# Optimizing Guard Time for TDMA in a Wireless Sensor Network*

## - Case Study -

Oday Jubran
Carl von Ossietzky Universität Oldenburg
26111 Oldenburg, Germany
jubran@informatik.uni-oldenburg.de

Bernd Westphal
Albert-Ludwigs-Universität Freiburg
79110 Freiburg, Germany
westphal@informatik.uni-freiburg.de

***Abstract*—When the Time Division Multiple Access (TDMA) protocol is used to schedule communication over a shared medium in wireless sensor networks, clock drift can be tolerated using guard time. A guard time is a short time interval that is added between the TDMA slots. Using guard time requires extra time and energy, which is critical in wireless sensor networks operating on limited energy. In this case study, we apply a formal approach to compute optimal safe guard times, i.e. shortest guard times which avoid message collision and loss due to clock drift in a wireless sensor network. We show the flexibility of the formal approach and its usefulness under several real-world conditions.**

***Keywords*—TDMA; Clock Drift; Slot Assignment; Guard Time**

## I. Introduction

Wireless sensor networks utilize the Time Division Multiple Access (TDMA) protocol [1] to allow nodes to share a communication medium without message collision or loss, by dividing time into slots, in which nodes are scheduled to send and receive messages. Indeed, in energy critical systems, TDMA allows to reduce energy drain by setting the nodes into sleep mode during the slots, in which the nodes do not send or listen to messages. In some network architectures, nodes are equipped with clocks to distinguish their assigned slots, and these clocks may exhibit clock drift. A consequence of clock drift can be message collision and loss. In time critical systems, message collision and loss hinder the communication between the nodes, and therefore, clock drift has to be treated.

Clock drift is mainly treated by clock synchronization mechanisms, which can limit the amount of clock drift, but which can in general not guarantee the absence of clock drift because clocks of course continue to drift between synchronization points. Some TDMA-based networks employ the notion of *guard time* [2], which is a time interval that is added to the slot, in which the assigned node listens, but does not send messages. By choosing a reasonable length of guard time, a small amount of clock drift can be tolerated. A guard time with a length that is large enough to tolerate a given maximal clock drift is called a *safe* guard time. Guard time extends the slot length thus it increases the duration between two slots of a node. As a consequence, the message delivery between the nodes requires longer time which reduces the network performance. Indeed, with guard time, receiving nodes require more energy during the listening mode. Therefore, the guard time length is required to be as short as possible.

The classical-engineering approaches to guard time minimization in a system comprise extensive testing, simulation, and calculations based on experience with the system architecture, its requirements, and environment conditions. Such approaches usually result in fair approximations of the optimal guard time length, even when certain safety margins are added to compensate for the incompleteness of these approaches. However, these approaches take a long time or many test systems to be performed, an effort which may be necessary for each modification of the system during further development.

In this work, we study the practical effectiveness of a formal technique [3] that minimizes guard time *analytically*. To this end, we consider a wireless fire alarm system which has been developed using a classical-engineering approach. This study investigates the usefulness and flexibility of the formal technique [3] to compute optimal guard times quickly, safely, and under several real-world conditions on the system. We show that the computed guard times have significantly shorter lengths than the one given in the fire alarm system design. We also show that by adding further restrictions, e.g., on the system's network topology and the slot assignment, guard time can safely be reduced even further.

**Outline**: Section II recalls the formal approach [3] to derive an optimal guard time.Section III presents the fire alarm system to be considered. Section IV presents computations of several guard times in different situations. Section V investigates how the guard times from Section IV affect response time and energy consumption. Section VI concludes.

## II. Preliminaries

For self-containedness, we recall the formal model of the class of systems that we consider and the theorems concerning guard time optimization. For further details, we refer to [3].

### A. Formal Model of the System

We use the notion of finite directed rooted trees. A *finite directed rooted tree* is a pair $T = (N, E)$ consisting of a finite

*This work was partially supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS, http://www.avacs.org/).

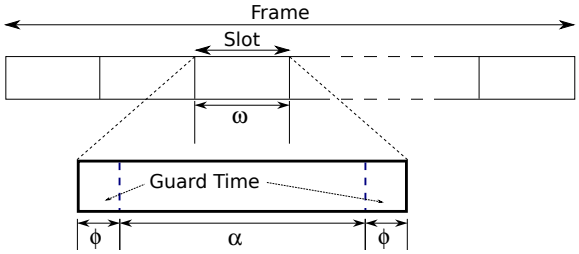

Figure 1: TDMA Frame.

set of *nodes* $N$ and a set of directed *edges* $E \subseteq N \times N$ where there exists a unique *root* node $n$ such that, for each $n' \in N$, there is a unique path from $n$ to $n'$ in $T$. A *path* of $T$ is a sequence of nodes $n_0, \ldots, n_q$ of *length* $q$ where the pairs $(n_i, n_{i+1})$, $0 \leq i < q$, are in $E$. Edge $(n, n') \in E$ is directed from $n$ to $n'$. For each edge $(n, n') \in E$, $n$ is called the *master* of $n'$, denoted by $ms(n')$, and $n'$ is called a *slave* of $n$. The number of nodes in $N$ is called the *size* of $T$. The *depth* $depth(n)$ of a node $n \in N$ is defined inductively as follows: The root has depth 0, and for $(n, n') \in E$, $n'$ has depth $depth(n) + 1$. We say a tree $T$ *has depth* $d \in \mathbb{N}_0$ if $d$ is the maximum depth observed for any node in $T$. Let $T = (N, E)$ be a tree with root $n$. A tree $T' = (N', E')$ with $N' \subseteq N \setminus \{n\}$, $E' \subseteq E$, and root $n'$ such that $(n, n') \in E$ is called *subtree* of $T$. The trivial tree, which consists of the root only, does not have any subtree. A subtree of $T$ is called *maximal* if $T$ does not have any strictly larger subtree.

A *network topology*, as considered in [3] is a directed rooted tree $T$, where the root is called a *central unit* ($cu(T)$), and other nodes are called *sensors* ($Sn(T)$), where $Sn(T) \neq \emptyset$. We consider a timed system, where each node has a clock, and may listen or send messages. The clock value of the central unit equals the reference time. Formally, an *evolution* over a topology $T = (N, E)$ is an interpretation $\mathcal{I}$ of the observables $clk_n : \mathsf{Time}\ (= \mathbb{R}_0^+)$, $send_n : \mathbb{B}$, and $listen_n : \mathbb{B}$ for $n \in N$ such that $\mathcal{I}(clk_{cu(T)})(t) = t$ for each $t \in \mathsf{Time}$. We write $clk_n^{\mathcal{I}}(t)$, $send_n^{\mathcal{I}}(t)$, and $listen_n^{\mathcal{I}}(t)$ to denote $\mathcal{I}(clk_n)(t)$, $\mathcal{I}(send_n)(t)$, and $\mathcal{I}(listen_n)(t)$, respectively. A point in time $t \in \mathsf{Time}$ is called *synchronization point* of sensor $n \in Sn(T)$ in $\mathcal{I}$ iff $clk_n^{\mathcal{I}}(t) = clk_{ms(n)}^{\mathcal{I}}(t)$.

We define notions of speed, drift, and drift rate of clocks. (1) The *clock speed* of node $n \in N$ in $\mathcal{I}$, denoted by $\varphi_n^{\mathcal{I}}$, is the first derivative of the interpretation of $clk_n$ with respect to time: $\varphi_n^{\mathcal{I}} = \frac{\partial}{\partial t}\, clk_n^{\mathcal{I}}(t)$. (2) The *clock drift* of node $n \in N$ in $\mathcal{I}$ at time $t \in \mathsf{Time}$, denoted by $\varrho_n^{\mathcal{I}}(t) \in \mathbb{R}$, is the difference between the clock values of $n$ and the central unit at time $t$ in $\mathcal{I}$: $\varrho_n^{\mathcal{I}}(t) = clk_n^{\mathcal{I}}(t) - clk_{cu(T)}^{\mathcal{I}}(t)$. (3) The *drift rate* of node $n \in N$ in $\mathcal{I}$, denoted by $\Delta_n^{\mathcal{I}}$, is the first derivative of the clock drift of $n$ in $\mathcal{I}$ with respect to time: $\Delta_n^{\mathcal{I}} = \frac{\partial}{\partial t}\, \varrho_n^{\mathcal{I}}(t)$.

We define message collision and loss as follows: An evolution $\mathcal{I}$ over a topology $T$ is said to have: (1) *message collision* at time $t \in \mathsf{Time}$ between two different sensors $n_1, n_2 \in Sn(T)$ iff both send at $t$, i.e. if $send_{n_1}^{\mathcal{I}}(t) \wedge send_{n_2}^{\mathcal{I}}(t)$, (2) *message loss* at time $t \in \mathsf{Time}$ for sensor $n \in Sn(T)$ iff $n$ is sending at $t$ while its master is not listening, i.e. if $send_n^{\mathcal{I}}(t) \wedge \neg listen_{ms(n)}^{\mathcal{I}}(t)$.

TDMA employs a partitioning of the time domain into frames and slots, as presented in Figure 1, with *slot length* $\omega \in \mathbb{R}^+$. We use the notions of scheduled and synchronized evolutions to schedule the actions of sending, listening, and clock synchronization over time slots. An evolution $\mathcal{I}$ over $T$ is called *scheduled* iff there exists an *assignment* of sensors to slots $assign : Sn(T) \to \{1, \ldots, k\}$, such that: (1) Each sensor $n \in Sn(T)$ sends messages only during the assigned slot according to its local clock. (2) For each sensor $n \in Sn(T)$, its master is listening in the slot assigned to $n$ according to the master's clock. On the other hand, evolution $\mathcal{I}$ is called *synchronized* if it is scheduled and each sensor has at least one synchronization point in each of its slots.

Guard time represents two time intervals added to the beginning and to the end of the slot, respectively. During the guard time of a slot, the assigned node is not allowed to send messages, but it is supposed to listen to its slaves. Formally, an evolution $\mathcal{I}$ over topology $T$ is said to have *guard time* $\phi \in \mathbb{R}_0^+$ iff (1) $\mathcal{I}$ is scheduled with a slot length $\omega \geq 2\phi$ and (2) sensors don't send for a duration of $\phi$ at the beginning and end of their slot. For each slot $[t_1, t_2)$ of $\mathcal{I}$ with guard time $\phi$, the time intervals $[t_1, t_1 + \phi)$ and $[t_2 - \phi, t_2)$ are called the *(left and right) guard intervals* of the slot. The time interval $[t_1 + \phi, t_2 - \phi)$ is called $\alpha$*-interval* of the slot.

We assume to have a fixed upper bound on the drift rate among all clocks, since the upper bound is typically part of the specification of clock modules. Let $\mathcal{I}$ be a synchronized evolution over topology $T$. $\Delta^{max} \in \mathbb{R}_0^+$ is a least upper bound on the magnitude of the drift rates in $\mathcal{I}$ iff $\Delta^{max}$ is the smallest number such that $\forall\, n \in Sn(T), t \in \mathsf{Time} \bullet |\Delta_n^{\mathcal{I}}(t)| \leq \Delta^{max}$.

A guard time $\phi \in \mathbb{R}_0^+$ is said to be *safe* for a topology $T$, slot length $\omega$, schedule $assign$, and least upper bound $\Delta^{max}$ on the drift rates iff no synchronized evolution of the given parameters exhibits message collision or message loss. For the systems we consider, the slot assignment order is critical.

### B. Formal Optimization of Guard Time

Following [3], there exist bounds on clock drift such that message collision and loss are avoided, given a guard time. The following theorem states these bounds.

THEOREM 1 (cf. [3], Theorem 1). *Let $\mathcal{I}$ be a scheduled evolution over topology $T$ with $k$ sensors and guard time $\phi$. (1) $\mathcal{I}$ does not have any message collision if $\forall\, n \in Sn(T), t \in \mathsf{Time} \bullet |\varrho_n^{\mathcal{I}} t)| \leq \phi$. (2) $\mathcal{I}$ does not have any message loss if $\forall\, n \in Sn(T), t \in \mathsf{Time} \bullet |\varrho_n^{\mathcal{I}}(t)| \leq \frac{\phi}{2}$.*

By [3], there are slot assignment orders that yield the highest value of maximum clock drift among all assignments. They are called worst case assignments. The following theorem shows how, in case of existence, an optimal safe guard time can be computed for worst case assignments, which is thus safe for all assignments.

THEOREM 2 (cf. [3], Corollary 3). *Let $assign$ be an assignment of slots to nodes for topology $T$ of depth $d$ with $k$ sensors. Let $\alpha \in \mathbb{R}^+$ be the length of the $\alpha$-interval. There exists a safe guard time for $T$ wrt. $\Delta^{max}$, and optimal for the worst case, iff $\Delta^{max} < \frac{1}{4(d(k-1)+2)}$. The optimal (minimal) safe guard*

*time for $T$ wrt. $\Delta^{max}$ is:*

$$\phi_{opt} = \alpha \cdot \frac{2(d(k-1)+2) \cdot \Delta^{max}}{1 - 4(d(k-1)+2) \cdot \Delta^{max}}.$$

On the other hand, there are slot assignment orders, namely the best case assignments, that yield the least value of maximum clock drift among all other assignments.

THEOREM 3 (cf. [3], Corollary 4). *Let assign be an assignment of slots to nodes for topology $T = (N, E)$ of depth $d$ with $k$ sensors and maximal subtree(s) of size at most $K \in \mathbb{N}^+$. Let $\alpha \in \mathbb{R}^+$ be the length of the $\alpha$-intervals. There exists a safe guard time for $T$ wrt. $\Delta^{max}$ iff $\Delta^{max} < \frac{1}{4(K+k)}$, and the optimal one is given by:*

$$\phi_{opt} = \alpha \cdot \frac{2(K+k) \cdot \Delta^{max}}{1 - 4(K+k) \cdot \Delta^{max}}.$$

## III. CASE STUDY: THE WIRELESS FIRE ALARM SYSTEM

In this work, we consider a Wireless Fire Alarm System (WFAS) in the sense of EN 54-25 [4], which states a set of requirements on wireless fire alarm systems. That is, EN 54-25 is not a definition of a particular standard protocol with, e.g., physical layer or preambles, but requires certain deadlines for alarm propagation and a self-monitoring function which need to be met in order to obtain an EN 54-25 compliance certificate. In the following sections, we present the system, and we model it by the means presented in Section II.

### A. Wireless Fire Alarm System (WFAS)

A WFAS is a wireless sensor network consisting of a central unit, sensors, and repeaters as nodes. Sensors are supposed to send information to the central unit. If the physical distance between a sensor and the central unit is large, repeaters are used to forward messages. The nodes communicate with each other via radio signals over a shared channel using TDMA. There is a communication scheme in this network; a sensor or repeater sends a message intended to a specific node, called its master, during certain slots, while the master is in listening mode. The master role can be assumed only by repeaters and the central unit. Each node other than the central unit is assigned exactly one slot per frame. To distinguish the slot assignments, each node is also equipped with a hardware clock.

There are restrictions on the network size: The maximum number of sensors and repeaters is 126, and the maximum number of repeaters linking a sensor to the central unit is 5. The slot length is 25 time units (ticks) and slots are further divided into certain time intervals as shown in Figure 2.

Sensors and repeaters in the WFAS are typically battery powered and EN 54-25 in particular requires a minimum battery lifetime [4]. The overall energy consumption is dominated by energy consumption during both, sending and listening phases. Therefore, energy efficiency is a prominent issue with EN 54-25 compliant WFAS.

Environment conditions like temperature and battery voltage lead to a variation in the crystal frequencies of the clocks, and thus *clock drift* may be exhibited at sensors and repeaters. The frequency of the clock modules used in sensor and repeater nodes is 32.768 kHz, the accuracy is 20 ppm[1] under

[1]ppm abbreviates 'parts per million'; 1 ppm = $10^{-6}$.

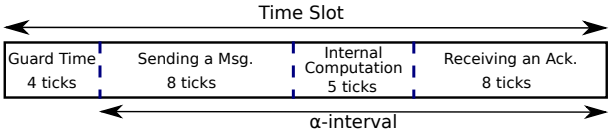


Figure 2: Time Slot of the WFAS.

the environment conditions specified by the manufacturer.

Due to clock drift, a node's clock may indicate that a slot assigned to the node is reached although, according to the reference time, the slot did not yet start or end already. Therefore, we may observe the following critical situations: (1) two nodes send messages at the same time yielding message collision which may inhibit proper decoding of the message at the receiver, and (2) a node sends a message to its master while the master is not listening, yielding message loss.

WFAS's are safety-critical and have to satisfy response deadlines. Since message collision and loss may lead to a communication failure, they have to be avoided. Therefore, clock synchronization mechanisms are required. The WFAS employs clock synchronization as follows: Within each slot assigned to a node, the node sends a message to its master, and the master replies with an acknowledgment within the same slot. The acknowledgment is time-stamped with the master's clock value. The node modifies its clock according to the received time-stamp. The time-stamped acknowledgments propagate the value of the central unit clock to the sensors, which adjust their clocks modules accordingly. This synchronization mechanism cannot guarantee the absence of clock drift for two reasons: (1) Between two subsequent clock synchronizations, clocks continue to drift. (2) Received clock values need not be correct if they are sent by a node which does not provide the reference clock, but in turn received its clock value from a third node.

A bounded clock drift can be tolerated using guard time. Note that the purpose of guard time in the WFAS is only to avoid message collision and those message loss cases, in which a node sends a message while its master it not listening. Message loss due to, e.g., weak signals, reflections, or other users of the frequency band during the $\alpha$-interval is treated as part of the communication protocol and not considered here. Loss of synchronisation messages is considered in Section IV-C.

### B. Modelling the WFAS

We model the WFAS by the means given in Section II. According to the communication scheme between the nodes in the network, the network topology is modelled as a directed rooted tree $T = (N, E)$, where $N = \{cu(T)\} \dot{\cup} Sn(T)$, such that $cu(T)$ (central unit) is the tree root, and $Sn(T)$ is the set of sensors and repeaters. Note that we do not distinguish between sensors and repeaters in the model, since both may exhibit the same amount of clock drift. A topology has a maximum depth of 6, since 5 repeaters at most are allowed to be on one path of the tree topology. The size $k$ of $Sn(T)$ is up to 126. There is no restriction on subtree size $K$, therefore, $1 \leq K \leq k$.

In the WFAS, the domain of the clock values is $\mathbb{N}_0^+$ because time is discretized into *ticks*. However, we model the clock of a node $n$ in an evolution $\mathcal{I}$ at time $t$ by the given function $clk_n^{\mathcal{I}}(t) : \mathsf{Time}\ (= \mathbb{R}_0^+)$, i.e. by real numbers, which results in

precise value of computed guard times. The clock accuracy is 20ppm. Thus, the maximum drift rate $\Delta^{max}$ equals 0.00002.

Since sensors and repeaters are assigned slots bijectively, and clocks are synchronized by the time-stamped acknowledgments, i.e. there is one synchronization point for each sensor and repeater within each of its assigned slots, the evolutions of the WFAS are scheduled and synchronized.

Recall from Figure 2 that the slot length $\alpha$ excluding guard time equals 21 ticks and that the guard time in this system is only one interval added to the beginning of a slot, while the guard time in the model comprises two intervals. We treat this issue in the following section.

## IV. Guard Time Optimization

We apply the formal technique (cf. Section II-B) to derive guard times for several situations of the WFAS. In Section IV-A, we apply Theorem 2 to compute a guard time which is safe for all assignments. In Section IV-B, we show how the optimal guard time can be further reduced when the assignment is restricted to be a best case assignment. In Section IV-C, we relax the assumption that there is no message loss due to communication failures, such as communication delays, and we show how to modify the formal technique to compute an optimal safe guard time for this case. In Section IV-D, we show how the formal approach can be extended for other situations.

Following Theorem 2, the number of nodes (which equals the number of slots) and the topology depth are directly proportional to the optimal guard time. Therefore, we consider a WFAS having the maximum values of its sizes as a worst case wrt. the parameters $n$ and $d$. The maximum number of sensors and repeaters is 126, the maximal topology depth is 6.

### A. Safe Guard Time

Firstly, we directly apply the formal technique [3] to compute a safe guard time as two intervals to the beginning and to the end of each slot, respectively. By Theorem 2, a safe guard time exists iff $\Delta^{max} < \frac{1}{4(d(k-1)+2)}$. Given that $\Delta^{max} = 0.00002$, $k = 126$, and $d = 6$, we obtain

$$\frac{1}{4(6 \cdot (126-1)+2)} = 0.000332447 > 0.00002,$$

thus, a safe guard time exists. By Theorem 2, and given that $\alpha = 21$ ticks, the optimal guard time for $T$ wrt. $\Delta^{max}$ is: $\phi_{opt} = 21 \cdot \frac{2(6(126-1)+2)\cdot 0.00002}{1-4(6(126-1)+2)\cdot 0.00002} \approx 0.67$

The safe guard time is, then, two intervals added to the beginning and to the end of a slot, respectively, where each interval is 0.67 ticks. By rounding this length to 1 tick, the overall length of both intervals is 2 ticks, implying that the guard time employed by the WFAS (4 ticks, cf. Figure 2) is reduced to half of its length.

### B. Guard Time for Best Case Assignments

As the system topology of the WFAS we consider here is not dynamically changing over time, but the result of a configuration process which involves external tools, it is possible to effectively restrict the assignment. For the best case assignments, the size of the maximal subtree $K$ is a parameter in the computation of guard time. We show detailed computations of optimal guard time for best case assignments with two maximal subtree sizes: (1) an assignment where $K = k = 126$; i.e. there is only one subtree in the topology, and (2) where $K = \frac{k}{6} = 21$; i.e. a quite restrictive condition where there are possibly many subtrees, but the size of the maximal subtrees is only about a sixth of the components. We have chosen the given values of $K$ to show how the optimal guard time can be reduced by restricting the size of the maximal subtree.

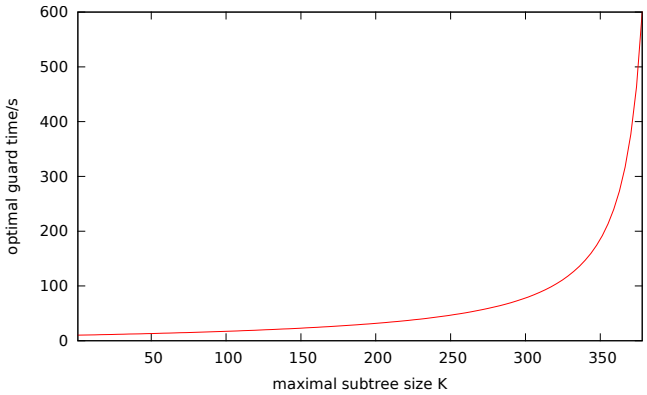


Figure 3: Effect of $K$ on the optimal guard time with fast drifting clocks ($\Delta^{max} = 0.000325$).

Following Theorem 3, the optimal guard time $\phi_{opt}$ for the best cases is as follows: (1) for case $K = 126$, $\phi_{opt} = 0.22$, and (2) for case $K = 21$, $\phi_{opt} = 0.12$. It is obvious that the case $K = 126$ requires (as expected) a larger optimal guard time than the one in case $K = 21$, implying that the required guard time length decreases if $K$ decreases, in the best case assignments.

For a deeper view on the best case assignments, we analyzed the possibility of having a reasonable length of optimal guard time, if the clocks' drift rate is large. Figure 3 shows the relation between the optimal guard time (y-axis) and $K$ (x-axis), when the clocks' drift rate is an order of magnitude worse than in the real WFAS. It is obvious that for large sizes of subtrees, a reasonable safe guard time does not exist (e.g. $\phi \approx 200$ for $K = 350$). On the other hand, if the subtree sizes are small, clock drift can still be treated by a safe guard time that is approximately half of the $\alpha$-interval ($\phi \approx 11$ for $K = 50$), even if the drift rate is this large.

### C. Guard Time Treating Other Sorts of Message Loss

In the former sections, we assumed that there is no message loss due to radio signal issues because we assumed that there is one synchronization point within each slot. In practice, messages with time-stamps can get lost due to e.g. signal weaknesses and delays in message delivery. In such cases, the computed guard times in the previous sections are in general not safe. If one time-stamped acknowledgment is lost then the one in the next frame may reach its target node too early or too late due to continuous clock drift in the node.

Consider a node $n$ at depth $d$ which is not synchronized within its assigned slot. Then, the clock of the node may

| | Sect. | Guard Time $2\phi$ (ticks) | Frame Length (ticks) | Energy Consumption ($Coulombs$) |
|---|---|---|---|---|
| Original | III | 4.00 | 3150.00 | $10.6000{\cdot}10^{-6} + e$ |
| Optimal for any assignment | IV-A | 1.34 | 2814.84 | $3.5510{\cdot}10^{-6} + e$ |
| Optimal for best case assignments, $K = 126$ | IV-B | 0.44 | 2701.44 | $1.1660{\cdot}10^{-6} + e$ |
| Optimal for best case assignments, $K = 21$ | IV-B | 0.25 | 2677.50 | $0.6625{\cdot}10^{-6} + e$ |

Table I: Optimal safe guard times (both, left and right interval) and effect on timing and energy consumption.

continue to drift for one additional frame until the next synchronization point. By [3], this is the same situation as if the node had depth $d + 1$. More general, if $n$ misses all synchronization points during $\ell$ adjacent frames, then $n$ may drift as if $n$ has a depth of $d + \ell$.

By Theorem 2, given a tree of depth $d$, an arbitrary assignment, and an evolution, if any node may miss all synchronization points during at most $\ell$ adjacent frames, then a safe guard time can be computed as follows:

$$\phi_{opt} = \alpha \cdot \frac{2((d+\ell)(k-1)+2) \cdot \Delta^{max}}{1 - 4((d+\ell)(k-1)+2) \cdot \Delta^{max}}. \quad (1)$$

For example, given a topology with $d = 6$, $k = 126$, $\omega = 21$, $\Delta^{max} = 0.00002$, and $\ell = 5$, then by (1), $\phi_{opt} \approx 1.3$.

Interestingly, Equation (1) provides an option to compute the number of subsequent message losses that can be tolerated by a given guard time. For example, given a WFAS with the maximum sizes, and the original guard time employed by the system $2\phi = 4$ ticks (cf. Figure 2), i.e. $\phi = 2$, the number of tolerated subsequent message losses, computed by (1), is 9.

### D. Guard Time Treating only Message Collision

The protocol employed by the WFAS does not utilize guard time to treat message loss caused by clock drift, these losses are treated by the employed protocol together with other communication failures caused by, e.g., other users of the frequency band. Yet the derived guard time in Section IV-A treats message loss caused by clock drift. By Theorem 1, treating message loss requires having smaller bounds on clock drift than the bounds allowed by message collision, which implies that the required guard time for treating only message collision is smaller. An equation for computing optimal guard time treating only message collision can be derived similar to the derivation process explained in [3, Proof of Theorem 2]. We omitted the derivation due to the limited space. The derived equation is as follows:

$$\phi_{opt} = \alpha \cdot \frac{(d(k-1)+2) \cdot \Delta^{max}}{1 - 2(d(k-1)+2) \cdot \Delta^{max}} \quad (2)$$

## V. Energy Consumption

We show a comparison of systems employing the computed guard times in Section IV wrt. the required time and energy. We investigate the time and energy consumption per frame, since a frame is a periodic cycle of a static length and schedule. The energy consumptions by any node during sleep, sending, and listening modes are $60\mu$A, 65mA, and 40mA, respectively.

By fundamental theorems of Physics, an $Ampere$ equals ${}^{Coulombs}/_{Second}$. Since slots and guard time are expressed as ticks, and given the clock speed as ticks per second, we express energy as $Coulombs$ per $Tick$ (CpT). The clock speed is 32.768kHz. The energy consumptions by any node during sleep, sending, and listening modes are, then, $2 \cdot 10^{-9}$CpT, $\approx 2 \cdot 10^{-6}$CpT, and $\approx 1.2 \cdot 10^{-6}$CpT, respectively.

We use the same system parameters as in Section IV: The number of slots per frame $k$ is 126, and an $\alpha$-interval (slot excluding guard time) is 21 ticks. Since we consider a constant length of $\alpha$-intervals, the variable in the comparison is only the guard time length. Therefore, we assume that the required energy during $\alpha$-intervals of all slots in one frame is $e$.

Recall that the number of nodes (incl. central unit) is 127. During the guard time of each slot, there exist two nodes in listening mode, where the other 125 are in sleep mode. Recall that the guard time is two intervals, each of length $\phi$. The required extra energy for guard time during one frame is:

$$2\phi{\cdot}((125{\cdot}2{\cdot}10^{-9}){+}(2{\cdot}1.2{\cdot}10^{-6})) = 2.65{\cdot}10^{-6}{\cdot}2\phi \text{ CpT} \quad (3)$$

Table I presents the required time and required energy by all nodes in one frame, while employing the guard times given in Sections III and derived in Section IV.

## VI. Conclusion

We applied a formal approach for guard time optimization on a wireless fire alarm system exhibiting clock drift. We have shown that the formal approach is useful for quickly deriving an optimal guard time that avoids message collision and loss. In addition, the formal approach is flexible to be modified to treat systems under different conditions, such as the existence of message loss due to signal weaknesses. We have also shown that by adding certain restrictions on the topology and the slot assignment, a guard time of smaller length can be achieved. This, in turn, saves time and energy in such time and energy critical systems.

## References


[1] T.S. Rappaport. *Wireless Communications: Principles and Practice*, volume 2. Prentice Hall, 2002.

[2] E. Perahia and R. Stacey. *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11 n*, volume 28. Cambridge University Press, 2008.

[3] O. Jubran and B. Westphal. Formal Approach to Guard Time Optimization for TDMA. In *RTNS*, pages 223–233. ACM, 2013.

[4] DIN e.V. Fire detection and fire alarm systems - Part 25: Components using radio links and system requirements, German version EN 54-25:2005, 2005.