ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Communication Technology Laboratory
Wireless Communications Group

Prof.Dr.-Ing. A. Wittneben

Design Considerations for Radio Controls in Orienteering

Martin Zoller

June 2010

Advisor: Raphael Rolny, rolny@nari.ee.ethz.ch
Professor: Prof. Dr. Armin Wittneben, wittneben@nari.ee.ethz.ch

Handout: February 22, 2010
Due: June 4, 2010

Abstract

When transmitting data from a mobile device to a base station, the range is limited by the device's transmit power. In a network with several nodes, a multi-hop protocol can be used to forward data over larger distances. This work discusses the possibility of using multi-hop networks for radio controls at orienteering competitions. At such events, data has to be transmitted over distances of up to 10 km. Since the control points are usually located in forests, the transmit signals suffer from strong attenuation. Existing radio controls achieve a sufficient range by using more than 500 mW transmit power - which requires a license - or by utilizing sophisticated infrastructure provided e.g. by cellular networks, which are often not available in remote areas. To this end, a multi-hop protocol is proposed that can reliably forward data from all source nodes to the base station, given that the nodes are placed at most 1 km apart. The protocol is analyzed by means of computer simulations.

Acknowledgements

I would like to thank my advisor, Raphael Rolny, for his valuable input. During our discussions, he contributed many good ideas to improve the communication protocols developed in this work. Furthermore I would like to thank Urs Friedrich from VELPOZ Switzerland for providing details about the existing radio controls in Switzerland. Last but not least, I owe my thanks to Prof. Armin Wittneben for letting me work on this interesting thesis at his laboratory.

Contents

Chapter 1

Introduction

The objective of this work is to define a radio system for data transmission over large distances. The system should achieve a range of up to 10 km in forests. Since the nodes are battery powered and the system should be license free, the transmit power is limited to 500 mW. However, the nodes are distributed such that each node has a neighbor at less than 1 km distance. This allows us to create a multi-hop network. The desired range can then be achieved even if each node can only send data over 1 km. A routing protocol for such a network has been defined. The main application of the system is to transmit split times of runners at orienteering competitions.

1.1 Motivation

Orienteering is a sport that is hardly visible to the public. Races usually take place in forests, with start and finish points somewhere near the border of the forest. There are some exceptions, like sprint races taking place in villages or cities, but even then the observability is low: If a spectator does not know the start time and course of a runner he sees, he cannot tell how fast the runner is. After the race, it is easy to analyze the results: Thanks to the electronic controls which are used nowadays, detailed split times are available as soon as the runners' chips have been read.

But especially during international races, the spectators want to have real-time information about what is happening in the forest. This is where radio controls come into play. They transmit split times to the finish area each time they are punched, and thereby allow the speaker to announce preliminary results. At world championships, such systems can be combined with TV controls, so that spectators can also see the runners passing. Another interesting possibility is to let the runners carry GPS trackers and show their position on a screen.

1.2 Electronic orienteering controls

Nowadays all bigger orienteering events are organized using electronic controls. There are two competing systems for controls: EMIT from Norway [1] and SPORTident from Germany [2]. In Switzerland only the SPORTident system is used; it was introduced in 1998 [3]. The system is based on RFID: Each runner carries a chip that is inserted into a coil at each control. The control station then writes a control number and a timestamp onto the chip. A successful punch is confirmed with a visual and an acoustical feedback. A typical SPORTident station with a chip is shown in Fig. 1.1.

There are special SPORTident stations available to read the chips after the finish. They have an RS232 or USB interface and are usually connected to a PC. However they can also be configured to behave like normal controls and send a small data record over the RS232 interface whenever a punch is registered. Radio

controls forward this signal over the air to the finish area and are connected to a PC there. The data can then be processed by an event software and displayed somewhere, e.g. on a screen on the speaker desk.

Figure 1.1: Chips and control stations of the SPORTident system.

1.3 Existing radio control systems

There are several existing radio control systems. Almost every country where orienteering is practised has its own radio controls. Some of the devices are based on GSM or GPRS, so they require coverage by a cellular network, which is often not available in the competition area. Others use direct radio transmission, usually with the possibility of relaying. However the devices known to us provide only limited relaying functionality, i.e. the relays need to be configured beforehand and are not detected automatically. We know of a 500 mW system used in the UK which is based on commercial radio handsets. It can be operated without a license and has a range of up to 1.5 km [4]. Most of the other systems have more transmit power and need a license each time they are used. For instance, the radio controls used in Bulgaria and Switzerland both send with 5 W transmit power.

1.3.1 Radio controls in Switzerland

The association “Verein für Elektronische Posten und Zeitmessung (VELPOZ)” provides radio controls for Swiss orienteering events. They are based on *MR25* [5] radio modems by the Czech company *RACOM*. The transmit power is 5 W and a license must be acquired each year from the Federal Communications Office to use the controls. As shown in Fig. 1.2, each control consists of a modem with lead-acid battery in a metal case, a separate antenna, and two *SPORTident* control stations. A modem weighs 6 kg including the battery, which makes up about 1.5 kg. The frequency of the system is 425.125 MHz, which lies in the 420-428 MHz Professional Mobile Radio (PMR) band [6]. Some more specifications of the modem are given in Tab. 1.3.1

We have discussed the existing system and possible improvements with Urs Friedrich from VELPOZ [7]. He has tested the line-of-sight range of the system: He placed a control on a mountain at 35 km distance from the reception antenna on the roof of his house. The signal was still received correctly. In forests, however, the achievable range is only about 10 km. In case of rain or in wet conditions, the range is reduced considerably. It can be enhanced by placing a relay station somewhere in the transmission path, e.g. on a hill near the control. However a laptop with special software is needed to configure the modems for relaying. Therefore, and due to the size and weight of the modems, the radio controls are usually placed by car.

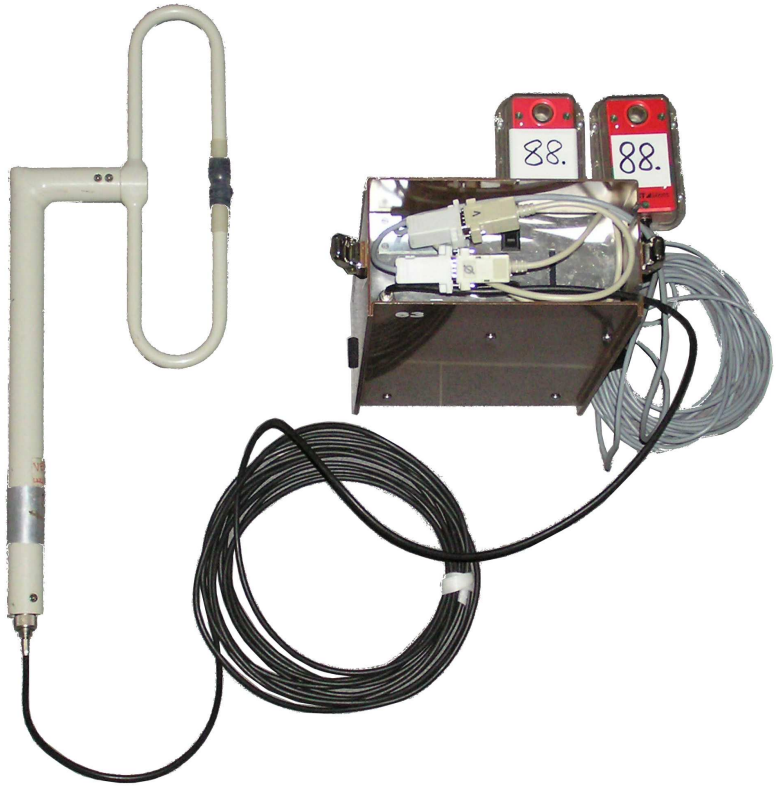

Figure 1.2: Radio controls currently in use in Switzerland

Racom MR25: Selected Technical Specifications

Channel spacing	25 kHz
Supply voltage	13.8 V
Current consumption (receiving)	0.5 A
Current consumption (sending, 5 W)	2.0 A
Receiver sensitivity for BER 10^{-3}	better than -170 dBm
Max. speed on user channels	115 kbps
Typical battery life	8 hours

Table 1.1: Some specifications of the MR25 modem, which is used in the Swiss radio controls.

1.3.2 Problems of existing systems

The existing radio controls work well, but they are not very user-friendly: A specialist is needed to configure them correctly. Sometimes it is necessary to place a relay between a radio control and the base station. Also the devices are much heavier than normal orienteering controls, so they have to be brought near their final locations by car. For some systems a license must be obtained before each use. Single licenses are more expensive than collective ones, so a collective license is applied for every year. This means that event organizers must plan the use of radio controls early enough to avoid extra costs. With today's technology it should be possible to fix these issues: The controls should automatically find a route to the base station and use other controls as relays where necessary. If they could be made small enough to be included with every control, they could be placed together with the normal controls and no additional personnel would be required. Furthermore, it should be possible to operate the controls without a license. If all these requirements can be fulfilled, it will become attractive to use radio controls even at regional orienteering events since the extra cost and effort will be much lower than today.

Chapter 2

Problem specification

In this chapter we will define the conditions in which the system should operate, the requirements it must meet, and the parameters that have to be optimized. One basic requirement is *user-friendliness*: The nodes should be auto-configuring and not require any configuration by the user. It should be possible to turn on the nodes in arbitrary order and to have a functional network as soon as all nodes are working. On the other hand, it is not important that the nodes recognize changes in the network topology, i.e. the nodes are assumed to be turned on only at their final positions. The only changes that may occur are additional nodes (those that are turned on later) and lost connections due to bad link quality. When a packet is lost, it has to be *retransmitted automatically* and repeated collisions must be avoided. A very important requirement is the *reliability*: All generated data packets must arrive at the base station. Even if the network is congested and it takes several seconds until the packet can be forwarded, it may not be discarded. The radio-transmitted split times are not used as official runtimes, i.e. a failure of the radio control system does not lead to the cancellation of the competition, but the information delivered to the speaker should be complete and correct. The *delay* of the data packets is not critical: The spectators will not notice if a split time arrives five seconds late. Also the *data rate* requirements are modest - as shown in the next section - but the *routing overhead* has to be low. If periodical routing updates are sent around by each node, they will easily generate more data than the actual punches.

2.1 Orienteering events

Most orienteering events are individual races with single start. Each competitor is assigned a start time and starts together with some runners of different classes. Between two runners of the same class there is a fixed time interval, usually one to three minutes. At a national competition in Switzerland, there are up to 2000 runners taking part, distributed over a start window of four hours. A typical set of courses will consist of about 50 controls. With the existing radio system, up to five radio controls are possible. A multi-hop network of 20 radio controls with 1 km range will always be sufficient to cover the needs of the speaker. If we assume that each runner punches half of the radio controls, the *punching rate* in the entire network will be 1.39 punches/s. Of each punch, the chip number (3 Bytes) and a timestamp (3 Bytes) have to be transmitted. Thus, only a few bytes of data are generated each second, and the data rate is secondary. It is more important that the *connection time* - the time it takes to start sending data - is short.

Some orienteering races also have mass start, i.e. all runners start at the same time. This is always the case at relay events, for instance. The world's biggest relay is the Jukola in Finland with about 1500 teams every year. Of course a radio control in such a race will generate much more data - especially when a big group of runners passes it - but this is a special case and beyond the scope of this work. Furthermore, one spectator control should be sufficient in such big races since it will generate enough of data for the speaker.

2.2 Multi-hop networks

Here some basic terminology for multi-hop networks is introduced. The following expressions will be used in the protocol descriptions in chapters 4 - 6.

A multi-hop network consists of several *nodes*, also known as terminals. Each node can act as a *sender*, a *receiver*, or both. The number of *sender-receiver pairs* determines the size of a network. A node which generates data packets is called a *source*, while one that only forwards packets is referred to as a *relay*. Multi-hop networks are *peer-to-peer* networks, i.e. there is no central base station that all nodes communicate with. Instead, each node communicates with its *neighbors*, which may forward data to other nodes. The neighbors of each node are the set of nodes which can receive data directly from the node. If the nodes are connected over a wireless channel, each node is assigned a *range*. All receivers which are located within the range of a sender can receive packets from that sender and are thus its neighbors. All data packets sent by a node are received by all its neighbors, except if packets get lost due to *collisions*. Two packets collide if they arrive at the same node at the same time. The networks discussed in this work always contain one special node called the *base station*. This node cannot send data, but it is the destination that all packets are forwarded to. If a source is not directly connected to the base station, its packets are forwarded by relays until they arrive there. The sequence of relays a packet passes on its way to the base station is called *route*. There may be many different routes from a source to the base station. The communication that is necessary to determine which packets should be forwarded over which route is called *routing*. Each transmission from one node to the next is called a *hop*, and the number of hops between the source of a packet and the base station is called the *hop count* of the route. A communication protocol's *routing overhead* is the amount of extra network traffic that is caused by routing.

2.3 Toolchain

All simulations were performed in MATLAB. The Condor High-Throughput Computing infrastructure of ETH Zurich [8] was used for many of the simulations. It allowed us to run simulations on 50 or more computers simultaneously, so that they took only a few minutes instead of an entire day. Since the protocol was written in MATLAB, it will have to be rewritten entirely in order to be run on a microcontroller when the system is implemented in hardware. Still it was not an option to use a programming language like C, as it would have taken much more effort there to create useful output data and plots.

Chapter 3

Radio link

If a new radio system for orienteering controls is to be developed, the first question is which frequency should be used. This depends on technical parameters such as propagation characteristics, but also on the legal regulations. We found solutions based on license-free frequency bands and commercially available radio modems. The physical layer was then mostly neglected in the simulations, since we did not want to make any implementation-specific decisions before implementing the system in hardware.

3.1 Regulations

In Europe there are several frequency bands available for so-called *non-specific short range devices (SRDs)*. Radio modems which work in these bands can be used without a license if they comply to the relevant standards. Devices up to 500 mW *Effective Radiated Power* (ERP) must meet the requirements regarding *"Electromagnetic compatibility and Radio spectrum Matters (ERM)"* described in EN 300 220 [9]. There is a similar standard for higher-power devices called EN 300 113. In Switzerland the Federal Office of Communications (BAKOM) is responsible for the technical regulations. Among others, the following frequency bands can be used for non-specific SRDs in Switzerland:

- **173 MHz:** The 173.0875 to 173.3625 MHz band consists of four channels with up to 25 kHz bandwidth and 500 mW ERP [10]. One of the channels (173.100 MHz) can be used with up to 2.5 W ERP [11].
- **433-434 MHz:** The 433.2375 to 434.5125 MHz band consists of 20 channels with a bandwidth of up to 25 kHz each [12]. Again the power is limited to 500 mW, with the exception of eight subchannels where up to 2.5 W are permitted [13]. This is an *Industrial, Scientific and Medical* band (ISM), i.e. interference is possible in urban areas, but we do not expect any problems in practice.
- **869 MHz:** The 869.400 to 869.650 MHz offers 10 channels with up to 25 kHz bandwidth each. band is intended for "non-specific SRDs" as well. Channel spacing must be 25 kHz except if the entire band is used as one channel. Devices must either comply to Draft EN 300 220-1 V2.4.1 (which currently available radio modems do) or have a duty cycle of less than 10%. Up to 500 mW ERP are possible [14].

We only considered frequencies with at least 500 mW maximum ERP since the range would be too limited with less power.

3.2 Commercial radio modems

To see what kind of products might be suitable for our application, we compared the specifications of several radio modems and transceivers. Each of the following examples is outstanding with respect to a specific

parameter. Be aware that the range specified in the devices' data sheets is valid only for line-of-sight connections in ideal conditions; the performance in a forest environment is much worse and is estimated in Sec. 3.6.

- **Amber Wireless AMB8355** [15]
  This is a modem on an embedded board. It is based on the AMB8315 transceiver with a line-of-sight range up to **20 km** at 869 MHz (license free). We did not find any other device with such a high range. The board has a receiver sensitivity of -110 dBm and uses 500 mW transmit power. It is 97×38×19 mm large. At 7 V supply voltage it consumes 530 mA in transmission mode, i.e. it dissipates 3.7 W. A 434 MHz version of the device is also available.

- **Telit PowerOne Terminal** [16]
  This modem uses the 869 MHz band and has a receiver sensitivity of **-115 dBm**, which is the best value we have encountered. It is 159×85×35 mm large in an IP65 casing. At a supply voltage of 12 V it dissipates 2.5 W during transmission. The range is up to 16 km.

- **Telit TinyOne Pro 868 MHz** [17]
  The TinyOne Pro 868 MHz RF module is a surface-mounted device and is **38×21 mm** large. It is by far the smallest 500 mW device we have found. The range with an external antenna is specified as 4 km.

- **Atim ARM-C8** [18]
  The ARM-C8 embedded board is 64×32×4 mm large. It is available for the 869 MHz and 433 MHz frequency bands. According to the datasheet it has a sensitivity of -105 dBm and achieves more than 5 km range. Furthermore it has the smallest power dissipation of all devices we considered (**1.75 W** in TX mode).

- **Radiometrix BiM1H** [19]
  This is just a transceiver, but it is one of the few devices we found which work in the **173 MHz** band. It supports frequencies between 120 and 180 MHy. The module is 33×23×12 mm large and has a transmit power of 500 mW. Its sensitivity is -120 dBm at the operating frequency of 151.3 MHz.

All the parameters compared above are important for our application: The modem should be compact, dissipate a small amount of power and achieve a high range. However these requirements are contradictory: Even though all presented examples have the same transmit power, the range and power dissipation varies considerably. The smallest device is not the most economic one, and although it has an external antenna, its range is only about one fourth of what larger devices achieve. Therefore a good trade-off between the requirements has to be found.

The main goal of our simulations was to determine whether a license-free system would achieve a sufficient range at all. Therefore we used the sensitivity value of the *Telit PowerOne Terminal*, which is the modem with the highest sensitivity among our examples. Since the implementation of the system is not the main topic of this thesis, we did not do an extensive search for 173 MHz and 433 MHz radio modems. Transceivers for the 173 MHz band and modems for the 433 MHz band are available, and their specifications are comparable to those of the 869 MHz radio modems. We therefore assumed that a modem with -115 dBm sensitivity is feasible in all three frequency bands. This information about the physical layer is sufficient to perform simulations. Since the radio modems detect erroneous packets automatically and errors occur with a probability of less than 10^{-3} if the received signal is strong enough, transmission errors do not have to be considered in detail.

3.3 Range and power requirements

As shown in Sec. 3.1, there are three possible frequency bands for a license-free radio control system. The transmit power may be up to 2.5 W in some channels. However, the devices should be as small as possible,

so the size of the battery has to be minimized. Therefore we were interested whether 500 mW transmit power would be sufficient. This is only 10% of the power that the existing radio controls use, so the range will be greatly reduced. It is therefore necessary to build a *multi-hop network* where data packets are forwarded automatically from each node to the base station. The limiting parameter is then the maximum distance between a node and its closest neighbors. In orienteering races, no control will usually be more than 1 km away from the others, so a range of 1 km would be sufficient. There may be special cases where additional relay nodes will be required, e.g. long-distance world championships, but the important thing is that the system works without special precautions in a normal race.

3.4 Attenuation in forests

In Sec. 3.2 we found the receiver sensitivity of a radio modem to be up to -115 dBm. The transmit power has been set to 500 mW, which corresponds to 27 dBm. This means that for the signal to be received correctly, it may not be attenuated by more than 142 dB. We compared this requirement to attenuation values found in the literature.

The attenuation of radio signals in forests is discussed in several papers. The work by T. Tamir [20] deals with frequencies of 1-100 MHz and introduces a model with two waves: A lateral wave which propagates along the tree tops, and a sky wave which is reflected at the ionosphere. Another paper by the same author [21] treats the case where only some parts of the communication path lie in a forest. In a more recent paper [22], a model based on electrodynamics theory is presented where two propagation mechanisms are present. At short distances, the direct wave is dominant and the attenuation increases exponentially with distance. At more than 100 m distance, the propagation characteristics are similar to those along the earth surface. The two mechanisms are also discussed in [23], where propagation of different frequencies over a distance of 112 m is studied. Attenuation rates at different frequencies are derived from the results.

For our simulations, we needed an estimate of attenuation at 1 km distance as a function of frequency. Experimental data on this are given in [20]. According to Fig. 9 in the paper, the attenuation is 120 dB at 173 MHz and 140 dB at 400 MHz. At 433 or 869 MHz the signal would be attenuated even more, so the 173 MHz band is clearly the best choice for a radio control system. Since the attenuation is less than our upper bound of 142 dB, we can conclude that a 173 MHz system will have more than 1 km range. However it is still to be shown that connections will be reliable enough; this is discussed in Sec. 3.6. To model the attenuation as a function of distance, the following formula for the *basic path loss* L_{b0} is given in [20]:

$$L_{b0} = 1570 \left[|n^2 - 1| \operatorname{Re}(n)\right]^2 \left(\frac{\rho}{\lambda_0}\right)^4$$

Here n denotes the refractive index of the forest medium, ρ is the distance between sender and receiver, and λ_0 is the wavelength. The formula is based on the assumption that both the transmitting and the receiving antenna are close to the tree tops. For our simulations, we further simplified the path loss model: We set the frequency to 173 MHz, so that the path loss is 120 dB at a distance of 1 km, and assumed all parameters except the distance to be constant. The formula then becomes

$$L_{b0} = \rho^4$$

since $1000^4 = 10^{12} = 120$ dB, i.e. the product of the constants is exactly 1. The simplified model is explained below.

3.5 Simple path loss model

We used a simple path loss model to predict the range and reliability of the radio link. The model was also implemented in some of the simulations. Its purpose is to obtain an outage probability as a function

of transmit power and distance between sender and receiver. It can only give a coarse approximation of the actual connection reliabilities since most physical layer issues are neglected. Each channel is modeled with a channel coefficient h_i and a noise coefficient n. When a symbol x_i is transmitted over a channel i, the arriving symbol y_i is given by

$$y_i = x_i \cdot h_i + n$$

where n is Additive White Gaussian noise ($n \sim \mathcal{N}(0, \sigma_n^2)$) and the symbols are normalized ($E\left[|x_i|^2\right] = P_{TX} \ \forall i$).

Attenuation

The path loss factor depends on the distance ρ and the terrain through which the radio signal ist transmitted. For propagation through forests at 173 MHz it can be approximated by

$$L_{b0} \approx \rho^4 \quad \text{(see Sec. 3.4)}$$

and the channel coefficients are normally distributed:

$$h_i \sim \mathcal{N}\left(0, \frac{1}{L_{b0}}\right)$$

The received power of the desired signal can now be calculated as follows:

$$P_{\text{desired},i} \ = \ P_{TX,i} \cdot |h_i|^2$$

If a standard Gaussian random variable $X \sim \mathcal{N}(0,1)$ is introduced, we get

$$P_{\text{desired},i} \ = \ P_{TX,i} \cdot \frac{1}{L_{b0}} \cdot X^2$$

Now X^2 can be replaced by a new random variable Y, which will then have a χ^2 distribution with one degree of freedom:

$$P_{\text{desired},i} \ = \ P_{TX,i} \cdot \frac{1}{L_{b0}} \cdot Y$$

This distribution was used in the simulation to determine the received power. If there is no interference, the power must be bigger than the receiver sensitivity S for a successful transmission. The probability of an outage is thus given by

$$\mathrm{P}_{\text{outage}} \ = \ \mathrm{P}\left[P_{\text{desired},i} < S\right] \ = \ f\left(\frac{S \cdot L_{b0}}{P_{TX,i}}; 1\right)$$

where $f(x;k)$ is the *cumulative distribution function* of the χ^2 distribution with k degrees of freedom. As stated in Sec. 3.2, we found radio modems with up to -115 dBm sensitivity, so we assumed $S = -115$ dBm. The transmit power was set to $P_{TX} = 500$ mW.

Signal to Interference-and-Noise Ratio (SINR)

If there is just one node transmitting, the criterion stated above is sufficient for successful transmission of a packet, i.e. it depends only on the random channel coefficient whether the packet will arrive. However, in a wireless network with many nodes there will often be interfering packets. Successful transmission may be possible despite a collision, though, if one signal is much stronger than the others. In the simulation we therefore calculated the SINR of each arriving packet to decide whether it should be dropped or not:

$$\text{SINR} = \frac{P_{\text{desired}}}{P_{\text{interference}} + \sigma_n^2}$$

This value was then compared to the minimum *signal-to-noise ratio (SNR)*:

$$\text{SNR}_{min} = \frac{S}{\sigma_n^2}$$

However, the SINR calculation is only valid if the interference is AWGN. Therefore our simulation can only provide a rough approximation of the actual reliability of connections.

Thermal noise power

The **noise voltage** is given by [24]

$$V_n = \sqrt{4k_B \cdot T \cdot R \cdot \Delta f} \qquad k_B = 1.38 \cdot 10^{-23} \frac{J}{K},\ T = 293.15\,K$$

The **thermal noise power** can be calculated from the noise voltage and does not depend on the noise resistance:

$$P_{n,th} = \frac{V_n^2}{R} = 4k_B T \cdot \Delta f \qquad (\Delta f:\ \text{Bandwidth})$$

To obtain the total noise power, $P_{n,th}$ has to be multiplied with the **noise figure**. We used 10 dB as an upper bound for this, i.e. the noise power was assumed to be $\sigma_n^2 = P_{n,th} + 10\,\text{dB}$.

3.6 Range at 173 MHz

We have seen in Sec. 3.4 that the expected attenuation when sending a 173 MHz signal through a forest is 120 dB at 1 km distance from the sender. However the channel coefficients are randomly distributed, so we need to determine the probability of an outage. Since the data packets are very small, we assume that the channel coefficient is determined once for every packet. With the formula derived from the simple path loss model, we get $\text{P}_{\text{outage}} = 0.063$. This is acceptable; 6% of the packets will get lost. At 1.26 km distance the outage probability is 0.1, and at 2 km it is already 0.25. A plot of P_{outage} against the distance is given in Fig. 3.1.

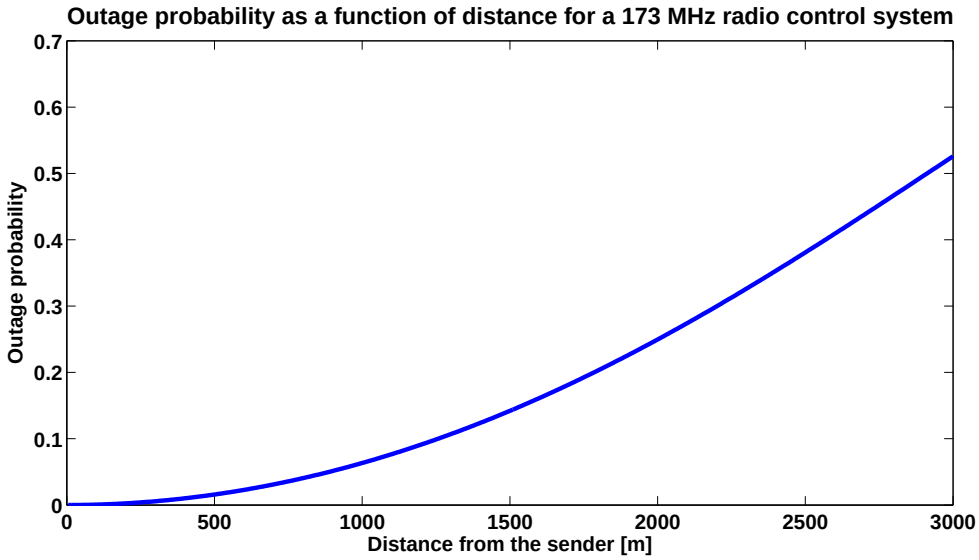

Figure 3.1: Outage probability as a function of distance, as derived from the simple path loss model for a 173 MHz signal with 500 mW transmit power.

Chapter 4

Simple circular network protocol

4.1 Protocol description

For the first simulations we chose a simple network topology: A circle of nodes. This would be a realistic scenario if the radio controls were used just for one course. In practice each orienteering race consists of many different courses, so it is optimistic to assume a circular network. See Fig. 4.1 for an example of a circle topology.

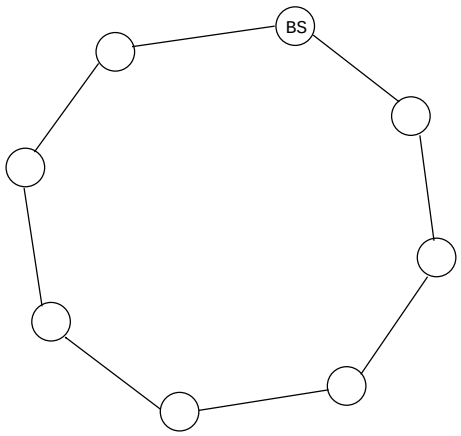

Figure 4.1: Example of a circular network topology.

We made the following assumptions for the modeled network:

- Time is discrete and each second is divided into 100 time units. A time unit thus corresponds to 10 ms, which is enough to make a connection and send a few bytes of data.
- All data packets are small enough to be transmitted within one time unit. If the data rate is 10 kbps, this means that a packet can be up to 13 Bytes large. As derived in Sec. 2.1, the payload of a data packet is just six Bytes, so seven Bytes remain for packet ID and other metadata. Thus the assumption is realistic.
- Each node has a unique ID and knows the IDs of its two neighbors. However the nodes are not aware of their distance from the base station.

- Whenever a node receives packets from both its neighbors in the same time unit, there is a collision and the packets get lost. Otherwise all packets arrive at the neighbors of their sender.

Within this network, we tested a simple protocol. It is based on the fact that each node generates a very small amount of data with a low duty cycle. This should allow us to broadcast all generated data through the entire network until, at some point, it arrives at the base station. An acknowledgement can then be sent back in the same way.

The behavior of the *sender* is given in Alg. 1. In each time unit, a new *message packet* (MSG) is added to the send queue with a small probability P_{punch}. If the node is idle and there are packets in the send queue, the next packet is sent and the timeout counter is set to its initial value. If the node is already waiting for an *acknowledgement* (ACK), it decrements the counter and enters random backoff if the counter reaches zero. After waiting for a random number of time units, the packet is retransmitted. If the send queue is empty and the node is idle, it checks for packets in the forwarding queue. They are sent and removed from the queue without getting acknowledged. ACK packets are sent only by the base station. They are fowarded through the network like normal data packets. Nodes can identify the ACKs of their messages using the message ID and source ID that is included with every packet.

Algorithm 1 Sender behavior in the Simple Circular Network Protocol.

```
if binRandom(P_punch) then
  addToSendQueue;
end if
if waitForACK > 0 then
  waitForAck ← waitForAck-1 ;
  if waitForAck = 0 then
    startRandomBackoff;
  end if
else if randomBackoffCnt > 0 then
  randomBackoffCnt ← randomBackoffCnt-1;
  if randomBackoffCnt=0 then
    retransmitMessage;
    waitForACK ← maxWaittime;
  end if
else if sendQueueSize > 0 then
  sendNextMessage;
  waitForACK ← maxWaittime;
else if forwardQueueSize > 0 then
  forwardNextMessage;
end if
```

The *receiver* processes all arriving packets which have not been dropped due to collisions. Its behavior is shown in Alg. 2. If a node receives an ACK, it checks whether the source ID matches its ID and the message ID is the ID of the last packet that the node sent out. In this case, the packet is marked as acknowledged. Otherwise, the ACK is searched in the node's message library. If it is not found there, it is added to the forwarding queue and to the library. This mechanism prevents repeated forwarding if a node receives a packet several times. A special case is that the ACK to a message is received while this message is still in the node's forwarding queue. If this situation is detected, the message is removed from the queue and an ACK is added instead. MSG packets are treated similarly: If a packet is not present in the message library, it is added both to the library and to the forwarding queue. The message queue is realized as a FIFO buffer, i.e. the oldest entry is overwritten when a new one is added.

Algorithm 2 Receiver behavior in the Simple Circular Network Protocol.

```
if received = ACK then
  if sourceID = myID and messageID = lastSentID then
    waitForACK ← 0;
  else if not inMessageLibrary(sourceID,messageID,ACK) then
    if inForwardQueue(sourceID,messageID) then
      removeFromForwardQueue(sourceID,messageID);
    end if
    addToForwardQueue(sourceID,messageID,ACK);
  end if
else if received = MSG then
  if not inMessageLibrary(sourceID,messageID) then
    addToForwardQueue(sourceID,messageID);
    addToMessageLibrary(sourceID,messageID);
  end if
end if
```

The *base station*'s behavior is quite simple, as shown in Alg. 3. If a MSG packet arrives, it is searched in the message library. Unknown messages are confirmed with an ACK and added to the message library. A real base station would also send out these messages over its serial port.

Algorithm 3 Base station behavior in the Simple Circular Network Protocol.

```
if received = MSG then
  if not inMessageLibrary(sourceID,messageID) then
    addToACKqueue(sourceID,messageID);
    addToMessageLibrary(sourceID,messageID);
  end if
end if
if ACKqueue then
  sendACK;
end if
```

4.2 Simulation results

For all simulations with this model, we made the assumptions derived in Sec. 2. Based on that, we determined appropriate values for the variable parameters, starting with the *timeout*. There the question was whether it would be sufficient to wait for the reply just from one direction, or whether the second and often much longer route would contribute significantly to the success probability. First simulations showed that it was optimal to wait for a reply over both routes, so the timeout was set to $2 \cdot (\#nodes - 1)$. A simulation with more repeats then showed that this is wrong and a timeout equal to the number of nodes is sufficient. The results of this simulation are shown in Fig. 4.2.

We also tried to determine an optimal value for the *duration of random backoff*. Nodes affected by a timeout will wait for a number of time units between 0 and this value before retransmitting their message. In Fig. 4.3 the number of timeouts is plotted against the backoff duration. The plot shows that the value should be at least 50 to avoid repeated timeouts due to collisions. For higher values the number of timeouts does not decrease anymore, so it does not make sense to choose a value larger than 50.

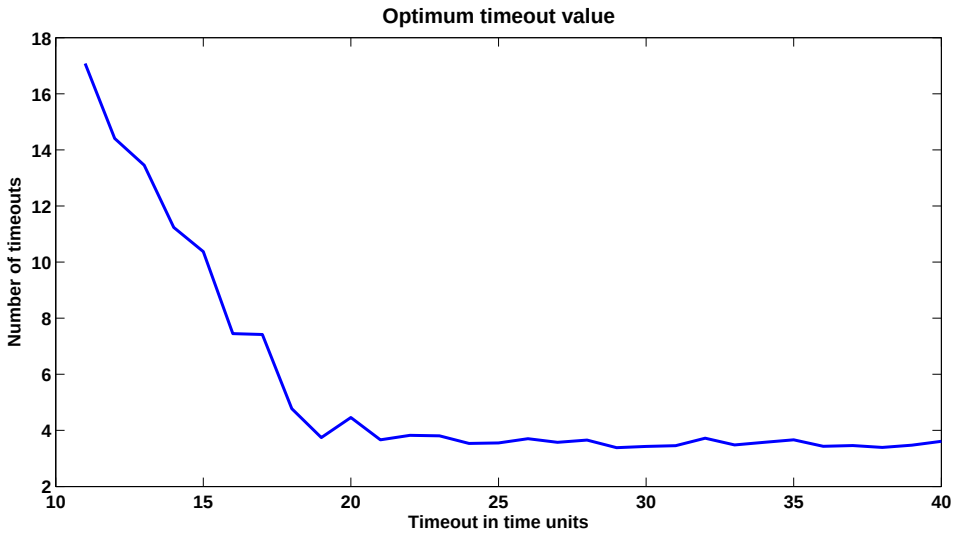

Figure 4.2: Simulation with 19 nodes and different timeout values. A timeout of 19 is sufficient in this case.

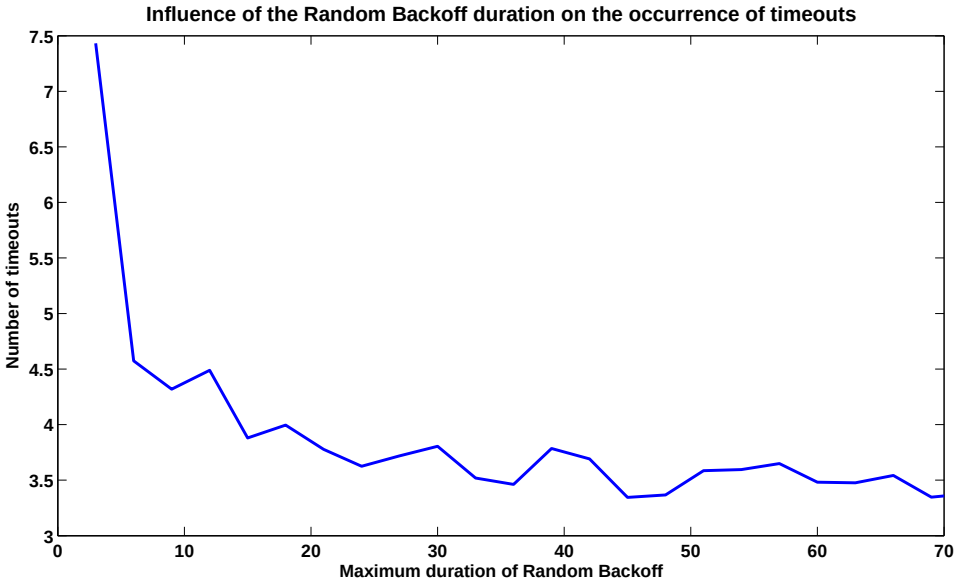

Figure 4.3: Number of timeouts with increasing duration of random backoff.

The *size of the message library* was fixed to 50 entries. The retransmission of a message will never be delayed so much that the receiver processes 50 messages between the first and the second arrival of the message, so 50 is certainly sufficient. It is not necessary to make the value smaller either, because 50 messages require only about 500 Bytes of memory. The *lengths of the message queues* at each node were assumed to be unlimited. They would only be important in situations of extreme traffic. Again, the memory requirements are very modest - even if the sending and forwarding queue each have 1000 entries, they will occupy no more than 20 kB.

The most interesting part of the simulations was to see how much traffic the system could deal with. A plot with different punching rates is given in Fig. 4.4. As stated in Sec. 2, the worst-case punching rate is 0.0007; a circular network with 19 nodes will work at up to twice this value. We also tried to add more nodes with a constant punching rate per node; this leads to more traffic, too, but additionally the routes get longer. For this simulation the timeout was scaled with the network diameter. The result is shown in Fig. 4.5: If the network contains more than 25 nodes, the collision probability gets too large and messages start queuing up. With more than 33 nodes, the congestion gets so bad that the number of messages which get transmitted and acknowledged decreases again.

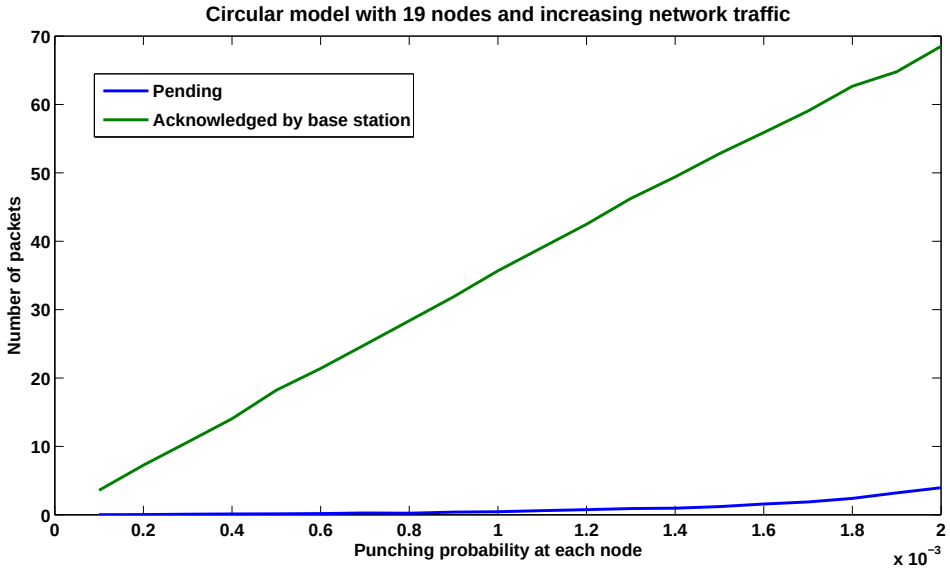

Figure 4.4: Number of acknowledged and stalled messages for increasing punching rate.

4.3 Interpretation

One problem of the system is that it does not work with an even number of nodes. If the node count is even, the base station has a direct opposite, and all packets coming from this node will collide at the base station since the two routes have the same length. We made all simulations with an odd number of nodes to avoid this problem. It could be dealt with by randomly delaying messages by one time unit, or by allowing traffic only in one direction.

As shown in Fig. 4.4, the network can take about twice as much traffic as the assumed worst-case scenario. If the punching probability per node is left constant, the number of nodes can be increased to 25 before messages start queuing up. This might seem like a good safety margin, but if we apply a broadcasting protocol to a more realistic network topology, it will not be sufficient. In the circular network, each message

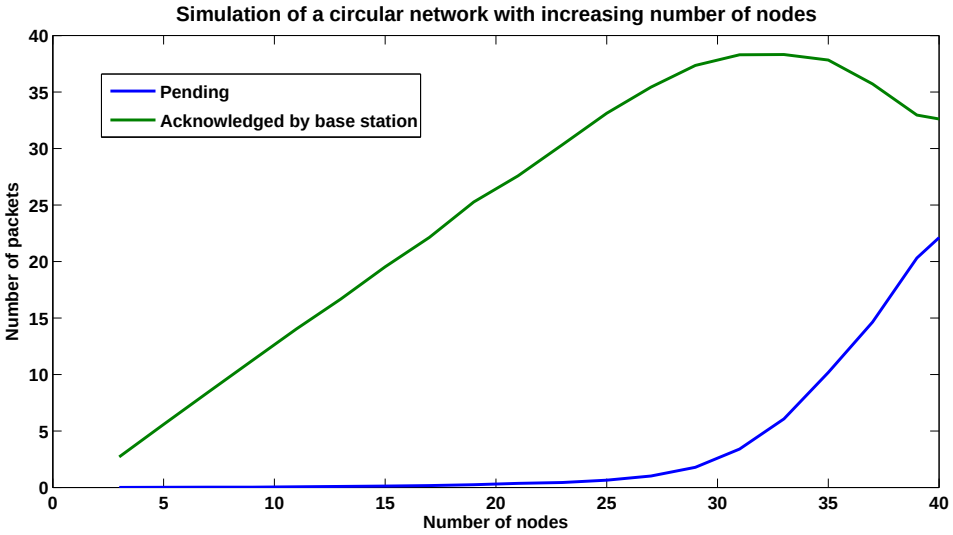

Figure 4.5: Simulation with increasing number of nodes.

is sent only over two paths, and it will only get lost if another message pops up on the way. But in a grid, for instance, most nodes have more than two neighbors, so the packets are forwarded over many paths and are likely to collide with copies of themselves. If all copies get lost, the packet has to be sent through the entire network again. The possibility of packet loss due to bad link quality further reduces reliability. Therefore we decided to develop a routing-based protocol that will allow to acknowledge transmissions from one node to the next.

Chapter 5

Routing in a random grid network

5.1 Protocol description

For this second protocol we made new assumptions about the network topology. The simulations are based on a "random grid topology", i.e. a grid of fixed size where only part of the fields are occupied by nodes. The number of nodes is also fixed, but they are randomly distributed. An example of a random grid topology is shown in Fig. 5.1. We further made the following assumptions:

- Time is discrete and each second is divided into 100 time units.
- All data packets are small enough to be transmitted within one time unit.
- The grid spacing is 400 m and each node has a range of 1 km. This means that each node has up to 20 neighbors. Fig. 5.2 illustrates this.
- Simulations are only performed on topologies whose network graph is *connected*, i.e. all nodes have at least one route to the base station.
- Each node has a unique ID. The nodes are initially not aware of their neighborhood.
- Whenever a node receives packets from several neighbors in the same time unit, there is a collision and the packets get lost. Otherwise all packets arrive at the neighbors of their sender.
- The node with ID 1 is assumed to be the base station. It transmits only ACK and RAK packets. Its position is randomly determined.

Challenges of Routing

The special requirement for our routing algorithm was that it should do without sending around routing updates at regular intervals. Such updates might otherwise cause more traffic than the payload data. The algorithm presented here is an adapted version of the *Distance Vector* algorithm, which is common in computer networks [25]. Since all payload data need to be forwarded to the base station, it is sufficient if every node knows the hop count and next hop of a shortest path to the base station. To keep this information up-to-date, the protocol takes advantage of the fact that packets sent over a wireless network are usually received by several nodes. This allows the nodes to verify the routing information in data packets even if the packets are intended for other nodes. If a node knows a shorter route than the one that is used by a packet, it sends a routing update to the packet's sender. This ensures correct operation with a minimal overhead. Routing query packets are only needed in the beginning and in case a node has lost all its connections. The protocol consists of four types of messages:

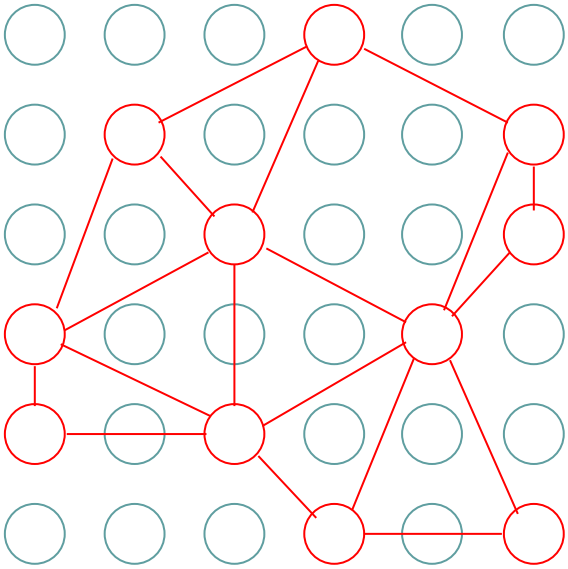

Figure 5.1: Example of a random grid topology.

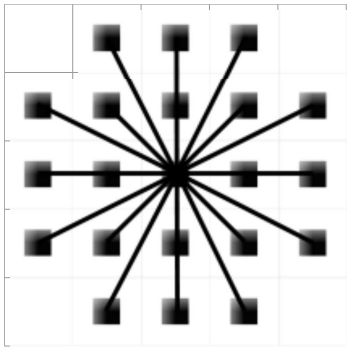

Figure 5.2: Range of a node within the random grid network.

- **MSG**: Contains a punching record and is forwarded over a route.
- **ACK**: Used to acknowledge transmission of a MSG over one hop.
- **NAK**: Sent in reply to a MSG if the receiver does not know any route to the base station.
- **QRY**: Nodes which are not yet connected or have lost their connections use this packet to obtain routing information. All receivers send a RAK in reply.
- **RAK**: Routing Acknowledgement. Sent in reply to QRY, but also to MSG if outdated routing information is detected.

Each node assigns a message ID to each MSG packet it transmits. A MSG is uniquely identified by the ID of its source and the message ID. Additionally each MSG contains a sender ID, receiver ID, and remaining hop count. These fields are assigned by the sender each time the packet is forwarded. ACK and RAK packets contain the same fields, but their source and message IDs are copied from the packet that caused them to be sent. The remaining hop count sent with ACK and RAK is used by the receiver to update the routing information. NAK packets do not contain a remaining hop count since they are only sent by nodes that do not know any valid routes. QRY packets only consist of a source ID and message ID because they address all nodes that receive them.

Header data

Our routing protocol needs more metadata to be included in every packet than a simple broadcasting protocol. It uses the following data fields:

- **Message ID** (2 Bytes) — A number assigned by the source of each message. It is unique only if combined with a node ID.
- **Source ID** (1 Byte) — ID of the node which created the packet.
- **Sender ID** (1 Byte) — ID of the node which sent the current copy of the packet.
- **Receiver ID** (1 Byte) — ID of the node to which the packet is being forwarded.
- **Remaining hop count** (0.5 Bytes) — Number of hops left on the packet's route to the base station. This value is decremented every time the packet is forwarded. It is present in each packet to allow neighbors of the sender to verify the routing information.
- **Packet type** (0.5 Bytes) — May be either MSG, ACK, RAK, NAK or QRY.

The header data thus occupy six Bytes, so a MSG packet with six payload Bytes has a total size of 12 Bytes. This can still be transmitted in less than 10 ms at a data rate of 10 kbps. Some of the other packets need less metadata, e.g. QRY packets have no receiver field.

Algorithms

Alg. 4 describes the behavior of the sender. If the node's routing database is empty, any ACKs in the queue are turned into NAKs and the first of them is sent. If the ACK queue is empty, a QRY packet is sent with the probability P_{QRY} to find new routes to the base station. ACK and RAK packets have first priority if the routing database contains some entries. They are sent immediately, if available. If the node is waiting for an ACK, a similar procedure as in the broadcasting protocol is used: After a timeout the node waits for a random number of time units (random backoff) before it retransmits the packet. However after a certain number of retransmissions the route is deleted and a different route is used for the next attempt. If there is no unacknowledged MSG packet and there are more messages in the queue, the next MSG is sent. As in the circular model, new messages are added randomly to each node's queue.

Algorithm 4 Sender behavior in the Reliable Random Grid Model.

```
if routeLibSize = 0 then
  if ACKqueueSize > 0 then
    sendNAK;
  else if binRandom(P_QRY) then
    sendQRY;
  end if
else if ACKqueueSize > 0 or RAKqueueSize > 0 then
  sendACKRAK;
else if waitForACK > 0 then
  waitForACK ← waitForACK - 1;
  if waitForAck = 0 then
    if retransCnt > retransMax then
      deleteRoute(0);
      retransCnt ← 0;
    else
      startRandomBackoff;
      retransCnt ← retransCnt+1;
    end if
  end if
else if randomBackoffCnt > 0 then
  randomBackoffCnt ← randomBackoffCnt-1;
  if randomBackoffCnt=0 then
    retransmitMessage;
    waitForACK ← maxWaittime;
  end if
else if MSGqueueSize > 0 then
  sendMSG(routeNextHops[0]);
  waitForAck ← maxWaittime;
end if
```

The receiver behaves as shown in Alg. 5. If a QRY packet arrives, a RAK is enqueued in reply. It is addressed to the sender of the QRY and contains its message ID as well as the length of the shortest route known to the replying node. ACK and NAK packets are only considered if their receiver field contains the ID of the current node, the source ID is that of the last MSG packet sent by the node, and the message ID is the ID of the last sent packet. If an ACK fulfills all these conditions, the current packet is removed from the send queue. When receiving a RAK for the current packet, the node will delete the route that was in use, so that the packet will be retransmitted over a different route. RAK packets addressed to the current node are used to update the routing database. If they contain a route that is not known to the node yet, it is added to the database. Otherwise the hop count of the route is updated as necessary. Normal ACK packets also contain a hop count, which is always checked against the count in the database when receiving an ACK. If a MSG packet arrives, its receiver field is checked first. Messages addressed to the current node are always confirmed with an ACK that contains the source ID, sender ID and message ID of the MSG packet. If the packet is not found in the message library of the node, it is added to the library as well as to the forwarding queue. Last but not least, the remaining hop count of all received MSG packets is compared to the shortest route in the routing database. If the node knows a shorter route than the one used in the packet, it enqueues a RAK packet to update the information of the packet's sender.

Algorithm 5 Receiver behavior in the Reliable Random Grid Model.

```
if received = QRY then
  if routeLibSize > 0 then
    addToACKqueue(RAK,senderID,messageID);
  end if
else if received = ACK then
  if toID = myID and sourceID=lastSentSourceID and messageID = lastSentID then
    waitForACK ← 0;
    retransCnt ← 0;
  end if
else if received = NAK then
  if toID = myID and sourceID=lastSentSourceID and messageID = lastSentID then
    deleteRoute(0);
    waitForACK ← 1;
    retransCnt ← −1;
  end if
else if received = RAK then
  if toID = myID then
    addRouteToLib(senderID,remainingHops);
  end if
else if received = MSG then
  if toID = myID then
    enqueueACK(sourceID,senderID,messageID);
    if not inMessageLibrary(sourceID,senderID,messageID) then
      enqueueMSG(sourceID,messageID);
      addToMessageLibrary(sourceID,senderID,messageID);
    end if
  else if remainingHops-1 ≥ destHopCount then
    enqueueRAK(sourceID,senderID,messageID);
  end if
end if
```

Also in this protocol, the base station's behavior is much simpler than that of the other nodes. It is shown in Alg. 6. When a QRY packet is received, the base station enqueues a RAK packet in reply. In case of

MSG packets, all packets are looked up in the message library. Unknown packets are added to the library and forwarded to the base station's serial port, even if they are not addressed to the base station. This ensures that packets arrive as quickly as possible. MSG Packets addressed to the base station are always answered with an ACK packet. This is also necessary if a message is known already, since the sender would otherwise retransmit it several times. Without multipath routing, it is not very likely that a packet arrives at the base station several times, but it happens if an ACK packet is lost. In this case a packet is retransmitted even though it has been received already.

Algorithm 6 Base station behavior in the Reliable Random Grid Model.

```
if received = QRY then
  addToACKqueue(RAK,senderID,messageID);
else if received = MSG then
  if not inMessageLibrary(sourceID,messageID) then
    addToMessageLibrary(sourceID,messageID);
    output(messagePayload);
  end if
  if toID = myID then
    enqueueACK(sourceID,senderID,messageID);
  else
    enqueueRAK(sourceID,senderID,messageID);
  end if
end if
if ACKqueueSize > 0 or RAKqueueSize > 0 then
  sendACKRAK;
end if
```

Design considerations

While developing this routing protocol, we evaluated several design options. They were discarded in an effort to optimize the performance of the routing protocol.

- **Multi-Path routing:** It is possible to forward packets over all known routes, or over all shortest routes, instead of just one route. A MSG packet can then have multiple receivers and must be acknowledged by all of them. At nodes where several routes converge, the first arriving copy of each packet is forwarded, while all others are automatically discarded. The base station behaves the same way, it processes only the first copy of each packet. The advantage of the multi-path approach compared to using just one route is that packets arrive quickly even if one of the copies gets lost. However, since each packet must be acknowledged by all receivers, much more traffic is generated. The network cannot handle the same amount of input data as with the single-route protocol. Due to the automatic retransmission of lost packets, reliability is guaranteed without sending packets over several routes. Therefore multi-path routing does not make sense in this protocol.

- **Best routes only:** Originally it was planned that nodes should store only the shortest routes to the base station. In this case, a node discards its entire routing database when it gets informed about a shorter route than the ones in its database. It stores several routes of equal length, if available. When transmission over a route fails several times, the route is deleted. If a node knows only one fastest route, and this route fails, the node has to send out a QRY packet to collect new routing information before it can send data again. This is detrimental to performance. On the other hand, there are no routing updates in regular intervals, so if each node saves all routes it is informed about, some of the routing information may get outdated. However, the routing protocol presented here is based on the assumption that the topology does not change after the nodes have been turned on, so this problem

can be neglected. Therefore each node saves all information it receives about routes. For transmitting data, the nodes select one of the shortest routes from their database.

5.2 Simulation results

We performed only basic simulations with this model since the model with unreliable connections provides more realistic results. In Fig. 5.3 a plot with increasing number of nodes is shown. Since the punching probability of each node remains constant, the number of packets arriving at the base station increases linearly with the number of nodes. Even with 40 nodes, all the data is forwarded safely. The same is true if the punching probability is increased in a network with 20 nodes, as shown in Fig. 5.4: At a probability of 0.002 per node and time unit, which is more than twice the worst-case value, the system still works nicely. To compare the routing protocol directly to the simple broadcasting protocol from Sec. 4.1, we ran a simulation with a circle topology. The system did not collapse even with 42 nodes. The result is shown in Fig. 5.5. We further noticed that in large circles it takes longer until all nodes have learned a route to the base station than in random topologies. In a circle with 20 nodes, the routes are established after 374 time units on average, while it takes only about 200 time units in a random grid network.

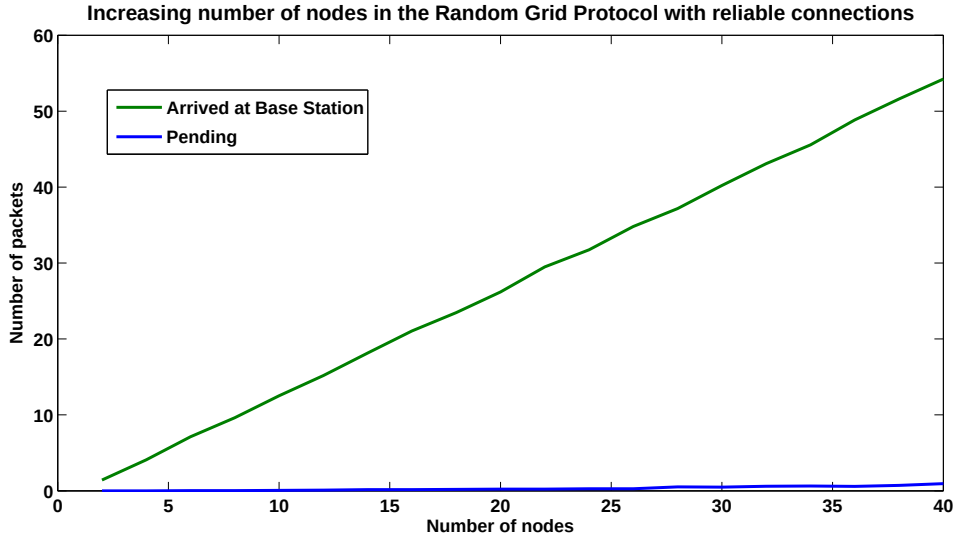

Figure 5.3: Simulation of the random grid routing protocol with increasing number of nodes and constant density, i.e. increasing size of the grid.

5.3 Interpretation

It becomes clear from the results that the presented routing protocol performs much better than a simple broadcasting protocol. The direct comparison of the two protocols in a circle topology, where broadcasting works with up to 25 nodes and routing with more than 42, is a good proof for this. The network can handle more traffic than required in the worst case. However the simple network model used in these simulations completely neglects the possibility of bad connections and lost packets. Only a simulation which takes this into account will show how much overhead is created by retransmissions and whether the protocol deals with lost connections correctly. Furthermore there may be interference with packets from very distant nodes, so

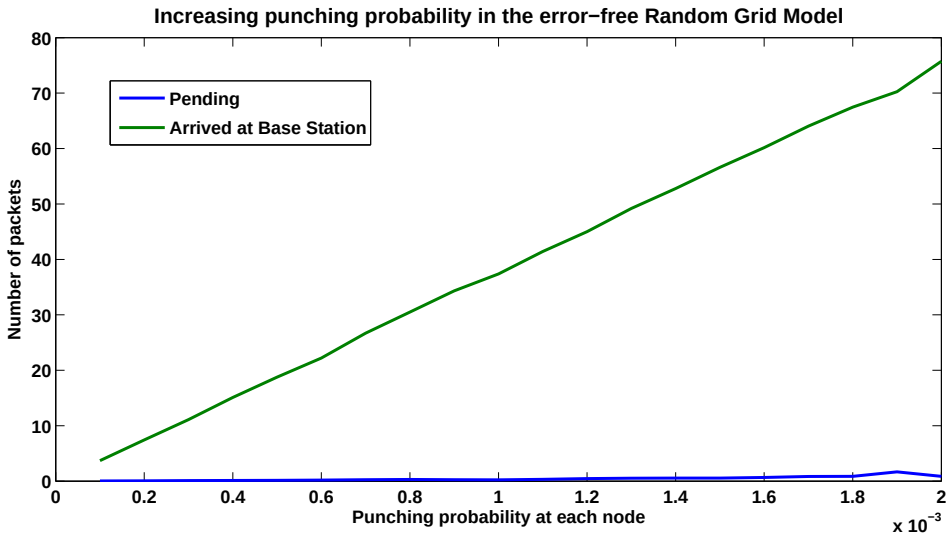

Figure 5.4: Simulation of the random grid routing protocol with 20 nodes and increasing punching probability.

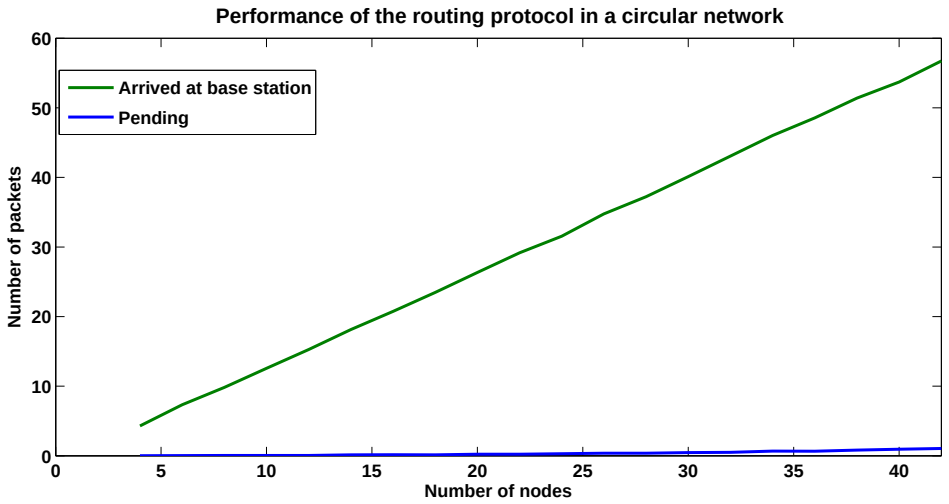

Figure 5.5: Application of the routing protocol to a circle topology with up to 42 nodes.

collisions will be more probable in a more realistic network model. We therefore performed most of the simulations in a network model with path loss (see Chapter 6).

Chapter 6

Routing with unreliable connections

One of the most important aspects of the simulations was to make sure that routing works well even when connections are unreliable. A simple path loss model was used to estimate the probability of packet loss for each connection and to handle colliding packets correctly.

6.1 Protocol description

As in the model with reliable connections, the topology was generated by randomly placing controls in a grid of fixed size. The grid spacing was assumed to be 400 meters, so that a distance of two units in horizontal and one in vertical direction (or vice versa) corresponds to 1 km. The requirement for an acceptable topology has been changed: We now consider a node to be *connected* if the probability that a sent packet arrives at one or more neighbors of the node is at least 95%. Only connected neighbors are considered. The nodes are marked as connected in an iterative algorithm: Initially only the base station is connected, then its closest neighbors fulfill the criteria to be marked as connected, and in the next iteration the neighbors of these neighbors can get connected, etc. If there are unconnected nodes left in the end, the topology is discarded and a new one is generated.

In this network model the definition of neighbors from Sec. 2.2 is no longer valid. Instead, all nodes which can receive a node's packets with at least 1% probability are considered neighbors of the node. There is no longer a defined range; instead it is determined separately for each packet at which neighbors it arrives. Most nodes will have neighbors which almost never receive their packets, but they still have to be in the nodes' neighbor list since their packets may cause collisions. Also, it is important that the nodes find out themselves which of their neighbors provide a reliable connection, so they must not be provided with any information about link quality.

During simulation, the simple path loss model described in Sec. 3.5 is applied to determine which packets arrive at each node. It is assumed that erroneous packets are detected and discarded by the receiving modem, so the protocol does not deal with error detection. The channel coefficients h_i are randomly chosen for each packet, and a packet arrives if its SINR is larger than the SNR_{min} of the receiver. This means that in case of a collision, the packet with the biggest received power may still arrive if the interfering packets are weak enough.

6.2 Lock-in prevention

When adapting the routing protocol to the unreliable network model, some small modifications were necessary to avoid that packets can be locked in. The problem was that each node forwarded a packet only

once. Packets were identified by the ID of their source and a message ID generated by the source. When a node received a packet it had already seen, it would acknowledge it as usual, but then the packet would be discarded. While this is useful to avoid duplicate packets (e.g. in case an ACK gets lost), it does not allow for a packet to be returned to a node that has previously forwarded it. Such returns can become necessary when a route fails due to a bad radio link. Therefore a simple mechanism has been introduced to avoid this problem: Now the nodes additionally store the ID of the sender of each packet. If a packet with a known source ID and message ID arrives a second time, but from a different sender, it is forwarded again. This way, lost ACKs will still not produce duplicates, but packets can be sent and returned several times. It is no longer possible to send a packet over several routes simultaneously, but that does not make sense with this protocol anyway.

6.3 Simulation results

First the parameters of the routing protocol were determined. The *maximum duration of random backoff* was varied; for results see Fig. 6.1. To avoid repeated collisions, the parameter should be at least 16; we set it to 20 for the remaining simulations. Then we determined the *maximum number of retransmissions* before a connection is considered broken. The number of lost connections is plotted against this parameter in Fig. 6.2. The parameter was set to 4. A lower value would cause connections to be lost after collisions, which is not intended. Larger values may be chosen if the nodes are further apart, but the parameter value should not be too high since retransmissions take a lot of time. The *timeout* was set to one time unit since all packets are acknowledged immediately. Furthermore, the *probability that a QRY packet is sent* had to be optimized. This parameter is mainly important when the nodes are turned on, since that is when all nodes send out QRY packets to get information about their neighbors. For statistical reasons the parameter was set to 0.05: In a network where each node has four neighbours, the collision probability would then be

$$\begin{aligned} \mathrm{P}_{\text{collision}} &= 1 - \mathrm{P}[\text{one QRY packet sent}] - \mathrm{P}[\text{no QRY packet sent}] \\ &= 1 - 4 \cdot \mathrm{P}_{QRY} \cdot (1 - \mathrm{P}_{QRY})^3 - (1 - \mathrm{P}_{QRY})^4 \;=\; 0.0140 \end{aligned}$$

To avoid collisions of QRY packets with RAK packets from other nodes, a special mechanism is used: Upon receiving a QRY packet, the nodes *refrain from sending a QRY packet* for a few time units. A reasonable duration of this delay was determined to be 10 time units. The remaining parameters were chosen as in the broadcasting protocol: The message library size was set to 50, and the send queues were not limited in length.

The simulations with increasing punching probability and increasing number of nodes have been done in the same way as for the error-free network model. Their results are given in Fig. 6.3 and Fig. 6.4, respectively. The maximum punching probability in a network with 20 nodes is 0.0009, which is only slightly more than the worst-case value. With higher probabilities the network collapses quickly. Also if the number of nodes is increased while the punching probability is left constant, the margin is small: The packets start queuing up if there are 24 or more nodes. In this simulation the density was constant, i.e. the number of grid fields was scaled along with the number of nodes. In another simulation, we reduced the grid spacing to 100 m and ran a simulation with up to 25 nodes in a 5×5 grid, to see whether a high density would be a problem. With 100% density and such a small grid, all nodes are connected to each other. As shown in Fig. 6.5, all messages arrive correctly even with 25 nodes.

6.4 Interpretation

The system works in worst-case conditions, so the routing protocol handles bad connections and interference quite well. However the margin is very small; a slightly increased traffic is enough to prevent successful transmission of all data. If the radio controls are used at competitions with many runners, i.e. close to the worst-case scenario, the traffic should be reduced to avoid excessive delaying of packets. This could be

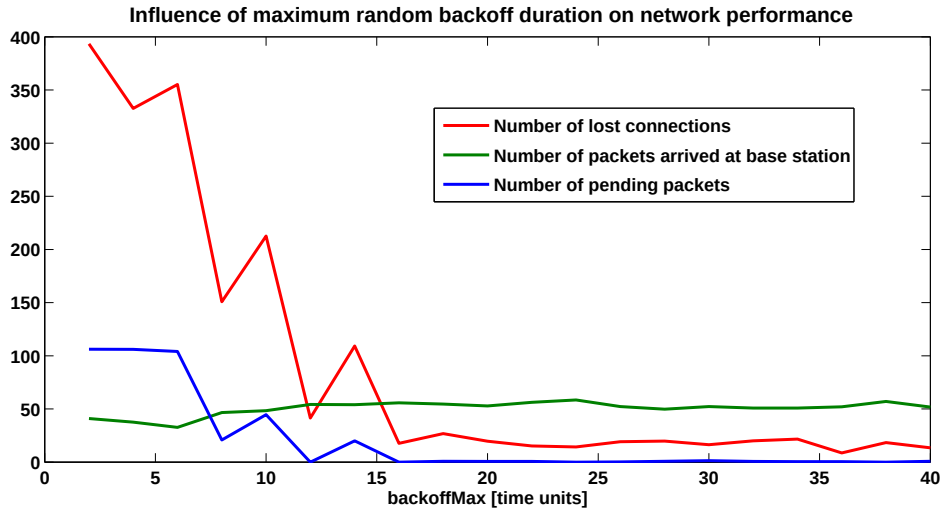

Figure 6.1: Impact of the maximum duration of random backoff on the number of lost connections and untransmitted packets.

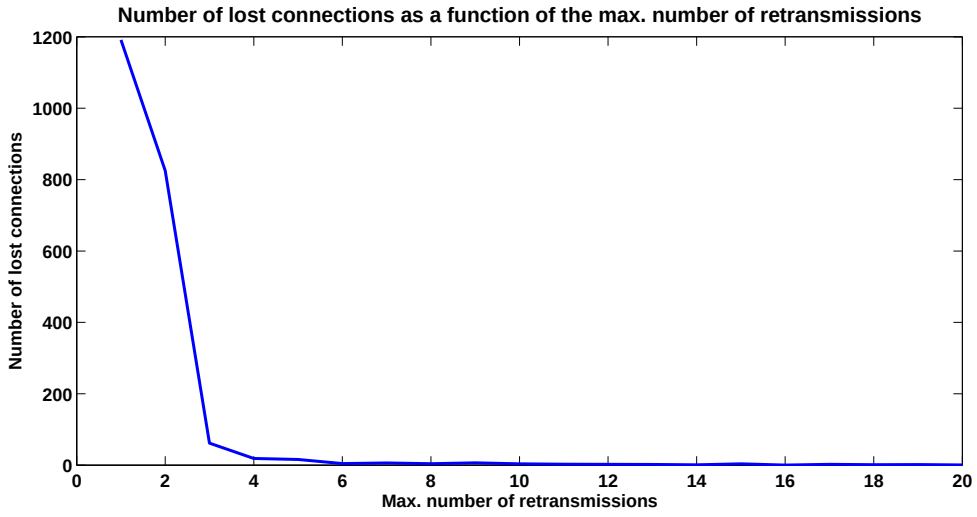

Figure 6.2: The maximum number of retransmissions is a trade-off between too many lost connections and too many pointless attempts to send over a bad connection.

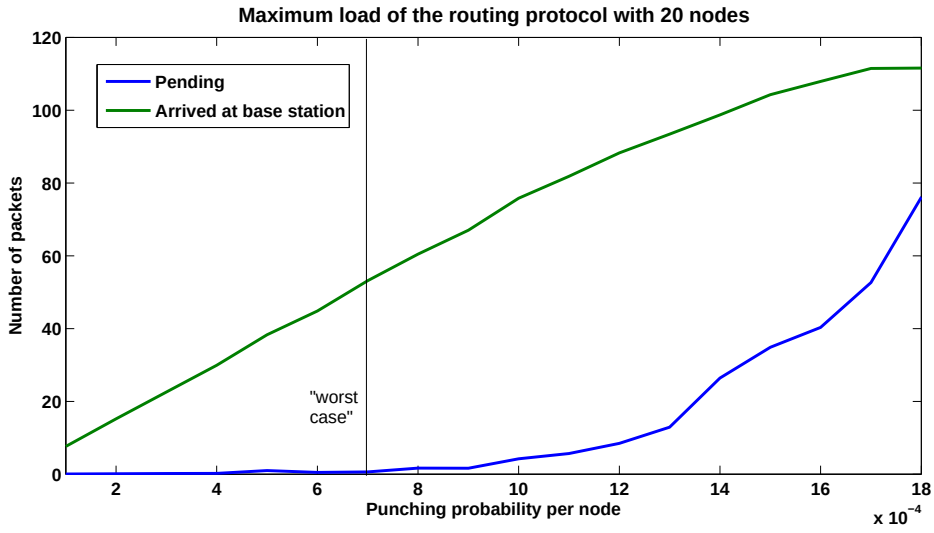

Figure 6.3: Simulation of the unreliable random grid model with 20 nodes and increasing punching probability at each node.

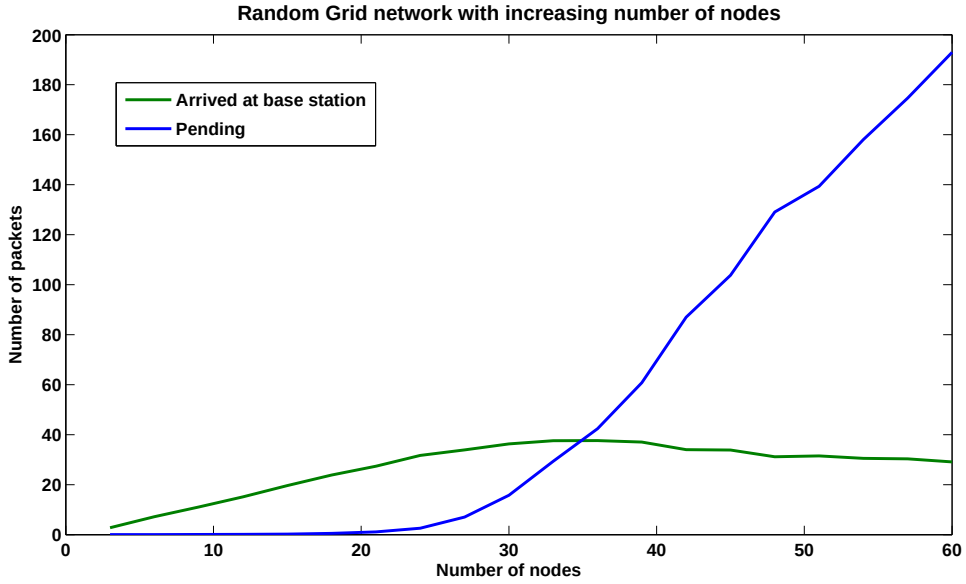

Figure 6.4: Simulation of the unreliable random grid model with increasing number of nodes and constant density, i.e. increasing size of the grid.

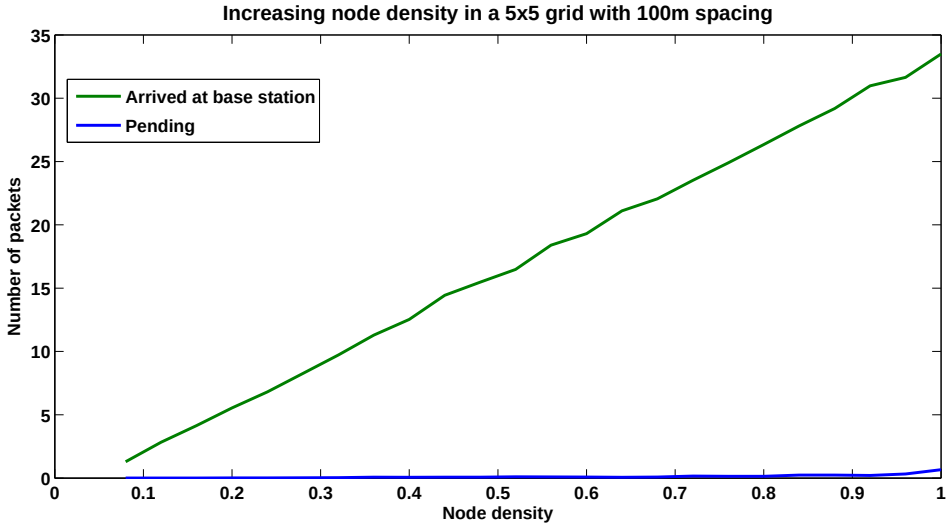

Figure 6.5: Simulation with 5x5 grid and increasing number of nodes. Even if the nodes are just 100m apart (i.e. the network graph is a clique), the network can handle the traffic of all 25 nodes.

achieved by using only ten radio controls and letting the other nodes act merely as relays.

A high node density is not problematic; even if 25 nodes are placed close together, the network does not get congested. This was to be expected since routing is easy when all nodes have a direct connection to the base station. Still, it shows that collisions are handled well by the routing protocol's random backoff mechanism. Also the collision probability is quite low in the first place due to the lack of periodical polling packets.

Chapter 7

Conclusions

We can conclude that it is feasible to build radio controls that will be more *user-friendly* than the ones currently in use in Switzerland. If a multi-hop radio system is implemented, the transmit power of each node can be reduced to 500 mW. This allows for devices that are no larger than a radio handset. The routing protocol that has been defined allows for *automatic configuration* of the multi-hop network, so that the radio controls can be installed without the need for an expert. The protocol is optimized for low overhead; it works without sending periodical polling packets by including some minimal routing information in each data packet. There are several license-free frequency bands available where a transmit power of up to 2.5 W is allowed. The one with the best propagation characteristics in forests is the *173 MHz band*. In this band, a range of at least 1 km can be achieved with 500 mW transmit power and a receiver sensitivity of -115 dBm. This is sufficient to connect all radio controls at a typical orienteering event without placing any relays outside the normal control locations. A simple path loss model was used to simulate the attenuation of the signal as a function of distance from the sender. The routing protocol has been tested in the assumed worst-case scenario and works well. However the *margin is small*: If there are more than 24 radio controls instead of the assumed 20, the network collapses. In events with more than 2000 runners or with very frequented radio controls, some of the nodes could be used only as relays to reduce traffic. Since the system has not been implemented in hardware, the range and performance figures have yet to be verified.

Lessons learned

It took longer than expected to write this report. Some simulations had to be repeated since the collected data were too distorted to create a meaningful plot. If the simulations had been properly finished and documented before introducing a new network model, this problem could have been avoided. Another insight was that it is very important to know the limitations and inherent errors of the models that are used.

Chapter 8

Outlook

Now that a system has been elaborated in theory, it would be interesting to implement it in hardware. This will be done within the scope of a Master's thesis starting in sprint 2011. A suitable radio modem for the 173 MHz band has yet to be found. The simulation models will certainly be useful in the Master's thesis as well, and after conducting some field experiments with radio nodes, it will hopefully be possible to improve the accuracy of the models.

The routing protocol could be enhanced with some configuration features. Especially it would be useful to have a command to poll nodes from the base station to see whether they are connected. This could be done by simply broadcasting the command in the entire network. The reply could then be forwarded over the usual route. To further improve user-friendliness, a small display could be added to each control to show the current status of the radio connection.

Bibliography

[1] “Emit AS,” 2010. [Online]. Available: http://www.emit.no/

[2] “SPORTident GmbH,” 2010. [Online]. Available: http://www.sportident.de/

[3] “Velpoz Schweiz: Was machen wir?” 2010. [Online]. Available: http://www.velpoz.ch/schweiz/about.php. Archived at: http://www.webcitation.org/5q1aRP4Dj

[4] “OTS - Orienteering Telemetry System,” 2010. [Online]. Available: http://www.gpprojects.com/si.htm. Archived at: http://www.webcitation.org/5pxgg89Iz

[5] “RACOM MR25 radio modem,” 2010. [Online]. Available: http://www.racom.eu/eng/products/mr25.html. Archived at: http://www.webcitation.org/5q0g0BV2m

[6] “Frequency Allocation Plan - 420-428 MHz,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/nafz.pl?freq_min=425&freq_max=426&freq_min_unit=1000000&freq_max_unit=1000000&Search=Search&use0=ON&usew0=9&use5=ON&usew5=27&use6=ON&usew6=21&use7=ON&usew7=7&use8=ON&usew8=18&use9=ON&usew9=18&set=PU&user=appnaf&pwd=saperl1p0pot&state=selected. Archived at: http://www.webcitation.org/5qhNH8GQ2

[7] “Velpoz Schweiz: Mitglieder,” 2010. [Online]. Available: http://www.velpoz.ch/schweiz/mitglieder.php. Archived at: http://www.webcitation.org/5pxh7NyqD

[8] “ISG.EE: Condor High-Throughput Computing,” 2010. [Online]. Available: http://computing.ee.ethz.ch/Services/Condor. Archived at: http://www.webcitation.org/5pxaFGsiF

[9] “ETSI EN 300 220 - Parts 1 to 3,” 2010. [Online]. Available: http://pda.etsi.org/pda/queryform.asp

[10] “Technical Interfaces Regulations - 173 MHz non-specific SRDs up to 500 mW,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/rir.pl?id=1021;nb=04. Archived at: http://www.webcitation.org/5pxfkBqD2

[11] “Technical Interfaces Regulations - 173 MHz non-specific SRDs up to 2.5 W,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/rir.pl?id=1021;nb=09. Archived at: http://www.webcitation.org/5pxfmQ02z

[12] “Technical Interfaces Regulations - 433/434 MHz non-specific SRDs up to 500 mW,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/rir.pl?id=1021;nb=05. Archived at: http://www.webcitation.org/5pxfpuuif

[13] “Technical Interfaces Regulations - 433/434 MHz non-specific SRDs up to 2.5 W,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/rir.pl?id=1021;nb=06. Archived at: http://www.webcitation.org/5pxfyFfqp

[14] “Technical Interfaces Regulations - 869 MHz non-specific SRDs,” 2010. [Online]. Available: http://www.ofcomnet.ch/cgi-bin/rir.pl?id=1008;nb=09. Archived at: http://www.webcitation.org/5pxg2mRMX

[15] "Amber Wireless AMB8355 Datasheet," 2010. [Online]. Available: http://amber-wireless.de/files/amb8355_db.pdf. Archived at: http://www.webcitation.org/5q0nvX8qo

[16] "Telit PowerOne Terminal Datasheet," 2010. [Online]. Available: http://www.telit.com/module/infopool/download.php?id=1381. Archived at: http://www.webcitation.org/5q0pyQmvg

[17] "Telit TinyOne Pro RF Modules Datasheet," 2010. [Online]. Available: http://www.telit.com/module/infopool/download.php?id=1374. Archived at: http://www.webcitation.org/5q0ph0nTw

[18] "Atim ARM-C8 Datasheet," 2010. [Online]. Available: http://www.atim.com/IMG/pdf/UKDS_ARMC8-3.pdf. Archived at: http://www.webcitation.org/5q1UUcffY

[19] "Radiometrix BiM1H Transceiver Datasheet," 2010. [Online]. Available: http://www.radiometrix.com/files/additional/bim1h.pdf. Archived at: http://www.webcitation.org/5q1Vy8REi

[20] T. Tamir, "On radio-wave propagation in forest environments," *Antennas and Propagation, IEEE Transactions on*, vol. 15, no. 6, pp. 806 – 817, nov 1967.

[21] ——, "Radio wave propagation along mixed paths in forest environments," *Antennas and Propagation, IEEE Transactions on*, vol. 25, no. 4, pp. 471 – 477, jul 1977.

[22] A. Magazinnikova and V. Yakubov, "Attenuation of coherent radiation in forest regions," *Microwave and Optical Technology Letters*, vol. 19, no. 2, pp. 164 – 168, 1998.

[23] V. Mironov, V. Yakubov, E. Telpukhovskiy, S. Novik, and A. Chukhlantsev, "Spectral study of microwave attenuation in a larch forest stand for oblique wave incidence," in *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, vol. 5, 25-29 2005, pp. 3204 – 3207.

[24] J. B. Johnson, "Thermal agitation of electricity in conductors," *Phys. Rev.*, vol. 32, no. 1, p. 97, Jul 1928.

[25] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach, 4th Edition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.