clauses C_{i_1}, C_{i_2}, and C_{i_3}. Arrival intervals of the four auxiliary messages are represented by dashed arrows.

Claim 1. Every MIN MAX ENERGY optimal solution to the subinstance obtained by ignoring messages $z_1, \ldots, z_m$ has the following properties:

(a) The cost of each node is $\max_l k_l + 1$;
(b) A message with release node v_j is either sent from v_j at its release date or at its due date, $j = 1, \ldots, n$;
(c) For each fixed $j = 1, \ldots, n$, either all messages x_i^j ($i = i_1^j, \ldots, i_{k_j}^j$) are sent at their release dates or all of them are sent at their due dates.

Proof of Claim 1. Let us consider a solution which minimizes the cost of nodes v_j. Since the $2k_j + 1$ arrival time intervals of messages with release node v_j form a partition of $[0, 3(m+1)(n+1)]$, at most one x_i^j-message and one of the auxiliary messages can be aggregated into a packet, which then has to be sent at the single intersection point of the two arrival time intervals. Thus the minimal number of packets that have to be sent from node v_j is $k_j + 1$, i.e. k_j pair-packets and one packet containing a single message. Hence the minimal cost of node v_j is $c(v_j, v)(k_j + 1) = \max_l k_l + 1$ for all nodes v_j.

Each pair-packet is sent at the common release and due date of its two messages and by construction of these dates no two pair-packets emanating from different nodes can be aggregated into a single packet at node v. Thus, there are $\sum_{l=1}^n k_l$ pair-packets passing v. And we have the two dummy messages z_0 and z_{m+1}, which are sent from node v at times 0 and $3(m+1)(n+1)$. Pair-packets cannot be aggregated with these dummy messages but each single-message-packet can be sent at time 0 or $3(m+1)(n+1)$ and hence it may join dummy message z_0 or z_{m+1} at node v. This gives a total of $\sum_{l=1}^n k_l + 2$ packets passing v. Thus, the cost of node v is $c(v, s)(\sum_{l=1}^n k_l + 2) = \max_l k_l + 1$. Notice that a single-message-packet contains either the first or the last auxiliary message released at node v_j. If the single-message-packet is the first auxiliary message then all pair-packets are sent on the due date of the x_i^j-message in the packet. Otherwise, all pair-packets are sent on the release date of the x_i^j-message in the packet.

Thus, we have constructed a solution which satisfies properties (a),(b) and (c). As the cost of node v_j is at least $\max_l k_l + 1$ the solution is an optimal solution. From the construction of this solution it can easily be verified that any solution which violates property (a),(b) or (c) has a node with a cost which exceeds $\max_l k_l + 1$. $\square$

6

This claim suffices to prove the following claim which in its turn implies the proof of the theorem directly.

Claim 2. The MIN MAX ENERGY sensor problem has a solution with maximum cost at most $\max_l k_l + 1$ if and only if the underlying instance of SAT is satisfiable.

Proof of Claim 2. Given a satisfying assignment for the SAT instance, a feasible solution to the sensor problem can be obtained as follows. Notice that in the construction of an optimal solution in the proof of Claim 1, for each j, there is a choice for the set of messages corresponding to X_j, to send either dummy message z_0 separately at time 0 or the dummy message z_{m+1} separately at time $3(m+1)(n+1)$. In both cases the cost of sending all messages corresponding to the variables and z_0 and z_{m+1} is $\max_l k_l + 1$ for each node. We make the choice now by sending z_{m+1} separately if X_j is true in the satisfying assignment and z_0 separately if X_j is false.

We claim that message z_i corresponding to clause C_i, $i = 1, \ldots, m$, can be aggregated at v with one of the pair-packets corresponding to a variable in the clause. Suppose that clause C_i is satisfied due to variable X_j. If X_j appears unnegated in C_i (thus X_j is true), then the pair-packet containing message x_i^j is sent at time $r_i^j := (3i+1)(n+1) + j \in [r_i, d_i]$, and message z_i can join this packet at no additional cost. Similarly, if X_j appears negated at C_i (thus X_j is false) then x_i^j is sent at $d_i^j := (3i+1)(n+1) + j \in [r_i, d_i]$. This concludes the proof of the "if" part.

It follows from Claim 1 that any feasible solution with maximum cost $\max_l k_l + 1$ yields an assignment of values to the boolean variables $X_1, \ldots, X_n$: variable X_j is set to true (false), if all messages x_i^j, $i = i_1^j, \ldots, i_{k_j}^j$, are sent at their release (due) dates. It also follows from Claim 1 that in an optimal solution message z_i, $i = 1, \ldots, m$ should not cause additional cost, therefore it must join one of the packets starting at a node v_j. Due to the construction of the instance, this is only possible if the value of variable X_j causes clause C_i to be satisfied. This concludes the proof of the "only if" part for this objective function. $\qquad\square$

NP-hardness of MIN SUM ENERGY is proved in a similar way. We write out the proof for completeness.

Theorem 2. *The off-line* MIN SUM ENERGY *sensor problem is strongly* **NP**-*hard*.

Proof. We prove the theorem using the same reduction from the Satisfiability Problem as in Theorem 1, but using a different cost function on the arcs. We define $M := 2nm$ and $K := 2 + nM + \sum_{j=1}^n k_j(M+1)$. Let the cost function be defined by $c(v_j, v) = M$ for $j = 1, \ldots, n$ and let $c(v, s) = 1$.

Claim 3. Every MIN SUM ENERGY optimal solution to the subinstance obtained by ignoring messages $z_1, \ldots, z_m$ has the following properties:

(a) The cost of the solution is K;
(b) A message with release node v_j is either sent from v_j at its release date or at its due date, $j = 1, \ldots, n$;
(c) For each fixed $j = 1, \ldots, n$, either all messages x_i^j ($i = i_1^j, \ldots, i_{k_j}^j$) are sent at their release dates or all of them are sent at their due dates.

Proof of Claim 3. Let us consider a solution which minimizes the total cost of all nodes. Since M is sufficiently large, an optimal solution will send as few packets of messages as possible from each node v_j. Since the $2k_j + 1$ arrival time intervals of messages with release node v_j form a partition of $[0, 3(m+1)(n+1)]$, at most one x_i^j-message and one of the auxiliary messages can be aggregated into a packet, which then has to be sent at the single intersection point of the two arrival time intervals. Thus the minimal number of packets that have to be sent from node v_j is $k_j + 1$, i.e. k_j pair-packets and one packet containing a single message.

Each pair-packet is sent at the common release and due date of its two messages and by construction of these dates no two pair-packets emanating from different nodes can be aggregated

7

into a single packet at node v. Each pair-packet costs $M + 1$ and in total, the transit costs for these pair-packets add up to $\sum_{j=1}^{n} k_j(M + 1)$.

And we have the two dummy messages z_0 and z_{m+1}, which are sent from node v at times 0 and $3(m + 1)(n + 1)$. Pair-packets cannot be aggregated with these dummy messages but each single-message-packet can be sent at time 0 or $3(m + 1)(n + 1)$ and hence it may join dummy message z_0 or z_{m+1} at node v. Only if each single-message-packet is sent with a dummy node, the total transit cost remains below K. Notice that there are n single-message-packets and each single-message-packet contains either the first or the last auxiliary message released at node v_j. If the single-message-packet is the first auxiliary message then all pair-packets are sent on the due date of the x_i^j-message in the packet. Otherwise, all pair-packets are sent on the release date of the x_i^j-message in the packet. □

This claim suffices to prove the following claim which in its turn implies the proof of the theorem directly.

Claim 4. The MIN SUM ENERGY sensor problem has a solution of total cost at most K if and only if the underlying instance of SAT is satisfiable.

Proof of Claim 4. Given a satisfying assignment for the SAT instance, a feasible solution to the sensor problem can be obtained as follows. Notice that, in the construction of an optimal solution in the proof of Claim 3, for each j, there is a choice for the set of messages corresponding to X_j, to send either dummy message z_0 separately at time 0 or the dummy message z_{m+1} separately at time $3(m + 1)(n + 1)$. In both cases the total cost for sending all of the messages corresponding to the variables plus the two dummy messages z_0 and z_{m+1} is K. We make the choice now by sending z_{m+1} separately if X_j is true in the satisfying assignment and z_0 separately if X_j is false.

We claim that message z_i corresponding to clause C_i, $i = 1, \dots, m$, can be aggregated at v with one of the pair-packets corresponding to a variable in the clause. Suppose that clause C_i is satisfied due to variable X_j. If X_j appears unnegated in C_i (thus X_j is true), then the pair-packet containing message x_i^j is sent at time $r_i^j := (3i + 1)(n + 1) + j \in [r_i, d_i]$, and message z_i can join this packet at no additional cost. Similarly, if X_j appears negated at C_i (thus X_j is false) then x_i^j is sent at $d_i^j := (3i + 1)(n + 1) + j \in [r_i, d_i]$. This concludes the proof of the "if" part.

It follows from Claim 3 that any feasible solution of cost K yields an assignment of values to the boolean variables $X_1, \dots, X_n$: variable X_j is set to true (false), if all messages x_i^j, $i = i_1^j, \dots, i_{k_j}^j$, are sent at their release (due) dates. It also follows from Claim 3 that in an optimal solution message z_i, $i = 1, \dots, m$ should not cause additional cost, therefore must join one of the packets starting at a node v_j. By construction of the instance, this is only possible if the value of variable X_j causes clause C_i to be satisfied. This concludes the proof of the "only if" part for this objective function. □

3.2 A 2-approximation

We give an ILP-formulation of the problem, based on Lemma 1, and show that rounding the optimal solution of the LP-relaxation yields a 2-approximation algorithm. For every message-arc pair $\{i, a\}$, we introduce a binary decision variable x_{ia}, which is set to 1 if and only if arc a is used by some message j which arrives at s at time d_i. We use the notation $j_{\min}$ for the smallest index i such that $d_i \geq r_j'$; that is, $j_{\min} := \min\{i : d_i \geq r_j'\}$. We use a_j to denote the first arc on the (unique) $v_j - s$- path. The LP relaxation of the MIN MAX ENERGY sensor problem is

$$
\begin{aligned}
\min\ & z \\
\text{s.t. } & z \geq c(a) \sum_{i=1}^{n} x_{ia} \quad \forall a \in A, \\
& \sum_{i=j_{\min}}^{j} x_{ia_j} \geqslant 1 \quad \forall\ 1 \leq j \leq n, \\
& x_{ia} \geqslant x_{ia'} \quad \forall\ 1 \leq i \leq n\ \forall\ a, a' \in A \text{ with } \mathrm{head}(a') = \mathrm{tail}(a), \\
& x_{ia} \in \{0, 1\} \quad \forall\ 1 \leq i \leq n\ \forall\ a \in A.
\end{aligned}
\tag{1}
$$

The set of constraints $\sum_{i=j_{\min}}^{j} x_{ia_j} \geqslant 1$ forces each message to leave its release node in time to reach the sink before its due date. The set of constraints $x_{ia} \geqslant x_{ia'}$ force that a message does not wait at intermediate nodes.

In the following lemma we develop a tool for rounding the corresponding LP-relaxations, which are obtained by replacing the integrality constraints with non-negativity constraints $x_{ia} \geqslant 0$.

Lemma 2. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{R}_{\geqslant 0}$ and $\beta_1, \ldots, \beta_n \in \{0, 1\}$ with*

$$\sum_{i=j}^{k} \alpha_i \geqslant 1 \quad \implies \quad \sum_{i=j}^{k} \beta_i \geqslant 1 \quad \forall 1 \leq k \leq n \; \forall 1 \leq j \leq k. \tag{2}$$

By decreasing some of the β_i's from 1 to 0, one can enforce the inequality

$$\sum_{i=1}^{n} \beta_i \leqslant 2 \sum_{i=1}^{n} \alpha_i \tag{3}$$

while maintaining property (2). Moreover, this can be done in linear time. $\square$

Proof. Consider the β_i's in order of increasing index. If $\beta_i = 1$, then round it down to 0, unless this yields a violation of (2). It is not difficult to see that this greedy algorithm can be implemented to run in linear time. It remains to be proven that inequality (3) holds for the resulting numbers $\beta_1, \ldots, \beta_n$.

For $h \in \{1, \ldots, n\}$, let $\bar{h} := \min\{i > h \mid \beta_i = 1\}$; if $\beta_i = 0$ for all $i > h$ or $h = n$, then $\bar{h} := n + 1$. Similarly, let $\underline{h} := \max\{i < h \mid \beta_i = 1\}$; if $\beta_i = 0$ for all $i < h$ or $h = 1$, then $\underline{h} := 0$. We prove the following generalization of (3):

$$\sum_{i=1}^{h} \beta_i \leqslant 2 \sum_{i=1}^{\bar{h}-1} \alpha_i \quad \forall \, 1 \leq h \leq n. \tag{4}$$

By contradiction, consider the smallest index h violating (4). Since h is chosen minimally, it must hold that $\beta_h = 1$; rounding β_h down to 0 would yield a violation of (2). In particular this would yield

$$\sum_{i=\underline{h}+1}^{\bar{h}-1} \alpha_i \geqslant 1 \tag{5}$$

while $\sum_{i=\underline{h}+1}^{\bar{h}-1} \beta_i = 0$. Notice that $\underline{h} \geqslant 1$, since, by choice of h,

$$\sum_{i=1}^{h} \beta_i \; > \; 2 \sum_{i=1}^{\bar{h}-1} \alpha_i \overset{(5)}{\geqslant} 2 \; .$$

Thus, $\beta_{\underline{h}} = \beta_h = 1$. We get a contradiction to the choice of h:

$$\sum_{i=1}^{h} \beta_i \; = \sum_{i=1}^{\underline{h}-1} \beta_i + 2 \overset{(4)}{\leqslant} 2 \sum_{i=1}^{\underline{h}-1} \alpha_i + 2 \overset{(5)}{\leqslant} 2 \sum_{i=1}^{\underline{h}-1} \alpha_i + 2 \sum_{i=\underline{h}+1}^{\bar{h}-1} \alpha_i \leqslant 2 \sum_{i=1}^{\bar{h}-1} \alpha_i \; .$$

The first inequality follows from (4) since $\overline{(\underline{h} - 1)} = \underline{h}$. This concludes the proof. $\square$

Theorem 3. *There is a polynomial time 2-approximation algorithm for the* MIN MAX ENERGY *sensor problem on intree $D = (V, A)$.*

Proof. We round optimal (fractional) solution (x, z) of the LP relaxation of (1) to an integral solution $(\bar{x}, \bar{z})$. Consider the arcs in order of non-decreasing distance from s. For arc a with $\text{head}(a) = s$, set $\hat{x}_{ia} = 1 \, \forall \, i = 1, \ldots, n$. Modify these values to $\bar{x}_{1a}, \ldots, \bar{x}_{na}$ by applying Lemma 2 to $x_{1a}, \ldots, x_{na}$ and $\hat{x}_{1a}, \ldots, \hat{x}_{na}$. For an arc a' with larger distance to s, take the arc a with $\text{head}(a') = \text{tail}(a)$ and set $\hat{x}_{ia'} := \bar{x}_{ia} \, \forall i = 1, \ldots, n$. We also modify these values into $\bar{x}_{1a'}, \ldots, \bar{x}_{na'}$ by applying Lemma 2 to the values $x_{1a'}, \ldots, x_{na'}$ and $\hat{x}_{1a'}, \ldots, \hat{x}_{na'}$. Premise (2) of Lemma 2 is satisfied for $x_{1a'}, \ldots, x_{na'}$ and $\hat{x}_{1a'}, \ldots, \hat{x}_{na'}$ since (2) holds for $x_{1a}, \ldots, x_{na}$ and $\bar{x}_{1a}, \ldots, \bar{x}_{na}$ and since $x_{ia'} \leq x_{ia}$.

By construction, the final solution $(\bar{x}, \bar{z})$ is feasible if we choose $\bar{z} = 2z$. $\square$

For the MIN SUM ENERGY sensor problem we may obtain a 2-approximation in a similar way. The LP relaxation of the MIN SUM ENERGY sensor problem is

$$\begin{aligned}
\min \ & \sum_{a \in A} c(a) \sum_{i=1}^{n} x_{ia} \\
\text{s.t.} \ & \sum_{i=j_{\min}}^{j} x_{ia_j} \geqslant 1 \quad \forall \ 1 \leq j \leq n, \\
& x_{ia} \geqslant x_{ia'} \quad \forall \ 1 \leq i \leq n \ \forall \ a, a' \in A \text{ with head}(a') = \text{tail}(a), \\
& x_{ia} \in \{0,1\} \quad \forall \ 1 \leq i \leq n \ \forall \ a \in A.
\end{aligned}$$
(6)

Note that this LP-relaxation differs from the MIN MAX ENERGY LP-relaxation only in the objective function and variable z. In particular, a feasible solution x to the MIN SUM ENERGY problem is a feasible solution to the MIN MAX ENERGY problem for appropriately chosen z. Similarly a feasible solution (x, z) to the MIN MAX ENERGY problem yields a feasible solution x to the MIN SUM ENERGY problem.

The constraints on z ensure that z is at least the transit cost of any node. The 2-approximation follows from the following Corollary, which is based on the proof of Theorem 3.

Corollary 1. *There is a polynomial time* 2*-approximation algorithm for the* MIN SUM ENERGY *sensor problem on intree* $D = (V, A)$.

Proof. We round optimal (fractional) solution x of the LP relaxation of (6) to an integral solution $\bar{x}$. As the constraint sets of both MIN MAX ENERGY and MIN SUM ENERGY are equal except for constraints on z, we obtain an integral vector $\bar{x}$ as described in Theorem 3. This proves the corollary. □

3.3   A polynomial time algorithm for the chain

For the MIN SUM ENERGY objective, the off-line sensor problem is polynomially solvable if graph D is a chain. We give a dynamic programming formulation and analyze its running time.

For $0 \leqslant i \leqslant k \leqslant n+1$, we denote by $N(i,k)$ the set of messages in $\{i+1, \ldots, k-1\}$ whose earliest arrival time is strictly later than i's due date. More formally,

$$N(i,k) := \{j \in N \mid i < j_{\min} \leqslant j < k\} \ .$$

$\mathcal{OPT}(i,k)$ denotes the cost of an optimal solution to the partial instance defined by the subset of messages in $N(i,k)$. The dynamic programming formulation is based on the following Lemma:

Lemma 3. *Let ℓ be a message in $N(i,k)$ whose release node v_ℓ has maximum total transit time t_{v_ℓ} among all messages in $N(i,k)$. Then,*

$$\mathcal{OPT}(i,k) \ = \ c_{v_\ell} + \min_{\ell_{\min} \leq j \leq \ell} \big(\mathcal{OPT}(i,j) + \mathcal{OPT}(j,k)\big) \ .$$
(7)

Proof. In an optimal solution for $N(i,k)$, message ℓ arrives at the sink at time d_j, for some $j \in \{\ell_{\min}, \ldots, \ell\}$. Moreover, there exists an optimal solution for $N(i,k)$ in which all messages in $N_{i,k}(j) := \{h \in N(i,k) \mid h_{\min} \leqslant j \leqslant h\}$ arrive at the sink together with message ℓ at time d_j (because they can join ℓ on its way to the sink at no additional cost). Notice that $N(i,k)$ is the union of the disjoint sets $N(i,j)$, $N_{i,k}(j)$, and $N(j,k)$. Moreover, since no two messages in $N(i,j)$ and $N(j,k)$ can ever reach the sink together in a feasible solution, the remaining problem can be decomposed into two subproblems for messages in $N(i,j)$ and $N(j,k)$. This yields the desired result. □

Theorem 4. *The off-line* MIN SUM ENERGY *sensor problem on a chain can be solved in $O(n^3)$ time.*

Proof. Consider a dynamic program which computes values $\mathcal{OPT}(i,k)$, $0 \leq i \leq k \leq n+1$, in order of non-decreasing $k - i$. It follows from Lemma 3, that the algorithm can compute these $O(n^2)$ values and since it takes $O(n)$ time to evaluate (7) the algorithm needs a total time of $O(n^3)$. The value of an optimal solution can be derived from $\mathcal{OPT}(0, n+1)$. Moreover, keeping track of the messages j that minimize the right hand sides of (7), we may derive for each message in N the time at which it is sent to s. This concludes the proof. □

10

4 The on-line problem

In real-world sensor networks, it is typically hard to predict whether and when new messages will be released. This is captured by the on-line model where messages are released over time and an algorithm has no information on messages before they are released.

We distinguish three different on-line models:

- the centralized model;
- the synchronous distributed model;
- the asynchronous distributed model.

In the *centralized* on-line model there is complete information on the presence of messages at nodes at a certain time instant. A centralized algorithm may use this information and determines at what times each node sends a packet of messages towards the sink.

In the *distributed* on-line models nodes communicate independently of each other. Each node is equipped with an algorithm, which determines at what times the node sends a packet of messages to the next node on the path to the sink. The input of each node's algorithm at any time t is restricted to the packets that have been released at or forwarded from that node in the period $[0, t]$.

We assume that all nodes are equipped with a clock to measure the latency of messages. We distinguish two distributed on-line models: in the *synchronous* model all nodes are equipped with a *common clock*, i.e. the times indicated at all clocks are identical. A common clock may facilitate synchronization of actions in various nodes. In the *asynchronous* model there is no such common clock; still, the duration of the time unit is assumed to be the same for all nodes.

We also assume in both models that each node v knows its total transit time τ_v to the sink. Moreover, for the asynchronous model we assume that all transit times $\tau(a)$ are equal, and without loss of generality we set $\tau(a) = 1 \ \forall a \in A$.

4.1 The centralized model

For the on-line centralized model we prove that no on-line algorithm can be better than 2-competitive. We also present an algorithm which is intuitively appealing and we prove that this algorithm is 2-competitive on a chain if latencies of all messages are equal. We also show that in case of a chain and equal latencies no algorithm can be better than $\sqrt{2}$-competitive. These results hold for both objectives. It is an open problem if there exists a constant competitive algorithm for the on-line centralized problem on a tree, for either objective.

Theorem 5. *No algorithm for the on-line centralized problem on a chain can have a competitive ratio less than $2 - \frac{1}{m}$ for* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. For the MIN MAX ENERGY objective consider a chain consisting of nodes u_2, u_1, s with unit cost. At some time t an adversary releases a message at u_2. If an algorithm sends the message at time t then at $t+1$ another message is released at u_2. This gives the algorithm a maximum cost per node of at least 2 against optimal maximum cost per node of 1. Else, if an algorithm delays the message then the adversary releases a message at u_1 with due date $t+2$, which gives the same cost structure as above.

For the MIN SUM ENERGY objective consider a chain with $m + 1$ nodes, consisting of nodes $u_m, \ldots, u_1, s$ with unit cost. Assume $L_j := L \geq 2m$ for all messages j released at u_m. Think of an adversary who releases messages at each time unit in the node u_m at distance m from s, until the algorithm decides to delay this message. Suppose the algorithm decides to hold up the i-th message.

If $i \leq L - m$ the adversary also releases a message at u_{m-1} with latency $m - 1$ at time instance i. The algorithm has sent $i - 1$ messages from u_m to u_{m-1} at a cost of $i - 1$ and has to send at least one packet from u_{m-1} and the packet at u_m separately to s. Total costs are at least $2m - 1 + (i - 1)$ which is minimal if $i = 1$. The optimal costs are m.

If $i > L - m$ then the adversary stops releasing messages after the $L - m$-th one, keeping optimal costs m against total cost $(L - m - 1) + m$ for the algorithm, which proves the theorem since $L \geq 2m$. □

Algorithm: Earliest Due Date First(EDD) At each time t do the following:
- select any message with earliest due date, t_{dd}, from the set of messages that have not arrived at the sink, as the *responsible message*;
- each sensor v which contains messages sends a packet containing all messages present at this node if $t + \tau_v = t_{dd}$.

Although time is continuous the algorithm only has to make decisions, i.e. select the responsible message and assign sending times to nodes, if a new message is released. Note, that in case of constant latencies once a message is chosen as responsible message we may assume without loss of generality that it remains the responsible message in the network until it reaches the sink. In case of arbitrary latencies, a newly released message may become the responsible message if its due date precedes the due date of the previous responsible message. It can easily be observed that any EDD-solution satisfies properties (i-iii) the optimal solution is proved to have in Lemma 1.

To analyze the competitive ratio of EDD we construct a graph by identifying its vertices with messages and its edges with pairs of messages that can be combined in a packet, i.e. whose arrival time intervals intersect. The resulting graph is an interval graph. A solution $\mathcal{P}$ consists of a set of packets. If packet $P \in \mathcal{P}$ arrives at the sink at time t, it contains only messages which have time t in their arrival time interval, i.e. $P \subseteq \{j | t \in I_j\}$. Each packet P corresponds to a clique C in the interval graph. We denote as C_t a clique whose messages arrive at time t at the sink, and we denote the corresponding solution set (of cliques) as $\mathcal{C}$.

Thus the sensor problem can be viewed as a clique partitioning problem on an interval graph. Clearly an optimal solution will consist of a collection of maximal cliques, but on-line solutions may not achieve this. The messages in a clique arrive at the sink in a single packet. It is the structure of the objective function that creates the algorithmic challenges. Indeed, the clique partitioning problem related to the sensor problem on an intree is **NP**-hard, as discussed in Section 3.

Theorem 6. EDD *is* 2-*competitive for the centralized sensor problem on a chain with constant latencies for both* MIN SUM ENERGY *and* MIN MAX ENERGY.

Proof. We prove that the number of packets sent from a node v in the EDD-solution is at most 2 times the number of packets sent from a node v in the optimal solution with respect to either objective.

Let $\mathcal{C}^*$ be an optimal solution of the off-line problem which satisfies property (i) in Lemma 1. We claim that any clique of optimal off-line solution $\mathcal{C}^*$ is intersected by at most two EDD-cliques. Suppose not, then $\mathcal{C}^*$ must contain a clique C_t^* which is intersected by at least 3 EDD-cliques. Consider any 3 such EDD-cliques, $C_{t_1}, C_{t_2}, C_{t_3}$ ordered by increasing arrival times at the sink. Let, for $i := 1, \ldots, 3$, $j_i \in C_t^* \cap C_{t_i}$ be a message with earliest due date amongst all messages in $C_t^* \cap C_{t_i}$ and let I_{j_i} be the corresponding arrival time interval. As $j_1, j_2, j_3 \in C_t^*$ we must have $I_1 \cap I_2 \cap I_3 \neq \emptyset$. Also, as $j_3 \notin C_{t_2}$, t_2 has to be smaller than the earliest possible arrival time of message j_3 as otherwise EDD would have added j_3 to clique C_{t_2}; hence $t_2 < t$. As EDD sends each packet of messages such that it arrives at the due date of some message in the packet, C_{t_2} must contain a message $j \notin C_t^*$ with due date t_2 by definition of j_2. Let I be the arrival time interval of this message. We must have $t_2 \in I$ and $t_1 \notin I$ as otherwise message j would be contained in C_{t_1}. This is only possible if the length of I is strictly less than the length of I_1. As we assume constant latency, this is only possible when message j is released at some node further away from the sink than message j_1.

Now, consider the optimal solution. As message j has due date $t_2 < t$ it must arrive at the sink before message j_1; but as message j would pass message j_1 it follows from property (i) in Lemma 1 that message j_1 is aggregated with j. As message j_3 cannot be aggregated with these messages this contradicts the assumption that j_1 and j_3 are aggregated into a single clique C_t^* in the optimal solution. This proves the claim.

12

By construction of the EDD-solution, all messages in an EDD-clique that have to be sent from a node v, are sent from this node simultaneously. Let C_t^* be an optimal off-line clique and let C_{t_1} and C_{t_2} be the two EDD-cliques intersecting C_t^*. If messages in C_t^* have to be sent from a node v, then in the EDD-solution node v has to send at most twice to send all messages in C_{t_1} and C_{t_2}. This proves the theorem. $\qquad\square$

The competitive ratio of EDD can be arbitrarily close to 2 for both objectives. even with constant latencies. Consider the chain u_2, u_1, s with $\tau_a = 1$ and costs $c(u_2, u_1) = l, c(u_1, s) = 1$, for some $l, l \geq 1$. Assume a latency of $L \geq 4$ for each message. Message 1 is released at node u_1 at time $r_1 = 0$. Message 2 is released at node u_2 at time $r_2 = L - 2$. Message 3 is released at node u_2 at time $r_2 = L - 1$. EDD aggregates messages 1 and 2 in a packet and sends message 3 separately, yielding a total cost of $2(l+1)$ and a maximum cost of $2l$. The optimal solution (for both objectives) aggregates messages 2 and 3 and has a total cost of $l + 2$ and a maximum cost of l. Hence, EDD is at best 2-competitive for MIN MAX ENERGY and it is $2 - \frac{2}{l+2}$-competitive for MIN SUM ENERGY.

As this example shows, EDD cannot be better than 2-competitive because it aggregates all messages that can be aggregated with the responsible message in one packet. It does not consider the extra cost of aggregating these messages with the responsible message. Also, note that EDD assigns messages to a clique irrevocably. Maybe these two insights help to construct an on-line algorithm with better competitive ratio, both on a chain and a tree. In the following theorem we generalize the above counter example to give a lower bound on the competitive ratio of any on-line algorithm. The theorem leaves open the possible existence of an on-line algorithm with better competitive ratio than EDD.

Theorem 7. *No algorithm can be better than $\sqrt{2}$-competitive for the on-line sensor problem on a chain with constant latencies, for* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. Consider the chain u_2, u_1, s with $\tau_a = 1$ and costs $c(u_2, u_1) = l, c(u_1, s) = 1$, for some $l, 1 \leq l \leq 2$. Assume a latency of $L \geq 2m$ for each message. Message 1 is released at node u_1 at time $r_1 = 0$. Message 2 is released at node u_2 at time $r_2 = L - 2$. If node u_2 does not send message 2 immediately to be aggregated with message 1, then the algorithm incurs a total cost of $l + 2$ whereas the optimal total cost is $l + 1$. Also, the maximum cost of a node in this case is $\max\{2, l\} = 2$ whereas the maximum cost of a node in the optimal solution is $\max\{1, l\} = l$. If messages 1 and 2 are aggregated, then message 3 is released at node u_2 at time L. Hence, a total cost of $2(l+1)$ is incurred, whereas the optimal total cost is $l + 2$. Also, the maximum cost of a node in this case is $2 \max\{l, 1\} = 2l$ whereas the maximum cost of a node in the optimal solution is $\max\{2, l\} = 2$. The theorem follows if we let $l := \sqrt{2}$. $\qquad\square$

In case the graph is a tree, EDD may perform even worse. We give an example which demonstrates that EDD is at least $\Omega(n)$-competitive in this case. Consider the tree of Figure 1. Let $\tau(v, s) = 0, \tau(v_i, v) = L, i = 1, \ldots, n-1, \tau(v_n, v) = 1$. The adversary releases $2(n-1)$ messages with some constant latency $L \geq 2n$. He releases message j_i at node v_n at time $r_i = L + i$ for $i = 1, \ldots, n-1$. Further, he releases messages j_i' at node v_i at time $r_i = i + 1$ for $i = 1, \ldots, n-1$. Thus $I_{j_i} = [L + i + 1, 2L + i]$ and $I_{j_i'} = [L + i + 1, L + i + 1]$. EDD aggregates each message j_i with message j_i' into a single packet, regardless of the arc costs. As a result, no two packets can be aggregated at v. This gives a cost of $nc(v_n, v)$ for leaf v_n, a cost of $c(v_i, v)$ for leaf $v_i, i = 1, \ldots, n-1$, and a cost of $nc(v, s)$ for node v. If we choose as cost function $c(v_n, v) = C \geq (n+1)$ and $c(a) = 1$ for all $a \in A \backslash (v_n, v)$, then the optimal solution, for both objectives, aggregates all messages at v_n into a single packet, thus (v_n, v) has to be traversed only once. This gives a cost of $c(v_n, v)$ for leaf v_n, a cost of $c(v_i, v)$ for leaf $v_i, i = 1, \ldots, n-1$, and a cost of $(n+1)c(v, s)$ for node v. This proves that EDD is $\Omega(n)$-competitive for both objectives.

4.2 The synchronous model

For the synchronous distributed model we derive upper bound on the competitive ratio of an algorithm for both the MIN MAX ENERGY and the MIN SUM ENERGY objective. We prove that

this algorithm is best possible (up to a multiplicative constant) among all deterministic algorithms for the MIN MAX ENERGY problem.

The algorithm is based on the following simple lemma.

Lemma 4. *Given any interval* $[a,b]$, $a,b \in \mathbb{N}$. *Let* $i^* = \max\{i \in \mathbb{N} \mid \exists k \in \mathbb{N} : k2^i \in [a,b]\}$, *then* k^* *for which* $k^*2^{i^*} \in [a,b]$ *is odd and unique. Also,* $i^* \geq \lfloor \log_2(b-a) \rfloor$. *We use notation* $t(a,b) = k^*2^{i^*}$. $\square$

Proof. Assume that $k_1 2^{i^*} \in [a,b]$ and $k_2 2^{i^*} \in [a,b]$, with $k_1 < k_2$. We may assume that $k_2 = k_1 + 1$, for if $k_2 > k_1 + 1$, then obviously also $(k_1 + 1)2^{i^*} \in [a,b]$. This means that either k_1 or k_2 is even. Suppose this is k_1 (if k_2 is even the arguments are analogous). Then, $k_1/2 \in \mathbb{N}$ and $(k_1/2)2^{i^*+1} \in [a,b]$, contradicting the definition of i^*. Observe that $i^* \geq \lfloor \log_2(b-a) \rfloor$ follows directly from the definition of i^*. $\square$

> **Algorithm:CommonClock** (CC) Message j is sent from v_j at time $t(r'_j, d_j) - \tau_{v_j}$ to arrive at s at time $t(r'_j, d_j)$ unless some other message (packet) passes v_j in the interval $[r_j, t(r'_j, d_j) - \tau_{v_j}]$, in which case j is aggregated and the packet is forwarded directly.

To give a competitive analysis of this algorithm we first derive a bound on the competitive ratio of CC for instances in which the arrival intervals I_j differ by at most a factor 2 in length.

Lemma 5. *If there exists an* $i \in \mathbb{N}$ *such that* $2^{i-1} < |I_j| \leq 2^i$ *for all messages* j, *then* CC *has a competitive ratio of at most* 3 *for both* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. We will prove that the transit cost of each arc in the CC-solution is at most 3 times the transit cost of this arc in the optimal solution.

Assume that in an optimal solution packets arrive at s at times $t_1 < \cdots < t_\ell$. Let N_h^* be the packet arriving at t_h at s. Since $t_h \in I_j \,\forall j \in N_h^*$ and $|I_j| \leq 2^i \,\forall j$, we have $I_j \subset [t_h - 2^i, t_h + 2^i] =: I$ $\forall j \in N_h^*$, and $|I| = 2 \cdot 2^i$. If $t_h = k2^i$ then in the CC-solution all messages in N_h^* may arrive at s at times t_h, $t_h - 2^i$ or $t_h + 2^i$. If $t_h \neq k2^i$ then I contains two different multiples of 2^i, say $k2^i$ and $(k+1)2^i$, such that $k2^i < t_h < (k+1)2^i$. In this case, since $|I_j| > 2^{i-1} \,\forall j$, we have $\forall j \in N_h^*$ that $I_j \cap \{k2^i, k2^i + 2^{i-1}, (k+1)2^i\} \neq \emptyset$. Lemma 4 implies that in a CC-solution every message $j \in N_h^*$ arrives at s at one of $\{k2^i, k2^i + 2^{i-1}, (k+1)2^i\}$. Hence, $\forall h = 1, \ldots, \ell$, all messages in N_h^* arrive at s at at most 3 distinct time instants in the CC-solution. CC does not delay messages at intermediate nodes. This implies that the arcs used by messages in N_h^* are traversed by these messages at most 3 times in the CC-solution, proving the lemma. $\square$

Let $U = \frac{\max_j |I_j|}{\max\{1, \min_j |I_j|\}}$ and let $U^* := \max\{\log U, 1\}$.

Theorem 8. CC *is* $O(U^*)$*-competitive for* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. For each $i \in \mathbb{N}$ with $\log(\max\{1, \min_j |I_j|\}) \leq i \leq \lceil \log(\max_j |I_j|) \rceil$, CC sends the messages in $N_i := \{j \in N \mid 2^{i-1} < |I_j| \leq 2^i\}$, at a cost of no more than 3 times the optimum, by Lemma 5. This proves $O(\max\{\log U, 1\})$-competitiveness if $\min_j |I_j| \geq 1$. In case $\min_j |I_j| = 0$ we observe that restricted to the class of messages $N_0 = \{j \in N \mid |I_j| = 0\}$ CC's cost equals the optimal cost, because there is no choice for these messages. $\square$

In fact CC can be $\Omega(\log U)$-competitive on instances. Consider a chain of 2^{n+1} nodes $u_1, \ldots, u_{2^{n+1}} = s$ for some $n \in \mathbb{N}$. Take $\tau(a) = 1$ and $c(a) = 1 \,\forall a$. For $j = 1, \ldots, n$, $v_j = u_{2^j}$, $r_j = 0$, and $d_j = 2^{n+1} - 1$. Hence $r'_j = 2^{n+1} - 2^j = k2^j$ for some odd $k \in \mathbb{N}$ and $|I_j| = 2^j - 1$. Therefore, CC makes each message j arrive at s at time r'_j, no two messages are aggregated, whereas in an optimal solution all messages are aggregated into a single packet arriving at s at time $2^{n+1} - 1$. Thus, the CC solution has value n against an optimal value of 1 for the MIN MAX ENERGY objective and $\sum_{j=1}^n (2^{n+1} - 2^j) = (n-1)2^{n+1} + 2$ against an optimal value of for the $2^{n+1} - 2$ for the MIN SUM ENERGY objective. Notice that $U = 2^n - 1$ here.

The following theorem shows that CC is best possible (up to a multiplicative constant) for the MIN MAX ENERGY objective. For the MIN SUM ENERGY objective we do not have any better lower bound on the competitive ratio of algorithms than the one of Theorem 5 for the centralized model.

14

Theorem 9. *Any deterministic synchronous algorithm is $\Omega(U^*)$-competitive for the* MIN MAX ENERGY *sensor problem.*

Proof. Consider an intree of depth $\delta = 2^{n+1}$ with n the number of messages, and where each node, except the leaves, has indegree n. We assume $\tau(a) = 1$ for all $a \in A$. For any on-line algorithm we will construct an adversarial sequence of n messages all with latency $L = \delta$, such that there exists a node at which the adversary can aggregate all messages in a single packet, but at which none of them is aggregated by the on-line algorithm. Using a similar argument as in the proof of Lemma 1 (i) the fact that all messages can be aggregated in a single packet implies that there exists a solution such that every node sends at most one packet, hence the cost of the adversarial solution is 1, whereas the cost of the on-line algorithm is n.

Fix any on-line algorithm. Given an instance of the problem, let $W_j(u)$ be the time interval message j spent at node u by application of the algorithm, the waiting time interval of message j on u. We denote its length by $|W_j(u)|$. Note that $\sum_u |W_j(u)| \leq |I_j|$ for each message j. We notice that the waiting time of a message in a node can be influenced by the other messages that are present at that node or have passed that node before. Since the algorithms are distributed the waiting time of a message in a node is not influenced by any message that will pass the node in the future.

The adversary chooses the source node v_j with total transit time $\tau_{v_j} := \delta - 2^j$ from s, for $j = 1, \ldots, n$, so that $|I_j| = 2^j$. Thus, $U = 2^{n-1} = \delta/4$. The choice of the exact position of v_j and the release time r_j is made sequentially and, to facilitate the exposition, described in a backward way starting with message n. The proof follows rather directly from the following claim.

Claim. For any set of messages $\{k, \ldots, n\}$ the adversary can maintain the properties:
(i) all messages in $\{k, \ldots, n\}$ pass a path p_k with 2^k nodes;
(ii) $I_k(u) = \bigcap_{j \geq k} I_j(u) \; \forall u \in p_k$;
(iii) if $k < n$, then $W_{k+1}(u) \bigcap I_k(u) = \emptyset \; \forall u \in p_k$;
(iv) if $k < n$, then $W_i(u) \bigcap W_j(u) = \emptyset \; \forall u \in p_k, \; i = k, \ldots, n, \; j > i$.

We notice that for any message j and any node u on the path from v_j to s, $W_j(u)$ may have length 0 but is never empty; it contains at least the departure time of message j from node u.

Note that properties (i) and (ii) for $k = 1$ imply that all messages can indeed be aggregated into one packet, hence as argued above, the adversarial solution has a cost of 1. Properties (iv) and (i) for $k = 1$ imply that the on-line algorithm sends all messages separately over a common path with 2 nodes, yielding a cost of n. This proves the theorem.

We prove the claim by induction. The basis of the induction, $k = n$, is trivially verified. Suppose the claim holds for message set $\{k, \ldots, n\}$ and p_k is the path between nodes $\overline{v}$ and $\hat{v}$. We partition p_k into two sub-paths $\overline{p}$ and $\hat{p}$ consisting of 2^{k-1} nodes each, such that $\overline{v} \in \overline{p}$ and $\hat{v} \in \hat{p}$. We denote the last node of $\overline{p}$ by $\overline{u}$ and the first node of $\hat{p}$ by $\hat{u}$. We distinguish two cases with respect to the waiting times the algorithm has selected for message k in the nodes on p_k.

CASE a: $\sum_{u \in \overline{p}} |W_k(u)| \geq (1/2)|I_k|$. The adversary chooses v_{k-1} with total transit time $\tau_{v_{k-1}} = \delta - 2^{k-1}$ such that its path to s traverses $\hat{p}$ but not $\overline{p}$. More precisely, we ensure that the first node message $k - 1$ has in common with any other message is $\hat{u}$. This is always possible, since the node degree is n. This choice immediately makes that setting $p_{k-1} = \hat{p}$ satisfies property (i). The release time of $k - 1$ is chosen so that $I_{k-1}(\hat{u})$ and $I_k(\hat{u})$ start at the same time, implying that $I_{k-1}(u)$ and $I_k(u)$ start at the same time for every $u \in \hat{p}$. Since $|I_{k-1}(u)| = |I_k(u)|/2$ we have $I_{k-1}(u) \subset I_k(u)$ for all $u \in \hat{p}$, whence property (ii) follows by induction.

Note that, as we consider distributed algorithms, message $k - 1$ does not influence the waiting time of $j, j > k - 1$, on $\overline{p}$ as $\hat{u}$ is the first node which both j and $k - 1$ traverse. In particular, $W_k(u), \forall u \in \overline{p}$ is not influenced by $k - 1$

Now, the equal starting times of $I_{k-1}(\hat{u})$ and $I_k(\hat{u})$ together with $\sum_{u \in \overline{p}} |W_k(u)| \geq (1/2)|I_k|$ and $|I_{k-1}(\hat{u})| = |I_k(\hat{u})|/2$ imply that k will not reach $\hat{u}$ before interval $I_{k-1}(\hat{u})$ ends. This, together with the consideration above, implies property (iii).

To prove (iv), note that by induction it is sufficient to prove that $W_{k-1}(u) \cap W_j(u) = \emptyset \; \forall j > k - 1 \; \forall u \in \hat{p}$. Since, as just proved, $W_k(u) \cap I_{k-1}(u) = \emptyset \; \forall u \in \hat{p}$ we have $W_{k-1}(u) \cap W_k(u) = \emptyset \; \forall u \in \hat{p}$.

15

We have by induction that, for $j > k$, $W_j(u) \cap I_{j-1}(u) = \emptyset \; \forall u \in \hat{p}$ and we just proved that $I_{k-1}(u) \subset I_{j-1}(u) \subset I_j(u) \; \forall u \in \hat{p}$, which together imply $W_{k-1}(u) \cap W_j(u) = \emptyset \; \forall j > k \; \forall u \in \hat{p}$.

CASE b: $\sum_{u \in \overline{p}} |W_k(u)| < (1/2)|I_k|$. As in the previous case, the adversary chooses v_{k-1} with total transit time $\tau_{v_{k-1}} = \delta - 2^{k-1}$ such that its path to s traverses $\overline{p}$ (therefore also $\hat{p}$) but does not intersect any of the paths used by messages $\{k, \ldots, n\}$ before it reaches $\overline{p}$ in $\overline{v}$. Again, this is always possible since the indegree of each node is n. Hence, choosing $p_{k-1} = \overline{p}$ satisfies property (i). The release time of $k-1$ is chosen so that $I_{k-1}(\overline{v})$ and $I_k(\overline{v})$ end at the same time, implying that $I_{k-1}(u)$ and $I_k(u)$ end at the same time for every $u \in \overline{p}$. Since $|I_{k-1}(u)| = |I_k(u)|/2$ we have $I_{k-1}(u) \subset I_k(u)$ for all $u \in \overline{p}$, whence property (ii) follows by induction.

The equal ending times of $I_{k-1}(\overline{u})$ and $I_k(\overline{u})$ together with $\sum_{u \in \overline{p}} |W_k(u)| < 1/2|I_k|$ and $|I_{k-1}(\overline{u})| = |I_k(\overline{u})|/2$ implies that k has left $\overline{u}$ before $I_{k-1}(\overline{u})$ begins, implying property (iii). Indeed, this gives $W_{k-1}(u) \cap W_k(u) = \emptyset, \forall u \in \overline{p}$. It also implies that $k-1$ could not influence the waiting time of k on $\overline{p}$.

The proof of (iv) follows the very same lines as in Case a, with the difference that we now refer to nodes in $\overline{p}$ instead of $\hat{p}$. $\qquad\square$

Since in the proof $U = \delta/4$ we also have the following lower bound on the competitive ratio of any deterministic synchronous algorithm.

Theorem 10. *Any deterministic synchronous algorithm is $\Omega(\log \delta)$-competitive for the* MIN MAX ENERGY *sensor problem.* $\qquad\square$

4.3 The asynchronous model

In this paragraph we consider deterministic algorithms for the asynchronous model. We propose a deterministic algorithm and analyze its competitive ratio. We also provide a lower bound on the competitive ratio for a broad class of algorithms including this algorithm.

In the asynchronous model nodes are equipped with a clock and a distributed algorithm. All clocks have the same time unit, but neither the time nor the start of a new time unit on clocks is synchronized. We assume that $\tau(a) = 1$ for all a, such that τ_{v_j} is equal to the number of nodes on the $v_j - s$-path.

We propose algorithm Spread Latency (SL) for this model, which divides the latency minus transmission time of each message j equally over the nodes on the $v_j - s$-path: at each node of this path the message is assigned a waiting time of $(L_j - \tau_{v_j})/\tau_{v_j}$ time units. As soon as messages appear simultaneously at the same node they get aggregated into a packet, which is sent over the outgoing arc as soon as the waiting time of at least one of its messages at that node has passed.

In this way, no message is delayed due to aggregation and thus the algorithm yields a feasible solution.

Let, as in the previous subsection, $U := \frac{\max_j |I_j|}{\max\{1, \min_j |I_j|\}} = \frac{\max_j (L_j - \tau_{v_j})}{\max\{1, \min_j (L_j - \tau_{v_j})\}}$.

Theorem 11. *The algorithm* SL *is $O(\delta U^*)$-competitive for* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. We prove that for all $a \in A$ the number of packets SL sends through a is at most $O(\delta U^*)$ times that number in an optimal solution.

Let $\lambda := \max\{1, \min_j (L_j - \tau_{v_j})\}$. Consider a packet P of messages sent by an optimal solution through arc (u, v) at time t. Without loss of generality we do not consider messages for which $\min_j (L_j - \tau_{v_j}) = 0$ as these messages have to be sent upon release by both SL and the optimal algorithm. To bound the number of packets sent by SL that contain at least one message from P, define $P_k := \{j \in P \mid 2^{k-1}\lambda \le L_j - \tau_{v_j} < 2^k\lambda\}$, for $k = 1, \ldots, \lceil \log U \rceil$. We *charge* any sent packet to the message that caused the packet to be sent due to its waiting time being over. It suffices to prove that the number of packets charged to messages in P_k is $O(\delta)$.

Since the waiting time of messages $j \in P_k$ at node u is at least $2^{k-1}\lambda/\delta$, the delay between any two packets that are charged to messages in P_k is at least $2^{k-1}\lambda/\delta$. Since the optimal solution sends

packet P at time t through arc (u,v), we get $t \in I_j(u) \ \forall j \in P$ and thus $I_j(u) \subseteq [t-2^k\lambda, t+2^k\lambda] \ \forall j \in P_k$. Thus, the number of packets charged to messages in P_k is at most $2 \cdot 2^k\lambda/(2^{k-1}\lambda/\delta) = 4\delta$. $\quad\square$

The competitive ratio of SL can be $\Omega(\delta \log \delta)$-competitive, for both objectives. Consider the chain $u_m, \ldots, u_1, s$ with unit transit times; i.e. $m = \delta$. We assume $\delta > 4$. An adversary releases messages $j_{i,k}$ at node u_m for $i = 1, \ldots, \delta/4$ and $k = 0, \ldots, \log_2 \delta - 1$. The release time of message $j_{i,k}$ is $r(i,k) = \delta + \frac{2^{k+1}}{\delta}i - 2^k$ and the latency is $L(i,k) = 2^k + \delta$. It follows from the observations $r(i,k) < r(i+1,k)$ and $r(\delta/4,k) < r(1,k-1)$ that messages are released ordered by decreasing order of k and then by increasing order of i. This induces the total order $\prec$ on pairs (i,k) and (i',k'). Formally, $(i,k) \prec (i',k')$ if either $k > k'$ or $k = k'$ and $i \leq i'$.

We have $I_{j_{i,k}} = [r(i,k) + \delta, r(i,k) + L(i,k)]$. As $r(i,k) + \delta \leq 2\delta$ and $r(i,k) + L(i,k) \geq 2\delta$ for each message $j_{i,k}$, we have $\bigcap_{j \in j_{i,k}} I_j \neq \emptyset$, hence the adversary may aggregate all messages at their common release node u_m. Let $W_j(u)$ be, as defined before, the waiting time interval of message j on u determined by SL. We have $W_{j_{i,k}}(u_m) = [r(i,k), r(i,k) + \frac{L(i,k)-\delta}{\delta}]$. It follows from simple arithmetics that $r(i,k) + \frac{L(i,k)-\delta}{\delta} < r(i',k')$ for all pairs $(i,k), (i',k')$ such that $(i,k) \prec (i',k')$. Hence, $W_j(u_m) \cap W_{j'}(u_m) = \emptyset$ for any two messages j and j' and SL sends all messages separately from u_m. This implies that SL is $\Omega(\delta \log \delta)$-competitive for MIN MAX ENERGY with unit costs. For cost structure $c(u_m, u_{m-1}) = C \geq \delta(\log \delta + 1), c(a) = 1 \forall a \backslash (u_m, u_{m-1})$ SL is $\Omega(\delta \log \delta)$-competitive for MIN SUM ENERGY objective.

SL determines the waiting time of each message at the nodes it traverses independently of all other messages. We call such an algorithm a WI-algorithm. To be precise, in a WI-algorithm node v determines the waiting time of message j based only on the message characteristics (v_j, r_j, d_j), transit time to the sink τ_v and clock time. The following lower bound shows that the competitive ratio of SL cannot be beaten by more than a factor U^* by any other WI-algorithm for the MIN MAX ENERGY problem. In the derivation of the lower bound we restrict to WI-algorithms that employ the same algorithm in all nodes with the same transit time to s. This is not a severe restriction, given that transit time to s is the only information about the network that a node has.

Again, as in the synchronous model we do not have any better lower bound on any subclass of algorithms for the MIN SUM ENERGY problems than the bound of 2 from Theorem 5 for the centralized model.

Theorem 12. *Any deterministic asynchronous WI-algorithm is $\Omega(\delta^{1-\epsilon})$-competitive for the MIN MAX ENERGY sensor problem, for any $\epsilon > 0$.*

Proof. Consider a binary intree with root s and all leaves at distance δ from s. Let $0 \leq \lambda < 1$ be such that $\delta^{1-\lambda} \geq 3$. An adversary releases message 1 with latency L at time r_1 in a leaf v_1. Notice that there are at most δ^λ nodes where the waiting time is at least $(L - \tau_{v_1})/\delta^\lambda$. Hence, the $v_1 - s$ path contains a sub-path consisting of at least $\delta^{1-\lambda} - 2$ nodes where in each node message 1 waits less than $(L - \tau_{v_1})/\delta^\lambda$. Choose such a sub-path and let u be the node on it which is closest to s.

Let V' be the set of leaves of the subtree with root u and depth $\delta^{1-\lambda} - 2$. $|V'| \geq 2^{\delta^{1-\lambda}-2} \geq \delta^\lambda/4$ for any fixed $\lambda \in [0, 1)$ and δ large enough. The adversary releases messages $j = 2, \ldots, \delta^\lambda/4$ with latency L at times $r_j = r_1 + j(L - \tau_{v_j})/\delta^\lambda$ in leaf v_j, such that each $v_j - s$ path passes through a different vertex of V'. Because $\tau_{v_j} = \tau_{v_1} \ \forall j$ and we assumed that any WI-algorithm applies the same algorithm in nodes at equal distance, all messages are sent non-aggregated to and from u, whereas they are aggregated as early as possible in an optimal solution, in particular at u. $\quad\square$

The lower bound does not hold for arbitrary algorithms as a node may adjust the waiting time of subsequent messages that traverse that node. However, we notice that only if a node delays subsequent messages longer the competitive ratio might be better. If the waiting time at a node does not increase for subsequent messages, the competitive ratio remains $\Omega(\delta^{1-\epsilon})$. The following theorem shows that the lower bound remains $\Omega(\delta^{1-\epsilon})$ if release nodes do not delay subsequent messages longer than preceding messages.

Theorem 13. *Any asynchronous WI-algorithm for which the waiting time of message j at its release node is at most $\frac{L-\tau_{v_j}}{K}$ is $\Omega(\min\{K, n\})$-competitive for the MIN MAX ENERGY objective.*

Proof. Consider a chain which consists of two nodes v and s. We assume a constant latency of L for each message. The adversary releases $K-1$ messages with an interval of $(L-\tau_{v_j})/(K-1)$ at v. Since the waiting time of message j at v is at most $(L-\tau_{v_j})/K$, none of these messages are aggregated in the on-line solution, whereas they are all aggregated in one packet in an optimal solution. $\qquad\square$

For arbitrary asynchronous algorithms we do not have any better lower bound than those in Theorems 9 and 10 for the synchronous case. Furthermore, notice that for any WI-algorithm to have a competitive ratio better than some constant times the number of messages, it should delay messages at their release node.

Improved algorithm for the case of a chain. We describe an algorithm with improved competitive ratio for the case that the network is a chain with s in one of its ends. We first introduce a classification of the nodes on the chain. Initially, all nodes are unclassified. For $p = \lfloor \log \delta \rfloor, \ldots, 0$ we assign all unclassified nodes v with $2^p | \tau_v$ to class p. I.e. v is of class p if p is the maximal integer such that $\tau_v = k2^p$ for some integer k.

Notice that the resulting classification of nodes has the property that between any pair of nodes of the same class there is at least one node of higher class.

To describe the algorithm for message j released at node v_j we set $w_j := (L_j - \tau_{v_j})/(\lfloor \log \tau_{v_j} \rfloor + 1)$. The rough idea of the algorithm is as follows: When message j reaches a node v of class p and j has not visited a node of higher class yet, it waits at node v for w_j time units. Otherwise, if j has already visited a node of class larger than p, it does not wait at node v at all. Since between any pair of nodes of the same class there is at least one node of higher class, a message will wait at most once at a node of class p for each $p = 0, \ldots, \lfloor \log \delta \rfloor$. Moreover, the highest class of a node message j will find on its path from v_j to the sink is $\lfloor \log \tau_{v_j} \rfloor$. Thus, the overall waiting time of message j accumulates to no more than

$$w_j(\lfloor \log \tau_{v_j} \rfloor + 1) = L_j - \tau_{v_j}.$$

Therefore the message arrives at the sink at or before time $r_j + \tau_{v_j} + L_j - \tau_{v_j} = d_j$.

As in the SL algorithm, messages are aggregated into a packet if they appear simultaneously at an intermediate node. This packet is sent over the outgoing arc as soon as the waiting time of at least one of its messages at that node has passed (in particular, if a packet contains a message that does not wait at a particular node, then the packet does not wait at that node). Notice that no message is delayed due to aggregation and the algorithm thus yields a feasible solution.

Theorem 14. *The described algorithm is $O(\log^3 \delta)$-competitive for* MIN MAX ENERGY *and* MIN SUM ENERGY.

Proof. We prove that the number of packets sent through an arc by the algorithm above is at most $O(\log^3 \delta)$ times the number of packets sent by an optimal solution through this arc.

Let $\lambda := \max\{1, \min_j(L_j - \tau_{v_j})\}$. Consider a packet P of messages sent by an optimal solution through arc $a \in A$ at time t. We derive a bound on the number of packets sent by our algorithm that contain at least one message from P.

As above, we define $P_k := \{j \in P \mid 2^{k-1}\lambda \leq L_j - \tau_{v_j} < 2^k \lambda\}$ for $k = 1, \ldots, \lceil \log U \rceil$. Moreover, for $p = 0, \ldots, \lfloor \log \delta \rfloor$, let $P_{k,p}$ denote the subset of P_k that consists of all messages j that have visited a node of class p but no node of class $p+1$ before being sent through arc a. Notice that the number of subsets $P_{k,p}$ is $O(\log \delta \log U) = O(\log^2 \delta)$.

We charge any packet sent by our algorithm to one of its messages $j \in P_{k,p}$ where k and p are chosen maximally. The waiting time of a message $j \in P_{k,p}$ at a node of class p is at least $2^{k-1}\lambda/(\log \delta + 1)$. On the other hand, messages in $P_{k,p}$ can only pass this node of class p within a time interval of $2 \cdot 2^k \lambda$ (see proof of Theorem 11). Thus, the number of packets charged to messages in $P_{k,p}$ is at most $4(\log \delta + 1) \in O(\log \delta)$. Since the number of subsets $P_{k,p}$ is $O(\log^2 \delta)$, the result follows. $\qquad\square$

We give a slightly better result for the case of common latency L.

Corollary 2. *If all messages share the same latency L, the competitive ratio of the algorithm is $O(\log^2 \delta)$.*

Proof. In the proof of Theorem 14, it can be observed that for $k > k'$ and $p < p'$ it holds that $P_{k,p} \neq \emptyset$ implies $P_{k',p'} = \emptyset$ (that is, messages that originate further away from the sink have seen nodes of higher classes). In particular, the number of nonempty subsets $P_{k,p}$ is $O(\log \delta + \log U) = O(\log \delta)$ in this case. This yields the improved competitive ratio $O(\log^2 \delta)$. □

It follows from Theorem 12 that this algorithm is $\Omega(\log \delta)$-competitive on a chain.

5 Conclusions and Open Problems

The results we presented in this paper are the first results on a rich class of problems which are both theoretically and practically interesting. We have described the great variety of problems in this class in the introduction.

The main results of this paper are summarized in Table 1. As the table shows many challenging questions remain open. First, we comment on the results and open questions within the models studied. Then, we conclude with research questions for various other models that we consider particularly interesting.

Model $Off-line$	Graph	Min Max Energy *complexity*	*approximation lower bound*	Min Sum Energy *complexity*	*approximation lower bound*
	tree	$NP-hard,\ 2-apx$	*open*	$NP-hard,\ 2-apx$	*open*
	chain	*open*	*open*	*polynomial time*	$-$
$On-line$		*competitive ratio upper bound*	*competitive ratio lower bound*	*competitive ratio upper bound*	*competitive ratio lower bound*
$-centralized$	*tree*	$O(U^*)$	2	$O(U^*)$	2
	chain const. latency	2	$\sqrt{2}$	2	$\sqrt{2}$
$-synchronous$	*tree*	$O(U^*)$	$\Omega(U^*)$	$O(U^*)$	2
$-asynchronous$	*tree*	$O(\delta U^*)$	$\Omega(U^*)$	$O(\delta U^*)$	2
	chain	$O(\log^3 \delta)$	2	$O(\log^3 \delta)$	2

Table 1. Approximation and competitiveness results on the sensor problem

For the off-line problem we have proved that the problem is **NP**-hard on a tree; we presented a 2-approximation for both objectives, but we have no lower bounds on the approximability. It would be interesting to see if there exists an approximation preserving reduction from an **APX**-hard problem. For the special case where the graph is a chain and message latencies are constant we presented a Dynamic Programming formulation which solves the Min Sum Energy sensor problem in $O(n^3)$ for this case.

Of all on-line models, the centralized model poses most open questions. It is not clear at all whether there exists a on-line algorithm with constant competitive ratio for arbitrary latencies. Although this model is least relevant in practice, the state of present knowledge is theoretically unsatisfactory.

For the synchronous model we presented an $O(U^*)$-competitive algorithm for both objectives and we showed that this algorithm is best possible for Min Max Energy. For the asynchronous

model we presented an $O(\delta U^*)$ -competitive algorithm for both objectives. There remains a gap between this bound and the lower bound of $\Omega(U^*)$ from the synchronous model. Theorem 12 shows that improvements in the upper bound should come from algorithms that are essentially different from the one we presented here.

We conclude our paper with listing some research problems which, in our opinion, are practically and/or theoretically interesting:

- Extend the results in this paper to more general cost functions. This looks achievable for transit cost functions that are concave in packet size.

- Consider a geometrical model of the sensor network in which transmission to other nodes is possible within some range, where transmission to more distant nodes is more expensive (in transit cost) than to nearer nodes. In this case, in addition to which time to send also to which node to send a message becomes part of the decision process. As a first attempt at solving this problem one could concentrate on the tree, with a length-function on the arcs, as the underlying model.

References

1. I. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks Journal*, 38(4):393–422, 2002.
2. A. Boulis, S. Ganeriwal, and M. B. Srivastava. Aggregation in sensor networks: an energy - accuracy tradeoff. *Elsevier Ad-hoc Networks Journal (special issue on sensor network protocols and applications)*, 1(2-3):317–331, 2003.
3. C. Brito, E. Koutsoupias, and S. Vaya. Competitive analysis of organization networks or multicast acknowledgement: how much to wait? In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 627–635, 2004.
4. A. Z. Broder and M. Mitzenmacher. Optmial plans for aggregation. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 144–152, 2002.
5. J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing*, pages 1965–1970, 2001.
6. J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th ACM Symposium on Operating System Design and Implementation (OSDI)*, pages 147–164, 2002.
7. J. Elson and K. Römer. Wireless sensor networks: a new regime for time synchronization. *Computer Communication Review*, 33(1):149–154, 2003.
8. G. Finke, V. Jost, M. Queyranne, and A. Sebö. Batch processing with interval graph compatibilities between tasks. In *Les Cahiers du Laboratoire, 118*, Leibniz-IMAG, 2004.
9. S. Ganeriwal, R. Kumar, and M. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems (SenSys)*, pages 138–149, 2003.
10. A. Goel and D. Estrin. Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 499–505, 2003.
11. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy efficient communication protocols for wireless microsensor networks. In *Proceedings of Hawaiian International Conference on Systems Science*, pages 3005–3014, 2000.
12. F. Hu, X. Cao, and C. May. Optimized scheduling for data aggregation in wireless sensor networks. In *International Conference on Information Technology Coding and Computing (ITCC)*, pages 557–561, 2005.
13. C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, pages 414–458, 2002.
14. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.
15. K. Kalpakis, K. Dasgupta, and P. Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks*, 42(6):697–716, 2003.
16. S. Lindsey and C. S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Proceedings of IEEE Aerospace Conference*, pages 1125–1130, 2000.

17. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: A tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 5th ACM Symposium on Operating System Design and Implementation (OSDI)*, pages 131–146, 2002.

18. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems (TODS)*, 30(1):122–173, 2005.

19. W. Yuan, V. S. Krishnamurthy, and S. K. Tripathi. Synchronization of multiple levels of data fusion in wireless sensor networks. In *Proceedings of IEEE Globecom*, pages 221–225, 2003.