NTNU
Innovation and Creativity

Adaptive Frequency Hopping with Channel Prediction

John Torjus Flåm

Master of Science in Communication Technology
Submission date: June 2006
Supervisor: Tor Audun Ramstad, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

A radio employing frequency hopping can be made adaptive to the noise and fading by extracting poor channels from the hopping pattern, and exluding them for usage. Such a procedure, implicitly expects that the extracted channels will remain poor in quite some part of the future. In a dynamic environment that expectation is not always met. Correspondingly, the extracted channels may coincide poorly with the actual noise and deep fades. Often the reason is that systems adapt only reactively. That is, a channel is extracted only after a significant share of data has been lost. If the systems had the ability to adapt proactively, meaning that they could foresee and void poor channels, performance gain could be expected. This provides the motivation for attempting at predicting channel quality. The objective is, of course, to identify and avoid poor channels. The assignment has been identified in accordance with Nordic Semiconductor ASA. It addresses a particular transceiver, the nRF24Z1, but the work and potential solutions should be as generic as possible.

Assignment given: 16. January 2006
Supervisor: Tor Audun Ramstad, IET

Preface

This Master's thesis summarizes my work during the 10th, and final, semester of the Master's program at the Department of Electronics and Telecommunications, NTNU. The assignment was identified in accordance with Nordic Semiconductor ASA and approved by NTNU. I enjoyed the work, and I would especially like to thank Michael Erstad, my contact at Nordic Semiconductor, and Professor Tor A. Ramstad, my tutor at NTNU, for support during the project. I would also like to thank my father, S.D. Flåm, for instructive discussions and advice.

Trondheim, June 8, 2006

....

John Torjus Flåm

Abstract

The number of radio systems operating in the 2.4 GHz band is rising as a result of increased usage of wireless technologies[1]. Since such devices interfere with one another, satisfactory co-existence becomes important. Several techniques serve to reduce the interference. Included among these are frequency hopping (FH) and power-control.

This report focuses only on FH, and particularly on methods that make FH schemes adaptive. An FH scheme is adaptive if it responds to the noise and fading by avoiding channels that are unfit for transmission.

An example of such a scheme is already implemented in an audio transceiver, the nRF24Z1, manufactured by Nordic Semiconductor. That transceiver provides the framework for this study, and the main objective is to suggest improvements to its FH algorithm. Better performance is particularly interesting in high quality audio streaming because such transmissions generally have strict real time requirements. Thus, the time to retransmit corrupted data is limited, and measures to reduce the impact of interference and fading are desired.

The FH scheme implemented in the nRF24Z1 broadly works as follows. If a channel distorts more than a certain share of the transmitted data, it is extracted from the FH routine and listed as banned for usage. The ban list has room for maximum 18 out of 38 channels and can therefore filter out significant parts of the spectrum. If the system identifies more poor channels than the list can hold, the oldest channel in the ban list is released, and the newly identified one takes its place. In a scenario where noise and deep fades come to occupy a rather stable and limited group of channels, the banned channels will match the unsuited parts of the spectrum quite accurately, and the scheme performs well.

However, when the noise and fading is changing, maybe quickly and non-periodically, the performance drops significantly. The reason is that channels are banned only *after* they have caused trouble, which has two negative effects. Firstly, because the bulk of the transmitted data was distorted, the need for retransmission is large. Secondly, since the transmission conditions are changing, the ban list becomes outdated and reflects poorly on the actual interference and fading. Therefore, in this report, the possibility of predicting poor channels in order to avoid them *beforehand*, is investigated.

For the purpose of prediction, small test packets are transmitted. In short, the principle of operation is as follows: if a test packet is readable at the receiver, the channel is used. Otherwise, it is avoided. Computer simulations indicate that this technique improves transmission conditions and reduces the need for retransmission when the noise and fading change significantly. Large changes are indeed common in practice. They occur, for example, if a broadband interferer is switched off or greatly varies its output power. Plainly, they could also come about when objects move.

Despite promising simulations, channel testing does not come without side effects. An audio streaming system like the nRF24Z1 must secure a certain flow of data to avoid audible errors. If prediction algorithms are to secure that flow, a compromise

[1] In 2002, sale of 2.4 GHz components represented 18% of Nordic Semiconductor's revenue, whereas in 2005 the figure was 63%.

must be made: the more time a system spends on channel testing, the less time remains for transmission of data. Therefore, at some point, testing must be terminated to leave room for the real job. In consequence, the key issue of finding the best trade-off between testing and transmission must be addressed. This report presents three adaptive FH schemes which approach that issue in their own manner.

The performance of the proposed prediction schemes has been investigated using a channel model for the ISM band (Industrial, Medical, and Scientific). It is coded and developed in MATLAB. The model mimics the effects of a real mobile channel quite well, which supports the credibility of the simulation results.

Contents

List of Figures

List of Tables

Abbreviations

FH	Frequency Hopping
ISM	Industrial, Scientific and Medical (radio band)
TX	Transmitter
CRC	Cyclic Redundancy Check
RX	Receiver
LOS	Line Of Sight
ISI	Inter Symbol Interference
WLAN	Wireless Local Area Network
DSSS	Direct Sequence Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
GFSK	Gaussian Frequency Shift Keying
QPSK	Quadrature Phase Shift Keying
C/I	Carrier to Interference and noise ratio
IEEE	Institute of Electrical and Electronics Engineers
FIFO	First In First Out
CDF	Cumulative Distribution Function
PDF	Probability Density Function
FCC	Federal Communications Commission
DP	Dynamic Programming
MDP	Markov Decision Process

Table 1: Abbreviations

1 Introduction

1.1 Background

Also in an earlier student project, I looked at improving the frequency hopping algorithm of the nRF24Z1. In fact, a modified algorithm was proposed, and it differs from the existing one in the way that channels are released from the ban list. During the course of operation, the proposed algorithm maintains a 'ban ratio' for each channel. It does so by noting how often each channel is banned compared to how often it is used. When a release becomes necessary, the channel exhibiting minimum ban ratio is cut loose.

Broadly, this modification is motivated as follows: whereas the existing scheme does not classify or distinguish between banned channels, and simply releases them by the first-in-first-out (FIFO) principle, the suggested scheme gradually identifies which channels are mostly impeding the system's performance. By doing so, it releases banned channels on a more reasonable basis.

The project indicated that the performance of an FH scheme strongly depends on its ability to learn from - and adapt to - the noise and fading profile. Of pivotal importance is the capacity to distinguish between transient and long-term poor channels. The modified FH scheme displayed better performance than the existing one, primarily because it distinguished these better. Further details on the findings in the above mentioned project can be found in the project report [1].

1.2 Current Problem

1.2.1 Description

Both the FH scheme currently implemented in the nRF24Z1 and the proposed modified FH algorithm, advocated in section 1.1, provide simple yet effective techniques to reduce usage of low quality channels. When the interference and fading deteriorate a rather stable and limited group of channels, these algorithms reach good peak performances rather quickly, and maintain them to acceptable degree thereafter.

However, under markedly non-stationary conditions the group of unsuitable channels has larger turnover, and the composition of that group becomes less predictable. As a result the algorithms perform significantly worse. The reasons are twofold. Firstly, a channel is banned only *after* it has been used to transmit data. Because a large share of the data has been distorted, considerable retransmission might be necessary. Secondly, the principle of banning poor channels is implicitly based on expectations that these channels will remain unfit for transmission in quite some part of the future. In a dynamic environment these expectations are not always met. Consequently, the banned channels may fit poorly to the actual noise and fading. In the extreme case, the system may end up employing low quality channels frequently and ban good ones too often, which has a disturbing effect on the performance. Obviously, it is desirable to mitigate such undesirable outcomes.

1.2.2 Approach

A main reason why the existing FH scheme performs poorly under changing transmission conditions is that the system adapts only *reactively*, as prompted by passed events. If the system had the ability to adapt *proactively* - meaning that it attempts to foresee and avoid poor channels, better performance could be obtained. This provides the motivation for attempting at *predicting the quality of channels*. The objective is of course to avoid poor channels, and that is the main focus in this report. The philosophy and the procedure are simple. Namely, for the purpose of prediction the system transmits short test packets on channels that are planned used in the near future. If the test packet arrives readable, the channel is used. Otherwise, it is avoided. Three adaptive FH schemes are proposed, differing from each other in their prediction part. They are introduced in section 3.3.

1.2.3 Problem orientation

Realistic prediction schemes are desired, and therefore they should only be based on available information at the transmitter (TX) and the receiver (RX)[2]. In search for suitable schemes, the existing protocols, limitations of memory and processing capabilities of the nRF24Z1 transceiver have been taken into account. The limitations of the transceiver have not been the primary concern, however. Prediction schemes which are not limited by a number of practical restrictions have been the first objective. The algorithms proposed in this report can therefore not be implemented in the nRF24Z1 right away, as it is designed today. They could however be taken into consideration for future designs.

1.2.4 Testing of proposed solutions

In order to test the performance of the proposed channel prediction schemes with some confidence, without implementing them in hardware, they are simulated using a channel model for the ISM (Industrial, Medical, and Scientific) band. That model was developed in MATLAB during the student project mentioned in section 1.1. The model is based on a three-dimensional matrix which is updated each time there is a physical change in the scenario. The high ability to perform matrix algebra, which is one of the most prominent features of MATLAB, is thus exploited. The channel model incorporates effects such as multi-path propagation, dynamic reflectors and various sources of noise. It has been tested under conditions where specific fading characteristics can be expected. The obtained characteristics were satisfying. In fact, the proposed channel prediction schemes are simulated in a model which mimics the effects of a real mobile channel quite well. This strengthens the confidence in the simulation results.

[2]The abbreviations TX and RX refer to the *audio* transmission direction. RX also transmits back to TX, but not audio data, only acknowledgments.

1.3 Key features of the nRF24Z1

A complete product specification of the nRF24Z1 can be found in [2]. Here only a few of the key features are mentioned because they will be referred to in what follows:

- Operating carrier range: 2.404 GHz - 2.478 GHz.
- Channel bandwidth: 4 MHz.
- Separation between center frequencies: 2 MHz.
- Number of channels: 38. Note that the channels are overlapping. If one disregards every second channel, the remaining channels are non-overlapping.
- Receiver sensitivity: -80 dBm.
- Capacity of receive buffer (in terms of packets[3].): 40.

1.4 Limitations

1.4.1 General limitations

Most radio transceivers that operate in the ISM band are short range transceivers. Maximum output power for the nRF24Z1 is 0 dBm, and the receiver sensitivity is -80 dBm. This corresponds to a range of approximately 10 meters or less. In a dynamic environment the LOS (Line Of Sight) component between TX and RX is often significantly attenuated and RX has to rely on the power contributions from the reflection paths. The indoor environment normally exhibits a larger number of reflectors than the outdoor environment. For this reason, the channel model which is used for simulations is restricted to an indoor scenario. Generally the number of reflection paths is stochastic and very high. In a computer model one cannot deal with an unlimited number of reflections. The model therefore considers only a limited number of reflection paths, and they are not chosen randomly.

When investigating the performance of FH schemes, there are two properties that are particularly interesting. Firstly, the instantaneous C/I (carrier to interference and noise ratio) that the scheme obtains indicates the current suitability of the channel. Secondly, since TX and RX have buffers, a certain latency can be allowed. Therefore, the schemes' ability to supply data to the receiver need not necessarily be immediate. It must only be able to supply data before it is too late. Both of these properties can be observed without considering the actual information content in the received signal. The information content is therefore of no importance in this report.

The nRF24Z1 uses GFSK (Gaussian Frequency Shift Keying) as its modulation scheme. This scheme does not affect the envelope of the carrier, and hence the received power is not affected by the modulation technique. Modulation and demodulation issues are therefore not considered in the following.

[3] The term *packet* will be properly explained in section 2.1

1.4.2 Specific limitations

Computer simulations can only indicate how well an FH scheme could work in hardware. After simulation one may proceed to implement the algorithm in hardware and test its performance in a controlled environment. These observations suggest that models and computer simulations had better be kept simple. On basis of the general limitations, a set of specific simplifying assumptions have been found reasonable. The previously developed channel model is based on the following simplifications:

- During a simulation, only one designated TX-RX pair is considered. All other applications are considered as interferers.

- In addition to the LOS component, the channel model only considers first order reflections (reflections that travel via one reflector only).

- Information content and modulation/demodulation issues are not considered. A packet is simply transmitted with a power of 0 dBm. A packet is considered readable at the receiver if it has power larger than -80 dBm and a C/I larger than 12 dB.

- The channel model has a small default background noise in order to avoid problems with the C/I taking potentially infinite values.

- There are no synchronisation issues. A common clock is available at TX and RX.

- Only interferers that transmit on channels which are partly or fully overlapping with the current TX-RX channel contribute to the interference. Sources transmitting in other bands do not produce interference.

- The antennas are vertically polarized dipoles which radiate omni directionally.

- Antenna gains are of no concern. The antennas are assumed to have constant and equal gain.

- A far-field condition is always assumed, meaning that the Friis-formula (presented in section 2.2.5) can be used to calculate the received power. A far-field condition can easily be provided by introducing a minimum allowed RX-TX separation.

- Dynamics do not introduce Doppler shifts. The channel is time-invariant. This statement will be better founded in section 2.2.4.

- The delay spread is small compared to the transmitted frame length, and the channel is considered to be ISI (Inter Symbol Interference)-free. This will be explained in section 2.2.3.

- Walls and non-human obstacles are of the same material. This is a simplification with respect to calculation of reflections, because one has only one value of relative permittivity. For the simulations, the relative permittivity of brick walls has been chosen: $\epsilon = 4.4$.

- Human obstacles attenuate the signal to a level which is below the receiver sensitivity. Therefore they can be considered as fully energy absorbing objects in the model. This will be further explained in section 2.2.6.

The channel prediction FH algorithms that are proposed have been developed under the following simplifying assumptions:

- In a dynamic scenario, a prediction of channel quality is valid only for about 15 ms or less. This statement will be better founded in section 2.2.4.

- When using dynamic programming[4] (DP), to address the problem of how much channel testing should be done, at the expense of reduced transmission time, the following assumptions have been made:

  1. All channels have the same probability of being poor. In addition, the suitability of one channel is considered independent from all other channels. Neither assumption is generally true, but convenient and largely simplifying. Otherwise much information about various channel would have to be stored, and the transceiver would need to find optimal action policies[5] while being in operation.
  2. If a channel is poor, it remains poor at least for the duration it is used, meaning that all the transmitted data are distorted. Similarly, when a channel is satisfactory, all transmitted data are readable at the receiver. This is generally not true, but the assumption will be justified in section 2.2.4. Such an assumption represents a simplification because it considerably reduces the number of possible next states[6] in the DP problem.

[4]An introduction to dynamic programming will be provided in section 2.4

[5]The notion of *policies* will be explained in section 2.4.2.

[6]This will also be explained in section 2.4.2.

2 Theory

This section contains theory on which the work presented in this report is based. It also justifies some of the simplifying assumptions made in section 1.4.2. The section starts by explaining some key terminology. Then introductions to the fields of mobile radio theory, frequency hopping and dynamic programming follow. Conclusively, the most common interferers of nRF24Z1 are presented.

2.1 Packets, buffers, frames and periods

There are two main data units that are referred to in this report: *packets* and *frames*. Transmitted and received packets are placed in the receive *buffer* before they are read out sequentially. The number of packets that a frame can contain depends on the length of the *period*. All these terms are used extensively throughout this report, and they are fundamental to understanding the current protocol of the nRF24Z1. A proper explanation of what they mean is therefore appropriate.

Packets and buffers. The term packet, actually refers to an *audio* packet. This is a data unit that carries a small part of the audio signal. Packets are read out of the receive buffer sequentially to produce the analog audio signal. Missing packets in the sequence will generally result in an audible error, and due to interference and deep fades, some packets are inevitably lost. Buffers reduce the number of audible errors, because they provide time to retransmit missing packets. When a packet can be transmitted multiple times, the probability that it arrives in time to be read out of the buffer increases. The buffer size is therefore an important parameter of wireless transmission systems. The basic structure of a packet is shown in figure 1. It has a preamble and an address field, followed by the payload. At the end of the packet, data for error detection (cyclic redundancy check - CRC) are included. Note that, upon successful reception, only the payload enters the buffer.

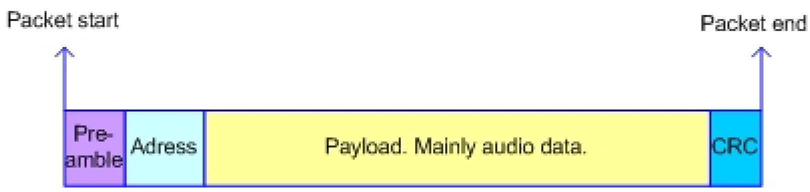

Figure 1: Packet structure

Frames and periods. A packet is transmitted as an integral part of a larger data unit - a frame. The default number of packets in a frame is 8, but the frame can hold more packets if necessary. The reason is as follows. The receive buffer has room for 40 packets. Packets are read out from that buffer one by one, and during the time it takes to transmit one frame, 8 packets will be read out. The receive buffer is refilled by the undistorted packets of the frame. If the channel is heavily interfered or is situated in a deep fade, the whole frame is likely to be unreadable, and the receive buffer will not be refilled at all. After such an event the receive buffer will contain 8 packets less than before. In order to cover such gaps, TX can increase the number of packets in each frame until the buffer is full again.

When RX receives a frame, it generally responds by transmitting an acknowledgment (ACK). The combined duration of the frame and the ACK is referred to as a *period*. In figure 2, the frame structure and the period is shown. The extra packets are transmitted first, because they are considered to be the most urgently needed at RX.

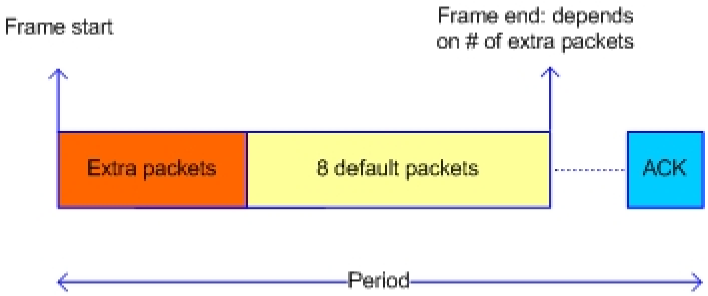

Figure 2: Frame structure

The length of the period and the maximum number of possible packets in a frame depend on the audio sampling rate. Table 2 shows the dependencies.

RATE	MAX PACKETS IN FRAME	PERIOD
48 KHz	13	2.67 ms
44.1 KHz	14	2.9 ms
32 KHz	21	4 ms

Table 2: Rates, packets and periods

2.2 General mobile channel theory

Much of the theory presented in here can also be found in [1]. It is based on [3], [4] and [5]. For a more detailed description of the theory related to mobile channels than presented in this paper, textbooks such as [4] and [6] can be recommended.

2.2.1 Causes of signal variation

The received signal power over a mobile channel is generally affected by the following phenomena:

- path loss (reduction of received power with distance)
- slow fading (due to physical obstacles)
- fast fading (due to constructive and destructive combination of multi-path components)

The nRF24Z1 transceiver is equipped with a power control which, to some extent, compensates the effect of path loss and slow fading. The desired effect of an adaptive FH scheme is therefore mainly to counteract the negative effects of fast fading and interference.

2.2.2 Fast fading and multi-path propagation

Consider a single sinusoid injected into an isotropic TX antenna. The signal travels from TX to RX via N different paths. This phenomenon is called multi-path propagation. At the receiver, the amplitudes and phase angles of the different multi-path components are superimposed such that a Gaussian band pass $r(t)$ signal originates:

$$r(t) = \sum_{i=1}^{N} A_i cos(2\pi f t + \varphi_i(t)) = A(t) cos(2\pi f t + \varphi(t)). \tag{1}$$

Notice that $r(t)$ is an envelope-varying signal with a time varying phase angle. The time variation of $A(t)$ is caused by the fact that N independent signals may add up constructively or destructively depending on their phase angles $\varphi(t)$, that is, depending on time. Assuming a strong LOS component, $A(t)$ is generally Rice distributed. If the LOS component is obstructed $A(t)$ follows a Raleigh distribution. $\varphi(t)$ is assumed to be uniformly distributed in both cases.

The N multi-path components add up destructively, causing severe attenuation, or so called "deep fades", whenever

$$2\pi f t + \varphi(t) \approx (2k+1)\tfrac{\pi}{2},\ k \in [0, 1, 2...].$$

Therefore, good transmission conditions can switch to very poor when the distance between TX and RX is changed by approximately half a wavelength. At 2.4 GHz, that corresponds to a movement of 6.25 cm. This is a small movement which in most cases happens fast; hence the term fast fading[7]. Obviously, fast fading makes transmission between dynamic radios far more challenging than when stationary conditions can be guaranteed.

2.2.3 Coherence bandwidth

The impact of frequency selectivity depends on the coherence bandwidth, B_c, of the channel. The coherence bandwidth specifies a bandwidth over which the channel can be regarded as constant. The coherence bandwidth can be approximated by

$$B_c = \frac{1}{T_m}, \tag{2}$$

where T_m is the delay spread. For a system with transmission bandwidth B, the following relations hold:

- $B << B_c$: narrow band system, negligible impact of frequency selectivity
- $B >> B_c$: wide band system, significant impact of frequency selectivity

The coherence bandwidth will depend on the environment that the nRF24Z1 operates in, but one can resort to an example to get an indication of the magnitude. Consider a quadratic room of size 100 m^2. Assume that TX and RX are positioned along the same wall with a separation of 10 m. This scenario is illustrated in figure 3. Disregarding the 3rd dimension, the reflection point which is the furthest away from

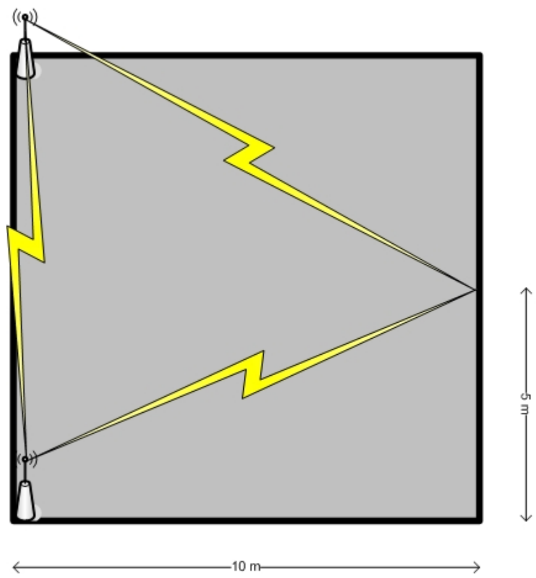

Figure 3: Calculating coherence bandwidth.

both RX and TX is on the middle of the opposite wall. Considering only first order reflections, the length of the path corresponding to this reflection point is:

$$d = 2\sqrt{5^2 + 10^2} \approx 22.4 \text{ m}.$$

[7]In the literature fast fading is also referred to as *small scale fading* because of the small movements that cause attenuation.

The time it takes the signal to travel this distance represents the maximum delay:

$$t_1 = \frac{d}{c} = \frac{22.4}{3 \cdot 10^8} \approx 75 \cdot 10^{-9} \text{ s} \Rightarrow B_1 = \frac{1}{t_1} = 13.4 \text{ MHz}.$$

The smallest delay corresponds to the direct path ($d = 10$ m):

$$t_2 = \frac{d}{c} = \frac{10}{3 \cdot 10^8} \approx 33 \cdot 10^{-9} \text{ s} \Rightarrow B_2 = \frac{1}{t_2} = 30 \text{ MHz}.$$

These delays and the corresponding bandwidths indicate the order of the coherence bandwidth. During simulations, the room size given in the example or smaller ones are used. For rooms with smaller dimensions the delays will be smaller, and the coherence bandwidth larger. Compared to the channel bandwidth of nRF24Z1, which is 4 MHz, the coherence bandwidth will generally be large. The channel might therefore be seen as non-frequency selective and the impact of inter symbol interference will on average be negligible.

2.2.4 Coherence time

The impact of time variance depends on the coherence time, T_c, of the channel, which is a measure of how fast the channel changes. The coherence time is given by

$$T_c = \frac{1}{B_d}, \tag{3}$$

where B_d is the Doppler spread. For a system with symbol duration T, the following relations hold:

- $T << T_c$: negligible impact of time variance
- $T >> T_c$: significant impact of time variance

To get an indication of the coherence time in our scenario, again an example is considered. Assume that TX moves with speed relative to RX of $v = 1$ m/s, and that the used frequency is $f = 2.4$ GHz. The wavelength is

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8}{2.4 \cdot 10^9} = 0.125 \text{ m}.$$

Correspondingly, the time needed to move half a wavelength is

$$\Delta t = \frac{\lambda/2}{v} = \frac{0.125/2}{1} = 0.0625 = 62.5 \text{ ms},$$

where Δt can be seen as an approximation of the coherence time T_c. The period for the nRF24Z1 depends on the audio fundamental rate (the sampling frequency). For the nRF24Z1 the minimum period is 2.67 ms and the maximum is 4 ms. Notice that even the largest period is only a fraction of the time it takes to change the fast fading situation, and for a single frame the channel can therefore be regarded as time invariant.

The coherence time also indicates how long a channel prediction could be valid if one disregards other noise sources. Since it has been assumed that within 62.5 ms a total change in transmission conditions may occur, it will only be sensible to predict within a fraction of that time. In the channel prediction schemes that are proposed in this report, a prediction is assumed to be valid for a quarter of the coherence time. That corresponds to around 15 ms. Accordingly, a prediction is valid for the duration of about 4 frames at the rate of 32 KHz, and about 6 at the rate of 48 KHz. Obviously, if the relative speed of TX and RX increases, or if dynamic reflectors are introduced , the predictability is reduced.

2.2.5 Calculating the received power from a multi-path signal

The prediction of channel quality is based on computed power levels of the received test packets. If the test packet has a power larger than the receiver sensitivity (-80 dBm), and the C/I is larger than 12 dB, the channel is assumed satisfactory to use, otherwise it is avoided. Therefore, the calculation of received power is a key operation during the course of the simulations, and a presentation of how it is done follows.

The power received by a receiver antenna in free space can be approximated by the Friis equation:

$$P_r = P_t \cdot g_t \cdot g_r \cdot \left(\frac{c}{4\pi f d}\right)^2, \tag{4}$$

where P_r is the received power, P_t is the transmitted power, g_t is the transmitter antenna gain, g_r is the receiver antenna gain, f is the carrier frequency, d is the distance between the antennas and c is the speed of light. In the channel model, antenna gains are not considered, and they are simply set to one. The received power then becomes:

$$P_r = P_t \cdot \left(\frac{c}{4\pi f d}\right)^2, \tag{5}$$

This expression is only valid in the far-field region of the transmitting antenna, which can easily be maintained by introducing a minimum allowed TX-RX separation. A multi-path signal combines at the receiver as described in equation 1 in section 2.2.2. From $r(t)$, the total received power can be derived. To obtain $r(t)$ one can proceed as follows. For each component in the multi-path signal, P_r is calculated by equation 5 if it is a LOS component, and by

$$P_r = P_t \cdot \Gamma^2 \cdot \left(\frac{c}{4\pi f d}\right)^2, \tag{6}$$

if it is a reflection component. Γ is a reflection coefficient which depends on the angle of incidence between the wave and the reflecting plane. It has a magnitude less or equal to one. How Γ can be calculated, is thoroughly described in [4, ch. 4]. The amplitude of each multi-path component is related to the power by:

$$P_r = \frac{A^2}{2R}, \tag{7}$$

where A is the peak amplitude of the sinusoidal voltage, and R is the resistance of the antenna. As for antenna gains, the expression is simplified by setting $R = 1$ and for the amplitude one obtains:

$$A = \sqrt{2P_r}, \tag{8}$$

The individual phase angles in equation 1 are computed directly from the length d of each multi-path component:

$$\varphi = 2\pi\frac{d}{\lambda} = 2\pi d\frac{f}{c}, \tag{9}$$

Given each amplitude A and phase angle φ, $r(t)$ can be calculated with equation 1. Finally the total received power is computed by:

$$P_r = \frac{1}{2\pi}\int_0^{2\pi} r^2(t)dt \tag{10}$$

2.2.6 Considering human obstacles as fully energy absorbing objects

A plausible explanation for this assumption can be found by considering the following example related to a microwave oven. A similar example can be found in [5, p 343]: This example shows how a microwave signal, at operating frequency of 2.45 GHz, is attenuated in a beef steak with relative permittivity of $\epsilon_r = 40$ and loss tangent of 0.35.

-The permittivity of free space is $\epsilon_0 = \frac{1}{36\pi} \cdot 10^{-9}$ F/m.

-The permeability of free space is $\mu = 4\pi \cdot 10^{-7}$ H/m.

The loss tangent can be approximated by

$$\frac{\sigma}{\omega \epsilon_0 \epsilon_r} = 0.35,$$

where σ is the conductivity of the material, and ω is the angular frequency. The conductivity σ is:

$$\sigma = 0.35 \omega \epsilon_0 \epsilon_r = 0.35 \cdot 2.45 \cdot 10^9 \cdot \frac{1}{36\pi} \cdot 10^{-9} \cdot 40 \approx 1.91.$$

The skin depth is calculated:

$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}} = \frac{1}{\sqrt{\pi \cdot 2.45 \cdot 10^9 \cdot 4\pi \cdot 10^{-7}}} \approx 0.0074 = 0.74 \text{ cm}.$$

The skin depth is the penetration distance corresponding to an attenuation of the signal by a factor of $1/e$. Assuming that for the transmitted signal, a person represents an obstacle with thickness of 20 cm having the same properties as flesh, one obtains for the attenuation A:

$$\frac{20}{0.74} \approx 27 \Rightarrow A = \frac{1}{e^{27}} \Rightarrow A = 10 \log(\frac{1}{e^{27}}) \approx -117 \text{ dB}.$$

This is only a rough estimate of the attenuation that a person represents, but it gives an indication of the order of magnitude. Attenuation of 117 dB will bring the signal far below the receiver sensitivity.

2.3 Frequency Hopping

The theory presented in this section is based on [7].

2.3.1 Definition of frequency hopping

FH is a spread spectrum technique, and is often abbreviated FHSS (Frequency Hopping Spread Spectrum). Spread spectrum techniques can be defined as techniques where the spectrum of the signal is spread out over a bandwidth which is much larger than that of the base band signal. Furthermore the spreading should be a result of an operation which is independent of the information bearing signal. In FH the carrier frequency of the system changes in discrete steps according to a hopping pattern during transmission, as shown in figure 4. The transmitted signal, which requires

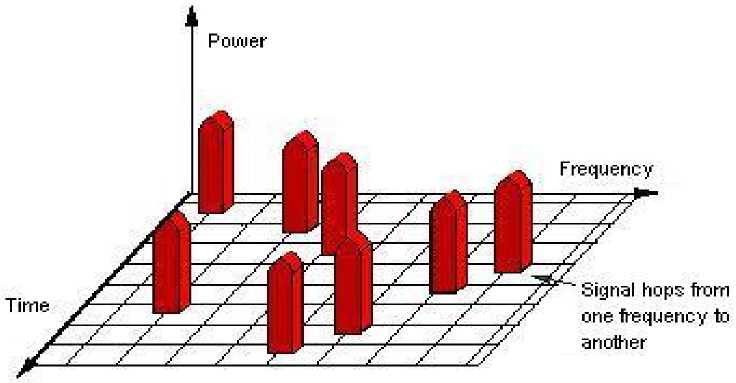

Figure 4: Illustration of frequency hopping

an instantaneous bandwidth of 4 MHz in our case, is constantly shifted within a larger band. Over time the signal occupies a bandwidth which is much larger than instantaneous bandwidth; a spreading is obtained.

2.3.2 The reason for using frequency hopping

The most prominent feature of spread spectrum systems is their robustness against intended and unintended interferers. Systems that use FH hopping normally remain on a particular channel just for a limited period of time. Accordingly, heavily interfered channels degrade the transmission only for that period, before the system again jumps to another channel. Mathematically, the motivation for spread spectrum techniques can be found from Shannon's channel capacity formula:

$$C = B \cdot \log_2(1 + \frac{\sigma_C^2}{\sigma_N^2}), \tag{11}$$

where C is the maximum obtainable channel rate over an AWGN (Additive White Gaussian Noise) channel with bandwidth B and channel signal to noise ratio $\frac{\sigma_C^2}{\sigma_N^2}$.

One recognizes that the channel capacity is linearly dependent on the bandwidth B, whereas it is logarithmically dependent on the channel power σ_C^2 . In order to increase the capacity, increasing the bandwidth is obviously more efficient than increasing the channel power if one assumes constant noise power σ_N^2 in all channels. In practical systems σ_N^2 will not be constant, and for certain channels σ_N^2 is likely to be very large compared to σ_C^2. FH exploits this by averaging the performance over all used channels, and thus masks the effect of poor channels.

In section 2.2.3 it was stated that for the considered transceiver the channel could be seen as non-frequency-selective. This does not mean that the mobile radio channel is free of deep fades. On the contrary, some of the channels that the transceiver uses have carriers that are likely to be located in deep fades. In the same way as with interfered channels, FH overcomes this problem to a large extent because the transmission performance is evened out over the carriers.

Further improvement of the performance of a FH system is normally obtained by making it adaptive to the environment. This means that the system responds to the noise and fading by avoiding channels that are unfit for transmission.

2.4 Dynamic programming

Thorough introductions to dynamic programming (DP) can be found in [8] and [9], and this section is based on those sources.

In our context, DP is used to solve the problem of how much time the radio should spend on channel testing - at the expense of transmission time. That problem, and the way it has been solved, will be described more thoroughly in section 3.3. In this section, general theory related to dynamic programming is presented. This is important theory, because it provides the foundation on which the channel prediction schemes are based.

2.4.1 Definition of dynamic programming

In [9, p. 89] DP is defined as follows:

> *"The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP)."*

In order to be representable as an *Markov decision process*, a dynamic system must satisfy the *Markov property*. That property holds, for an autonomous system, if and only if its future evolution is independent of the strict prehistory. That is, the current state is what decides the immediate actions (not previous states and actions).

2.4.2 Key ideas of dynamic programming

Value functions. The key idea of dynamic programming is to use *value functions* in order to search for good policies. The notion of value functions can be explained by introducing state diagrams. The state diagram in figure 5 should be interpreted

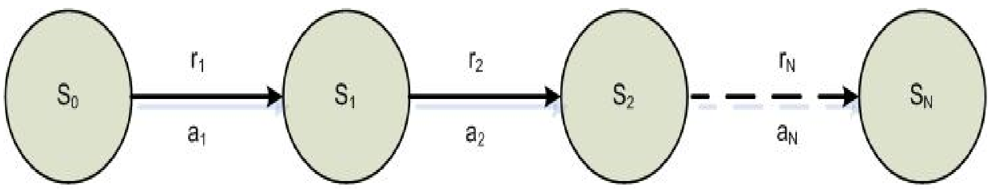

Figure 5: State diagram

as follows: a system starts in an initial state s_0, takes an action a_1 with probability $p(a_1|s_0)$, which changes the state to s_1 with probability $p(s_1|a_1, s_0)$, and generates a reward, r_1 (which may be stochastic). From state s_1 it proceeds in a similar manner until it reaches the final state s_N, and gets a reward of r_N.

The sum of all the rewards that the system receives in the long run is a stochastic variable which is called the *return*:

$$R = r_1 + r_2 + \cdots + r_N \tag{12}$$

If a system naturally stops operating after a finite number of actions, the return will be bounded. However, if it operates indefinitely ($N \to \infty$), the return cannot be calculated as in equation 12, because it may diverge. Therefore one rather works with *discounted*[8] returns:

$$R = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{k+1} \tag{13}$$

where γ is a discount factor, and $0 \leq \gamma \leq 1$.

What the system should maximize is the *expected* return. The expected return, starting at the initial state $s_0 = s$, is defined as the *value function* of s:

$$V(s) = E\left\{R|s_0 = s\right\} = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{k+1}|s_0 = s\right\} \tag{14}$$

In order to see that the value function $V(s)$ satisfies a recursive relationship, equation 14 is expanded:

$$\begin{aligned} V(s) &= E\left\{\sum_{k=0}^{\infty} \gamma^k r_{k+1}|s\right\} = E\left\{[r_1 + \gamma \sum_{k=0}^{\infty} \gamma^k r_{k+2}]|s\right\} \\ &= \sum_a p(a|s) \sum_{s'} p(s'|a,s) \left[E(r_1|a,s,s') + \gamma E(\sum_{k=0}^{\infty} \gamma^k r_{k+2}|a,s,s')\right] \end{aligned} \tag{15}$$

The first term within the square brackets of equation 15 is the expected first reward to be collected, under the condition that the current state of the system is s, the action taken is a and the next state is s'. One simplifies that term by writing:

$$E(r_1|a,s,s') = R^a_{s,s'}.$$

The second term is the discounted return that is expected when starting from state s'. Since the Markov property is assumed, the previous history (s,a) is irrelevant and therefore:

$$\gamma E(\sum_{k=0}^{\infty} \gamma^k r_{k+2}|a,s,s') = \gamma E(\sum_{k=0}^{\infty} \gamma^k r_{k+2}|s') = \gamma V(s').$$

Equation 15 can now be simplified to obtain:

$$V(s) = \sum_a p(a|s) \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V(s')\right]. \tag{16}$$

Equation 16 says that the value of s equals the expected discounted value of the next state s' plus the reward that can be expected on the way to s'. It is called the *Bellman Equation*, and it satisfies a recursive relationship, as can be seen.

[8] *Discounting* is economically motivated, and reflects that a future reward (or payment) is generally less valuable, than one that can be collected immediately.

Policy. Obviously, the value $V(s)$ depends on the probability density function (PDF) $p(a|s)$. Generally, there is one such PDF related to each state, and the set of all these PDFs determines the *policy* of the system. Thus, the policy might be seen as the system's rules for taking actions. The policy is often denoted π, and $\pi(s,a)$ should be read as: the probability of action a when in state s. Accordingly, the value function generated under policy π is often written as $V^{\pi}(s)$. Adapting to this notation, the Bellman equation can be written as

$$V^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi}(s')\right]. \tag{17}$$

Obviously, some policies can generate higher expected returns than others, and particularly interesting is the optimal policy, π^*, such that

$$V^{\pi^*}(s) \geq V^{\pi}(s), \tag{18}$$

for all $s \in S$, where S is the state space and π denotes an arbitrary policy. Before investigating how optimal policies can be generated, it is instructive to see how the Bellman equation can be written when the maximal value function $V^{\pi^*}(s)$ has actually been found. From equation 17 one has:

$$\begin{aligned}
V^{\pi}(s) &= \sum_a \pi(s,a) \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi}(s')\right] \\
&\leq \sum_a \pi(s,a) \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right] \\
&\leq \sum_a \pi(s,a) \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right] \right\} \\
&= \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right] \right\}.
\end{aligned} \tag{19}$$

Since π is an arbitrary policy (which could be the same as the optimal one), equation 19 implies that

$$V^{\pi^*}(s) \leq \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right] \right\},$$

and since $V^{\pi^*}(s)$ is the maximum value function, one obtains:

$$V^{\pi^*}(s) = \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right] \right\}. \tag{20}$$

Equation 20 is called the *Bellman optimality equation*, and from it can be seen that if $V^{\pi^*}(s)$ was known, the optimal policy π^* could easily be determined. It would simply be: in state s, choose that action, a, where the combination of the immediate expected reward and the expected discounted value of the next state, s', is maximum. The optimal value function, $V^{\pi^*}(s)$, can, in fact, be found by three different methods.

Method 1: Successive approximations. With the method of successive approximations[9], $V^{\pi}(s)$ is updated in successive steps toward $V^{\pi^*}(s)$. The update is done according to the Bellman optimality equation. To see how it works, a notation which separates different versions (updates) of $V^{\pi}(s)$ is necessary: If $V_j^{\pi}(s)$ is the most recent expected return of s, then the next and updated version will be $V_{j+1}^{\pi}(s)$.

Now, let $V_0^{\pi}(s)$ denote an initial, arbitrary and bounded expected return for all $s \in S$. Then the update is defined as

$$V_1^{\pi}(s) = \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R_{s,s'}^a + \gamma V_0^{\pi}(s') \right] \right\},$$

and generally one has:

$$V_j^{\pi}(s) = \max_a \left\{ \sum_{s'} p(s'|a,s) \left[R_{s,s'}^a + \gamma V_{j-1}^{\pi}(s') \right] \right\}. \tag{21}$$

The key property of this method is that when $j \to \infty$, $V_j^{\pi}(s) \to V^{\pi^*}(s)$. The proof is not included here, but it can be found in [8, p.35-36].

When using the method of successive approximations, there are two ways to go about. Either one obtains the jth update of the value function for each state by consistently using the $(j-1)$-th estimate, or one inserts the jth update on the right side of equation 21 as soon as it is available. Arguably, the latter way to proceed is preferred, because it usually converges faster and requires fewer arrays when implementing it on a computer. It is also the method I am most comfortable with, and have used during this project.

Method 2: Policy improvement. This method can be explained as follows. Assume that the current policy has been evaluated using equation 17, and that $V^{\pi}(s)$ therefore is known. Notice that $V^{\pi}(s)$ is obtained by averaging over all the actions where $\pi(s,a) \neq 0$. It would be interesting to see, when in s, if it is better to *deterministically* choose an action and follow the policy π thereafter. In other words, one tests if there are actions that perform better than the average. The value of s, given that action a is chosen , and π is followed thereafter is denoted as:

$$Q^{\pi}(s,a) = E\left\{ [r_1 + \gamma V^{\pi}(s')] | s_0 = s, a_1 = a \right\} = p(s'|a,s) \left[R_{s,s'}^a + \gamma V^{\pi}(s') \right]. \tag{22}$$

Obviously, if $Q^{\pi}(s,a) \geq V^{\pi}(s)$, choosing a when in s would be an improvement to the existing policy π. The policy is therefore improved, evaluated by computing the value function, and one starts the same procedure over again, for other actions and other states. One proceeds in this manner until the most recent value function is no greater than the previous one (for all states). At that point, the optimal value function has been obtained, and the optimal policy can easily be derived.

Method 3: Linear programming. This method of finding the optimal value function is based on the following observation: from the Bellman optimality equation

[9] This method is in some parts of the literature also referred to as value iteration.

(equation 20), one has

$$V^{\pi^*}(s) \geq \sum_{s'} p(s'|a,s)\left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right], \tag{23}$$

for all allowed actions a, and for all $s \in S$. This inequality is not difficult to satisfy, which can be demonstrated by adding a constant term C to the value function:

$$V^{\pi^*}(s) + C \geq \sum_{s'} p(s'|a,s)\left[R^a_{s,s'} + \gamma\left(V^{\pi^*}(s') + C\right)\right].$$

Rewriting the right side of the inequality, one obtains:

$$V^{\pi^*}(s) + C \geq \gamma C + \sum_{s'} p(s'|a,s)\left[R^a_{s,s'} + \gamma V^{\pi^*}(s')\right].$$

Observe that an increase of the value function on the left side, only results in a fraction of that increase on the right side. Accordingly, the inequality can easily be satisfied by choosing $V^{\pi^*}(s)$ sufficiently large. Therefore, the criterion for finding the optimal value function is:

$$\min \sum_{s} V^{\pi^*}(s), \tag{24}$$

under the conditions provided by inequality 23. This is a problem of linear optimization, or a so-called linear programming problem.

Remark - speed of convergence. If one considers the method of successive approximations, the speed of convergence will naturally depend on the size of γ. If one chooses γ close to 0, the optimal value function will be found in relatively few updates. The consequence of such a choice, is that the system discounts heavily. That is, during an update, it places little emphasis on the previous estimate of the value function and rather goes for the immediate reward. In the extreme case, when $\gamma = 0$, immediate rewards are all that count and the system has no ability to learn from previous events. This would result in a greedy policy which only looks one step ahead.

If one chooses γ close to one, on the other hand, the previous estimate of the value function will weigh heavily when updating, and the resulting policy will become more far-sighted. The problem with such a choice, is that it will generally take many updates until the value function converges. If the system of rewards constantly changes because it represents an environment where the terms are changing, the policy must be updated regularly, and preferably the update should be fast. Otherwise, the system will operate according to outdated policies a large portion of the time.

Obviously, the more one knows about the dynamics of the rewards, the more reasonable choice of γ can be made. If one knows that the rewards will remain stable, one should arguably choose γ close to one. If the conditions of the environment are constantly changing, such that action policies should be updated frequently, a smaller γ would be wise.

2.5 Some common interferers of nRF24Z1

This section gives a brief introduction on some common devices radiating in the ISM band. These devices are modeled as noise sources that can be included when the simulating the performance of the algorithms that are proposed, and some basic understanding of how they work is therefore necessary. The operation of many of these devices are based on the IEEE (Institute of Electrical and Electronics Engineers) standards.

2.5.1 Bluetooth

The operation of Bluetooth is based on the IEEE 802.15.1 standard. It transmits in the frequency range of 2400-2483.5 MHz in USA and Europe. There are 79 non-overlapping channels of 1MHz each from 2402 to 2480 MHz. It employs frequency hopping and jumps between channels 1600 times per second. Bluetooth uses GFSK as modulation scheme, and Bluetooth 1.0 offers data rates up to 1 Mbps. It is allowed a maximum output power of 20 dBm = 100 mW. Receiver sensitivity is minimum -70 dBm. Bluetooth employs adaptive power control.

2.5.2 WLAN

WLAN (wireless local area network) transceivers can support different data rates according to various IEEE standards. WLAN transceivers based on the standard 802.11b and the newer 802.11g both transmit in the 2400-2483.5 MHz band, and they both support two spread spectrum techniques: FHSS and DSSS (Direct Sequence Spread Spectrum). When FHSS is used, there are a total of 79 non overlapping channels (in Europe and USA) each with a bandwidth of 1 MHz and spaced 1 MHz apart. According to FCC (Federal Communications Commission) regulations a channel can be occupied 0.4 sec at most, of a period of 0.4 sec times the total number of hopping channels. When DSSS is used, higher data rates can be achieved but the channels are overlapping. For USA there are 11 channels and for Europe there are 13 channels. The channels are spaced 5 MHz apart and each channel has a spectral mask of 22 MHz. This implies that if one channel is occupied, the nearest neighbor channel that can be used simultaneously without overlap is minimum 5 channels away. WLAN based on 802.11b and 802.11g support channel agility, meaning that when DSSS is used, channel hopping may also be implemented. For both FHSS and DSSS a maximum output power of 100 mW is allowed, and receiver sensitivity is around -75 to -80 dBm. WLAN employs adaptive power control. When including WLAN as an interfering source in our simulations it is assumed to operate in DSSS mode and therefore it is regarded as a broad band interferer.

2.5.3 Microwave ovens

Most residential microwave ovens operate at around 2450 MHz. Leakage of radiation is mostly caused by a damaged or badly fitted door. The maximum allowed leakage power, upon manufacture, according US food and drug administration, is 1 mW/cm^2 a distance of 5 cm from the microwave oven. Assuming isotropic radiation, this corresponds to powers of 2.5 μW/cm^2 which is -56 dBm/cm^2 at one meter distance, and 0.625 μW/cm^2 which is -62 dBm/cm^2 at 2 meters distance. Older and used microwave ovens are likely to radiate significantly more power than described here.

The leakage emission bandwidth from microwave ovens is largest during the start-up of operation and it can be as large as 20 MHz [10]. Under normal operation the leakage emission bandwidth is only a few MHz. Commercial microwave ovens which are used in restaurants etc. have a much larger leakage emission bandwidth. It may cover tens of MHz, and it thus affects much larger part of the ISM band than its domestic counterpart [11]. It also radiates with higher power. When including microwave ovens in the simulations, the commercial type has been assumed. It has therefore been modeled as a broad band, high power interferer.

2.5.4 ZigBee

ZigBee transceivers operate according to the IEEE 802.15.4 standard. The 802.15.4 standard divides the frequency range 2400-2483.5 MHz in 16 non overlapping channels spaced 5 MHz apart, each channel occupying a bandwidth of 2 MHz. The 802.15.4 standard dictates a minimum capable output power of 0.5 mW = -3 dBm. Maximum output power should be specified by local regulations. Maximum output of 0 dBm is common. Receiver sensitivity is specified to -85 dBm or better. 802.15.4 uses offset-QPSK (Quadrature Phase Shift Keying) as modulation technique. ZigBee transceivers may employ FHSS and DSSS as spread spectrum techniques or a combination of both. 802.15.4 dictates that adaptive power control shall be used. When including a ZigBee transceiver as an interfering source in the simulations, it is assumed that it uses FHSS as its spread spectrum technique.

3 Method

This section presents how the channel prediction schemes work. Since all the tests and simulations are done in a channel model (a computer program) that tries to mimic the effects of a real wireless channel, the key features of that model are of relevance. Therefore, this section starts with a presentation of how it works. The currently implemented FH scheme of the nRF24Z1 is also presented, together with the different channel prediction schemes that are proposed here.

3.1 Description of the channel model

In channel modeling, it is not unusual to include random variables as parameters of the model. In [12], a statistical model for indoor propagation is presented where both the path loss and the delay spread are random variables. In our channel model, random parameters are not introduced. The motivation is to have a model where the received signal is purely a result of combining multi-path components, each attenuated according to distance and reflector properties. This is desired because it gives a large degree of control. With a random and varying path loss exponent for instance, it might be difficult to specify whether poor performance of a system is due to large path loss exponent or due to an FH algorithm that does not work as intended. Therefore a deterministic model has been developed, and randomness has only been introduced with the casual movement of human obstacles.

3.1.1 The scenario of operation

It is desired that a channel model can represent challenging conditions with respect to interference. Such a simulation setting is provided by the "Starbucks scenario". It refers to the typical situation in a Starbucks coffee shop, and implies a confined space with many customers and variety of ISM band applications in use. Some applications may be:

- WLAN
- cell-phones with Bluetooth-based earplugs
- audio streaming from MP3 player to headsets
- commercial microwave ovens

The Starbucks scenario may be described by high interferer density, high dynamics and high interference power. Our channel model is based on this scenario, but in addition the model provides flexibility to simulate less challenging conditions. This can be done by reducing the number of interferers, limiting dynamics and setting obstacle density to whatever level is wanted.

3.1.2 MATLAB as basis for the channel model

The channel model is implemented in MATLAB which is a powerful program with respect to matrix algebra. It uses a three-dimensional matrix as a model of a room.

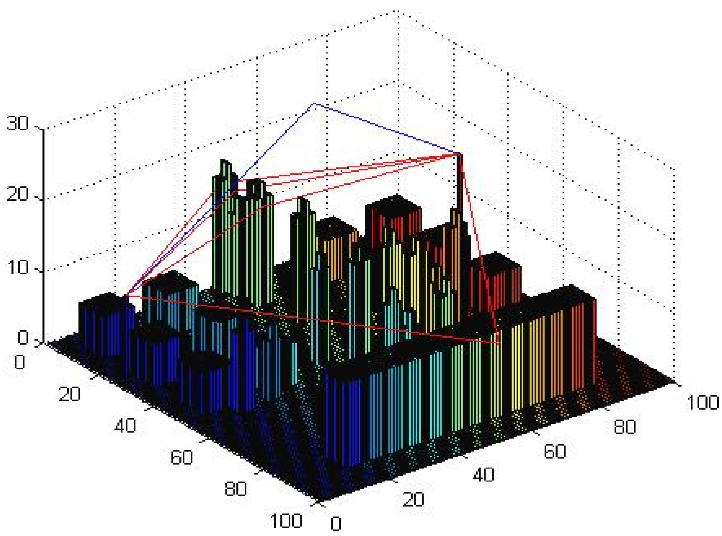

Figure 6: Visualization of the channel model with multi-path propagation

The entries of the matrix at coordinates (x, y) represent the height of objects. When the height is zero, there is no object present. Objects can thus be seen as non-zero sub matrices within a larger matrix. Dynamics are introduced by shifting the sub matrices to vacant entries. Using a matrix as basis for the channel model enforces a rectangular geometry on the room and on objects. The contour of an object forms planes that are perpendicular to the floor or to the walls. Furthermore, planes perpendicular to the floor are always parallel to either a row or a column. Together with the limitation that the model is restricted to consider first order reflections only, this greatly reduces the number of possible reflection points, and thus provides a simplification that is desired to reduce the complexity. Figure 6 visualizes the channel model. The reflection paths that the model considers are also illustrated. For further details about the channel model, such as how reflection points and reflection coefficients are determined and what fading characteristics it provides, [1] contains a thorough description. Here, it suffices to mention that the model has approximately the same fading characteristics that can be empirically obtained for a real mobile channel.

3.2 The FH scheme currently implemented in nRF24Z1

The nRF24Z1 has 38 frequency hopping table registers which dictate the hopping pattern. Theoretically, this can be any pattern, but in practice the pattern is chosen in a systematic manner. If one assumes channel numbers from 1 to 38, a routine that generates a systematic pattern is given by:

```
if (channel+nr) <= 38
channel=channel+nr
else
channel=(channel+nr)modulo38
end
```

Here, `nr` represents the number of channels that the FH algorithm jumps every time it updates to a new channel, and a channel update takes place every time a frame has been transmitted. The only criterion that `nr` must fulfill, is that it must be chosen in such a way that all channels will be used. This is guaranteed when `nr` and 38 are relatively prime numbers, that is, the greatest common divisor is 1. However, one might have additional requirements to the hopping length which accommodates the system to the expected noise. Setting `nr` = 11, for instance, will make the transceiver jump 11 channels, corresponding to 22 MHz, when updating. This could be beneficial because 22 MHz is approximately the spectral mask of a WLAN transceiver.

The nRF24Z1 features adaptive FH. Adaptivity is provided by a channel banning list which can hold up to 18 banned channels. A channel is banned when the quality of a received frame is unsatisfactory. A frame contains a varying number of packets depending on the transmission quality and the rate (see section 2.1). If less than 3 packets in the frame are readable, the channel is banned. When the banning list is full, the oldest channel in the list must be released before new channels can be banned. Therefore, channels are released by the FIFO principle.

3.3 The proposed channel prediction schemes

Before presenting the proposed channel prediction schemes, it is appropriate to look more closely at the problem of how much time the radio should spend on channel testing - at the expense of reduced transmission time. This problem deserves attention because its solution in principle defines how the channel prediction scheme operates. In fact, the solution to the problem might be seen as the channel prediction scheme itself.

3.3.1 Description of the underlying problem

Consider a nRF24Z1 transceiver pair under operation. If the radio operates in presence of much noise, transmitted frames are frequently distorted. After some time, it is likely that the receive buffer holds few packets, and in the worst case, runs empty. Obviously, that should be avoided, and the technique that is proposed here is to transmit a test packet from TX to RX before transmitting an entire frame. If the test packet is readable, the channel is interpreted as good and the frame can be sent. Otherwise, the channel is considered unsuited and it is not used.

Under very noisy conditions, it is possible that a number of test packets must be transmitted, on different channels, in order to find a good one. The advantage of multiple testing is that the probability of finding a good channel increases. The drawback is that for each transmitted test packet, the extra capacity in the frame (packets in addition to the compulsory 8) is reduced. In the extreme case, if TX consistently tests channels to an extent that there is no room for extra packets, a gap in the receive buffer will never be refilled. That will result in an audible error for the user of the equipment.

3.3.2 Desired result

An optimal trade-off between channel testing (transmission of test packets) and, at the same time, maintaining some extra capacity in the frames, is wanted. In order to simplify the problem, all channels are assumed to have equal probability, p, of being poor. Furthermore, a poor channel is assumed to distort the entire frame, and a good channel distorts no part of the frame.

Firstly, an optimal policy which dictates how many test packets, k, should be transmitted when p is known, is wanted. Secondly, an optimal policy which gives k as a function of both p and the inventory I (the number of packets present in the receive buffer) is desired. DP provides techniques that can be used to find these policies.

3.3.3 Proposed solution

Notation

$p :=$ probability of a channel being poor (for example $p = 0.1$).
$k :=$ number of test packets/channel tests.
$n :=$ default number of packets in a frame ($n = 8$).
$\bar{I} :=$ max number of packets in the receive buffer($\bar{I} = 40$).
$I :=$ Inventory; actual number of packets in the receive buffer: $I \in \left[0, 1 ... \bar{I}\right]$.
$\bar{e}_k :=$ max number of *extra* packets (in addition to n) in a frame, given k transmitted test packets, and is defined as

$$\bar{e}_k = \left\{ \begin{array}{l} 13, \text{ for k=0} \\ 12, \text{ for k=1} \\ 10, \text{ for k=2} \\ 8, \text{ for k=3} \\ 6, \text{ for k=4} \\ 5, \text{ for k=5} \\ 3, \text{ for k=6} \\ 1, \text{ for k=7} \\ 0, \text{ for k=8} \end{array} \right\}, \tag{25}$$

at a rate of 32 KHz. For the two other rates, the function $\bar{e}_k$ is defined differently.
$e :=$ actual number of extra packets in the frame: $e \in \left[0, 1 ... \min(\bar{e}_k, \bar{I} - I)\right]$.
$C^{\bar{e}_k}(I) :=$ cost function[10]; the expected cost of having I packets in the receive buffer when maximum $\bar{e}_k$ extra packets are allowed in the frame.

Solution with DP

Actions and states. In order to solve the problem with DP, the possible actions and next states must be clarified. Assume that the number of packets in the receive buffer is I. Because all channels are assumed to have identical properties, only I defines the state of the system, and correspondingly there are 41 different states.

[10] This is equivalent to a value function $V^{\pi}(s)$; only here one seeks to *minimize* the sum of expected *penalties* rather than maximizing the sum of expected rewards.

Given that k channels have been tested, the system can transmit any number of extra packets, e, within a frame, in addition to the compulsory n default packets. Those are the possible actions.

If all the k tested channels are poor, it is proposed that the radio shall simply use the next untested channel from the hopping sequence to transmit the frame. The probability of ending up using a poor channel is therefore p^{k+1}. A poor channel will result in an unreadable received frame, and the buffer will be reduced to $I-n$, which is one of the possible next states of the system[11]. Upon such an event the system receives an immediate penalty of $P(I-n,e)$, which will be explained shortly.

In the same manner, the probability of using a good channel is $1-p^{k+1}$. If the channel is good the frame will be readable at the receiver, and the buffer will be increased to $I+e$. That is the other possible next state of the system, which leads to a (smaller) penalty of $P(I+e,e)$.

Cost function. An expression for the optimal cost function $C^{\bar{e}_k}(I)$ can now be formulated. Using the Bellman optimality equation (equation 20 in section 2.4.2), the cost function is

$$\begin{aligned} C^{\bar{e}_k}(I) &= \min_e\{p^{k+1}\left[P(I-n,e)+\gamma C^{\bar{e}_k}(I-n)\right] \\ &+ (1-p^{k+1})\left[P(I+e,e)+\gamma C^{\bar{e}_k}(I+e)\right]\}, \end{aligned} \tag{26}$$

and the update procedure, according to the method of successive approximation (equation 21), is

$$\begin{aligned} C_j^{\bar{e}_k}(I) &= \min_e\{p^{k+1}[P(I-n,e)+\gamma C_{j-1}^{\bar{e}_k}(I-n)] \\ &+ (1-p^{k+1})[P(I+e,e)+\gamma C_{j-1}^{\bar{e}_k}(I+e)]\}. \end{aligned} \tag{27}$$

Penalties. How the penalties are chosen, can greatly affect the resulting policy. In our case, it is sensible that the penalty is large when the next state corresponds to few packets in the receive buffer, and conversely, the penalty should be relatively small if the next state results in a buffer that is almost full. In addition, when the system chooses to test channels to an extent that there is no room for extra packets, it should be penalized additionally. Therefore the following penalty function is proposed:

$$P(I,e) = (\bar{I}-I)^2. \tag{28}$$

And subsequently, the following (pseudo)code takes care of the additional penalty:

```
if e==0
P(I,e)=100*P(I,e)
end
```

Equation 28, in combination with this code has been used when calculating the cost functions in this project. Because the penalties are time invariant in our context, $\gamma = 0.95$ has been used.

[11] Actually, negative values of I are not allowed, and the correct next state would be $\max[0, I-n]$, but $I-n$ is written here for simplicity.

Remark - policies versus cost functions. One should keep in mind that, in our case, the optimal cost function is actually of secondary interest. It is only a means to determine what the system really needs: the best policy. Furthermore, it is rather common that the best policy can be determined long before the optimal cost function is obtained. As an example, a computer program might require 500 iterations (updates) for each state in order to obtain a cost function which is close to the optimal one. In the process, one might find that the optimal policy was obtained after only 50 iterations, and that the subsequent 450 were actually superfluous. This happened when the proposed prediction schemes were developed, and another example illustrating this can be found in [9, p. 92-94].

3.3.4 Channel prediction scheme 1: k as a function of p

In this case, it is desired to find a policy which dictates how much channel testing should be done when the channel ineptness probability is known. The idea is to implement such a policy as a simple look-up table in the radio. By estimating p regularly, the radio can read out a corresponding value for k, which is the number of test packets to be transmitted.

Now, the focus is on how to choose an optimal k, given p. The procedure to find this is quite straightforward. It is listed stepwise below:

1. Choose a fixed value for p. (For example $p = 0.1$).

2. Compute the optimal cost function for each possible value of k by repeating the update procedure described in equation 27 for each I until convergence. For instance, at the rate of 32 KHz, this implies nine different cost functions, because $k = 0, 1, .., 8$. At the two other rates fewer cost functions will result.

3. From each cost function, determine how many extra packets it is optimal to transmit for each buffer size[12].

4. For each of the k cost functions, *simulate* a transmission, by consistently transmitting frames with the optimal number of extra packets found in the previous step. Observe the corresponding development of the receive buffer. After each transmitted frame in the simulation, note the buffer size, I, and record the corresponding cost, $C^{\bar{e}_k}(I)$.

5. When all the k simulations are terminated, sum up the recorded costs for each simulation.

6. The best k to choose is that k which corresponds the minimum accumulated cost.

7. Go back to the first step, choose a new value for p, and proceed as before.

[12] Actually, when evaluating the cost function, the result was quite trivial: As long as the receive buffer misses more than $\bar{e}_k$ packets, add $\bar{e}_k$ extra packets to the frame. If the receive buffer misses less than, or equal to $\bar{e}_k$ packets, add $\bar{I} - I$ extra packets to the frame. In other words, with respect to extra packets: always transmit as many as allowed. This general rule proved to remain unaffected by the values of p and k.

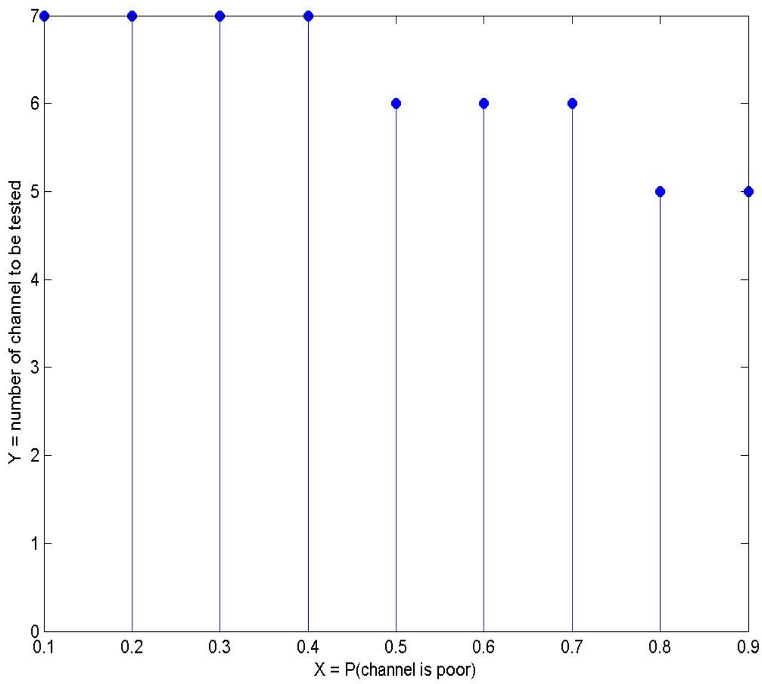

Figure 7: Look-up table: k as a function of p (rate: 32 KHz).

8. Continue in this manner until k has been determined for as many values of p that are wanted.

If one sets $p = 0.1, 0.2, ...0.9$, and for each value of p one finds a corresponding k, one has a simple look-up table that can be stored in the radio. In order to use it sensibly, the radio must estimate p periodically. Briefly, that is how the first of the proposed channel prediction schemes works.

Figure 7 shows the look-up table for 32 KHz. Notice that, according to this table, the radio shall test less channels, when the transmission conditions become worse. This makes sense, because under poor transmission conditions the number of packets in the receive buffer is likely to be reduced. The ability to refill the buffer therefore becomes more important than testing channels.

3.3.5 Channel prediction scheme 2: k as a function of (p, I)

Intuitively, it would make sense that when the receive buffer is almost full, a large capacity for extra packets in the frame is not critical, and therefore more channel testing could be performed. Similarly, when the buffer is close to running empty, one would want to refill it with as many extra packets as possible, and less channel testing would be done. Therefore, one could imagine that the number of channels the radio should test, should not only depend on p but also on how many packets there are in the receive buffer I. Such a policy is what is desired here.

In this case, in order to obtain the optimal cost function, one follows the update procedure in equation 27, with a slight modification:

$$\begin{aligned} C_j^{\bar{e}_k}(I) &= \min_{e,k}\{p^{k+1}[P(I-n,e)+\gamma C_{j-1}^{\bar{e}_k}(I-n)] \\ &+ (1-p^{k+1})[P(I+e,e)+\gamma C_{j-1}^{\bar{e}_k}(I+e)]\}. \end{aligned} \tag{29}$$

That is, one chooses that number of extra packets e, and test that number of channels k, who together minimize the expected penalty and subsequent expected cost. The stepwise procedure to find the desired policy then becomes:

1. Choose a fixed value for p. (For example $p = 0.1$).
2. Compute the optimal cost function by repeating the update procedure described in equation 29 for each I until convergence.
3. For each value of I, note the corresponding k and how many extra packets it is optimal to transmit.[13]
4. Go back to the first step, choose a new value for p, and proceed as before.
5. Continue in this manner until k has been determined for each value of p.

Note that no simulation is done to generate the policy. That is because multiple cost functions do not have to be compared in this case, as there is only one.

If the cost function is calculated for each allowed value of p, ($p = 0.1, 0.2, ...0.9$), one obtains k as a function of (p, I). It can also be stored in the radio as a look-up table, and again, the radio must estimate p regularly in order to benefit from it. Briefly, this is how the second of the proposed channel prediction schemes works.

In figure 8, k is shown as a function of p and I for 32 KHz. It shows what could be suspected: when the buffer is nearly full the radio should give priority to channel testing, and when the buffer is close to empty it should test less and give priority to transmitting extra packets.

Remark. In order to avoid confusion, it should be mentioned that in the proposed prediction schemes, the intention is to compute look-up tables only once and store them in the radios. Accordingly, it is not the idea that the radio shall update and compute new look-up tables regularly under operation. That would also be possible, but it would require much higher processing capability and more memory than is available in the nRF24Z1 today.

3.3.6 Practical considerations and implementation

Proposed frame structure. Because a variable number of channel tests and extra packets must be included within a period, the frame structure in figure 2 on page 7, must be modified. The suggested structure is shown in figure 9, for a period of 2.67 ms (rate: 48 KHz). In the first part of the frame, 8 default packets are transmitted. If there are extra packets, they follow next, and after that, the channel testing is

[13] Also here, when evaluating the cost function with respect to extra packets, the result was the same as in section 3.3.5: always transmit as many extra packets as allowed.

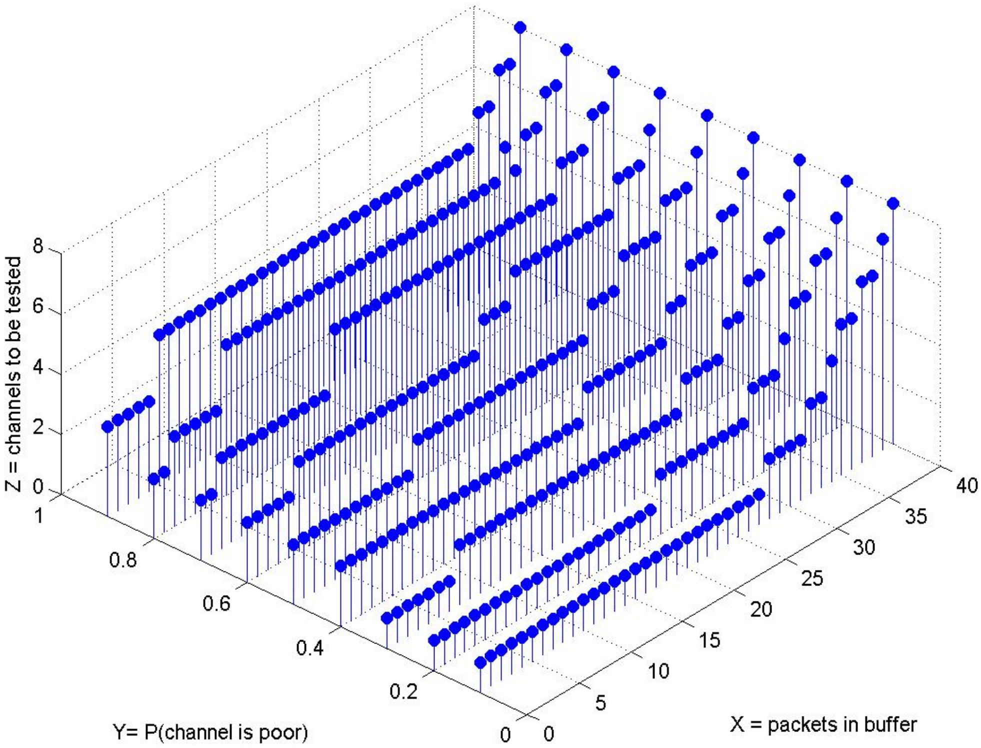

Figure 8: Look-up table: k as a function of (p, I) (rate: 32 KHz).

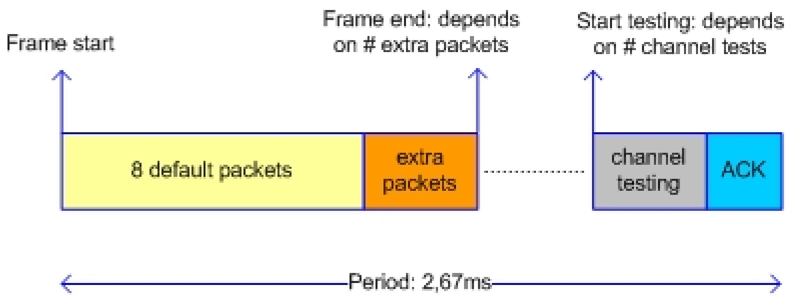

Figure 9: Proposed frame structure (rate: 48 KHz).

done. Upon reception, RX transmits an ACK to TX, before a new frame can be transmitted. Note that the 8 default packets are always transmitted first in a fixed and reserved part of the frame. One could argue that the extra packets should be transmitted first because they are most urgently needed at RX. Then, the 8 default packets should follow and finally the channel testing should be done in order to test channels as close to usage as possible. This is however not practical, because it would imply that the 8 default packets of the frame would have floating start and end points depending on the number of extra packets before, and the number of channel tests after. If RX and TX have different ideas about the frame structure (due to loss of ACKs, which will be explained shortly), the 8 default packets might be expected at RX at a different time than TX transmits them, and the result is that packets are lost. Therefore, it is proposed to reserve a fixed part of the frame for the 8 default packets in order to reduce possible confusion and packet loss.

Constraints. The channel prediction schemes presented in sections 3.3.4 and 3.3.5 suggest that before transmitting, k channel tests are performed, and one of the suitable channels are immediately used to send the next frame. Due to practical constraints, however, this is not a realistic implementation. In order to indicate the main problem with such a technique, an example can be considered.

Assume that TX and RX test channels, and that one or more of them are suitable. In order for RX and TX to agree on which one to use, TX must be informed about which of the test packets seemed most promising at RX. Therefore, RX could include that information in the ACK that it sends after the frame has been received. The problem is that RX will not know whether the transmitted ACK has arrived TX in readable shape or not. If TX transmits an ACK back to confirm that the acknowledgment was OK, the same problem occurs, and the result is potentially a never-ending series of ACK packets back and forth. This is a well known problem in network theory, and is often referred to as the *two army* problem. It can be tackled by introducing a three-way handshake with timers on the ACKs, as suggested in [13, p503-506]. In our case however, this is not a particularly attractive option because, in addition to complicating the protocol, transmission of multiple ACKs is time consuming and reduces both the available transmission time and the number of channels that can be tested.

Channel lists and test lists. Due to the above mentioned constraints, RX and TX use another solution to communicate which channel to use: they both maintain *channel lists* of four to six tested and ready-to-use channels depending on the rate (the period length). Such lists already exist in the current protocol of the nRF24Z1. Here, it is suggested that they also maintain *test lists* on which channels to test next. The lists are proposed used as follows.

Before sending a frame, the oldest channels on the channel lists of both RX and TX are read out, and used for the transmission. After such read outs, RX and TX have vacancies in their channel lists. If the read out channels are not the same, one can be sure the audio packets will have to be retransmitted. If they are equal, the noise and fading determines whether the packets are received or not. In the subsequent channel testing, channels are read out from the test lists, again leaving vacancies. These channels are used to transmit test packets. Also here, test packets can only be interpreted as suitable if RX and TX test the same channels. Those tested channels that RX sees fit, enter the vacant spaces of RX's channel list. RX's test list is refilled locally, using the code described in section 3.2[14]. In addition, RX determines the number of channels to be tested during the next frame by considering its estimate of p (and I, if the scheme in section 3.3.5 is used), and using the look-up table. TX has a replica of that table.

After each received frame, RX transmits an ACK to TX. A replica of RX's channel list, RX's test list, the number of packets in the receive buffer (I), and RX's estimate of p are included in that ACK. If the ACK is readable at TX, this included information overwrites TX's version of the same information. Correspondingly, if TX receives a readable ACK, and the included channel list contains one or more channels, there is no confusion about which channel to use for the next frame, and which channels to test next. If the ACK is lost, the situation is generally not hopeless because TX still has a channel list - it only has vacancies because it was not refilled. In this case, TX will refill its test list locally the same way RX does, and determine k by using its most recently received p (and its most recently received I if the scheme in section 3.3.5 is used). If the transmission conditions are not too poor, it is likely that TX will receive a readable ACK in due time, such that RX and TX again possess the same information.

By using this technique, tested (predicted) channels are generally not used immediately, but they will be used within the period that is defined as sensible to predict, as explained in section 2.2.4. That is also why the length of the channel lists depend on the period length.

Interaction between TX and RX. In the schemes that are proposed here, RX decides which channels to test and use, and TX adapts to the information in the ACK. In order to visualize how they interact, let us consider a transmission example.

If ACKs are lost, TX may transmit a frame structure that is different from the one RX expects. The reason is that TX must use its look-up table locally, possibly with different (older) values of p and I than those RX uses. Accordingly, RX and TX may operate with different versions of k, and the expected structures of the following

[14] RX uses the channel that most recently entered in the test list as the initial channel, and runs through the code multiple times until the test list is filled up.

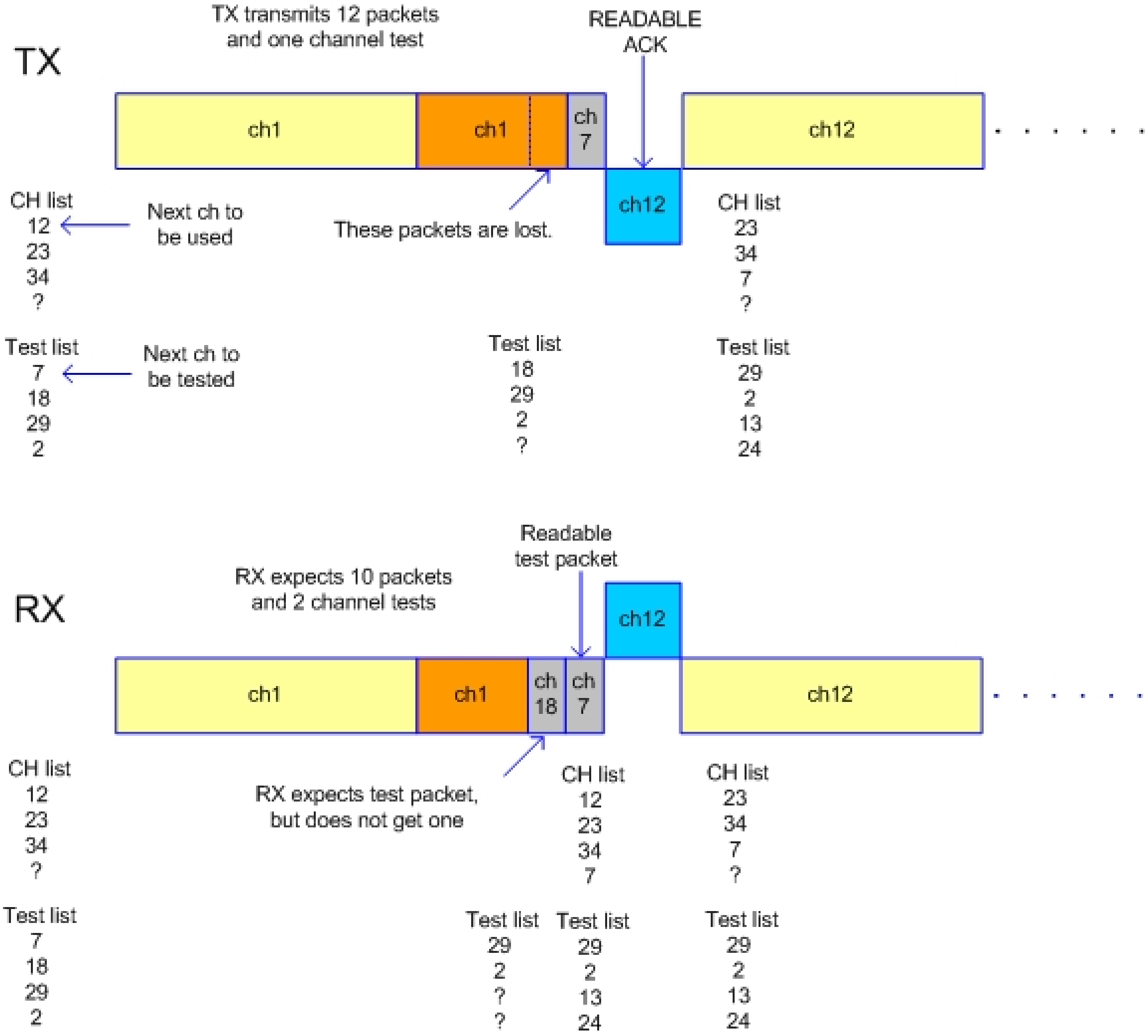

Figure 10: Transmission example (rate: 48 KHz).

frame will differ. To be more specific, in this example it is assumed that the period length is 2.67 ms, and that TX transmits a frame that has 12 packets and includes one channel test, but RX expects a frame containing 10 packets and two channel tests. This is illustrated in figure 10.

From figure 10 one sees that the channel lists and test list can hold four channels each. There are vacancies in the leftmost channel lists, indicated by question marks, because channel 1 has been read out to use for the frame transmission. At this point, it is assumed that RX and TX have equal test lists. Upon reception of the 10th packet, RX goes into channel testing mode. RX reads out the two oldest channels from the test list, leaving two vacant spaces. Notice that the order of which RX expects to test channels is crucial. If RX expected to test the two channels in reverse order (first channel 7 and then channel 18), one could be sure that RX would receive no test packets.

Because TX is still transmitting audio packets on channel 1, RX does not receive a test packet on channel 18. The consequence is that both the audio packets are

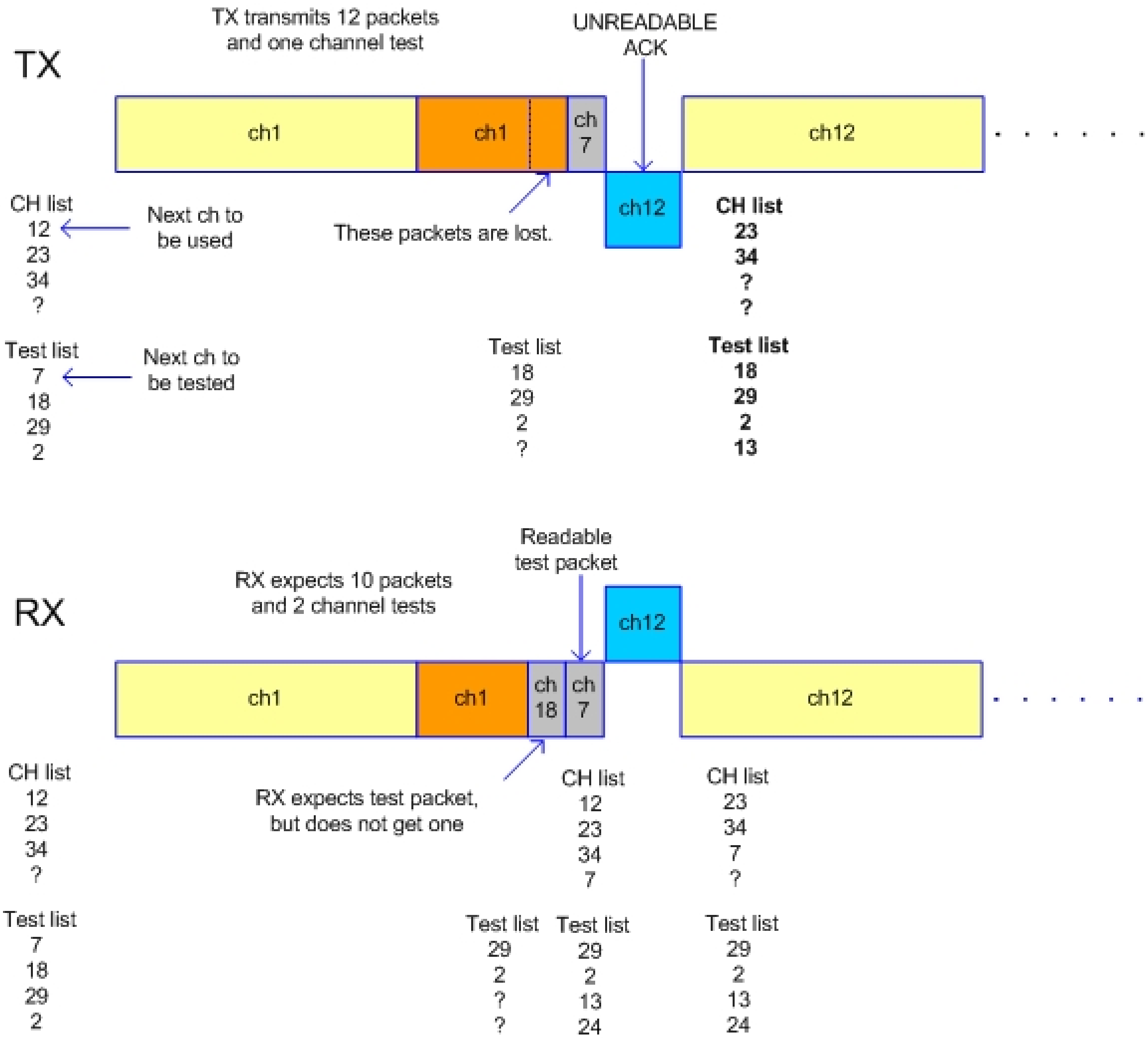

Figure 11: Transmission example - ACK is lost (rate: 48 KHz).

lost and channel 18 is interpreted as useless. When TX has transmitted 12 audio packets, it changes to channel testing mode. The oldest channel on its test list is read out leaving a vacancy. The subsequent test packet is transmitted on channel 7 and received in readable condition at RX. Therefore, RX notes channel 7 as OK, and inserts it in the vacant place of its channel list. RX also refills its test list by adding 11 (the hopping length) to the most recently entered channel in the list until it is full, as does the code described in section 3.2.

Before the ACK is transmitted, RX and TX jump to a new channel (channel 12) simultaneously. The channel is read out from the channel lists as before. Notice that, before transmission of the ACK, RX and TX have different channel and test lists. Because the ACK is successfully received at TX, the lists are identical shortly after, and during the subsequent frame there will be no confusion about frame structure.

Let us also consider what would happen if the ACK was unreadable at TX. That situation is illustrated in figure 11. In this case, because TX gets no readable information from RX in the ACK, channel 7 does not enter TX's channel list. The consequence is that TX's channel list has more vacancies than RX's, but there is still no confusion about which channel to use next, and there will not be for the two next frames either.

TX's test list is in this case generated locally, the result being that it differs from RX's test list on all entries. Therefore, the channel testing that follows, will actually be a waste of time. If the next ACK is also not readable at TX, there will only be one channel left in TX's channel list (channel 34). In this case, it is proposed that TX simply fills its channel list with untested channels using the code in section 3.2. The resulting channel list would then be: $[34, 7, 18, 29]$. RX does the same thing if RX's channel list also contains only one channel. This will in many cases provide a sequence of next channels to use, that are identical at RX and TX, at least for some time. The channel lists may, however, also differ after such local refills. That will happen if the single channel that remains on top of one of the channel list before a refill, differs from the channel on top of the other channel list. In that case, the receive buffer will generally run empty rather quickly.

If 8 subsequent packets are missing when they should be read out from the receive buffer, it is proposed that RX initiates a link re-initialisation procedure. This involves filling up the receive buffer and filling the channel lists at both RX and TX. The procedure brings the system back to the optimal state, but it will represent an audible error to the user of the equipment.

3.3.7 Channel prediction scheme 3: k as a function of the length of the channel list

Both prediction schemes that have been presented so far, use channel and test lists. One could imagine that, rather than considering parameters such as p and I to determine how much channel testing should be performed, the system can simply concentrate on maintaining a channel list which is as full as possible.

The principle of operation would then be, if the list is full, minimum testing is performed just to keep it that way, and if it is nearly empty more testing is done in order to refill it. Such a scheme would eliminate the need for DP and look-up tables all together, and is therefore attractive. Basically, that is what the last proposed channel prediction scheme does. If the ACK from RX to TX is lost TX will determine how many and which channels to test locally, by regarding its own channel and test lists, otherwise TX will have the same information as RX and there will be no confusion about the frame structure.

In order to maintain the channel list, a minimum of one channel test will always be performed, even when the list is full. When the list holds no channels, most channel testing is done. At the rate of 32 KHz, the maximum number of channel tests according to equation 25 is 8, but then the frame has no extra capacity. In order to avoid that, 7 channel tests is defined as the maximum allowed at the rate of 32 KHz. Similarly, at 48 KHz and 44.1 KHz, the largest number of channel tests, which also allows extra packets, is defined as maximum.

4 Simulations and results

This section presents the simulation results. They are obtained by testing the performance of the various FH algorithms in the channel model described in section 3.1. Five different FH schemes are simulated under various levels of noise and dynamics. The schemes are:

- *Ban list*: The scheme currently implemented in the nRF24Z1.
- *Ban ratio*: The modified scheme presented in section 1.1.
- *Ch. list*: The prediction scheme presented in section 3.3.7.
- $k = f(p)$: The prediction scheme presented in section 3.3.4.
- $k = f(p, I)$: The prediction scheme presented in section 3.3.5.

Results are, in addition to being presented, also commented and interpreted. Interpretations are instructive because they illuminate and explain key characteristics of the various FH schemes.

4.1 Graphical results from simulations

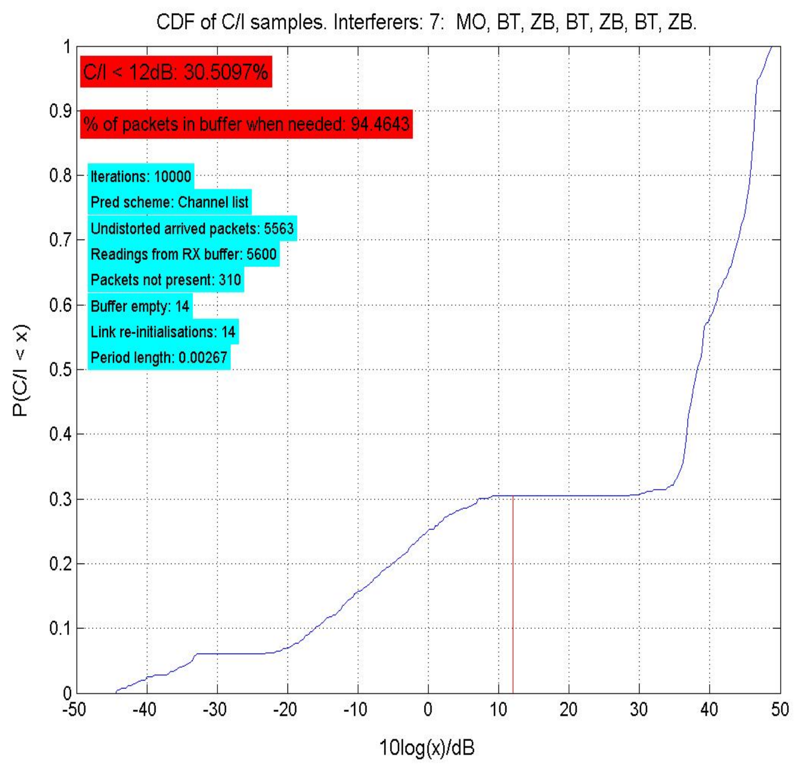

Figure 12: Graphical presentation of results

Figure 12 shows how the results are displayed upon simulation. It illustrates a cumulative distribution function (CDF) where the x-axis represents the C/I of the received audio packets at RX. The y-axis shows the share of packets with C/I less than x. An example demonstrates how the graph should be read: during the simulation, approximately 25% of the received packets had C/I less than 0 dB, and around 30.5% of the received packets had C/I less than 12 dB (12 dB is the quality threshold required by the radio).

As can be seen in the title of the plot, there were 7 interferers during this simulation: one microwave oven, three Bluetooth sources and three ZigBee sources. In spite of more than 30% of the received packets being unreadable at RX, the receive buffer provides time to retransmit, and the result is that around 94.5% of the packets are readable at RX in time to be read out of the buffer.

After an initialisation phase, to get stable conditions, the simulation consisted of 10 000 iterations, and during each iteration, either an audio packet is transmitted, a channel is tested or an ACK is sent from RX to TX. The FH scheme tested during this simulation is "Pred scheme: Channel list", which refers to the channel prediction scheme described in section 3.3.7. There are a total of 5563 packets that have arrived at RX in readable shape, and 5600 readings from the buffer have been performed. Note that a reading is counted regardless of whether the packet is present or not. In 310 of the readings, the needed packets were not present, and in 14 of these cases the buffer was totally empty. Also, in 14 cases, 8 subsequent packets were not present in the buffer, and accordingly RX launched the link re-initialisation procedure 14 times. The simulation was done using an audio sampling rate of 48 KHz, and therefore the period length is 2.67 ms. In addition, which can not be seen from figure 12, the room size during this simulation was $10 \cdot 10 = 100$ m^2, the maximum number of people in the room were 9, and both people and radios were allowed to move.

4.2 Levels of interference and parameters of interest

In the following, all the considered FH schemes are tested under three different levels of interference. These are:

- *Low* interference: refers to 1 WLAN transceiver, 1 Bluetooth transceiver and 1 Zigbee transceiver.
- *Medium* interference: refers to 1 WLAN transceiver, 2 Bluetooth transceivers and 2 Zigbee transceivers.
- *High* interference: refers to 1 WLAN transceiver, 3 Bluetooth transceivers and 3 Zigbee transceivers.

The WLAN transceiver has been selected as broad band interferer, because it is generally more common than a microwave oven.

When comparing the performance of various FH schemes, some of the information provided in figure 12 is more important than the rest. Therefore, in what follows, only the following parameters are considered:

- The percentage of packets that are unreadable at RX.
- The percentage of packets that are in the buffer when needed.

Parameter	FH scheme				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	13.4	11.3	12.4	16.6	11.4
% of packets present in buffer when needed	98.9	99.2	100	99.9	99.8
# re-initialisations	0	0	0	0	0

Table 3: Performance in a static environment with stable and low interference.

- The number of link re-initialisations.

Furthermore, tables will be used in order to compare FH schemes because they are more convenient than plots like the one shown in figure 12.

4.3 Simulations in a static environment

In this first set of simulations, no dynamics are introduced. All interferers are static, TX-RX are static and people are not allowed to move. Such simulations are interesting because the obtained results are functions of the interference only, and not of varying transmission conditions due to moving obstacles and reflectors. Thus, the results indicate the various FH schemes' ability to suppress interferers.

When simulating in a static environment, the channel model provides periodic noise. The reason is that the narrow band interferers (Bluetooth and ZigBee) are modeled to use FH algorithms that have constant hopping patterns. Accordingly, they loop through the same set of channels, in the same order, repeatedly. In addition, the broad band WLAN interferer is set to operate either at one constant carrier, or to shift periodically between two different carriers throughout the simulation.

Due to periodic noise, a limited number of iterations are needed for the simulations. 10 000 iterations have been used for all the simulations that are presented in this section.

4.3.1 Simulations in a static environment - stable interference

Stable interference refers to the case when the broadband interferer uses a constant and single carrier frequency. In addition, narrow band interferers are included in the simulation. Tables 3, 4 and 5 show the obtained simulation results for a static environment with increasing levels of interference. The room size is 100 m^2, TX-RX separation is 10 m, and the period length is 2.67 ms.

Interpretations and comments. From tables 3, 4 and 5 a few observations can be made. Firstly, as the number of interferers increases, the performance drops for all schemes, which is to be expected.

In table 3, which represents the least interference, there is hardly any difference in performance at all. Whether one goes for banning schemes[15] or prediction schemes[16],

[15] *Banning schemes* refers to "Ban list" and "Ban ratio" in the tables, because they both employ ban lists.

[16] *Prediction schemes* refers to "Ch.list", "$k = f(p)$" and "$k = f(p, I)$" in the tables.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	27.6	21.8	23.3	23.4	25.9
% of packets present in buffer when needed	97.7	98.7	98.0	97.3	97.8
# re-initialisations	0	0	1	4	3

Table 4: Performance in a static environment with stable and medium interference.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	36.6	32.8	29.8	31.6	39.3
% of packets present in buffer when needed	94.4	97.1	91.1	91.7	90.7
# re-initialisations	3	0	8	13	9

Table 5: Performance in a static environment with stable and high interference.

is not critical because they all perform well. The high performance can be explained as follows.

The two banning schemes perform well because they identify the poor channels rather quickly, and ban them for usage thereafter. In addition, poor channels are, on average, relatively seldom released. The reason can be found when considering the bandwidth of the interferers. A WLAN transceiver has a spectral mask of 22 MHz, Bluetooth 1 MHz, and ZigBee 2 MHz. Accordingly the interferers will occupy, at maximum, an instantaneous bandwidth of $22 + 2 + 1 = 25$ MHz. The ban list in the nRF24Z1 has room for 18 channels. That corresponds to 36 MHz of the spectrum, and therefore, releases are relatively rare.

The three prediction schemes also perform well, because the predictions are rather accurate in this context: firstly, the WLAN transceiver occupies almost 30% of the available spectrum, all of the time. Accordingly, all transmitted test packets in this part of the spectrum are distorted, and the corresponding channels are avoided. Secondly, when the nRF24Z1 transmits outside the WLAN lobe, the narrow band interferers occupy, at most, an instantaneous bandwidth which is only around 5% of what is available. Therefore, a channel predicted to be OK outside the WLAN lobe is, in most cases, trustworthy.

In table 4, the banning schemes perform better[17] than the prediction schemes. In this case, the predictions of channel quality are less reliable outside the WLAN lobe, because the number of narrow band interferers is larger. Accordingly, it happens more frequently that a channel is measured as OK, but at the time when the frame is transmitted it is actually interfered. Table 5 shows the same trend.

Notice that, in tables 4 and 5, the ban list scheme produces a noticeably larger share of packets with unsatisfactory C/I than the ban ratio scheme. The reason is

[17]Good performance refers mostly to few audible errors, and less to achieved C/I. Errors are generally audible when needed packets are missing, and under re-initialisation of the link.

Parameter	FH scheme				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	38.7	38.1	17.2	18.1	17.3
% of packets present in buffer when needed	93.6	95.9	99.8	99.5	99.3
# re-initialisations	9	8	0	0	1

Table 6: Performance in a static environment with unstable and low interference.

Parameter	FH scheme				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	44.5	50.3	28.0	27.1	32.1
% of packets present in buffer when needed	90.9	89.8	92.7	94.7	92.7
# re-initialisations	16	25	6	8	10

Table 7: Performance in a static environment with unstable and medium interference.

that, when banned channels are released, the ban ratio scheme consistently releases the channel that has proven to do least damage.

From tables 3, 4 and 5 it seems as if the channel list prediction scheme is the superior of the prediction schemes. However, simulation results for the other period lengths (2.9 ms and 4 ms) do not confirm this tendency. These simulations are not included here, but they show that the ranking between the prediction schemes varies with the number of interferers, and conclusions about which of them is better can therefore not automatically be drawn. Nevertheless, a clear tendency can be seen: for all period lengths, it is observed that under stable noise conditions, the banning schemes perform better than the prediction schemes.

4.3.2 Simulations in a static environment - unstable interference

Unstable interference refers to the case when the broadband interferer switches between two different carrier frequencies. In addition, narrow band interferers are included in the simulation. This scenario is of interest because, when a broadband interferer suddenly changes its carrier, the set of suitable channels has a large turn over. The two FH hopping schemes employing ban lists adapt to such changes rather slowly, and therefore one might expect to see reduced performance in comparison with the prediction schemes. Tables 6, 7 and 8 show the obtained simulation results with increasing levels of interference.

Interpretations and comments. Tables 6, 7 and 8 show that when the broad band interferer does not occupy a stable part of the spectrum, the banning schemes perform worse than the prediction schemes.

Note particularly that the ban ratio scheme performs worse than the ban list scheme. The reason is that after the broad band interferer shifts to a different carrier,

Parameter	FH scheme				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	49.3	54.5	34.3	36.4	39.4
% of packets present in buffer when needed	89.2	87.5	89.5	89.2	90.8
# re-initialisations	23	39	19	20	14

Table 8: Performance in a static environment with unstable and high interference.

most of the banning rates represent historical data that are no longer valid, because the carrier shift is large (30MHz). The banning rates will change only gradually and it might take a long time before they reflect the actual noise profile. During that time, the ban ratio scheme will perform poorly. The ban list scheme does not have banning rates to update, and therefore adapts quicker.

One observes that the prediction schemes can not automatically be ranked here either. They perform differently with varying number of interferers. All the above mentioned observations are repeated if one simulates for the two other period lengths. The results are therefore not included here.

It may seem surprising, maybe even disappointing, that none of the prediction schemes stand out as superior across various levels of interference and period lengths. One could for instance expect that the $k = f(p, I)$ scheme should outperform the channel list scheme, because it considers more parameters and therefore is more accurate. The simulation results do not fully meet these expectations, so it is appropriate to reflect over some possible reasons.

Although seemingly attractive, the $k = f(p)$ scheme and the $k = f(p, I)$ scheme have some weak points that should be mentioned. First of all, they both rely on estimates of p. Both schemes obtain their estimate of p by noting how many of the channels deliver readable test packets. Accordingly, the estimate of p is obtained only after all channels have been looped through, and therefore it is consistently based on old events and potentially it is outdated. Secondly, by observing figure 8 on page 30, which applies to the $k = f(p, I)$ scheme, one sees that if TX has a false understanding of how many packets are actually in the receive buffer, RX and TX are likely to have different ideas about how much channel testing should be done. Mismatches in frame structure at RX and TX will be the result, which reduces the performance. With 8 possible values of p and 41 possible values of I, mismatches can be frequent.

The channel list scheme on the other hand does not depend on p or I. It depends only on the number of channels in the channel lists at RX and TX. Consequently, if RX and TX have the same information about channel lists, that information is up to date and true - it is not some estimate which may be false. If RX and TX have different ideas about the channel lists, mismatches happen also here, but they are less frequent. The reason is that the number of channels in the list is maximally 6, for the shortest period, and 4 for the longest period.

4.4 Simulations in a dynamic environment

In order to test how the the FH schemes counteract the effects of fast fading, one needs a simulation scenario with dynamics. That is provided by imposing movement on both people and radios, and when moving, the speed has been set to 1 m/s. However, in a dynamic scenario, all objects are not constantly on the move. For instance, some people are ordering coffee at the bar, while others are seated at tables. Therefore, it is desired to do simulations that correspond to certain time lengths in real life to average out such effects. In our case, that means simulations that go over a significant number of iterations. To illustrate this, one can consider an example. The audio packet length is approximately 0.15 ms at all rates. Accordingly, 10 000 transmitted packets correspond to 1.5 s of real time. In the program, one iteration corresponds to the duration of one packet, and therefore it takes 100 000 iterations to simulate approximately 15 seconds of real time. Preferably, one would want to simulate at least a couple of minutes of real time. That has proven to be problematic, because the program does a substantial amount of calculations per iteration, and therefore such simulations become very time consuming.

In order to handle this problem, two considerations have been made. Firstly, simulations that correspond to several coherence periods, T_c, should generally be sufficient to generate changing multi-paths patterns, and accordingly, should provide the effects of fast fading. Short simulations corresponding to only a few seconds should therefore be sufficient. Secondly, if short simulations are utilized, two different simulations may provide very different transmission conditions. The reason is that the location, and movement of obstacles and reflectors may give very different fading situations in the two cases. Therefore, it might be difficult to conclude whether the superiority of one FH scheme to another is due to a particularly favorable environment, or to the fact that it is more cleverly designed. In order to reduce this problem, all the FH schemes are tested simultaneously in the same simulation. Then, all the FH schemes experience identical, but varying, conditions with respect to dynamics and interference throughout the simulation. Consequently, one should have a better basis for drawing potential conclusions. 20 000 iterations have been used for all the simulations in this section, and therefore one can generally expect a higher number of link re-initialisations here, than in the simulation results from section 4.3.

Also here, constant separation of 10 m is kept between the designated TX-RX pair. Thus, the effects of path loss (section 2.2.1) are equal for all simulations, and the resulting differences in performance can mainly be attributed to how the FH algorithms handle deep fades and interference. Such results are interesting because they are obtained under conditions which try to copy a real-world situation.

4.4.1 Simulations in a dynamic environment - stable interference

These simulations refer to the same case as in section 4.3.1, the only difference being that people and radios are now moving. Tables 9, 10 and 11 show the obtained simulation results for a dynamic environment with increasing levels of interference. The room size is 100 m^2, and the period length is 2.67 ms.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	15.1	8.8	12.0	13.6	10.7
% of packets present in buffer when needed	99.6	99.7	99.9	100	99.2
# re-initialisations	0	0	0	0	0

Table 9: Performance in a dynamic environment with stable and low interference.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	28.5	21.4	24.4	23.1	25.0
% of packets present in buffer when needed	97.3	98.6	96.7	97.1	97.6
# re-initialisations	2	0	6	5	5

Table 10: Performance in a dynamic environment with stable and medium interference.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	42.4	32.0	36.4	37.1	41.5
% of packets present in buffer when needed	92.6	97.3	87.9	88.7	90.2
# re-initialisations	29	4	39	43	37

Table 11: Performance in a dynamic environment with stable and high interference.

Parameter	FH scheme				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	34.8	35.2	17.8	18.8	16.3
% of packets present in buffer when needed	95.6	94.6	99.6	99.4	99.7
# re-initialisations	12	18	2	1	0

Table 12: Performance in a dynamic environment with unstable and low interference.

Interpretations and comments. From tables 9, 10 and 11 one observes, again, that as the number of interferers increases, the performance drops for all schemes.

From tables 3, 4 and 5, in section 4.3.1, it was concluded that the banning schemes outperform the prediction schemes in a static environment with stable noise. Now, the set of unsuited channels can no longer be assumed stable because of the dynamics. Therefore one might expect the banning schemes and prediction schemes to perform more evenly, but the simulation results do not support such a theory. As can be seen, there is still a considerable difference in performance in favor of the banning schemes.

One explanation can be that in the channel model, persons move in discrete steps of 10 cm. That is, they do not move in continuous motion. In fact, for most of the iterations in the simulation, they are actually static. This can be illustrated through an example. If a person has a speed of 1 m/s, and each iteration corresponds to 0.15 ms, it takes approximately 667 iterations before the person moves 10 cm. For the next 666 iterations it remains static in this new position, before it again moves 10 cm. Accordingly, one can argue that this dynamic environment is actually more static than dynamic (!), which of course favors the ban list schemes.

Another explanation is that, such discrete steps remove many of the deep fades that TX and RX would experience if the movements were continuous, and therefore deep fades occur less often in the channel model than in real life.

The observed trend, that banning schemes outperform the prediction schemes in a dynamic environment with stable interference, is confirmed also for the two other period lengths (2.9 ms and 4 ms). Those simulation results are not included here.

4.4.2 Simulations in a dynamic environment - unstable interference

These simulations refer to the same case as in section 4.3.2, the only difference being that people and radios are now moving. Tables 12, 13 and 14 show the obtained simulation results for a dynamic environment with increasing levels of interference.

Interpretations and comments. From tables 12, 13 and 14 one sees the same tendency as in section 4.3.2. Again, the prediction schemes outperform the ban list schemes when the dominant and broad band interferer shifts carrier regularly.

Here, the transmission conditions are also affected by dynamics, which one could have expected to favor the prediction schemes even more. The shown results, and also those for the other period lengths which are not displayed here, neither confirms nor invalidates these expectations. If the channel model could provide more continuous

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	43.6	42.9	27.6	26.4	29.3
% of packets present in buffer when needed	91.2	92.6	94.6	96.2	95.9
# re-initialisations	32	26	15	16	15

Table 13: Performance in a dynamic environment with unstable and medium interference.

Parameter	**FH scheme**				
	Ban list	Ban ratio	Ch.list	$k = f(p)$	$k = f(p, I)$
% of received packets received with $C/I < 12$ dB	429.3	52.2	40.0	37.1	41.1
% of packets present in buffer when needed	88.5	86.3	87.0	88.0	89.3
# re-initialisations	48	56	38	39	29

Table 14: Performance in a dynamic environment with unstable and high interference.

motion, it is likely that the prediction schemes would give results that are even more superior than those in tables 12, 13 and 14.

5 Discussion

This section discusses some of the simplifying assumptions, and it also indicates some potential weak points in the proposed prediction schemes. In addition, some alternative techniques are suggested that might lead to improvements.

5.1 Differentiating between channels in the channel list

When generating the look-up tables in section 3.3.4 and section 3.3.5 one assumption is that a channel search either results in a poor or a good channel. It is assumed that a tested channel found to be poor will distort 100% of the frame, whereas a tested channel found to be good will let 100% through. In other words, a channel tested to be poor, is poor with probability one, and channel tested to be good, is good with probability one. Medium good channels do not exist in this context.

This assumption makes the DP problem more feasible, but in real life it is generally not true. In a practical system, a tested channel found to be suitable for transmission, cannot guarantee 100% delivery. The reason is that the channel might deteriorate after it has been tested but before the transmission of the frame has come to an end. The channel testing that is proposed here cannot foresee such events, because the testing only involves an instantaneous measurement of the channel quality.

Nevertheless, when considering the coherence time, and if there are not too many interferers, it is clear that a channel test will be rather reliable immediately after it has been performed. In the implementation that is proposed though, a channel is generally not used immediately after testing, as described in section 3.3.6. Instead, it is placed on a channel list and remains there for a while before usage. As time passes, the goodness probability of the channel drops. It goes from a probability near one, immediately after testing, to some probability p_g, were $1 - p \leq p_g < 1$, and p is the a priori probability of a channel being poor. When generating the look-up tables in this project, tested channels have been assumed suitable with probability one as long as they are on the channel list. That is generally not correct, but it provides a simplification.

If one wants to generate more realistic look-up tables, the channel list could be taken into account from the beginning. A possible way to proceed, could be to link unique goodness probabilities to each channel in the list. A channel that has just entered the list could be assumed to have goodness probability of one, as before. For the other channels, who have been on the channel list longer, the probability should be less. The oldest channel on the list should be given the lowest goodness probability, but it should always be larger than or equal to $1 - p$.

The exact assignment of goodness probabilities to each channel in the channel list is not investigated further here. Yet, it is appropriate to mention that the possibility of differentiating between tested channels has been identified, and that it could possibly improve the look-up tables.

5.2 The vulnerability of the proposed schemes

As indicated in section 3.3.6, stable communication, between radios employing one of the proposed channel prediction schemes, depends on ACKs arriving at TX. If ACKs are repeatedly unreadable, the communication link gradually becomes vulnerable. The reason is that TX and RX are likely to operate with different frame structures, and in the worst case, disagree on which channels to use for transmission.

The $k = f(p, I)$ scheme, advocated in section 3.3.5, is most vulnerable to loss of ACKs. In that scheme, TX depends on receiving ACKs with updated values of both p and I from RX. If an ACK is not received, TX will simply use its most recent values of p and I, and can therefore not be certain that it tests the correct number, and those channels, that RX expects. From the look-up table in figure 8 one sees that if TX's version of I differs from RX's true I, RX and TX are likely to have different expectations on how many and which channels to test. Because I generally changes after each transmitted frame, loss of ACKs can seriously threaten the performance of the system.

The $k = f(p)$ scheme, presented in section 3.3.4, is less vulnerable to loosing ACKs because TX only needs an estimate of p to determine how many channels to test. That estimate is only updated by RX whenever the radio has looped through all 38 channels, and p is therefore likely to be unchanged from frame to frame. Accordingly, TX will in most cases use the same estimate for p as RX. Confusion about how many channels to test is therefore more rare, but disagreement over which channels to test may nevertheless occur.

In the channel list prediction scheme, in section 3.3.7, RX and TX must have the same number of channels in their respective channel lists in order to test an equal number of channels. If the ACKs do not arrive at TX, the channel lists are likely to differ. Because the number of channels in RX's channel list may change after each transmitted frame, it is important that TX has the same list, and accordingly the reception of ACKs is therefore important.

Because readable ACKs are more or less critical to all the proposed schemes, it could be an option to investigate techniques that secure them better than what is proposed here. A possibility could be to send multiple ACKs after each frame, possibly on different channels. That would, however, have to be done at the expense of reduced audio transmission time, and it would also complicate the protocol. Such techniques have not been investigated further in this report.

5.3 The probability of channels being poor

The look-up tables suggested here, are generated under the assumption that all channels have equal probability of being poor. In a scenario were for instance only Bluetooth interferers produce the noise, such an assumption is not far from being true. The reason is that these interferers employ FH, and they occupy the various channels of the spectrum almost equally often.

However, if a broadband interferer which occupies a stable group of channels is introduced, then those channels are more likely to be unsuitable for usage than the rest, and the probability of channels being poor can no longer be considered as equal across the spectrum. Because such broadband interferers are common in real life, one might argue that the look-up tables are suited, at best, only for a few scenarios.

It might be possible to create more correct look-up tables if one had estimates of p for each channel that the system employs. The DP problem would, however, become much more involved because then the currently used channel would have to be considered as a part of the system's state. Accordingly, the state space would increase dramatically.

Furthermore, estimates for p for each channel would only be possible to obtain during operation. That means that the system would have to be able to solve DP problems on line, which requires significant processing capabilities. Also, with dynamics and changing noise profiles, p for each channel would have to be monitored and updated regularly, and accordingly, the policy of the system would also have to be recomputed regularly. Keeping in mind that Nordic Semiconductor aims at producing devices with low cost and low power consumption, this option seems rather unrealistic at the time being.

5.4 The suitability of the channel model

5.4.1 Discrete versus continuous motion

In sections 4.3 and 4.4, it was seen that the stability of broad band interferers clearly distinguishes between banning schemes and prediction schemes: if the broad band interferer remains on one carrier, the banning schemes are better. Conversely, if the broad band interferer switches between carriers the prediction schemes are preferred.

Dynamics, on the other hand, distinguish the FH schemes less. Because the prediction schemes adapt to changing transmission conditions faster than the banning schemes, one would expect dynamics to favor the prediction schemes more clearly than the simulation results in section 4.4 have shown. The reason why this is not the case, was argued in section 4.4.1. In short, the discrete movement of objects and persons provides the explanation.

Therefore, in order to see the expected distinction clearer, one could reduce the discrete steps, say from 10 cm to 1 cm, to enforce less staccato motion in the model. The problem with such a modification, is that the corresponding mesh grid in the three dimensional matrix representing the scenario, would have to be 10 times finer, possibly, along each axis. That implies a thousand times more entries in the matrix than before.

Prior to many of the calculations that are done for each iteration in a simulation, searches are done through a large share of the matrix' entries. For instance, this happens when the reflection points are determined. Obviously, when the number of entries increase, searching becomes even more time consuming. Therefore, smoother movements do not come without slowing down the simulations even further.

5.4.2 The fading characteristics of the channel model

In section 3.1.2 it was said that the computer model has approximately the same fading characteristics that can be empirically obtained for a real mobile channel. That is true, but only under conditions when TX and RX can transmit via more than a minimum number of reflection paths. The explanation is that in a real indoor setting, the number of reflection paths is generally infinite, and that explains why $r(t)$ in equation 1 on page 8 is a gaussian signal.

The channel model used here, considers only first order reflection points. Such reflection points are, in general, not numerous in a completely rectangular setting. Furthermore, persons are modeled as fully energy absorbing obstacles, something that reduces the number of reflection paths even more. Accordingly, in our setting $r(t)$ cannot be considered gaussian, and the the envelope, $A(t)$, is not Raleigh- or Rice distributed as is often the case in empirical measurements.

In [1], the channel model provided very realistic fading characteristics when people were given reflective properties, in order to increase the number of reflection paths. Here, people do not have such properties, and therefore the fading effects are not entirely realistic. This provides an additional explanation to why the dynamic scenario distinguishes less between banning and prediction schemes than could be expected.

5.5 Techniques that might point out the best scheme

As mentioned in section 4.3.2, the obtained simulation results do not clearly point out a winner among the proposed prediction schemes. A possible way to select a winner, or to confirm that there is none, would be to repeat all simulations a number of times. Afterwards, confidence intervals should be determined for the test results. On that basis, comparison between various schemes would be better founded, and conclusions would be more certain.

Nevertheless, one must keep in mind that, if one proceeds with repeated simulations, all these simulations would also result from the same *simplified* computer model. Therefore, regardless of the number of simulations, one can never be totally confident in such results. Confidence can only be obtained after implementing the proposed algorithms in hardware, and testing them in a controlled environment.

5.6 Sensitivity analysis

The obtained simulation results are sensitive to how some of the variables in the computer model are chosen. Changing the room size or the reflection coefficient, for instance, would certainly lead to different results than those obtained here. Also, certain variables influence the results more than others. As an example of an influencial variable, it was seen in section 4 that the level and stability of the interference greatly affects the results.

How sensitive the computer model is to adjustments of the variables, has not been properly analyzed in this project. Such an analysis might reveal that even small changes to some variables can greatly change the results, while the impact of other variables is limited, even when they are chosen from a large range of values. Obviously, pinpointing the variables that the model is most sensitive to, is of interest because these must be chosen carefully.

5.7 Exploiting the strengths of both banning schemes and prediction schemes

The simulations indicate that, under stable interference, the banning schemes are better than the prediction schemes. On the other hand, if the interference is unstable, the situation is turned around. Therefore it is appropriate to ask whether it is possible to extract some of the benefits from each of the two different techniques, and compose an FH algorithm that exploits both. It has not been investigated here, but the answer is likely to be yes.

Ban lists could possibly be used in combination with channel prediction. One possibility could be to ban channels in the exact same way that the nRF24Z1 does today. In that case, channel testing should mainly be focused on those channels that are not in the ban list. In order to obtain a scheme that is also proactive, test packets should also be transmitted on channels that are banned, every once in a while. Thus, sudden changes in interference could be detected quicker than the ban list schemes are currently able to, and suitable channels could be released straight away.

6 Conclusion

During this project, three different adaptive FH schemes using channel prediction have been developed. The first, determines how many channels to test by using an estimate of the channel ineptness probability as the input of a look-up table. The second, does the same, but in addition to the channel ineptness probability, it also uses the number of packets the receive buffer as input parameter. The third algorithm does not use look-up tables. This prediction scheme only transmits test packets in order to maintain a list of ready-to-use channels. Therefore, it uses no estimates, nor does it consider the load of the buffer.

The performance of these algorithms has been simulated in a computer model, which to a large extent, mimics a real mobile channel. Indications are, that the prediction schemes perform better than schemes which use ban lists, when the interference is subject to large and sudden changes.

Implementing the proposed algorithms in hardware, and simulating in a controlled environment, might verify whether these indications are consistent with real life tendencies. Such simulations might also suggest which of the proposed prediction schemes is the best one to choose. The results from the computer simulations do not give that answer, partly because the computer model only allows discrete motion, in rather infrequent and large steps. To some extent, this type of staccato motion masks the negative impact of fast fading. Real fading, on the other hand, might distinguish clearer between the proposed algorithms, and potentially point out the best one.

In that case, it is possible that the most sophisticated scheme, which considers most parameters, would be the winner. However, it is not certain. One should remember that this scheme is rather vulnerable to loosing ACK packets. Therefore, when operating under much interference, its performance might drop below that of the other schemes. The reason is that the two other schemes are potentially more robust, because they operate according to simpler plans.

From an implementation perspective, the simplest algorithm would be the channel list scheme, which only concentrates on maintaining a channel list which is as full as possible. It does not require look-up tables, or estimation of parameters, and is therefore attractive. On the other hand, these look-up tables do not require much memory, and the processing needed to estimate channel ineptness probability is almost negligible. Therefore, it might be over-hasty to choose the channel list scheme, only because of simplicity - especially if real simulations indicate that one of the two others perform better.

Another likely conclusion from real simulations, would be that the performance of the prediction schemes, relative to the banning schemes, is actually better than the computer results indicate, at least in a dynamic environment.

The computer simulations indicate that under conditions where the noise and interference occupy a rather stable and limited group of channels, the banning schemes are superior to the prediction schemes. The banning schemes adapt according to past events. Thus, they have memory, but no anticipation. The prediction schemes, on the other hand, anticipate because they consistently test before transmitting, but they have no ability to learn from previous events. Simulations have shown that both schemes have their strengths. An FH scheme which extracts the benefits from both, would therefore be attractive. For future works, FH schemes that are able to both memorize and anticipate, should be considered.

References

[1] *nRF-12: Mobile radio channel modelling and testing of adaptive frequency hopping algorithms for 2.4GHz ISM Systems*, John Torjus Flåm, Student project report, NTNU, 2005

[2] *Preliminary product specification: 2.4GHz wireless audio streamer nRF24Z1*, NORDIC SEMICONDUCTOR ASA, November 2005

[3] *Mobile Communications*, lecture notes by A. Klein, Technische Universität Darmstadt, 2004

[4] *Wireless Communications - Principles and Practice, 2nd edition*, by Theodore S. Rappaport, Prentice Hall, 2002

[5] *Field and Wave Electromagnetics, 2nd edition*, by David K. Cheng, Addison Wesley, 1992

[6] *Communication Systems, 4th edition*, by Simon Haykin, Wiley, 2001

[7] *UMTS - The Universal Mobile Telecommunications Systems*, by A. Springer and R. Weigel, Springer Verlag, 2002

[8] *Introduction to Stochastic Dynamic Programming*, by Sheldon Ross, Academic press, 1983

[9] *Reinforcement learning - An introduction*, by R. S. Sutton and G. Barto, The MIT press, 1998

[10] *Radio spectrum measurement of individual microwave ovens*, by P.E Gawthorp, F.H. Sanders, K. B. Nebba and J.J Sell, NTIA report 94-303-2

[11] *Microwave oven interference on Wireless Lan operating in the 2.4GHz ISM band*, by A. Kamerman and N. Erkocevic, The 8th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1997. 'Waves of the Year 2000'. PIMRC '97

[12] *A statistical model for indoor multipath propagation*, by A. A. M. Saleh and R. A. Valenzuela, IEEE journal on selected areas in communications, Vol. SAC-5, NO. 2, February 1997

[13] *Computer Networks, 4th edition*, by Andrew S. Tanenbaum, Prentice Hall, 2003