Advanced Mechanisms for Radio Base Station Synchronization

M. Sc. Thesis Rachid Ait Yaiz
21.01.1999

University of Twente

Advanced Mechanisms for Radio Base Station Synchronization

M. Sc. Thesis report Rachid Ait Yaiz
Period 01.05.1998 - 31.12.1998

Committee:
Dr.ir. G.J. Heijenk (Ericsson, CTIT)
Dr. P.F. Chimento (CTIT)
Dr.ir. S.M. Heemstra de Groot (TSS/EL)
Prof.dr.ir. I.G.M.M. Niemegeers (CTIT, TSS)

Ericsson Business Mobile Networks B.V.
Enschede, The Netherlands

University of Twente

Preface

From May 1998 till December 1998 I have done my M.Sc. assignment at the Core Department of Ericsson Business Mobile Networks in Enschede (The Netherlands).

Within the Core department, I worked on the topic of DECT Base Station Synchronization in which I first made a study of synchronization. Next I evaluated some possible mechanisms for DECT Base Station Synchronization. After the evaluation I have selected one mechanism which I evaluated further concluding with a description of the hardware implementation.

I would like to thank Dr. ir. Geert Heijenk, dr. Phil Chimento, dr.ir. Sonia Heemstra, dr. ir. Ignas Niemegeers, dr. Victor Nicola, the members of the Core department at Ericsson and ir. Cor van de Water for their support and advice during my thesis period.

Table of Contents

List of Figures

List of tables

1 Introduction

This report presents the results of an investigation on advanced mechanisms for base station synchronization. Nowadays DECT support besides speech also data communication. This has leaded to integration of DECT base station and Local Area Networks and resulted in the uselessness of conventional synchronization mechanisms.
In this chapter, we present the problem description in Section 1.1, the previous work in Section 1.2, the approach in Section 1.3 and the roadmap of this report in Section 1.4.

1.1 Problem description

Within wireless systems according to the Digital Enhanced Cordless Telecommunication (DECT) Standard, it is needed to synchronize the radio base stations within a system. This is for a number of reasons:

- to increase traffic capacity by minimizing interference between base stations
- to avoid large numbers of forced handovers (e.g. to other time slots on the same base station) because of interference from other base stations with sliding clocks
- to allow for fast (seamless) and predictable handovers between base stations

For Ericsson's current generation DECT systems, base stations are synchronized by distributing clock information from a central system using the (ISDN based) infrastructure which base stations are connected to. In the near future, we also expect base stations to be connected to infrastructures designed for packet data transfer, which are often less suited to transfer timing information e.g., because of unpredictable delays (Ethernet).

The purpose of the D-assignment is to investigate new, advanced mechanisms for base station synchronization. Alternatives to be considered include mechanisms to transfer clock information over data infrastructures, a common 2nd source, and synchronization over the air.

1.2 Previous work

A lot of work concerning DECT synchronization and synchronization via Ethernet has been done. DECT synchronization via wires is described in [ETS 300 175-2]. Requirements for DECT synchronization are worked out in the same reference. Synchronization via DECT Air Interface, in a non-fading environment, is discussed in [Daalen]. In this report we will investigate the effect of a fading environment on the synchronization via DECT Air Interface.
As far as synchronization via Ethernet is concerned, a lot of documents, concerning PC clock synchronization, are written. In all those documents accuracy of milliseconds is reached. As will be seen in Section 3.5, this accuracy is not sufficient. The mentioned PC clock synchronization documents can be found on the World Wide Web (e.g. [PCClock1], [PCClock2] and [PCClock3]). In this report we will investigate if higher accuracy's are possible when synchronizing via Ethernet.

1.3 Approach

The D-assignment comprises the following phases:

- A study of DECT (Physical layer, MAC layer).
- A study of the requirements / criteria with respect to base station synchronization (e.g., accuracy, stability, cost, fault tolerance).
- Evaluation of the possible alternatives.
- Based on a first evaluation, select one mechanism, and design it in detail.
- Consider implementation aspects, and if possible, test design by prototype implementation

1.4 Roadmap

In Chapter 2, an overview of the Digital Enhanced Cordless Telecommunication (DECT) is given. Chapter 3 describes the need for synchronization, the synchronization types and the DECT requirements for synchronization. A synchronization architecture will also be given. Chapter 4 will investigate some possible solutions. The distribution algorithm, performance aspects, cost and complexity will be compared in order to choose the best option. Chapter 5 will explain base station synchronization via Ethernet. First the Ethernet protocol is clarified. Next a more detailed evaluation of the performance of Ethernet is given. After presenting the transmission and reception requirements a delay compensation technique is presented. The chapter is concluded with the description of the hardware implementation. Finally Chapter 6 summarizes the conclusions and recommendations.

2 Digital Enhanced Cordless Telecommunications

2.1 Introduction

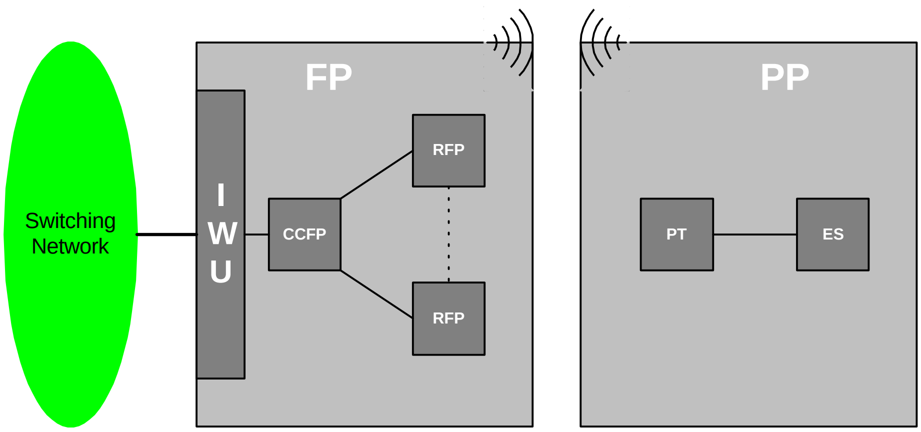
In order to understand the context of the synchronization problem, the environment in which it takes place must be understood. Therefore the relevant aspects of DECT will be explained in this chapter.

Wireless technologies are split into two categories, Mobile and Cordless. Besides a different historical background, the cordless and mobile technologies differ in the mobility they offer.
Base stations of a cordless system are spaced from 10 meters (in indoor application) up to a few kilometers (for outdoor applications) while base stations for mobile terminals were designed to be used over large distances.

DECT stands for Digital Enhanced Cordless Telecommunications. This European standard for digital cordless systems was defined by the Conference of European Posts and Telecommunication (CEPT) and the European Telecommunications Standards Institute (ETSI). It was ratified in July 1992 [EMN94].

2.2 Architecture of DECT systems

DECT is an access technology [DECTCD]. The standard describes how to provide cordless access to a large number of networks. Only the air-interface and the controlling parts on both sides of the air-interface are specified, whereas the local and public switching networks are not.



FP	= Fixed Part
PP	= Portable Part
IWU	= Intwerworkin unit
CCFP	= Common Control Fixed Part
RFP	= Radio Fixed Part
PT	= Portable Radio Termination
ES	= End System

Figure 1. Architecture of DECT systems

Figure 1 gives the architecture of DECT systems. There is a switching network to which DECT provides cordless access. This can be a public switched network (PSTN), a private branch exchange (PBX), a GSM network and so on. A DECT system always consists of a Fixed Part (FP) and a Portable Part (PP).

The Fixed Part contains all the DECT elements on the network side and connects to the switching network via an Interworking unit (IWU). The Portable Part contains all DECT elements on the user side.
Communication between the Fixed Part and the Portable Part is achieved by means of radio signals. This is called the DECT Common Interface (CI) or DECT Air Interface.

The Fixed Part consists of a Common Control Fixed Part (CCFP) and one or more Radio Fixed Parts (RFP also called base station). The CCFP provides transparent access to the switching network, which means that all the switching network functionality and services are available for the user. The CCFP also controls all traffic and signaling between the network and the RFPs.

The RFP provides the radio functionality on the fixed side of the DECT Air Interface, containing a transmitter/receiver. The RFP also takes care of modulation and demodulation of the digital speech or data. Single cell DECT systems are DECT systems with one RFP. By adding RFPs a multi-cell system can be built. This is a system with two or more overlapping cells, forming one large coverage area in which calls can be handed over from one cell to another.

The Portable Part consists of the Portable radio Termination (PT) and the End System (ES). The PT provides the radio functionality on the portable side of the DECT Air Interface containing a transmitter/receiver. The PT also takes care of the modulation and demodulation of the digital speech or data.
The ES is the application that gives the user cordless communication. This can be a telephone, computer, printer and so on.

2.3 DECT Protocol Layers

DECT's architecture covers only the lower three layers of the Open System Interconnection (OSI) reference model, as higher functionality is not required, leaving a high degree of freedom for further development.

OSI takes no account of radio transmission uncertainties and handovers. DECT consists therefore of four layers and also a management entity, as can been seen in Figure 2. Figure 3 shows the relationship between the data units that are used by the individual layers.

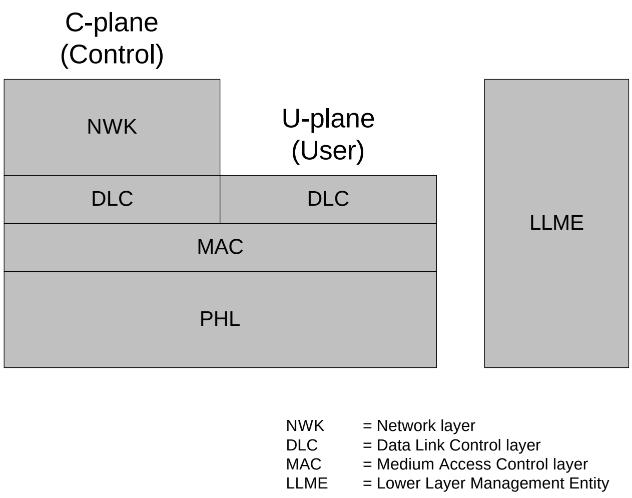


Figure 2. DECT layers

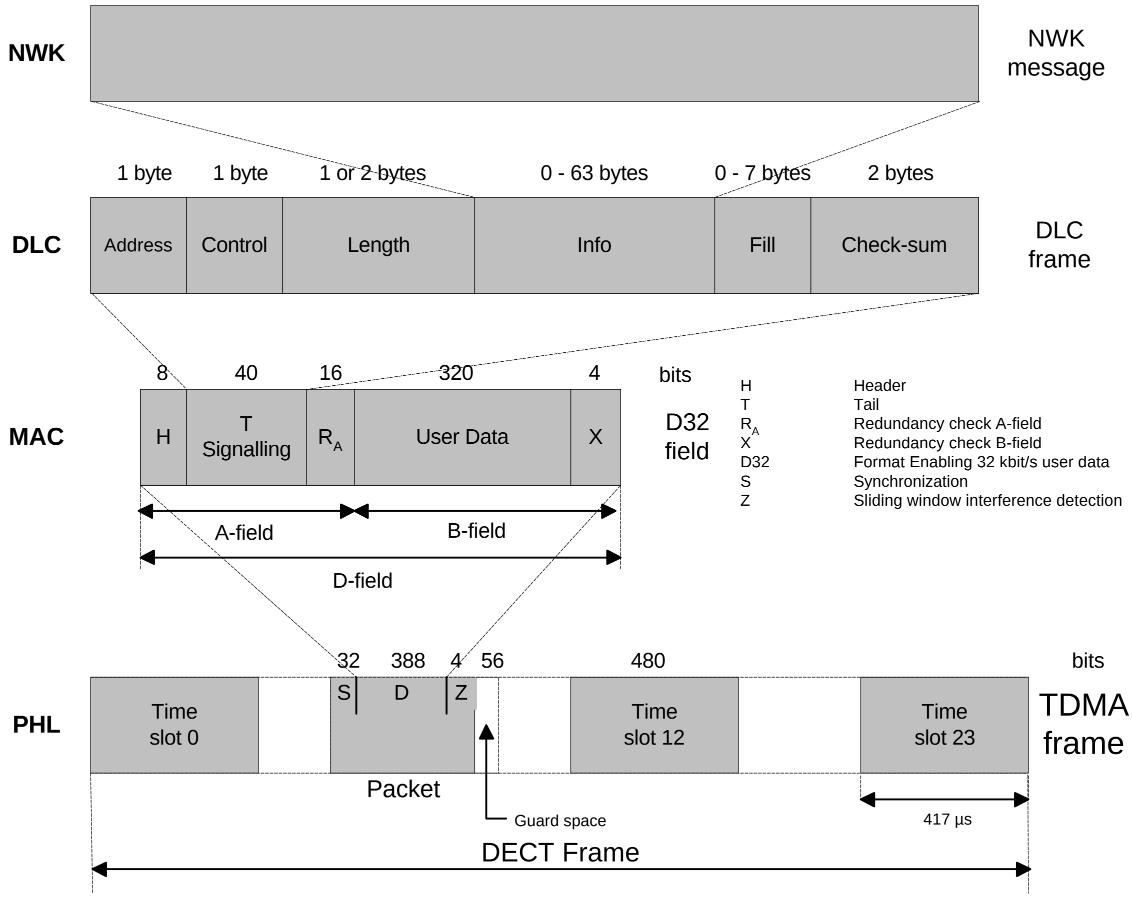


Figure 3. Structure of DECT frames and packages

<u>Remarks on Figure 3:</u>

- D32 means that a user data rate of $100\dfrac{frames}{\sec}\cdot 1\dfrac{slot}{frame}\cdot 320\dfrac{bit}{slot} = 32kbit/\sec$ is provided. Other bitrates are also possible. D80 for example uses two slots to provide a bitrate of $100\dfrac{frames}{\sec}\cdot 1\dfrac{slot}{frame}\cdot(320+480)\dfrac{bit}{slot} = 80kbit/\sec$ (second slot contains only user data).
- NWK is used for control plane messages.

2.3.1 Physical Layer (PHL)

The purpose of the physical layer is to divide the assigned radio spectrum into physical channels. There are 10 available carrier frequencies in the 1880 - 1900 MHz band (The DECT standard also provides for possible extension of the band to meet future demand). Each carrier frequency is subdivided into timeslots numbered 0 to 23 as can be seen in Figure 4. Time-slots 0 to 11 are used for the downlink (base station to portable), whereas time-slots 12 to 23 are used for the uplink (portable to base station). Time-slots 0 to 23 can also be used for data transmission, but all transmissions in time-slots 0 to 11 must be in the same direction. The transmissions in time-slots 12 to 23 must also be in the same direction but not necessary the same transmission direction as in time-slots 0 to11.

This access technique is known as Time Division Multiple Access / Time Division Duplex (TDMA/TDD). It provides 12 duplex channels on each carrier leading to a total of 120 channels (with 10 carrier frequencies). Table 1 depicts these and other radio-related characteristics specified by the Physical Layer.

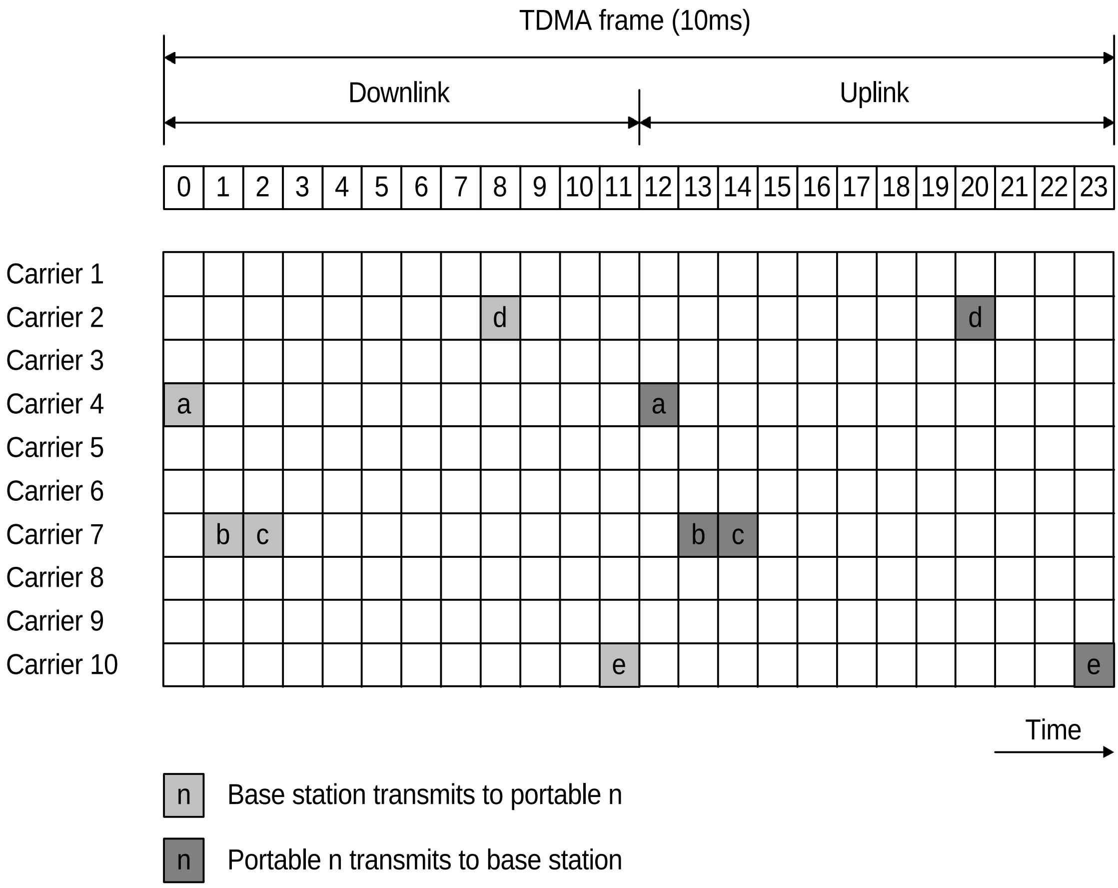


Figure 4. DECT channel allocation

Frequency band	1880 - 1900 MHz
Number of carriers	10
Carrier spacing	1728 kHz
Carrier modulation	GFSK (Gaussian Filtered Frequency Shift Keying)
Peak transmit power	250 mW
Channel selection	CDCS (Continuous Dynamic Channel Selection)
Gross bit rate	1152 kbps
Carrier multiplex	TDMA/TDD; 24 slots per frame (12 duplex)
Total number of channels	120
TDMA frame duration	10 ms
TDMA slot duration	0.417 ms

Table 1. Characteristics of the DECT physical layer

2.3.2 Mac Layer

The DECT Mac layer performs two functions:

1. Selection of physical channels, establishment and release connections on those channels.
2. Multiplexing control information with higher layer information and error control information, into slot-sized packets.

The MAC layer divides each data package sent from the physical layer into an A-field and a B-field. The A-field is used to transport control information for the higher layers. The B-field is normally used to carry user data. To enable reliable data transmission, each field has a separate redundancy check field.

Three independent services provided by the MAC layer are [ETS 300 175-3]:

1. Connection oriented services
2. Connection less services
3. Broadcast services

2.3.2.1 Connection Oriented Services

The DECT MAC layer provides two types of logical channels for connection-oriented services, named I_N-channel and I_P-channel. They are used for the exchange of user data in the B-field (U-plane).

The I_N-channel provides a limited X-CRC error detection (see Section 2.3.2.4) on the user data. This is sufficient for speech, and a data rate of 32 kbps. The I_p-channel offers a better R-CRC error detection (see Section 2.3.2.4) capability at the expense of a lower data rate of 25.6 kbps. The I_p-channel is used for services like facsimile, modem and data.

We can distinguish between services on the C-plane. As Figure 3 depicts, higher layer (C-plane) information is transmitted in the tail field. This is called the C_s higher layer channel associated signaling and offers a capacity of 40 bits per C_s segment. Larger messages will be segmented and sent using more segments. With a maximum number of 8 C_s segments per Multiframe, the maximum signaling capacity is:

$$C_{\max} = \frac{8 \cdot 40}{160 \cdot 10^{-3}} = 2kbit / s \qquad [1]$$

The information is protected by the MAC layer control, which uses error correction (A-field CRC).

The C_f higher layer associated signaling on the other hand provides 25.6 kbps signaling transmission capacity by using the B-field for transmission of C-plane data.

The receiving side has to acknowledge the correct reception of the signaling data. For this purpose C_s and C_f higher layer signaling is transmitted with ARQ (Automatic Repeat Request) window size (currently 1). The ARQ window size determines how many successive frames can be acknowledged at one time. The receiving side does this by setting two bits in the header of the next MAC frame it sends; thus it is not necessary that an extra frame has to be transmitted for acknowledgement.

2.3.2.2 Connectionless services

The connectionless services allow multicast transmission of higher layer C-plane and U-plane data from FT (Fixed Terminal) to PT (Portable Terminal), and point to point transmission of higher layer C-plane data from a PT to one FT. These services are controlled by the CMC (Connectionless Message Control). The FT to PT connectionless service may be continuous (i.e. one transmission in every frame). In the direction PT to FT, transmission is limited to a maximum of two slots in two successive frames.

2.3.2.3 Broadcast services

The broadcast service is a special service by which a portable can quickly identify all the base station within its range. These "beacon" transmissions are always present on at least one physical channel per base station.

We distinguish two broadcast services; namely a continuous broadcast service or a non-continuous broadcast service.
The continuous broadcast service is a simplex service in the direction FT (Fixed Terminal) to PT (Portable Terminal). This type of broadcast allows PTs to lock on a FT and to obtain access rights and service related information (called N and Q channel). The service also takes care of transmission of paging messages (called P channel) from FT to PT. These paging messages are used for instance for the Call Set-up procedure. The continuous broadcast service is available on all bearers with continuous transmissions in the direction FT to PT.

The non-continuous broadcast service allows the PTs to obtain extended system information on request. The service needs a limited number of transmissions in both directions. The request and reply data are transmitted either in the A-field or in the B-field.

2.3.2.4 Error control [ETS 300 175-3]

The MAC layer provides error control for all logical channels, using a combination of Two Cyclic Redundancy Codes (CRC):
- R-CRC; a 16 bit CRC
- X-CRC; a 4 bit CRC

Besides these error detecting mechanisms, an automatic repeat request (ARQ) mechanism for the I_p channel is also used.

<u>R-CRC overview</u>

The R-CRC is used to provide the main MAC layer error control. The MAC layer calculates 16 redundancy bits over several fixed length data blocks:
- All A-fields
- All B-subfields in protected format

In each case, the redundancy bits are appended to the data blocks and allow a redundancy check in the receiver.

<u>X-CRC overview</u>

For the error control of B-field data, a limited error detection scheme is always applied, even for unprotected B-field formats. The MAC layer calculates 4 redundancy bits from selected B-field data bits. These four bits are transmitted in the X-field. The X-field occupies the last four bits of the B-field.

<u>ARQ in I_p channel</u>

The ARQ in I_p channel is also known as MOD-2 retransmission scheme for I_p data. This modulo-2 procedure uses a 2-state packet number in the A-field header. This packet number applies to the complete B-field of I_p data. The first I_p packet sent on a logical bearer is labeled with packet number "1".

Successful reception of the data is acknowledged independently for each logical bearer. Following successful acknowledgment, the transmitter may advance to the next packet, toggling the packet number. If a packet is not received correctly, the receiving side requests that the sending side to transmit the last packet again until no errors are detected or until a timer expires.
If the transmitter has no new packet to send, it may repeat the old data.

2.3.3 DLC Layer

The DLC layer provides reliable data links. It provides higher levels of data integrity than can be provided by the MAC layer. The DLC layer is divided into a part used by the Control-plane (C-plane) and a part used by the User-plane (U-plane).

The C-plane DLC provides a reliable transmission of data, used for transmission of signaling information. The U-plane provides a range of alternate services, with a varying degree of reliability, optimized to the specific type of service. When, for instance, speech transmission is required; an unprotected service is used.

As can be seen in Figure 3, DLC frames are allowed to be of much greater size than a single MAC Tail field. The reason for this is that the DLC layer takes care of segmenting and re-assembling individual sub-messages.

2.3.4 Network Layer (NWK)

As Figure 2 shows, only the Network layer uses the Control-plane. The Control-plane is used for signaling purposes. The Network Layer provides the DECT service primitives to the user. These services are grouped into the following families of signaling procedures:

- Call Control (CC)
- Supplementary Services (SS)
- Connection-Oriented Message Service (COMS)
- ConnectionLess Message Service (CLMS)
- Mobility Management (MM)
- Link Control Entity (LCE)

Mobility Management contains the procedures that support the special cordless mobility of portable parts, for example authentication and location registration.

2.3.5 Lower Layer Management Entity (LLME)

The LLME performs the following functions:

1. Channel management
2. Connection management
3. Service management

Due to these functions the LLME deals with all four layers of DECT as can be seen from Figure 2. The management procedures for the different layers are:

- MAC layer

1. Creation, maintenance and release of bearers, by activating and deactivating pairs of physical channels.
2. Physical channel management, including the choice of free physical channels and the assessment of the quality of received signals.

- DLC layer

1. Connection management, for example the establishment and release of DLC connections.
2. Routing of C-plane and U-plane data to suitable connections.

- NWK layer

1. Service negotiation and mapping.

2.4 Datalab

Datalab is an Ericsson project aiming to investigate several aspects concerning data over DECT. Within Datalab a testbed is introduced. This testbed consists of the following components (Figure 5):

- DECT PCMCIA cards interface to notebooks and/or handheld computers running Windows 95 and Windows CE.
- PC Base Stations attached to Ethernet LAN supporting multi-slot DECT traffic.
- A standard router for Inter(/Intra)net access.
- Mobility gateway (a PC) with DECT higher layer software for call control and mobility.

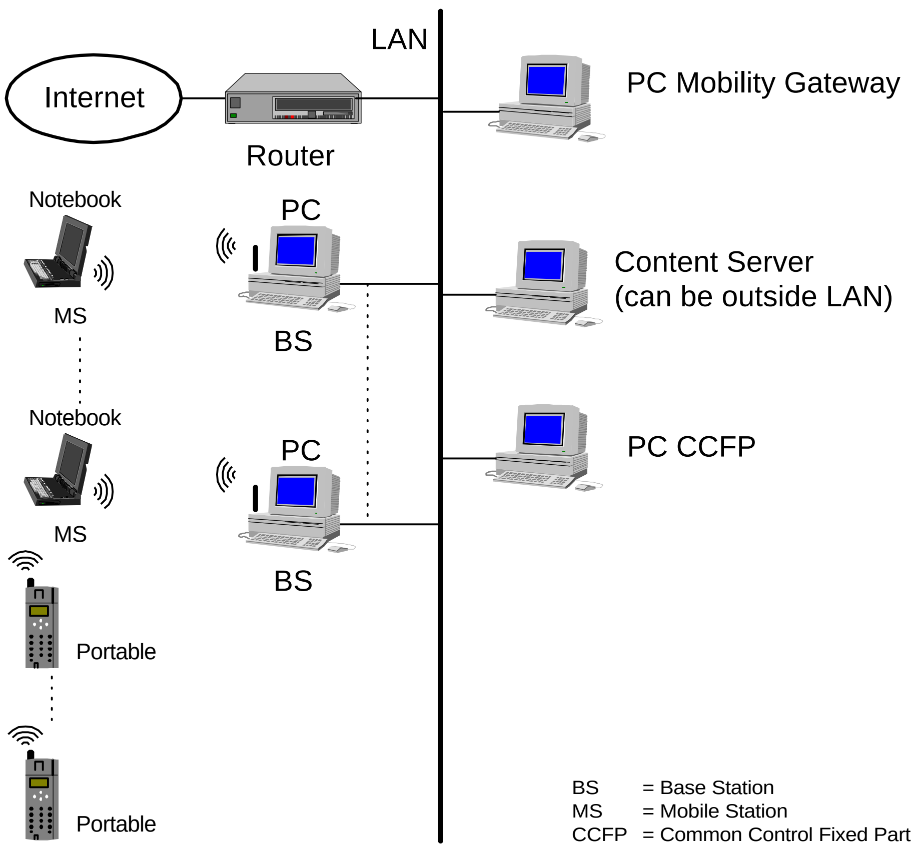


Figure 5. Test-Bed Set-Up

This testbed is used to test new applications and techniques and contains multiple base stations. Due to reasons explained in Section 3, it is necessary to design a good solution for synchronization of base stations, which can be used in future products based on the results of the Datalab project.

3 Synchronization

3.1 Introduction

In this section we will discuss some basic aspects of synchronization. First, the need for synchronization will be explained in Section 3.2, Frame synchronization is explained in Section 3.3 and in Section 3.4 frame number synchronization will be explained.
From DECT a number of requirements concerning synchronization are defined. These requirements will be listed in Section 3.5. Finally a generic synchronization architecture will be explained in Section 3.6.

3.2 Need for synchronization

There are three major reasons for synchronization:

- Increase traffic capacity.
- Decrease the number of forced handovers.
- Provide handover between base stations.

These reasons are discussed in Section 3.2.1, Section 3.2.2 and Section 3.2.3.

3.2.1 Increasing traffic capacity

Portable parts located in regions covered by one or more base station, must search for an unused time slot in a frame if they want to communicate with a base station. With an unused time slot we mean a time slot where during the whole time slot no other transmission at the same carrier frequency takes place. If frames from different base stations are not aligned, a used time slot on a base station may mean two used time slot on another base station. This is illustrated by Figure 6.

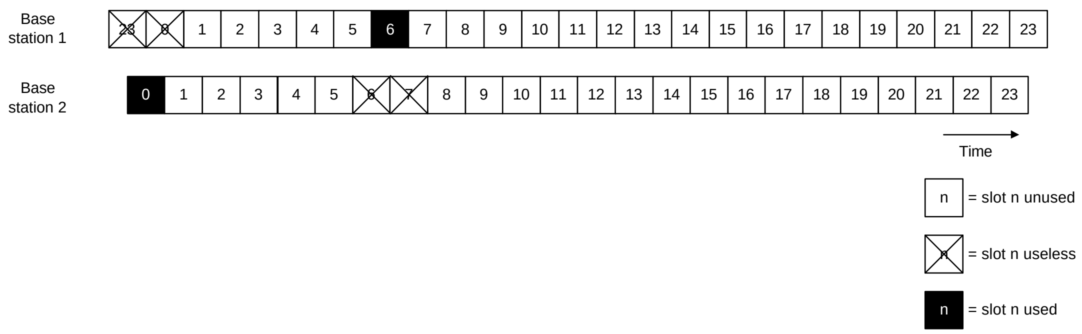


Figure 6. Capacity decrease due to unaligned frames

We see that if base station 1 uses time slot 6, base station 2 is not allowed to use time slot 6 and 7 on the same carrier frequency. We also see that when base station 2 uses time slot 1, base station 1 is not allowed to use slot 0 and 1 on the same carrier frequency. It is clear that when a base station uses a time slot on a particular carrier frequency, it makes two time slots on the same carrier frequency useless for the other base station leading to a decrease of capacity. Aligning the start of frames from different base stations is required to solve this problem.

In the DECT standard [ETS 300 175-2] this requirement is known as Class 1 synchronization.
Class 1 synchronization is intended for increased traffic capacity of adjacent systems by aligning guardbands. This class provides frame synchronization, which is explained in Section 3.3.

3.2.2 Decreasing the number of forced handovers

When alignment of frames from different base station does not take place, there will be a time difference between the start of frames of different base stations. This difference is variable in time due to the drift of the reference timers[i] of the base stations. This means that when a base station uses a particular time slot n it is possible that a used time slot m from another base station shift into time slot n forcing both base station to perform a handover to an unused slot. This is illustrated by Figure 7.

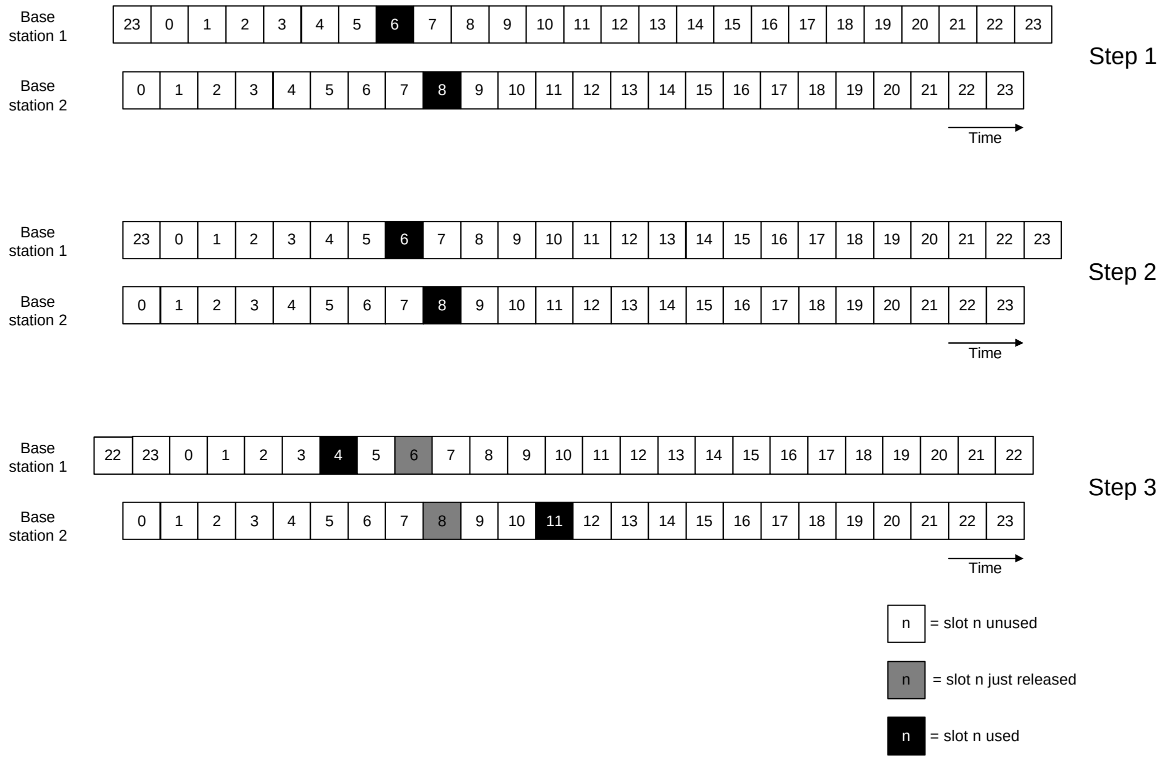


Figure 7. Forced handover due to sliding frame

In Figure 7 we see that base station 1 use time slot 6 and base station 2 uses time slot 8 on the same frequency. We also see that the reference timer of base station 1 is slower than the reference timer of base station 2. As a result time slot 6 of base station 1 and time slot 8 of base station 2 will overlap after a given time (step 3), leading to a forced handover to another time slot on both base stations.
Synchronizing reference timers of different base stations solves this problem.

3.2.3 Providing handover between base stations

Consider a DECT system consisting of more than one overlapping base station and handover between these base stations is required. During the handover process, a portable part must be able to lock on both base stations that are involved in the handover process.
According to [ETS 300 175-2], a portable part can lock on a base station only if the time difference between the start of the frame from that base station and the start of the frame from the portable part is less than ± 4 bits ($\pm 3.47 m \sec$ at 1152000 bps). As a result, the time difference in the start of frames from different base stations is not allowed to exceed 8 bits ($6.94 m \sec$). Figure 8 show the maximum time difference between the reference timers of base stations involved in the handover process, such that portable part locking on both base stations is still possible, which means that a handover requirement is met.

[i] "The reference timer of a RFP or a PP is a notional clock to which the timing parameters of the TDMA framing are related" [ETS 300 175-2].

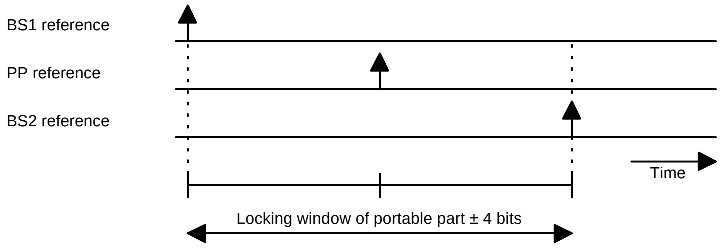


Figure 8. Locking window of a portable part

In the DECT standard [ETS 300 175-2] this requirement is known as Class 2 synchronization. Class 2 synchronization is intended for the case when handover has to be provided between base stations. This class provides frame synchronization and Multiframe synchronization. Frame synchronization is explained in Section 3.3 while Multiframe synchronization is explained in Section 3.4.1.

3.3 Frame synchronization

Frame synchronization means that the time difference between the start of frames from different base stations is bound. This is shown by Figure 9.

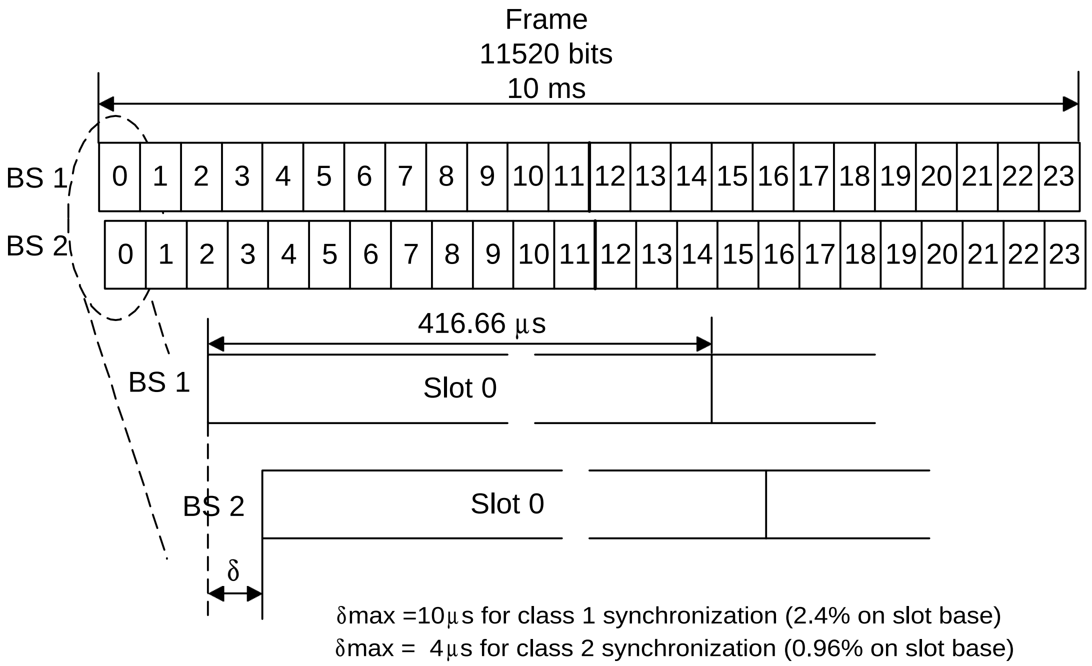


Figure 9. Frame Synchronization

Remarks:

- If the start of frames is synchronized the slots and slot numbers are also synchronized.
- Class 2 allows a maximum time difference between the start of frames from different base stations equal to $d_{\max} = 4ms$, which satisfies the handover requirement mentioned in Section 3.2.3.

An important fact is to keep in mind that the time differences in the start of frames belonging to different base stations are actually the time differences (offset which can also be variable due to drift) between the reference timers of the Radio Fixed Parts (RFP). We will see in Section 3.5 that the jitter[i] of an RFP packet transmission shall be less than ± 1 µs at extreme[ii] conditions. As a result the time differences between the start of frames from different RFPs at the Air Interface can reach a value of δ+2 µs, with δ depending on the synchronization class (see Figure 9).

[i] The jitter of an RFP packet transmission in a slot refers to the occurrence of the start of the first bit of that packet at the antenna. The jitter is defined in relation to the reference timer of that RFP. [ETS 300 175-2]

[ii] Extreme condition when environment temperature is below 15°C or above 35°C. [TBR 006]

3.4 Frame number synchronization

In the previous sections we discussed frame synchronization where the time difference between the start of frames from different base stations is bound. In this section we discuss frame number synchronization.

Every frame sent by a base station has a 28 bits frame number. By letting frame numbers of frames from different base stations be linked to each other in a predefined way, a specific frame number synchronization is performed. We distinguish the following types frame number synchronization:

- Multiframe synchronization (explained in Section 3.4.1)
- Multiframe number synchronization (explained in Section 3.4.2)
- PSCN synchronization (explained in Section 3.4.3)

3.4.1 Multiframe synchronization

Once every 16 frames (Multiframe) a transmitting RFP places system information in the Q Tail (See Tail in A-field in Figure 3). This information is used by the portable part for locking on the RFP and for bearer set-up. Examples of this information is slot number, carrier number and Primary Scan Carrier Number (PSCN explained in Section 3.4.3).
DECT frame reception by a portable part is accompanied with energy (battery) consumption. To save energy at the portable part, the portable part has to know when it must look for the mentioned information (once every Multiframe). In other words the portable part has to know which frame within a Multiframe contains the Q Tail. This means that the portable part and the base station with which it communicates must be Multiframe synchronized.
When a portable part wants to initiate a seamless handover from one base station to another, it must also be Multiframe synchronized with the base station it wants to be connected with after the seamless handover. In order to make it possible for a portable part to be Multiframe synchronized with all base station, all base stations must be Multiframe synchronized with each other.. Figure 10 shows the Multiframes of two Multiframe synchronized base stations.

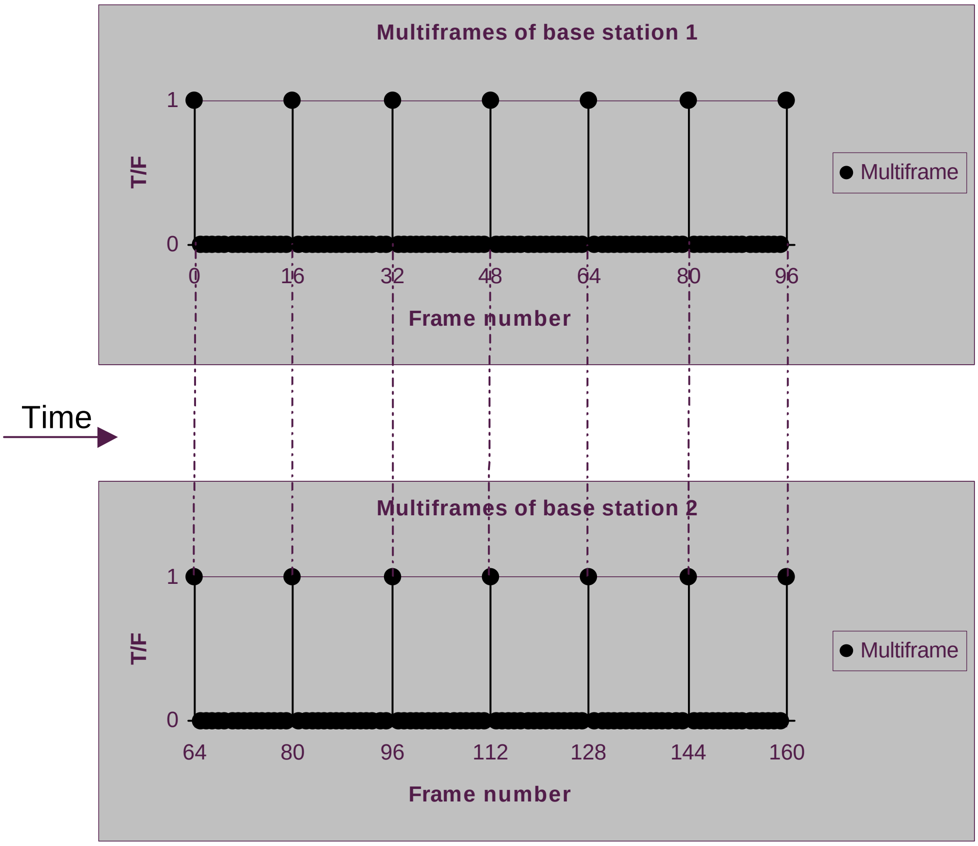


Figure 10. Multiframe Synchronized (10 carrier frequencies)

Multiframe synchronization means that the four Least Significant Bits (LSB) of the frame numbers (each frame has a 28 bit frame number) of frames from different RFPs have to be equal. In this Figure we see that the four Least Significant Bits (LSBs) of the frame numbers of the two Base Stations are equal (equal frame number within a Multiframe), and Multiframes occur at the same time. This is also necessary for Scrambling[i] [ETS 300 175-3], where the frame number within a Multiframe is used for Scrambling.

i Scrambling is a mechanism used to avoid long “0” or “1” sequences occurring several times due to unaltered data or retransmission protocols [ETS 300 175-3].

3.4.2 Multiframe number synchronization

Because the encryption[i] mechanism uses the Multiframe number (MFN) at both a base station and a portable part, Multiframe number synchronization between a base station and a portable part is required for encryption [ETS 300 175-3]. With Multiframe number synchronization, 24 MSBs of the 28 bits of a frame number have to be equal.
When a portable part wants to initiate a seamless handover from one base station to another, it must also be Multiframe number synchronized with the base station it wants to be connected with after the seamless handover. In order to make it possible for a portable part to be Multiframe number synchronized with all base station, all base stations must be Multiframe number synchronized with each other. Together with Multiframe synchronization (equal 4 LSBs of the 28 bits of the frame number), this means that frame numbers are equal. We see this in Figure 11.

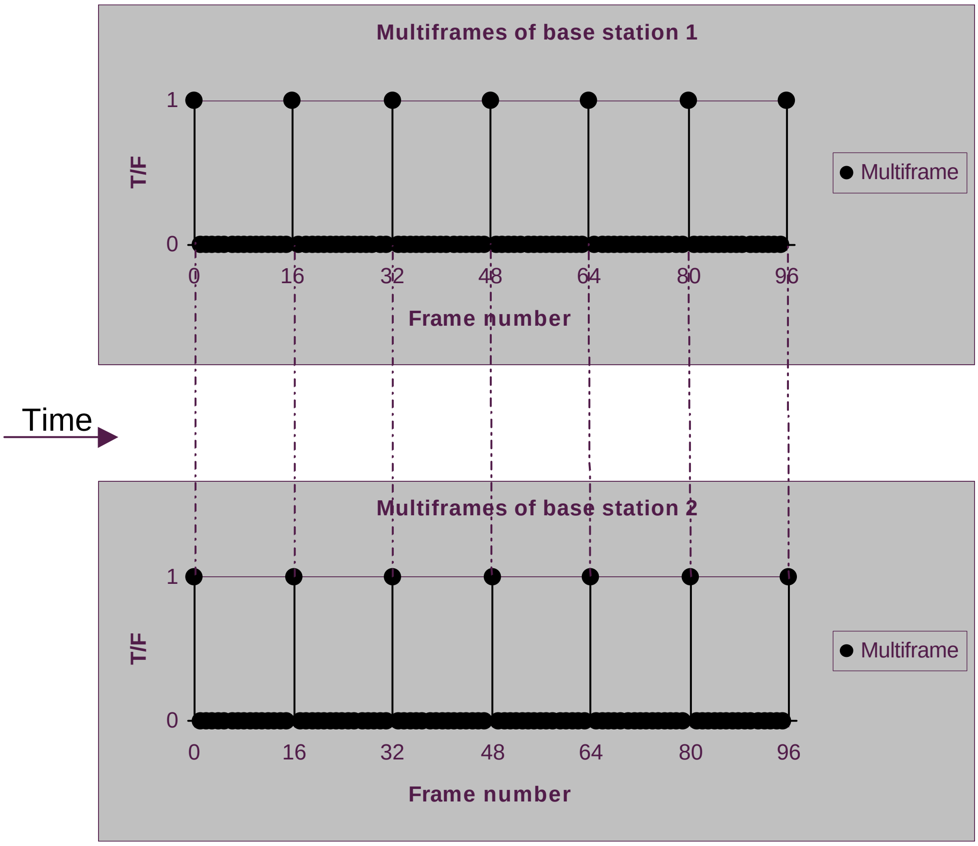


Figure 11. Multiframe and MFN synchronization (10 carrier frequencies)

i Encryption is a privacy mechanism which may be provided to encrypt data of a connection oriented call [ETS 300 175-3]. Encryption applies only to the B-field whereas R-CRC and X-CRC bits are never encrypted.

3.4.3 PSCN synchronization

Every DECT frame, base stations scan the Air Interface during the uplink period for the presence of a bearer request made by a portable. Each frame a new carrier frequency (from the set of allowed[i] carrier frequencies) is examined. The carrier frequency that will be scanned during the next frame is given by the Primary Scan Carrier Number (PSCN).

A portable part that wants to have a connection set-up with a base station must know the PSCN of that base station. As a result the portable part can place a bearer request at the carrier frequency that will be scanned by that base station during the next frame. Otherwise a portable part may place a bearer request on a frequency that is just scanned, which means that the base station will not see the portable.
When a portable part knows the PSCN of a base station, we say that the portable part is PSCN synchronized with that base station.

When a portable part wants to initiate a seamless handover from one base station to another, it must also know the PSCN of the base station it wants to move to. This means that the portable part must also be PSCN synchronized with the base station it wants to be connected with after the seamless handover.
In order to make it possible for a portable part to be PSCN synchronized with all base station, all base stations must be PSCN synchronized with each other.

There is no absolute relation between a frame number and the start of a PSCN cycle[ii]. A base station has to know the PSCN of the base station it wants to synchronize with, and so modify its own PSCN to start its PSCN cycle, synchronous with that base station.

[i] Within a DECT system, the use of particular carrier frequencies can be prohibited. We then say that these frequencies are not allowed.

[ii] PSCN cycle is a set of serial numbers of allowed (occupied or not) carrier frequencies (e.g. 0 to 9 in the case of 10 allowed carrier frequencies). The size of a PSCN cycle is equal to the number of allowed carrier frequencies in the DECT system.

3.5 DECT requirements

According to [ETS 300 175-2], a multi channel RFP shall have its reference timer accuracy[i] better than 5 ppm (parts per million) at normal[ii] conditions and better than 10 ppm at extreme[iii] conditions (worst case).

In Section 3.3 we have mention that the difference between the start of frames from different base station must be bound. Due to the inaccuracy of the reference timers of the slave base stations there will be an increasing time difference between the start of frames from the master base station (accurate, e.g. 0.1 ppm) and the start of frames from a slave base station. This time difference will increase in time. The lower the accuracy of the reference timer of a slave base station the faster the maximum time difference is reached.

Given the accuracy (acc) of the reference timer of a slave base station and the maximum allowed time difference ($\pm\frac{d}{2}$) between the start of frames from that base station and the start of frames from the master base station, the maximum time between two synchronization pulses can be derived. We call this maximum time the Maximum Inter-Synchronization pulse Time ($MIST$). The relation between these quantities is shown by Figure 12.

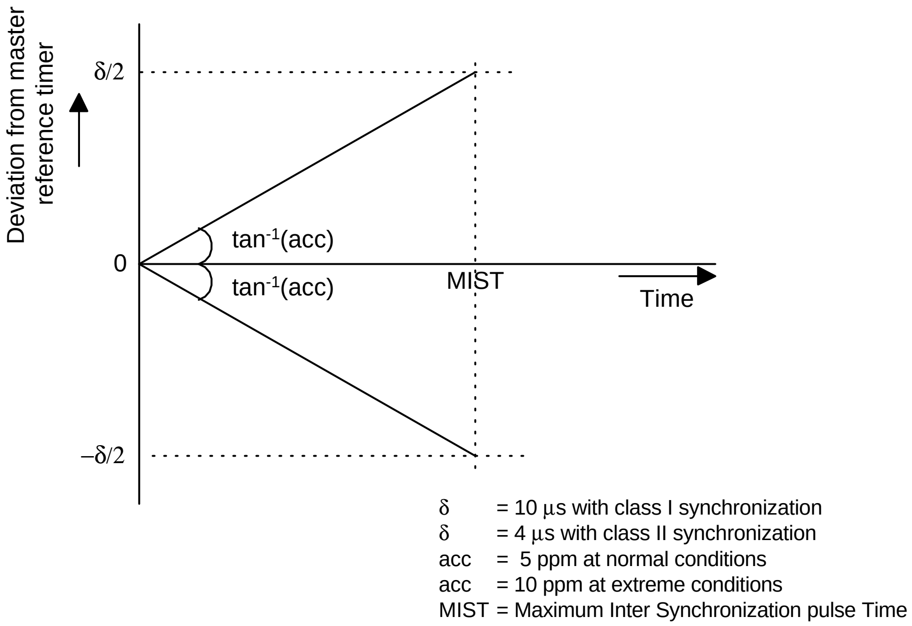


Figure 12. Relation between accuracy and synchronization pulse

At time 0, the slave base station is just synchronized and we assume the deviation of the reference timer of the slave base station from the reference timer of the master base station to be zero at that moment.

Considering accuracy acc, the maximum allowed deviation $\pm\frac{d}{2}$ (see Section 3.3) would be reached within

[i] When a crystal clock is produced with the intention to have a clock frequency of f_c and it has an accuracy of x ppm, the measured frequency will be in the range of $[f_c \cdot (1 - x \cdot 10^{-6}), f_c \cdot (1 + x \cdot 10^{-6})]$. However, the measured frequency is almost constant when using crystal clocks. This means that once a crystal clock shows a deviation from the intended frequency, this deviation does not change much.

[ii] Normal condition when the environment temperature is within the range of 15°C to 35°C [TBR 006].

[iii] Extreme condition when environment temperature is below 15°C or above 35°C [TBR 006].

a time span of $MIST$. In other words a valid synchronization pulse must be received by the slave base station at least once every time period equal to $MIST$.

The relation between the maximum difference in the reference time of slave base stations and a master base station, the accuracy of the reference timer of the slave base stations and the Maximum Inter-Synchronization pulse Time is given by the next formula:

$$\tan(\tan^{-1}(acc)) = acc = \frac{d/2}{MIST} \qquad [2]$$

In this formula, $MIST$ is the maximum time period between two successive synchronization pulses. The values of Maximum Inter-Synchronization pulse Time ($MIST$) under different circumstances are shown in Table 2.

	Condition	
	Normal	extreme
class I	1 s	0.5 s
class II	0.4 s	0.2 s

Table 2. *MIST* under different circumstances

Jitter of an RFP packet transmission in a slot, is the variable time difference between the production of a packet and its appearance at the antenna. According to [ETS 300 175-2] this jitter shall be less than ± 1 µs at extreme conditions (in relation to the reference timer of that RFP). This means that the start of frames of different RFPs at the Air Interface may differ $\pm(\frac{d}{2}+1)m\sec = \pm(2+1)m\sec = \pm 3m\sec$ from the reference timer when class II synchronization is provided.

Important DECT requirements are summarized below [ETS 300 175-2]:

1) The synchronization signal shall have long-term frequency accuracy better than ± 5 ppm at nominal conditions or ± 10 ppm at extreme conditions.

2) The jitter of an RFP packet transmission in a slot shall be less than ± 1 µs at extreme conditions (in relation to the reference timer of that RFP).

3) The difference between reference timers of RFPs of the same FP shall be less than 4 µs if handover is required between these RFPs.

4) Related to its reference timer, the RFP synchronization window should be at least ± 14 bits ($12.15m\sec$), when expecting a first reception and if intracell handover is provided, else ± 4 bits ($3.47m\sec$).
   The synchronization window is the maximum size of the time difference (in bits) between the start of frames from synchronizing base stations, such that synchronization is still possible.

5) To obtain system and inter-system synchronization, an RFP may alter the length of a single frame by any amount, or, it may alter the length of successive frames by up to 2 bits (this in accordance with the reference timer stability and accuracy requirements).

6) Base station can be in master mode, generating synchronization pulses via a port called SYNC-OUT, or they can be in slave mode receiving synchronization pulses via a port called SYNC-IN.
   Base stations in slave mode also can pass the incoming (via SYNC-IN) synchronization pulses to other base station via SYNC-OUT. The DECT standard mentions a requirement concerning

the time difference between the incoming synchronization pulse and the outgoing synchronization pulse: “If a valid synchronization signal is detected at the SYNC-IN (synchronization input port), the slave base station shall regenerate the signal at its SYNC-OUT (synchronization output port). The propagation delay in the regenerated signal, between the input and output synchronization ports shall not exceed 200 ns”.

From the user’s point of view, there are also requirements. Some of those requirements are:

1) Adding synchronization capabilities to existing DECT systems, without adding extra wires.

2) Low probability of a synchronized system losing synchronization (e.g. once a month).

3) When new base station have to be added to a synchronized DECT system, there must be no need for manual set-up procedures. In other words, just plug the new base station in to the existing DECT system and it will synchronize (after some time) with the rest of the system.

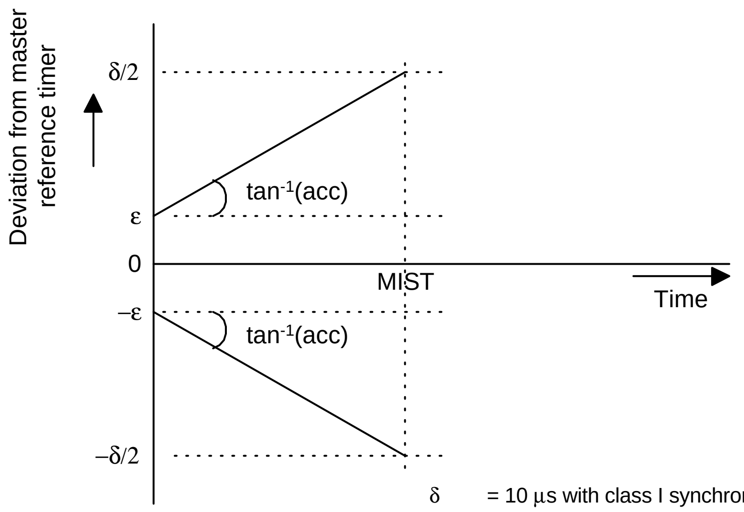
4) When base stations synchronize over a given medium, the propagation delay influences the performance of the synchronization mechanism. It is required that this delay is compensated for without the need for manual calculation.

5) No excessive increase in price when adding synchronization functionality to a system.

3.5.1 Exchanging MIST and Synchronization Jitter

In the previous section, we assumed that a synchronization pulse, wherever generated, arrives within a constant time at an RFP. With knowledge of this constant time we can compensate for the delay, providing synchronization of RFPs within a given error bound.

In the system to be developed, we allow some jitter in the arrival of a synchronization pulse at an RFP in order to allow a solution with possibly variable delays of the synchronization pulses. Figure 13 show the impact of these variable delays on the Maximum Inter-Synchronization pulse Time.



δ = 10 µs with class I synchronization
δ = 4 µs with class II synchronization
ε = Jitter in synchronization pulse arrival
acc = 5 ppm at normal conditions
acc = 10 ppm at extreme conditions
MIST = Maximum Inter Synchronization pulse Time

Figure 13. Relation between Jitter, Accuracy and Synchronization pulse

When a slave base station receives a synchronization pulse, the time difference between the start of frames from that slave base station and the start of frames from the master base station can reach a value of $\pm e$. As a result, the maximum time difference between the start of these frames is reached earlier than when $e = 0$ (see Figure 12). The larger e , the smaller the Maximum Inter-Synchronization pulse Time ($MIST$).

The relation between the maximum difference in the reference time of a master base station and a slave base station, the accuracy of the reference timers of the slave base stations, the jitter in the arrival of the synchronization pulse and the Maximum Inter-Synchronization pulse Time is given by the next formula:

$$\tan(\tan^{-1}(acc)) = acc = \frac{\frac{d}{2} - e}{MIST} \qquad [3]$$

Thus:

$$MIST = \frac{\frac{d}{2} - e}{acc} \qquad [4]$$

We see that $MIST$ will be small for large ε. It will be impossible to synchronize RFPs if $e \geq d$.

Assumption:

Given:

- $d = 4m\sec$, due to DECT requirement
- $e = 0.434m\sec$, Clock can be corrected with half bit time accuracy (1 bit time = 0.868µsec)
- $acc = 1ppm$, due to virtual correction of the clock (see 3.5.1.2).

So:

$$MIST = \frac{\frac{d}{2} - e}{acc} = \frac{2m - 0.434m}{10^{-6}} = 1.566\sec \rightarrow 156\,frames \qquad [5]$$

This means that synchronization must take place, at least once every 156 frames.

3.5.1.1 Using a Stable clock

We see that a higher accuracy (smaller acc), leads to a larger $MIST$, which means that the system can stay synchronized (within the bounds) for a longer time before reception of a new synchronization pulse is necessary.

In current base station synchronization systems, every base station is set as a slave base station and they all have a less accurate local clock. Once every frame (10ms) the burst mode controllers[i] receive a synchronization pulse from a master clock that is attached to all slave base stations.
In order to increase $MIST$, we can add a stable clock (e.g. acc=10^{-6}) to every RFP. Such a stable clock produces a synchronization pulse every 10 ms. At least once every $MIST$ (now larger than before), a synchronization pulse from the master clock is required to synchronize the stable clock.

Figure 14 shows a possible implementation of the system mentioned before, while Figure 15 shows the effect on the RFP timing.

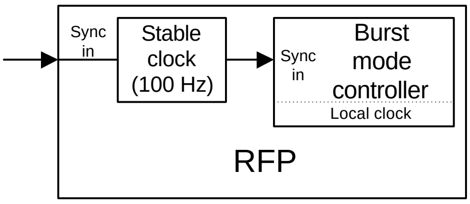


Figure 14. Using a Stable clock

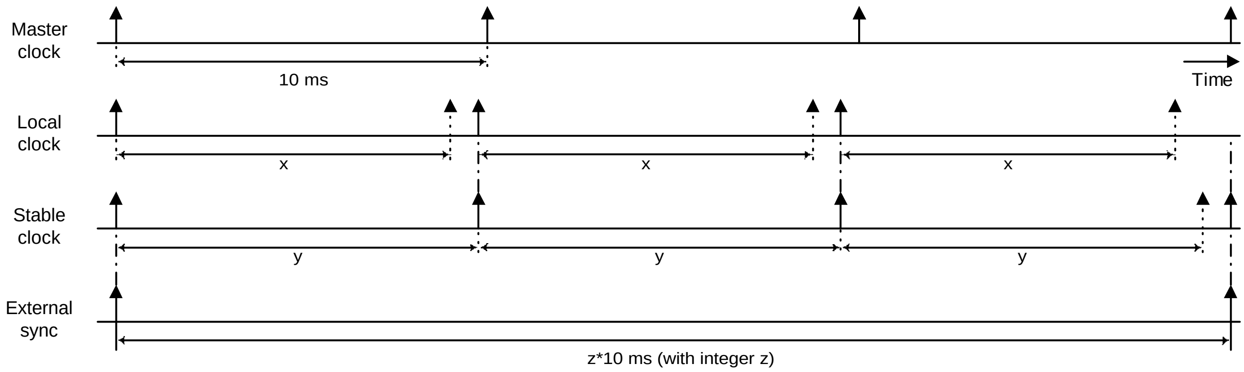


Figure 15. Effect of a stable clock (not drawn to scale)

In Figure 15 we see that the local clock (in the Burst Mode Controller) is less accurate than the stable clock (see dashed arrows). The stable clock corrects the local clock every 10 ms.
Although the stable clock is more accurate that the local clock, it drifts from the master clock, however slowly. This requires an external synchronization pulse that corrects the stable clock. The result of this construction is that we need fewer external synchronization pulses to keep the local clock synchronized with the master clock.

[i] Integrated Circuit in every base station, performing DECT (MAC)functions and timing functions

3.5.1.2 Virtual correction of the clock

When we say that a clock (e.g. crystal oscillator) has an accuracy of 10^{-6}, we mean that the frequency of the clock is found to differ relatively from the production specification (of the clock) for a value of at most 10^{-6}. Once a clock differs for a given value from the specification, this difference is almost constant. A result of this is that we can use a mechanism that determines the difference between the local clock and the master clock, in order to compensate for that difference every frame. This leads to a very reference timer accuracy.

3.6 Synchronization architecture

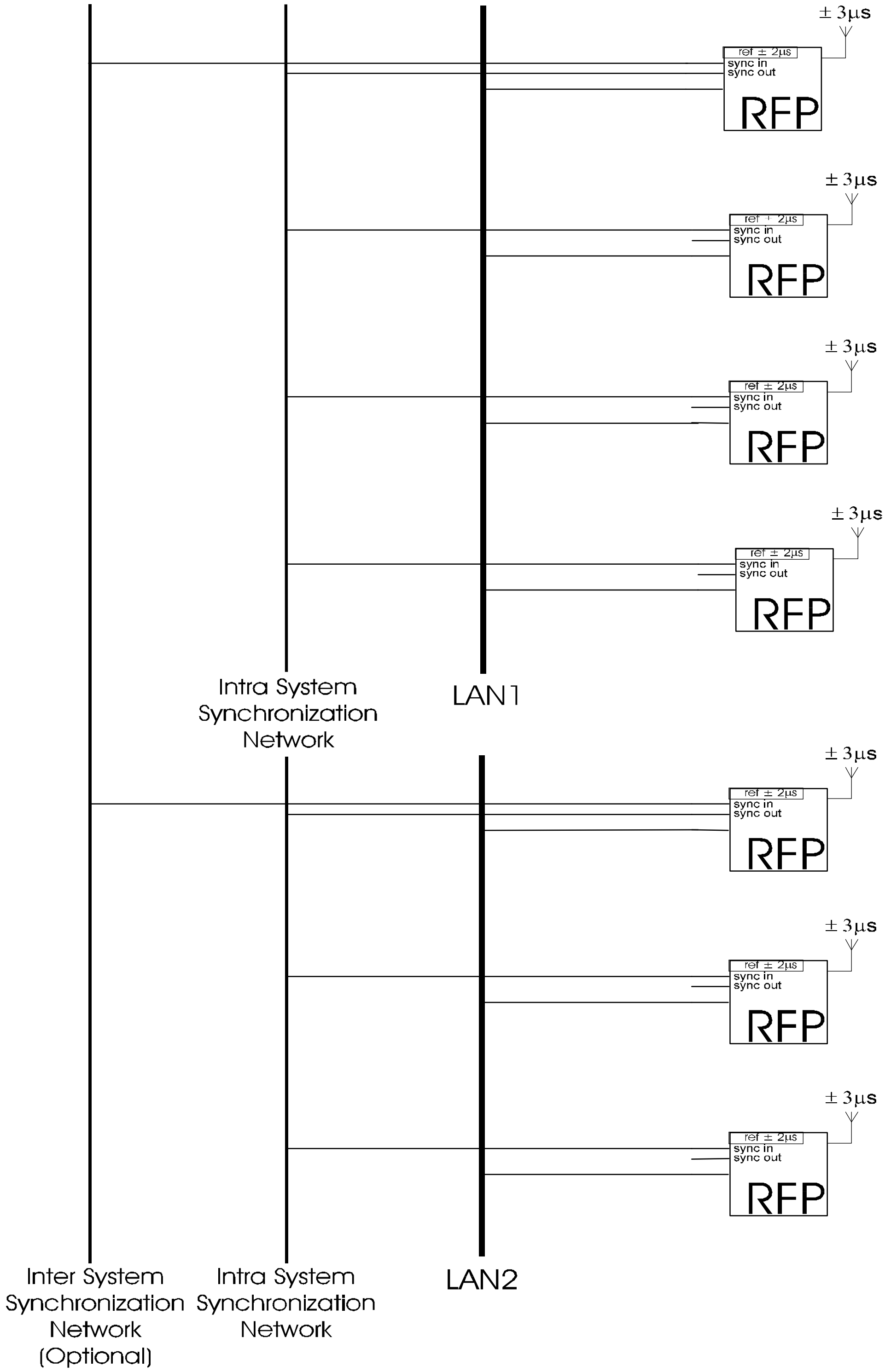


Figure 16. Synchronization network

In Figure 16 the intra-system synchronization network is receiving its synchronization signal from one master RFP. This master RFP can autonomously produce a synchronization signal when inter-system synchronization is not needed. In this case there is only intra-system synchronization.

When seamless handover between different DECT systems is needed, inter-system synchronization is needed. For this purpose an overall synchronization network is used to synchronize the intra-system synchronization networks via the master RFPs. The inter-system synchronization network can get its synchronization pulse from a global time source (e.g. GPS receiver), or from a master RFP which will then be master of a higher synchronization level.

We also see in Figure 16 that the intra-system synchronization network is a LAN. This is a possible implementation but is not the only possibility. Intra-system synchronization networks can cover more than one LAN and in the same way a LAN can cover more than one intra-system synchronization network.

4 Possible solutions to the synchronization problem

4.1 Introduction

This chapter will discuss the possible solutions to the synchronization problem. First an overview of the alternatives will be given in Section 4.2. Synchronization via the DECT Air Interface will be discussed in Section 4.3 and synchronization via Ethernet will be discussed in Section 4.4.
When discussing the possible solutions, several topics will be inspected including distribution algorithm of the synchronization pulses, delay experienced by the synchronization pulses, cost, complexity and problems to solve. These aspects will be compared in Section 4.5 where also one solution is chosen. A detailed evaluation of the chosen solution will be given in the Chapter 5.

4.2 Overview of the alternatives

Synchronization of the base station requires a common physical medium between these base stations. The common physical mediums are:

- Air Interface
- Wires

4.2.1 Air Interface

Via the Air Interface there are different possibilities of connecting base stations to each other or to an external source:

- Light
  This possibility does not offer a solution, because base stations are not necessarily in a line of sight.

- Sound (ultrasonic)
  Base stations do not have to be in a line of sight, but this possibility is also not an option because sound is not very suitable for traversing large distances (>1 km).

- RF (Radio Frequency)
  Base Stations do not have to be in a line of sight and RF signals can traverse large distances.

- GPS (Global Positioning System)
  The global positioning system consists of a number of satellites transmitting pulses at accurate (ns accuracy) times. These pulses can be used to synchronize systems. Unfortunately GPS receivers are very expensive and they have to be placed outdoor. As a result this possibility is not an option

Radio Frequency (RF) signals are the most suitable signals for transport of synchronization information. Using this type of signals we can choose to use the DECT Air Interface or to define a new frequency on which synchronization information can be transported. When choosing the latter option, a free (unused) frequency must be found for each country where the base stations will be sold. In order to avoid this difficult (may be impossible) task we choose to use the DECT Air Interface for synchronization via RF signals.

4.2.2 Wires

At this moment, coax cables or optical fibers (both an Ethernet physical medium) are the common wires of base stations within Datalab . Because it is undesirable placing additional wires, synchronization via the Ethernet physical medium is the only way of synchronizing via wires.

4.3 Synchronization via DECT Air Interface

Using the DECT Air Interface we can use the base stations to distribute the synchronization signal, in such a way that every base station listens to the surrounding base stations in order to get synchronized with a specific base station. Figure 17 shows a possible implementation of this approach.

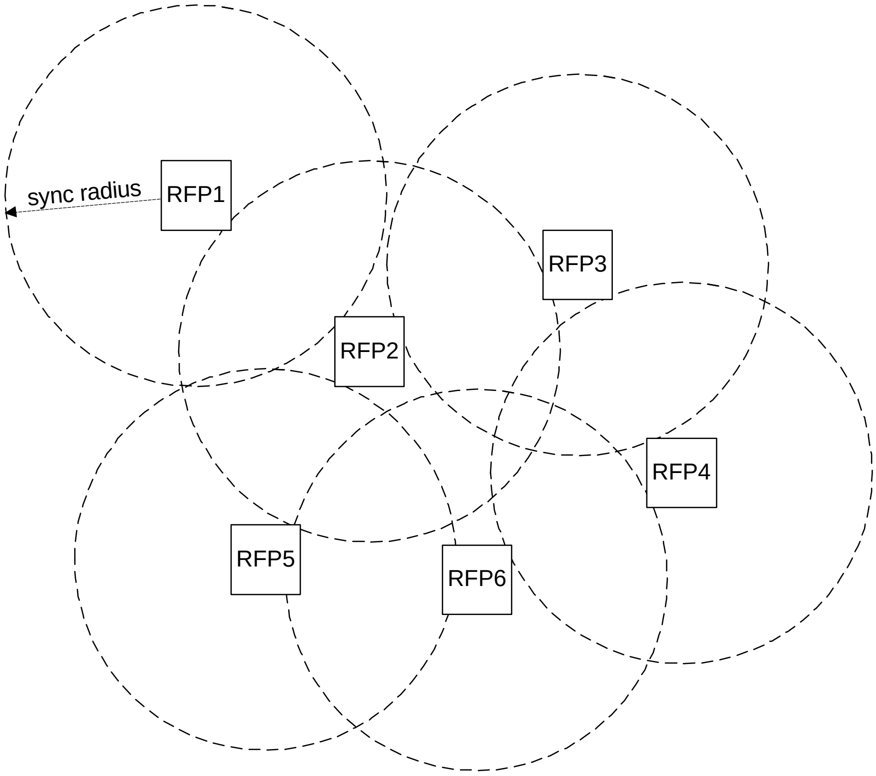


Figure 17. Air Interface Connected Base Station

We define two terms:

- Cover radius: the radius of the circle[i] (coverage circle from now on), around a particular base station, in which portable parts can communicate with that base station at the minimum Bit Error Rate.

- Sync radius: the radius of the circle (sync circle from now on), around a particular base station, in which other base stations lose synchronization with that base station with a given synchronization loss probability (see Section 3.5). This means that the size of the sync radius depends on requested probability of losing synchronization.

[i] Circular radiation patterns of the base station antennas are assumed

The sync radius is not necessary the same as the cover radius. When synchronizing via the Air Interface, a higher DECT Frame Error Ratio[i] than when potables receive speech frames is allowed, which means that the sync radius is larger than the cover radius.

The cover radius is in direct relation with the Signal to Noise Ratio (SNR) at the border of the coverage circle. Because we know the maximum allowed Bit Error Rate (for a portable part) at the border of the coverage circle, we also know the Signal to Noise Ratio at that place.
Given a maximum probability of a base station losing synchronization with another base station we can determine the necessary Frame Error Ratio and thus the necessary Bit Error Rate. With this Bit Error Rate, we can calculate the required Signal to Noise Ratio required for synchronization at the border of the sync circle.
Comparing the two Signal to Noise Ratios will give an indication of the ratio of cover radius an sync radius.

In Section 4.3.4 we will compare the sync radius and the cover radius in order to give an indication of the price increase when adding synchronization via the DECT Air Interface to a DECT system.

4.3.1 Distribution algorithm

In this section we will find out how the synchronization pulses can be distributed over the base stations. We do this in order to get a first idea about the complexity of the solution.
In Figure 17, we see that each RFP is within the sync range of at least one other RFP. By letting one RFP act as master RFP (RFP1 is an example), in which that RFP uses its own clock as a master clock, or extract its clock from another timing source, synchronization can be performed by telling each RFP with which RFP it has to synchronize.

A possible synchronization sequence for the system of Figure 17 is shown by Figure 18.

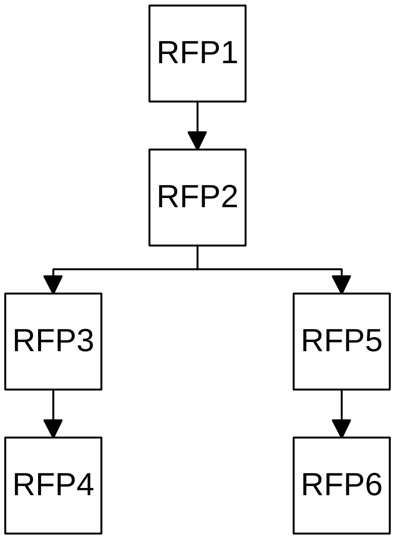


Figure 18. A possible synchronization sequence of RFPs

The synchronization dependencies of the solution given by Figure 18 are shown in Table 3. All the RFP are listed together with the adjacent RFPs. For each RFP the shortest way to the master RFP is chosen, unless there is a deadlock (RFPA synchronizes with RFPB and vice versa).

---

[i] The frame error ratio is the fraction of frames with corrupted A-field.

synchronizing RFP	Available RFPs to synchronize with			RFP to synchronize with	
RFP number	**RFP number**	**#hops from Master RFP**	**Remark**	**RFP number**	**#hops from Master RFP**
RFP2	RFP1 RFP3 RFP5	0 2 2	Master RFP RFP3 is synchronizing with RFP2 RFP5 is synchronizing with RFP2	RFP1	1
RFP3	RFP4 RFP2	3 1	RFP4 is synchronizing with RFP3	RFP2	2
RFP4	RFP3 RFP6	2 3	Shortest Propagation time	RFP3	3
RFP5	RFP2 RFP6	1 3	Shortest Propagation time RFP6 is synchronizing with RFP5	RFP2	2
RFP6	RFP4 RFP5	3 2	Shortest Propagation time	RFP5	3

Table 3. Synchronization dependencies

In order to get the synchronization sequence of Figure 18, an algorithm performing this function must be implemented. The algorithm must also:

- Determine the synchronization sequence that leads to the minimum average number of hops to the master base station (e.g. smaller average number of hops if RFP2 is the master base station).
- Determine the new synchronization sequence when adding or removing a base station to the system.

We expect that implementing such an algorithm will increase the complexity of synchronization via the DECT Air Interface.

4.3.2 Delay

The relevance of the delay aspect can best be explained on the basis of Figure 19 where RFP2 is synchronizing with RFP1.

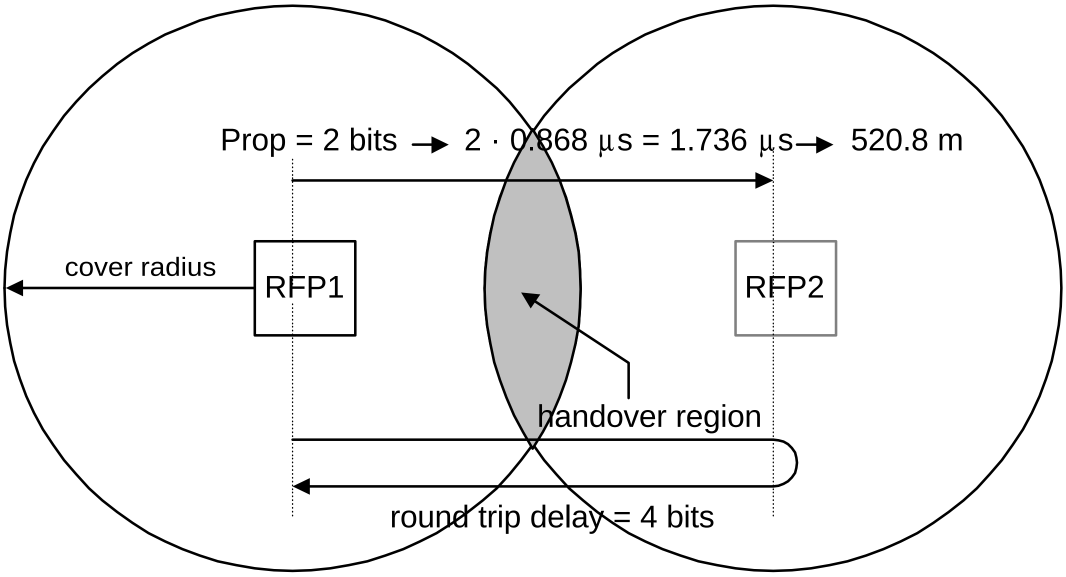


Figure 19. Delay in the Air Interface

If, for instance, base stations are 520 meters away from each other, we deal with a propagation delay of about 1.7 µs, which means a shift of two bits. Frames transmitted by RFP1 will arrive 2 bit times later at RFP2. RFP2 can not notice this delay and will therefore synchronize on the delayed frame.

If a handover would take place near to RFP2 then there would not be any problem (from the portable part's point of view) because in that region there are delayed frames from RFP1 and frames from RFP2 synchronous to these delayed frames.
Unfortunately handovers mostly take place at the handover region. In the handover region frames from RFP1 suffer a delay of 1 bit time and frames from RFP2 are shifted 3 bit times (2 bit times delay from RFP1 to RFP2 and 1 bit time delay from RFP2 to the handover region) from the frames transmitted by RFP1. This means that the time difference between frames from RFP1 and frames from RFP2 at the handover region is 2 bit times. When the distance between RFP1 and RFP2 exceeds 2 km, the time difference between the frames from RFP1 and RFP2 will be larger than 8 bit times making handovers impossible. A handover will also be impossible if we have a synchronization sequence like given in Figure 20. The reason for this is that in the handover region between RFP1 and RFP10, the time difference between the start of frames from RFP1 and RFP10 is 9 bit times ($7.8 m\sec$) which makes a handover impossible (see Section 3.2.3).

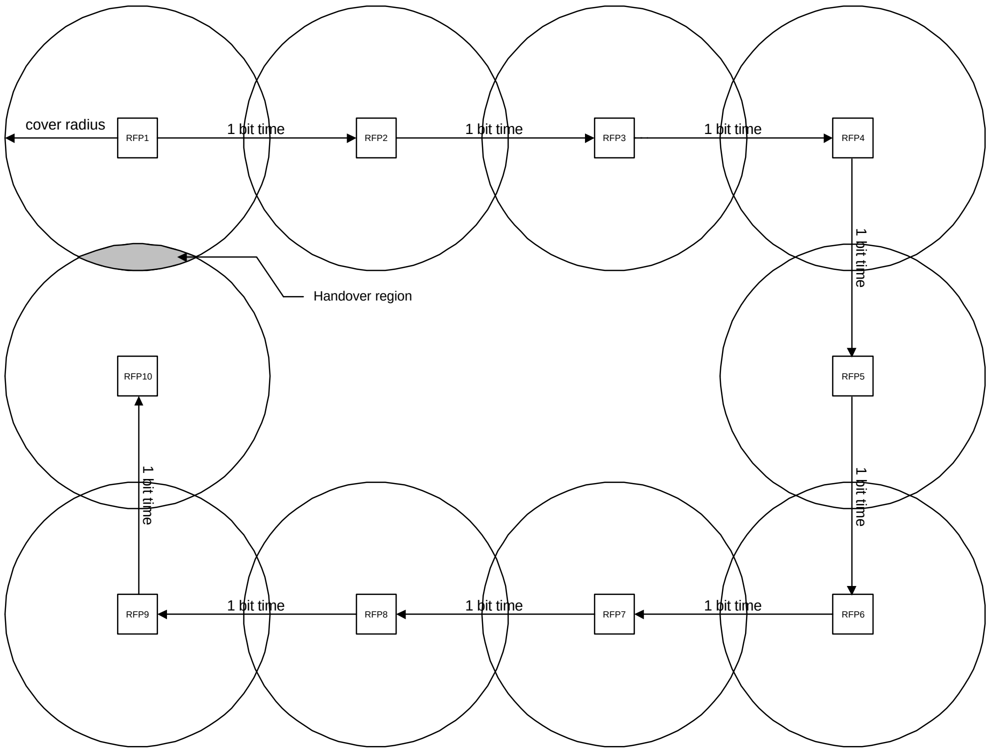


Figure 20. Long synchronization sequence

The synchronization mechanism to be developed must compensate for the delay between two base stations in order to avoid this problem.

In the case of Figure 20, the time difference between frames can be made smaller by letting the synchronization with RFP1 take place in two directions, which means that both RFP2 and RFP10 are synchronizing with RFP1. The result is a maximum time difference of 4 bit times between frames from RFP7 and RFP6 in the handover region between RFP7 and RFP6. The difference between frames from RFP5 and RFP6 in the handover region between RFP5 and RFP6 is also 4 bit times. This modification leads to a time difference between frames (in handover regions) of at most half the maximum time difference before the modification.

A generic solution is to let a base station listen to frames sent by the base station that is synchronizing with it, and extract the propagation time and hence the correction factor (number of bits) by comparing the received frames with its own frames. The listening base station then has to send a timing modification command to the synchronizing base station in order to let the synchronizing base station modify its timing. In the example of Figure 19, RFP1 will listen to frames sent by RFP2 and will notice a time difference of 4 bit times with its own frames. RFP1 will then ask RFP2 to start its frames 2 bits earlier. In absolute sense, this means that the base stations start sending their frames at the same time, with a possible initial difference of
$\pm$ 0.5 bit which means a difference of $\pm$ 0.434 μs.

4.3.3 Jitter

When base station synchronization via the DECT Air Interface is used, jitter is introduced when the start of the next frame is corrected. At that moment a shift in time, which can be seen as jitter, will take place.

The first way to face this problem is to correct more often, decreasing the size of the time shifts. Another, more advanced manner, is to measure the drift of the local clock after synchronization takes place. This is done by comparing the bit-numbers (which can be extracted using the existing patterns in a frame) of the received frame and the internal bit-number of the receiving RFP. The measured drift (in bits) can then be used to regulate the internal clock (shift the internal bit-number) of the synchronizing base station.

4.3.4 Cost

The cost of a synchronization solution based on the use of the DECT Air Interface is hard to predict at this level. The increase of the price of a DECT air synchronized system in comparison with a non-synchronized system is introduced by bringing the base stations closer to each other, which means a higher number of base stations to cover the same area.
When a base station wants to listen to another base station in order to synchronize with that base station, the listening base station must be in receive mode during at least one time slot in the Downlink period (see Figure 4). The synchronizing base station can not use this time slot for communication with a portable part, which means a decrease of the capacity and hence a price increase.

In order to determine the increase in the number of base station necessary for synchronization, we will compare two distances. The minimum distance between two base stations at which a handover between them is still possible at one hand, and on the other hand the minimum distance between two base station at which synchronization between them is still possible with the required probability of losing synchronization. If the last mentioned distance is at least equal to, or larger than the first mentioned distance, base stations do not have to be placed closer to each other, which means no price increase due to base station repositioning.

Assumptions:

1) A base station that wants to synchronize with another base station always has a free time slot in the Downlink period available for synchronization. A base station can be programmed to always have a free time slot.
2) When a synchronizing base station listens in a free time slot in the Downlink period, the base station it wants to synchronize with is transmitting.
   In reality the base station with which a synchronizing base station wants to synchronize is not necessary transmitting during the time slot in which the synchronizing base station listens, which means that this assumption is optimistic. If the following sections show that the distance between base station does not have to be much smaller, then calculations must be repeated without this assumption, otherwise we do not repeat the calculations.
3) The maximum distance between two base stations at which handovers between those base stations are still possible is equal to twice the cover radius. A that distance there still is a region (very small) in which a portable part can communicate with both base stations. In reality this maximum distance is smaller than twice the cover radius in order to increase the area of the handover region.
4) When the sync radius is larger than twice the cover radius, base station do not have to be placed closer to each other because the distance between them is smaller than twice the cover radius (see Figure 21 and assumption 3) and subsequently smaller than the sync radius.
5) If we want a base station to synchronize with another base station, the synchronizing base station must receive a correct S-field at least once every time period equal to $MIST$ and a correct A-field at least[i] once every 10 seconds (1000 DECT frames). In our calculations we will first determine the maximum sync radius for both situations independently. If the calculations show that the distance between base station does not have to be much smaller, then calculations must be repeated considering a simultaneous synchronization loss probability due to an incorrect S-field reception or incorrect A-field reception. Otherwise we do not repeat the calculations.
6) In Section 4.3.4.1 and Section 4.3.4.2 calculation are made assuming that base stations and portable parts have the same receive sensitivity, transmission power and antenna gain. In Section 4.3.4.3 we will see what the effect of different receive sensitivity, transmission power and antenna gain is.

[i] In order to ensure that a synchronizing base station is synchronizing with the correct base station, identification (in A-field) of that base station must be received at least once every 10 seconds. This is an implementation choice analogue to the requirement in [ETS 300 175-3] for a synchronizing portable part.

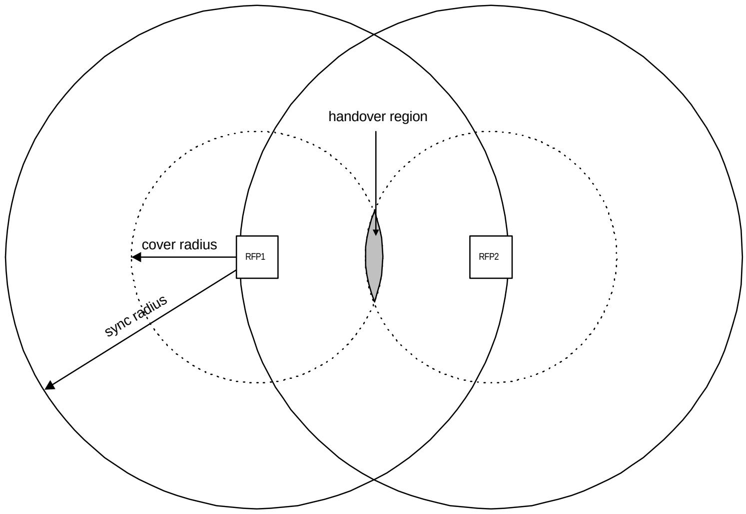


Figure 21. Base stations within the sync radius

In the following sections we will compare the cover radius and the sync radius in both a non-fading environment and a fading environment.

4.3.4.1 Distance comparison in a non-fading environment

In this section, we will compare the maximum tolerated distance between a base station and a portable part on one hand, and the maximum tolerated distance between two base stations on the other hand. Both distances will be calculated in a non-fading environment.
The distance between a portable part and a base station will depend on the speech requirement, whereas the distance between the base stations will depend on the synchronization requirement. The ratio of the two distances to be compared will give an idea about the adjustment to be made in the position of base stations in order to let them synchronize with each other.

As mentioned in Section 4.3.4, if the ratio is found to be higher than 2, then there is no need for any adjustment. The reason for this conclusion is that the distance between two base stations has an upper-bound equal to twice the maximum distance between a base station and a portable part if handover is provided between those base stations. Otherwise the portable part can not find a region where it can be in contact with both base station, and so it can not make a handover from one base station to the other, which means that those base stations do not have to be synchronized (in relation to handovers).

In this section we consider a non-fading environment, which means that every bit error is independent of another.

For distance comparison in relation to synchronization losses due to incorrect S-field reception we follow the next steps:

1. Given the minimum Bit Error Rate (BER) for speech calculate the minimum Signal to Noise Ratio (SNR_{speech}) for speech.
2. Given the accuracy of the reference timer of a base station (acc), the jitter e and the maximum difference in the reference timers of two synchronized base stations, we calculate the Maximum Inter-Synchronization pulse Time ($MIST$). If no synchronization pulse arrives within a time period equal to $MIST$ base station lose synchronization with each other.
3. Given the Synchronization Loss Ratio (SLR) and $MIST$,the Synchronization Error Rate (SER) will be determined. SER is the fraction of frames, which can not be used for synchronization due to corrupted
S-field.
4. Given SER, the BER necessary for proper synchronization can be calculated.
5. Given the BER necessary for proper synchronization a Signal to Noise Ratio ($SNR_{S-field}$) is calculated.
6. Assuming that the base stations and portable parts have the same sensitivity, transmission power and antenna gain, the difference between SNR_{speech} and $SNR_{S-field}$ is caused by the propagation loss. As a result we can determine the (relative) difference in the distances at which these Signal to Noise Ratios can be found.

For distance comparison in relation to synchronization losses due to incorrect A-field reception we follow the next steps:

1. Given the minimum Bit Error Rate (BER) for speech calculate the minimum Signal to Noise Ratio (SNR_{speech}) for speech (already calculated for S-field comparison).
2. Given the Synchronization Loss Ratio (SLR) and maximum period before reception of a correct A-field (10 seconds, see Section 4.3.4), A-field Error Ratio (AER) will be determined. AER is the fraction of frames, which can not be used for synchronization due to corrupted A-field.
3. Given AER, the BER necessary for proper synchronization can be calculated.
4. Given the BER necessary for proper synchronization a Signal to Noise Ratio ($SNR_{A-field}$) is calculated.
7. Assuming that the base stations and portable parts have the same sensitivity, transmission power and antenna gain, the difference between SNR_{speech} and $SNR_{A-field}$ is caused by the propagation loss. As a result we can determine the (relative) difference in the distances at which these Signal to Noise Ratios can be found.

SNR for speech:

According to [ETS 300 175-2] the minimum requested BER for speech is:

$$BER = 10^{-3} \qquad [6]$$

According to [ERICSSON1] a typical radio in a hypothetical DECT system, the instantaneous BER can be approximately described as a function of the signals to noise ratio (SNR in dB) as follows:

$$BER(SNR) = \frac{1 - erf\left(\sqrt{0.34 \cdot 10^{\frac{SNR}{10}}}\right)}{2} \qquad [7]$$

Substituting Equation 6 in Equation 7 gives the requested SNR for speech:

$$SNR_{speech} = 11.475 dB \qquad [8]$$

Comparison between minimum distances for speech and S-field reception:

In Section 3.5 we have mentioned that a synchronization loss probability of at most once a month is required. The fact that every seconds contains 100 DECT frames gives:

$$SLR = \frac{1}{100 \cdot 3600 \cdot 24 \cdot 31} = 3.734 \cdot 10^{-9} \text{ (implementation choice)} \qquad [9]$$

When we implement the clock correction mentioned in Section 3.5.1.2, the accuracy of the base station clock will reach:

$$acc = 10^{-6} \qquad \text{(with virtual correction of the clock)} \qquad [10]$$

When correcting the base station clock, correction are made in bits introducing an possible difference e between the reference clock and the corrected clock equal to a $\frac{1}{2}$ bit time (at 1152000 bps). This means:

$$e = \frac{1}{2 \cdot 1152000} = 4.34 \cdot 10^{-7} \sec \qquad \text{(see Figure 13)} \qquad [11]$$

The maximum difference between the start of frames from two synchronized base stations is:

$$d = 4 \cdot 10^{-6} \sec \qquad \text{(see Figure 13)} \qquad [12]$$

The maximum time between two successive receptions of a correct synchronization pulse ($MIST$) is:

$$MIST = \frac{\frac{d}{2} - e}{acc} = 1.566 \sec \text{ (156 frames)} \qquad [13]$$

Synchronization is lost when 156 subsequent DECT frames do not contain a correct S-field.

Thus the Synchronization Loss Ratio (SLR) as a function of Synchronization Error Ratio (SER) is:

$$SLR(SER) = SER^{156} \qquad [14]$$

Substituting Equation 9 in Equation 14 gives the Synchronization Error Ratio (SER), which is the fraction of frames with corrupted S-field:

$$SER = 0.883029 \qquad [15]$$

The S-field is 16 bits long. Base stations are programmed to allow a single error in the S-field. In other words more than one bit in the S-field is received incorrectly, the S-field is said to be corrupted. The Synchronization Error Ratio as a function of the Bit Error Rate is:

$$SER(BER) = 1-(1-BER)^{16} - 16 \cdot BER \cdot (1-BER)^{15} \qquad [16]$$

$BER_{S-field}$, which is the Bit Error Rate required to achieve the Synchronization Error Ratio of Equation 15, can be found by substituting Equation 15 in Equation 16. The result of the substitution is:

$$BER_{S-field} = 0.212148 \qquad [17]$$

$SNR_{S-field}$, which is the Signal to Noise Ratio required to achieve the Bit Error Rate of Equation 17 and thus the Synchronization Error Ratio of Equation 15, can be found by substituting Equation 17 in Equation 7. The result of the substitution is:

$$SNR_{S-field} = -0.274dB \qquad [18]$$

According to [KEMO], loss as a function of distance in buildings is:

$$L(d) = 38 + 35 \cdot \log(d) \qquad [19]$$

The assumption made in Section 4.3.4, that the difference in SNR is caused by the difference in loss and hence by the difference in distance to the transmitting base station gives:

$$L(d_{S-field}) - L(d_{speech}) = SNR_{speech} - SNR_{S-field} = 11.475 - (-0.274) = 11.749dB \qquad [20]$$

SNR_{speech} and $SNR_{S-field}$ can be found in Equation 8 and Equation 18.

Substitution of Equations 19 in Equation 20 gives:

$$35 \cdot \log(d_{S-field}) - 35 \cdot \log(d_{speech}) = 35 \cdot \log(\frac{d_{S-field}}{d_{speech}}) = 11.749dB \qquad [21]$$

The ratio of the minimum distance for speech and the minimum distance for S-field reception is:

$$\frac{d_{S-field}}{d_{speech}} = 10^{\frac{11.749}{35}} = 2.166 \qquad [22]$$

Comparison between minimum distances for speech and A-field reception:

As mentioned in Section 4.3.4, synchronizing base stations have to receive a correct A-field at least once every 10 seconds. Measurements have shown that 90% of the frames, sent by a base station, contain an A-field with the desired identification information.

Synchronization is lost when all the frames containing the desired information are received with a corrupted A-field. The Synchronization Loss Ratio as a function of the A-field Error Ratio (AER is the fraction of frames with corrupted A-field) is:

$$SLR(AER) = AER^{(100 \cdot 0.9 \cdot 10)} \qquad [23]$$

Substituting the Synchronization Loss Ratio of Equation 9 in Equation 23 gives an A-field Error Ratio of:

$$AER = 0.978669 \qquad [24]$$

The A-field is 64 bits long. The A-field uses a CRC error control mechanism in which errors can be detected but not corrected. As a result, when one or more bits in the A-field are received incorrectly, the A-field is said to be corrupted. The A-field Error Ratio as a function of the Bit Error Rate is:

$$AER(BER) = 1 - (1 - BER)^{64} \qquad [25]$$

$BER_{A-field}$, which is the Bit Error Rate required to achieve the A-field Error Ratio of Equation 15, can be found by substituting Equation 24 in Equation 25. The result of the substitution is:

$$BER_{A-field} = 0.058347 \qquad [26]$$

$SNR_{A-field}$, which is the Signal to Noise Ratio required to achieve the Bit Error Rate of Equation 26 and thus the A-field Error Ratio of Equation 24, can be found by substituting Equation 26 in Equation 7. The result of the substitution is:

$$SNR_{A-field} = 5.586dB \qquad [27]$$

The assumption made in Section 4.3.4, that the difference in SNR is caused by the difference in loss and hence by the difference in distance to the transmitting base station gives:

$$L(d_{A-field}) - L(d_{speech}) = SNR_{speech} - SNR_{A-field} = 11.475 - 5.586 = 5.889dB \qquad [28]$$

SNR_{speech} and $SNR_{A-field}$ can be found in Equation 8 and Equation 27.

Substitution of Equations 19 in Equation 28 gives:

$$35 \cdot \log(d_{A-field}) - 35 \cdot \log(d_{speech}) = 35 \cdot \log(\frac{d_{A-field}}{d_{speech}}) = 5.889dB$$

The ratio of the minimum distance for speech and the minimum distance for S-field reception is:

$$\frac{d_{A-field}}{d_{speech}} = 10^{\frac{5.889}{35}} = 1.47 \qquad [29]$$

<u>Conclusion:</u>

We see that the distance ratio for synchronization is higher than 2, which means that we don't have to adjust the position of the base stations for correct reception of the S-field in this non-fading environment.
To receive the A-field correctly the distance between two base stations has to be decreased by ±26.5% (2→1.47). In other words the minimum distance between base station necessary for correct reception of an A-field at the required frequency (at least once every 10 seconds), is the minimum distance between base station required for speech transmission (at least $BER = 10^{-3}$) multiplied by 0.735. This means that the number of base stations necessary to cover the same are is multiplied by $(1/0.735)^2 = 1.85$, which means an increase in the number of base stations of 85%.

4.3.4.2 Distance comparison in a Rayleigh fading environment

In Section 4.3.4.1 the maximum distance between a portable part and a base station and the maximum distance between two base stations are compared in a non-fading environment. This means that bit errors are assumed to be independent of each other. In reality bit errors are not independent of each other. When a bit is corrupted, there is a high probability that the following bits are also corrupted.

In order to take this effect in account, we will compare the maximum distance between a portable part and a base station and the maximum distance between two base stations in a Rayleigh fading environment.
With this type of fading, burst of errors lasting more than one frame (dependent on the speed of the receiver) can be simulated.

We will distinguish between two different situations. To calculate the maximum distance between a portable part and a base station, we consider a moving portable part. To calculate the maximum distance between two base stations we assume the base stations to be stationary.

To model a fading environment we imagine several (6 gives a good approximation) sources at infinity all with random phases and random angles in reference to the receiver (PP or base station). Once arrived at the moving receiver, the signals transmitted by these sources form the modeled Rayleigh fading.

To distinguish between the two situations mentioned we assume that the portable part is moving with a velocity of 1 m/s (slow walking speed). On the other hand we assume the fixed part to move with a velocity of 0.1 m/s in order to model slow fades caused by the slow changes of the environment (e.g. moving objects).

Figure 22 and Figure 23 give two typical Rayleigh fading graphs. To get these graphs we wrote a simulation program in Mathcad (see Appendix D).

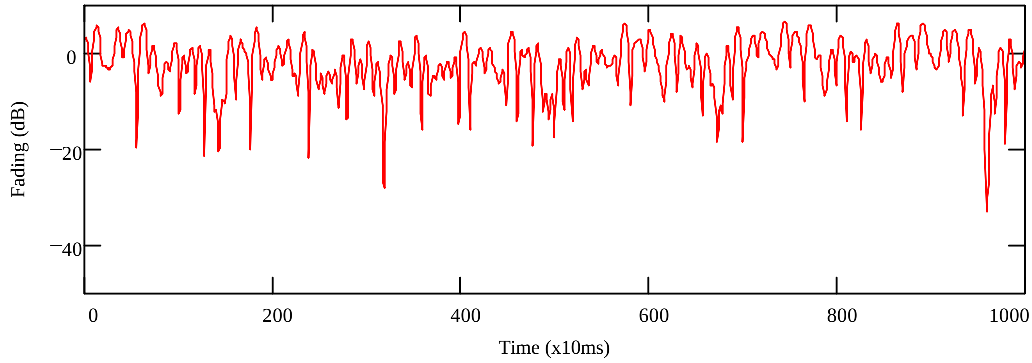


Figure 22. Rayleigh fading for a portable part

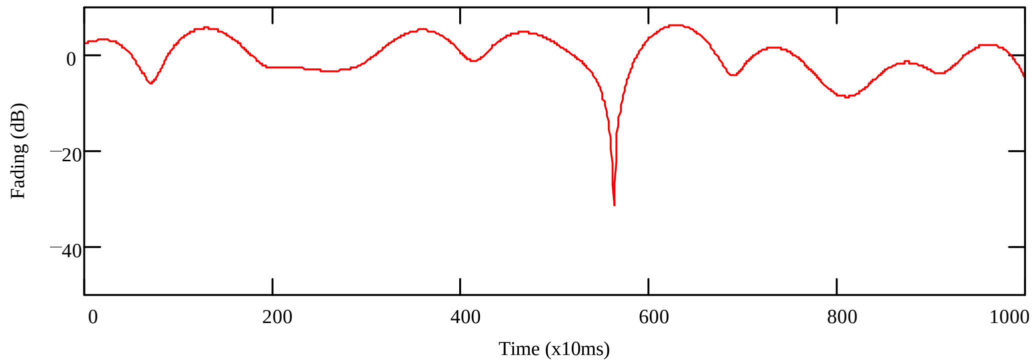


Figure 23. Rayleigh fading for a fixed part

As we can see from Figure 22 and Figure 23, the fades for a fixed part continues for a longer time than the fades for the portable part, which means that there is a high probability that several successive frames will contain errors.

In a Raleigh fading environment the Signal to Noise Ratio at the receiver at a given moment is equal to the sum of the Signal to Noise Ratio that would be experienced in a non-fading environment, and the fading at that moment. The formula for the Bit Error Rate as a function of the Signal to Noise Ratio can be found by rewriting Equation 7:

$$BER(SNR) = \frac{1 - erf\left(\sqrt{0.34 \cdot 10^{\frac{SNR + fading_{Rayleigh}}{10}}}\right)}{2} \qquad [30]$$

This Bit Error Rate can be converted to a Synchronization Loss Ratio as we have seen in a previous section. Comparing the Signal to Noise Ratio needed to reach the Synchronization Loss Ratio of Equation 6 and the Signal to Noise Ratio needed to reach $BER = 10^{-3}$ both in different Rayleigh fading environments, we can extract a distance ratio. In order to do this comparison we wrote three programs in C++ (see Appendix A, B and C for listing and comments).

Figure 24 shows the simulation results for speech. In Figure 24 the 95% confidence intervals are also plotted.
We have eliminated the worst 1% of the received frames, according to [TBR 006] which says that 1% of the received frames can be suppressed when providing speech transmission. As we can see, a Signal to Noise Ratio of ≈24dB is necessary to reach a Bit Error Rate of 10^{-3}.

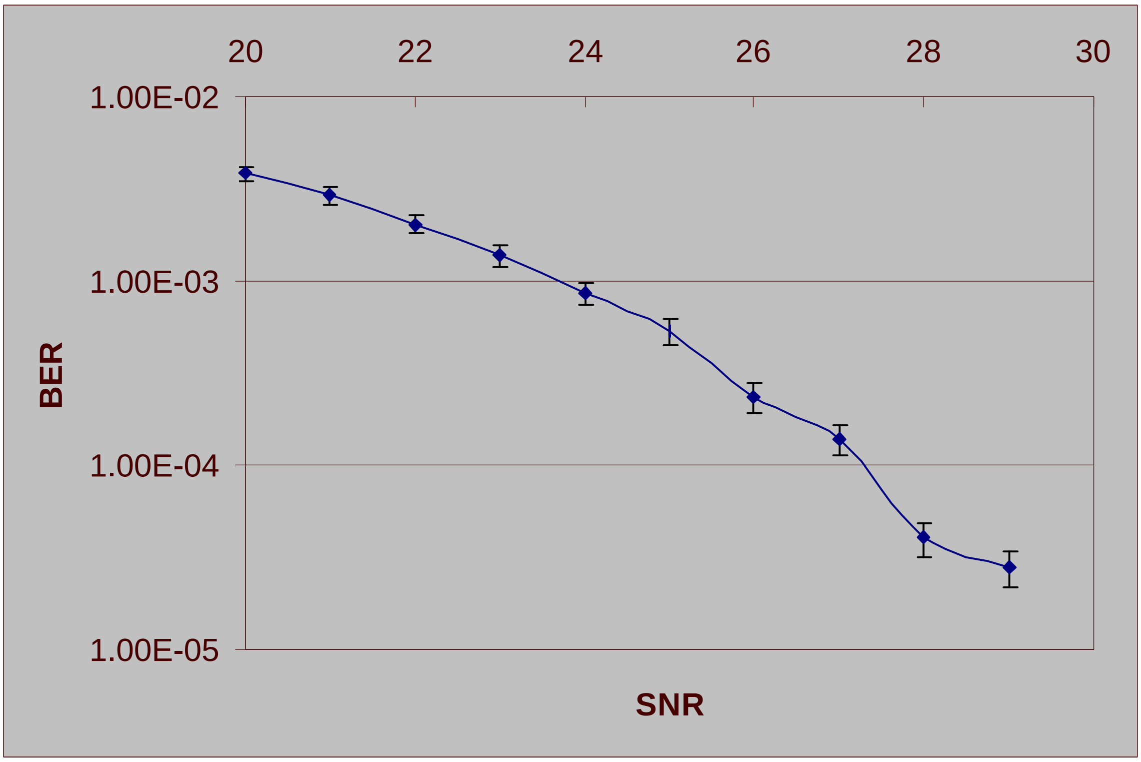


Figure 24. BER as a function of SNR for speech in a fading environment (no diversity[i])

Thus the minimum Signal to Noise Ratio necessary to reach the Bit Error Rate of Equation 6 in a fading environment is:

$$SNR_{speech} = 24dB \qquad [31]$$

[i] Diversity is a communication receiver technique that provides wireless link improvement at relatively low cost. It makes use of the fact that if one radio path undergoes a deep fade, another independent (or at least highly uncorrelated) path may have a strong signal. By having more than one path (more than one antenna) to select from, both the instantaneous and average Signal to Noise Ratios at the receiver may be improved, often by as much as 20dB to 30dB [Rapp].

Comparison between minimum distances for speech and S-field reception:

We made a simulation with $8 \cdot 10^6$ independent sets of 156 frames for every Signal to Noise Ratio (SNR) in a fading environment. The fading environment is modeled with Rayleigh fading with 12 sources at infinity and base station speeds of 0.1 m/s.
For every set of 156 frames the Synchronization Loss Ratio (SLR) due to incorrect S-field reception is determined, which leads to $8 \cdot 10^6$ Synchronization Loss Ratio (due to corrupted S-fields) values for every Signal to Noise Ratio. The large number of simulation is chosen to get a small confidence interval (confidence level of 95%).

Figure 25 shows the Synchronization Loss Ratio due to corrupted S-field as function of the SNR.

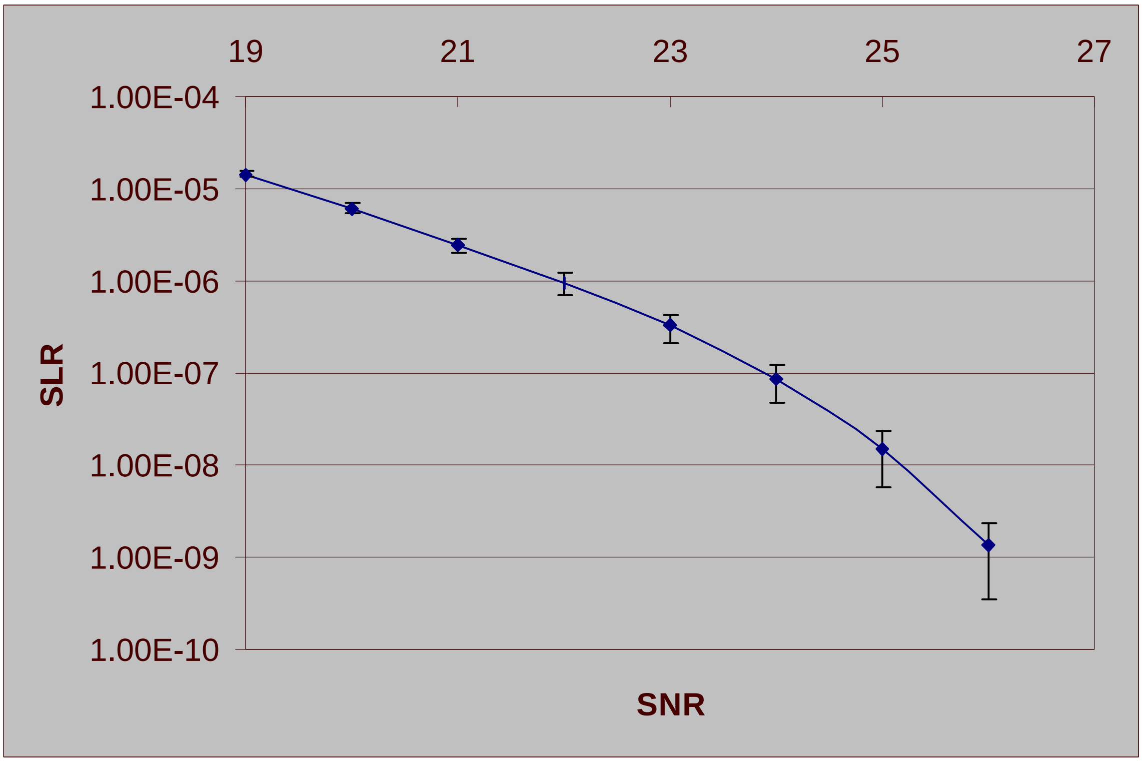


Figure 25. *SLR* as a function of SNR

As we can see, to reach the Synchronization Loss Ratio of Equation 9 ($SLR = 3.7 \cdot 10^{-9}$) we need:

$$SNR_{S-field} \approx 24.5dB \qquad [32]$$

The assumption made in Section 4.3.4, that the difference in SNR is caused by the difference in loss and hence by the difference in distance to the transmitting base station gives:

$$L(d_{S-field}) - L(d_{speech}) = SNR_{speech} - SNR_{S-field} = 24 - 24.5 = -0.5dB \qquad [33]$$

SNR_{speech} and $SNR_{S-field}$ can be found in Equation 31 and Equation 32.

Substitution of Equation 19 in Equation 33 gives:

$$35 \cdot \log(d_{S-field}) - 35 \cdot \log(d_{speech}) = 35 \cdot \log(\frac{d_{S-field}}{d_{speech}}) = -0.5dB \quad [34]$$

The ratio of the minimum distance for speech and the minimum distance for S-field reception is:

$$\frac{d_{S-field}}{d_{speech}} = 10^{\frac{-0.5}{35}} = 0.968 \quad [35]$$

This means that the tolerable distance between base stations is smaller for correct S-field reception between base station than for correct speech transmission between portables and base stations. This implies that the distance between base stations must be halved, which means that the number of base stations will increase by a factor 4 to cover the same area.

Comparison between minimum distances for speech and A-field reception:

We made a simulation with $2 \cdot 10^6$ sets of 900 frames (see Equation 23) for two Signal to Noise Ratios (SNR) in a fading environment. The fading environment is modeled with Rayleigh fading with 12 sources at infinity and base station speeds of 0.1 m/s.
For every set of 900 frames the Synchronization Loss Ratio (SLR) due to incorrect A-field reception is determined. Table 4 shows the results.

SNR	$\overline{SLR}$	Confidence Interval (confidence level of 95%)
22 dB	$3.12 \cdot 10^{-6}$	$2.15 \cdot 10^{-6}$
23 dB	$1.64 \cdot 10^{-6}$	$1.50 \cdot 10^{-6}$

Table 4. Simulation results of the *SLR* due to incorrect A-field reception

We see that the confidence intervals (confidence level of 95%) are relatively high in comparison to the mean SLR. To halve the confidence interval we have to make a simulation with $8 \cdot 10^6$ sets of 900 frames (factor $2^2 = 4$). Unfortunately resources to make such long simulation are not available, which means that we content our selves with the results of Table 4. The values presented, show that the SLR due to incorrect A-field reception is higher than SLR due to incorrect S-field reception for a given SNR. However, we expect that the SLR due to incorrect A-field reception will decrease rapidly by increasing the SNR because at higher Signal to Noise Ratios, fades will not last (very small probability) longer than 900 frames.

We say that synchronization is lost when during 156 successive frames no correct S-field is received or when during 900 frames no correct A-field is received or when both events happen. In the previous section we have mentioned that due to incorrect S-field reception, we need more (4x) base station on the same area to reach the desired SLR. We conjecture that incorrect A-field receptions further increase the required number of base stations. We will not prove this.

4.3.4.3 Link Budget

In Section 4.3.4.1 and Section 4.3.4.2 we have used the assumption that the difference in Signal to Noise Ratios is caused by the difference in loss and hence by the difference in distance to the transmitting base station. This is only true when a portable part and a base station have the same transmission power, receive sensitivity and antenna gain.

Portables and base stations within the GAP standard have transmission powers ranging from 19dBm to 24dBm and their reception sensitivity ranges from –86dBm to –90dBm [ETS 300 175-2].
Base stations have to cover portables with transmission powers of 19dBm (lowest power) and reception sensitivities of –86dBm (lowest sensitivity). For base stations we consider the maximum gain in account.

After the following two examples we will see that due to base stations and portable parts having different transmission powers, receiver sensitivities and antenna gains, base station positioned at a given place will experience a higher Signal to Noise Ratio than a portable part will experience at the same place.
Both example 1 and example 2 will be based on Figure 26.

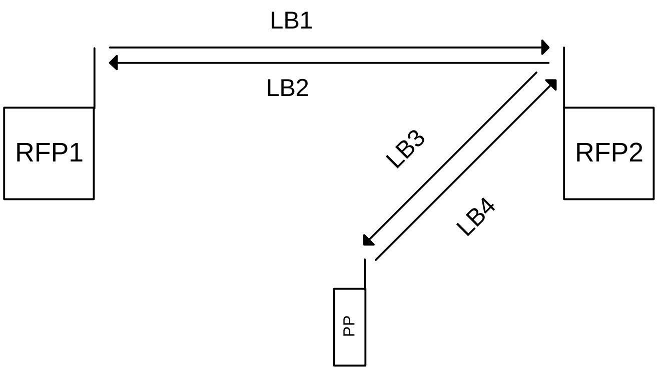


Figure 26. Link budget of RFP-RFP and Portable-RFP connection

During both example 1 and example 2 the following parameters will be used:

Because base stations use a half-wave dipole, the antenna gain of a base station is (see [Cheng]):

$$G_{RFP} \approx 2dBi \qquad [36]$$

Antenna gain of a portable part:

$$G_{PP} \approx 0dBi \qquad [37]$$

We consider portable parts transmitting with the minimum required transmission power:

$$P_{PP} = 19dBm \qquad [38]$$

We also consider portable parts with the minimum allowed sensitivity:

$$S_{PP} = -86dBm \qquad [39]$$

We consider base stations with the maximum transmission power:

$$P_{RFP} = 24dBm \qquad [40]$$

Example 1:

Consider a base station with the maximum sensitivity:

$$S_{RFP} = -90dBm \quad [41]$$

The link budget for the link from RFP1 to the portable part is build up of the transmission power of the base station, the receiver sensitivity of the portable part and the antenna gains of both the base station and the portable part:

$$LB3 = P_{RFP} - S_{PP} + G_{RFP} + G_{PP} = 24dBm + 86dBm + 2dBi + 0dBi = 112dB \quad [42]$$

The link budget for the link from the portable part to RFP1 is build up of the transmission power of the portable part, the receiver sensitivity of the base station antenna gains of both the base station and the portable part:

$$LB4 = P_{PP} - S_{RFP} + G_{RFP} + G_{PP} = 19dBm + 90dBm + 2dBi + 0dBi = 111dB \quad [43]$$

The link budget for the link from a base station to another is build up of the transmission power of a base station, the receiver sensitivity of a base station and the antenna gain of a base station:

$$LB1 = LB2 = P_{RFP} - S_{RFP} + G_{RFP} + G_{RFP} = 24dBm + 90dBm + 2dBi + 2dBi = 118dB \quad [44]$$

The gain is the difference between the link budget of an RFP-RFP link and the minimum link budget of the RFP-PP link or PP-RFP link:

$$gain = LB1 - \min(LB3, LB4) = 118dB - 111dB = 7dB \quad [45]$$

Consider a transmitting base station RFP2. When using a base station RFP1 as a receiver at a given place, the Signal to Noise Ratio experienced by RFP1 would be $7dB$ higher than the Signal to Noise Ratio experienced by a portable part at the same place given the same transmitting base station RFP2.

Example 2:

Consider a base station with the minimum sensitivity:

$$S_{RFP} = -86dBm \quad [46]$$

The link budget for the link from RFP1 to the portable part is build up of the transmission power of the base station, the receiver sensitivity of the portable part and the antenna gains of both the base station and the portable part:

$$LB3 = P_{RFP} - S_{PP} + G_{RFP} + G_{PP} = 24dBm + 86dBm + 2dBi + 0dBi = 112dB \quad [47]$$

The link budget for the link from the portable part to RFP1 is build up of the transmission power of the portable part, the receiver sensitivity of the base station antenna gains of both the base station and the portable part:

$$LB4 = P_{PP} - S_{RFP} + G_{RFP} + G_{PP} = 19dBm + 86dBm + 2dBi + 0dBi = 107dB \quad [48]$$

The link budget for the link from a base station to another is build up of the transmission power of a base station, the receiver sensitivity of a base station and the antenna gain of a base station:

$$LB1 = LB2 = P_{RFP} - S_{RFP} + G_{RFP} + G_{RFP} = 24dBm + 86dBm + 2dBi + 2dBi = 114dB \quad [49]$$

The gain is the difference between the link budget of an RFP-RFP link and the minimum link budget of the RFP-PP link or PP-RFP link:

$$gain = LB1 - \min(LB3, LB4) = 114dB - 107dB = 7dB \quad [50]$$

We see that the gain is again $7dB$. In Appendix E we show that this is also the maximum gain.

A gain can also be explained in another way. Suppose that we have a reference base station RFP1 and a portable, at distance d, communicating with this base station RFP1. A gain of 7dB, means that we can place a base station X at a distance $10^{\frac{7}{35}} \cdot d = 1.58 \cdot d$ and reach the same link quality as for link between base station RFP1 and a portable part at distance d. This means that when we compare speech with S-field or A-field reception, we can take benefit of this additional $7dB$.

4.3.4.4 Conclusions

	RFP and PP with same transmission power, receiver sensitivity and antenna gain		RFP and PP with different transmission power, receiver sensitivity and antenna gain	
	$\frac{d_{S-field}}{d_{speech}}$	$\frac{d_{A-field}}{d_{speech}}$	$\frac{d_{S-field}}{d_{speech}}$	$\frac{d_{A-field}}{d_{speech}}$
Non-fading environment	2.17	1.47	3.43	2.33
Rayleigh fading environment	0.96	Not available	1.53	Not available

Table 5. Distance ratios under different circumstances

From Table 5 we see that the distance ratios in a non-fading environment, with RFP and PP having different transmission power, receiver sensitivity and antenna gain, are higher than 2. This means that base stations do not have to be placed closer to each other. However the most realistic situation is a Rayleigh fading environment and RFP and PP having different transmission power, receiver sensitivity and antenna gain. In that situation, the S-field requirement increases the number of base station with a factor $(2/1.53)^2 = 1.7$. We also saw in Section 4.3.4.2 that correct reception of the A-field requires a higher Signal to Noise Ratio than correct reception of the S-field, which means:

$$\frac{d_{S-field}}{d_{speech}} > \frac{d_{A-field}}{d_{speech}} \qquad [51]$$

Because both corrupted S-fields and corrupted A-fields cause synchronization errors we conjecture:

$$\frac{d_{DECT\_Air\_Synchronization}}{d_{speech}} << 1.53 \qquad [52]$$

This means that the number of base station must be multiplied with a factor much higher than 1.7 in order make synchronization via the DECT Air Interface possible.

Remark:

We mentioned in Section 4.3.4 (assumption 5) that if the distance between base station is found to be appropriate for synchronization via the DECT Air Interface, we will repeat calculation considering a simultaneous synchronization loss probability due to an incorrect S-field reception or an incorrect A-field reception. However, we have found that the distance between base station must decrease (substantially) in order to make synchronization via the DECT Air Interface possible. As a result we will not repeat our calculations.

4.3.5 Complexity

The complexity with DECT air synchronized system is introduced by:

- Base stations must be positioned in such a way that every base station has another base station to synchronize with.

- Implementation of an algorithm that performs the following functions:

    - Determine the synchronization sequence that leads to the minimum average number of hops to the master base station (e.g. smaller average number of hops if RFP2 is the master base station in the system of Figure 17).
    - Determine the new synchronization sequence when adding or removing a base station to the system.

- Implementation of a delay compensation mechanism that solves the problem mentioned in Section 4.3.2

4.3.6 Problems to solve

If synchronization via the DECT Air Interface is chosen the following problems have to be solved:

- Measurement of the locking range, and its relation to the price increase.
- Development of an algorithm that decides on which base station a new inserted base station has to lock in order to be synchronized with all surrounding base stations. This algorithm must also define the actions taken, when a base station fails, in order to give base station on a lower synchronization level a new reference base station.
- Find the possibilities of the DECT standard, in relation to the synchronization feedback (reference base station listens to locking base station).

4.4 Synchronization via Ethernet

At this moment, the Ethernet medium is the common wire of base stations within Datalab. Because it is undesirable placing additional wires, synchronization via the Ethernet medium is the only way of synchronization via wires.

4.4.1 Distribution Algorithm

Ethernet is a Multiple Access Protocol, which is suitable for broadcast from one station (can also be a base station) to a group of stations. By letting one base station act as a master base station and by defining a packet that can be recognized by all base station as a synchronization packet, Ethernet can be used for distribution of the synchronization packet.
Because Ethernet is already suitable for broadcast to multiple stations, it is not needed to develop a distribution algorithm.

4.4.2 Delay

When using the Ethernet Multiple Access Protocol for synchronization, the synchronization packet from a master base station to a slave base station suffers a delay that consists[i] of:

- A deterministic propagation delay (see Figure 27) due to the propagation time of the packet on the medium. This delay is fixed for every two stations and can be compensated for.
- A stochastic delay (which can not be compensated for) introduced by non-empty transmission buffers and busy lines. It is unknown how long a synchronization packet must wait before it is put on the line.
- A stochastic delay introduced by collisions. When a packet collides, a backoff algorithm comes into operation introducing stochastic waiting times.
- A stochastic delay introduced by processing the synchronization packet by both the master base station and the slave base stations.

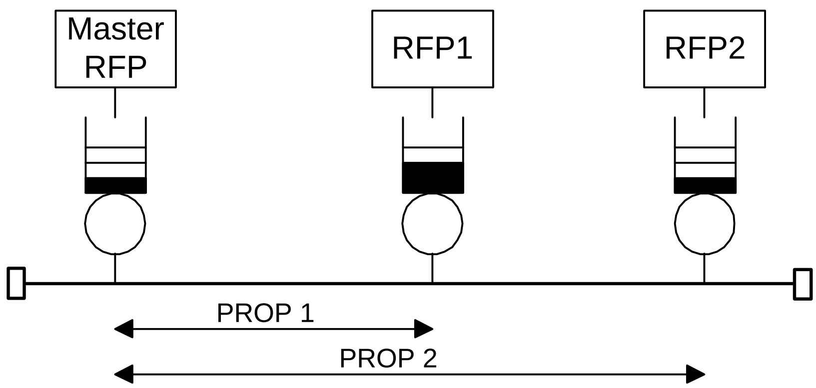


Figure 27. Propagation delay with Ethernet synchronization

[i] There may be other delays than the ones mentioned in this section. If evaluation of the possible solutions show that synchronization via Ethernet is the best solution, we will expand this list in the following chapter.

Assuming that base station are fit out with a stable clock like mentioned in Section 3.5.1.1, a slave base station needs not to receive a synchronization pulse every 10ms (see Figure 15). However, a slave base station must receive a synchronization pulse at least once every $MIST$.
As a result, we can define a number of attempts[i] to make during a period of $MIST$, and choose the number of attempts such that the probability of losing synchronization after a period of $MIST$ is at most equal to the Synchronization Loss Probability of Equation 9.

Stochastic delay can not be compensated for and must be eliminated. Stochastic delays due to non-empty transmission buffers and busy lines can be eliminated by letting a base station transmit a synchronization pulse only when the line in free and the transmission buffer is empty. In behalf of this, the probability of a busy line and the probability of a full transmission buffer at the master base station must be taken in account when determining the number of attempts to make during a period of $MIST$.

When evaluation of the possible solution for synchronization show that we have to continue with the Ethernet solution, a mechanism must be developed for elimination (or at least limitation) of the stochastic delay due to collisions and processing times of the synchronization packet.

4.4.3 Jitter

Due to the stochastic delays mentioned in Section 4.4.2, synchronization pulses will not arrive at the expected[ii] time and introduce jitter. In order to limit the jitter, the size of stochastic delays must be limited or eliminated when possible. In the previous section we mentioned how stochastic delay due to non-empty transmitter buffer and busy line can be eliminated. In order to eliminate stochastic delays due to collisions, retransmission of collided packets must be canceled.
Stochastic delays due to processing times of the synchronization packet by both a master base station and slave base stations can not be eliminated. They can be limited by:

- Developing a synchronization mechanism in which processing of a synchronization can not be interrupted.
- Making access to the system bus deterministic.

4.4.4 Cost

The price of this solution is mainly introduced by the capacity decrease of the Ethernet due to the transmission of synchronization packets. This illustrated by the following calculations.

A correct synchronization packet has to be received by the slave base station once every $MIST$ (Maximum Inter-Synchronization pulse Time). In this, the probability of losing synchronization (see also Equation 9) is:

$$SLR = (P_{nosync})^{attempts} = 3.734 \cdot 10^{-9} \quad [53]$$

In Equation 53, P_{nosync} is the probability that a synchronization attempt fails, this means that no correct (or even no) packet is received by the slave base station. Notice that attempts are made at deterministic times, we will explain this in Section 5.3.

---

[i] When a master base station tries to send a synchronization pulse to a slave base station, we say that the master base station is making an attempt. Due to the possibility of collisions and a full transmission buffer at the master base station, it is not guaranteed that the synchronization pulse arrives in time at the slave base station. Slave base stations discard a late synchronization pulse and the attempt is said to fail.

[ii] Considering a 100Hz reference timer at the master base station, the time between two synchronization pulses at the slave base station is an integer multiple of 10ms.

A synchronization attempt fails if the line is busy or the Ethernet frame (packet) is received incorrectly. Assume that a synchronization frame is 1024 bits long and that if one or more bits are corrupted that the frame is corrupted. When the Bit Error Rate on the Ethernet medium is 10^{-6} the Ethernet Frame Error Ratio is:

$$FER = (1-(1-BER)^{1024}) = (1-(1-10^{-6})^{1024}) = 1.023 \cdot 10^{-3} \quad [54]$$

Assume that the probability of a busy line is equal to the line occupation, and that the line occupation has a value of 30%:

$$P_{busy-line} = 0.3 \quad [55]$$

Remark: If evaluation shows that we must continue with the Ethernet solution, we will try to give a better approximation of the probability of a busy line.

In order to give a first approximation we assume that a synchronization attempt fails if the synchronization packet is corrupted or when the line is busy:

$$P_{nosync} = 1-((1-FER)\cdot(1-P_{busy-line})) = 0.3007 \quad [56]$$

We see that a Bit Error Rate of 10^{-6} has very low influence on the probability P_{nosync} that a synchronization attempt fails. In other words P_{nosync} is almost equal to $P_{busy-line}$.

Substitution of Equation 56 in Equation 53 gives the minimum number of synchronization attempts that must be made during a time period equal to $MIST$ in order to reach the Synchronization Loss Ratio of Equation 9:

$$attempts = \frac{\log(SLR)}{\log(P_{nosync})} = \frac{\log(3.734 \cdot 10^{-9})}{\log(0.3007)} = 16.149 \rightarrow 17 \quad [57]$$

We assumed that the size of a synchronization packet is equal to 1024 bits. During a time period equal to $MIST$, an amount of $MIST \cdot 10 \cdot 10^6$ bits can be transported over a 10 Mbps Ethernet medium. However, the synchronization mechanism transmits at most 17 synchronization packet over the Ethernet medium, each with a packet size of 1024 bits. As a result the usage[i] (h) by the synchronization packets will be:

$$h = \frac{17 \cdot 1024}{MIST \cdot 10 \cdot 10^6} = \frac{17 \cdot 1024}{1.56 \cdot 10 \cdot 10^6} = 1.116 \cdot 10^{-3} = 0.112\% \quad [58]$$

Where $MIST$ is given by Equation 4.

The value in Equation 58 is equal to a bit rate (R_{sync}) of:

$$R_{sync} = h \cdot 10 \cdot 10^6 = 11159 \text{ bps} \quad [59]$$

From Equation 58 we see that the capacity usage by the synchronization mechanism is very low. We conclude that there is almost no price increase due to capacity decrease of Ethernet. However, we expect some price increase due to additional hardware for processing of the synchronization packets (e.g. forwarding of synchronization pulses from Ethernet chipset to DECT Burst Mode Controller).

[i] The percentage of the Ethernet capacity that is used by the synchronization mechanism. This percentage of the Ethernet capacity can not be used for data-communication.

4.4.5 Complexity

The complexity of this solution is introduced by the detection of a busy line. As we have mentioned before, a synchronization packet is sent, only when the line is free. To be able to take such a decision, we have to monitor the carrier sense. Most Ethernet Network Interface Cards handle the carrier sense information on the chipset. In other words, the carrier sense information in unknown at higher levels.
Applications just forward packets to the Network Interface Card, which decides whether to put the packet on the line or not, depending on the carrier sense information. As a result Ethernet chipsets that provide carrier sense information or Ethernet chipsets that can be modified to provide carrier sense information must be found.

Another aspect that increases the complexity is the development of hardware that provides at least the following functions:

- Generate synchronization packet at appropriate times.
- Transmit the synchronization packet when allowed.
- Detect the reception of a synchronization packet and apply a synchronization pulse to the Burst Mode Controller after the delay is compensated for.

The hardware to develop must be developed such that stochastic delays are minimized, which minimizes the jitter (see Section 4.4.3).

4.4.6 Problems to solve

When choosing the solution of DECT base station synchronization via Ethernet, at least the following problems must be solved:

- Modify Ethernet source code in order to be able to “sense” the line.
- Compensate for (propagation) delay between the master RFP and a slave RFP.
- Study the possibility of interconnected networks (e.g. Switched Ethernet).

4.5 Evaluation of the possible solutions

The properties of each of the solutions are listed in Table 6.

	Synchronization via	
	DECT Air Interface	Ethernet
Distribution algorithm	Find or develop a distribution algorithm	Use Ethernet protocol
Delay	Deterministic due to fixed distance between every pair of base stations	Stochastic delay must be eliminated
Jitter	Due to clock correction	Due to stochastic delays and clock correction
Cost	At least 70% more RFPs to cover the same area RFP	Slight Ethernet capacity decrease and additional Hardware
Complexity	Distribution algorithm	Ethernet driver modification, delay compensation and Hardware to develop
Reuse	In all kind of networks with DECT base stations	In a Ethernet network with all kind of base stations (cordless systems)
Quality	Required Synchronization Loss Ratio is reachable of the distance between base stations is small enough	Required Synchronization Loss Ratio is reachable if the stochastic delays can be limited or avoided and if the carrier sense information can be used by the synchronization mechanism.

Table 6. Properties of the different solutions

Cost and quality are the most important properties of Table 6. As can be seen, in order to make synchronization via the DECT Air Interface possible, at least 70% more base stations must be placed covering the same area. Once these extra base stations are placed, we expect that the required Synchronization Loss Ratio can be reached.
In the case of synchronization via Ethernet, the required Synchronization Loss Ratio can be reached if the mentioned requirements are met. However, we expect the cost of synchronization via Ethernet to be lower than synchronization via the DECT Air Interface. As a result we have decided to continue with the **Ethernet solution**.

5 Base station synchronization via Ethernet

5.1 Introduction

In the previous section it was concluded that synchronization via Ethernet is the most promising solution for DECT base station synchronization. In this chapter we analyze and design the solution in more detail.
An overview of Ethernet is given in Section 5.2 while the synchronization process via Ethernet is introduced in Section 5.3. In Section 5.4 we evaluate the performance of Ethernet synchronization.
In Section 5.5 we explain how delay can be measured and compensated for while in Section 5.6 we explain how synchronization via Ethernet can be performed. Finally we give a possible Hardware implementation of synchronization via Ethernet in Section 5.7.

5.2 Ethernet overview

Within Ethernet all the nodes share the same medium, which is the bus[i] [Quinn]. Because only one node at a time can use the bus, every node 'listens' to the carrier before transmitting. A carrier exists only when another node is transmitting, which means that the presence of a carrier indicates that the medium is in use and the listening node (or nodes) should defer to the currently transmitting node.
A node that has a frame to transmit (pending frame) must wait for a minimum period of time after the end of the last transmitted frame before transmitting its next frame. This time is called *Inter-Packet Gap* (IPG). All nodes on the network must comply with this rule. Even if a node has multiple frames to transmit and is the only node that needs to transmit, it must ensure there is at least the IPG time between each packet that it transmits. The reason for this is that the Inter-Packet Gap allows the station that last transmitted, to cycle its circuitry from transmit mode to receive mode [IOCGAP]. Figure 28 illustrates the Inter-Packet Gap.

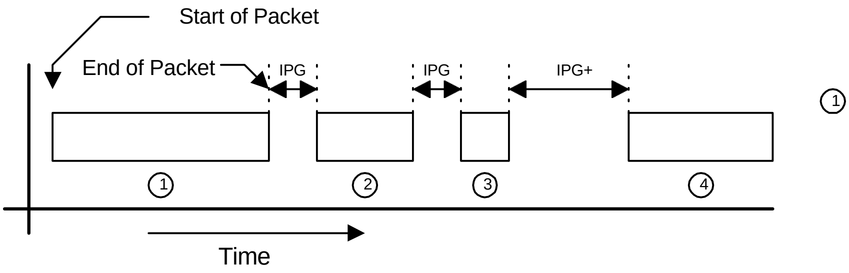


Figure 28. Inter-Packet Gap

As mentioned before, when a node senses no carrier, it can start transmitting a frame after the IPG has expired. However collision-less transmission is not guaranteed because a packet leaving a node reaches another node after a certain propagation time in which the another node is still experiencing a free channel and is allowed to put a packet on the line. The result is that the two packets will be corrupted. This effect is known as a collision.

With a collision detect (CD) mechanism, nodes are able to detect collisions and recover from them. After a collision takes place, the transmitting nodes send a jam pattern and enter a backoff mode.
The Ethernet protocol IEEE 802.3 uses a Binary Exponential Backoff algorithm. After a collision, time is divided up into discrete time slots whose length is equal to the worst case round-trip propagation time on the medium [Quinn]. To accommodate to the longest path allowed by IEEE 802.3 (2.5 km and four repeaters)

[i] In most Ethernet version this is a coax cable. Sometimes fiber-optics are used.

the slot time has been set to 512 bit times. Briefly explained, when a packet has collided i times, a random number between 0 and $2^i - 1$ is chosen, and that number of slots is skipped.
Figure 29 shows the mentioned aspects.

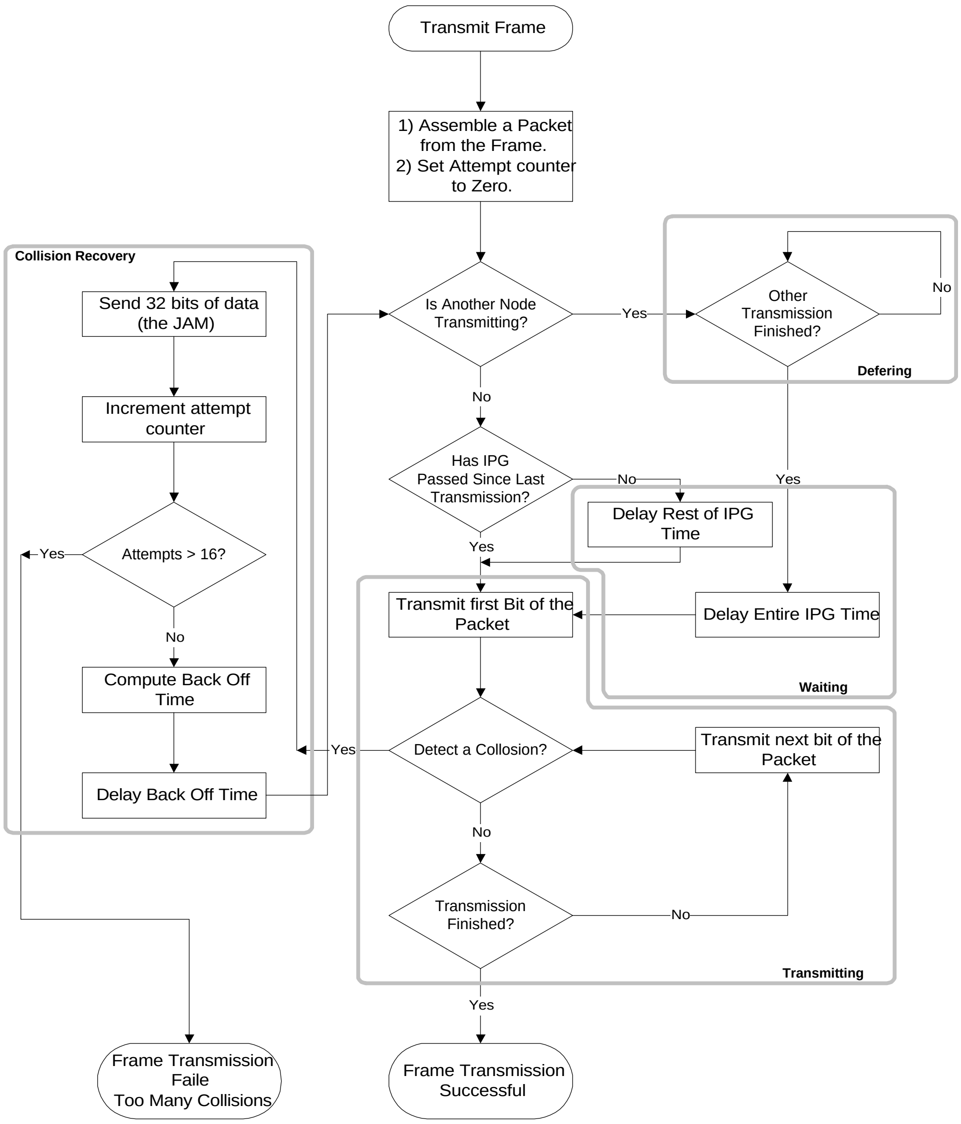


Figure 29. Schematic view of Ethernet

5.3 Synchronization process via Ethernet

In Section 3.5.1.1 we mentioned that we use a stable clock that must be synchronized at least once every $MIST$ seconds. The stable clock does not expect a synchronization pulse every 10 ms, but expects the synchronization pulse to come at a predefined place within a DECT frame (10 ms). In other words the time difference between successive synchronization pulses must be an integer multiple of 10 ms, with a maximum of $MIST$ seconds.
If the probability of a synchronization success is 1, we need not to try to synchronize more than once every $MIST$. But since this is not the case, more attempts have to be made. The number of attempts depends on the allowed Synchronization Loss Ratio (SLR), the Maximum Inter-Synchronization pulse time ($MIST$) and the failure probability of a synchronization attempt (see also Equation 53).
When the number of attempts and $MIST$ are known, the Inter-Attempt Time (IAT) can be derived.

In our solution, we want to synchronize via Ethernet. A way to do this is to define a synchronization packet that is recognized as such by all the attached base station. The idea is to try to send a synchronization packet whenever the attempt time has come. Of course, when we want to send a synchronization packet, the Ethernet medium (line from now on) must be free. If we let the Ethernet protocol do its work, the synchronization packet will be stored in a buffer until the line is free. Delayed transmission will then take place introducing a stochastic delay. In order to avoid this, we cancel the transmission of a synchronization packet when the line is not free at attempt times, and make another attempt when the next attempt time has come.
To know how many attempts have to be made every $MIST$, the probability of an unsuccessful attempt must be known. In order words, we have to know the probability of a busy line at attempt times. In Section 5.4 we will try to determine the necessary number of attempt by means of performance evaluation of the Ethernet. The number of attempts found will also give an indication of the impact of synchronization on the Ethernet capacity.

5.3.1 Transmission requirements

Normally, when an application wants to send a packet, it moves this packet to the Network Interface Card (NIC), which in its turn keeps the packet in a buffer until the line is free and all preceding packets or packets with a higher priority have been transmitted.

Suppose that the time to make a synchronization attempt has come. If the master base station just moves the synchronization packet to the network interface card, it will be possible that the synchronization packet finds a busy transmission channel, or even a non-empty network interface card buffer. In that case the synchronization packet will suffer an unpredictable delay that can not be compensated for.

In order to avoid this stochastic transmission delay, we have set up some requirements concerning the decision to send a synchronization packet or not, i.e. the base station is only allowed to send a synchronization packet if both the line is idle and the network interface card buffer is empty.
An alternative would be just requiring an idle line, but placing the synchronization packet at the front of the network interface card buffer.

Another important aspect is that even when the line is free at attempt times, collisions still can take place. The Ethernet protocol initializes a backoff algorithm and delayed retransmission takes place. This means that an unpredictable delay is introduced. In order to avoid these unpredictable delays, retransmission of collided synchronization packets must not take place and a new attempt must be made when the next attempt time has come.

Furthermore, there is a predictable delay between a master base station and the slave base stations (signal propagation delay, download delay etc…). This delay must be determined and compensated for. The determination of this delay is explained in Section 5.5.

5.3.2 Reception requirements

Normally, when a packet arrives at the network interface card, the packet is uploaded and an interrupt is generated. After a certain time called ISR (Interrupt Service Response time), the computer begins handling the interrupt. We found that ISR has a large variance (with VxWorks, which is one of the fastest kernels), which is unacceptable for our synchronization solution. In other words, processing the synchronization packet by means of a software interrupt introduces an intolerable stochastic delay. A hardware solution is required.

5.4 Performance of Ethernet

When the medium is Idle, and the master base station (which is also a node) has no other packets to transmit, a synchronization packet can be transmitted with a deterministic transmission delay.
In Section 4.4.4 we assumed that the probability of a busy line is equal to the network utilization. This is not the reality. In this section we will examine the performance of Ethernet more accurately. The reason for this is that we want to have some quantitative measures about the idle state of the medium. We then will transform these measures to number of required synchronization attempts, which in turn can be converted to an additional load on the Ethernet.

5.4.1 Introduction

Consider the case in which two nodes are trying to transmit a packet within the collision window (time window in which a node cannot notice that another node is transmitting). After the packets collide and the nodes notice this, time will be divided into discrete time slots (contention slots) of twice the round trip time (standardized to 512 bit-times).

For our performance evaluation, we assume a heavily loaded system with N nodes where all N nodes compete for a time slotted medium [Tanenbaum]. An analysis of the binary exponential backoff algorithm is complicated. Instead we will assume a constant retransmission probability (p) in each slot and that the binary exponential backoff algorithm optimizes p .

If each station (node) transmits during a contention slot with probability *p*, the probability that some station acquires the channel in that slot is [Walrand]:

$$a\,(p,N) = N \cdot p \cdot (1-p)^{N-1} \qquad [60]$$

In Equation 60, a successful acquiring of the channel takes place when one station tries to transmit and $N-1$ stations do not, which is possible in N ways.

The maximum $a\,(p,N)$ in relation to p occurs when the derivative of $a\,(p,N)$ to p is zero, so:

$$\frac{d(a\,(p,N))}{dp} = N \cdot (1-p)^{(N-1)} - N \cdot p \cdot (1-p)^{(N-1)} \cdot \frac{(N-1)}{(1-p)} = 0 \qquad [61]$$

Division of Equation 61 by $N \cdot (1-p)^{(N-1)}$ gives:

$$1 - \frac{p}{1-p} \cdot (N-1) = 0 \qquad [62]$$

Multiplication of Equation 62 by $1-p$ gives:

$$1 - p - p \cdot (N-1) = 0 \qquad [63]$$

Rewriting Equation 63 gives:

$$p = \frac{1}{1+N-1} = \frac{1}{N} \qquad [64]$$

From Equation 64, we see that the maximum probability of a success is reached for $p = \frac{1}{N}$, and that this maximum approaches $\frac{1}{e}$ for very large N. Equation 65 and Figure 30 illustrate this fact:

$$\lim_{N\to\infty} a\left(\frac{1}{N}, N\right) = \lim_{N\to\infty} N \cdot \frac{1}{N} \cdot \left(1-\frac{1}{N}\right)^{N-1} = \left(\frac{N-1}{N}\right)^{N-1} = \frac{1}{e} \qquad [65]$$

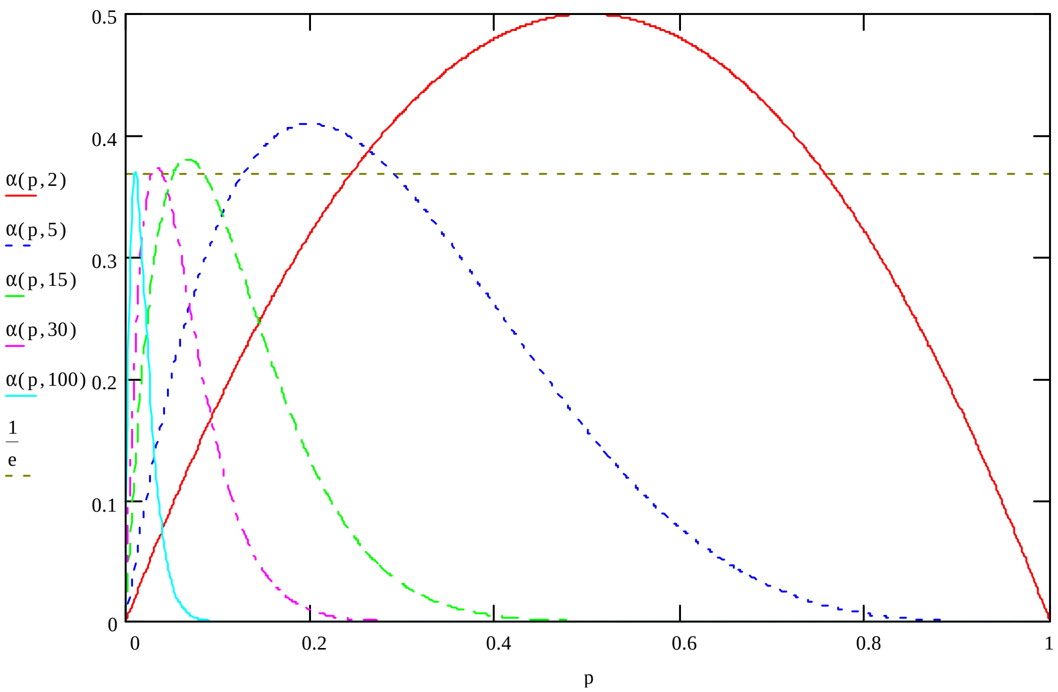


Figure 30. Probability of a success as a function of the probability that a node transmits a packet

Suppose that each node produces a packet with probability p. If p is optimized to the number of stations ($p = \frac{1}{N}$), we get the $a(p, N)$ (probability of success) shown by Figure 31.

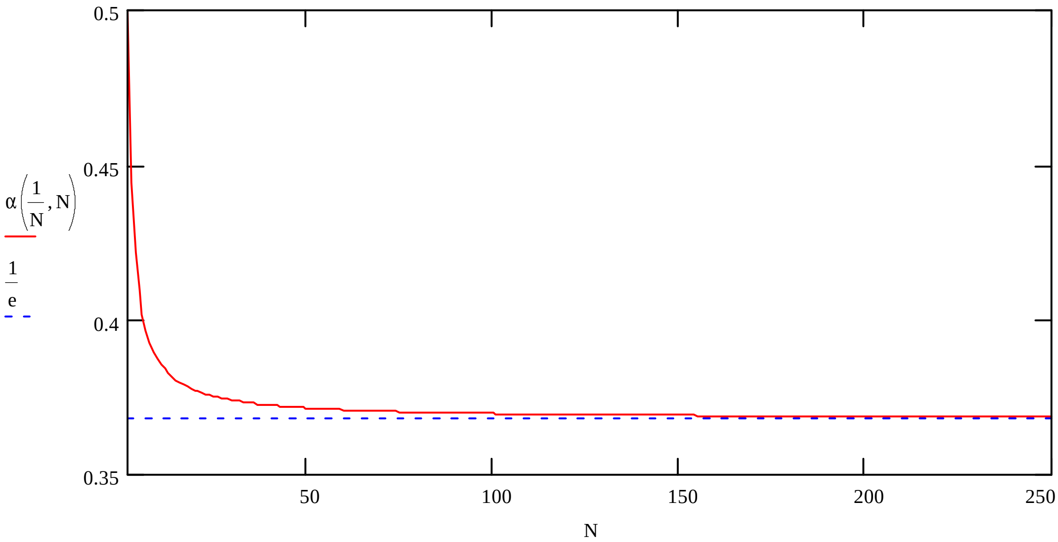


Figure 31. Maximum probability of a success in a time slot as a function of N

In the DECT synchronization problem, synchronization succeeds only when no node has a packet to send. We then say that the line is idle.
The probability of an idle line during a contention slot is:

$$b(p,N) = (1-p)^N \qquad [66]$$

When p is optimized to the number of nodes ($p = \frac{1}{N}$) and then number of stations is large, the maximum probability of an idle line during a contention slot is equal to:

$$\lim_{N\to\infty} b\left(\frac{1}{N}, N\right) = \lim_{N\to\infty} \left(1-\frac{1}{N}\right)^N = e^{-1} \qquad [67]$$

This fact that is illustrated by Figure 32.

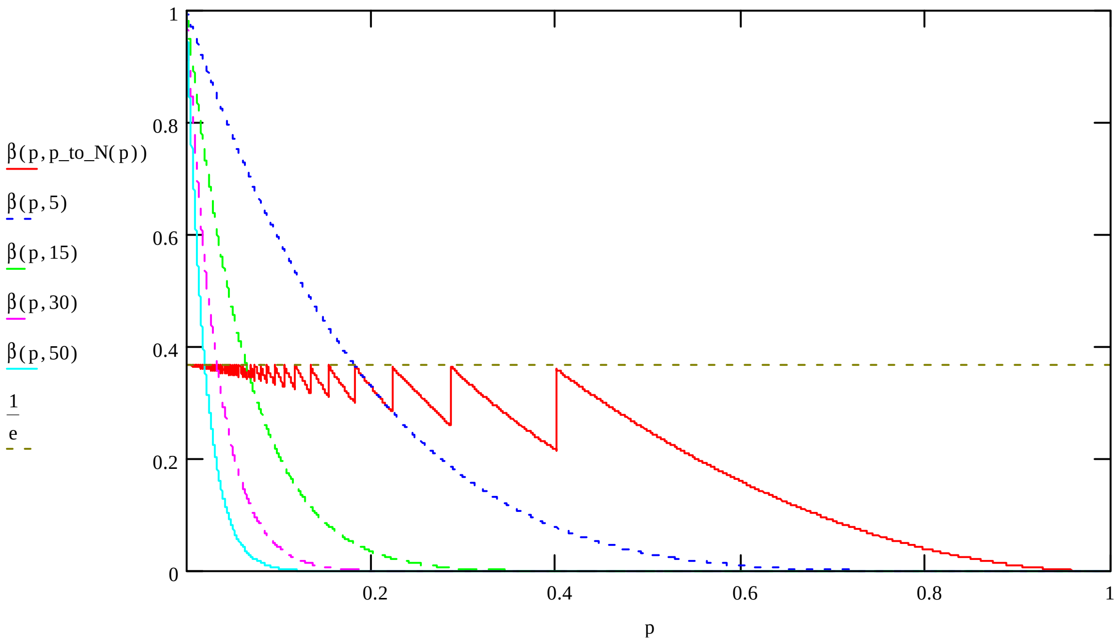


Figure 32. Maximum probability of an idle line during a contention slot

- Figure 32 $p\_to\_N(p) = round(\frac{1}{p})$ is an auxiliary function that converts the probability p that a node transmits a packet to a number of nodes N for which the value of p is an optimum ($p = \frac{1}{N}$). The minimum value of N is two nodes, because there is no network with less than two nodes.

5.4.2 Wasted time slots

Consider the diagram in Figure 33. The figure contains two states: “start” and “stop”. State “start” is the state in which contending stations try to acquire the line. State “stop” is the state in which one station acquires the line and begins a successful transmission. The arrows are labeled with an expression of the form $[W,P]$. W indicates the number of wasted contention slots while P is the probability of the transition.

The diagram in Figure 33 starts in state “start” where contending stations try to acquire the line during a contention slot. With probability $1-a$ more than one station tries to begin transmission during that contention slot, which means that one contention slot is wasted and the diagram return to state “start”. With probability a one station acquires the line and starts transmission during that contention slot, which means that the diagram moves to state “stop” in which successful transmission starts. In this case no contention slot is wasted.

The average number of wasted contention slots is the average number of extra time slots (besides the transmission time) needed to transport a packet from one node to another. When a packet transmission succeeds at the first attempt, there will be no wasted contention slot. When transmission does not succeed at the first attempt, one time slot will be wasted and a new attempt will be made.

The average number of wasted contention slots can be extracted from Figure 33 by calculating the average number of wasted contention slots when moving from state “start” to state “stop”.

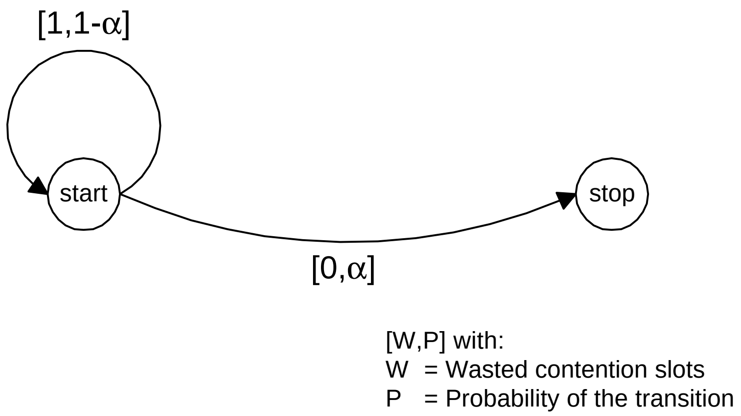


Figure 33. Wasted contention slots

In Figure 33, a station acquires the line with probability a (i.e. no wasted contention slot), whereas $(1-a)$ denotes the probability of more than one station trying to acquire the line (i.e. a wasted contention slot).

Let A be the average wasted time. According to [Walrand] the average wasted time is given by:

$$A = a \cdot 0 + (1 - a) \cdot (1 + A) \qquad [68]$$

Thus:

$$A = a^{-1} - 1 \qquad [69]$$

According to [Walrand] simulations have shown that the amount of wasted time is close to $A = 2.5$ timeslots, which means that the probability of a success in a contention slot is equal to:

$$a(p, N) = \frac{1}{A+1} = 2/7 \qquad [70]$$

We see that $a < e^{-1}$. One of the reasons for this is that in real networks $p \neq \frac{1}{N}$.

For a given N, the value of a found in Equation 70 can be converted numerically (Mathcad) to a value of p using Equation 60. The results are shown in Figure 34.

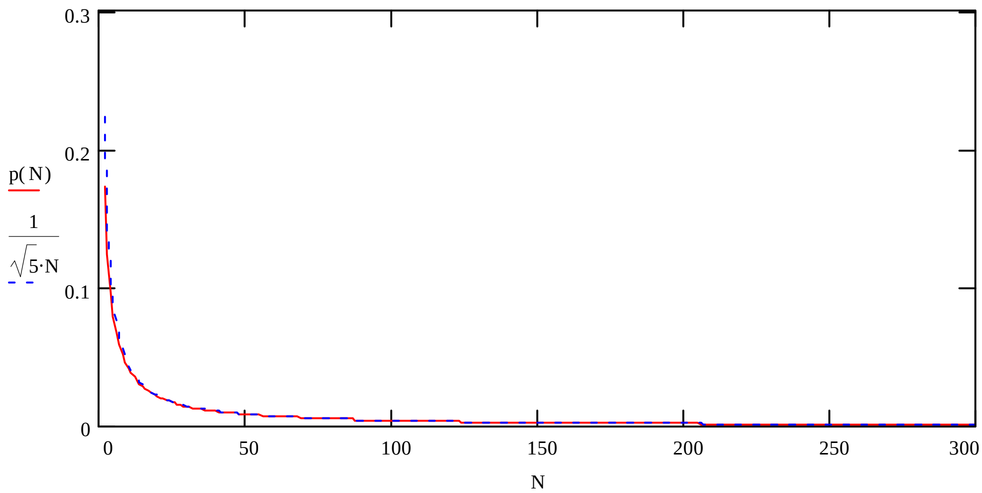


Figure 34. *p* as a function of *N* for α=2/7

In Figure 34 we see that the value of p is approximately equal to $\frac{1}{\sqrt{5 \cdot N}}$ (we found it experimentally), especially for large N. This means for large N, that the probability of an idle line ($b(p, N)$) within a contention slot is equal to:

$$\lim_{N \to \infty} b\left(\frac{1}{\sqrt{5 \cdot N}}, N\right) = \lim_{N \to \infty} \left(1 - \frac{1}{\sqrt{5 \cdot N}}\right)^N = e^{-\frac{1}{\sqrt{5}}} \qquad [71]$$

This is illustrated by Figure 35.

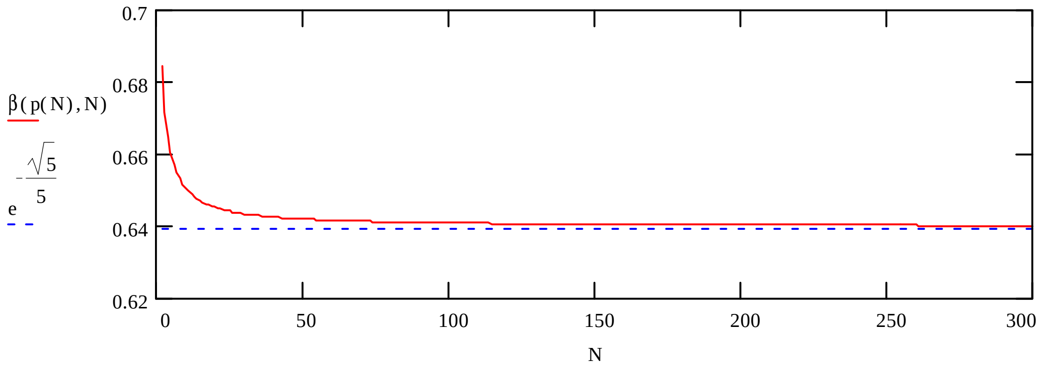


Figure 35. Probability of an Idle line (during a contention slot) as function of N for α=2/7

It is shown that the minimum probability of an idle line is equal to $b_{\min} = e^{-\frac{1}{\sqrt{5}}}$, which is the probability of an idle line within a contention slot for large N.

5.4.3 *From network utilization to idle line probability*

Unfortunately, $b(p, N)$ does not inform us about the total probability of an idle line. The following derivations will lead a total probability of an occupied line ($= 1 - p_{idle}$).

Over any given period of time on any particular Ethernet, some number of databytes will be transmitted. When this number is divided into the maximum possible throughput of Ethernet, we get a number called *network utilization*. Network utilization is a good measure of total LAN performance. For busy Ethernet systems, a network utilization of 30% percent is considered to be a good target [Quinn]. This means that the average packet transmission rate is equal to:

$$S = 0.3 \cdot C / PL[packets/\sec] \quad [72]$$

With C the line capacity and PL the packet length. However every packet resides on the line for an average period of [Walrand]:

$$R = TRANSP + 2.5 \cdot slots[\sec] \quad [73]$$

With $TRANSP$ the transmission time of a packet.
A slot is standardized to 512 bit-times, which means that the average time that a packet resides on the line is equal to:

$$R' = PL + 2.5 \cdot 512 = (PL + 1280)[bit - times] \quad [74]$$

Where bit-times is the transmission time of a bit.
The fraction of time that a line is busy is equal to:

$$B(PL) = \frac{S \cdot \frac{R'}{[bit - times]}}{C} = \frac{(0.3 \cdot C / PL) \cdot (PL + 1280)}{C} = \frac{0.3 \cdot (PL + 1280)}{PL} \quad [75]$$

We see that this value varies for different average packet lengths, with a worst case for minimum average packet lengths, i.e. for packets with a length of 64 bytes. The best case is when the average packet length is the maximum packet length, i.e. for packets with a length of 1518 bytes.

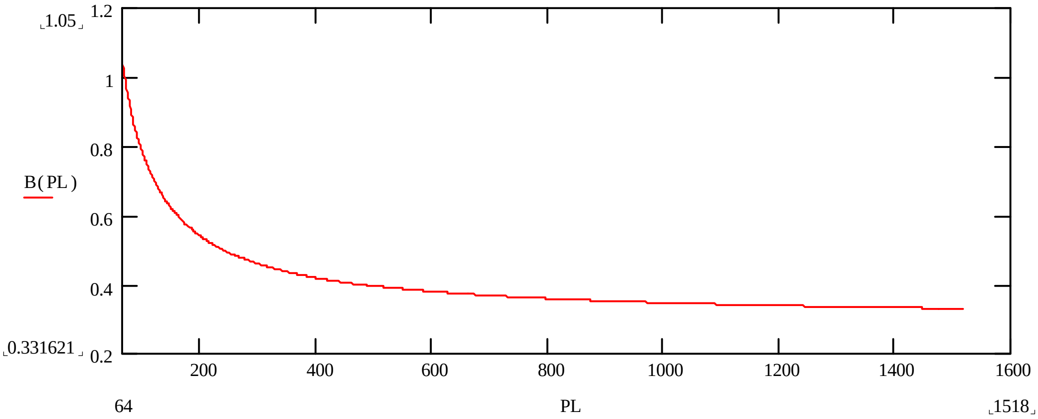


Figure 36. Line occupation versus packet length for a typical [Walrand] Ethernet

If we assume a typical average packet length of 500 bytes a line occupation of $B(500) \approx 0.4$ will be found. This will lead a required number of attempts equal to (see also Equation 57):

$$attempts = \frac{\log(3.7 \cdot 10^{-9})}{\log(0.4)} \approx 21 \quad [76]$$

A period of 1.56 seconds contains 156 DECT frames. So a synchronization packet has to be sent at the beginning of every $100 \cdot 1.56 / 21 = 7.4 \approx 7^{\text{th}}$ DECT frame. We give this value the name IAF (Inter-Attempt Frames).

Assuming that synchronization packets have a length of 1000 bits, and given that every second 100 DECT frames are transmitted. An extra load of $\frac{1.56 \cdot 100}{7} \cdot 1000 = 22.29 kbits$ is generated every 1.56 seconds.

This can be translated to an increased Ethernet usage due to synchronization packets of:

$$h = \frac{22.29 \cdot 10^{3}}{10 \cdot 10^{6} \cdot 1.56} = 1.429 \cdot 10^{-3} \quad [77]$$

5.4.4 Remarks on the performance evaluation

We have made a lot of assumptions about Ethernet. A few of them are:

- A heavy loaded system is assumed, giving us the opportunity to analyze a slotted CSMA/CD network. However, this is not true for nodes with new packets. Such nodes will try to send whenever they sense a free line (at least IPG after the last busy period)
- The IEEE 802.3 Ethernet protocol uses a Binary Exponential Backoff Algorithm, whereas our CSMA/CD protocol defines the probability p, as the probability a node has a new transmission or a retransmission.
- The traffic is assumed to be Poisson. Network traffic is rarely Poisson, but self-similar. This means that averaging over long periods of time does not smooth out the traffic [Tanenbaum].
- We assumed that the probabilities of an idle line at synchronization times are independent. However if the number of attempts increases, the inter-synchronization time decreases and the line idle probabilities will become dependent of each other.

If we really want to give an idea about the expected number of attempts, we have to measure real Ethernet networks in different business areas. However at this moment we are satisfied with the results found in the previous subsection.

We conclude that an attempt to transmit a synchronization packet must be made every 70ms to obtain a synchronization mechanism that will not lose synchronization more than once a month. This will generate an additional load of the Ethernet in the order of 10^{-3} (0.1%).

5.5 Delay compensation

If the master base station transmit a packet only if the line is idle and the network interface card buffer is empty, only deterministic delays remain. The remaining deterministic delays are:

- Generation delay (GD) of the synchronization packet, which is the time needed for generation or production of the synchronization packet in addition to the time needed to place the packet in the download buffer. The download buffer is the buffer where the packets reside before they are downloaded by the Ethernet chipset (from now on chipset) to the chipset buffer.
- Download[i] delay (DD) of the synchronization packet, which is the time to move the packet from the download buffer to the chipset buffer.
- Transmission delay (TD) which is the time needed to transmit the synchronization packet at the given bitrate (i.e. the time needed to put the bits of the synchronization packet on the medium).
- Propagation delay (PD), which is the time, needed to transport a synchronization packet from a master base station to a slave base station. This is also the propagation delay in the opposite direction.
- Upload[ii] delay (UD) of the synchronization packet, which is the time needed to move the synchronization packet from the chipset buffer to the upload buffer, which is the buffer where the packets reside before they are processed.
- Processing delay (PRD) of the synchronization packet, which is the time needed to find out that the uploaded packet is a synchronization packet plus the time needed to generate a synchronization pulse.

The difference in delay between different slave base station is due to different transmission delays.

Remarks:

- GD and PRD are only deterministic if generation and processing of a synchronization packet can not be interrupted.
- DD and UD are only deterministic if access to the system bus is deterministic.
- With Switched Ethernet, packets transmitted from one station to another may be buffered on their way to the destination. This introduces a stochastic Propagation Delay (PD) which can not be compensated for. As a result, Switched Ethernet will not be supported by the mechanism of synchronization via Ethernet.

---

[i] Process of moving the packet to transmit, from the packet buffer (which is accessible for other hardware) to the chipset buffer (which is not accessible for other hardware)

[ii] Process of moving the received packet from the chipset buffer to the packet buffer

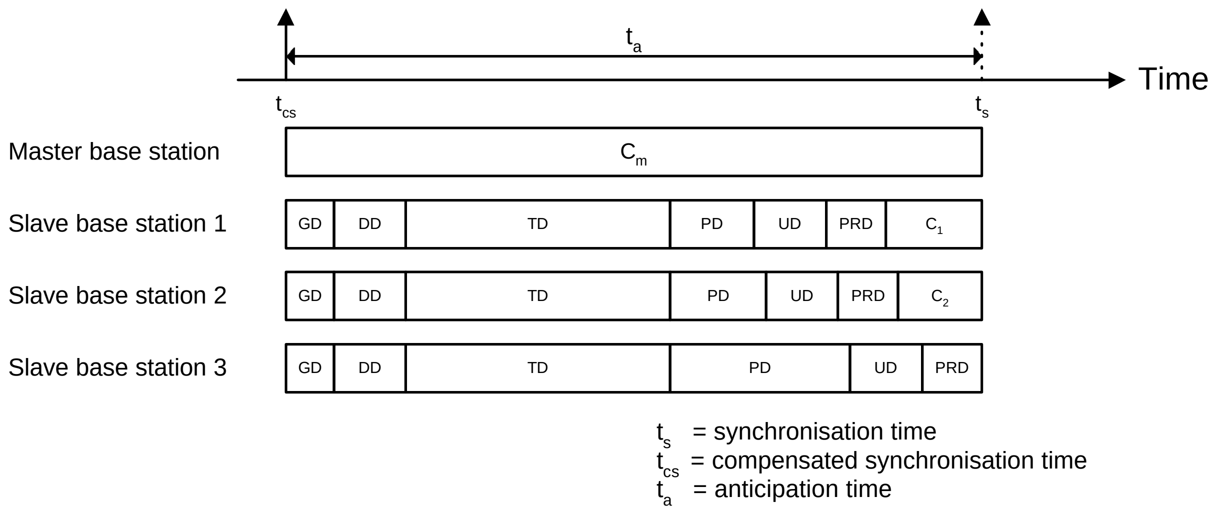


Figure 37. Delay compensation

Considering Figure 37, suppose that slave base station 1 is the nearest base station to the master base station followed by slave base station 2 and 3. The result is that the signal propagation time (PD) of slave base station 2 will be larger than that of slave base station 1. The signal propagation time (PD) of slave base station 3 will be larger than that of slave base station 2.

In order to make synchronization at slave base station 3 possible at time t_s, the master base station has to generate (with the intention to send) the synchronization packet at time t_{cs} and transmit it. The master base station has to anticipate to the maximum delay, which means that the synchronization pulse must be generated t_a seconds before t_s. As a result of this, slave base stations 1 and 2 will receive their synchronization packet too early. Letting slave base station i wait for an appropriate compensation time (C_i) can solve this problem. The master base station must also compensate for the premature synchronization pulse generation by waiting for a time C_m before reacting (synchronizing) to the synchronization pulse it has generated

The difficulty here is to determine the value of C_i. Figure 38 shows how the value of C_i can be determined.

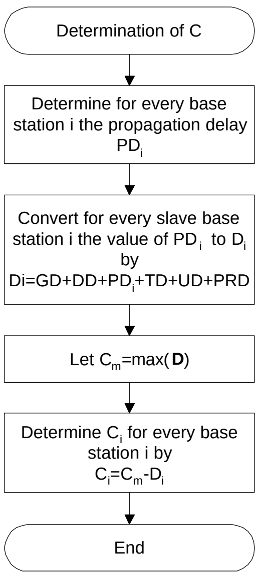


Figure 38. Determination of *C*

Remarks on Figure 38:

- D_i is the sum of all delays experienced by a synchronization packet when transmitted from a master base station to a slave base station.
- $\mathbf{D}$ is the vector $[D_1, D_2, \ldots, D_n]$ with n the number of slave base station.
- The function $\max(\mathbf{D})$ returns the largest value D_i.

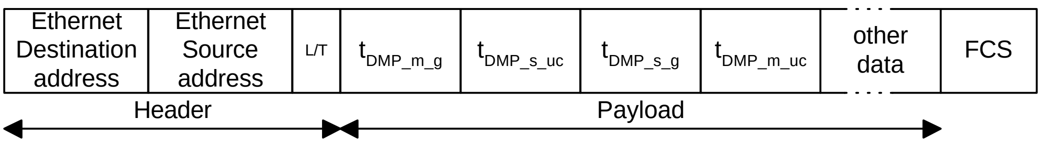
As shown in Figure 38, the propagation delay for every slave base station must be determined. In order to do this, a Delay Measurement Packet (DMP) will be defined. The structure of such a packet will be explained in the following section. Subsequently, the send and receive procedures on both the master base station side and the slave base station side will be explained.

Remark: GD, DD, TD, UD and PRD are equal for different base stations of the same type

5.5.1 Delay Measurement Packet (DMP)

As mentioned in the previous section, a Delay Measurement Packet must be defined. DMP is a packet that will be recognized as such at both the master and slave base stations. By sending the DMP from the master base station to a specific slave base station and back, the one way propagation delay can be calculated if the departure and arrival time of the DMP, at both the master base station and the slave base station are stored in the DMP.

To calculate the one way propagation delay (half the two-way propagation delay), four time stamps are necessary in a DMP. Additional information such as Upload Delay, Download Delay and others can be stored elsewhere. A possible structure of DMP is given in Figure 39.



L/T	= Length/Type
$t_{DMP\_m\_g}$	= timestamp of DMP generation at master base station
$t_{DMP\_s\_uc}$	= timestamp of DMP upload completion at master base station
$t_{DMP\_s\_g}$	= timestamp of DMP generation at slave base station
$t_{DMP\_m\_uc}$	= timestamp of DMP upload completion at slave base station
FCS	= Frame Check Sequence

Figure 39. Possible structure of DMP

By keeping up the four mentioned time stamps, the time spent at the slave base station as well as the outgoing time (time between departure and arrival of DMP at master base station), A value for C can be calculated.
For each slave base station i a value of C_i is maintained at the master base station. C_i is the value of the needed compensation time for slave base station i, but can also serve as a status indicator to show that the DMP for slave base station i is outgoing (sent to slave but not yet back) or that a DMP has not yet been sent.
Notice that every slave base station i must be informed about its delay compensation value C_i because delay compensation takes place at the slave base stations. A possible way to do this is defining a packet that contains the delay compensation values for all slave base station. After all the delay compensation values are determined such a packet must be broadcast to all slave base stations.

In the following sections, flowcharts of the events concerning the delay determination will be shown and explained.

Remarks:

- Besides the delays GN, DD, TD, PD and UD that are explained in Section 5.5, a DMP experiences the following delays:
  - Processing delay 1 ($PRD1$) is the time needed to find out that the received packet is a DMP.
  - Processing delay 2 ($PRD2$) is the time needed to take a time stamp (i.e. from the moment of request to the record moment).
  - Processing delay 3 ($PRD3$) is the time between the record moment of the time stamp and the moment the time stamp is available in addition to the time needed to place the time stamp in the DMP.
  - Processing delay 4 ($PRD4$) is the time needed to store a DMP in the DMP buffer, in order to calculate the propagation delay when resources are available.

- Clocks at different base stations, used to take time stamps may be offset, skewed and they may drift with respect to one another. However, we assume these factors to have low impact on the delay calculation.

- The Length/Type (L/T) field is used in two different ways [Quinn]. It indicates the length of the data field of the frame whenever the content of the L/T field is between 0 and 1500 (maximum payload size). If the content of the L/T field is 1500 or greater, then it is a protocol type indicator. These types are called **Ether types** and are not standardized. A list of Ether types or new Ether Type assignments can be requested by contacting Xerox System Institute (see also [IOCETYPE]).

5.5.2 DMP transmission at the master base station

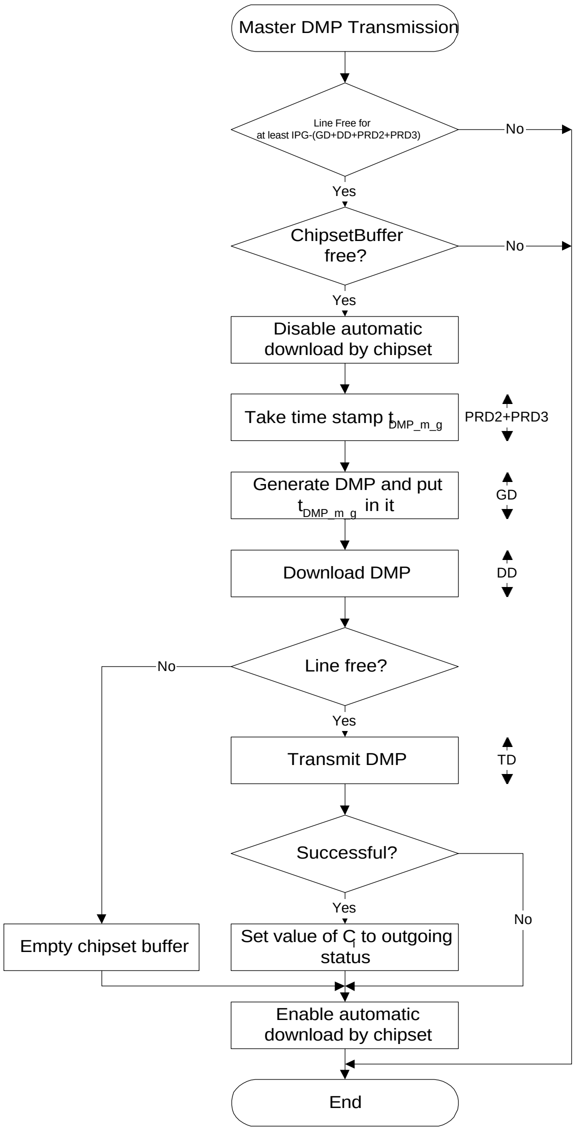


Figure 40. DMP transmission by the master base station

The transmission procedure is executed for all slave base stations i whose C_i value is still not determined and for which a DMP is not yet sent.

First, the line status is checked. As can be seen in Figure 40, the line must be free for a period of at least $IPG-(GD+DD)$. Actually the Ethernet protocol states that after every transmission, a line idle period of IPG is mandatory. Because DMP suffers an additional delay of $(GD+DD+PRD2+PRD3)$ it takes at least the mentioned amount of time before transmission starts, which means that we require a line free period of at least[i] $IPG-(GD+DD+PRD2+PRD3)$.

[i] It is assumed that the time needed to check the line or the chipset buffer is negligible in comparison with other delays. The same assumption holds for the time to enable or disable automatic download by the chipset

Next the chipset buffer is checked. If this buffer is not empty, the chipset will transmit the other downloaded packets first, causing a stochastic delay, which can not be compensated for.
Just before generating a DMP with the appropriate slave base station address, a time stamp $t_{DMP\_m\_g}$ is taken. Before taking the time stamp, the chipset must be prevented from downloading packets other than the DMP during the rest of this procedure.
Next the generated DMP is downloaded. If the line is still free, the DMP is transmitted. The value of C_i is set, such as it indicates that a DMP is transmitted to slave base station i.
If, after generation of DMP, the line is not free anymore (other stations start sending) the downloaded packet is removed from the chipset buffer in order to start the procedure another time.

Even after all checks made, a collision still can take place. Two solutions are conceivable:
1) When the collision is detected (chipset status), the transmission is aborted after transmitting a JAM signal. Remove the DMP from the chipset and try to start the transmission procedure of Figure 40 later, replacing the old time stamp $t_{DMP\_s\_g}$ by a new one.
2) Let the chipset handle the collision as usual (backoff algorithm) and discard the correctly retransmitted DMP when received at the master base station, in order to start a new DMP transmission procedure at the master base station.

In order to comply with solution 2, the receiving base station have to distinguish collided DMP frames from non collided DMP frames. In following example we will see that this is impossible.

<u>Example:</u>
Consider a 100Mbs Ethernet with one master base station and a slave base station A connected. Furthermore:
- Slave base station A is placed at a distance of 26 meters from the master base station.
- Signal propagation speed is $2.6 \cdot 10^8 m/s$
- Inter-Packet Gap $IPG = 0.96ms$ (standard, see [Quinn])
- Size of JAM pattern 32 bit (standard, see [Quinn] and [IOCJAM])
- Other (slave base)stations are quiet

Suppose slave base station A tries to transmit a packet to the master base station. Just before the head of the frame reaches the master base station (i.e. $e \to 0$), the master base station also begins transmitting a frame. A collision will take place and the channel becomes time-slotted after transmitting a JAM signal. Suppose that the master base station is programmed to transmit in the first time slot (after IPG) whereas slave base station A is programmed to transmit in the second time slot. A retransmission of the master base station will succeed now. See Figure 41.

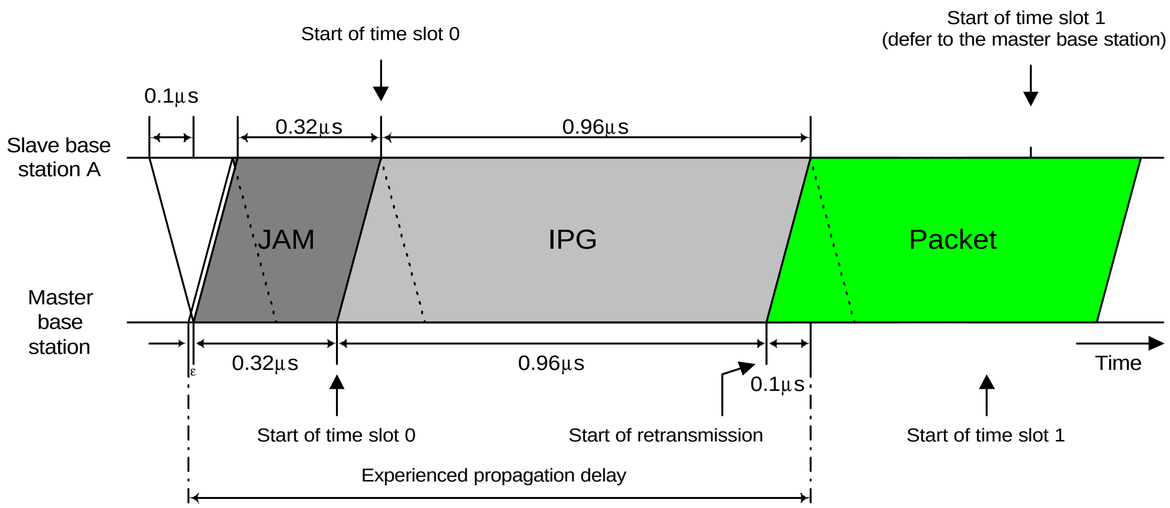


Figure 41. Effect of collisions on the propagation delay

Normally frames transmitted by the master base station to slave base station A would experience a propagation delay of:

$$PD = \frac{26}{2.6 \cdot 10^8} = 0.1ms \qquad [78]$$

In our example the transmitted frame experiences, due to the collision and the recovery from it, a propagation delay of:

$$PD = 0.1 + 0.32 + 0.96 + e \approx 1.38ms \qquad [79]$$

This is the same propagation delay experienced by a successfully transmitted frame (zero collisions) from the master base station to a slave base station placed at a distance of:

$$D = PD \cdot 2.6 \cdot 10^8 \approx 1.38 \cdot 10^{-6} \cdot 2.6 \cdot 10^8 = 359m \qquad [80]$$

This distance is allowed within the standard[i], so base stations can not distinguish a new transmission (here from 359m) from a retransmission (here from 26m). This means that solution 2 is not an option. As a result collided DMP frames should be removed from the chipset buffer and a new DMP must be generated. In order to do this, the Ethernet transmission procedures shown in Figure 29 must be modified. The modified transmission procedures are shown in Figure 42.

[i] If the worst case Path Delay Value (PDV = roundtrip time) is no more than 512 bit times, the Fast Ethernet LAN is a legal one [Quinn]

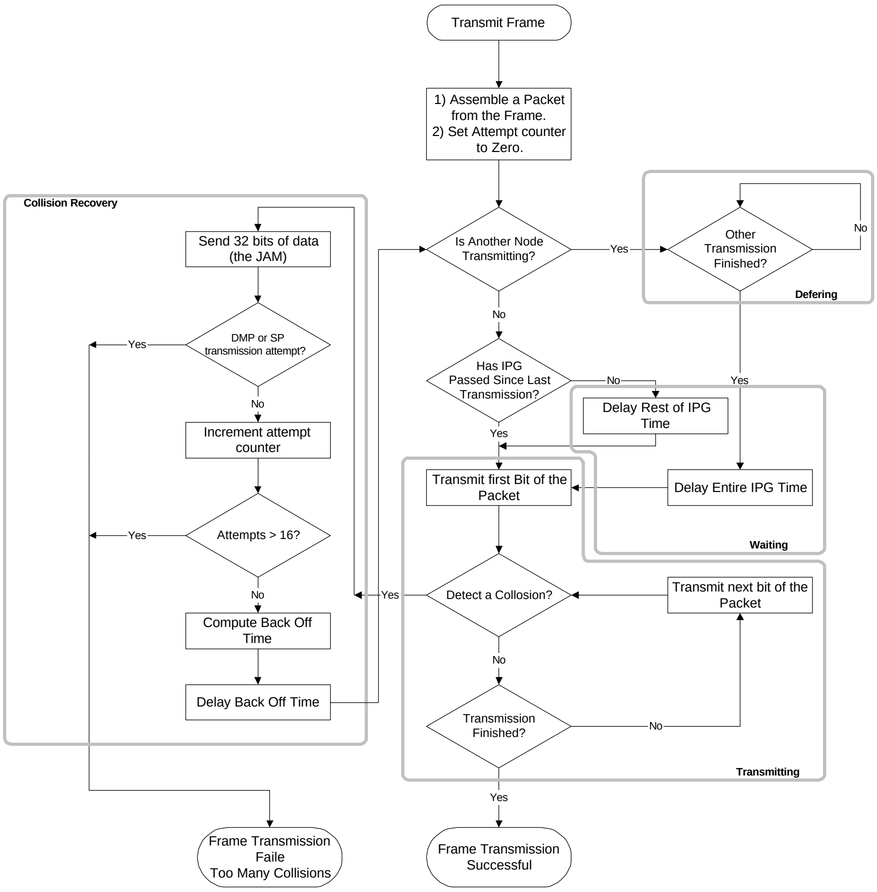


Figure 42. Schematic overview of modified Ethernet transmission procedures

5.5.3 DMP reception at slave base stations

The DMP reception procedure at the slave base station starts after the packet is uploaded and after it is determined that this packet is a DMP. The two conditions together form an additional delay of $UD + PRD1$.

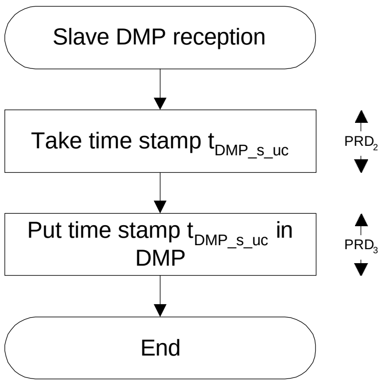


Figure 43. DMP reception by the slave base station

When entering the Slave DMP reception procedure, a time stamp $t_{DMP\_s\_uc}$ is taken and placed in the DMP. This time stamp can be used in other procedures to calculate the residence time of DMP at the slave base station.

5.5.4 DMP transmission at slave base stations

It is not necessary to send the DMP back right upon reception. It is sufficient to know how long the DMP has resided at the slave base station. This fact enables us to send the DMP back when possible (free line and empty chipset buffer). We just have to know when it is sent back in order to calculate the residence time of the DMP at the slave base station.

Just like the master base station, the slave base station must ensure that the line is idle for at least IPG before beginning with the transmission (Figure 44). The slave base station does this by beginning the DMP generation procedure after the line was idle for at least $IPG-(GD+DD+PRD2+PRD3)$. If furthermore the chipset buffer is empty, the slave base station prevents the chipset from automatic download of other packets and takes a time stamp $t_{DMP\_s\_g}$ just before the regeneration of the packet. Regeneration of the DMP means in this situation that the new taken time stamp is also placed in the DMP. The DMP can now be downloaded. After download has taken place, transmission of the DMP can begin if the line is still free. If this is not the case, the DMP is removed from the chipset buffer. Another attempt will be made later.
We see that after transmission, the DMP status is modified. This DMP status indicated that a DMP is either not yet received, received but still in house or received and sent back. This DMP status can also be used by higher level procedures to enter the Slave DMP transmission procedure if the status shows that a DMP is received but still in house.

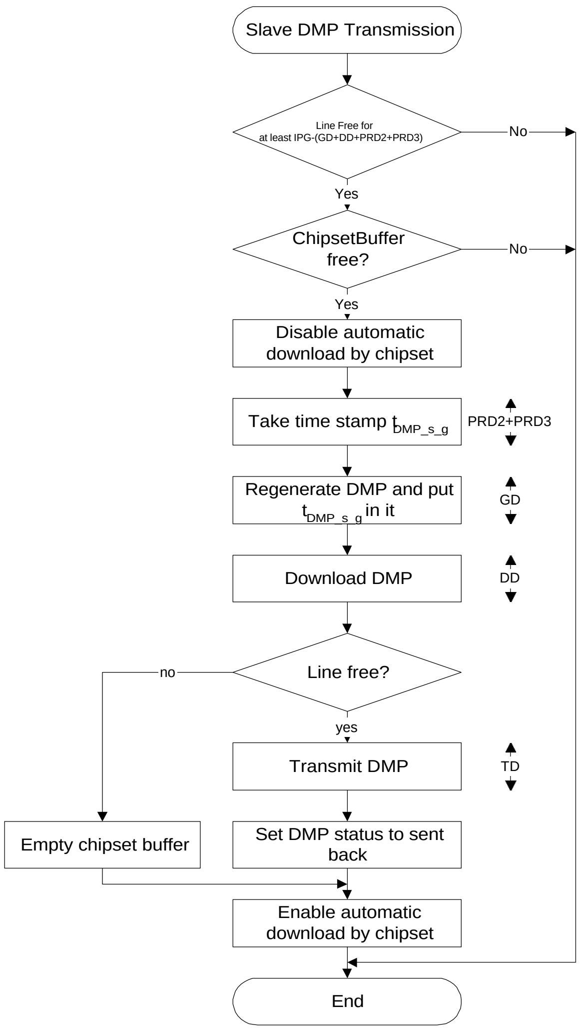


Figure 44. DMP transmission by the slave base station

Remark: The remarks made in Section 5.5.2 concerning collision recovery hold also for the transmission of DMP from the slave base station to the master base station.

5.5.5 DMP reception at the master base station

Upon arrival at the master base station, a packet is uploaded and identified. This takes an amount of time equal $UD + PRD1$. Once it is clear that the received packet is a DMP the Master DMP reception procedure is started.

In the Master DMP reception procedure, a time stamp $t_{DMP\_m\_uc}$ is taken and placed in the DMP. The DMP now contains all the required information for propagation delay calculation. Because of this, the DMP can be stored in a DMP processing buffer for later calculation of the propagation delay.

Figure 45 show the actions taken during the procedure Master DMP reception.

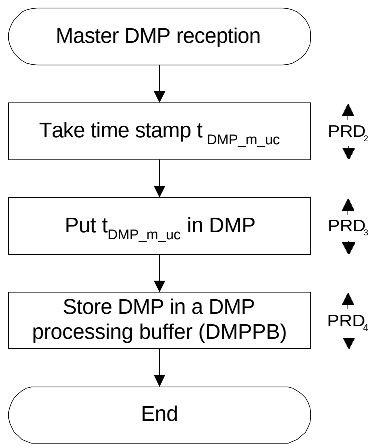


Figure 45. DMP reception by the master base station

5.5.6 Propagation delay calculation

Before presenting the propagation delay formula a timing overview of the delays experienced by the DMP will be shown in Figure 46.

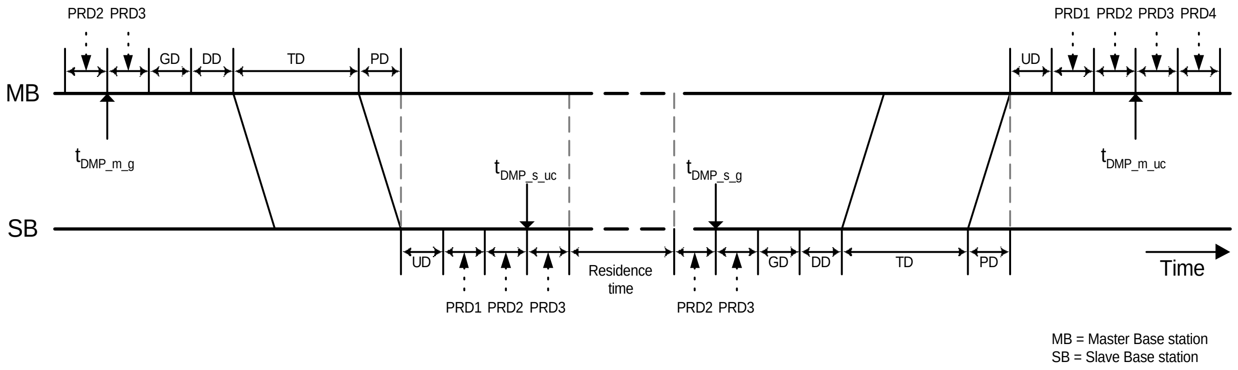


Figure 46. Timing overview of delays

Some important remarks have to be made:

- Generation delay (GD) is the time to generate a packet with already available information in addition to the time needed to place the generated packet in the download buffer. Choosing the size of the DMP as large as the size of the Synchronization Packet (SP), GD can be made equal for both DMP and SP. By taking this decision DD , UD and TD will also be equal for both DMP and SP.
- Processing delay 1 ($PRD1$) is the time needed to find out that the received packet is a DMP. By using the same recognition technique for DMP and SP, $PRD1$ can be made equal for both DMP and SP.
- Processing delay 2 ($PRD2$) is the time needed to take a time stamp (i.e. from the moment of request to the record moment).
- Processing delay 3 ($PRD3$) is the time between the record moment of the time stamp and the moment the time stamp is available in addition to the time needed to place the time stamp in the DMP.
- Processing delay 4 ($PRD4$) is the time needed to store a DMP in the DMP buffer, in order to calculate the propagation delay when resources are available.
- Residence is the time that DMP resides at the slave base station until the slave base station is allowed to transmit.

We assume that generation and processing of a DMP can not be interrupted and that access to the system bus is deterministic. As a result, we assume the mentioned delays to be deterministic.

With the given data a propagation delay can be calculated as follows:

$$PD = \frac{1}{2}\left\{t_{DMP\_m\_uc} - t_{DMP\_m\_g} - (t_{DMP\_s\_g} - t_{DMP\_s\_uc}) - k\right\} \quad [81]$$

With

$$k = (2 \cdot GD + 2 \cdot DD + 2 \cdot TD + 2 \cdot UD + 2 \cdot PRD1 + 2 \cdot PRD2 + 2 \cdot PRD3) \quad [82]$$

When using a particular type of base stations, the value of k can be measured and used as a constant. As a result, we can calculate the propagation delay if the values of the four time stamps and the value of k are available.

Once all propagation delays are determined (i.e. to every slave base station), a value of C_i (compensation time) for every slave base station can be calculated. All the base station must be informed about their value of C . In order to minimize the additional load on the Ethernet, it is conceivable that one setup packet is broadcast to all slave base stations. Such a setup packet will contain a table consisting of sets of a slave base station number and its C_i value.

5.5.7 Optimization of the calculation

Let us define a variable k_i' for each slave base station i such that:

$$k_i' = \frac{1}{2}\left\{t_{DMP\_m\_uc} - t_{DMP\_m\_g} - (t_{DMP\_s\_g} - t_{DMP\_s\_uc})\right\} \quad [83]$$

Substitution of Equation 81 in Equation 83 gives:

$$k_i' = PD_i + \frac{k}{2} = GD + DD + TD + PD_i + UD + PRD1 + PRD2 + PRD3 \quad [84]$$

In Figure 38 we saw that:

$$D_i = GD + DD + PD_i + TD + UD + PRD \quad [85]$$

Combining Equation 84 and Equation 85 gives:

$$D_i = k_i' + PRD - PRD1 - PRD2 - PRD3 = k_i' + typeconst \quad [86]$$

We assumed PRD, $PRD1$, $PRD2$ and $PRD3$ to be deterministic. Given a master base station type and a slave base station type, the values of PRD, $PRD1$, $PRD2$ and $PRD3$ are fixed. As a result, the value of $typeconst$ is also fixed for a given master base station type and slave base station type. This means that the value of $typeconst$ is known when the master base station type and the slave base station are known. For calculation of D_i, a DMP belonging to the concerned base station i in addition to the constant $typeconst$ (which is known when master RFP type and slave RFP type are known) are the only givens required.

5.6 Synchronization

Once every slave base station is informed about its compensation value, synchronization can start. In the following sections we will introduce the synchronization packet. Next we describe the transmission and reception procedures by the master base station and slave base station respectively.

5.6.1 Synchronization Packet (SP)

When the compensated synchronization time t_{cs} (see Figure 37) has come, the master base station must broadcast a synchronization packet. Such a packet must contain the following information:

- A unique pattern, by which the slave base station can recognize the received packed as an SP. This unique pattern can be a series of bytes in the payload, or a unique type in the type field (see Remark in Section 5.5.1).
- A (master base station) DECT frame number in order to let the master base station and the slave base station be frame number synchronized.

We mentioned before that the size of SP and the size of DMP will be chosen equal in order to simplify the compensation time C_i determination. A possible structure of SP is given in Figure 47.

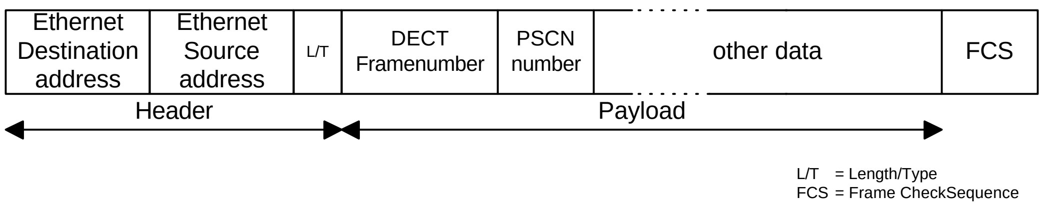


Figure 47. Possible structure of SP

5.6.2 SP transmission by the master base station

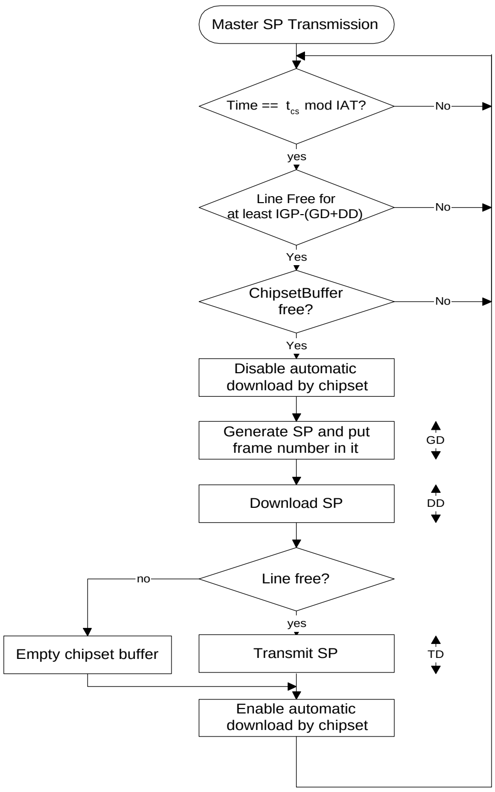


Figure 48. SP transmission by the master base station

As can be seen from Figure 48, the procedure Master SP transmission is always running. When time t_{cs} mod IAT has come[i], an SP must be prepared in order to transmit it to the slave base station.
The major difference between the SP transmission procedure and the DMP transmission procedure is that in the former no time stamp is taken.

5.6.3 SP reception by the slave base station

When using the DECT Air Interface as a synchronization medium, the slave base station compares bit numbers of frames sent by the master base station with its local bit numbers. Differences in bit times can then be noticed. Although the burst mode controller allows for timing corrections in steps of $\frac{1}{4}$ bit times, timing is in this case adjusted in steps of one or two bit times. This means $e = 0.434 m\sec$ in Figure 13 and leads to $MIST = 1.56 \sec$.

[i] Inter-Attempt Time ( ) is the deterministic time between two attempts and is equal to $IAT = 1.56 / attempts$

When using Ethernet as a synchronization medium, synchronization pulses can be applied to the burst mode controller whenever we want to. Every timing difference between the master base station and the slave base station can be noticed.

Because slave base stations can correct their timing in steps of $\frac{1}{4}$ bit time, an error of $\pm\frac{1}{8}$ bit time = $\pm 0.1085ms$ is introduced.

To reach $MIST = 1.56\sec$ a maximum jitter of $e = 0.434m\sec$ was allowed. This means that the remaining jitter available for the synchronization pulse arrivals is $e_{SP} = 0.434 - 0.1085 = 0.3255m\sec$, which means that a synchronization pulse must be fed to the burst mode controller during the interval $[t_s - 0.3255m\sec, t_s + 0.3255m\sec]$ (see Figure 37). If the size of this interval is found to as small that synchronization pulses are fed to the Burst Mode Controller outside the mentioned interval, the size of the interval can be increased by decreasing the size of $MIST$ and decreasing the Inter-Attempt Time.

The SP reception procedure at the slave base station starts after the packet is uploaded and after it is determined that this packet is an SP. The two conditions together form an additional delay of $UD + PRD1$.

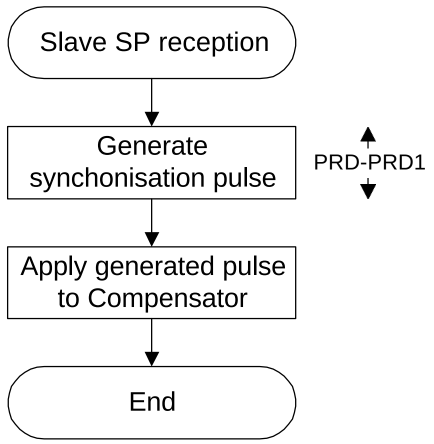


Figure 49. SP Reception by the slave base station

As shown in the flowchart of Figure 49, once the SP Slave transmission procedure is entered, a synchronization pulse is generated. This pulse is not directly fed to the burst mode controller but to a Compensator. The compensator is a hardware block that has a synchronization input and output in addition to a register that contains de desired delay time. The relation between the input and output is that the output is a delayed version of the input. We will discuss this in Section 5.7.

5.7 Hardware blocks

In this section we will present a possible implementation of the flowcharts discussed in the previous sections. We describe the hardware implementation using a block diagram of the hardware.

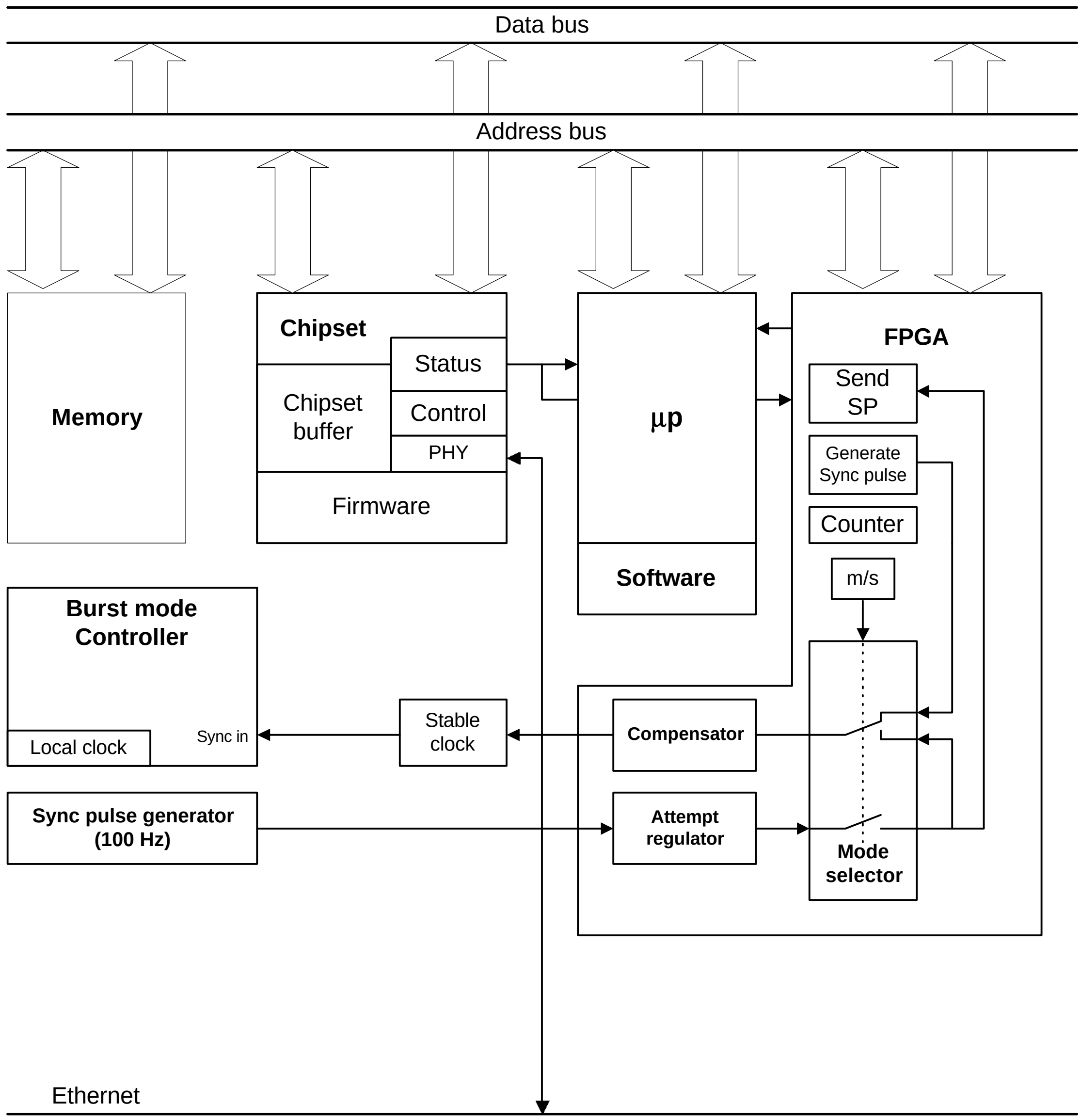


Figure 50. Block diagram of the hardware

In Figure 50 a number of blocks are shown, we will name them and explain them one by one.

Memory

This part performs the following functions:

- Storage of the uploaded packets
- Storage of the packets that must be downloaded
- Storage of the returned DMP's that must be processed (Slave mode)
- Storage of the id's of the connected slave base station in combination with their compensation value or their DMP status[i] (Master mode).

[i] If a C value is not yet calculated, C is a status that shows that a DMP is still not sent, sent but not back or back but still not processed.

- Storage of the framework (Source address, Length/Type, recognition pattern etc…) of an SP and a DMP. The idea here is that the FPGA puts the correct data in these frames (time stamps, destination base station address, frame numbers etc…) and orders the chipset to download these packets (SP or DMP). After successful download, the microprocessor refreshes these frameworks.
  The reason for doing this, is to keep the generation time of an SP or a DMP as constant as possible.

Ethernet chipset

This is the chipset that handles the Ethernet protocol. The chipset block contains the following parts:

- "Firmware" that contains the Ethernet procedures as mentioned in Figure 29. The "firmware" is the part we have to modify to comply with the modified Ethernet procedures of Figure 42.
- Chipset buffer where the packets reside before they are transmitted or uploaded (after download or reception).
- Status register which contains information about the status of the chipset. It gives us information about the line (free or busy), chipset buffer (empty or not), collision detected, transmission succeeded, transmission failed, transmission completed, reception completed and many others. Some of the mentioned indicators can generate an interrupt if desired. Such an interrupt can be fed to the FPGA, which generates another interrupt for the microprocessor after it finished processing of the uploaded packet.
- The control part is accessed via the Data bus. Via this control part we command the chipset (e.g. Enable/Disable automatic download, empty chipset buffer etc…).
- PHY is the interface part between the Ethernet and the rest of the chipset.

Microprocessor μp

The microprocessor performs the following functions:

- Interface between the DECT part and the Ethernet part (not depicted in Figure 50)
- Calculation of the C_i values mentioned in Figure 38 (with the modifications of Section 5.5.7)
- Processing of the uploaded packets after they are inspected and eventually processed by the FPGA.
- Setting the Compensator value in the FPGA.

FPGA

FPGA stands for Field Programmable Gate Array. An FPGA can be used to integrate logical functions on a single chip. In our case it can be used to perform the following functions:

- Identification of the uploaded packet. In order to perform this function, the FPGA must be able to perform[i] DMA functions. Whenever a packet is received, the chipset uploads this packet to a predefined place in the memory and informs the FPGA via an interrupt. The FPGA on its turn identifies the uploaded packet.
- Generation of a synchronization pulse if the identified packet is an SP.
- Take and place a time stamp in the uploaded packet if the uploaded packet is a DMP.
- Because the time stamps in the DMP need not to be of an absolute type, we build a clock cycle (e.g. 25MHz) counter within the FPGA. The value of this counter can be used as a time stamp. Of course, the range of the counter must be large enough due to the roundtrip delay of the DMP.
- Generate an interrupt to the microprocessor if the uploaded packet is neither a DMP nor an SP or when finished handling the uploaded DMP or SP.
- When the synchronization pulse generator applies a synchronization pulse to the FPGA, the FPGA must perform the checks included in the procedure of Figure 48 (i.e. line free etc.).
- When allowed, the FPGA must order the chipset to download an SP (always available in memory). This function is only valid in master mode.

[i] Time critical DMA functions will be handled by the FPGA

- Orders (if allowed) the chipset to download the DMP to transmit it to a specific slave base station after the FPGA takes a time stamp and places it in the DMP (master mode).
- Requests the microprocessor to place the arrived DMP in a buffer where the DMP resides before it is sent back. This request is made after a time stamp is taken and placed in the DMP (slave mode).
- Orders (if allowed) the chipset to download the DMP to transmit it back to the master base station after the FPGA takes a time stamp and places it in the DMP (slave mode).
- Requests the microprocessor to place the returned DMP in a processing buffer after the FPGA takes a time stamp and places it in the DMP (master mode).
- When the base station is in master mode, the synchronization pulse generated by the Sync pulse generator is fed to the stable clock (via the Compensator) and the FPGA which also takes care of the generation of an SP.
  If the base station is in slave mode, the synchronization pulse generated by the Sync pulse generator is not fed to the SP sending entity. Furthermore, the synchronization pulse from the sync pulse generation entity in the FPGA, is fed to the stable clock (via the Compensator).
- The Compensator is a part that delays the incoming synchronization pulse for the appropriate compensation value. In slave mode, the compensation value is equal to the calculated C value. However in master mode the compensation value is equal to the maximum D_i value (C_m, see Figure 37). The reason for this is that the master base station must correct for the maximum delay to the slave base stations.
  The compensation value is asserted internally by the FPGA after it is calculated by the microprocessor.
- The attempt regulator causes the number of attempt to be equal to the desired value (maximum of 100 attempts per second).

Sync pulse generator

This part generates a synchronization pulse once every DECT frame (10 ms). This pulse is applied to the FPGA, which allows one synchronization pulse to pass every $IAT \cdot frametime = 7 \cdot 0.01 = 70m\sec$. This means that every $MIST$ (1.56sec), at least the number attempts determined in Equation 76, are made. Note that the synchronization pulse is generated t_a (see Figure 37) before the real synchronization time.

Stable clock

This is the clock that synchronizes the burst mode controller. Every DECT frame (10 ms) a synchronization pulse is applied to the burst mode controller. The stable clock is synchronized via the FPGA (see also Figure 14) and has an accuracy of at least 10^{-6} (by using virtual correction of the clock).

Burst mode controller

Burst mode controller is an integrated circuit that performs DECT (MAC)functions and timing functions. It is also the part that shortens or lengthens frames in order to get synchronized with the presented synchronization pulse. The burst mode controller contains a local clock which is corrected by the stable clock (see also Figure 14).

6 Conclusions and recommendations

6.1 Conclusions

The aim of this report was to investigate possible new advanced mechanisms for DECT base station synchronization. Some possible solutions where:

- Synchronization via GPS
- Synchronization via the DECT Air Interface
- Synchronization via Ethernet

The first option is not evaluated in this report. Price inquiries have shown that this eventual solution is not a good option for us. GPS systems are far more expensive than a base station and lead to a price overhead of more than 100%.
The second option is evaluated in a non-fading environment and is found to be a good option. However, further evaluation in a fading environment has shown that we need to increase the number of base station with at least 70% in order to make synchronization via the DECT Air Interface possible. So synchronization via the DECT Air Interface was also not feasible.
The third option is found to be a good option after it was evaluated. Simplified performance analysis of the system has shown that synchronization over Ethernet has small influence on the available Ethernet capacity. The major problem here is that processes and procedures concerning the synchronization mechanism must be deterministic in time. The latter fact implies that software modification of Ethernet drivers is required and that some hardware must be added in order to avoid large stochastic delay introduced when only software processes synchronization packets.
This report describes the requirements on these software modifications and hardware additions.

6.2 Recommendations

Detailed design of hardware and software is still needed. This should be done in conjunction with (new) Ethernet base station design.

In the Ethernet performance evaluation section, a lot of simplifications had to be made. To get a better estimate of the Synchronization Loss Ratio, simulations are necessary. In such a simulation the real Ethernet mechanism must be implemented and traffic must be closer to real traffic (e.g. self similar). By adding a master base station and slave base stations which synchronize conform the procedures of Section 5.6, the Synchronization Loss Ratio can be measured.

Furthermore, it is recommended to check the nature of the processing times of the different components that take part in the synchronization mechanism. A good working synchronization mechanism requires all components to have deterministic processing time (see Section 5.5.1 and Section 5.6).

Abbreviations

AER	A-field Error Rate
ARQ	Automatic Repeat request
BER	Bit Error Rate
BMC	Burst Mode Controller
CC	Call Control
CCFP	Common Control Fixed Part
CD	Collision Detect
CDCS	Continuous Dynamic Channel Selection
CEPT	Conference of European Posts and Telecommunication
CI	DECT Common Interface
CLMS	Connectionless Message Service
CMC	Connectionless Message Control
COMS	Connection-Oriented Message Service
CRC	Cyclic Redundancy Check
DD	Download Delay
DECT	Digital Enhanced Cordless Telecommunication
DLC	Data Link Control layer
DMP	Delay Measurement Packet
DMPPB	Delay Measurement Packet Processing Buffer
ES	End System
ETSI	European Telecommunications Standard Institute
FCS	Frame Check Sequence
FP	Fixed Part
FPGA	Field Programmable Gate Array
GAP	Generic Access Profile
GD	Generation Delay
GFSK	Gaussian Filtered frequency Shift Keying
GPS	Global Positioning System
GSM	Global System for Mobile communication
IAF	Inter-Attempt DECT Frames
IAT	Inter-Attempt Time
IPG	Inter-Packet Gap
ISDN	Integrated Services Digital Network
IWU	Interworking Unit
LAN	Local Area Network
LCE	Link Control Entity
LLME	Lower Layer Management Entity
MAC	Medium Access Control Layer
MFN	Multiframe Number
MIST	Maximum Inter-Synchronization pulse Time
MM	Mobility Management
NIC	Network Interface Card
NWK	Network layer
OSI	Open System Interconnection
PBX	Private Branch exchange
PD	Propagation Delay
PP	Portable Part
PRD	Processing Delay
PSCN	Primary Scan Carrier Number
PSTN	Public Switched Telephone Network
PT	Portable Radio Termination
RF	Radio Frequency
RFP	Radio Fixed Part
SER	S-field Error Rate
SLR	Synchronization Loss Ratio

SNR	Signal to Noise Ratio
SP	Synchronization Packet
SS	Supplementary Services
TD	Transmission Delay
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
UD	Upload Delay

References

[3COM] "3C90x and 3C90xB NICs Technical Reference", 3COM, August 1998

[Aker2] D. Åkerberg, "DECT Explained", PHS World Congress '96, Singapore, November 1996

[BCSHA] S. Bink and G. van Lieshout, "Introduction to Business Cordless System Hardware Architecture", Internal information, August 1995

[C2Sync] E. Schreuder, "RFP Synhronization", Internal information report, January 1998

[Cheng] D.V. Cheng, "Field and Wave Electromagnetics", Addison-Wesley, 1989

[Daalen] R.P.W. van Daalen, "Synchronization of Base-stations in Cordless Telephone Systems", Master's thesis, Enschede, the Netherlands, January 1993

[DECTCD] "DECT, The cordless Revolution, Interactive training", CDROM, Ericsson Business Mobile Networks, 1997

[EMN94] "Digital Cordless Telephony and the Personal Communications revolution", Ericsson Business Mobile Networks, Press background information, March 1994

[ERICSSON1] "BER behavior of a typical DECT link", RES-03/Data, Ericsson, Italy 1996

[ETS 300 175-1] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 1: Overview"

[ETS 300 175-2] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 2: Physical layer"

[ETS 300 175-3] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 3: Medium Access Control layer"

[ETS 300 175-4] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 4: Data Link Control layer"

[ETS 300 175-5] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 5: Network layer"

[ETS 300 175-6] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 6: Identities and addressing"

[ETS 300 175-7] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 7: Security features"

[ETS 300 175-8] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 8: Speech coding and transmission"

[ETS 300 175-9] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT) Common Interface Part 9: Public Access Profile"

[ETS 300 444] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Generic Access Profile"

[ETR 185] "Radio Equipment and Systems (RES); Digital European Cordless Telecommunications (DECT); Data Service Profile (DSP); Profiles overview"

[Jain] R. Jain, "The art of computer system performance analysis", Wiley, 1991

[KEMO] J.M. Keenan and A.J. Motley, "Radio coverage in buildings", Br Telecom Technol J, Vol 8 No1, January 1990

[O'Reilly] J.L. Hammond and P.J.P. O'Reilly, "Performance analysis of local computer networks", Addison Wesley, 1988

[IOCETYPE] "Ethertypes", "Optimized Engineering Corporation", http://www.optimized.com/COMPENDI/REF-Etyp.htm

[IOCGAP] "Interframe gap", "Optimized Engineering Corporation", http://www.optimized.com/COMPENDI/EN-Ifgap.htm

[IOCJAM] "Runts and collisions", "Optimized Engineering Corporation", http://www.optimized.com/COMPENDI/EN-Colli.htm

[PCClock1]" "The clock synchronization package", Intecs Sistemi S.p.A., http://www.pisa.intecs.it/products/ClockSynchronisation/Overview.html

[PCClock2] "Network time server", TrueTime, http://www.truetime.com/DOCS/PDTS/ntnts_100.html

[PCClock3] "Computer clock synchronization techniques", TrueTime, http://www.truetime.com/DOCS/PDTS/ntCC_sync.html

[Ploeg] H.J. van der Ploeg, "DRA1900 Signaling Performance Analysis", TWAIO Final Report, November 1997

[Quinn] Liam B. Quinn and Richard G. Russell, "Fast Ethernet", Wiley Computer Publishing, pp 71-76

[Rapp] T.S. Rappaport, "Wireless Communications, Principles and Practice", Prentice Hall, 1996, pp 325-335

[Rayleigh1] F. Brouwer, "Elementary model for bit and synchronization quality in a Rayleigh fading channel with 2-branch diversity", January 1997

[RTL8019] "RTL 8019 Realtek Full-Duplex Ethernet Controller with Plug and Play Function (RealPNP)", Advanced information, Realtek Semi-conductor Co. LTD., April 1995

[Tanenbaum] Andrew S. Tanenbaum, "Computer Networks", Prentice Hall, pp 280-285

[TBR 006] "Digital Enhanced Cordless Telecommunications (DECT): General terminal attachment requirements", European Telecommunications Standards Institute, 1998

[VP23030] "VP23030 ADPCM Interface; Application Note", VLSI Technology, January 1996

[Walrand] Jean Walrand, "Communication Networks", Aksen Associates, pp 214-219

Appendix A. *SLR* due to incorrect S-field reception (Program listing)

Main.cpp (*SLR* due to incorrect S-field reception)

Description: Calculates the mean *SLR* by simulating the S-field error for DECT frames in a fading environment

```
#include <iostream.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "complex.hpp"
#include "statistic.hpp"
#include "probability.hpp"

int main(void)
{
        FILE *of;                     // outputfile

        probability prob;             // to generate a erf(x) value
        statistic stat;               // to provide statistical operations

        complex rs;                   // Rayleigh fading

        long double dBpower;          // Power in dB to be used for BER calculation
        long double power;            // Power used for BER calculation
        long double BER;              // Bit error rate

        long double D;                // Distance between two consecutive samples
        int n;                        // Number of rays in fading generation
        const F = 1.89e9;             // Dect frequency
        const c = 3e8;                // Speed of light
        const PI = 4*atan(1);
        const k=2*PI*F/c;
        int N;                        // Number of frames within MIST

        long double *fd;              // n angles
        long double *fi;              // n phases

        long double *SER;             // Synchronisation Error Ratio (for a frame)
        long double *SLR;             // Synchronisation Loss Ratio
        long double *SLR_SNR;         // SLR as a function of SNR

        long double *conf;            // for confidence interval calculation

        int SLRs;                     // Number of SLR simulations per SNR
        int snrs;                     // Number of SNR simulations

        long double minSNR;           // Minimum SNR to simulate
        long double maxSNR;           // Maximum SNR to simulate

        long double snrss;            // SNR step size

        long double realSNR;          // for output file
        int i,j;                      // counters


        system("clear");
        cout << "Enter the number of Rayleigh rays: ";
        cin >> n;
        cout << "Enter the distance between two consecutive samples (0.001 to 0.01): ";
        cin >> D;
        cout << "Enter the number of frames within MIST: ";
        cin >> N;
        cout << "\nEnter the number of SLR simulations per SNR: ";
        cin >> SLRs;
        cout << "Enter the number of SNR simulations ";
        cin >> snrs;
        cout << "\nEnter the minimum SNR to simulate: ";
        cin >> minSNR;
        cout << "Enter the maximum SNR to simulate: ";
```

```
cin >> maxSNR;

snrss=(maxSNR-minSNR)/snrs;

if((fd=(long double *)malloc(n*sizeof(long double)))==NULL)          // allocate memory for n fd's
{
        printf("allocation failed\n");
        exit(0);
}

if((fi=(long double *)malloc(n*sizeof(long double)))==NULL)          // allocate memory for n fi's
{
        printf("allocation failed\n");
        exit(0);
}

if((SER=(long double *)malloc(N*sizeof(long double)))==NULL)         // allocate memory for N SER's
{
        printf("allocation failed\n");
        exit(0);
}

if((SLR=(long double *)malloc(SLRs*sizeof(long double)))==NULL)      // allocate memory for SLRs SLR's
{
        printf("allocation failed\n");
        exit(0);
}

if((SLR_SNR=(long double *)malloc(snrs*sizeof(long double)))==NULL) // allocate memory for snrs SLR(SNR)
{
        printf("allocation failed\n");
        exit(0);
}

if((conf=(long double *)malloc(snrs*sizeof(long double)))==NULL)     // allocate memory for snrs conf
{
        printf("allocation failed\n");
        exit(0);
}

int SLRsi=0;

for (int snri=0;snri<snrs;snri++)
{
        for (SLRsi=0;SLRsi<SLRs;SLRsi++)
        {
                for (i=0;i<n;i++)
                {
                        fd[i]=cos(2*PI*rand()/RAND_MAX);
                        fi[i]=2*PI*rand()/RAND_MAX;
                }
                for (i=0;i<N;i++)
                {
                        rs = complex(0,0);
                        for (j=0;j<n;j++)
                        {
                                rs = rs + cexp(complex(0,fi[j]+k*fd[j]*i*D));
                        }
                        dBpower = (minSNR+snri*snrss)+20*log10(abs(rs)/sqrt(n));
                        power = pow(10,dBpower/10);
                        BER = 0.5*(1-erf(sqrt(0.34*power)));
                        if (BER<1e-6) {BER = 1e-6;}
                        SER[i]=1-pow((1-BER),16)-16*BER*pow((1-BER),15);
                }
                SLR[SLRsi]=stat.product(SER,N);
        }
        SLR_SNR[snri]=stat.mean(SLR,SLRs);
        conf[snri]=stat.confidence(0.05,SLR,SLRs);
```

```
		if( (of = fopen("progress.dat","w")) == NULL )
		{
			perror("fopen failes");
			exit(1);
		}
		realSNR = snri*snrss+minSNR;
		fprintf(of,"SNR = %Le gives SLR = %Le and Confidence interval = %Le\n",realSNR,SLR_SNR[snri],conf[snri]);

		fclose(of);

		printf("SNR stapnummer: %d    SLR_SNR[%Le]=%Le\n",snri,(snri*snrss)+minSNR,SLR_SNR[snri]);
	}



	if( (of = fopen("output.dat","w")) == NULL )
	{
		perror("fopen failes");
		exit(1);
	}

	fprintf(of,"n=%d D=%Le\n",n,D);
	fprintf(of,"%d sets of %d frames per SNR value\n",SLRs,N);
	fprintf(of,"SNR from %LedB to %LedB in steps of %LedB\n\n",minSNR,maxSNR,snrss);

	for (i=0;i<snrs;i++)
	{
		realSNR = i*snrss+minSNR;
		fprintf(of,"SNR = %Le gives SLR = %Le and Confidence interval = %Le\n",realSNR,SLR_SNR[i],conf[i]);
	}

	fclose(of);

return 0;
}
```

Appendix B. *SLR* due to incorrect A-field reception (Program listing)

Main.cpp (*SLR* due to incorrect A-field reception)

Description: Calculates the mean *SLR* by simulating the A-field error for DECT frames in a fading environment

```
#include <iostream.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "complex.hpp"
#include "statistic.hpp"
#include "probability.hpp"

int main(void)
{
        FILE *of;                       // outputfile

        probability prob;               // to generate a erf(x) value
        statistic stat;                 // to provide statistical operations

        complex rs;                     // Rayleigh fading

        long double dBpower;            // Power in dB to be used for BER calculation
        long double power;              // Power used for BER calculation
        long double BER;                // Bit error rate

        long double D;                  // Distance between two consecutive samples
        int n;                          // Number of rays in fading generation
        const F = 1.89e9;               // Dect frequency
        const c = 3e8;                  // Speed of light
        const PI = 4*atan(1);
        const k=2*PI*F/c;
        int N;                          // Number of frames within MIST

        long double *fd;                // n angles
        long double *fi;                // n phases

        long double *AER;               // Synchronisation Error Ratio (for a frame)
        long double *SLR;               // Synchronisation Loss Ratio
        long double *SLR_SNR;           // SLR as a function of SNR

        long double *conf;              // for confidence interval calculation

        int SLRs;                       // Number of SLR simulations per SNR
        int snrs;                       // Number of SNR simulations

        long double minSNR;             // Minimum SNR to simulate
        long double maxSNR;             // Maximum SNR to simulate

        long double snrss;              // SNR step size

        long double realSNR;            // for output file
        int i,j;                        // counters


        system("clear");
        cout << "Enter the number of Rayleigh rays: ";
        cin >> n;
        cout << "Enter the distance between two consecutive samples (0.001 to 0.01): ";
        cin >> D;
        cout << "Enter the maximum number of frames between to succesfull A-field receptions (consider 90% rule): ";
        cin >> N;
        cout << "\nEnter the number of SLR simulations per SNR: ";
        cin >> SLRs;
        cout << "Enter the number of SNR simulations ";
        cin >> snrs;
        cout << "\nEnter the minimum SNR to simulate: ";
        cin >> minSNR;
        cout << "Enter the maximum SNR to simulate: ";
```

```
cin >> maxSNR;

snrss=(maxSNR-minSNR)/snrs;

if((fd=(long double *)malloc(n*sizeof(long double)))==NULL)          // allocate memory for n fd's
{
        printf("allocation failed\n");
        exit(0);
}

if((fi=(long double *)malloc(n*sizeof(long double)))==NULL)          // allocate memory for n fi's
{
        printf("allocation failed\n");
        exit(0);
}

if((AER=(long double *)malloc(N*sizeof(long double)))==NULL)         // allocate memory for N SER's
{
        printf("allocation failed\n");
        exit(0);
}

if((SLR=(long double *)malloc(SLRs*sizeof(long double)))==NULL)      // allocate memory for SLRs SLR's
{
        printf("allocation failed\n");
        exit(0);
}

if((SLR_SNR=(long double *)malloc(snrs*sizeof(long double)))==NULL) // allocate memory for snrs SLR(SNR)
{
        printf("allocation failed\n");
        exit(0);
}

if((conf=(long double *)malloc(snrs*sizeof(long double)))==NULL)     // allocate memory for snrs SLR(SNR)
{
        printf("allocation failed\n");
        exit(0);
}

for (int snri=0;snri<snrs;snri++)
{
        for (int SLRsi=0;SLRsi<SLRs;SLRsi++)
        {
                for (i=0;i<n;i++)
                {
                        fd[i]=cos(2*PI*rand()/RAND_MAX);
                        fi[i]=2*PI*rand()/RAND_MAX;
                }
                for (i=0;i<N;i++)
                {
                        rs = complex(0,0);
                        for (j=0;j<n;j++)
                        {
                                rs = rs + cexp(complex(0,fi[j]+k*fd[j]*i*D));
                        }
                        dBpower = (minSNR+snri*snrss)+20*log10(abs(rs)/sqrt(n));
                        power = pow(10,dBpower/10);
                        BER = 0.5*(1-erf(sqrt(0.34*power)));
                        if (BER<1e-6) {BER = 1e-6;}
                        AER[i]=1-pow((1-BER),64);

                }
                SLR[SLRsi]=stat.product(AER,N);
```

```
                if (((float)SLRsi/1000)==prob.round(SLRsi/1000))
                {
                        if( (of = fopen("progress.dat","w")) == NULL )
                        {
                                perror("fopen failes");
                                exit(1);
                        }

                        fprintf(of,"SLRsi=%d  snri=%d\n",SLRsi,snri);

                        fclose(of);
                }

        }

        SLR_SNR[snri]=stat.mean(SLR,SLRs);
        conf[snri]=stat.confidence(0.05,SLR,SLRs);
    }

    if( (of = fopen("output.dat","w")) == NULL )
    {
        perror("fopen failes");
        exit(1);
    }

    fprintf(of,"n=%d  D=%Le\n",n,D);
    fprintf(of,"%d sets of %d frames per SNR\n",SLRs,N);
    fprintf(of,"SNR from %LedB to %LedB in steps of %LedB\n\n",minSNR,maxSNR,snrss);

    for (i=0;i<snrs;i++)
    {
        realSNR = i*snrss+minSNR;
        fprintf(of,"SNR = %LedB gives SLR = %Le and Confidence interval = %Le\n",realSNR,SLR_SNR[i],conf[i]);
    }

    fclose(of);

return 0;
}
```

Appendix C. Auxiliary procedures (Program listing)

Complex.hpp

Description: Performs complex operation procedures

```
// complex.hpp

#ifndef REKENWONDERCLASS
#define REKENWONDERCLASS

class complex
{
	long double real;
	long double imag;

public:
	friend complex sum(complex x, complex y);
	friend void show(complex x);
	friend complex cexp(complex x);
	friend long double abs(complex x);
	friend complex conj(complex x);
	friend complex operator+(complex x, complex y);
	friend complex operator-(complex x, complex y);
	friend complex operator*(complex x, complex y);
	friend complex operator/(complex x, complex y);

	complex(){}
	complex(long double a,long double b) {real = a; imag = b;}
};
#endif
```

Complex.cpp

```
// complex.cpp
#include <stdio.h>
#include <math.h>
#include "complex.hpp"

complex sum(complex x, complex y)
{
	complex z;
	z.real = x.real +y.real;
	z.imag = x.imag+y.imag;
	return(z);
}

void show(complex x)
{
	printf(" %f + i %f\n", x.real, x.imag);
}

complex cexp(complex x)
{
	complex z;
	z.real = exp(x.real)*cos(x.imag);
	z.imag = exp(x.real)*sin(x.imag);
	return(z);
}

long double abs(complex x)
{
	return (sqrt(x.real*x.real+x.imag*x.imag));
}


complex conj(complex x)
{
	complex z;
	z.real = x.real;
	z.imag = -x.imag;
```

```
		return(z);
}

complex operator+(complex x, complex y)
{
		complex z;
		z.real = x.real + y.real;
		z.imag = x.imag + y.imag;
		return(z);
}


complex operator-(complex x, complex y)
{
		complex z;
		z.real = x.real - y.real;
		z.imag = x.imag - y.imag;
		return(z);
}


complex operator*(complex x, complex y)
{
		complex z;
		z.real = x.real*y.real-x.imag*y.imag;
		z.imag = x.real*y.imag+x.imag*y.real;
		return(z);
}


complex operator/(complex x, complex y)
{
		complex z;
		long double abs2;

		z=(x*conj(y));
		abs2=abs(y)*abs(y);
		z.real=z.real/abs2;
		z.imag=z.imag/abs2;

		return(z);
}
```

Statistic.hpp

Description: For determination of the mean, sum, product, standard deviation and confidence interval of an array of numbers.

```
// statistic.hpp


#ifndef STATISTICCLASS
#define STATISTICCLASS

class statistic
{

public:
		long double sum(long double *p,int n);
		long double mean(long double *p,int n);
		long double product(long double *p,int n);
		long double stdev(long double *p,int n);
		long double confidence(float alpha,long double *p,int n);
		statistic(){}
};
#endif
```

Statistic.cpp

```
// statistic.cpp

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "statistic.hpp"

long double statistic::sum(long double *p,int n)
{
	long double z=0;
	for (int i=0;i<n;i++)
	{
		z=z+p[i];
	}
	return(z);
}

long double statistic::mean(long double *p,int n)
{
	return ((sum(p,n))/n);
}

long double statistic::product(long double *p,int n)
{
	long double prod=1;
	for (int i=0;i<n;i++)
	{
		prod = prod * p[i];
	}
	return (prod);
}

long double statistic::stdev(long double *p, int n)
{
	long double *p2;
	long double average;
	long double sd;

	if((p2=(long double *)malloc(n*sizeof(long double)))==NULL)	// allocate memory for E[(x-u)^2]
	{
		printf("allocation failed\n");
		exit(0);
	}

	average = mean(p,n);

	for (int i=0;i<n;i++)
	{
		p2[i]=pow((p[i]-average),2);
	}

	sd = sqrt(mean(p2,n));

	return(sd);
}

long double statistic::confidence(float alpha, long double *p, int n)
{
	long double conf;
	float q;

	q = 1-alpha/2;
	conf = 4.91*(pow(q,0.14)-pow((1-q),0.14))*stdev(p,n)/sqrt(n);

	return(conf);
}
```

Probability.hpp

Description: Determination of erf(x) via a lookup table.

```
// probability.hpp

#ifndef PROBABILITYCLASS
#define PROBABILITYCLASS

class probability
{


public:

		FILE *f;
		long double p[50001];

		probability();
		int round(long double x);
		long double erf(long double x);
};
#endif
```

Probability.cpp

```
// probability.cpp
#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "probability.hpp"

probability::probability()
{
		if( (f = fopen("erfvalues.dat","r")) == NULL )
		{
				perror("fopen failed");
				exit(1);
		}

		for(int i=0;i<50001;i++)
		{
				fscanf(f,"%lf\n",&p[i]);
		}
		fclose(f);
}

int probability::round(long double x)
{
		if ((x-floor(x))<0.5)
		{
				return (floor(x));
		}
		else
		{
				return (ceil(x));
		}
}


long double probability::erf(long double x)
{

		long double result;
		long double slope;
		long double step;
		long double ceilstep;
		long double floorstep;
		long double stepsize=0.0001;
```

```
	if (x<0)
	{
		x=0;
	}
	if(x>5)
	{
		x=5;
	}


	floorstep = floor(x/stepsize);
	ceilstep  = ceil(x/stepsize);
	step      = (long double)(x/stepsize);
	slope = p[(int)ceilstep]-p[(int)floorstep];
	result = (slope*(step-floorstep))+p[(int)floorstep];
	return (result);
}
```

Appendix D. Rayleigh fading simulation program (Mathcad)

Distance between two samples

$D := 0.01$

Number of Rayleigh sources

$n := 6$

Simulation time (x10ms)

$N := 100($

DECT carrier frequency

$F := 1.89\,10^{9}$

Speed of light

$c := 3 \cdot 10^{8}$

$k := 2 \cdot \pi \cdot \frac{F}{c}$

$k = 39.584$

Determination of n random phases and n random angles

$$fd := \left| \begin{array}{l} fdh \leftarrow 0 \\ \text{for } i \in 0..n-1 \\ \quad fdh_i \leftarrow \cos(2 \cdot \pi \cdot \mathrm{rnd}(1)) \\ fdh \end{array} \right.$$

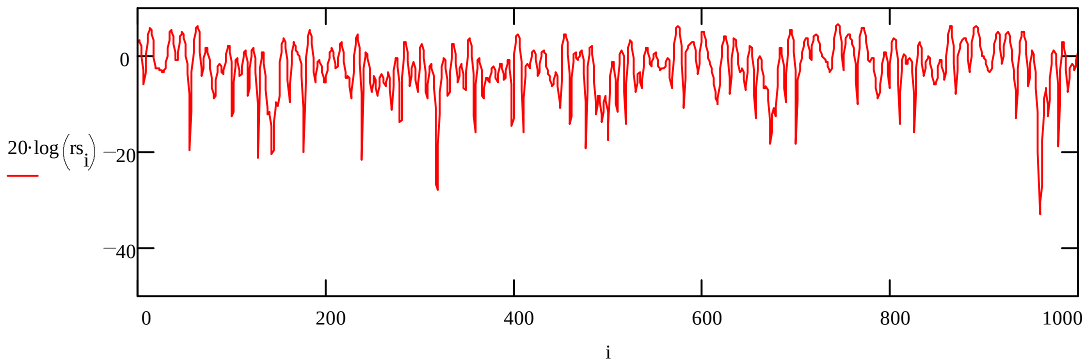
$$\phi := \left| \begin{array}{l} \phi h \leftarrow 0 \\ \text{for } i \in 0..n-1 \\ \quad \phi h_i \leftarrow 2 \cdot \pi \cdot \mathrm{rnd}(1) \\ \phi h \end{array} \right.$$

Determination of fading for every frame (10ms)

$$rs := \left| \begin{array}{l} rsh \leftarrow 0 \\ \text{for } i \in 0..N-1 \\ \quad \left| \begin{array}{l} rsh_i \leftarrow 0 \\ \text{for } j \in 0..n-1 \\ \quad rsh_i \leftarrow rsh_i + e^{-i \cdot \left(\phi_j + k \cdot fd_j \cdot i \cdot D\right)} \\ rsh_i \leftarrow \frac{\left|rsh_i\right|}{\sqrt{n}} \end{array} \right. \\ rsh \end{array} \right.$$

Simulation range

$i := 0..N - 1$



Appendix E. Link budget calculation in Maple V

```
> restart;
>
> Grfp:=2;

                         Grfp := 2

> Gpp:=0;

                          Gpp := 0

> Ppp:=19;

                         Ppp := 19

> Spp:=-86;

                         Spp := -86

> LB1:=(Prfp,Srfp)->Prfp-Srfp+2*Grfp;

          LB1 := (Prfp, Srfp) -> Prfp - Srfp + 2 Grfp

> LB3:=(Prfp,Srfp)->Prfp-Spp+Grfp+Gpp;

          LB3 := (Prfp, Srfp) -> Prfp - Spp + Grfp + Gpp

> LB4:=(Prfp,Srfp)->Ppp-Srfp+Grfp+Gpp;

          LB4 := (Prfp, Srfp) -> Ppp - Srfp + Grfp + Gpp

> gain:=(Prfp,Srfp)->LB1(Prfp,Srfp)-min(LB3(Prfp,Srfp),LB4(Prfp,Srfp));

  gain := (Prfp, Srfp) -> LB1(Prfp, Srfp) - min(LB3(Prfp, Srfp), LB4(Prfp, Srfp))

> plot3d(gain(Prfp,Srfp),Prfp=19..24,Srfp=-90..-86,labels=["RFP Power (dBm)","RFP Sensitivity
(dBm)","Gain"]);
```