Energetic and Temporal Analysis of a Desynchronized TDMA Protocol for WSNs

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

***Abstract*—These days, wireless radio communication of sensor nodes is still very power consuming. Thus lots of MAC protocols yet exist to manage the collision free access to the common transmission medium with respect to energy consumption. In this paper, we focus on the biologically inspired and self-organized TDMA protocol DESYNC. We analyze its potential for energy-saving, and network latency respectively. We could identify some parameters, which may help network designers to adjust the DESYNC protocol according to their preferences, i.e. to save more energy or to get a lower latency.**

I. Motivation

One characteristic feature of Wireless Sensor Networks (WSNs) is the shared transmission medium of interacting sensor nodes. Concurrent assignment of common radio channels might cause loss of data due to packet collisions which require retransmission and consume additional energy. That's why the access of each node to the medium has to be coordinated carefully. Several Medium Access Control (MAC) protocols for WSNs already exist, amongst others a couple of Time Division Multiple Access (TDMA) protocols. They divide the radio channel into *time slots* offering collision free medium access for a specific node, like Z-MAC [1], TRAMA [2] or HashSlot [3]. Here we analyze the decentralized but self-organized DESYNC protocol [4], [5] with respect to energy savings and network latency.

In the next section we briefly introduce the decentralized TDMA protocol DESYNC and its underlying paradigm of desynchronization. In Section III we identify the potential for energy-savings, whereas Section IV discusses the emanating changes in network latency. Section V closes this paper with a short conclusion and an outlook to further research.

II. Introduction to the DESYNC Protocol

The biologically inspired paradigm of desynchronization [6] denotes the equidistant distribution in time of *oscillators*, for example periodically transmitting sensor nodes. Based upon this, Degesys et al. [4] developed the DESYNC protocol, a self-organized TDMA protocol for single-hop topologies. Because real-world deployments usually contain multi-hop topologies, an extended version of the DESYNC protocol is subject to current research (cf. [5]).

First of all, each element of the set N of nodes has a unique identifier i and oscillates at an identical frequency ω within the common period $T = \frac{1}{\omega}$. The period T must be long enough to provide at least one time slot for each of the n participating nodes, e.g. for single-hop topologies $n = |N|$, for multi-hop topologies n equals the cardinality of the maximum clique of two-hop neighbors. Next, the communication links are symmetrical and each node uses Carrier Sense (CS) just before transmission to avoid collisions in the first place.

To fulfill the paradigm of desynchronization, i.e. to spread out the time slots of all participating nodes equidistantly, each node i of the WSN tries to maximize the time lag relative to its neighbors. Therefore, the phase $\phi_i \in [0.0, 1.0]$ of a node i denotes the elapsed time since its last transmission normalized to T, e.g. $\phi_3 = 0.7$ means, that node 3 has already finished 70 % of its current period. When node i finishes its period, i.e. $\phi_i = 1.0$, it broadcasts a so called *firing packet* and immediately resets its phase to $\phi_i = 0.0$. The column vector $\overrightarrow{\phi} = \left[\phi_1 \cdots \phi_i \cdots \phi_{|N|}\right]^T$ describes the global system state, i.e. the phases of all nodes.

Two nodes are of special interest for node i: the *previous phase neighbor* $p(i)$ broadcasts its firing packet just before, whereas the *successive phase neighbor* $s(i)$ broadcasts its firing packet just after node i. Hence, node i can calculate the midpoint of its phase neighbors as

$$\mathrm{mid}(\phi_{s(i)}, \phi_{p(i)}) = \frac{\phi_{p(i)} + \phi_{s(i)}}{2}$$

and finally estimate its new phase ϕ_i' unassisted by itself as

$$\phi_i' = (1 - \alpha) \cdot \phi_i + \alpha \cdot \mathrm{mid}(\phi_{s(i)}, \phi_{p(i)}).$$

The jump size parameter $\alpha \in (0.0, 1.0)$[1] regulates how fast a node moves towards the assumed midpoint of its phase neighbors.

The stable state, when each node has the same temporal distance to its phase neighbors and thus the times of firing do not change anymore (unless the system changes), is called *desynchrony*. The convergence to desynchrony for single-hop topologies was proved in [4]. Figure 1 exemplifies the progress of desynchronization for a single-hop topology consisting of five sensor nodes.

[1] If $\alpha = 0.0$, there's no movement at all, and, according to [7], $\alpha = 1.0$ forces straight movement onto the midpoint under unstable emergence of new configurations. Thus, a reliable value would be $\alpha \approx 0.9$.

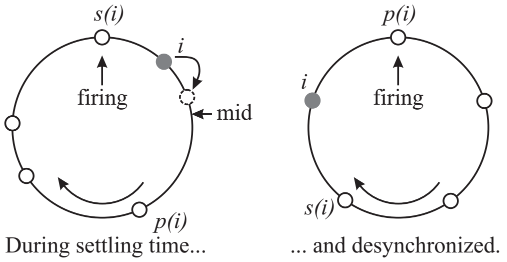

Fig. 1. Snapshots of the desynchronization progress

For a successful operation in real-world deployments, the DESYNC protocol requires an extension for multi-hop topologies as well as further improvements, e.g. back-off algorithms for concurrent start-ups of nearby nodes, or unreliable links. But space does not permit a discussion of that here, that's why we just analyze potential energy savings and network latency within this paper.

III. Energy

Still, the energy consumption of radio transceivers used at sensor nodes is much higher than that of current microcontrollers. Thus, to save much energy – especially at periodically transmitting sensor nodes – the radio controller has to be switched off as often and as long as possible. Since the comprehensive and constant period T depends on the maximum number n of supported nodes, we divide T into n *frames* $F(i)$ of equal size f, i.e. $f = |F(i)|$ for any $i \in \{1, \ldots, n\}$. Similar to other protocols like LMAC [8] or Crankshaft [9], each frame $F(i)$ again is subdivided into k *slots* $F(i,j)$, where $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$. The first slot $F(i,1)$ is reserved for the firing packet of node i (cf. Section II), whereas the remaining slots for $k \geq 2$ can be used for further data transmissions, if desired. Please note, for $k = 1$ the length of a frame equals the length of its single firing slot, i.e. $f = |F(i,1)|$.

To avoid collisions when nodes (re)join the network dynamically, and to compensate potential but individual clock drifts or other hardware or software delays, a safety gap $\sigma = \varepsilon \cdot |F(i,1)|$ is prefixed to each firing slot $F(i,1)$ (cf. Fig. 2). The factor ε should be selected carefully, because it shall cover possible drifts but not delay the firings of the nodes unnecessarily. Since there need not be any data slot, we made σ to be a function of the length of a firing slot. Finally, the period T has to hold

$$\begin{aligned} T &= n \cdot (\sigma + f) \\ &= n \cdot \Big(\sigma + |F(i,1)| + \sum_{j=2}^{k} |F(i,j)|\Big) \\ &= n \cdot \Big((1+\varepsilon) \cdot \underbrace{|F(i,1)|}_{\text{firing slot}} + \underbrace{\sum_{j=2}^{k} |F(i,j)|}_{\text{data slot(s)}}\Big) \end{aligned} \tag{1}$$

Because parameter n is specified by the network and the length of a firing packet dominates the safety gap σ but can only be influenced marginally as well, the only way to save energy is determined by the sum of the lengths of the data slots. Of course, the period T can be padded out to prolong the sleep periods of a radio transceiver, but for our further examinations we just consider the minimum value for period T specified by equation 1.

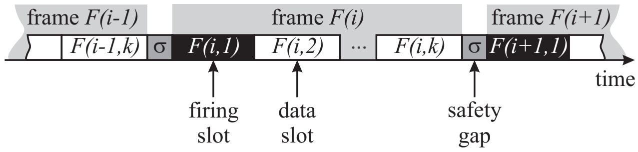

Fig. 2. Arrangement of frames, slots and safety gaps

To stay desynchronized, a node $i \in N$ must be just interested in the firing packets of its $n-1$ neighbors, thus it has to turn on its radio transceiver for at least

$$\Delta t_{i,RF} = \Delta t_{i,RX} + \Delta t_{i,TX},$$

where $\Delta t_{i,TX} = \sigma + |F(i,1)|$ denotes the duration[2] for broadcasting a firing packet, and $\Delta t_{i,RX} = (n-1) \cdot (1+\varepsilon) \cdot |F(i,1)|$ terms the elapsed time for reception of the firing packets of all its neighbors. With it, the uptime of the radio unit of node i is

$$\Delta t_{i,RF} = n \cdot (1+\varepsilon) \cdot |F(i,1)| \leq T. \tag{2}$$

Assuming all data slots have the same length f_k, i.e. for all $i \in \{1, \ldots, n\}$ and $m \in \{2, \ldots, k\}$ holds $f_k = |F(i,m)|$, the gain of energy γ_i compared to an always activated radio controller of a node i in percent per period T is

$$\begin{aligned} \gamma_i &= \frac{T - \Delta t_{i,RF}}{T} \\ &= 1 - \frac{n \cdot (1+\varepsilon) \cdot |F(i,1)|}{n \cdot \big((1+\varepsilon) \cdot |F(i,1)| + (k-1) \cdot f_k\big)} \\ &= \frac{(k-1) \cdot f_k}{(1+\varepsilon) \cdot |F(i,1)| + (k-1) \cdot f_k} \end{aligned} \tag{3}$$

If we additionally suppose that firing slots and data slots are of the same length, i.e. for all $i \in \{1, \ldots, n\}$ holds $|F(i,1)| = f_k$, equation 1 reduces to

$$T = n \cdot (k + \varepsilon) \cdot f_k,$$

and furthermore equation 3 simplifies to

$$\gamma_i = \frac{k-1}{k+\varepsilon}$$

That means, if there are only firing slots but not a single data slot (i.e. $k = 1$), and if the relevant number of neighbors is at maximum support for period T, it is not possible to save energy by reducing the uptime of the radio unit.

[2]Before transmission, each node has to use CS to detect joining nodes or drifting neighbors. Here we expect the same safety gap σ as length for the CS phase.

However, further savings could be achieved if node i only needs to receive some $\eta \in \{0, \ldots, n-1\}$ of firings of its neighbors per period T. This way, equation 3 adjusts to

$$\begin{aligned}\gamma_i^\eta &= \frac{T - \Delta t_{i,RF}^\eta}{T} \\ &= 1 - \frac{(\eta+1)\cdot(1+\varepsilon)\cdot|F(i,1)|}{n\cdot\big((1+\varepsilon)\cdot|F(i,1)| + (k-1)\cdot f_k\big)} \\ &= \frac{\left(\frac{n-\eta-1}{n}\right)\cdot(1+\varepsilon)\cdot|F(i,1)| + (k-1)\cdot f_k}{(1+\varepsilon)\cdot|F(i,1)| + (k-1)\cdot f_k}.\end{aligned} \tag{4}$$

With equal length for firing slots and data slots, equation 4 again reduces to

$$\gamma_i^\eta = \frac{k - 1 + (1+\varepsilon)\cdot\frac{n-\eta-1}{n}}{k+\varepsilon}.$$

If the radio transceiver is powered down for several periods, even more energy could be saved. But such a long down time implicates additional problems which can destabilize collision free communication and require extra administrative costs to keep track of the down times of nearby nodes. Hence, we won't go into detail here, but discuss in the next section the effect on network latency using some results from this section about energy-savings.

IV. Latency

So far, the period T mainly depends on n, the maximum number of supported nodes, and the slot lengths. But when examining the network latency, further parameters are of interest, like data rate or minimum packet length. Thus, if the firing packet contains additional information about the neighbors of the transmitting node, for instance to prevent the hidden node problem, the length of a firing slot $|F(i,1)|$ indeed depends on n. Introducing an adequate factor β with subject to network specific variables and leaving n fixed, the length of a firing slot can be specified as

$$|F(i,1)| = \beta\cdot n.$$

With it and according to equation 1, the minimal period T to support just firing packets (i.e. $k=1$) plus safety gap for n nodes is

$$T = (1+\varepsilon)\cdot\beta\cdot n^2,$$

which is quite similar to equation 2.

Assuming that the length δ of the data section within a frame $F(i)$ is independent of n, the period T must hold

$$\begin{aligned}T &= \big((1+\varepsilon)\cdot\beta\cdot n + \delta\big)\cdot n \\ &= (1+\varepsilon)\cdot\beta\cdot n^2 + \delta\cdot n.\end{aligned}$$

But if the length of a firing slot is the disposing base unit as mentioned in Section III, where all slots have the same length $f_k = |F(i,1)|$, the length δ of the data section can be rephrased as a function of the number n of supported nodes

$$\delta = \delta_0\cdot\beta\cdot n$$

by using another factor δ_0. Thus, the minimal period T now modifies to

$$\begin{aligned}T &= \big((1+\varepsilon)\cdot\beta\cdot n + \delta_0\cdot\beta\cdot n\big)\cdot n \\ &= (1+\varepsilon+\delta_0)\cdot\beta\cdot n^2.\end{aligned}$$

Overall, the number n of supported nodes has a much stronger influence on the length of period T, if n affects the length of all slots. That means, if the maximum number n of supported nodes increases, the period T grows with the square of n, the same is true for the network latency. Thus, a node has to wait in order of n^2 until its next firing, and so will a joining node, especially if they reside within an area of low density. For this reason, it seems not clever to make the slot lengths dependent on the number n.

That's why the trade-off between energy savings and network latency is quite complex – especially in networks of non-uniformly distributed nodes, containing areas of high density, causing a great value of n and – as a result – a great period T, and areas of low density, containing lots of unused slots.

V. Conclusion and Outlook

After a short motivation, we first introduced the biologically inspired and self-organized TDMA protocol DESYNC for Wireless Sensor Networks in Section II. Next, we analyzed its energetic characteristics in Section III, where we identified some adjustable parameters to save energy. In Section IV, we examined the latency performance of the DESYNC protocol with subject to the number n of supported nodes within a period. As a remarkable result, the length of the period is in order of square of n, if the slot lengths also depend on it.

For further research, we want to build a real-world testbed to specify some of our factors, like σ and δ. We also want to analyze the impact of an additional energy-harvesting unit at sensor nodes, which may influence the duty-cycle of the radio unit, too. As well, we try to promote a more universal version of the DESYNC protocol for multi-hop topologies using extra but locally available information. This additional information within a firing packet may be sufficient to support further add-ons, like time synchronization or routing.

References

[1] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min, "Z-MAC: a Hybrid MAC for Wireless Sensor Networks," in *SenSys*, Jason Redi, Hari Balakrishnan, and Feng Zhao, Eds. ACM, 2005, pp. 90–101.

[2] V. Rajendran, K. Obraczka, and J. Garcia-Luna Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Networks*, vol. 12, no. 1, pp. 63–78, Feb. 2006, tRAMA. [Online]. Available: http://dx.doi.org/10.1007/s11276-006-6151-z

[3] Marcel Baunach, "Speed, Reliability and Energy Efficiency of HashSlot Communication in WSN Based Localization Systems," in *EWSN*, ser. Lecture Notes in Computer Science, Roberto Verdone, Ed., vol. 4913. Springer, 2008, pp. 74–89.

[4] Julius Degesys, Ian Rose, Ankit Patel, and Radhika Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks," in *IPSN*, Tarek F. Abdelzaher, Leonidas J. Guibas, and Matt Welsh, Eds. Cambridge, MA, USA: ACM, 2007, pp. 11–20.

[5] Julius Degesys and Radhika Nagpal, "Towards Desynchronization of Multi-hop Topologies," in *SASO*. Venice, Italy: IEEE Computer Society, 2008, pp. 129–138.

[6] Renato E. Mirollo and Steven H. Strogatz, “Synchronization of Pulse-Coupled Biological Oscillators,” *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.

[7] Alessandro Giusti, Amy L. Murphy, and Gian Pietro Picco, “Decentralized Scattering of Wake-up Times in Wireless Sensor Networks,” in *EWSN*, ser. Lecture Notes in Computer Science, Koen Langendoen and Thiemo Voigt, Eds., vol. 4373. Springer, 2007, pp. 245–260.

[8] L. F. W. van Hoesel and P. J. M. Havinga, “A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches,” in *1st International Workshop on Networked Sensing Systems (INSS)*. Tokio, Japan: Society of Instrument and Control Engineers (SICE), 2004, pp. 205–208.

[9] Gertjan P. Halkes and Koen Langendoen, “Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks,” in *EWSN*, ser. Lecture Notes in Computer Science, Koen Langendoen and Thiemo Voigt, Eds., vol. 4373. Springer, 2007, pp. 228–244.

[10] Koen Langendoen and Thiemo Voigt, Eds., *Wireless Sensor Networks, 4th European Conference, EWSN 2007, Delft, The Netherlands, January 29-31, 2007, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4373. Springer, 2007.