Micro Protocol Based Design of MacZ - A Highly Adaptive, Integrated QoS MAC Layer for Ambient Intelligence Systems

T. Kuhn, I. Fliege

Technical Report 347/05

Micro Protocol Based Design of MacZ - A Highly Adaptive, Integrated QoS MAC Layer for Ambient Intelligence Systems

T. Kuhn, I. Fliege

Computer Science Department, University of Kaiserslautern, Kaiserslautern, Germany
{kuhn,fliege}@informatik.uni-kl.de

Technical Report 347/05

Computer Science Department
University of Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Germany

Technical Report

Micro-protocol based design of MacZ – A highly Adaptive, Integrated QoS MAC Layer for Ambient Intelligence Systems

T. Kuhn, I. Fliege

This report introduces MacZ, a hardware-independent MAC layer with QoS capabilities. The focus is on the basic layer of MacZ that provides common services, in particular distributed multihop synchronization, signaling of alert messages and conflict detection and resolution services. The intended application domain for MacZ is within Ambient Intelligence, where small, energy-efficient, reliable and adaptive communication networks are required. Besides the hardware independent description, a mapping to an existing hardware using an IEEE 802.15.4 compliant transceiver chipset is performed. MacZ consists of several components that model distinct protocol functionalities. The components are specified as micro protocols in SDL and composed using a micro protocol framework.

1. Introduction

1.1. Motivation

Wireless networks have to serve a variety of purposes. As of now, more and more different types of wireless networks arise – for example wireless LAN networks, wireless AmI networks, and networks that are designed to operate on a very short distance with a high bandwidth. All types of these networks are relevant in the domain of *ambient intelligence* [LWS04]. Ambient intelligence is about to create an intelligent, yet mostly invisible environment. The heterogeneity of this environment encourages the use of very different types of nodes, depending on the tasks that they are performing.

Our work focuses on wireless ad-hoc networks for ambient intelligence systems. In these networks, the nodes have very scarce resources. However, there are many possible communication scenarios, ranging from the transmission of infrequent sensor data to the transmission of multi-media traffic, like audio communication. Also these networks comprise various types of nodes, differing in to their hardware and energy resources as well as in the applications that they are executing.

This work describes MacZ, a QoS-MAC layer that can be used as underlying technology for ambient intelligence networks. Most current MAC layers for wireless ad-hoc networks use a contention-based media access scheme. For offering a medium that is also capable of providing QoS-services, we decided to create a time-synchronized medium that can be divided into contention-based and into contention-free periods to adapt to the needs of an application.

Since this report describes work in progress, only the finished parts of MacZ are described. These finished parts are the basic structure of MacZ, the time synchronization mechanism, the announcement mechanism for specific events and the conflict detection and resolution. Each of these functionalities forms a logical component, which has been specified as a *micro protocol* [FGGS05], a structuring technique for protocol development that has been developed by our networked systems group at the University of Kaiserslautern.

1.2. Related work

In this section, we survey work related to MacZ, the QoS MAC layer presented in this report. We structure this survey into approaches to time synchronization and medium access control in wireless networks. Media access schemes that use contention based reservations only, that provide no time synchronization among multiple hops, or that require specific hardware like transceiver chips with configurable frequency hopping or modulation schemes are omitted in this survey.

1.2.1. Time synchronization in wireless networks

Multiple methodologies exist for achieving time synchronization in wireless networks. One methodology is the server based synchronization – one server transmits its current clock value through the network synchronizing all other nodes to its clock value. NTP [Mil94] is a protocol for server based clock synchronization. Although it is widely used in the Internet, it has also disadvantages that become especially visible in wireless networks. NTP needs to transmit the current clock value of one or multiple servers through the network. This makes

the protocol vulnerable to variances in the time required for accessing the wireless media. Depending on the MAC-protocol being used, these variances can be in the range of several milliseconds [IEEE03] in wireless networks, yielding those protocols that need to transmit the current clock value of one node unusable for achieving synchronization at a microsecond scale across multiple hops.

Another possibility for achieving clock synchronization across a network is to use external clock receivers, like GPS receivers that also provide a highly accurate clock. Unfortunately, GPS receivers require a substantial amount of energy, a clear sky view and an additional receiver. This renders GPS unusable for most time synchronization tasks in wireless networks.

The methodologies mentioned above attempt to achieve global clock synchronization between the nodes of a sensor network. An alternative to global clocks are virtual clocks [Lam78]. Lamport proposes virtual clocks for systems where the ordering of events is more important, than the absolute time of an event. For the design of a distributed MAC layer, the global time is also not relevant. Rather there is the need that all nodes have an accurate time scale that is relative to a specific point of time that is equal to all nodes.

A methodology that achieves distributed time synchronization without providing global clock synchronization is described in [EGE02], introducing a technique called "reference broadcasts" which synchronizes all receivers around a transmitter node to each other. To avoid jitter with contention based medium access, only the nodes receiving the broadcast at almost the same time are synchronized – a transmitter node is not synchronized by its own broadcast. After receiving a reference broadcast, all nodes exchange their observations about the reception time of the broadcast. This can be used to create a function for every pair of nodes that converts clock values from one to another node.

1.2.2. Medium access control in wireless networks

Medium access in wireless AmI networks is usually determined by saving as much energy as possible. Since most nodes have only scarce energy resources, saving energy is an important point. Energy can be saved by reducing the time the transceiver chip of every node has to be active. Retransmissions of packets also cost large amounts of energy at the sender and at the receiver, so collisions – the main source for retransmissions, are to be avoided as good as possible.

Current MAC-protocols for sensor networks rely at least partially on contention based media access. The SMAC protocol presented in [YHE02] is a contention based MAC-protocol that reduces the energy consumption of nodes by sending them periodically into a sleeping state. Nodes in 1-hop neighborhood synchronize their sleeping periods to each other. Every node has only a short period where it wakes up to see if any other node wants to talk to it. [CC02] proposes a medium access protocol called TBMAC that mixes space division with time division. All hops within a 1-hop cluster form a cell. Within each cell, a time-slotted medium is used, while different clusters use different channels for communication. Specific time slots are used for performing inter-cell communication. [CC02] relies on additional techniques for obtaining time synchronization and for obtaining the current position of every node.

The work presented in [PDÖ02] uses black bursts for providing a contention based accessing scheme for wireless networks. The described MAC protocol supports a deadline based medium access scheme for ad-hoc networks. The description is based on wireless LANs, but the basic techniques could also be adapted to wireless AmI networks.

ZigBee [IEEE03] has the ability to provide a contention-free period. This is implemented by a master node sending a beacon that is followed by a fixed number of slots. Between the last slot and the next beacon, there is a period of time for contention-based access or beacons from other masters. This beacon-based synchronization only works within the 1-hop range of a master node. The point coordinator functionality of 802.11 [IEEE97] realizes also a very basic slotting of the wireless medium by having one station polling the other nodes. However contention-free and contention-based access periods are mixed with a nondeterministic switching time, the methodology depends on exactly one master node and it is not usable across multiple hops. GAMA-PS presented in [MA98] does offer contention-free access without having the need of synchronizing the network. However, GAMA-PS requires a fully connected network, so every node has to be able to receive the transmissions of all other nodes to maintain its reservation tables. This makes this protocol unusable for AmI networks that usually span multiple hops and contain nodes that periodically turn off their transceiver chipsets.

1.3. Objectives

Our QoS MAC layer MacZ was designed with multiple applications in mind. To serve a high variety of applications, reservation-based contention-free access should be possible, as well as contention-based medium access, for nodes and applications that do not produce regular traffic. Therefore, MacZ creates a time-synchronized virtual medium. This medium can be divided into periods with contention-based and contention-free access, supporting both types of possible traffic. The partitioning is variable in size, so it can be adapted to the needs of a specific network.

In the domain of ambient intelligence networks, the propagation of alert messages might also be one of the tasks of a network. Therefore, this MAC layer supports a special technique for propagating alert messages with high priority through the network. This mechanism can be seamlessly integrated into the energy saving strategies of every node due to the time-synchronization of the medium. By adapting the frequency and the position of these signaling slots, a developer can adapt the propagation characteristics to its specific needs.

Another objective of MacZ is high reliability and the possibility to operate without a fixed infrastructure. Therefore, two different synchronization algorithms are presented later in this work, each algorithm with very different characteristics. This also demonstrates the modularity of MacZ. We have used micro-protocols for specifying the behavior of every protocol component of the MAC layer. This causes the components of MacZ to be exchangeable and largely independent of each other. As a result, the developer might exchange the synchronization algorithms that are presented in this work with different algorithms without having to adapt the functionality of the other algorithms, as long as the documented constraints are still met.

1.4. Outline

Section 2 gives a short overview on the general design of MacZ. Section 3 introduces some basics of wireless medium access and transmission techniques. Section 4 introduces multihop synchronization and common techniques for achieving it. In Chapter 5, the main part of this report, the various functionalities of MacZ are presented together with their micro-protocol based implementations. Section 6 shows an example for the instantiation of MacZ on a realistic hardware platform. Section 7 draws the conclusions and indicates future work.

2. Overview of MacZ

This section documents the general design of and the general ideas behind MacZ, our QoS MAC layer for AmI networks. The detailed description of its components and algorithms will be presented in Section 5.

2.1. Design rationale

The main idea behind our MacZ is, to create a MAC layer that supports a variety of access mechanisms while still remaining robust to network topology changes. This is achieved by using a fully distributed approach for node synchronization. Unlike in many wireless and cellular networks, no designated master node is required. This makes many tasks particularly challenging, because no central master is available to create and to control the synchronized medium or to resolve conflicts. This has to be done by fully distributed algorithms, which will be presented later in this work.

Since in the domain of ambient intelligence a variety of applications are possible, MacZ must be able to adapt itself to the needs of different applications and node types. Therefore, MacZ supports a time-synchronized medium that the developer may fill with different slot types, for example for signaling, for contention-based or for contention-free transmissions. Section 2.2 will regard this in more detail. Also the implementation of the MAC layer itself is modular. Micro protocols have been used for specifying the behavior of the different, independent protocol components. A micro protocol is a communication protocol with a single (distributed) functionality and the required protocol collaboration [FGGS05]. The modularity facilitates the exchange of protocol units for different application needs. For example, the micro protocol used for multi-hop synchronization could be changed to a simpler one, if a network for a single hop scenario is being created.

To achieve the variety of functionalities, two different transmission methodologies are being used at once: Black bursts and regular packets. Black bursts are transmissions, whose only information is their length. Since the actual transmitted data is meaningless, these transmissions are resistant to collisions on the medium. There is a specific protocol functionality described below that takes care of receiving and decoding black bursts, including the detection of duplicates due to timer drift of the transmitting nodes. Most functionalities of the MacZ basic layer (see Figure 1) solely rely on the transmission and reception of black bursts, while the functionalities of MacZ service layer primarily rely on the transmission and reception of regular packets.

2.1.1. Basic structure of MacZ

MacZ is structured into two layers; the MacZ basic layer and the MacZ service layer (see Figure 1). The MacZ basic layer synchronizes the medium and reports the current media state to the MacZ service layer. The MacZ service layer handles the transmission of the network traffic and uses therefore the services provided by the MacZ basic layer. As indicated in Figure 1, the MAC service layer provides services required for applications to transmit various types of data.

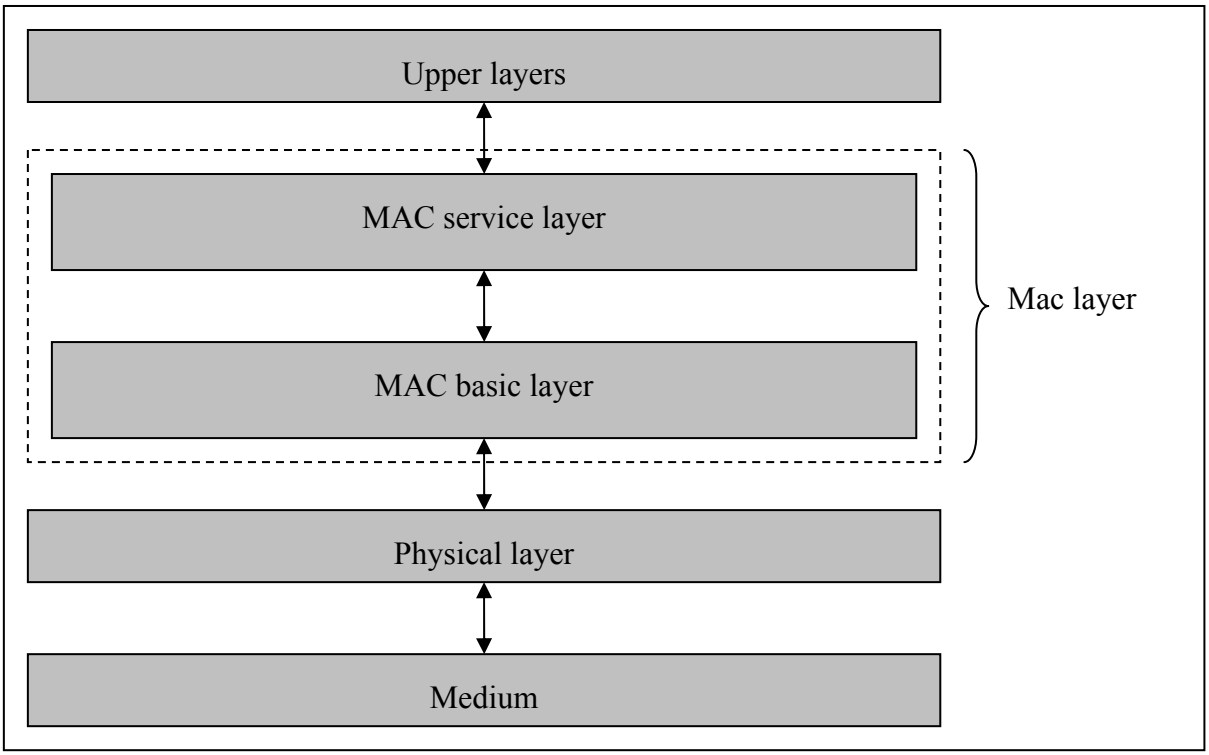


Figure 1: Basic structure of our QoS-MAC layer

The MacZ basic layer provides basic functionalities to the MacZ service layer. These functionalities are:

- Multi-Hop synchronization
  Time-synchronizes the whole network across multiple hops, creating the premises for a synchronized medium.

- Signaling of special events to all nodes
  Signals the pending transmission of specific, high priority messages to nearby nodes.

- Conflict detection
  Detects unsynchronized networks and nodes.

- Conflict resolution
  Attempts to resynchronize an unsynchronized network or to join two networks together.

The MacZ service layer provides more advanced functionalities that depend on the MacZ basic layer. It is also required for bootstrapping the synchronization process, because therefore, a defined set of masters must be selected and synchronized. The set of offered services by the MacZ service layer strongly depends on the state of the MacZ basic layer. As long as the MacZ basic layer is unsynchronized, only the following services are available:

- Election of masters (available only to MacZ service layer)
- Transmission of alert messages
- Transmission of generic, low-priority messages

The need of the master voting functionality depends on the implementation of the MacZ basic layer; the other services are always required. Once the MacZ basic layer is in synchronized state, the following additional services are offered:

- Electing replacements for lost masters
- Signaling of high priority alert messages
- Contention free, reservation based access
- Contention based access with collision avoidance
- Energy saving strategies

As a result of the decomposition of the MacZ layer into a service layer and a basic layer, further service layers for different purposes could be developed on top of the basic layer.

The MacZ basic layer falls back to unsynchronized state if the synchronization of the medium fails for any reason. Depending on the used synchronization algorithm, this could be the case if all elected masters fail at once and no re-election is possible, or if other networks produce interferences that prevent the MacZ basic layer from synchronizing correctly. In these cases, the synchronization is lost; the MacZ basic layer attempts to join the networks together and then starts the re-synchronization sequence whose implementation depends on the used synchronization algorithm.

2.1.2. Hardware independent design

For the development of MacZ, a clear separation between the MAC layer and the physical layer was made. As a result, the whole MacZ layer is described in a hardware independent manner, structured by the use of micro protocols. Protocol functionalities are defined as separate micro protocols, forming the functionality of this MacZ basic layer as a macro protocol.

Although the design MacZ is independent of a specific hardware platform, possible limitations from real hardware platforms must be considered. For example, real platforms are limited with respect to their computation resources, resulting in *delay* and *jitter* when processing received signals. These effects also influence the possible reaction time to events. So, whenever possible, the required reaction time to events should be as variable as possible, to support a variety of hardware platforms and implementation methodologies.

Another constraint is the used transceiver chip or transmission technique. The interface to the radio hardware must provide at least the following basic functionalities for being able to be supported by MacZ:

- Sending with clear-channel assessment
- Sending without clear-channel assessment for supporting black bursts
- Reception of ordinary packets
- Notification about the current media state, whether it is idle or busy

For being able to support the reception of black bursts, the interface must either provide a low delay and jitter signaling for the change of the media state, or the signals for the media state changes must contain accurate time stamps. This is necessary, because the length of a black burst must be accurately determined. It must also be possible to transmit a packet of a specific length at a specific time. The more accuracy of these timings can be guaranteed, the better the performance of MacZ will be on this platform. Section 6 describes the implementation of our MAC layer on a real hardware platform.

2.2. Synchronized medium

One requirement of the MAC layer is the variety of applications that are to be supported. Also, the ability to save energy is crucial, because many nodes will have very limited energy resources. On the other hand, it must be possible to reserve bandwidth for high-priority and multimedia applications. This calls for a time synchronized medium that allows idle nodes to sleep most of their lifetime.

Our time synchronized medium is divided into micro- and macro slots (see Figure 2). Macro slots have a defined length $T_{MacroSlot}$. They always start with a synchronization slot, followed by micro slots which could have signaling and transmission slots assigned to them. The assignment whether a micro slot is used for signaling or for transmissions is taken by the developer. Transmission slots are usable by the MAC service layer for either contention-based or contention-free traffic. The possibility of changing the slot distribution can radically change the behavior of the network, with respect to possible reaction time on alert messages and network synchronization errors. It can also change the energy consumption of nodes.

MacZ creates a synchronized medium by time-synchronizing all nodes within a wireless network. This is achieved by a combination of synchronization-, signaling-, and transmission slots. The following Figure 2 shows one possible distribution of the different slots on the medium.

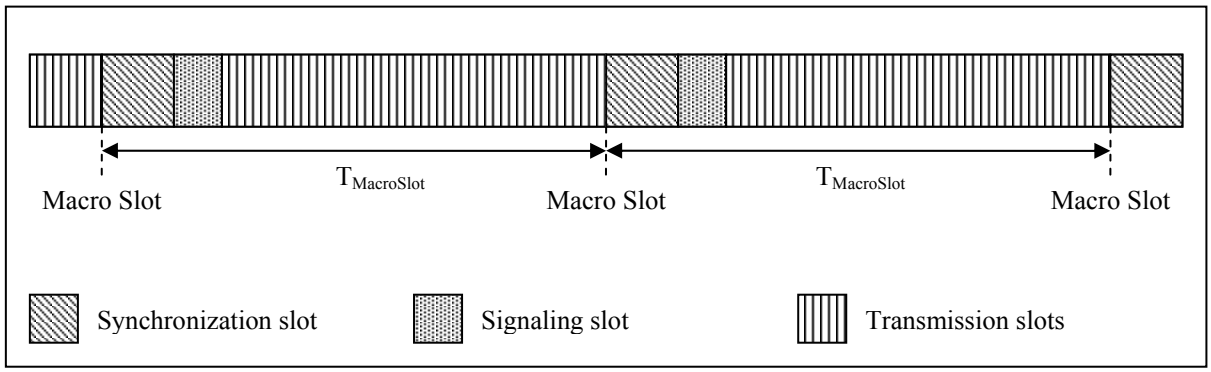


Figure 2: Example slot distribution

As shown in the example presented in Figure 2, the medium is basically divided into three types of slots. The synchronization slots are used by the multihop synchronization functionality to ensure that all nodes have synchronized timers – within specific constraints. The signaling slots are used for two purposes: Conflict resolution, which will be laid out further in Section 5.8, and the signaling of alert messages. The remaining time is filled with transmission slots. The MacZ service layer can decide how these slots are to be filled – they can be filled with (possibly priority based) best-effort traffic, with reservation based contention-free traffic slots, or with a mix of both.

All nodes must be awake during their signaling slots, to ensure that synchronization errors and alert signals will be received by every node. Afterwards, every node may decide to sleep during the transmission slot. This decision usually depends on the needs of the upper layers. One should note that the slot distribution shown in Figure 2 is only one example for aligning the different slot types. To facilitate adaptation, the distribution of these slots may be changed by the developer to reflect the needs of the used hardware or the needs of the domain that MacZ is about to work in.

2.2.1. Modularity

To provide maximum adaptivity, the developer may align the slots that serve different purposes to reflect the specific needs of the network. It is also possible to replace the component that provides, for example, the network synchronization without having to modify other components. This provides potential for adapting MacZ to different networks, for example, to single hop scenarios.

To achieve this modularity, every component of MacZ must be solitary. Concretely, this means for this case, that no slot may depend on the specific location of another slot of a different type. This ensures that developers may move the locations of slots around, i.e. for providing better reaction times or energy savings. However, every type of slot must be present at least once in every macro slot to ensure the functionality of MacZ.

3. Wireless-medium access

For clarifying the specific problems that arise with medium access for wireless networks, the basics of wireless transmissions must be understood. This section outlines the basics of wireless medium access and explains a special transmission technique, called “black bursts”.

3.1. Wireless media access

When designing wireless networks, it must be taken into account that a transceiver chip can only be either in sending or in receive mode. Only in receive mode, the medium state can be monitored, for example for detecting foreign transmissions. The monitoring of the medium state is called *clear-channel assessment* (CCA). Usually, this information is available after the transceiver chip has been in receiving mode for a certain amount of time. In transmission mode, there is no possibility for detecting collisions on the medium. Transmissions must be either resistant to collisions – this is the case for black burst transmissions, or some sort of collision avoidance must be used. Collision avoidance is complicated by the fact that there is a period of time where a node can neither send nor receive anything between mode changes of its transceiver chip. Also, the clear-channel assessment is not working during this period. This *blind period* may occur when changing from sending to receive mode or when changing from receive to sending mode and may have a different length for each change. The duration of the blind period strongly depends on the used physical layer and raises the chance of collisions, because the node that is about to transmit data cannot detect transmissions that started in its blind period and other nodes cannot see its transmission yet. Figure 3 illustrates the different operation modes of a wireless transceiver chip.

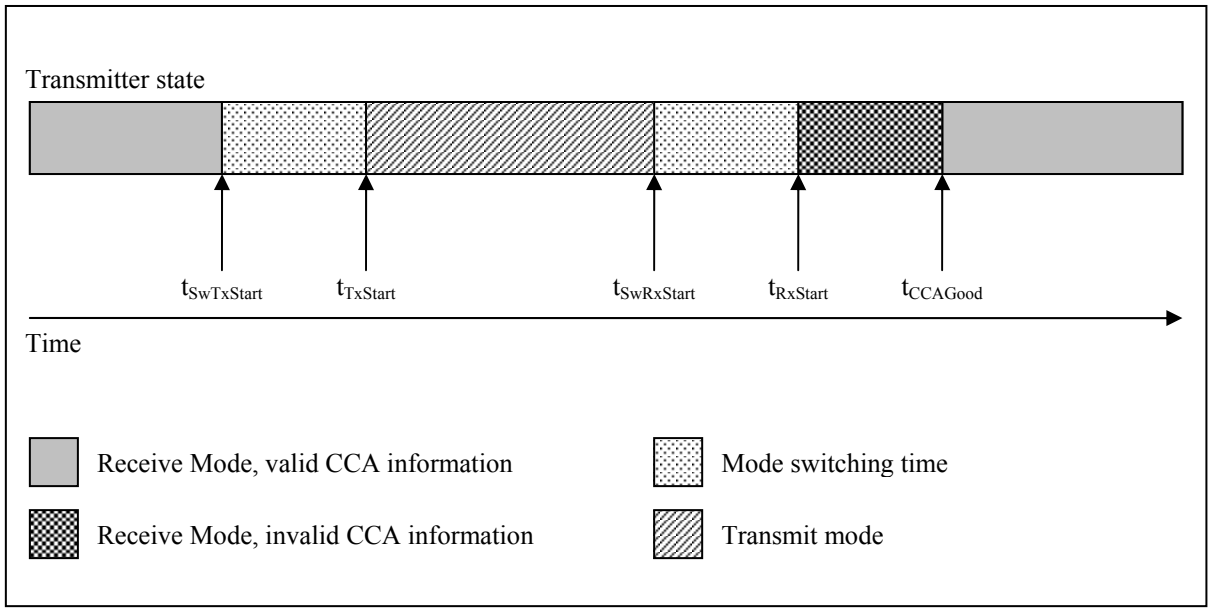


Figure 3: Operation modes of a wireless transceiver chip

Two time periods can be identified in Figure 3: The time that is required for changing from transmission to receive mode, $T_{SwitchRX}$, and the time that is required for changing from receive to transmission mode, called $T_{SwitchTX}$. The length of $T_{SwitchRX}$ is defined as the duration between the beginning of switching to receive mode ($t_{SwRxStart}$) and the point of time when the clear channel assessment information becomes valid ($t_{CCAGood}$). As a result, $T_{SwitchTX}$ and $T_{SwitchRX}$ are defined as following:

- $T_{SwitchTX} = t_{TxStart} - t_{SwTxStart}$
- $T_{SwitchRX} = t_{CCAGood} - t_{SwRxStart}$

So there are two fundamental problems that developers of MAC layers for wireless networks have to cope with: It is impossible to directly detect collisions on the wireless medium, and there are periods of time in which the state of the medium is unknown to a node. These periods of time are before, during, and shortly after a transmission, as indicated in Figure 3. This makes the design of medium access strategies for wireless networks a challenging task.

Normally, a collision avoidance scheme is used in wireless networks. Basic collision avoidance schemes select a random number between defined borders that is decreased as long as the medium is idle for a specific period of time, called “slot time”. For every slot that the medium is idle, the counter is decreased by one. The slot time is selected to be greater than $T_{SwitchTX}$ to avoid that nodes finish their counter while another node is already switching to transmit mode. The probability of collisions decreases with increasing range of the random numbers. Although collisions are avoided by using this technique, this unpredictable delay, called “contention window”, is very disturbing for time synchronization. A transmission technique, that is resistant to collisions, at least to a certain extent, is the transmission of black bursts. This technique will be described in Section 3.2.

3.2. Black bursts

Frames that are transmitted regardless of their content are referred to as “black bursts”. The only information that a black burst carries is its length. Unlike regular frames, the payload of a black burst is not considered to be relevant. Although black bursts are a considerable waste of bandwidth, due to the limited amount of information that can be transmitted by them, they

also have significant advantages over regular transmissions. Black bursts are resistant to collisions as long as the length of the burst is not significantly changed – so several nodes may transmit a burst at the same time and it is still ensured, that every receiving station is able to understand the burst. Another advantage of black bursts is their increased transmission range compared to regular transmissions. Since it is not necessary to decode a packet correctly for receiving a black burst, black bursts can be transmitted over a much greater distance than conventional frames. These properties make black bursts an interesting mechanism for synchronization and for emergency signaling.

Drawbacks of black bursts include the collision handling. Although black bursts are quite resistant to collisions, because the data that they transmit is meaningless anyway, there length might be changed significantly when two or more black bursts collide. This will change their length, and as a result of this, also the information that they are carrying. Figure 4 illustrates this problem.

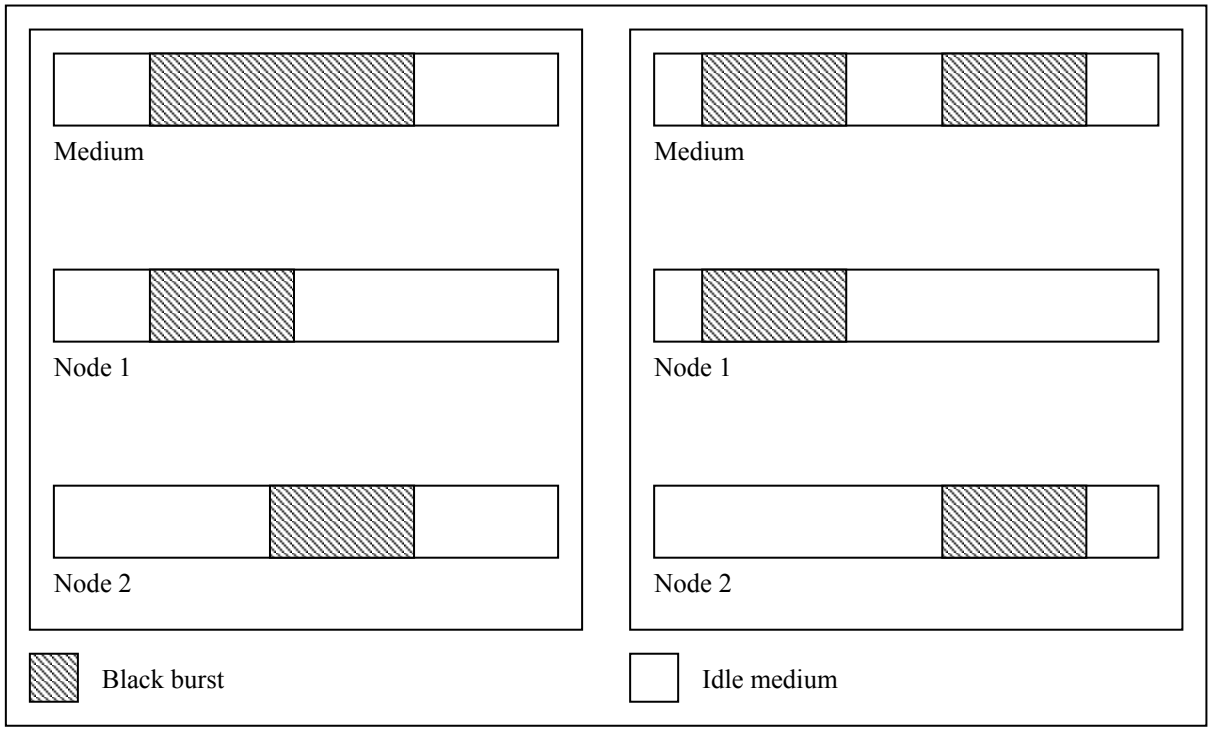


Figure 4: Possible effects of timer drift to black burst transmissions

As indicated in Figure 4, there are two possible issues that may rise with black bursts. Two bursts that are to be sent at the same time may either be received as two separate bursts, due to timing drift, or they may be melted to a longer burst. Since more than two nodes may be transmitting a burst at the same time, the length of the burst may vary between the original length of the black burst D_{Burst} and $D_{Burst} + T_{MaxDrift}$, where $T_{MaxDrift}$ is the maximum tolerable timer drift among all transmitting nodes. The MAC layer has to ensure sufficient time synchronization of all nodes.

Another issue is the separation of one black burst into multiple bursts. This might happen if two nodes are transmitting a black burst at the same time, with a timer drift that is greater than the length of the black burst. As a result, receiving nodes will see two black bursts with a certain idle time between them. This must be handled by the components that are receiving the black bursts as well as the possible length increase due to overlapping burst transmissions.

4. Multihop synchronization

This section describes the general ideas of our two multihop synchronization algorithms. The exact specifications and implementations are presented in section 5.

For the creation of a synchronized medium, sufficient synchronization of all hops in the network is mandatory. Our multi-hop synchronization also has to cope with the constraint that most nodes in the network will eventually be asleep for most of the time. So there must be periods of time, called synchronization slots, where all nodes are awake and where all nodes are synchronized to ensure, that the synchronization drift of a node never exceeds the maximum allowed drift.

We decided to use physical bursts on the medium as our synchronization methodology as it was proposed in [EGE02]. In [EGE02], the receiver nodes exchange the information about the reception time of a frame to achieve a function for converting times between nodes. Since our only concern is to synchronize all nodes to a specific point of time, the exchanging of reception time is not necessary. The main idea of our synchronization algorithms is that every node starts a timer on the reception of a packet that controls the time division of the medium. This way, all receiver nodes within the 1-hop range of a transmitter node can be synchronized. However, the transmitter node itself will not be synchronized due to the unpredictable jitter that might be added by the randomly selected contention window.

In practice, there are some problems that arise: There is always jitter due to contention window or processing delays, the synchronization depends on one node transmitting the synchronization frame and wireless networks usually span multiple hops. We will explain in the following paragraphs, how our two synchronization algorithms overcome these problems.

The first problem for time synchronization that arises is the difference between the point of time when one node starts to transmit a packet and the point of time when another node is notified of its reception. This period of time has a specific duration and a specific unpredictable jitter which originate from four effects:

- MAC delay at the transmitter: This is the time that is required for creating the packet, for creating the packet header and for transmitting the packet to the transceiver chip. Since we do not assume preemption during this activity, this time can be considered as being almost constant.
- MAC accessing delay: This period of time is highly variable and depends on the value of the contention window that is randomly selected by the MAC layer.
- Propagation delay: This is the time that the wave transmitting the signal requires to get from one node to another. Although the value of this delay theoretically depends on the range between two nodes, it is assumed to be constant, because its value is small enough to be not significant.
- MAC delay at the receiver: This is the time that is required for receiving the packet header and for either notifying the upper layer or for time stamping the packet. If the packet is time stamped in an interrupt routine, this time can also be considered as constant.

So there are three major factors that add a constant value to the time that passes between the transmission request at the sender and the receive event, and one highly variable time, the MAC accessing delay. The constant times are known and can be accounted for during time

synchronization. If it would be possible to eliminate the MAC accessing delay, it would be possible to synchronize a set of receivers not only to each other, but also to the transmitter node. To eliminate the MAC accessing delay, we decided to use black bursts. Black bursts carry only their length as information, while their payload is not significant. Therefore, they are resistant to collisions, because a collision with a packet of the same length renders the payload unusable, but the length of the packet is not changed. So if two nodes transmit a black burst with the same length at roughly the same point of time, the message will still be understood by all other nodes. Since black bursts are resistant to collisions, no media contention is necessary – so the MAC accessing delay is eliminated by our synchronization algorithms.

Our synchronization algorithms follow two different paradigms for overcoming the problems of multihop synchronization and of depending on one master. We have developed two different algorithms. The first algorithm depends on a set of pre-elected masters that transmit different sequences of black bursts. Every sequence has a different priority, synchronizing the entire network to the node transmitting the sequence with highest priority. The second algorithm does not depend on master nodes – with this algorithm, all nodes are synchronized in a completely distributed manner. The specifications of these algorithms are presented in sections 5.3 and 5.4.

5. Services of the MacZ Basic layer

This section describes the details of the functionalities and the realization of the MacZ basic layer. First, the overall structure is described. Then, the concrete realization of all components is documented. All algorithms are self-contained – as a result, each algorithm of the MacZ basic layer may be replaced by another algorithm that realizes the same functionality in a different manner. This can be used to adapt MacZ to very specific needs of special application domains.

5.1. Structure overview

Basically, four different services build up the functionality of the MacZ basic layer: The synchronization service, the signaling service, the conflict detection, the handling of black burst transmissions and the conflict resolution by transmitting a jamming sequence. Figure 5 outlines the basic structure of the MacZ basic layer.

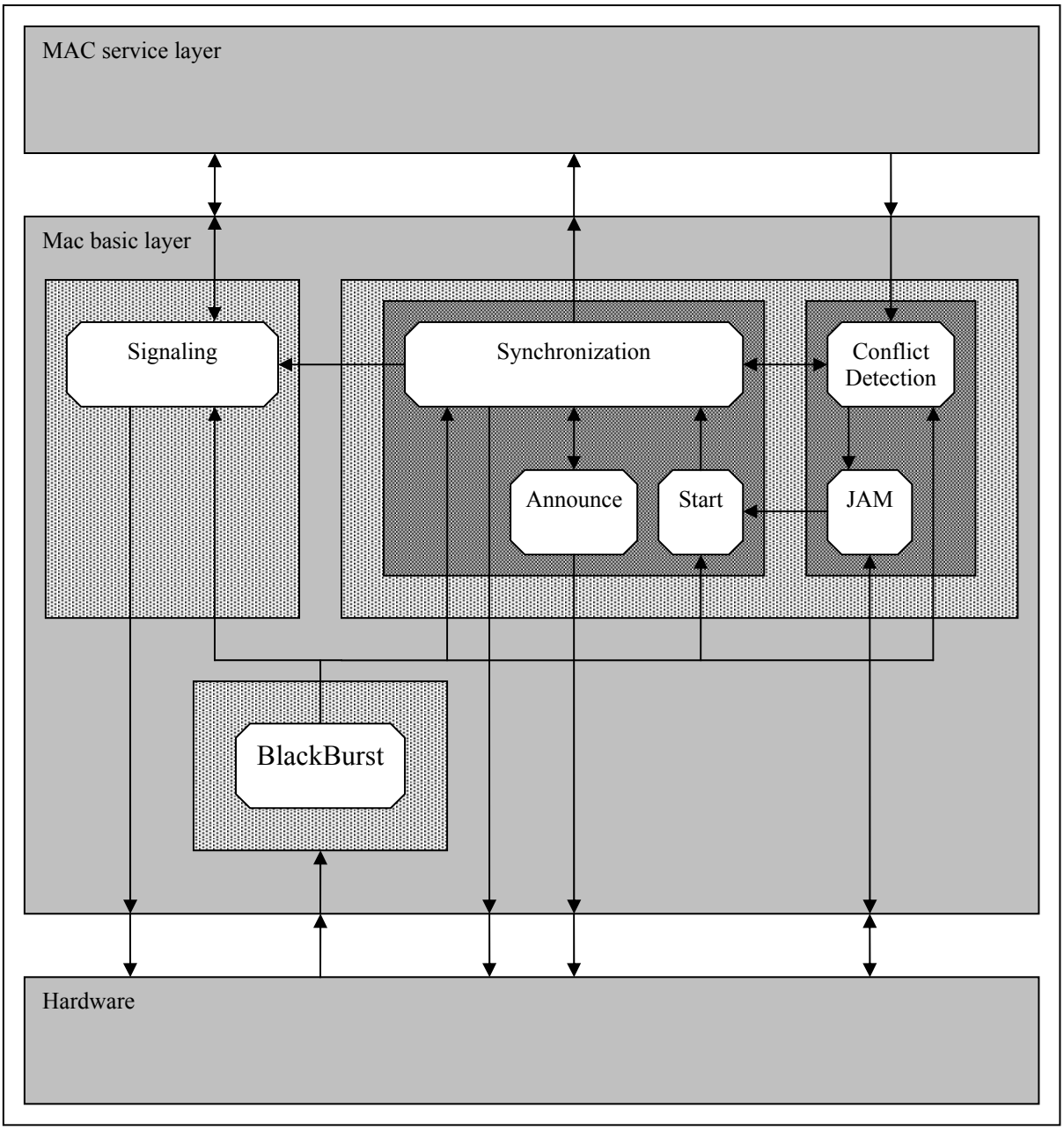


Figure 5: Structure of the MacZ basic layer

As shown in Figure 5, the services of the MacZ basic layer are grouped together into four groups. The protocol functionalities “Synchronization”, “Announce” and “Start” form the synchronization group that takes care of synchronizing and re-synchronizing the medium. The functionalities “ConflictDetection” and “JAM” form the conflict handling group that detects and propagates conflicts. The functionalities “Signaling” and “BlackBurst” form the remaining two groups. Each of these functionalities will be described in detail in the following, specified as a self-contained micro protocol.

5.2. Handling of black bursts

One integral technique that has been used for nearly all of the following algorithms is the transmission and reception of black bursts. This section describes the micro protocols that have been specified for sending and receiving black bursts.

5.2.1. General description

The transmission of a black burst is done by sending a frame of a specific length without checking the clear-channel assessment information of the medium. The reception and decoding of a black burst is more complex, because the timer drift of a node might change the recognized size of a specific burst. The actual size of a black burst on the medium may also vary; this depends on the timer drift of the transmitting nodes. When multiple nodes with a timer drift of $T_{MaxDrift}$ transmit a black burst at the same time, the length of the burst might be increased by the value of $T_{MaxDrift}$. If $T_{MaxDrift}$ is greater than the length of the transmitted burst, one burst could be visible as two separate bursts on the medium. The micro protocol for receiving a black burst must be able to handle these effects, and it must also be able to correctly decode the type of a specific black burst. Also, it must be able to distinguish a black burst from an ordinary frame. Currently, duplicate black bursts are detected by the short pause time between them – the protocol functionalities that transmit black bursts must ensure that the pause time between two regular transmitted bursts is long enough to ensure that this micro protocol is able to detect the second transmitted black burst as an independent burst.

5.2.2. Types of black bursts

MacZ uses two different types of black bursts, for signaling different types of events. Each of the two burst types represents, depending on its length, either a 0 or a 1. The length of bursts with ID 0 and ID 1 is defined as D_{Burst0} and D_{Burst1}, respectively. The following constraints must hold for the two types of black bursts.

- $D_{Burst0} > D_{Burst1} + T_{MaxDrift} + 4*T_{HWJit}$
  Both types of black bursts must be clearly distinguishable from each other, so their length difference must be greater than the four times timing jitter of the hardware platform, since the beginning and the end of both bursts must be measured, plus the maximum possible synchronization jitter, since multiple nodes might send a burst at the same time (see Figure 6 – the overlapping short bursts of Node 1 and Node 2 with maximum timer drift $T_{MaxDrift} + 4*T_{HWJit}$ are shorter than the long burst that is transmitted by Node 3). D_{Burst0} represents the more dominant burst and must therefore be the longer one.

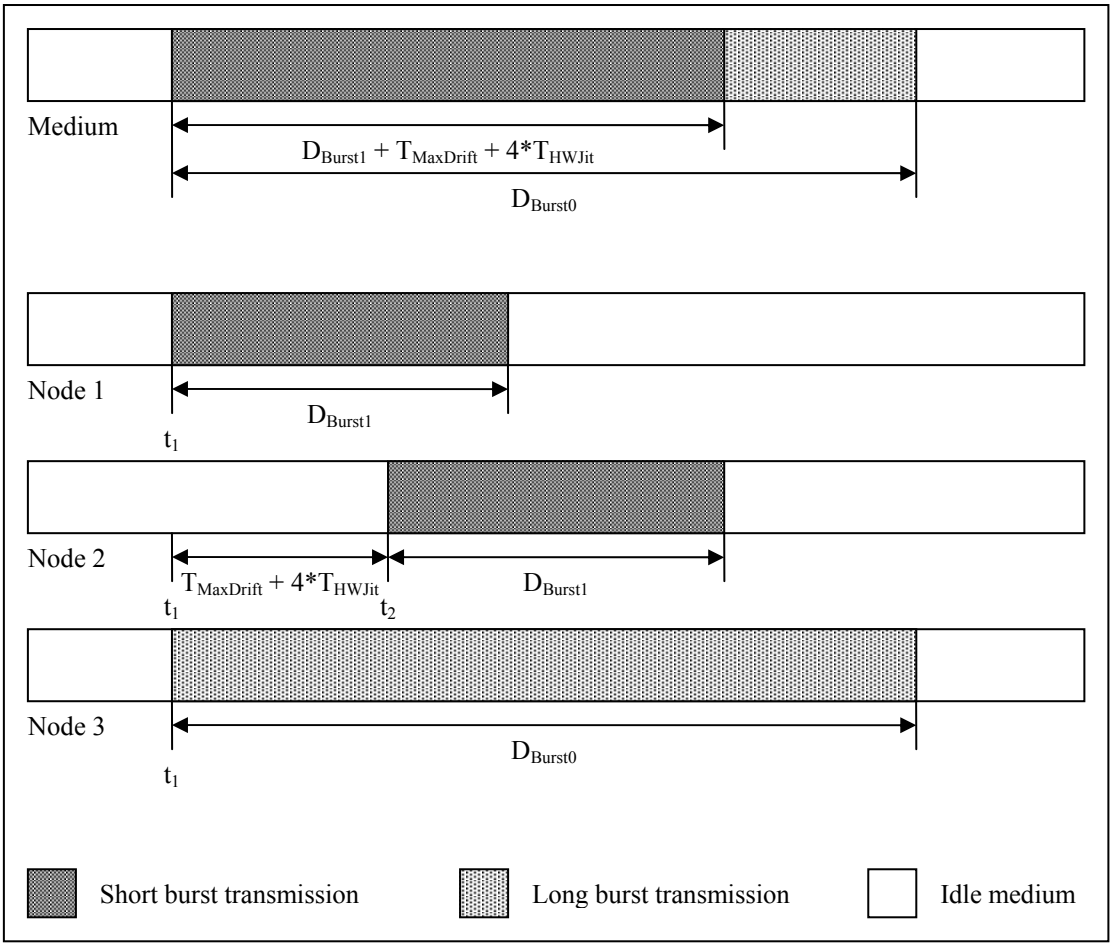


Figure 6: Overlapping black bursts

- $D_{Burst0} > D_{Burst1} + T_{SwitchRX} + T_{MaxDrift} + 4*T_{HWJit}$
  This ensures that a node is able to detect a burst with ID 0, even if it started transmitting a burst with ID 1 at the same time. The tolerable synchronization jitter must also be considered when selecting the length of D_{Burst0} and D_{Burst1}, because the drift of the transmitting nodes will be between 0 and the maximum synchronization jitter.

- $D_{Burst0} + 2 * T_{HWJit} + T_{MaxDrift} > D_{MinFrame} - 2 * T_{HWJit}$
  All black bursts must be clearly distinguishable from regular frames, whose minimum length is referred to as $D_{MinFrame}$. This must still hold, if multiple nodes start transmitting a maximum length burst with the maximum tolerable synchronization drift.

5.2.3. Micro protocol design

Figure 7 shows the micro protocol design that encapsulated the black burst decoding and duplicate detection functionality, specified with SDL [SDL100].

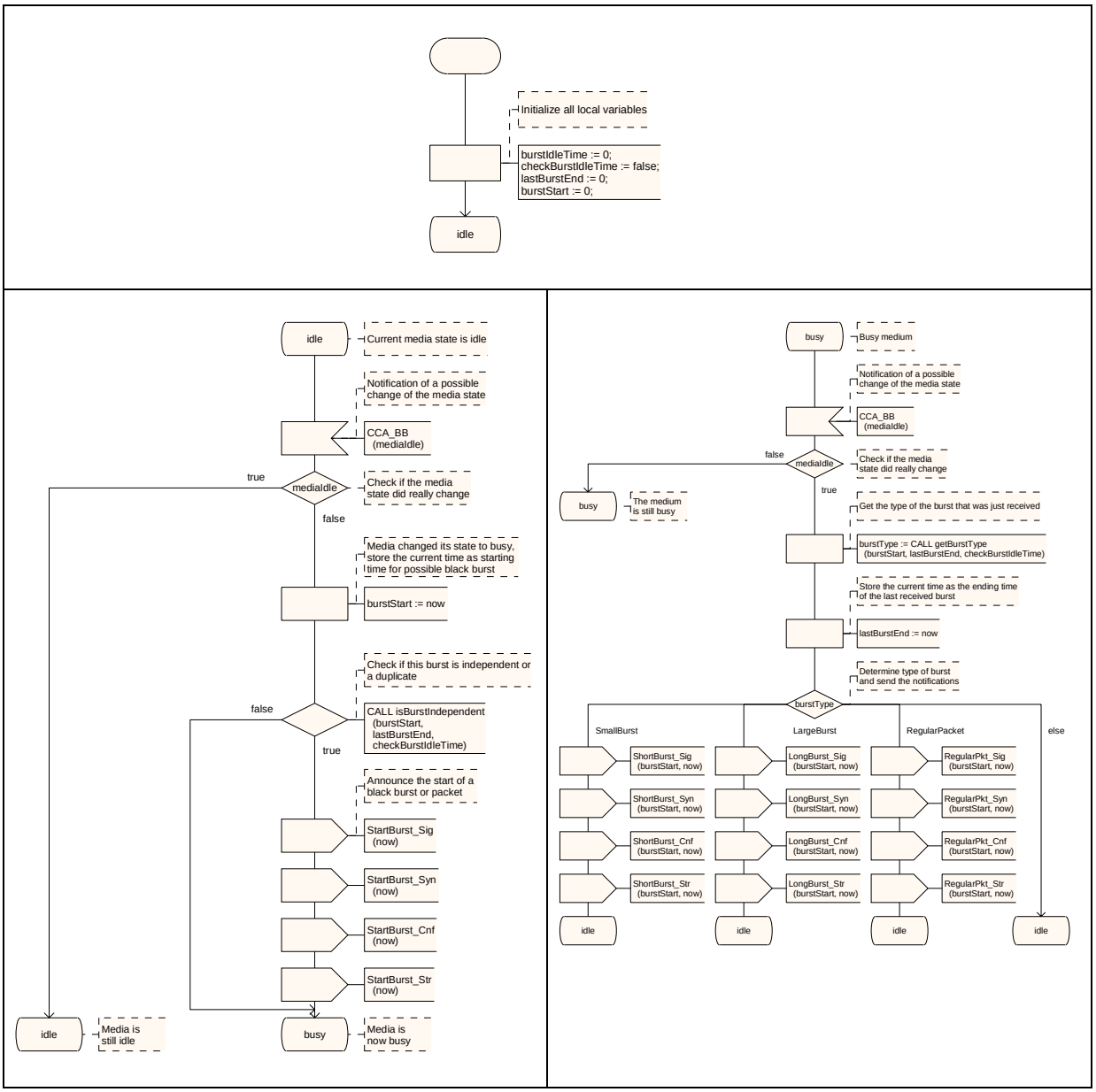


Figure 7: Design of black burst decoding functionality

5.3. Time Synchronization

This section describes our synchronization mechanism, which is based on exchanging black bursts of different length. The usage of black bursts makes the use of collision avoidance and medium access strategies unnecessary. It also ensures that a high number of nodes may send at the same time, without destroying the transmitted information. This is especially important for multi-hop synchronization in networks that potentially have a large number of nodes. When broadcasting ordinary time stamps, no collisions must occur during transmission. To reduce collisions, these networks must either use a large range of backoff slots during medium contention, or they risk a high number of collisions. This problem is not existent when black burst transmissions are used for medium synchronization.

Before starting the synchronization mechanism, a set of masters must be elected. These masters start sending black bursts sequences that are unique for each master in every synchronization slot. The other nodes start forwarding the most dominant black burst sequences. All nodes, including the masters, synchronize on the most dominant burst

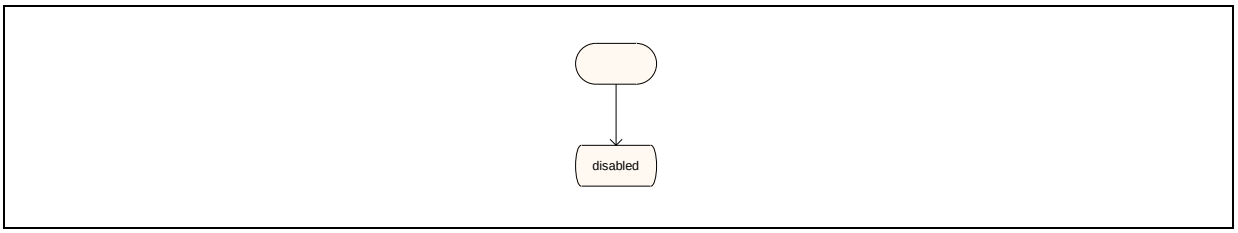
sequence. If one master fails, all nodes will still be synchronized to one of the remaining masters – the one that has the most dominant sequence of the remaining masters. Since all master nodes get synchronized too, there is no need to explicitly synchronize the masters, once the MacZ basic layer has started synchronizing all nodes. If one or multiple master nodes fail, the MacZ service layer is notified to revote the remaining masters. When a node was elected to become a master, it keeps this state until it leaves the network. This is feasible, because wireless transceivers usually require less energy for sending than for receiving packets – so no additional energy resources are necessary. This behavior also encourages the stability of the network, because with every vote due to a lost master, chances rise that a more stationary node will be elected and keep the master state. The synchronization should be performed at a rate high enough to ensure, that a few synchronization sequences may be missed without losing synchronization. This way, the MacZ service layer is able to reelect masters and to restart synchronization before the MacZ basic layer hat to switch to unsynchronized state. This will be done by the synchronization algorithm if too many synchronization slots are omitted. The MacZ service layer is notified when the state of the medium is changed. Some of the required functionality is expected to be implemented in the MacZ service layer.

5.3.1. Prerequisites

The distributed synchronization algorithm depends on a set of masters that have been elected by the MacZ service layer during the startup phase of the network. The MacZ service layer is also responsible for maintaining the master nodes. For synchronization purposes, only one master is required – the remaining master nodes serve as backups to ensure the correct functionality of the medium even if some of the masters fail or leave the network.

5.3.2. Announcement

Before the actual synchronization is started, a specific sequence of short black bursts is transmitted to announce the beginning of a synchronization phase. This is useful for synchronizing nodes that are still unsynchronized with the network. Therefore, short bursts may be used only for synchronization purposes – signaling mechanisms as described in the following sections will use long bursts. The sequence used for announcing a synchronization sequence is two short black bursts, separated by an idle period of length D_{Idle1}, (see Section 5.3.3). The announcement is followed by an idle time of the length D_{Idle1}, after which the synchronization sequence starts. The announcement sequence is transmitted by every node in the network, regardless of whether it is a master node or not. This ensures that the announcement sequence is transmitted as far as possible. The micro protocol design is shown in Figure 8.



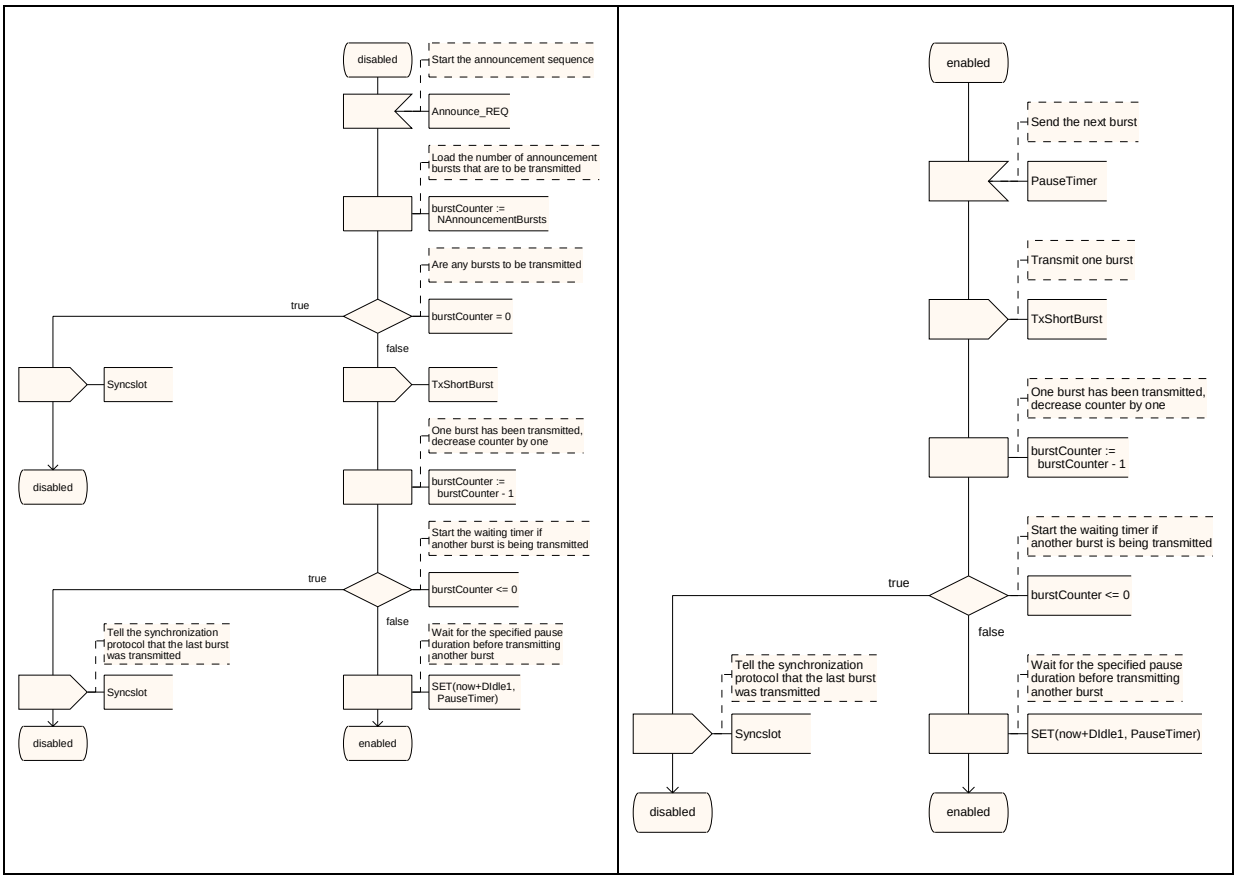


Figure 8: Design of synchronization announcement

5.3.3. Description

The synchronization algorithm is based on the usage of black burst sequences that are unique for every master. Masters are elected prior to the starting of the synchronization by the MAC service layer. For the synchronization, both types of black bursts are used. To ensure an equal length of every black burst sequence, the idle times between two black bursts must outweigh the length difference of the black bursts. The idle times following the two types of black bursts with ID 0 and ID 1 are defined as D_{Idle0} and D_{Idle1}. The length of the two different idle times must be selected according to the following criteria:

- $D_{Idle1} = D_{Burst0} - D_{Burst1} + D_{Idle0}$
  The idle times between two bursts must outweigh the length differences of these bursts.

- D_{Idle0} must be large enough to guarantee correct processing by the hardware platform.

To each master, a unique synchronization sequence is assigned. Since the sequences must be unique, the number of transmitted bursts depends on the maximum number of masters in the network. The bursts with ID 0 are longer, and therefore are considered being dominant. The master with the lowest master ID 0 has the highest priority and is assigned the most dominant burst sequence. This sequence consists of bursts with only the ID 0. The length of the synchronization sequence equals the highest possible master ID. For every master, whose master ID is above zero, its burst sequence is filled with bursts of ID 1, starting with the last burst. Table 1 illustrates this for a network with a maximum of four masters.

Master ID	Burst sequence
0	000
1	001
2	011
3	111

Table 1: Example burst sequences for every master in the network

Due to the timing constraints explained above, nodes that are transmitting a burst with ID 1 can detect the presence of a master with higher ID in their 1-hop range. Higher IDs have lower priorities and therefore, contain more bursts with ID 1. The timing of the burst sequence is to be specified by the developer. Figure 9 illustrates the timing of a black burst sequence.

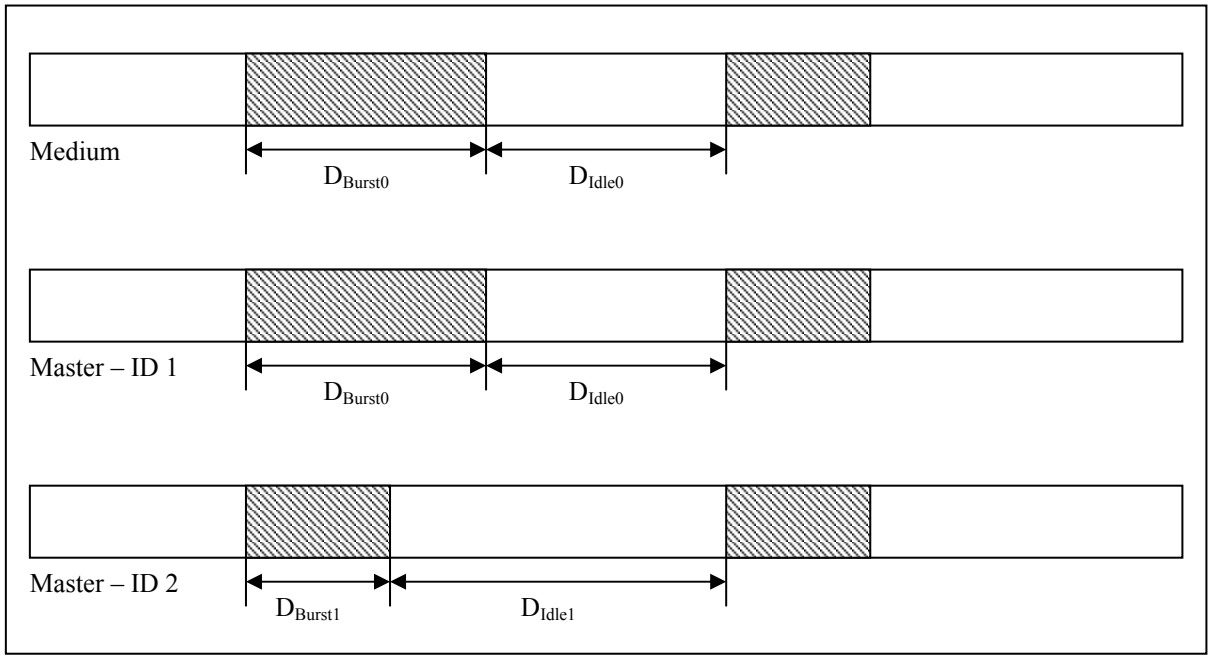


Figure 9: Timing of a black burst sequence – 2 masters within 1-hop range

Synchronization of the medium is performed in phases. When a synchronization slot starts, all masters start transmitting their unique synchronization sequences that they have been assigned to during the election process, at the same time. In the first phase, only the nodes within 1-hop signaling range of a master can hear the burst that is transmitted by the master.

Since a burst representing a 0 is longer than bursts representing a 1, bursts with ID 0 are more dominant. Master nodes that hear a longer burst than the one they are transmitting will synchronize on this longer burst. The timing constraints for black bursts ensure that nodes that are transmitting shorter bursts will be able to hear longer bursts that are transmitted at the same time. This ensures that also all masters synchronize on the master with the highest priority, which is represented by the sequence with the most leading zeros. The black burst sequences also ensure that collisions of the bursts do not harm the synchronization sequence. At the same time, redundancy is guaranteed, because synchronization takes place, as long as at least one master is available. If all masters fail at the same time, no synchronization occurs, and a master re-election in the MacZ service layer is triggered after a number of synchronization failures. Figure 9 illustrates the transmission of synchronization bursts by two masters within 1–hop range.

In Figure 9, the two masters with master ID 1 and 2 are in their 1-hop signaling range. The master with ID 2 detects the presence of the master with the higher ID due to its longer burst

and therefore, is able to synchronize on this master. This way, it is guaranteed that not only regular nodes, but also all masters synchronize on the master with the highest ID.

When any node receives a black burst sequence that is transmitted by any master, it synchronizes its medium timer to the end of the first burst. Since the bursts with ID 0 are dominant, nodes will synchronize on the more dominant masters. After each phase, a pause of a fixed length is inserted by the transmitting node. The length of this pause depends on the last burst that was transmitted by this node. The pause time has to outweigh the length difference of shorter bursts to longer bursts. Therefore, if the last transmitted burst was a burst with ID 0, a pause with the duration $D_{SyncPause0}$ is inserted. If the last transmitted burst was of ID 1, a pause with the duration $D_{SyncPause1} = D_{SyncPause0} + D_{Burst0} - D_{Burst1}$ is inserted (see Figure 10).

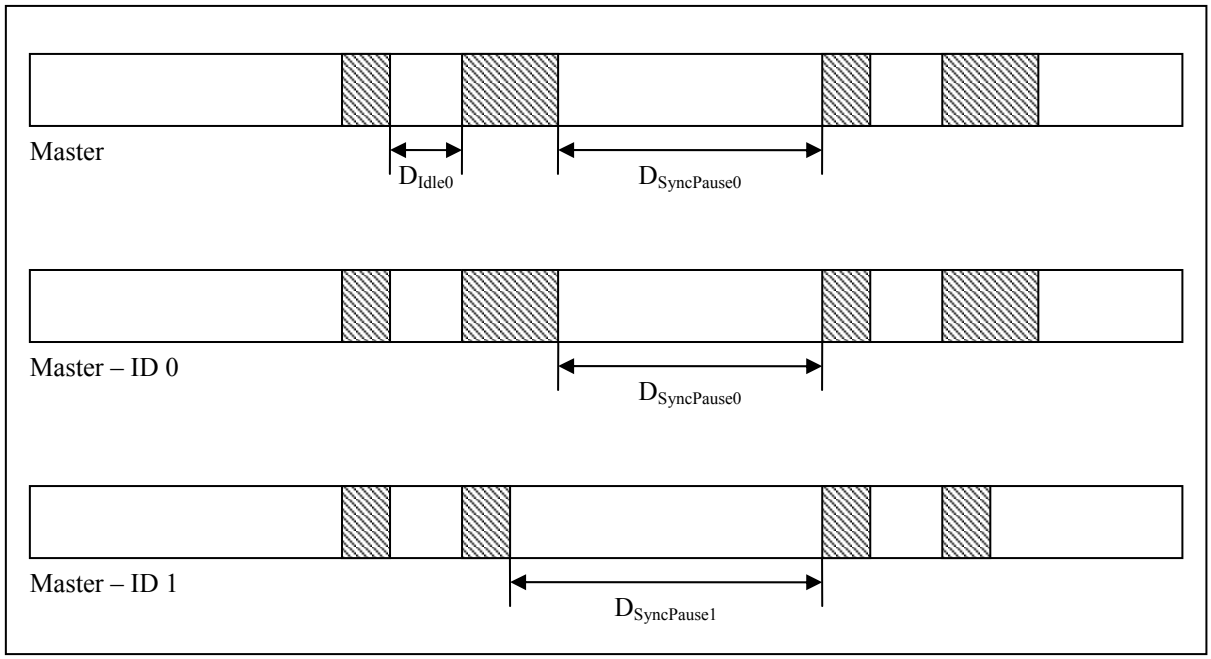


Figure 10: Synchronization pause times

The length of $D_{SyncPause0}$ and $D_{SyncPause1}$ must be known to every node, and it must also be long enough to guarantee correct processing of the black burst sequence. After the pause has passed, all nodes that received a synchronization sequence will transmit the most dominant sequence that they did receive. Since the nodes that received a sequence did synchronize on its transmitter or transmitters, they will propagate the synchronization one hop further. If any node receives a synchronization sequence with a higher priority than the one it is transmitting, it will start transmitting the higher priority sequence in the next phase. This also holds for master nodes, although, in the next synchronization slot, they will start transmitting their own sequence again. This ensures that the sequence with the highest priority will be propagated through the whole network after the number of iterations that represent the maximum diameter of the network, decreased by one (see Figure 11)

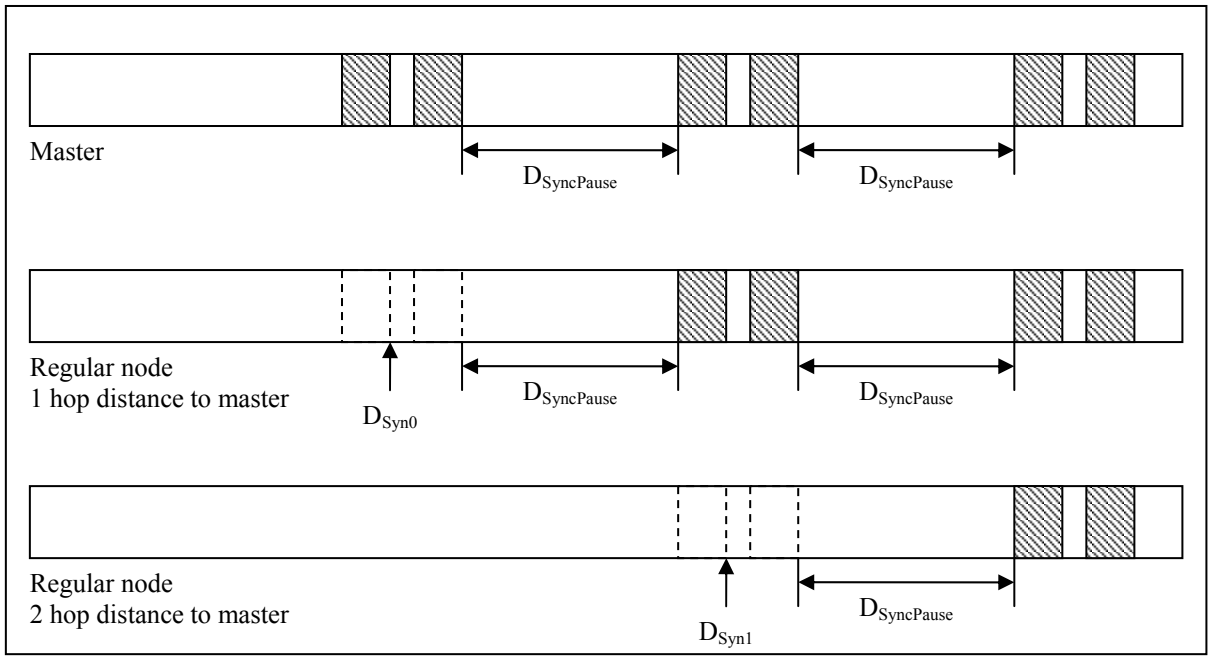


Figure 11: Synchronization across multiple hops

5.3.4. Termination

Currently, the algorithm terminates after a specific number of synchronization phases. The number of iterations N_{Iter} should be the maximum possible diameter of the network $N_{MaxDiameter}$. It is possible to set the number of iterations to a higher value to be on the safe side. Although it is theoretically possible to change this number at runtime, in our current design it is specified by the developer. This has the advantage, that the maximum time for network synchronization can be predicted, which facilitates the possibility for guaranteeing a specific bandwidth and delay to transmissions – as long as the network remains synchronized. A reference implementation of this synchronization algorithm for the Chipcon CC2420 Transceiver chip is presented in Section 6.

5.3.5. Synchronization error

Since synchronization is done iteratively, the nodes within 1-hop distance ti the master nodes are synchronized first. In every iteration, one more hop is synchronized. The achievable synchronization accuracy with this algorithm depends on the number of hops in the network – every hop adds its timer jitter T_{HWJit} to the maximum drift right after synchronization. For the tolerable timer drift $T_{MaxDrift}$ of the network, the equation $T_{MaxDrift} > N_{MaxDiameter} * T_{HWJit}$ must hold.

5.3.6. Micro protocol design

The micro protocol design of this synchronization algorithm is currently ongoing work.

5.4. Fully distributed synchronization

This section describes an alternative synchronization algorithm. The main difference of this algorithm to the synchronization algorithm described in Section 5.3 is the absence of the need of a set of elected master nodes. However, the synchronization error for every hop might be significantly higher – depending on the value of $T_{SwitchTX}$.

5.4.1. Description

This synchronization algorithm works fully distributed across multiple hops. The main benefit of this algorithm is that the failure of a specific node or of a set of nodes will not disturb the communication capabilities of the network. This algorithm also synchronizes with black bursts – however – only one type of black bursts is required. Since long black bursts are also used for signaling, this algorithm depends solely on short black bursts. Every node transmits the same synchronization sequence, consisting of a single black burst.

Before the synchronization is started, the announcement sequence described in Section 5.3.2 is transmitted by all nodes in the network. The startup behavior of the network is described later in this work.

Synchronization of the network is achieved in several phases. The number of necessary phases depends on the diameter of the network. In every phase, each node will send a black burst. All nodes will synchronize themselves on the first received burst. This results in all nodes synchronizing to the first burst within their 1-hop distance. Figure 12 illustrates the synchronization algorithm.

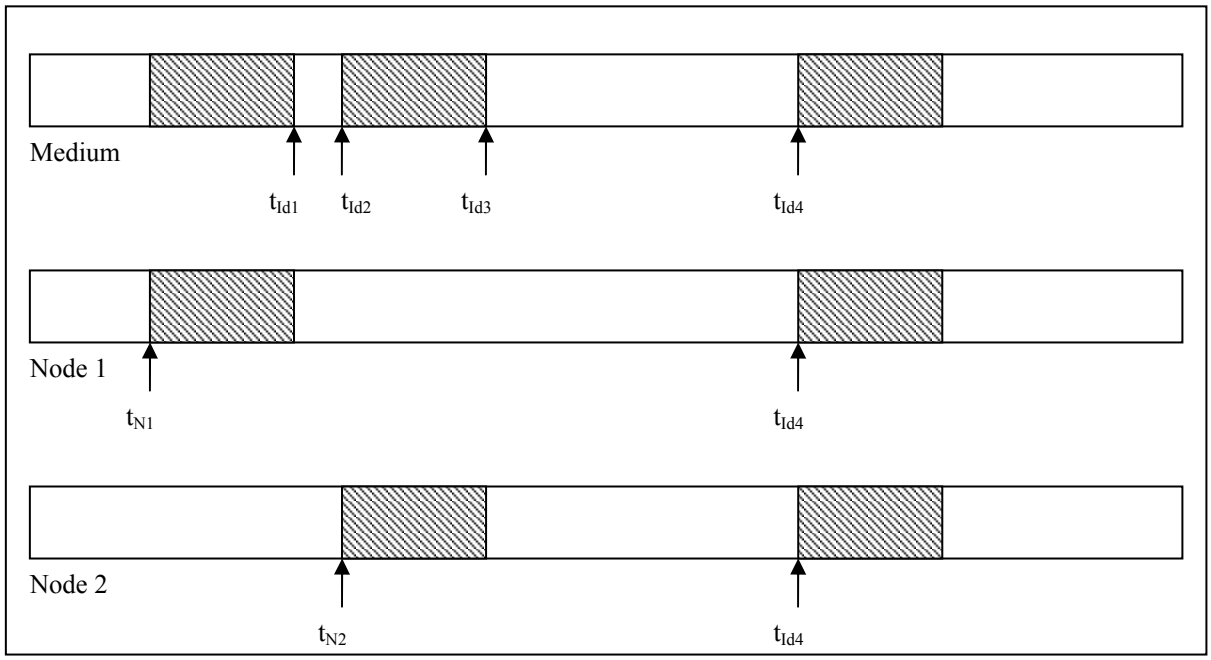


Figure 12: Synchronization with fully distributed algorithm

In the example in Figure 12, two nodes communicate over a wireless medium. At the beginning, both nodes have a timer drift of $t_{N2} - t_{N1}$. Every node sends a synchronization burst at the same "virtual" time. Due to timer drift, these bursts might be transmitted at different points in real time. Node 1 sends its synchronization burst at time t_{N1}. Node 2 receives the beginning of the burst at time t_{N1} plus propagation delay. This will cause Node 2 to synchronize its virtual time to Node 1, because this Node was the first one transmitting a synchronization burst. This causes all nodes to synchronize to the first transmitting node within their 1-hop distance. Node 2 transmits its own burst in this period as soon as possible – in Figure 12 this is at time t_{N2}. This causes other nodes to possibly synchronize on Node 2 – if Node 2 is the first transmitting node within their 1-hop distance.

In the next phase, Node 2 will transmit its burst synchronously with Node 1. In Figure 12, this is at the point of time t_{Id4}. This way, all nodes synchronize through the whole network to the node that was the first transmitting node.

The algorithm terminates after a defined number of phases that equals $N_{MaxDiameter}$, the maximum possible diameter of the network. This ensures the predictability of the length of every synchronization slot. In the case that $T_{SwitchTX}$ is significantly larger than $T_{SwitchRX}$, it is also possible to synchronize with the end of black bursts instead of synchronizing with their beginnings.

5.4.2. Synchronization error

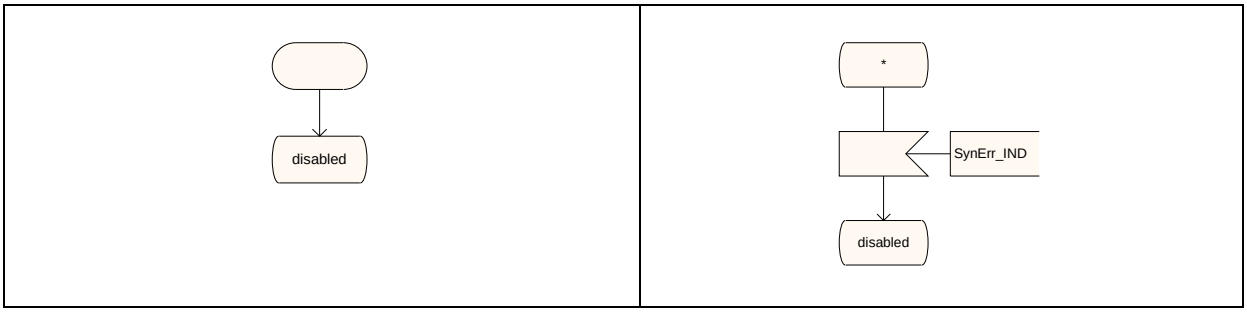
Every hop might add its timing jitter T_{HWJit} to the synchronization drift. Since every node transmits a black burst in every iteration, the achievable accuracy is also decreased by the length of the blind period prior to sending. During this period, a node is not able to sense the medium state. This affects the synchronization drift, because a node is only able to detect bursts that are sent before this period. Therefore, all nodes within 1-hop range cannot synchronize more accurately than $T_{HWJit} + 2*T_{SwitchTX}$. The reason is that the timer drift of every node may vary by $T_{SwitchTX}$ either into the future or into the past. As a result, the accuracy through the complete network with n hops is limited to $N_{MaxDiameter} * (T_{HWJit} + 2*T_{SwitchTX})$.

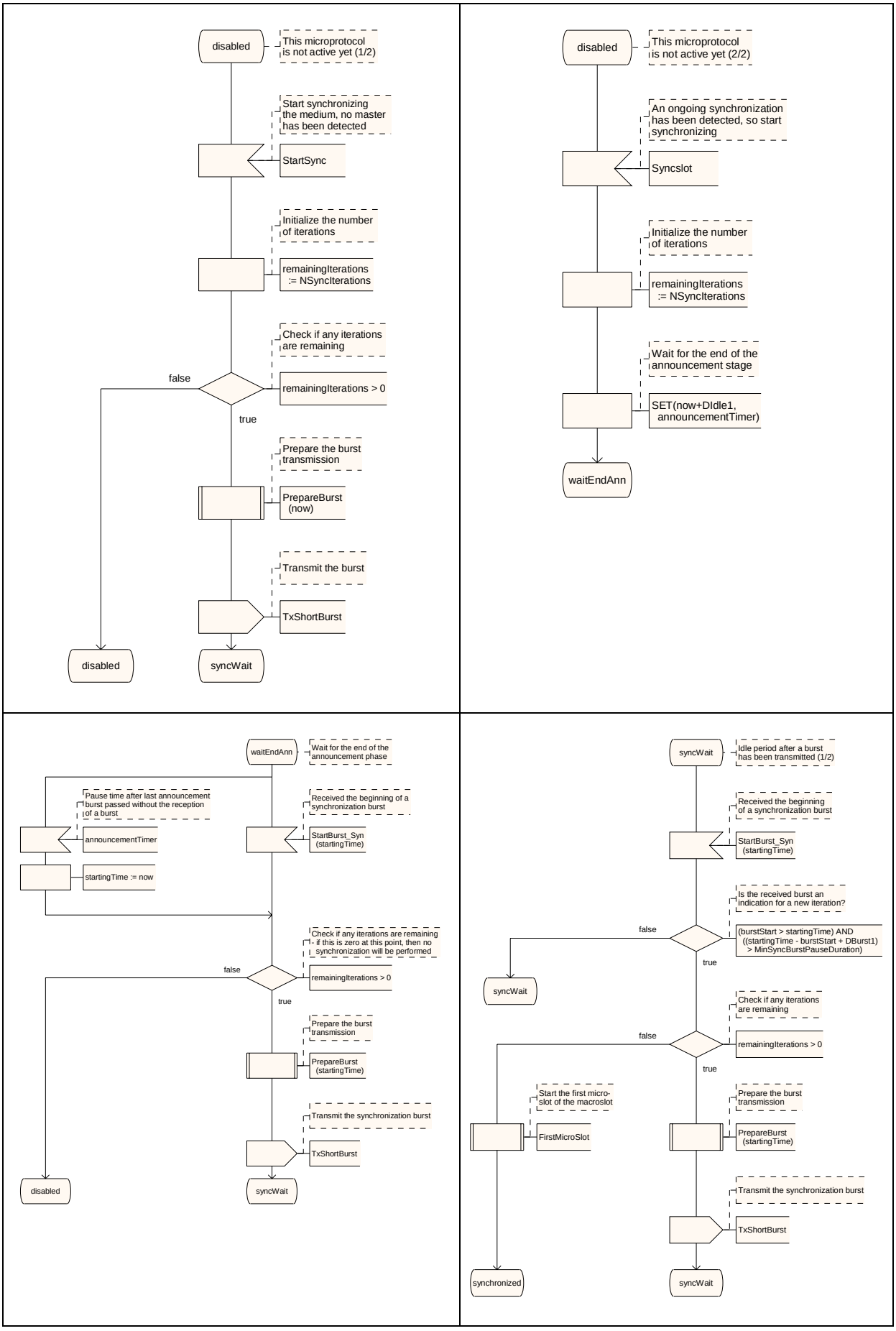
5.4.3. Comparison

This algorithm requires no master election. However, its maximum possible accuracy is significantly lower than the accuracy of the algorithm described in Section 5.3. Both algorithms can be used to synchronize the nodes, which is needed for slotting the medium. The algorithm described in this section can be used when a slotted medium with low bandwidth requirements is needed. Since the synchronization drift must be added to the beginning of every slot, a larger amount of the medium is wasted than it would be necessary when synchronizing with the algorithm described in Section 5.3. So for applications where a higher throughput and a more accurate synchronization is necessary, the algorithm with master election should be used.

5.4.4. Micro protocol design

Figure 13 shows the main parts of the micro protocol design of the multi hop synchronization functionality.




disabled
This microprotocol is not active yet (1/2)
Start synchronizing the medium, no master has been detected
StartSync
Initialize the number of iterations
remainingIterations := NSyncIterations
Check if any iterations are remaining
false
remainingIterations > 0
true
Prepare the burst transmission
PrepareBurst (now)
Transmit the burst
TxShortBurst
disabled
syncWait
disabled
This microprotocol is not active yet (2/2)
An ongoing synchronization has been detected, so start synchronizing
Syncslot
Initialize the number of iterations
remainingIterations := NSyncIterations
Wait for the end of the announcement stage
SET(now+DIdle1, announcementTimer)
waitEndAnn
waitEndAnn
Wait for the end of the announcement phase
Pause time after last announcement burst passed without the reception of a burst
announcementTimer
startingTime := now
Received the beginning of a synchronization burst
StartBurst_Syn (startingTime)
Check if any iterations are remaining if this is zero at this point, then no synchronization will be performed
false
remainingIterations > 0
true
Prepare the burst transmission
PrepareBurst (startingTime)
Transmit the synchronization burst
TxShortBurst
disabled
syncWait
syncWait
Idle period after a burst has been transmitted (1/2)
Received the beginning of a synchronization burst
StartBurst_Syn (startingTime)
Is the received burst an indication for a new iteration?
false
(burstStart > startingTime) AND ((startingTime - burstStart + DBurst1) > MinSyncBurstPauseDuration)
true
syncWait
Check if any iterations are remaining
false
remainingIterations > 0
true
Start the first micro-slot of the macroslot
FirstMicroSlot
Prepare the burst transmission
PrepareBurst (startingTime)
Transmit the synchronization burst
TxShortBurst
synchronized
syncWait

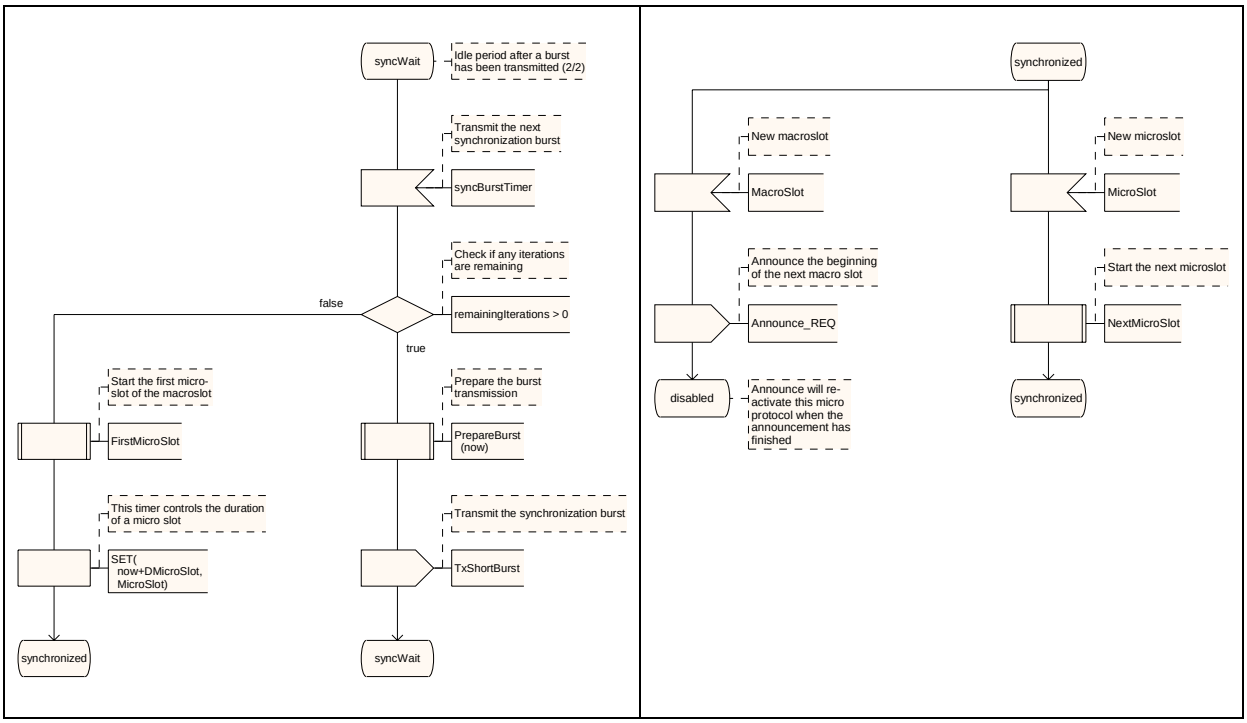


Figure 13: Design of synchronization functionality

5.5. Event signaling methodology

Signaling is done in specific slots of the medium, called “signaling slots”. Our current design supports the signaling of three different messages: Transmission of an alert message, a pending master election sequence and a network synchronization error. The first two messages will be addressed in this section; the network synchronization error will be treated in section 5.8.

5.5.1. Description

For the signaling of specific events, a mechanism based on black bursts is used. Black bursts have the advantage that multiple nodes may signal the same events, without loss of information due to collisions. Also the range of black bursts is higher than the range of ordinary frames, ensuring that at least every node within 1-hop distance will receive the signal. As already mentioned in Section 5.2, black bursts of two different lengths are defined. The developer must ensure that the length of black bursts, the minimum frame length, the tolerable timer drift of all nodes, and the timing jitter of the operating system are selected correctly, so that our MacZ is able to distinguish between the two types of black bursts and regular frames. Since the shorter black bursts are reserved for the announcement of synchronization sequences, the signaling mechanisms must use the longer black bursts only.

Every node must be awake during all signaling slots. This ensures that all nodes will receive important signals. When no event has been signaled, nodes may, depending on the needs of their upper layers, go asleep to save energy. When an event that concerns a node is signaled, this node must not go asleep. Currently, there are two possible events apart from network synchronization errors that are signaled: The pending transmission of an alert message and the pending election of one or multiple synchronization masters. When an alert message is signaled, all nodes should remain active, for example for supporting to route the message quickly to its destination. When an upcoming master election is signaled, all nodes that can act as synchronization masters should remain active and participate on the election.

Currently, a signaling slot is separated into different partitions, where every partition is reserved for signaling a specific event. The length of a partition is defined as $D_{SigPart}$. The first partition number 0 is used for signaling alert messages; the second partition is used for signaling an upcoming master election (see Figure 14).

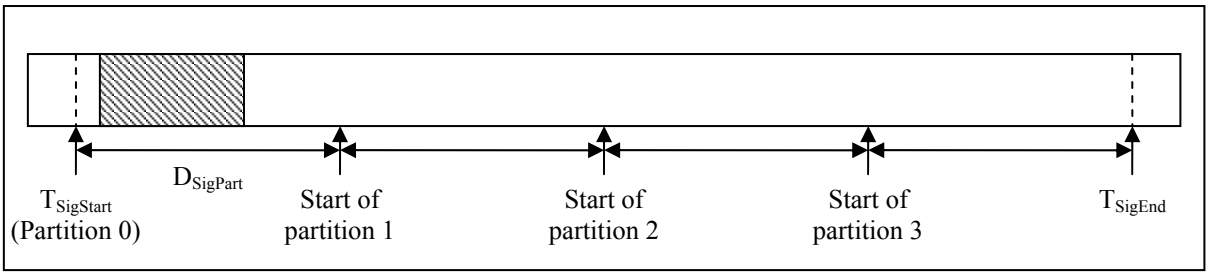


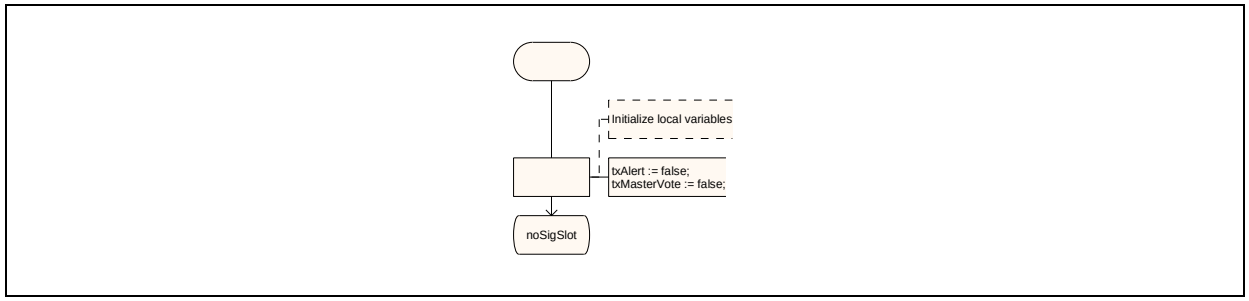
Figure 14: Signaling of an alert message

In Figure 14, the signaling of an alert message is shown. The last partitions of a signaling slot are reserved for the detection of synchronization errors. Section 5.6 describes this functionality in greater detail.

To support the rapid propagation of signals, multiple signaling slots may be inserted in every macro slot. Every signaling slot helps speeding up the transmissions of signals, because the signal is transmitted one hop further, but it also consumes energy, because all nodes will have to be awake during an additional slot, even if no signals are transmitted by any node.

5.5.2. Micro protocol design

The main parts of the design of the signaling functionality are shown in Figure 15.



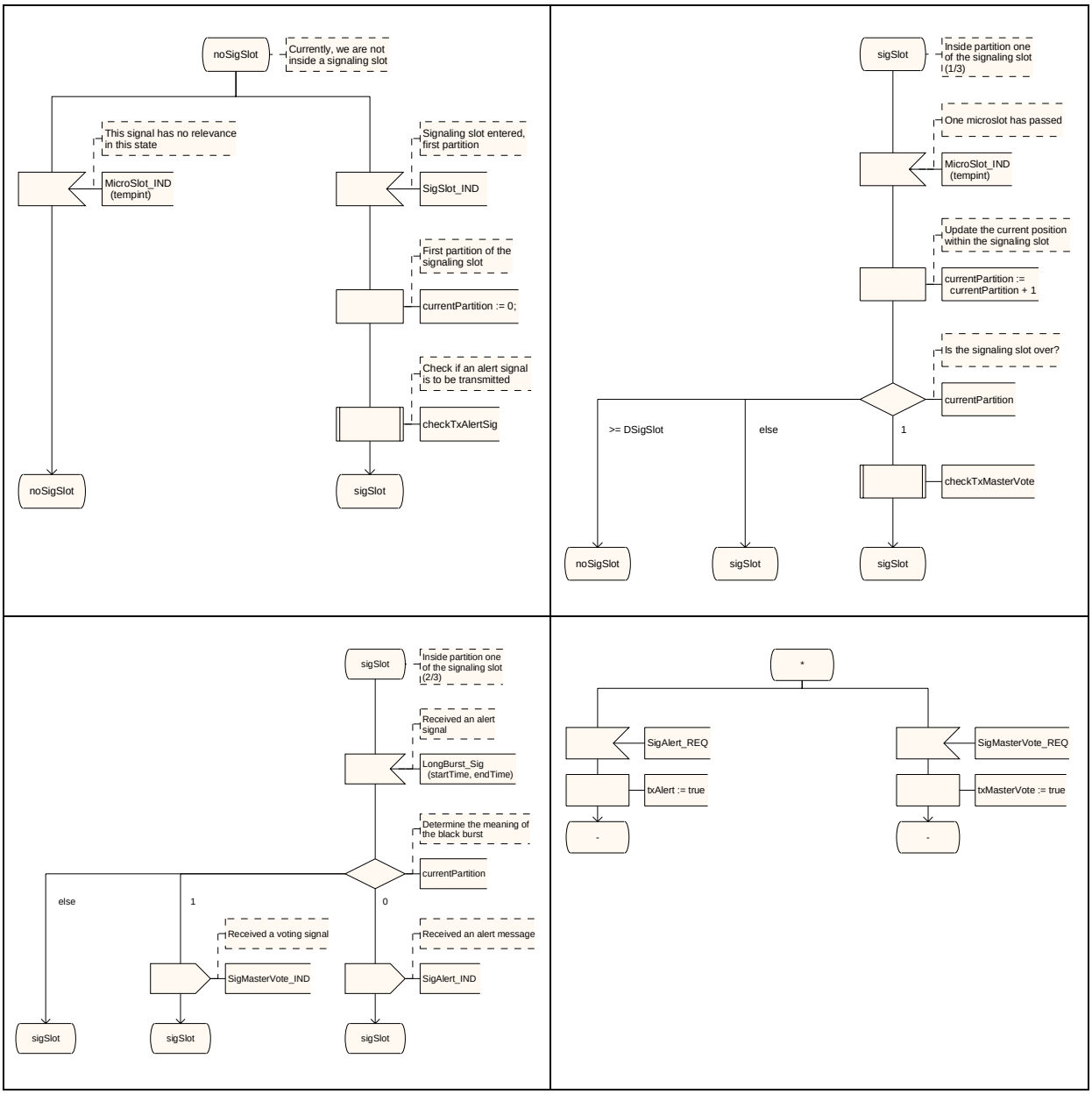


Figure 15: Signaling functionality

5.6. User defined signaling

In addition to signaling slots used for alert and master election signaling, the MacZ basic layer also offers signaling slots for user defined purposes. Normally, using such a service is only feasible for the MacZ service layer – it depends on the realization of this layer whether the user defined signaling mechanism is also available to the upper layers.

5.6.1. Description

The user defined signaling is similar to the signaling mechanism that is described in Section 5.4, with the exception that any possible black burst sequence that fits into the user defined signaling slots may be used. Therefore, the conflict detection is not active during these slots. To ensure that freshly powered-up nodes will not wrongly detect a black burst sequence inside user defined signaling slots as the start of a synchronization sequence, only long black bursts may be used for user defined signaling. The MacZ basic layer cares about transmitting the bursts at the correct time and reports received bursts to the MacZ service layer.

5.6.2. Micro protocol design

The design of this specific component is ongoing work.

5.7. Conflict detection

Whenever a time-synchronized medium is created, there must be a mechanism to detect unsynchronized nodes or networks that may cause interferences. The time synchronization of a network is mandatory for the functionality of the synchronization, signaling and contention-free algorithms, because unsynchronized nodes that are transmitting bursts or packets in these slots may cause collisions or may hide regular signals. Therefore, a conflict detection algorithm must be implemented.

5.7.1. Description

Conflicts may arise when two networks, that are not synchronized, come close to each other. Due to the fact that most nodes will probably be sleeping, this might not be detected for a while. Two possible detection strategies can be implemented:

- Detecting synchronization errors at the MacZ basic layer
- Detecting synchronization errors at the MacZ service layer

The MacZ service layer can detect network synchronization errors, when unexpected frames are being received, for example frames with an unknown network id or frames that are sent by an unknown node within a reserved time slot. Since the MacZ service layer is not treated in this report, its detection capabilities will be regarded as “detected by upper layers”. Apart from having to rely on the upper layers for network error detection, the MacZ basic layer can detect network errors also on its own. For this purpose, signaling slots are used.

During signaling slots, every node that is not transmitting a signal must be listening to the medium. Nodes must also listen to partitions of the signaling slots in whose they are not transmitting a signal in. Since only the first part of every signaling slot is used for signaling, the remaining slot is expected to be idle. If unexpected transmissions are detected in these signaling slots – either frames that are too long to be black bursts or unexpected burst sequences, a conflict has been detected by the MacZ basic layer. This information is then sent to all component of the MacZ basic layer and also to the MacZ service layer. Only those signaling slots that are defined in Section 5.4 are used for conflict detection. Since user defined signaling slots, which are introduced in Section 5.6, allow for any user defined burst sequence, the conflict detection is not performed in these slots.

If a node remains active outside the signaling and synchronization slots, its collision detection functionality also remains active. The medium is permanently monitored for a sequence of two short bursts – which would clearly announce a foreign synchronization sequence. The minimum packet length ensures that no regular traffic can be misinterpreted as a sequence of two short black bursts. Section 5.8 describes the behavior of a node that detects a foreign or unsynchronized network.

5.7.2. Micro protocol design

Figure 16 presents the design of the conflict detection functionality of the MacZ basic layer.

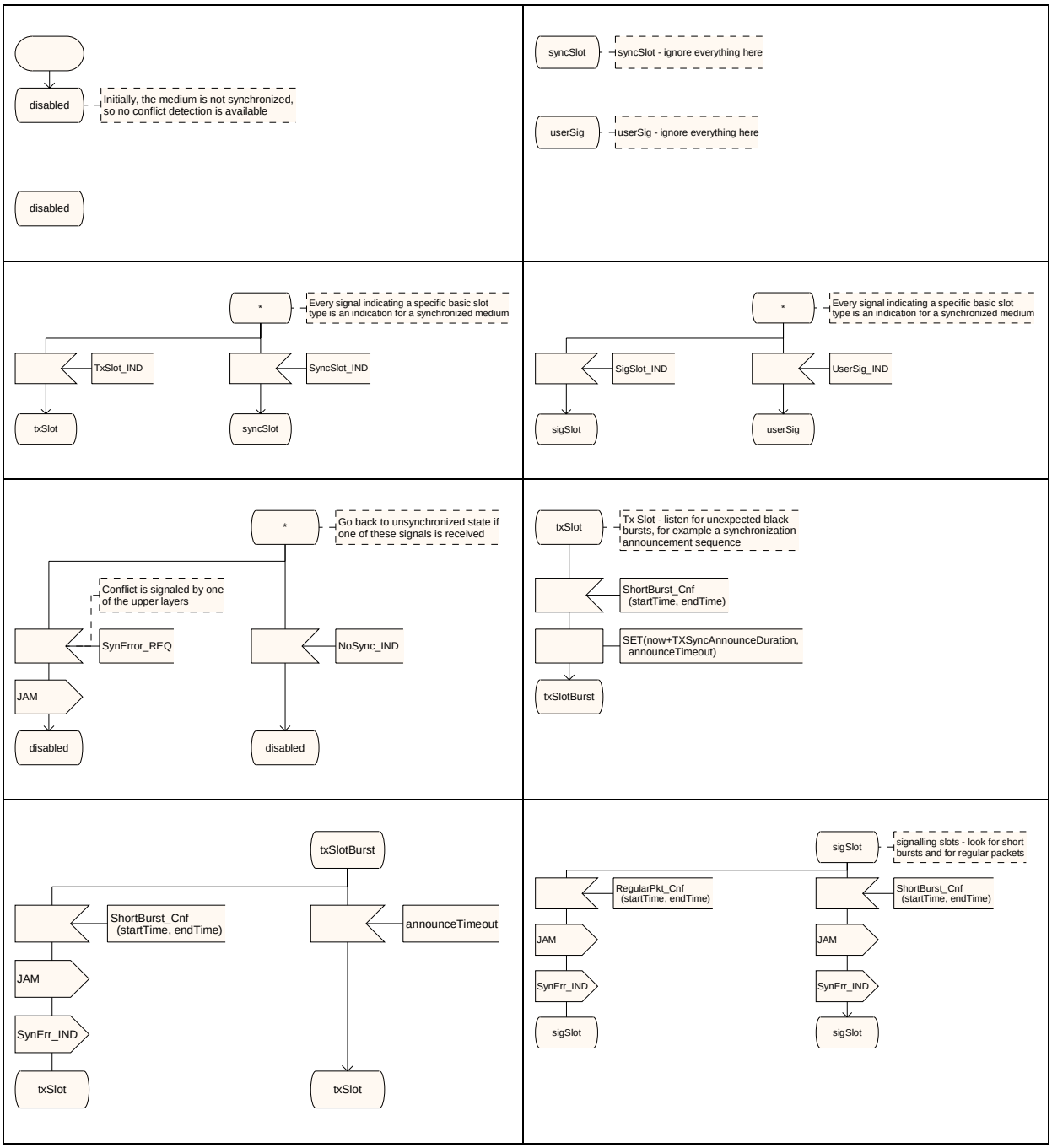


Figure 16: Collision detection functionality

5.8. Conflict resolution

When a conflict has been detected by one node, it must be propagated to all other nodes – to the nodes that belong to the same network, and also to the conflicting network. The methodology for propagating this conflict information is described in the following.

5.8.1. Description

When a node detects an unsynchronized network, it will move itself into unsynchronized state and start sending a jamming sequence of black bursts for the duration of a whole macro slot. For the jamming sequence, the shorter black bursts with ID 1 are used. Figure 17 illustrates the jamming sequence. The first jamming burst is transmitted in this example from $t_{MacroSlot}$ to

t_1, and then an idle period until the time t_2 follows. This is repeated for the duration of a whole macro slot.

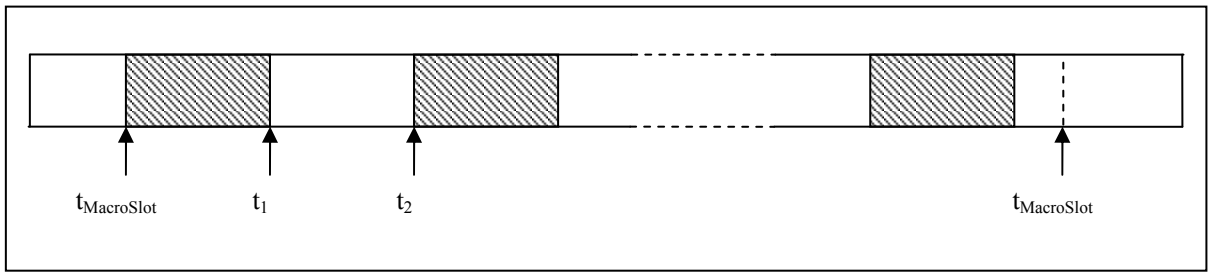


Figure 17: Jamming sequence

The jamming sequence is sent for a whole macro slot - this explicitly includes all synchronization and signaling slots. Within their signaling slot, all nodes within 1-hop distance will recognize the jamming sequence – even if they are not synchronized with the network. The signaling slots have to be long enough to hold at least three of the used, short bursts. This ensures, that the jamming sequence is received, even if other events are being signaled (see Figure 18).

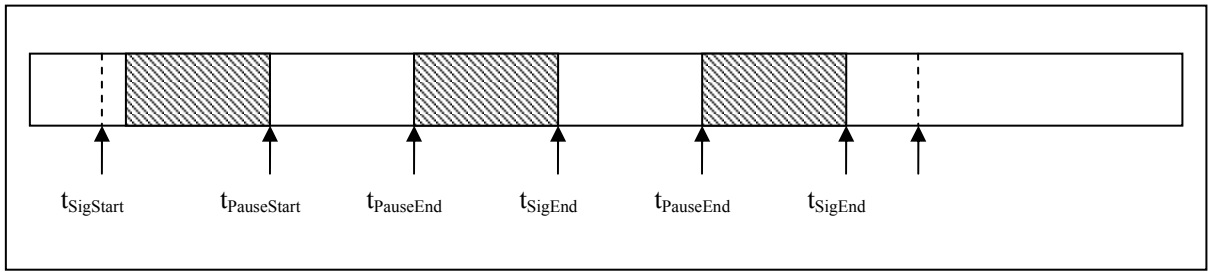


Figure 18: Jamming sequence within synchronization slot

The length of the macro slots of all networks that may probably get in range of each other must be equal for the jamming algorithm to work reliably. To propagate the information through all hops of both networks, the jamming sequence must be repeated $N_{MaxDiameter}$ times, where $N_{MaxDiameter}$ is the maximum number of hops across all networks. This value depends on the application domain and must be predefined by the developer. If several nodes transmit jamming sequences at the same time, which is likely to happen, multiple bursts may overlap, forming a long burst transmission that may even exceed the maximum size of a regular frame. So the detection of jamming sequences in the signaling component may not rely on detecting an ideal jamming sequence consisting of short bursts. To avoid interferences from nodes that have just been powered on, the start-up phase of every node has to be explicitly specified.

5.8.2. Termination

The transmitted jamming sequence terminates, after $N_{MaxDiameter} * T_{MacroSlot}$ time has passed. After a node has sent the jamming sequence for that period of time, it will start listening on the medium. To ensure the propagation of the jamming sequence through the whole medium, every node will wait the amount of time specified by $N_{MaxDiameter} * T_{MacroSlot}$. After $N_{MaxDiameter} * T_{MacroSlot}$ has passed, the jamming sequence has propagated through the network. During this time, the first node has transmitted its jamming sequence. Then it has to wait for at least $N_{MaxDiameter} * T_{MacroSlot}$, before the nodes with the maximum distance have transmitted their complete jamming sequence.

After a node has waited the time specified by $N_{MaxDiameter}$ * $T_{MacroSlot}$, it terminates its jamming algorithm and triggers the algorithm that controls the startup phase, which is described in the next section.

5.8.3. Micro protocol design

Figure 19 shows the design of the jamming functionality of MacZ.

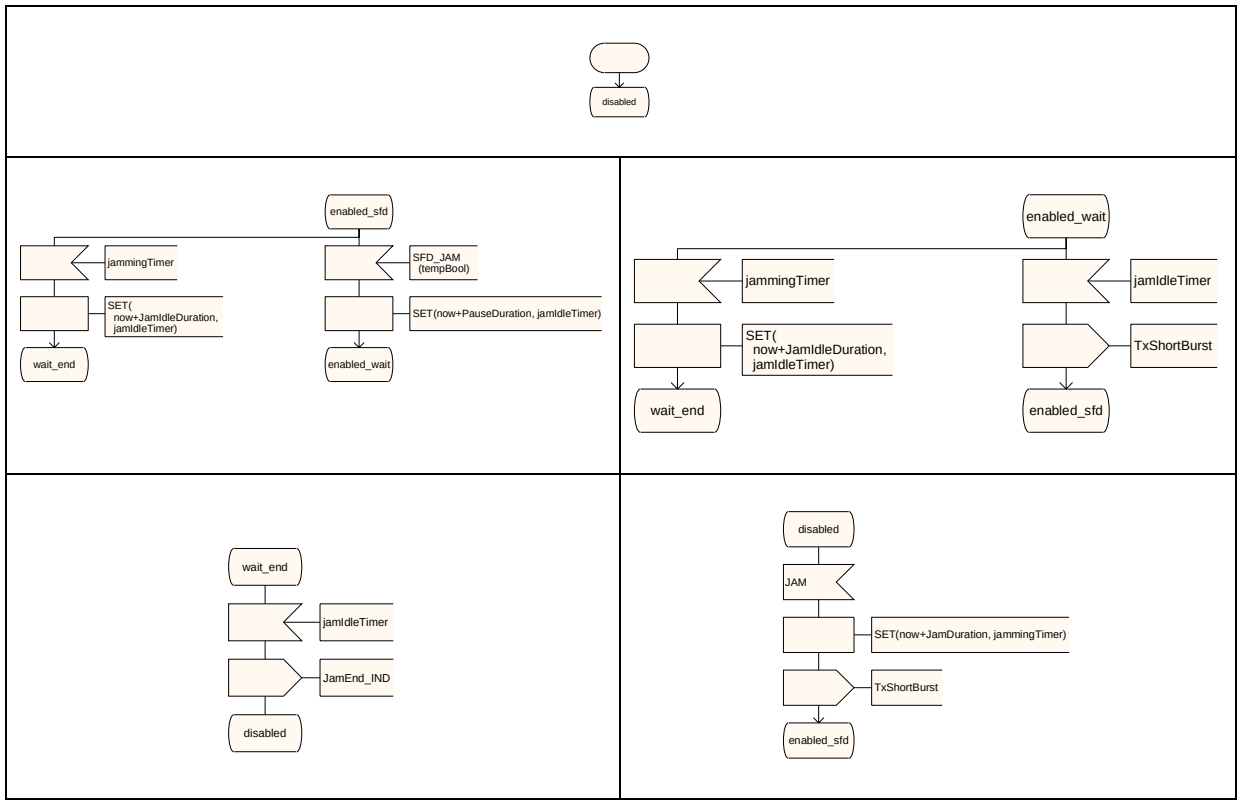


Figure 19: Design of the jamming functionality

5.9. Startup phase

The startup is performed after powering on a node, or after a jam sequence has been transmitted.

5.9.1. Description

When the startup is triggered, a node waits for $N_{MaxDiameter}$ + 1 macro slots, so every node waits for the time specified by ($N_{MaxDiameter}$ + 1) * $T_{MacroSlot}$ before it starts synchronizing the network. There are two reasons for this behavior:

- The first reason is that this idle time gives the node the possibility of synchronizing with an already existing network. If it receives a synchronization announcement sequence within this period of time, it will start synchronizing with the preceding synchronization sequence.
- The second reason is that it must be ensured that all nodes are ready for starting resynchronization after a jamming sequence has been transmitted.

Every node will wait for the time specified by ($N_{MaxDiameter}$ + 1) * $T_{MacroSlot}$ before it triggers the startup algorithm, so a idle time ensures that every node of the network is in the startup phase.

After $N_{StartupDelay}$ has passed without hearing the announcement sequence, the node starts with an unsynchronized medium. The synchronization service of this node then begins synchronizing the medium. The actual synchronization behavior depends on the used algorithm. The synchronization algorithm described in Section 5.3 will start by using the MacZ service layer and an unsynchronized medium for electing an initial set of master nodes. The alternative algorithm, described in Section 5.4 will start without having to elect any master nodes.

5.9.2. Micro protocol design

Figure 20 shows the micro-protocol design of the startup phase that is triggered after synchronization was lost or whenever a node is powered on.

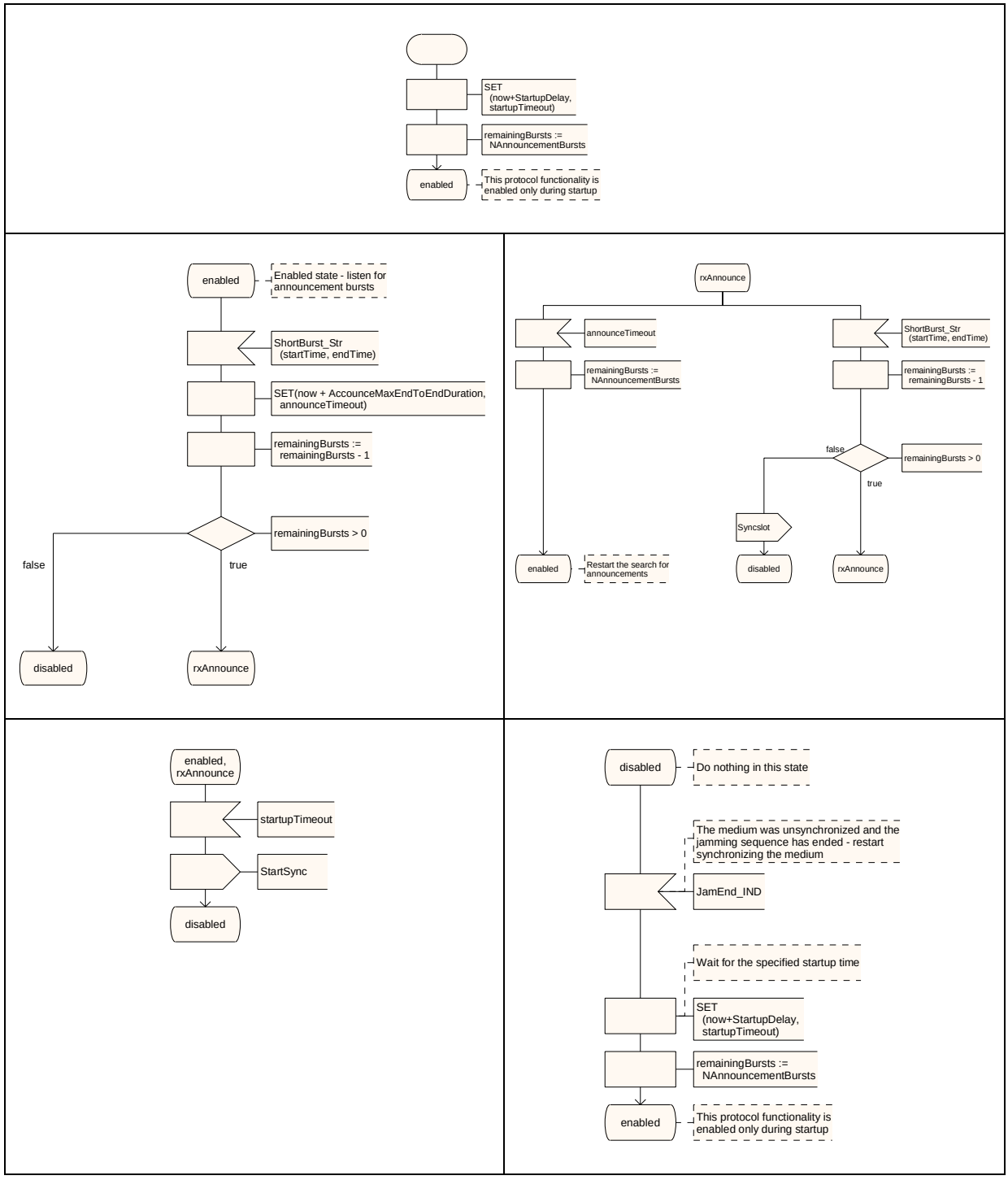


Figure 20: Implementation of start-up phase

5.10. Limited nodes

As an exception to the full featured nodes, which have to be awake during every synchronization and signaling slot, also limited nodes are possible within the network. These nodes may also sleep during signaling and synchronization slots. Limited nodes usually have very low transmission requirements and also very scarce energy resources. The possibility of being idle during synchronization and signaling slots too, allows for much greater idle time to these nodes, and as a result, much greater energy savings.

Limited nodes only synchronize prior to transmitting anything. As a result, they are not synchronized with their network most of the time. Therefore, it is mandatory for them to participate in at least one synchronization sequence before starting any transmission. The startup phase for limited nodes is also different to the startup phase of regular nodes. A limited node will never start synchronizing the medium after it has been powered up. Instead, it will transmit using the unsynchronized medium if it was not able to synchronize after $N_{StartupDelay}$ has passed. Limited nodes must not participate in master elections, if the synchronization mechanism relies on virtual masters. It should also be noted that the higher layers in the protocol stack must ensure, that the receiver(s) for the frame that is being transmitted by the limited node is also awake when the limited node wakes up and starts to transmit. In a network with only limited nodes, it must be ensured that all nodes wake up at a specific time. Therefore, an application level synchronization protocol may be necessary.

Since limited nodes are not capable of providing a network infrastructure, in most domains a number of full featured nodes should be available for providing a reliable infrastructure, for example for providing the time synchronization and multi-hop routing services.

6. Implementation on MicaZ motes

This section describes the adaptation of MacZ a specific hardware platform, which is an Atmel ATMega128L micro controller and a Chipcon CC2420 transceiver [CC2420] chipset.

6.1. Runtime platform

This section describes the characteristics of the hardware platform “MicaZ”. The relevant properties of the transceiver chip are described in this section, while the concrete timing of our MAC layer for this transceiver chip is described in Section 6.2.

6.1.1. Description of the CC2420

The following timing characteristics of the Chipcon CC2420 transceiver are relevant for the adaptation of MacZ:

Symbol	Value	Description
$t_{SwitchTX}$	192 µs	Required time for switching from receiving to transmission mode
$t_{SwitchRX}$	320 µs	Required time for switching from transmission to receiving mode, until the CCA signal is valid
$N_{Bandwidth}$	250.000 Bit/s	The available bandwidth

t_{Byte}	32 µs	Required time for transmitting a byte
$t_{Preamble}$	128 µs	Size of preamble of every transmitted packet.
t_{Header}	64 µs	Size of the physical header, this is the start of frame delimiter and the length information on this platform
$N_{MaxSize}$	121 Bytes	Maximum packet length without physical header and preamble
$N_{MinSize}$	0 Bytes	Minimum packet length without physical header and preamble

6.1.2. Processor and available timers

The accuracy of the available timers is important, because this is the granularity at which the length of black bursts can be measured. The available hardware timer was considered to offer a granularity of T_{HWJit} = 32µs. This means, that every measured time has an accuracy of these 32µs, which must be considered when determining values for slot- and black burst sizes.

6.1.3. Domain specific constraints

The application domain of the wireless network is in indoor and outdoor scenarios with a maximum diameter $N_{MaxDiameter}$ = 5 Hops. The duration of a macro slot is set to 1 second. Since alert messages may be signaled, two signaling slots per macro slot are assigned, so that an alert message may pass the network in about 2.5 seconds in the worst case of all nodes being asleep and depending on the signaling. For the domain, a maximum of 4 concurrent masters were deemed to be sufficient.

6.2. Adaptation of MacZ

The timing constraints for the MAC layer must be set to values that guarantee that the equations below hold.

$$D_{Burst0} > D_{Burst1} + T_{MaxDrift} + 4 * T_{HWJit}$$

$$D_{Burst0} > D_{Burst1} + T_{SwitchRX}$$

$$D_{Burst0} + 2 * T_{HWJit} + T_{MaxDrift} > D_{MinPkt} - 2 * T_{HWJit}$$

$$D_{Idle1} = D_{Burst0} - D_{Burst1} + D_{Idle0}$$

$$T_{MaxDrift} > N_{MaxDiameter} * T_{HWJit}$$

Further, the correct processing of all signals from the hardware must be guaranteed. Based on the data of the runtime platforms, and the domain requirements, the following values for the MAC layer have been defined:

- $T_{MaxDrift}$ = 192µs
- D_{Idle0} = 1ms
- $D_{SyncPause0}$ = 1ms
- D_{Burst1} = $t_{Preamble} + t_{Header}$ = 192µs
- D_{Burst0} = $D_{Burst1} + T_{MaxDrift} + 4 * T_{HWJit} + 4 * t_{Byte}$ = 640µs
- D_{Idle1} = $D_{Idle0} + D_{Burst0} - D_{Burst1}$ = 1,448ms
- $D_{SyncPause1}$ = $D_{SyncPause0} + D_{Burst0} - D_{Burst1}$ = 1,448ms
- $D_{MinFrame}$ = $D_{Burst0} + 4 * T_{HWJit} + T_{MaxDrift} + 2 * t_{Byte}$ = 960µs

6.3. Synchronization duration and accuracy

This section describes the required time and achievable synchronization accuracy when using the algorithm with master election, described in Section 5.3. With these values, the synchronization across four hops requires the following amount of time:

1,64 ms are required for transmitting a black burst. For the sequence of two bursts, 3,28ms are required. This already includes the pause $D_{SyncPause}$ between two iterations (D_{Burstn} + D_{Idlen} + D_{Burstm} + $D_{SyncPausem}$).

For synchronizing across a maximum of $N_{MaxDiameter}$ = 5 hops, 3,28ms * 5 - $D_{SyncPause0}$ = 15,4ms are required, because after the last synchronization iteration, no additional pause is necessary. If the synchronization sequence ends with a short burst, the additional 448µs of the longer burst are saved, resulting in 14,952ms for the whole synchronization. The duration of the synchronization-announcement sequence must be added to this time.

The achievable timer accuracy also depends on the maximum number of hops $N_{MaxDiameter}$. Every hop might add a jitter of T_{HWJit} to the timer drift, resulting in an accuracy of T_{HWJit} * $N_{MaxDiameter}$ = 160µs.

Since $T_{MaxDrift}$ = D_{Burst1} in this example, it is possible, due to propagation delay and due to the hardware jitter, that one short burst is seen as two individual short bursts, if two nodes with maximum delay are transmitting this burst. All services that can decode short bursts must be able to handle this.

6.4. Synchronization with fully distributed algorithm

This section describes the achievable accuracy and required time for synchronization with the fully distributed algorithm described in Section 5.4.

For every iteration of the fully distributed algorithm, only the transmission of one black burst is required. Every burst is to be followed by the smallest idle time D_{Idle0}. So one iteration has the duration D_{Burst1}+ D_{Idle0} = 1,192ms. For synchronizing across a maximum of $N_{MaxDiameter}$ = 5 hops, 1,192ms * 5 = 5.96ms is required. This is much faster than the master based algorithm which needs to transmit multiple bursts in every iteration.

The achievable timer accuracy is substantially lower when the fully distributed algorithm is used. Every hop adds T_{HWJit} + 2 * $T_{SwitchTX}$ to the maximum synchronization error. For a maximum diameter of 5 hops, the guaranteed maximum synchronization time with this algorithm is T_{HWJit} + 2 * $T_{SwitchTX}$ * $N_{MaxDiameter}$ = 2,08ms.

7. Conclusion & future work

We have presented MacZ, an adaptive QoS MAC layer that provides a decentralized synchronization of a wireless ad-hoc network. By changing the distribution and frequency of the transmission slots, the network developer can adapt MacZ to the needs of a specific application or domain. MacZ does not limit the number of nodes – however, the maximum diameter of the network must be predefined to ensure a correct synchronization for both algorithms. Removing this limitation is an area for future work.

The synchronized medium is able to offer a predictable set of time slots, which supports reservations for bandwidth and delay, as well as contention-free access to the medium. The reservation functionalities, as well as the functionalities for providing contention-based access are part of the MacZ service layer, which is the subject of ongoing work.

The application to a real hardware, as described in Section 6, has shown that the duration of the synchronization depends largely on the idle times that were chosen. With some optimization effort, a much shorter and more efficient synchronization sequence could be possible. To minimize the synchronization error, the accuracy of the hardware timer should be re-evaluated and probably increased. This would significantly reduce the jitter across multiple hops.

The possibility of replacing the micro protocols by others that provide the same functionality could also be evaluated. For example, networks that have an infrastructure could use a synchronization algorithm that supports a designated master.

8. References

[LWS04] L. Litz, N. Wehn, and B. Schürmann. *Research Center "Ambient Intelligence" at the University of Kaiserslautern*. In VDE Kongress 2004, volume 1, pages 19-24, Berlin/Germany, 2004. VDE Verlag, ISBN 3-8007-28273.

[FGGS05] I. Fliege, A. Geraldy, R. Gotzhein, P. Schaible: *A Flexible Micro Protocol Framework*. In D. Amyot, A.W. Williams (Eds.), System Modeling and Analysis, Lecture Notes in Computer Science 3319, Springer, 2005, pp. 224-236.

[Mil94] D. L. Mills. *Internet Time Synchronization: The Network Time Protocol*. In Zhonghua Yang and T. Anthony Marsland (Eds.), Global States and Time in Distributed Systems. IEEE Computer Society Press, 1994.

[Lam78] Leslie Lamport. *Time, Clocks and the ordering of events in a distributed system.* Communications of the ACM, 21(7): 558-67, 1978.

[YHE02] W. Ye, J. Heidemann, D. Estrin. *An Energy-Efficient MAC Protocol for Wireless Sensor Networks.* In IEEE INFOCOM, New York, June 2002.

[PDÖ02] A. Pal, A. Doğan, F. Özgüner. *MAC Layer Protocols for Real-time Traffic in Ad-hoc Wireless Networks.* In IEEE ICPP, 2002.

[CC02] R.Cunningham, V. Cahill. *Time Bounded Medium Access Control for Ad Hoc Networks.* In ACM POMC, Toulouse, France, October 2002.

[KRD04] S. Kumar, V. Raghavan, J. Deng. *Medium Access Control Protocols for ad hoc wireless networks: a survey.* Elsevier Ad-Hoc Networks Journal, 2004.

[EGE02] J. Elson, L. Girod and D. Estrin. *Fine-Grained Network Time Synchronization using Reference Broadcasts.* Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI),

Boston, MA, December 2002.

[MA98] A. Muir, J.J. Garcia-Luna-Aceves. *An Efficient packet sensing MAC protocol for wireless networks.* Mobile Networks Applic. 3 (3), pp. 221-234, 1998

[IEEE97] IEEE 802.11 Working group. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*. 1997.

[IEEE03] IEEE 802.15 WPAN™ Task Group 4. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks*. 2003.

[CC2420] Chipcon CC2420 Datasheet.
http://www.chipcon.com/files/CC2420_Data_Sheet_1_2.pdf

[SDL100] SDL standard. *http://www.sdl-forum.org/Publications/Standards.htm*