Geographic Routing in Wireless Sensor Networks for Surveillance

Introducing a novel routing algorithm: GZOR

Date last modification	24/06/08
Author	Arjan Dam arjandam@gmail.com
Supervision	Ir. J. Scholten Ing. P. J. M. Olde Damink M.Sc. J. Slagman Dr. Ing. P. Havinga
Classification	Unclassified

THALES

University of Twente
Enschede - The Netherlands

Geographic Routing in Wireless Sensor Networks for Surveillance

Master of Science final project: "Introducing a novel routing algorithm: GZOR"

Student
Arjan Dam
0020818
arjandam@gmail.com

Supervisors:
J. Scholten, University of Twente
P.J.M. Olde Damink, Thales Nederland B.V.
J. Slagman, Thales Nederland B.V.
P. Havinga, University of Twente

ABSTRACT

Wireless Sensor Network (WSN) technology is an upcoming field of research throughout the last decade. Thales Nederland B.V. is investigating if this technology is applicable for surveillance purposes, in cooperation with Twente University. In this scenario such a network will have to exist of thousands of position-aware sensor nodes. The nodes must monitor a large area to detect and report incidental hostile intrusions. Intrusion data has to be transmitted to a base station. Therefore the network will be equipped with dozens of gateway nodes, which have a stronger radio and battery and are able to communicate with a base station. The sensor nodes must route the detection data to a gateway by means of a multi hop routing algorithm.
Since the nodes need to be position aware for the surveillance purpose, this information can be utilized to increase efficiency and performance of routing algorithms. Geographic routing algorithms form a subclass of WSN routing algorithms. They are developed to deliver reliable any-to-any connections between all nodes in an energy efficient and scalable manner. The intended surveillance network does not share these hard requirements. This research has explored the possibilities to improve the energy efficiency by loosening the delivery requirements. This has led to the development of a novel geographic routing algorithm which is introduced in this document. Geographic Zero Overhead Routing (GZOR) is a state-free algorithm based on the concepts of volunteer forwarding and multipath routing. This combination creates robust and dynamic routing paths. The algorithm is intended to route packets from nodes to gateways with an acceptable delivery rate. This can be summarized as best-effort many-to-some routing. GZOR nodes do not explore the network topology and do not provide transmission feedback. As a result, GZOR does not require proactive or reactive communication overhead. This property causes GZOR to be very energy efficient and scalable. It also allows nodes to engage in asynchronous energy-conserving sleep cycles, which can greatly extend the lifetime of a network.
GZOR's performance and behaviour is quantified and analysed by simulation. It is compared with Greedy Perimeter Stateless Routing (GPSR), which is a well-known and studied geographic routing algorithm. GPSR is a routing algorithm that maintains position information on neighbouring nodes to decide to which nodes packets have to be send. Both algorithms are simulated onto various deployments and network densities. This research demonstrates that GZOR is able to achieve an acceptable delivery rate with a significantly smaller amount of communication than GPSR.

SAMENVATTING

Draadloze sensornetwerk-technologie is een opkomend onderzoeksveld sinds de laatste tien jaar. Thales Nederland B.V. onderzoekt in samenwerking met de Universiteit Twente of deze technologie toepasbaar is voor bewakingsdoeleinden. In dit scenario zou zo'n netwerk moeten bestaan uit duizenden positie-bewuste sensor nodes. De nodes moeten een groot gebied monitoren en zodoende vijandige indringers detecteren. De detectie-informatie moeten worden doorgezonden naar een basis station. Hiervoor moet het netwerk worden uitgerust met zogenaamde toegangsnodes. Deze nodes hebben een sterkere radio en batterij, waardoor ze in staat zijn tot communicatie met het basisstation. De sensor nodes moeten elkaars detectie data routeren naar een toegangsnode door middel van een multi-hop algoritme.

De nodes moeten hun positie kennen voor het bewakingsdoeleinde van het netwerk. Deze informatie kan ook gebruikt worden om de energie efficiëntie en schaalbaarheid van routing algoritmes te verbeteren. Geografische routeringsalgoritmes maken hiervan gebruik en vormen een subklasse van routeringsalgoritmes. Deze algoritmes zijn ontworpen om betrouwbare één-op-één verbindingen tussen alle individuele nodes in het netwerk tot stand te brengen op een efficiënte en schaalbare manier. Voor het beoogde bewakingsdoeleinde hoeven nodes alleen naar toegangsnodes te routeren en een acceptabele hoeveelheid aan dataverlies is toegestaan. Daarom is er onderzocht of het versoepelen van de netwerkeisen kan leiden tot een algoritme dat nog zuiniger omgaat met energie. Dit heeft geleid tot de ontwikkeling van een nieuw algoritme, dat in dit document geïntroduceerd wordt. Geographic Zero Overhead Routing (GZOR) is een toestandsloos algoritme, gebaseerd op de concepten van vrijwillig doorsturen en routering over meerdere paden. Deze combinatie stelt GZOR in staat om robuuste en dynamische routeringspaden te creëren. Het algoritme is bedoeld om pakketten te routeren van nodes naar toegangsnodes met een acceptabele aankomstratio. Dit kan worden samengevat als beste-poging, veel-naar-enkelen routering. GZOR-nodes hebben geen informatie over de netwerk-topologie nodig en geven ook geen terugkoppeling over het succes van transmissies. Daarom heeft GZOR geen proactieve of reactieve communicatie-overhead nodig. Deze eigenschap zorgt ervoor dat GZOR heel erg energie-efficiënt en schaalbaar is. Het algoritme staat ook toe dat nodes zich in asynchrone energie-conserverende slaapcycli begeven. Dit kan de levensduur van een netwerk aanzienlijk verlengen.

De prestaties en het gedrag van GZOR zijn gekwantificeerd en geanalyseerd door middel van simulatie. De resultaten zijn vergeleken Greedy Perimeter Stateless Routing (GPSR). GPSR is een algemeen bekend en bestudeerd geografisch routeringsalgoritme. GPSR-nodes onderhouden tabellen met positie-informatie van de omringende nodes en baseren daarop naar welke nodes pakketten moeten worden doorgestuurd. Beide algoritmes zijn gesimuleerd op verschillende netwerkopstellingen en -dichtheden. Dit onderzoek toont aan dat GZOR in staat is om een acceptabel aankomstratio te leveren en hiervoor een significant kleinere hoeveelheid communicatie nodig heeft dan GPSR.

Contents

GLOSSARY

ACK	Acknowledgement. Feedback of a receiving node to the sender whether the transmission was successful.
Ad hoc functionality	The functionality of nodes to form a functional network autonomously, without central control.
Anchor	GPS-equipped node.
Asynchronous sleep cycle	The sleep cycle is the proportion of time a node is not operational. Asynchronous sleep cycles are uncoordinated sleep cycles. Nodes do not know when their neighbours are asleep.
Connectivity void	A physical area in the network without links between the nodes on either side of this area.
Delivery rate	Percentage of transmitted packets received by a gateway.
Dijkstra's algorithm	Graph search algorithm which can find the shortest path between two vertices.
Duty cycle	Proportion of time during which a node is in reception mode.
ETX	Expected transmission count. The average amount of packets that must be transmitted to get a packet over a link with a certain PRR.
Face routing	Routing algorithm which uses the planar graph to route around connectivity voids.
Gateway	A node capable of communication with the base station.
Greedy routing	Routing algorithm where a node forwards packets to the neighbour which would induce the most progress towards the packet's destination.
GPSR	Greedy Perimeter Stateless Routing. A WSN routing algorithm which combines greedy and face routing.
GZOR	Geographic Zero Overhead Routing. The novel algorithm described in this document.
Isotropic	Uniform behaviour in all directions.
Localization	The process where nodes acquire their position through communication with surrounding nodes and anchors.
MADM	Multiple Air Deployment Model. A network model which consists of several overlapping aerial deployments.
Mica2 node	Sensor node developed by crossbow, equipped with a CC1000 radio module.
Overhead	Transmitted data required for routing purposes. All data that is not detection data is overhead. In this document the term overhead points at packets without detection data.
Planarization	A process where all crossing links of a network graph are removed from routing tables.
PRR	Packet Reception Ratio. The probability on successful reception of a packet across a link between two nodes.
RSS	Received Signal Strength. The intensity (dB) of the received signal in relative to some reference. This value is commonly used as an indication on distance between two nodes.
Stretch	The average summed amount of transmissions the nodes in a network must to do to get a single packet from sender to destination.
Thales simulator	The Matlab WSN simulator developed as a student research project within Thales.
TinyOS	The operating system running on the Mica2 sensor nodes.
WSN	Wireless Sensor Network. A network of sensor nodes equipped with a radio model.

1 Introduction

Context of this document

This document contains the fourth student research on wireless ad hoc sensor networks performed in a cooperation of Twente University and Thales Nederland B.V. Thales is interested in this field of research since this technology has potential to offer an efficient mechanism to extend the ground surveillance possibilities.
A wireless sensor network (WSN) consists of a large amount of small, low-cost, radio equipped, battery-powered sensors, which are called nodes. These nodes, once deployed, have to localize themselves and form a network autonomously. An intruder, entering the network deployment area, will be detected by the nodes. After a detection, the nodes try to report this information to the military base, where the detection data can be post-processed. Such a network has to be operational for at least several months after deployment, therefore energy efficiency is a major factor in the design of this technology.
Currently, Thales is specialized in radar technology and offers a ground radar solution for compound security and area surveillance. This technology, however, does have some vulnerabilities. A radar installation is a very expensive piece of equipment and has to be manned and guarded continuously, which could impose high risks in a hostile environment. Besides, as radar is a centralized solution, it is susceptible to objects blocking its line of sight. Wireless sensor network technology might offer a good solution for monitoring enemy activity in blind areas of the compound radar or in hostile territory where deployment of a radar installation is difficult, dangerous or completely impossible.
Wireless ad hoc network technology is currently still in its infancy. Thales would like to explore the possibilities of this new technology and gain knowledge about the viability of this solution to meet military surveillance requirements. Prior to this research, the global concepts of WSN technology were explored and an assessment was made about applicability for surveillance purposes [Bos06]. This survey was followed by a research that focused on node localization algorithms based on signal strength indications (range-free localization) and the presumption that a subset of nodes is equipped with a GPS module (*anchors)* [Sla07]. The third research focused on simulation of high-level WSN algorithms for operation in large networks [Dam08].

Problem statement

A wireless sensor network consists of a large amount of sensor nodes with a low-power radio. When a node detects an intruder, it must report this to the base station. The radio of a node itself, unfortunately, is not strong enough to send its detection packet directly to the base. It can, however, reach its neighbouring nodes. Therefore the network has to be equipped with *gateway* nodes, also known as *sinks*, which have a stronger radio and can directly transmit to the base. The network nodes can now cooperate, forwarding the detection message until it reaches a gateway, where it will be transmitted to the base. This method is called multi-hop routing. Since all nodes are battery powered, this forwarding has to be done according to an efficient algorithm, where as much energy has to be saved as possible. Since the nodes are aware of their position, or at least have an estimation of it, this information can be used to aid routing. This class of routing is calling geographic routing.
This research aims at the development of such an algorithm. This algorithm has to be robust, reliable, efficient and scalable. The algorithm has to be simulated in order to monitor its behaviour and assess whether it holds the above mentioned properties.

About this document

This document starts with an overview of the related work in chapter 2. This chapter describes correlated research on WSN routing algorithms. Chapter 3 contains an analysis of the intended network and states the requirements of a routing solution. The research methodology is described in chapter 4. The design issues, solutions and implementation details of the developed algorithm are presented in chapter 5. The algorithm is named Geographic Zero Overhead Routing (GZOR).

Chapter 6 describes another geographic routing algorithm; Greedy Perimeter Stateless Routing (GPSR). This algorithm is developed and analysed by other geographic routing research groups. The GPSR algorithm is used as a comparison to evaluate GZOR's performance. The algorithm was also implemented on the utilized simulator. Details on this implementation can also be found in this chapter 6. Chapter 7 describes how both algorithms are simulated and analysed. The performance results of these simulations are presented in chapter 8. An analysis of the behaviour of both algorithms, which gives insight on applicability and future optimizations, can also be found in chapter 8. Chapter 9 contains a discussion where the performance results and findings are placed in a more general context. Chapter 10 summarizes this research and provides recommendations for future work.

2 Related work

Multi-hop routing algorithms have been a major issue in the WSN research field for many years. Traditional solutions to this problem are *dynamic source routing* (DSR) [Jon96] and *ad-hoc on-demand distance vector routing* (AODV) [Per99]. DSR tries to discover routes from sender to destination with network floods. Intermediate nodes forward route discovery packets and append themselves to it. When the destination receives such a packet, it knows the route from sender to destination, and replies to sender along this route with this information. In AODV, nodes will construct a local routing table on demand. This table contains information through which neighbour a gateway node can be reached and at what cost. Nodes broadcast this information to each other, so every node knows at a local level to which node specific packets have to be forwarded. Both algorithms require a relatively high amount of overhead to share topology information resulting in scalability and energy performance issues.

For surveillance purposes nodes are required to know their position, or at least have a good estimation. Otherwise the detection information they broadcast would be useless, since the location of a detection is essential. Position information can be utilized to aid routing and reduce topology information overhead. This subclass of multi-hop routing is calling geographic or location-based routing. Since this method offers great potential to reduce overhead and thus save energy, a great amount of research has been done to create a solution to exploit this information as much as possible [Gio04].

Karp proposes a solution called geographic forwarding [Kar00]. This method uses greedy forwarding to decide to which node a packet must be forwarded. Every node knowns the location of all of its neighbours and packets contain the location of the destination. Nodes forward packets to the neighbour closest to the destination, which ensures that every broadcast will result in the packet travelling towards its destination. Since topology information only has to be exchanged locally, greedy forwarding is very scalable and many variants have been proposed [Bar01], [ZSK06]. The largest problem with this method is overcoming holes in network connectivity, nodes that do not have a neighbour closer to the destination than itself. Most algorithms use a variant of greedy perimeter stateless routing (GPSR) [KaK00] to bypass such a void (FACE [Bos01], GOAFR+ [Kuh03]).

Ko and Vaidya suggest an approach where DSR is optimized by restricting a route detection network flood to the direction of the destination [KoV00]. This method is known as location-aided routing (LAR) and prevents network-wide floods [Mar04]. Variations to this technique also explore possibilities to use multipath routing for increasing robustness and reliability [Gan01], [Dul03].

The mentioned routing algorithms all implement forwarding nodes to send packets to a specific next hop. *Zorzi and Rao* suggest that method restricts nodes from energy saving power-off strategies, because this causes network connections to become lost [ZoR03]. They propose volunteer forwarding as a solution to this restriction [Zor03]. Nodes do not specify the next hop, they simply broadcast the packet on the network and receiving nodes must decide which one will forward the packet. Nodes closest to the destination should get a higher priority to forward the packet. This technique introduces new problems, such as packet duplication and node contention schemes and it also does not solve the connectivity void issues. It does however offer a solution to highly dynamic networks with unstable links and nodes with asynchronous sleeping cycles, therefore several studies on this topic have been performed. They have produced many variations of this algorithm which incorporate volunteer forwarding [XuL05], [Blu03], [Wit05].

3 Requirements analysis

The goal of this research is the design of a routing solution for a surveillance network proposed by Thales. For this purpose the boundary conditions and network requirements first have to be identified. This chapter states the global requirements and gives insight into smaller design issues.

3.1 Energy efficiency

Once the network is deployed, it has to be operational for at least several months. This requirement introduces the need for energy efficiency. Therefore it is important to identify the different states of a node and the power consumption concerned with such a state.

- When a node does not participate in the communication process at all, is it said to be in sleep mode (radio off). The sensor of the node is functional, but the node will not hear other nodes broadcasting. This state has the lowest current draw ($<100\mu$A) [Shn04].
- If the node is able to receive packets from neighbouring nodes and thus is listening to the radio channel, it is said to be in idle mode (radio on). A node in idle mode consumes 100-150 times more power than in sleep mode (current draw: 10mA) [Lan05]. In idle mode the node can use a duty cycle mechanism [YeS06], which means the node's radio is rapidly alternating between listening and sleeping (for instance 1% listening, 99% sleeping). When another node wants to send a packet, it first sends a preamble, which takes long enough so that all the nodes will have some duty time to hear that a packet will be sent shortly. With a 1% duty cycle the current draw is 0.1mA [Chi06].
- When the node is receiving a packet, it consumes 20% more power than in idle mode. The current draw is 12.5mA [Shn04].
- If the node is transmitting a packet, it draws twice as much current as in receiving mode (25mA) [Shn04].
- Communication is in terms of energy consumption the most dominant factor, internal calculations are only invoked by sensor readings or received transmissions. While executing instructions, the node draws 8mA [Shn04].

When analysing the above, transmission is clearly the most costly state, also since it causes the reception state in all receiving nodes [Fee01]. However, it can also be concluded that the difference between the transmission state and the idle state (incorporating a duty cycle mechanism) is of the same significance level as the difference between idle and sleep state. Energy conservation in terms of routing therefore not only lies in the reduction of transmissions, but also in the ability of a routing algorithm to cope with some amount of nodes in longer sleep states. This is especially important because the network will have no transmissions at all during most of its lifetime (since the presence of intruders is considered a rare event).

3.2 Ad hoc functionality

Since the network must be capable of being deployed by an aerial vehicle in a hostile environment, the nodes must be able to autonomously form a network environment capable of localization and routing, and maintain this property. The network will consist of a certain amount of anchor and gateway nodes. The nodes must for themselves decide to which gateway their packets should be sent and how to do this. Nodes are considered to have a stable position.
The network should also be able to be extended with another air drop. The new nodes must be able to easily integrate with the operational network, forming a new larger (or denser) network. When a node stops functioning, it should be able to easily leave the network, without causing network routing to fail.

3.3 Robustness and reliability

We define robustness as the capability of the network to cope with changing network conditions. Reliability is the capability of the network to ensure some average transport delivery rate. We state the following properties of a WSN.

- Radio nodes do not have isotropic transmission ranges, links can be asymmetric and unstable [Zho06].
- The transmission ranges of nodes are not stable, they change due to weather conditions and battery charge depletion [Sla07].
- Due to energy conserving sleep modes, nodes might temporarily be unavailable.
- Concurrent packet transmission can cause packets to get lost [Son06].
- The network will not be uniformly distributed and can contain network connectivity voids.

From these network properties it can be concluded that the topology of the network is continuously changing. A routing algorithm should be robust enough to handle the changing topology and offer the availability of node-gateway connections without an extreme amount of topology overhead.
When detection occurs, the nodes will start to send data at a rate of at most 1 Hz. Postprocessing requires that at least 90% of the data arrives at the base and that data must arrive with a maximum delay of 5 seconds [Bos06]. The reliability can be defined as the minimum delivery rate, which must be at least 90%. Reliability can be improved by retransmissions in case of packet loss and multiple routing paths for the same packet (these methods add redundancy in time and space respectively). Note that the surveillance purpose of the network loosens the requirement of ensuring packet delivery guarantees. As long as the delivery rate is high enough, a best-effort network is a viable solution.

3.4 Scalability

The intended network must be able to monitor an area of about 10 hectare. With a common per-node detection range of 10 meters maximum, this causes an extent of the network to at least 1000 nodes. Larger monitoring areas or non-heterogeneous node distribution increments the required number of nodes rapidly.
This implies that it is impossible to have full network topology information for every node. Routing algorithms on networks at this scale must ensure to have at maximum a linear increase of topology overhead. Maintaining routing tables or node-to-gateway-routes induces too much overhead, especially when the network topology is rapidly changing (section 3.3). Optimal paths conserve energy, but the trade-off against network overhead to find and maintain such paths might lead to the situation where the use of non-optimal paths is a better strategy.
Furthermore, loosening the constraint of packet duplication prevention could potentially increase robustness (emergence of multi-paths) and decrease communication overhead (no contention schemes on packet forwarding). However, this could also incite a flooding of the network by duplicate packets causing the network scalability to decrease; here lies a delicate trade-off.

4 Research Methodology

The usage of wireless sensor network technology for security purposes is a new concept of which there currently does not exist a functional implementation. The requirements stated in the previous chapter differ from what is considered standard in terms of scalability and durability in the field of WSN research. Most research focuses on many-to-many reliable communication between nodes in a 100 node network. Our intended solution, however, involves many-to-some *reliable-enough* communication in a 1000+ node network. Although the basic hardware technology is the same, the difference in application requires to re-examine current routing algorithms and their applicability in our intended solution. Within this research the latter has led to the design of a new routing algorithm specifically designed for the surveillance purpose solution of Thales. The algorithm is called Geographic Zero Overhead Routing and will be referred to as GZOR throughout the rest of this document. Although the developed algorithm could be applicable to other applications in the WSN field, this has not been the focus of this research.

4.1 Simulation

The process from design to final implementation of a routing algorithm goes through several stages, including theoretical design, simulation, prototyping, implementation and testing. This research only involves the first two steps and aims at the demonstration of the theoretical viability of the created solution. The latter justifies taking the future next step of creating a prototype, which in this field can be very costly. The research scope is illustrated in Figure 1.

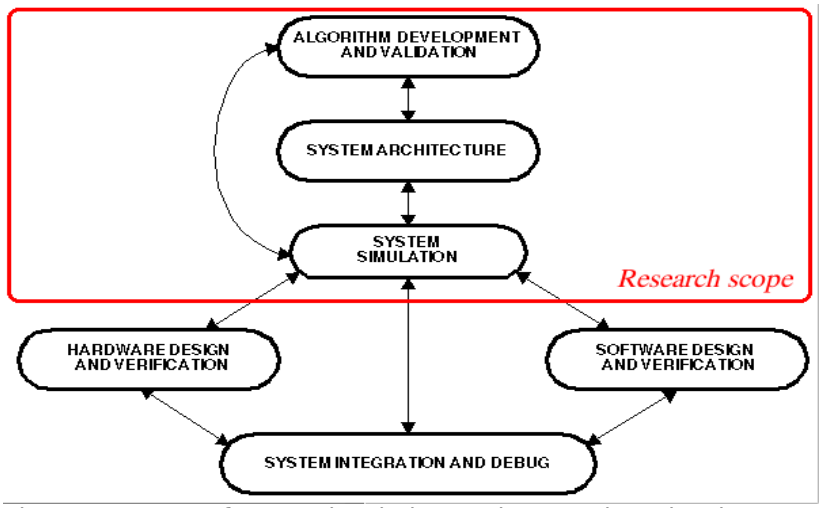

Figure 1: Scope of research relative to the complete development process [Ber93].

In order to demonstrate the viability of the designed algorithm it is very important to have a realistic simulator. The simulator must not only be founded by state-of-the-art knowledge on radio behaviour, it must also be able to give insight in the properties of the developed conceptual algorithm. The previous research performed within Thales by the author of this document has focused on the development of such a simulator [Dam08]. This simulator is based on a radio model of the Mica2 node, the development platform adopted by Thales for research on WSN technologies. Another Mica2 simulator is Tossim [Lev03] (included in the standard TinyOS release), which can execute and simulate NesC code written for the TinyOS [Lev05] platform. This simulator can be well used to evaluate, test and debug algorithms during future implementation stages. However, Tossim requires a full NesC implementation of algorithms, which could distort the focus of design at the current stage of development. This prevents from quickly exploring new fundamental concepts of geographic routing. Furthermore, this simulator does not allow reproducible experiments and interaction with the simulation is considered laborious by the Tossim developers. Therefore the Thales simulator is a better platform for design and analysis of routing algorithm concepts.

The simulator described in [Dam08] only supports methods for quantification of algorithms. Therefore the simulator had to be extended for support of analysis of algorithm behaviour. This includes reproducible experiments where the path of a packet traversing through the network can be evaluated, as well as the routing decisions made on each specific node along the way. Details on the extensions can be found in section 7.3.

4.2 Evaluation and quantification

The research goal is the development of a routing algorithm specifically suited to deal with the requirements imposed by the hostile security environment. It is important to evaluate this algorithm by means of a simulation, which closely resembles the conditions an actual implementation will have to cope with in practice. This means that in order to evaluate the performance of GZOR, the simulation must include non-trivial imperfections, such as localization errors and non-optimal node deployment. In chapter 7 details on the simulation conditions are elaborated.
For performance measurement of GZOR it is important define calibration points for evaluation. In addition, metrics need to be defined for comparison with the calibration points. The approach of this research includes two calibration points: a lower bound and a competing algorithm. The lower bound is a mathematical analysis of the network and a calculation of the optimal path. An average better performance than the lower bound is theoretically impossible. The competing algorithm is the geographic WSN routing algorithm Greedy Perimeter Stateless Routing, which is developed and thoroughly examined by other research groups [Sea07] [Kim05] [Fre06]. This algorithm and the implementation used for comparison is described in chapter 6.
The metrics used for quantification, evaluation and comparison of GZOR's performance are defined as follows:

- Delivery rate (%): The percentage of packets that successfully arrive at the intended destination.
- Stretch (#transmissions): The average sum of transmissions that the nodes in the network must do to get a single packet from a source node to destination.
- Node density (neighbours/nodes): The average amount of neighbour nodes a single node has in the network. Two nodes are defined to be neighbours if they can communicate with each other with a packet loss probability of less than 30% (bidirectional).
- Percentage of asynchronous sleep (%): The percentage of time operational (non-gateway) nodes asynchronously switch off their radio to engage an energy-saving modus.

Both algorithms are quantified according to these metrics. The comparison of GZOR with the two calibration points gives a solid view on GZOR's performance. This provides a foundation on which the conclusion on the viability of the new routing concept for the intended solution is based.
Evaluation of the behaviour of both algorithms is achieved by analysis of individual simulations. Such evaluation is necessary to demonstrate the strengths and weaknesses of both algorithms. The latter is important to explain the difference in performance and to provide well-founded conclusions on the applicability of GZOR and GPSR in specific situations.

5 Development of Geographic Routing Algorithm GZOR

This chapter describes the design issues and functioning of the developed geographic routing algorithm. The results of simulations of this algorithm can be found in chapter 8.

5.1 Design issues

Chapter 3 states several requirements that a routing algorithm for this specific purpose must meet. This section tries to describe how these issues lead to design decisions and what solutions are adopted.

5.1.1 Best effort delivery versus reliable data transport

A fundamental issue in the algorithm design process is the decision whether a best effort delivery or reliable data transport strategy is adopted [Wil05]. In a best effort delivery strategy (also known as stochastic delivery) packets are simply sent and no feedback is given back to the sender on the delivery status of the packet. The network tries to deliver the packet, but if it gets lost along the way, no retransmissions will take place. Reliable data transport, on the other hand, offers guaranteed delivery. This can be implemented in two ways. Either the sender gets feedback on the delivery status of the packet in the form of an acknowledgement packet (ACK) from the destination, so that it can consider to retransmit the packet when this ACK stays out (this is called end-to-end reliability), or via a per-hop strategy, where intermediate hops send feedback messages to each other after each single transmission, so the network itself guarantees delivery at the MAC-layer. In both cases, a reliable connection strategy will at least double the number of transmissions needed to get a single packet from sender to gateway. The extra overhead could result in a packet loss increase caused by an increased chance of concurrent transmissions, which leads to even more retransmissions. This vicious circle occurs when the bandwidth in the network is limited; the effect is amplified when connections are highly unreliable. Retransmissions become the dominant factor, causing the throughput decreasing to a level where the end-to-end delay requirement cannot be met. Asymmetric links also induce a problem in a reliable data transport strategy, an ACK of a successfully received packet might become lost, resulting in retransmissions and an unnecessary waste of bandwidth.

How important is a single detection packet in a surveillance network? We assume that, when an intruder enters the network, the detecting nodes will start to send data at 1 Hz. This means that when a packet is lost, a new packet with roughly the same information will arrive within a second. Since an individual packet does not contain unique, indispensable information, guaranteed delivery is not required. So for surveillance purposes the overall solution can handle a certain amount of packet loss as long as the average delivery rate is high enough. The risk of using a best effort strategy is, however, that individual nodes, whose only path to the gateway is unreliable, will not reach a high enough delivery rate. This could potentially be solved by extra transmissions (what basically happens in a reliable transport strategy), but since the sender does not get feedback this will never happen.

Summarized, a best effort strategy offers a higher energy efficiency and throughput and can be used when guaranteed delivery of single packets is not a requirement; it is also less complex from an implementation perspective. Individual nodes, however, are more likely to fail in reaching a required delivery rate when their shortest connection to the gateway is unreliable. This means that the network would require more nodes to reach a certain detection coverage. The marginal costs of individual nodes is assumed less important than the benefits of extending the network's lifetime. Besides that, guaranteed delivery of individual packets is not a hard requirement in a surveillance network. Therefore we adopt a best effort delivery strategy in our approach.

5.1.2 State-based versus stateless

Geographic routing algorithms can roughly be divided into two categories: state-based and stateless. State-based algorithms assume that each node maintains some information to which node certain packets must be forwarded, so it can select the next hop accordingly. This can either be through the means of path finding (e.g. LAR) or by maintaining a vector of the locations of neighbouring nodes (e.g. GPSR). Either way, nodes have to exchange messages to keep this information up-to-date. This

can be done on-demand, which causes delay on packet transmissions, but does not waste energy on maintaining unused links. Maintaining state potentially offers fast, optimal paths. However, in a highly unreliable network where the network topology continuously changes, the amount of overhead required to keep the network connected might be insuperable.
The second category involves stateless algorithms, in which the sender does not specify a receiver and simply broadcasts the packet on the network (e.g. GeRaF). The receivers decide which node forwards the packet by means of volunteer forwarding, using some mechanism where the node belonging to the most optimal path should be favoured. This method will severely reduce overhead and thus result in a higher energy efficiency. Stateless routing also allows nodes to be in asynchronous, periodic energy conserving sleep states, which is impossible in state-based algorithms because links would become disconnected. It does, however, induce new challenges such as the risk of packet duplication and the inability to overcome connectivity voids (because opposite to state-based routing, nodes are unaware of connectivity voids). Another risk lies in possible concurrent transmissions emerging when two nodes both want to volunteer as the forwarding node.
The proposed network has a very dynamic network topology, caused by sleeping states and unreliable links. Also overhead reduction, as a result of the absence of topology information packets, will increase the scalability and lifetime of the network. Therefore the stateless routing paradigm is chosen as a basis for our algorithm.

5.1.3 Multiple short hops versus long hops

The transmission range of a radio node depends on the transmit power. Besides, the further away two nodes are from each other, the more likely the link between them is unstable or asymmetric [Zha03]. This raises the question whether routing over many short-distance hops is preferred over messages travelling long distances. This way the transmit power could be lowered and also the added interference of a single transmission to the network would be smaller. Or, instead of lowering the transmit power, unstable links could simply be rejected. The complete connection from sender to destination would be more reliable since far and unstable links, were packet loss is more likely, are not used.
In [Hae04], *Haenggi* argues that transmit power reduction does not lead to reduced energy consumption because of the nature of low-power radios (power usage of local oscillators and bias circuitry dominate). It is also unclear whether a signal transmission at high power causes more interference in the network than multiple low power transmissions. On top of that, lowering the transmit power of network nodes will cause more connectivity voids in the network and will result in a larger burden to the nodes closest to a gateway. Transmit power reduction is clearly not an attractive option in our case.
The rejection of unstable links is another point that needs consideration. Packets can be lost on such a link. In case of reliable data transport or state-based routing link rejection could be an interesting strategy, because it reduces the overhead caused by lost packets or links becoming unconnected. In our case, however, it might be a better strategy to exploit such unstable (or asymmetric) links whenever possible since this reduces the amount of transmissions needed to get a packet from sender to gateway and it increases the chance of leaping over a connectivity void.

5.1.4 Packet duplication constraint

Duplicate packets in a network waste bandwidth, that is why most algorithms try to prevent this. Preventing packet duplication in a stateless routing algorithm requires some overhead to let nodes mutually decide who will forward the packet. Also a short-hop propagation strategy can be used, where every node can hear all other nodes possibly volunteering to forward the packet and back-off.
It is interesting to explore what happens in a network, such as ours, when this constraint is loosened. We assume that this can cause two scenarios: a single node doing an unnecessary transmission or the emergence of (disjoint or braided) multi-paths.

This is illustrated in Figure 2; red lines represent successful transmissions, the numbers indicate the order of events. In the first scenario, node *a* transmits a packet, node *b* and *c* receive the transmission. Node *c*, which lies closest to the destination, forwards this packet. Node *b* fails to receive this transmission and also forwards the packet it received from node *a*. Node *d,* who has received the transmission from *c,* will now also receive the transmission from *b*. Node *d* already had this packet, so node *b* did an unnecessary transmission. Note that since node *d* also receives the same packet from node *e*, which is closer to the destination, it will reject the packet anyway.

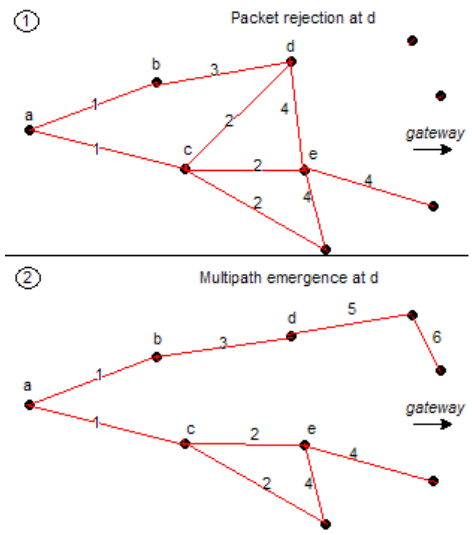

Figure 2: Example of effects caused by packet duplication

Now consider the second scenario, where node *d* did not receive the transmission from *c*, but does receive the packet through the transmission from *b*. Since node *d* also does not receive the transmission from node *e*, it will forward this packet and a multipath emerges.

Dropping the constraint of packet duplication could deprive our network of any overhead and exploit unstable connections, at the cost of some unnecessary transmissions. Furthermore, possible multipath emergence (which offers more resilience to connectivity voids) will increase the robustness of our routing solution in a trade-off with increased transmission redundancy. Therefore we drop the constraint of duplicate packet prevention. This method does require packets to have a unique identification number, which must be stored at intermediate nodes, so a single node will not repeatedly forward the same packet. This causes our algorithm to be semi-stateless; nodes do not keep information about links or neighbours but do (temporarily) store information about received packets.

5.2 Geographic Zero Overhead Routing (GZOR)

The previous section outlines the design issues of the routing algorithm. We call our algorithm Geographic Zero Overhead Routing (GZOR). GZOR has zero topology overhead, but does presume to have a small amount of knowledge of the network (as a result of localization). Since there is no overhead, a node receiving a packet has to solely consider whether to forward or reject the packet. It can, however, hear other nodes broadcasting the same packet and incorporate this information in its decision. This section explains how the nodes make these decisions. Also a mechanism to prevent floods of duplicate packets in the network is presented.

We assume that each node participates in the localization process. As a result of this, the node has estimates (by ranging) about the distances between itself and its neighbours. We also assume that a node knows the location of the packet's destination (in this case: a gateway).

The global concept of GZOR is a form of volunteer forwarding, where a single node broadcasts a packet without determining the next hop. Ideally, only the furthest receiving node (the one closest to the gateway) forwards this packet. Also, since nodes do not have equal transmission ranges opposite to each other (due to hardware variation, differences in battery power and environmental factors), the node with the furthest reach should be favoured.

GZOR tries to approximate this behaviour by using a timing mechanism such that nodes closer to the gateway are more likely to be the first to broadcast. Nodes less close to the gateway will hear this broadcast and decide that their broadcast is unnecessary and therefore back off. When the timer of a node expires without having heard closer nodes broadcasting, it will conclude that there is no node closer to the gateway than itself and forward the packet. This global concept comes with some important issues, we distinguish four:

1. A node must determine whether it should forward or reject a received packet.
2. The node must have some mechanism to assess its distance to the gateway and transmission range relative to other receiving nodes, so it can determine at what value its timer has to be set.
3. When receiving transmissions from other nodes concerning the same packet, a node could use this information to adjust its decision.

4. Nodes may not hear other nodes broadcasting, this causes duplicate packets. Some mechanism has to assure that this phenomenon does not cause network wide floods.

For easy comprehension we define the following:

- The *sender* is the node that detects an intruder. This node will create a packet and broadcast it on the network.
- The *gateway* is the destination of a packet. The sender knows the position of this gateway and appends it to the packet.
- A *receiver* is a node who successfully receives a packet and must decide whether to forward or reject it.
- An *intermediate* (relay node) is what a node becomes when it actually forwards a packet. When a sender broadcasts a packet, it is also the intermediate. Receivers refer to the previous sender of the packet as the intermediate, the initial detecting node still remains the sender. So within a packet travelling through the network, the sender and gateway remain unchanged, while the intermediate changes continuously with every transmission.
- *distanceToGateway(x)* defines the distance of node x the the gateway, which can easily be calculated by using Pythagoras.

5.2.1 Forwarding decision

The solution to the first issue is simple. Since nodes know their location, an *intermediate* can append its location to the message; *receivers* can calculate whether they lie closer to the *gateway* than the *intermediate*. If they lie closer, they set their timer and place the packet in their broadcast buffer, else they reject the packet.

```
If (distanceToGateway(intermediate) > distanceToGateway(receiver))
        forward the packet
else
        reject
end
```

5.2.2 Priority determination

For the second issue, the setting of the timer, GZOR uses two parameters: the distance that the packet has travelled and the estimated transmission strength of the *receiver*. For the latter parameter GZOR uses the information gathered by the localization process. [Zun04] describes a correlation between the transmission and reception abilities of a single node. In the localization process, the node estimates the distance to all of its neighbours by ranging (measurement of the signal strength); we call the maximum of all ranging values *maxRange*. We assume that the value of *maxRange* will be higher for strong senders. This is caused by the above correlation; strong senders will correctly receive packets from nodes further away and thus estimate their distance. Weak senders will not be able to correctly receive such a packet. We verified this assumption in the simulator; the result is presented in Figure 3. We simulated a localized network with 1216 nodes, distributed according to the group-based air deployment model (section 5.1.3). For all nodes the value of *maxRange* was calculated, the nodes were then arranged by transmission power in groups differing 0.25dBm, and the mean of all *maxRange* values of the group was taken. The result indicates that the *maxRange* value is indeed a good indicator for the nodes transmission strength.

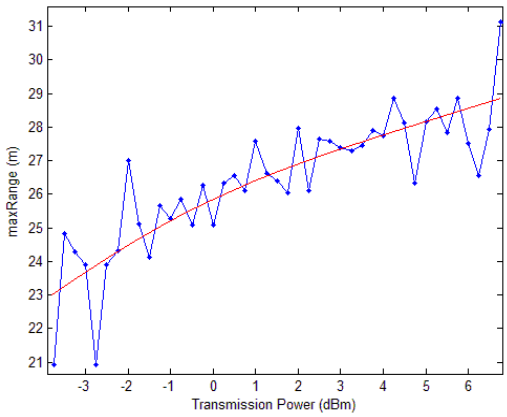

Figure 3: Correlation between transmission power and maxRange

Figure 4 explains the importance of range estimation. The circles in this figure represent the ranges of the blue and red nodes. In this example node *2* has the ability to reach the *destination* while node *1* does not. However, node 1 lies closer to the *destination* than node *2*. It is favourable that node *2* forwards the packet since this is the only way to reach the *destination*. However, if node *1* forwards

first, then node *2* will cancel its timer and the packet will be lost. When appending the range estimate into the timer equation, node *2* will have a shorter timer than node *1* and therefore it will forward the packet before node *1* gets a chance to do so. As a result, the chance of a packet arriving at the *destination* increases. Without range estimation, node *1* will forward first since it is the closer node.

It should be noted that in an actual implementation also the battery power indicator could be taken into this evaluation. This may be necessary since strong senders may not remain to be so during the lifetime of the network, as a result of being favoured.

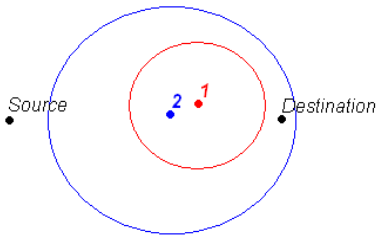

Figure 4: Importance of range estimation in the timer equation

The distance travelled by the packet (relative towards the *gateway*) can easily be calculated, since we know the (estimated) location of the *intermediate* (which is appended to the packet) and the *receiver*. A node can calculate the progress of a packet as follows:

packetProgress = distanceToGateway(intermediate) – distanceToGateway(receiver)

Note that the packet progress will always be positive since the *receiver* will only calculate a timer value if it is closer to the *gateway* than the *intermediate*, else it would have rejected the packet.
The node can now make an estimation about the desirability (and consequently the *priority*) of itself forwarding the packet and set its timer accordingly. We do this by multiplying the strength indication with the packet progress:

*priority = packetProgress * maxRange*

The relation between *packetProgress* and *maxRange* is 1:1 in this equation, this can be altered in a practical implementation. When we now divide a *constant* by the node's *priority*, a value for the *timer* is acquired which is high for low *priority* and low for high *priority*.

timer = constant / priority

Note that the timer values of different *receivers* are very likely to differ, because they all have different *packetProgress* and *maxRange* estimates. A packet's total delay in the network is directly related to the value of the *constant*. It should be chosen low enough such that the end-to-end delay requirement can easily be met, and high enough to prevent concurrent transmissions of duplicate packets.

5.2.3 Network traffic eavesdropping

The third issue concerns a *receiver* that has correctly received a packet, placed it in its broadcast buffer and is waiting for its timer to expire. In the mean time, the *receiver* receives the same packet again, but only from a different *intermediate*. Based on this new packet it can make a judgement about the progress of the packet in the network and possibly alter its own *priority* (and consequently, its *timer*) to forward this packet. We distinguish two cases, the *intermediate* is either closer to or further away from the *gateway*. In the first case remember the ideal behaviour that GZOR tries to approximate. Apparently, the packet has already travelled past the *receiver*, so another transmission by this *receiver* would be unnecessary. The *receiver* thus responds to this case by removing this packet from its broadcast buffer and stopping its timer (backing off). In the second case, another node has broadcast the packet although it was not the closest receiving node. Since the new *intermediate* may have reached nodes closer to the *gateway* which did not receive the packet before, the *receiver* has to recalculate its timer based on the location of the new *intermediate*. This prevents nodes, reached by earlier transmissions, to broadcast before closer nodes have the chance, simply because their timer was set earlier.
In a network with randomly spread gateways, it is possible that a packet en-route to a certain *gateway* comes along another gateway. This gateway can of course also accept the packet and send it to the base, in which case no more transmissions in the network are needed. Problem is that in

this case there are still nodes with the packet in their broadcast buffer with sharp timers. This of course also happens when the intended *gateway* has successfully received a packet and surrounding nodes are uninformed about this event. We circumvent this problem by implementing gateways to broadcast the fact that they received a packet. Neighbour nodes that hear this transmission can consequently remove the packet from their broadcast buffer.

5.2.4 Network flood prevention

The fourth issue is a more theoretical problem. Consider a node with a large distance from its *gateway*. We would like packets from this node to travel to the *gateway* in close-to straight path (a near optimal route). Packet forwarding based on distance alone can cause unwanted effects. In theory, a packet can travel completely around the *gateway*, while still coming closer to it with every single hop. In the worst case almost all nodes around the *gateway,* closer than the detecting node, could receive this message at some time and forward it accordingly.

To prevent this kind of behaviour, we add another consideration constraint for a *receiver* on which a packet can be rejected. Because the location of the *sender* and *gateway* are known to the *receiver*, the *receiver* can draw a straight line between these points and calculate its distance relative to this line, which we define the *distanceToSenderGatewayPath*. This line from sender to gateway resembles the optimal path and forwarding nodes should not differ from this path by an amount of distance such that a single transmission would not be able to reach nodes close to this path. Therefore, we restrict the receiving node from forwarding if its *maxRange* value is smaller than the distance from the *receiver's* location to the *sender-gateway* path. This is illustrated in Figure 5.

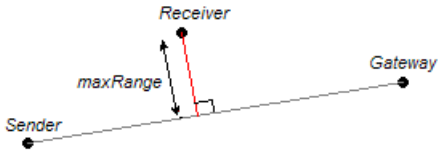

Figure 5: A node may only consider forwarding if it lies within maxRange to the sender-gateway path

Figure 6 summarizes the decision a *receiver* will make upon packet reception. *Receiver r* will only forward a packet intended for *gateway D* if the original *sender* lies in the dark grey area and the *intermediate* lies either in the dark or light grey area. Packets from the white area will always be dropped. If this packet already was in the broadcast buffer, it will be removed and the timer will be cancelled.

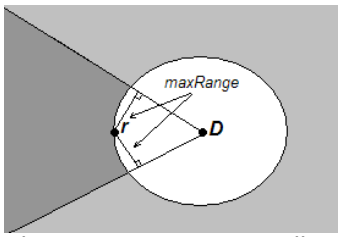

Figure 6: Forwarding consideration of GZOR

5.2.5 Pseudo code

The solutions above are now combined with the global concept. The following pseudo-code describes the behaviour of a single node when it is triggered by the reception of a packet:

```
if (rejectionList.contains(packetNumber))
        reject;
elseif (broadcastBuffer.contains(packet))
        if (distanceToGateway(intermediate) > distanceToGateway(receiver)
        & intermediate != gateway)
                packetProgress = distanceToGateway(intermediate) – distanceToGateway(receiver);
                priority = packetProgress * maxRange;
                timer(packetNumber) = constant / priority;
        else
                broadcastBuffer.remove(packet);
                rejectionList.add(packetNumber);
                timer(packetNumber).stop
        end
else
        if (distanceToSenderGatewayPath < maxRange
        & distanceToGateway(intermediate) > distanceToGateway(receiver)
        & intermediate != gateway)
                packetProgress = distanceToGateway(intermediate) – distanceToGateway(receiver);
                priority = packetProgress * maxRange;
                timer(packetNumber) = constant / priority;
                broadcastBuffer.add(packet);
        else
                rejectionList.add(packetNumber);
        end
end
```

6 Implementation of GPSR

The currently most evaluated geographic routing algorithm for WSNs is GPSR (Greedy Perimeter Stateless Routing) [KaK00]. This algorithm theoretically offers delivery guarantees while only maintaining and sharing position information locally between nodes in a network (node information does not travel further than a single hop). It was originally designed to function on a unit disk model, which at that time seemed to be an accurate model of a wireless network. Later research, however, showed the opposite. Simulations on arbitrary connected graphs showed that the GPSR algorithm contained possible deadlocks or failed to deliver a packet even when a path from sender to destination was present. Several fixes and variants were proposed to solve these problems.
Since this algorithm is widely examined and offers a high delivery rate while maintaining scalability, it seems a logical choice to compare the performance of GZOR with GPSR.

6.1 Basics of Greedy Perimeter Stateless Routing

The principle of GPSR is quite simple. Every node knows its own location and also that of its neighbours, which is stored locally in a table. From a global perspective, the combination of these tables form a connectivity graph. By a beacon mechanism the nodes proactively communicate with each other to keep their tables up-to-date. The GPSR solution belongs to a subclass of geographical routing algorithms, all of which combine greedy forwarding and face routing. All algorithms belonging to this subclass consist of a set of individual components, each of which have many variants (e.g. the GOAFR+ algorithms family [Kuh03]). The basic idea, however, is always the same.
When a node receives a packet, it evaluates the location of the packet's destination and forwards the packet to the neighbour closest to the destination. This process is called greedy forwarding. A problem arises when the forwarding node does not have a neighbour closer to the destination than itself; this situation is called a local maximum. When such a maximum occurs, a backup algorithm (face routing) is initiated to route the packet further and eventually recover from this local maximum. This backup algorithm operates on a planar subgraph of the connectivity graph. The packet traverses along this planar graph until it comes across some node closer to the destination than the local maximum. At such a node, greedy forwarding can proceed. Since the backup algorithm operates on a planar graph, nodes must decide which of the neighbours in their table are part of the planar graph and which are not. This planarization process requires a third algorithm, which must be initiated after each alteration of the neighbour table.
The GPSR solution alternates between two routing algorithms, hence it must have information about which one to use. This is stored in the packet itself, which can be in greedy mode and perimeter mode. On encounter of a local maximum, the packet switches from greedy to perimeter mode. On encounter of a node closer to the destination than the location where the perimeter mode was entered (which must be stored in the packet), the packet switches back to greedy mode. An example is given in Figure 7, the destination of the packet is node *D*. At node *a* the packet is in greedy mode. It forwards the packet to the neighbour closest to *D*, which is node *x*. Node *x* does not have a closer neighbour and switches the packet to perimeter mode. The packet is now routed by a backup algorithm via *c* and *d* to node *e*. Since node *e* is closer to node *D* than node *x*, at node *e* the packet is toggled back to greedy mode, after which it is forwarded to node *f* and finally reaches its destination.

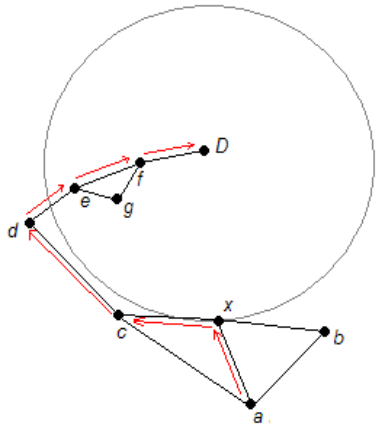

Figure 7: Example of the greedy and face routing combination

Since GPSR forwards every packet only to a single node, some MAC layer support (acknowledgements, retransmissions) is required to provide guaranteed packet delivery. In the following sections the different components of the algorithm are folded out. Also the decision is clarified about which variant of each component is chosen for the qualitative comparison with GZOR.

6.2 Neighbour detection and MAC layer support

Unlike GZOR, GPSR contains of several types of packets. These packets can be divided into two categories: pro-active topology packets and reactive (detection induced) packets. Although the size of these packets may vary (and consequently, the chance of bit error occurrence), for simplicity they are all treated the same in our simulations.

Nodes in GPSR must be aware of the position of their neighbours. In order to maintain their information on their neighbours, they periodically broadcast beacon messages (this is the proactive component). On receiving such a beacon, neighbours reply with their position information. When the beaconing node receives such a reply, it stores the sender and its position in its own neighbour table. The original GPSR algorithm was also designed for mobile (moving) nodes with dynamic positions. In between beacons, nodes could move in and out of range. Therefore upon reception of a data packet of an unknown node, this node is added to the table. When a transmission to a certain node failes, it is removed from the table. Detection of failed transmissions requires MAC layer feedback.

During simulation the nodes do not change their tables upon new node detection or transmission failure.

Simulations of GPSR done by other research groups [KaK00], [Kim05] all assume bidirectional links and discard links with a Packet Reception Rate less than 70% (nodes with a packet failure probability of 30% or more between each other are not considered to be neighbours). We have implemented this in our simulator as follows. Each node broadcasts ten beacons (serialized), when it receives seven or more replies from an individual node, it adds that node to its neighbour table. Afterwards, all nodes correspond with all of their neighbours to assure that every pair of nodes both agree on whether they are neighbours or not. This avoids asymmetric neighbour pairs. Nodes consider themselves neighbours when at least one of them has received seven beacon replies from the other. Note that this is an optimistic evaluation of link quality, nodes could also drop neighbours if they are unknown to them. Both approaches have been simulated and the latter showed a poorer performance. The communication between nodes for reaching agreement on the neighbour status is considered to be 100% reliable, which in practice can be achieved by MAC layer retransmissions.

The reactive component of the algorithm involves nodes sending data packets only to an individual neighbour. Because links with a packet failure rate of up to 30% are used, the algorithm heavily relies on a MAC layer providing reliable transport on such a link. This is implemented in the following manner. When a node receives a data packet, it sends an acknowledgement to the sender. When the sender does not receive this acknowledgement, it will conclude that the transmission has failed and retransmit the packet. The sender will retransmit three times at maximum, after which it drops the packet.

6.3 The right-hand rule and face routing

If greedy routing fails, a backup algorithm is activated to route the packet further. This algorithm is called face routing and there are many different variants available. They all have one thing in common, they route along the graph using the right-hand rule. This rule is basically a strategy to find the exit of a maze. When the right hand is continuously placed on the wall while walking forward through the maze, the exit will always be reached (in case the maze does not contain loops). In a graph, the right-hand rule implies that the next hop is the first node counter-clockwise from the previous visited node. This means walking through links as if it were corridors of a maze. When applying this rule in a graph, it will automatically traverse along the interior of a closed polygonal region in clockwise direction, which is illustrated in Figure 8. Such a closed region is called a face. Every graph has several inner faces and one outer face, as can be seen in Figure 8. Alternatively, the left-hand rule can be applied to traverse along a face in counter-clockwise direction. A packet can be

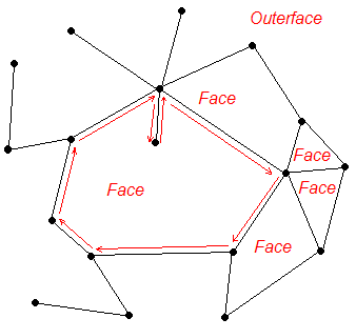

Figure 8: Example of face traversal according the right-hand rule

routed across a network along the interiors of a set of adjacent faces, using the right-hand rule and changing to new faces according some set of face-change rules. A sequence of visited hops is called a perimeter.

6.4 Planarization of the connectivity graph

When a packet encounters a local maximum, greedy forwarding cannot take it any further. To solve this problem, the network needs to switch to another routing algorithm (perimeter mode) to route the packet around the network void. This can be done by face routing according to the right-hand or left-hand rule. Face routing is a well-known class of graph routing algorithms which guarantees delivery on a planar graph. A graph is planar when it has no crossing edges (links) and all links are symmetric. So in order to apply this algorithm, the connectivity graph as a whole has to be planar. This means that nodes have to decide locally which links belong to the planar graph and which do not. There are several algorithms to do so, the most famous are the Relative Neighbour Graph (RNG) and the Gabriel Graph (GG) algorithms. These algorithms both guarantee to provide a connected planar graph, under the assumption that all nodes have an equal transmission range (unit disk model).

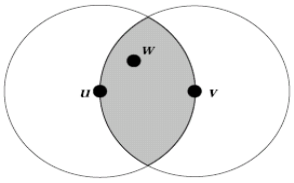

Figure 9: An edge uv is only included in the RNG if there is no witness w in the grey area [KaK00]

In RNG every node considers all of its neighbours and assesses if this neighbour should belong to the planar graph. It does so in the following manner, which is also illustrated in Figure 9. Node *u* removes the link to node *v* from the planar graph, if and only if it knows another node *w* of which the distance to *u* and to *v* is smaller than the distance between *u* and *v*. Such a node *w* is called a witness.

In GG also every link is evaluated, but in a slightly different manner, which is illustrated in Figure 10. Node *u* defines a virtual point *m* at the middle of the link between *u* and *v*. Node *u* removes the link to node *v* from the planar graph if and only if it knows another node *w* that is located closer to this point *m* than *u* itself. Because the witness area in a RNG is larger, the RNG is a subset of the GG. An example of a graph planarized by RNG and GG is illustrated in Figure 11.

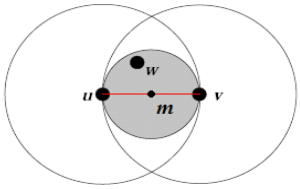

Figure 10: An edge uv is only included in the GG if there is no witness w in the grey area [Kak00]

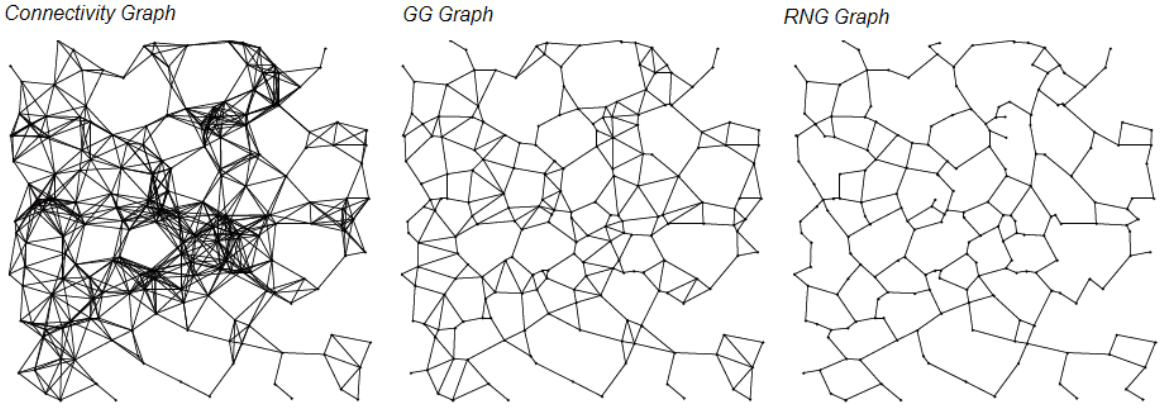

Figure 11: Example of graph planarization [KaK00]

As mentioned above, both algorithms rely on the assumption that all nodes have equal transmission ranges, implying that both nodes assessing link *uv* will see witness *w*. Empirical results, however, show that this assumption does not hold in practice, as described in chapter 4. When for instance node *w* does not have an equal transmission range, the situation could occur where node *u* does see witness *w* and removes to the link *uv* from the planar graph, while node *v* does not. Anisotropic transmission ranges and localization errors can induce crossing edges or asymmetric links in the RNG or GG graph, causing such graphs not to be planar. Routing algorithms relying on the fact that the graph is planar will fail in this case. There is a well-known fix to this problem, which is the Mutual Witness (MW) extension. In MW, nodes *u* and *v* will remove link *uv* based on witness *w* if and only if they both see this witness *w*. This means that nodes *u* and *v* have to verify this with each

other by some communication before removing a link from the planar graph. This fix effectively solves the problem of asymmetric edges in the planar graphs, but does not succeed in removing all crossing links. Note that removal of some crossing links could disconnect the graph. These links do not impose failures in face routing protocols, which is proven in [Kim05]. Links that do cause loops are named Loop-Inducing Cross-Links (LICL) [Kim06].

A different algorithm exists which is able to remove all LICLs. This is the Cross-Link Detection Protocol (CLDP) [Kim05]. CLDP is not based on RNG or GG and is designed to operate on an arbitrary graph. All nodes send probes along every link, which are redirected by the right-hand rule until the probe is again received by the original sender. When the probe contains information on a crossing link, it removes this link from the planar graph only if this does not disconnect the graph. Problem with this algorithm is that probes are not local, but travel along the network. This means that the original concept of GPSR, where topology information propagates only a single hop, is abandoned. Non-local pro-active topology overhead seriously threatens the scalability of GPSR. In [Kim06] Lazy Cross-Link Removal is proposed, which is a variant of CLDP. In this algorithm the planar graph is produced by RNG (or GG) with the MW fix. A node will only send CLDP probes on-demand after a transmission failure, reducing the large proactive overhead of CLDP. In [Sea07] it is proven that the production of a planar graph while only sharing local topology information is impossible when the unit disk model does not hold.

Although RNG and GG do not guarantee the graph to be planar, combined with the MW fix both succeed in removing all asymmetric links and a large portion of the cross-links. Together with the fact that CLDP does not offer localized topology information exchange, RNG and GG seem to be the better option from a scalability point of view. Since RNG and GG do not guarantee the removal of all cross-links, a Time To Live (TTL) variable has to be added to the packet so it can be dropped in case of a loop. All three algorithms (RNG/MW, GG/MW, CLDP) were simulated in combination with the GPSR algorithm. The performance of the reactive component of the face routing algorithm did not greatly vary. The results are presented in Figure 13 and 12 (simulation details are in appendix A). As expected, CLDP shows a slightly better performance. In our simulations GG performed a little better than RNG in terms of transmission count. This is caused by the fact that GG leaves more links in the planar graph than RNG, allowing shorter routes.

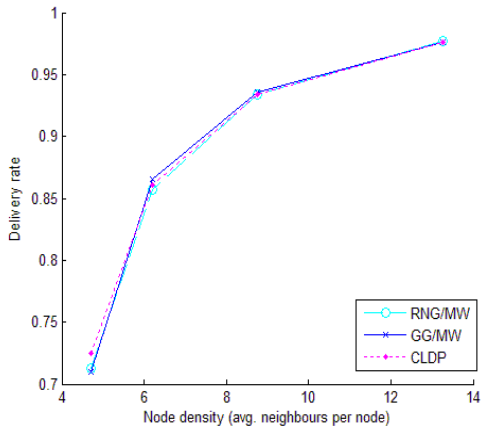

Figure 13: Delivery rate comparison of RNG, GG and CLDP

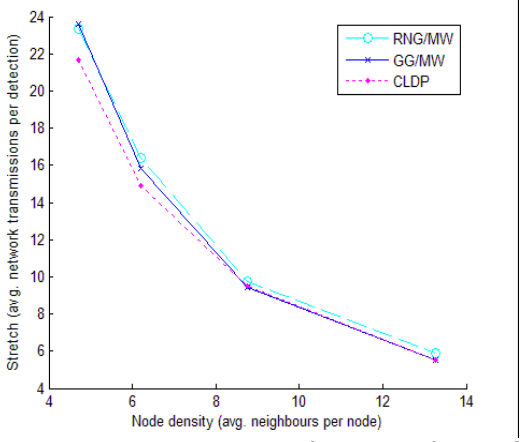

Figure 12: Average stretch comparison of RNG, GG and CLDP

6.5 Face routing algorithm: First intersection

Individual faces are traversed by the right-hand rule. In order to make progress towards the destination, a face algorithm must have a strategy on when and where to change to a different face. There are various strategies to do so, which are examined and compared in [Fre06] and [KiG05]. When the perimeter mode is entered, a line can be drawn from the current node *x* to the final destination *D*. This line intersects a sequence of adjacent faces, this is illustrated in Figure 14. There are basically two classes of face change rules; the first is based on the line *xD* and the intersection of a node-to-node link with this line, the second is based on the distance to *D*. The original face routing of GPSR is intersection based. A node selects a new hop according to the right-hand rule. When the

link to this new hop intersects line *xD*, the hop counter-clockwise from this hop is chosen. This corresponds to changing the face. All faces visited in this manner are intersected by line *xD*. An example of this process is given in Figure 15. The first link on a new face is recorded in the packet, so when a packet travels around a face without finding a point where it can change to a new face, it will come across the recorded link. If this happens, the packet will be dropped, which should only happen where no path exists between *x* and *D*. Other face routing algorithms adopt strategies where the packet first travels completely around a face before a decision is made about the best location for a face change. Other strategies involve the alternation of right- and left-hand rule usage.

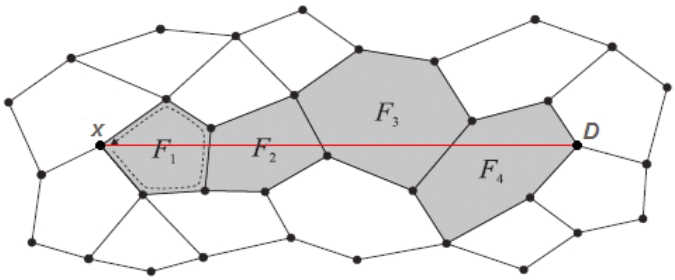

Figure 14: Line xD intersects a set of adjacent faces

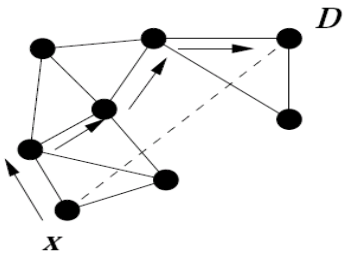

Figure 15: Example of First Intersection right-hand rule face routing

The face routing algorithm of GPSR is proven not to be loop-free [Fre06], though rather efficient since it does not involve a complete face traversal. Combined with the greedy algorithm, it does offer guaranteed delivery in a planar graph under the unit disk model.

Another problem with the GPSR face routing algorithm is that at the border of the network the outerface can be chosen. This means that packets will be routed along the entire perimeter of the network. In a large network, the amount of transmissions involved in such an event is enormous. An example of this event is illustrated in Figure 16, the red line indicates a right-hand rule route, the blue line is the left-hand rule route. The TTL variable, as described in the previous section, limits this behaviour. A smart choice whether to route right- or left-handedly can also minimize the occurrence of outer face traversal. Since the right- and left-hand rule are basically the same algorithm (only mirrored), adopting the left-hand rule within the GPSR algorithm does not change the fundamentals of the algorithm. The choice whether to route according to the right- or left-hand rule is made on the following basis. On entering perimeter mode, the right-hand rule selects the first node counter-clockwise to the destination. If the angle between this selected node and the destination is smaller than 180 degrees, the right-hand rule is adopted. If it is larger, the first node in clockwise direction from the destination is chosen and the left-hand rule will be applied. Although this reduces the occurrence of outer face traversal, it does not solve the problem. GOAFR+ [Kuh03] uses a different face routing algorithm, which limits the distance a packet is allowed to travel away from the node where the perimeter mode was entered. When this distance is crossed, the packet turns around and the algorithm toggles between the right- and left-hand rule (to stay on the same face).

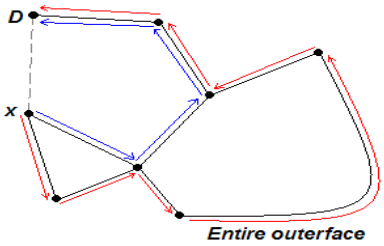

Figure 16: The right-hand rule (red) induces an outer face traversal

Summarising, the face routing algorithm of GPSR is not loop free nor has a bounded worst-case performance. Surprisingly however, together with greedy forwarding it is very efficient according to [Kim05]. This is because this algorithm greedily changes face at the first intersection with line *xD*. Furthermore, face changes in practice seldom occur, most of the time a node closer to the node where the perimeter mode was entered is encountered on the very first face. Besides that, in our simulations, a packet traversing the outer face of the network is likely to encounter an unintended gateway, which is also willing to accept the packet.

7 SIMULATION

7.1 Set-up and parameters

As described in the research methodology (chapter 4) it is very important to have a clear picture on the expected performance of the GZOR algorithm in a real-world implementation. Therefore we have aimed to create a simulation process closely resembling conditions argued to be encountered in the security solution.

7.1.1 Lower bound: Dijkstra's algorithm

Simulation of both routing algorithms produces information on the average stretch (average number of transmissions induced per detection) on some random topology. To place this information in perspective we would like to compare it with the optimal average stretch. The latter shows what the network is capable of and can be seen as a lower bound. This can be done by Dijkstra's algorithm in combination which an ETX [Cou05] evaluation of every link in the network. The ETX value is the average amount of transmissions required to get a packet across a link. This assumes the sender somehow knows if packets were successfully received or retransmission is needed. For example, if a link has a packet loss of 20%, on average $1/(1-0.2)=1.25$ packets have to be sent to achieve a delivery rate of 100%. The ETX value of every pair of nodes is calculated, which results in a graph were paths on reliable links have a lower value. On this graph Dijkstra's algorithm can calculate the shortest path between two nodes and the corresponding cost of the path. The latter represents the optimal average amount of transmissions the network has to do to get a detection from some node to its gateway (with 100% reliability). An example of this analysis is given in Figure 17. In this example *a* shows the graph with the packet reception rates of every link, *b* shows the corresponding ETX values and *c* shows the optimal path and the corresponding amount of transmissions.

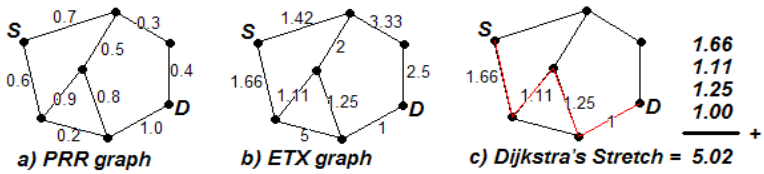

Figure 17: The lower bound of the communication stretch from S to D is 5.02 network transmissions. This is the optimal average value necessary to achieve reliable transport.

Especially with an air deployment network there will always be nodes that lie too far away from any other node to communicate with. If a node does not have a neighbour with an ETX valued link less than 50 (this corresponds to 98% packet loss), it is considered to be lost and will not participate in the routing simulations.

It should be noted that it is impossible to design an algorithm with the same performance as the lower bound, since this would imply a node can somehow 'know' whether its transmission was successfully received, without feedback from the receiver.

7.1.2 Node arrangement/deployment

For performance evaluation of the routing algorithms, it is important to create a realistic node arrangement. This strengthens later conclusions on the applicability in a final solution. The objective of this research is to create a routing algorithm for a surveillance network capable of performing in a hostile environment. In this type of environment it is not always possible to have full control of the arrangement of the nodes, resulting in a distribution of nodes not optimal from a routing standpoint. When considering an air drop, the nodes may be scattered in a heterogeneous way with large connectivity holes. This may have severe impact on the network performance in terms of routing.

The Thales WSN simulator supports several models for deployment of the network. These include the scenarios most likely to be implemented in a final real-world solution, which are a manual deployment and an air drop. Details on the deployment models can be found in Appendix B. For the evaluation of the algorithms this research has focused on the model which combines multiple air deployments. This model corresponds with the intended purpose of the implementation and also partly resembles the grid layout used in correlated research [Kim05] [Whi07].

The used implementation of the Multiple Air Deployment Model (MADM) consist of four overlapping air deployments. The centres of these deployments are placed on the corners of a square with 100m sides, which is illustrated in Figure 18. Each of the deployments consists of 346 nodes, which are subdivided in 85% normal nodes, 10% GPS-equipped anchors and 5% GPS-equipped gateways. In the centre of the square the four deployments overlap, which causes the entire network to form a closed square with varying densities. This closely resembles a network where nodes are randomly placed inside a predefined square grid.

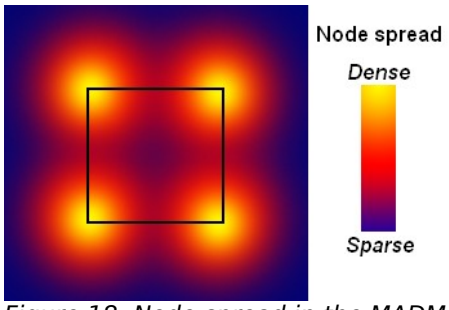

Figure 18: Node spread in the MADM

Figure 19 shows the connectivity graph of a generated network. This figure shows that in the overlapping zones the chance of connectivity voids increases.

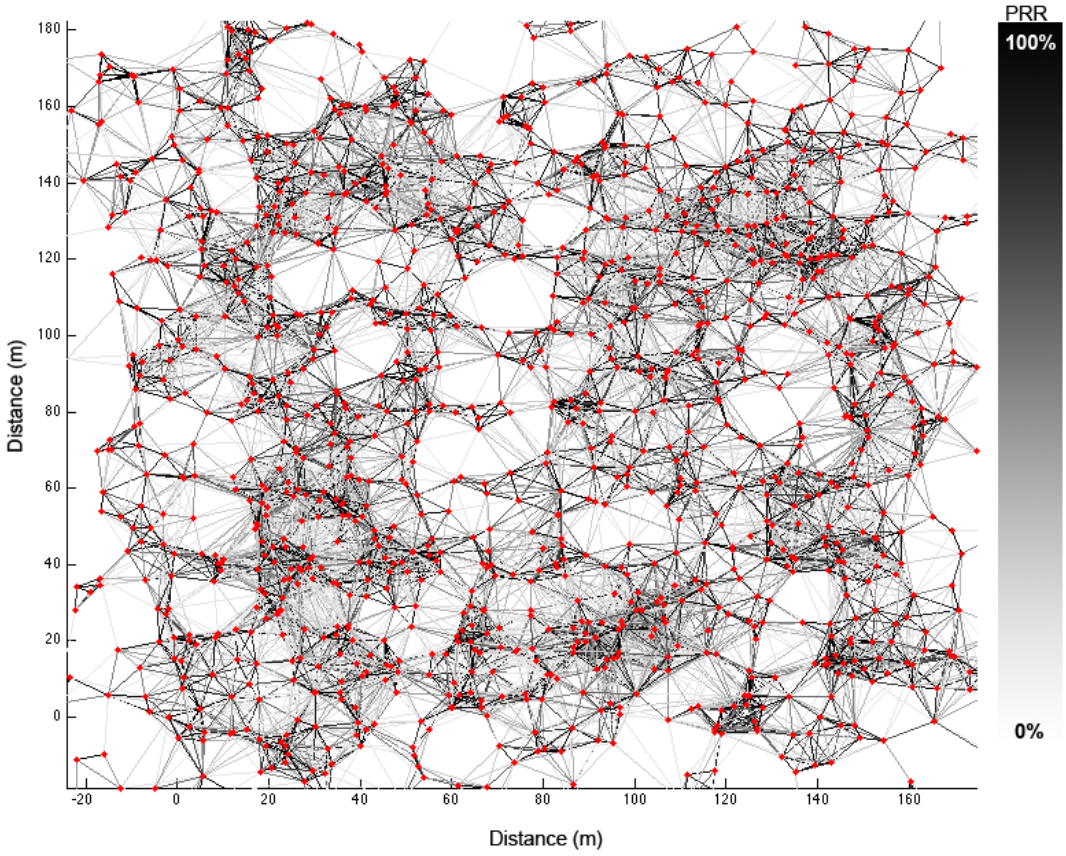

Figure 19: Connectivity graph showing the packet reception rate between node pairs

7.1.3 Localization simulation

Both GZOR and GPSR base their decisions on the estimation of their own location and the information they receive from other nodes concerning their estimated location. Localization is never error-proof, especially if it is a result of ranging, which is very unreliable [Sla07]. Localization errors can have a significant impact on geographic routing algorithms [Sea04]. Therefore a localization algorithm is included into the simulation. Before simulating any geographic routing algorithm, a localization algorithm is executed on the deployed network. The used algorithm is a simplified

version of the algorithm described in [Sla07], which is a result of previous research within Thales. Normal nodes acquire their position information through localization and therefore will have realistic deviations from their actual position. Details on this process can be found in Appendix C.
Figure 20 shows an example of a localised network. The red dots are the actual position of the normal nodes. The black lines indicate the position the nodes have acquired through the localization algorithm.
As a result of the localization process nodes have an estimate on their position. Since gateways also conduct in this process as an anchor, the nodes are aware of the position of the closest gateway. The latter information is used by GZOR and GPSR to determine the destination of their packets.

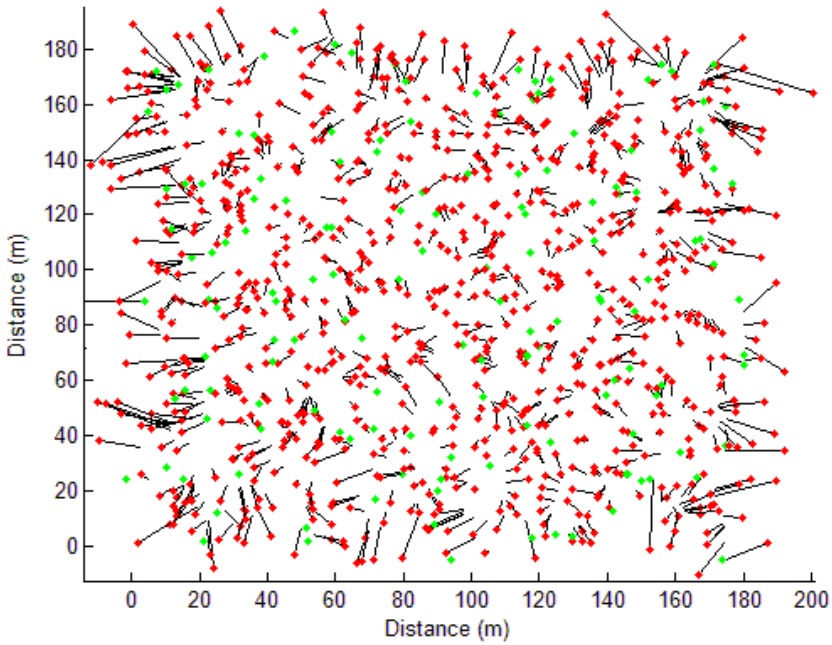

Figure 20: Example of a localized network. Red and green dots are normal and anchor nodes respectively. The line points to the node's estimated location

7.1.4 Operational nodes definition

The simulation metrics in chapter 4 state that the performance of the *operational* nodes is measured. The operational nodes are defined as all nodes containing the following properties:

- A node is either a GPS-equipped anchor or has found enough (at least three) anchors during the localization process to estimate its own position. Nodes without position estimates can not participate in a geographic routing algorithm.
- A node is either a gateway or has acquired information on the position of at least one gateway during localization. Note that a node without knowledge on the position of a gateway does not know where to send packets. Therefore these nodes are not allowed to broadcasts detection packets.

Nodes which do not have these properties after the localization process are considered unconnected. They do not send detection packets and thus their performance is not reflected in the delivery rate statistics. However, unconnected nodes may participate in the routing process if they have a position estimate, since they are perfectly capable of forwarding packets. Figure 21 shows an example of a generated network according to the MADM where the localization algorithm is executed. Cyan nodes do not send detection packets. The blue, green and red nodes are operational nodes.

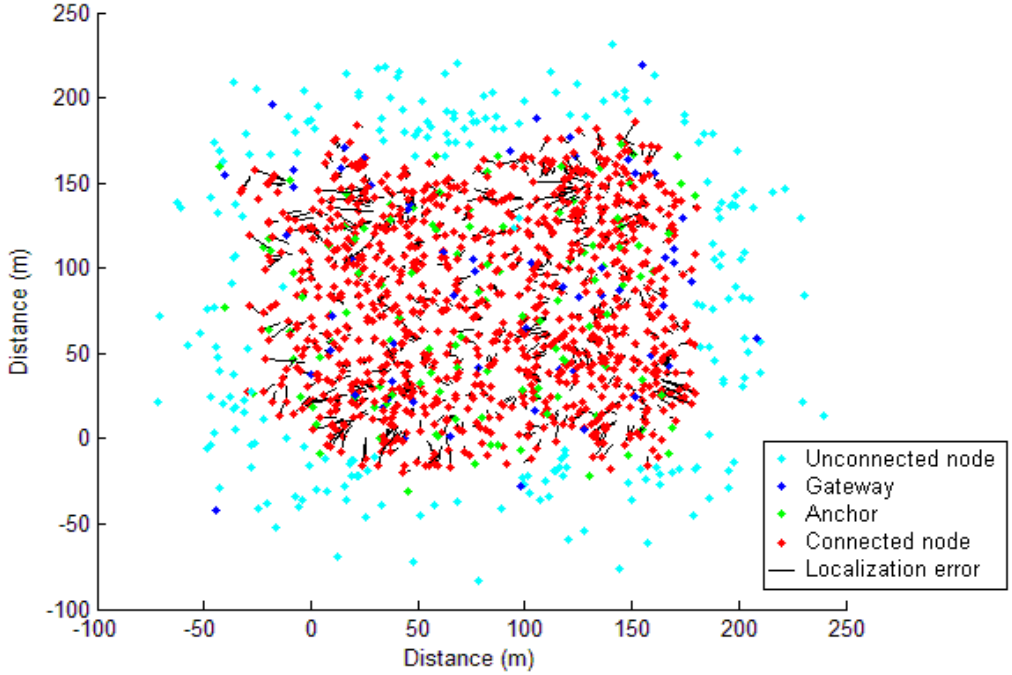

Figure 21: Example of a localized network deployed by four overlapping single air deployments (MADM)

7.1.5 Asynchronous sleep

The GZOR algorithm is designed to support asynchronous sleep cycles. During such a sleep cycle a node completely turns off its radio. This method can significantly increase the total lifetime of the network. Both algorithms were tested with increasing percentages of time in sleep mode, ranging from 0 to 50%. This means in simulation that nodes will reject packets when they are asleep. This is implemented as a uniform stochastic function where e.g. a 50% sleep cycle corresponds with a 50% chance of packet rejection. This function will not be active when nodes are either waiting for a reply (in GPSR) or have a live timer (in GZOR), thus the nodes are constantly awake during such events. Gateways never enter a sleep mode, anchors are considered to be normal nodes after the localization process.

7.2 Quantification

The quantification of both algorithms GZOR and GPSR is done according a set of network simulations where the algorithms are stressed. The behavioural analysis is performed on individual network deployments where the handling of both algorithms is compared in characteristic situations.

7.2.1 Network topology generation and density variation

Section 7.1 described the generation and composition of the network. Once a network topology is generated, the positions of all nodes are static throughout the simulation. A common way in WSN research to present the performance of an algorithm is to show the delivery rate against a variation of network densities. The network density is defined as the average number of neighbours per node (Section 4.2). Variation of the density can be done in several ways. Examples are: Expansion of the space between nodes, reduction of the amount of nodes, reduction of the radio power, variation on the radio path loss exponent. All methods are viable, but some are more tangent with reality than others. The first two alternatives do not agree with the previous stated static network. The latter two, variation of radio power and path loss, represent battery depletion and weather condition variations (e.g. humidity) respectively. Both variables are part of the same function used by the simulator to calculate signal strength. The path loss variable is equal among all nodes, while the radio power varies, therefore the first is chosen to achieve the variation in density.

Since the network only does the localization process at the initialization of the network, this remains unchanged throughout the simulation. Note that localization with different path loss variables would yield different results, therefore localization is performed with the mean of all simulated path loss exponents.

7.2.2 GZOR quantification

Since the GZOR algorithm allows packet duplication, routing simulation is not straightforward. Multiple nodes might decide to forward a packet at different time intervals. On reception of other packets, nodes might cancel their timers. Since nodes are event driven, they will only consider packet forwarding (or timer adjustment) after a successful packet reception.
The GZOR quantification process involves each operational node to send 10 packets to the closest known gateway.
The simulator handles only one unique packet at a time, although copies of that unique packet may exist. From the moment the packet is injected into the network all transmissions performed by nodes are counted. The simulation runs until there are no more nodes with live timers. Afterwards the simulator determinates if there was a gateway among the receiving nodes. If this is the case the packet is counted as received, else it is lost.
Summarising, every node sequentially transmits a fixed amount of packets on the network, which corresponds with a fixed amount of detections. The percentage of these packets that actually reach a gateway is recorded (this is the delivery rate) together with the total amount of transmissions the network must do as a result of the detections (stretch).
This process will be repeated onto 25 randomly created network topologies, the mean values form the data points in the resulting figures.

7.2.3 GPSR quantification

The GPSR simulation consists of two phases, the network table creation and the detection routing phase. During the first phase every node broadcasts 10 beacon messages (section 6.2). Neighbour nodes reply on the beacon message. When the amount of beacon-replies received from a single node exceeds the threshold value of 7, that node is added to the neighbour table. Afterwards the mutual witness algorithm is initiated to ensure all nodes are in agreement whether they are neighbours or not. This is followed by the Gabriel Graph algorithm where each single node builds its planarized neighbour table (section 6.4). The (pro-active) transmissions performed during the first phase are not included in the calculation of the stretch. Only reactive communication contributes to the stretch.
The second phase involves the same 10 detection packets per operational node as in GZOR. The GPSR algorithm only sends packets to a single receiver. Therefore GPSR requires MAC layer support in the form of retransmissions after a failed transmission. All retransmissions and ACK messages are counted. A node will retransmit a packet when it fails to receive an ACK message up to 3 times.
If the intended receiver indeed receives the packet and is a gateway, the packet will be accepted. The percentage of packets accepted by gateways is the delivery rate.
Since GPSR is not loop-free (see section 6.4) a packet is only allowed to travel a fixed amount of hops. The Time-To-Live value is incremented at each hop and is set to 50. When this value is reached the next hop will drop the packet, ensuring nodes will not forward deadlocked packets endlessly.
This process is performed on the same 25 random topologies as the GZOR simulation.

7.3 Behavioural analysis

The behavioural analysis is done by individual experiments where the journey of a packet through the network from source node to gateway is monitored step by step. This is done with both algorithms, where the same source node in the same network is sending the same packet. The only difference is the routing algorithm.
The analysis tools for simulation feedback are custom made per algorithm as a part of this research. The feedback information provided by the simulator is not straightforward however. Especially the GZOR algorithm is not easy to interpret since it allows packet duplication and depends on timers. Therefore the following sections explain how the simulator information should be interpreted.

Each experiment starts with an optimal path analysis by Dijkstra's algorithm indicating the optimal path from source node to gateway. An example of such an analysis is given in Figure 22. In this example the cyan node *4* is the source node. All gateways are blue, all operational nodes are red. Node *4* needs to send a packet to its known gateway *826*, the only knowledge node *4* possesses is the location of gateway *826*. The green lines indicate the optimal path to do so. The values on the axes are in meters.

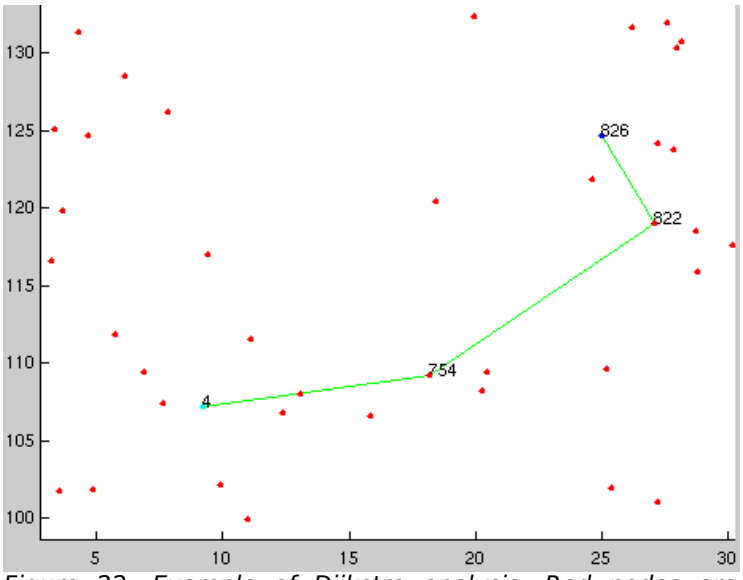

Figure 22: Example of Dijkstra analysis. Red nodes are operational nodes, the source node is cyan, the blue node is a gateway

7.3.1 GZOR simulator feedback

Initialization

The GZOR experiments basically use the same colour pattern as the Dijkstra experiments. The source node is cyan, gateways and normal nodes are blue and red respectively. A transmission of an individual node is indicated with a black * around that node.

Broadcast and reception

Since nodes in GZOR are unaware of their neighbours, node *4* simply broadcasts the packet. All nodes that have received the packet are indicated with a purple circle around them (e.g. Figure 23).

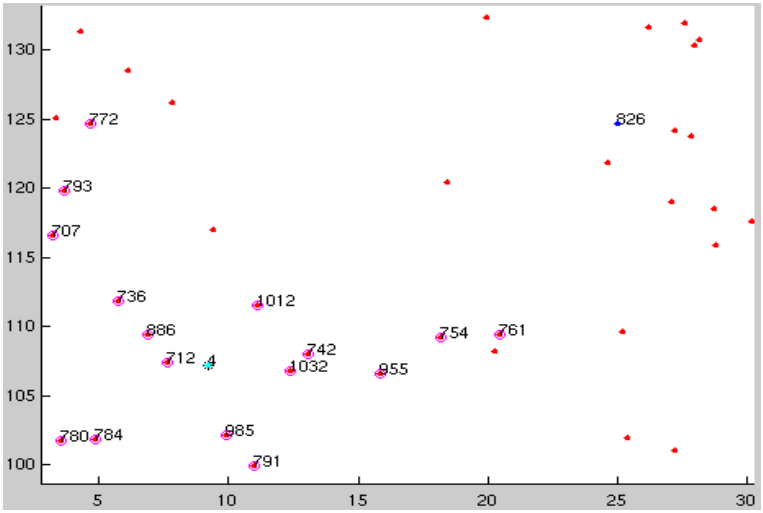

Figure 23: Example GZOR broadcast by node 4 and reception of the surrounding red nodes. Nodes with magenta circles have received the packet.

Decision and timer expiration

The nodes which have successfully received the packet will now decide what to do with it. Some nodes will decide that it is not favourable for them to forward the packet, these nodes become black. The rest of the nodes calculate their timer value for packet forwarding. The node of which its timer expires first forwards the packet. When the timer of a node expires and it transmits the packet accordingly, it becomes red with a black * surrounding it (e.g. node *1032* in Figure 24).

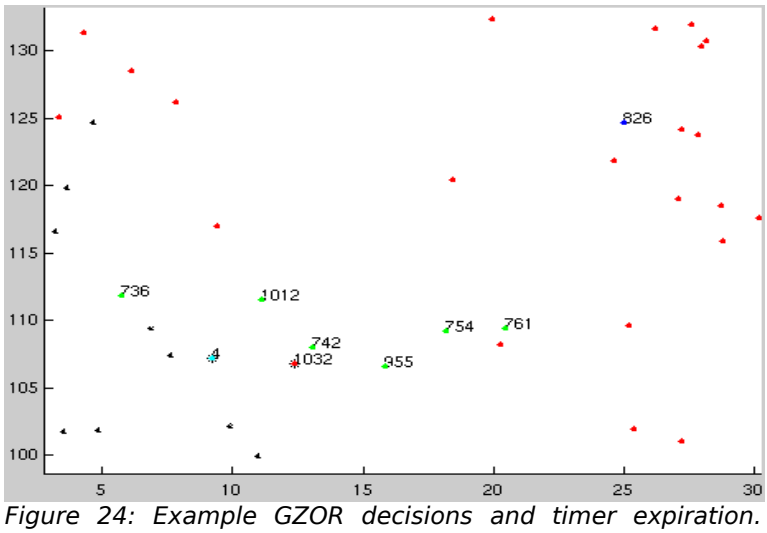

Figure 24: Example GZOR decisions and timer expiration. Black nodes have received the packet but dropped it, green nodes have live timers.

Final analysis

When a gateway receives a packet it will signal its surrounding nodes to cancel their timers. The process repeats until there are no nodes left with live timers. When this happens, the packet is either received by a gateway or lost.

The simulator provides feedback of the packet's journey through the network. An example is given in Figure 25. The green lines indicate the direct path of the packet from source to gateway. The black lines belong to emerged multi-paths. When a node is connected by a black line, it has successfully received the packet and decided to forward it at some time during the experiment. We make a distinction between nodes that have actually forwarded the packet themselves and nodes that have had a live timer but cancelled it afterwards on reception of other packets. Forwarders are black, cancellers are red.
Notice that node *757* in this example indeed has forwarded the packet but without success. All receiving nodes of this transmission have decided not to forward it, so the branch stops. The location of the nodes indicate the direction of a line, since the packet is not allowed to travel backwards. For example, node *1012* has two lines towards itself (from node *4* and *1032*), and one line outwards (to node *761*). This means that on reception of the packet from node *1032*, node *1012* has recalculated its timer value. Note that this overview figure does not give information which transmissions have caused nodes to cancel their timers.

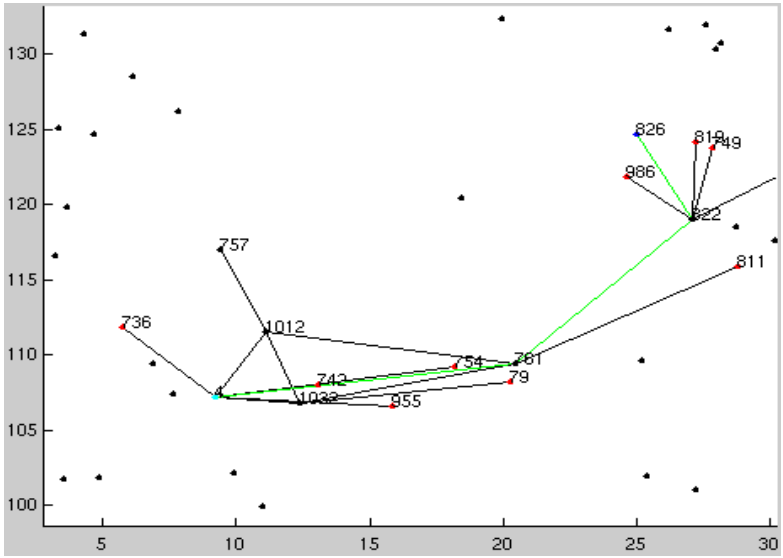

Figure 25: Example GZOR path analysis. The cyan node 4 is the source, the blue node 826 is the gateway. The green line indicates the path travelled by the first arrived copy of the packet. The black lines are multi-paths. Red nodes have had live timers, but cancelled them on packet reception of a more favourable forwarder.

7.3.2 GPSR simulator feedback

Initialization

The GPSR algorithm consists of two modes: greedy mode and perimeter mode. At initialization the packet is set to greedy mode. In order to analyse the behaviour of the GPSR algorithm it is important to have an overview of the routing tables of the nodes, because every routing decision is based on these tables. This includes both the neighbour tables required for greedy routing and the planarized neighbourtable required for face routing. The combination of all tables can be represented as a graph. Two nodes are connected by an edge in such a graph when they are in each others table. We use the same example of source node *4* and gateway *826*. Figure 27 and Figure 28 show the graphs formed by respectively the neighbourtables and planarized neighbour table of the nodes.

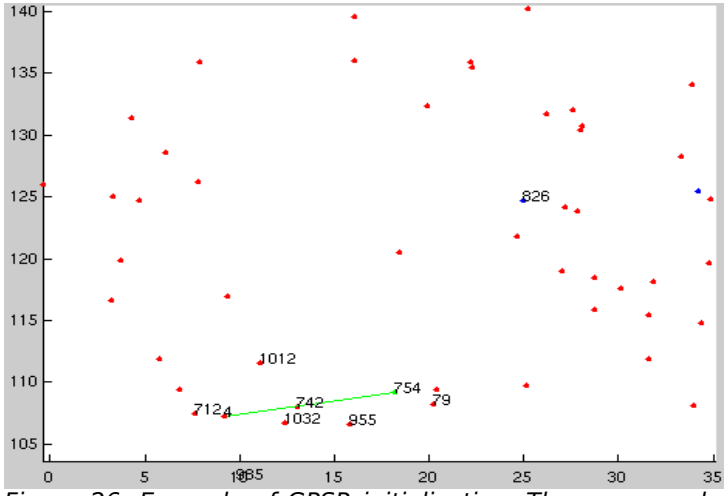

Figure 26: Example of GPSR initialization. The green node currently has the packet.

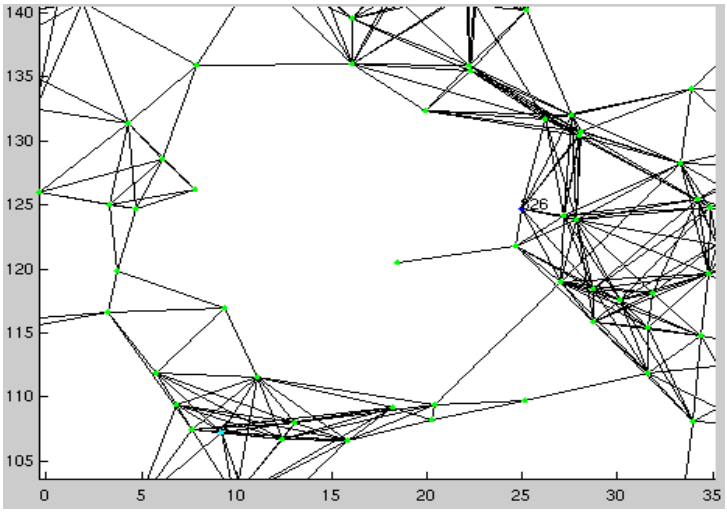

Figure 27: GPSR neighbour table graph.

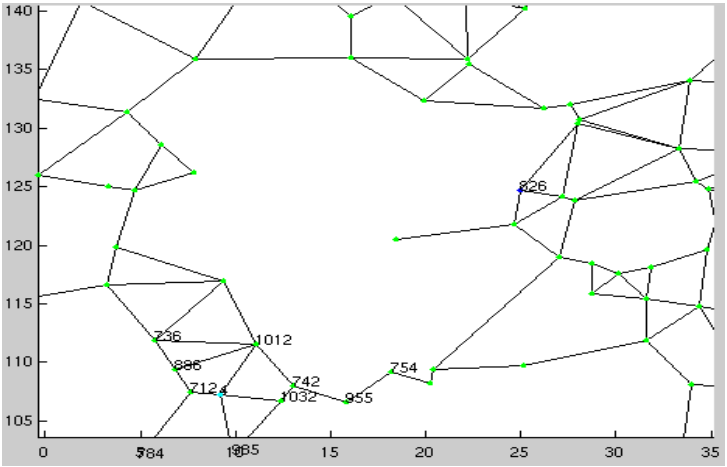

Figure 28: GPSR planarized neighbour table graph.

Transmissions and routing

In the GPSR algorithm there is always a single node holding the packet. The node which has the packet is coloured green. When a node does a transmission a line appears between the two nodes involved. Transmissions in greedy mode are green, transmissions in perimeter mode are blue. Figure 26 shows the first transmission in this experiment example.

Final analysis

Figure 29 shows an example of a packet traversing through the network. The packet starts in greedy mode, at node *754* the packet switches to perimeter mode since *754* does not have a neighbour closer to *826*. The packet then routes along the planarized neighbour graph until it reaches node *739*. This node is closer to the gateway *826* than *754* (where the packet went to perimeter mode) and therefore the packet switches back to greedy mode. It stays in greedy mode until it reaches the *gateway*.

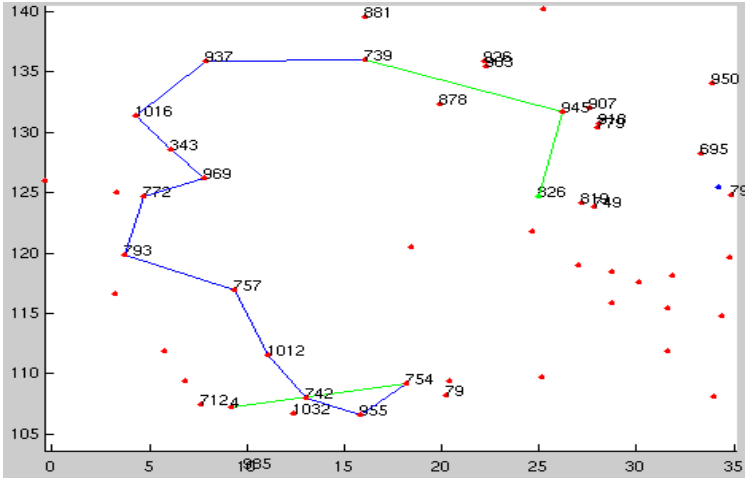

Figure 29: Example GPSR path. Node 4 is the source, node 826 is the gateway. Green lines are greedy routing decisions. Blue lines are face routing decisions.

8 Results and Analysis

8.1 Performance results

Quantification of the algorithm gives insight in the performance that can be expected in a real-life implementation.

8.1.1 Quantification of GZOR and GPSR

Figures 30 and 31 show the quantification results. The nodes in this simulation are deployed by four separate instances of the single air deployment model, 1384 nodes in total (Multiple Air Deployment Model). The centres of these single air deployment lie on the corners of a square with sides of 100m. Each data point in the figures is the mean across 25 randomly generated topologies. On each topology both routing algorithms let every connected node send 10 packets sequentially. This is done with 5 different path loss exponents, resulting in 5 data points. The middle data point (at density 8.2) corresponds with path loss parameter 4.7 which is the average value measured during an extensive empirical survey [Zun06].

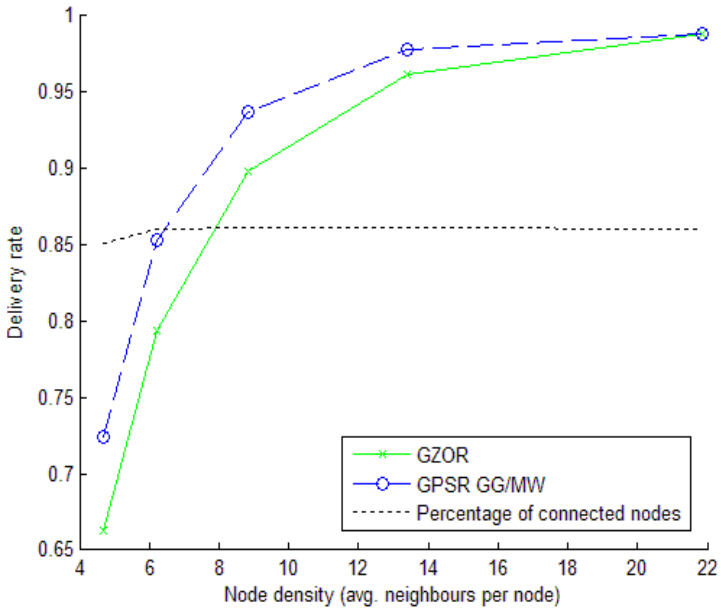

Figure 30: Delivery rate in the MADM

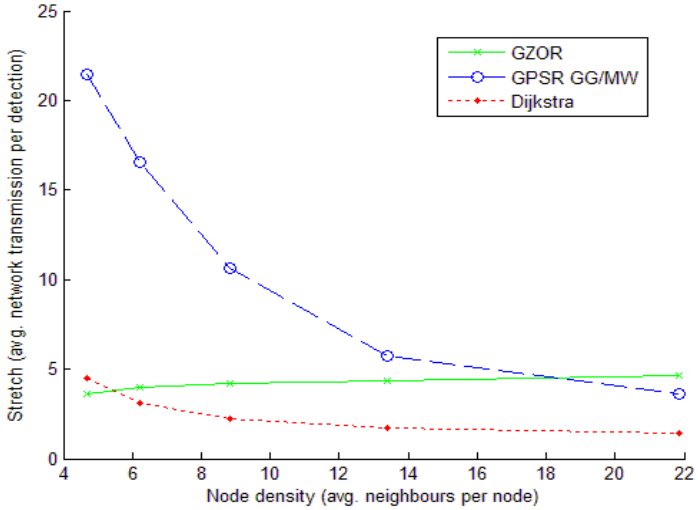

Figure 31: Stretch in the MADM

The figures show that GZOR consistently has a lower stretch, while GPSR has a higher delivery rate. Figures 32 and 33 show the distribution of the delivery rate and the induced stretch per node (sorted) of a single simulation. This simulation is performed with a path loss exponent of 4.7, which corresponds to a node density of 8.2 neighbours per node. These figures correspond with the data points of Figure 30 and 31 at node density 8.2.
The red line in Figure 32 indicates the number of connected nodes. Gateways and unconnected nodes have a stretch of zero in Figure 33 because they do not broadcast packets. Figure 32 shows that GZOR allows nodes to have a delivery rate lower than 80%. This is a result of the best-effort nature of the GZOR algorithm. With the GPSR algorithm, nodes either have a delivery rate of 80% and higher or no successful deliveries at all.

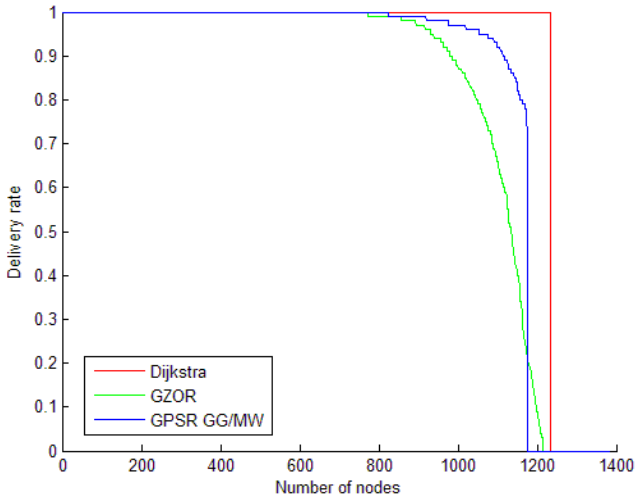

Figure 32: Delivery rate distribution of a MADM simulation

The peak of the GPSR algorithm in Figure 33 is caused by outer face traversals or loops in combination with retransmissions (the latter does not append to the TTL value). The small peak of Dijkstra's algorithm in the same figure is caused by nodes which have a very unreliable path to their gateway. These nodes are very likely to be disconnected in both simulated geographic routing algorithms.

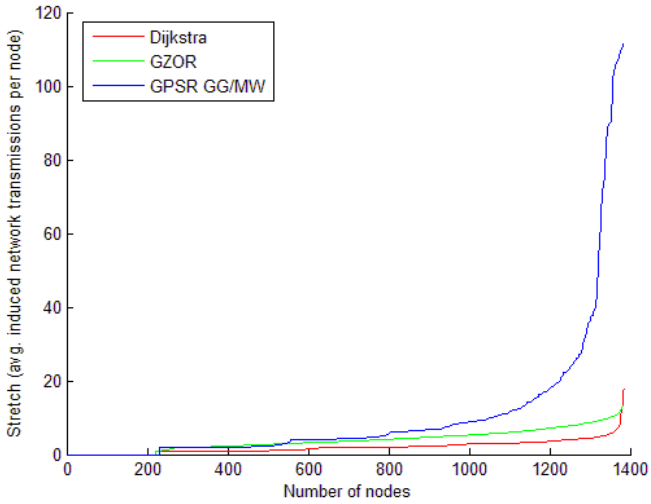

Figure 33: Stretch distribution of a MADM simulation

8.1.2 Asynchronous sleep cycles

The lifetime of the network can be extended by introducing asynchronous sleep cycles. To maintain the same amount of awake nodes available for routing, the number of deployed normal nodes is increased, correlated to the percentage of asynchronous sleep. Figure 34 shows the impact on delivery rate of increasing sleep cycles. The simulation is performed with a path loss exponent of 4.7.

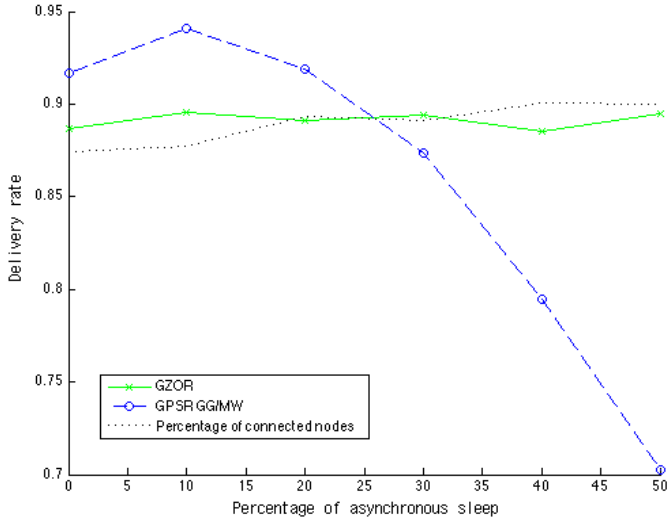

Figure 34: Delivery rate when increasing sleep

The figure clearly shows that GZOR is completely tolerable to these sleep cycles. As a result of the increase of network size, there will always be a path from sender to gateway, no matter which nodes are asleep. GPSR can tolerate about 10% of asynchronous sleep with use of retransmissions and actually benefits from the increased node amount.
Figure 35 shows the distribution of the delivery rate per node in a simulation with 50% of asynchronous sleep cycles. The figure indicates that the amount of connected nodes does not decrease with GPSR. Because of the temporarily unavailability of the nodes in the routing path, a larger percentage of transmissions fail along the way. This could be improved by increasing the amount of allowed retransmissions, but that would defeat the energy conserving purpose of the sleep cycles.

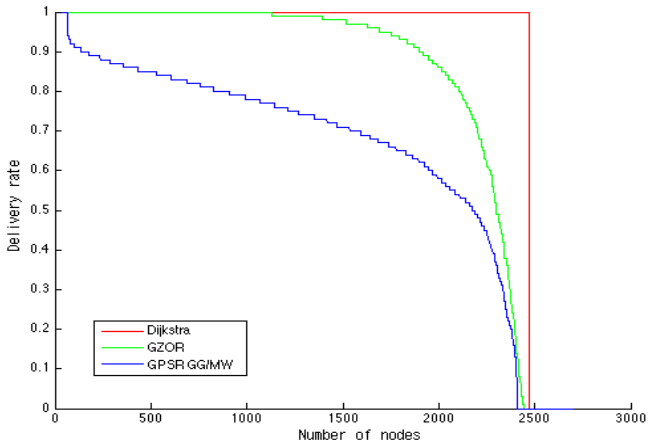

Figure 35: Distribution of delivery rate at 50% of asynchronous sleep

8.1.3 Proactive overhead GPSR

The GPSR algorithm completely depends on its routing tables. To cope with the dynamic behaviour of the network and the probability of node failure, these tables have to be updated proactively. The original specifications of GPSR state that each node sends a beacon message every 0.5 second [KaK00]. This interval is based on a network with a high rate of topology change caused by moving nodes. Since a security network does not contain moving nodes, its topology will be less dynamic. The optimal interval length of GPSR in the specific purpose of surveillance is unknown. It depends on the Mean Time To Failure of the nodes and the effect and frequency of changing weather conditions on the link quality between nodes. To gain a perspective of the proactive overhead costs, a simulation is performed on the MADM network. The amount of transmissions required per node in a refresh of the routing tables is presented in Figure 36. A refresh involves every node to broadcast 10 beacons to reassess the quality of links between itself and its neighbours. This process is simulated at different percentages of sleep cycles to evaluate the impact of the correlated network size increase on the proactive overhead.

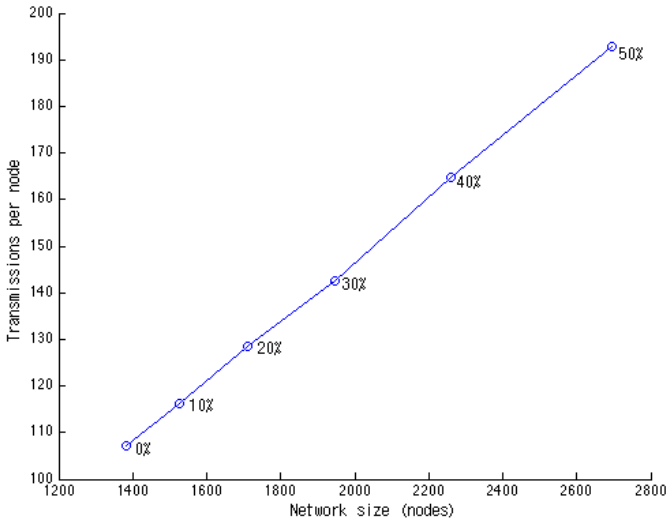

Figure 36: Required proactive overhead of GPSR. The circles represent the required network size corresponding with the percentage of sleep cycles.

8.2 Behavioural analysis

This section contains several phenomena that emerge from the GZOR algorithm's characteristics. During the design phase these phenomena where predicted as well as the effect on the algorithm's performance. Through several examples we demonstrate the occurrence of the predicted phenomena and how it differs from the behaviour of GPSR in the same situation.

8.2.1 Multipath emergence

During design of GZOR the decision was made to loosen the packet duplication constraint. This allows several copies of the same packet to travel different routes at the same time. It was argued that, as a result of the anisotropic radio ranges and the hardware variations among the nodes, this could lead to multi-path behaviour.

The behavioural analysis of the GZOR algorithm shows that multi-paths emerge frequently. Multi-path emergences are caused by only one single phenomenon. When a node receives a packet and sets a forwarding timer, it starts to listen to the channel. Only if it hears a node closer to the packet's destination broadcast the same packet, the node's timer will be cancelled. When this does not happen, the node will forward the packet itself. The only case a multi-path can emerge is when the transmission of a closer node fails to arrive at the more distant node with a live timer.

There are two kind of multi-paths: braided and disjoint. Braided multi-paths are very common in the GZOR algorithm. Braided multi-paths are often a result of the following situation. A sender transmits a packet and reaches a set of nodes. The most distant node (and closest to the gateway) forwards this packet. A node very close to the sender does not receive this packet and therefore it also forwards it. The nodes in between these two forwarders have already received this packet from the first forwarder and therefore drop it, so the newly formed path is rejoined. These transmissions are often redundant and do not add to the robustness of the algorithm. An example is given in Figure 37. In this example source node *183* reaches nodes *748*, *17*, *142*, *21* and *1352* in the first transmission. Since node *748* is closest to the gateway, it has the shortest timer value and therefore transmits first. In this transmission, besides its surrounding nodes and the gateway, only node *17* is reached. Therefore the other nodes still have live timers. They will forward the packet even though the only nodes they can reach already are aware of the fact that the packet has successful arrived at the gateway. We can see for instance that at node *748* the right multi-path rejoins.

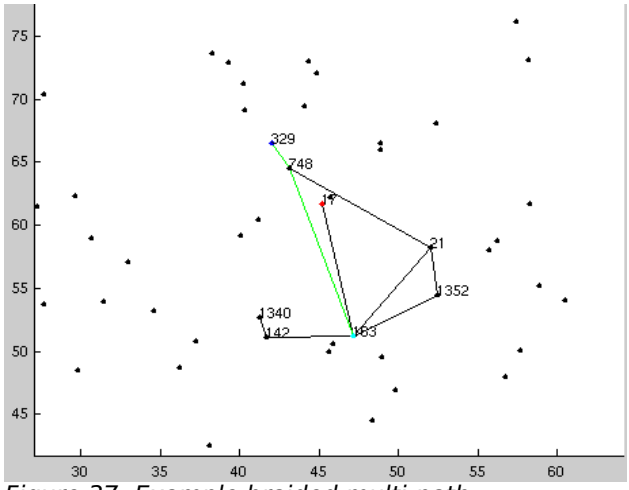

Figure 37: Example braided multi-path

Disjoint multi-paths are less common and happen when two nodes receive the packet from the sender, but although they are more or less evenly close to the gateway, the distance between each other is quite large. This not only causes failure of reception of each other's transmissions, they also reach a distinct set of new nodes along the path. This causes a new branch in which the packets are forwarded. On the outer sides of these branches both paths are completely disjoint. Figure 38 gives an example of such an event. Source node *54* reaches *926* and *721*, which are too far away from each other to achieve successful communication. When the packet on one of these branches becomes lost due to encounter of a network void, the other branch still has a chance of bringing the packet to the gateway. The branches meet at the intended gateway. Therefore disjoint multi-paths increase the robustness of the GZOR algorithm.

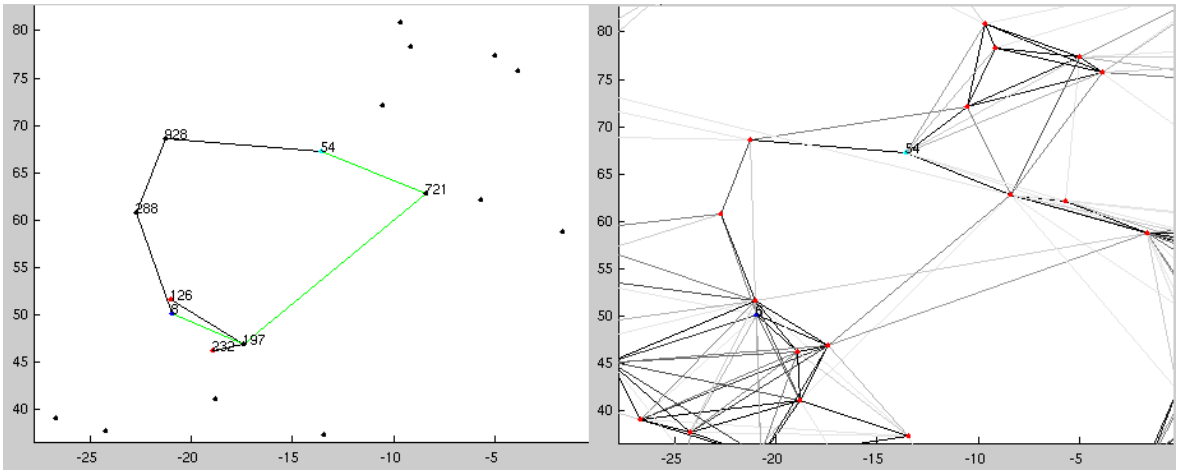

Figure 38: Example of a disjoint multi-path

To summarize: disjoint multi-paths often widen the path from source to gateway and increase robustness, these paths explore new ground. Disjoint paths are often caused by network voids in the centre of the path from source to gateway. Braided multi-paths do not widen the path from source to gateway and travel along ground that is already been covered by the packet.
Figure 39 illustrates these differences. All red nodes have received the original packet from the source node. Because of the circular nature of radio range, the node closest to the sender-destination path and in the centre of the receiving nodes is most likely to be the closest node to the gateway. Therefore it is allowed to forward first. In scenario 1 the transmission between node *1* and node *4* and *5* fails. Therefore these nodes still have live timers. However, when they finally transmit, they will only cover ground already reached by node *1* (nodes *2* and *3*). These nodes will drop the packets sent by node *4* and *5* because they have already received this packet from closer node *1*.
In scenario 2, a network void lies at the centre of the source's radio range. Therefore nodes *2* and *3* are now the closest nodes. Transmissions between them are likely to fail. As a result both nodes will forward the packet on either side of the void and thus create a disjoint multi-path. Combinations of both disjoint and braided are also very common.

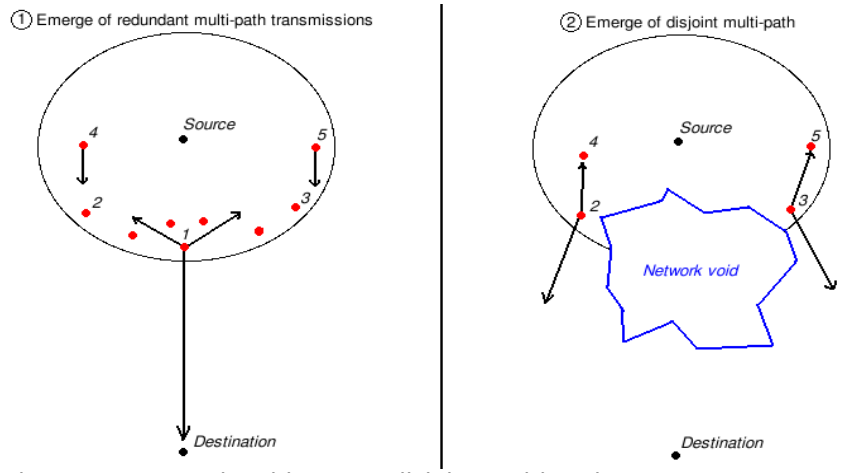

Figure 39: Network voids cause disjoint multi-paths

8.2.2 Link filtration

Chapter 5 outlines the fact that the GPSR algorithm is a state-based algorithm. Nodes build routing-tables filled with the locations of their neighbours. They do this by the method of broadcasting beacon messages. When they receive enough replies from a single node to their beacon message,

that node will be stored in the neighbour table. To prevent deadlocks in the planarization algorithm, a mutual witness protocol is added to ensure that all nodes agree on the fact whether they are neighbours or not. As a result of this process, weak links are filtered and will not be used in the routing process. Furthermore, links that are only strong in a single direction (asymmetric) are also filtered by the MW protocol. All the above is necessary since GPSR nodes specify the link which will be used for packet forwarding. If this link would be too weak, the packet would become lost (after a certain amount of retransmissions which are also likely to fail). Building and maintaining routing tables adds up to a significant amount of transmissions (see section 8.1.3).
The GZOR algorithm is stateless and as a result all links can potentially be used for routing. This section gives an analysis on the impact of link filtration.

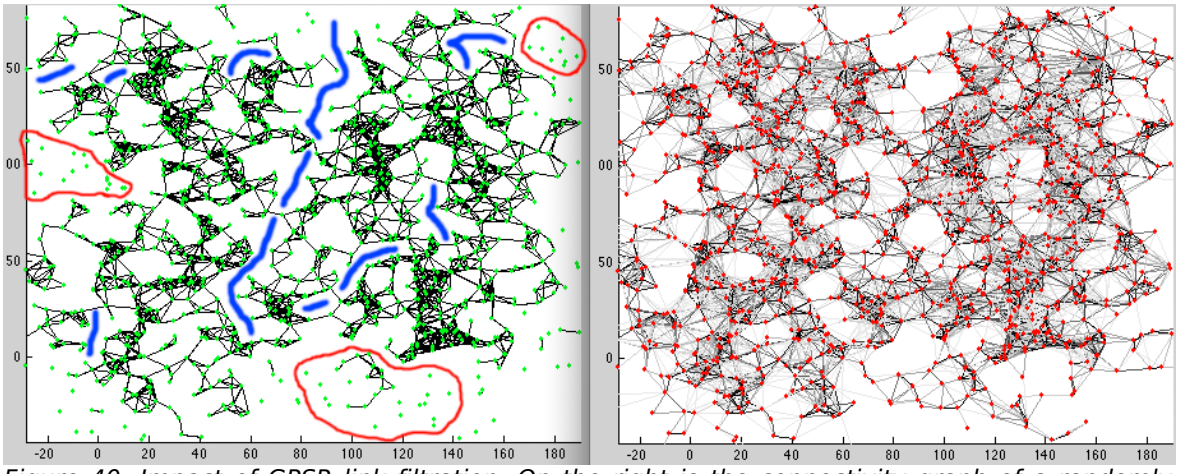

Figure 40: Impact of GPSR link filtration. On the right is the connectivity graph of a randomly generated network (darker links are stronger). On the left is the graph representation of the GPSR neighbour-tables. Emerged voids by link filtration are indicated with blue. Disconnected groups of nodes are circled red.

Figure 40 displays a randomly generated network. The left part of the figure represents the graph generated by the combination of neighbour-tables, the right part is the connectivity graph. When we analyse this figure it is clear that the filtration of low-quality links causes groups of nodes to become completely disconnected from the main network. These groups are circled red. Furthermore, it can be concluded that the filtration of links also increases the size of the connectivity voids in the network. Examples of this are marked blue.
Link filtration is necessary for GPSR, because it aims at reliable end-to-end connections between nodes [Woo03]. The consequence of this is that the filtration enlarges the present network voids. GPSR therefore has to take effort to route around the voids in order to achieve its high reliability. Unconnected nodes have no chance at all to successfully send packets to their gateway. GPSR only claims a high reliability on node-pairs which are connected by the neighbour table graph. [KaK00]

8.2.3 Network connectivity void handling

In the field of geographic routing, voids in the network topology always induce a threat to end-to-end reliability. The fundamental principle of geographic algorithms is to use geographic progress as an indicator for routing direction. On encounter of a network void, the nodes are required to route the packets away from the intended destination in order to find a path around the void. The network void handling of GPSR and GZOR is very different. This section demonstrates the difference in behaviour and also the consequences of the different strategies.

The previous section has demonstrated that the GPSR algorithm requires link filtration. As a result the connectivity voids can be enlarged. Low-quality links offering a chance to leap over a void are removed. GPSR has a powerful algorithm to route around these voids: the face routing algorithm, described in section 6.5. This algorithm can route packets away from the intended destination if necessary and can find a path around a void unconstrained by the size of it. Face routing offers a highly reliable connection between two nodes on either side of a void, if they are in fact connected by the neighbour table graph. When the latter is not the case, face routing offers no connectivity at all, and in the worst case consumes a high amount of transmissions before it finds out.

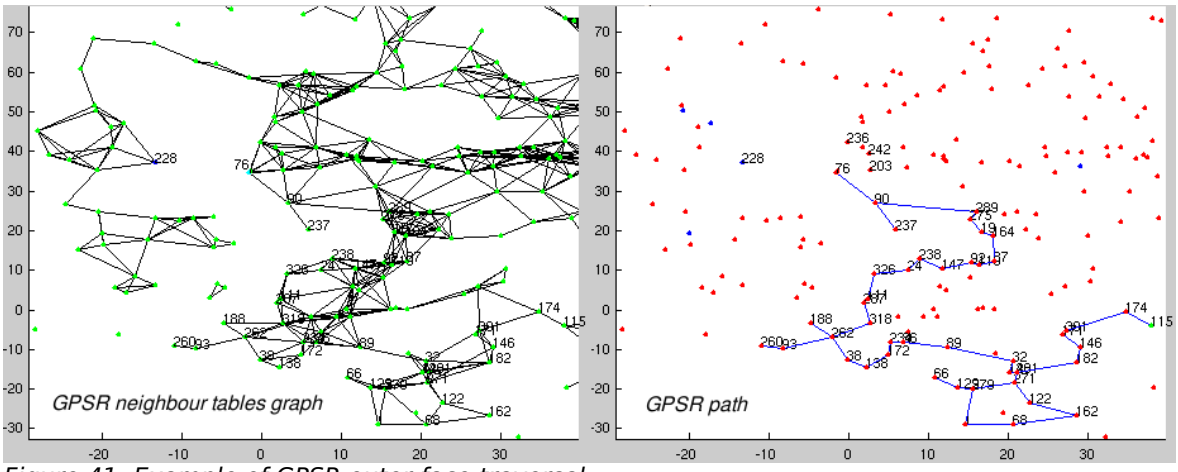

Figure 41: Example of GPSR outer face traversal

This worst case scenario can occur when the link filtration disconnects a gateway at the border of the network from the main network itself. This results in the traversal of the outer face (described in section 6.5), an example is given in Figure 41. This figure shows the bottom-left side of the network in Figure 40. Node *76* can only reach node *228* through the upper path. GPSR routes the packet downwards, where the packet will be routed around the entire network. The figure shows that outer face traversals can also happen when there is only one path around the void and GPSR chooses to face route according to the right-hand rule instead of the left-hand rule. It should be noted that there are newer face routing algorithms available which enable a packet to turn around and explore the opposite side of the face (e.g. GOAFR+ [Kuh03]).

Routing along the border of a face is costly in terms of transmission count. Because the Gabriel Graph planarazation algorithm (section 6.4) removes all crossing links, the links that offer the largest progress along the border of the graph are removed as well. Figure 42 illustrates this principle. As a result, when GPSR routes around a void it tends to visit all nodes that lie on the border of this void. For instance, the path from node *237* to *238* can be routed in 5 hops, but because all the nodes at the border of the face are visited, the result is a 10 hop path. The greedy nature of the GPSR algorithm is not reflected in the way it routes around voids.

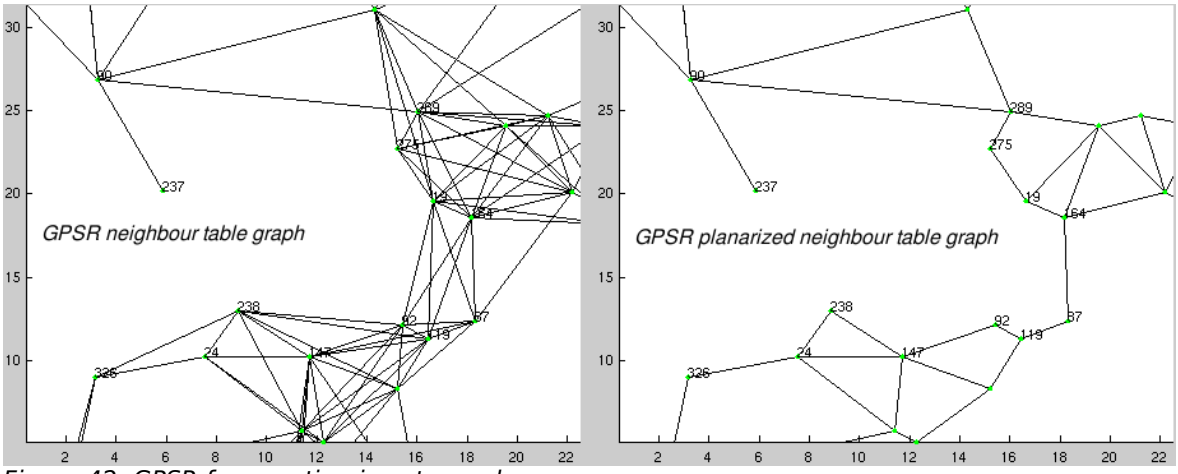

Figure 42: GPSR face routing is not greedy

The GZOR algorithm does not support different modes or connectivity void detection. It has a best-effort strategy to route along the existing links. GZOR does not filter links nor specifies the receiver of a packet. This provides the ability to jump over connectivity voids even when the available links are of low-quality. Since GZOR is not allowed to travel away from the destination, it does not have the ability to exploit strong links around a void. As a result, the delivery rate of GZOR tends to decrease on encounter of a void. This is illustrated in Figure 43. GPSR filters the grey links in this figure and routes around the void along nodes *1*, *2* and *3*. GZOR cannot travel backwards and will not exploit the reliable path around the void. However, when the source node broadcasts the packet, the chance that the *destination* or its surrounding nodes (*3* and *4*) will not receive the transmission is only 12%. As a result GZOR achieves a delivery rate of 88%. Note that if node *5* would be the source node, the path along nodes *1*, *2* and *3* is allowed to be used. Thus, resulting from the fact that packets are only allowed to travel forwards, nodes with a running start towards a void have a higher probability to route around it in GZOR.

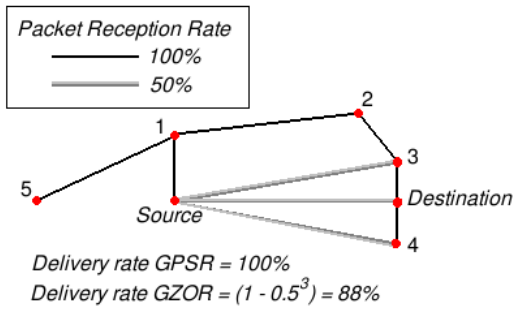

Figure 43: GPSR filters links which GZOR exploits

Section 8.2.1 demonstrates that small network can induce multi-paths, which widens the path of the packet. Section 5.2.4 explains that the maximum width of the path is constrained to prevent network floods. This leads to the following conclusion on GZOR's capability to handle network voids.

GZOR's definition of network voids is different from the GPSR definition. If there are low-quality links, it is not considered a void. GZOR can route around a network voids when the width of the void is smaller than the maximum range of the nodes. As a result, the position of the nodes relative to the void is very important. Figure 44 illustrates this conclusion. The connectivity graph shows that there exists a path from node *76* to gateway *228*. By means of left-hand face routing, GPSR is able to route around the void. GZOR can only route forward and since there are no links from *76* across the void, this node will never be able to reach node *228*. Nodes at the same horizontal height of *76* but further to the right, do have forward paths (running start). However, they might be constraint by the maximum width of the path. The nodes around coordinates *(10,55)* will be able to reach *228*. Their transmissions have a high probability for multi-path emerge, leading towards nodes *189* (upper path) and *76* (lower path). The upper path can deliver the packets but at a decreased delivery rate.

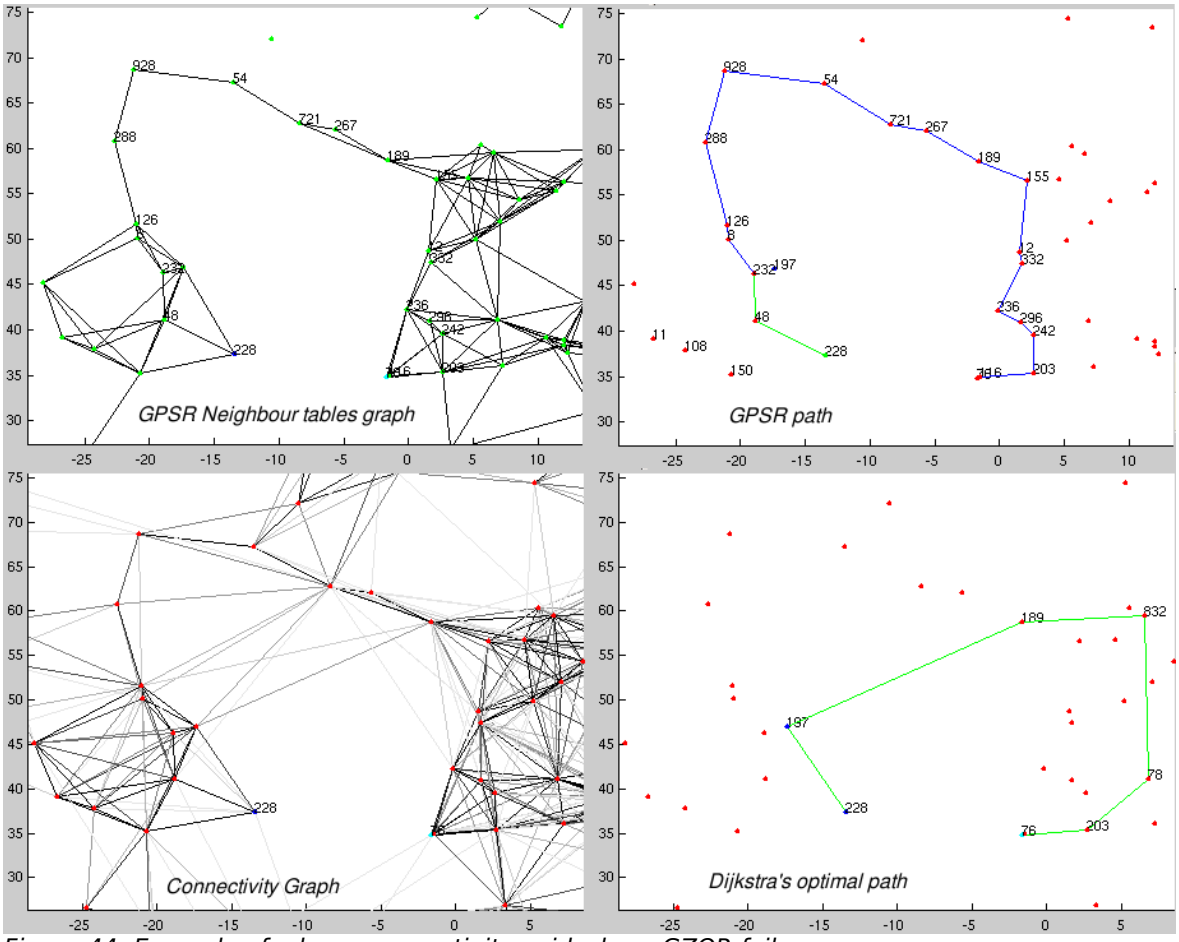

Figure 44: Example of a large connectivity void where GZOR fails

8.2.4 Dynamic routing paths

In the GPSR algorithm the path between sender and destination is predetermined by the routing tables. Therefore, as long as these tables are not updated, packets will always visit the same sequence of nodes. Packets routed by the GZOR algorithm do not have a predetermined path. The sequence of nodes visited by a packet depends on which nodes are reached with each transmission. This section demonstrates this behaviour with an example.

Figure 45 and 46 show a routing path of GZOR and GPSR respectively between two nodes in a randomly generated network. The node in the bottom left corner is the source. Characteristic behaviour emerges in both algorithms, multi-paths in Figure 45 and greedy/face routing switches in Figure 46. We performed a simulation with both algorithms where the source sequentially transmits 100 packets. The achieved delivery rate and stretch are the following: GZOR has an end-to-end delivery rate of 93% with a stretch of 40 transmissions per detection, GPSR has an end-to-end delivery rate of 95% with a stretch of 40 transmissions per detection.

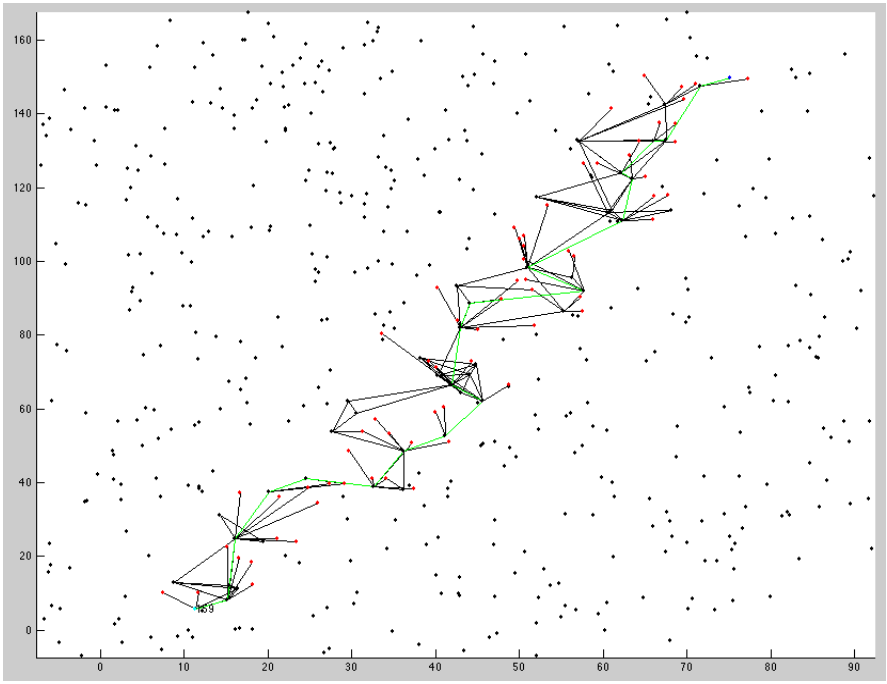

Figure 45: GZOR routing path

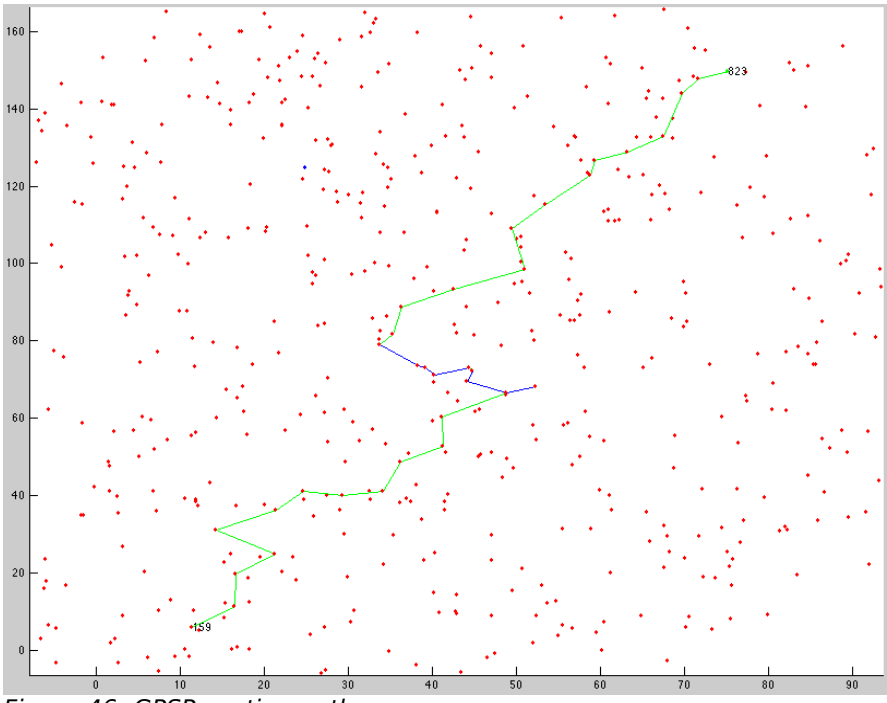

Figure 46: GPSR routing path

The amount of packets transmitted by each node is recorded. Figures 47 and 48 show the distribution of transmitted packets. Green nodes did not participate in forwarding the packet. The intensity of red indicates the amount of transmissions per node. These figures show that GZOR employs four times more individual nodes than GPSR. The amount of packets transmitted by each individual node is much higher with the GPSR algorithm. From these figures it can be concluded that it is likely that GPSR will drain individual nodes more quickly. As a result of the dynamic routing paths, the nodes' energy depletion will be more uniform with GZOR.

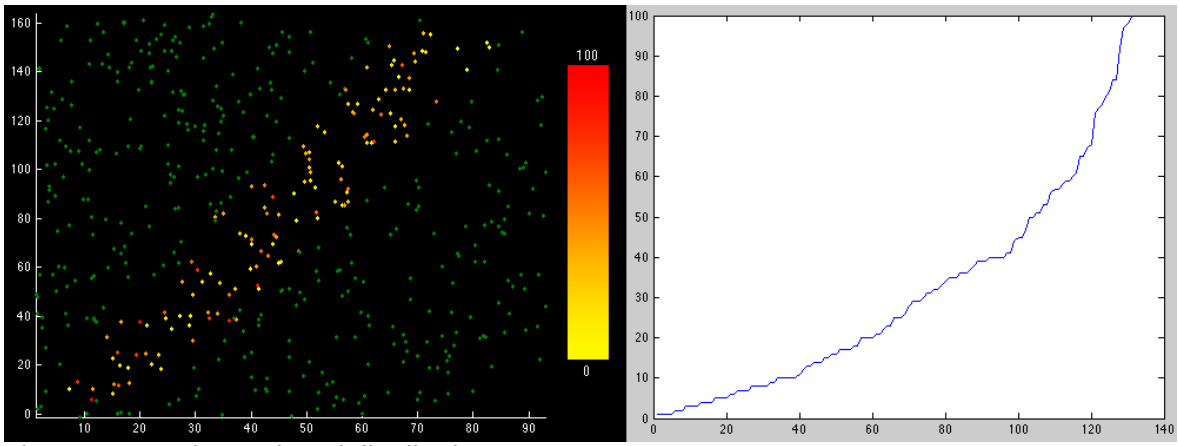

Figure 47: Routing path and distribution GZOR

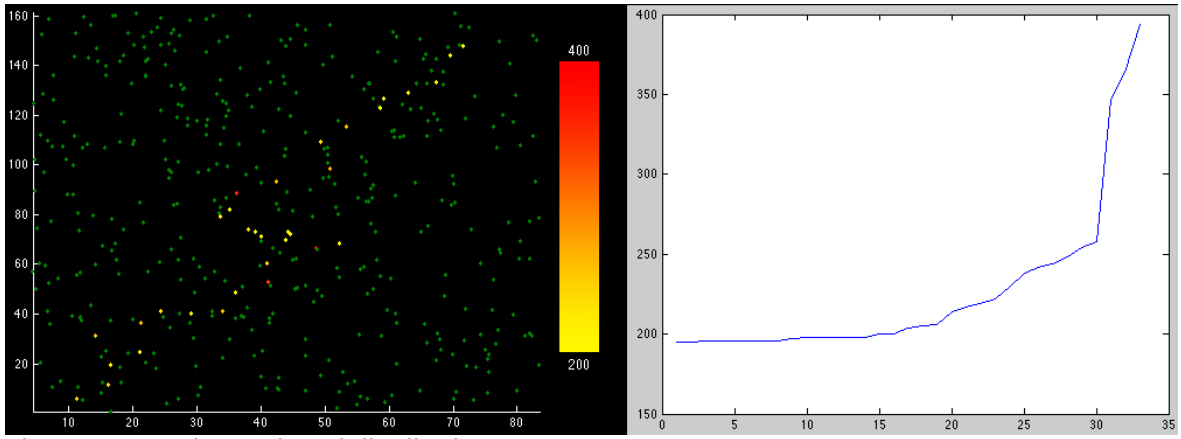

Figure 48: Routing path and distribution GPSR

It should be noted that GZOR's dynamic routing paths can create a significant amount of random jitter. For evaluation of the extent of jitter an implementation can be made on a real-time tinyOS simulator (e.g. AVRORA [Tit05]). This falls beyond the scope of this research.

8.2.5 Simultaneous crossing data streams

In a large network it is inevitable that streams of data collide during the lifetime of the network. We have run simulations to evaluate the effect of such an event on the delivery rate of colliding data streams. In these simulations the data stream from the previous section is intersected by another data stream that runs from the lower right corner to the upper left. We refer to both streams as data stream *1* and *2* respectively. Figure 49 illustrates the intersection of both streams and the involved transmission intensity per node. Data stream *1* nodes indicate increased intensity from yellow to red, data stream *2* nodes from blue to cyan. White nodes are used by both streams, the brightness of these nodes indicate the intensity of transmission.

During the simulation process the delivery rate of each data stream is monitored individually. The amount of packets per second from the crossing stream was increased to reveal the effect on the delivery rate of the intersecting data stream. This is done by calculation of the amount of transmissions and receptions per node of the crossing stream and the time nodes are occupied with it. The average time nodes are occupied by the crossing stream corresponds to a chance of transmission failure in the first stream. The length of a transmission was set to 10ms, thus 20 transmissions or receptions per second corresponds to 20% of transmission failure probability. The averages of both streams as well as the delivery rates were acquired by transmission of a 1000 packets per stream.

The results are presented in Figure 50. The results indicate that GZOR is slightly more tolerable to crossing traffic. Although this is only a single experiment, the small amount of white nodes in Figure 49 of the GPSR algorithm do indicate a possible higher probability of transmission failure due to concurrent transmissions of both streams.

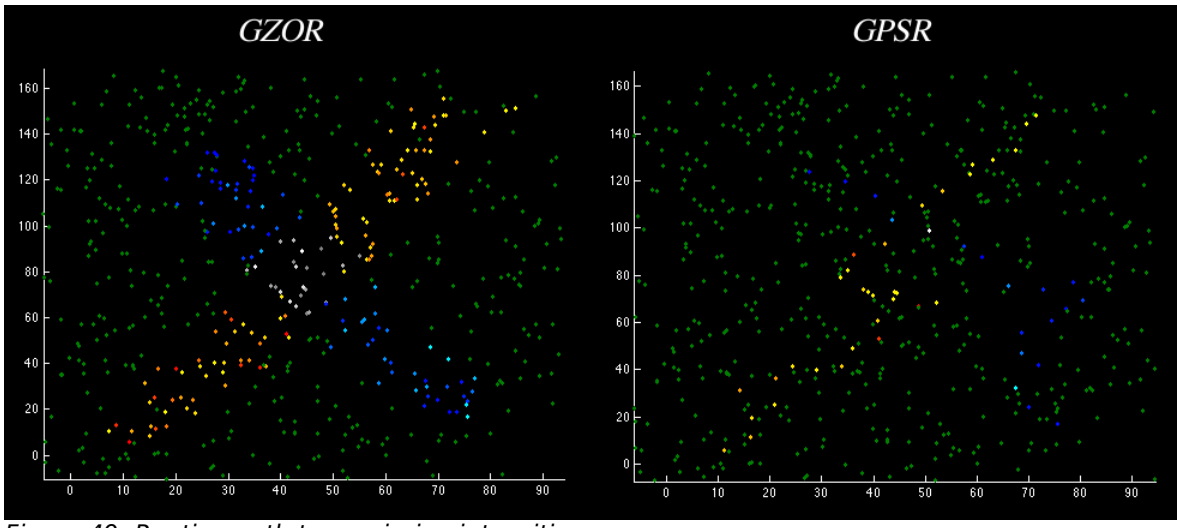

Figure 49: Routing path transmission intensities

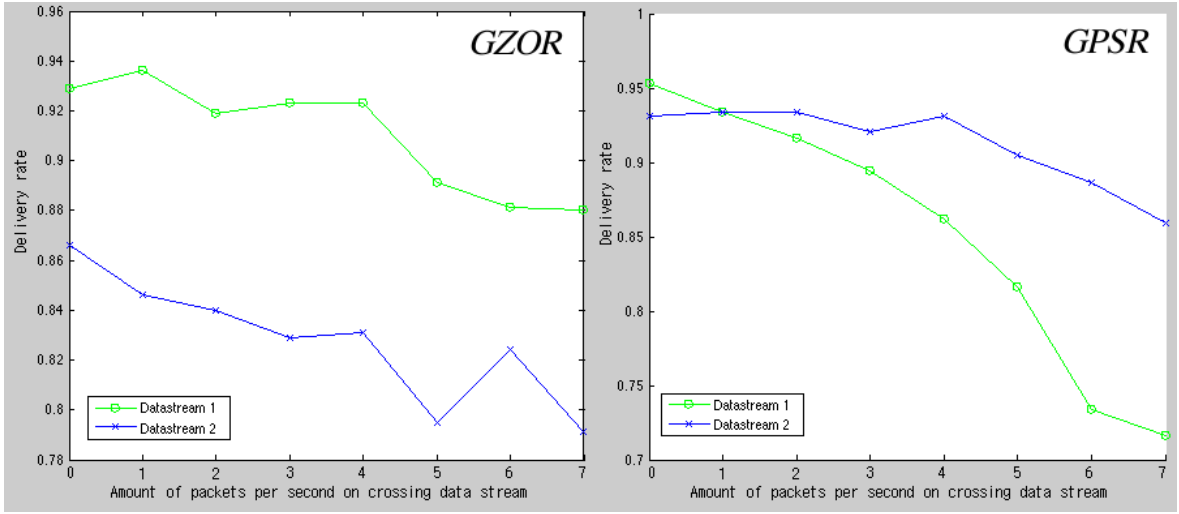

Figure 50: Decline of delivery rate on encounter of a crossing data stream

8.2.6 Lifetime estimation

We have made an estimation on the lifetime of a network simulated in this document. The estimation contains several simplified assumptions. Therefore the estimation is only meant to give insight in the factors that influence the lifetime of a network. It is by no means a claim on guaranteed lifetime. A more complete estimation of the lifetime of a network can be done with use of energy profiling tools (e.g. AEON [Lan05]). This requires a complete implementation in NesC.

Assumptions:

- The network size is 1384 nodes of which ~95% is functional ~=1300 nodes
- There are five intruders a day who invade the network for 15 minutes at a time
- An intruder is on average detected simultaneously by 5 sensor nodes
- packet rate per node = 1Hz → detections/day = 5 * 3600 * 5/4 = 22500
- Every node has one false positive per hour. → False positives/day: 1300 * 24 = 31200
- GPSR needs to update its neighbour tables four times a day to keep the network connected, sending 10 beacons per node (at once or during the day)
- Detection coverage is completely uniform so all nodes send an equal amounts of packets.
- The network functions the same throughout the lifetime of the network until all nodes are depleted at once. Therefore the stretch is always equal.

To estimate the total lifetime of a network we calculate two figures: the amount of transmissions a network has to do per day and the total network cost of a single transmission. The product of these figures indicates the amount of energy depletion per day. An estimate on the amount of days a network will last is acquired by dividing the capacity of a node's battery by the depletion per day.

Reactive communication:
At path loss 4.7, which corresponds to node density 8.2 (Figure 31):
- Stretch GZOR = 5 network transmissions/detection
- Stretch GPSR = 11 network transmissions/detection

Proactive communication:
At path loss 4.7, 0% asynchronous sleep cycles (Figure 36):
- 110 * 4= 440 transmissions/node/day

Daily traffic per node:
GZOR:
- False positives: 24 * 1300 * stretch GZOR= 156000
- Intruders: 22500 * stretch GZOR= 112500
- (156000 + 112500) / 1300 = 207 transmissions/day

GPSR:
- False positives: 24 * 1300 * stretch GPSR = 343200
- Intruders: 22500 * stretch GPSR = 247500
- (343200 + 247500) / 1300 + proactive GPSR = 894 transmissions/day

Summary power consumption [Chi06]:
- The capacity of the batteries are 2400 mAh → 1.5V * 2400mAh = 3600 mWh
- A node has two batteries in serial, 95% of the batteries energy can be used until the nodes stops functioning: 3600mWh * 2 * .95 = 6840 mWh [Suh04]
- Transmission cost: 25mA * 3V= 75 mW
- Listening (excl duty cycle): 10mA * 3V = 30 mW
- Receiving (excl duty cycle): 12.5mA * 3V = 37.5 mW
- Idle (incl 1% duty cycle): 10mA * 0.01 * 3V= 0.3mW
- Depletion of nodes/day when idle: 0.3mW * 24 = 7.2 mWh
- Lifetime of a node without communication: 6800mWh / 7.2mWh = 950 days

Data rates [Tin03]:

Mode	Duty cycle %	Effective data rate (kB/s)
0	100	12.364
1	35.5	5.671
2	11.5	2.488
3	7.53	1.737
4	5.61	1.336
5	2.22	0.559
6	1.00	0.258

We assume the packets have the same length as set in the simulator parameters [Dam08]:
- frame length = 55 bytes
- preamble = 23 bytes

Total network cost per transmission averaged per node:
Assumptions:
- A node has 8.2 neighbours, thus, when one node transmits, 8.2 neighbours are receiving

- Due to the duty cycle on average a receiving node is in reception mode during half the time of the preamble
- All neighbours receive the complete packet
- Routing decisions take 5 ms to calculate
- Idle nodes are in mode 6
- On detection of the preamble, nodes switch to mode 0
- When the transmission is completed, nodes switch back to mode 6

The mode switching can be optimized in many ways for both routing algorithms. We make the same mode switching assumptions for both algorithms to achieve a better comparison. For instance, on constant data streams switching to mode 0 and remaining in that mode will decrease the transmission time of a packet.

Summary:
- Sender sends preamble at mode 6 = 23 / (0.26 * 1024) = 87 ms
- Sender sends packet at mode 0 = 55 / (12.4 * 1024) = 4 ms
- Receiver wakes up on preamble and switches to mode 0. It waits in mode 0 until the packet will be sent (on average half of the preamble) = 44ms
- Receiver receives the packet in mode 0 = 4 ms
- Sender is 97 ms transmitting → 25mA * 3V * 91ms = 6.8 mWs
- Receiver is 44ms in receive mode 0 = 10mA * 3V * 44ms = 1.3 mWs
- Receiver is 10ms receiving mode 0 = 12.5mA * 3V * 4ms = 0.15 mWs
- CPU time per node required for routing decision: 8mA * 3V * 5ms = 0.12 mWs [Cro07]

We add the cost of reception of all surrounding nodes to the transmitting node's energy depletion. Since we assume a uniform detection coverage, on average this will have the same result.

If 1 node sends, 8.2 nodes receive. Average drain per transmission of the total network:
- 6.8 + 8.2 * (1.3 + 0.15 + 0.12) = 19.7 mWs

Average depletion per day per node:
GZOR:
- 207 transmissions/day *19.7mWs/detection + 7.2mWh (idle drain per day) = 8.3mWh

GPSR:
- 894 transmissions/day * 19.7mWs/detection + 7.2mWh (idle drain per day) = 12.1mWh

Expected lifetime:
- GZOR: 6840 / 8.3 = 824 days
- GPSR: 6840/ 12.1 = 565 days

9 Discussion

9.1 Delivery rate

The main assumption made in the GZOR development process was that a best effort delivery strategy offers a delivery rate sufficient for the intended purpose. The simulation results show that the achieved delivery rate of GZOR never falls below 10% difference with the delivery rate achieved with GPSR. However, both algorithms have troubles with keeping up their delivery rate when the network density decreases. The cause of delivery rate decrease is different in both algorithms. Previous section revealed insight on this matter and offers arguments of the applicability of both algorithms in different purposes.

The cause of delivery rate decline with GZOR lies in the size of connectivity voids when these lie in the direct path from sender to destination. As a consequence GZOR will not be able to successfully route in a network with large objects such as walls or buildings. This can be circumvented by assuring that the gateway of a node never lies behind the other side of such a wide void. For instance: gateway broadcast messages are also not allowed to travel around voids. However, when the intended purpose is a many-to-many network with large obstacles, the GZOR algorithm will probably not be able to deliver the delivery rate demonstrated in this document.

Decline of delivery rate with GPSR is caused by several factors. Link filtration can cause nodes to become disconnected. Failure of individual nodes or decline of external conditions can cause nodes to route their packets along non-existent links. Incorrect unit-disk model assumptions cause the planarization algorithm to fail on removing all crossing links from the planar graph. This causes packets to become dropped as a result of deadlocks. An added Time-To-Live field to detect deadlocks causes packets to become dropped during outer face traversals. Most of the causes of delivery rate decline can eased by increasing the frequency of beacon messages to keep the network connected. The planarization algorithm can also be improved by abandoning the local topology constraint. This decreases the scalability and lifetime of GPSR networks but improves the delivery rate. Summarising, GPSR is a better solution in small networks (with or without obstacles) where reliable many-to-many routing is a requirement. The 90% delivery rate stated in the requirements is achievable with both algorithms on average. When environmental conditions deteriorate, the delivery rate can fall below 90%.

9.2 Stretch

Figure 31 and 33 show that the stretch of GZOR seldom exceeds the optimal path by a factor 3. On average the stretch of GZOR is twice the stretch of the optimal path. Note that an algorithm travelling along the optimal path would always need transmission feedback (ACKS) to achieve a reliable connection. Although GZOR does not offer reliable connections, it almost routes with a stretch that would be considered near-optimal with reliable routing. The amount of duplicate packets or detours of GZOR seem to be more or less equal to the amount of required ACKs in an optimal routing algorithm. When network conditions deteriorate, lost packets causes the size of the stretch to decrease.

GPSR does have transmission feedback, therefore all routing paths have at least a stretch twice as large as the optimal path. Section 8.2.3 shows that face routing is expensive. This causes the stretch of GPSR to increase to four times the optimum on average when network conditions deteriorate. The worst-case stretch is much higher and is only bounded by the value of the TTL field. Since deteriorating conditions cause a decline in delivery rate, GPSR delivers less packets at a higher cost. This relation was also demonstrated by simulation in [Kim05].

9.3 Energy efficiency

From the properties of GZOR and the simulation results it can be concluded that GZOR is a very energy efficient algorithm. The stretch is near-optimal and the network only has to do transmissions as a result of detections. Furthermore, GZOR can also handle asynchronous sleep cycles and the nodes are drained in a uniform manner as a result of dynamic routing paths. Therefore the lifetime of the network depends mainly on the lifetime of the nodes and the amount of detections.
GPSR is less efficient with energy. Routing table maintenance and lack of asynchronous sleep support causes the lifetime of the network to decrease even in the absence of detections. The higher stretch of GPSR also indicates that GPSR is less energy efficient than GZOR.
The lifetime of the network can be increased by an optimal duty-cycle strategy. Longer duty-cycles causes nodes to last longer in absence of communication. Communication itself becomes more expensive because a longer preamble is required [Lan05]. The energy usages are not quantified in this research. A thorough energy estimation and algorithm optimizations are left for future work.

9.4 Overhead

GPSR needs proactive communication overhead to maintain routing tables in order to keep the network connected. The amount of communication required in such a maintenance cycle is measured and seems to increase linearly with the size of the network. This was to be expected since all topology sharing information is local (between neighbours only). The amount of maintenance cycles required to keep the network connected is unknown. Therefore we cannot state hard conclusions on the impact of this on the network's lifetime. It depends on the frequency of topology change.
In short, if the topology is very dynamic, the amount of transmissions required by GPSR to keep the network connected will severely shorten the lifetime of the network. If the topology is very static, proactive overhead is not a significant factor.
Nodes with the GZOR algorithm do to not have to communicate with each other to keep the network connected. This implies that GZOR would also be a good solution in a network where position aware nodes are moving constantly and only the position of the gateways are static.

9.5 Scalability

By design, both GZOR and GPSR are very scalable. The local topology overhead of GPSR ensures that routing table maintenance does not induce scalability issues. The outer face traversals do impose scalability issues, but this can easily be prevented. The GZOR solution has no theoretic scalability problems at all. The size of the network is irrelevant to the lifetime and communication extent. Besides, GZOR allows asynchronous sleep cycles together with an increase in network size. Implementation of this has no negative consequences to the stretch or delivery rate. Asynchronous sleep cycles increase the lifetime of a network and the increased amount of nodes decreases the average size of detection area voids.

9.6 Localization

Geographic routing algorithms strongly depend on their position information system. The quantification of this research was performed on nodes which acquired their position by simulation of a localization algorithm (details in appendix C). Localization errors cause GZOR nodes to make wrong decisions on whether to forward a packet. Nodes that incorrectly forward packets, increase the amount of redundant transmissions. Nodes that incorrectly drop packets decrease the delivery rate. Localization errors have a different impact on GPSR nodes. The errors cause more crossing links in the planar graph, resulting in more deadlocks [Sea04]. Both algorithms seem to tolerate localization errors when they do not change the relative topology of the nodes. Thus, if one node is closer to a destination, it does not matter whether this is 5 meters or 3 meters from a routing perspective. The decision on which node will forward the packet remains the same in both algorithms, although in the GZOR algorithm end-to-end delay and stretch can be influenced.

10 Conclusion and Recommendations

This document describes a research on wireless sensor network routing. The goal was development and evaluation of a multi-hop routing algorithm intended for a specific surveillance network solution. Current geographic algorithms focus on delivering reliable connections between all nodes in the network in an efficient manner. A surveillance solution only requires a high percentage of nodes to have a connection to gateway nodes with an acceptable delivery rate. Therefore the trade-off between reliability and energy efficiency can be shifted towards the latter.

The Geographic Zero Overhead Routing algorithm is specifically created to cope with network conditions that can be expected in the surveillance domain. Design fundamentals such as best-effort routing, volunteer forwarding and packet duplication are adopted to clear the network from communication overhead. This makes the algorithm more scalable, robust and energy efficient. The scalability gain is largely achieved by the absence of proactive communication overhead or path discovery strategies. The increased robustness is achieved by a volunteer forwarding mechanism, which causes routing paths not to be dependent on a single node. The improved energy efficiency is achieved by the fact that the algorithm exploits weak and unreliable node-to-node links, instead of ignoring them to increase reliability. Ignoring such links causes more connectivity holes in the network, which require an energy expensive backup algorithm to route around them. The absence of a packet duplication constraint together with volunteer forwarding enable this property to exploit weak links and jump across potential connectivity voids. It also enables the algorithm to function without transmission feedback. This comes at a price: a decline of the packet delivery rate is a consequence of the fundamental design decisions. GZOR also has trouble routing around large objects which cause large connectivity voids without weak links crossing them. Therefore, in a many-to-many network where reliability is a hard requirement, GZOR is not the best solution.

An implementation of Greedy Perimeter Stateless Routing was made to perform a comparison with GZOR. The GPSR algorithm was originally designed to function on a unit disk model. The applied WSN simulator (correctly) violates this model [Kot03] [Kot04], which results in failures of GPSR. Some fixes and modifications had to be applied in order to improve GPSR to prevent deadlocks stalling the simulator.

Both algorithms were simulated with various environmental parameters. The results show that GPSR consistently offers a higher delivery rate. GZOR is more energy efficient and is able to sustain asynchronous sleep cycles (not to be confused with a MAC-layer duty cycle mechanism), which contribute largely to energy conservation. The latter also indicates that GZOR is more robust against individual node failures.

The behavioural analysis gives more insight on how the algorithm functions. This can guide future developers in the optimization of the algorithm in an implementation stage. It also helps explaining the strengths and weaknesses of both GZOR and GPSR. Such knowledge offers arguments of which algorithm should be applied in specific situations.

This research shows that in the specific situation of a large volume of low-power radio nodes, the GZOR algorithm is a viable energy efficient solution. No effort has been taken to improve the GZOR parameters for better simulation performance. Such an approach would only result in curve-fitting the algorithm instead of providing fundamental knowledge.

The scope of this research has been the development of an algorithm with simulation as a tool. The next logical step would be to perform real-world experiments. This means that practical issues have to be solved, such as packet length, timer delays etc. This can be done using the same methodology as in [Kim05]. They make use of a process-level simulator TOSSIM, which is able to execute TinyOS code. The same code required to run this simulator can later be used for a real-world implementation. Berkeley (developer of the Mica2 platform) offers a shared testbed infrastructure where simulation time can be rented. The number of nodes in this testbed is limited (<100 nodes) and the environment (indoor) does not match the intended deployment conditions, but despite of these shortcomings a real-world experiment could deliver valuable data. An outdoor experiment with 300 nodes or more is of course preferred, but also more costly.

An estimation of the lifetime of a network was made to gain some perspective how the different aspects of a routing algorithm relate to an increase of network lifetime. This estimation is not a claim to achievable performance. Accurate energy profiling can be performed on simulators dedicated to this purpose (e.g. AEON [Lan05]).

Furthermore, assumptions on the deployment methods were made. Not much research has yet been done on this topic. Especially an air deployment comes with a lot of practical problems. Scattering of the nodes, survival after impact, resilience against weather conditions and security are topics that have to be solved to come closer to a practical implementation.

11 Acknowledgements

I would like to thank Peter Olde Damink from Thales Nederland BV and Hans Scholten from Twente University for the insightful guidance that has benefited this work. In particular I would like to thank Johan Slagman from Thales Nederland BV for his helpful support and countless reviews, this has significantly contributed to the quality of this document.

12 REFERENCES

[Bar01] L Barriére, L Narayanan - *Robust position-based routing in wireless Ad Hoc networks with unstable transmission ranges* - Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications, 2001

[Ber93] Berkeley Design Technology, Inc. - *DSP Algorithm Development Tools* - DSP & Multimedia Technology, November 1993. - http://www.bdti.com/articles/info_dspmt93algorithm.htm

[Blu03] B Blum, T He, S Son, J Stankovic - *IGF: A state-free robust communication protocol for wireless sensor networks* - Department of Computer Science, University of Virginia, USA, Tech. Rep. CS-2003-11, 2003

[Bos06] S Bosch - *Wireless sensor networks for surveillance* - Thales, 2006

[Bos01] P Bose, P Morin, I Stojmenović, J Urrutia - *Routing with Guaranteed Delivery in Ad Hoc Wireless Networks* - Wireless Networks, 2001

[Cam06] T Camilo, A Rodrigues, JS Silva, F Boavida - *Lessons Learned from a Real Wireless Sensor Network Deployment* - IFIP Networking Conference, Performance Control in Wireless SensorNetworks, pages 71-78, 2006

[Chi06] Chipcon Products from Texas Instruments - *CC1000 Single Chip Very Low Power RF Transceiver* - http://www.ti.com/lit/gpn/cc1000, 2006

[Cou05] DSJD Couto, D Aguayo, J Bicket, R Morris - *A high-throughput path metric for multi-hop wireless routing* - Wireless Networks, 2005

[Cro07] Crossbow Wireless Sensor Networks - *Mica2 Datasheet* - http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf, 2007

[DuD04] W Du, J Deng, YS Han, S Chen, PK Varshney - *A key management scheme for wireless sensor networks using deployment knowledge* - IEEE INFOCOM, VOL 1, pages 586-597, 2004

[Dul03] S Dulman, T Nieberg, J Wu, P Havinga - *Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks* - Wireless Communications and Networking, 2003

[Fee01] LM Feeney, M Nilsson - *Investigating the energy consumption of a wireless networkinterface in an ad hoc networking environment* - INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2001

[Fre06] H Frey, I Stojmenovic - *On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks* - Proceedings of the 12th annual international conference on Mobile computing and networking, 2006

[Gan01] D Ganesan, R Govindan, S Shenker, D Estrin - *Highly-resilient, energy-efficient multipath routing in wireless sensor networks* - ACM SIGMOBILE Mobile Computing and Communications Review, 2001

[Gio04] S Giordano, I Stojmenovic, L Blazevic - *Position based routing algorithms for ad hoc networks: a taxonomy* - Ad Hoc Wireless Networking, 2004

[Hae04] M Haenggi - *Twelve Reasons not to Route over Many Short Hops* - IEEE Vehicular Technology Conference, 2004

[Jon96] DB Johnson, DA Maltz - *Dynamic source routing in ad hoc wireless networks* - Mobile Computing, 1996

[Kar00] BN Karp - *Geographic Routing for Wireless Networks* - PhD. Dissertation, Harvard University, Cambridge, 2000

[KaK00] BN Karp, HT Kung - *GPSR: greedy perimeter stateless routing for wireless networks* - Proceedings of the 6th annual international conference on Mobile computing and networking, 2000

[Kim05] YJ Kim, R Govindan, B Karp, S Shenker - *Geographic routing made practical* - Proceedings of the 2nd Symposium on Networked Systems Design and Implementation, Boston, Massachusetts, 2005

[Kim06] J Kim, R Govindan, B Karp, S Shenker - *Lazy cross-link removal for geographic routing* - ACM SenSys, 2006

[KiG05] YJ Kim, R Govindan, B Karp, S Shenker - *On the pitfalls of geographic face routing* - Proceedings of the 2005 joint workshop on Foundations of mobile computing, 2005

[KoV00] YB Ko, NH Vaidya - *Location-Aided Routing (LAR) in mobile ad hoc networks* - Wireless Networks Volume 6, 2000

[Kot03] D Kotz, C Newport, C Elliott - *The mistaken axioms of wireless-network research* - Dartmouth TR2003-467, 2003 - cs.dartmouth.edu

[Kot04] D Kotz, C Newport, RS Gray, J Liu, Y Yuan, C Elliott - *Experimental evaluation of wireless simulation assumptions* - Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, 2004

[Kuh03] F Kuhn, R Wattenhofer, Y Zhang, A Zollinger - *Geometric ad-hoc routing: of theory and practice* - Proceedings of the twenty-second annual symposium on Principles of distributed computing, 2003

[Lan05] O Landsiedel, K Wehrle, S Gotz - *Accurate Prediction of Power Consumption in Sensor Networks* - Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on Volume, Pages 37-44, 2005

[Lev03] P Levis, N Lee, M Welsh, D Culler - *TOSSIM: accurate and scalable simulation of entire tinyOS applications* - Proceedings of the 1st international conference on Embedded networked sensor systems, 2003

[Lev05] P Levis, S Madden, J Polastre, R Szewczyk, K Whitehouse, A Woo, D Gay, J Hill, M Welsh, E Brewer, D Culler - *TinyOS: An Operating System for Sensor Networks* - Ambient Intelligence, Springer Berlin Heidelberg, 2005

[Mar04] M Maróti - *Directed flood-routing framework for wireless sensor networks* - Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware, 2004

[NiN03] D Niculescu, B Nath - *Ad hoc positioning system (APS) using AOA* - INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, 2003

[Per99] CE Perkins, EM Royer - *Ad-hoc on-demand distance vector routing* - Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999

[Sea04] K Seada, A Helmy, R Govindan - *On the effect of localization errors on geographic face routing in sensor networks* - Proceedings of the third international symposium on Information processing in sensor networks, 2004

[Sea07] K Seada, A Helmy, R Govindan - *Modeling and analyzing the correctness of geographic face routing under realistic conditions* - Ad Hoc Networks, Volume 5, Issue 6, Pages 855-871, 2007

[Shn04] V Shnayder, M Hempstead, B Chen, GW Allen, M Welsh - *Simulating the power consumption of large-scale sensor network applications* - Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004

[Sla07] J Slagman - *Localization for Wireless Ad Hoc Sensor Networks for Surveillance* - Thales, 2007

[Son06] D Son, B Krishnamachari, J Heidemann - *Experimental Analysis of Concurrent Packet Transmissions in Low-Power Wireless Networks* - Proceedings of the 4th International Conference on Embedded networked sensor systems, 2006

[Suh04] J Suh – *MICA2 AA Battery Pack Service Life Test* – Crossbow Technology inc. 2004 - www.xbow.com/Support/Support_pdf_files/MICA2_BatteryLifeTest.pdf

[Tin03] Tiny OS Tutorial - *MICA2 Radio Stack for TinyOS* - http://www.tinyos.net/tinyos-1.x/doc/mica2radio/CC1000.html, 2003

[Tit05] BL Titzer, DK Lee, J Palsberg - *Avrora: scalable sensor network simulation with precise timing* - Proceedings of the 4th international symposium on Information processing in sensor networks, 2005

[Whi07] K Whitehouse, C Karlof, D Culler - *A practical evaluation of radio signal strength for ranging-based localization* - ACM Sigmobile Mobile Computing and Communications Review, 2007 - portal.acm.org

[Wil05] A Willig, H Karl - *Data Transport Reliability in Wireless Sensor Networks: A Survey of Issues and Solutions* - Praxis der Informationsverarbeitung und Kommunikation, 2005 Data Transport Reliability in Wireless Sensor Networks—A Survey of Issues and Solutions

[Wit05] M Witt, V Turau - *BGR: blind geographic routing for sensor networks* - Intelligent Solutions in Embedded Systems, 2005

[Wit06] M Witt, V Turau - *The Impact of Location Errors on Geographic Routing in Sensor Networks* - Proceedings of the International Multi-Conference on Computing in the Global Information Technology, 2006

[Woo03] A Woo, T Tong, D Culler - *Taming the underlying challenges of reliable multihop routing in sensor networks* - Proceedings of the 1st international conference on Embedded networked sensor systems, 2003

[XuL05] Y Xu, WC Lee, J Xu, G Mitchell - *PSGR: priority-based stateless geo-routing in wireless sensor networks* - Proceedings of IEEE International Conference on Mobile Ad-hoc Sensor Systems, 2005

[YeS06] W Ye, F Silva, J Heidemann - *Ultra-low duty cycle MAC with scheduled channel polling* - Proceedings of the 4th international conference on Embedded networked sensor systems, 2006

[Zha03] J Zhao, R Govindan - *Understanding packet delivery performance in dense wireless sensor networks* - Proceedings of the 1st international conference on Embedded networked sensor systems, 2003

[Zho06] G Zhou, T He, S Krishnamurthy, JA Stankovic - *Models and solutions for radio irregularity in wireless sensor networks* - ACM Transactions on Sensor Networks (TOSN), 2006 - portal.acm.org

[Zor03] M Zorzi, RR Rao - *Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance* - Mobile Computing, IEEE Transactions on, 2003

[ZoR03] M Zorzi, RR Rao - *Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance* - Mobile Computing, IEEE Transactions on, 2003

[Zun06] M Zuniga, B Krishnamachari - *An analysis of unreliability and asymmetry in low-power wireless links* - under submission TOSN, 2006 - http://www-scf.usc.edu/~marcozun/

[ZSK06] M Zuniga, K Seada, B Krishnamachari, A Helmy - *Efficient Geographic Routing over Lossy Links in Wireless Sensor Networks* - under submission TOSN, 2006 - http://www-scf.usc.edu/~marcozun/

Appendix A: Simulation details

This chapter contains the details of the performed simulations. The results can be found in chapter 8.

GG RNG CLDP Comparison Simulation	
number of nodes	1384 (=4xSADM)
simulated path loss exponents	[5.5,5.1,4.7,4.3]
gateway percentage	5%
anchor percentage	15%
number of random topologies	10
path loss component localization	4.7
deployment centres	4
locations centres	[(30;30),(130;30),(30;130),(130;130)]

Table 1: GG RNG CLDP comparison simulation details

Multiple Air Deployment Model Simulation	
number of nodes	1384 (=4xSADM)
simulated path loss exponents	[5.9,5.5,5.1,4.7,4.3,3.9]
gateway percentage	5%
anchor percentage	15%
sleep time percentages	0%
number of random topologies	25
path loss component localization	4.7
deployment centres	4
locations centres	[(30;30),(130;30),(30;130),(130;130)]
deployment scatter deviation	50m
transmission power Mica2 node	5dB
noise floor	-105dB

Table 2: MADM simulation details

Asynchronous sleep cycles simulation	
number of nodes	[1384 1530 1713 1948 2260 2699]
simulated path loss exponents	4.7
sleep time percentages	[0% 10% 20% 30% 40% 50%]
number of random topologies	10
path loss component localization	4.7
deployment centres	4
locations centres	[(30;30),(130;30),(30;130),(130;130)]

Table 3: Asynchronous sleep cycles simulation details

GPSR proactive overhead simulation	
number of nodes	1384 (=4xSADM)
simulated path loss exponents	4.7
sleep time percentages	[0% 10% 20% 30% 40% 50%]
number of random topologies	10
path loss component localization	4.7
deployment centres	4
locations centres	[(30;30),(130;30),(30;130),(130;130)]

Table 4: GPSR proactive overhead simulation details

Single Air Deployment Model Simulation	
number of nodes	346
simulated path loss exponents	[5.9,5.5,5.1,4.7,4.3,3.9]
gateway percentage	5%
anchor percentage	10%
sleep time percentages	0%
number of random topologies	10
path loss component localization	4.7
deployment centres	1
locations centres	[(150;150)]
deployment scatter deviation	50m
transmission power Mica2 node	5dB
noise floor	-105dB

Table 5: SADM simulation details

Group-Based Air Deployment Model Simulation	
number of nodes	1216 (64x19)
simulated path loss exponents	[5.9,5.5,5.1,4.7,4.3,3.9]
gateway percentage	1 gateway per group: 5.3%
anchor percentage	2 anchors per group: 10.5%
sleep time percentages	0%
number of random topologies	10
path loss component localization	4.7
deployment centres	64
locations centres	8x8 square grid, centres lie 25m apart in x and y
deployment scatter deviation	50m
transmission power Mica2 node	5dB
noise floor	-105dB

Table 6: GBADM simulation details

Hand Deployment Model Simulation	
number of nodes	1024
simulated path loss exponents	[5.9,5.5,5.1,4.7,4.3,3.9]
gateway percentage	5%
anchor percentage	10%
sleep time percentages	0%
number of random topologies	10
path loss component localization	4.7
Grid size	183mx183m
transmission power Mica2 node	5dB
noise floor	-105dB

Table 7: MDM simulation details

Geographic Zero Overhead Routing Simulation	
unique packets transmitted per node	10
Number of measurements per *prrTable* entry	15

Table 8: GZOR simulation details

Greedy Perimeter Stateless Routing	
unique packets transmitted per node	10
Number of measurements per *prrTable* entry	15
maximum retransmissions	3
Time To Live value	50
Planarization algorithm	Gabriel Graph with Mutual Witness fix (GG/MW)

Table 9: GPSR simulation details

Appendix B: Node deployment

In chapter 8 the performance results of algorithm simulation on the Multiple Air Deployment Model were presented. Several others deployment models were also investigated, this chapter explains the models and presents simulation results. Simulation details are outlined in appendix A.
The simplest way to deploy a wireless network is doing this manually. This means that someone must walk around the area and deploy every single node. With this method a semi-optimal distribution of nodes can be assured. This comes down to a network without large node connectivity holes, a minimum distance between individual nodes and a semi-uniform spread of anchors and gateways. We assume that the person who deploys the network walks over the network area in some structured way without knowledge of his exact location. This results in a deployment where the nodes are not in an exactly structured grid position, but all nodes have some deviation from their intended position.
We model this by starting with a general plan of where the nodes are supposed to be deployed and calculate for each node an error in their deployment position according to a Gaussian function with a deviation of 2m in x and y direction. Although this deployment method could offer possibilities to help the nodes localize themselves, we do not assume that this is the case in this model. Nodes still have to localize themselves based on the anchor nodes information. An example of such a manual deployment is illustrated in Figure 51. The simulation results of both routing algorithms are in Figure 52.

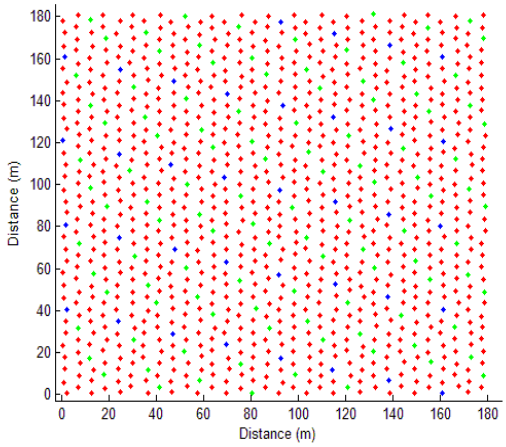

Figure 51: Example of a manual deployment. Red, green and blue dots represent regular, anchor and gateway nodes respectively

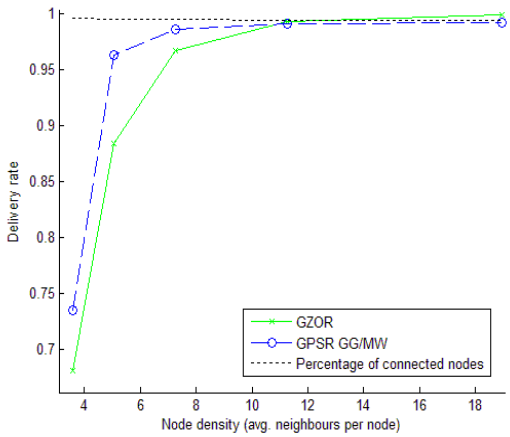

Figure 52: Manual Deployment Model simulation results

Since the area in which the network has to be deployed could be hostile or impassable, it is unfortunately not always possible to deploy the network by hand. Therefore we also consider a deployment by air; this however comes with some inconvenient side-effects.
Firstly, all nodes are dropped from a single point, which is the position of the aerial vehicle. Scattering of the nodes has to be accomplished by gravity and wind influences. Not only does this severely limit the size of network area, nodes are also more likely to land in the centre than at the border of such a network. This results in a high network density in the centre and a decrease of density proportional to the increase of distance from the centre [Cam06]. This leads to the situation where both detection probability and network performance will decrease the further one moves away from the centre.
Secondly, there is no control over were the nodes actuality land, so the network may contain large connectivity (or detection) holes and the distribution of anchor and gateway nodes could be far from optimal.
Unfortunately, the research on the distribution of an air deployed wireless sensor network is very limited. We assume that a two-dimensional Gaussian distribution of the nodes is a realistic way to model this. We do not know what the deviation of this distribution is and how it correlates with the

speed and height of the aerial vehicle. We assume that an airplane travelling at a height of about 500 meters corresponds with a node deviation of 50 meters. This means that 95% of the nodes land within 100 meters of the deployment centre, limiting the grid size to a circle with a radius of 100 meters. This is actually a lot smaller, since it would require a relatively high amount of nodes to reach a critical network density to ensure some required percentage of detection coverage. We also assume that the probability of a node landing anywhere on the deployment area is equal for all types of nodes (anchors, gateways etc.). Figure 53 shows an example of a generated Single Air Deployment according to this model. This model can easily be extended to incorporate multiple air deployments, each with a different centre location. A variation of the latter is utilized in this research and referred to as the Multiple Air Deployment Model.

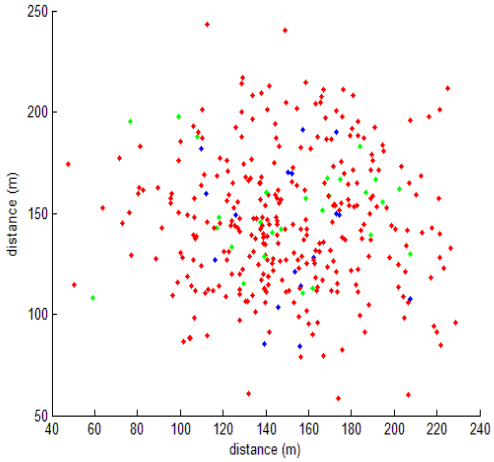

Figure 53: Example of the Single Air Deployment Model. Red dots are normal nodes, blue en green nodes represent gateways and anchors respectively

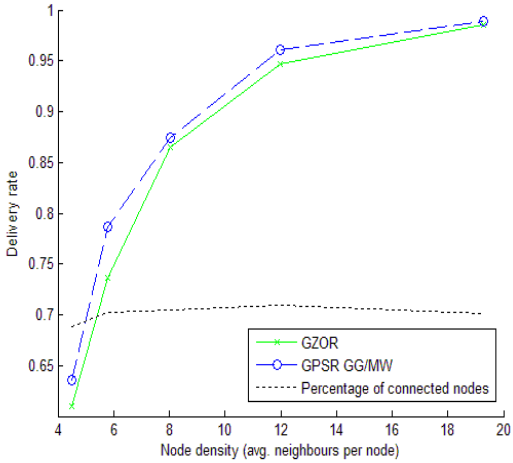

Figure 54: SADM simulation results

As shown above, a single air deployment does not provide a grid with an equal node density on every point. Besides, since the scattering of nodes is limited by physical boundaries, the size of the grid inherits that limitation. A solution to this problem is to construct the network by using multiple smaller, overlapping deployments (groups all consisting of an equal amount of nodes, anchors and gateways). This would provide a more uniformly distributed network and would also solve the size boundaries. Besides that, nodes that belong to the same sub-deployment could be programmed with some topology information, for instance which gateways are more likely to be close or perhaps on a higher level, security key distribution could be optimized [Dud04].
We assume that for this group-based deployment an aerial vehicle would have to fly over the network area several times, and distribute the groups according to some predefined grid. The grid in this case consists of some uniform distribution of deployment points. In order to reach a certain amount of accuracy, we assume the aerial vehicle has to fly slower and/or lower than in the single air deployment strategy. This leads to the assumption that the nodes deviations from the centre of such a deployment point is smaller, we assume 15 meters (so 95% of the nodes lie within 30 meters of the deployment point). When we place the deployment points 25 meters apart, we should get a quite evenly distributed network. Figure 55 shows an example of a generated group-based air deployment according to this model.

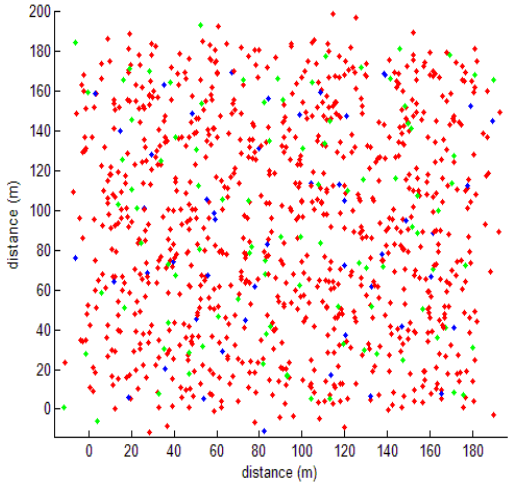

Figure 55: Example of a group-based air deployment. The network consists of 49 deployments, each group contains one gateway, two anchors and 18 regular nodes

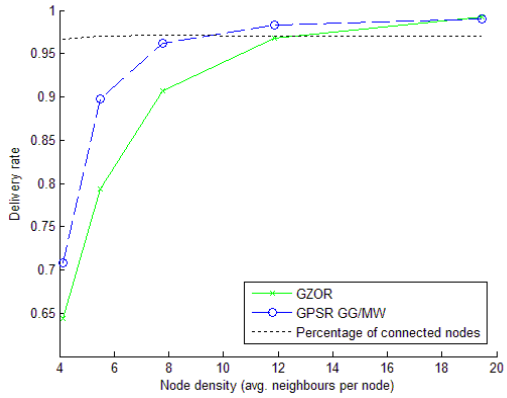

Figure 56: GBADM simulation results

APPENDIX C: LOCALIZATION

GPS position deviation is modelled as a two-dimensional Gaussian distributed error with a standard deviation of 1m. Since nodes localize themselves according to information they receive from their neighbours, their localization errors might have some correlation. It is important to model this correlation when we evaluate a routing algorithm since it may have some impact on its performance (this can be both positive or negative) [Wit06].
Therefore we simulate a localization algorithm based on the received signal strength (RSS) values that the Mica2 model provides us. As a result we should get realistic values of the localization error of the nodes, both in direction as in distance. These values are dependent on parameters such as network density and the amount of anchor nodes and should also show correlation between different nodes in the same area.
Note that no effort was put into optimizing the localization algorithm to obtain better results; we are only interested in obtaining realistic localization errors for a better routing evaluation.
For the localization process we use an algorithm based on the research described in [Sla07]. This algorithm uses a sequence of known algorithms to get a location estimation for each node. This sequence is DV-distance [NiN03] -> range sharing [Whi04] -> bounding box [Bia05].
This algorithm is initiated by the anchor nodes in the network; they broadcast their location estimation. Receivers of this broadcast measure their distance to this anchor through the RSS indication and broadcast this information over the network. Nodes that are not in the vicinity of anchor nodes will get information about the anchors through their neighbour nodes. When they receive such a broadcast, they add their distance from the broadcaster to information sent by the broadcaster about its anchors distances. In this way the node will get an estimate about the length of the shortest path reaching from anchor to itself. Scalability of this algorithm is assured by a time-to-life (TTL) value; an anchor broadcast message will only be forwarded TTL times.
Nodes optimize their distance estimations between each other through a range sharing algorithm, they try to reach mutual agreement on their distance from each other by sharing their ranging information and calculating the mean.
Ultimately, the nodes will have built an internal list of anchors and neighbours, with their locations and the estimated distance between themselves and the anchors. The nodes can use this list to calculate what their position is. This is in our case done by the bounding box algorithm. Every node draws a virtual box around every anchor from its list, with the size of its distance estimation to this anchor. All these boxes will have an overlapping area, the node estimates its own position at the centre of this area.
For a detailed overview of the localization algorithm we refer the reader to [Sla07].
In Figure 57 and Figure 20 the results of the localization phase of a random network constructed by the Group-based Air Deployment Model are presented. This simulation consist of 49 deployments on a grid of 180x180 m^2, each deployment consists of three anchor nodes and 18 normal nodes. The localization error closely resembles a Gaussian distribution with standard deviation of roughly 6 meters. Figure 20 shows that the localization errors are indeed correlated between the nodes. Nodes at the border of the network have a greater error than nodes in the centre, caused by the fact that they have less anchor information.
A by-product of localization is that every node has an idea about the surrounding network. It probably knows which gateways are in close proximity and it also has some information about the distance to its neighbours. This information is stored since it could be used to aid routing.

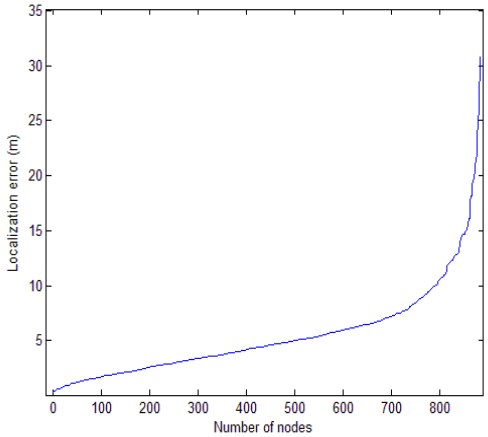

Figure 57: Distribution of localization errors.