© IEEE, 2010. This is the authors' version of the work. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purpose or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. The definite version is published in Proc. IEEE International Communications Conference (ICC), Cape Town, South Africa, May 23-27, 2010. ieeexplore.ieee.org.

How Does a Faulty Node Disturb Decentralized Slot Synchronization over Wireless Networks?

Alexander Tyrrell Gunther Auer
DOCOMO Euro-Labs
80687 Munich, Germany
Email: lastname@docomolab-euro.com

Christian Bettstetter Rahul Naripella
University of Klagenfurt, Mobile Systems Group
9020 Klagenfurt, Austria
Email: Firstname.Lastname@uni-klu.ac.at

***Abstract*—Decentralized synchronization requires cooperation among network participants so that all nodes agree on a common reference timing. What happens if one node does not follow local synchronization rules and randomly transmits? This paper studies the resilience of two classes of decentralized slot synchronization against random disturbances, and quantifies the impact of the random behavior. It is shown that the coupling strength is a key factor for resilience, and that the synchronization approach based on the theory of coupled oscillators generally behaves better and is more robust than the approach that updates clocks based on the average neighboring timings.**

I. INTRODUCTION

Many communication systems require an agreement on a common time slotted structure for correct operation. For instance, in Time Division Multiple Access (TDMA), nodes transmit in allocated time slots. Synchronization is also very useful in wireless sensor networks to correctly estimate the speed and direction of an object moving in the proximity of the sensors. With the development of wireless sensor networks and ad hoc networks, the importance of synchronization led to the development of a variety of synchronization schemes. While centralized synchronization approaches exhibit some drawbacks, e.g. high overhead related to dispersing the timing of the central timing reference, low reactivity to a changing environment, and a central point of failure, decentralized synchronization circumvents these drawbacks but requires nodes to exchange their timing information locally.

The local exchange of timing information is either done explicitly through the exchange of timestamps or implicitly by detecting the timing of a transmission at the receiver. For the purpose of slot synchronization, implicit timing information is provided by transmitting at the beginning of a time slot. It has been utilized, for example, by two algorithms available in the literature to perform decentralized slot synchronization:

- Meshed Emergent Slot Synchronization (MEMFIS) [1] is based on the synchronization rules of pulse-coupled oscillators [2]. It updates internal clocks in a discrete manner when detecting a neighboring transmission.
- Power-Weighted Average Synchronization (PWASync) records neighboring timings over one slot and updates internal clocks based on a power-weighted average of recorded timings [3–5].

Previous studies of MEMFIS and PWASync assume that all nodes in the network follow the same synchronization rules. In the following a node that transmits randomly is introduced in the network, and its disruption is evaluated based on a synchronization metric that measures the level of local synchrony. This random behavior may mimic a malfunctioning node that does not follow the correct synchronization rules, or it could originate from a malicious node that intentionally tries to disrupt the synchronization state of surrounding nodes. In this paper we investigate the ability of MEMFIS and PWASync to maintain an acceptable level of synchronization in the presence of faults to normal operation, i.e. the resilience of the algorithms. This is an important first step before investigating possible detection methods and countermeasures.

The remainder of this article is structured as follows. Section II reviews the two slot synchronization schemes MEMFIS and PWASync, and compares their characteristics. The impact of randomly transmitted reference timings on MEMFIS and PWASync is studied in Section III. The behavior of both algorithms is analyzed as a function of the node coupling, the rate of random transmissions, and the network connectivity. It is shown that MEMFIS tends to be more resilient against such kind of disturbance, i.e. its behavior is more constant under a range of parameters. Finally Section IV concludes.

II. DECENTRALIZED SLOT SYNCHRONIZATION

MEMFIS and PWASync propose fundamentally different rules to achieve slot synchronization in a decentralized manner. In spite of their different rules, both schemes assume similar system constraints. Time is divided in slots of equal duration T, and nodes are either in a receive or a transmit slot. In a receive slot, both algorithms obtain implicit timing information by detecting transmissions of a synchronization word (sync-word) that is common to all nodes and transmitted along with data. In a transmit slot, each data packet is composed of the sync-word and data, and the common sync-word is transmitted only when data is to be transmitted. Half-duplex transmission is considered, i.e. sync-words cannot be received in a transmit slot. In this section, the update rules of MEMFIS and PWASync for achieving synchronization are reviewed, and their performance is compared in wireless meshed networks.

A. MEMFIS: Meshed Emergent Firefly Synchronization

MEMFIS is adapted from the theory of pulse-coupled oscillators (PCOs), which provides simple local rules leading

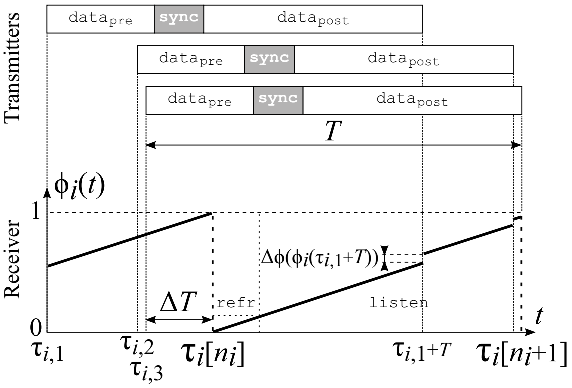

Fig. 1. Reference timing update with MEMFIS [1].

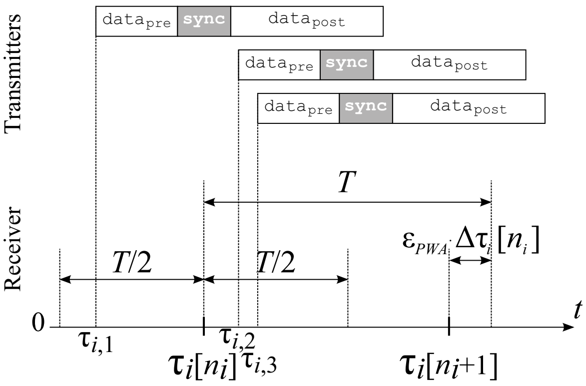

Fig. 2. Reference timing update with PWASync [4].

to system-wide synchronization. This mathematical model describes entities that periodically interact with each other at discrete time instants via infinitely short pulses. It is used in many fields of science to describe dynamic natural phenomena such as firefly synchronization [2].

The PCO model is not directly applicable to wireless networks, and MEMFIS adapts it in several ways so that synchronization is reached from any initial condition [1]. Fig. 1 describes the adaptation process of a receiving node when detecting three transmissions within a time slot.

In Fig. 1 the receiving node, denoted node i, observes the transmission of three nodes with respective timings $\tau_{i,1}$, $\tau_{i,2}$, and $\tau_{i,3}$. During a receive slot, node i maintains a phase function $\phi_i(t)$ that increases linearly over time from 0, marking the start of a slot, to 1. A reference instant τ_i is reached when the phase reaches 1. At this instant, a node decides whether to transmit, if data is to be transmitted, or to receive. A receive slot is composed of a refractory period of duration T_{refr} where no phase increment is possible, and of a listening period. In Fig. 1, the three increment instants $\tau_{i,1}{+}T$, $\tau_{i,2}{+}T$, and $\tau_{i,3}{+}T$ all fall in the `listen` state and cause three discrete phase increments that update the receiver's phase function according to:

$$\phi_i(\tau) \to \phi_i(\tau^+) = \phi_i(\tau) + \Delta\phi\left(\phi_i(\tau)\right) \ , \tag{1}$$

where τ^+ denotes an infinitesimal time instant after τ. The increment function $\phi_i(\tau){+}\Delta\phi\left(\phi_i(\tau)\right)$ is referred to as phase response curve (PRC).

In [2] conditions on the PRC to reach synchrony for arbitrary initial time offsets are identified. A simple PRC leading to synchrony is the piecewise linear function:

$$\phi_i(\tau^+) = \min\left((1+\alpha)\cdot\phi_i(\tau)+\beta, 1\right) \ . \tag{2}$$

The terms α and β are the coupling parameters: $1{+}\alpha$ is the slope and β is the initial value of the PRC. Assuming that each node maintains the same PRC, if $\alpha{>}0$ and $0{<}\beta{<}1$, a set of nodes always synchronizes independently of the initial conditions.

B. PWASync: Power-Weighted Average Synchronization

In PWASync, slot synchronization is performed by adjusting local clocks based on the average of detected timings. This approach was initially proposed for inter-base station synchronization in cellular networks [3], and was later extended to the dynamic environment of inter-vehicle communication [4, 5]. The application of PWASync in [5] fits similar constraints as MEMFIS: sync-words are considered instead of pulses; these are time-multiplexed with data to form packets.

The basic mechanism of PWASync is summarized in Fig. 2. In slot n_i node i monitors the detected reference instants $\tau_{i,j}$ and the received power level of transmitting node j, and determines the power-weighted average timing error [4]:

$$\Delta\tau_i[n_i] = \frac{\sum_{j\in\mathcal{N}_i[n_i]} P_{ij}\cdot\Delta\tau_{i,j}[n_i]}{\sum_{j\in\mathcal{N}_i[n_i]} P_{ij}} \ , \tag{3}$$

where $\mathcal{N}_i[n_i]$ is the set of transmitting nodes detected during slot n_i, P_{ij} is the power level of transmitter j detected at node i, and with

$$\Delta\tau_{ij}[n_i] = \tau_{i,j} - \tau_i[n_i] \ . \tag{4}$$

Based on this average, the timing reference of node i is updated according to:

$$\tau_i[n_i{+}1] = \tau_i[n_i] + \epsilon_{\text{PWA}}\cdot\Delta\tau_i[n_i] \ , \quad 0 < \epsilon_{\text{PWA}} < 0.5 \ , \tag{5}$$

where ϵ_{PWA} is the coupling strength associated with a timing update.

C. Performance Comparison

In this section MEMFIS and PWASync are briefly evaluated in a meshed network based on a local synchronization metric.

1) Meshed Network: In the following, the network topology is modeled as a *random geometric graph* $\mathcal{G}(N,d)$: N nodes are placed on a square area using a uniform random distribution, and nodes are connected if their distance is lower or equal than d. The set of links is denoted by $\mathcal{E}$, and two connected nodes are called *neighbors*. The set of neighbors of node i is defined as $\mathcal{N}_i{=}\{j:(i,j)\in\mathcal{E}\}$.

A common measure to characterize topological properties of a network is its *algebraic connectivity* [6]. It is denoted by κ in the following, and is defined as the smallest non-zero eigenvalue of the Laplacian matrix $\mathbf{L}(\mathcal{G})$ [6]. This eigenvalue is strictly greater than 0 if and only if $\mathcal{G}$ is a connected graph [6]. For a fully-meshed network, i.e. all nodes connected to each other, $\kappa{=}N$. For a given N, the algebraic connectivity κ is varied by changing d: from the minimal distance ensuring a connected graph where κ is very close to zero, the connectivity increases very rapidly until the maximum $\kappa{=}N$.

2) Synchronization Metric: In order to characterize the local misalignment in timing references, a metric that quantifies the level of local synchronization was proposed for MEMFIS in [7], and is here adapted for PWASync. The objective of a synchronization metric is to identify whether a network is synchronized and to quantify the state of the system. When all nodes agree on a common reference instant, the metric approaches 1. When the system is in disorder, the metric approaches 0.

The local synchronization metric for node i is defined as:

$$r_i = 1 - \frac{1}{|\mathcal{N}_i|}\left|\sum_{j\in\mathcal{N}_i}\left(\exp\left(2\pi\frac{\tau_i}{T}\right)-\exp\left(2\pi\frac{\tau_j}{T}\right)\right)\right| . \qquad (6)$$

The local metric is defined in the interval $r_i{\in}[-1,1]$. If node i is synchronized with all its neighbors, then the complex exponentials in (6) cancel each other, and the local metric yields $r_i{=}1$. The metric decreases rapidly as the difference between τ_i and τ_j increases, e.g. when $\tau_i{-}\tau_j{=}0.1$, the local metric yields $r_i{\approx}0.4$. The metric is equal to 0 either when reference instants are equally distributed in $[0,T]$, or when nodes are anti-phase synchronized, i.e. two groups of equal size have references shifted by $0.5\,T$. For further details on the synchronization metric, please refer to [7].

3) Comparison: Synchronization Metric over Time: To illustrate the behavior of the local synchronization metric with MEMFIS and PWASync, Fig. 3 plots the evolution over time of the mean local synchronization metric for different network connectivity values. The coupling values are set to $\alpha{=}0.2$ and $\beta{=}0.01$ for MEMFIS and to $\epsilon_{\mathrm{PWA}}{=}0.5$ for PWASync, and the number of nodes in the network is equal to $N{=}8$. The data traffic follows a Poisson distribution with an average load set to $\lambda{=}1.0$ packets per slot for the whole network.

In Fig. 3 reference instants are initially randomly distributed in $[0,T]$, and the mean metric is close to 0. Synchronization emerges rapidly under all conditions, and within $40\,T$, fully-meshed networks and networks with a connectivity of $\kappa/N{=}0.5$ are synchronized. In networks with a low connectivity value of $\kappa/N{=}0.1$, synchronization is not always reached within $50\,T$. However MEMFIS is better able to cope with sparse networks, as the metric with PWASync grows very slowly when $t{\geq}30\,T$.

4) Comparison: Influence of the Coupling: The coupling parameters α and ϵ_{PWA} are very important for MEMFIS and PWASync respectively. They determine how strongly a node reacts when detecting a transmission from a neighboring node. Strong coupling values lead to quick convergence, but also cause instability because nodes tend to adjust their timing reference too strongly when receiving a sync-word. On the other hand, weak coupling implies that nodes are less reactive after detecting a sync-word, but the system requires more time to synchronize. Fig. 4 plots the mean time to synchrony over the coupling values α and ϵ_{PWA} for different network connectivities.

Fig. 4 confirms the superior scalability of MEMFIS in sparse networks. Provided that $\alpha{\geq}0.2$, further increasing the coupling value α has little effect on the mean time to synchrony with MEMFIS. Networks with higher connectivity synchronize faster, a behavior that was mathematically proven in [8]. For PWASync, a high coupling value ϵ_{PWA} is required so that an acceptable time to synchrony is achieved. Furthermore the algorithm does not scale well in sparse networks, as the mean time to synchrony abruptly increases when the connectivity is decreased from $\kappa/N{=}0.5$ to $\kappa/N{=}0.1$.

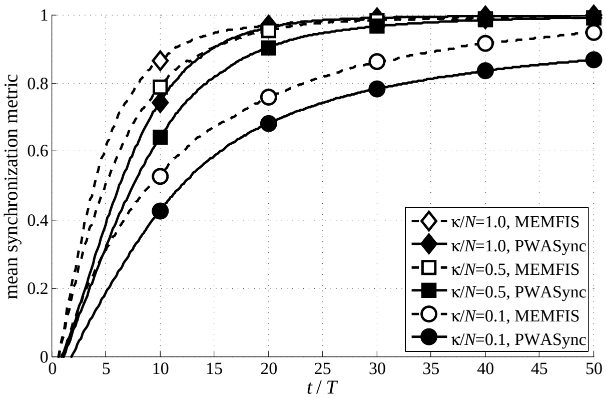

Fig. 3. Evolution of the mean local synchronization metric over time for MEMFIS and PWASync for different network connectivity values.

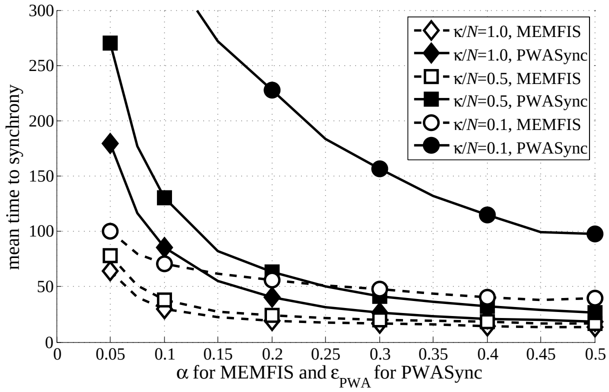

Fig. 4. Mean time to synchrony over the coupling strength for MEMFIS and PWASync.

III. RESILIENCE TO RANDOMLY TRANSMITTED SYNC-WORDS

In this section, we investigate the impact of randomly transmitted sync-words on MEMFIS and PWASync. Both schemes do not identify the transmitter of a sync-word, and rely only on its implicit timing information to perform synchronization. In Section III-A we detail the random transmissions, which reflect a malfunctioning node or the strategy of a malicious node that attempts to disrupt a synchronized network. The impact of these random transmissions are investigated in Section III-B through the synchronization metric level.

A. Random Behavior

We model a faulty or malicious node as follows. The node neither transmits sync-words periodically nor reacts to sync-words from other nodes, but transmits sync-words randomly without any predefined time slot structure. The sync-words are transmitted according to a Poisson process with rate λ_r. The Poisson distribution is known to maximize the entropy in the system [9], and thus seems to be a natural choice for a malicious node to choose in order to disrupt synchrony.

An important point for this disruptive approach to be effective is that nodes transmit only when there is data scheduled to be transmitted, otherwise they remain in listen state. The randomness in receive and transmit slots and the half-duplex condition of normal nodes make it possible for a random sync-word to disrupt synchronized nodes, because this sync-word is not received by the same set of nodes.

B. Impact on MEMFIS and PWASync

In the following the impact of a node randomly transmitting sync-words is evaluated through simulations. Unless mentioned otherwise, the coupling parameters are set to α=0.2 and β=0.01 for MEMFIS and to ϵ_PWA=0.3 for PWASync. The default number of nodes and traffic density are set to N=8 and λ=1.0 in Sections III-B1, III-B2, and III-B3, and to N=20 and λ=2.0 in Section III-B4. Finally the network connectivity is set by default to κ/N=0.5.

1) Synchronization Metric over Time: Fig. 5 plots the evolution of the mean synchronization metric over time. Initially reference instants are distributed in $[0, T]$, and the network synchronizes, i.e. the metric increases from 0 to 1. Random sync-word transmissions start at t=100 T. Their impact is very rapid; within 10 periods, the mean synchronization metric drops from 1 to a constant level that depends on the rate of random transmissions. For a low arrival rate, e.g. λ_r=0.1 pkt/slot, the synchronization metrics for both MEMFIS and PWASync settle around 0.9. This level of synchrony decreases to approximately 0.6 for random transmission rates of λ_r=1.0 and λ_r=2.0. The synchronization level of PWASync is proportional to the arrival rate of random sync-words, i.e. a higher rate of random transmissions disturbs more the synchronization state. Interestingly, using MEMFIS, the level of synchronization does not necessarily decrease with increasing rate, and its level of synchrony is higher for λ_r=2.0. In the following the constant level of the mean metric after the start of the random transmissions in the steady-state is termed "sync-metric level" and serves as the metric in the following sections.

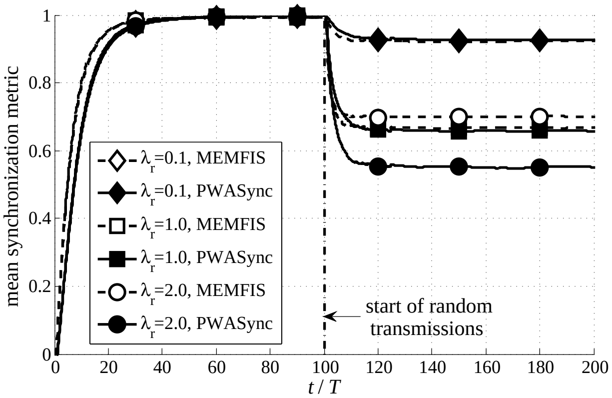

Fig. 5. Mean time to synchrony over time when random transmissions start at t=100 T.

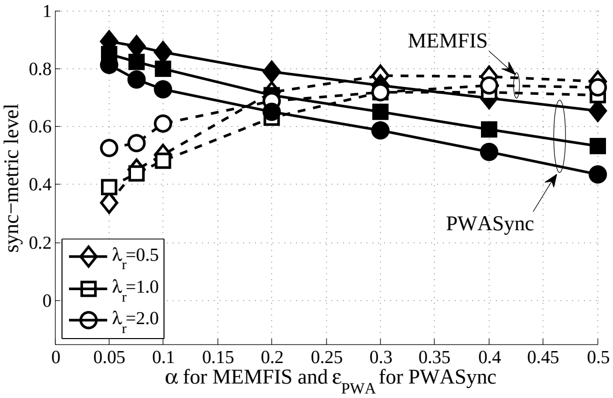

Fig. 6. Mean synchronization level after random transmissions over the coupling strength.

2) Influence of Coupling: As detailed in Section II-C4, the coupling between nodes is an important parameter for MEMFIS and PWASync. Fig. 6 investigates the impact of the coupling parameter on the sync-metric level.

As randomly transmitted sync-words are interpreted as normal sync-words, the coupling strength is as strong between the faulty or malicious node and normal nodes. From Fig. 6, increasing the coupling in MEMFIS minimizes the impact of the random transmissions, and keeps the mean sync-metric level around 0.7. On the other hand, increasing the coupling in PWASync privileges the sync-words from the malicious node, and the mean sync-metric decreases linearly with the increase of ϵ_PWA. This behavior in Fig. 6 is incompatible with an appropriate time to synchrony: from Fig. 4, a high coupling value is required, in particular in sparsely connected networks, so that nodes synchronize quickly. On the other hand, MEMFIS behaves optimally in Fig. 6 when α=0.3, a value that also yields a low time to synchrony.

The superior behavior of MEMFIS, i.e. its better resilience to random sync-words with high coupling, is explained by the

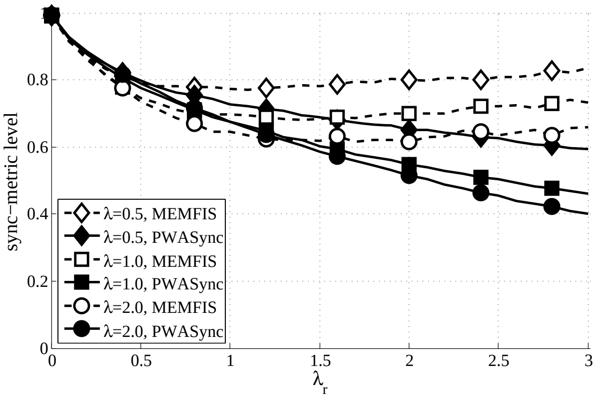

Fig. 7. Mean synchronization level after random transmissions over the arrival rate of random transmissions.

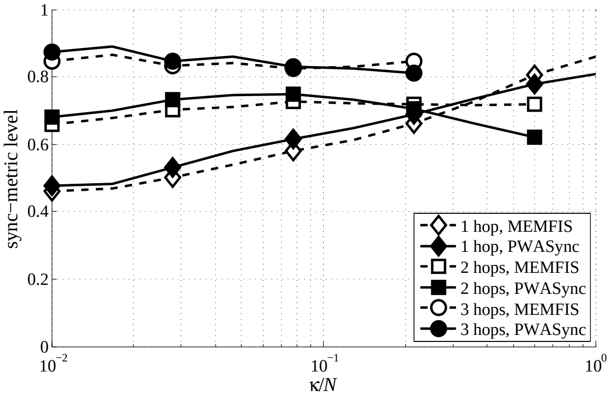

Fig. 8. Mean synchronization level after random transmissions over the network connectivity.

different dynamics than PWASync. In PWASync, the timing adjustment in (5) depends on the coupling factor ϵ_{PWA} and, more importantly, on the perceived timing difference at the receiver $\Delta\tau_i[n_i]$. Therefore, if a received sync-word is far from the local timing reference, the receiver adjusts its timing more strongly than when if $\Delta\tau_i[n_i]$ is small. In MEMFIS, on the contrary, when a node detects a sync-word close to its reference instant, it adopts it immediately by setting its phase to 1, before entering the refractory period where it neglects subsequent sync-words.

3) Influence of the Rate of Random Transmissions: In Fig. 5 it is observed that MEMFIS is less affected by the random sync-words when they are transmitted more often. Fig. 7 investigates the behavior of MEMFIS and PWASync as the arrival rate of random sync-words λ_{r} increases and the arrival rate of normal transmissions λ varies.

Fig. 7 confirms the better resilience of MEMFIS with regards to random transmissions. After reaching a minimum value for $\lambda_{\mathrm{r}}\approx 1$, the level of synchrony with MEMFIS increases as a higher number of sync-words are transmitted randomly. On the other hand, the level of synchrony with PWASync decreases as λ_{r} increases. In all cases a higher rate of normal transmissions leads to a decreased level of synchrony, which is due to the increased number of disagreeing transmissions.

4) Diffusion of the Random Behavior: The faulty or malicious node is randomly chosen in a meshed network, and its transmissions are received by a subset of nodes, i.e. its neighbors. Fig. 8 investigates the diffusion of the random transmissions for different connectivity values, relative to the distance, i.e. hop count, of a normal node from the malfunctioning or malicious node.

In Fig. 8 both MEMFIS and PWASync behave similarly. For a connectivity of $\kappa/N{=}1.0$, the network is fully-meshed, and all nodes are thus directly connected to the misbehaving node. In this case, its impact is compensated by the high connectivity among normal nodes, i.e. transmissions from normal nodes compensate the misaligned transmissions. As the connectivity decreases, the impact of the random sync-words is stronger, and nodes that are directly connected to the misbehaving node, i.e. "1 hop" nodes, are strongly affected. Nodes that are placed two or three hops away from the faulty or malicious node, and thus are not directly affected by it, are less impacted than neighbors of the misbehaving node, but their synchronization level is nevertheless below 1.

IV. CONCLUSION

This paper studied the resilience of two classes of decentralized slot synchronization algorithms in the presence of a faulty node that transmits sync-words randomly. This study was done using a synchronization metric that quantifies the level of synchrony in the network. It was shown that MEMFIS, which is based on the theory of coupled oscillators, is generally more robust than PWASync, which adjusts timing references based on the average of received sync-words.

The used methodology is a useful tool for future work, which will investigate the detection of faulty or malicious nodes and possible counter-measures.

REFERENCES

[1] A. Tyrrell, G. Auer, and C. Bettstetter, "Emergent slot synchronization in wireless networks," *IEEE Trans. Mob. Comp.*, vol. 99, no. PrePrints, 2009.

[2] R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. on Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, Dec. 1990.

[3] Y. Akaiwa, H. Andoh, and T. Kohama, "Autonomous decentralized inter-base station synchronization in TDMA microcellular systems," in *Proc. IEEE Vehic. Technol. Conf. (VTC)*, St. Louis, USA, May 1991.

[4] E. Sourour and M. Nakagawa, "Mutual decentralized synchronization for intervehicle communications," *IEEE Trans. Vehic. Technol.*, vol. 48, no. 6, pp. 2015–2027, Nov. 1999.

[5] A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "Decentralized slot synchronization in highly dynamic ad hoc networks," in *Proc. Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Hawaii, USA, Oct. 2002.

[6] N. L. Biggs, *Algebraic Graph Theory*, 2nd ed. Cambridge University Press, 1994.

[7] A. Tyrrell, G. Auer, and C. Bettstetter, "A synchronization metric for meshed networks of pulse-coupled oscillators," in *Proc. Int. Conf. on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS)*, Hyogo, Japan, Dec. 2008.

[8] D. Lucarelli and I.-J. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proc. ACM Conf. Embedded Networked Sensor Sys. (SenSys)*, Baltimore, USA, Nov. 2004.

[9] J. Walrand, "Entropy in communication and chemical systems," in *Proc. Int. Symp. on Applied Sciences in Biomedical and Commun. Tech. (ISABEL)*, Aalborg, Denmark, Oct. 2008.