Cumulus Humilis: Wireless mesh-networking for gliders

Roel Baardman

UNIVERSITY OF TWENTE.

PERVASIVE SYSTEMS
RESEARCH GROUP

Cumulus Humilis: Wireless mesh-networking for gliders

Roel Baardman

Master Thesis

Pervasive Systems group
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

Supervisor:
Dr. Ir. Nirvana Meratnia

Graduate Committee:
Dr. Ir. Nirvana Meratnia
Ir. Hans Scholten
Prof. Dr. Ing. Paul Havinga

Coverpage image by Tobias Oort

Abstract

During cross-country flights, it is common to carry a mobile computer in the cockpit of the glider. The mobile computers are used to assist the pilot with navigation and computation of the optimal speed for the pilot to fly. For these purposes, the mobile computer is connected to a GPS receiver and gathers a vast amount of information during the flight. The exchange of this information has the potential to provide insight not only to the pilot himself but also to other pilots and people on the ground. If the information could be sent to the ground, family and friends of the pilots could watch the flight live. If a pilot experiences an emergeny situation, a search-and-rescue team can directly know his location. This can be vital for the pilot's survival. Also, information from the ground can be of great value to the pilot. Delivery of large-scale weather developments to the pilot enables him to adapt his strategy early and not to be surprised by the changing weather conditions. For example when high clouds approach, block incoming sunlight and thus reduce the formation of rising air, the pilot could then be informed to head home earlier to prevent an outlanding. During measurement flights, the real-time communication of sensor readings and feedback from the ground to the pilot can greatly improve the productivity of each measurement flight.

To the best of our knowledge, there exists no low-cost solution for communication and information exchange to this extent between gliders. In this thesis, we aim at designing such a solution using *opportunistic dissemination* through a *mesh network*. Using existing long-range transceivers, a mesh network - a network in which each participant(node) can act as a router - is formed. Each node in the network can move around freely and the network can still operate when a node breaks down or a connection breaks down. To cope with situations in which very few gliders fly, we use *opportunistic dissemination*. Opportunistic dissemination utilizes the storage available on the nodes in the network to physically transport information. A node can remember information, to forward it when it is connected to other nodes interested in this information. Over the network thus created, information can be disseminated. In this paper, we focus on disseminating three types of information, namely (i) High-priority periodical positional information: used for example for tracking and emergencies, (ii) Low-priority large bursts of information: e.g. weather information, and (iii) Low-priority periodical large amounts information: e.g. sensor readings.

A challenge faced in designing our opportunistic dissemination is how to handle the different types of information. While the network should be able to cope with the movement of the nodes(*mobility*), it should still be able to scale to a large number of nodes(*scalability*). It may also not waste much communications on operations not directly related to dissemination of information (*communication overhead*), for example maintaining connections to nearby nodes. Therefore, we take support for mobility, scalability and low communication overhead as the main requirements while designing our dissemination protocol. In addition, we investigate applicability of existing routing and dissemination protocols for information exchange between gliders.

Routing approaches can be categorized into proactive and reactive approaches. Proactive protocols attempt to create an overview of the network topology, and use this overview to route content across the network. Reactive protocols try to create an overview of network topology when it is needed, that is when data are sent by

a node. A proactive approach generally imposes more overhead, as it attempts to continuously keep track of changes in network topology. Reactive approaches do not impose extra overhead, as they only try to discover the current network topology when data is to be sent. This scales better to large numbers of nodes, since the amount of control information sent does not grow linearly with the number of nodes in the network.

Dissemination protocols can be categorized into approaches dependent on infrastructure, such as routing information, and approaches independent of infrastructure. Approaches dependent on infrastructure work well up to a certain level of mobility. Above this level of mobility, the infrastructure can not be maintained and the dissemination stops working. Approaches independent on infrastructure to exploit mobility and the more or less random connections between nodes. Nodes opportunistically send information to other nodes, which eventually reaches a node interested in this information.

In order to gain insight on the mobility, we analyzed GPS logfiles of real simultaneous flights offline. Analyzed parameters include: (i) duration of connectivity between nodes, (ii) average distance between two connected nodes, and (iii) common speed component of two connected nodes. We also evaluate existing approaches to opportunistic dissemination and routing approaches in mesh networks based support for mobility, scalability and communication overhead. To test solutions for disseminating information over the network, a simulator was written which uses the same GPS logfiles as were used in analysis. This simulator emulates the movement of nodes and models the signal strength of transmissions.

Over 2600 analyzed connections, the average duration of a connection between two gliders is 54 minutes and during the connection the average distance is 26 kilometers. The common speed component between two nodes has a distribution resembling normal distribution. Around a common speed component of 100 to 120 km/h the longest connections are established. Above 140 km/h connection times are rapidly getting shorter.

Contents

List of Figures

List of acronyms

AODV	Ad-Hoc On-demand Distance Vector
API	Application Programming Interface
CRC	Cyclical Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
EDF	Earlyest Deadline First
EIRP	Equivalent Isotropically Radiated Power
FLARM	Flight Alarm
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
Kbps	Kilo bits per second
KML	Keyhole Markup Language
MAC	Medium Access Control
ODMRP	On-Demand Multicast Routing Protocol
OLSR	Optimized Link Source Routing
QoS	Quality of Service
RP-SMA	Reverse Polarity SubMiniature version A
RS232	Recommended Standard 232
RTS	Request To Send
SNR	Signal-Noise Ratio
SRD	Short Range Device

TDMA Time Division Multiple Access

UMTS Universal Mobile Telecommunications System

USB Universal Serial Bus

1 Introduction

1.1 An introduction to gliding

> If flying were the language of mankind, gliding would be its poetry.
>
> Unknown

Gliding is a recreational activity and sport in which pilots fly aircrafts with no engine, commonly known as gliders. Gliding may be in a form of aerobatic contests, in which pilots perform pre-defined sequences of aerobatic maneuvers as precisely as possible, or cross-country flying, in which pilots regularly fly hundreds of kilometers. National and international competitions are held worldwide. Many glider pilots enjoy the interaction with nature and the silence of unpowered flight and experience gliding as the most pure form of flight.

1.1.1 An introduction to unpowered flight

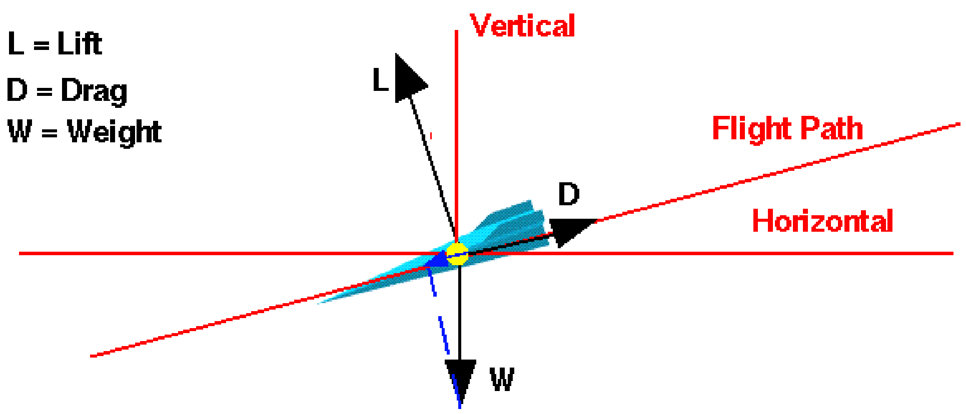

Figure 1.1: *Gliding flight. The forces acting on the glider - lift, drag and gravity - are shown. Gravity is decomposed along the longtitudal axis of the glider in blue. Image adapted from Nasa [1].*

When a glider flies with a constant speed, it is in uniform motion. Like in any uniform motion, there exists an equilibrium of forces. The forces acting on a glider and the equilibrium of forces are shown in Figure 1.1 . The forces acting on the glider are gravity, drag and lift produced by the wings. Unlike powered aircrafts, there is no engine pulling the glider forward. Therefore, the drag acting on the glider must be compensated in another way. This is why gliders fly with the nose underneath the horizon and glide

Figure 1.2: *A Jonker Sailplanes JS1 Revelation, a glider with a wingspan of 18 meters and a lift-drag ratio of 53 to 1. [2]*

downwards constantly. Because the nose of the glider points towards the earth, the force of gravity can be decomposed into a force parallel to the longtitudinal axis of the glider. This force compensates for the drag and keeps the glider flying at a constant speed.

Gravity is compensated by the lift. Lift is generated by the wings when air flows over the wing, due to the shape of the wing. The shape of the wing is such that air flowing over the top of the wing has to travel a larger distance than air flowing underneath the wing. The air flowing over the wing flows faster than the air flowing underneath the wing. This causes the pressure of the air over the wing to decrease. Due to the pressure difference over and below the wing, the glider is pulled upwards.

Gliders have great aerodynamic efficiency in order to create as much lift with as little drag as possible. Therefore gliders have aerodynamic features that can be seldom found in other aircrafts, and can be seen in Figure 1.2 . The wings are long and have a special low-drag airfoil. Vertical winglets at the end of the wing improve handling of the aircraft and performance even more. The aerodynamic features, such as the shape of the airplane and the vertical winglets, result in a lift-drag ratio of up to 60 to 1. This means each unit of drag that is produced, 60 units of lift are generated. In practice this means that with 1 kilometer of altitude the glider can travel 60 kilometers horizontally before hitting the ground.

Most modern gliders are able to fly from roughly 70 kilometers per hour, and have a maximum speed of 200 to 300 kilometers an hour. The construction of the glider usually allows G forces up to 5 G's: at 5G the wings of the airplane carry 5 times the weight of the airplane. The weight of a glider also varies, from 200 kilos to 850 kilos.

1.1.2 Cross-country gliding

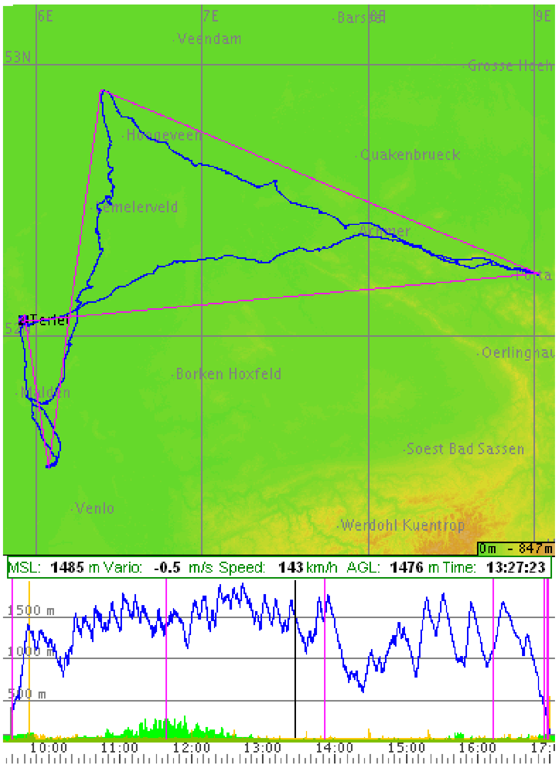

Figure 1.3: *A cross-country flight visualized. The top of the image shows the path the pilot flew. The bottom shows the altitude of the glider during the cross-country fllight. Copyright: Online Contest [3].*

During cross-country flights, distances of several hundreds of kilometers are flown in the course of up to eight hours. An example of analysis a cross-country flight can be seen in Figure 1.3 . To travel distances bigger than the gliding capacity of the glider (e.g. 60 kilometers per 1 kilometer of altitude) the pilot seeks thermals i.e., bubbles of warm air. When the sun shines on the surface of the earth, air is heated and a sticky layer of hot air is formed above the ground. This layer of air can be observed for example when the asphalt of a highway appears to reflect in the distance. From this layer of hot air, thermals separate. The thermals are warmer than the surrounding air, which causes them to rise. When the air in the thermal becomes saturated with water vapour, the thermal forms a cumulus cloud. This cumulus cloud is an indication for the pilot where thermals may be found. Thermals are generally several hundreds of meters in diameter, which enables glider pilots to circle in them. When the pilot circles in a thermal, he gains altitude through the thermal. He can then convert this altitude into distance, using the aerodynamic efficiency of the glider. The altitude of the glider ranges from around 500 to 3000 meters above the ground in The Netherlands, depending on weather conditions and airspace restrictions. For example, the west of The Netherlands has very little gliding activity due to the airspace reserved for Schiphol Airport.

Most pilots carry a mobile computer, in combination with a Global Positioning System (GPS) receiver, in the cockpit. During the flight, the mobile computer helps the pilot with navigation and flight optimization and creates a GPS logfile, which the pilot can use to prove where he has been. During competitions this proof is used to verify wether the pilot actually flew the flight path.

To prepare for a cross-country flight, the pilot evaluates the weather predictions and temporary changes in airspace structure. The pilot plans his intended route. If the pilot uses a mobile computer, this mobile computer is started before take-off. Since the pilot does not have access to up-to-date information during the flight, he relies solely on his preparation and his observations in the cockpit.

1.1.3 Measurement flight

Gliders are also used for collecting information and measurements for research purposes. One example is the mountain wave project [4], in which measurements are performed on turbulence at the lee side of large mountain chains. At the lee side of mountains, the air can start to oscilate vertically. This oscilation can cause rotor turbulence and breaking waves [5, 6, 7, 8]. Rotor turbulence and breaking waves can cause accidents for the aircrafts in both recreational and professional aviation [6]. Rotor turbulence and breaking waves and where these phenomonae occur can be seen in Figure 1.4. The mountain wave project has the goal of global classification and analysis of mountain waves and their associated effects. The project analyzes the propagation of the mountain waves by flying into the waves and measuring air movement [6]. For this, a glider equipped with a probe is used which is able to precisely measure turbulence. This probe measures the turbulence during flight, which enables the project to verify and correct the mathematical models describing the mountain wave phenomenae.

Similar to pilots of cross-country flights, the pilots of measurement flights rely on good preparation. Before the flights, mathematical models are consulted for possible interesting locations to measure at. The glider then takes off and flies to those locations. After the flight, collected data will be evaluated. During the flight, there is no data exchange between a ground crew performing analysis and the flight crew collecting measurements. Measurements are stored by an on-board computer system and transfered when the glider has landed.

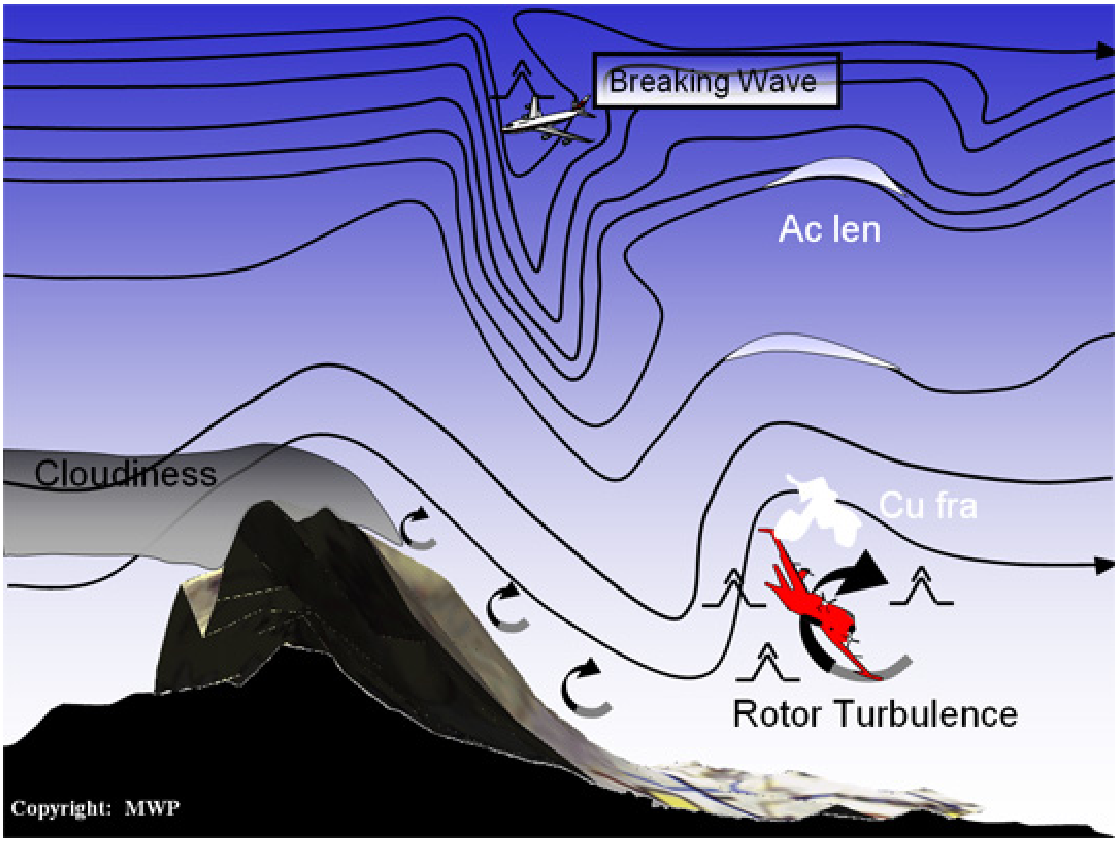

Figure 1.4: *On the lee side of large mountains, mountain waves can form. Oscilating air can cause rotor turbulence and breaking waves. Copyright: Mountain Wave Project [4].*

1.2 Challenges

Besides dealing with aspects of gliding, this also looks at wireless networking. Therefore we give a short introduction on some aspects of wireless networking in this section.

1.2.1 The hidden terminal problem

In wireless transmission, the *hidden terminal problem* is a problem related to medium access control, and is visualized in Figure 1.5. One node, in this case the "Hub", is within the transmission range of both node A and node B. When both node A and node B are transmitting simultaneously, the Hub will receive none of the two transmissions, since these will interfere each other. However, node A and node B are not in each other's transmission range. This makes node A and node B unable to coordinate their respective transmissions such that the Hub node can receive those. This is called the *hidden terminal problem*.

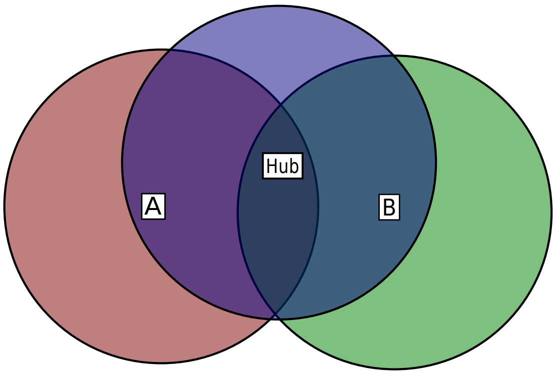

Figure 1.5: *The hidden terminal problem visualized*

1.2.2 Dynamic topology

A common problem in wireless networking is the changing network topology. Due to the fact that nodes in the network move around, connections can break and new connections can become available. To allow two nodes to continue sending packets to each other when a connection breaks, these changes should be known and packets should be routed appropriately considering the new network topology. When topology is such that no direct route from the source of a packet to the destination of a packet is readily available, a *store-and-forward* approach can be utilized to deliver the packet at the destination node.

1.2.3 Quality of Service

Providing Quality of Service (QoS) in mesh networking is another challenge. Since both network topology and signal strength change, prioritizing one type of data over another type of data is not straightfoward. Packetloss and channel characteristics are not predictable, thus sending high-priority data before low-priority data can result in packetloss of the high-priority data.

1.2.4 Broadcast storm

The broadcast storm problem is a problem which accors when many nodes are sending the same information in a network, which causes a lot of medium contention, packet collisions and results in a single node receiving the same information many times. When, in order to spread a single message through the entire network, each node simply broadcasts every packet it receives the broadcast storm problem occors.

1.3 Problem statement

In this thesis we answer the following questions:

- What form of wireless communication is most suitable for the exchange of information between gliders?
- How can be dealt with the hidden terminal problem?
- How can Quality of Service be provided with wireless communication?
- How can be dealt with the mobility caused by gliders flying around?

1.4 Overview of this thesis

In this thesis we design and evaluate a communication system for communication system between gliders.

In Chapter 2 we describe existing communication systems, and evaluates communication media. A communication medium is chosen at the end of this chapter, and this is used in this thesis. In Chapter 3 we analyze the hardware we choose to use, and look at the mobility of gliders during good and poor cross-country days. We also analyze the scenarios in which communication would be benificial, and what information should be sent in this scenario. In Chapter 4 we evaluate existing protocols which provide inspiration for the solution present in this thesis, in Chapter 5 .

In Chapter 5 we describe the solution we present, and we evaluate this solution in Chapter 6 . In Chapter 6 we describe the method we use for evaluation and we show the results we obtain from our evaluation. In Chapter 7 we conclude the thesis, discuss recommendations and describe future work.

2 Communication systems between gliders

Doing more with less

Paul MacCready

In the forms of gliding described in Chapter 1, no digital communication between the glider and other gliders, or the glider and individuals on the ground exists. This would however be benificial.

During cross-country flights, receiving information about the weather can be valuable for the pilot. Delivery of large-scale weather developments to the pilot enables him to adapt his strategy early and not to be surprised by the changing weather conditions. For example when high clouds approach, block incoming sunlight and thus reduce the formation of thermals, the pilot could then be informed to head home earlier to prevent an outlanding.

Being able to follow gliders can be valuable in two ways. Firstly, this enables friends and family of the pilot to see where he is and thus share part of the excitement of cross-country gliding. They can see how his flight progresses. Secondly, if an accident takes place, this allows for quickly determining the location of the pilot. Knowing the last location of the pilot can dramatically reduce the search area for the ground crew, and thus increase the chance for survival of the pilot.

During measurement flights information exchange enables feedback to the measuring glider. Analysis on the ground can determine if the region, which is currently being measured is relevant to the goal of the measurement, and give feedback to the pilots performing the measurements. If multiple gliders are performing measurements simultaneously, they can collaborate and scan an area for interesting phenomonae. When an interesting phenomonon is found, all gliders can then fly towards the area and perform detailed measurements of this phenomonon.

2.1 Generic requirements of a communication system for gliders

A communication system for gliders is used in airborne gliders and is bought as part of a hobby. This yields the following generic requirements for a communication system for gliders:

- *Usability.* Since the target audience is a large group of people - there are for example around 8000 glider pilots in The Netherlands - it is not feasible to require the pilot to pass tests before using a communication system. The system should be usable by everybody.
- *Cost.* Ideally, the system should cost as little as possible. In order to allow as many people as possible use the system, it is especially important that operating

costs are as low as possible, or even free. Hardware does not have to be cheap per se, however this will make the system more accessible.

- *Coverage.* Since pilots usually fly at altitudes ranging from 500 to 3000 meters during cross-country flights, it is important the system has good coverage at these altitudes.

2.2 Existing communication systems

2.2.1 FLARM

FLARM [9] is an active and cooperative traffic and collision warning system for general aviation and recreational flying. FLARM operates in a licence-free radio band, with a typical range of 4 to 8 kilometers. Using GPS information, FLARM performs trajectory prediction. The current position, together with the prediced trajectory, is then broadcasted over the air. Nearby aircrafts receive this information and check for potential collisions. When a device finds a potential collission, it notifies the pilot on a small display. FLARM indicates the horizontal direction of the potential collider and whether it is above of below the pilot.

FLARM is not capable of performing other tasks besides traffic and collision warning. Correspondence with the designers of FLARM revealed that the system has little room for the exchange of other traffic than GPS information and predicted trajectories between FLARM devices. Sending more information is likely to disrupt the traffic and collision warning for which FLARM was designed.

2.2.2 Iridium based tracking systems

During international competitions, tracking systems are often utilized. These tracking systems are often built for the specific competition by local companies. What these tracking systems share is Iridium as their carrier for data.

Iridium is a group of satellites used to provide voice and data coverage to satellite phones, pagers, and integrated transceivers over earth's entire surface. The satellites are owned and operated by *Iridium Communications Inc.* [10]. This company also sells equipment and access to the services provided by the satellites.

Although Iridium will work in most conditions, it is too expensive for every-day use. Iridium hardware generally costs a few hundred dollars. Beside these costs, a monthly fee is required. Moreover there is a fee per data unit.

2.3 Potential data carriers

2.3.1 GSM

At the time of writing this thesis GSM has data service available. These are General Packet Radio Service (GPRS) and Universal Mobile Telecommunications System (UMTS).

An advantage of these data services is the relatively cheap hardware(mobile phones), which is also widely spread. However, there is a fee per data unit. Another downside to GSM is its coverage. Coverage in areas far from urban areas is generally very bad. Also, coverage mostly ends at about 400 to 800 meters above the ground. Since a pilot will mostly fly above 800 meters altitude during cross-country flgiht, he will not have a connection for long periods during the flight.

GSM also has speed limitations that could lead to problems. GSM 1800 has a speed limitation of 130 km/h, whereas GSM 900 has a limit of 250 km/h. It is very likely that gliders will fly above 130 km/h when traveling from thermal to thermal. Analysis of the speed of a glider in several flights has shown us that the characteristics of the glider's performance often lead to traveling speeds of 140 km/h and more between thermals. This would mean that only GSM 1800 is usable.

2.3.2 Iridium

Iridium as a data carrier has the same downsides as Iridium when it is used as a communication system in existing Iridium based tracking systems. The cost for an Iridium modem is high, and the cost per unit of data sent or received is also high.

2.3.3 Amateur Packet Radio

Amateur Packet Radio [11] is a form of packet switching used to transmit digital data via radio links. It uses datagrams to send information. Amateur Packet Radio can operate on a number of frequency bands. Some of these bands are free to use, whereas others require a licence to use.

The frequency bands requiring a licence to use are not an option for a widely used network, since this would require every pilot using the network to pass an exam and effectively become a radio amateur. The frequency bands which are free to use, are in general very much used by all sorts of devices. This, combined with the mostly large range of Amateur Radio, will introduce contention for the medium. Beside this, the regular hardware used for Amateur Packet Radio reaches speeds up to 9600 bits per second, typically 1200 bits per second. This would, combined with large amounts of contention, probably result in unusable speeds.

2.3.4 X-Bee PRO 868

Digi International [12] produces X-Bee PRO 868 transceivers. These transceivers operate on the Short Range Device (SRD) frequency band, which is free to use in Europe. Within the SRD frequency band, there exist a few sub-bands. The X-Bee PRO 868 transceivers operate on the G3 sub-band. This sub-band allows the devices to transmit with a maximum power of 500mW Equivalent Isotropically Radiated Power (EIRP) [13, 14], with a 10% duty cycle averaged over 1 hour. The transceivers have a maximum range of 40 kilometers [15] and have a transmission speed of 24 Kilo bits per second (Kbps). The transceivers have an RS232 interface, which is often used in serial ports.

The X-Bee PRO 868 transceivers cost 72 US Dollar at the time of writing this thesis [16], which we consider cheap enough to satisfy the *cost* requirement, and sending and receiving does not cost extra money. Since the transceivers use a freely operatable frequency band, the *usability* requirement is also met. Since the range of the transceivers is 40 kilometers, coverage can also be satisfied given that enough gliders are airborne. On days with few gliders airborne, coverage can be a problem.

In the rest of this thesis the X-Bee PRO 868 transceivers are used to build a mesh network, i.e. a network without a fixed topology. The design and development of a protocol to handle the exchange of information between gliders, using the X-Bee PRO 868 transceivers, will be the focus of the rest of this thesis.

3 Hardware, software and traffic analysis

Hope is not a strategy.

Unknown

In Chapter 2 we present the reasons for choosing the X-Bee PRO 868 transceivers with mesh networking as the communication medium. Before we look at related work, we perform data analysis. This provides insight on the circumstances we can expect in terms of network topolgy changes and insight on the performance of the chosen transceivers.

First, we describe the X-Bee PRO 868 hardware. Since the X-Bee PRO 868 transceivers use a proprietary MAC layer, we perform measurements to determine the time it takes for the MAC layer to send packets of different sizes. In addition, we mention properties of the MAC layer which are specified by the manufacturer.

Furthermore, we analyze log files from real glider flights to gain insight on possible connection times, the distance between two connected nodes, the number of connections a single node has, and the effect of the speed of and direction to which two nodes are traveling on the time they can be connected. This provides insight on conditions a dissemination protocol should be able to cope with.

We translate the forms of gliding described in Chapter 1 into scenarios for which a dissemination protocol - which we describe in Chapter 5 - could be used. Furthermore, we derive QoS requirements for this dissemination protocol which we can use in Chapter 6 .

3.1 X-Bee PRO

We analyze the X-Bee PRO 868 transceivers with respect to features relevant to this thesis. Figure 3.1 illustrates X-Bee PRO 868 transceivers.

3.1.1 Overview

The X-Bee PRO 868 transceivers operate on the 868 MHz SRD G3 band, which is a free band in Europe. In this band, the transceivers can transmit with 500mW and are limited to a 10% duty cycle. With this transmission the transceivers have a maximum Line-of-Sight range of 40km, using a dipole antenna. The transceivers have an RF data rate of 24Kbps, and are controllable over an Recommended Standard 232 (RS232) interface.

3.1.2 MAC layer

The MAC layer used by the X-Bee PRO 868 transceivers is similar to the 802.11 MAC layer [17]. It utilizes Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with RTS and CTS messages to address the hidden terminal problem for unicast messages. An ACK message is used to notify the sending node, that the receiving host has succesfully received the sent packet. For broadcast messages Carrier Sense

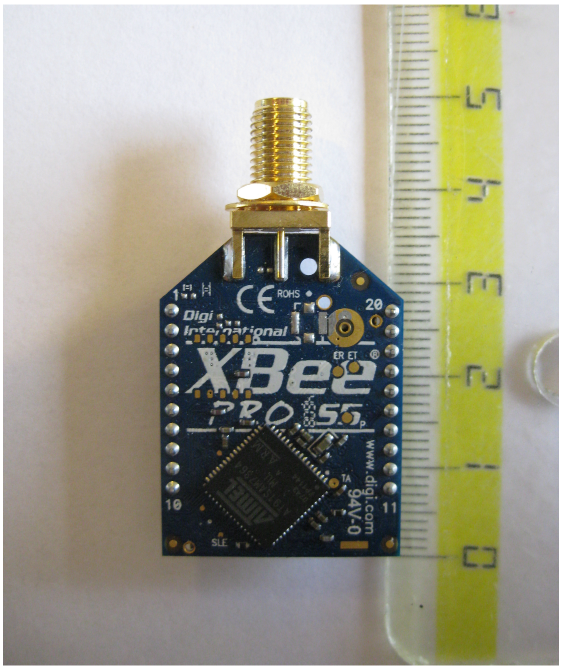

Figure 3.1: *An X-Bee PRO 868 module, with an RP-SMA connector.*

Multiple Access (CSMA) is used without RTS and CTS messages. All packets include a Cyclical Redundancy Check (CRC) at the end to ensure packet integrity.

3.1.3 Time required to send a packet

Since the transceivers utilize a proprietary MAC layer, we determine the performance of the MAC layer. For this, we write a C program which communicates with the transceiver. The program sends packets of random sizes containing random data. For each packet, we determine the time it takes to to be sent. After a packet of each packetsize is sent 250 times, we determine the average time required for each packetsize to be sent.

The time required to send a packet is determined through interaction with the X-Bee PRO 868 transceiver. First, we send an Application Programming Interface (API) command to send the packet to the the X-Bee PRO 868 transceiver, after which a timer is started. The X-Bee PRO 868 transceiver then attempts to send the packet. After sending the packet, the X-Bee PRO 868 transceiver notifies the C program. The timer is then stopped. The value of the timer is then used as the time to send a packet. Although this includes the time it takes for the API command to travel over the RS232 link to the X-Bee PRO 868 transceiver and the time it takes for the X-Bee PRO 868 transceiver to send the notification over the RS232 link, it is representative. Early attempts to send multiple frames and not wait for the notifications resulted in errornous behaviour of the X-Bee PRO 868 transceivers. Thus we conclude the notification sent by the X-Bee PRO

868 transceivers *must* first be received before a new packet can be sent.

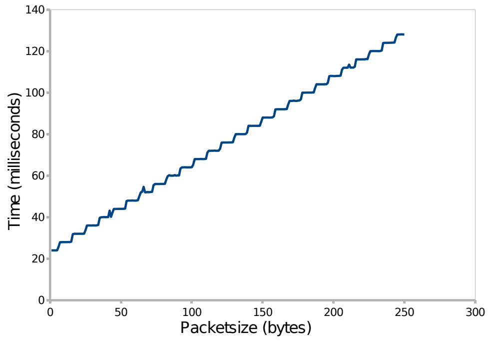

Figure 3.2: *Packetsize and the time it takes to send a packet of that size.*

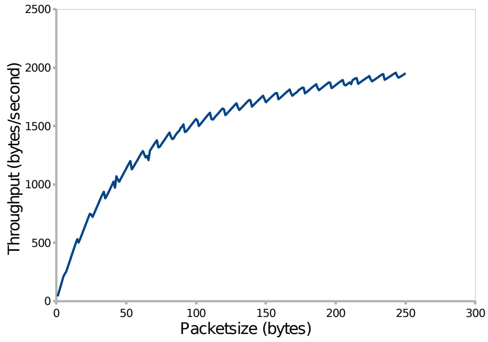

Figure 3.3: *Packetsize and the throughput with that packetsize.*

Figure 3.2 shows the time it takes to send packets of a particular size. It can be seen that a pattern with the shape of a staircase emerges. This pattern can be explained by buffering performed by the RS232 controllers in the computer and the X-Bee PRO 868 transceiver. Early measurements used a Universal Serial Bus (USB) connection to the X-Bee PRO 868 transceiver, with the X-Bee PRO 868 transceiver mounted on a USB-to-RS232 converter board. These measurements showed the same pattern with larger intervals between the jumps. Investigation into RS232 emulation in the USB protocol shows this interval was similar to the maximum packet size of the USB protocol. We conclude that the USB controller attempts for a certain amount of time to fill packets with data before sending the packets. We conclude that the RS232 controller has the same behaviour, using a smaller buffer. Figure 3.3 shows the throughput for different packet sizes.

3.1.4 Conclusion

In this section we evaluated the X-Bee PRO 868 transceivers which are used in this thesis. We analyzed the MAC layer and measured the time required to send a packet of a certain size. This revealed only one packet can be sent to the transceivers at a time, and showed a linear relation between the size of a packet and the time required to send it. This relation is can be estimated as $\lfloor 0.4202 \cdot \text{bytes} + 23.531 \rfloor$.

3.2 Mobility Analysis

3.2.1 Approach

We use logfiles, downloaded from Online Contest [3] of real simultaneous flights for mobility analysis. Each file is stored on the mobile computer carried by the pilot in the cockpit. A logfile contains among other things a series of time-stamped GPS locations.

Different logfiles have a different sampling frequency at which GPS data is stored. The sampling frequency remains constant during one flight.

Downloaded from the Internet, we group logfiles of the same days. The reason for doing so is to have groups containing good and poor cross-country flying conditions. This enables us to analyze both conditions with many and few gliders airborne.

To analyze the files, we create a script written in PHP scripting language. This script reads all the logfiles and produces a table for each logfile. Each entry in this table contains a timestamp and the location of the glider at that time. However, not all logfiles contain GPS locations at precisely the same timestamps. Therefore, for fixed timestamps the location of the glider is calculated. If the location for a specific timestamp is not known, it is calculated via interpolation.

Optionally we also use another file, which contains the locations of all dutch gliding clubs. This file enables simulation with stationary nodes at airfields, which can improve connectivity and allow individuals on the ground to access the network.

After reading all these files, we check connectivity. At every time instance, connectivity between every pair of two nodes - computers in a gliders cockpit or a computer at an airfield - is checked. Two nodes are assumed to have a connection when their relative distance is less than 40 kilometers, which is the maximum range of the X-Bee PRO 868 transceivers used in this thesis. In the case of available connectivity, a record for each link will be created containing:

- The nodes involved,
- The first moment the connection is available,
- The last moment the connection is available,
- The duration of the connection,
- The average distance between the involved nodes during the available connectivity,
- The vector dot-product between the average speeds of the two nodes involved,

Next, we analyze the fan-out of each node at each point in time. At each point in time, for each node, a set with all the node's direct neighbours and the node's neighbour's neighbours is created. We store the overall maximum fan-out, as an indication to how sparse or dense the network is for certain conditions. This provides insight on, for example, the effect of nodes at airfields.

We also write a Keyhole Markup Language (KML) file - which can be viewed in Google Earth - containing records of all GPS locations at all available timestamps, and records of all links. This allows a visual check on the correctness of the script and to get insight on the circumstances. Links are colored depending on the distance between the two involved nodes: a distance of 20 to 40 kilometers is colored red, a distance of 10 to 20 kilometers is colored yellow, and a distance smaller than 10 kilometers is colored green. Figure 3.4 and Figure 3.5 illustrate the generated KML files.

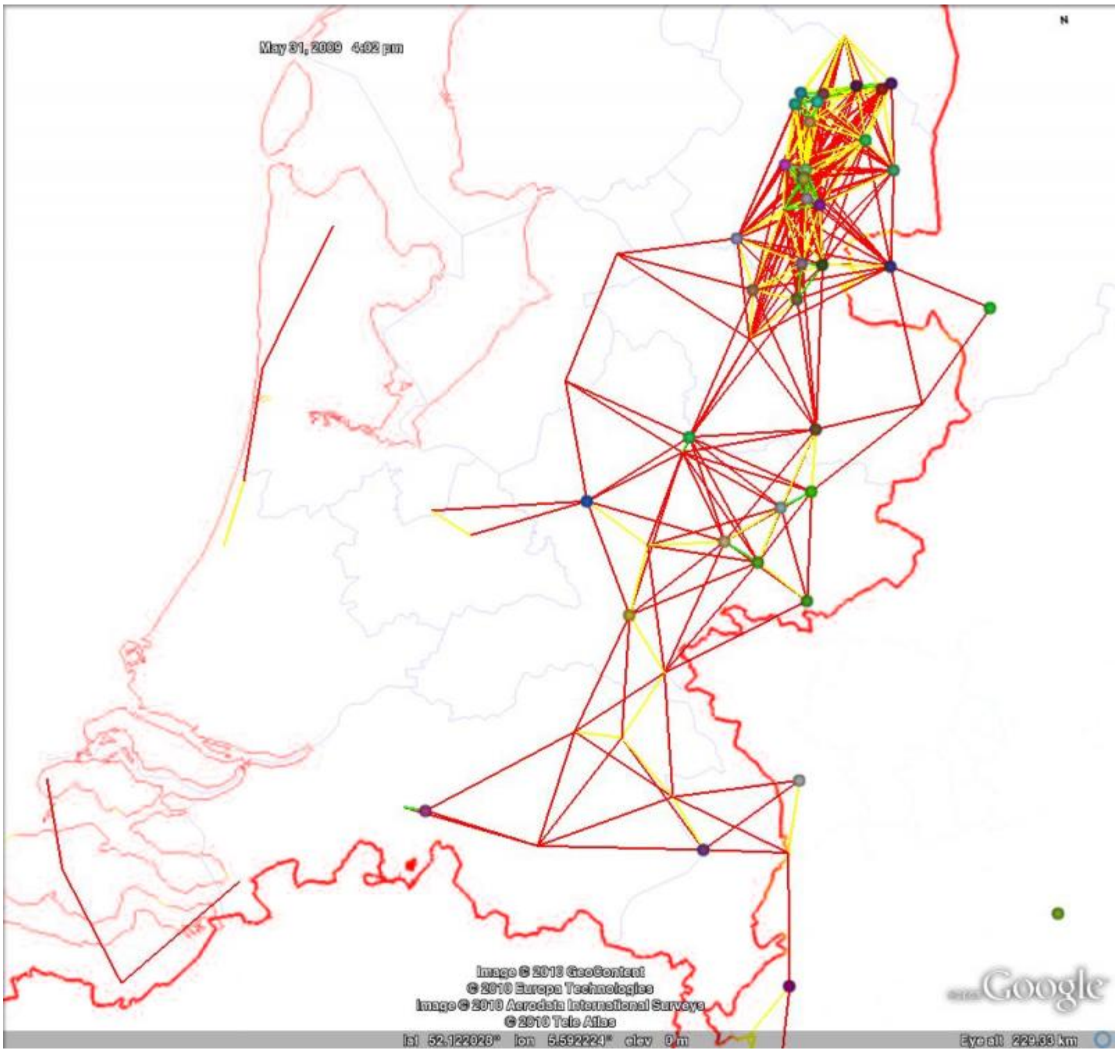

Figure 3.4: *Flight analysis visualized in Google Earth. The 31th of May 2009, a day of the Dutch National Championships with very good cross-country flying conditions, is shown. Communication is possible from the north to the south of The Netherlands. Image edited with Pixlr [18].*

3.2.2 Duration of a connection

We analyze the duration of a connection between two nodes to gain insight on how long connections typically last between nodes in various scenarios. The link duration has consequences on what kind of networking approach should be applied. If connections last very long, discovering topological information can be rewarding. If connections only last very short, topological information is invalidated very quickly and it makes little sense to exchange this information, as continously keeping this information fresh

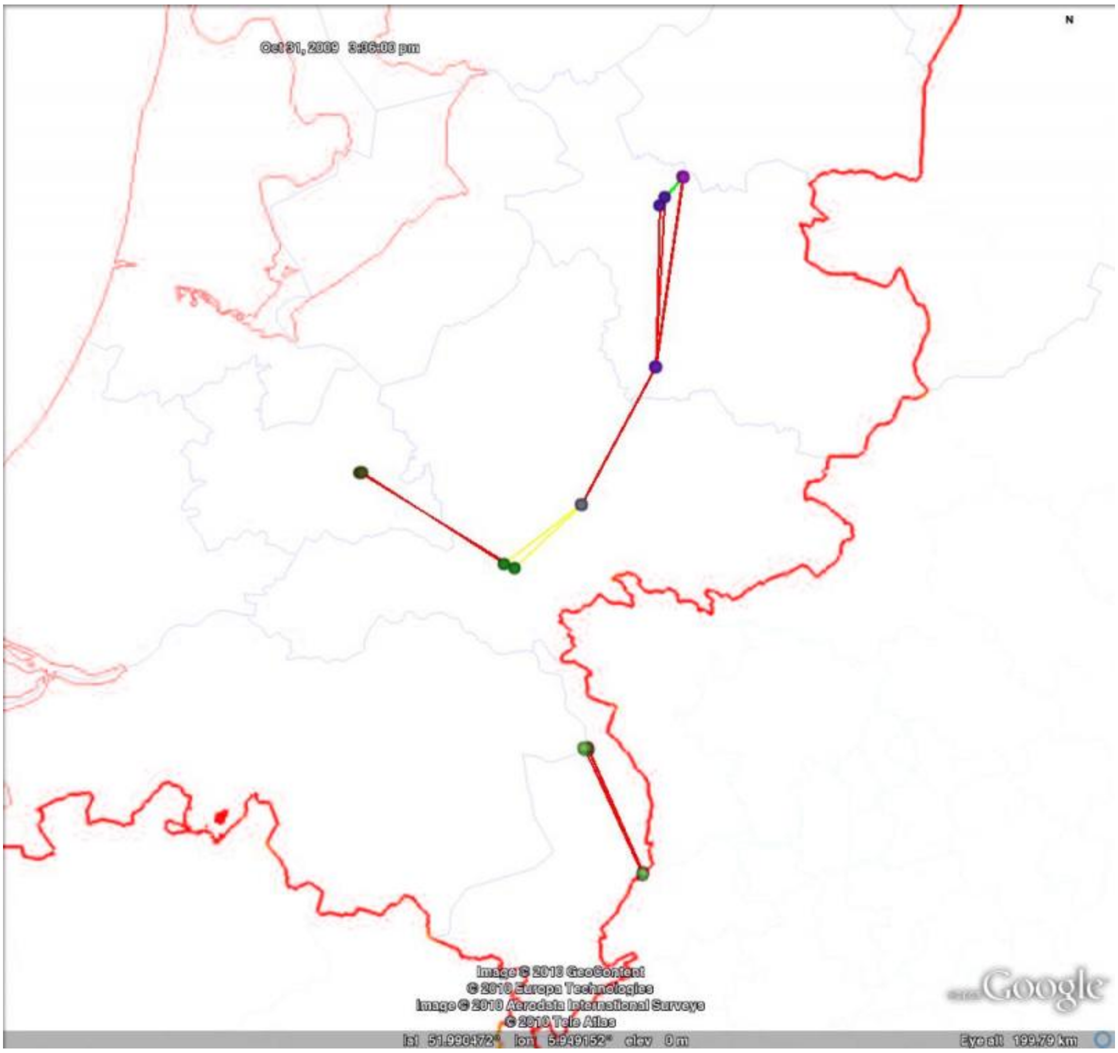

Figure 3.5: *Flight analysis visualized in Google Earth. The 31th of October, a day with not so good cross-country conditions, is shown. Airfields are not turned on. The network during this day is almost always partitioned. Image edited with Pixlr [18].*

generates high network traffic.

Figure 3.6 shows the link duration on a day with good cross-country conditions, and the visualization of this day can be seen in Figure 3.4 . Figure 3.6 shows many connections last up to 5000 seconds, or 80 minutes. However, some connections last much longer. These are connections between nodes which are flying next to each other for long periods of time.

Figure 3.7 shows the link duration on a day with poor cross-country conditions, and the visualization of this day can be seen in Figure 3.5 . Figure 3.7 shows an even

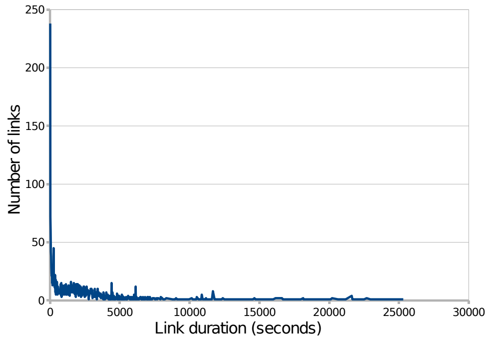

Figure 3.6: *Link duration during a day with good cross-country conditions.*

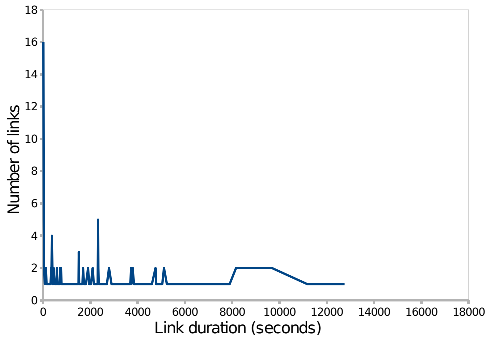

Figure 3.7: *Link duration during a day with poor cross-country conditions.*

distribution of connection duration.

3.2.3 Distance between two connected nodes

We analyze the average distance between two connected nodes to gain insight on how far apart nodes are in general when connected in different scenarios. We also analyze the duration of a connection to see if a relation exists between the distance and the duration of a connection.

Figure 3.8 shows the link distance between two connected nodes on a day with good cross-country conditions, and the visualization of this day can be seen in Figure 3.4 . Figure 3.10 shows the relation between link distance and link duration for this day. Many connections are made over a large distance. However, these connections have a short duration. Thus, if we want to take full advantage of all connections possible the setup phase of a connection should be as short as possible.

Figure 3.9 shows the link distance between two connected nodes on a day with poor cross-country conditions, and the visualization of this day can be seen in Figure 3.4 . Figure 3.11 shows the relation between link distance and link duration for this day. It can be seen that the same pattern emerges as during the day with good cross-country conditions.

3.2.4 The dot-product of between the average speeds of the connected nodes

We look at the dot-product between the average speeds of the connected nodes to see if a relation between the velocity and heading of the two nodes and the time the connection between the nodes exists. The vector is an indication of how well the velocities and heading of both nodes mach. The dot-product is calculated as follows:

$$a \cdot b = |a|\,|b|\,\cos(\alpha)$$

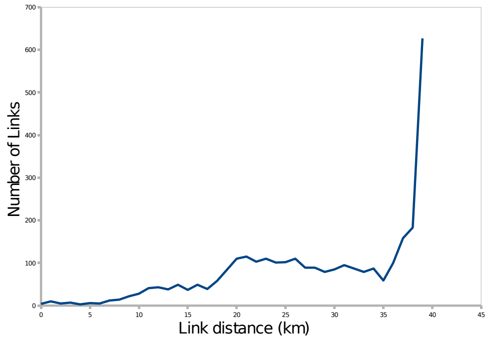

Figure 3.8: *Link distance during a day with good cross-country conditions.*

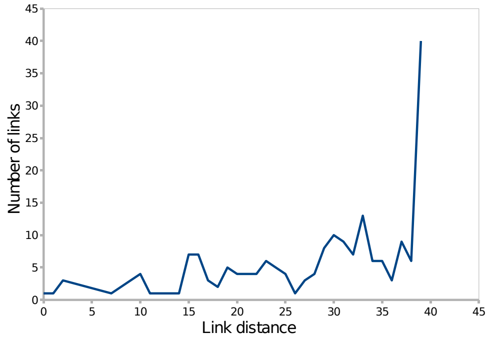

Figure 3.9: *Link distance during a day with poor cross-country conditions.*

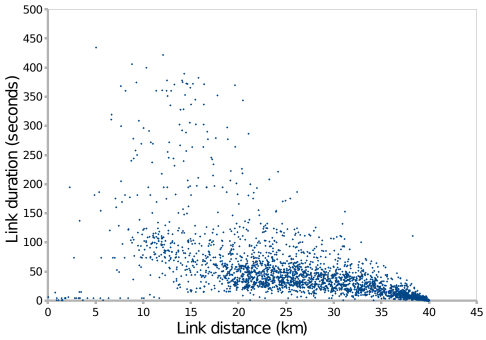

Figure 3.10: *Link distance versus link duration during a day with good cross-country conditions*

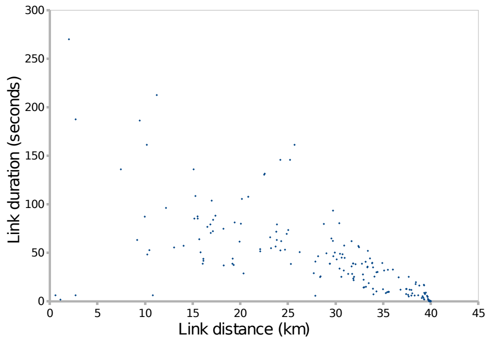

Figure 3.11: *Link distance during a day with poor cross-country conditions*

Here a and b are the velocity vectors of the two nodes involved in the connection, and α is the angle between these two velocity vectors. If the two nodes are moving with the same speed in the same direction, the dot-product will roughly be the square of the velocity of the two nodes, since the cosine in the dot-product expression will roughly be 1.

In Figure 3.12 , the dot-product can be seen for a day with good cross-country conditions. We see that connection duration decreases as the dot-product becomes negative, which indicates nodes moving in the opposite direction. When the nodes travel in the same direction, connection times become longer, with a peak around a dot-product of 12000.

In Figure 3.13 , the dot-product can be seen for a day with poor cross-country conditions. Although data points are sparse, the same patterns as in Figure 3.12 can be observed.

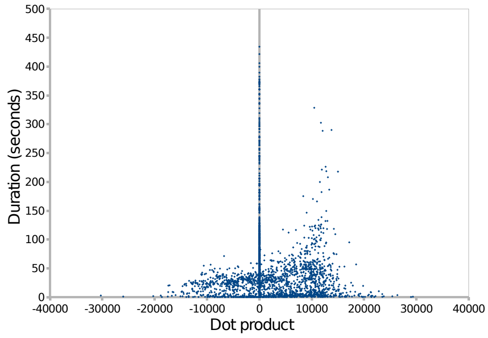

Figure 3.12: *Dot-product during a day with good cross-country conditions.*

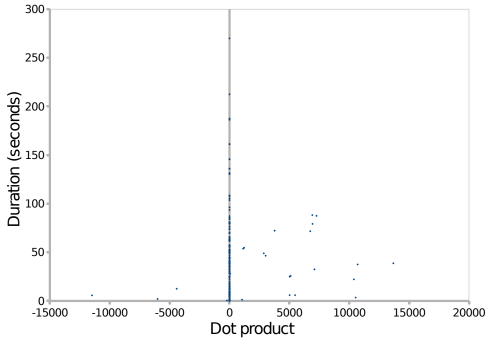

Figure 3.13: *Dot-product during a day with poor cross-country conditions.*

3.2.5 Maximum number of nodes within two hops

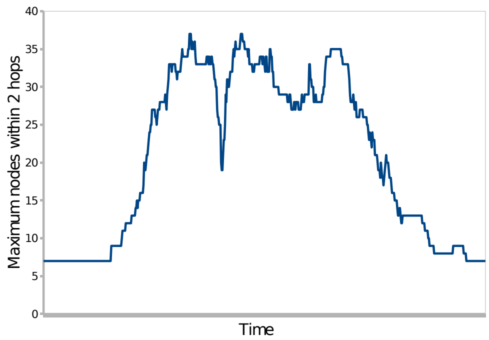

Figure 3.14: *Fanout during a day with good cross-country conditions.*

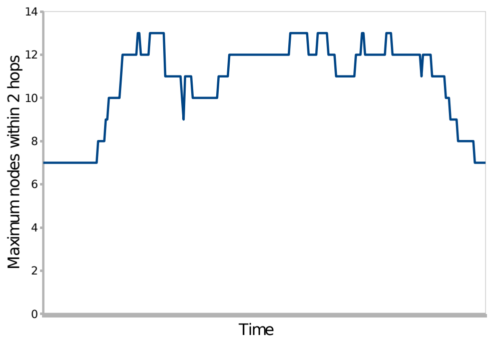

Figure 3.15: *Fanout during a day with poor cross-country conditions.*

We look at the maximum number of nodes within two hops to gain insight on how severe the hidden terminal problem is in various situations. If two nodes are separated by two hops, this means that they share a common neighouring node. This is the scenario in which the hidden terminal problem is present. Thus we take the maximum number of nodes within two hops as a measure of how severe the hidden terminal problem is in the network at that time.

In Figure 3.14 , the maximum number of nodes within two hops can be seen during a day with strong conditions. The visualization of this day can be seen in Figure 3.4 . During this day the Dutch National Championships was held, causing many gliders to fly close to each other for a long period of time. This results in a high overall maximum number of nodes within two hops, which is over 35 nodes. A decrease in the maximum number of nodes within two hops can be seen during the day, which is a result of two large groups of gliders taking different routes. These two groups later merge, causing the maximum number of nodes within two hops to increase again.

In Figure 3.15, the maximum number of nodes within two hops can be seen during a day with poor cross-country conditions. The visualization of this day can be seen in Figure 3.5 . This results in a much less overall maximum number of nodes within two hops.

3.2.6 Conclusion

Although connections with a long duration exist, most connections exist over a long distance and have a short duration. During a day with good cross-country conditions the hidden terminal problem is a serious issue, since many nodes are able to interfere transmissions.

3.3 Traffic Analysis

For each type of gliding described in Chapter 1, we define a set of scenarios. These scenarios are listed here, along with their respective requirements. We view these scenarios to define the requirements they impose on the network. The reason for doing so is to gain insight on the amount of data that is generated by each scenario. We show the requirements in Figure 3.19. Also, the values shown here are also used during evaluation in Chapter 6.

3.3.1 Dissemination of location information upon event detection

In case a glider crashes, it is very usefull for rescuers to know the last known location of the glider. It is very important that this information is received by the rescuers. Because the location of the rescuer is not known in advance, it is important that this information is widely spread through the network. To cope with lack of connectivity in the network, the information should be stored at as many nodes in the network as possible.

In practice, however, it appears to be infeasible to detect an event that might lead to a crash way in advance. According to correspondence with the dutch gliding safety board [19], most crashes happen during take-off and landing. Since this is within sight of individuals who are able to rescue the pilot, the location is known and does not need to be disseminated. Another category where crashes frequently happen are mountain-sides, which cannot be detected beforehand. It also happens that pilots suffer from medical conditions, such as a heart attack during the flight. This usually results in the glider crashing into the ground within a few seconds. This is hard to detect since the common resolution one gets from a GPS receiver is one GPS position per second. The time-resolution combined with the occasional inaccuracy of a GPS receiver make it very hard to detect a crash before the airplane hits the ground. We assume that all equipment is destroyed when the glider hits the ground. Since it is not feasible to detect a crash before it actually happens, no information can be disseminated. The last location of a crashed glider should otherwise be disseminated.

3.3.2 Dissemination of location information during cross-country flights

Sending and receiving location information during cross-country flights have two purposes. The main focus is visualization of the glider's position on a map application like Google Earth. Existing solutions exist for this purpose. For instance, Irridium sends location information once per minute to a satellite. The information arrives after roughly 3 minutes at a server, which will handle the visualization and optional broadcasting on the internet. Sending positional information is feasible. Assuming a maximum speed of 300 km/h, a glider can move a maximum of 5 kilometers in one minute. Thus, when the glider crashes, the last location is known with a minimum accuracy of 10 kilometers. The mentioned delay of three minutes has proved not to be distrubing for spectators watching. Therefore, this is taken as the maximum delay. Most logfiles have a log entry every 10 seconds, which has also proven to be enough. This is also used. Thus, every 10 seconds a position is recorded, and every 60 seconds six positions are transmitted. The transmission should arrive within three minutes.

Spatial resolution is also important when representation of the glider's position is concerned. For the latitude, longtitude and altitude 32 bits of accuracy are used. Given 32 bits of accuracy, spatial resolution can be calculated. The equatorial and meridional circumference of the earth is roughly 40.000 kilometers, or 40.000.000 meters. Thus the minimum resolution can be calculated as:

$$\frac{40000000}{2^{32}} = \frac{40000000}{4294967296} \approx 0.01 meters$$

The maximum altitude reached so far by a glider is 15460 meters [20]. Using 32 bits to represent the altitude of the glider, and rounding the maximum altitude to 16000 meters, the resulting resolution in height becomes:

$$\frac{16000000}{2^{32}} = \frac{16000000}{4294967296} \approx 0.003 meters$$

With a 32-bit timestamp, this results in 128 bits per position, or 16 bytes. The representation of positional information can be seen in Figure 3.16 . Six positions take 96 bytes, which fits inside a packet.

Bits	Description
32	Timestamp
32	Latitude
32	Longtitude
32	Altitude
128	*Total*

Figure 3.16: *The representation of positional information.*

3.3.3 Dissemination of weather information

A wide variety of weather information is available. Images from geo-stationary satellites, orbiting satellites and images from precitipation radar images are just a few examples. All images differ in resolution and the area they cover. For example, precitipation radar images can reach a resolutions of 1 km^2 per pixel and orbiting satellites can reach resolutions up to 250 meter^2 per pixel. Time resolution also varies, from one or two passes a day at a certain location for orbiting satellites to one image every 15 minutes from precitipation radars.

Because the information varies a lot, and remote sensing technology might advance even further in the future, we do not choose a fixed scheme for representing remotely sensed data and choose a model instead. This model assumes that:

- the area covered by one packet with data has the shape of a rectangle, where the edges are parallels and lines of latitude.
- the measured quantity can be represented by a 8-bit greyscale bitmap.

Since weather information is not vital, it should have a low priority, as its receipt is not important compared to the other types of traffic in the network. How quickly the information should arrive at interested nodes depends on what that content is. Satellite images, which usually describe large-scale developments, can be sent relatively slowly. Precitipation can vary a lot within 5 minutes, which implies it should be sent quicker. We choose the deadline for arrival as the time the next image will be ready. For a precitipation image this is 5 minutes, which for a satellite image can be up to one hour.

In this thesis will use a precitipationradar image as an example. This image has a resolution of 1 km^2 and a coverage of 240 by 330 kilometers. Every 15 minutes, a new image will be available. One byte in this image describes the precitipation at 1 km^2, a complete image is described as $240 \cdot 330 = 79200$ bytes. In one packet, an area of 15 by 15 kilometers can be described. Thus, the total image is transmitted in $(240/15) \cdot (330/15) = 352$ packets.

3.3.4 Dissemination of measurement information

To measure turbulence, like in the mountain wave project, a five-hole probe can be used. A five-hole probe, as can be seen in Figure 3.17 , measures the pressure at the front and at the four holes around the center: Yaw-1, Yaw-2, Pitch-1 and Pitch-2. The hole in the center measures the dynamic pressure generated by the speed of the airplane. Turbulence manifests itself as local changes in direction of the airflow. This will create a pressure difference, for example between Pitch-1 and Pitch-2 if the turbulence locally flows up. By measuring the pressure difference between Yaw-1 and Yaw2, and measuring the pressure difference between Pitch-1 and Pitch-2, the direction of the local airflow can be determined. The result is a 3D vector.

The phenomenon which is being measured has to be considered when deriving the interval at which a glider performing measurements will have to measure. We will assume

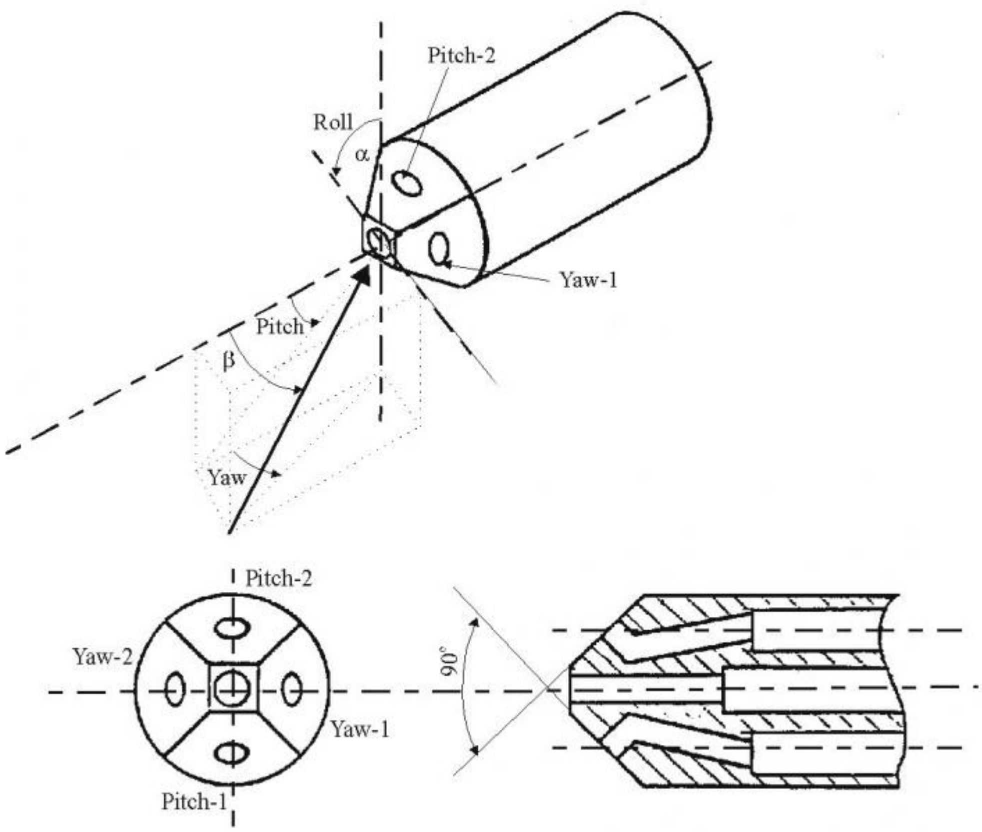

Figure 3.17: *A five-hole probe. The hole in the center gives total pressure. Between holes at the sides a pressure difference exists, through which the angle of the airflow can be determined. Copyright: University of Cambridge [21].*

the glider is measuring on lee waves, as described in Chapter 1 . Lee waves are a large scale phenomenon, as they are several kilometers in size [7, 6, 8]. Also, the rotorwinds which can be measured by gliders are several kilometers in diameter [7]. Another factor is the maximum speed of the glider. The maximum speed is again estimated at 300 kilometers per hour. This means that the glider travels a maximum of approximately 83 meters per second. We choose the interval between two measurements to be 3 seconds. This means that the maximum distance between two measurements is 249 meters. With rotor turbulence of 6 kilometers in diameter, this would mean 24 measurements would take place if the glider were to cross the rotor turbulence at the maximum speed of 300 kilometers per hour.

We estimate the precision of the measurements to be in the order of 0.1 degrees resolution. In order to represent angles from 0 to 360 degrees with a 0.1 accuracy, twelve bits are needed. Thus, for the two angles, α and β, twelve bits are used for each. The wind speed is estimated to range from 0 to 50 kilometers per hour. We choose 8 bits to represent the length of the vector, which gives the wind speed a resolution of roughly 0.2 kilometers per hour. In total 32 bits are needed to describe a measurement from the five-hole probe: 2 times 12 bits for angles α and β and 8 bits for the length of the vector.

Beside the measurement itself, positional information is also inserted in the packet. This positional information is sent in the same format as described earlier in thie section for location information. Thus, 16 plus 4 bytes totals 20 bytes per measurement are used. Sending a packet very 30 seconds, and each packet contains 10 measurements, results in each packet containing 200 bytes.

Bits	Description
32	Timestamp
32	Latitude
32	Longtitude
32	Altitude
12	Angle α
12	Angle β
8	Vector length
160	*Total*

Figure 3.18: *The representation of measurement information.*

Traffic type	Frequency	Deadline	Packetsize
Positional information	Every 60 seconds	180 seconds	96 bytes
Weather information	Every 300 seconds	900 seconds	225 bytes
Measurement information	Every 30 seconds	90 seconds	200 bytes

Figure 3.19: *The requirements summed up for each type of traffic.*

4 Inspiring existing protocols

> Never underestimate the bandwidth of a truck full of tapes hurling down the highway.
>
> Andrew S. Tanenbaum

Chapter 3 presented circumstances where connectivity is good enough not to have disconnected segments in network coverage. For these circumstances we look at existing approaches for unicast routing and multicast routing in mesh networks. For circumstances, in which connectivity is not good enough and the network sometimes has disconnected segments of network coverage, we look at existing dissemination approaches.

Since there are different quality of service requirements for different types of traffic, we look at existing approaches for quality of service in wireless networks.

Because the fan-out in Chapter 3 is sometimes high, we look at approaches to minimize the hidden-terminal problem.

4.1 Data dissemination

Dissemination can be categorized into reactive approaches - like the typical client/server model in which consumers request for specific information - and proactive approaches - like broadcasting, publish/subscribe mechanisms and multicast - in which information is proactively disseminated whenever such information is available in the network.

Some existing paradigms make use of infrastructure or build an overlay network on top of existing unicast routing [22].

Opportunistic dissemination approaches [23, 24, 25, 26, 27] transport the information on the network, without the information specifically being requested. When information is required by a node, this information is already available at that node. Thus, the node does not need to send requests to the network, instead the network delivers information to the node.

Dissemination protocols for wireless mesh networks exist in many forms, where often geospatial data is disseminated [26, 27]. Dissemination approaches can be categorized in structured approaches with a specific network architecture [22] sometimes with a publish/subscribe model, sometimes based on metadata [25]. Other approaches perform opportunistic dissemination [23, 24].

4.1.1 Autonomous Gossiping

Autonomous Gossiping [23] is an example of an opportunistic dissemination protocol. It is a self-organizing epidemic algorithm for selective information dissemination and attempts to exploit mobility to achieve randomness. The protocol utilizes a ecological paradigm to achieve selectivity. The protocol does not exchange topology information and is a completely stateless protocol.

Autonomous Gossiping attempts to utilize mobility instead of approaching mobility as a source of difficulties. The opportunity to come in contact with new nodes are a source for disseminating messages to these nodes. This is similar to how in real life people gossip information.

Selectivity is reached through the epidemic paradigm, which is used in Autonomous Gossiping. Each unique data item is considered as an epidemic, and each epidemic attempts to selectively infect network members based on their vulnerability (as stored in a profile) to the epidemic.

Nodes advertize their profile such that data items resident at neighboring nodes can see this profile. The data items can then decide to migrate to that node. Through the rewarding of good decisions - such as migration to nodes which have a well matching profile - and the punishing of bad decisions - such as migration to nodes which have a poorly matching profile, data items migrating towards areas where they are not desired will eventually extinct in the event of storage shortage. Thus, the protocol utilizes the *extinction of the unfit* paradigm.

4.2 Routing algorithms in mesh networks

Routing in mesh networks can be categorized into pro-active protocols and reactive protocols. Pro-active protocols, like Destination-Sequenced Distance Vector (DSDV) [28] and Optimized Link Source Routing (OLSR) [29], exchange routing information continuously. Each node in the network broadcasts its current routing table to its neighbours, who do the same. When a node wants to send a packet, it can simply look up the destination of the packet up in its routing table to determine the next hop for the packet. Thus, the time it takes for a node to send the packet is relatively short. However, since the nodes are continuously exchanging routing information, traffic is sent constantly between nodes. Even when no data packet is sent over the network, the nodes are busy exchanging routing information. Thus, control overhead is high in this situation. When connectivity in the network is short, there is also the risk that nodes route information based on outdated routing information.

In reactive protocols, like AODV [30, 31, 32, 33], Dynamic Source Routing (DSR) [34] and ODMRP [35, 33, 36], the nodes in a network do not continuously exchange routing information, but only when a node wants to send a packet. This node then initiates the process of path discovery. During path discovery, the node discovers only the current route to the destination it wants to send the packet. All other available routes in the network should be discovered separately. The latency before sending a packet is higher, since the node originating the packet has to discover the path to the destination of the packet first.

4.2.1 AODV

AODV [30, 31, 32, 33] is an example of a reactive routing protocol. AODV uses destination sequence numbers to find the most recent route to a destination. It does so

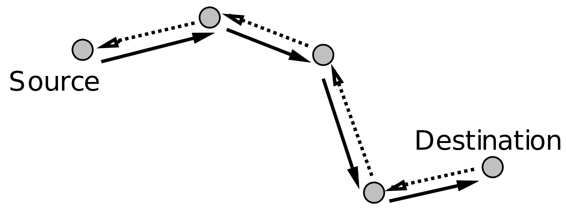

Figure 4.1: *AODV route discovery. Solid arrows represent* RouteRequest *packets, dashed arrows represent* RouteReply *packets. Image adapted from Perkins [30].*

on-demand, that is only when a source node has a packet to send to a destination node for which no known route is known at the source node.

When a node wants to send data packet, to a destination to which it does not know a route, the source node broadcasts a *RouteRequest* packet. This can be seen in Figure 4.1 . If in response to this *RouteRequest* packet a route is returned, AODV uses this route to send the data packet. When multiple routes are returned, AODV uses a *destination sequence number* to select the most recent route.

A *RouteRequest* packet carries a *source identifier*, a *destination identifier*, a *source sequence number*, a *destination sequence number*, a *broadcast identifier* and a *time to live*. When a node receives a *RouteRequest* packet, it checks if it has a route entry stored for the specified *destination identifier*. If the node has indeed a route entry for the specified *destination identifier*, the node checks if the information is valid by checking if the stored *destination sequence number* is equal to or greater than the *destination sequence number* in the received *RouteRequest*. If this is the case, the node will prepare a *RouteReply* packet containing information about the route. If the node has no route stored, it will itself prepare a *RouteRequest* and broadcast this packet.

When a node receives a *RouteRequest*, packet it will also store a route entry for the source of the *RouteRequest*. This route can then be used when the destination node of the data packet for which the route is discovered wants to send a packet back to the source of the data packet.

AODV prevents nodes from receiving *RouteRequest* packets multiple times using the *broadcast identifier* to identify packets. If a node receives a *RouteRequest* packet, the node will store the *broadcast identifier* of that *RouteRequest*. When the node then receives a *RouterRequest* packet with the same *broadcast identifier*, it will discard it.

Timers are used to invalidate route entries stored in memory. This triggers a node to find the current route when it receives a *RouteRequest* packet for that route.

4.2.2 ODMRP

ODMRP [35, 33, 36] is a reactive multicast routing protocol. It can be compared to AODV in many ways [33]. ODMRP uses packets similar in nature to the *RouteRequest* and *RouteReply* packets from AODV to discover routes. In ODMRP, the *RouteRequest* and *RouteReply* packets are called a *JoinRequest* packet and *JoinTable* respectively.

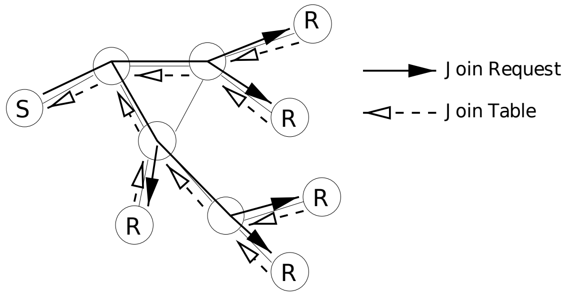

Figure 4.2: *ODMRP route discovery. Node* S *is the sender of multicast data. All nodes marked* R *are receivers. [35].*

When an ODMRP node has multicast data to send, it will periodically broadcast a *JoinRequest* packet. This packet will be flooded over the network, causing all ODMRP nodes to receive the *JoinTable* packet. The packet will traverse the nodes in the network similarly to how the *RouteRequest* packet in AODV traverses the nodes in an AODV network. When the *JoinRequest* arrives at a multicast receiver for the multicast data associated with the *JoinRequest*, the node will respond by sending a *JoinTable* packet to the node it received the *JoinRequest* packet from. This node will conclude it has to forward the multicast data in question and will also send a *JoinTable* packet to the node it received the *JoinRequest* packet from. Thus *JoinRequest* packets propagate from multicast receivers to the source of the *JoinTable* packet. This builds a multicast graph, which can be seen in Figure 4.2 . Nodes in the multicast graph between the multicast sender and multicast receivers are called the forwarding group. The forwarding group supports the shortest paths between multicast receivers and the multicast sender. When multiple shortest paths between the multicast sender and a multicast receiver exist, both paths are used. Unlike AODV, ODMRP builds redundant paths.

4.3 Medium Reservation in wireless networks

Medium reservation protocols, i.e. protocols designed to solve the hidden terminal problem, can be categorized into two groups: proactive and reactive.

Proactive protcols exist in the form of Time Division Multiple Access (TDMA) protcols [37, 38, 39], which devide time into fixed-size frames with fixed-size timeslots. A timeslot can be reserved by a node, after which that node can send data in the reserved timeslot. These timeslots can be seen in Figure 4.3 . The reserved timeslots have to be communicated to nearby nodes and to nodes 2 hops away to solve the hidden-terminal problem. The communication to nodes 2 hops away takes time, and if in the meanwhile conditions change invalid information is communicated to the nodes 2 hops away. Overhead is not very high, since a node can communicate numerous reservations in one packet. The node can listen for one framelength and communicate all reservations it has heard about during his timeslot.

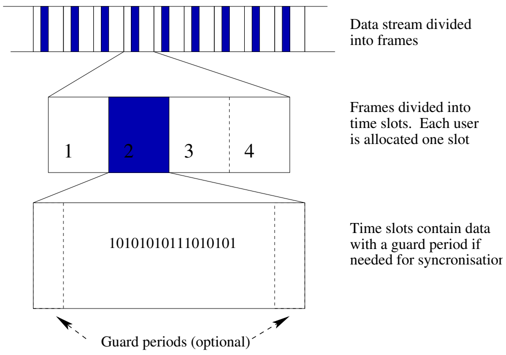

Figure 4.3: *The structure of a TDMA frame.*

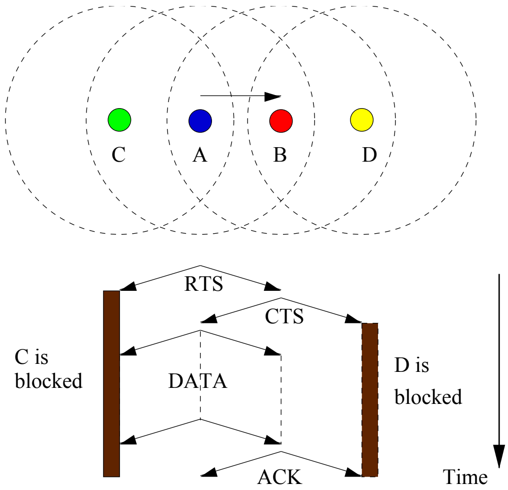

Figure 4.4: *RTS/CTS medium reservation*

Reactive protocols [40, 41] only reserve the medium when a node wants to send a packet. An example of a reactive protocol is RTS/CTS, which is shown in Figure 4.4 . The reservation will be communicated directly to nodes 2 hops away, which reduces the chance that the information is outdated by the time it arrives at nodes 2 hops away from the sender. The direct communication however means a node will not listen for several reservations before it will communicate the reservation it heard. Thus, for every reservation one packet is sent per node. For TDMA protocols this can be one packet per frame: the more timeslots fit in a frame, the more efficient it gets.

5 Design of an Opportunistic Quality-aware Mesh protocol

Als het niet kan zoals het moet,
dan moet het maar zoals het kan

My father

This chapter will focus on the design of an opportunistic quality-aware mesh protocol. The conclusions in Chapter 3 and the protocols evaluated in Chapter 4 are translated in a new dissemination protocol, called *Cumulus Humilis*.

5.1 Cumulus Humilis

Bits	Description
32	Unique packet ID
8	Packet utilization factor
8	Packet rebroadcast period
16	Packet age in seconds
16	Packet deadline in seconds
80	*Total*

Figure 5.1: *The layout of a data dissemination packet.*

Our Cumulus Humilis protocol aims to disseminate data among gliders in both sparse and dense glider networks. More specifically, it aims to achieve a high delivery ratio with a low propagation delay while honoring relative packet priorities. For this purpose, we take the following approaches:

- We assign each type of traffic a *utility factor*, allowing nodes to prioritize packets with a high *utility factor* over packets with a low *utility factor*.
- Nodes apply a *store-and-forward* communication model to allow nodes to physically carry packets to new locations whenever a multi-hop connection is not available between gliders.
- We assign each packet an *age* and a *deadline*. Whenever a package's *age* has reached the *deadline*, the packet will be removed.

The header layout can be seen in Figure 5.1 .

5.1.1 Requirements and Assumptions

For proper operation of the protocol we assume:

- a mobile computer - for example an iPaq - is available in the cockpit,
- a GPS receiver is connected to the mobile computer,
- enough batteries are present to power all electronic equipment for the entire duration of all flights during the day,
- the transceivers present at the gliders have a maximum transmission range of 40 km,
- signal strength decays in the order of the distance to the forth power (d^4).

Cumulus Humilis performs opportunistic dissemination, without knowing the network topoligy. It does not attempt to discover routes like AODV or create multicast graphs like ODMRP. No information about network topology is stored or requested from nearby nodes. *Cumulus Humilis* will simply broadcast any information available at a node. When neighbouring nodes receive this information, they will in turn store the information and broadcast it. Effectively a broadcast storm is generated.

To prevent nodes from receiving duplicate packets, every packet carries a *unique packet identifier*. When a node receives a packet, it stores its *unique packet identifier* of the packet. When the same node again receives a packet with the same *unique packet identifier*, it will discard it.

To limit the time a packet exists in the network, each packet carries a *packet age* and a *packet deadline*. The *packet age* is incremented each second. When the *packet age* reaches the value of the *packet deadline*, the packet is deleted from the storage of the node. The *packet deadline* describes the time the data in the packet remains valid.

5.2 Desynchronizing hosts

When two nodes receive the same data packet, and will store this data packet for later broadcasting, those nodes can become synchronized. If the nodes are synchronized they attempt to send the same data packet at the same time. This can result in two colliding transmissions, such that none of the two transmissions will be readable by other nodes. To prevent the two nodes from becomming synchronized, upon startup each node chooses a *random delay time*. Each transmission the node schedules will be delayed with this time, not exceeding the *rebroadcast period* of the packet. Thus, different nodes will broadcast the same packet at different times, due to their different *random delay time*.

To further desynchronize hosts, each node will pick a random moment, ranging from 0 to 1000 milliseconds, within a second to broadcast packets.

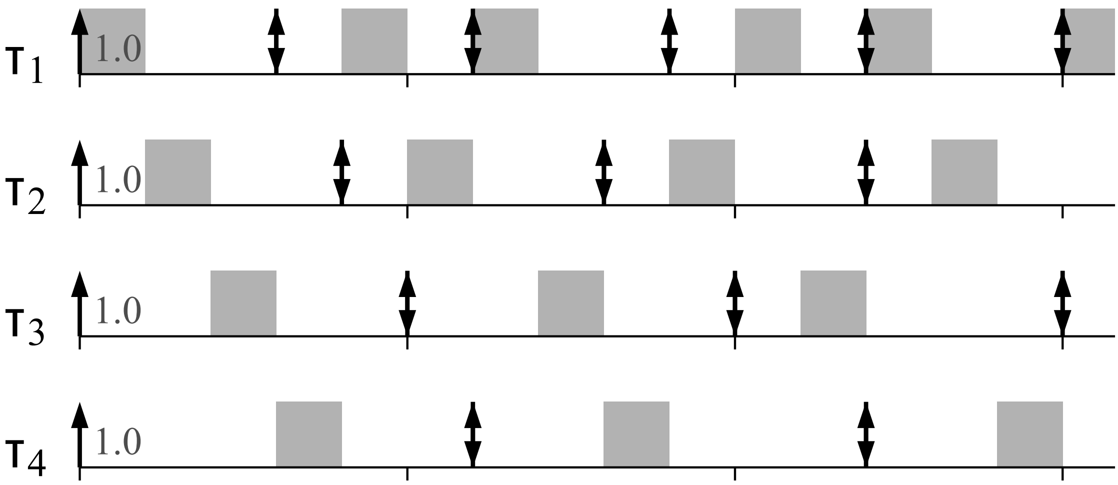

Figure 5.2: *EDF scheduling visualized for 4 packets. Arrows pointing upwards indicate release times, arrows pointing downwards indicate deadlines. Grey blocks indicate the transmission of the packet.*

5.3 Scheduling transmissions

In a mobile environment, no assumptions can be made on when a direct route will be available to the destination of a packet, or when a node which is able to physically transport the packet towards a suitable location will be around. Thus, the best effort a node can do is spreading the transmissions of a packet as evenly over time as possible. This increases the chance that during the time it periodically broadcasts the packet, a suitable node will be around. To maximize the chance different broadcasts of a packet from one node are received by as many nodes as possible, we spread the different broadcasts evenly over time. However, when multiple packets have to be broadcasted at the same time by a node, scheduling is needed to optimally spread all different packet broadcasts over time. We utilize EDF [42] for this purpose, and check the feasibility of the set of different broadcasts. We make sure that the 10% average duty cycle of the X-Bee PRO 868 transceivers is honored. In case the set is not feasible, priority is given to the broadcasts with the highest priority. An example of the packet scheduling as performed by EDF can be seen in Figure 5.2 .

6 Performance evaluation

> If debugging is the process of removing bugs, then programming must be the process of putting them in.
>
> — Unknown

In this chapter we evaluate the new dissemination protocol proposed in Chapter 5, using a discrete-time simulator. Since no existing solution exists for the communication between gliders, we do not compare the protocol with another solution. Instead we verify if the protocol meets the requirements set in Chapter 1.

6.1 Evaluation technique

To evaluate *Cumulus Humilis*, we write a discrete-time simulator in the C++ programming language. The simulator uses the same GPS logfiles used in Chapter 3 to model glider movement. The locations of gliding airfields are also used.

The simulator uses discrete time steps of 1 millisecond. Every millisecond we determine the movement of every node, and the protocol behaviour of every node.

The radio model used in the simulator assumes that signal strength decays over distance following d^4. A transmission is received by a node, when during the entire transmission the Signal-Noise Ratio (SNR) is higher than 10. For all other transmissions, the noise is calculated again through d^4. If there exists a transmission which has a ratio of 10 in signal strength over all other transmissions, it is assumed this transmission is received by the node.

The length of a transmission in milliseconds is calculated from the number of bytes, using the formula $\lfloor 0.4202 \cdot \text{bytes} + 23.531 \rfloor$. This follows the results of the measurements performed on the X-Bee PRO 868 transceivers described in Chapter 3.

6.1.1 The simulated MAC layer state machine

To simulate the MAC layer of the X-Bee PRO 868 transceivers, a state machine is used which is shown in Figure 6.1. A node starts at the *Listen* state, where it listens to the medium for incoming transmissions. Once an incoming transmission is heard and it has been determined that this transmission just started, the state machine moves to the *Receive* state and stores which transmission was heard. When the transmission is finished, it is passed onto the protocol running on top of the MAC layer and the state machine returns to the *Listen* state. If during the transmission, the SNR drops below the value of 10 the state machine also returns to the *Listen* state, but does not pass the transmission to the protocol running on top of the MAC layer.

When a transmission is ready to be sent, the state machine first checks what the duty cycle is. If the duty cycle indicates the node has exceded the 10% duty cycle of

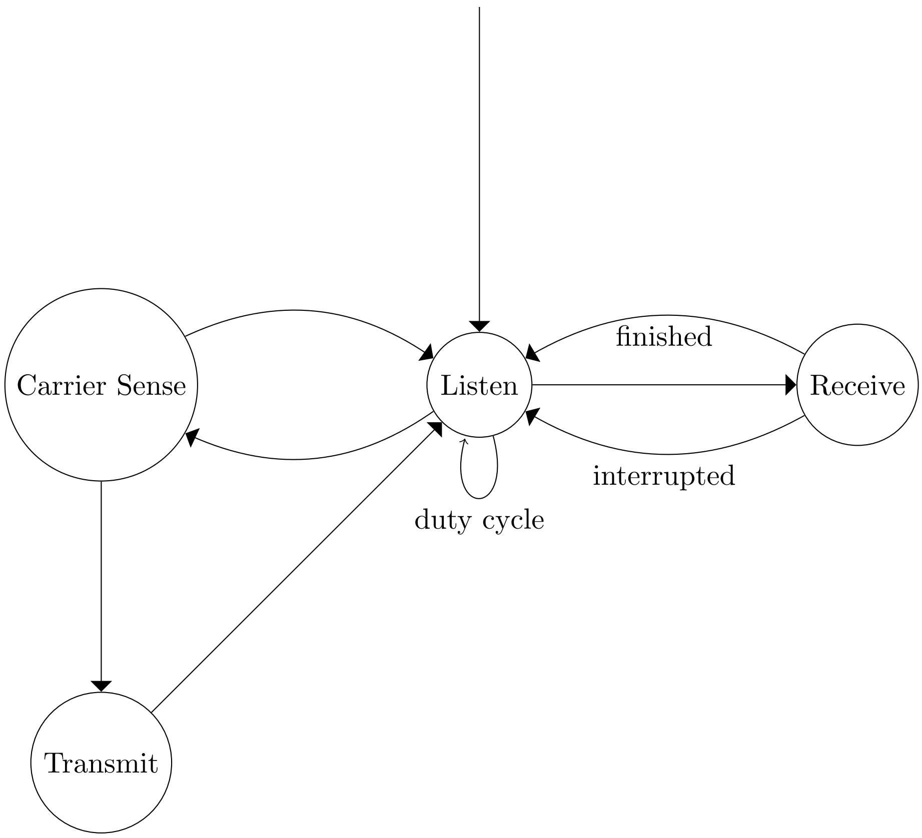

Figure 6.1: *The state machine used to simulate CSMA.*

the X-Bee PRO 868 transceivers, the node remains in the *Receive* state. If it has not exceded the 10% duty cycle, the state machine will switch to the *Carrier Sense* state. In the *Carrier Sense* state the medium is sensed for other transmissions. In the case of no transmissions, the state machine switches towards the *Transmit* state. In the case transmissions are sensed, the state machine returns to the *Listen* state.

6.1.2 Metrics used in simulation

We use the following metrics for our evaluation.

- *Successfull Transmission.* We define a transmission to be succesfull when all the bytes in the transmission have been sent by the transceiver at the sending node.

- *Successfull Reception.* We define a reception of a packet to be succesfull when the receiving node has succesfully received all the bytes of a transmission and has decoded the transmission.

- *Number of Succesfull Receivers of a packet.* We define the number of receivers of transmissions to be number of *Reception*s devided by the number of *Transmission*s.

- *Latency.* We define the latency of a packet as the age of the packet when it is received for the first time by a node.

- *Packetloss.* We define packetloss as the ratio of all packets not succesfully received devided by all packets succesfully sent. Since all traffic in the network is broadcast traffic, one packet transmission has multiple receivers. Thus, the amount of packets not succesfully received can be larger than the amount of packets successfully sent.

- *Delivery ratio.* We define the delivery ratio of an information type as the number of nodes which on average receive packets for the type of information, devided by the number of nodes. Thus, it illustrates the probability of any node receiving a packet from this type of information.

6.2 Results

During simulation we look at two specific scenarios:

- *Scenario 1*: A scenario with very good connectivity and airfields turned on, to evaluate the performance of the protocol in circumstances in which a lot of traffic is sent.

- *Scenario 2*: A scenario with very bad connectivity and airfields turned off, to evaluate the performance of the protocol in circumstances in which the opportunistic dissemination features of the protocol have to be used to deliver packets.

First we evaluate a scenario with very good connectivity, for which we use GPS logfiles from a day of the Dutch National Championships. To increase connectivity we also enable airfields. One of the airfields disseminates weather information. One of the gliders is performing a measurement flight during this day and disseminates these data. Figure 6.2, Figure 6.3, and Figure 6.4 show the latency of packets containing positional information, measurement information, and weather information respectively. We see that positional information and measurement information deliver most packets with a low latency. Positional information has priority over measurement information, causing the latter to quickly be erased from the network. However, it spreads better than weather information. This information spreads much less, which can be seen by the much lower number of packets overall delivered. Figure 6.7 shows the delivery ratio during this scenario. We see that positional information has the best delivery ratio, followed by weather information and measurement information. Although measurement information has a higher priority in the network, and therefore would have a higher delivery ratio, it has an earlier deadline compared to weather information. This prevents the measurement information from reaching more nodes.

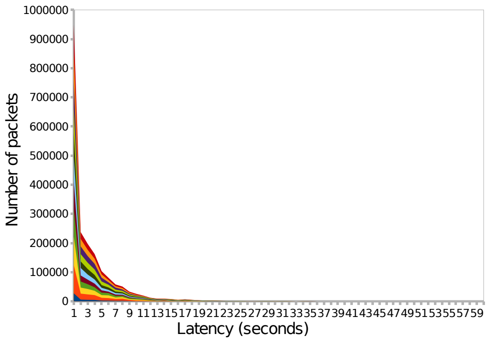

Figure 6.2: *Latency for positional information during a day with good connectivity.*

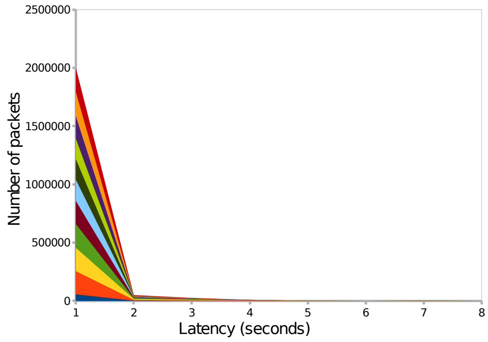

Figure 6.3: *Latency for measurement information during a day with good connectivity.*

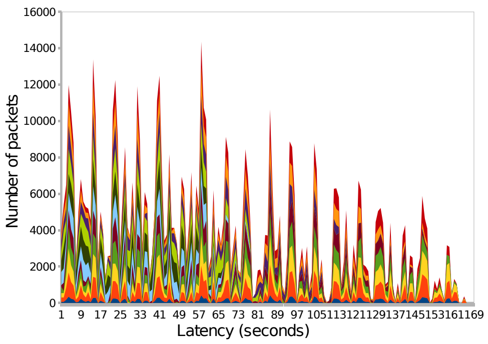

Figure 6.4: *Latency for weather information during a day with good connectivity.*

The other scenario we evaluate is a scenario with very sparse connectivity, for which we use GPS logfiles from a day during October 2009. To further decrease connectivity we disable airfields. During the day network segments meet and separate again. The connections between these segments are through one route, with several hops. Figure 6.5 and Figure 6.6 show the latency of packets containing positional information and measurement information respectively. In these figures the effect of the regular broadcast can clearly be seen. Since two large networks of gliders are connected through a few hops, both positional information and measurement information propagates through these hops. The result of this is the increase of number of packets for the latency between 20 and 45 seconds, which illustrates the time it approximately takes for the packets to travel from one large network of gliders to another. The second increase of packets for the latency above 50 seconds shows opportunistic dissemination. Packets which were stored in memory are eventually delivered when a connection is possible between two segments of the network. Figure 6.8 shows the delivery ratio during this scenario. We

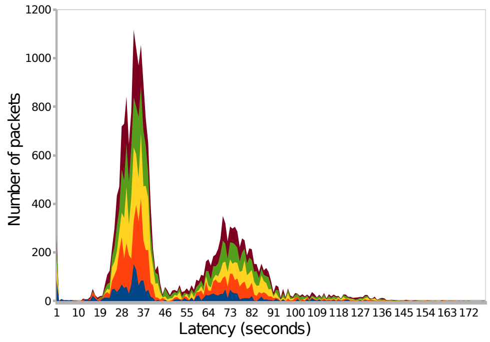

Figure 6.5: *Latency for positional information during a day with sparse connectivity.*

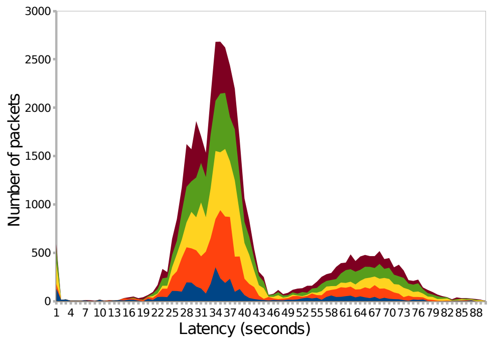

Figure 6.6: *Latency for measurement information during a day with sparse connectivity.*

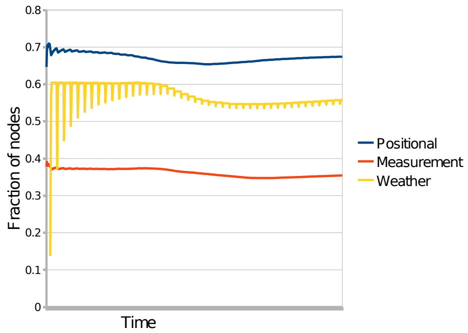

Figure 6.7: *Packet arrival rate plotted over time for a day with sparse connectivity*

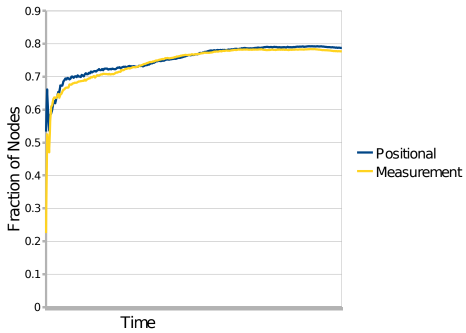

Figure 6.8: *Packet arrival rate plotted over time for a day with good connectivity*

see that the limited number of nodes in the network causes both types of information to be delivered equally well. Since the network is often disconnected in this scenario, not all nodes can be reached.

6.3 Conclusion

We show that our protocol is able to handle situations with good connectivity and situations with sparse connectivity. In both cases our protocol gives better results for high-priority information than for low-priority information. We show that in situations with good connectivity high-priority traffic has a very low latency. In situations with sparse connectivity, the *store-and-forward* communication model is able to deliver information by physically carrying packets.

We deal with the hidden terminal problem by assigning each node a random timer, and provide QoS by assigning each packet a priority. The *store-and-forward* communication model handles mobility causes by gliders flying around.

7 Conclusion, future work and recommendations

> The best way to predict the future is to invent it.
>
> Alan Kay

In this thesis we have proposed Cumulus Humilis, an *opportunistic dissemination* protocol for the exchange of information between airborne gliders using a *wireless mesh network*. It applies a *store-and-forward* communication model to physically carry packets to new locations and assign information a priority. To overcome the hidden terminal problem, it utilizes random timers at each node. Since no existing solutions exist for the communication between gliders, we evaluate the protocol against our requirements. Simulations show it is able to handle situations with good connectivity and situations with sparse connectivity.

We show that a wireless mesh network is the most suitable communication system for communication between gliders, when considering usability, cost and coverage. We show the hidden terminal can be dealt with using random timers at each node, and we show QoS can be dealt with by assigning each type of information a priority. We deal with the mobility caused by gliders moving around implicitly, by spreading information through the entire network.

7.1 Future work and recommendations

7.1.1 Traffic subscriptions

When disseminated data contains metadata describing the data, nodes can subscribe to data described by a set of constraints. When requests for data meeting a set of constraints can be sent, and nodes receiving these requests would locally increase the *utilization factor* for any data meeting the constraints, this would prioritize packets traveling to a node subscribed to the information in the packet. When this proves to be effective, the dissemination as it is performed in *Cumulus Humilis* in this thesis can be limited. This may be more efficient than the dissemination described in this thesis.

7.1.2 Medium Reservation

It has become apparent that CSMA limits our abilities to perform distributed medium reservation for broadcast transmissions. For the problem of Medium reservation two approaches can be chosen: pro-active medium reservation and reactive medium reservation. Better mechanisms for medium reservation, be it a pro-active or a reactive apparoach, can result in more medium capacity and therefore can allow more information to be disseminated over the same network. This would disqualify the X-Bee PRO 868 transceivers for use as transceiver, since the MAC layer of these transceivers cannot be altered.

Pro-active approaches, like LMAC [37] could assign nodes a timeslot to broadcast information in. However, mobility remains a challenge, and the drawbacks [40] RTS/CTS involves also apply for any TDMA approach.

Reactive approaches might not suffer from mobility like TDMA approaches, since the medium is only reserved when a node has data to send. However, performing medium reservation when broadcasting to multiple nodes - who are not aware of each others existence - is also challenging.

7.1.3 Information visualization

When the problem of how to deliver information to a glider pilot's mobile computer is solved, a challenge is how to display the information to the pilot. Since cross-country gliding takes place on days with a lot of sunshine, the sun is shining on the instruments in the cockpit and on the screen of the mobile computer used. Also, since glider pilots have no collission warning system that warns for all airplanes (FLARM only warns for other aircraft equipped with FLARM), the pilot should look outside as much as possible. Therefore, if the pilot reads information from the screen of his mobile computer, this should take as little time as possible. Whenever information can otherwise be presented to the pilot, for example through sounds or speech synthesis, this should be preferred over presenting the information graphically to the pilot. When the information is offered via sound effects, the pilot does not need to shift his focus to his intrument panel and instead he can continue looking outside while listening.

Information overload is also an issue, especially when presenting the pilot with many sorts of graphical information. Information presented on a screen should have strict hierarchy, such that the attention of the pilot turns to the most critical event at that time. Studies have been performed in this area, for example by NASA [43]. However, the specific circumstances present in the cockpit of a glider and the limited room for sophisticated coomputer systems require research in this area.

Bibliography

[1] T. Benson, "Vector force balance for a glider," Jul 2008. http://www.grc.nasa.gov/WWW/K-12/airplane/glidvec.html.

[2] Jonker Sailplanes South Africa, "Jonker sailplanes," Aug 2010. http://jonkersailplanes.co.za/specifications.htm.

[3] Segelflugszene GmbH, *OLC - Gliding - Overview*, Jun 2010. http://www.onlinecontest.org/olc-2.0/gliding/.

[4] R. Heise, K. Ohlmann, and J. Hacker, "Mountain wave project," Apr 2010. http://www.mountain-wave-project.com/index-1.html.

[5] E. J. Carter and E. H. Teets, "Stratospheric mountain waves: Observations and modeling for a proposed sailplane that will use these waves to reach 100,000 feet," Jan 2002. http://www.perlanproject.com/pdfs/AMS-Perlan.pdf.

[6] H. Mühlenberg, "Die windreiter der anden," May 2007. Documentary broadcasted on Arte. Spoken in German.

[7] W. Adriaansen, "Theorie van het zweefvliegen," 1995. ISBN: 90-9000722-9.

[8] W. Bleeker, J. H. Heiërman, and F. H. Schmidt, "De meteorologische grondslagen van het zweefvliegen," 1948. ISBN: 49-21634.

[9] U. Rothacher, A. Schlapbach, and U. Mäder, "Traffic and collision warning for general aviation," Jun 2010. http://www.flarm.com.

[10] Iridium Communications Inc., "Iridium satellite communications," Jul 2010. http://www.iridium.com/.

[11] G. Jones, "Introduction to packet radio," 1995. http://www.tapr.org/pr_intro.html.

[12] Digi International Inc., "Digi international," Jul 2010. http://www.digi.com.

[13] "Electromagnetic compatibility and radio spectrum matters (ERM); short range devices (SRD); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with power levels ranging up to 500 mw; part 1: Technical characteristics and test methods," Apr 2006. ETSI EN 300 220-1.

[14] "Electromagnetic compatibility and radio spectrum matters (ERM); short range devices (SRD); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with power levels ranging up to 500 mw; part 2: Harmonized en covering essential requirements under article 3.2 of the r&tte directive," Apr 2006. ETSI EN 300 220-2.

[15] Digi International Inc., *XBee-PRO®868 OEM RF Modules*, Nov 2008. http://ftp1.digi.com/support/documentation/90001020_a.pdf.

[16] Digi International, *XBee-PRO 868 OEM RF Modules - Digi International*, Jun 2010. http://www.digi.com/products/wireless/point-multipoint/xbee-pro-868.jsp#models.

[17] L. C. of the IEEE Computer Society, "Wireless LAN medium access control MAC and physical layer PHY specifications," Jun 2007. http://standards.ieee.org/getieee802/download/802.11-2007.pdf.

[18] Pixlr Team, "Photo editor online," Jul 2010. http://pixlr.com/editor/.

[19] Commissie Instructie en Veiligheid KNVvL, "Website commissie instructie en veiligheid," Jul 2010. http://www.civ.zweefportaal.nl/main/news.php.

[20] Fédération Aéronautique Internationale, *World Record Claims - Class D (Gliders)*, Jun 2010. http://records.fai.org/data?v1=275&v5=8425.

[21] H. Hodson, "Pressure-probes," Jun 2010. http://www-g.eng.cam.ac.uk/whittle/current-research/hph/pressure-probes/pressure-probes.html.

[22] Y. Li, S. Peng, and W. Chu, "K-mcore for multicasting on mobile ad hoc networks," *Parallel and Distributed Computing Applications and Technologies, International Conference on*, vol. 0, pp. 109–114, 2006.

[23] A. Datta, "Autonomous gossiping in ad-hoc networks," 2003.

[24] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," Aug 2007.

[25] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," pp. 174–185, 1999.

[26] A. P. Sistla, O. Wolfson, and B. Xu, "Opportunistic data dissemination in mobile peer-to-peer networks," in *Advances in Spatial and Temporal Databases*, pp. 346–363, 2005.

[27] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," tech. rep., 2001.

[28] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," pp. 234–244, 1994.

[29] T. Clausen and P. Jacqueet, "Rfc3626: Optimized link state routing protocol (olsr)," Oct 2003. http://tools.ietf.org/html/rfc3626.

[30] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pp. 90–100, 1999.

[31] C. Perkins, E. Belding-Royer, and S. Das, "RFC 3561: Ad hoc on-demand distance vector (AODV) routing," Jul 2003.

[32] C. E. Perkins, *Ad Hoc Networking.* Addison Wesley, 2001.

[33] T. Kunz and E. Cheng, "On-demand multicasting in ad-hoc networks: Comparing aodv and odmrp," 2002.

[34] D. Johnson, Y. Hu, and D. Maltz, "Rfc4728: The dynamic source routing protocol (dsr)," Feb 2007. `http://tools.ietf.org/html/rfc4728`.

[35] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," pp. 1298–1302, 2002.

[36] S.-J. Lee, *Routing and Multicasting Stategies in Wireless Mobile Ad hoc Networks.* PhD thesis, University of California, 2000.

[37] L. F. W. V. Hoesel and P. J. M. Havinga, "A lightweight medium access protocol for wireless sensor networks," tech. rep., 2004.

[38] I. R. Philp and J. W. S. Liu, "Ss/tdma scheduling of real-time periodic messages," in *Intl. Conf. on Telecomm. Systems*, pp. 244–251, 1996.

[39] G. Pei and C. Chien, "Low power TDMA in large wireless sensor networks," vol. 1, (Vienna, VA), pp. 347–351, Oct. 2001.

[40] K. Xu, M. Gerla, and S. Bae, "How effective is the ieee 802.11 rts/cts handshake in ad hoc networks?,"

[41] J. L. Sobrinho, R. de Haan, and J. M. Brázio, "Why rts-cts is not your ideal wireless lan multiple access protocol," 2005.

[42] W. Horn, "Some simple scheduling algorithms," 1974.

[43] R. Mogford, "Using color in information display graphics," Jul 2010. `http://colorusage.arc.nasa.gov/`.