Analysis and Enhancement of Safety-Critical Communication for Railway Systems

Morten Lisborg Jørgensen

AALBORG UNIVERSITET

Department of Electronic Systems
Fredrik Bajers Vej 7
9220 Aalborg Ø
Telephone: +45 96 35 86 00
http://es.aau.dk/

Title:

Analysis and Enhancement of Safety-Critical Communication for Railway Systems

Project period:
1 September 2007 –
1 August 2008

Specialization:
Wireless Communication

Author:
Morten Lisborg Jørgensen

Supervisors:
Hans-Peter Schwefel
Tatiana Kozlova Madsen
Torben Larsen

Number of copies: 5 + 1 electronic

Main report pages: 64

Appendix pages: 6

Total pages: 70

Synopsis:

Wireless communication rising in popularity due to the ease of use and the low cost of installing such equipment. However, safety-critical applications are holding back due to the risk introduced by the uncontrollable nature of wireless connections. This report analyzes the prospect of using wireless communication based on 802.11 off the shelf technology to support communication between the train computer and the control panel operated by the driver.
First the target application with existing solutions is described. The basics of 802.11 communication is then described, then the risks associated with a simple implementation are analyzed and necessary defenses are described, these defenses are then analyzed to evaluate the safety and availability of such a system.
In the analysis, both analytical and numerical methods are employed to assess the performance of the system. The models show a high frequency of occurrence of long delays of packets, and it is thus concluded that 802.11 under the assumptions of the report, is not a viable solution for on-board communication in the train environment.

Preface

This report is the result of my master thesis project conducted from September 2007 to August 2008. The project takes its starting point the in the SafeDMI programme and in cooporation with the partners of this programme. I would especially like to thank Prof. Andrea Bondavalli and his associates at the University of Florence where I spent part of the project period. I would also like to thank Iovino Danilo of Ansaldo Segnalamento Ferroviario (ASF – Ansaldo Signal) for his cooporation.

At the back of this report, a CD containing the report in PDF-format and the relevant MATLAB code is attached. The plots in this report are all made with MATLAB and a reference is provided to the employed MATLAB file. One vital MATLAB file can be found in appendix B.1.

Morten Lisborg Jørgensen

Contents

Chapter 1

Introduction

In many industrial systems, a reliable network connecting key parts, is an absolute necessity for safe operation. Errors in these connections can lead to breakdowns causing material damage or even loss of human lives. E.g. if the driver of a train or automobile activates the brake it is absolutely critical that this message is conveyed to the brakes and is not misinterpreted or lost along the way. Since wired connections provide a guide for the electric signal which is easily controllable, they inherently offer more robust communication than unguided, wireless connections. As such, wires are the weapon of choice in many safety-critical applications.

Wired connections are however inferior when it comes to installation costs. In trains and automobiles the problem is not a single wire, but rather the fact that a large amount of electronic components all need to be interconnected. It is thus attractive to replace the wired connection with wireless technology, given that it can provide the proper level of safety.

During the last ten years wireless network technology has matured to the point at which they are now ready to be developed for carrying safety-critical data. This landmark in wireless technology will also allow new safety-critical applications to be developed like car-to-car communication.

This project takes this next step working on the specific challenge of replacing the safety critical networks of a train with a wireless network connection based on off-the-shelf technologies. More specifically it targets the communication link between the train's on-board computer (the EVC – European Vital Computer), and the console which the engine driver operates (the DMI – Driver Machine Interface).

The project takes its starting point in the requirements imposed on the existing wired solution. Based on this, the risks introduced by a wireless solution are analyzed qualitatively, and risk reduction techniques are proposed. The full system is then modeled, and the performance is evaluated with respect to safety and availability.

The rest of the report is organized as follows:

Chapter 2: Pre-Analysis — In this chapter, the initial analysis of the problem is described. This incorporates descriptions of the basic system and analysis relevant standards and technology.

Chapter 3: Risk Analysis — In the risk analysis, an assumed system is analyzed further to identify the risks associated with its operation. Based on this assessment, risk reduction methods are proposed and the resulting risk is evaluated.

Chapter 4: Delimitation — From the solutions proposed by in the risk analysis, the most interesting

parts of the solution proposed in the risk analysis is selected, and thus the analyzed problem is reduced.

Chapter 5: System Analysis — In this section the proposed system architecture is analyzed in detail, this is done by dividing the system in sub-system and defining models for each entity. Based on the analysis the availability of the system is expressed and used for evaluation of safety.

Chapter 6: Conclusion — Finally the outcome of the project is evaluated and discussed, and future work is proposed.

List of Abbreviations

AES – Advanced Encryption Standard

AH – Authentication Header

ASF – Ansaldo Segnalamento Ferroviario – Ansaldo Signal

BER – Bit Error Rate

CCK – Complimentary Code Keying

CENELEC – Comité Européen de Normalisation Electrotechnique – European Committee for Electrotechnical Standardization

COTS – Commercial off the shelf

DCF – Distributed Contention Function

DMI – Driver Machine Interface

DSSS – Direct Sequence Spread-Spectrum

ERTMS – European Rail Traffic Management Systems

ESP – Encapsulated Security Payload

EVC – European Vital Computer

EDCA – Enhanced Distributed Channel Access

FER – Frame Error Rate

HCA – Hybrid Coordination Function

ICV – Integrity Check Value

IP – Internet Protocol

IPsec – Internet Protocol Security

MAC – Medium Access Control

PHY – Physical Layer

PLCP – Physical Layer Convergence Protocol

PPH – Packets Per Hour

SHA – Secure Hash Algorithm

SIL – Safety Integrity Level

SNR – Signal-to-Noise Ratio

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

WLAN – Wireless Local Area Network

Chapter 2

Pre-Analysis

2.1 Background

To ensure safe interoperability between Automatic Train Control (ATC) systems, while coping with the increasing speed of trains in an environment of increased traffic, the European countries has set up the European Rail Traffic Management Systems (ERTMS/ECTS) programme. This programme is to standardize ATC systems of which the on-board computer (the EVC – European Vital Computer) and Driver-Machine Interface (DMI) are two components. It is in this context that the SafeDMI project takes it's starting point.

The SafeDMI project is a European project which is partly funded by the European Commission which consists of five partners. The goal of the project is to design a DMI which can perform safely in the increasingly complex train systems of today. The output of the project is a number of deliverables which are gradually made available to the public at `www.safedmi.org`. These deliverables are in the rest of this report denoted by e.g. D1.2 for deliverable 1.2.

The DMI is the console through which the driver interacts with the EVC, which in turn controls the components of the train: engine, brakes, etc. The DMI consist of several devices, most notably CPU, keyboard, audio speaker, and LCD display (possibly touch sensitive). The DMI is conventionally connected to the EVC through a PROFIsafe fieldbus, a solution which is SIL 3 (Safety Integrity Level 3) certified. The implications of this certification is described in section 2.3.

2.2 EVC-DMI Communication

The communication between the EVC and DMI is described as a number of requirements imposed in D1.2 and in general terms in D2.1. This section first summarizes the requirements and then moves on to the more general description.

2.2.1 Requirements

This section summarizes the general requirements imposed on the system in D1.2 which are relevant to the EVC-DMI communication link. In the terminology of the deliverables the "wireless interface" in general refers to an update/configuration link which is used for servicing the SafeDMI software while the train is stationary and is thus auxiliary to the scope of this report. Each requirement (REQ) is postfixed with a

letter-code in brackets: [M], [HR], and [O] corresponding to “Mandatory”, “Highly Recommended”, and “Optional” respectively. Some requirement numbers are prefixed with a letter (like REQ C.6 [M]) which refers to the risk analysis of D1.1.

- General requirements
  - **REQ 7 [HR]** – “The SAFEDMI maximum response time towards EVC and driver commands shall be less than 1 second.”
  - **REQ 56 [O]** – “The SAFEDMI may use the wireless connection for on-board communication.”
  - **REQ 68 [M]** – “The communication between SAFEDMI and EVC shall be safe.”
  - **REQ 57 [M]** – “The SAFEDMI and its wireless interface shall be compliant according to the CENELEC standard EN 50 159-2.”
- Hardware-specific requirements
  - **REQ 1 [M]** – “The SAFEDMI shall support a COTS (Commercial Off-The-Shelf) hardware architecture.”
  - **REQ 10 [M]** – “The SAFEDMI power supply shall be 24V. For vehicles with other board voltage, another device (converter) must be added.”
  - **REQ 12 [O]** – “The SAFEDMI hardware design shall minimise the cost of managing hardware obsolescence in an area of rapidly evolving technology.”
  - **REQ E.1 [M]** – “The SAFEDMI shall operate regularly in the environmental temperature (T) range of $[T_{low}, T_{high}]$ where $T_{low} = -10°C$ and $T_{high} = +55°C$. The lower limit comes from the COTS LCD low temperature constraint.”
  - **REQ 14 [M]** – “The SAFEDMI shall operate regularly when the relative humidity (non-condensing) is at most 70% at an environmental temperature of T_{high}.”
  - **REQ 15 [M]** – “The lowest SAFEDMI storage temperature shall comply -20°C.”
  - **REQ C.4 [M]** – “The SAFEDMI electromagnetic compatibility shall be compliant with EN50121-3-2 [3].”
  - **REQ C.5 [M]** – “The SAFEDMI electric/electronic equipment shall be compliant with EN50155 [4].”

The following requirements and sub-requirements describes the messages passed between EVC and DMI. They provide no useful information about package contents and are thus referenced purely for completeness.

- **REQ 33 [M]** (and 17 sub-requirements) details the information which the SafeDMI shall display to the driver in response to EVC commands
- **REQ 34 [M]** (and 14 sub-requirements) details messages from EVC to SafeDMI which allows the SafeDMI to allow the driver to perform certain actions
- **REQ 37 [M]** specifies that the SafeDMI shall, on EVC request, produce a certain sound to draw the drivers attention
- **REQ 46 [M]** specifies that the DMI shall do self-checking and inform the EVC
- **REQ 59 [M]** specifies that the DMI should disconnect from the EVC when it goes into a safe state

- **REQ K.7 [M]** discusses a 2 step acknowledgment (by the driver) which is initialized by the EVC
- **REQ V.24 [M]** specifies that the DMI should inform the EVC if the LCD back-light lamp is not in the desired state

Note here that REQ 7 specifies a soft limit on the delay of packets, it is not stated in any requirements what delays are acceptable in order to maintain safe operation. The standard mentioned in REQ 57 is described in section 2.3.2.

REQ 1 states that the implementation should use COTS hardware. In some scenarios this implies the use of standard software as well in order to ensure interoperability with future hardware versions as described in REQ 12 which is however optional.

REQ E.1, REQ 14, and REQ 15 discusses temperature ranges and relative humidity where the DMI must be able to function or be stored. A 802.11 card [TRENDnet, 2007] does not comply with these requirements, but there is probably some interface cards in retail that can cope with these requirements. These requirements are thus not considered further.

REQ C.4 and REQ C.5 discusses electromagnetic compatibility and shock and vibration resilience, these requirements are not considered in this report either.

2.2.2 General Description

The communication between EVC and DMI is made through a number of information objects which are described in D2.1. The driver interacts with the DMI which in turn conveys the driver commands to the EVC.

Downstream is defined as communication from EVC to DMI while upstream is defined as communication from DMI to EVC. In D2.1 the information objects passed between the EVC and DMI are defined, they are:

- Downstream (from EVC to DMI)
  - **CyclicInfo** – Periodic information about the state of both the train in general, and the EVC e.g. speed, ceiling speed, etc.
  - **AperiodicInfo** – Aperiodic information, e.g. text notifications.
  - **EVCRequest** – Aperiodic requests for the execution of a specific procedure by the DMI, e.g. safe acknowledgment request, request for DMI status information. It is unclear if this message necessarily implies a response from the DMI.
- Upstream (from DMI to EVC)
  - **DMIReply** – Information about DMI status in response to a specific message called EVC_STATUS_REQ which is contained in the EvcRequest information object. The information object contains information about the status of the DMI.
  - **AckReply** – Response by driver or DMI to content previously received, e.g. confirmation of safe acknowledgment procedure.
  - **AperiodicEvent** – Genererated when the driver interacts with the DMI, e.g. pushes a button on the display.

The EVC can at any point be put into "safe mode" which means that the train is stopped by an emergency brake procedure. When the train is moving at high speeds this can be a violent action which can cause injury to the passengers. The safe mode is always initiated by the EVC. The DMI can also go into safe mode itself in order to avert error propagation to the EVC. When the DMI goes into safe mode it terminates the connection to the EVC which then probably brings the train into safe mode.

According to mail correspondence with ASF the messages passed between the EVC and DMI has a maximum size of 236 bytes which is a limit imposed by the PROFIBUS maximal packet size. Periodic packets are send each 400 ms to update the DMI display.

2.3 Relevant Standards

This section describes the two standards which are relevant for the present application: IEC 61508 and CENELEC EN 50159-2, relate to safety applications in the train environment.

2.3.1 IEC 61508

This section describes the safety standard IEC 61508 which introduces Safety Integrity Levels (SIL), the section is based on [Bell, 2006].

IEC 61508 is titled "Functional safety of electrical/electronic/programmable electronic safety-related systems". It was created with the purpose of defining practices in obtaining a proper safety level. The creation of the standard came as a reaction to the increased complexity introduced by microprocessors. With this introduction, testing alone was no longer practical to guarantee a certain level of safety, and thus a new methodology had to be introduced.

The standard addresses risk while recognizing two important properties:

1. Risk can only be reduced, not eliminated
2. Risk reduction must be approached holistically

To explain property one, consider a system which fails catastrophically with probability p in some given time period, e.g. the brakes of a train. A natural way to reduce the risk of failure is to introduce modular redundancy, in the example application: multiple, independent brake systems. With n-ary redundancy the probability of failure is reduced to p^n which is not zero for any value of n. In the example this means that no matter how much care that is put into assuring the availability of the brake system, availability can never be guaranteed.

The second property means that risks can arise from any part of the products life span, from the design phase, through implementation to the way the system is used. In the train example, redundancy in the braking system is useless if the designers has chosen brake button which has a too short life span. Thus it is necessary to reduce risk in all phases of the systems lifetime.

The standard defines safety as "the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment" [Bell, 2006]. Thus the definition is very wide.

The standard further defines two classes of system operation (from [Bell, 2006]):

"Low demand mode – Where the frequency of demands for operation made on a safety-related system is no greater than one per year and no greater than twice the proof-test frequency"

"High demand or continuous mode – Where the frequency of demands for operation made on a safety-related system is greater than one per year or greater than twice the proof-check frequency"

The analyzed system falls in the second category since the train must be assumed to be in almost continuous operation.

Figure 2.1 shows the target failure rates of a continuously high demand or continuously operated system for the different SIL levels. The EVC is SIL 4 and DMI along with the EVC-DMI connection is SIL 2 (see D2.1).

SIL	Average dangerous failure rate pr. hour	Corresponding MTBF
4	$[10^{-9}, 10^{-8}[$	11,416 – 114,155 years
3	$[10^{-8}, 10^{-7}[$	1,142 – 11,416 years
2	$[10^{-7}, 10^{-6}[$	114 – 1,142 years
1	$[10^{-6}, 10^{-5}[$	11 - 114 years

Table 2.1: Target failure measures for a safety function in high demand or continuous mode of operation. MTBF mean Mean Time Between Failures

2.3.2 CENELEC EN 50159-2

The CENELEC standard EN 50159-2 [CENELEC, 2001] titled "Railway applications - Communication, signalling and processing systems Part2: Safety related communication in open transmission systems" is relevant for the present application. EN 50159-1 addresses open communication systems. The two system classes are defined in [CENELEC, 2001] as

Closed – "a fixed number or fixed maximum number of participants linked by a transmission system with well known and fixed properties, and where the risk of *unauthorised* access is considered negligible."

Open – "a transmission system with an unknown number of participants, having unknown, variable and non-trusted properties, used for unknown telecommunications services, and for which the risk of *unauthorised access* shall be assessed."

A wired (e.g. PROFIBUS) system is considered a closed transmission system since the wires are difficult to access and only a fixed amount of nodes are attached. A wireless system like off-the-shelf 802.11 is considered an open transmission system since anyone is able to become part of the communication system without authorization.

The standard identifies a number of threats and defenses against them, the threats are:

- Delay – The message is delivered later than some deadline.
- Loss (deletion) – The message is not delivered.
- Corruption – The message arrives at the destination in changed condition.
- Resequence – The message arrives before the previous message or after the following message at the destination.
- Insertion – An arbitrary message which has not been sent by an authorized node, arrives at the receiver.

- Repetition – A valid message is reinserted at a later time instant.
- Masquerade – A message is tagged with a source which has not sent the message.

and the proposed defenses are

- Sequence number
- Time stamp
- Timeout
- Source and destination identifiers
- Feed-back message – Normally called acknowledgments, can also contain extended proof of reception, e.g. a checksum of the received data
- Identification procedure
- Safety code – Detection of corruption
- Cryptographic techniques

The threats and defenses are shown in Figure 2.1.

Threats	Defences							
	Sequence number	Time stamp	Time-out	Source and destination identifiers	Feed-back message	Identification procedure	Safety code	Cryptographic techniques
Repetition	X	X						
Deletion	X							
Insertion	X			X [2)]	X [1)]	X [1)]		
Resequence	X	X						
Corruption							X [3)]	X
Delay		X	X					
Masquerade					X [1)]	X [1)]		X [3)]

1) Application dependent
2) Only applicable for source identifier
Will only detect insertion from invalid source
If unique identifiers cannot be determined because of unknown users, a cryptographic technique shall be used, see 6.3.8.
3) See 7.3 and A.2.

Figure 2.1: Threats and defenses matrix, figure from [CENELEC, 2001]

2.4 Conventional Solution

Conventionally the EVC and DMI are connected through the PROFIsafe protocol. PROFIsafe is an application layer protocol designed to be used on top of PROFIBUS-DP protocol. This protocol stack

has SIL-3 certification. The rest of this section addresses the PROFIBUS-DP and PROFIsafe protocols separately.

The PROFIBUS (PROcess FIeld BUS) protocol is standardized as IEC 61158. The following is based on [Willig, 2002], [Acromag Inc., 2002], and [Tovar and Vasques, 1999].

PROFIBUS has widespread adoption, according to PROFIBUS International, more than 20 million nodes are in use worldwide [PROFIBUS international, 2008a]. The PROFIBUS physical layer (PHY) can use RS485 over twisted pair or coaxial cables for physical layer rates up to 12 Mbit/s, fiber optic cabling is also supported.

In the data link layer (DLL) which is called FDL (Fieldbus Data link Layer), PROFIBUS uses a token passing mechanism in a master-slave configuration. The nodes in a PROFIBUS network form a logical ring with unique successive addresses assigned at installation. Each node maintains two separate queues, one for high and one for low priority packets.

At initialization, the token is given to the first node which starts the token rotation. Upon reception of the token, the node starts a timer which is stopped upon reception the next reception of the token, this is the real token rotation time (t_{RR}). This value should be below the Target Token Rotation Time (t_{TR}) which is set at initialization. When receiving the token, the token holding timer is set to $t_{TH} = t_{TR} - t_{RR}$. If the t_{TH} counter is initialized with a negative value, the node can transmit maximally one packet from the high priority queue, if positive the node may send high and low priority packets in that order, as long as the holding timer is above zero. Whether a packet can be sent or not is solely based on the value of t_{TH} at the beginning of the transmission, thus overrun can happen for both high and low priority packets, the overrun is however limited due to the maximal packet size.

When t_{TH} becomes negative, the token transfer procedure must be initiated. When the token is received, the next transmission is started immediately, either a packet is transmitted or the next token passing procedure is started. The next transmission is used for implicit acknowledgment of the reception of the token. If the transmission by the receiver does not start before the slot time (t_{SL}) has passed, the token is retransmitted. The token is retransmitted three times after which the receiver is deemed dead, after which the sender passes the token to the next successor in the logical ring. Each node is responsible of polling the range of addresses between itself and the next node in the ring every gt_{TR} interval in order to find newly added, or reactivated nodes. g is called the gap factor.

The PROFIBUS packets are called telegrams and has a length of maximally $l_t = 255$ bytes of which 9 or 11 bytes is header data depending on the telegram type, and $l_d = 244$ bytes are payload data. The header contains an 8 bit checksum which is the simple sum of certain fields, including the payload, modulo 256.

The maximal throughput under saturated conditions is thus

$$S = r\frac{l_d}{l_t}\frac{t_{TR} - nt_{TP}}{t_{TR}} \tag{2.1}$$

where r is the physical layer transmission speed, n is the number of nodes, and t_{TP} is the time used by the token passing procedure for a single token pass.

PROFIsafe [PROFIBUS international, 2008b] is an extension to PROFIBUS which is currently progressing toward standardization within IEC 61784-3-3. PROFIsafe is an intermediate layer which is inserted in the protocol stack between the application layer and the data link layer (in the OSI model). The safety layer adds sequence numbers, time-out, source/destination identifiers, and CRC to each frame in order to increase safety compared to conventional PROFIBUS. This is in line with the defenses proposed in CENELEC EN 50159, see section 2.3.2, however PROFIBUS is considered a closed communication system. With these remedies, the protocol conforms to SIL 3. PROFIsafe can also be used over a wireless connection if these networks live up to certain security requirements which is a requirement set in order

to maintain that the system is a “closed communication system”.

2.5 Wireless Solution

This section analyzes wireless solutions. First the general restrictions of wireless systems are described, then the specific solutions of the 802.11 standards are described, first the original 802.11 standard with extensions a, b, g, and the draft n, then the Quality of Service (QoS) extension 802.11e.

2.5.1 General Wireless Characteristics

This section describes in general terms the problems encountered when replacing a wired system with a wireless one. Thus the discussed protocol, in this section, is an arbitrary, simple physical layer protocol.

If a sender and a receiver are suspended in free space, the transmission gain between them is

$$G \propto r^{-\alpha} \tag{2.2}$$

where r is the distance and α is the path-loss exponent, which in free space is 2. However in environments with obstructions, α is higher [Seidel and Rappaport, 1991]. This means that the distance between sender and receiver along with the physical layout of objects in the signal path, is critical for the quality of the link.

In addition to this effect which is primarily due to static properties of the environment, dynamic changes in the signal are caused by the multi-path propagation of the signal. This is caused by multiple versions of the signal with different delay combining at the receiver. The resulting signal can vary widely, and the variations occur quickly over time. [Rappaport, 1996]

In addition to these propagation phenomena, a third problem is caused by the shared nature of the propagation medium. In contrast to wired systems, it is difficult to refuse physical access to the medium. As such anyone can transmit or receive signals in the communication medium. Even simple interferers can easily cause connection outage, while more sophisticated interferers can insert false packages or perhaps even change them in transmission. All these problems exist in wired systems, but they are more evident in wireless systems.

2.5.2 802.11

Today the most widespread and versatile wireless protocol is probably IEEE 802.11 which is specified in [ANSI/IEEE, 2003]. Though the specification defines both RF and infrared physical layer, this report will focus only on the RF PHY which by far is the most widely employed, and due to the lack of line-of-sight between transceivers, the only feasible choice.

The RF PHY operates in the 2.4 GHz band for 802.11legacy (the original standard), b, and g or in the 5 GHz band for 802.11a, both bands are ISM-bands (Industrial, Scientific, Medical). This means that anyone can send and receive in these spectrums as long as they adhere to some limitations which the 802.11 protocols does by design. 802.11legacy specifies both a Direct Sequence Spread Spectrum (DSSS) RF and a Frequency Hopping (FH) RF, the latter is however not adopted to the same extent as 802.11a, b, g, or legacy.

Currently (2008) an amendment, 802.11n, is under development with draft 2.0 as the latest release. 802.11n uses MIMO (Multiple-Input Multiple-Output) technology with which multiple antennas both

at sender and receiver are used to improve performance with respect to range, throughput and error resilience. 802.11n is expected to become final in March 2009. However products based on the draft 2.0 specification are currently available in retail.

Based on the throughput requirements for the present application, 802.11b is deemed sufficient for the bandwidth needs of the present application. It is the simplest protocol after 802.11legacy. Based on this, the present report focus only on this edition of the 802.11standard.

In the physical layer, 802.11b offers four different transmission speeds: 1 Mbit/s using DBPSK modulation and Barker coding, 2 Mbit/s using DQPSK modulation and Barker coding, and 5.5 and 11 Mbit/s using DQPSK with CCK coding. Though not specified in the standard, transceivers usually select the optimal speed based on RF conditions. Both Barker coding and CCK coding is based on spread spectrum techniques.

In the physical layer, the Physical Layer Convergence Protocol (PLCP) is utilized. Any message passed down from the MAC-layer is prefixed by a 144 bit PLCP Preamble and a 48 bit PLCP header both of which are always transmitted at the basic rate, 1 Mbit/s. The preamble is used for synchronization and the header contains information about transmission speed, length of the transmission, and a 16 bit CRC to protect the header from corruption.

The physical layer offers roughly three service primitives to the MAC layer: Carrier Sense / Clear Channel Assesment (CS/CCA), Transmit (Tx), and Receive (Rx). Comparing to the wired alternative, 802.3 (Ethernet), the main difference is that send and receive operations cannot be conducted simultaneously. The CCA primitive is a special function offered by the PHY in which the state of the medium is inferred from the headers of other transmissions. At the beginning of each transmission, the length field of the PLCP header specifies the amount of time that the medium will be busy during the transmission, this is called the Network Allocation Vector (NAV). If the header is received correctly, the node does not need to sense the medium again before the end of the reserved period, thus saving energy.

In the MAC layer 802.11 offers two approaches, the mandatory DCF (Distributed Coordination Function) and the optional PCF (Point Coordination Function). PCF is always used in conjunction with DCF and is a deterministic approach in which an access point polls individual nodes for traffic, this is however very rarely implemented in actual devices.

DCF is a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol, thus contention is used to assign medium time to the different nodes. Optionally the sender can issue a RTS (Request To Send) packet before transmission which is answered by a CTS (Clear To Send) packet after which the transmission is initiated. This method both decreases the collision period (because the RTS packet is short) and addresses the hidden and exposed terminal problems. However, if packets are short the RTS/CTS-mechanism is inefficient due to the extra airtime consumed by these extra messages [Chatzimisios et al., 2004].

Figure 2.2 shows a transmission of a packet under DCF. In the situation shown, the medium is inactive when the source gets a packet in queue, in this case the source can thus wait a DIFS interval and then immediately initiate the transmission. If the transmission is successful, the destination waits a SIFS interval and transmits an acknowledgment packet. After the ACK packet has been received, any node wanting to send a packet must wait for a DIFS interval before the contention process is initiated.

When initiating contention, each contending node must choose a random backoff counter uniformly in the interval [0; CW-1], where CW is the contention window size which takes an initial value of $CW_{\min}$. For each aSlotTime interval that passes with an idle medium, each node must decrease their backoff counter by one. When the backoff counter at a node reaches 0, the node must transmit. This process in presence of 5 nodes is illustrated in Figure 2.3.

If the transmission fails, i.e. no ACK is received after the transmission, the sender node must wait an

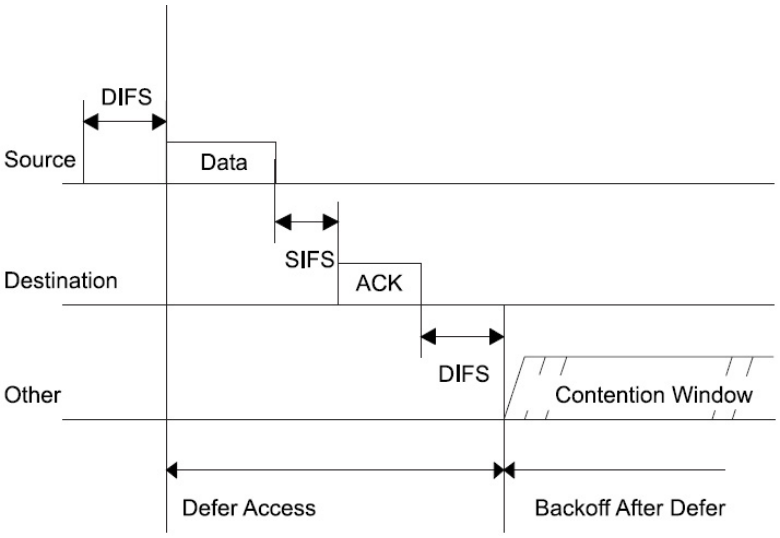

Figure 2.2: The two-way handshake operation of DCF. Figure from [ANSI/IEEE, 2003, p. 84]

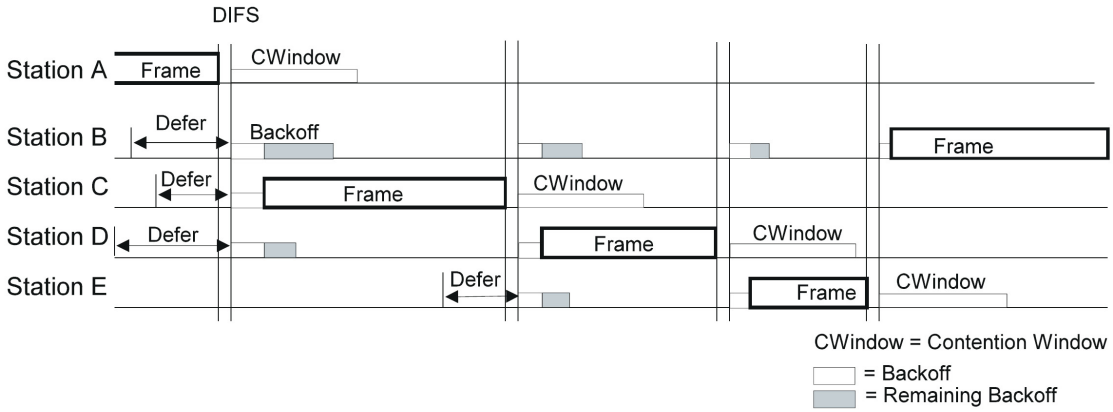

Figure 2.3: Backoff counters in DCF with five nodes in the collision domain. Notice how the remaining backoff interval for each node decreases when no transmission is going on. Figure from [ANSI/IEEE, 2003, p. 78]

ACK_TIMEOUT interval before contention is re-initiated for retransmission. For each retransmission the contention window size is doubled such that the i'th contention window has size $2^{i-1}CW_{\min}$. The contention window size increased in this way for each retransmission until it reaches $\mathrm{CW}_{\max}$ at which level it remains until the maximal number of transmissions is reached.

A MAC frame header has a maximal length of 34 bytes including 4 bytes of CRC. An ACK packet is purely header, and contains only a few fields for a total length of 14 bytes including 4 bytes of CRC. The utilized CRC polynomial is the standard CRC-32-IEEE-802.3 which is used in 802.3 (Ethernet).

2.5.3 802.11e

The 802.11e [ANSI/IEEE, 2005] standard has been developed to incorporate QoS requirements in the 802.11 standard. 802.11e introduces a third coordination function: HCF (Hybrid Coordination Function). The HCF contains two sub-methods for channel access: Enhanced Distributed Channel Access (EDCA) and, HCF Controlled Channel Access (HCCA) which roughly are extensions of DCF and PCF; both methods use traffic classes to prioritize traffic.

EDCA is a distributed technique much like DCF with the single change that aggressiveness in the contention period is weighted by traffic priority such that high priority data is more likely to be transmitted first. Also the transmission time "won" in contention is restricted such that slow senders do not occupy the channel for disproportionate amounts of time.

HCCA is the counterpart of PCF, in this mode a coordinator called Hybrid Coordinator (HC), usually an access point, controls transmissions in contention free periods. Like PCF HCCA also leaves room for contention periods in a fashion determined by the HC. The HC can poll the individual nodes for buffer levels and allow them to transmit for a selected amount of time. This approach is very versatile since the HC can be implemented to schedule in any way deemed prudent.

802.11e can be used to improve real-time performance and availability of the system, especially it might be useful to give high priority to emergency traffic.

2.6 IPsec

IPsec is a network layer protocol which can be inserted transparently on top of IP in the protocol stack. IPsec is specified in RFCs 2401–2412 and runs on top of IP as a transport layer protocol and offers both authentication, which is provided through the AH (Authentication Header) packet, and confidentiality (with authentication) which is offered by the ESP (Encapsulated Security Payload).

The AH contains a sequence number and an ICV (Integrity Check Value) which is calculated over both payload and relevant IP-headers. The ICV can be calculated with almost any hash-function depending on implementation.

IPsec is used to add cryptographic methods to the protocol stack in a simple way. It is widely adopted in several implementations in various operating systems, and integrating it in the implementation is thus unproblematic.

There are several available keyed hash functions (HMACs) which can be used with IPsec. Historically, hash functions have a certain lifetime before weaknesses are found, when this happens the function is regarded as insecure and should be replaced by a more powerful, unbroken alternative. When IPsec is used, it is simple to change HMAC, it will only require a simple configuration change and possibly a software update. With this in mind it is not necessary to incorporate any other responses to a potential

break in the cryptographic hash function.

The most widespread family of hash functions is the SHA functions. The newest generation of SHA is SHA-2 which comes in different versions with varying key length. The 256-bit function (called SHA-256) is the function with the lowest number of bits in the key for which a possible attack has not yet been publicized. This hash-function has undergone several public review processes to ensure that it performs properly and does not contain back doors or is vulnerable to simple attacks.

[Niedermayer et al., 2006] analyzes the delay penalty introduced by IPsec. The paper finds that a 300 byte packet is delayed approximately 150 calculation cycles pr. byte, i.e. 45000 cycles for the full packet when using the 128 bit AES encryption protocol and the SHA-1 hash function, both are considered very secure. On a 200 MHz processer this is approximately 225 μs.

Chapter 3

Risk Analysis

In this section, first a preliminary system layout is assumed, the risks associated with this system are identified, and methods are presented to reduce these risks.

3.1 Assumed system layout

For further analysis the system is assumed to consist of an 802.11b PLCP physical layer, 802.11b DCF MAC layer connected as shown in Figure 3.1. The choice of 802.11b is argued in section 2.5.2.

In the network layer the Internet Protocol is assumed, this is done because it is the most widespread COTS protocol in existence and thus interoperability is optimal. In the transport layer the User Datagram Protocol (UDP) is assumed. The choice of UDP is made because the protocol is assumed to be connection-less as opposed to TCP which offers connection-oriented service.

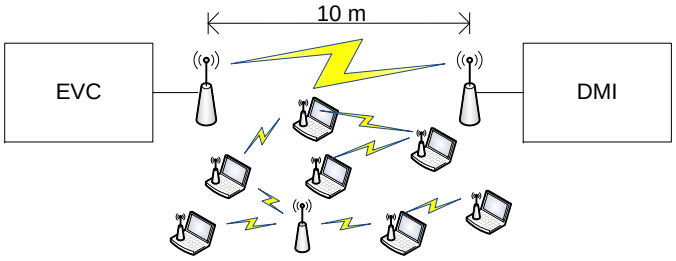

Figure 3.1: The assumed system configuration for the risk analysis including EVC and DMI, other nodes (the passengers) and their access point.

Based on information from ASF the EVC and DMI are assumed to be spaced 10 m apart. Also based on information from ASF, the maximal payload length is 236 bytes. The EVC sends a packet every 400 ms and it is further assumed that 1 aperiodic packet is also send every second for a total load of 3.5 packets pr. second. The size of packets is assumed to be constant at the 236 bytes which is the maximum in PROFIsafe.

The system is assumed to be in continuous operation and the EVC is assumed to put the system into safe mode after a substantial delay or a packet loss. Attackers are assumed to have sufficient skill and

equipment to mount both noise transmission attacks as well as sending out intelligent interference across multiple channels simultaneously. The attacker is assumed to have both directed and omni-directional antennas.

3.2 Methodology

This risk analysis is conducted in a process similar to the analysis conducted in D1.1 which is inspired by "The Yellow Book" [Rail Safety and Standards Board, 2007], a rail safety management guide. In the yellow book, 6 stages are described, these are depicted in 3.2 except for the "Demonstration of Acceptability", which is not relevant before a final product is presented.

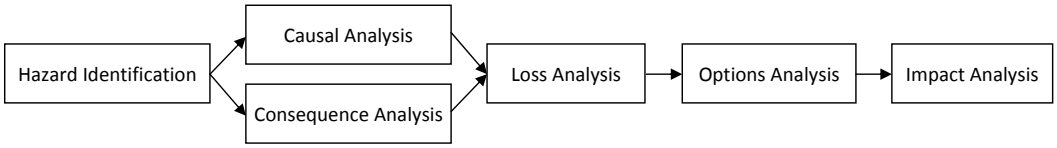

Figure 3.2: Stages of the Risk Analysis

In the hazard analysis, the hazards are identified, i.e. the situations which can lead to risks. Then in the Causal Analysis, the causes of the hazards are identified and in the consequence the consequences of hazards are identified. These results are then collected in the Loss Analysis where the overall risk level is identified, i.e. the loss associated with a hazard weighted by its frequency. Then in the Options Analysis, possible risk reduction techniques are identified, and the resulting loss after risk reduction is identified in the Impact Analysis.

The risk analysis presented is made through qualitative assessments of system performance. While the disadvantage is the precision of the analysis, the advantages are the dramatical reduction in resources that has to be devoted to the task. The results are still useful as they show which parts should be further investigated, and which parts that should not.

3.3 Hazard Identification

The yellow book defines a hazard as "any situation that could contribute to an accident". In section 2.3.2 the possible errors that can occur in transmission were identified. These errors are almost always hazardous and thus the hazards are defined as these events.

Defining the scope of a hazard is arbitrary, the narrower a hazard is defined, the more hazards has to be analyzed, but the more precise the identification of risk will be. However the above hazards are selected for further analysis as risk reduction will typically focus on reducing these risks.

Packets transmitted between the EVC and the DMI are the information objects described in Section 2.2.2. There are six information objects which each can contain a variety of messages. The seven error conditions mentioned above can be applied to any message passed over the network connection. Each combination of error and message constitutes a potentially hazardous condition, not all combinations does however cause a hazard since the assumed protocol already protects against certain errors.

3.4 Causal Analysis

The Causal Analysis identifies causes and frequency of hazards. In Section 3.3 the causes of risks were established, this section thus focuses on determining their frequency. The analyzed errors can occur naturally in any wireless system, both sporadically and periodically. However they can also be artificially introduced by a malicious attacker. All kinds of malicious attacks are considered to be equiprobable since it must be assumed that if an attacker is determined to break into a system, he will attempt any known attack vector.

For qualitative frequency assessment, this section uses the same terms for cause frequencies as D1.1 which are derived from EN 50126 ([CENELEC, 1999]):

- Frequent – Daily to monthly occurrence
- Probable – Monthly to yearly occurrence
- Occasional – Occurs between once every year and once every 10 years
- Remote – Occurs between once every 10 years and once every 100 years
- Improbable – Occurs less than once every 100 years
- Incredible – Events that are extremely unlikely to occur. It can be assumed that the hazard may never occur.

The rest of this section analyzes the frequencies of the causal factors while assuming:

- All messages occur with the same frequency
- The length of a message has no influence on its hazard frequency
- The type of message has no influence on the hazard frequency

3.4.1 Accidental Delay

Delay is caused by two phenomena: CRC-detected bit errors in the physical layer (both from collisions and other detection errors) and contention delay in the MAC layer. Bit errors detected by the CRC results in a discarded packet, which causes retransmission until the maximal amount of retransmission has been reached. When the retransmission limit is reached, the packet is dropped, and it is up to the higher layers to perform further error handling. Contention delay is introduced when multiple users attempt to access the medium simultaneously.

[Zheng et al., 2006] analyzes the average delay in 802.11b by simulation. For small loads the delay is around 10-20 ms for 10 nodes in contention, but can potentially be a lot larger depending on the amount of traffic generated, and the number of nodes in the network. With a load below $0.5\,\mathrm{MBit/s}$ in a $2\,\mathrm{MBit/s}$ network the delay is below 20 ms with a BER of 10^{-5}. However as [Sakurai and Vu, 2007] documents, there is a small probability of packets being delayed substantially longer than this, even more than a second. However the data from the latter source does not conform well with the results from the former source.

REQ 7 states that the response time between EVC and DMI should be less than 1 second, however it is not clear if this relates to safety. It is however assumed to be unlikely that a delay of reply of one second

is considered hazardous. Thus it is assumed that a hazardous delay is longer than 5 seconds. This leaves 2.5 seconds per transmission direction. Based on the above, significant delays of more than 2.5 seconds are very rare when seen on a per packet basis, for the present application with an assumed 3.5 packets pr. second in continuous operation, more than 9 million packets are sent pr. month. With this volume to be a frequent cause, only one packet has to be delayed for more than 2.5 seconds corresponding to a probability of long delays of 10^-7, the probabilities of such a delay is found in section 5.9 confirming that the occurence is *frequent*.

3.4.2 Accidental Loss

Packet loss is very similar to delay since it can be regarded as a large delay. The only difference is that it does not cause queuing of packets behind it. Based on this, the event is assumed to have the same frequency as delay: *frequent*.

3.4.3 Accidental Corruption

Corruption of data occurs when transmitting through the wireless medium, this is generally detected by the data link layer which triggers a retransmission. The MAC layer has the strongest corruption detection mechanism as it employs mathematically founded CRC codes. Specifically the CRC-32 which is also used in 802.3. CRC-32 has a hamming distance of 5 bits for codewords up to 2000 bits [Koopman, 2002] which is length of packets in the analyzed application. Based on the volume of packets, this event is assumed to be *frequent*.

3.4.4 Accidental Resequence

Resequence happens when errors in the protocol stack causes the packets in a queue to switch positions. Events leading to such a swap are very unlikely, though programming errors cannot be ruled out, the event is thus assumed to be *probable*.

3.4.5 Accidental Insertion

Insertion of packets can happen accidentally, but it is very rare that packets are created from nothing, especially with the assumed protocol stack. This can happen if

- An programming error or similar causes undesired writes to a queue, e.g. to a queue level counter
- Noise in the receiver takes the form of a valid packet including a correct CRC code

Each of these events are unlikely in absence of design and programming flaws, however these must also be taken into consideration and insertion is thus *probable*.

3.4.6 Accidental Repetition

In the MAC layer, repeated frames are generally detected by the 12 bit sequence number, so if a MAC frame is repeated at a random time, assuming periodic packet transmissions, there is a $\frac{1}{2^{12}} \approx 2.4 \cdot 10^{-4}$ chance that the packet is not discarded by the receiver. By the time of such a transmission, the MAC layer

at the sender will have passed on $2^{12} - 1 = 4095$ other packets. The only cause of such an event would be a software error, and this is unlikely due to the low complexity of the protocol.

In the higher layers (UDP/IP), the protocol is more complex. Again programming error is the only cause of repetition, but this is slightly more likely due to the increased complexity. It should be noted that if a packet is repeated in the higher layers, the MAC layer has no way of detecting it.

Thus the frequency of this cause is assessed to be *probable*.

3.4.7 Accidental Masquerade

Accidental masquerade occurs when the source field is changed, the notion of source exist both in the transport layer (UDP), network layer (IP) and in the MAC layer (source address), and as such both has to be changed in order for masquerade to occur. For this to happen, one of the following events has to occur

- In the transmission in the physical layer, the relevant bits are changed such that the source address is changed while maintaining consistence between MAC and IP addresses. The CRC is also changed such that this check is correct
- Due to software error, the source IP address is changed in the UDP layer. Since the MAC address is obtained from the IP address, this address is by design consistent and no further error has to occur.

The latter is clearly the most likely and its frequency is assessed to be *remote*.

3.4.8 Malicious Attack

It is difficult to assess the frequency of malicious attacks. Initially the attacker must be assumed not to know anything about the system, in this scenario the frequency of attacks are very low. But over time, hackers will learn more and more facts about the system until they reach a point at which they can mount a dangerous attack. At this point the frequency will go up. Depending on the reward for compromising the system, which for e.g. terrorist organizations is large, the time needed to get to this level of knowledge is short or long [Bishop, 2007], but once the level of knowledge is reached by a potential attacker, it is likely to spread to other hackers. Regarding only this worst case situation, the frequency of malicious attacks is assessed to be *probable*.

3.4.9 Summary

Table 3.1 summarizes the assessments of this section,

Error	Delay	Loss	Corruption	Resequence
Frequency	*Frequent*	*Frequent*	*Frequent*	*Probable*
Error	Insertion	Repetition	Masquerade	Malicious Attack
Frequency	*Probable*	*Probable*	*Probable*	*Probable*

Table 3.1: Results of the Causal Analysis. Malicious attacks can cause any combination of the other hazards.

3.5 Consequence Analysis

The Consequence Analysis assesses consequences that may arise from the previously identified hazards. While the previous sections addresses accidental and malicious hazards separately, this section treats them jointly. This is because no matter how the a hazard has been created, the consequence will be the same. Each hazard is associated with a severity level, these levels are as in D1.1 derived from EN 50126 [CENELEC, 1999]:

- Catastrophic – Fatalities and/or multiple severe injuries and/or major damage to the environment
- Critical – Single fatality and/or severe injury and/or significant damage to the environment / Loss of a major system
- Marginal – Minor injury and/or significant threat to the environment / Severe system damage
- Insignificant – Possible minor injury/Minor system damage

In the risk reduction section of D1.1 (specifically RD-18 and RD-29) the consequence of a brake procedure is assessed to having "insignificant" consequence, the same is assumed here. Corruption, Insertion, Repetition, and Masquerade (CIRM) has equal consequences since all of them are different ways of introducing false information to the communication.

3.5.1 CyclicInfo

This information object is used by the EVC to periodically transmit information to the DMI about the current state of both the EVC or the train in general. This information could be speed, maximal speed, position of the train etc. which is displayed to the driver. The CyclicInfo information object is transmitted every 400 ms if possible, or at least every second. When received by the DMI the information contained in the message is queued for display on the screen. When the information has been displayed a DMIReply message is sent back to the EVC after which the EVC is allowed to send a new CyclicInfo message with new data. This method is used to assure that only one CyclicInfo object is queued for display so that the most recent data is always shown to the driver. (Based on information from ASF)

Delay If the message is delayed the DMI will continue to show the most recent information that it has received. Thus old state information is being displayed to the driver which leaves him/her less informed than the design intended. When the driver is uninformed he/she could make a wrong decision which could bring the train into a dangerous state. However, if the acknowledgment message (DMIReply) is not received by the EVC withing e.g. 5 seconds the train would go into a safe state. Thus the age of the information is limited and the resulting severity is assessed as *insignificant*.

Loss If this information object is lost, no acknowledgment is sent back which causes the present information to remain on the display. After some time the EVC will detect that no response has been sent and it will go into safe mode. In the time before the EVC goes into safe mode, the driver could make an unsafe decision which is equally dangerous as with delay and thus *insignificant*.

CIRM If wrong information is introduced it might cause the driver to make an unsafe decision. CIRM is more severe than loss and delay because the difference between the true values and the values actually shown to the driver can be a lot larger. Thus the severity of CIRM is *critical*.

Resequence This message is not transmitted before the previous has been acknowledged, there can thus be no confusion and this error thus *cannot happen*.

3.5.2 AperiodicInfo

AperiodicInfo contains informative data generated aperiodically by the EVC, it contains a text message to be displayed to the driver.

Delay If this message is delayed, critical information might not be readable to the driver before it is outdated. However, it is assumed that the EVC will go into safe mode if the delay is dangerous. So the possible consequence is *marginal*

Loss Loss of this information object has similar impact as delay: *Marginal*.

CIRM as with CIRM of CyclicInfo: *Critical*.

Resequence Sequence change cannot cause any confusion to the driver and is thus *insignificant*.

3.5.3 EVCRequest

The EVCRequest information object is transfered from the EVC to the DMI aperiodically to make the DMI perform a certain action. This could be a safe acknowledgment by the driver, status of the DMI, etc.

Delay Delay could cause the status of the DMI to be inconsistent with the EVC, this can cause the EVC to misunderstand driver commands which can be *Catastrophic*.

Loss Similar impact as delay: *Catastrophic*.

CIRM Wrong information in this information object might cause the status of the DMI to be different from what the EVC expects, this can be *Catastrophic*.

Resequence Resequence can cause the status of the DMI to be different from what the EVC expects, this can cause misinterpretation of driver commands which can be *catastrophic*.

3.5.4 DMIReply

DMIReply is used to inform the EVC about the status of the DMI. The information object is sent in response to the EVC_STATUS_REQUEST message which is a EvcRequest.

Delay A delay longer than a certain threshold might cause the EVC to go into safe mode, the consequence is *insignificant*.

Loss Similar to delay: *Insignificant*

CIRM If the packet contains wrong information the EVC might be lead to believe that the DMI is working properly in cases where it is not, this can potentially be *catastrophic*.

Resequence Sequence change is not possible with information objects of the same kind since the EVC will not request the status of the DMI again before response has been received. Sequence change with other information objects cannot be confusing and thus *cannot happen*.

3.5.5 AckReply

AckReply is a response to a previously received message which can be of different kinds.

Delay A delay longer than a certain threshold might cause the EVC to go into a safe state because of lack of reply, the impact of this is *insignificant*.

Loss As delay: *Insignificant*.

CIRM If this information object contains wrong information, the EVC might think the driver has acknowledged information which he has not, depending on the acknowledgment protocol, this can potentially be *catastrophic*.

Resequence Since resequence can possibly cause the EVC to be confused about what is being acknowledged, the impact can be the same as CIRM: *Catastrophic*.

3.5.6 AperiodicEvent

The AperiodicEvent information object is caused by driver interaction with the DMI, e.g. button press.

Delay A delay might cause time critical data to arrive too late at the EVC which is potentially *catastrophic*.

Loss A loss is similar in impact to a delay: *Catastrophic*.

CIRM If wrong information arrives at the EVC, it might react improperly to a driver command which is *catastrophic*.

Resequence Resequence can cause the DMI to be in a different state than expected by the EVC which can be *catastrophic*.

3.5.7 Summary

Table 3.2 summarizes the results of this section.

	Delay	**Loss**	**CIRM**	**Resequence**
CyclicInfo	*Insignificant*	*Insignificant*	*Critical*	*N/A*
AperiodicInfo	*Marginal*	*Marginal*	*Critical*	*Insignificant*
EVCRequest	*Catastrophic*	*Catastrophic*	*Catastrophic*	*Catastrophic*
DMIReply	*Insignificant*	*Insignificant*	*Catastrophic*	*N/A*
AckReply	*Insignificant*	*Insignificant*	*Catastrophic*	*Catastrophic*
AperiodicEvent	*Catastrophic*	*Catastrophic*	*Catastrophic*	*Catastrophic*

Table 3.2: Results of the Consequence Analysis (CIRM is as noted earlier any hazard arising from Corruption, Insertion, Repetition, or Masquerade error). Hazards marked with N/A cannot happen, and thus have no associated risk.

3.6 Loss Analysis

The Loss Analysis assesses the magnitude of safety losses for each individual hazard. This is done by evaluating frequency of occurrence as determined in 3.4 with severity as determined in 3.5. For this analysis the loss levels of D1.1 which are derived from EN 50126 [CENELEC, 1999] are used:

- Intolerable – The risk shall be eliminated
- Undesirable – Shall only be accepted when the risk reduction is impracticable and with the agreement of the Railway Authority
- Tolerable – Acceptable with adequate control and the agreement of the Railway Authority
- Negligible – Acceptable without any agreement of the Railway Authority

The mapping from the causal and Consequence Analysis of sections 3.4 and 3.5 is done as in D1.1, see Fig. 3.3.

	Frequency of Occurrence of a Hazardous Event	**RISK LEVELS**			
Daily to monthly	**FREQUENT (FRE)**	Undesirable (UND)	Intolerable (INT)	Intolerable (INT)	Intolerable (INT)
Monthly to yearly	**PROBABLE (PRO)**	Tolerable (TOL)	Undesirable (UND)	Intolerable (INT)	Intolerable (INT)
Between once a year and once per 10 years	**OCCASIONAL (OCC)**	Tolerable (TOL)	Undesirable (UND)	Undesirable (UND)	Intolerable (INT)
Between once per 10 years and once per 100 years	**REMOTE (REM)**	Negligible (NEG)	Tolerable (TOL)	Undesirable (UND)	Undesirable (UND)
Less than once per 100 years	**IMPROBABLE (IMP)**	Negligible (NEG)	Negligible (NEG)	Tolerable (TOL)	Tolerable (TOL)
	INCREDIBLE (INC)	Negligible (NEG)	Negligible (NEG)	Negligible (NEG)	Negligible (NEG)
		INSIGNIFICANT (INS)	**MARGINAL (MAR)**	**CRITICAL (CRI)**	**CATASTROPHIC (CAT)**
		Severity Levels of Hazard Consequence			

Figure 3.3: Mapping from causality and consequence to loss. Figure from D1.1.

This section again treats malicious interference separately from accidental interference to the system. Tables 3.3 and 3.4 shows the resulting losses arising from the hazards for hazards introduced inadvertently and maliciously respectively.

From these tables it can be seen clearly that risk reduction should be conducted against all hazards, both accidental and malicious.

	Delay	Loss	Corruption	Resequence	Insertion	Repetition	Masquerade
CyclicInfo	*Undesirable*	*Undesirable*	*Intolerable*	*N/A*	*Intolerable*	*Intolerable*	*Intolerable*
AperiodicInfo	*Intolerable*	*Intolerable*	*Intolerable*	*Tolerable*	*Intolerable*	*Intolerable*	*Intolerable*
EVCRequest	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*
DMIReply	*Undesirable*	*Undesirable*	*Intolerable*	*N/A*	*Intolerable*	*Intolerable*	*Intolerable*
AckReply	*Undesirable*	*Undesirable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*
AperiodicEvent	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*

Table 3.3: Result of the Loss Analysis of accidentally induced hazards.

	Delay	Loss	Corruption	Resequence	Insertion	Repetition	Masquerade
CyclicInfo	*Tolerable*	*Tolerable*	*Intolerable*	*N/A*	*Intolerable*	*Intolerable*	*Intolerable*
AperiodicInfo	*Undesirable*	*Undesirable*	*Intolerable*	*Tolerable*	*Intolerable*	*Intolerable*	*Intolerable*
EVCRequest	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*
DMIReply	*Tolerable*	*Tolerable*	*Intolerable*	*N/A*	*Intolerable*	*Intolerable*	*Intolerable*
AckReply	*Tolerable*	*Tolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*
AperiodicEvent	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*	*Intolerable*

Table 3.4: Result of the Loss Analysis of maliciously introduced hazards.

3.7 Options and Impact Analysis

The Options Analysis identifies measures which can be taken to reduce the risks identified previously. In the Impact Analysis the resulting risk after risk reduction is found.

Risk reduction focuses on two different kinds of hazards, malicious and accidental. Even though the errors introduced are similar the measures taken to protect against them are in some case different.

In the following each error type is analyzed to find protection against both accidental and malicious errors. Risk reduction can focus on either reducing the hazard frequency (prevention) or the consequence if a hazard should arise (protection). All risk reduction strategies are derived from EN 50159-2 [CENELEC, 2001], see section 2.3.2.

3.7.1 Delay

Delay can be introduced both accidentally and maliciously, however the countermeasures against these two categories of hazards are the same since a malicious attacker cannot cause delays in ways that separate them from accidental ones.

It has previously been assumed that a one-way delay of less than 2.5 second is safe. The specific number chosen for is a design decision which is a trade-off between safety and availability. The value chosen should be the highest possible that provides an adequate level of safety since safety is the most important. This is discussed further in Chapter 5.

The strategy for preventing delays shall reduce the frequency of delays which are longer than the maximal delay. The protection strategy shall detect delays longer than the maximal delay and upon detection initiate a safe response.

Delays arise from different low-level phenomena as described in section 2.5.1, however delays longer than 2.5 seconds are most often caused by long term effects (long term meaning events lasting more than a few tenths of milliseconds): Shadowing, and interference by other equipment operating in the ISM band.

Interference is frequency selective, thus adding redundancy in frequency can reduce the amount of longer delays. This could be done by adding more wireless devices in both ends using different channels. At the sender, all messages should be sent on all connections, and at the receiver only the first version of

a message to arrive should be let through to the application. However, it is unlikely that this will be effective in reducing the amount of hazardous delays by a factor of 10-100 as is needed to make the event "probable" as opposed to frequent.

When assuming static positions of transceivers and no influence on the dynamic environment, these physical layer effects cannot be prevented. However in short periods of congestion, prioritization of traffic can make more important packets arrive quicker at the destination. E.g. AckReply could be assigned a low priority because the impact of delay on this packet is *insignificant* according to table 3.2, while AperiodicInfo could be assigned a high priority because a delay of this packet has catastrophic consequences. This approach does however not reduce the frequency substantially.

Since hazard prevention does not reduce the risk of delays substantially, the frequency of delays remain at *frequent*.

Thus hazard protection, i.e. reduction of the loss when delays occur, is necessary. This can be done by introducing an acknowledgment protocol between the transport layer (UDP) and the application. Based on the delay of acknowledgments, the current delay can be assessed and the system can be put in safe mode if the delay becomes too big. This can also be done using cross-layer methods to tap into the information which is available to the MAC-layer, but such a solution is too complicated since it will require changes to the wireless driver and/or firmware which is outside the scope of this project.

With this reduction, the consequence of a delay is reduced to *insignificant*.

Thus with double redundancy of the wireless link, prioritization, and introduction of the described delay detection mechanism, the risk associated with accidental and maliciously introduced delay is reduced to *undesirable*.

3.7.2 Loss

All the risk reduction techniques introduced to counter delay, also has a positive effect on the packet loss. Additionally retransmissions can be introduced, this effectively converts packet losses to delays, and if infinite retransmissions are employed, losses become *incredible*.

The consequence of losses are still the same, the worst case being *catastrophic*. Thus the risk after risk reduction is *negligible*.

3.7.3 Corruption

Corruption can be introduced both accidentally and maliciously, however the way of introducing corruption is different.

The common way of dealing with general corruption in communication systems is to add redundancy to the message, i.e. checksums. When random errors occur CRC (Cyclic Redundancy Check) codes are very effective in detecting them, and their effectiveness can be mathematically analyzed. But since an attacker must be assumed to know how to calculate the CRC, he will be able to intelligently change a message such that both message and CRC code is changed to match, instead cryptographic methods must be used.

A cryptographic hash function is a one-way function which returns a hash of a fixed number of bits when used on any amount of data. The goal of such function is to make it computationally difficult to create a message that matches a hash. A keyed hash function uses a key in conjunction with the hash function such that only the parties knowing the key are able calculate the correct hash which proves the origin and authenticity of the message.

However hash functions has the weakness that their effectiveness against random errors cannot be proved mathematically. Thus they cannot replace CRC codes and both protection methods thus has to be used in conjunction to ensure protection against both accidental and malicious errors.

Upon detection of corruption, the corrupt message shall be discarded and requested retransmitted from the sender. The system shall repeat this procedure until the message has come through intact or it has been delayed enough to cause the system to go into safe mode.

The methods described reduces the frequency of accidentally introduced corruption to *remote* because the CRC codes can be chosen to give a certain accidental corruption frequency corresponding to a *negligible* risk level. The frequency of malicious corruption is reduced to *improbable* under the assumption that the chosen hash function is updated at regular intervals as weaknesses are found.

This reduces the risk to *negligible* for both accidental and malicious attacks.

3.7.4 Resequence

Resequence can be introduced both accidentally and maliciously, the countermeasures against both kinds of errors are the same.

Sequence numbers as introduced in the solution of delay can be used to detect resequencing. When detected, resequencing can either be fixed by buffering packets and releasing them to the application layer or packets can be passed on to higher layers in the order they arrive. The former approach is relevant when resequence is a hazard while the latter is relevant when it is not, see Table 3.2. If withholding a packet means delaying it more than `max_delay`, the system should go into safe mode.

This addition will not change the consequence which remains at *catastrophic* in the worst case. The frequency of the hazard is reduced to *incredible* both for malicious and accidental errors. Thus the risk is *negligible* both with respect to accidental and maliciously introduced hazards.

3.7.5 Insertion

Insertion can occur both accidentally and as a result of a malicious attack, the defenses against the two are the same. There exist several defenses against this hazard: Sequence numbers, source identifiers, feed-back messages, and message authentication.

Sequence numbers helps in detecting when a message has been inserted, as sequence numbers in the previous has been deemed necessary for reduction of the risk of resequence, this feature will be included. When the same sequence number is seen twice by the receiver, a message must have been inserted, it is however impossible to know which message is the inserted one, thus the system must be put in safe mode.

By using source and destination identifiers, the receiver of an inserted message can detect if he is the right receiver and if the sender is valid. Whether the sender is who he claims is a masquerading problem which is addressed in a later section. Upon detection of such an error, the message should be dropped.

The previously introduced keyed hash authentication technique will also provide protection against this error as an inserted message must have a proper authentication code to be accepted at the receiver.

With these additions the frequency of accidental insertion will be reduced to *remote*. Maliciously introduced messages will be reduced in frequency to *improbable*.

With these additions the risk of insertion will be reduced to *undesirable* and *tolerable* for both accidental and maliciously introduced hazards.

3.7.6 Repetition

Repetition can be introduced both accidentally and maliciously, the defenses against these two methods are the same. Repetition is very similar to insertion as an extra message is inserted, however a repeated message can have some properties that a random inserted message does not, e.g. the message will have a proper message authentication code.

The sequence numbers introduced earlier will assure that a repeated message will not get through. When a repeated message is detected, because of the precautions introduced to counter insertion, the system is put into safe mode. The consequence of this is *insignificant*.

This makes the risk of repetition *negligible* and *tolerable* toward accidental and malicious hazards respectively.

3.7.7 Masquerade

Masquerade can occur both as a result of accidental and maliciously introduced errors. The defenses against the two are different as maliciously masquerade is not random. The defenses introduced previously, i.e. cryptographic hash functions and source / destination identifiers, also protect against masquerade.

With these additions the frequency of accidental masquerade is reduced to *incredible* and the frequency of malicious masquerade is reduced to *improbable*. The consequence of masquerade cannot be reduced. The risk is thus reduced to *negligible* and *tolerable* with respect to accidental and maliciously introduced hazards respectively.

3.7.8 Summary

In the previous sections, options for reducing the risks have been identified and specific risk reduction techniques have been chosen. Based on these measures the resulting risks has been identified, these risks are summarized in Table 3.5.

	Delay	**Loss**	**Corruption**	**Resequence**
Accidental	*Tolerable*	*Negligible*	*Negligible*	*Negligible*
Malicious	*Tolerable*	*Tolerable*	*Negligible*	*Negligible*

	Insertion	**Repetition**	**Masquerade**
Accidental	*Negligible*	*Negligible*	*Negligible*
Malicious	*Negligible*	*Tolerable*	*Tolerable*

Table 3.5: Results of the Options Analysis.

In this analysis, the following defense techniques has been chosen to reduce risks:

- Use of multiple transceivers for frequency diversity
- Prioritization of traffic
- Time stamp and sequence numbers
- CRC codes and keyed cryptographic hash functions

- Retransmissions
- Selective recreation of packet order
- Source and destination identifiers

To assure safety, it has been assumed that the EVC puts the system in safe mode when the following events occur:

- A packet with the same sequence number is received twice
- A packet response is delayed for longer than a certain threshold set by the EVC

Chapter 4

Delimitation

The rest of this report focuses only on parts of the solution proposed in the previous chapter. The focus is placed on the impact of the lower layers of the protocol stack on the system.

The following is thus not considered

- Transceiver diversity – Which is not analyzed in order to reduce the complexity of the analysis.
- Prioritization of traffic – The effect of prioritization is deemed not to have a major impact on the performance of the system since it cannot increase the attainable throughput, only optimize the transmission of packets to favor safety critical packets.
- Selective recreation of packet order, Source/destination identifiers, time stamps, and sequence numbers – Which are not considered as neither are interesting to analyze due to their low complexity. Neither one has an impact on the availability of the system, except for a slightly increased packet size and processing delay. They are however assumed present which is a reasonable assumption considering the minimal overhead they are likely to produce.
- CRC codes – Though CRC is assumed to be present, it is not analyzed and thus it is assumed that the CRC in the MAC layer detects all corrupted packets. To allow for such operation, further CRC should be added.
- The cryptographic methods are assumed to be handled by IPsec, but the impact of this assumption on the protocol stack is not considered. The impact of IPsec on a similar connection is described in D4.1.

This leaves a system for analysis with focus on the impact of the lower layers on the availability and safety of the system. The following section starts by setting up a framework for the analysis in which the safety vs. availability trade-off is discussed. In this discussion the safety problem is mapped to a problem of availability which is simpler to analyze when the EVC reaction patterns are not known.

Chapter 5

System Analysis

In this chapter the system is analyzed in depth. First the modeling approach is described, then the system is divided into sub-models, and then the sub-models are described. The primary goal of this analysis is to set up a framework for the safety such that the availability of the system can be expressed.

The protocol stack for analysis which is dictated by the delimitation in the former chapter is shown in Figure 5.1. In the PHY and MAC layers 802.11b is used, then IP in the network layer and UDP on the transport layer. Then a safety layer for added safety-related functionality which is defined later in this section. On the top, the application layer protocol of the EVC and DMI. The details of the EVC-DMI protocol are to a wide extend unknown and the analysis is thus based on assumptions which are elaborated in the following sections.

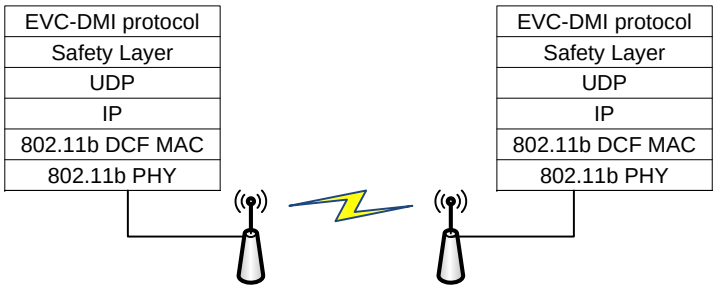

Figure 5.1: The analyzed protocl stack

5.1 Safety versus Availability

The definition of safety is a vital problem which arise in the analysis of the present application.

When observing the system in presence of a perfect delay and loss free communication system, it is obvious that the movement of the train introduces a certain amount of risk. The higher the speed becomes, the shorter the response time of the driver becomes and thus unexpected events become more dangerous. E.g. if a car is locked between the gates of a level crossing, the driver initiates the braking procedure but if the speed is high, the braking is a slow procedure, causing the train to hit the car. This kind of event

is outside the reach of the communication link as it should not interfere with application layer functions such as restricting the speed of the train or warning the driver.

Additionally it should be noted that the train might be going so fast that the driver does not get any useful information from looking out the window. In such a case it is not apparent what the exact role of the driver is because the EVC is presented with the same information as the driver while being able to perform extremely fast responses compared to the driver. In these conditions the EVC receives information about the track conditions ahead from short-range wireless transmitters embedded in the track (called Balise). At low speeds however, the driver is useful for navigating the train in ways that the EVC cannot; in this environment the human "sensors" are more useful.

Investigating the information shown to the driver, there is bound to be different classes ranging from simple non time-critical information like the distance to the next scheduled stop to the ambient temperature to highly critical information such as the current speed of the train or warning signals. Obviously the impact of errors in the communication protocol has variable impact depending on the type of message transmitted. But also within each class there is variations. E.g. if the speed of the train is reported too low to the driver as a result of a packet loss or corruption, then the driver might decide to increase the speed even further which increases the risk. In the opposite case the driver unnecessarily decreases the train speed which is unlikely to be risky.

Adding to this, in some situations, a loss of a packet or the display information might not be dangerous at all. E.g. while the train is at a stand-still or going at a low speed the loss of a display update message is less time critical.

Deciding when the system is safe is not a problem which is easily solvable by means of analyzing the communication system. The responsibility of keeping the system safe should be placed on the EVC. In order for the EVC to decide if the system is safe or unsafe such that it can bring the system into safe mode, the EVC must be presented with the relevant information from the communication link. Based on the delimitation of the problem, only delay and loss of packets are considered. In the following, a retransmission mechanism in the safety layer protocol is introduced, this eliminates packet loss, then introduction of application layer retransmissions is analyzed further in section 5.10. Thus only delay is a variable under the stated assumptions.

Considering such a system, the responsibility of keeping the system safe has been shifted from the communication system to the EVC where it is more easily managed. This means that the system can be assumed to be safe, given that the delay information is passed to the application layer, and that the EVC takes proper precautions.

In section 2.2.2, the different message types in the EVC-DMI protocol was described. Each of these message types (information objects) can contain an abundance of different information. As mentioned earlier in this chapter, this means that the different messages passed might have different levels of urgency and thus cause hazards after variable amounts of time. It is thus necessary for the safety layer protocol to pass information about every single package to the EVC in order for it to ensure the safety of the system. If this is done, the system can be analyzed for a generic packet, the content of which is unknown while still maintaining safety. For simplicity, the analyzed packets are assumed all to have the maximal telegram size in PROFIBUS which is 236 bytes (see section 2.2.2).

Under these assumptions, the goal of the following analysis is shifted from safety to availability since this is the central parameter of importance.

5.2 Modeling Availability

The system availability is a measure of how often the system is able to perform the tasks which it was constructed for. Obviously this figure should be quite high such that a number of missions can be run without interruption. It should be noted here that the availability discussed here is the availability of the modeled subsystem, the communication system, the measure derived here is thus not descriptive of the entire train system.

In order to provide the delay information needed by the application layer, each packet must be acknowledged in the safety layer. This can be done in two ways:

1. Periodic messages detailing past deliveries
2. Acknowledgment of every packet

Both has advantages and disadvantages. Option 1 has the advantage of taking up the least airtime in the physical layer due to fewer packets being sent and thus less overhead. However this approach leads to additional delay of the acknowledgment for the individual packets. Option 2 uses more airtime, but the acknowledgment is transmitted immediately. The choice is thus a delay vs. bandwidth trade-off. Since a lower delay is essential in this application, and the increase in bandwidth consumption is moderate, option 2 is selected.

Each packet transmission is thus acknowledged as shown in Figure 5.2. An information object (INFO) is sent to the DMI which responds with an acknowledgment (INFO_RECEIVED). Also shown in the figure is the incoming messages from other devices (e.g. the speed of the train). Based on this, it is assumed that the EVC puts the system in safe mode if a packet is delayed for longer than some threshold t_s. This means that the system is safe given that a proper value for t_s is chosen since the EVC puts the system in safe mode after this delay causing the system to become unavailable.

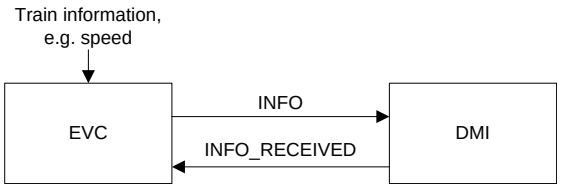

Figure 5.2: Message flow between EVC, DMI, and auxiliary devices.

Figure 5.3 shows a state space model of the system from the application layers perspective. In the neutral state 0, the EVC is waiting for data from the train sensors, when this arrives, a timer is started and the transition to state 1 is taken. In state 1 the packet is being delivered by the MAC layer, when the transmission is complete the model transitions to state 2 where the packet INFO_RECEIVED is being transmitted. When it has been transmitted by the MAC, the state machine transitions to state 0 again, ready for a new piece of data from the train.

However if while in state 1 or 2, the timer at the EVC runs out, the EVC goes into safe mode. If the system is put in safe mode from state 1, the safe mode is necessary since there indeed has existed an inconsistency between data available to the EVC and DMI for longer than t_s. However if the EVC goes into safe mode from state 2, there is no inconsistency between the EVC and DMI data, but the acknowledgment has not yet arrived to confirm this.

This model is insufficient in its description of the system as it tells very little about the complicated architecture which causes the transitions in the model. The impact of these functions are important to

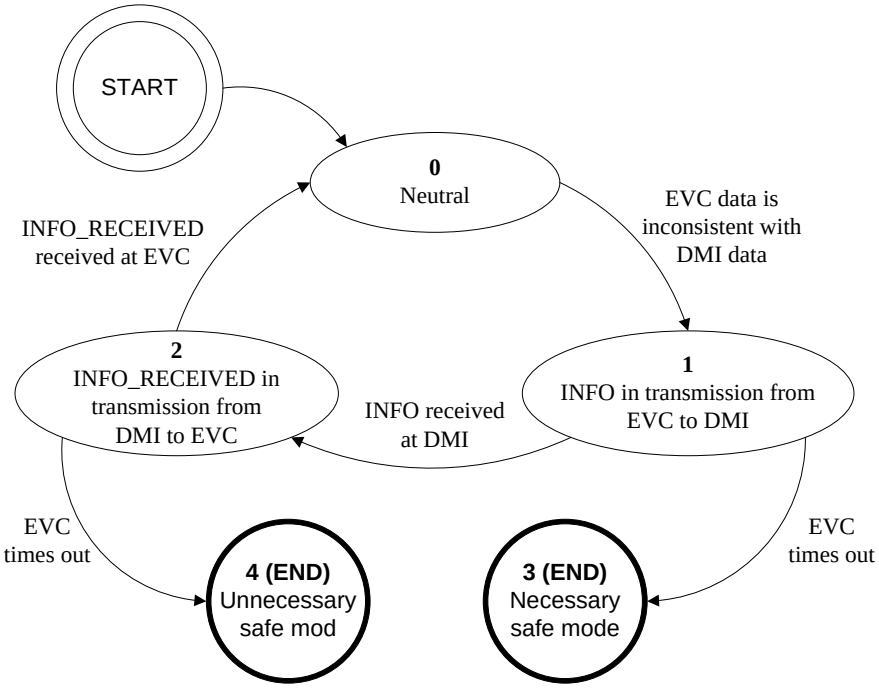

Figure 5.3: Early system model.

model such that the contributions of each part on the availability can be identified. Thus it is necessary to model the system in a separate sub-models, the following section describes this sub-division.

5.3 Sub-models

This section describes the division of the system model in sub-models. A sub-model is defined by its function, and by the interfaces to the other sub-models. In each model, a number of standard parameters are defined, these values are assumed in models described later.

The system model is divided into logical models using the terminology of the OSI reference model [ITU-T, 1994], though not all layers are present and some layers are sub-divided for further clarity.

The physical layer (PHY) is the lowest layer of the OSI-model, this layer comprises the medium of communication and the signals transmitted through the medium. This layer is divided into four models:

1. Wireless Medium Model
2. Frame Error Rate Model
3. Transmission Time Model
4. Interference Model

Each model representing a logical subdivision of the physical layer.

1) sets up a model for the medium itself; the model primarily addresses propagation effects and the received signal quality. The variables on the input side is the distance between the EVC and DMI, and on the output side a Signal-to-Noise Ratio (SNR) which describes the quality of the received signal.

2) sets up a model for the Frame Error Rate (FER) in the physical layer, which is dependent on the SNR. The input variable is the modulation type and the output is is the FER as a function of the packet length passed down by the MAC layer.

3) models the time used on transmissions in the physical layer.

4) models the other nodes which are bound to be present in the train, their traffic patterns and how many they are.

The MAC layer is divided into two models, one modeling contention, specifically the collision probability, and the other the delay.

Finally the availability model models the remaining layers: IP (network layer), UDP (transport layer) and part of the application layer denoted the safety layer, the other part of the application layer being the "black-box" EVC-DMI protocol.

Figure 5.4 shows the sub-blocks of the system model and the interfaces between them. The arrows represent in- and output variables for each block.

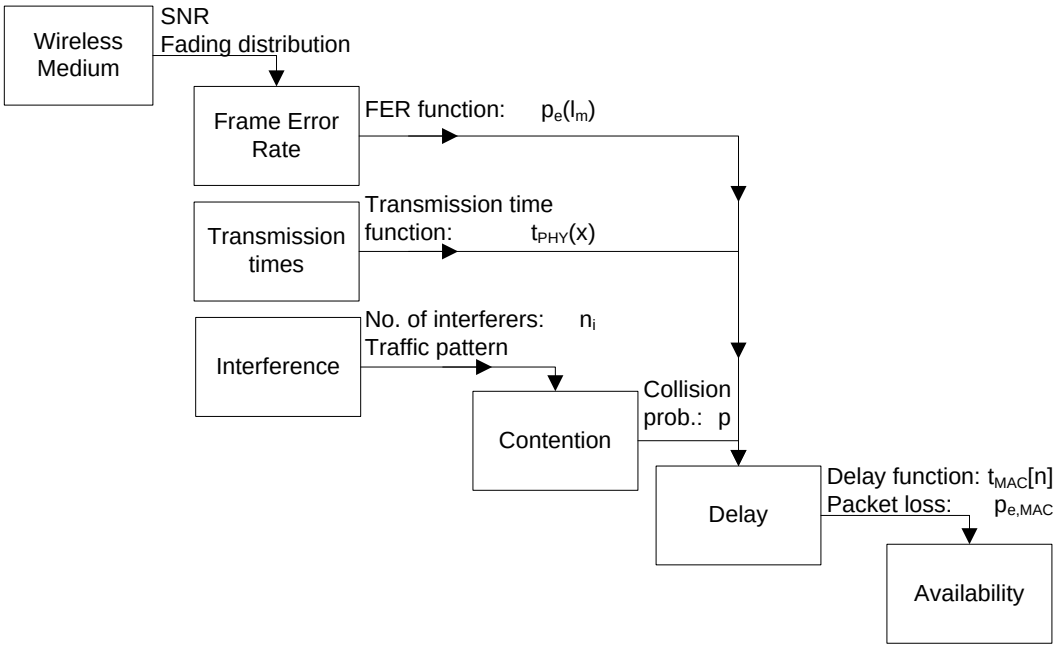

Figure 5.4: The system sub-models with interfaces.

5.4 Wireless Medium Model

The wireless medium model describes the wireless medium through which the radio waves propagate. This sub-model incorporates sender and receivers including antennas and filtering. The model maps

distance to Signal-to-Noise Ratio (SNR).

The model only treats the cases when no other nodes are transmitting, section 5.7 treats the cases where interferers are transmitting under the assumption that collision always results in the affected frame being dropped which is a worst-case assumption. Because of the MAC protocol, most of the time there is only one on-going transmission. It thus makes sense to make a special model for this special case, as opposed to a general, more complicated model.

The impact of the channel on the signal can be divided in three contributions: path loss, fading, and noise. Path loss is large scale attenuation which is caused by the the spherical dispersion of the radio signal over distance, while fading is caused by reflection of the signal by objects in the environment. Noise is unwanted signals in the receiver, in this report only thermal noise is considered. By definition, the average SNR at the receiver is given by the path loss and the noise power, while the instantaneous SNR (which fluctuates over time) is caused by both path loss, noise and fading. Thus the i'th received symbol in complex baseband representation is

$$y[i] = x[i]Lh[i] + w[i] \tag{5.1}$$

where $x[i]$ is the transmitted symbol which is assumed, without loss of generality, to have zero mean and an average power of unity, L is the path gain, $h[i]$ is the fading component and $w[i]$ is the noise which is circularly symmetric white Gaussian noise with zero mean and power N_T.

While path loss is the generally used notion in the literature, however this section uses the notation of gains which is the reciprocal of the loss. The path gain is defined as the gain between the sender and receiver including antennas. With the expression from [Friis, 1946], slightly rearranged as in [Haykin, 2001, p. 522], and generalized to environments with higher (or lower) attenuation than free space, it is given by

$$L = \frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d} \right)^{\alpha} \tag{5.2}$$

where G_t and G_r is the gain of the transmitting and receiving antenna respectively, λ is the wavelength, α is the path loss exponent, and d is the distance. In vacuum, the wavelength is given by

$$\lambda = \frac{c}{f} \tag{5.3}$$

where $c = 299,792,458\,\mathrm{m/s}$ is the speed of light, and f is the frequency.

The path loss exponent α is dependent on the environment in which the signal propagates. [Seidel and Rappaport, 1991] describes measurements of this value (denoted as n) in the 914 MHz band in a walled office environment. In the measurements the path loss exponent has values between 2.68 and 5.22, with an average value in the measured locations of 3.14. [Giannopoulou et al., 2000] also treats the path loss, though in the 2.4 GHz band, with similar results. Obviously the path loss exponent is highly dependent on the environment, so a measurement in the target environment would be beneficial, this is however not possible.

Considering the environment of the train, it is assumed that there is no line of sight between sender and receiver, this situation can be modeled by the Rayleigh fading model. In the the model, it is assumed that signal reflections take a large number of different paths, which changes randomly in time, before reaching the receiver. Each received component has a unique phase, because of the different distances traveled, and zero mean as the original signal. In some cases the signals adds up constructively at the receiver causing a gain larger than one (amplification), while in other cases they add up destructively causing gain smaller than one (dampening).

If the number of signal components received is sufficiently large, the central limit can be applied yielding the Rayleigh distribution:

$$p_r(x) = \frac{x}{\sigma^2} \exp - \frac{x^2}{2\sigma^2}, \quad \text{for } x \in [0, \infty[\tag{5.4}$$

which is the probability distribution of the fading function $h(t)$ in equation (5.1).

In the receiver, noise is added to the received signal, external noise sources such as sky noise is assumed to be negligible, and thus only thermal noise is considered. Thermal noise arises from the random motion of electrons which occur in any conductor at temperatures above absolute zero (0 Kelvin). The phenomenon was first described in [Johnson, 1928] and later explained in [Nyquist, 1928]. The theory asserts that any electronic circuit "generates" noise with power which is proportional to the temperature of the electronics. Most importantly, the power of the noise generated is independent of any other parameter of the system.

The thermal noise in a system is given by [Stelzried, 1968]

$$N_T = k_B T B \tag{5.5}$$

where $k_B = 1.380650e-023 \frac{\mathrm{J}}{\mathrm{K}}$ is Boltzmann's constant, T is the temperature of the circuit, and B is the bandwidth.

The amplifier in the receiver contributes to the noise as well, this means that the input to the amplifier has a higher SNR than the output signal. This degradation is represented by the noise figure, which is a quantification of the degredation of SNR between the input and output of the amplifier. It is given by

$$F = \frac{SNR_o}{SNR_i G} \tag{5.6}$$

where SNR_i and SNR_o is the input and output SNR respectively, and G is the amplifier gain.

Combining previous equations in this section, the received SNR is given by

$$SNR = \frac{P_t L F}{N_T} \tag{5.7}$$

Figure 5.5 shows the model SNR as a function of distance, also plotted is comparative measurements from the testbed, see Appendix A. The measurements where made with TRENDnet TEW-443PI interface cards, thus parameters from this card is used in the calculated SNR as well. The path loss exponent is set to $\alpha = 3.14$ based on the previosuly mentioned [Seidel and Rappaport, 1991].

The deviation between measurements and calculations are moderate, the largest being at 2.2 m at 8.2 dB.

In the following chapters, unless otherwise specified, the following parameters are assumed when this model is used: Send power $P_t = 18\,\mathrm{dBm}$, sender and receiver antenna gain $G_t = G_r = 2\,\mathrm{dBi}$ (dipole antenna), carrier frequency $f = 2.45\,\mathrm{GHz}$, temperature $T = 290\,\mathrm{K}$, sender and receiver amplifier noise figure $F = 3.7\,\mathrm{dB}$, distance $d = 10\,\mathrm{m}$, and path loss exponent $\alpha = 3.14$.

This section described the wireless medium model which is a mapping from distance to SNR, the ratio between signal and noise power. In the following this relationship is mapped to bit loss rates.

5.5 Frame Error Rate Model

This section models the Frame Error Rate (FER), since the goal of the MAC protocol is to allow only one node to transmit at any given time, it makes sense only to model the collision free periods. The collision

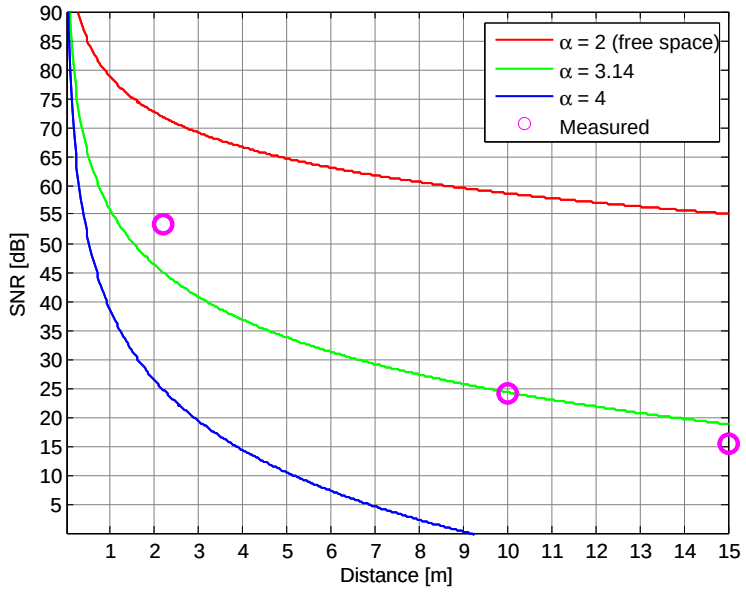

Figure 5.5: Calculated and measured (see Appendix A) SNR vs. distance for different path loss exponents. Parameters corresponding to a TRENDnet TEW-443PI 802.11b adapter [TRENDnet, 2007] which was used for the measurements. $P_t = 18\,\text{dBm}$, $G_t = G_r = 2\,\text{dBi}$ (dipole antenna), $f = 2.45\,\text{GHz}$, $T = 290\,\text{K}$, $F = 3.7\,\text{dB}$, $\alpha = 3.14$. MATLAB file: `do_medium.m`.

periods are handled by the model described in section 5.7 in which it is assumed that a collision always results in a packet loss.

As described in section 2.5.2, 802.11b supports four different data rates supported by different modulation schemes. The lowest rate of 1 MBit/s (called the basic rate) is always used for transmitting the PLCP (Physical Layer Convergence Protocol) header while the rest of the packet can be transmitted at any rate. All modulations are spread spectrum modulations, for the lowest two rates Barker coding is utilized while the highest two rates use CCK. In both cases the information bits are mapped to a smaller number of bits, i.e. a sequence of information bit is mapped to a codeword for transmission which is called a chip sequence. The number of information bits per chip for the rate r is denoted by

$$k_r = \frac{n_c}{n_i} \tag{5.8}$$

where n_c is the length of the chip sequence, and n_i is the number of information bits contained in the chip sequence. Since the energy of one chip is spread over multiple bits, the energy per bit is given by [Proakis, 2001]:

$$E_b = E_c k_r \tag{5.9}$$

where E_b is the energy per bit. Table 5.1 shows the coding rates for the different transmission speeds.

Transmission speed	Modulation	Code rate
$r_{1M} = 1\,\text{MBit/s}$	DBPSK with Barker coding	$k_{1M} = 11$
$r_{2M} = 2\,\text{MBit/s}$	DQPSK with Barker coding	$k_{2M} = 5.5$
$r_{5.5M} = 5.5\,\text{MBit/s}$	DQPSK with CCK	$k_{5.5M} = 2.75$
$r_{11M} = 11\,\text{MBit/s}$	DQPSK with CCK	$k_{11M} = 1.375$

Table 5.1: Properties of the available transmission rates [ANSI/IEEE, 2003].

The energy per chip vs. noise spectral density ratio is given by:

$$\frac{E_c}{N_0} = \text{SNR}\frac{B}{f_b} \tag{5.10}$$

where f_b is the bit rate and B is the bandwidth. The energy per bit vs. noise spectral density is then

$$\frac{E_b}{N_0} = \frac{E_c}{N_0} k_r = \bar{\gamma} \tag{5.11}$$

the $\bar{\gamma}$ notation is used for shorthand in the following.

In the following the Bit Error Rate (BER) for the different modulations is expressed from the $\frac{E_b}{N_0}$ under assumption of optimal reception and detection, the BER formulas are as such approximations. Every formula shown here takes Rayleigh fading into account as assumed in section 5.4. None of the modulation employ Forward Error Correction (FEC).

The basic rate r_{1M} is transmitted with Differential Binary Phase Shift Keying (DBPSK) with barker coding, the BER for DBPS is given by

$$p_{b,1M} = \frac{1}{2(1+\bar{\gamma})} \tag{5.12}$$

The r_{2M} rate is transmitted using the same Barker sequence, but over Differential Quadrature Phase-Shift Keying (DQPSK) which doubles the bit rate. The BER is given by:

$$p_{b,2M} = \frac{1}{2}\left[1 - \left(\sqrt{\frac{(1+2\bar{\gamma})^2}{2\bar{\gamma}^2} - 1}\right)^{-1}\right] \tag{5.13}$$

The $r_{5.5M}$ and r_{11M} rates are transmitted using CCK modulation over DQPSK, the only difference is the amount of bits encoded in each CCK sequence. The BER is given by

$$p_{b,\mathrm{CCK}} = \frac{2^{k-1}}{2^k - 1}\sum_{m=1}^{M-1}\frac{(-1)^{m+1}\binom{M-1}{m}}{1+m(1+k\bar{\gamma})} \tag{5.14}$$

where M is 4 for $r_{5.5M}$ and 8 for r_{11M}, and $k = \log_2 M$.

Figure 5.6 shows a plot of the BER vs. SNR for these formulas.

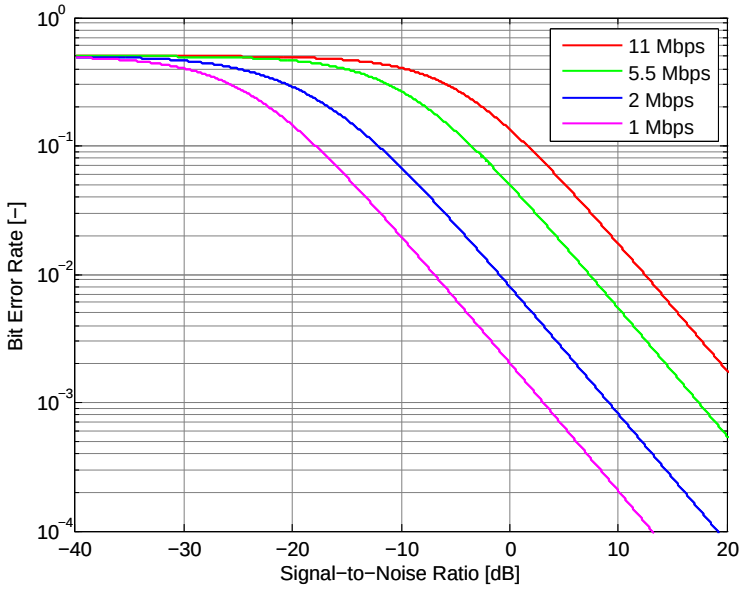

Figure 5.6: Calculated approximate Bit Error Rate vs. SNR under assumption of optimal receivers, based on Equations (5.12), (5.13), and (5.14). $B = 22\,\mathrm{MHz}$. MATLAB file: `plot_ber_vs_snr.m`.

Assuming that errors are independent and identically distributed, the FER for a frame of length n is given by

$$p_e = 1 - (1 - p_b)^n \tag{5.15}$$

The PLCP header has a length of $l_p = 24\,\mathrm{bytes} = 192\,\mathrm{bits}$, and is sent at the basic rate (r_{1M}), the payload portion, which is passed down from the MAC-layer, is sent at any rate r. A frame is thus lost with probability

$$p_{e,PHY}(l_m) \quad = \quad 1 - (1 - p_{b,1M})^{l_p}(1 - p_r)^{l_m} \tag{5.16}$$

where l_m is the length of the frame which is passed down from the MAC.

Figure 5.7 shows the maximal throughput as a function of the length of the frame passed down from the MAC.

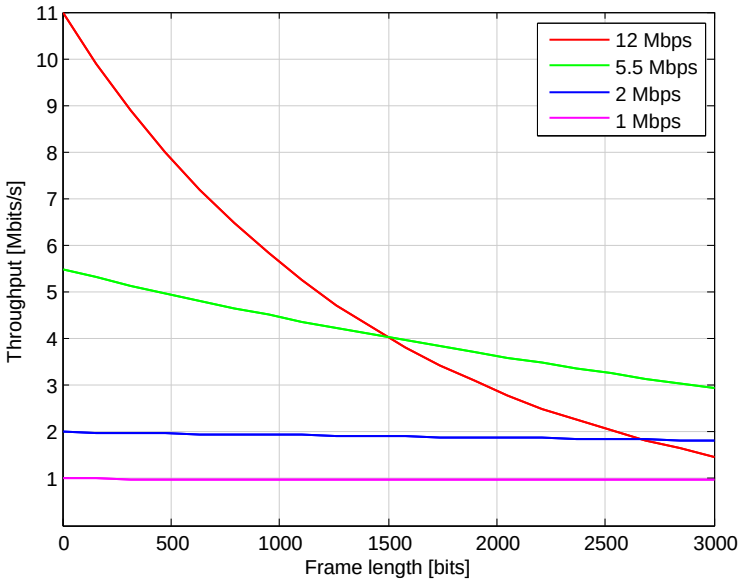

Figure 5.7: Calculated maximal throughput vs. payload length (l_m) based on equation (5.16) with input SNR calculated from the wireless medium model with standard parameters (most notably a distance of 10 m) as described in section 5.4. Note that the payload is the full data received from the MAC layer including MAC headers. The PLCP header length is $l_p = 144+48 = 192$ bit, and the throughput is calculated as $S = r(1-p_{e,PHY}(l_m))$. MATLAB file: `plot_fer_vs_packet_length.m`.

$B = 22$ MHz and $l_p = 192$ bit are defined as a standard parameter. In later sections it is assumed that the packet length which is passed to the MAC has a length of 2432 bits, thus the optimal rate is $r = 5.5$ Mbit/s which is also a standard parameter.

5.6 Transmission Time Model

This model addresses the time used by the wireless medium for transmissions. The MAC layer models described in the following section uses this as an input in order to simulate the total delay seen by application layer packets.

Each transmission is prefixed with a PLCP header with a length of $l_{PLCP} = 192$ bits. The PLCP header is always transmitted at the basic rate, i.e. $r_{1M} = 1$ Mbit/s. The transmission of the payload (from the PHY layers perspective) is conducted at the speed r which is any one of the possible speeds, i.e. r_{1M}, r_{2M}, $r_{5.5M}$, or r_{11M}. The total time used for sending a PHY packet is thus:

$$t_{PHY}(n) = t_{PLCP} + t_{payload}(n) \quad (5.17)$$

$$= \frac{l_p}{r_{1M}} + \frac{n}{r} \quad (5.18)$$

Figure 5.8 shows a plot of this equation.

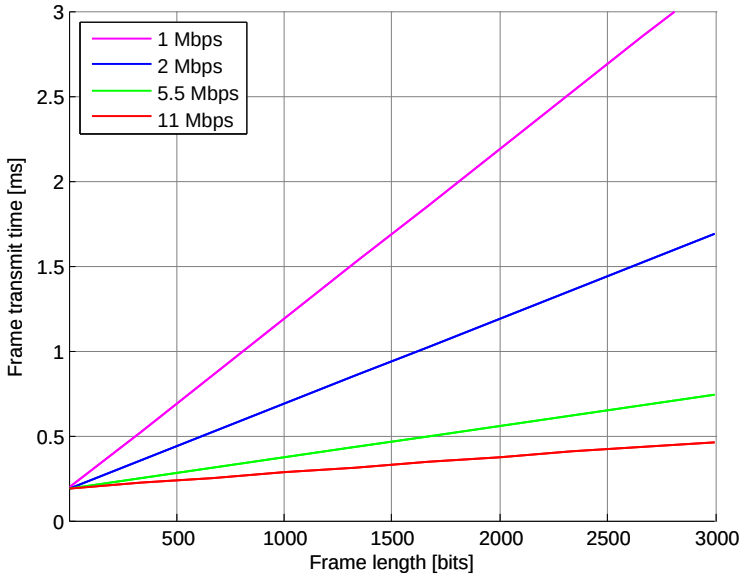

Figure 5.8: Calculated send time vs. payload length based on equation (5.18). Each line corresponds to a different rate r, the fame length n is varied as the x-axis. MATLAB file: `plot_send_time_vs_frame_length.m`.

5.7 Interference Model

This section describes a model of the interferers which are present in the target environment. It defines the quantity of nodes present, their transmission patterns, and their effect on the analyzed connection when transmitting.

The passengers of the train must be assumed to also use 802.11 WLAN, either for connecting to each other, or for connecting to in-train hot spots which are currently being introduced on some railroad stretches in Denmark [DSB, 2008a] and probably also in many other countries.

The nature of 802.11b hot spot traffic is examined in [Na et al., 2004] in which the traffic through a restaurant hot spot is analyzed. The vast majority of traffic which passes through the access point is HTTP. Most packets are either close to the minimum size, or close to the maximum size i.e. inbound (from AP to user) $\approx 40\,\%$ are shorter than 100 bytes and $\approx 40\,\%$ are 1500 bytes, outbound $\approx 80\,\%$ of the packets are shorter than 100 bytes. This is consistent with HTTP traffic in which small request packets are sent outbound and responded to by large TCP packets. This traffic pattern is possibly comparable to that exhibited by users of a train. This report will however not dwell further on the anthropological aspects of train users WLAN usage but instead simply assume that the traffic consist solely of HTTP traffic.

Assuming that each user requests a 100 KByte web page every 30 seconds, the average traffic from each user is 3.33 KByte/s. It is also assumed that the requests occur as a Poisson arrival process, this is a reasonable assumption since users are likely to act independent and randomly, though realistic traffic is likely to be more bursty.

All interfering nodes are placed inside the train, and it is assumed that only seated passengers are using

the WLAN. This is more probable in high speed trains since all passengers must have seat reservations for safety reasons. The layout of the train is assumed to be that of a Danish IC3 train which is shown in Figure 5.9. The total length of a train is 60 m and it is assumed that the train is full (144 seated passengers) and that the passengers are distributed uniformly on the length axis of the train, i.e. 48 evenly spaced passengers per car. The DMI is placed in the front of the train with the EVC approximately 10 m away. This means that the EVC is most affected by interferers, and the DMI less so, however for simplicity this analysis considers the worst case, i.e. the EVC as receiver, and it is then assumed that the DMI sees the same collision domain. It is also assumed that nodes outside the train do not interfere with the EVC–DMI communication.

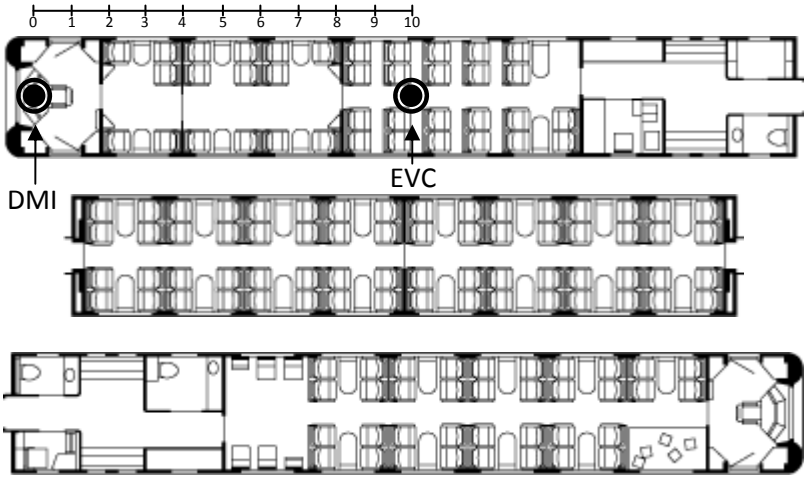

Figure 5.9: Floor plan of a Danish IC3 train [DSB, 2008b]

802.11 has 13 channels with 5 MHz separation, and a width of 22 MHz. This means that each channel overlaps with up to 8 adjacent channels. E.g. channel 1 overlaps channels 2, 3, 4, and 5, channel 5 overlaps channels 1, 2, 3, 4, 6, 7, 8 and 9. Table 5.2 shows the channel overlap and the attenuation of a signal between the channels when assuming that the channel energy is distributed evenly across the entire channel width.

Channel tier	Self	1st	2nd	3rd	4th	5th and beyond
Overlap [MHz]	22	17	12	7	2	0
Overlap [%]	100	77	55	32	9.1	0
Attenuation [dB]	0	1.1	2.6	5.0	10	∞

Table 5.2: Channel overlap with adjacent channels.

It is assumed that interference from other channels are white noise which is a worst-case assumption from an information-theoretic perspective since white noise has infinite information content. It is further assumed that transmissions by other nodes start and end at the same time as EVC–DMI communication which is in optimal assumption since then one transmission by another node can only conflict with one transmission, thus minimizing the impact. It is further assumed that only one node transmits at any time in a channel which is an optimal assumption.

Figure 5.10 shows the throughput as a function of distance between receiver (EVC) and sender (the interferer node), the distance between EVC and DMI is fixed at 10 m. The plot only shows the throughput

during a collision with one node in another channel. The thick line shows the mean throughput, i.e. the average throughput when nodes from the interfering tiers alternate in interfering with equal probability. The mean throughput is above 50 % of the maximally posible throughput at 44.69 m from the DMI.

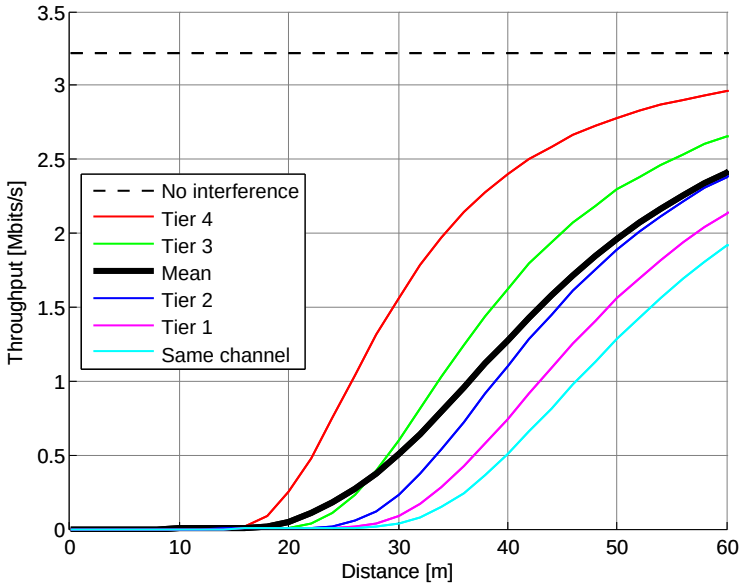

Figure 5.10: Throughput vs. distance for different interference tiers, made with the models described in the earlier sections of this chapter with their standard parameters. MATLAB file: `interferer_test_fixrate.m`.

Based on this, it is assumed that any node within 44.69 m can interfere, if it is transmitting on one of the interfering channels. This means that $\frac{44.69}{60}144 \approx 107$ nodes can interfere when the DMI is receiving, and $\frac{44.69+10}{60}144 \approx 131$ nodes when the EVC is receiving. Regarding only the worst case, 131 nodes are in the collision domain.

Figure 5.11 shows the average number of nodes in the collision domain of each channel as a result of uniform random distribution of 144 nodes across the available channels. Each node is place randomly in one of the channels, with all nodes placed, the number of nodes in the collision domain (the adjacent channels and the channel itself) are counted.

The best choice of channel is one of the outermost channels (channel 1 or 13) on which 50.4 nodes on average are in the collision domain. With the traffic per node previously calculated to be 3.33 KByte/s the total traffic becomes 168 KByte/s.

5.8 Contention Model

This section and the following describes the MAC layer model. The model chosen was proposed in [Bianchi, 2000]. While this section addresses the probabilities of collision, the following section describes the temporal behavior of the protocol which is dependent on the collision and loss probabilities.

The contention model is divided in two parts with different modeling approaches. One part of the model describes the other nodes which are in contention with the EVC-DMI communication, the other part

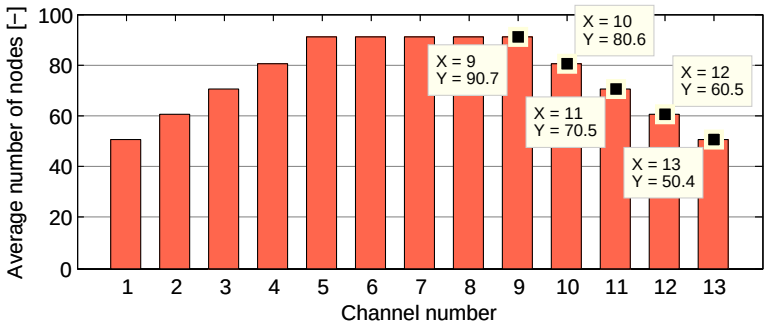

Figure 5.11: Average number of interfering nodes in each channels collision domain as a result of random uniform assignment of 144 nodes. 1000000 repetitions. MATLAB file: `channel_assignments.m`.

describes the EVC-DMI communication. The other nodes are described as the probability of transmission from any one of them in the steady state while the EVC-DMI communication model is a transient model with some important deviations from the Bianchi model. In the Section 5.9 the temporal aspects of the protocol is described and simulated. The advantage of the chosen approach as compared to an event driven simulation is the speed of which the simulation can be executed. During the project, a model of this kind was created (denoted `model4` in the MATLAB files on the CD), it could process around 252 packets pr. second while the present analytical-numerical model can process 32258 packets pr. second. The former simulation was performed on two cores while the latter was only performed on one core, this means a theoretical factor of 256 improvement.

The Bianchi model is a discrete time Markov chain model for saturated conditions. In the model, the contention process at each node is modeled by a two-dimensional random process: $(s(t),b(t))$, where $s(t)$ is a stochastic process representing the backoff stage, $b(t)$ is a stochastic process representing the backoff counter, and t is a the discrete time scale. Note that each step in this time scale means a progression of the model by one step, this step can take different amounts of time depending on the events of the overall system, it is thus not proportional to the "real world time". E.g. the discrete time is only progressed by one for each transmission, no matter how long the transmission is. The timing aspects of the model is explored further in section 5.9.

Figure 5.12 shows the states of the Bianchi model, where p is the probability that a packet collides, given that a transmission occurs. $W_i = 2^i W_0$ is the size of the i'th backoff step, and m is the highest backoff step. Note, that when the highest backoff step is reached, the backoff window is kept constant for future transmissions, and the packet is retransmitted until it is delivered, i.e. there is no retransmit limit.

Transmissions occur whenever the transition $(i,0) \rightarrow (0,j), i \in (0,m), j \in (0,W_0-1)$ occurs, the probability of this event is given in [Bianchi, 2000] as:

$$\tau = \frac{2(1-2p)}{(1-2p)(W_0+1)+pW_0(1-(2p)^m} \tag{5.19}$$

and the collision probability p is given as:

$$p = 1-(1-\tau)^{n-1} \tag{5.20}$$

where n is the number of nodes including either the EVC or the DMI. The two equations (5.19) and (5.20) form a non-linear system in two unknowns to which a unique solution exist which can be found numeri-

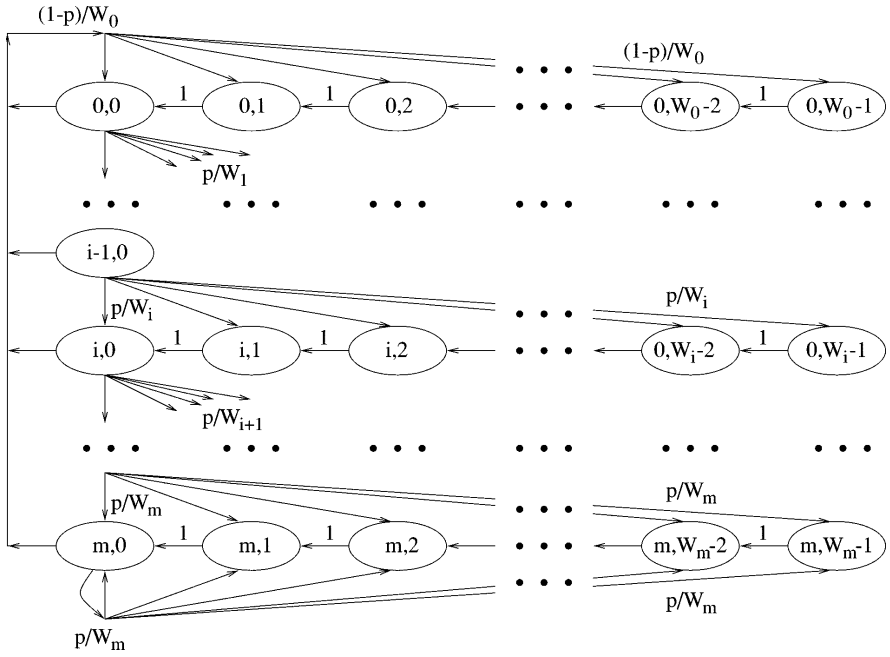

Figure 5.12: The states of the Bianchi model, figure from [Bianchi, 2000]

cally. Packet losses are not accounted for in this model, they can however be incorporated by expanding p with a packet loss probability as well.

The Bianchi model is designed to be an average throughput model for saturated systems. However in this model, it is used to describe the probability of other nodes transmitting at any point in time. The EVC-DMI communication, is modeled separately. In the analysis, the case where the EVC is receiving is the worst case since most other nodes are in the collision domain, so only this case is considered. Note also that it is assumed that all nodes are in the same collision domain and thus hidden and exposed terminal problems are not considered.

This means that $n-1$ nodes (the other nodes), follow the Markov chain of Figure 5.12, and that they see the DMI following this Markov chain as well. However, the DMI is modeled in such a way that the backoff chain is simulated with a finite number of retransmissions. This means that the model of the other nodes employs a different world view than the model of the DMI. This modeling choice was made to model the temporal behavior of the DMI in order to obtain a delay distribution, this is described in the following section. The effect of the inconsistency is that the value of τ in equation (5.20) does not represent the DMI, in fact the τ corresponding to the DMI should be lower resulting in a p which is too large. This causes the other nodes to tend more towards later steps in the backoff chain than should be the case. This effect is reduced when the number of nodes is increased.

Since the Bianchi model uses infinite retransmissions which is probably not what can be expected in actual systems, the stationary distribution of the other nodes tends slightly more towards a low backoff step since the collision probability is slightly lowered. The absence of channel errors in the model of the other nodes has the effect that the value of p is too low making the model tend towards the lower steps.

In all, there is three major modeling inconsistencies two has the effect of making the backoff chain of the other nodes tend towards the low steps in the backoff chain, and one making it tend towards later steps.

The assumed traffic dynamic from other nodes is HTTP which uses TCP in the transport layer, this means that packets are retransmitted without limit. However, a retransmission in the transport layer means resetting the backoff counter in the MAC-layer. The packets from the other nodes are transmitted following a Poisson process, however the Bianchi model is a saturated load model, thus it is necessary to map a number of Poisson arrival processes to a number of saturated, i.e. constant flow arrival processes. This mapping is made in the following section.

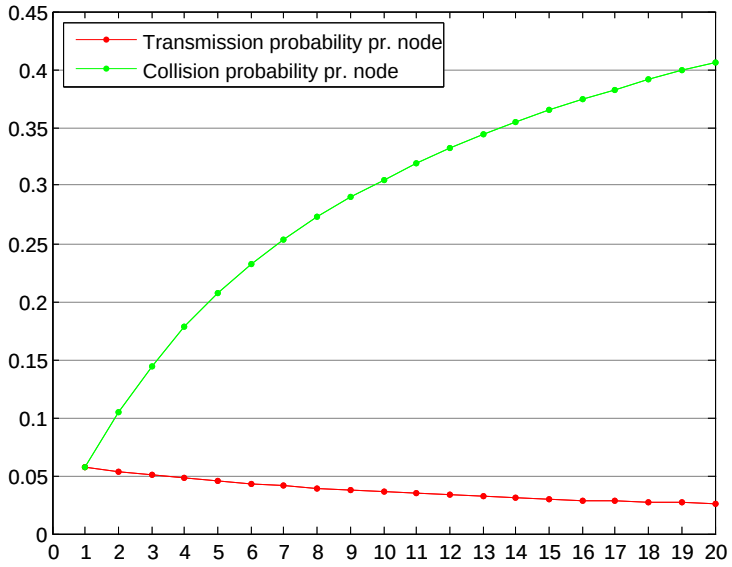

Figure 5.13: Calculated per node transmission and collision probability based on numerical solution of equations (5.19) and (5.20). The minimal contention window size is $CW_{min} = 32$ and the maximal is $CW_{max} = 1024$ based on the 802.11b standard, this corresponds to $m = 4$ backoff steps (the first being step 0 following the notation of the Bianchi paper). MATLAB file: `plot_mac_coll_prob.m`.

In the 802.11 standard, the minimal and maximal contention windows size is given as $CW_{min} = 32$ and $CW_{max} = 1024$ respectively. These values are used as parameters for this model.

5.9 Delay Model

This section describes a model of the delay seen by application layer packets. It models the 802.11 MAC protocol in a manner which is fast in simulation.

In section 5.5 the frame error rate in the physical layer was defined. The frame error rates seen by the MAC layer for data packets and acknowledgment packets respectively are thus

$$p_{e,d} = p_{e,PHY}(l_n + l_d) \qquad p_{e,a} = p_{e,PHY}(l_a) \tag{5.21}$$

where l_n is the length of the payload received from the network layer, l_d is the MAC header length, and l_a is the length of an acknowledgment packet. The error rate during a transmission is assumed to be $100\,\%$. The transmission rate is the optimal $r_{5.5M} = 5.5\,\mathrm{MBit/s}$.

The transmission time of data packets and acknowledgment packets respectively are given by

$$t_d = t_{PHY}(l_n + l_d) \qquad t_a = t_{PHY}(l_a) \tag{5.22}$$

where $t_{PHY}(x)$ is the transmission time of a packet of length x which is given in equation (5.18).

A packet delivery is defined as the successful transmission of both data packet and acknowledgment packet. In reality the data packet is delivered to the higher layers as soon as it arrives at the receiver, thus the chosen approach gives a higher delay, but only slightly since the loss rate of the acknowledgment is substantially lower than that of the data packet. This simplification is introduced to allow for calculation of the actual airtime usage which is necessary for calculating the obtainable throughput.

For simplicity, it is assumed that the other nodes transmit at the same rate and uses the same packet sizes as the EVC and DMI. With this assumption, a slotted time scale can be adopted for simpler modeling.

From the perspective of the sender, given that a packet transmission is attempted, four distinct events can occur:

1. With probability $p_c = p$, the packet collides with another transmission
2. With probability $p_d = (1-p)p_{e,d}$, the packet does not collide, but the data part is not received at the receiver
3. With probability $p_a = (1-p)(1-p_{e,d})p_{e,a}$, the packet does not collide, the data portion is received at the receiver, but the acknowledgment is lost
4. With probability $(1-p)(1-p_{e,d})(1-p_{e,a})$, the packet is received at the receiver and acknowledged to the sender and is thus delivered

Only the last event (4) results in a packet delivery meaning that with probability

$$p_t = 1-(1-p)(1-p_{e,d})(1-p_{e,a}), \tag{5.23}$$

a transmission is unsuccessful.

The number of events 1-3 during the transmission of a packet is given by Y_c, Y_a, and Y_d respectively. In the following they are addressed by their sum $Y = Y_c + Y_a + Y_d$. The distributions of Y_c, Y_a, and Y_d are thus given from the distribution of Y and the probabilities mentioned above such that e.g. a transmission failure is with probability $(1-p)p_{e,d}$ a collision.

If there is no retransmission limit, the number of transmissions needed to deliver a certain packet is denoted by the discrete random variable Y_g with the probability density function

$$f_{Y_g}(y) = \Pr(Y_g = y) = (1-p_t)p_t^{y-1} \tag{5.24}$$

which is identified as geometric distribution with success parameter $(1-p_t)$ and support $y \in 1,2,....$

Taking the retry limit n_{rl} into account, if a sample of Y_g is larger than the retry limit, the packet is dropped. The probability of this happening is

$$p_d = \Pr(Y_g > n_{rl}) \tag{5.25}$$

$$= p_t^{n_{rl}+1} \tag{5.26}$$

This is the packet drop probability seen by the network layer and as such it is descriptive of the portion of the packets which are dropped. The remainder of this section thus only addresses delivered packets, i.e. probabilities of delays given that the packet is delivered.

The number of transmission needed to successfully transmit a certain packet is given by the random variable Y, which is a limited version of Y_g which only takes values on the interval $\{1, n_r l\}$. A sample of Y is made by taking a sample of Y_g, if the value is larger than $n_r l$, it is thrown away and the process is repeated. If the sample is less than or equal to $n_r l$, the value is the sample of Y. The probability density function is thus

$$f_Y(y) = \Pr(Y = y) = \Pr(Y_g = y | y \leq n_{rl}) \tag{5.27}$$

$$= \begin{cases} \frac{(1-p_t)p_t^{y-1}}{1-p_d} & \text{for } 1 \leq y \leq n_{rl} \\ 0 & \text{otherwise} \end{cases} \tag{5.28}$$

$$= \begin{cases} \frac{(1-p_t)p_t^{y-1}}{1-p_t^{n_{rl}+1}} & \text{for } 1 \leq y \leq n_{rl} \\ 0 & \text{otherwise} \end{cases} \tag{5.29}$$

Before each transmission attempt, the nodes must contend for the medium, as previously described this is done by waiting a random number of time slots. Backoff counter for the i'th backoff step is given by the discrete random variable X_i. X_i is uniformly distributed on $(0, W_i)$ where W_i is the i'th maximal contention window size.

What remains is then the delay caused by other nodes transmitting, since another node transmits with probability p in each backoff time slot of the sender, the number of other transmissions in a given backoff period of X_i steps, is binomially distributed with trial parameter X_i and success parameter p. The number of transmissions in the i'th step is denoted by the discrete random variable K:

$$\Pr(K_i = k) = \binom{X_i}{k} p^k (1-p)^{X_i - k} \tag{5.30}$$

The total time needed for the transmission of a certain packet is then given by:

$$Z = T_s + Y_c T_c + Y_a T_a + Y_d T_d + \sum_{i=0}^{Y-1} (X_i t_{st} + K_i T_s) \tag{5.31}$$

where the times are given by

$$T_s = T_a = t_d + \text{SIFS} + t_a + \text{DIFS} \tag{5.32}$$

$$T_d = T_c = t_d + \text{DIFS} \tag{5.33}$$

and t_{st} is the slot time.

The minimal and maximal value of Z is given by

$$Z_{\min} = T_s \tag{5.34}$$

$$Z_{\max} = T_s + \sum_{i=0}^{n_{rl}-1} W_i (T_a + t_{st}) \tag{5.35}$$

To create the output of this model, the value of Z is simulated by a MATLAB script which is listed in appendix B.1. The simulation is run over n_{MAC} repetitions, and the output values are created by evaluating the samples of Z. For each packet in the simulation, random numbers are drawn to represent each of the random variables described in these section. From these random numbers the delay of each packet is calculated along with the different categories of airtime usage.

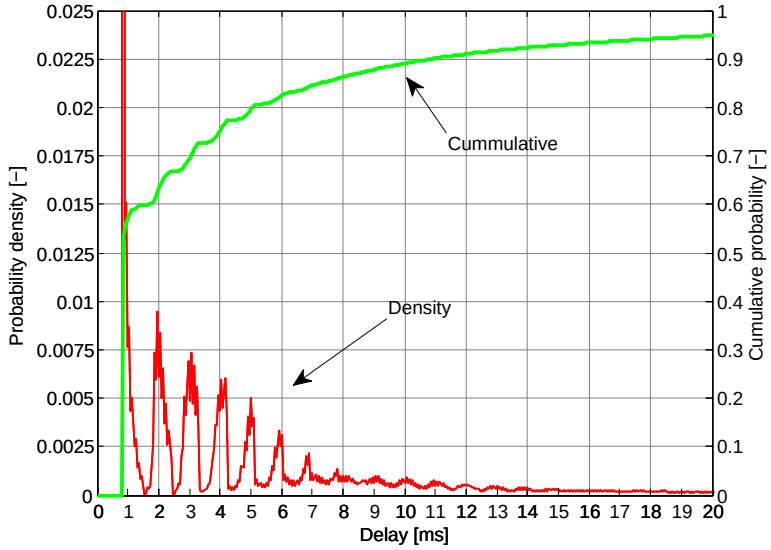

Figure 5.14: Simulated delay probability for intervals of 50 μs. Not shown in this figure is variations on the short time scale as each point is a sum over the time interval, i.e. due to the resolution of the graph. The simulation is based on 10,000,000 packets, the top of the first spike is not shown in the plot. The calculated minimal delay was $Z_min = 898$ ms and the maximally possible delay was $Z_max = 2.80$ s. MATLAB file: `model7/strict_models/plot_mac_probabilities.m`

Figure 5.14 shows the probability density function and the cumulative distribution function of the delay. From the graph it can be seen that the majority of packets are delivered within 1 ms.

Figure 5.15 shows the throughput as a function of the total number of nodes. The throughput is the application layer throughput.

Figure 5.16 shows the air utilization split in groups of similar events.

The throughput required to accommodate the needs of the other nodes is earlier assumed to be 168 KiByte/s = 1.376 Mbit/s, in reference to Figure 5.15, at least two nodes with saturated traffic must be used to model the other nodes. Assuming a total available maximal throughput of 1.5 Mbit/s, 794 packets can be transmitted per second. This leaves used 1.26 ms per transmitted packet while the application produces a packet every 1.37 ms, i.e. 0.92 packets are produced per transmit opportunity in average. Calculating the probable number of nodes attempting transmission in the top 95 % percentile, the cumulative distribution function for the Poisson distribution with $\lambda = 0.92$ shows that three nodes are active in these cases.

The following parameters are standard values from the 802.11 standard. Retry limit: $n_{rl} = 7$, data frame header size: $l_{dh} = 34$ bytes, payload size: l_n, data frame size: $l_d = l_{dh} + l_n$, ack frame size: l_a. These values are used as standard parameters for this model.

5.10 Availability Model

This sub-model describes the upper bound availability of the system.

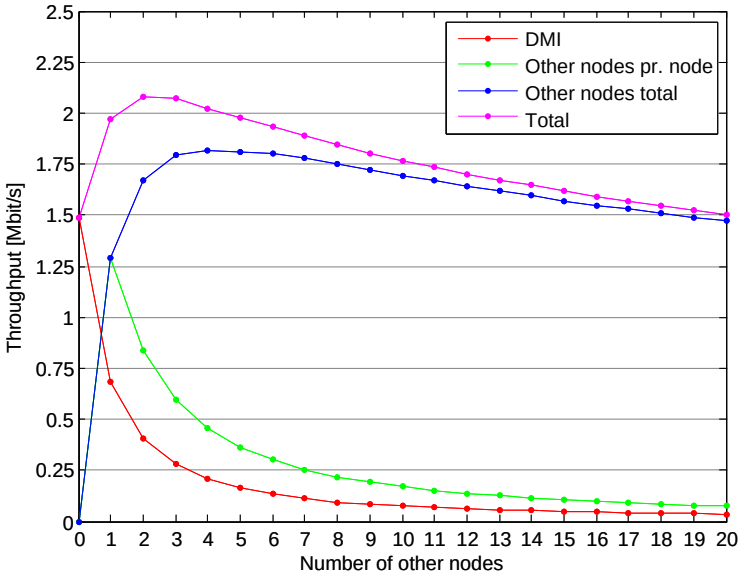

Figure 5.15: Simulated throughput vs. number of other nodes, 1,000,000 packets simulated. MATLAB file: `model7/strict_models/plot_mac_airtime_tp.m`

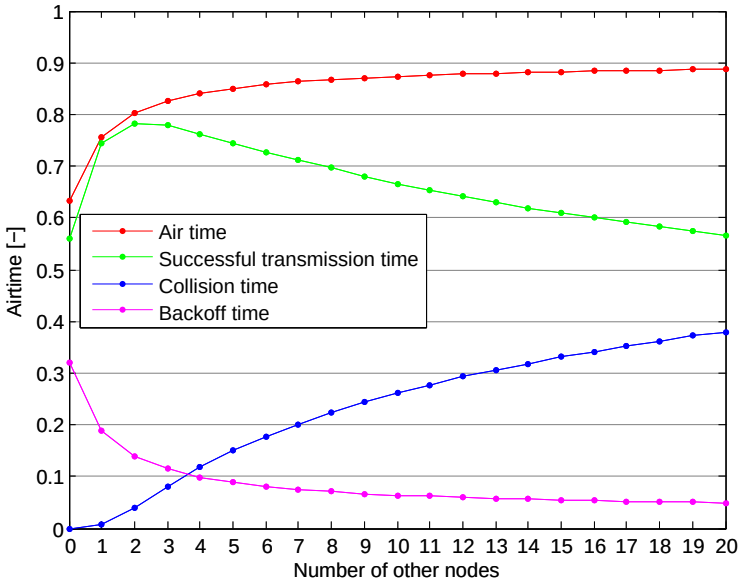

Figure 5.16: Simulated airtime vs. number of nodes. 1,000,000 packets simulated. MATLAB file: `model7/strict_models/plot_mac_airtime_tp.m`

As described in section 5.1, the basic assumption is that the system is always safe, given that a proper safety threshold is selected by the EVC. The delay model divides packets into two categories: lost and delivered packets. To improve availability, and to ensure that packet losses does not necessarily lead to safe mode, retransmissions are introduced in the safety layer. The retransmission can be initiated when the packet has been dropped by the MAC-layer, this is guaranteed to have happened after $2Z_{max}$.

A better solution is obviously using cross-layer mechanisms, i.e. using a unified stack to gather intelligence across the otherwise strictly defined layer boundaries. With this approach, the safety layer could be informed exactly at the point at which the packet is dropped. However this increases the complexity of the stack and makes integration with standard hardware and software more costly which is in conflict with the goals of this project, thus it is not considered further. Also, the retry limit in the MAC-layer could be increased to it's maximum of 256 retransmissions. It is not clear whether this option is implemented in standard hardware or drivers and as such it is not considered.

Since packets are lost with probability $p_{da} = 1-(1-p_{e,d})(1-p_{e,a})$, the acknowledgment is not received with probability $p_{d,sl} = 1-(1-p_{da})^2$. The number of retransmissions needed in the safety layer in order to deliver a packet is given by random variable A with probability density function

$$f_A(a) = \Pr(A = a) = (1-p_{d,sl})p_{d,sl}^{a-1} \tag{5.36}$$

which is identified as a geometric distribution with success parameter $1-p_{d,sl}$ and support $a \in \{1,2,\cdots\}$.

The random variable Z_1 represents the delay of the data packet from the sender to the receiver, and the random variable Z_2 represents the delay of the safety layer acknowledgment. Both are distributed with the input distribution from the MAC layer. The delay from transmission of the data packet to reception of the acknowledgment in the safety layer is given by:

$$B = A2Z_{max} + Z_1 + Z_2 \tag{5.37}$$

In order to calculate the upper bound on availability, queuing delays are disregarded. The per packet probability of going into safe mode is thus $\Pr(B > T_{sl})$. With n_p packets per hour, the probability of the system being available for an entire, randomly selected hour is then given by

$$p_a = (1-\Pr(B > T_{sl}))^{n_p} \tag{5.38}$$

Figure 5.17 shows a plot of the upper bound availability represented as the upper bound Mean Time Between Failures (MTBF) where a failure is defined as the system going into safe mode. The step-shaped function is a result of the safety layer retransmissions which occur with period $2Z_{max} \approx 5.6\,\text{s}$. A simple description of the figure is that it is the cumulative distribution function shown in Figure 5.14, repeated for every safety layer retransmission.

Figure 5.18 shows the upper bound MTBF as a function of the packet load in packets per second.

To obtain the lower bound or exact availability, queuing delays must be taken into account. This vastly complicates the model and is thus not considered.

The assumed load on the system is 12,600 packets per hour corresponding to 3.5 packets per second, this load is represented by the green line in Figure 5.17. Though the requirements for the availability of the system are unknown, the target MTBF must be assumed to be very large since a single part of the system cannot be allowed to bring down the entire system too often. Thus a MTBF of 1 every 10-100 years is probably a good estimate. This would require a safety threshold of around 23 seconds.

Such a safety threshold might be prudent for certain types of packets, e.g. messages to the driver about the temperature in the cabin. Information like speed and speed limits might be more urgent. Thus this

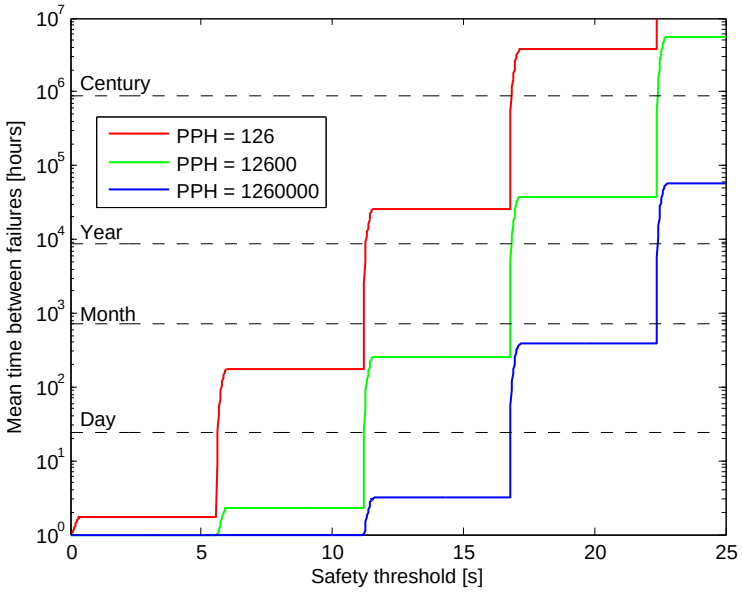

Figure 5.17: Calculated (from simulated data) upper Mean Time Between Failure (MTBF) for various loads, earlier assumptions defined a standard load of $3.5 \times 60 = 210$ packets per hour for the analyzed system. MATLAB file: `plot_availability_vs_thresh.m`

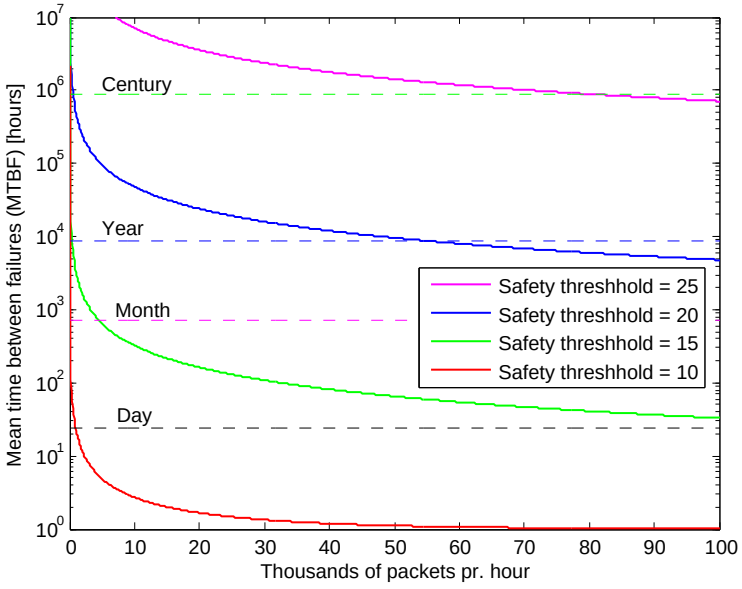

Figure 5.18: Calculated (from simulated data) upper bound Mean Time Between Failure (MTBF) vs. system load in packets per hour for various safety thresholds. MATLAB file: `plot_availability_vs_pph.m`.

graph cannot describe the availability of the train, only the communication link. However, with the long safety thresholds required to obtain high availability in the communication links, it is necessary for a very large portion of the transmitted packets to be non-time-critical. This seems as an unlikely condition in the high-speed trains of the target application, and the link thus seems like an unrealistic replacement for the PROFIBUS solution.

Looking at the graph shown in Figure 5.17, it is apparent that most of the time (in the flat sections of the plot), not much is going on. Based on the simulation which created Figure 5.14, there is a 99.999 % chance that a packet is delivered before 305 ms, thus the two way delay is with probability 99.998 %, the acknowledgment at is received at the EVC after 610 ms. So the retransmission in the safety layer could instead be initiated after this time or event earlier to allow for increased availability without substantial loss of throughput or increased delay of later packets due to queueing. This does however require additional analysis including a queueing model.

Chapter 6

Conclusion

This report presents an analysis of a wireless communication protocol based on Commercial Off The Shelf (COTS) hardware and software for use between the primary computer and the driver interface of a train. First the premise of the system was presented along with descriptions of possible solution technology. Then the risks associated with an assumed solution protocol were analyzed and a full solution framework was presented. The problem was then reduced in order to facilitate analysis of the most important sub-systems. Finally the proposed system was analyzed in depth through both analytical and numerical methods.

The basic assumption of the system analysis was that the EVC should be informed of the magnitude of delays and then it should guarantee the safety of the system. This meant transforming the analysis problem from focusing on safety to focusing on availability.

For the analysis the system was divided into a number of models representing logical divisions of the system. The physical layer was modeled through four models; the wireless medium model mapped the distance and physical parameters to an SNR. Based on the SNR the Frame Error Rate (FER) was calculated through a simple 802.11b model under assumption of optimal receivers. The transmission time model calculated the transmission time in the physical layer and the interference model set up a model for the number of interferers active in the environment. The Bianchi model was then used to calculate approximate probabilities of collisions which in the delay model was used for simulating the temporal behavior of the 802.11 MAC. In the availability model this simulated delay was used for calculating the upper bound availability of the system.

The analysis shows that the safety threshold under the stated assumptions must be above 23 seconds to ensure a proper level of upper bound availability. This means that the application layer protocol must accept a two-way delay of at least 23 seconds. The primary contributing factor to this large two-way delay is the required level of availability coupled with the choice of not allowing packets to be retransmitted in the safety layer before they can be guaranteed to have been dropped in the MAC layer.

Considering the time-critical nature of the analyzed application, the proposed wireless solution does not seem as a viable alternative to the PROFIsafe solution which is bound to offer much better delay performance because of the controlled nature of its physical medium.

There are however some options fro mitigation of this problem which can be asserted through further analysis. Introducing faster retransmissions in the safety layer would with high probability improve the availability dramatically, the analysis of such a solution does however require a queuing model placed between the delay model and the availability model. Such a queuing model is possibly computationally

heavy since the output delay distribution from the delay model is not a simple one.

With this addition, the primary cause of delays is probably going to be the interference caused by the passengers of the train. Since the amount of interference from other nodes is likely to vary based on the chosen channel, switching channels as a reaction to the level of interference might improve performance. This could be combined with the introduction of multiple radios for delay improving radio diversity. The effectiveness of these solutions depend highly on the correlation of traffic levels across the available channels which would have to be analyzed further.

Chapter 7

Bibliography

[Acromag Inc., 2002] Acromag Inc. (2002). Introduction to profibus dp. Technical report, Acromag Inc., Wixom, MI, U.S.A.

[ANSI/IEEE, 2003] ANSI/IEEE (2003). 802.11 1999 (r2003).

[ANSI/IEEE, 2005] ANSI/IEEE (2005). 802.11e-2005.

[Bell, 2006] Bell, R. (2006). Introduction to iec 61508. In *SCS '05: Proceedings of the 10th Australian workshop on Safety critical systems and software*, pages 3–12, Darlinghurst, Australia. Australian Computer Society, Inc.

[Bianchi, 2000] Bianchi, G. (2000). Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547.

[Bishop, 2007] Bishop, M. (2007). About penetration testing. *Security & Privacy, IEEE*, 5(6):84–87.

[CENELEC, 1999] CENELEC, E. S. (1999). En 50126: Railway applications. the specification and demonstration of reliability, availability, maintainability and safety (rams).

[CENELEC, 2001] CENELEC, E. S. (2001). En 50159-2: Railway applications - communication, signalling and processing systems - part 2: Safety related communication in open transmission systems.

[Chatzimisios et al., 2004] Chatzimisios, P., Boucouvalas1, A. C., and Vitsas, V. (2004). Optimisation of rts/cts handshake in ieee 802.11 wireless lans for maximum performance. In *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pages 270–275.

[DSB, 2008a] DSB (2008a). Dsb: Internet i tog – tdc hotspot i dsb tog. Web. Available from: `http://www.dsb.dk/hotspot/`.

[DSB, 2008b] DSB (2008b). Dsb: Togsæt – litra mfa-ff-mfb. Web. Available from: `http://www.dsb.dk/cs/Satellite?pagename=DSB/Page/Indholdsside_med_sidemenu_og_introindhold&c=Page&cid=1098341129632&a=Artikel&aid=1099378075858&pid=1099565562725&p=Artikel`.

[Friis, 1946] Friis, H. (1946). A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256.

[Giannopoulou et al., 2000] Giannopoulou, K., Katsareli, A., Dres, D., Vouyioukas, D., and Constantinou, P. (2000). Measurements for 2.4 ghz spread spectrum system in modern office buildings. *Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean*, 1:326–329 vol.1.

[Haykin, 2001] Haykin, S. (2001). *Communication systems*. John Wiley & sons, Inc., 4th edition.

[ITU-T, 1994] ITU-T (1994). X.200 standard.

[Johnson, 1928] Johnson, J. B. (1928). Thermal agitation of electricity in conductors. *Phys. Rev.*, 32(1):97.

[Koopman, 2002] Koopman, P. (2002). 32-bit cyclic redundancy codes for internet applications. *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*, pages 459–468.

[Na et al., 2004] Na, C., Chen, J., and Rappaport, T. (2004). Hotspot traffic statistics and throughput models for several applications. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 5:3257–3263 Vol.5.

[Niedermayer et al., 2006] Niedermayer, H., Klenk, A., and Carle, G. (2006). The networking perspective of security performance - a measurement study. In German, R. and Heindl, A., editors, *MMB*, pages 119–136. VDE Verlag. Available from: `http://dblp.uni-trier.de/db/conf/mmb/mmb2006.html#NiedermayerKC06`.

[Nyquist, 1928] Nyquist, H. (1928). Thermal agitation of electric charge in conductors. *Phys. Rev.*, 32(1):110–113.

[Proakis, 2001] Proakis, J. G. (2001). *Digital Communications*. McGraw-Hill, 4th edition.

[PROFIBUS international, 2008a] PROFIBUS international (2008a). Official profibus website. Available from: `http://www.profibus.com/`.

[PROFIBUS international, 2008b] PROFIBUS international (2008b). Official profisafe website. Available from: `http://www.profisafe.net/`.

[Rail Safety and Standards Board, 2007] Rail Safety and Standards Board (2007). Engineering safety management (the yellow book).

[Rappaport, 1996] Rappaport, T. S. (1996). *Wireless Communications: Principles and Practice*. Prentice Hall, Inc., Upper Saddle River, NJ, USA, 1st edition.

[Sakurai and Vu, 2007] Sakurai, T. and Vu, H. (2007). Mac access delay of ieee 802.11 dcf. *Wireless Communications, IEEE Transactions on*, 6(5):1702–1710.

[Seidel and Rappaport, 1991] Seidel, S. and Rappaport, T. (1991). Path loss prediction in multifloored buildings at 914 mhz. *Electronics Letters*, 27(15):1384–1387.

[Stelzried, 1968] Stelzried, C. (1968). Microwave thermal noise standards. *Microwave Theory and Techniques, IEEE Transactions on*, 16(9):646–655.

[Tovar and Vasques, 1999] Tovar, E. and Vasques, F. (1999). Real-time fieldbus communications using profibus networks. *Industrial Electronics, IEEE Transactions on*, 46(6):1241–1251.

[TRENDnet, 2007] TRENDnet (2007). Trendnet tew-443pi data sheet. Available from: `http://trendnet.com/products/proddetail.asp?prod=140_TEW-443PI`.

[Willig, 2002] Willig, A. (2002). Analysis of the profibus token passing protocol over wireless links. *Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on*, 1:56–60 vol.1.

[Zheng et al., 2006] Zheng, Y., Lu, K., Wu, D., and Fang, Y. (2006). Performance analysis of ieee 802.11 dcf in imperfect channels. *IEEE Transactions on Vehicular Technology*, 55(5):1648–1656.

Appendix A

Measurement of Background Noise and Signal Power

Introduction

The purpose of this experiment is to measure background noise and signal levels in multiple non-line of sight setups. The measurements were made in connection with the experiment described in D4.1, section 12.2 and thus uses the same basic layout.

Methods

The setup consist of two PCs with TRENDnet TEW-443PI 802.11legace/b/g interfaces set for 802.11b [TRENDnet, 2007]. The cards are set to a send power of 18 dBm through a 2 dBi dipole antenna. Both computers run Ubuntu Linux 7.04 (feisty fawn) with the MadWIFI v0.9.4 driver for the wireless interface. PC1 is a 400 MHz Pentium II (512 KB cache) with 192 MB RAM, and PC2 is an 800 MHz Pentium III (256 MB cache) with 512 MB RAM.

Unrelated to the experiment, an IPsec tunnel is set up over the wireless link. Additionally PC1 has 3 fast Ethernet cards and PC2 has 2 fast Ethernet cards. None of this has an impact on the present experiment.

In the experiment PC1 is kept stationary and PC2 is placed in three different locations named “Room 1”, “Room 2”, and “Room 4” as shown in Figure A.1. The location of the testbed is the cellar of the NOVI building, Niels Jernes Vej 12-14, in Aalborg, Denmark. The walls are made from concrete, probably reinforced concrete. At all locations the antennas as pointed directly up, and the back of the computers point towards each other.

PC2 contains a script which stores the output of `iwconfig` to a file every second. In each location the dump script is executed and left running for at least 60 minutes while a transmission is on-going. The transmission is initiated to ensure that the figures which `iwconfig` output are updated as often as possible. After the experiment, the output is parsed to obtain signal and noise powers in dBm. These numbers are imported to MATLAB and converted to Watts. The mean and variance is then calculated and converted to dBm and dB respectively. The SNR is calculated by subtracting the mean noise power from the mean signal power.

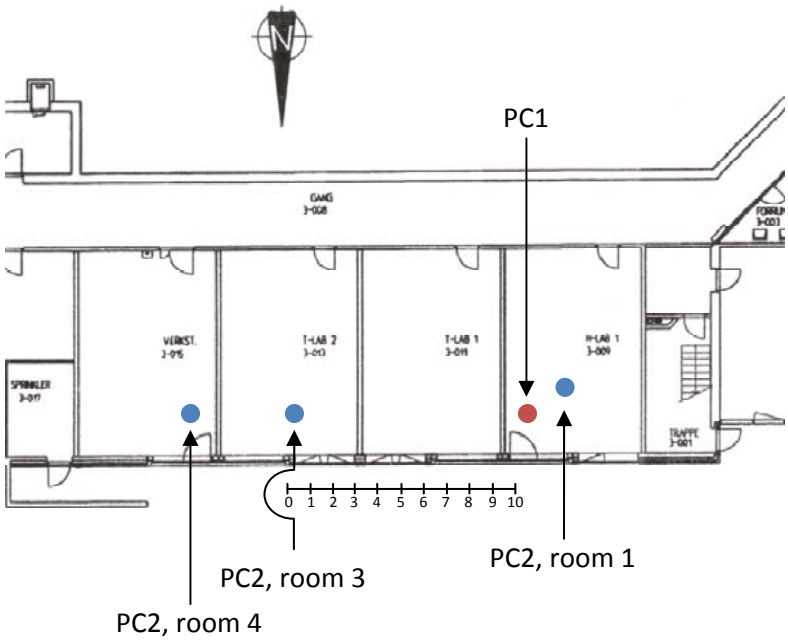

Figure A.1: Physical configuration of the experiment. The distance from PC1 to the three locations are approximately, 2.2 m, 10 m, and 15 m.

Results

The results of the experiment are shown in Table A.1.

Location	Distance	Noise mean	Noise variance	Signal mean	Signal variance	SNR
Room 1	2.2 m	-96.2 dBm	-272 dB	-43.0 dBm	-139 dB	53.2 dB
Room 3	10 m	-96.8 dBm	-271 dB	-72.8 dBm	-214 dB	24.1 dB
Room 4	15 m	-96.7 dBm	-270 dB	-81.4 dBm	-232 dB	15.3 dB

Table A.1: Results from the experiment

Discussion

The variance is extremely low, this might be attributed to the fact that it is unknwon how often the actual physical layer measurements occur and how often the `iwconfig` output is updated. Even so, if the update rate is much lower than the polling rate by the script, variations should be equally amplified leading to the same level of variance.

It is unknown exactly how these measurements are made in the physical layer, this is problematic. However, the values obtained from this experiment show the same tendency as the wireless medium model developed in section 5.4.

Appendix B

MATLAB Code

B.1 mac.m

```
function [tp, cumbins, per, bins, bin_edges, mean_delay, counters, ...
    times, air_usagefrac, tp_other_nodes, Z_min, Z_max] = ...
    mac(data_packet_sendtime, ack_packet_sendtime, ped, pea, ...
    maxretry, CW_min, CW_max, t_st, n, granu, ACK_TIMEOUT, SIFS, ...
    DIFS, bin_size, nocontenders, payload_length)
%Input:
%  data_packet_sendtime - Time to send a data packet [s]
%  ack_packet_sendtime - Time to send an ACK packet [s]
%  ped - Probability of losing a data packet [-]
%  pea - Probability of losing an ACK packet [-]
%  maxretry - Retry limit [-]
%  CW_min - Minimum contention window size [-]
%  CW_max - Maximal contention window size [-]
%  t_st - Time slot size [s]
%  n - Number of repetitions [-]
%  granu - Repetition granularity [-]
%  ACK_TIMEOUT - Timeout before assuming that an ACK is lost [s]
%  SIFS - Short Inter Frame Spacing [s]
%  DIFS - Distributed Inter Frame Spacing [s]
%  bin_size - Width of bins of the "PDF" [s]
%  nocontenders - Number of contending nodes [-]
%Output:
%  tp - Throughput figure [bits/s]
%  cumbins - Cumulative Distribution Function bins [-]
%  per - Packet Error Rate [-]
%  bins - Pseudo Probability Densitiy Function bins [-]
%  bin_edges - Edges of the bins [s]
%  mean_delay - Mean delay [s]
%  counters - Counters for each kind of delay event
%  times - Times associated with each delay event
%     The array position for counters and times corresponds to the events:
%       (1) Successfull transmission
%       (2) Collision
%       (3) Data OK, ACK lost
%       (4) Data lost
%       (5) Backoff time slot
%       (6) Other nodes transmitting for collision
```

```
%       (7) Other node transmitting OK packet

%%% Derived parameters
%Time used by the different possible events
T_s = data_packet_sendtime + SIFS + ack_packet_sendtime + DIFS; %Success
T_c = data_packet_sendtime + DIFS; %Collision
T_a = T_s; %Data OK, ACK lost (bit error)
T_d = T_c; %Data lost (bit error)
times = [T_s T_c T_a T_d t_st T_s T_s];

%%% Simulate
n = ceil(n/granu);

%Calculate contention window sizes for each step
W(1:maxretry) = CW_min*2.^(0:(maxretry-1));
W(W>CW_max) = CW_max;
W_cumsum = cumsum(W);

%Find tau (send probability pr. node)
backoff_stages = (log(CW_max)-log(CW_min))/log(2) - 1;
%p_err = 1-(1-ped)*(1-pea); %Bit error prob
tau = fzero(@mac_tau_eq, [0,1], [], nocontenders, CW_min, ...
    backoff_stages);

%Probability of one of the other nodes sending
p_on_send = 1-(1-tau).^nocontenders;

%Probability of a transmission by another node resulting in a collision,
%given that a node transmits, and that the DMI does not
p_cond_coll_on = 1-(1-tau)^(nocontenders-1);

%Probability of my transmission failing
p_trans_fail = 1-(1-p_on_send)*(1-ped)*(1-pea);

%Conditional probabilities given that a packet is lost, how is it lost?
p_coll = p_on_send/p_trans_fail; %Collision
p_acklost = (1-p_on_send)*(1-ped)*pea/p_trans_fail; %ACK lost
%p_datalost = (1-p_on_send)*ped/p_trans_fail; %Data lost

%Calculate bin edges
Z_min = T_s;
Z_max = T_s + sum(W)*t_st + W_cumsum(end)*T_s + (maxretry-1)*T_a;
bin_edges = 0:bin_size:(Z_max+bin_size);
bins = zeros(1, length(bin_edges));
fprintf('Min delay: %.0fus   Max delay: %.0fs   Bins: %d\n', ...
    Z_min*1e6, Z_max, length(bin_edges));

%Initial data for estimating simulation time
[start_time total_iterations iterations_done last_iteration_start ...
    last_display] = time_init(n);

counters = zeros(1, 7);
run_no = 0;
while run_no<n
    run_no = run_no + 1;

    %Generate Y = number of transmission attempts needed
    Y = geornd(1-p_trans_fail, 1, granu);

    %Remove packets which are dropped due to too many retries
    no_pdrop = sum(Y>maxretry);
    Y(Y>maxretry) = [];
```

```
%Generate backoff steps for each retry
X = zeros(maxretry, length(Y));
X_lost = zeros(maxretry, no_pdrop);
for i=1:maxretry
    X(i, :) = ceil(rand(1, length(Y))*W(i));
    X_lost(i, :) = ceil(rand(1, no_pdrop)*W(i));
end

%Remove unused backoff steps for delivered packets
for i=1:length(Y)
    X((Y(i)+1):end, i) = 0;
end

%Generate a binary mask of the failed transmissions for all packets
X_mask = (X ~= 0);
X_mask = X_mask(2:end, :);
X_mask = [[X_mask; zeros(1, granu-no_pdrop)], ...
    ones(maxretry, no_pdrop)];
[X_msize X_nsize] = size(X_mask);

%Generate samples of the different kinds of failure for all packets
X_mask_prob = X_mask.*rand(X_msize, X_nsize) + (X_mask-1);
Y_c = sum((X_mask_prob >= 0).*(X_mask_prob < p_coll));
Y_a = sum((X_mask_prob >= p_coll).*(X_mask_prob < p_coll+p_acklost));
Y_d = sum((X_mask_prob >= p_coll+p_acklost).*(X_mask_prob <= 1));
Y_c_lost = Y_c( :, granu-no_pdrop+1:end );
Y_a_lost = Y_a( :, granu-no_pdrop+1:end );
Y_d_lost = Y_d( :, granu-no_pdrop+1:end );
Y_c = Y_c( :, 1:granu-no_pdrop);
Y_a = Y_a( :, 1:granu-no_pdrop);
Y_d = Y_d( :, 1:granu-no_pdrop);

%Generate other number of transmissions by interferers
on_send = binornd(sum(X), p_on_send);
on_send_lost = binornd(sum(X_lost), p_on_send);

%Divide other transmissions in collisions / successful
on_send_coll = round(p_cond_coll_on*on_send);
on_send_ok = on_send-on_send_coll;
on_send_coll_lost = round(p_cond_coll_on*on_send_lost);
on_send_ok_lost = on_send_lost-on_send_coll_lost;

% Calculate delays for delivered packets
Z = 1*T_s + ... %The succesful transmission (always one)
    Y_c*T_c + ... %Collisions
    Y_a*T_a + ... %Succesful data transmission, but lost ACK
    Y_d*T_d + ... %Lost data packet
    sum(X)*t_st + ... %Waiting time while decreasing backoff counter
    on_send_coll*T_s + ... %Other nodes transmitting collisions
    on_send_ok*T_s;    %Other nodes transmitting ok packets

%Count different events:
%       (1) Successfull transmission
%       (2) Collision
%       (3) Data OK, ACK lost
%       (4) Data lost
%       (5) Backoff time slot
%       (6) Other nodes transmitting for collision
%       (7) Other node transmitting OK packet
counters = counters + [...
    length(Z) ... %(1)
```

```
        sum(Y_c) + sum(Y_c_lost) ... %(2)
        sum(Y_a) + sum(Y_a_lost) ... %(3)
        sum(Y_d) + sum(Y_d_lost) ... %(4)
        sum(sum(X)) + sum(sum(X_lost)) ... %(5)
        sum(on_send_coll) + sum(on_send_coll_lost)... %(6)
        sum(on_send_ok) + sum(on_send_ok_lost)]; %(7)

    % Add delays to respective bins
    bins = bins + histc(Z, bin_edges);

    %Calculate and print simulation time remaining
    [iterations_done last_iteration_start, last_display] = ...
        time_remaining(start_time, last_iteration_start, ...
        total_iterations, iterations_done, last_display);
end
%Print total simulation time elapsed
time_elapsed(start_time);

%Calculate total simulate time
sim_time = sum(counters.*times);

%Calculate mean delay and tp
mean_delay = sim_time/sum(bins);
%tp = (payload_length/mean_delay);
tp = payload_length*counters(1)/sim_time;
if nocontenders ~= 0
    tp_other_nodes = payload_length*counters(7)/(sim_time*nocontenders);
else
    tp_other_nodes = 0;
end

%Normalize bins to get the probability functions pdf and cdf
bins = bins/sum(bins);
cumbins = cumsum(bins);

%Calculate drop rate
per = p_trans_fail^(maxretry+1);

%Calculate airtimes
airtime = ...
    counters(1)*(data_packet_sendtime + ack_packet_sendtime) + ...
    counters(2)*data_packet_sendtime + ...
    counters(3)*(data_packet_sendtime + ack_packet_sendtime) + ...
    counters(4)*data_packet_sendtime + ...
    counters(6)*(data_packet_sendtime + ack_packet_sendtime) + ...
    counters(7)*(data_packet_sendtime + ack_packet_sendtime);
air_usagefrac = airtime/sim_time;

end %function
```