Desynchronization in Multi-Hop Topologies: A Challenge

Clemens Mühlberger
Chair of Computer Science V
Department of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg
Email: muehlberger@informatik.uni-wuerzburg.de

***Abstract*—The biologically inspired primitive *desynchronization* was successfully implemented and tested within single-hop topologies in form of the self-organizing TDMA protocol DESYNC for wireless sensor networks (WSNs). Two extensions of that MAC protocol for multi-hop topologies have been discussed, but either the extended protocol is not all-purpose but specified for just a specific subset of multi-hop topologies, or each node has to broadcast all of its neighboring information at every single packet, which enlarges the packet size and thus consumes additional energy and bandwidth. One reason for this limitation, and packet overhead respectively, is the hidden terminal problem which is inherent in all multi-hop topologies. In this paper we compare the characteristics of single-hop and multi-hop topologies – with respect to the primitive of desynchronization. We will further analyze one special multi-hop topology in detail, which not only shows the complexity of a multi-hop desynchronization, but also provides new opportunities to support all sorts of multi-hop topologies with reduced overhead for the neighboring information.**

I. DESYNC – A BRIEF INTRODUCTION

In 2006, Degesys et al. [1] first published DESYNC, a self-organized TDMA protocol for WSNs [2]. This MAC protocol follows the biologically inspired primitive of *desynchronization* [3] to achieve an equidistant distribution of participating oscillators, e.g. periodically transmitting sensor nodes. As logical opposite of synchronization, desynchronization in general means that each device tries to perform its (periodic) tasks as far away as possible from all other affected devices. Within the scope of WSN, desynchronization describes the temporally equidistant transmission of radio packets.

So, the (idealized) network is composed of a set of nodes N. All communication links are symmetric, and each node $i \in N$ oscillates at an identical frequency. The phase ϕ_i of a node i denotes the elapsed time since its last transmission relative to its current period. When a node finishes its period, it broadcasts a so called *firing packet* and immediately resets its phase, i.e. if the node is desynchronized already, it will broadcast its next firing packet exactly one period after the start of the current transmission. Each one-hop neighbor of the currently transmitting node receives this firing packet (if there was no collision), and logs the sender's ID together with its local time of reception to calculate its individual phase shift towards the sender.

Each node i can determine by itself a more appropriate firing phase (according to an equidistant distribution), based on its individual knowledge of the phases of its so called *phase neighbors*:

- *previous phase neighbor* (predecessor) $p(i) \in N \setminus \{i\}$ broadcasts its firing packet (from i's point of view) just before node i,
- *successive phase neighbor* (successor) $s(i) \in N \setminus \{i\}$ broadcasts its firing packet (from i's point of view) just after node i.[1]

With it, node i can now calculate the midpoint of its phase neighbors, and finally estimate its new firing phase ϕ'_i as

$$\phi'_i = (1-\alpha) \cdot \phi_i + \alpha \cdot \frac{\phi_{s(i)} + \phi_{p(i)}}{2}, \qquad (1)$$

where ϕ_i denotes the last phase of node i, and the *jump size parameter* $\alpha \in (0.0, 1.0]$ regulates, how fast the node moves toward the assumed midpoint of its phase neighbors. Convergence to the stable state of *desynchrony* is achieved, if each node has the same distance to its phase neighbors (cf. Fig. 1) and thus the transmission times do not change anymore - unless the system changes.

II. EXTENSIONS FOR MULTI-HOP TOPOLOGIES

The handling of single-hop topologies is quite simple[2], because every node can directly communicate with each other. On the other hand, the so called *hidden terminal problem* inheres in multi-hop topologies, which complicates collision-free communication. This section presents two yet available extensions of the DESYNC protocol for multi-hop topologies: M-DESYNC and EXTENDED-DESYNC.

A. The M-DESYNC *Approach*

The M-DESYNC algorithm [4] for (single-hop and) acyclic multi-hop topologies is mainly based on the *local max degree* of each node, i.e. the maximum degree among a node i and its one-hop neighbors $N_1(i)$. Here, the *degree* of a node equals the cardinality of its one-hop neighborhood $|N_1(i)|$.

This algorithm requires an initial phase, at which each node exchanges its degree with all its one-hop neighbors to determine its local max degree. This phase may take quite long, because the algorithm uses just a random back-off protocol

[1] Besides, within a connected topology of size $|N| = 2$ both phase neighbors are the very same node $p(i) = s(i)$.

[2] This statement will be confirmed in detail in Section III-A.

without further optimization. After this preliminary phase, every node requires local max degree plus an additional time slots for a collision-free communication within its interference range. At the next step, each node just has to occupy its individual time slot. For this slot selection, a modulo pre-coloring as well as a priority-based strategy are suggested instead of just a random competition.

Using the local max degree, the minimum number of required time slots per period for each node was proven. However, the M-DESYNC approach is not very flexible to topology changes due to the lengthy exchange phase, but even not applicable for cyclic topologies, which will be demonstrated in Section IV.

B. *The* EXTENDED-DESYNC *Approach*

To solve the hidden terminal problem at multi-hop topologies, each node needs knowledge about its two-hop neighborhood. Therefore, for the EXTENDED-DESYNC algorithm [5] each node broadcasts its (currently known) one-hop neighbors in combination with their relative phase shifts, always corresponding to the point of view of the current sender. With it, each node gets to know its two-hop neighborhood in addition. The relative phase shifts may become stale, because phase changes of two-hop neighbors emerge after two periods. But this delayed information becomes more accurate and reliable with each subsequent period and thus just slows down convergence rate a little.

Here, no initial exchange phase is required. Instead, a new joining node just has to listen for a few periods to make itself familiar with its local topological conditions. Afterwards, it can interact immediately with its well-known one-hop neighbors and thus be integrated into the network easily.

Hence, the EXTENDED-DESYNC approach is very flexible and reacts quite fast on topology changes. It thus scales well with network size, but exhibits a large packet overhead. Every node has to broadcast its whole one-hope neighborhood, which takes bandwidth and energy for algorithmic purposes, especially in dense networks and at nodes with a high degree.

III. COMPARISON

Before we oppose the characteristics of desynchronization in single-hop topologies to multi-hop topologies, we specify some general assumptions. For all nodes we assume that their communication range equals their interference range. Next, the network is build upon symmetrical links, i.e. communication between two nodes always works bidirectional. And finally, the network consists of $|N|$ nodes, where every node owns a unique identifier as well as a finite buffer for storing (incoming) packets. But each node has just one transceiver in half-duplex mode, i.e. no node can transmit and receive packets simultaneously.

A. *Single-Hop Topology*

Within a single-hop topology, every node is able to interact with each other, hidden nodes do not exist. Thus, everyone knows everyone, each node has knowledge about the whole network. This enables a fast and easy self-adaption on start-up and topological modifications. Furthermore, all nodes share the common communication medium, that means for a desynchronized TDMA protocol there are exactly $|N|$ slots required at every period.

In single-hop topologies, a packet transmission is considered to be successful, if there are no other packet transmissions at the same time, i.e. the shared communication medium is assumed to be error-free. Thus, at every point in time just one single node is allowed to send a radio packet. In terms of desynchronization, the stable state (*desynchrony*) is reached, if each node transmits its radio packets temporally equidistant to its phase neighbors. With it, we can draw the following conclusions for desynchronization in single-hop topologies.

- S1 All nodes within a single-hop topology have the very same degree $|N| - 1$.
- S2 If node i is phase neighbor (w.l.o.g. predecessor $p(k) = i$) of another node $k \in N \setminus \{i\}$, then node k in return is the corresponding phase neighbor (here, k is successor $s(i) = k$) of i.
- S3 Every node i with degree ≥ 1 has at most one predecessor $p(i) = j$ and at most one successor $s(i) = k$. Following from S2, node i will be the corresponding phase neighbor (successor, and predecessor respectively) of its phase neighbors j and k in return.
- S4 Using S3, every node i with degree ≥ 1 is always predecessor $p(j) = i$ and successor $s(k) = i$ of nodes $j, k \in N \setminus \{i\}$.
- S5 Due to S2 and according to equation 1 (every node tries to maximize the temporal distance to both its phase neighbors), all nodes are distributed equidistant along the unified period. In other words, the temporal distances between each pair of subsequently firing nodes are identical.
- S6 The initial start-up order determines, when a node will (re)join or leave the network, mainly affects the order of firings.

B. *Multi-Hop Topology*

Within a multi-hop but connected topology, there exists at least one node i which is not able to interact with every node $j \in N \setminus \{i\}$ of the network in a direct way. For this reason, there exists at least one such "hidden" node $h \in N \setminus \{i\}$ outside the communication range of node i. Hence, every node has just a local view and thus limited knowledge about the whole network. Although all nodes share the same communication medium. Indeed, it will be possible now, that two or more nodes can transmit their packets simultaneously within the same time slot without interference. Therefrom, for a desynchronized TDMA protocol at most $|N|$ transmission slots are required to support a collision-free communication within the network.

This is the reason, why a packet transmission is considered to be successful, if there are no other packet transmissions at the same time within the interference area of the sender and all of its potential receivers. Thus, more than one node may be allowed to transmit a radio packet concurrently. Desynchrony is reached here, if each node transmits its packets temporally equidistant to its phase neighbors without interference with any other node of the network. For desynchronization in multi-hop topologies the following phenomena can be observed:

M1 The degree of the nodes within a multi-hop topology now may diverge, but is at most $|N|-1$.

M2 Due to the nodes' different degrees in multi-hop topologies (cf. M1), $s(i) = j \Leftrightarrow i = p(j)$ as well as $p(i) = k \Leftrightarrow i = s(k)$ (cf. S2) do not hold any longer for a node i and its phase neighbors $j, k \in N \setminus \{i\}$. For example, node i is predecessor $p(j) = i$ of node j, but in turn node j is not i's successor $s(i) \neq j$, but instead node $k \neq j$ is now successor $s(i) = k$ of i.

M3 As for single-hop topologies (cf. S3), every node i with degree ≥ 1 has at most one predecessor and at most one successor. But now, in multi-hop topologies there can be a set of nodes $S = \{x|s(x) = i\} \subseteq N \setminus \{i\}$ with $|S| \geq 2$ sharing the same successor i. Analogously, there can be a set of nodes $P = \{x|p(x) = i\} \subseteq N \setminus \{i\}$ with $|P| \geq 2$ sharing the same predecessor i. Changing the firing time of such a multiple successor (and predecessor respectively) will affect at once the time of firing of every node $x \in S$, and $x \in P$ respectively, which initiates the recalculation of x's next firings and thus slows down convergence rate.

M4 In single-hop topologies (cf. S4), every node i with degree ≥ 1 is always predecessor and successor at once. But due to observation M2, multi-hop topologies can contain nodes i with degree ≥ 1, which are either just predecessors, or just successors, or none of another node. That means, changing the time of firing (within a specific interval) of such a node i does not initiate recalculation of many (if any) time of firings, but maybe contradicts the primitive of desynchronization (cf. Sec. I).

M5 The observation M2 of not-being phase neighbor of node's phase neighbors, linked to the availability of different degrees in multi-hop topologies (cf. M1), leads to non-identical temporal distances. That is, each node tries to maximize its temporal distance towards its phase neighbors (cf. S5), but within multi-hop topologies the temporal distance between each pair of subsequently firing nodes are not identical anymore.

M6 As for single-hop topologies (cf. S6), the initial start-up order not only mainly affects the order of firings, but also whether a node becomes phase neighbor of other nodes – or not.

The nodes' temporal order and the phase neighbors of a node within a multi-hop topology strongly depend on the initial start-up order. Because of this large configuration space and observations M1 – M6, the proof of convergence for any kind of multi-hop topology is quite difficult – especially from an arbitrary initial start-up order into the stable state of desynchrony. To get a first impression of the difficulty of such a proof see [6]. Such a proof will be object for our future research. Therefore, we will exemplify in the next section, why cyclic topologies are not covered by the M-DESYNC approach and which information could be sufficient to get such a multi-hop topology desynchronized – always depending on the initial start-up order.

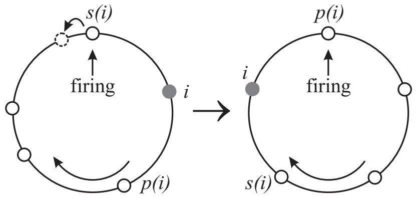

Fig. 1. Snapshots of the process of desynchronization from a global point of view.

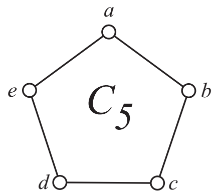

Fig. 2. Diagram of the examined topology C_5.

IV. MULTI-HOP EXAMPLE

In this section we analyze the desynchronization of a 2-regular Hamiltonian cycle C_5 of size $|N| = 5$, i.e. there are five nodes $a, \ldots, e \in C_5$, all have degree two, according to Fig. 2. For a collision-free communication within C_5, there are five time slots required: If for example node a transmits a packet, neither its one-hop neighbors e and b, nor its two-hop neighbors d and c are allowed to transmit any packet at the same time. Due to the symmetry properties, this holds for all other nodes of topology C_5. Thus, each node claims one of totally five slots.

Using the local max degree method of the M-DESYNC approach does not lead to a correct and collision-free time slot assignment by the following reasons. First, the degree of each node is two, just as any local max degree. With an additional slot for itself, each node schedules three slots in total. But for a collision-free communication within topology C_5, at least five instead of just three disjoint slots are required (see above). For this reason, the M-DESYNC algorithm is non-applicable for cyclic multi-hop topologies.

In contrast, the EXTENDED-DESYNC algorithm schedules five time slots according to the five nodes. Because each node transmits its currently known one-hop neighborhood, each node also gets to know its two-hop neighborhood. With this knowledge, each node can take care of its one-hop – and more important – of its two-hop neighbors. Due to the symmetry properties of this topology C_5, still every node has two one-hop and also two two-hop neighbors and thus schedules five slots in total. Therefore, each node desynchronizes itself according to its phase neighbors, which in turn depend on the initial network configuration (cf. M6).

To reduce the packet overhead which has to be propagated at the EXTENDED-DESYNC algorithm, we will go step-by-step through one (of many) possible desynchronization procedures for the formation of this multi-hop topology C_5, using the following packet format $[i_{id}, p(i)_{id}, s(i)_{id}, |N_1(i)|]$ which contains the following data[3]:

[3]The relative phase shift of the phase neighbors are also transmitted but not shown here to cut short the example.

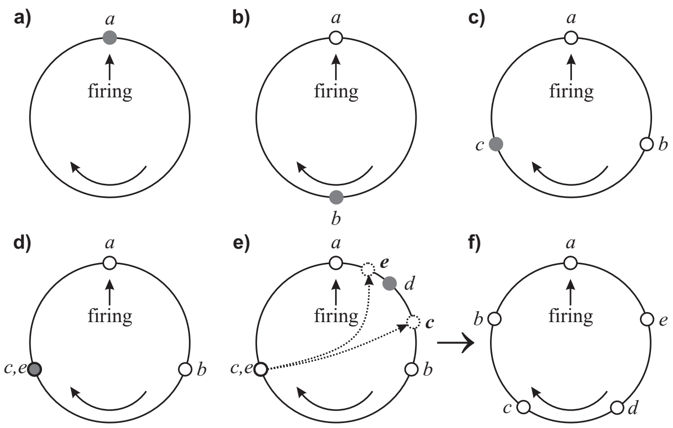

Fig. 3. Step-by-step desynchronization for topology C_5 with less overhead from a global point of view.

- i_{id}: the ID of the sender named i,
- $p(i)_{id}$: the ID[4] of the current predecessor $p(i)$ of the transmitting node i,
- $s(i)_{id}$: the ID[4]of the current successor $s(i)$ of the transmitting node i,
- $|N_1(i)|$: the current number of one-hop neighbors of the transmitting node i.

The desynchronization process for the start-up order $a \to b \to c \to e \to d$ runs as follows (cf. Fig. 3).

a) First, node a starts up and listens, but receives not a single packet. Thus after a while, node a builds a new network and broadcasts $[a,,,0]$ after every period.

b) Next, node b starts up, listens and receives $[a,,,0]$ from node a. Thus, node b broadcasts $[b,a,a,1]$. Node a in return receives b's packet and from now on broadcasts $[a,b,b,1]$ accordingly.

c) Node c wants to join the network and listens, but just receives b's broadcast. With it, node c in return broadcasts $[c,\underline{a},b,1]$. This causes node b and – with some delay – node a to adjust their time of firings, as well as the content of their packets to $[b,c,a,2]$, and $[a,b,\underline{c},1]$ respectively.

d) Later, node e starts up and listens. It receives just the broadcast of node a. Using this information, node e chooses the same time of firing as node c and broadcasts $[e,a,\underline{b},1]$. This is possible, because node e and c are currently more than two hops away and thus do not interfere with each other. The broadcast of node e causes a just to update its packet content into $[a,b,e,2]$.

e) Last but not least, node d tries to join the network and listens. Because both its one-hop neighbors e and c transmit their firing packets at the very same time, d receives just corrupt data – if any, and thus just broadcasts $[d,,,0]$. Assuming that this broadcast does not collide with any other packet, i.e. the time of firing of node d does not overlap with any time slot of the remaining network, and in this example temporally lies in between node a and b (cf. Fig. 3.d), its one-hop neighbors e and c receive d's broadcast. But because node d states not to know any neighbors (especially not e and c), each receiver concludes to cause a collision. With it, node e, and c respectively, changes its time of firing in such a way, to be in between the joining node d and its one-hop neighbor a, and b respectively. The firing packets of e and c also change to $[e,d,a,2]$, and $[c,b,d,2]$ respectively. These changes cause the corresponding neighbors to adjust their time of firing and content of their firing packets into $[a,e,b,2]$, $[b,a,c,2]$, and $[d,c,e,2]$.

f) Finally, after the nodes rearranged themselves along the period, each node holds the same distance to its both phase (and one-hop) neighbors. Remarkably, all nodes are temporally equidistant distributed, although this is not a single-hop topology.

[4]If the identifier is underlined, the corresponding node is a two-hop neighbor of sender i.

V. Conclusion and Outlook

In this paper we initially introduced the primitive of desynchronization as TDMA protocol for WSNs. We then analyzed the difference between desynchronization in single-hop and multi-hop topologies. The detailed example of desynchronization in a specific cyclic multi-hop topology on the one hand presented a collision-free slot assignment using reduced firing data. But on the other hand, this idealized example leaves many question open, e.g. what, if the transmission of node d always interferes with a's broadcast, thus d never will be received? Will the system converge for any other start-up order, too? Is the reduced firing data sufficient or too much limiting for other multi-hop topologies?

These questions are subject to our future research. Also, we plan to strengthen and to generalize the approach of reduced firing data. This may help us to prove the convergence of our reduced data approach for arbitrary multi-hop topologies in any start-up order. To get our approach fit for practice, we plan to implement it within a real-world testbed: for instance, realistic scenarios do not have symmetrical links in any case.

References

[1] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks," Havard University, Cambridge, MA, USA, Tech. Rep. TR-18-06, Dec. 2006.
[2] A. Patel, J. Degesys, and R. Nagpal, "Desynchronization: The Theory of Self-Organizing Algorithms for Round-Robin Scheduling," in *SASO*, Cambridge, MA, USA, 2007, pp. 87–96.
[3] R. E. Mirollo and S. H. Strogatz, "Synchronization of Pulse-Coupled Biological Oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.
[4] H. Kang and J. L. Wong, "A Localized Multi-Hop Desynchronization Algorithm for Wireless Sensor Networks," in *INFOCOM*, 2009, pp. 2906–2910.
[5] C. Mühlberger and R. Kolla, "Extended Desynchronization for Multi-Hop Topologies," Institut für Informatik, Universität Würzburg, Tech. Rep. 460, Jul. 2009.
[6] J. Degesys and R. Nagpal, "Towards Desynchronization of Multi-hop Topologies," in *SASO*, Venice, Italy, 2008, pp. 129–138.