An Energy-Efficient Wireless Routing Protocol for Distributed Structural Health Monitoring

Andreas Engel
LOEWE Research Center AdRIA,
Darmstadt
engel@esa.cs.tu-darmstadt.de

Andreas Koch
Embedded Systems and Applications Group,
Technische Universität Darmstadt
koch@esa.cs.tu-darmstadt.de

***Abstract*—Congestion-induced packet loss leads to throughput degradation in wireless sensor networks and often requires energetically expensive retransmissions. Though collisions can be avoided (CSMA/CA), the required carrier-sensing also increases the sensor node energy consumption. In this paper, an energy-efficient routing protocol for a distributed monitoring application is proposed. Although the development of the new protocol was motivated by a specific use-case, it is applicable to all static-topology wireless sensor networks where multiple sensor nodes need to simultaneously distribute information to all other nodes in the network. First, we model the globally known network topology as two directed graphs representing reachability and interference. Based on this model, we define an integer linear program to find a collision-free communication schedule with the minimum number of transmissions and receptions required for distributing the data (multi-commodity information flooding). As these ILPs are hard to solve for larger networks, we also propose a heuristic for scheduling. Compared to other advanced flooding protocols, the proposed schedule can reduce wireless activity by up to $90\ \%$, with the heuristic solver achieving a solution quality just $4\ \%$ worse than the optimal ILP solution.**

I. Introduction

Routing protocols have a significant impact on distributed applications relying on multi-hop wireless communication. The number of nodes involved to deliver an information (a data item) from a source to a sink determines the networks overall energy consumption as well as the data throughput. Often, shortest routes can not be easily discovered, or are undesirable (e.g., when balancing communication load equally over all network nodes or to provide robust communication over redundant paths). In general, application-tailored routing protocols can exploit special characteristics to improve throughput and energy consumption.

We propose a routing protocol optimized for a distributed Structural Health Monitoring (SHM) application. In this application, modal parameters of large civil infrastructures such as bridges or buildings are periodically estimated from acquired vibration measurements. Serious damage of the structures is reflected by significant changes of the observed model parameters, and can thus be detected or even localized automatically. The modal parameters are extracted from acceleration signals captured by the sensor nodes distributed all over the structure. To dispense with energetically expensive actuators, output-only SHM applications rely on natural excitation such as wind or traffic. This requires the use of special algorithms such as Random Decrement Technique (RDT) [1] to recover the necessary structural information from the noisy sensor data measured at the randomly excited structure. In RDT, a small number of nodes is selected as so called *reference nodes*. Each time the sensor signal of a reference node over- or undershoots a predefined threshold, a trigger information describing the time and location of the trigger event has to be forwarded to all other nodes in the network. These trigger events are used to cancel the random parts of the measurements by averaging across multiple nodes, thus leaving only structural information in the captured signals [2].

From the perspective of the routing protocol, this application has some specific characteristics worth exploiting. First, there is a fixed (and small) number $s \in \mathbb{N}$ of information sources (the reference nodes), with all nodes acting as information sinks. Thus, an s-to-all flooding protocol is required. Second, time and location of trigger information generation is strongly correlated. If the structure is sufficiently excited to cause a trigger event at a certain reference node, it is quite likely that other nearby reference nodes will also generate trigger events, and that even more trigger events will be generated shortly afterwards. Third, the sensor nodes are attached to the structure. Apart from node failures, the network topology can thus be assumed to be static. Furthermore, the trigger information generated at the source nodes is quite small. Thus, information on multiple triggers can be aggregated into a single radio packet. And finally, the monitoring application relies on a highly accurate time synchronization mechanism for correct data gathering. This can be exploited by the routing algorithm to optimize the scheduling of data reception and transmission.

The main contribution of this work is the description of an ILP and a heuristic for scheduling receivers and transmitters in a wireless sensor network, such that the overall radio energy cost is minimized while simultaneously flooding information from multiple source nodes throughout the network.

This paper is organized as follows. Section II provides an overview over available routing protocols that may be used for the described application. In Section III, a cost-function and the boundary conditions defining the abstract routing problem are declared to simplify the transfer of the proposed solutions to similar problems. An optimal and a heuristic solution for the specific routing problem are proposed afterwards. These are compared to conventional routing protocols regarding the defined cost function in Section IV. The discussion of possible improvements in Section V concludes this work.

978-1-4799-3060-9/14/$31.00 ©2014 IEEE

II. Related work

Flooding information into a wireless network has been widely adopted for general tasks such as time synchronization [3]–[5] or route discovery [6]. However, the basic blind flooding is quite inefficient especially in dense networks, as each receiver rebroadcasts the message to be flooded [7]. Intense research has been carried out to reduce the number of necessary rebroadcasts. Prior work commonly utilizes the knowledge about the nodes locations [8], [9] or their local neighborhood [10]–[13] to select the forwarding nodes covering the most as-yet uncovered network nodes. However, all of these flooding protocols assume a 1-to-all data propagation. Unfortunately, unlike real water waves, multiple data waves originating from different source nodes interfere with each other due to channel congestion and signal interferences such as the hidden terminal problem [14].

Furthermore, the information from multiple sources should be aggregated as soon as possible to avoid repeated data transfers between the same nodes. Many data aggregation and clustering protocols have been reported [15]. By selecting specific nodes as cluster-heads, data is collected locally before it is forwarded to a single sink node. The basic idea of clustering is picked up in the design of the greedy heuristic in Section III-C.

Using Integer Linear Programs to solve scheduling problems is a widely adopted approach. In the context of wireless sensor networks, ILPs were used for frequency channel assignment [16], task scheduling [17], and routing [18]. The later one is closely related to this work as it generates the ILP formulation based on a directed graph representing the network topology. The ILP solution proposed in [18] describes a multi-cycle schedule for transmissions and receptions with a minimal amount of overall required energy. However, the schedule is used to transport information from multiple source nodes just to a *single* sink, it does not perform flooding. Furthermore, it does not provide information aggregation and does not distinguish between transmission and interference ranges of the radio transceivers. All of these aspects are considered in our approach, and will be discussed in the following section.

III. Proposed Routing Algorithm

As detailed in Section I, a small number of source nodes is assumed to generate information that has to be distributed to all other nodes in the network. Furthermore, the source nodes will generate new information approximately at the same time as the sensed information is physically correlated. The main goal of the proposed routing scheme is to simultaneously flood the information from all source nodes throughout the network with as little overall radio activity as possible.

As the network topology is assumed to be static and the nodes are time-synchronized to each other, we propose to calculate a schedule that selects appropriate nodes as transmitters and receivers in subsequent time slots (cycles). This static schedule must be known by all nodes and is executed periodically, i.e. the source nodes buffer collected information until the restart of the schedule. The main advantage of this scheme is twofold. First, all nodes not scheduled for reception in a certain cycle can keep their radios turned off as they do not have to listen for potential incoming data. Second, the global topology knowledge can be used to deliberately aggregate information from different source nodes.

Due to the correlation of the sensor nodes, at each restart of the schedule it is most likely that either *all* source nodes have buffered some information, or that *no* source node has buffered any information. In the first case, the schedule can be executed as intended. In the second case, the source nodes actually scheduled for transmission will not start sending, which must be recognized at the receivers scheduled for listening by a short timeout. These receivers, in turn, cancel their scheduled forwarding transmission, thus propagating the transmission cancellation until the end of the schedule. The energy overhead for an unused schedule is thus restricted to a small number of clear channel assessments at the scheduled receivers. The same mechanism of transmission cancellation must be applied if only a subset of source nodes provides new information at the start of a new schedule. In this case, the resulting flooding routes may not be optimal for the specific subset of source nodes. However, the probability of those cases is small and will not significantly impact the overall energy efficiency.

The reminder of this section focuses on calculating an optimal schedule for a given network topology.

A. Network model and cost function

We define an application independent network model M as

$$
\begin{aligned}
M &= (N, N_S, E_I, E_C, h) && (1)\\
N &\subseteq \mathbb{N} && (2)\\
N_S &\subseteq N && (3)\\
E_I &\subseteq N^2 && (4)\\
E_C &\subset E_I && (5)\\
h &: N^2 \to \mathbb{N}. && (6)
\end{aligned}
$$

This model combines two directed graphs $M_I = (N, E_I)$ and $M_C = (N, E_C)$ with a common set of network nodes N. M_C represents the networks single-hop connectivity, i.e., node $b \in N$ may receive information directly from node $a \in N$ iff $(a, b) \in E_C$. M_I represents the interference relationship within the wireless network, i.e., the transmission of node a prevents the successful reception of any other information at node b iff $(a, b) \in E_I$.[1] M_C is thus a subgraph of M_I. The information-generating source nodes are denoted as N_S. The function h determines the minimum number of hops between two nodes. These hop counts h are pre-computed by the all-pairs-shortest-path algorithm [19].

The network model M can be generated from the known locations and the transmission and reception characteristics of all network nodes, shown in Section IV. A more accurate model accounting for obstacles can be obtained from well known online topology discovery algorithms (e.g. recursive neighborhood exchange [20]), executed once after network deployment. However, detailed topology detection is outside the scope of this paper and we assume M to be known and static.

[1]For clarity, we will use a for nodes acting as transmitters, b for nodes acting as receivers, and v for arbitrary network nodes.

Based upon the network model, we define

$$S \quad : \quad N \times \mathbb{N} \rightarrow \{\mathit{RX}, \mathit{TX}, \mathit{IDLE}\} \tag{7}$$

$$I \quad : \quad N \times \mathbb{N} \rightarrow 2^{N_S} \tag{8}$$

a schedule S assigning a radio transceiver status $S(v,t)$ to each network node $v \in N$ in every scheduling cycle $t \in \mathbb{N}$. The information assignment function $I(v,t)$ denotes the subset of source nodes that have already forwarded their information to node v before cycle t such that the information is known to v in cycle t. Initially, only the source nodes know about their information:

$$I(v,0) = \begin{cases} \{v\}, & \text{if } v \in N_S \\ \varnothing, & \text{otherwise} \end{cases} \tag{9}$$

The information is then propagated within each cycle t from any node a scheduled for transmission to any node b within the transmission range of a if b is scheduled for reception and not disturbed by another transmitter c:

$$\begin{aligned} I(b,t+1) \quad = \quad & I(b,t) \\ \cup \quad & \begin{cases} I(a,t), & \text{if } (a,b) \in E_C \;\wedge \\ & S(a,t) = \mathit{TX} \;\wedge \\ & S(b,t) = \mathit{RX} \;\wedge \\ & \forall (c,b) \in E_I \backslash \{(a,b)\} : \\ & S(c,t) \neq \mathit{TX} \\ \varnothing, & \text{otherwise} \end{cases} \end{aligned} \tag{10}$$

The length $L(S)$ of a schedule S is the minimum number of cycles required to flood all source node information to all nodes in the network:

$$L(S) \quad = \quad \min_{t \in \mathbb{N}} \forall v \in N : I(v,t) = N_S \tag{11}$$

As we try to minimize the networks overall energy consumption $C(S)$ for a schedule S, we define the sum of scheduled receptions and transmissions (more precisely, the number of time intervals with enabled transmitters or receivers) as the primary optimization goal:

$$C(S) = \sum_{t=0}^{L(S)-1} |\{v \in N : S(v,t) \neq \mathit{IDLE}\}| \tag{12}$$

Radio transceivers draw about the same power when receiving and transmitting data. Thus, receiving and transmitting the same amount of data consumes about the same amount of energy. The cost function therefore does not distinguish between transmission and reception costs. However, weighting factors could be easily integrated as needed.

For two schedules of equal cost, the shorter schedule yields smaller latency and increased data throughput. Therefore, the schedule length is the secondary optimization goal. Definition 1 summarizes the optimization problem to be solved in Section III-B and III-C.

Definition 1 (*Optimal Flooding Schedule*)

For any network model M find a schedule S of finite length, such that

$$C(S) < C(\hat{S}) \vee \Big(C(S) = C(\hat{S}) \wedge L(S) \leqslant L(\hat{S})\Big) \tag{13}$$

for any valid schedule $\hat{S}$.

B. Integer Linear Program for Optimal Scheduling

Finding an optimal solution for the problem stated in Definition 1 requires solving multiple set-cover problems, e.g. finding the smallest number of forwarding nodes to cover the remaining, not yet reached nodes. The optimal scheduling problem is thus NP-complete. To find an optimal solution for non-trivial network sizes in a reasonable time, parallelizable branch-and-bound algorithms have to be utilized. We therefore translate the network model (Equation 1) and the information propagation rules (Equations 9 to 12) into an Integer Linear Program (ILP). A commercial ILP-solver is then utilized to compute the optimal schedule (see Section IV).

The ILP-solver determines integer values for a set of variables. The solution space is restricted by a set of constraints, which are relations between linear combinations of the variables and constant values. Within this solution space, a single objective is minimized. This objective is also a linear combination of the variables.

When working with binary variables, basic operations like disjunctions and conjunctions have to be replaced by the following constraints:

$$y = \bigvee_{i=1}^{n} x_i \quad \Leftrightarrow \quad y \geqslant x_1 \wedge \ldots \wedge y \geqslant x_n \wedge y \leqslant \sum_{i=1}^{n} x_i \tag{14}$$

$$y = \bigwedge_{i=1}^{n} x_i \quad \Leftrightarrow \quad y \geqslant \sum_{i=1}^{n} x_i - (n-1) \wedge n \cdot y \leqslant \sum_{i=1}^{n} x_i \tag{15}$$

To formulate the ILP for the optimal scheduling, the following binary variables are used:

$$\alpha_{a,t} \quad \Leftrightarrow \quad S(a,t) = \mathit{TX} \tag{16}$$

$$\beta_{b,t} \quad \Leftrightarrow \quad S(b,t) = \mathit{RX} \tag{17}$$

$$\gamma_t \quad \Leftrightarrow \quad \exists v \in N : S(v,t) \neq \mathit{IDLE} \tag{18}$$

$$\delta_{s,v,t} \quad \Leftrightarrow \quad s \in I(v,t) \tag{19}$$

$$\epsilon_{s,a,b,t} \quad \Leftrightarrow \quad \delta_{s,a,t} \wedge \alpha_{a,t} \wedge \beta_{b,t} \tag{20}$$

To define the necessary constraints for each schedule cycle t, an upper bound $L_{\max}$ for the length of the resulting schedule is required. This upper bound can be derived from the network model, as a naive sequential blind flooding of all information would not require more then $|N| \cdot |N_S|$ cycles. Alternatively, the outcome of the heuristic scheduling in Section III-C can be used to define tighter bounds for the schedule length. This is important, as the ILP formulation requires a total number of $L_{\max} \cdot (2|N| + 1 + |N_S|(|N| + |E_C|))$ variables.

Let $T = \{0, \ldots, L_{\max} - 1\}$ be the set of potential schedule cycles. The objective to be minimized is described as

$$\textbf{Minimize} \quad L_{\max} \cdot \sum_{t \in T} \sum_{v \in N} (\alpha_{v,t} + \beta_{v,t}) + \sum_{t \in T} \gamma_t \tag{21}$$

to satisfy Equation 13. The right sum calculates the number of actually required cycles $L(S)$. The complete distribution of information tested in Equation 11 is ensured by constraints (Equation 27). The left sum calculates $C(S)$ according to Equation 12. It is weighted by $L_{\max}$ to make $C(S)$ the primary objective.

The following constraints are required to distinguish between active and idle cycles:

$$\forall t \in T, v \in N : \gamma_t \geqslant \alpha_{v,t} \tag{22}$$
$$\forall t \in T, v \in N : \gamma_t \geqslant \beta_{v,t} \tag{23}$$
$$\forall t \in T : \gamma_t \leqslant \sum_{v \in N} (\alpha_{v,t} + \beta_{v,t}) \tag{24}$$
$$\forall t > 0 : \gamma_t \leqslant \gamma_{t-1} \tag{25}$$

According to Equation 14, $\gamma_t = \bigvee_{v \in N} \alpha_{v,t} \vee \beta_{v,t}$ is realized by Equations 22 to 24. Thus, a cycle is active, iff any node is scheduled for transmission or reception in this cycle. Equation 25 moves all idle cycles to the end of T.

The following constraints ensure correct information propagation:

$$\forall s \in N_S, t \in T : \delta_{s,v,t} = 1 \tag{26}$$
$$\forall s \in N_S, v \in N \backslash \{s\} : \delta_{s,v,L_{\max}-1} = 1 \tag{27}$$
$$\forall s \in N_S, v \in N \backslash \{s\}, t < h(s,v) : \delta_{s,v,t} = 0 \tag{28}$$
$$\forall s \in N_S, b \in N \backslash \{s\}, t \geqslant h(s,b), (a,b) \in E_C :$$
$$\epsilon_{s,a,b,t-1} \geqslant \delta_{s,a,t-1} + \alpha_{a,t-1} + \beta_{b,t-1} - 2 \tag{29}$$
$$3 \cdot \epsilon_{s,a,b,t-1} \leqslant \delta_{a,s,t-1} + \alpha_{a,t-1} + \beta_{b,t-1} \tag{30}$$
$$\delta_{s,b,t} \geqslant \epsilon_{s,a,b,t-1} \tag{31}$$
$$\forall s \in N_S, b \in N \backslash \{s\}, t \geqslant h(s,b) :$$
$$\delta_{s,b,t} \geqslant \delta_{s,b,t-1} \tag{32}$$
$$\delta_{s,b,t} \leqslant \delta_{s,b,t-1} + \sum_{(a,b) \in E_C} \epsilon_{s,a,b,t-1} \tag{33}$$

All source nodes know about their own information at any time (Equation 26). In the last cycle, all nodes must know about all information (Equation 27). The information s can not reach node v before cycle $h(s,v)$ (Equation 28). According to Equation 15, the Equations 29 and 30 realize $\epsilon_{s,a,b,t-1} = \delta_{s,a,t-1} \wedge \alpha_{a,t-1} \wedge \beta_{b,t-1}$. Therefore, an information s is transmitted from node a to node b iff a knows about s and is scheduled for transmission and b is scheduled for reception (Equation 20). According to Equation 14, the Equations 31 to 33 realize $\delta_{s,b,t} = \delta_{s,b,t-1} \vee \bigvee_{(a,b) \in E_C} \epsilon_{s,a,b,t-1}$. Thus, node b knows about information s in cycle t iff it knew about s in cycle $t-1$, or received the information from any node a in cycle $t-1$.

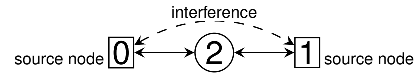
Finally, the following constraints prevent interference and invalid transceiver usage:

$$\forall t \in T, v \in N : \alpha_{v,t} + \beta_{v,t} \leqslant 1 \tag{34}$$
$$\forall t \in T, b \in N : |N| \cdot \beta_{b,t} + \sum_{(a,b) \in E_I} \alpha_{a,t} \leqslant |N| + 1 \tag{35}$$

Equation 34 ensures that no node is scheduled for transmission and reception in the same cycle. Due to Equation 35, a receiver b must not hear signals from multiple transmitters. If b is not scheduled for reception, Equation 35 does not constrain the solution space as the number of potential transmitters can not be larger than $|N|-1$.

After passing the Equations 21 to 35 to the ILP solver, the resulting schedule can be constructed from the α and β variables.

Figure 1 provides a small example network. For $L_{\max} = 4$, the generated ILP formulation comprises 68 variables and 141 constraints. In the ILP solution, 29 variables are set to 1 representing an optimal schedule $(0 \to 2; 1 \to 2; 2 \to 0, 1)$.



$\alpha_{0,0}$	$\alpha_{1,1}$	$\alpha_{2,2}$	
$\beta_{2,0}$	$\beta_{2,1}$	$\beta_{0,2}, \beta_{1,2}$	
γ_0	γ_1	γ_2	
$\delta_{0,0,0}$	$\delta_{0,0,1}, \delta_{0,2,1}$	$\delta_{0,0,2}, \delta_{0,2,2}$	$\delta_{0,0,3}, \delta_{0,2,3}, \delta_{0,1,3}$
$\delta_{1,1,0}$	$\delta_{1,1,1}$	$\delta_{1,1,2}, \delta_{1,2,2}$	$\delta_{1,1,3}, \delta_{1,2,3}, \delta_{1,0,3}$
$\epsilon_{0,0,2,0}$	$\epsilon_{1,1,2,1}$	$\epsilon_{0,2,1,2}, \epsilon_{1,2,0,2}$	

Fig. 1. Example network with $N_S = \{0,1\}$, $N = N_S \cup \{2\}$, $E_C = \{(0,2),(2,0),(1,2),(2,1)\}$, $E_I = E_C \cup \{(0,1),(1,0)\}$ and its ILP solution (variables set to 1)

C. Heuristic Scheduling

As discussed in Section III-B, finding an optimal schedule is NP-complete and the ILP-solvers will thus not be able to find solutions for larger networks in an acceptable time. To this end, we propose an heuristic algorithm to find good (see Section IV for an evaluation), but not necessarily optimal solutions for larger networks.

The main idea of Algorithm 1 is to select the set of transmitting nodes in each scheduling cycle that transfers the maximum amount of new information to their neighboring nodes. If called with an enabled *collect* flag, the algorithm tries to aggregate the information of all source nodes at a central collector node v_{collect} before flooding the whole network. Not all network topologies benefit from this aggregation. Thus the algorithm should be run twice (with and without the *collect* flag) to obtain best results.

In lines 1 to 4, the cycle counter t, the schedule S and the information assignment I are initialized. In lines 6 to 9, the central node v_{collect} for the optional *collect* mode is selected as the node with minimal distance to all source nodes and maximum number of reachable neighbors. Each execution of the while-loop in line 11 generates another schedule cycle, until all information is distributed to all nodes. In line 12, a list T of potential transmitting nodes is selected. To keep the subsequent search-space as small as possible, T is restricted to those nodes carrying the most information (line 14). Due to possible mutual interferences, it is not always the best solution to activate all selected transmitters in the same cycle. Therefore, the subset P of T actually maximizing a performance metric $w_{\max}$ is searched by the loop in line 16. The performance metric varies depending on operating mode (*collect* true or false), and over the course of the search (see below).

In lines 18 to 23, a list of edges E from the transmitters P to appropriate receivers is determined. A node b is an appropriate receiver if it is not included in the transmitters list (line 18), if it is influenced by exactly one transmitter $a \in P$ (line 20) and if it can actually receive additional information from a (lines 21, 22).

Input: Graph model $M = (N, N_S, E_I, E_C, h)$
Input: collect flag
Output: Schedule S

1 $t \coloneqq 0;$
2 $S(v,t) \coloneqq$ *IDLE* $\quad \forall v \in N;$
3 $I(v,t) \coloneqq \varnothing \quad \forall v \in N;$
4 $I(s,t) \coloneqq \{s\} \quad \forall s \in N_S;$
5 **if** *collect* **then**
6 $\quad d \coloneqq \min_{v \in N} \sum_{s \in N_S} h(s,v);$
7 $\quad D \coloneqq \{v \in N : \sum_{s \in N_S} h(s,v) = d\};$
8 $\quad d \coloneqq \max_{a \in D} |\{(a,b) \in E_C\}|;$
9 $\quad v_{\text{collect}} \coloneqq a \in D : |\{(a,b) \in E_C\}| = d;$
10 **end**
11 **while** $\exists v \in N : I(v,t) \neq N_S$ **do**
12 $\quad T \coloneqq \{v \in N : \exists (a,b) \in E_C : I(a,t) \nsubseteq I(b,t)\};$
13 $\quad d \coloneqq \max_{v \in T} |I(v,t)|;$
14 $\quad T \coloneqq \{v \in T : |I(v,t)| = d\};$
15 $\quad w_{\max} \coloneqq -\infty;$
16 $\quad$ **foreach** $P \in 2^T$ **do**
17 $\quad\quad E \coloneqq \varnothing;$
18 $\quad\quad$ **foreach** $b \in N \backslash P$ **do**
19 $\quad\quad\quad D \coloneqq \{a \in P : (a,b) \in E_I\};$
20 $\quad\quad\quad$ **if** $|D| \neq 1$ **then** next;
21 $\quad\quad\quad$ **if** $(a,b) \notin E_C$ **then** next;
22 $\quad\quad\quad$ **if** $I(a,t) \subseteq I(b,t)$ **then** next;
23 $\quad\quad\quad E \coloneqq E \cup M;$
24 $\quad\quad$ **end**
25 $\quad\quad$ **if** *collect* $\wedge\ I(v_{collect}, t) \neq N_S$ **then**
26 $\quad\quad\quad w \coloneqq 0;$
27 $\quad\quad\quad$ **foreach** $s \in N_S$ **do**
28 $\quad\quad\quad\quad D \coloneqq \{v \in N : s \in I(v,t)\} \ \cup$
29 $\quad\quad\quad\quad\quad \{b \in N : \exists (a,b) \in E : s \in I(a,t)\};$
30 $\quad\quad\quad\quad w \coloneqq w - \min_{v \in D} h(v, v_{\text{collect}});$
31 $\quad\quad\quad$ **end**
32 $\quad\quad$ **else**
33 $\quad\quad\quad w \coloneqq \sum_{(a,b) \in E} |I(a,t) \backslash I(b,t)|;$
34 $\quad\quad$ **end**
35 $\quad\quad$ **if** $w > w_{\max}$ **then**
36 $\quad\quad\quad w_{\max} \coloneqq w;$
37 $\quad\quad\quad E_{\max} \coloneqq E;$
38 $\quad\quad$ **end**
39 $\quad$ **end**
40 $\quad I(v, t+1) \coloneqq I(v,t) \quad \forall v \in N;$
41 $\quad$ **foreach** $(a,b) \in E_{\max}$ **do**
42 $\quad\quad S(a,t) \coloneqq$ *TX*;
43 $\quad\quad S(b,t) \coloneqq$ *RX*;
44 $\quad\quad I(b, t+1) \coloneqq I(b,t) \cup I(a,t);$
45 $\quad$ **end**
46 $\quad t \coloneqq t + 1;$
47 $\quad S(v,t) \coloneqq$ *IDLE* $\quad \forall v \in N;$
48 **end**
49 postProcessing(M, S, I);

Algorithm 1: Heuristic scheduling for flooding

In non-greedy mode (or in greedy mode, but with all information already aggregated at the collector node v_{collect}), the algorithm aims to maximize the overall appearance of new (previously unknown) information at receiver nodes. This computation is performed in line 33 for the currently examined subset P. In greedy mode, the heuristic initially drives all information towards the collector node. Thus, until all information has arrived there ($I(v_{\text{collect}}, t) \neq N_S$), the current subset P is rated with regard to the minimal hop distance of each information s to the collector node v_{collect}, summed in the loop at line 27 across all s. Once all information has arrived at v_{collect} ($I(v_{\text{collect}}, t) = N_S$), the quality metric falls back to aiming for maximum information distribution.

The best-so-far solution is maintained in $w_{\max}$ and $E_{\max}$ (line 35). It is that solution that will be used to schedule the nodes in the current cycle and expand the information assignment for the next cycle (lines 40 to 47).

The resulting schedule is further optimized in a post-processing step (line 49) by eliminating redundant data transfers. A data transfer from node a to node b in cycle t is redundant, if it is dominated by a later data transfer from node c to node b in cycle $k > t$, i.e. $\Delta I \coloneqq I(b, t+1) \backslash I(b,t) \subseteq I(c,k)$ and b is not scheduled for transmission between cycles t and k. In this case, $S(b,t)$ is set to *IDLE* and $I(b,i)$ is reduced by ΔI for $t < i \leqslant k$. Afterwards, the transmission $S(a,t)$ may have become superfluous if b (now set to *IDLE*) was its *only* receiver in cycle t. Then, $S(a,t)$ is set to *IDLE*. This may result in a completely idle schedule cycle t, which has to be removed. Furthermore, removing a transmission for node a may make another reception of a redundant. Thus, these steps are performed repeatedly until no more redundant transfers can be found.

IV. Evaluation

The difficulty to find an optimal schedule heavily depends on the network topology. In sparse networks, the number of possible information routes to choose from is smaller than in dense networks. However, in totally connected graphs the optimal solution is straightforward. To evaluate the proposed scheduling algorithms on a large number of topologies, we rely on random test-case generation. For each test-case, a number of $|N|$ nodes is randomly placed in a two dimensional plane of a certain size l^2 by assigning $x, y : N \mapsto [0, l]$. Without loss of generality, the first $|N_S|$ nodes are selected as the source nodes. Two thresholds $d_C, d_I \in \mathbb{R}$ are used to derive the network-topology from the euclidean distance between two nodes $(a,b) \in N^2$:

$$d(a,b) = \sqrt{(x(a) - x(b))^2 + (y(a) - y(b))^2} \tag{36}$$

$$(a,b) \in E_C \Leftrightarrow d(a,b) \leqslant d_C \tag{37}$$

$$(a,b) \in E_I \Leftrightarrow d(a,b) \leqslant d_I \tag{38}$$

The inference and connectivity thresholds are derived from the 2-ray-ground radio propagation model [21]:

$$\frac{P_{RX}}{P_{TX}} = \frac{G_{TX} G_{RX}}{L} \frac{h_{TX}^2 h_{RX}^2}{d(TX, RX)^4} \tag{39}$$

where P_{TX} is the transmitted signal power, P_{RX} is the received signal power, $d(TX, RX)$ is the distance between sender and receiver, h_{TX} and h_{RX} are the vertical height of the transmitter and receiver antennas over ground. The system loss factor L and the antenna gains G_{TX}, G_{RX} are usually disregarded.

As a practical example, the TI CC2530 radio system-on-chip [22] actually used in our SHM sensor node [2] provides a maximum P_{TX} of 4.5 dBm (2.82 mW) and may successfully receive -82 dBm (6.31 pW) signals while being subject to another -85 dBm (3.16 pW) noise signal (co-channel rejection). With an assumed antenna height of 0.24 m, the connectivity and interference ranges are calculated as

$$d_C = 0.24 \text{ m} \cdot \sqrt[4]{\frac{2.82 \text{ mW}}{6.31 \text{ pW}}} = 35 \text{ m} \quad (40)$$

$$d_I = 0.24 \text{ m} \cdot \sqrt[4]{\frac{2.82 \text{ mW}}{3.16 \text{ pW}}} = 41 \text{ m} \quad (41)$$

With these fixed settings, the density of the network topology can be adjusted by the number of network nodes and the size l^2 of the deployment area.

To evaluate the efficiency of the proposed optimal and heuristic scheduling algorithm, the resulting cost for various network configurations is compared against traditional and advanced flooding algorithms. These reference implementations share some common characteristics. In each cycle, a set of potential forwarding nodes $F \subseteq N$ is selected by a protocol specific mechanism (see below). Out of these, a subset $T \subseteq F$ is selected by CSMA/CA and a hidden terminal detection, i.e. $\nexists a, \hat{a} \in T : (a, \hat{a}) \in E_I \vee (\exists b \in N : (a, b) \in E_C \wedge (\hat{a}, b) \in E_I)$. All non-transmitting nodes in the interference range of any transmitter are counted as receivers $R = b \in N \backslash T : \exists a \in T : (a, b) \in E_I\}$. All other nodes will turn off their radio after clear channel assessment and thus do not consume a considerable amount of energy when compared to actually receiving data.

The main difference between the various reference implementations is the selection of the potential forwarding nodes F. In blind flooding, all nodes that have received, but not yet forwarded an information are included in F. This does not require any knowledge about the network topology. In contrast, the self-pruning (*sp*) [11], dominant-pruning (*dp*) [11], partial dominant-pruning (*pdp*) [12], total dominant-pruning (*tdp*) [12] and the history-based 2-hop dominant-pruning (*h2dp*) [13] try to reduce F based on knowledge about the one- or two-hop neighborhood of each node. Please refer to the original work for further details.

For this evaluation, CPLEX 12.5.1 is used as ILP solver running up to 8 threads. The ILP formulation is passed to CPLEX in the LP file format. The solver is supplied with an upper bound for the objective function derived from a preceding run of the scheduling heuristic. All computations are executed on a compute server equipped with 128 GB RAM and 32 AMD Opteron 6134 CPUs running at 2.3 GHz.

For networks of $|N| = 20$ nodes, 10 network configurations are considered by varying the number of source nodes $N_S \in \{1, \dots, 5\}$ and the size of the deployment area $l^2 \in \{100 \times 100 \text{ m}^2, 150 \times 150 \text{ m}^2\}$. For each configuration, 50 networks are generated at random (within the given bounds). For each network, the various algorithms (flooding, pruning, heuristic and ILP) are applied to generate a schedule. To simplify comparison, the schedule costs (Equation 12) of the enhanced algorithms are normalized to the outcome of the naive blind flooding. The resulting relative costs are averaged over the 50 random test-cases per network configuration.

Figure 2 shows the results for the larger deployment area. For a single source node, the pruning algorithms reduce the communication costs by up to 35 % compared to the blind flooding baseline costs. These pruning algorithms assume a certain preferred direction of information expanding from a single source throughout the network. With a growing number of source nodes (as required for SHM), this assumption is no longer valid, resulting in the degraded improvement of less than 30 % cost reduction for five source nodes. In contrast, the heuristic and the ILP scheduler are designed to manage multiple source nodes interfering with each other. Thus, their performance advantage over blind flooding increases with the number of source nodes up to 83 % for five source nodes.

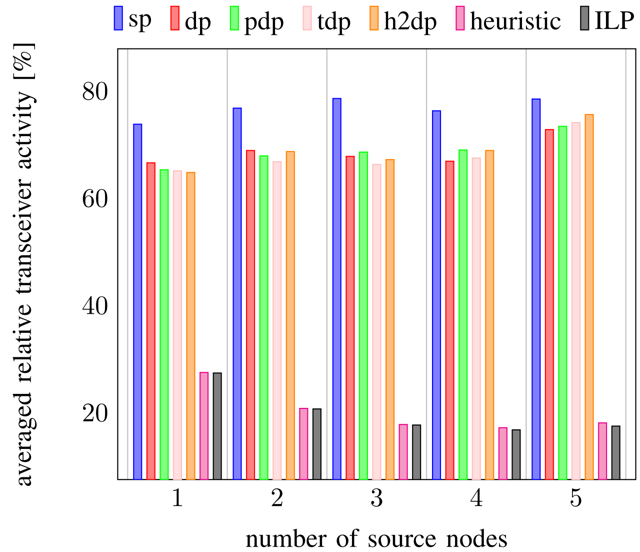


Fig. 2. Averaged scheduling costs relative to blind flooding for 20 nodes in an 150×150 m^2 deployment area

Similar observations can be made in Figure 3. Here, the smaller deployment area results in denser networks. The number of redundant transmissions and receptions performed in blind flooding thus increases and the relative cost of all other algorithms decreases. Furthermore, the knowledge of the

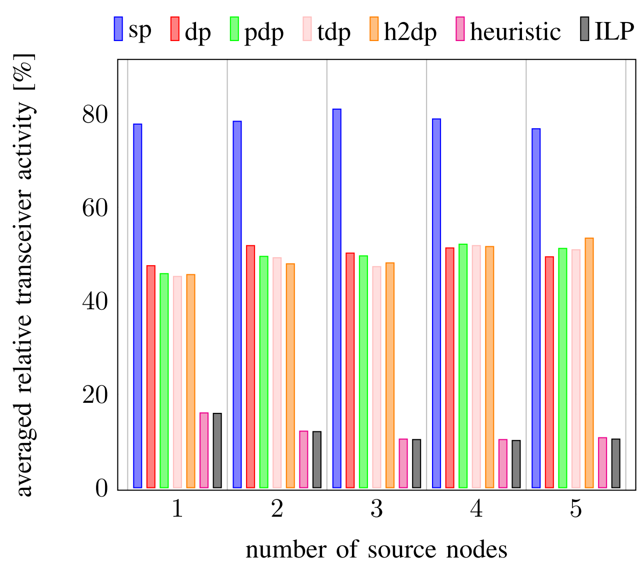


Fig. 3. Averaged scheduling costs relative to blind flooding for 20 nodes in an 100×100 m^2 deployment area

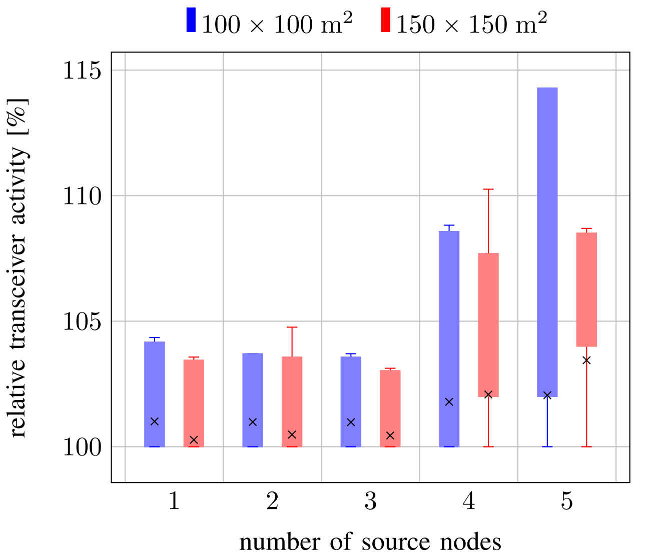


Fig. 4. Heuristic scheduling costs relative to optimal scheduling costs: 0 % to 100 % percentile ( ), 60 % to 95 % percentile ( ), average (×)

pruning algorithms about the two-hop neighborhood becomes more valuable as it covers a larger part of the network. The pruning algorithms achieve up to 55 % cost reduction for a single source node while the heuristic and optimal schedules achieve up to 90 % cost reduction for five source nodes.

Figure 4 details the heuristic schedule costs normalized to the optimal schedule costs (the number of transmissions/receptions). For networks with up to three source nodes, the average heuristic results are less than 1 % worse than the optimal results. Furthermore, the heuristic found an optimal solution in at least 60 % of the generated test cases with less than four source nodes. With an increasing number of source nodes, the gap between heuristic and ILP becomes larger, but does not exceed 4 % on average for five source nodes.

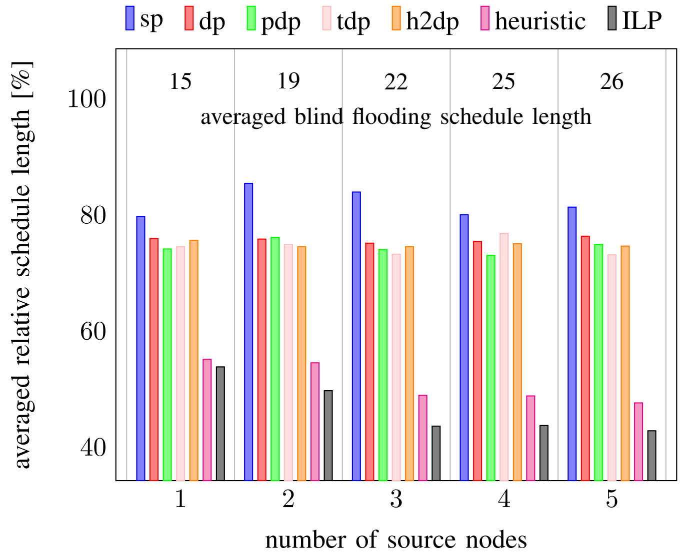


Fig. 6. Averaged schedule length relative to blind flooding for 20 nodes in an 150 × 150 m^2 deployment area

This slight quality loss is the price for the highly accelerated computation, detailed in Figure 5. E.g., it can be seen that for four sources in the smaller area, the heuristic (run in two passes with *collect* both enabled and disabled) executes in just 0.1 % of the time of the optimal ILP solution. A speedup of more than 1000x can be observed for networks with five source nodes. Note that all execution times are measured as pure wall-clock time, not counting the eight-core processing of the ILP solver as separate execution times.

The length of a schedule is the secondary optimization target defined by Equation 13. Figures 6 and 7 detail the average length of the schedules determined by the considered flooding schemes. As for the transceiver activity, the proposed ILP and heurisitcs clearly outperform the conventional pruning

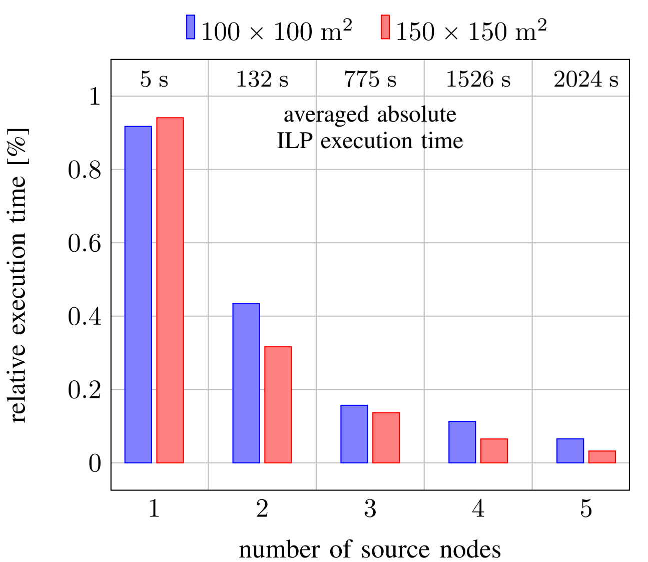


Fig. 5. Averaged runtime of heuristic relative to runtime of ILP-solver (both executed on the same compute server)

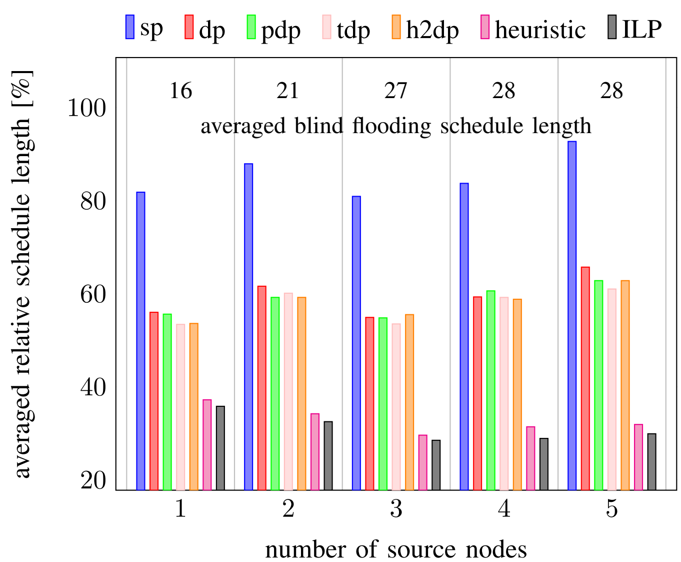


Fig. 7. Averaged schedule length relative to blind flooding for 20 nodes in an 100 × 100 m^2 deployment area

algorithms especially for an increasing number of source nodes and denser networks.

V. Conclusion and future work

In this work, the specific characteristics of a wirelessly distributed structural health monitoring application were analyzed to derive an energy-efficient routing strategy for information exchange between the sensor nodes. To support the simultaneous flooding of data generated at multiple source nodes, a scheduling problem was defined whose solution assigns a minimum amount of radio activity to each sensor node for consecutive transmission cycles. An integer linear program and a heuristic algorithm were proposed to solve the scheduling problem. Compared to conventional flooding protocols, the amount of required radio activity can be reduced by up to 90 $\%$. For the network configurations considered, the approximation error of the heuristic is restricted to just 4 $\%$ over the optimal solution, while providing a speedup of more than 1000x over the ILP solver.

Future improvements of the scheduling algorithm should utilize the transmission power regulation and multi-channel capabilities of modern radio transceivers to mitigate interference-related limitations in dense networks. Furthermore, instead of just reducing the overall network activity, the transmission load should be equally distributed to avoid early failure of heavily loaded relay nodes.

Acknowledgment

This work is part of the LOEWE program funded by the Hessian Ministry of Higher Education, Research and Arts.

References

[1] H. A. Cole, *On-line failure detection and damping measurement of aerospace structures by random decrement signatures*. Nielsen Engineering and Research Inc., 1973.

[2] A. Engel, B. Liebig, and A. Koch, "Energy-efficient heterogeneous reconfigurable sensor node for distributed structural health monitoring," in *Conference on Design and Architectures for Signal and Image Processing (DASIP)*, D. A. Morawiec and J. Hinderscheit, Eds. Electronic Chips & Systems design Initiative, 2012.

[3] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 39–49.

[4] L. Gheorghe, R. Rughinis, and N. Tapus, "Fault-tolerant flooding time synchronization protocol for wireless sensor networks," in *Proc. Sixth Int Networking and Services (ICNS) Conf*, 2010, pp. 143–149.

[5] N. Xu, X. Zhang, Q. Wang, J. Liang, G. Pan, and M. Zhang, "An improved flooding time synchronization protocol for industrial wireless networks," in *Proc. Int. Conf. Embedded Software and Systems ICESS '09*, 2009, pp. 524–529.

[6] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, Feb 1999, pp. 90–100.

[7] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 151–162.

[8] J. Arango, M. Degermark, and A. Efrat, "An efficient flooding algorithm for ad hoc networks," in *Proceedings of the Second Workshop on Modeling and Optimizations in Mobile Ad Hoc and Wireless Networks (WiOpt 2004)*, 2004.

[9] H.-C. Jeong, K.-D. Kwon, and Y. Yoo, "Cross-layer counter-based flooding without location information in wireless sensor networks," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, april 2010, pp. 840 –845.

[10] M. Sheng, J. Li, and Y. Shi, "Relative degree adaptive flooding broadcast algorithm for ad hoc networks," *Broadcasting, IEEE Transactions on*, vol. 51, no. 2, pp. 216–222, June 2005.

[11] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Communications*, vol. 24, no. 3-4, pp. 353 – 363, 2001.

[12] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 111–122, 2002.

[13] S. Agathos and E. Papapetrou, "Efficient broadcasting using packet history in mobile ad hoc networks," *Communications, IET*, vol. 5, no. 15, pp. 2196 –2205, 14 2011.

[14] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 4204–4213, November 2012.

[15] M. Alnuaimi, K. Shuaib, K. Nuaimi, and M. Abdel-Hafez, "Performance analysis of clustering protocols in wsn," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, April 2013, pp. 1–6.

[16] N. Ahmed, S. Kanhere, and S. Jha, "Mitigating the effect of interference in wireless sensor networks," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, Oct 2010, pp. 160–167.

[17] T. De Pauw, S. Verstichel, B. Volckaert, F. De Turck, and V. Ongenae, "Resource-aware scheduling of distributed ontological reasoning tasks in wireless sensor networks," in *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*, June 2010, pp. 131–137.

[18] T. D. Hoa and D.-S. Kim, "Minimum latency and energy efficiency routing with lossy link awareness in wireless sensor networks," in *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, May 2012, pp. 75–78.

[19] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, pp. 345–, Jun. 1962.

[20] V. Dheap, M. Munawar, S. Naik, and P. Ward, "Parameterized neighborhood based flooding for ad hoc wireless networks," in *Military Communications Conference, 2003. MILCOM '03. 2003 IEEE*, vol. 2, oct. 2003, pp. 1048 – 1053 Vol.2.

[21] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[22] *CC253x System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*. [Online]. Available: http://www.ti.com/ww/de/analog/cc2530