UNIVERSITÉ DE GRENOBLE

No attribué par la bibliothèque

|_|_|_|_|_|_|_|_|_|_|

THÈSE

pour obtenir le grade de **DOCTEUR DE L'UDG**

Spécialité : Informatique

préparée au Laboratoire d'Informatique de Grenoble dans le cadre de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

préparée et soutenue publiquement par

Eryk Schiller

le 12 juillet 2010

Titre :

Les architectures des réseaux pour des environnements entierement sans fil

sous la direction de M. Andrzej Duda et M. Franck Rousseau

JURY

M. Yves Ledru	Président
M. Serge Fdida	Rapporteur
Mme. Isabelle Guérin-Lassous	Rapporteur
M. Guillaume Chelius	Examinateur
M. Andrzej Duda	Directeur de thèse
M. Franck Rousseau	Co-encadrant

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor and mentor Professor Andrzej Duda for his continuous support of my PhD studies and research, his expertise, leadership, enthusiasm, immense knowledge and for the sleepless nights we were working together before deadlines. His guidance helped me in all the time of my research and writing of this thesis.

I owe my deepest gratitude to Doctor Franck Rousseau for his advice, encouragement and insightful comments. I would like to show my gratitude to Doctor Paul Starzetz for the endless discussions and his motivation in the early stage of my research. I am also thankful to many of my colleagues from the Drakkar Team who supported me and believed in my success.

I offer my sincere regards to members of the thesis committee: Professor Isabelle Guérin Lassous from University of Lyon and Professor Serge Fdida from University Pierre and Marie Curie that they found some time in their busy schedules and accepted the role of external readers of my dissertation.

Last but not the least, I would like to thank my fiance and best friend Kasia. Without her love, help, editing and graphical assistance I would never accomplished my thesis. I also think about my whole family for their unconditional support and understanding of all my undertakings and decisions.

Contents

List of Figures

Part I

English

1 Introduction

Progress in electronics, telecommunications, and computer science during the XX century led us to the present situation in which electronic devices with wireless interfaces are cheap and accessible by everybody. Nowadays, many people possess mobile phones, smart phones, notebooks, and access points, but the access to the Internet is still archaic, as we explain below.

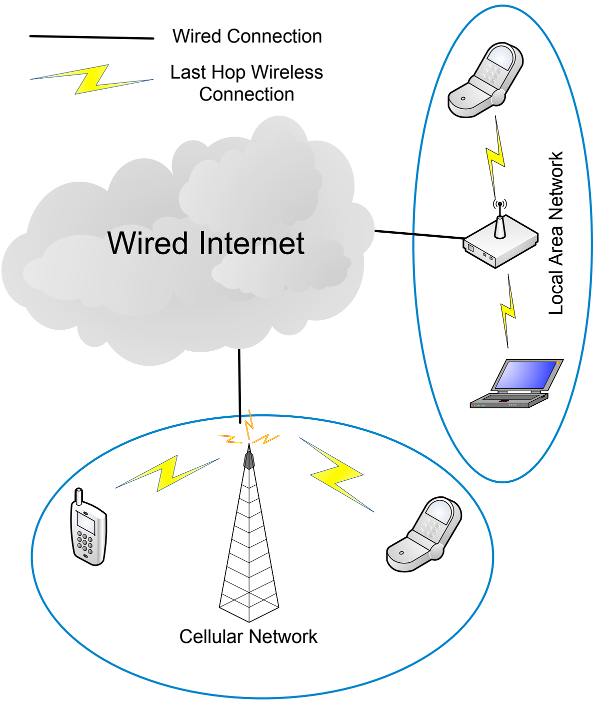

Figure 1.1 – Current view of the Internet

We have shown in Figure 1.1 that an access point in case of local area networks or a base transceiver station in case of cellular networks normally plays the role of interface between a wireless station and the wired Internet. Current solutions offer high accessibility, e.g., cellular telephony provides spatial coverage of about 98%, however, they suffer from many problems like not optimal routes, a traffic aggregation, and high load, because devices communicate over a wired infrastructure even if they are a few wireless hops away from each other. Wireless devices become more robust and powerful, and they are able to run as a wireless backhaul replacing the wired Internet in some situations.

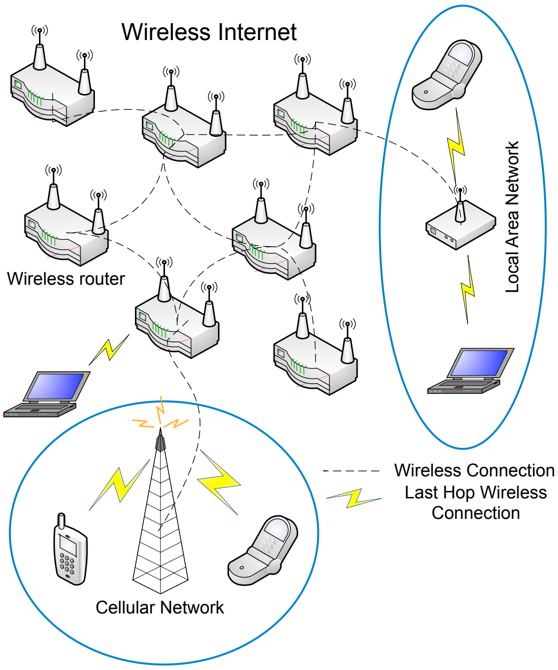

Figure 1.2 – Future view of the Internet

In this thesis, we explore possibilities of cooperation among wireless devices to develop a wireless Internet as shown in Figure 1.2. Wireless devices have completely different properties comparing with a wired equipment like: mobility, limited range and energy consumption, lower data rates and computational power. We need to redefine current standards and protocols to obtain a solution having minimal configuration needs, using simple algorithms, and offering high accessibility.

We study Spontaneous Wireless Mesh Networks that begin to emerge to provide wide coverage connectivity to mobile nodes. One of their characteristics is self-forming–users just add another mesh router at some place to increase the coverage. The organization of mesh networks needs to be autonomic to a great extent, because unlike the current Internet, such networks are not managed, so they cannot rely on highly skilled personnel for configuring, connecting, and running mesh routers. We assume that such mesh networks may be composed of a large number of routers providing connectivity to mobile nodes. Mesh routers benefit from abundant resources (memory, energy, computation power, GPS devices in some cases) and may only move occasionally. To provide a sufficient capacity to mobile stations, they may use multiple wireless network interfaces and different types of antennas such as sectorial or directional for increased space coverage. Neighboring mesh routers directly communicate over wireless links. To support end-to-end connectivity between any source and destination in the mesh, communication may require intermediate nodes that forward packets in a multi-hop way. We want to connect user devices to an unstructured flat network of wireless routers running without any commercial Internet providers nor supervision. We focus on the network layer and decentralized routing algorithms fulfilling all above requirements and providing robustness and good accessibility.

We start with the definition of the main elements in the network architecture. As in many fundamental contributions in this domain [1, 2, 3, 4, 5], we distinguish between the following entities:

node – it is an entity capable of communicating and computing. It may be mobile like a client terminal or stable as a mesh router.

end-point – it is a logical communicating entity corresponding to a single node.

end-point identifier (EID)– it is a short binary identifier for an end-point. It is persistent, i.e., it does not change when a node changes its position.

name – it is a human readable unique identifier associated with an EID.

address – it is a locator in the addressing space: a short binary identifier used for locating an end-point in the mesh network so that routers can forward packets to it. The address may change when a node changes its position and its network interface.

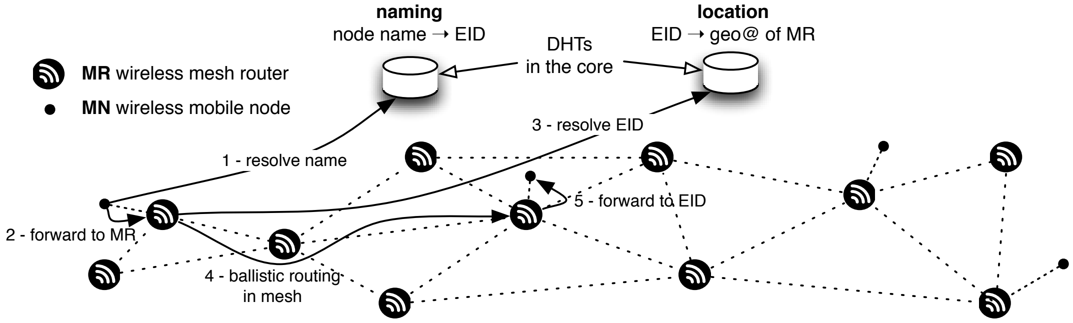

Figure 1.3 – Wireless mesh architecture

Figure 1.3 presents the architecture of the wireless mesh. Mesh routers form an interconnection infrastructure for mobile nodes moving inside and getting connectivity from the nearest mesh routers. The mesh network runs core services: the Naming Service and the Location Service. The naming service provides mapping between names and EIDs of nodes like the standard DNS service. The location service handles the information about mapping between EIDs and addresses. This approach to mobility is based on the assumption that the address reflects the position of a mesh router and other mobile nodes handled by the router. When a mobile node joins the mesh, it establishes a relation with a mesh router that knows its EID. The mesh router updates the location service with the mapping between its address and the EID of the mobile node. When another mobile node wants to send packets to the remote mobile node, it first resolves its name to get the corresponding EID and sends the packet with the destination EID to the neighboring mesh router, which in turn locates the address of the destination mesh router. Then, the packet takes a route to the destination address established by geographical routing. Finally, the destination mesh router forwards the packet to the destination mobile node. So, the communicating nodes only see packet forwarding between their respective EIDs and mesh routers make use of addresses to actually forward packets to the destination.

As EIDs are persistent, the transport layer can use them across different network interfaces and across different locations to maintain long-lived transport connections in spite of mobility. We can construct EIDs in several ways: we can derive them from a public key like in Host Identity Protocol (HIP) [5] or we can take into account a person that uses a node and derive EIDs from the public key of the user. A node may use multiple network interfaces, but a single EID.

1.1 Addressing space

We want to define a Coordinate Addressing Space used for assigning addresses to EIDs in an autonomous and distributed way. Such a coordinate space may be either a virtual or real one based for example on the GPS position of a node. Many proposals considered addressing and routing in virtual coordinate spaces [6, 7, 8]. Their advantage is relative ease of generation compared to the requirement of GPS positioning. However, they present several drawbacks. The first one is related to merging two subnetworks. Imagine that two regions of the networks develop in an independent and unplanned way. Each part creates its addresses in a given portion of the virtual subspace (for instance, P2P approaches for constructing a virtual space such as Content Addressable Network (CAN) [9] can be used). The problem is how we can merge two parts when we place a mesh router that interconnects them. One part needs to change its addresses to accommodate for the addresses of the other part. We can also face address clashes if parts allocate addresses from intersecting portions of the coordinate space. The second problem is that if we want to handle mobility, the distance in the addressing space needs to be correlated with the real world distance, because nodes physically move in the real space. When the distances in the addressing space reflect real movements, the most common mobility of a node will be located in its neighborhood, so changing addresses will only be limited to this part of the space.

If addresses are derived from the geographical positions of nodes, the problem of merging does not exist, because nodes in different locations use different addresses. However, this approach requires the knowledge of the exact positions of all nodes, which may be too difficult or too expensive to obtain. Several authors have already considered the problem of relative positioning. In the absence of GPS, for instance indoors, the location information can be obtained from relative positioning based on radio localization techniques described in Chapter 4.

1.2 Routing

Routers use geographical routing to forward packets. We propose an approach to reach far destinations that we call Waypoint Routing. Instead of the greedy geographical forwarding, we use additional information about the trajectories to the limited number of distant destinations called Waypoints. A router forwards packets along the best trajectory for a given destination. Each router keeps only partial information about trajectories, because a complete route is stored in the distributed manner. The set of trajectories is constructed and adapted based on the information from previously forwarded packets, so the protocol has no additional overhead. The Waypoint routing presented here is just an idea. The possible specifications of the Waypoint Routing are presented in Chapters 6 and 7 of this dissertation.

1.3 Organization of the Manuscript

This thesis has the following organization. Chapter 2 contains a brief description of the PHY and MAC layers for the most popular standards of wireless networking. In Chapter 3, we describe routing protocols starting from the classical specifications up to the most prominent examples of the protocols being very interesting from our network architecture point of view. In Chapter 4, we study the mathematical model of localization protocols. This is important, because localization allows a network address to be a geographical coordinate of node. Unlike the GPS localization, this technique may be cheap and energy efficient. We have developed a Monte-Carlo simulator able to handle large-scale networks to obtain the topological requirements for the model of wireless mesh networks that aim at network-wide localization. In Chapter 5, we focus on the greedy geographical routing in spontaneous wireless mesh networks to show several interesting properties. The greedy geographic routing still remains one of the building blocks of many geographical algorithms. First, we can approximate the dependence of the mean node degree on the packet loss probability with a Sigmoid function. When the mesh network grows, it becomes opaque to packets, i.e., the probability of message delivery between distant nodes tends to 0, regardless of the average node degree. We also show that the packet loss probability in mesh networks with the greedy geographical routing does not exhibit the behavior of percolating systems. Finally, we propose an analytical model of the greedy geographical forwarding and we use it to derive the packet loss probability. In Chapter 6, we propose Binary Waypoint Routing, a novel geographical routing protocol for wireless mesh networks. Its idea is to learn and maintain source routes to a small number of nodes called waypoints, placed in subspaces constructed as a result of binary space partitioning. A source node sends a packet to a waypoint for a given destination and intermediate nodes try to adapt a packet route by aiming at waypoints that are closer to the destination. Our simulation results show that the proposed scheme achieves a high packet delivery rate with a traffic pattern similar to the Optimal Shortest Path Routing. In Chapter 7, we propose a scalable geographical routing protocol for dynamic multi-hop wireless networks. Unlike the previous scheme, our protocol does not require source routing. The volume of routing information in any node remains very small compared to the total size of the network, because the number of waypoints grows in a logarithmic way with the number of nodes and the volume of the routing information is limited. Our simulation results show that the proposed scheme achieves a high packet delivery rate and the distribution of route length similar to the Optimal Shortest Path. We end with some conclusions and perspectives for the future work.

2

A Survey on Wireless Networks

In this thesis, we investigate new possibilities of addressing and routing in Wireless Mesh Networks (WMNs). A Wireless Mesh Network [10] is a communication network of mesh topology containing wireless stations with radio interfaces. Each station may use wireless devices of different vendors and standards. We can distinguish WMNs' basic components like wireless routers, i.e., special devices constructed for packet forwarding and wireless client stations like laptops or mobile phones without routing capabilities. Normally, routers are much more stable than client stations and they move occasionally.

Wireless technologies are very popular. Many different vendors exist offering specialized radio hardware. Different devices are able to communicate if they follow the same standards. In this chapter, we present the most popular wireless communication standards defined by the Institute of Electrical and Electronics Engineers (IEEE) for networks of different scales and purposes. They define low level mechanisms that permit wireless devices to exchange information, i.e., they provide definitions of radio transmitters and data transfer mechanisms: a physical (PHY) layer so that one device is able to successfully decode a signal emitted by other device if received signal power exceeds a desired device limit and a Media Access Control (MAC) layer defining sharing rules for several terminals or a network of a multi-point structure such as a LAN or a MAN. IEEE worked out popular standards of PHYs and MACs for wireless networking starting from Personal Area Networks (PANs) like IEEE 802.15.4 [11] or Bluetooth [12], Local Area Networks (LANs) like IEEE 802.11 [13] and Metropolitan Area Networks (MANs) like IEEE 802.16 [14].

PANs are suitable for a personal use, i.e., for creating a link between a mobile phone and a headset or among devices located nearby. Operational coverage does not exceed 10 meters. LANs may connect a group of people located at the same place like at an office or inside a building. The coverage of this network does not exceed 50 meters. MANs are networks for connecting a large number of people, so their operational range is measured in kilometers. In this thesis, we are interested in the large scale wireless mesh networks built out of LANs.

2.1 Bluetooth 1.2

Bluetooth 1.2 defined by the IEEE 802.15.1 standard works in the 2400-2483.5 MHz Industrial, Scientific, Medical (ISM) band and provides a method of communication among low data rate devices. Bluetooth has 79 channels spaced by 1 MHz and uses the Gaussian Frequency Shift Keying (GFSK) modulation offering bit rates up to 1 Mbps. A data transmission is made reliable by Forward Error Correction (FEC) codes and Automatic Repeat reQests (ARQs). There are three power classes of transmitters allowing communication among devices from 10 cm up to 100 m. Bluetooth uses frequency hopping techniques over 79 separated channels and additionally a Time-Division Duplex (TDD) to allow simultaneous communication among different devices. Bluetooth specifies a connection oriented and connection-less MAC. There are two types of communication links: Synchronous Connection-Oriented (SCO) and Asynchronous Connection-Less (ACL) providing data rates up to respectively 64 kbps and 723.2 kbps. Notice that the 723.2 kbps mode uses the 1 Mbps modulation, but the offered throughput is reduced due to the protocol overhead. There is no native Internet Protocol (IP) support and the IP traffic can be exchanged using a serial port emulation and the Point-to-Point (PPP) protocol. A synchronized entity of 1 master device and 1 up to 6 active slaves forms a piconet. In Figure 2.1, we present the possible topologies of a Bluetooth network.

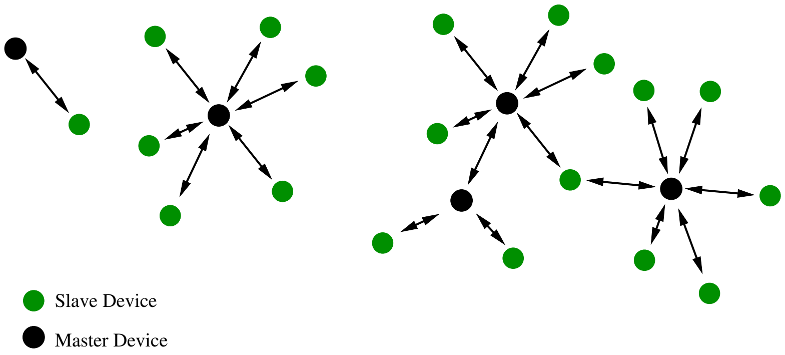

Figure 2.1 – Examples of 802.15.1 networks

2.1.1 Bluetooth 2.0

This version of the Bluetooth specification was released on November 10, 2004 [15]. The Enhanced Data Rate (EDR) is an optional part of the Bluetooth specification providing higher data rates. This version has significant advantages compared to the older version 1.2. Bluetooth 2.0 uses two additional modulations for data transmissions: Differential Quadrature Phase Shift Keying (DQPSK) and 8 Differential Phase Shift Keying (8DPSK) carrying respectively 2 and 3 bits per symbol. When the 8DPSK mode is used, the throughput is tripled in comparison to Bluetooth 1.2. Bluetooth 2.0 is able to achieve data rates of around 2.1 Mbps and it is backward compatible with the previous Bluetooth version.

2.1.2 Bluetooth 3.0

The 3.0 High Speed (HS) specification was released on April 21, 2009 [16]. The most important improvement is an introduction of a new high-speed transport mechanism–the Alternate MAC/PHY (AMP). Bluetooth 3.0 gets its higher data rates from the IEEE 802.11 Protocol Adaptation Layer (PAL). IEEE 802.11 offers a 54 Mbps PHY with approximate throughput of around 24 Mbps due to the MAC overhead. Bluetooth devices use the older versions of the Bluetooth protocol for device discovery in the initial phase of connection, however, when the user sends lots of data, the higher throughput is achieved due to a momentary use of the IEEE 802.11 protocol.

2.1.3 Bluetooth 4.0

The Bluetooth Special Interest Group (SIG) revealed new innovations in December 2009 [17] to feature its next version of Bluetooth. Unlike 3.0, which concentrates on throughput improvements, the updated 4.0 specification aims to consume less power.

2.2 802.15.4

IEEE 802.15.4 provides a PHY and a MAC for network devices requiring low data rates and low energy consumptions. It defines the PHY operating at 868-868.3 MHz (1 channel), 902-928 MHz (10 channels) and 2400-2483.5 MHz (16 channels) with BPSK, ASK and QPSK modulations and data rates varying from 20 to 250 kbps. It supports Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) and an optional allocation of Guaranteed Time Slots (GTSs). The IEEE 802.15.4 network contains two different device types: Full-Function Devices (FFDs) and Reduce-Function Devices (RFDs). FFDs can talk to any other device, they implement the whole IEEE 802.15.4 standard and may act as a PAN coordinator while RFDs can only serve as devices supporting reduced functionality. It is an opportunity, because RFDs may implement an ultra-light micro-code with extremely low complexity. A PAN coordinator has some additional functionality and acts as a central point of the whole network, e.g., allowing devices to join it. IEEE 802.15.4 supports two topologies: star and mesh. In star topology, the PAN coordinator is a central point of the network and other devices are its leafs. Other nodes can only communicate with the PAN coordinator. In mesh topology this mode of communication is also possible, but additionally, nodes may directly communicate within their coverage using the CSMA-CA mechanism. In Figure 2.2, we present the possible IEEE 802.15.4 topologies. A Cluster Tree Network looks like a real decentralized mesh network, however, it still requires the global PAN coordinator.

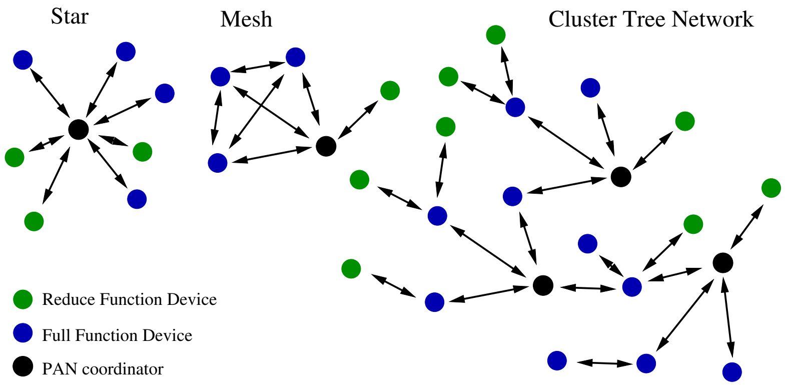

Figure 2.2 – Examples of 802.15.4 topologies

2.3 802.11

IEEE 802.11 [13] is a popular and mature standard of WLANs. The standard defines a physical layer able to work with a highly efficient Orthogonal Frequency-Division Multiplexing (OFDM) technique that transmits at data rates up to 54 Mbps using different modulations and FEC codes with different coding rates. OFDM efficiently uses a spectrum and minimizes multipath fading by dividing the spectrum into narrowband subcarriers. The standard also defines older techniques able to work with lower data rates (1 Mbps up to 11 Mbps) and an infrared access method. When the network does not contain older devices not supporting the OFDM PHY, the OFDM mode is used. In case of the 2.4 GHz PHY [18, 13], presence of older devices enforces the network to use lower data rates. Radio devices operate in two different spectrum ranges: the first region covers the 2400 MHz-2483.5 MHz band with 13 overlapping and 3 orthogonal channels of 22 MHz and one additional channel for Japan centered at 2484 MHz. The second band is located close to 5 GHz. The standard exactly precises a channel list allowed in a particular geographical area. The ISM band is busy, so it is good to switch to 5 GHz, however, a transmission in this band may be affected by a stronger signal attenuation.

IEEE 802.11 networks come in two main configurations called Extended Service Set (ESS) and Independent Basic Service Set (IBSS). A cellular architecture of an ESS contains different Basic Service Sets (BSSs) connected to a Distribution System (DS). We can use a wired or another wireless network as the DS. An Access Point (AP) along with a set of Stations (STA) associated with it form the BSS. Generally, we say that the ESS is a set of STAs associated with the APs connected by some sort of medium. The IBSS is another type of architecture. It is a set of STAs without any wired infrastructure. STAs in range may directly communicate while multi-hop communication requires additional routing functionality.

IEEE 802.11 supports a full connection-less MAC layer. There are two types of MAC coordinating functions: a Distributed Coordination Function (DCF) and a Point Coordinated Function (PCF). The first one is a fully distributed mechanism based on CSMA-CA implemented on the majority of 802.11 devices. The PCF is only implemented by a few vendors. When a BSS works under the PCF, the AP acts as a central channel coordinator.

2.4 802.11n

IEEE 802.11n [19] is an addition to the IEEE 802.11 standard family. It provides a significant throughput increase. The highest data rate was improved from 54 Mbit/s to 600 Mbit/s. IEEE 802.11n adds a Multiple-Input Multiple-Output technique (MIMO) and 40 MHz channels to the PHY and a frame aggregation to the MAC layer. MIMO uses multiple antennas to improve a communication performance. It includes sending different data streams at the same time over several antennas. Every receiving device obtains a different outcome from each of receiving antennas due to the multipath signal propagation. It is then able to decode each of the transmitted data streams, which is known as Spatial Division Multiplexing (SDM). The 802.11 MAC protocol specifies the fixed overhead of interframe spaces. When a device uses the 600 Mbit/s mode with MIMO, a data frame duration is much shorter than interframe spaces introduced by the contention avoidance mechanism, so each device waits longer for the channel than uses it. In order to improve a channel efficiency, the standard introduces a frame aggregation sending many data frames during a single transmission. The 802.11n standard allows a 4x4 MIMO transmission in which both transmitting and receiving nodes use 4 antennas to simultaneously transport 4 different streams.

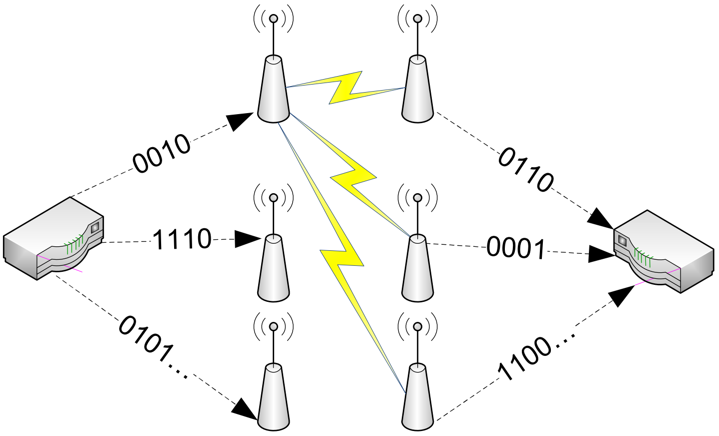

Figure 2.3 – Node to node communication using the MIMO technique

2.5 802.16

IEEE 802.16 [14] is a network standard designed for MANs. The standard defines two bands: 10-66 GHz and 2.5-11 GHz. Communication in the first band requires a Line-of-Sight (LOS) and may be affected by a strong attenuation. Because of these problems, the 802.16 committee decided to provide additional communication in the 2.5-11 GHz band without the Line-of-Sight requirement (NLOS). The IEEE 802.16 networks use a Frequency Division Duplex (FDD) with highly efficient OFDM and Time Division Multiple Access (TDMA). The maximal theoretical throughput in the 10-66 GHz band is 144 Mbps and 70 Mbps in 2.5-11 GHz. The 10-66 GHz spectrum is divided into constant 20 MHz channels while 2.5-11 GHz band is divided into variable length channels up to 20 MHz wide. A network can operate in the Point-to-Point (PtP), Point-to-Multipoint (PtMP) or Mesh architectures, so it is highly capable to work as a multihop wireless mesh network.

2.6 Summary

In this chapter, we have presented different versions of wireless standards able to work in a multihop network. Bluetooth is not suitable to provide a wireless Internet due to its limitations: no native IP support, low data rates and a limited number of devices in a single piconet. It may rather serve as a Wireless USB to exchange a small amount of data among dedicated devices like headsets or modems. IEEE 802.15.4 low data rates and a centralized architecture are suitable for a PAN, but it is not designed to work as a large scale Wireless Mesh Network. Its advantage is that TCP/IP may be natively used, but a limited packet payload up to 127 bytes may be a problem, because, e.g., TCP/IPv4 header of about 40 bytes seems to be too long for the IEEE 802.15.4 network. The IEEE 802.11 and IEEE 802.16 networks are fully capable to work in a multihop architecture, but they require an external routing algorithm to forward packets from different sources to destinations. In the next chapter, we focus on routing protocols.

3 Routing Protocols

In the previous chapter, we have described the standards that define wireless networks suitable to work in mesh topology. Now, we study upper layer protocols that deliver information between different points in a network not necessarily directly connected. Routing Protocols are responsible for constructing information required for packet forwarding. Nodes exchange routing messages and store the necessary information in their routing tables. Nodes use the information from routing tables to forward packets and deliver them from their origins to destinations. There are many routing protocols designed for wireless networks, e.g., AODV [20], DSDV [21], DSR [22], OLSR [23], GFG [24], GPSR [25], and many others. Protocols may be divided into several categories based on different criteria. We describe the most important routing protocol families.

- Protocols based on a distance vector strategy in which each node maintains a vector of distances to all nodes in the network. A forwarding node selects a next hop using the Bellman-Ford algorithm.
- Protocols based on a link state approach in which each node in the network has all information on the network topology. To keep the information up-to-date, each node disseminates its immediate neighborhood. A forwarding node runs the Dijkstra algorithm over the connectivity graph to compute a next hop.
- Geographical routing protocols benefit from a node position and use coordinates to forward packets. Unlike traditional algorithms they do not use graph topology.
- Proactive or table-driven routing protocols maintain paths between all nodes in the network even if the paths are not used and not needed. Keeping network topology up-to-date requires periodical updates.
- Reactive protocols establish a path on demand, when nodes wish to communicate. This strategy slows down the forwarding of the first packet in comparison with proactive protocols.

3.1 Bellman-Ford Algorithm

We start our discussion with one of the earliest distance vector routing protocols called Distributed Asynchronous Bellman-Ford (DBF) Algorithm [26]. Imagine that each link between two neighbors i, j has non-negative weight w_{ij}. Node i maintains distance vector d_{ij}^k, where j varies over all immediate neighbors, and k varies over all known destinations. d_{ij}^k is the minimal distance from node i to node k via immediate neighbor j. At the beginning routing tables are empty. Routing agents periodically advertise their minimal distances to all known destinations $\forall_l \left\{l,\, d_i^l = \min_j d_{ij}^l\right\}$. Suppose that node i receives a routing message from neighbor j. The message contains minimal distances d_j^k to remote destinations k. Node i updates its distance vector using the received information increased by the link weight $d_{ij}^k = d_j^k + w_{ij}$. When node i forwards a data packet to destination k, it sends the packet to the closest neighbor to the destination j such that: $d_{ij}^k = \min_l d_{il}^k$, what guarantees the shortest path delivery.

3.1.1 Count-To-Infinity

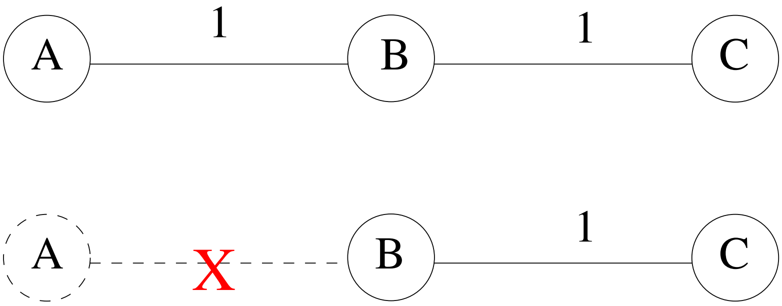

Figure 3.1 – Count-to-Infinity Problem

The major problem of the Distributed Bellman-Ford algorithm is referred as count-to-infinity. Imagine the situation presented in Figure 3.1 in which the network converged. Node B recognizes node A as its immediate neighbor. The distance between A and B is 1. Node C finds the shortest path to A via B with the distance equal to 2 and advertises it to B. B discards this advertisement, because it has shorter path to A. The situation changes dramatically when link AB disappears. B deletes immediate neighbor A from its routing table, however, it still gets advertisements from C about the route to A. B adds the route to A via C of length 3 even though the route is no longer valid. C notices that B changes its distance to A and updates it respectively to 4. Nodes B and C count-to-infinity advertising a non-existing route.

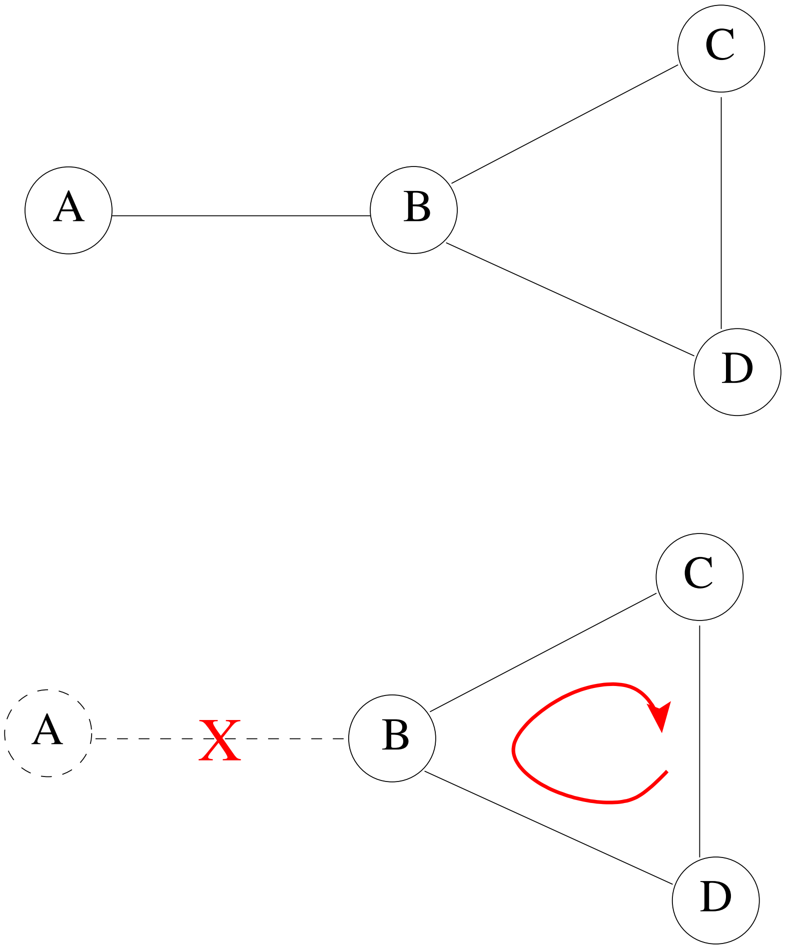

Figure 3.2 – Count-to-Infinity in looped topology

3.1.2 Split horizon, split horizon with with poison reverse

A split horizon [27] is a partial solution to the count-to-infinity problem. The idea is that a node does not advertise a route on the link it forwards on. Consider once again the example in Figure 3.1. The difference between the previous situation is that node C running DBF with the split horizon does not inform B that it has a path to A, because B is its predecessor to A. The split horizon eliminates counting-to-infinity in this case. The split horizon does not eliminate all count-to-infinity problems in the network. Let us consider a network of looped topology in Figure 3.2. Imagine that B and D discard path to A, but node C still announces it. In this case D and finally B store this path in their routing tables. B-C-D still count-to-infinity even if they use the split horizon approach.

3.2 DSDV

DSDV is a distance vector protocol based on the DBF algorithm. DBF suffers from the count-to-infinity problem in dynamic networks. There are different partial solutions to this problem, e.g., a split horizon, however, performance of the split horizon in a dynamic environment of looped topology is low and it requires Point-to-Point (PtP) links. Wireless networks seem to be highly dynamic of looped topology, wireless links are not PtP, because every immediate neighbor hears a routing message sent by the neighboring router. Distance-Vector Routing (DSDV) [21] maintains a sequence number to discard stale information. Each change in the topology increases the sequence number for a given destination. Nodes prefer up-to-date information, so the obsolete routing entries with older sequence numbers are discarded all over the network.

3.3 AODV

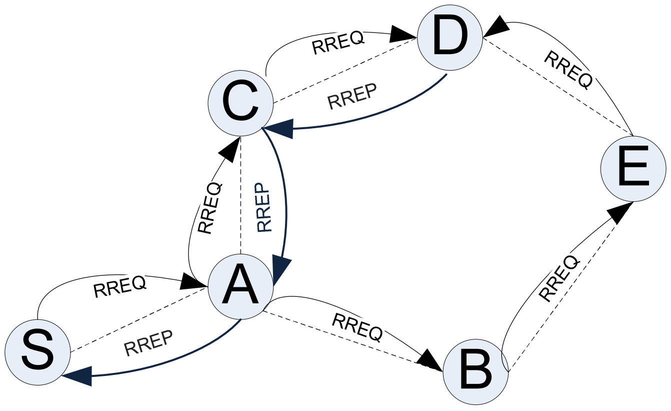

Figure 3.3 – Path establishing process under the AODV protocol

Ad hoc On-Demand Distance Vector (AODV) Routing [20] is a reactive routing protocol establishing a path from a source to a destination node when the path is required. When a node looks for a new path, it broadcasts a Route Request (RREQ) packet. A receiving node forwards RREQ packets if none of the following conditions is satisfied: it possesses a valid route to the destination, it has seen the same packet before or it is the destination for the RREQ packet. The forwarding node keeps the best predecessor of the RREQ message (i.e., the node that delivered RREQ in the smallest number of hops, counting from the origin of the message). When the receiving node has a valid route to the destination or it is a destination point by itself, it returns the unicast Route Reply Message (RREP) to the RREQ origin node using the predecessor pointers established before. Each node on the RREP path updates its routing table to be able to forward packets to the RREQ destination point. The whole idea suffers from routing loops, because nodes may use the obsolete information while computing next hops. AODV maintains sequence numbers in order to discard stale information.

3.4 DSR

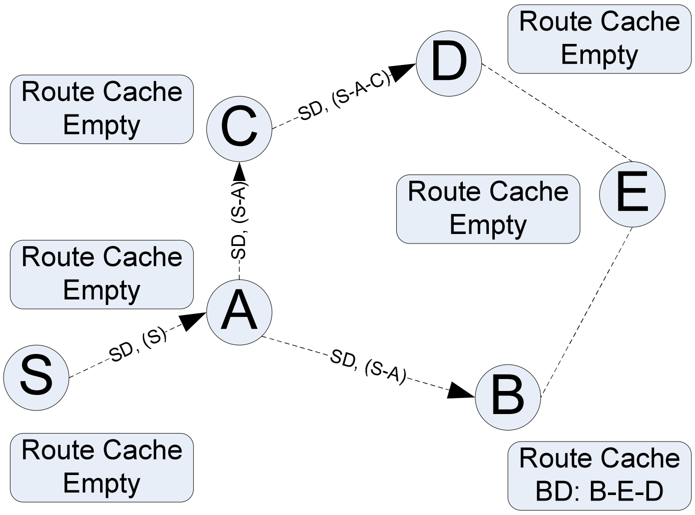

Figure 3.4 – Source node sends a DSR route request

The Dynamic Source Routing Protocol (DSR) [22] for Mobile Ad Hoc Networks is also a reactive routing protocol. Imagine that node S needs to send packets to a distant node. If S does not have a path to the destination, it broadcasts a RREQ packet. The idea is that every node receiving RREQ that does not have a valid source route to the destination forwards the request, however, multiple instances of the same message are discarded. Each RREQ contains a complete list of all intermediate forwarding nodes, so every node benefits from the request getting paths to reach all predecessors. When the RREQ packet reaches the destination or other node that already knows the valid path to the destination, a unicast RREP is returned to the origin of the RREQ. The RREP packet contains a valid source route from the origin to the destination, so the intermediate nodes can also use this information updating their routing tables.

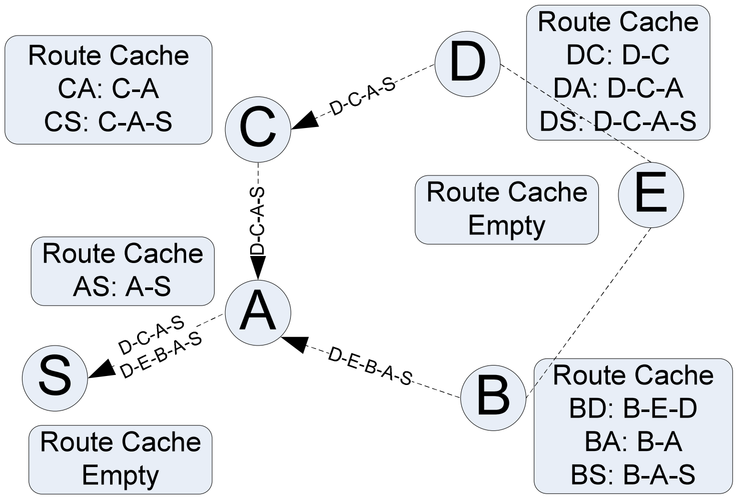

Figure 3.5 – Source node obtains a DSR route reply

3.5 OLSR

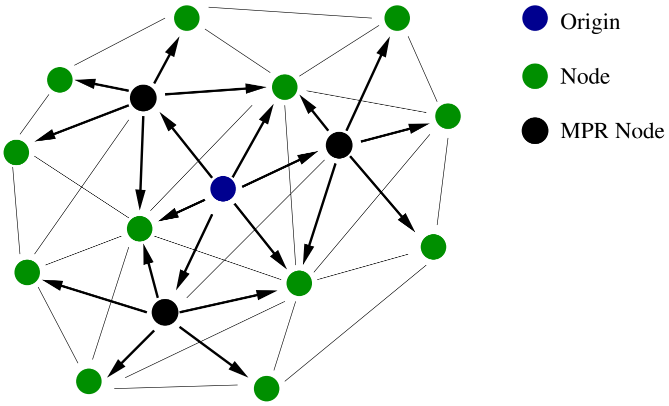

Figure 3.6 – MPR optimized flooding scheme

Optimized Link State Routing Protocol (OLSR) [23] is an adaptation of a Link State Routing protocol for Wireless ad-hoc Networks. In a link state protocol, every node has a global view on the network topology and runs a shortest path algorithm, e.g., the Dijkstra algorithm over the network connectivity graph. The idea is that each node announces every change in the local neighborhood to all nodes in the network in an optimized way. Each node discovers a 2-hop neighborhood and some of the immediate neighbors become Multipoint Relays (MPRs). Each node from the 2-hop neighborhood is accessible through the set of MPRs. Only MPRs forward topology information, while the ordinary nodes do not. This method guarantees that OLSR can efficiently spread the topology information through the network using a set of locally established MPRs.

3.6 Geographical routing protocols

The previously described protocols adapt classical ideas known from the wired Internet to wireless networks. The wired Internet is a computer network with interesting properties. The Internet follows the preferential attachment rule and some of its parameters like, e.g., the node degree exhibit a power law behavior [28]. When the node degree behaves like a power function, there are nodes in the Internet having much more neighbors than the ordinary node. These nodes play the role of hubs exchanging lots of data among different sources and destinations. The notion of a communication range does not exist, because a packet may advance 1000s kilometers in a single hop. There are also pairs of nodes geographically close to each other exchanging information via distant intermediate nodes. Wireless Mesh Networks are different. They contain devices with wireless interfaces characterized by a small communication range. In our opinion Random Geometric Graphs (RGGs) may be used as a first approximation of WMNs. The Random Geometric Graph is a graph in which all nodes have equal circular coverage and the position of each node is taken from a uniform random distribution. A link between two nodes exists iff one node belongs to the coverage of the second node and the other way around.

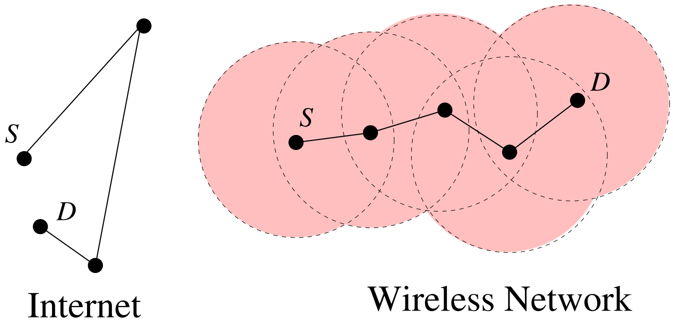

Figure 3.7 – Typical paths in the Internet and in Wireless Mesh Networks

When two wireless devices lie very close to each other, they can directly exchange information, which is not necessarily true in case of the wired Internet. When the direct communication is not possible because of the distance between a source and a destination node, then a set of intermediate nodes is necessary to forward packets. In order to reach the destination, a packet, roughly speaking, approaches the destination in terms of the Euclidean distance. This is a motivation for another class of routing protocols suitable for Wireless Mesh Networks called geographical routing protocols.

These protocols built upon information about geographical coordinates of nodes, which can be provided by means of global positioning techniques such as GPS. In the absence of GPS, for instance indoors, the location information can be obtained from positioning based estimation of the signal strength [6] or through relative positioning of nodes with respect to a few nodes with the exact geographical position. Imagine an analogy to road traffic in which we want to establish a route from Lyon to Marseilles using the system of motorways in France. A classical routing algorithm would consider all possible paths via Paris or Rennes, but geographical protocols choose a route leading to the south. This is a good choice, because in this case the motorway directly leads to the destination. Geographical routing is particularly interesting for spontaneous wireless mesh networks: it does not require any information on the global topology, but rather it uses the position of the destination to forward packets.

3.7 Cartesian Routing

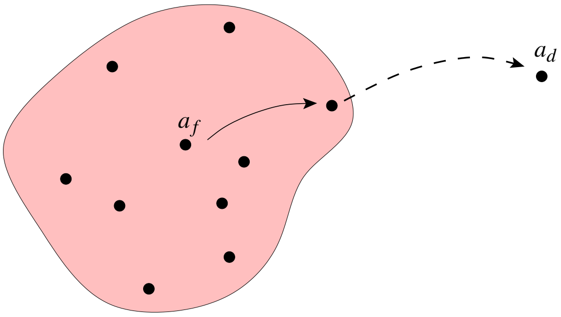

Figure 3.8 – Greedy forwarding: node a_f chooses a next hop for a_d

Algorithm 1 Greedy(a_d)

```
$d_{min} \leftarrow d(current, a_d)$
$next\_hop \leftarrow$ NULL
**for all** $n \in Neighbors$ **do**
    **if** $d(n, a_d) < d_{min}$ **then**
        $next\_hop \leftarrow n$
        $d_{min} \leftarrow d(n, a_d)$
    **end if**
**end for**
**return** $next\_hop$
```

Finn [29] did a pioneering work in this domain by pointing out main problems of topological routing algorithms and proposing a method called Cartesian Routing. To the best of our knowledge this is the first geographical routing protocol. Cartesian Routing introduces greedy forwarding in which a node forwards a packet to the neighbor that has the least distance to the destination as shown in Figure 3.8. The greedy forwarding guarantees a loop-free operation, but it only works for high density networks without voids or obstacles–a concave node that has no further neighbors closer to the destination may need to drop a packet as shown in Figure 3.9. Concave nodes usually appear at some places close to voids–uncovered areas or obstacles to radio waves in a given direction. In order to cope with that problem Cartesian Routing implements limited flooding to find a closer intermediate nodes when no neighbor lies in the direction of the destination. This solution suffers from heavy traffic around obstacles.

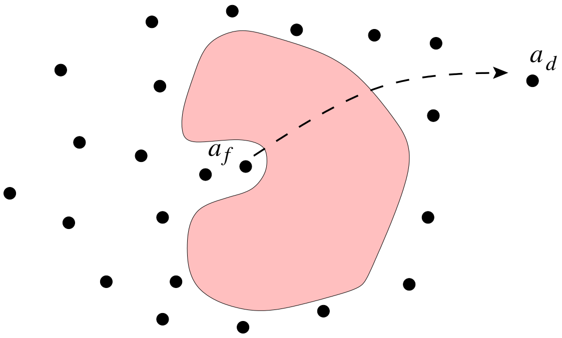

Figure 3.9 – Greedy forwarding fails at the border of the concave region

3.8 Compass routing II, Face Routing

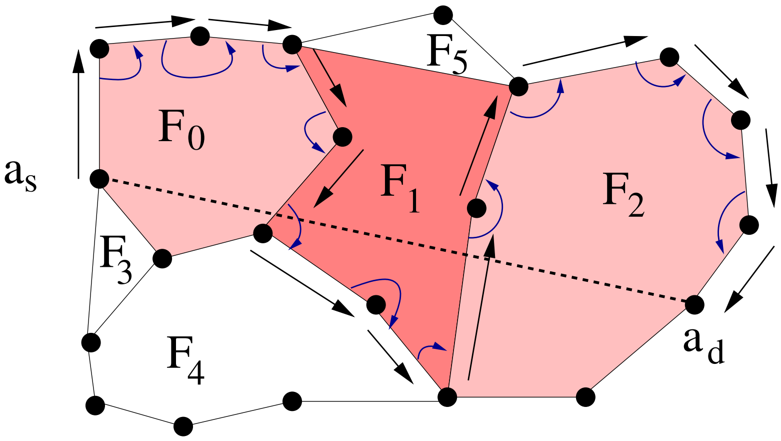

Figure 3.10 – Face routing

A large amount of work considered methods for coping with voids, and face routing is one of the first solutions, however, the face routing requires a construction of a plane graph. A planar graph is a graph that can be drawn in the two dimensional plane without intersecting edges. A plane graph or planar embedding of the graph is a planar graph drawn in the plane. A face is a cycle in a plane graph that does not contain nodes nor edges in its interior. In Figure 3.10, we see a plane graph containing 6 faces: $F_0, F_1, \ldots, F_5$.

Kranakis et al. [30] proposed Compas Routing II or face routing to surround void areas in plane graphs. Let us consider the situation in Figure 3.10 in which source node a_s sends a packet to destination node a_d. At the beginning nodes forward the packet along F_0 boundary. To do it, a_s chooses the first anticlockwise link starting from section $\overline{a_s, a_d}$ and intermediate nodes forward the packet to the first anticlockwise link counting from the predecessor. This procedure moves the packet around the face and the packet may arrive at the source node once again. To prevent from this situation Compass Routing II switches the currently surrounded face to the face closer the destination when the last encountered edge intersects with $\overline{a_s, a_d}$. In Figure 3.10, we see an immediate switch from F_0 to F_1 and F_1 to F_2 when it happened. To implement it, nodes change the method of selecting next hops from anticlockwise to clockwise or vice versa after each intersection. A packet explores several faces and finally reaches the destination.

Algorithm 2 Face-Routing(a_d, $predecessor$, $clockwise$)

```
$\alpha_{min} \leftarrow \infty$
$next\_hop \leftarrow$ NULL
if $predecessor$ = NULL then
    $predecessor \leftarrow a_d$
end if
if intersects($predecessor$, $current$, $a_s$, $a_d$) = TRUE then
    $clockwise \leftarrow \neg clockwise$
end if
for all $n \in Neighbors$ do
    if angle($predecessor$, $n$, $clockwise$) < $\alpha_{min}$ then
        $next\_hop \leftarrow n$
        $\alpha_{min} \leftarrow$ angle($predecessor$, $n$, $clockwise$)
    end if
end for
return ($next\_hop$, $clockwise$)
```

3.9 Local Planarization

Bose et al. [24] provide a method of a local planarization algorithm working under the Unit Disc Graph (UDG) assumption, which is a common model for representing multi-hop wireless networks. The UDG contains vertices with geographical coordinates a_i. An edge between two of vertices a_i and a_j exists iff the Euclidean distance $|a_i, a_j| < R$, where R is a constant range. Now, we provide important definitions and explain how to construct a plane graph.

3.9.1 Euclidean Minimum Spanning Tree

Euclidean Minimum Spanning Tree (EMST) for set P of points is a minimum spanning tree in which a weight of each link is equal to the Euclidean distance between the vertices.

3.9.2 Delaunay Triangulation

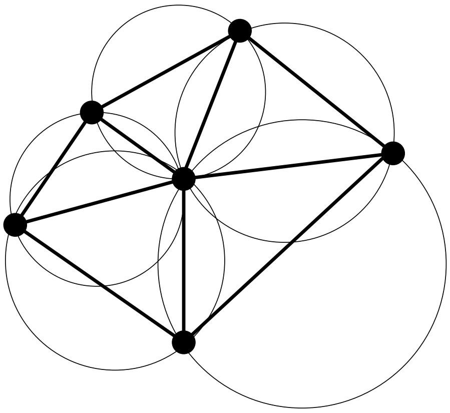

Figure 3.11 – Typical Delaunay Triangulation

We say that a triangulation of the 2 dimensional plane is its subdivision into disjoint triangles. Delaunay Triangulation DT(P) [31] for set P of points contains triangles built out of three vertices connected by edges. Each circle circumscribing any triangle that belongs to DT(P) does not contain in its interior any other points from P. By definition DT(P) is planar.

3.9.3 Voronoi Diagrams

Voronoi Diagrams [31] for set P of points are dual structures to Delaunay Triangulation DT(P) and they divide a plane into convex Voronoi Cells (a set is convex iff

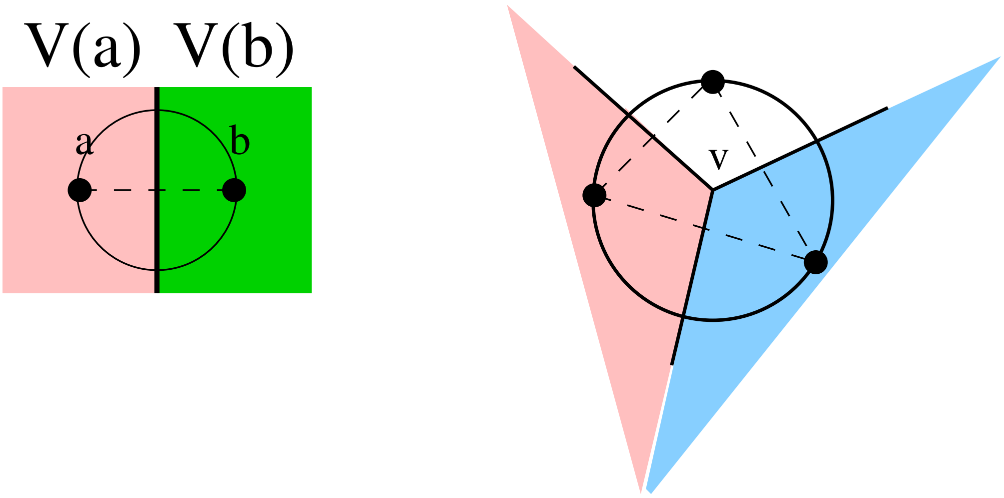

Figure 3.12 – Voronoi diagrams

it contains all lines connecting any points from the set). Some point $x \in \mathcal{R}^2$ belongs to the Voronoi cell of point $a \in P$ iff the Euclidean distance between x and a: $|x, a|$ is smaller than the distance between x and any other point from P. In Figure 3.12, we present the subdivisions of a plane into Voronoi Cells. In the left picture, P only contains 2 points, so a straight line separates two cells $V(a)$ and $V(b)$. In the general case, two Voronoi cells border with each other iff a circle centered in the middle between a and b of diameter $|a, b|$ does not contain other vertices in its interior. The right picture shows a Voronoi Diagram of 3 points. The distance from Voronoi vertex v to all three points from P is the same, so v is a center of a circle circumscribing these points. Now, we connect points from the neighboring cells to obtain a triangle circumscribed by a circle centered at v. In the non-degenerated case when P contains more points, each vertex borders at most with three Voronoi cells. We draw the circles centered at the vertices. The radius of each circle is equal to the distance between the vertex and the closest points from P. We connect points from P from neighboring cells (we test each pair of points by the circle as explained in the previous example with 2 points) by links to get a set of triangles (i.e., a border of a triangle contains 3 links) circumscribed by circles so that each circle does not contain other points from P in its interior (only three points on the circumference). This statement defines the Delaunay Triangulation, so graph $G(P, E)$ is planar when E is a set of links among the neighboring cells.

3.9.4 Relative Neighborhood Graphs and Gabriel Graphs

We introduce two families of graphs: Gabriel Graphs (GGs) and Relative Neighborhood Graphs (RNGs) [31, 32, 33]. These graphs are important because of their properties. For RNGs or GGs, a link between two vertices exists under certain conditions (cf., Figure 3.13). For RNGs, we take two circles centered at two different vertices.

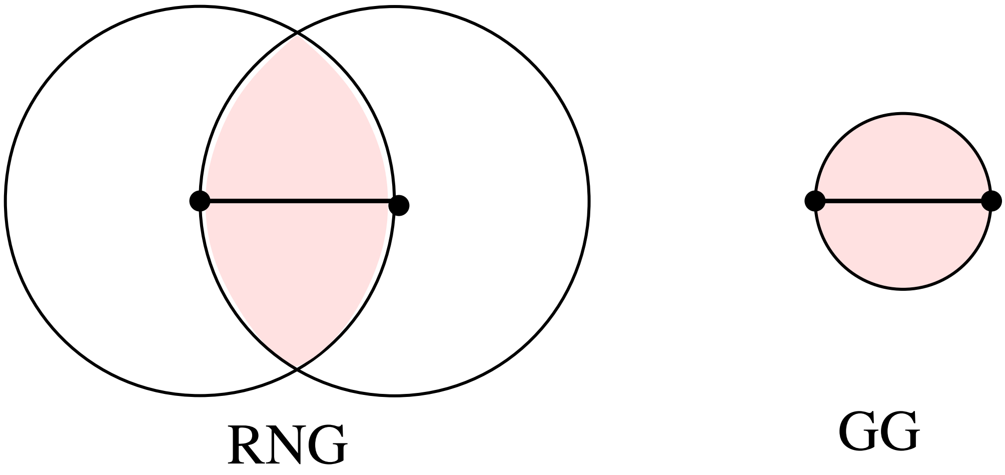

Figure 3.13 – Construction of RNG and GG

The radius of each circle is equal to the distance between the vertices. An intersection of two circles form a lune. We say that a link between two vertices exists in the RNG iff the lune is empty and does not contain other nodes inside. For GGs, we consider a circle centered in the middle between two vertices. The diameter of the circle is equal to the Euclidean distance between the vertices. If the circle is empty, then the link between the vertices exists. It implies that $GG \subseteq DT$ (cf., Subsection 3.9.3). The RNG lune contains the GG circle, so $RNG \subseteq GG$, because some links existing in GGs do not exist in RNGs. This is important issue, because it means that both RNGs and GGs are plane graphs. Assume that we have three points as shown in the left picture in Figure 3.14 and we want to construct a spanning tree. In this case, the spanning tree contains maximally two edges. The horizontal edge is the longest edge from the set of possible connections, so it does not belong to the spanning tree and it does not exist in the RNG. However, when distances among all three nodes are equal, then all links

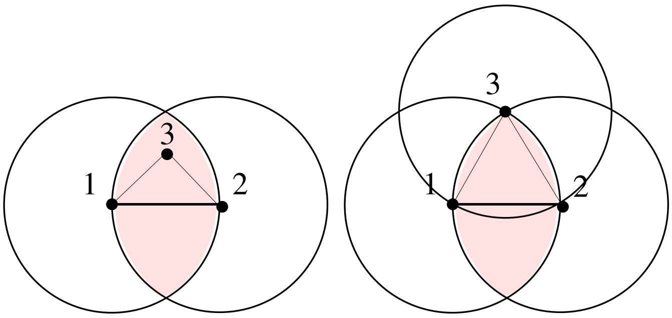

Figure 3.14 – Relation between RNG and EMST

belong to the RNG and thus: $EMST \subseteq RNG$. Summing up all relations we get:

$$EMST \subseteq RNG \subseteq GG \subseteq DT. \tag{3.1}$$

Algorithm 3 GG planarization algorithm

```
set $GG \leftarrow Neighbors$
**for all** $n \in Neighbors$ **do**
    $x \leftarrow 0.5 \times (current + n)$
    **for all** $m \in Neighbors$ **do**
        **if** $m \neq n$ and $d(x, current) > d(x, m)$ **then**
            delete($GG$, $n$)
            break
        **end if**
    **end for**
**end for**
return GG
```

The local GG (or RNG) planarization algorithm (cf., Algorithm 3) computes a plane graph without any connectivity loss in UDGs. In real wireless environments, the conditions for obtaining plane graphs are not satisfied due to asymmetric links and not circular radio coverage, so routing protocols based on face routing over a locally constructed GG fail to provide a sufficient packet delivery rate [34].

3.10 Greedy-Face-Greedy Geographical Algorithm

Bose et al. [24] proposed the first geographical algorithm combining greedy forwarding (cf., Section 3.7) with face routing (cf., Section 3.8) over a locally constructed GG (cf., Section 3.9). GFG maintains two graphs at the same time. The first one is the network connectivity graph and the second one is the GG computed locally in the distributed way by each node in the network. GFG has two routing phases. The first one called greedy forwarding fails in some situations, so the recovery phase is required. During the greedy phase a forwarding node chooses a next hop with the least distance to the destination from the set of its immediate neighbors. It tries to forward each packet in the direction of the destination and does not take into account that it may fail at the border of concave regions in which the forwarding node does not have any neighbor closer to the destination. When the greedy forwarding fails and it cannot return a next hop, GFG uses the face routing from Section 3.8. GFG stores the last position a_l in which the greedy forwarding was used. When a packet arrives at the closer node to the destination than a_l, GFG switches back to the greedy forwarding.

3.11 Right hand rule face routing

Karp et al. [25] proposed a simplified version of face routing called a right hand or resp. left hand rule. It applies the strict right hand rule to surround faces. Like in Section 3.8, source node a_s sending a packet chooses the first anticlockwise link counting from section $\overline{a_s, a_d}$. Intermediate nodes forward the packet to the first anticlockwise link from the predecessor. When the forwarding node realizes that the chosen link intersects with $\overline{a_s, a_d}$, it changes the face to the face closer to the destination and recomputes the next hop to be the first anticlockwise link from this crossed by $\overline{a_s, a_d}$. The difference between the right hand rule face routing and Compass II (cf., Section 3.8) is that the former changes faces before crossing $\overline{a_s, a_d}$ and the latter after encountering the intersection. Additionally, Compass II changes the right hand to left hand rule or vice versa.

Karp et al. [25] proposed Greedy Perimeter Stateless Routing (GPSR) for Wireless Networks. The difference between GFG and GPSR is that the latter uses their simplified face routing.

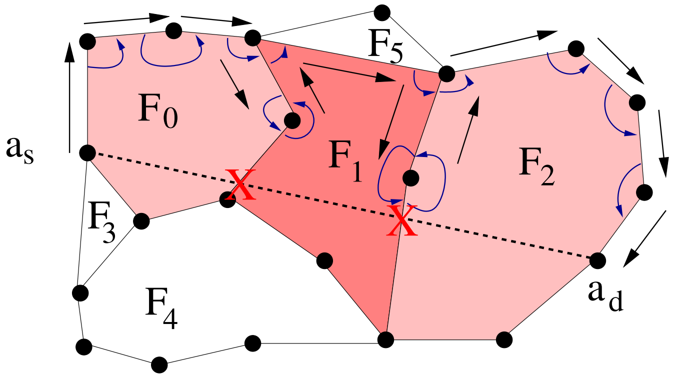

Figure 3.15 – Right hand rule face routing

3.12 Greedy Other Adaptive Face Routing

Kuhn et al. [35] proposed an optimized protocol called Greedy Other Adaptive Face Routing (GOAFR+) pronounced as *gopher plus*. Their protocol has two phases: the greedy geographical forwarding and Other Adaptive Face Routing (OAFR). The protocol maintains three parameters: ρ_0, ρ, and σ with default values: $\rho_0 = 1.4$, $\rho = \sqrt{2}$, and $\sigma = 0.01$. When source node a_s sends a packet to destination a_d, it initializes circle C centered at a_d of radius $r_C = \rho_0 \, |a_s, a_d|$. Each time the packet greedily advances towards the destination and the distance $|a_f, a_d|$ is smaller than r_C/ρ, forwarding node a_f reduces $r_C^{(n+1)} = r_C^{(n)}/\rho$. The OAFR is an optimized face routing, it tries to entirely explore each face starting from a local minimum. When a packet arrives at the local minimum once again, it is forwarded to the closest node to the destination on the face.

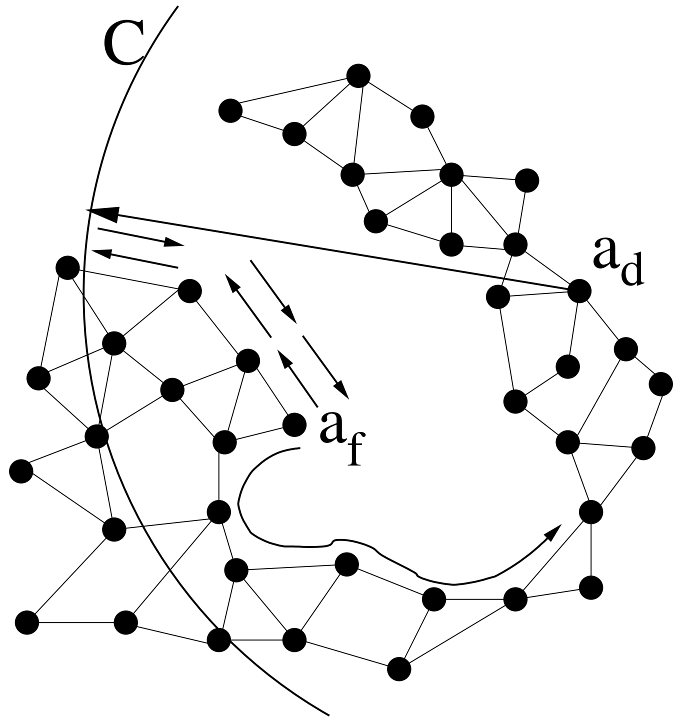

Figure 3.16 – Other Adaptive Face Routing

The OAFR also counts a number of nodes closer to destination p and further from the destination q than the local minimum. If a forwarding node realizes that $p > \sigma q$, it terminates the face routing and forwards the packet to the closest node seen so far. When the packet hits C for the first time, it goes back and explores a face in the opposite direction as shown in Figure 3.16. The second hit means two possibilities. If none of visited nodes lies closer to the destination than the local minimum, a forwarding node enlarges radius of C: $r_C^{(n+1)} = \rho r_C^{(n)}$, reset hit counter, and continues face routing. It terminates face routing and sends a packet to the closest node seen so far in the other case. Bounding circle C prevents from exploring faces in the wrong direction and improves the OAFR performance.

3.13 Path Vector Face Routing

Leaon et al. [36] specified the Path Vector Exchange (PVEX) protocol. PVEX disseminates information around each face by using the right or left hand rule. Each node around a face gets the cyclic list of all nodes located on the face boundary. During the packet forwarding, each node advances a packet to the closest node to the destination located on the currently surrounded face minimizing a hop distance by choosing the appropriate routing direction: clockwise or anticlockwise. This method improves the face routing performance.

3.14 Cross Link Detection Protocol

The main disadvantage of face routing is that it requires a plane graph. Wireless network connectivity graphs are usually not planar, so face routing fails in some situations as shown in Figure 3.17. The GG planarization algorithm computes a plane graph, however, the algorithm requirements are not satisfied in real world experiments as reported by Kim et al. [37], because the coverage varies. Normally, the algorithm inspects the link between two neighbors checking whether the circle (the link is its diameter) contains other neighbors in its interior. It may happen that the circle is not empty from one neighbor point of view, but another node does not see any node in this interior. In this situation, the link cannot be simply deleted, because the resulting GG may loose its connectivity. The method of keeping such links is also not good, because it may not eliminate all loops in the network graph, so the efficiency of face routing over the locally computed GG dramatically drops in real topologies.

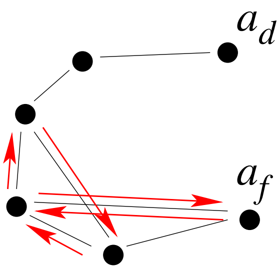

Figure 3.17 – Recovery procedure fails due to crossing edges

Kim et al. [37] proposed a Cross Link Detection Protocol (CLDP). A node running the CLDP tests each of its links whether it intersects with other links in the network. It sends a probing packet forwarded by the right hand rule face routing. Each node inspects whether its forwarding link crosses the tested link of the origin. When the packet arrives at the origin once again, it has the list of crossing edges. Links are carefully removed to prevent from disconnecting the original graph. This is a very expensive technique, because it implies the important traffic overhead. One year later, Kim et al. [34] published another paper about removing crossing edges. In this case, the CLDP runs on demand when the routing really fails because of the face routing loop. This technique improves the protocol performance.

3.15 Summary

Routing in small static networks is easy. However, most real wireless networks are dynamic (links may go up and down, nodes join or leave the network) and their size may grow, which implies the problem of large routing tables $O(N)$. New kinds of spontaneous networks amplify this trend: wireless mesh [38, 39] or sensor networks may include thousands of nodes acting as routers [40]. In large scale wireless networks with high churn, link-state protocols such as OLSR are not suitable, because each modification of topology requires propagation of the routing information to recompute the best routes. Distance vector protocols imply periodic distribution of routing information or in case of on-demand versions for ad-hoc networks such as AODV they may suffer from frequent route failures and search phases that rely on flooding. Several experimentations reported on scaling problems of familiar wireless routing protocols such as AODV, DSDV, or DSR [41, 42].

When considering these drawbacks, geographical routing appears as an interesting alternative approach to traditional multi-hop routing that presents many advantages: forwarding is localized–it only depends on the state of a closed neighborhood and not on the global topology, it is also scalable–the size of routing information is limited.

The pure greedy geographical routing presents several drawbacks that we want to avoid. For instance, when routing fails at topological defects such as voids or concave regions, a backtrack method such as face routing should be used. Topological defects can be detected at concave nodes that only have neighbors in the backward direction from the void area. Face routing can circumvent topological defects, but it requires the construction of a plane graph of links between neighboring nodes. Several papers reported on the difficulty of constructing plane graphs in real wireless environments, a non-planar graph leads to the significant probability of packet loss.

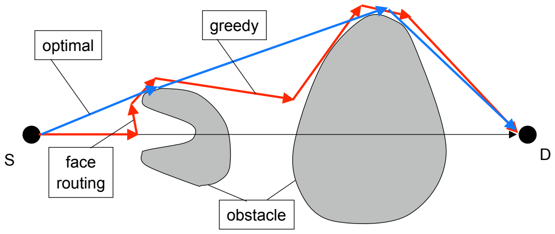

Figure 3.18 – Greedy routing vs. optimal trajectory

Figure 3.18 illustrates another drawback of geographical routing compared to the optimal trajectory for a given topology with obstacles. The greedy routing coupled with a face routing algorithm for circumventing obstacles compute much longer paths than the optimal route. In the next chapter, we study mathematical properties of Network-Wide Localization.

4

Properties of Localization Techniques

In this chapter, we study Network-Wide Localization. We assume that nodes in the network have completely random physical positions. We are interested in localizing all nodes in the network when starting with a small location reference group containing the nodes with known positions called anchors. Other nodes do not know their coordinates. If an unlocalized node can estimate its position through a radio location method, it becomes an anchor node. We study a percolation approach to the localization. We take the basic property of localization techniques that an unlocalized node can compute its position when it neighbors with at least three anchors. We study an influence of the network topology on the process of Network-Wide Localization. We count the fraction of nodes in the network that can be localized, but we do not care about localization accuracy. Localization techniques are important, because they are suitable for network addressing required by geographical forwarding.

4.1 Localization

Now, we provide mathematical definitions of localization techniques suitable for radio localization.

4.1.1 Trilateration

Trilateration is a method of estimating a node position using Euclidean distances to three points with known positions: AN_1, AN_2, and AN_3. Imagine the situation in Figure 4.1 in which an Unlocalized Node (UN) estimates the distance d_1 to AN_1. We know that a set of points with a fixed distance from single point c forms a circumference centered at c. The UN realizes that it is located somewhere on the circumference of the circle centered at AN_1 of radius d_1: $UN \in O(AN_1, d_1)$. The UN estimates the distance d_2 to AN_2 and analogically it gets that $UN \in O(AN_2, d_2)$. Summing up both relations, we get that the UN lies at the intersection of two circles $UN \in O(AN_1, d_1) \cap O(AN_2, d_2)$. Two circles may have 2 intersection points, so the third circle $O(AN_3, d_3)$ is necessary to choose one of these points and estimate the position of the UN. Finally, we get that the UN lies at the intersection of 3 circles: $UN \in O(AN_1, d_1) \cap O(AN_2, d_2) \cap O(AN_3, d_3)$.

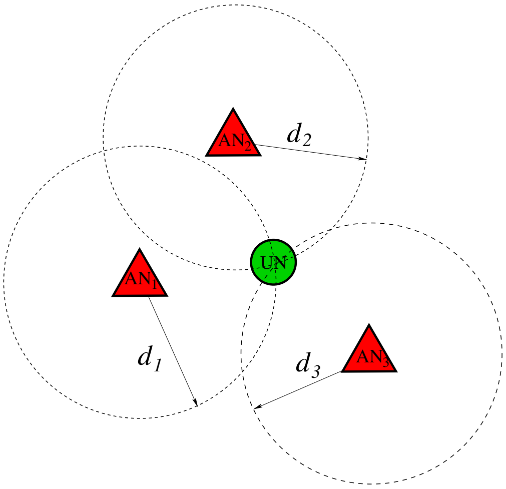

Figure 4.1 – Trilateration

4.1.2 Multilateration

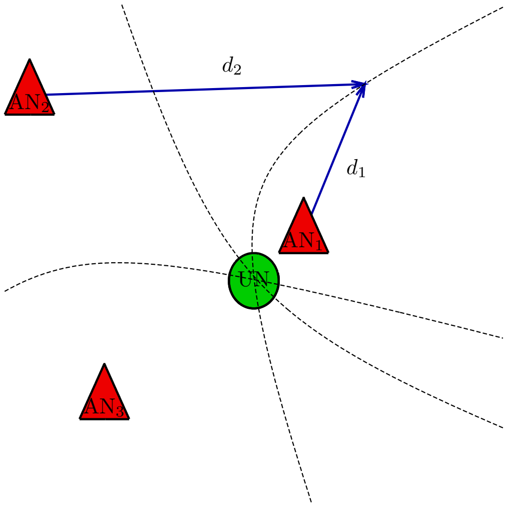

Figure 4.2 – Multilateration

Multilateration or hyperbolic positioning (cf., Figure 4.2) is another method of localization. An UN measures distance difference $\delta_{12} = d_1 - d_2$ to two points: AN_1 and AN_2. A branch of hyperbola $H(AN_1, AN_2, \delta_{12})$ is a set of points such that distance difference δ_{12} to two points called foci (AN_1, AN_2) is constant. The UN realizes then, that it is located at branch of hyperbola $H(AN_1, AN_2, \delta_{12})$. Analogically, we get that $UN \in H(AN_2, AN_3, \delta_{23}) \cap H(AN_1, AN_3, \delta_{13})$. Instead of solving complicated equations, the UN may also use Maximum Likelihood (ML) methods of position estimating. The estimated position should keep distance differences $\{\delta_i\}$ as close as possible to the real values. Now, we describe the position estimation techniques by means of Angle of Arrival (AoA) measurements that can be done either at anchors (ANs) or at UNs.

4.1.3 Triangulation at ANs

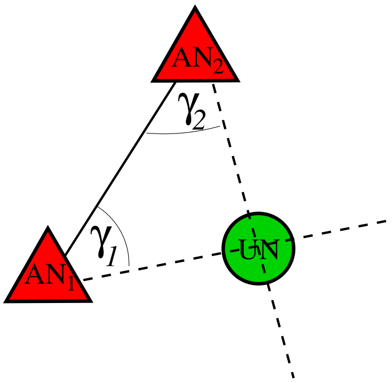

Figure 4.3 – Triangulation, measurements at AN

When AN_1, AN_2, and UN are non-collinear, AN_1 and AN_2 are directly connected by section $\overline{AN_1, AN_2}$ as shown in Figure 4.3, and angles $\gamma_1 : \angle AN_2, AN_1, UN$ and $\gamma_2 : \angle AN_1, AN_2, UN$ are known, an UN can estimate its position as the intersection point of 2 straight lines originating at ANs that go in the destination of the UN. Two, non-collinear straight lines have only one intersection point, so these two ANs are required to estimate the position of UN, but the direct connectivity between each pair of ANs is required.

4.1.4 Triangulation at UN

This triangulation method estimates a node position using 3 angles: $\gamma_1 : \angle AN_1, UN, AN_2$, $\gamma_2 : \angle AN_2, UN, AN_3$, and $\gamma_3 : \angle AN_3, UN, AN_1$ among an UN and 3 anchors: AN_1, AN_2, and AN_3. Imagine the situation in Figure 4.4 in which the UN estimates γ_1. We know that geometrical locus of points observing two anchors at a fixed angle forms an arc AN_1, UN, AN_2. The arc is a part of a circle circumscribing triangle $\triangle AN_1, UN, AN_2$. The UN knows now, that it is located somewhere on the arc. Taking into account other nodes, the UN estimates its position $UN \in$ arc$(AN_1, UN, AN_2) \cap$ arc$(AN_2, UN, AN_3) \cap$ arc(AN_3, UN, AN_1).

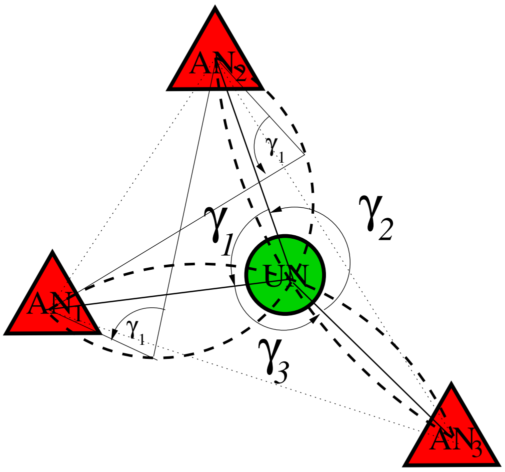

Figure 4.4 – Triangulation, measurements at UN

Many authors studied different radio location techniques required for localization. Time of Arrival (ToA) [43] uses Time of Flight (ToF) between ANs and an UN to estimate the distance among devices according to the formula $d = cT$, where d is the estimated distance, c is a speed of light and T is ToF. Position is computed through trilateration. In Time Difference of Arrival (TDoA) [44, 45, 46] a distance difference is estimated as: $\delta = c\Delta T$, where δ is the distance difference, c is a speed of light, and ΔT is TDoA. Position is computed through multilateration. One way measurements of the time of signal propagation requires synchronized clocks, which is difficult to obtain in WMNs. Two Way Ranging (TWR) technique [11, 47, 48, 49, 50] measures the Round-Trip-Time using communication sequence in two directions and it does not require clock synchronization to estimate the distance. Recent increased precision of this approach owes to the progress in low-power implementations of TWR using UWB (Ultra Wide-Band) or chirp transmissions [50].

Angle of Arrival (AoA) [51] uses highly directional antennas for the signal direction estimation. Position is computed through triangulation at an UN. Received Signal Strength Indicator (RSSI) [52] and RSS profiling [53] are poor ranging techniques. They estimate the distance using the RSSI-distance mapping. Position is computed through trilateration. Each technique performs a node positioning and it might be used for the network-wide localization. To estimate node positions all mechanisms require the UN to neighbor with 3 anchor nodes.

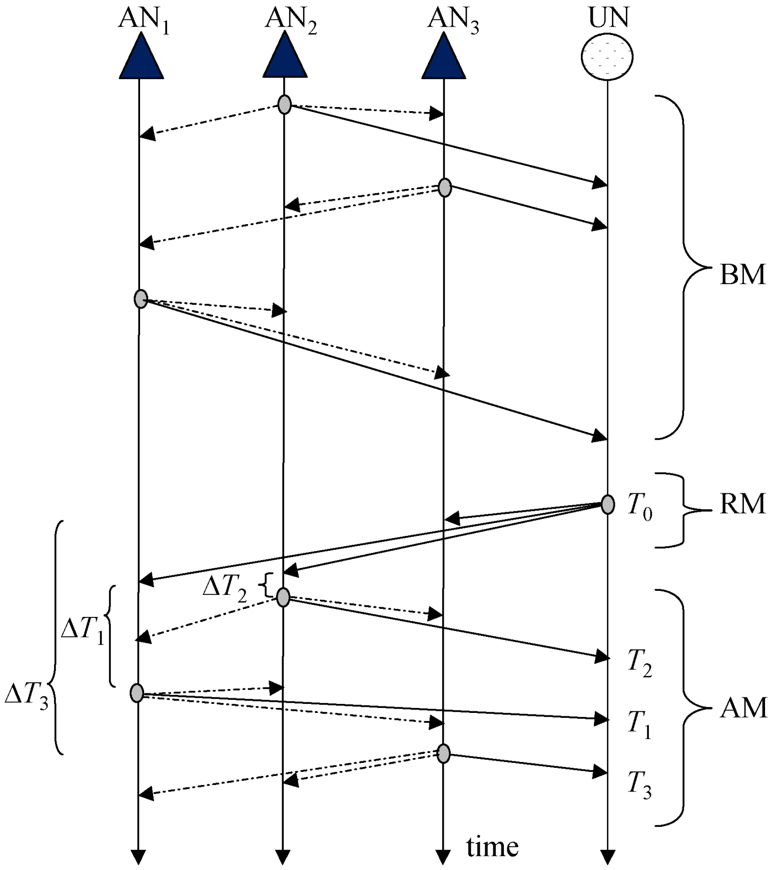

Figure 4.5 – The principle of the Beacon protocol

4.1.5 Beacon Protocol

In the Beacon protocol proposed in the literature [45], anchor nodes initiate the localization process by periodically sending broadcast Beacon Messages (BM) each time interval T_B. When an unlocalized node receives three beacons from at least three different anchor nodes, it waits for an uniformly distributed period with max T_R and sends a broadcast Range Message (RM) to perform ranging with the anchor nodes in parallel. An additional delay is introduced to reduce collisions. The anchor nodes reply to the RM message with unicast ACK Messages (AM) after an uniformly distributed delay with max T_A to avoid simultaneous arrivals at the unlocalized node. Timestamping the range and the ACK messages enable the estimation of the distance to three anchor nodes yielding thus the position. The unlocalized node then becomes an anchor node and starts sending periodic beacons to propagate the localization process to other unlocalized nodes. Figure 4.5 from [54] illustrates the principle of the Beacon protocol.

4.1.6 Optimized Beacon Protocol

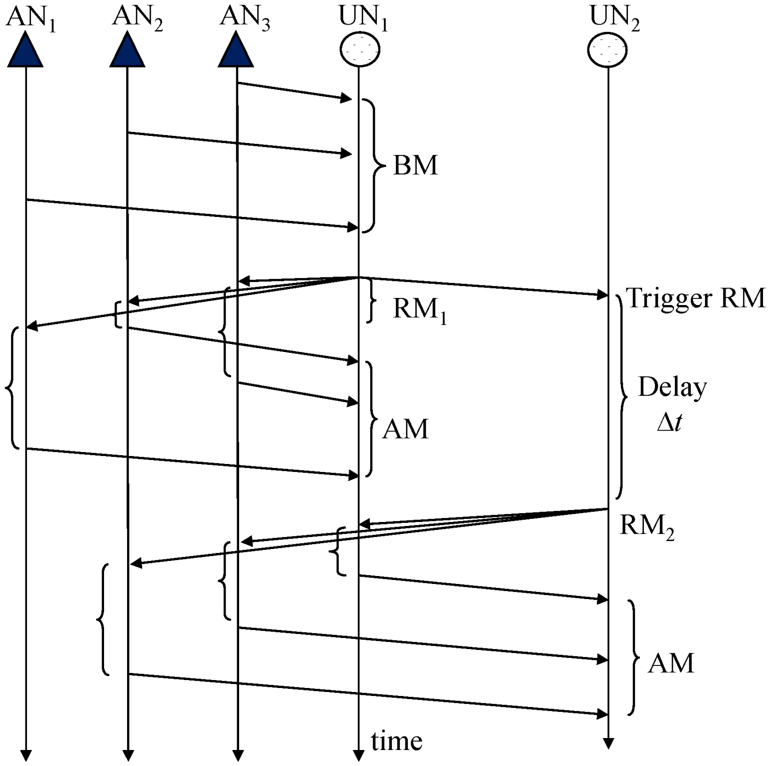

Figure 4.6 – The principle of the Optimized Beacon protocol

An optimized version of the beacon protocol reduces the number of messages [54]. In the Optimized Beacon protocol, anchor nodes initiate the localization process by periodically sending a limited number of broadcast beacon messages so that any unlocalized node in their neighborhood can start ranging. However, the difference is that once an unlocalized node gets localized, it does not send beacons. Unlocalized nodes sense ranging packets in order to start their ranging operations. If an unlocalized node overhears three broadcast range messages sent by other nodes, it can start its ranging operation without waiting for beacons (cf., Figure 4.6 from [54]). It waits, however, for interval T_B to be sure that when its neighboring nodes have started the ranging operation, they finished it, so they are localized and ready to respond with AM messages. Delay T_A is the same as in the previous scheme.

4.2 Random Geometric Graphs

We consider Random Geometric Graphs [55] with open boundary conditions (as opposed to continuous or toroidal conditions) that model a wide range of Multi-Hop Wireless Networks: ad hoc, mesh, or sensor networks [56]. Loosely speaking, a random geometric graph results from random placement of wireless nodes in an area with edges connecting neighbor nodes, if they are within a communication range. If the communication region of a node is a disk with radius 1, the corresponding model is called an Unit Disk Graph [57]. Although it is a simplistic model that does not capture all of the details and complexity of radio communications, it is useful for analyzing the most important characteristics of a wireless network such as capacity, connectivity, and coverage [58, 56, 59, 60].

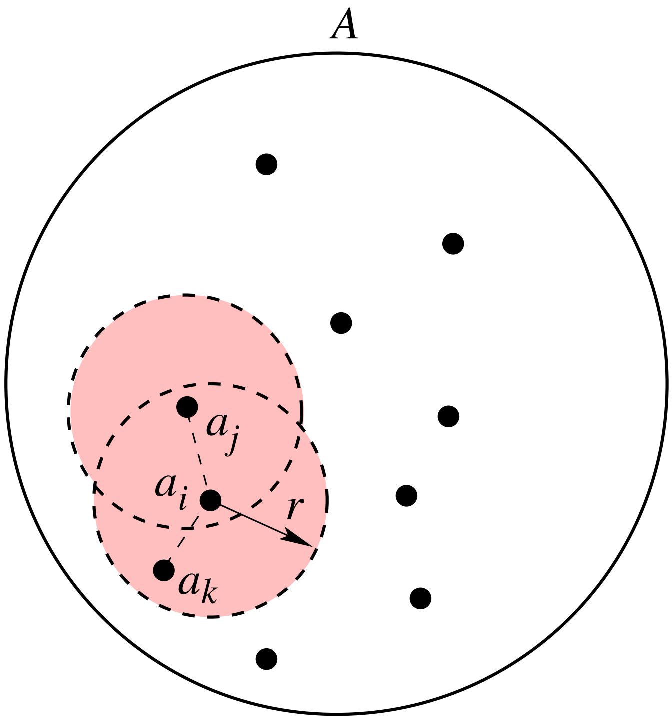

Figure 4.7 – Random geometric graph

We randomly place N nodes inside disk $\mathcal{A}$ of center $O = (0,0)$, radius 1, and area π: coordinates a_i of node n_i are drawn from uniform distribution $f(a_i)$:

$$f(a_i) = \begin{cases} \pi^{-1} & a_i \in \mathcal{A} \\ 0 & a_i \notin \mathcal{A} \end{cases} \tag{4.1}$$

We assume that nodes have the same circular transmission range of radius r. Nodes a_i and a_j are immediate neighbors, which means that they are connected by a link, iff a_j

is within the transmission range of a_i and vice-versa. Figure 4.7 illustrates disk $\mathcal{A}$ and the notion of the neighborhood with nodes at a_j and a_k directly connected to a_i.

The number of immediate neighbors of node n_i is called a node degree $d_i = \text{deg}[n_i]$ ($d_i = 2$ in Figure 4.7). We define the average node degree as:

$$D = \frac{1}{N}\sum_{i=1}^{N} \text{deg}[n_i]. \tag{4.2}$$

Let us choose a node at a_i and compute the probability that another node at a_j is its neighbor (the probability that it lies inside the circle of origin a_i and radius r), which is a Bernoulli trial with p of success (resp. q of failure) given by:

$$p = \frac{\pi r^2}{\pi} = r^2,\ q = 1 - p. \tag{4.3}$$

$Pr(d_i)$, the probability that the node at a_i has d_i neighbors is the sum of N independent Bernoulli trials given by the binomial distribution:

$$Pr(d_i) = \binom{N-1}{d_i} p^{d_i} q^{N-1-d_i}, \tag{4.4}$$

with mean degree

$$D = (N-1)p \approx Np = Nr^2. \tag{4.5}$$

In this chapter, we consider the case in which the graph has a given average node degree $D = Np < \infty$ and large N values: $N > 1000$. This assumption allows us to approximate the binomial distribution of the node degree with the Poisson distribution:

$$Pr(d_i) = \frac{D^{d_i}}{d_i!}\exp(-D) \tag{4.6}$$

with expected value D.

4.2.1 Connectivity

Connectivity–the property that any two nodes on the network can communicate, is essential to the network design. By changing deployment density of nodes or adjusting the transmission range of nodes through power control, we can obtain different topologies (connected or not) with different characteristics of performance and interference. For this reason, much of the existing literature on wireless networks studied connectivity and addressed the issue of finding the minimum transmission range or the critical node degree (the number of neighbors that depends on node density) for obtaining a connected network [61, 55, 56, 62, 63, 64]. The work on these aspects showed that connectivity is a percolation phenomenon–when the transmission range or the average node degree goes over a critical value, the probability of the resulting connected graph sharply increases [56]. The main objective of this kind of analysis was to find the critical transmission range or critical node degree. In some cases, theoretical tools are insufficient to derive analytical expressions, for instance Dall et al. [55] found the critical connectivity in d-dimensional random geometric graphs by means of Monte Carlo simulations [55].

4.2.2 k-connectivity

Connectivity extends to the notion of k-connectivity: a graph is k-connected if there is no set of $k-1$ nodes whose removal partitions the network into two components. The notion of a k-connected network is important for fault tolerance, because such network can continue to operate in spite of the failure of $k-1$ nodes [61, 59]. It is also important for performance: as there are k disjoint paths between any pair of nodes, routing protocols can take advantage of multiple disjoint routes [65].

Much research work has addressed the problem of connectivity in wireless multi-hop networks. Several authors proposed schemes for power control and analyzed the critical power for connectivity in mobile wireless networks [60, 66, 67]. Another important issue is the minimum transmission range or the critical node degree for obtaining a connected network [60, 56, 62, 64]. Bettstetter analyzed the minimum node degree and k-connectivity to derive expressions for determining the required transmission range that results in k-connected network [56]. Xue and Kumar showed that each node should be connected to $\Theta(\ln n)$ nearest neighbors to achieve network connectivity [68]. They also provided more results on the impact of parameters such as range, number of neighbors, power consumption, and connectivity on spatial reuse and capacity [69]. Li et al. analyzed the notion of a k-connected network in the context of fault tolerance and proposed a localized method for controlling topology that achieves k-connectivity [59]. Penrose showed that as the node transmission range increases in a random geometric graph, it becomes almost surely k-connected when its minimum node degree $> k$ [61]. Dall and Christensen studied the behavior of the critical giant cluster in many dimensions. The goal of their work was to show similarities between random graphs and random geometric graphs in infinite number of dimensions [55]. As theoretical tools are insufficient to derive analytical expressions, they used Monte Carlo simulations to find the critical connectivity in d-dimensional random geometric graphs–their estimates for the critical node degree 4.52 (2D) and 2.74 (3D) are in accordance with known values. Moreover, they report on the critical values in higher dimensions. We use a similar approach based on efficient Monte-Carlo simulation of large-scale graphs, but we focus on the localization like process in 2D.

Most of the work on connectivity in wireless multi-hop network consider unrealistic assumptions of the unit disc graph. Miorandi and Altman analyzed the extension of connectivity and coverage results by adopting more realistic transmission propagation models that take into account shadowing [70].

Propagation processes arise in the context of network-wide broadcast communications or flooding. Several authors addressed the issue of probabilistic broadcast protocols that reduce the number of packets in the network and improve performance as well as energy consumption [71, 72, 73]. The technique of gossiping is similar and leads to the same type of percolation behavior. All these classes of protocols operate according to the connectivity and their analysis as a percolation phenomenon exist [72, 74].

Social networks and epidemic propagation have a similar behavior to k-propagation (cf., Section 4.3), however, researchers that work in these domains adopt different assumptions such as random graphs and look at different critical parameters, for instance they analyze a critical threshold T in the complex propagation process involving multiple active neighbors mentioned previously [75]. Such a model corresponds well to stock market herds, lynch mobs, riots, or trust establishment, but it does not represent the behavior of the protocols that we have analyzed in this chapter.

4.2.3 Percolation

To illustrate percolation, we take connectivity in random geometric graphs as an example, because it is a well analyzed case [55]. We consider an infinite number of nodes on an infinite plane setting the transmission range so that average node degree D is fixed: $D < \infty$. We divide the graph into a set of non connected clusters and consider two metrics: $R(D)$, the radius of the largest cluster existing in the network (average Euclidean distance between the origin and the nodes that form the cluster) and $P(D)$, percolation probability that a randomly chosen node belongs to the infinite cluster.

Figures 4.8 and 4.9 present two metrics. The first curve corresponding to $R(D)$ grows like:

$$R(D) \propto (D_c - D)^{-\nu}, \tag{4.7}$$

where D_c is called a critical node degree and ν is a critical exponent. We can observe that when D is much lower than D_c, the radius of the maximal cluster is quite small, so there are a lot of small disconnected clusters in the network with low D. When D grows, the radius of the maximal cluster also grows and begin to grow very fast in the vicinity of the critical point. Note that the curve has a vertical asymptote at $D = D_c$, because the maximal cluster has infinite radius when D reaches the critical node degree. This is so called phase transition at $D = D_c$, because at this point, one giant cluster emerges that covers the infinite number of nodes.

Percolation probability $P(D)$ is 0 for $D < D_c$ and has a non zero positive value above, because the infinite cluster does not exist below D_c. $P(D)$ also grows very rapidly in the vicinity of critical node degree D_c:

$$P(D) = \left\{ \begin{array}{ll} 0 & D \leq D_c \\ (D - D_c)^{\beta} & D > D_c \end{array} \right. , \tag{4.8}$$

where β is a critical exponent describing the critical behavior of the curve $P(D)$. The critical node degree is known and equal to $D_c = 4.51$ for connectivity in random geometric graphs on the 2D plane [55].

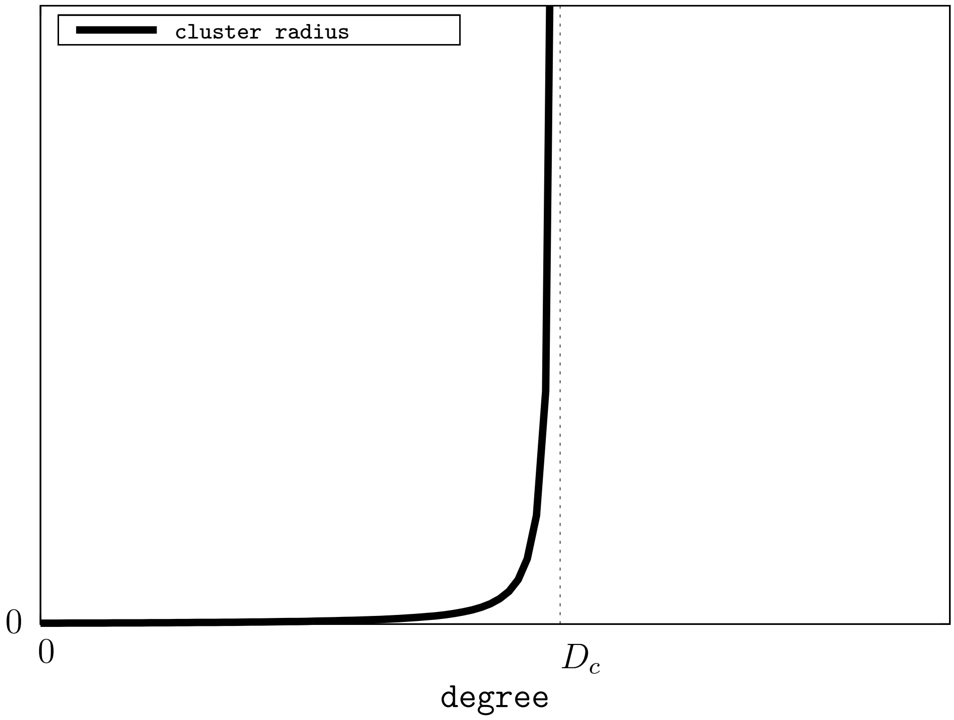

Figure 4.8 – Critical behavior of cluster growth

4.3 k-propagation

The design of protocols in wireless networks sometimes requires analyzing a different notion–propagation of information. Consider for instance flooding: a node wants to send a packet to all nodes in the network. We can consider this protocol as the propagation of a black token in the network initially composed of white nodes. When a node receives the token from a neighbor, the node becomes black and propagates the token to all other neighbors. Similarly to the notion of the k-connected graph, we can define a k-propagated graph: a node becomes black if it receives the token from its k neighbors. Note that flooding corresponds to 1-propagation. Figure 4.10 illustrates the principle of 2-propagation.

Optimizing performance of flooding led to probabilistic protocols in which a node propagates a packet with a certain probability to reduce the number of reforwarded packets in the network [71, 72, 73]. Protocols based on gossiping operate in a similar way [76]. Several authors analyzed 1-propagation related to this class of protocols and showed its percolation behavior [72, 74].

There are other protocols that operate according to k-propagation, for instance network-wide localization protocols based on trilateration, multilateration and triangulation require a 3-propagated graph [77, 44]. Assume that two nodes can estimate the distance between them through the TWR technique in which nodes exchange short

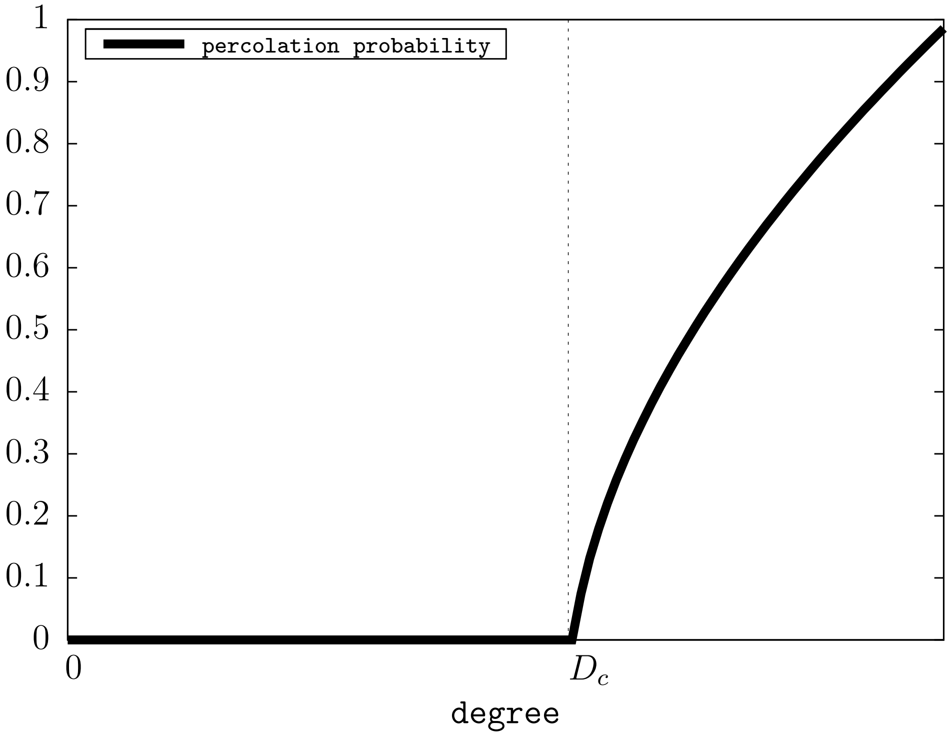

Figure 4.9 – Critical behavior of percolation probability

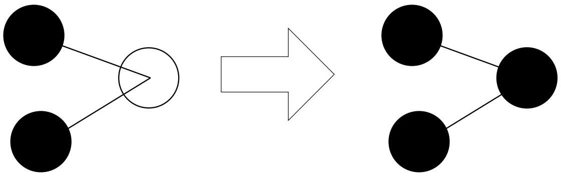

Figure 4.10 – 2-propagation: a node becomes black if its two neighbors are black too.

messages and record the instants of their emission and reception. The protocol corresponds to the propagation of a black token from $k = 3$ neighbors until all nodes become black (anchors). An important design and deployment parameter is the critical node degree for which all nodes become localized. Moreover, we are also interested in speed of convergence. Note that to improve accuracy of localization, some localization protocols may use greater values of $k > 3$ [78].

A similar, but different model corresponds to social networks with Complex Propagation in which node activation requires simultaneous exposure to multiple active neighbors in random graphs [75]. In this model, each node is in two states (active or inactive) and a node becomes active if the proportion of its active neighbors exceeds a given threshold T. Such a model corresponds well to stock market herds, lynch mobs, riots, or trust establishment, but it does not represent the behavior of localization protocols.

The goal of this chapter is to analyze k-propagation and to find the conditions in which information k-propagates to all nodes in the network. The k-propagation process is much more complex than connectivity in wireless networks or analytically calculated models of the complex propagation in social networks, so to our best knowledge, there are no analytical results. For $k = 1$, the existing results come from simulations [72]. We propose to study k-propagation for larger k to show that it also behaves as a percolation phenomenon and find its critical parameters. Our approach is based on Monte-Carlo simulations as in a previous work on geometric random graphs, however, to obtain precise estimates of critical parameters, we push the numerical method to a large-scale by analyzing networks with a million of nodes, which is one original aspect of our work. Although many authors have already studied 1-propagation in the context of flooding and gossiping protocols, as well as epidemic or malware propagation, the analysis of k-propagation is also original.

To obtain our results, we have developed a simulator that efficiently handles large-scale multi-hop wireless networks, which is impossible with traditional network simulators such as ns-2 [79]. Finally, we analyze a protocol for network-wide localization as an example of a k-propagation process. We show that it behaves exactly as predicted by our percolation results.

We consider a random geometric graph of N nodes with open boundary conditions and average node degree D embedded in disk $\mathcal{A}$. We assign a Boolean color (black or white) to nodes–at the beginning all nodes are white. We create a second disk at the origin of $\mathcal{A}$ with radius $r/2$ and place k connected black nodes at random coordinates (every black node has $k - 1$ black neighbors at the beginning). Next, we start k-propagation: each white node with at least k black neighbor nodes changes its color to black. At the beginning all white nodes inside the small disk change to black, because they are connected to all initial k black nodes and the process continues until there is no node that can change the color. Note that a node that changes the color influences others, so the process propagates in a wave from the origin in the direction of the network boundary.

4.4 Monte-Carlo Simulations of k-propagation

In this section, we describe our simulator and a method for deriving the critical node degree for k-propagation from simulation results. We are interested in the same metrics as in Figure 4.8, but for k-propagation and for different values of k.

4.4.1 Monte-Carlo Simulator

We have developed a simulator able to handle very large wireless multi-hop networks up to a million of nodes. We take square embedding $\mathcal{S}$ of disk $\mathcal{A}$: it is divided into a set of $x \times y$ small squares of equal size $\mathcal{S}_{i/x,j/y}$ as shown in Figure 4.11. We choose

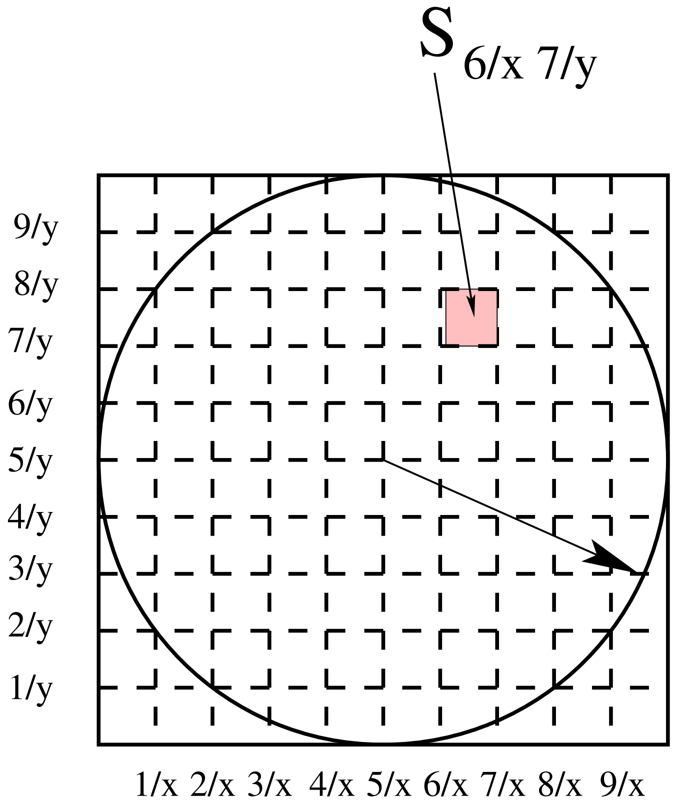

Figure 4.11 – Square embedding of disk $\mathcal{A}$

x and y in function of the number of nodes N in the following way:

$$x = \lceil \sqrt{N}/x_0 \rceil \quad (4.9)$$

$$y = \lceil \sqrt{N}/y_0 \rceil \quad (4.10)$$

Assuming that the node density inside $\mathcal{S}$ is constant and knowing that the area of a square element is $x \times y$ smaller than the area of $\mathcal{S}$, we obtain a constant number of nodes ($x_0 \times y_0$) per single square $\mathcal{S}_{i/x,j/y}$.

We randomly place N nodes inside $\mathcal{A}$: each node obtains its coordinates from the uniform random variable with the probability density function given by Eq. 4.1 and it is assigned to one of the small squares: $a_i \in S_{i/x,j/y}$. The squares do not overlap and they cover whole space $\mathcal{A}$ so that one node lies exactly inside one small square. We create the network connectivity graph for each node in the network by finding the set of all squares that lie within the node transmission range r and iterate over all nodes inside the squares to check if the nodes are within the transmission range. In this way, the computational complexity of the algorithm is smaller than $O(N^2)$, complexity achieved by a straightforward check of all edges between N points. The simulator thus allows us to simulate very large wireless multi-hop networks.

4.4.2 k-propagation analysis: average radius of the giant cluster

To obtain $R^k(D), k = 1, 2...$[1], the average radius of the cluster composed of black nodes in function of the average node degree, we run simulations for different values of D while keeping N, the total number of nodes, constant and large. For one value of D, we perform a large number of simulation runs. In a single simulation run, we wait until the k-propagation process terminates and we compute the average distance between origin O and all black nodes. Next, we fit the curve given by Eq. 4.7 and compute ν and D_c. As we repeat simulations for different values of k, we add index k to the parameters: ν^k and D_c^k.

4.4.3 k-propagation analysis: percolation probability

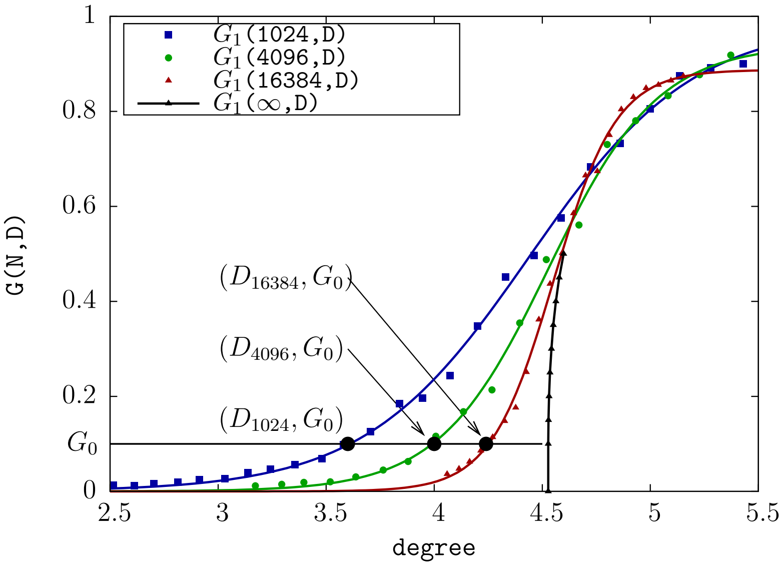

Figure 4.12 – 1-propagation: approximation of percolation probability by the finite size network

We know that percolation probability $P(D)$ for an infinite network is a non-analytic function at $D = D_c$. Percolation probability corresponds to the ratio of black nodes to the total number of nodes in an infinite network. We cannot simulate such a network, so we need to simulate k-propagation for an increasing number of nodes and extrapolate for infinity. We measure the ratio for different numbers of nodes N and node degrees D denoted by $G(N, D)$, which tends to percolation probability as N increases. For a fixed value of N and varying D, we fit a sigmoid curve to the simulation results:

$$\text{Sig}(D) = \frac{\gamma}{1 + \exp\left(-\zeta(x - \xi)\right)} \tag{4.11}$$

[1] k is an index and not an exponent.

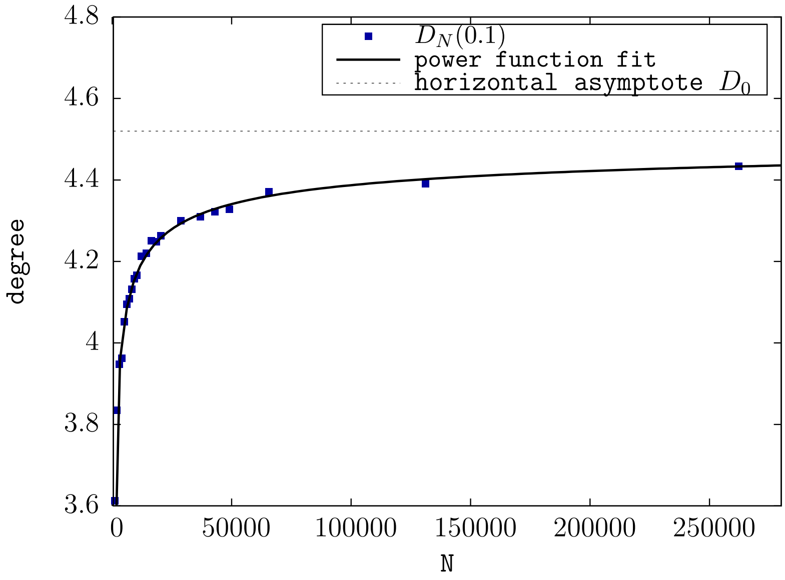

Figure 4.13 – 1-propagation: extrapolation of the infinite size network

as shown in Figure 4.12, where γ governs the height of the sigmoid function, ζ corresponds to sharpness (how fast it goes up), and ξ shifts it to the left or right. Fitting the sigmoid function for different N values results in function Sig$_N(D)$. We can see that the curves are sharper and they grow faster with the increasing value of N. We have expected this behavior, because greater N means a better model of the infinite network, so that $G(N, D)$ closer approximates the non-analytic behavior of the infinite network $G(\infty, D)$.

Figure 4.12 also presents a horizontal line corresponding to G_0 taken at $G_0 = 0.1$ that crosses three sigmoid curves at different values of D. This is the key for finding the value of D for the infinite system for which $G(\infty, D) = G_0$. We compute the solution of the system of equations for different values of N from our simulations:

$$\text{Sig}_N(D) = G_0 \in (0, 0.5] \rightarrow D_N(G_0) = \text{Sig}_N^{-1}(G_0). \tag{4.12}$$

In order to continue the example with $G_0 = 0.1$ we plot pairs $(D_N(0.1), N)$ in Figure 4.13, plot behaves like a power function: $D_N \propto N^{-\phi}$, $\phi > 0$ so we can easily find the horizontal asymptote D_0 of this function, because term $N^{-\phi}$ is irrelevant when $N \rightarrow \infty$. The asymptotic value is very important, because this is the behavior of the infinite network for a given value D_0: $G(\infty, D_0) = G_0$. Repeating simulations runs for different values of $G_0 \in (0, 0.5]$ finally leads to the reconstruction of $G(\infty, D)$. Figure 4.14 presents its critical behavior by making a zoom at the values of D in $[4.53, 4.6]$. We can observe a very rapid growth near the critical value of the node degree within this small interval.

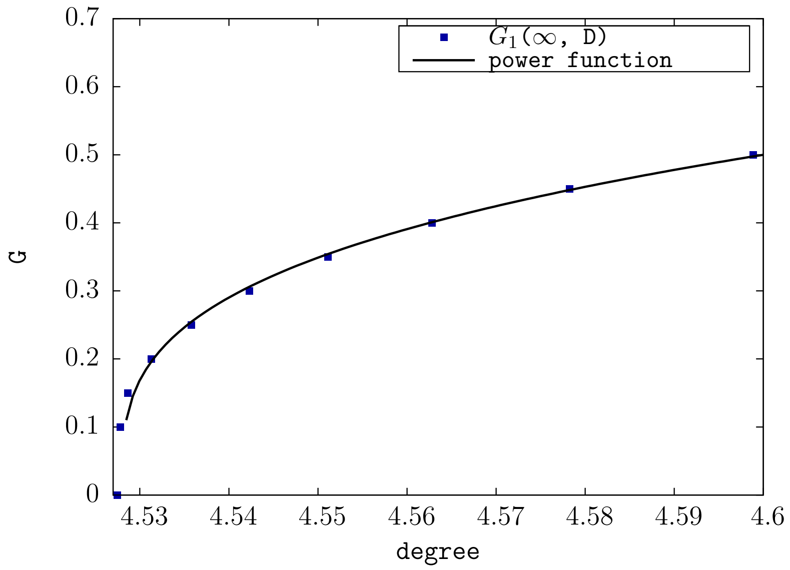

Figure 4.14 – 1-propagation: percolation probability

4.5 Percolation Results

We present the results for a large network of $N = 262144$ nodes. To obtain good precision, we run 200 simulations per transmission range r that results in a given average node degree $D(r)$.

Figure 4.15 presents the average node degree D in function of the average radius of the giant cluster R. We fit the curve with the percolation equation (the inverse of Eq. 4.7):

$$D = \text{const} \times R^{-\frac{1}{\nu_k}} + D_c \tag{4.13}$$

We obtain the following critical parameters for $k = 1$:

$$\nu_1 = 1.43 \pm 0.08\,(5.9\%) \tag{4.14}$$
$$D_c^1 = 4.58 \pm 0.04\,(0.9\%) \tag{4.15}$$

Note that the critical node degree obtained from our simulations is close to the critical value for 1-connectivity in 2D equal to 4.51 reported in the literature [55].

We have run simulations for $k = 2$ and $k = 3$ and obtained the following values:

$$\nu_2 = 1.33 \pm 0.11\,(8.1\%) \tag{4.16}$$
$$D_c^2 = 7.79 \pm 0.05\,(0.6\%) \tag{4.17}$$

$$\nu_3 = 1.53 \pm 0.13\,(8.4\%) \tag{4.18}$$
$$D_c^3 = 10.84 \pm 0.08\,(0.8\%) \tag{4.19}$$

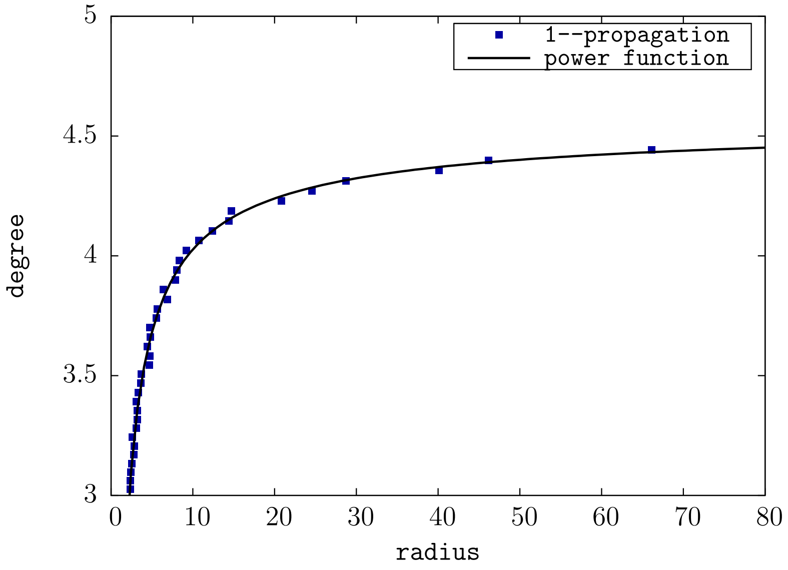

Figure 4.15 – 1-propagation: average node degree in function of the average radius of the giant cluster

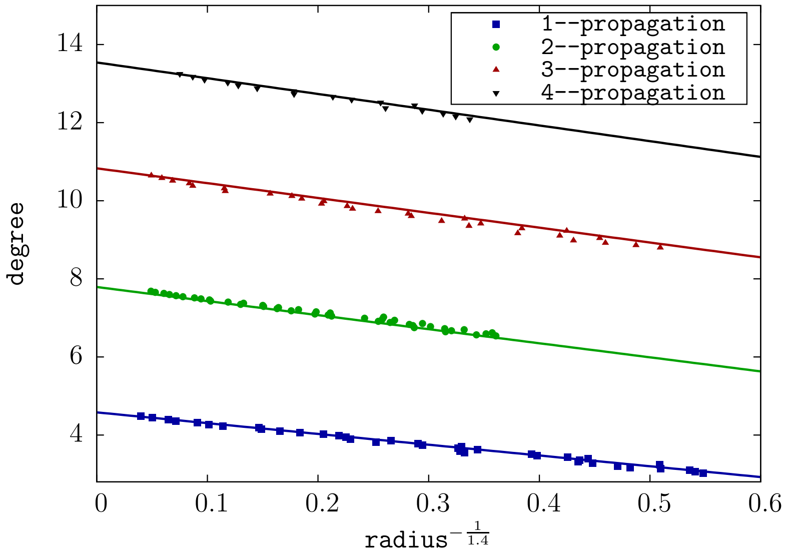

Figure 4.16 – k-propagation: average node degree in function of $R^{-\frac{1}{\nu_k}}$

When presenting the results in function of $R^{-\frac{1}{\nu_k}}$ for $k = 1, ..., 4$, we can observe a perfect fit (cf., Figure 4.16). It enables extrapolation for $X \to 0$ so that $R \to \infty$ while $D \to D_c^k < \infty$ (note that in this presentation of data, larger values of R are closer to 0 on the x-axis). In this way, we obtain the black node cluster of the infinite size for the critical value of the average node degree. These results also confirm that k-propagation exhibits percolation behavior.

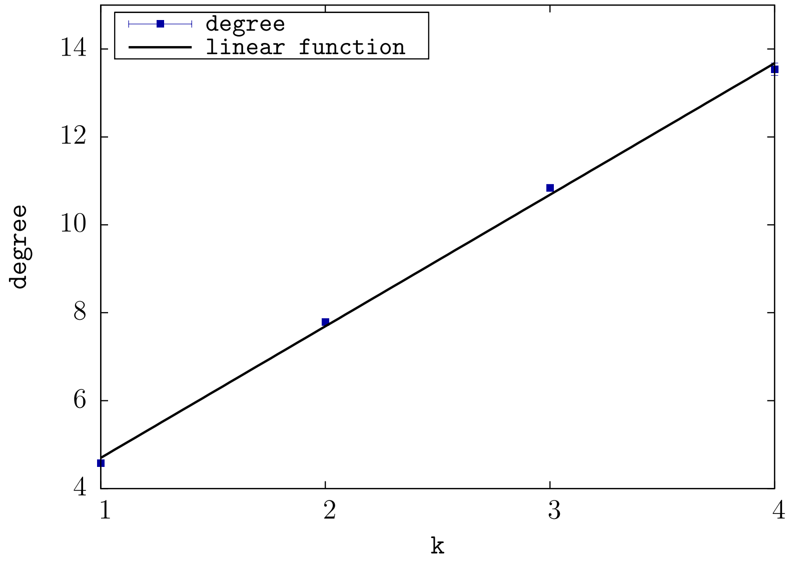

Figure 4.17 – Critical node degree in function of k

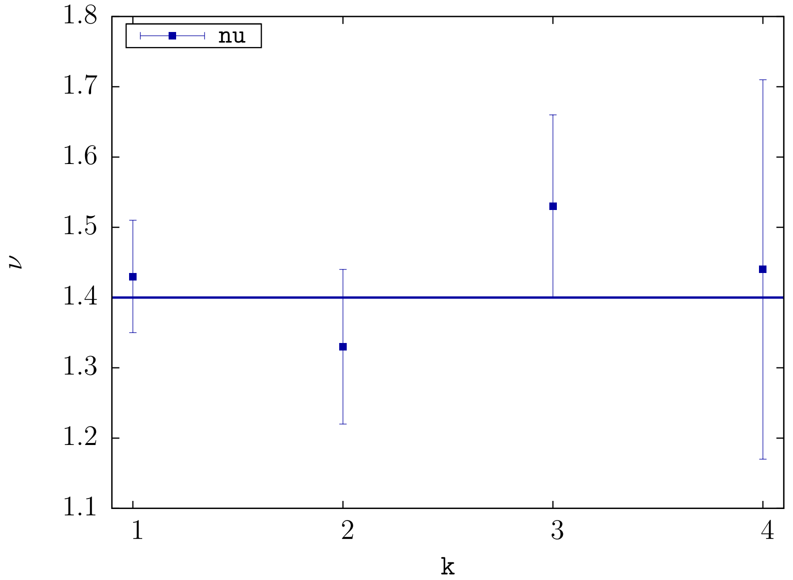

Figure 4.18 – Critical exponent ν in function of k

Figure 4.17 shows that the critical node degree linearly depends on k with linear coefficient 2.99 ± 0.08. This dependence allows us to predict the critical degree for any k-propagation process.

We can observe in Figure 4.18 that critical exponent ν remains close to the value of 1.4 with variations within the error margin. Note that we used the value of 1.4 for all curves in Figure 4.16.

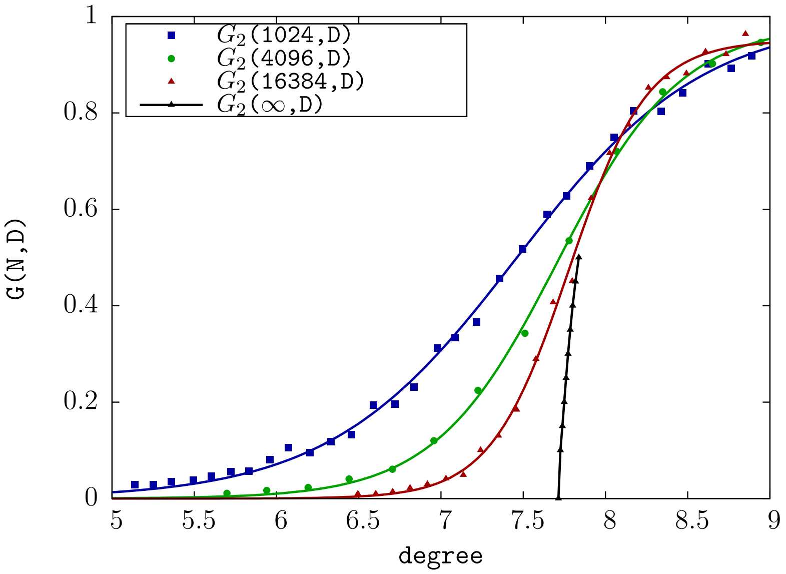

Figure 4.19 – 2-propagation: approximation of percolation probability by the finite size network

Figures 4.19 and 4.20 present the approximation of percolation probability by the finite size network for 2-propagation and 3-propagation, respectively. Finally, Figure 4.21 shows percolation probability for different values of k.

4.6 3-propagation in localization protocols

In this section, we take an example of network-wide localization protocols in sensor networks to show that they exhibit 3-propagation percolation. We compare our theoretical results with a detailed protocol simulation in ns-2 [79].

We assume that nodes use a two-way ranging technique consisting of measuring the round-trip time of signal transmission between two nodes. Timestamping at the physical layers enables accurate time measurements [47, 48, 49].

At the beginning, the network contains at least three non-collinear anchor nodes with overlapping communication ranges deployed in the center of the network. All other nodes are unlocalized. The localization process starts when the unlocalized nodes in the communication range of the anchor nodes perform ranging, estimate their positions, and become anchor nodes. The process repeats until all nodes know their positions.

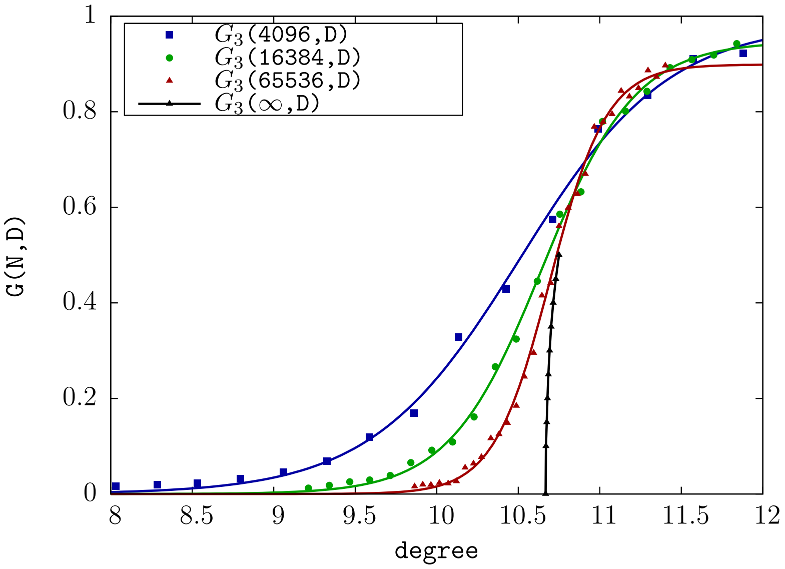

Figure 4.20 – 3-propagation: approximation of percolation probability by the finite size network

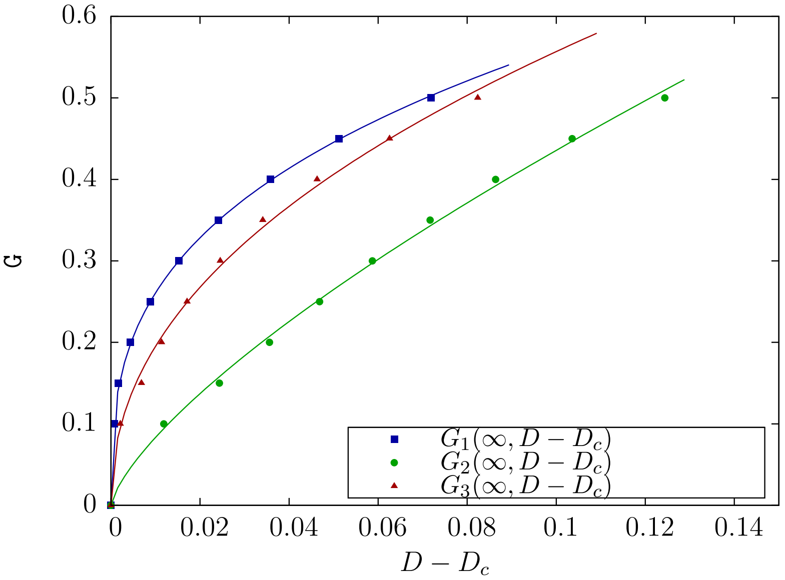

Figure 4.21 – Percolation probability

4.6.1 Evaluation

We have simulated the localization protocols with ns-2 [79] to take into account their detailed operation. We want to identify the critical node degree and study the convergence speed of propagation. The placement of nodes is slightly different from the Monte-Carlo simulator: we randomly place N sensor nodes in a square of size $L \times L$ m^2 according to the uniform distribution. We use Two Ray Ground propagation model for the radio communication range r and we assume the IEEE 802.11b PHY-MAC layers. As our propagation protocols generate low load, the MAC influence also remains low, because few collisions occur and do not have significant impact on propagation performance. We set the configuration parameters of the protocols described above to the following values: $T_B = 20$ s, $T_R = 1$ s, and $T_A = 3$ s.

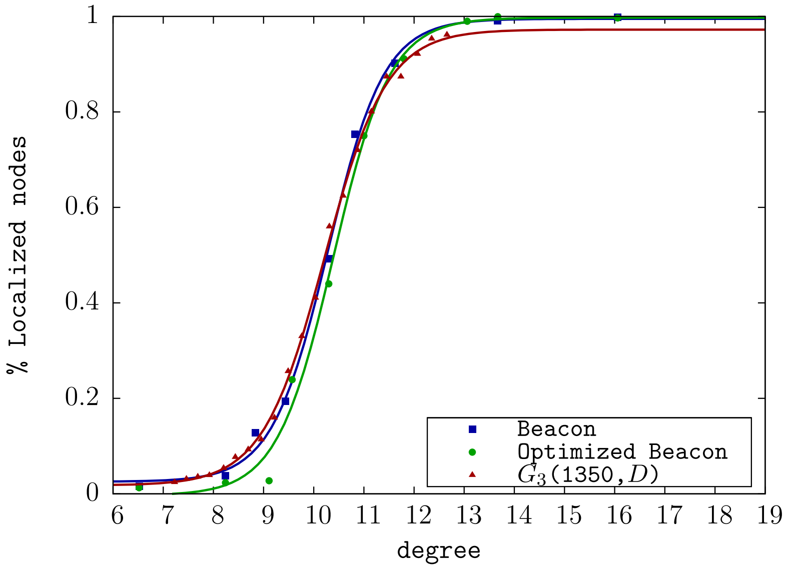

Figure 4.22 – Proportion of localized nodes in the network under different localization protocols, $N = 1350$ nodes, L=300 m

Figure 4.22 shows the proportion of localized nodes in the network after the end of localization in function of the average node degree. We can observe that the network requires a critical node degree to achieve localization. Moreover, we can also see that the results for both protocols perfectly match our curve $G_3(1350, D)$ and the overall shape is similar to the results in Section 4.5 (Figure 4.20). The figure shows that the protocols behave like our token model in a finite network [55].

4.7 Summary

In this chapter, we have introduced the model of k-propagation arising in several classes of protocols in wireless multi-hop networks. We have developed a Monte-Carlo simulator able to handle large-scale networks and studied the conditions in which information k-propagates to all nodes in the network. Our simulations show that k-propagation behaves as a percolation phenomenon and we can estimate its critical parameters. It may seem that many phenomena behave like percolation, so the result is straightforward, however, it is not always the case. In a next chapter, we show that the probability of successful packet delivery in a random geometric graph under the greedy geographic routing does not exhibit a percolation behavior [80].

Finally, we have analyzed protocols for network-wide localization as examples of the k-propagation processes and compared our theoretical results with detailed ns-2 simulations. The protocols behave exactly as predicted by our model of k-propagation.

We are aware that the random geometric graphs considered in this chapter only provide a first-order approximation of the real behavior of radio networks. The most important result from the network architecture point of view is that the network-wide localization may exchange GPS localization in case of really dense networks in which the average density exceeds 10-11 neighbors per node which is a very large value. We have not studied the accuracy of the localization techniques nor the influence of the accuracy on the behavior of geographical routing protocols.

5

Properties of Greedy Geographical Routing

In this chapter, we study general properties of geographical routing in spontaneous wireless mesh networks. We use a circular arena to place wireless nodes and study routing performance for different mesh sizes and densities. We adopt the Unit Disk Graph (UDG), a common connectivity model for representing multi-hop wireless networks. We consider it as a kind of a first-order approximation useful for studying main trends in performance behavior. Our goal is to identify and understand the behavior of the greedy geographical routing and to quantify its performance.

We start with the definition and description of a graph model based on a uniform distribution of nodes in a circular arena. We derive the distribution of node degree (the number of its neighbors). Then, we analyze the greedy geographical routing and show how the packet loss probability varies in function of network size. In particular, we demonstrate that the probability of packet delivery (probability of reaching the destination) is not a percolation phenomenon in contrast to network connectivity related to the existence of a network path between any two nodes or k-propagation. We also propose an analytical model of the greedy geographical routing and use it to derive the probability of packet loss. Our simulation results come from a dedicated graphical simulator written in C++ able to handle up to a million nodes.

5.1 Modeling spontaneous wireless mesh networks

5.1.1 Unit disk graph

As in previous chapter, we consider a spontaneous wireless mesh network with N nodes uniformly distributed in a disk $\mathcal{A}$ of radius $L = 1$ with area $A = \pi$. The probability density function of node position a_i is thus:

$$f(a_i) = \begin{cases} \pi^{-1} & a_i \in \mathcal{A} \\ 0 & a_i \notin \mathcal{A} \end{cases} \tag{5.1}$$

We assume that nodes use omnidirectional antennas and their transmission range is r with a perfect propagation model. Denoting by $|\cdot,\cdot|$ the Euclidean distance between any two nodes, there is a link in the graph between nodes a_1, a_2, if $|a_1, a_2| < r$.

5.1.2 Properties of the graph

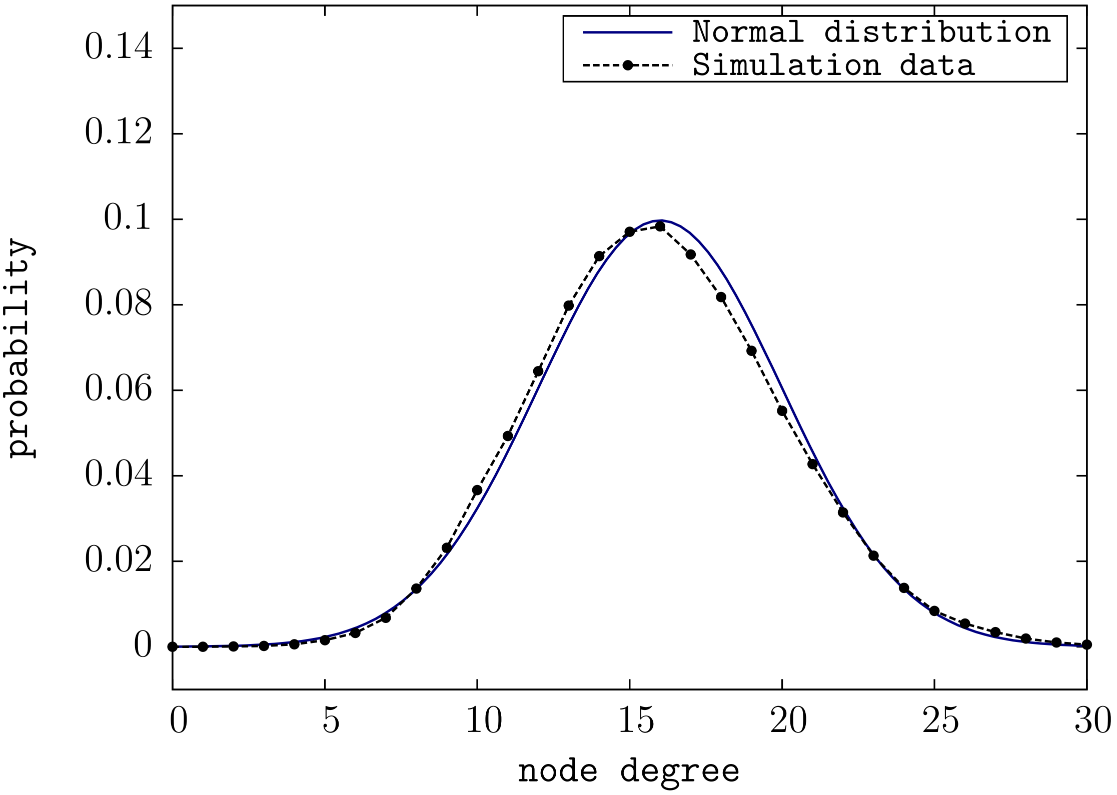

Figure 5.1 – Probability distribution of node degree, $N = 10000$, $L = 1$, $r = 0.04$

First, we want to find the distribution of node degree, i.e., the probability that a node a_i has d_i links with its neighbors. If we know (x, y), the position of node a_1, the probability p (resp. q) that node n_2 is inside (resp. outside) the disk of radius r placed at (x, y) is a Bernoulli distribution with parameters

$$p = \frac{r^2}{L^2} \quad \text{and} \quad q = 1 - p. \tag{5.2}$$

As the sum of Bernoulli distributions is a binomial distribution

$$P(d_i) = \binom{N-1}{d_i} p^{d_i} q^{N-1-d_i}, \tag{5.3}$$

the mean node degree $D = Np$ and its variance is $\mathrm{Var}[d_i] = Npq$. For large values of N, the distribution of the node degree converges to the normal (Gaussian) distribution $\mathcal{N}$:

$$\lim_{N\to\infty} P(d_i) = \mathcal{N}\left(D, \mathrm{Var}[d_i]\right). \tag{5.4}$$

Figure 5.1 shows a good match between the above theoretical result and simulation for $N = 10000$ nodes and $L = 1$, $r = 0.04$. In this model, D and $\mathrm{Var}[d_i]$ are finite, because from the relation

$$p = \frac{D}{N} = \frac{r^2}{L^2} \tag{5.5}$$

we can use $r = L\sqrt{\frac{D}{N}}$ to find its value for a network with given mean node degree D. Likewise, in the simulations below, we use the relation $D = N\frac{r^2}{L^2}$ to generate graphs with different values of the mean node degree by varying N, r, and L.

5.2 Greedy geographical routing

In this section, we study the behavior of the greedy geographical routing in a spontaneous mesh network: nodes forward packets based on the Euclidean distance $|\cdot,\cdot|$ by choosing the next hop with the least distance to the destination. If there is no such neighbor, the packet is dropped. We assume that all nodes know the positions of their neighbors.

We start with simulations of a large number of unit disk graphs for given mean node degree D and observe packet loss probability p_e of greedy geographical routing for a sufficiently large sequence of randomly chosen source and destination nodes (we simulate transmission of over 10^6 packets between randomly chosen nodes). We then derive an experimental expression for packet loss distribution as a function of mean node degree D.

5.2.1 Packet loss probability for the varying mean node degree

Figure 5.2 presents example simulation results, the experimental distribution of packet loss probability p_e in function of mean node degree D for $N = 1000$. A point in the figure represents the packet loss probability p_e obtained in a simulation of one randomly generated graph of given mean node degree D. We have confirmed by χ^2 testing that the distribution can be approximated by the Sigmoid function described by the equation:

$$p_e(D) = \frac{1}{1 + \exp(\frac{D-D_c}{\gamma})} \tag{5.6}$$

We have used the function to find the value of critical average node degree D_c for which the packet loss probability equals $p_e(D_c) = 0.5$ by fitting D_c and γ to simulation results with the least-square method. We can see in Figure 5.2 a steep decrease of packet loss

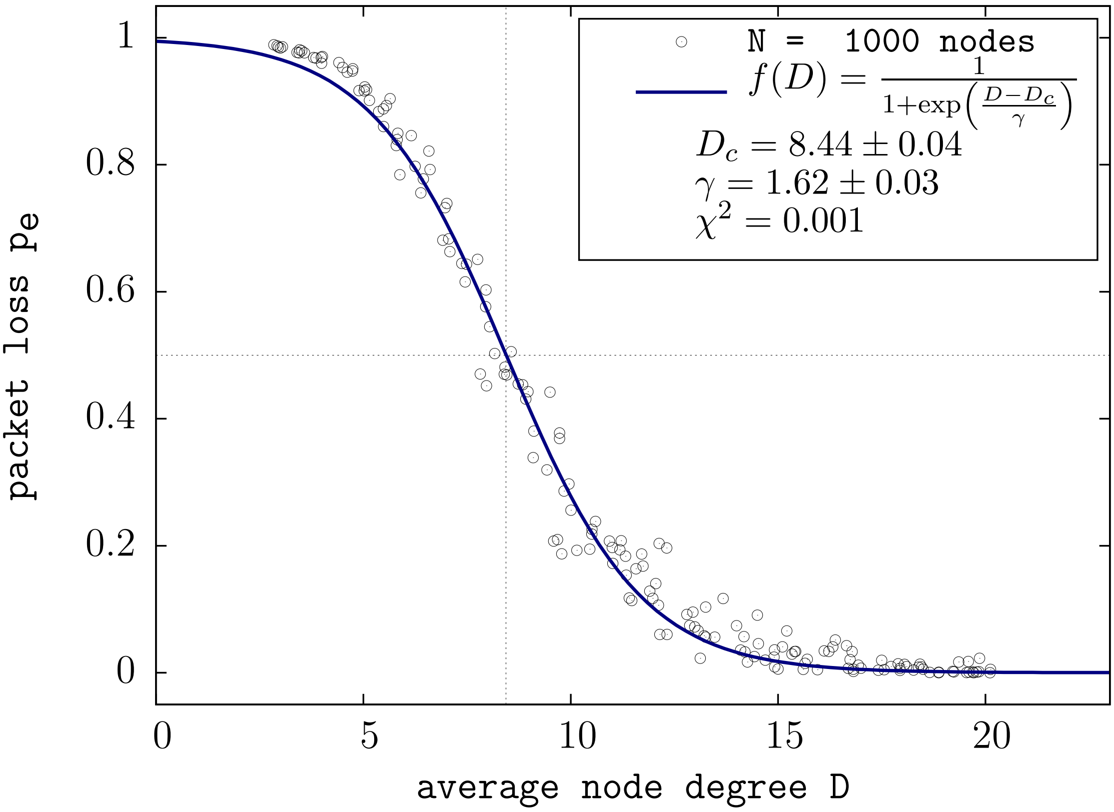

Figure 5.2 – Packet loss probability for $N = 1000$ nodes, $D_c = 8.44 \pm 0.04, \gamma = 1.62 \pm 0.03$

probability $p_e(\bar{k})$ around the critical average node degree of $D_c = 8.44$. For $N = 4000$, the results are similar, but astonishingly the critical average node degree D_c does not stay constant, but rather slowly grows with the increasing mesh size, i.e., $D_c = 9.90$ in this case.

Our observation is that packet losses occur only in fairly few places in the simulated graphs. For example in a graph with $N = 1000$ nodes and for $p_e = 0.5$, there are just few nodes (typically 10-15) that drop packets. These nodes can be identified as topology defects called concave nodes, because they only have neighbors in the backward direction–they appear at the border of holes in the mesh corresponding to nodes surrounding an empty concave region without links. Thus, an adequate model of the interaction of the geometry of the unit disk graph with greedy routing is a forest of defects with the constant (for given D) defect probability per unit area of the mesh. This means that a unit area of the graph has a specific opacity for packets under greedy routing analogous to the light passing through an opaque medium (extinction).

Let us study the behavior of the critical average node degree D_c and the packet loss probability when the size of the network increases. Let $p_e^{(1)} = \epsilon$ and $p_t^{(1)} = 1 - \epsilon$ denote the packet loss and transmission probabilities per unit area, respectively. If we quadruple the area of the mesh network while keeping the average node degree D, the average distance a packet travels in the quadrupled mesh doubles. Thus, a packet will cross two units of the graph on the average and transmission probability $p_t^{(2)}$ in the quadrupled mesh becomes $p_t^{(2)} = (1-\epsilon)^2$. So, the packet loss probability in the quadrupled mesh will be $p_e^{(2)} = 1-(1-\epsilon)^2 = 1-(1-p_e^{(1)})^2$, since the packet can be lost on each half of the way it travels (losses are approximately independent, if we neglect the interaction of the borders).

We can apply the above result to predict the value of the critical average node degree D_c for the 4000-node simulation using Eq. 5.6 and the value of the critical degree from the 1000-node simulation:

$$0.5 = 1-(1-p_e^{(1)})^2 = 1-\left(1-\frac{1}{1+\exp(\frac{D-D_c}{\gamma})}\right)^2, \tag{5.7}$$

which defines a new critical value of D for which $p_e^{(2)} = 0.5$. Putting the values $D_c = 8.44, \gamma = 1.62$ obtained for 1000 nodes into Eq. 5.7 and solving for D yields the value of $D_c \approx 9.87$ that fits the value of $D_c = 9.90$ from the simulation of 4000 nodes quite well. This enables us to estimate D_c of a larger network by simulating a much smaller mesh network.

We can further derive an approximate expression for the packet loss probability in a mesh with the area of 4^m units by generalizing the above reasoning. We obtain an expression for the critical node degree $D_c(m)$ given that the critical node degree D_c for the unit mesh is known: $p_e^{(m)} = 1-(1-p_e^{(1)})^m$. After inserting it into Eq. 5.6 and solving for $D(m)$ we get:

$$\lim_{m\to\infty} D_c(m) = \infty \quad \text{and} \quad \lim_{m\to\infty} p_e^{(m)} = 1. \tag{5.8}$$

Thus, an infinite mesh network under greedy geographical routing is completely opaque independently of the average node degree. This also means that the performance of greedy forwarding becomes worse with increasing mesh size when the mean node degree is kept constant. We have compared our results on the greedy geographical routing in spontaneous mesh networks with the classical percolation theory [81]. It states that an infinite percolating system needs to have a unique but finite critical average node degree D_c for which a measurable quantity of interest (in our case the packet loss probability) experiences a sharp transition. The critical average node degree may slightly depend on the size of the observed system (so called a finite size effect), but it can be estimated for an infinite system size, if a sequence of critical average node degree $\bar{k}_0(m)$ for differently-sized finite systems is plotted against $Z_m^{-\frac{1}{\nu}}$, that is the system size Z_m (corresponding to the radius L of the mesh network in our case) to the power of a dimension dependent, but universal percolation exponent $-\frac{1}{\nu}$. The exponent is known to be constant and equal to $\nu = \frac{4}{3}$ for any two dimensional system. The crossing of the resulting line with the x-axis should then give the critical average node degree for an infinite system.

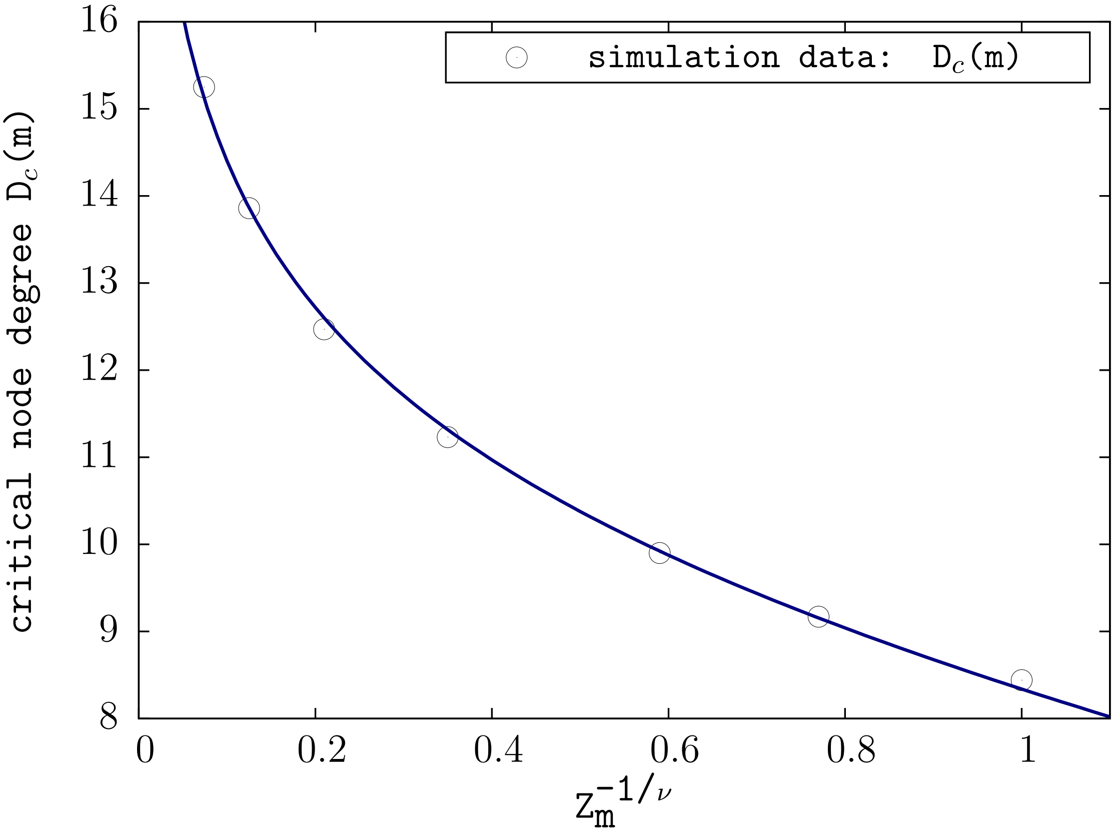

Figure 5.3 – Packet loss probability in mesh networks with greedy geographical routing in mesh networks does not behave like a percolating system

Figure 5.3 presents the corresponding percolation graph. We can see that no straight line can be drawn in the case of greedy geographical routing in a mesh network, but rather we can observe an exponential growth of the critical average node degree. Thus, despite the fact that a mesh network itself is a percolating system in the sense of the global connectivity [56, 82], this property does not hold for greedy geographical routing in spontaneous mesh networks with the packet loss probability as the percolation measure.

5.3 Analytical Model of Greedy Geographical Routing

In this section, we present an analytical model for packet loss probability p_e (and thus for successful transmission probability p_t of delivering a packet) in a mesh network with greedy geographical routing and a given average node degree for any randomly chosen source and a destination.

5.3.1 Probability that a neighbor is closer to the destination

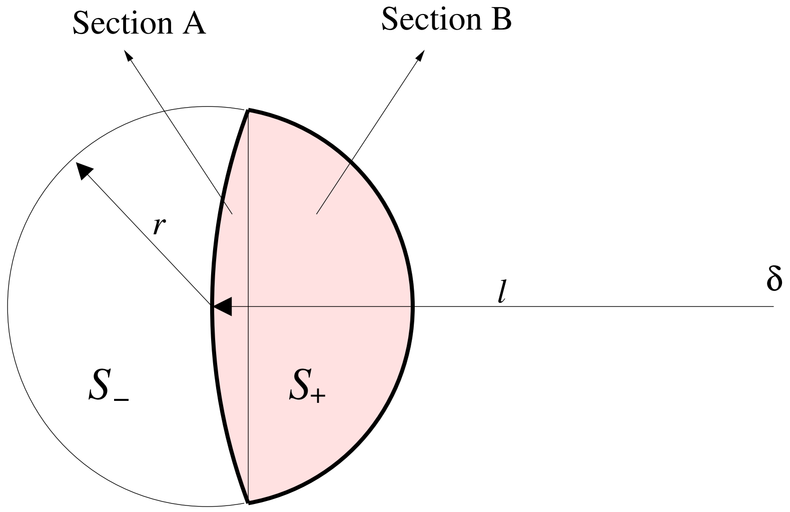

Figure 5.4 – Packet loss model

Let us consider two disks $\mathbb{D}(0,r)$ and $\mathbb{D}(\delta,l)$ where $|\delta - l| < r$ (cf., Figure 5.4 for $l = \delta$). Disc $\mathbb{D}(0,r)$ is placed at the position of the node that forwards a packet, while $\mathbb{D}(\delta,l)$ is located at the destination node. For $l = \delta$, S_+ (resp. S_-) is the area of positions closer to (resp. farther of) the destination node. The intersection points (x,y) of $\mathbb{D}(0,r)$ and $\mathbb{D}(\delta,\delta)$, $\delta \geq r$ satisfy the following system of equations:

$$\begin{cases} x^2 + y^2 & = & r^2 \\ (x-\delta)^2 + y^2 & = & \delta^2 \end{cases} \tag{5.9}$$

By solving it and summing up the area of disk sections $S_+ = A + B$, we can express probability p_c that a randomly selected neighbor of a forwarding node is closer to the destination than the forwarding node as the ratio of S_+ to the area of the smaller disk πr^2:

$$p_c = \frac{\left(r^2 \arcsin\left(\xi_r\right) + \delta^2 \left(\arcsin\left(\xi_\delta\right) - \xi_\delta\right)\right)}{\pi r^2}, \tag{5.10}$$

where

$$\xi_r = \frac{1}{2r}\sqrt{4r^2 - \frac{r^4}{\delta^2}}, \quad \xi_\delta = \frac{1}{2\delta}\sqrt{4r^2 - \frac{r^4}{\delta^2}}. \tag{5.11}$$

Figure 5.5 presents probability p_c in function of the distance between the forwarding node and the destination expressed in multiples of range r. Its minimum value is $p_c|_{\delta=r} \approx 0.39$ for the angle α corresponding to the case of $\delta = r$:

$$\alpha = 2\arcsin\left(\xi_r\right) = \frac{2}{3}\pi = 120^\circ \tag{5.12}$$

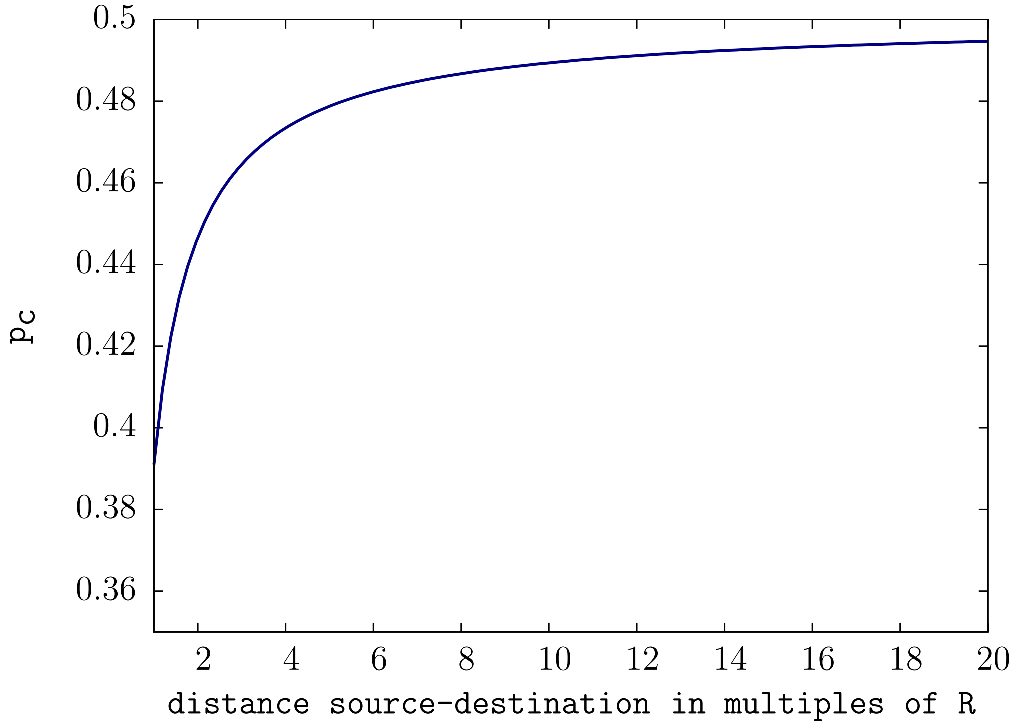

Figure 5.5 – Probability p_c that a neighbor is closer to the destination

Note that $\lim_{\delta \to r^+} 2 \arcsin (\xi_r) = \alpha$, which means that if a node has a disk sector with an angle bigger then or equal $\alpha \geq 120^\circ$ not containing any neighbor, it may drop packets. Whether a packet is lost or not, depends on the direction towards the destination.

5.3.2 Packet loss probability

To find the packet loss probability, we randomly choose source and destination nodes. Let denote the distance between them by z. We consider topology defects on the route to the destination–what is the probability of dropping a packet at a concave node. If not dropped, a packet will advance toward the destination by choosing the neighbor node for which the gain in distance is the best. So, first we compute the probability distribution of the distance that a packet will gain towards the destination when being forwarded by a given neighbor node. We then find the distribution of the maximum distance gain over the neighbors that are closer to the destination and we obtain the final result by considering the distribution of the distance between any two nodes in a mesh network.

Recall that $P(d_i)$ given by Eq. 5.3 is the probability that a given node has d_i neighbors in its transmission range. Forwarding fails if packet arrives at a concave node (none of its neighbors is closer to the destination). A neighbor lies backwards with respect to the destination with probability $q_c = 1 - p_c$. We can express p_x, the probability of being a concave node as:

$$p_x = \sum_{j=1}^{N-1} \binom{N-1}{j} p^j q_c^{j-1} q^{N-j-1} = \frac{(pq_c + q)^{N-1} - q^{N-1}}{q_c}. \tag{5.13}$$

Notice, that a forwarding node always has one node in a backward direction, because this node delivers a packet. This is a reason why, we have q_c^{j-1} term.

To find the probability distribution that a given neighbor is closer to the destination, we consider the case in Figure 5.4 for $\delta > r$ and $|\delta - l| \leq r$. To obtain the positions of the disk intersection, we can solve Eq. 5.9 for this case to obtain:

$$\begin{cases} y' &= \pm\sqrt{r^2 - \frac{(-l^2+\delta^2+r^2)^2}{4\delta^2}} \\ x' &= \frac{-l^2+\delta^2+r^2}{2\delta}. \end{cases} \tag{5.14}$$

Let us define

$$\zeta = \frac{1}{l}\sqrt{r^2 - \frac{(-l^2+\delta^2+r^2)^2}{4\delta^2}}. \tag{5.15}$$

The length of the arc of the larger circle located inside the smaller disc is $A_L = 2l \arcsin(\zeta)$. Knowing that

$$S_+ = \int_{l-r}^{l} A_L \, dl \rightarrow \int_{l-r}^{l} \frac{A_L}{S_+} \, dl = 1, \tag{5.16}$$

we can define the probability distribution that a given neighbor is closer to the destination by length ρ as $p_d(\rho) = \frac{A_L}{S_+}$. When the forwarding node has on average D neighbors, approximately $d' = p_c \cdot D$ of them are closer to the destination. To find how far a packet advances towards the destination in one hop, consider neighbor $i, i = 1, \ldots, d'$. It is closer to the destination by length ρ_i distributed according to p_d. Thus, the distance by which a packet advances towards the destination in one hop is $\Delta = \max\{\rho_1 \ldots \rho_{d'}\}$. We can find the distribution of Δ in the following way. First, we need to find the cumulative density function of ρ_i as $F_{\rho_i}(x) = \int_0^x p_d(\rho)\, d\rho$. Then, the cumulative distribution of Δ is $F_\Delta(x) = (F_{\delta_i}(x))^{d'}$ and its probability density function $f_\Delta(x) = \frac{d}{dx} F_\Delta(x)$. So, if the destination is at distance z, a packet will reach the destination in approximately $\bar{h} = \frac{z}{E[\Delta]}$ hops on average.

The probability density function of distance $z \in [0, 2L]$ between two randomly chosen points in a disc with radius L is the following[83]:

$$\Lambda(z) = \frac{z}{\pi L^2}\left(4 \arctan \frac{\sqrt{4L^2 - z^2}}{z} - \frac{z}{L}\sqrt{4 - \left(\frac{z}{L}\right)^2}\right). \tag{5.17}$$

Finally, we can obtain the upper bound of the packet loss probability as:

$$p_e(D) = \int_0^{2L} \Lambda(z) \left(1 - (1 - p_x|_{\delta=r})^{\bar{h}|_{\delta=z}}\right) dz, \tag{5.18}$$

where $r = L\sqrt{\frac{D}{N}}$.

5.3.3 Numerical results

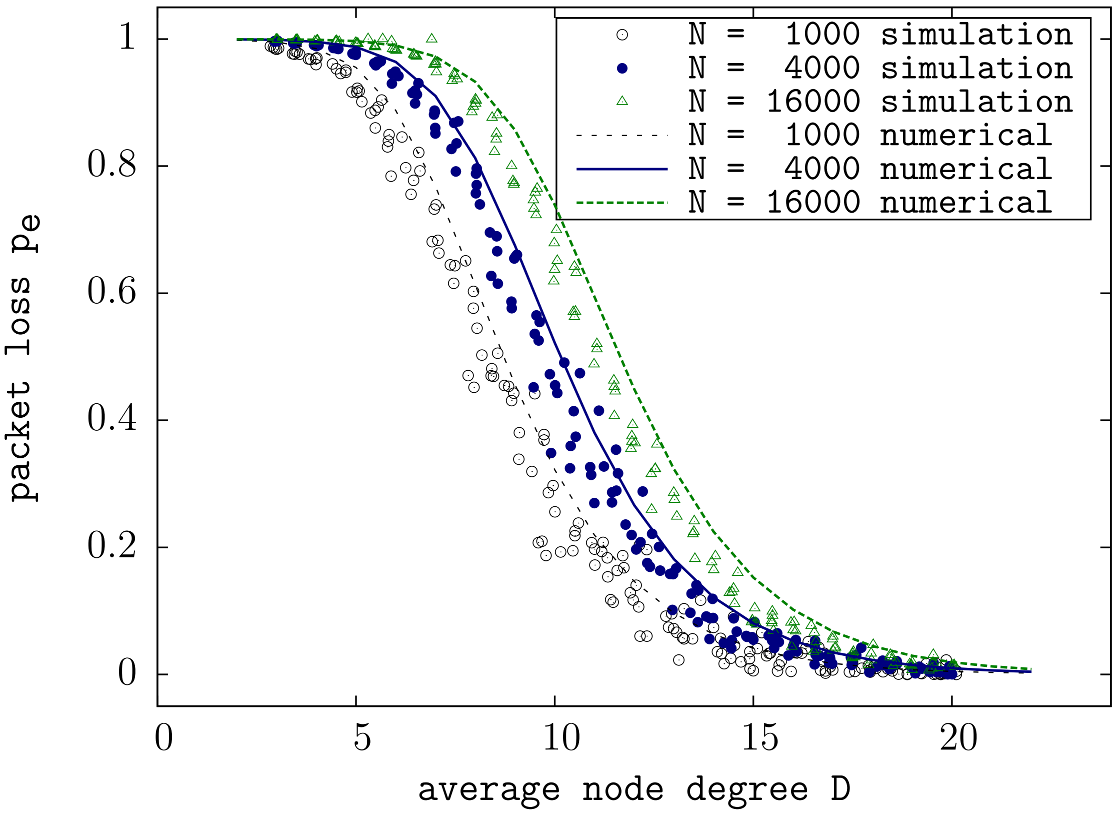

Figure 5.6 – Packet loss probability: numerical computation vs. simulation

We numerically compute the packet loss probability derived above. Figure 5.6 shows the analytical results compared with the curves fitting the simulation data. We can see that the form of the analytical curve corresponds to the curve that fits simulation data, the slight difference coming from several approximations made in the derivation of packet loss probability. To evaluate the approximation error, we can compare critical average node degree $D_c = 10.15$ obtained from the numerical results for $N = 4000$ with the corresponding value of $D_c = 9.90$ given by simulation data.

5.4 Summary

The goal of this chapter is to analyze the greedy geographical routing in spontaneous wireless mesh networks. Although we have used a simplistic model based on the unit disk graph, our analysis yields several interesting properties. First, we can approximate the dependence of the mean node degree on the packet loss probability with a Sigmoid-like function. When the average node degree D remains constant, but the mesh size increases, the network becomes more opaque to packets. We also show by simulation that the packet loss probability in mesh networks with the greedy geographical routing does not exhibit the behavior of a percolating quantity. Finally, we propose an analytical model of the greedy geographical routing and use it to derive the packet loss probability. In the next chapter, we show how to improve the greedy geographical forwarding using the knowledge on previously forwarded packets.

6 Binary Waypoint Routing

In this chapter, we present a novel geographical routing protocol that offers high packet delivery rates in wireless mesh networks with voids. The idea of Binary Waypoint Routing is to learn and maintain source routes to a small number of nodes called binary waypoints that are placed in subspaces constructed as a result of binary space partitioning. A set of subspaces whose size decreases exponentially, covers the whole addressing space. To forward a packet, a node sends it to the waypoint corresponding to the subspace of the destination. Each intermediate node tries to adapt the route by aiming at its waypoint that may be closer to the destination.

The source routes to waypoints provide a means for reaching any destination in the addressing space. As the number of subspaces grows in a logarithmic way with the space size, the size of routing tables in any node remains very small compared to the size of the mesh network. The proposed scheme does not require explicit construction of routing tables and its operation does not depend on any graph construction algorithm such as UDG, so the protocol works for any type of wireless connectivity between nodes.

In this chapter, we explain the principles of the protocol and show how it guarantees packet delivery. We also evaluate its performance through simulations and compare it with other approaches (Greedy Routing and Optimal Shortest Path). In our simulations, we randomly place wireless nodes in a circular arena and create connectivity based on the UDG model. Even if UDG is only a first-order approximation of real wireless connectivity, such a model enables performance comparisons between the routing protocols and results in an initial insight into main performance trends. We observe forwarding performance of different routing protocols for randomly chosen pairs of a source and a destination. Our comparisons show that our scheme achieves high packet delivery rate without the need for learning the global topology. It results in a traffic pattern similar to the Optimal Shortest Path Routing.

The rest of this chapter is organized as follows. We explain the principles of the proposed routing scheme (Section 6.1) and analyze its performance (Section 6.2). Finally, we present some conclusions (Section 6.3).

6.1 Binary Waypoint Routing

Chapter 5 shows that packet losses only occur in fairly few concave nodes that are topology defects: they only have neighbors in the backward direction and they appear at the border of concave voids in the mesh. The idea of Binary Waypoint Routing is to define a forwarding scheme that attempts to go around voids based on routes learnt from destinations.

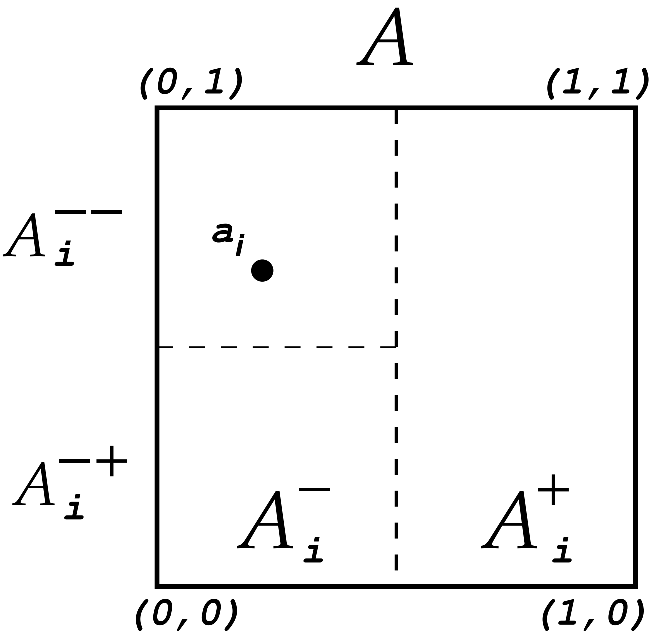

Figure 6.1 – Binary address space partitioning

6.1.1 Binary Address Space Partitioning

Assume that each node knows its geographical coordinates $a_i = (x_i, y_i) \in \mathcal{A}$, where $\mathcal{A} = [0,1] \times [0,1]$ is a finite square addressing space. Node a_i divides the addressing space into an unbalanced binary tree of progressively smaller subspaces $\mathcal{S}_i$ that fulfill the following conditions:

$$\bigcup_j \mathcal{S}_i^j = \mathcal{A},\ i = 1, ..., N,\ j = 0, ..., m \tag{6.1}$$

$$\forall j \neq k :\ \mathcal{S}_i^j \cap \mathcal{S}_i^k = \emptyset \tag{6.2}$$

Partitioning terminates when the length of the shortest border edge of the smallest subspace is not shorter than parameter d_l, which can be the maximal or the average distance to an immediate neighbor node. In the end of partitioning, the smallest subspace contains node a_i.

Figure 6.1 illustrates partitioning of the addressing space. At the beginning, the tree of node a_i is empty: $\mathcal{S}_i = \emptyset$. Let us define two subspaces of node a_i [1]:

- $\mathcal{A}_i^+$ such that it does not contain node a_i, i.e. $a_i \notin \mathcal{A}_i^+$, in our case $\mathcal{A}_i^+ = [0.5, 1] \times [0, 1]$,
- $\mathcal{A}_i^-$ such that it contains node a_i, i.e. $a_i \in \mathcal{A}_i^-$, in our case $\mathcal{A}_i^- = [0, 0.5] \times [0, 1]$.

The node inserts $\mathcal{A}_i^+$ into $\mathcal{S}_i$, i.e. $\mathcal{S}_i^0 = \mathcal{A}_i^+$ and checks whether it can further partition $\mathcal{A}_i^-$ by comparing its border edge to d_l. If so, it partitions $\mathcal{A}_i^-$ into $\mathcal{A}_i^{-+}$ such that $a_i \notin \mathcal{A}_i^{-+}$ and $\mathcal{A}_i^{--}$ such that $a_i \in \mathcal{A}_i^{--}$, and inserts $\mathcal{A}_i^{-+}$ into $\mathcal{S}_i$, i.e. $\mathcal{S}_i^1 = \mathcal{A}_i^{-+}$. Partitioning continues until the space is too small for further splitting: the last subspace $\mathcal{A}_i^{-\cdots-}$ that contains node a_i is inserted into $\mathcal{S}_i$.

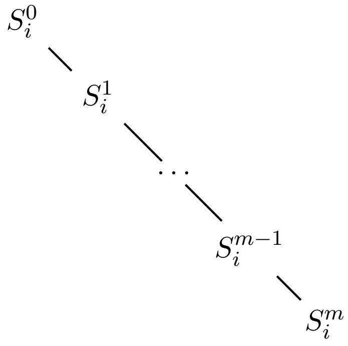

Figure 6.2 – Unbalanced binary tree of addressing subspaces

Partitioning returns a set of $m + 1$ subspaces S_i that fulfill Eqs. 6.1, 6.2–they cover the addressing space and their size decreases exponentially:

$$\mathcal{S}_i = \left\{\mathcal{A}_i^+, \mathcal{A}_i^{-+}, \ldots, \mathcal{A}_i^{-\cdots-+}, \mathcal{A}_i^{-\cdots--}\right\}. \tag{6.3}$$

Figure 6.2 shows the resulting unbalanced binary tree $\mathcal{S}_i$ of length $m + 1$. Note that the area of the address subspaces varies from very large to very small. As binary partitioning divides the size of an address subspace by two, the number of subspaces in $\mathcal{S}_i$ grows logarithmically with the number of nodes in the network:

$$m \sim \log N. \tag{6.4}$$

6.1.2 Routing tables

Node a_i maintains a routing table $\mathcal{R}_i$ of size $m + 1$. Entry j of the routing table contains a source route to node $a_j \in \mathcal{S}_i^j$ called a binary waypoint for subspace $\mathcal{S}_i^j$:

$$\mathcal{R}_i[j] = (a_1, a_2, \ldots, a_j) \text{ such that } a_j \in \mathcal{S}_i^j \tag{6.5}$$

[1] Note that the node can split square spaces either vertically or horizontally, but rectangles need to be cut into squares.

i.e. the source route is composed of all intermediate nodes between node a_i and waypoint a_j in $\mathcal{S}_i^j$. So, the routing table $\mathcal{R}_i$ contains one route to a waypoint node per subspace.

At the beginning, routing table $\mathcal{R}_i$ is empty. We assume that each packet records the route taken towards a destination: it is a list of all nodes that have forwarded a packet between source node a_s and forwarding node a_f:

$$r = (a_s, \ldots, a_f) \tag{6.6}$$

When node a_f receives a packet, it analyzes its source a_s and identifies to which subspace $\mathcal{S}_f^s$ it belongs. Then, it fills $\mathcal{R}_f[s]$ with the inverted route $r^{-1} = (a_f, \ldots, a_s)$ which leads to a_s, so a_s becomes the waypoint node a_w for the subspace $\mathcal{S}_f^s$.

Nodes keep the best routes to waypoints in all subspaces. To compare routes they need to use a metric–we propose the following one: the average distance per hop of route r defined as:

$$H(r) := \begin{cases} |a_f, a_s|/\text{Length}[r] & \text{Length}[r] \neq 0, \\ 0 & \text{Length}[r] = 0, \end{cases} \tag{6.7}$$

where $|a_i, a_j|$ is the Euclidean distance between nodes a_i and a_j, Length$[r]$ is the length of route r in hops.

If the routing table $\mathcal{R}_f[s]$ is already loaded, node a_f replaces its routing table entry with the route recorded in a received packet, if its average distance per hop is greater than that of the route stored in the routing table entry, i.e. $H(r) > H(\mathcal{R}_f[s])$

6.1.3 Binary Waypoint Forwarding

When a source node a_s wants to send a packet to destination a_d, it first finds route $r_d = \mathcal{R}_i[d]$ to waypoint node a_w in subspace $\mathcal{S}_i^d$ such that $a_d, a_w \in \mathcal{S}_i^d$. If the route exists it then inserts the route in the packet and sends it to the first node in the source route. If the routing entry is empty node uses greedy forwarding and sends the packet to the neighbor that has the least distance to the destination.

Each node that forwards the packet checks whether it has a better waypoint node for the destination, e.g., an intermediate node a_j compares its route $r'_d = \mathcal{R}_j[d]$ such that $\mathcal{S}_j^d : a_d \in \mathcal{S}_j^d$ with the remaining source route of the packet. Assume that route r'_d has a waypoint $a_{w'}$, if $|a_w, a_d| > |a_{w'}, a_d|$ or the remaining source route in the packet is empty the intermediate node replaces it with its route r'_d, because its waypoint is closer to the destination. At the end node sends the packet to the first node in the source route or uses greedy forwarding if remaining source is still empty.

Let us show that when routing tables $\mathcal{R}_i$ contain sufficient information (all nodes have at least one route to each of their subspaces), waypoint forwarding guarantees packet delivery. Consider Figure 6.3: in this example destination a_d is in subspace $\mathcal{A}_s^+$ of source node a_s. It sets the route of a packet to waypoint a_w in $\mathcal{A}_s^+$: $r_d = \mathcal{R}_s[\mathcal{S}_s^0 = \mathcal{A}_s^+]$. Assume the worst case in which there is no intermediate node along route r_d that knows

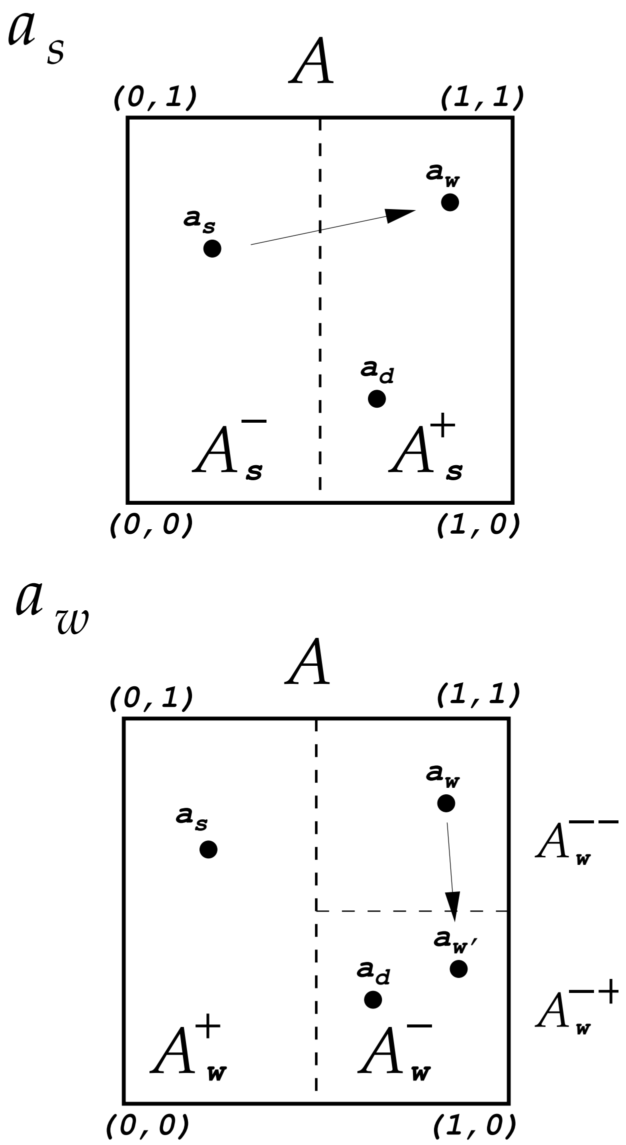

Figure 6.3 – The worst case in Binary Waypoint Forwarding. The view on the addressing space of nodes a_s and a_w

a better route to the destination. When waypoint node a_w receives the packet, it finds new route $r'_d = \mathcal{R}_w[\mathcal{A}_w^{-+} = \mathcal{S}_w^1]$ with another waypoint node $a_{w'}$. As the size of each subspace exponentially decreases, this forwarding process advances the packet into an increasingly smaller area near the destination. In the last step, the packet arrives in a node within distance d_l from the destination. This last node can directly deliver the packet to the destination.

Figure 6.4 illustrates route adaptation when an intermediate node knows a better

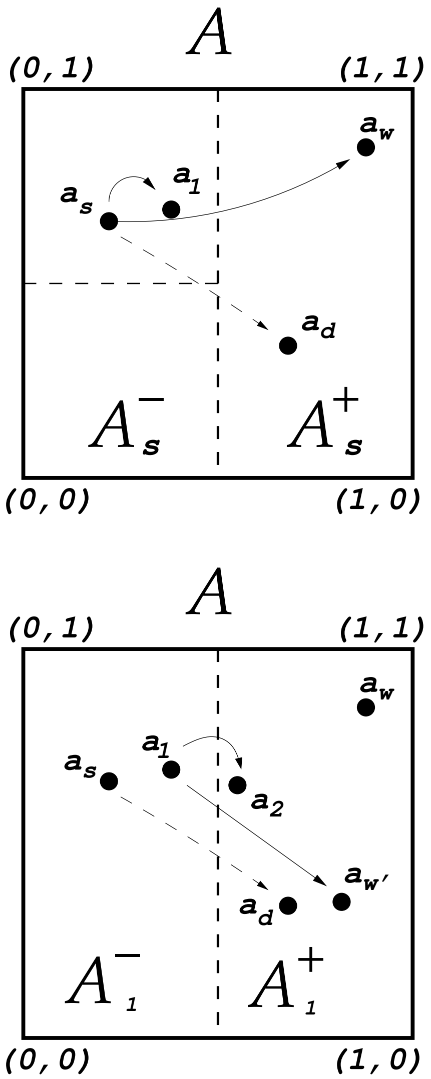

Figure 6.4 – Binary Waypoint route adaptation

route to the destination. Assume that source node a_s sends a packet to destination node a_d. As previously, it finds that a_d lies in $\mathcal{A}_s^+$ and sets the packet source route $r_d = \mathcal{R}_s[\mathcal{S}_s^0 = \mathcal{A}_s^+]$ to waypoint a_w in $\mathcal{A}_s^+$. Assume that the next hop on the route is node a_1. It finds that the destination lies in its $\mathcal{A}_1^+$ and it has a route $\mathcal{R}_1[\mathcal{A}_1^+]$ with waypoint node $a_{w'}$ that is closer to destination a_d. It thus updates the packet source route with the new route and sends the packet towards the new waypoint $a_{w'}$. In this way, a packet can advance towards the destination in few hops even if at the beginning the source node only knows a route to a waypoint node, which may be far away from the destination.

6.1.4 Voids and obstacles

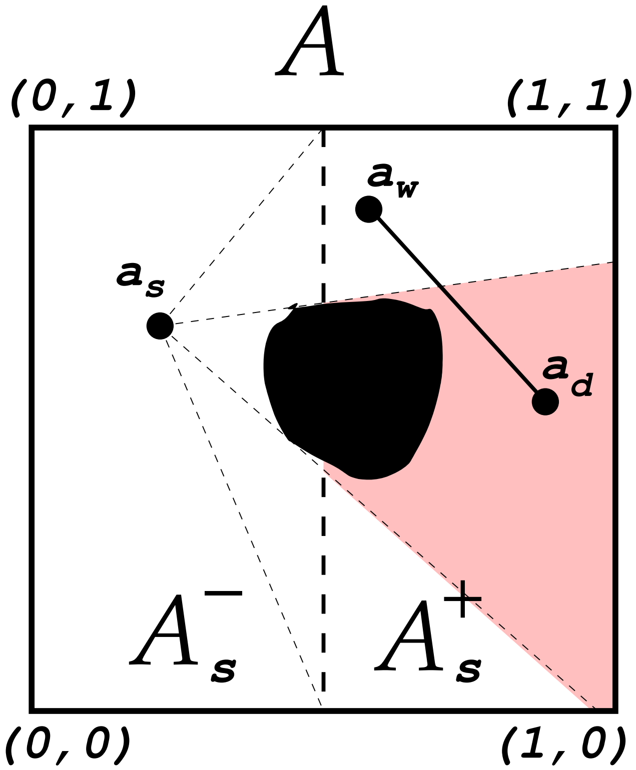

Figure 6.5 – Binary Waypoint goes around voids

Figure 6.5 shows the behavior of the protocol when there is a void in the network. Source node a_s wants to send a packet to destination node a_d in A_s^+. Assume that the void shadows some part of the subspace $\mathcal{A}_s^+$ with respect to node a_s: the source can receive packets along straight routes from the white area of subspace $\mathcal{A}_s^+$, but it does not receive packets from the shadow region. Recall that at the beginning nodes use greedy forwarding that results in almost straight routes. The source node thus learns routes to nodes from the visible area and stores them in its routing table. Assume that the source node has a route to waypoint a_w for subspace $\mathcal{A}_s^+$. At the beginning, packets will go to waypoint a_w, which is a good thing, because they will not go in the direction of the void. A packet on the route to waypoint a_w will probably pass through other nodes that have already received packets from nodes closer to the destination. It is likely, because we can connect a_w and a_d with a straight line: nodes close to a_w receive packets coming from regions near a_d and they record this information in their routing tables, which is then used to adapt routes of packets going to the destination.

6.2 Performance of Binary Waypoint Routing

In this section, we present simulation results of the proposed routing scheme in a large scale wireless mesh network.

6.2.1 Unit disk graph

We generate our simulations based on a model of wireless mesh networks considered previously [80]: a network is composed of N nodes uniformly distributed in a disk $\mathbb{D}$ of radius L with area $Z = \pi L^2$. We assume that disk $\mathbb{D}$ is embedded in the addressing space $\mathcal{A}$. The probability density function of node position (x, y) is thus:

$$p_l(x,y) = \begin{cases} 1/Z & (x,y) \in \mathbb{D} \\ 0 & (x,y) \notin \mathbb{D} \end{cases} \tag{6.8}$$

We assume that nodes use omnidirectional antennas and their transmission range is R with a perfect propagation model: there is a link in the connectivity graph between nodes a_1, a_2, if $|a_1, a_2| < R$. Under such assumptions, probability p_r (resp. q_r) that node a_2 is inside (resp. outside) the disk of center a_1 and radius R is a Bernoulli distribution with parameters

$$p_r = \frac{R^2}{L^2} \quad \text{and} \quad q_r = 1 - p_r \tag{6.9}$$

so that the mean node degree $\bar{k}$ is Np_r and its variance $\mathrm{Var}[k] = Np_rq_r$. For large values of N, the distribution of the node degree converges to the normal (Gaussian) one.

6.2.2 Comparisons with Greedy Routing and Optimal Shortest Path

We generate a random uniformly distributed network with $N = 10,000$ nodes in a disk embedded in the address space $\mathcal{A} = [0,1] \times [0,1]$. We assume $L = 1.0$ and the radio range of $R = 0.01425$. Under these conditions after 10 simulations the average node degree is equal to 9 ± 0.03. In each simulation, we randomly choose a pair of source and destination for 200,000 packets. They are routed according to the Greedy Geographical Routing. We measure the packet loss rate p_e and obtain the value of $p_e = 0.791 \pm 0.023$, which is fairly high. Next, we simulate the behavior of Binary Waypoint Routing: we have measured p_e as a function of the total number of packets that have been sent in the network (cf., Figure 6.6). We can see that after 100,000 packets, which is 10 packets per node on the average, nodes have learned routes to waypoints and sending more packets almost does not improve the delivery rate in the network. The loss rate for Binary Waypoint Routing attains $p_e \sim 9\%$. The figure also presents the performance of the Optimal Shortest Path Routing in which all nodes have the complete knowledge of the network topology and use the shortest route to the destination. We obtain very

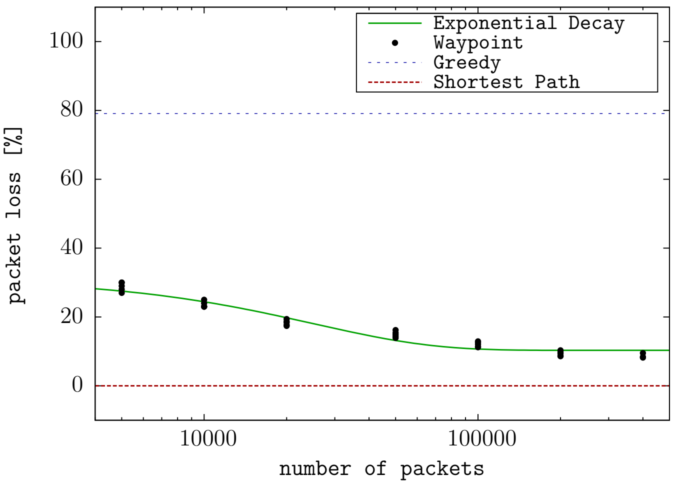

Figure 6.6 – Binary Waypoint Routing, $N = 10,000, R = 0.01425$

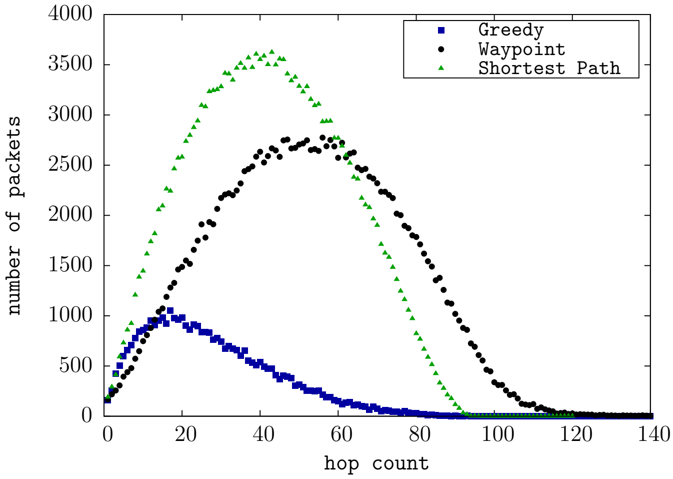

Figure 6.7 – Packet statistics for parameters $N = 10,000, R = 0.01425$ and $200,000$ packets

small value of $p_e = 570 \pm 365$ ppm, which is not zero, because the simulated mesh network is not fully connected for $R = 0.01425$.

Figure 6.7 shows the total number of packets delivered after a specific number of hops for each routing scheme. We can see that Greedy Routing cannot forward packets to destinations that are far away. Moreover, we can see that the shape of the curve for Binary Waypoint Routing is not much different from the shape for Optimal Shortest Path Routing, but more packets are lost.

We ran another simulations for a larger network ($N = 40,000$), but for the radio range $R = 0.00731$ such that the average node degree is almost the same as in the first

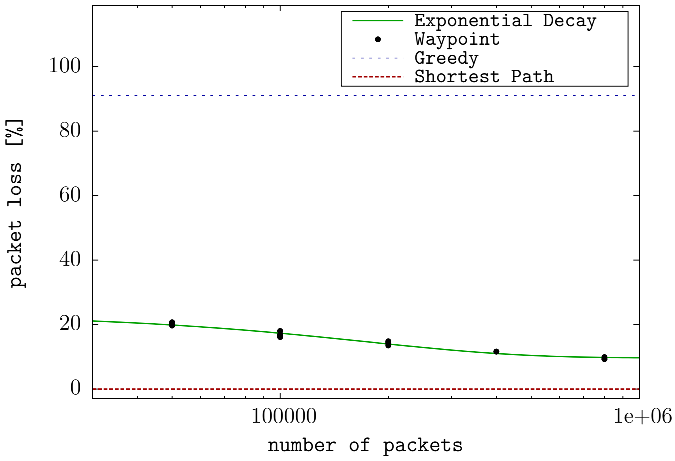

Figure 6.8 – Binary Waypoint Routing, $N = 40,000, R = 0.00731$

experiment: 9.01 ± 0.03. We obtain $p_e = 0.91 \pm 0.005$ for Greedy Routing. Binary Waypoint Routing attains almost the same performance as in the first experiment: $p_e \sim 9\%$ (cf., Figure 6.8). This means that the delivery rate remains constant for a given average node degree even though we have increased the network area four times.

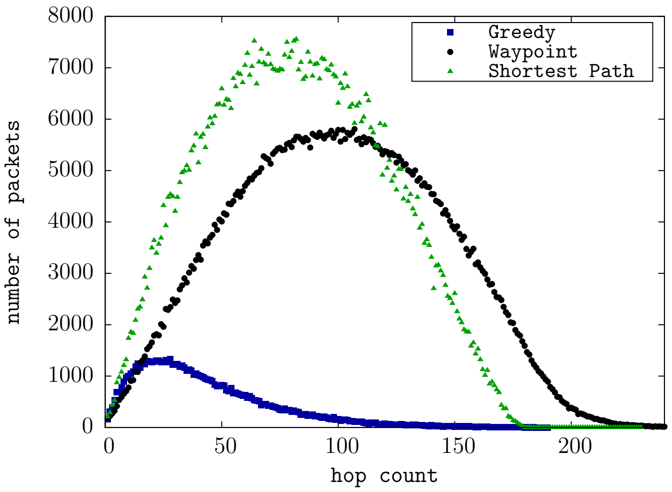

Figure 6.9 – Packet statistics for parameters $N = 40,000, R = 0.00731$, and $800,000$ packets

Figure 6.9 presents hop statistics for this case: the average hop count for Binary Waypoint Routing is 102.5 and 82.4 hops for the Optimal Shortest Path. This means that routes in Binary Waypoint Routing are on the average almost 25% longer.

To decrease the packet loss rate in Binary Waypoint Routing we need to increase average node degree–in the next simulation, we have increased the coverage area up

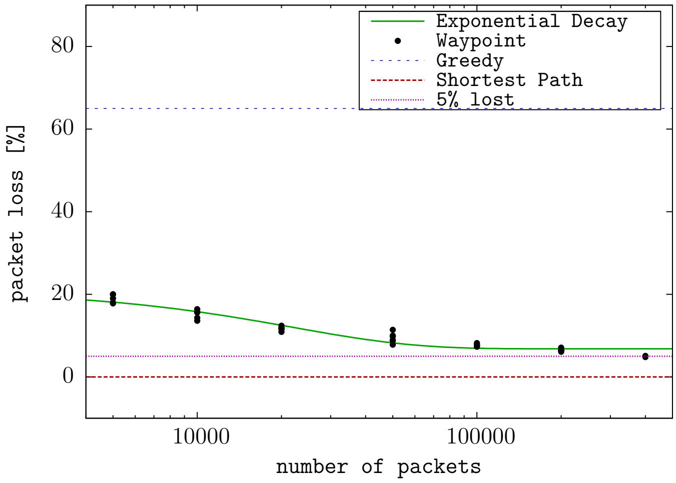

Figure 6.10 – Binary Waypoint Routing, $N = 10,000, R = 0.015$

to $R = 0.015$ for $N = 10000$ nodes and after 10 simulations we obtained the average node degree equal to 10.03 ± 0.07. Figure 6.10 shows that the network under Binary Waypoint Routing tends to the state with a smaller loss rate than in the previous case: it achieves the loss rate of 5%.

6.2.3 Routing in a network with voids

We use a simple model of concave voids (cf., Figure 6.11) for which we can easily modify size and shape by varying its parameters. A concave void placed inside our circular arena results in deleting all mesh routers lying inside the black area.

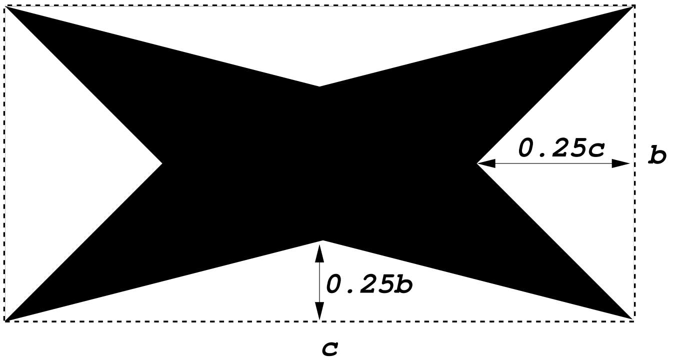

Figure 6.11 – Model of a void

We place a void with parameters $b = 0.4$, $c = 0.4$ in the center of the circular network with parameters $N = 10,000, R = 0.01425$. We have observed in 5 simulations that

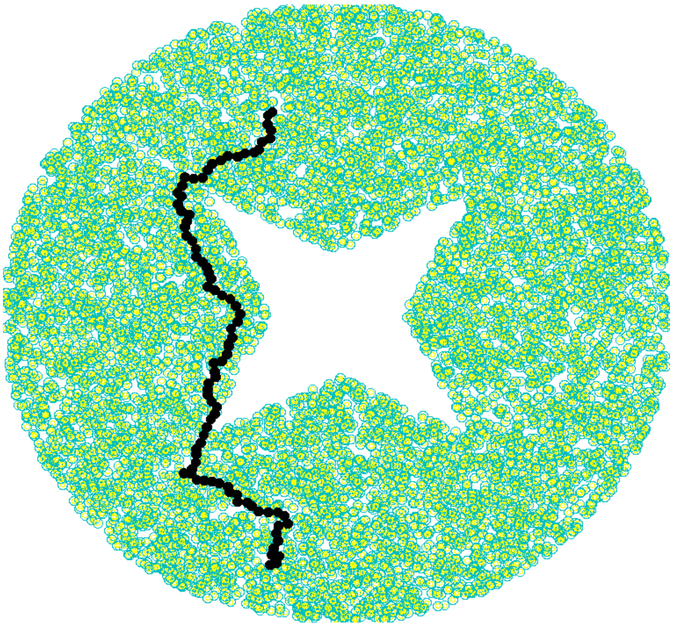

Figure 6.12 – Network with a void, an example route used in Binary Waypoint Routing

the void with the chosen parameters removes $N' = 1093 \pm 29$ nodes on the average. We have generated 400,000 packets per simulation between random pairs of sources and destinations. The packet drop rate p_e for Greedy Routing significantly grows to $p_e = 0.85 \pm 0.01$. Figure 6.13 presents its traffic map that shows packet loss and delivery statistics at nodes for a single simulation: a black bar represents the number of packets dropped by a node whereas a red one represents the number of packets forwarded by the same node. We can see an empty area in the center of the network and large loss bars at the border of the concave region.

We have simulated a mesh network with the same parameters, but operating under Binary Waypoint Routing. After 5 simulations we obtained the packet drop rate of $p_e = 0.11 \pm 0.015$. Figure 6.12 shows an example route used in Binary Waypoint Routing in this network while Figure 6.14 presents the corresponding traffic map. We can see that the border of the void does not drop as many packets as in the greedy case. Note also that the neighborhood of the void does not forward as many packets as under Greedy Routing. We can observe that Binary Waypoint Routing prefers certain paths related to the topology of the void and results in a cobweb-like traffic pattern observed also for the Optimal Shortest Path Routing (cf., Figure 6.15). Moreover, we can see that traffic does not go along the concave face of the void as it is the case for face routing (e.g., GPSR). Rather, the nodes have organized themselves into a more efficient ring-like structure to route around the void. We have also observed networks with voids placed in other regions, not only in the center. The packet loss probability p_e remains constant and does not depend on the placement of the void.

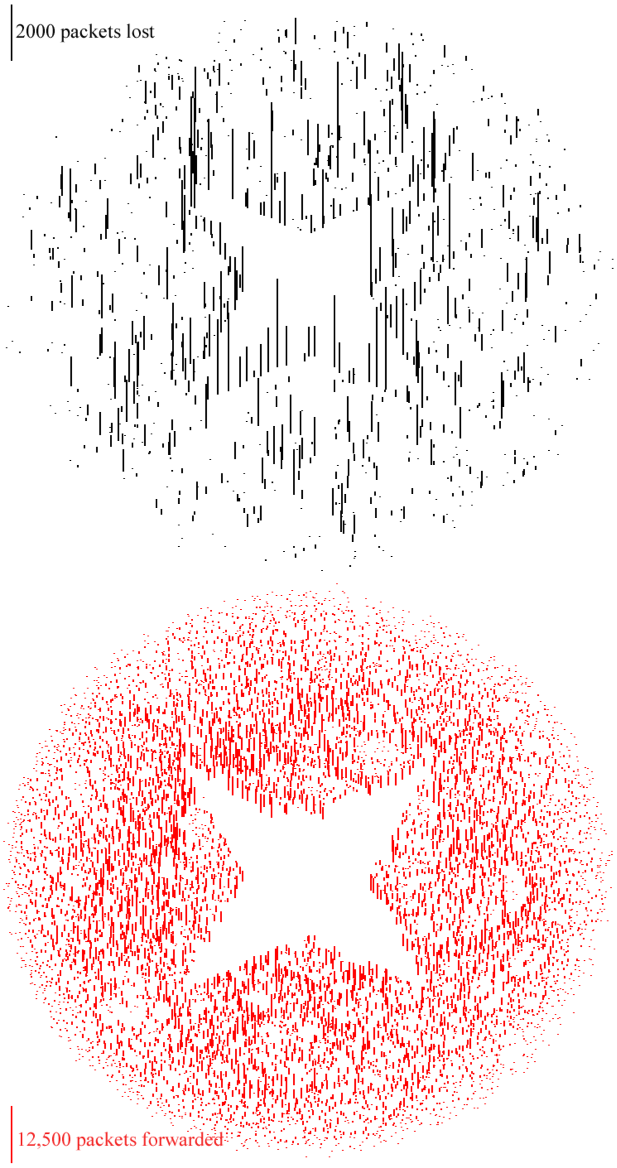

Figure 6.13 – Traffic map for Greedy Routing

Finally, we have generated a more complex network topology with two voids shown in Figure 6.16. The figure shows an example route of a packet under Binary Waypoint Routing. It is important to note that the route is admittedly suboptimal, but it adapts to the mesh topology. Even though we started the learning process of Binary Waypoint Routing with greedy geographical forwarding that prefers straight paths, we have obtained a curved route. The packet loss rate of Binary Waypoint Routing is $p_e = 0.127$ af-

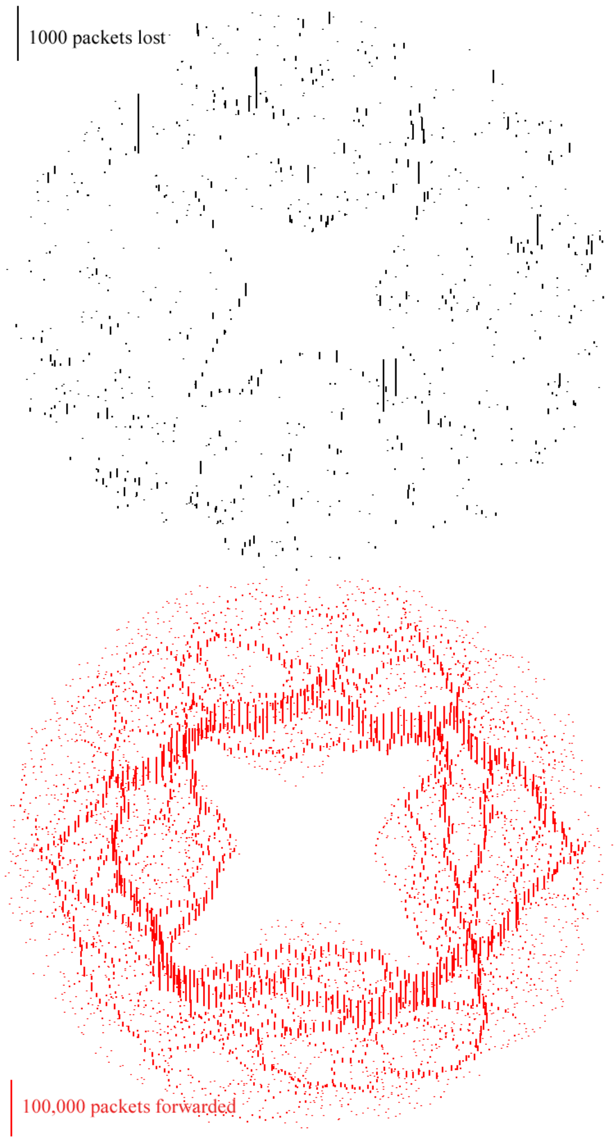

Figure 6.14 – Traffic map for Binary Waypoint Routing

ter 400,000 packets, which is much better than in the greedy case ($p_e = 0.871$). Greedy Geographical Routing simply does not work well in such complex topologies. We obtain the value of $p_e = 6450$ ppm > 0 for the Optimal Shortest Path routing, because some parts of the graph are not connected. The average hop count for Binary Waypoint Routing is 74.5, which is still larger than in the case of the Optimal Shortest Path routing: 58 hops.

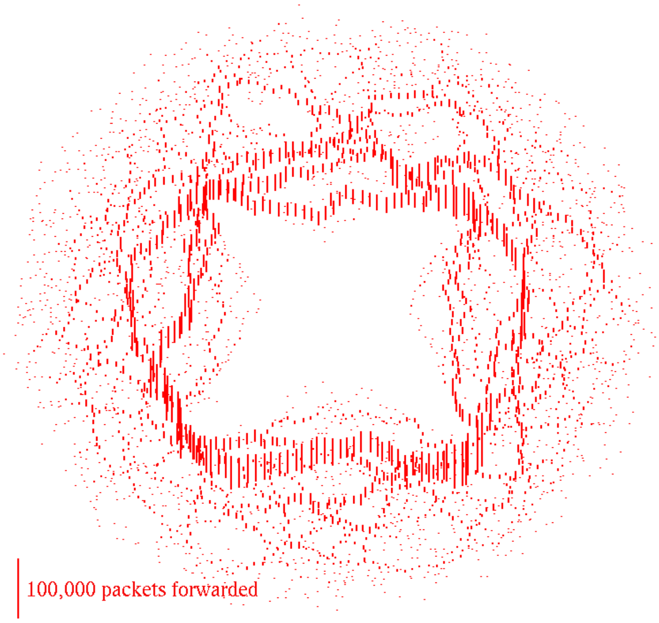

Figure 6.15 – Traffic map for Optimal Shortest Path

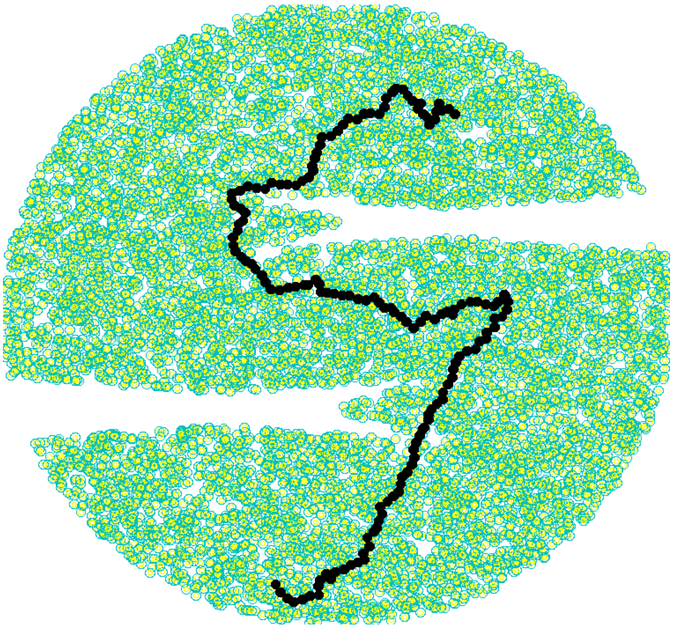

Figure 6.16 – Example route under Binary Waypoint Routing in a complex topology with two voids

6.3 Summary

We have presented Binary Waypoint Routing, a novel geographical routing protocol for wireless mesh networks based on the idea of learning and maintaining source routes to a small number of nodes in disjoint addressing subspaces.

Our simulation results show that the proposed scheme achieves high packet delivery rate with a traffic pattern similar to the Optimal Shortest Path Routing. A higher node degree (the number of neighbors) results in an improved packet delivery rate, so we can achieve desired protocol performance by constructing a network with a required node density. The proposed scheme does not require explicit construction of routing tables and its operation does not depend on any graph property such as UDG, so the protocol works for any type of wireless connectivity between nodes.

7 Scalable Waypoint Routing

In this chapter, we propose a scalable geographical routing protocol that offers high packet delivery rates while keeping the size of routing tables small and slowly growing with the network size. The idea of Scalable Waypoint Routing is to maintain routes to a small number of nodes called waypoints and use them to forward packets to any destination. Our protocol partitions the geographical address space through a quad-tree creation process that yields subspaces of exponentially decreasing size. By observing traces of incoming packets, a node learns routes to waypoints associated with each subspace and uses them to forward packets to all destinations in a given subspace. As each node forwards a packet through a sequence of subspaces of a decreasing size, a packet finally reaches its destination. The volume of routing information in any node remains very small compared to the size of the whole network, because the number of waypoints grows in a logarithmic way with the network size.

In the initial phase of operation, nodes without routing information forward packets in the greedy way while observing traces in received packets to find waypoints for all subspaces–if we receive a packet going through a node in a given subspace, we can choose it as a waypoint. Each intermediate router on the route to a waypoint tries to optimize the path by forwarding a packet to its waypoint, which may be closer to the destination. Unlike the previously proposed scheme [84], our protocol does not require source routing, uses almost optimal routes (route stretch is small), and achieves scalability through the contraction of the addressing subspaces. Moreover, it only requires a constant size of information per packet so we can use IPv6 header extension for prototype implementation.

In this chapter, we present the principles of the protocol and evaluate its performance through simulations by comparing with other approaches (Greedy Routing and Optimal Shortest Path). Our comparisons show high packet delivery rates and the distribution of routes similar to the Optimal Shortest Path. We also test its ability to operate in dynamic wireless mesh networks with high churn.

The highlights of the work are the following:

– we propose a geographic routing scheme that achieves high delivery ratio in large scale dynamic multi-hop wireless networks,

- unlike other proposals, it combines efficiency with scalability–when compared with the optimal routing scheme that requires the total topology knowledge, it results in routes of similar lengths, while only maintaining a small volume of routing information,
- it can operate in real wireless networks, because it does not rely on any stringent assumptions about the radio propagation model, i.e. radio ranges need not to be uniform and cover unit disks.

The rest of the chapter is organized as follows. We introduce the principles of the proposed scheme illustrated by some examples (Section 7.1). Section 7.2 evaluates its performance and behavior in dynamic networks. Finally, we present some conclusions (Section 7.5).

7.1 Scalable Waypoint Routing

We adopt similar assumptions to other geographical routing protocols: we assume that nodes know their coordinates and neighbors can exchange packets. Unlike previous approaches such as face routing, we do not require any unit disk assumptions nor other properties of the underlying network graph (e.g., planar graph). In particular, we assume that each node lies inside a finite square addressing space:

$$\mathcal{A} = [x_l, x_m] \times [y_l, y_m] \tag{7.1}$$

and knows its geographical position $a_i = (x_i, y_i)$, a pair of coordinates such that $x_l \leq x_i \leq x_m$ and $y_l \leq y_i \leq y_m$. In the rest of the chapter, we denote a node by its address a_i.

We start with the observation that packet loss in greedy geographical forwarding only occurs in fairly few places corresponding to topology defects: concave nodes only have neighbors in the backward direction and they appear at the border of concave voids (cf., Chapter 5). The idea of Scalable Waypoint Routing is to learn directions to some remote nodes by observing arriving packets and to keep the information in a routing table. Note that a given node can receive packets from remote nodes on a given direction if there is no void nor obstacle between them. So, we use some chosen waypoints that send us packets for forwarding packets to destinations to which we do not have direct access.

We introduce below the principles of the proposed protocol by first explaining Binary Waypoint Routing and then defining the operation of Scalable Waypoint Routing. We also illustrate its operation with some examples.

7.1.1 Binary Waypoint Routing

Let us first consider the principles of Binary Waypoint Routing [84]. Node a_i partitions addressing space $\mathcal{A}$ into a set of smaller disjoint subspaces $\mathcal{A}_i^k, k = 1, \ldots, n$

through binary space partitioning process $P_{binary}(\mathcal{A}) \rightarrow \{\mathcal{A}_i^{+}, \mathcal{A}_i^{-+}, \ldots, \mathcal{A}_i^{-\cdots-+}\}$: first, it divides $\mathcal{A}$ into two subspaces $\mathcal{A}_i^{+}$ and $\mathcal{A}_i^{-}$ (in a vertical or horizontal way randomly chosen by nodes at each step), then it further divides $\mathcal{A}_i^{-}$ into $\mathcal{A}_i^{-+}$ and $\mathcal{A}_i^{--}$ and so on (see also the explanations of a similar scheme in Section 7.1.2). Note that due to the random choice each node may obtain a different partitioning of the addressing space. The process stops when the smallest subspace is too small for further splitting (for example its area corresponds to the radio coverage of a node). The resulting subspaces $\mathcal{A}_i^{k}$ cover addressing space $\mathcal{A}$, their number n is relatively small and logarithmically grows with the size of the network.

At the beginning, a node uses greedy geographic forwarding to send packets to any destination–it forwards a packet to its closest neighbor in the direction of the destination. Each packet contains trace $r = (a_s, \ldots, a_i)$, a list of all nodes that have forwarded the packet between a_s, the source of the packet, and a_i. Node a_i inspects packet traces and if it finds that node a_s lies inside a given subspace $a_s \in \mathcal{A}_i^{s}$, then a_s becomes the waypoint for subspace $\mathcal{A}_i^{s}$: it stores source route $r^{-1} = (a_i, \ldots, a_s)$ in routing table $\mathcal{R}[\mathcal{A}_i^{s}]$. When the node needs to send a packet to any destination $a_d \in \mathcal{A}_i^{s}$, it will send it using source route $\mathcal{R}[\mathcal{A}_i^{s}]$.

Nodes may receive packets from different nodes belonging to the same subspace. They choose the best routes according to the following metric:

$$H = \begin{cases} \frac{|a_i, a_s|}{\text{Hops}[r]} & \text{Hops}[r] \geq 0 \\ 0 & \text{Hops}[r] = 0 \end{cases}, \tag{7.2}$$

where $|a_i, a_j|$ is the Euclidean distance between two nodes.

The main drawback of Binary Waypoint Routing is its dependence on packet traces and sources routes. The size of traces may rapidly grow, especially in large scale wireless mesh networks so that their overhead may become too large for a given packet size. Moreover, traces are of variable size, so it is difficult to insert them in a fixed size packet header. Packet forwarding based on source routes also suffers from the same problems.

7.1.2 Scalable Waypoint Routing

In Scalable Waypoint Routing, we keep the principle of learning waypoints for destination address subspaces from packet traces, however we define a different partitioning scheme. Moreover, instead of an unlimited size packet traces and source routes, we limit the size of packet traces and we do not use source routing, more precisely, we keep the information on partial routes limited to a fixed small length (typically up to five hops). In fact, we split the information about a given route between a node and a waypoint into short parts kept by different nodes on the path between them. Forwarding based on this information results in the reconstruction of the whole route from pieces of a limited size.

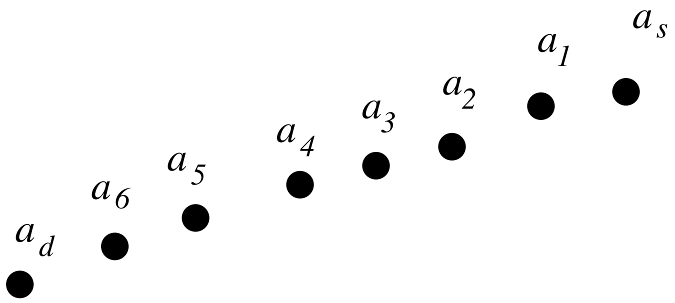

Figure 7.1 – Packet trace example

Learning partial routes

We start the description of Scalable Waypoint Routing by explaining how to limit the size of packet traces and only work with fixed size routing information. Each packet contains partial trace r limited to last h_l hops and a hop counter h. h_l is a protocol parameter set to a small value (e.g., it varies from 1 to 5 in our simulations). The partial trace contains the information about up to last h_l nodes that have forwarded the packet. Note that when $h_l = 1$, the protocol only keeps the information about the next hop. Figure 7.1 presents an example of a route from source a_s to destination a_d going through six intermediate nodes $a_i,\ i = 1, \ldots, 6$. Assume that $h_l = 3$. The partial trace at a_1 is (a_s), (a_s, a_1, a_2) at a_3, and (a_1, a_2, a_3) at a_4. Note that node a_3 has deleted a_s from the trace and added itself, because its size cannot exceed the limit h_l. The source node initializes hop counter h with 0 and each intermediate node increments it, so it becomes 7 at the destination.

To learn routes to waypoints, each node inspects packets and fills its routing table with the best information it can find in the partial trace. Consider the example in Figure 7.2: when node a_i receives a packet from a_s to forward to destination a_d while its routing table for the subspace of a_s is empty, it creates w, a waypoint routing entry containing three fields: waypoint address a_w that becomes equal to a_s, waypoint metric $H_w = |a_i,\ a_w|\,/h$, and the partial route $r_w = r^{-1}$, where r is the inverse of the partial trace (e.g., $(a_1, a_2, a_3)^{-1} = (a_3, a_2, a_1)$), the hop counter h coming from the packet.

A node stores several waypoint entries $W = \{w_1, \ldots, w_k\}$ for each subspace. In this way, it optimizes the choice of a route–it can choose waypoint routing entry $w_j \in W$ with a_{w_j} closest to the destination: $\forall w_i \neq w_j \in W : \ |a_{w_i}, a_d| \geq \left|a_{w_j}, a_d\right|$.

Packet forwarding consists of inserting the best waypoint routing entry into a packet (we reserve a fixed size field in the packet header for this purpose) and sending it to the next hop defined in the partial route r_{w_j}. The waypoint routing entry in the packet stays valid up to h_l hops. The next forwarding node may have a different set of waypoint entries W' and applies the same procedure with the difference that it chooses the best waypoint from its internal set W' and the waypoint routing entry in the packet, if it is still valid. Then, the node inserts the new best waypoint entry in the packet and forwards it to the next hop defined by the waypoint entry.

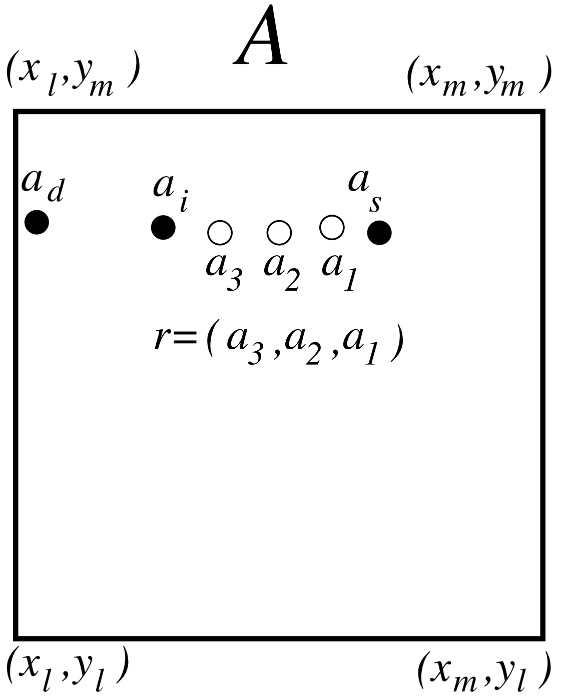

Figure 7.2 – Scalable Waypoint Routing learning process

Partitioning addressing space

We define address partitioning according to a quad-tree data structure that fits our objectives better that binary partitioning, because the latter privileges some directions (the first vertical or horizontal splitting yields a large subspace oriented either vertically or horizontally). This scheme enables route optimization in Binary Waypoint Routing when nodes use long source routes. As nodes may partition the addressing space in different ways, a packet has more chances to go through a node with a different partitioning and benefit from a waypoint closer to the destination. In the case of Scalable Waypoint Routing, nodes only know about short partial routes and operates the best with the same partitioning of the space at least for the farthest and largest subspaces.

At the beginning, each node a_i discovers its neighborhood–all directly reachable neighbors and estimates the neighborhood diameter d_l defined as the double geographical distance to the farthest neighbor:

$$d_l = \begin{cases} 2\max_{a_j} |a_i, a_j| & a_j \in \text{Neighborhood}\,[a_i] \\ \text{Edge}\,[\mathcal{A}] & \text{Neighborhood}\,[a_i] = \emptyset \end{cases} \tag{7.3}$$

or the size of the addressing space $\mathcal{A}$ if the neighborhood is empty (the node does not have any neighbor).

Each node a_i applies a quad-tree decomposition process $P_{quad-tree}$ to partition the addressing space $\mathcal{A}$ into a set of disjoint subspaces $\mathcal{S}_i^j$:

$$P_{quad-tree}(\mathcal{A}) \rightarrow \begin{pmatrix} \mathcal{S}_i^1, \mathcal{S}_i^2, \mathcal{S}_i^3 \\ \mathcal{S}_i^{01}, \mathcal{S}_i^{02}, \mathcal{S}_i^{03} \\ \vdots \\ \mathcal{S}_i^{0\ldots0} \end{pmatrix} \tag{7.4}$$

that cover the whole addressing space: $\bigcup_j \mathcal{S}_i^j = \mathcal{A}$.

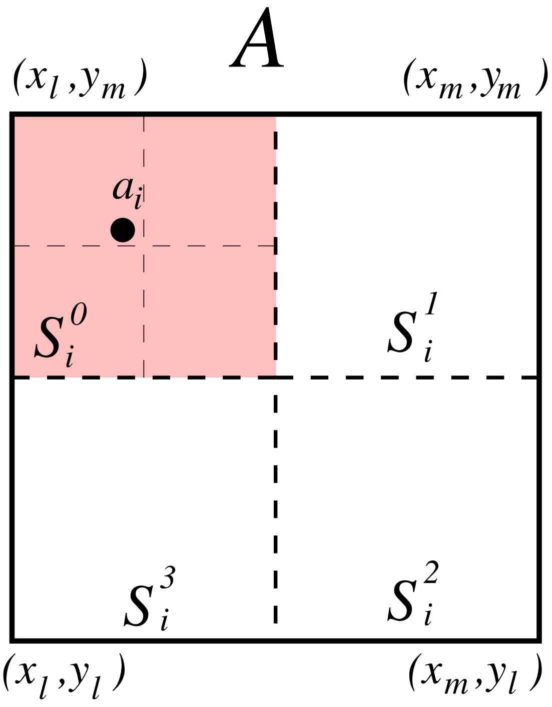

Figure 7.3 – Quad-tree address space partitioning

Figure 7.3 illustrates the first step of the partitioning process in which node a_i partitions $\mathcal{A}$ into four subspaces: $\mathcal{S}_i^0$ that contains node a_i and three other subspaces $\mathcal{S}_i^1, \mathcal{S}_i^2$, and $\mathcal{S}_i^3$. Next, the node repeats partitioning of S_i^0, if Edge $\left[S_i^0\right] > d_l$, which results in $\mathcal{S}_i^{00}$ and $\mathcal{S}_i^{01}, \mathcal{S}_i^{02}, \mathcal{S}_i^{03}$, and so on. The process continues until Edge $\left[\mathcal{S}_i^{0\ldots0}\right] \leq d_l$.

Packet forwarding

Let us consider packet forwarding in Scalable Waypoint Routing. We describe the operation of node a_i on the path between source a_s and destination a_d. The packet sent by source a_s arrives after some hops in a_i (cf., first part of Figure 7.4, we assume $h_l = 2$ hops in this example). Let us assume that the waypoint entry in the packet is empty. Node a_i first identifies the subspace that contains the destination address: $a_d \in \mathcal{S}_i^1$ and

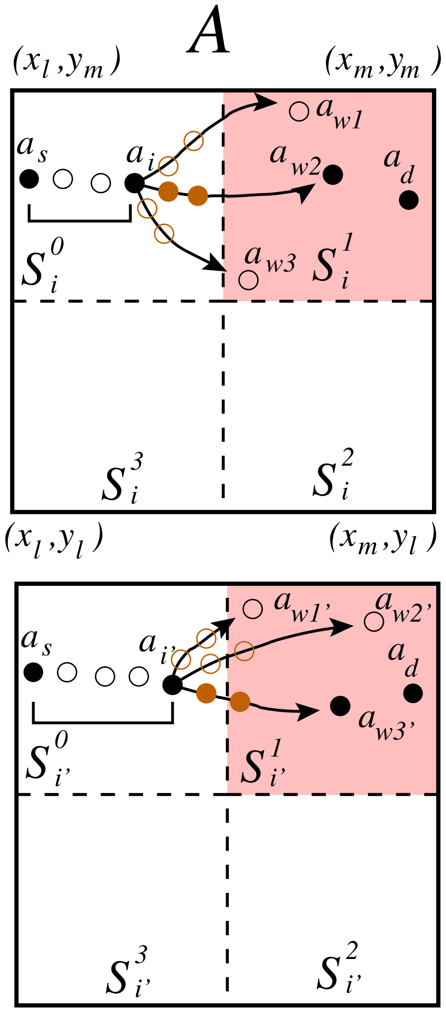

Figure 7.4 – Scalable Waypoint packet forwarding

the set of waypoint entries associated with $\mathcal{S}_i^1$: w_1, w_2, w_3 at locations $a_{w_1}, a_{w_2}, a_{w_3} \in S_i^1$. Node a_i can choose between three different partial routes $r_{w_1}, r_{w_2}, r_{w_3}$ towards three waypoints $a_{w_1}, a_{w_2}, a_{w_3}$, respectively. Assume that the node chooses waypoint entry w_2, because a_{w_2} is the closest to the destination: $\forall w_i \neq w_2 : |a_d, a_{w_i}| \geq |a_d, a_{w_2}|$, so it inserts waypoint entry w_2 into the packet and forwards it to $a_{i'}$, the next hop in r_{w_2}. The second part of Figure 7.4 shows what happens next at node $a_{i'}$ that has a different set of waypoint entries corresponding to the same subspace (note that both nodes a_i and $a_{i'}$ have the same partitioning of the addressing space). It chooses w_3', which is the best one out of its internal waypoint entries and the one from the packet. In a similar way as previously, it forwards the packet to the next hop defined by the best waypoint entry.

We can observe that even if each node only knows some partial information about paths–partial routes to known waypoints, nodes construct the whole path between a source and a destination. Note also that each node keeps the waypoint entries having the largest value of H_w metric. As the whole path $(a_s, \dots a_d)$ is a concatenation of small pieces (partial routes), the waypoint metric computed at the destination a_d : $H_w = |a_s, a_d|/h$ is also large. This means that the resulting path is close to the shortest one– the protocol constructs it without taking advantage of any global information, a graph structure, or a graph optimization algorithm.

Number of waypoints

We now consider the problem of choosing the right number of waypoint entries kept by a node. It is better to have more waypoints, but we also want to limit the volume of the information stored at nodes to achieve scalability. We propose to keep at most $L_i[\mathcal{S}_i^w] = \mathcal{L}$, $a_w \in \mathcal{S}_i^w$, where $\mathcal{L}$ is constant, best waypoint entries (best with respect to metric H_w) per subspace. If a node has several entries for the same waypoint, it will only keep the best one.

Waypoint metric H_w is an important parameter: we want to keep a set of waypoint entries with the largest H_w, because in this case packets cross long distances per hop count. Note that the shortest path between a_s and a_d computed by OSPF will have the maximal value of H_w for this pair of nodes.

Refreshing waypoints

The last issue addressed by our protocol is the problem of dynamic adaptation to changing state of the network. In a large-scale network, we need to deal with a substantial part of nodes that may disappear and return back. In a spontaneous mesh network based on user access points for instance, there will be a high variation of the number of active nodes between days and nights, because some nodes can be switched off.

Our principle is to refresh waypoint entries after some time to learn new information about the current network state. To achieve this, each node a_i has a counter c_i initially set to 0 and incremented for each received packet. A node also maintains a shadow waypoint table for each subspace with the same size as the waypoint routing table that contains recently seen waypoint entries.

A node stamps waypoint entries in the routing and shadow tables with the value of counter c_i. We define $\bar{c}_i[\mathcal{S}_i^j]$ as the average counter value of waypoint entries in the shadow table. In order to refresh the routing table, the node deletes obsolete waypoint entries that fulfill the following condition:

$$c_i - c \geq C_L(c_i - \bar{c}_i[\mathcal{S}_i^j]), \tag{7.5}$$

where c is the counter value associated with the waypoint $a_w \in \mathcal{S}_i^j$ and C_L is a parameter. Roughly speaking, when the waypoint entry lifetime (counted in the number of received packets) is C_L times longer than the average lifetime of a waypoint entry to a subspace, the node deletes the entry from its routing tables.

7.2 Simulations

We have implemented a simulator of large-scale wireless mesh networks able to handle up to 40,000 nodes (available network simulators such as NS are limited to several tens of nodes). The simulator focuses on packet forwarding and routing, and for this first evaluation of the proposed protocol, we assume that MAC and PHY layers are ideal (no frame loss). We are aware of this limitation, but its purpose is to validate the routing principles and compare them with other protocols in a controlled environment on a large-scale.

In our simulations, we model a wireless mesh network as a random geometric graph and we adopt the Unit Disc Graph assumptions to construct the connectivity graph: each node has the same constant radio range R and a circular propagation area–if two nodes are in range $|a_i, a_j| < R$, we consider that there is a stable link between them and nodes become neighbors. Note that the proposed protocol does not rely on these assumptions, the only information needed by a node being the list of its neighbors. In realistic wireless conditions, a node can maintain a threshold of channel quality to decide with which node communication benefits from sufficient quality to consider it as a neighbor.

We obtain the connectivity graph by drawing the coordinates of N nodes from the uniform distribution in the square arena $A = [0, 1] \times [0, 1]$. Nodes send a large number of packets between randomly chosen source and destination pairs. We choose sufficient network density to obtain good connectivity–the network operate beyond the percolation threshold so the network is almost connected (there are only few disconnected nodes). Our protocol has three parameters and we consider the following values: $h_l = 1, 3$, $\mathcal{L} = 5$, and $C_L = 4$.

If relevant, we compare the proposed protocol with Greedy Forwarding and the Optimal Shortest Path protocol, but not with the Binary Waypoint Routing. Performance results for both Waypoint Routing protocols are similar, however Scalable Waypoint Routing does not operate according to the principle of source routing.

7.2.1 Stationary network, $N = 10,000$ nodes

In the first experiment, we consider a stationary network (no topology modification) with $N = 10,000$ nodes and radio range of $R = 0.015$. The resulting network connectivity graph has the average node degree of $D = 6.97$. We can see in Figure 7.5 that packet loss rate quickly decreases with the number of packets sent as nodes progressively acquire more information about routes in the network. After some time, packet loss rate does not improve and stays almost constant. Sending more packets at the beginning improves network reachability, because nodes learn more information about routes. Then, after the initial learning phase, packet loss rate mainly depends on network defects. We can also observe the influence of h_l parameter, the size of partial routes stored at nodes. For $h_l = 5$, the protocol decreases packet loss rate to around 10% while for $h_l = 1$, which corresponds to keeping the information about only the

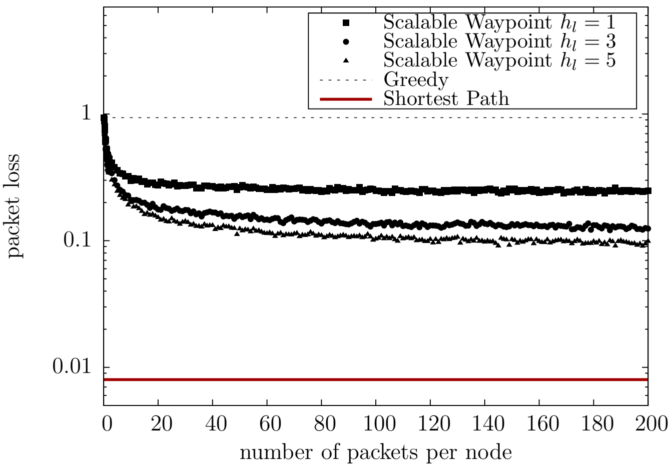

Figure 7.5 – Packet loss rate, $N = 10,000, D = 6.97, h_l = 1, 3, \mathcal{L} = 5$

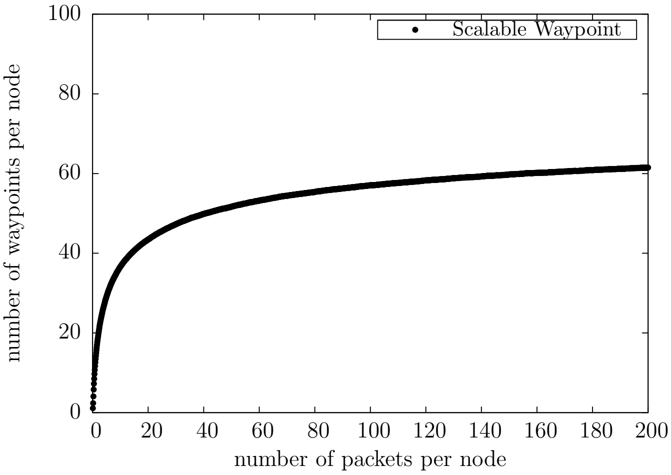

Figure 7.6 – Number of waypoints, $N = 10,000, D = 6.97, h_l = 3, \mathcal{L} = 5$

next hop, packet loss rate is higher, around 25%. The network is not fully connected, because there is non zero packet loss rate in the case of the Optimal Shortest Path protocol. Compared with Greedy Forwarding with over 90% packet loss rate, our protocol achieves much better performance.

Figure 7.6 presents the number of waypoint entries per node. It increases during the initial learning phase and stabilize at the level of 60, which is small compared to the number of nodes.

Figure 7.7 presents the distribution of route length–probability that variable h has a given value. The average route length is around 60 for our protocol and we can observe that the Optimal Shortest Path protocol achieves a quite similar distribution. This result shows that Scalable Waypoint Routing converges to almost optimal routes.

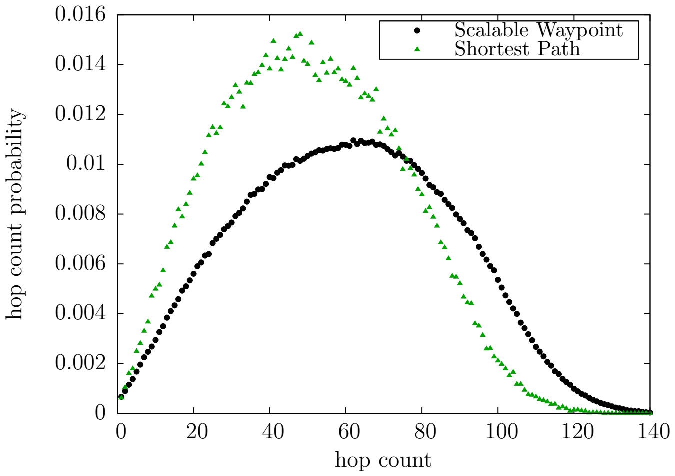

Figure 7.7 – Distribution of route length, $N = 10,000, D = 6.97, h_l = 3, \mathcal{L} = 5$

7.2.2 Stationary network, variable number of nodes N

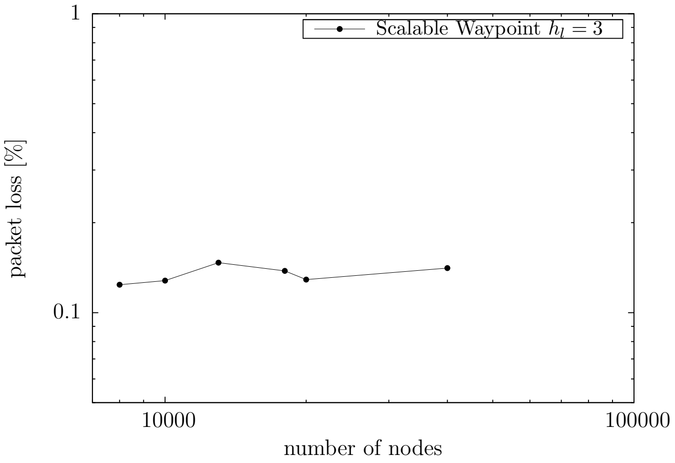

Figure 7.8 – Packet loss rate, variable $N,\ D = 6.97,\ h_l = 3,\ 100$ packets sent by node

Next, to observe protocol scalability we have simulated a network with different sizes while keeping the same density–the number of nodes increases from $N = 8,000$ to $N = 40,000$. Figure 7.8 shows the packet loss rate in the network after sending 100 packets on the average. We can see that the probability of packet delivery is almost the same and it does not depend on the size of the network.

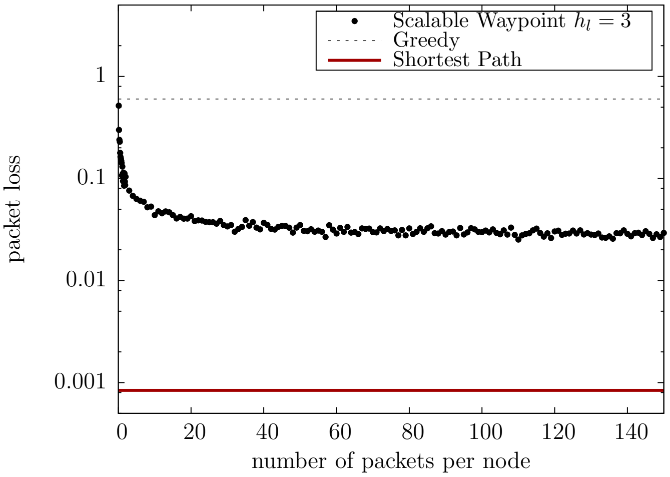

Figure 7.9 – Packet loss rate, $N = 10,000, D = 9.99, h_l = 3, \mathcal{L} = 5$

7.2.3 Stationary network, $N = 10,000$ nodes, higher density

Another experiment tests the behavior of the network for higher density. Network connectivity improves so that the Optimal Shortest Path protocol does not experience any loss. We can see in Figure 7.9 that for our protocol, the stationary packet loss rate is also much better and attains a value around 3%.

7.2.4 Network with voids, $N = 10,000$ nodes

To test our protocol in a network with obstacles we have simulated the network with the addressing space presented in Figure 7.10. Figures 7.11, 7.12, and 7.13 show the corresponding results for this network. We can see that packet loss rate is almost the same with a slightly higher stationary rate of 13% while the number of stored waypoint entries is also similar. The distribution of route length presents an anomaly due to the obstacle, but we can still observe almost the same route lengths for our protocol and the Optimal Shortest Path.

7.2.5 Dynamic network, varying number nodes

Finally, we have simulated a network with high churn of nodes. At the beginning, we have placed $N = 10,000$ nodes with $R = 0.018$, which results in a network with the average node degree of $D = 10.01$. Then, we have run the simulation for some time so that nodes have sent 100 packets on the average and we have removed 15% of randomly selected nodes. After that, nodes have generated additional 200 packets on the average and we have added nodes to obtain the initial number of nodes. Finally, nodes have sent additional 100 packets on the average. We can observe very good behavior of the

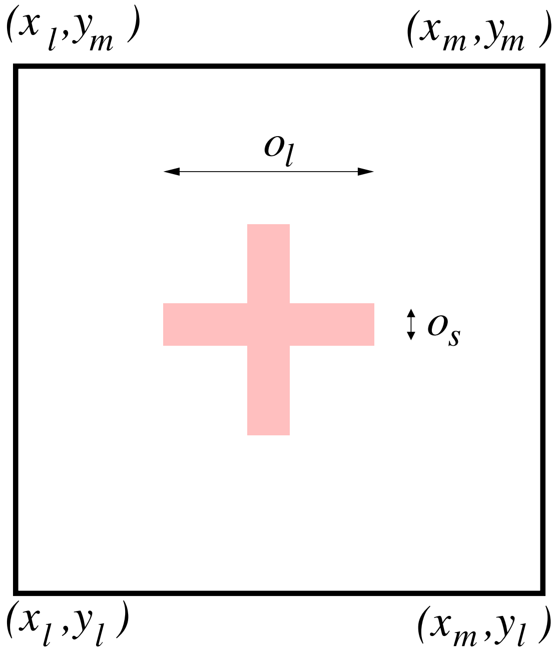

Figure 7.10 – Network with an obstacle, $o_s = 0.1, o_l = 0.4$

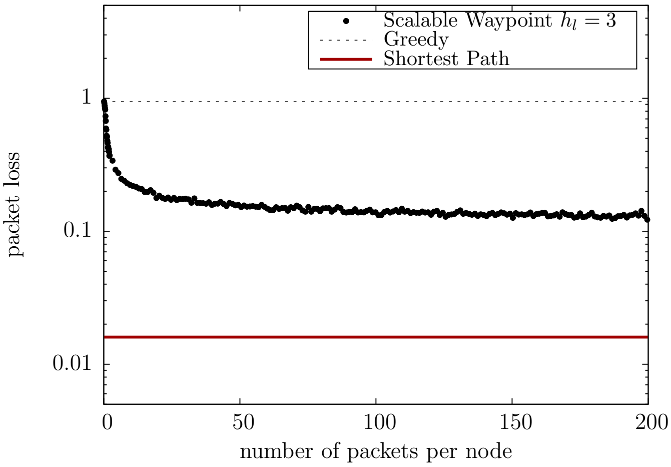

Figure 7.11 – Packet loss rate, network with an obstacle, $N = 9316, D = 6.97, h_l = 3, \mathcal{L} = 5$

protocol: after the departure of nodes, the packet loss rate quickly decreases to attain the low value. Adding nodes just improves performance–loss rate further decreases.

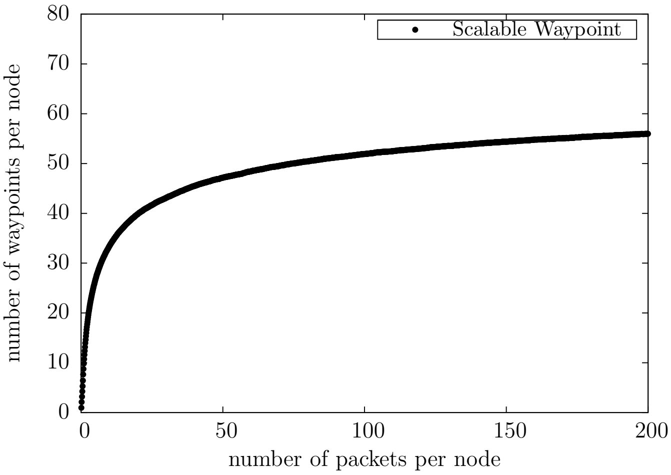

Figure 7.12 – Number of waypoints, network with an obstacle, $N = 9316, D = 6.97, h_l = 3, \mathcal{L} = 5$

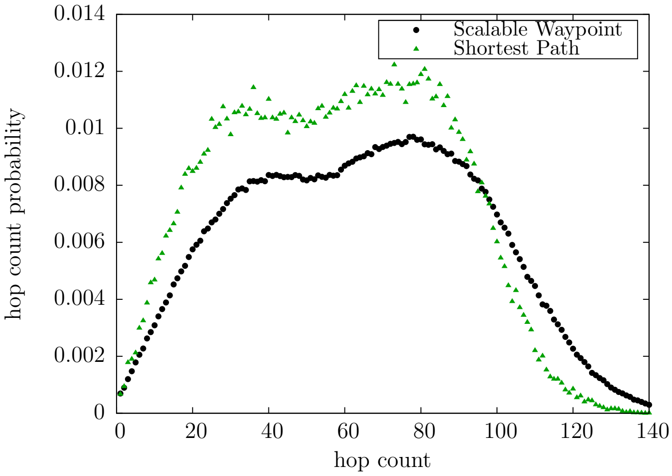

Figure 7.13 – Distribution of route length, network with an obstacle, $N = 9316, D = 6.97, h_l = 3, \mathcal{L} = 5$

7.3 Scalable Waypoint Routing with Guaranteed Delivery

It is not acceptable that a routing protocol drops a large number of packets. We showed that our algorithm behaves better than the greedy geographical forwarding, but its performance is still far away from the OSPF, dropping, e.g., 10% of packets in some situations (cf., Figure 7.5). In order to guarantee packet delivery, we integrate our Scalable Waypoint Routing with other solutions: GPSR [25] and GOAFR+ [35] described in Chapter 3. These algorithms contain two phases: greedy geographical routing and

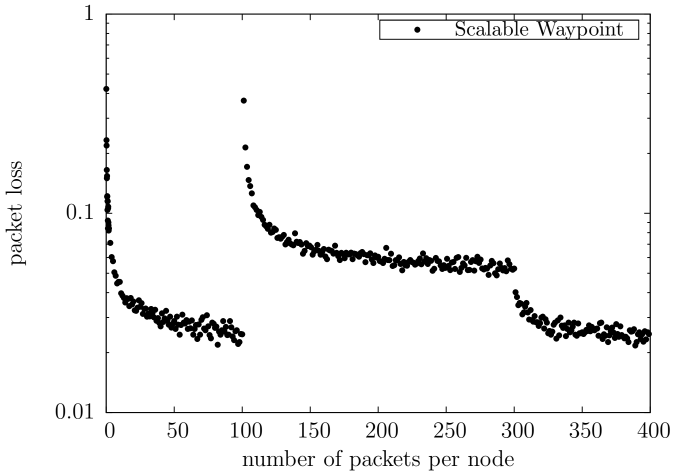

Figure 7.14 – Packet loss rate in a dynamic network, $N = 10,000, D = 10.01, h_l = 3, \mathcal{L} = 5$

the face routing. Greedy geographical forwarding frequently fails in the networks with low densities as we showed in Chapter 5, so obviously algorithms spend a lot of time in the face routing phase. Face routing also suffers from many problems: it may fail in real topologies and it guarantees poor efficiency, finding much longer paths than OSPF. Face routing should be used as a backup and not as a mainstream routing algorithm contrary to the situation in the low density networks. Our Scalable Waypoint Algorithm fails less frequently than the greedy geographical forwarding, so the hybrid of Scalable Waypoint Routing and GPSR or GOAFR+ should ameliorate the performance of these algorithms. We propose a very simple design of the hybrids. In the greedy phase, the hybrid uses Scalable Waypoint Routing to compute a next hop. If Scalable Waypoint Routing fails and it can not deliver a valid next hop, algorithm switches back to the greedy geographical forwarding. If the forwarding fails, the algorithm uses face routing. We use only limited traces, so we should avoid looped paths in waypoint learning. To guarantee loop-free operation a node uses the closest waypoint entry only if the distance between the waypoint and the destination point is smaller then the minimal distance between the forwarding node and the destination seen so far. Nodes do not use packets forwarded by the face routing nor along the hop sequence a_i, a_j, a_i (forward, backward) for learning waypoints.

7.4 Evaluation in NS2

In this section, we evaluate the proposed protocols through simulations and study their communication costs. We use ns-2 to run simulations [79]. We randomly place N nodes in a square of size LxL m^2 according to the uniform distribution. We use the Two Ray Ground propagation model for the radio communication range and we assume the IEEE 802.11b PHY-MAC layers. We set the transmitter power to obtain different average network node degree. We study the behavior of GPSR and GOAFR+ with $\rho = \sqrt{2}$, $\rho_0 = 1.4$, $\sigma = 0.15$ and their hybrid counterparts. We measure the route stretch as the average hop count on the route computed by the geographical routing protocol divided by the average route length computed by AODV. We set $N = 400$ nodes, and $L = 400$ m, we randomly choose pairs of a source and a destination. We send on average 25 packets per node. In Figure 7.15 we see the route stretch evaluated through the simulation for GPSR, GOAFR+, the hybrid of Scalable Waypoint and GPSR (SW+GPSR), and the hybrid of Scalable Waypoint and GOAFR+ (SW+GOAFR+). The hybrids offer much better performance than their counterparts.

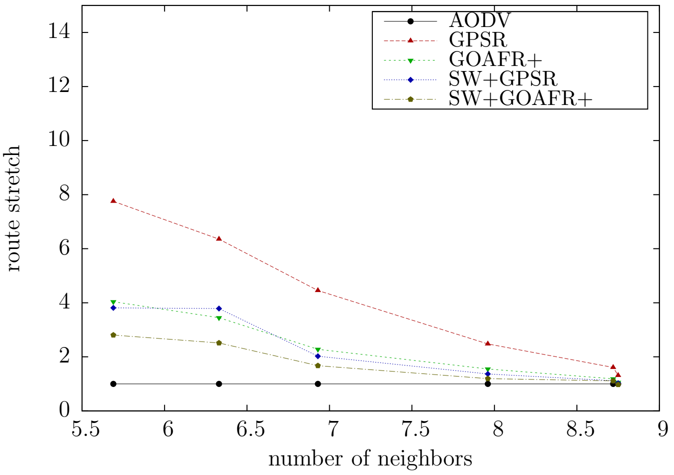

Figure 7.15 – Route stretch, $h_l = 5, \mathcal{L} = 5$, general comparison

7.5 Summary

We have presented Scalable Waypoint Routing, a geographical routing protocol for large-scale dynamic multi-hop wireless networks. Its main idea is to maintain routes to a small number of waypoints and use them to forward packets to any destination. A node partitions the geographical address space through a quad-tree creation process that yields subspaces of exponentially decreasing size and learns directions to waypoints for subspaces from partial packet traces. Unlike similar previously proposed schemes, our protocol does not require source routing, yet the volume of routing information in any node remains very small compared to the size of the whole network, because the number of waypoints grows in a logarithmic way with the network size. Nodes limit the volume of the trace information and partial routes to a configurable fixed size. In this way, the protocol achieves good scalability.

We have run simulation experiments under simple Unit Disc Graph assumptions to validate the principles of the protocol and compare it with the Optimal Shortest Path Routing. Our results show that the proposed protocol achieves high packet delivery rate even for large-scale networks with 40,000 nodes and for networks with high churn. Interestingly, the distribution of the route length is similar to the Optimal Shortest Path Routing, which shows that our scheme achieves almost optimal routes.

8 Conclusions

Spontaneous wireless mesh networks promise a completely different architecture of the future Internet in comparison with its current state. There are many problems that have to be solved prior to the network deployment. Classical routing algorithms should be replaced with suitable technologies that guarantee good scalability and offer robust connectivity. We considered geographical routing algorithms, because they do not require complete global topology of the network to compute routes and thus they are much more scalable then the topological algorithms. There are, however, many problems we need to solve. The most important one is that geographical algorithms have poor performance. In some situations they compute much longer paths than the topological routing algorithms. In order to solve that problem in Chapter 5, we studied the behavior of the greedy geographical forwarding, the basic building block of geographical algorithms. We found that that the poor performance of geographical algorithms emerges, because the greedy geographical forwarding does not have any information on the global topology and thus it does not correctly go along the network connectivity graph. The greedy geographical forwarding frequently fails to deliver packets, so the highly inefficient face routing should be used. In order to solve that problem, we specified Binary Waypoint Routing and Scalable Waypoint Routing protocols. These protocols do not require any additional overhead to discover network topology, but instead they passively analyze traffic. What is amazing, we do not gather pure topological information on the graph, but instead the limited geographical information that helps to forward packets taking into account the network topological properties. In Chapter 6 and Chapter 7, we present an extensive study on our schemes analyzed through simulation methods. In order to find the basic properties of our algorithms, we developed a Monte-Carlo simulator able to simulate networks up to 40’000 nodes. We found very interesting properties of our algorithms, so we proposed the hybrids of Scalable Waypoint Routing and GPSR or GOAFR+ and we studied their behavior in smaller situations using the ns-2 simulator. According to our simulations, these hybrids have much better performance than the original algorithms, because they compute much shorter paths. Binary Waypoint Routing and Scalable Waypoint Routing can be easily extended to three dimensions, so it is one of possible directions to be studied in the future work. Our algorithms can be also interesting from the Wireless Sensor Networks point of view.

Geographical routing protocols use node positions to forward packets. In order to obtain node coordinates some localization methods should be used. Nodes may use GPS or other radio location techniques to obtain geographical coordinates. This is also an important issue, because localization has an impact on the geographical forwarding, so the topological conditions to localize a node should be studied. In Chapter 4, we assumed that we perform a Network-Wide Localization using radio location techniques starting from a small reference group that contains a few nodes with known positions. We found that the radio localization requires a high network connectivity with more than 11 neighbors on average to address an important number of nodes. Operating in the networks with densities below 11 may prevent from using Network-Wide Localization and the geographical forwarding of packets may not be possible. We did not study the accuracy of the localization techniques, but it also has a big impact on the behavior of geographical routing protocols, so this issue should be taken in account in the future work.

Deuxième partie

Résumé français

1 Introduction

Avec les récents progrès en électronique, télécommunications et informatique, les appareils équipés d'interfaces sans fil sont devenues monnaie courante. Ils ont un prix sans cesse plus bas et sont accessibles dans le monde entier. De nos jours, beaucoup possèdent téléphone, smartphone, ordinateur portable et point d'accès. Pourtant, l'accès à l'Internet est toujours archaïque.

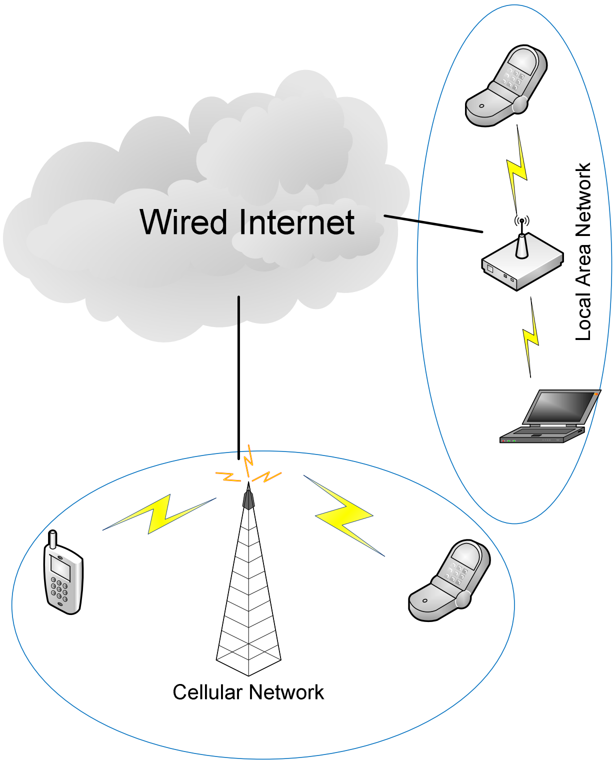

FIGURE 1.1 – La situation actuelle de l'Internet

Nous montrons sur la figure 1.1 que le point d'accès dans le réseau local est une sorte d'interface entre la station sans fil et l'Internet filaire. Les équipements sans fil deviennent plus robustes et puissants. Ils sont capables de fonctionner en relais sans fil et de remplacer l'Internet filaire dans certaines situations. Les solutions actuelles offrent une grande accessibilité (par exemple la téléphonie cellulaire avec une couverture d'environ 98%). Pourtant, ils rencontrent de nombreux problèmes tels que des chemins non optimaux, l'agrégation du trafic et une charge élevée. Considérons le réseau des points d'accès (AP). Les clients mobiles peuvent joindre et quitter le réseau dans lequel ils échangent l'information via l'infrastructure.

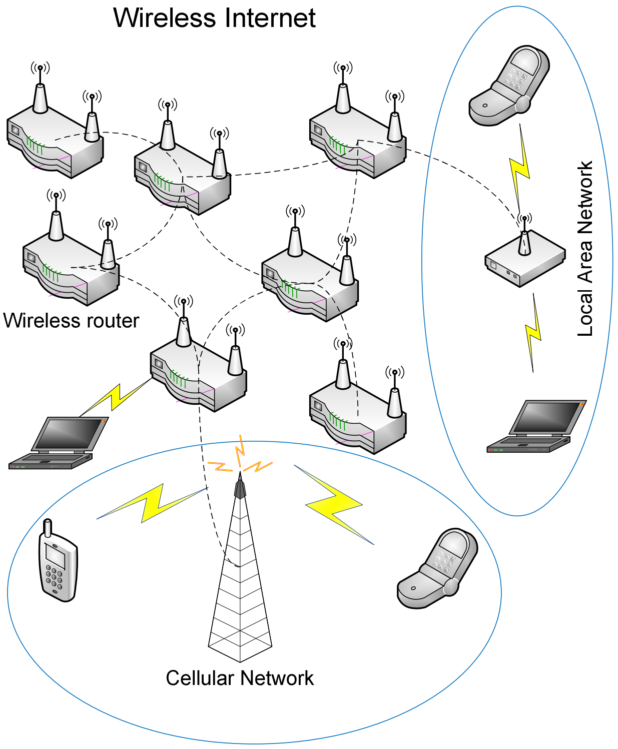

FIGURE 1.2 – La situation dans l'Internet à l'avenir

Dans cette thèse nous examinerons les possibilités de coopération entre des stations sans fil afin de créer l'Internet sans fil comme illustré sur la figure 1.2. Les appareils sans fil sont complètement différents des équipements filaire. Ils sont mobiles, possèdent une quantité d'énergie limitée et les débits sont plus faibles. Nous devons redéfinir les standards et les protocoles actuels afin d'obtenir une solution ayant des besoins de configuration minimale, utilisant des algorithmes simples et offrant une grande accessibilité.

Nous étudions les réseaux sans fil multi-sauts spontanés qui commencent à apparaître, et fournir une connectivité à couverture étendue aux nœuds mobiles. Une de leurs caractéristiques est l'auto-formation — il suffit d'ajouter un autre routeur dans un endroit afin d'augmenter la couverture. L'organisation des réseaux doit être très autonome parce que contrairement à l'Internet actuel, ces réseaux sont auto-gérés : ils ne peuvent pas être gérés par du personnel qualifié pour configurer, connecter et lancer les routeurs.

Nous supposons que les réseaux sans fil multi-sauts peuvent être composées d'un grand nombre de routeurs offrant une connectivité aux nœuds mobiles. Les routeurs bénéficient de ressources abondantes (mémoire, énergie, puissance de calcul et même GPS dans certains cas) et peuvent se déplacer de temps en temps. Ils peuvent utiliser plusieurs interfaces sans fil et différents types d'antennes afin de fournir une capacité suffisante pour les stations mobiles. Les routeurs voisins communiquent directement par les liens sans fil. La communication peut exiger que les nœuds intermédiaires retransmettent les paquets en multi-saut afin de fournir une communication bout en bout entre chaque source et destination du réseau.

Nous voudrions connecter les appareils des utilisateurs sans aucune structure supplémentaire ni fournisseurs d'accès et créer un réseau sans aucune autorité de contrôle. Nous nous concentrons sur la couche réseau et les réseaux décentralisées qui garantissent une bonne connectivité.

Nous commençons par la définition des éléments principaux d'architecture d'un réseau. Comme dans nombreuses contributions fondamentales dans ce domaine [1, 2, 3, 4, 5], nous distinguons les notions suivantes :

nœud — une entité capable de communiquer et de calculer. Il peut être mobile comme un terminal client ou stable comme un routeur.

point terminal — une entité logique correspondant à un nœud unique.

identifiant de point terminal (EID) — un court identifiant binaire d'un point terminal. Il est persistant (il ne change pas quand le nœud change de position).

nom — un identifiant unique, mémorisable, qui est associé à un EID.

adresse — un localisateur dans l'espace d'adressage. Un identifiant court binaire utilisé pour localiser un point terminal dans le réseau pour que les routeurs puissent lui transmettre les paquets. L'adresse peut changer quand le nœud change de position et d'interface.

La figure 1.3 présente l'architecture du réseau multi-sauts spontané sans fil. Les routeurs forment une infrastructure d'interconnexion pour les nœuds mobiles qui se déplacent à l'intérieur et se voient offrir la connectivité par routeurs les plus proches. Le réseau sans fil gère les services de base : le service de noms et le service de localisation. Le service de nom fournit l'associaton entre les noms et les EID des nœuds, comme dans le service DNS standard. Le service de localisation gère les informations sur l'association entre les EID et les adresses. Cette approche de la mobilité résulte de l'hypothèse qu'une adresse reflète la position d'un routeur et des autres nœuds mobiles gérés par ce routeur. Quand un nœud mobile entre dans un réseau, il établit une relation avec un routeur qui connaît son EID. Le routeur fait la mise à jour de la localisation avec l'association entre son adresse et l'EID du nœud mobile. Quand un autre

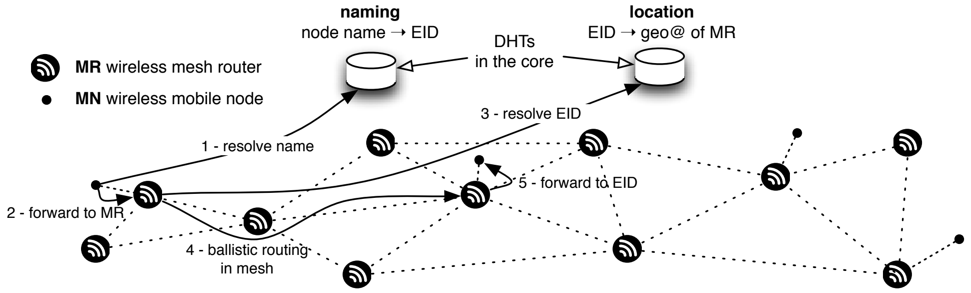

FIGURE 1.3 – Architecture du réseau sans fil multi-sauts

nœud mobile veut envoyer les paquets vers un nœud distant, il identifie d'abord son nom pour obtenir son EID correspondant et envoie le paquet avec la destination EID vers le routeur voisin, qui à son tour localise l'adresse du routeur de destination. Le paquet prend un chemin à l'adresse de destination établie par le routage géographique. Enfin, le routeur de destination envoie le paquet vers le nœud mobile de la destination. Ainsi les nœuds de communication ne voient que les paquets avec leur EID respectifs et les routeurs utilisent les adresses afin de transmettre les paquets à la destination.

Comme les EID sont persistants, la couche transport peut les utiliser sur des interfaces réseaux différents et en des endroits différents afin de maintenir des liens de transport à long terme en cas de mobilité. Nous pouvons construire les EID de plusieurs façons : nous pouvons les obtenir d'une clé publique comme dans HIP [5] ou nous pouvons prendre en compte une personne qui utilise un nœud et dériver l'EID de la clé publique de l'utilisateur. Un nœud peut utiliser différentes interfaces réseaux mais un unique EID.

1.1 L'espace d'adressage

Nous voudrions définir un espace d'adressage basé sur des coordonnées et l'utiliser pour attribuer une adresse à un EID d'une façon autonome. Un tel espace de coordonnées peut être soit virtuel soit réel, basé par exemple sur la position GPS d'un nœud. De nombreuses propositions ont pris en considération l'adressage et le routage dans des espaces virtuels de coordonnées [6, 7, 8]. Leur avantage est une facilité relative de gestion par rapport aux exigences du positionnement GPS. Toutefois, ils présentent plusieurs inconvénients. Le premier est lié à la fusion de deux sous-réseaux. Imaginons que deux régions d'un réseau se développent d'une façon indépendante et non planifiée. Chaque partie crée ses adresses dans une partie de sous-espace virtuel (nous pouvons utiliser par exemple les approches P2P pour construire un espace virtuel comme dans Content Adressable Network (CAN) [9]). Le problème est de savoir comment nous pouvons fusionner deux parties quand nous plaçons un routeur qui les relie. Une partie a besoin de changer ses adresses afin d'accueillir les adresses de la deuxième partie. Nous pouvons aussi avoir des collisions d'adresses si des adresses ont été attribuées dans des espaces de coordonnées non disjoints. Le deuxième problème est que si nous voulons gérer la mobilité, la distance dans l'espace d'adressage doit être en corrélation avec la distance dans le monde réel, car les nœuds se déplacent physiquement dans l'espace réel. Lorsque les distances dans l'espace d'adressage reflètent les mouvements réels, la mobilité la plus fréquente d'un nœud sera dans son voisinage : les adresses modifiés seront limités uniquement à cette partie de l'espace.

Si les adresses viennent des positions géographiques des nœuds, le problème de la fusion n'existe pas, parce que les nœuds dans des positions différentes utilisent des adresses différentes. Cependant, cette approche nécessite la connaissance de la position exacte de tous les nœuds, ce qui peut être trop difficile ou trop coûteux à obtenir. Plusieurs auteurs ont déjà examiné le problème de positionnement relatif. En absence de GPS (par exemple pour les localisations en intérieur) les informations sur la localisation peuvent être obtenues à partir d'un positionnement relatif basé sur les techniques de la localisation radio décrites dans le chapitre 4.

1.2 Routage

Les routeurs utilisent le routage géographique pour transmettre les paquets. Nous proposons une approche pour atteindre des destinations lointaines, que nous appelons « Waypoint Routing ». Au lieu du routage géographique glouton simple, nous utilisons des informations supplémentaires sur quelques chemins vers des destinations lointaines appelées « waypoints ». Le routeur transmet les paquets sur le meilleur chemin. Chaque routeur ne garde que des informations partielles sur les chemins, le chemin complet est gardé d'une manière distribuée. L'ensemble des chemins est construit et adapté en fonction des informations à partir des paquets déjà transmis : de cette façon le protocole n'a pas de surcoût. Nous venons de présenter uniquement l'idée du Waypoint routing. Les descriptions précises du routage Waypoint sont présentés dans les chapitres 6 et 7 de cette thèse.

1.3 Organisation du manuscrit

Cette thèse est organisée de la manière suivante. Le chapitre 2 contient une brève description des couche physique et couche MAC pour les standards les plus communs des réseaux sans fil. Dans le chapitre 3 nous décrivons les protocoles de routage. Nous commençons avec les caractéristiques classiques et finissons avec les exemples les plus intéressants du point de vue de l'architecture de réseau. Dans le chapitre4, nous étudions le modèle mathématique des protocoles de localisation. C'est important, parce que la localisation permet à une adresse d'être une coordonnée géographiques d'un nœud. Contrairement à la localisation GPS, cette technique peut être moins coûteuse et économique en énergie. Nous avons développé un simulateur de Monte-Carlo capable de gérer les réseaux à grande échelle et d'obtenir des conditions topologiques pour le modèle des réseaux sans fil qui vise à la localisation totale du réseau. Dans le chapitre 5, nous nous concentrons sur le routage géographique glouton simple dans les réseaux sans fil multi-sauts spontanés pour montrer plusieurs caractéristiques intéressantes. Le routage géographique glouton simple reste toujours un des éléments de nombreux algorithmes géographiques. Tout d'abord, nous pouvons rapprocher la dépendance de degré moyen sur la probabilité de perte de paquets avec la fonction sigmoïde. Lorsque la taille du réseau augmente, il devient opaque aux paquets (par exemple la probabilité d'une transmission réussie entre des nœuds éloignés s'approche de 0), quel que soit le degré moyen de nœud. Nous montrons aussi que la probabilité de perte de paquets dans les réseaux sans fils avec le routage géographique glouton simple ne présente pas le comportement des systèmes à percolation. Pour finir, nous proposons un modèle analytique du routage géographique glouton simple et nous l'utilisons pour calculer la probabilité de perte des paquets. Dans le chapitre 6, nous proposons « Binary Waypoint Routing », un nouveau protocole de routage géographique pour les réseaux multi-sauts sans fil. Son idée est d'apprendre et de maintenir sur les routeurs le routage par la source vers un petit nombre de nœuds appelés « waypoints », placés dans des sous-espaces construits à la suite du découpage binaire de l'espace. Un nœud source envoie un paquet à un waypoint associé à une destination et les nœuds intermédiaires tentent d'adapter la route des paquets, en visant les waypoints qui sont les plus proches de la destination. Nos résultats de simulation montrent que notre proposition atteint un taux élevé de la livraison des paquets avec un modèle de trafic similaire au routage du chemin le plus court. Dans le chapitre 7, nous proposons un protocole de routage géographique pour les réseaux multi-sauts sans fil dynamiques. Contrairement au principe précédent, notre protocole ne nécessite pas de routage par la source. Le volume de l'information de routage dans un nœud reste très faible par rapport à la taille du réseau, parce que le nombre de waypoints augmente de manière logarithmique avec le nombre de nœuds et le volume d'information de routage est limité. Nos résultats de simulation montrent que le schéma proposé atteint un taux de livraison de paquets élevé et une distribution de longueur des routes semblable au routage optimal. Nous terminerons avec quelques conclusions et perspectives pour les travaux futurs.

2

Une étude sur les réseaux sans fil

Dans ce chapitre, nous avons présenté différentes versions des standards sans fil. Bluetooth n'est pas capable de créer un réseau multi-saut sans fil à grande échelle à cause de ses limitations : manque de support IP natif, un bas débit et un nombre de machines dans un pico-net unique limité. Il peut plutôt servir, à l'instar de l'USB sans fil, pour échanger une petite quantité de données entre des périphériques dédiés, par exemple les casques ou les modems. 802.15.4 a un bas débit de données et une architecture centralisée. Il est adapté pour les réseaux PAN, mais il n'est pas conçu pour fonctionner comme un réseau sans fil à grande échelle. Son avantage est que le protocole TCP/IP peut être utilisé d'une manière native. Cependant, la taille de paquet limitée à 127 octets peut être un problème, parce que, par exemple, l'en-tête TCP/IPv4 d'environ 40 octets semble être trop longue pour ce type de réseau. Les réseaux 802.11 et 802.16 sont parfaitement capables de travailler dans une architecture multi-saut, mais ils ont besoin d'un protocole de routage externe afin de transmettre les paquets de différentes sources aux destinations. Dans le chapitre suivant, nous nous concentrons sur les protocoles de routage.

3
Les protocoles de routage

Le routage dans les petits réseaux statiques est facile. Cependant, la plupart des réseaux sans fil réels sont dynamiques (les liens peuvent apparaître et disparaître, les nœuds rejoindre ou quitter le réseau) et leur taille peut augmenter, ce qui implique un problème de grandes tables de routage $O(N)$. Les nouveaux types de réseaux spontanés amplifient cette tendance : les réseaux multi-sauts ou les réseaux des capteurs peuvent inclure des milliers de nœuds agissant comme des routeurs. À grande échelle, dans les réseaux sans fil à dynamique élevée ayant un taux de connexion/déconnexion élevé, les protocoles à état de lien comme OLSR ne sont pas appropriés parce que chaque modification de la topologie exige la propagation des informations du routage pour recalculer les meilleurs chemins. Les protocoles à vecteur de distance nécessitent la distribution périodique des informations de routage ou en cas de versions à la demande pour les réseaux ad hoc tels que AODV, ils peuvent souffrir de défaillances de chemins fréquentes et des phases de recherche qui appliquent l'algorithme d'inondation. Plusieurs expérimentations ont montré l'intensification des problèmes des protocoles de routage sans fil familiers tels que AODV, DSDV ou DSR.

Lors de l'examen de ces inconvénients, le routage géographique se présente comme une approche intéressante et une alternative au routage multi-sauts traditionnel qui présente de nombreux avantages : la décision est localisée — elle ne dépend que de l'état d'un voisinage immédiat et non de la topologie globale. Cette solution passe également bien à l'échelle, la taille des informations de routage étant limitée.

Le routage géographique glouton présente plusieurs inconvénients que nous voulons éviter. Par exemple, quand le routage perd des paquets à cause de défauts topologiques comme des régions vides concaves, une méthode de récupération comme le « face-routing » doit être utilisée. Les défauts topologiques peuvent être découverts au niveau des nœuds concaves, qui ont seulement les voisins dans la direction opposée à la région vide. Le face-routing peut contourner les défauts topologiques, mais il exige la construction d'un graphe de surface des liens entre les nœuds voisins. Plusieurs articles ont montré la difficulté de cette construction en environnement sans fil réel, un graphe non-planaire conduisant à une forte probabilité de perte de paquets.

4

Caractéristiques des techniques de localisation

Dans ce chapitre, nous avons introduit le modèle de k-propagation qui apparaît dans plusieurs classes de protocoles dans les réseaux multi-sauts sans fil. Nous avons développé un simulateur de Monte-Carlo capable de gérer les réseaux à grande échelle et nous avons étudié les conditions dans lesquelles l'information se propage dans tous les nœuds du réseau. Nos simulations montrent que la k-propagation se comporte comme un phénomène de percolation et nous pouvons estimer ses paramètres critiques. Il peut sembler que de nombreux phénomènes sont à percolation, et ce résultat serait évident, mais ce n'est pas le cas. Dans le chapitre suivant, nous montrons que la probabilité de succès de livraison des paquets dans un graphe géométrique aléatoire sous routage géographique glouton simple ne présente pas de comportement de percolation.

Enfin, nous avons analysé des protocoles de localisation totale des réseaux comme les exemples de k-propagation et nous avons comparé nos résultats théoriques avec les simulations ns-2. Les protocoles se comportent exactement comme notre modèle l'avait prévu.

Nous sommes conscients que les graphes géométriques aléatoires examinés dans ce chapitre ne fournissent qu'une approximation du premier rang du comportement des réseaux radio réels.

Le résultat le plus important du point de vue de l'architecture du réseau est que la localisation peut se passer de GPS dans le cas de réseaux très denses, dans lesquels la densité moyenne dépasse 10–11 voisins par nœud, ce qui est un très grand nombre. Nous n'examinons pas ici la précision des techniques de localisation ni l'influence de la précision sur le comportement des protocoles de routage géographique.

5

Caractéristiques du routage géographique glouton simple

L'objectif de ce chapitre est d'analyser le routage géographique glouton simple dans les réseaux multi-sauts spontanés sans fil. Notre analyse montre plusieurs propriétés intéressantes, même si nous avons utilisé un modèle simple basé sur le graphe de disques unitaires. Tout d'abord, nous pouvons rapprocher la dépendance du degré moyen dans le réseau sur la probabilité de perte de paquets de la fonction sigmoïde. Lorsque le degré moyen dans le réseau est stable, mais la taille du réseau multi-saut augmente, le réseau devient plus opaque aux paquets. Nous montrons aussi par la simulation que la probabilité de perte des paquets dans les réseaux multi-sauts avec le routage géographique glouton simple ne présente pas le comportement d'un système à percolation. Enfin, nous proposons un modèle analytique du routage géographique glouton simple et nous l'utilisons pour calculer la probabilité de perte de paquets. Dans le chapitre suivant, nous améliorons le protocole géographique glouton simple pour obtenir de meilleures propriétés.

6

Binary Waypoint Routing

Nous avons présenté le « Binary Waypoint Routing », un nouveau protocole de routage pour les réseaux multi-sauts sans fil basé sur l'idée de l'apprentissage et du maintien de routes de source vers un petit nombre des nœuds dans des sous-espaces d'adressage disjoints. Nos résultats de simulation montrent que le schéma proposé atteint une probabilité de livraison de paquets élevée avec un modèle de trafic similaire au routage du plus court chemin. L'augmentation du degré moyen des nœuds (le nombre de voisins) se traduit par une probabilité de livraison de paquets améliorée, nous pouvons obtenir les performances désirées du protocole par la construction d'un réseau avec une densité des nœuds adéquate. Le principe proposé ne nécessite pas la construction explicite de tables de routage et son fonctionnement ne dépend d'aucune propriété du graphe (par exemple graphe de disques unitaires). Le protocole fonctionne pour tout type de connectivité sans fil entre les nœuds.

7

Scalable Waypoint Routing

Nous avons présenté « Scalable Waypoint Routing », un protocole de routage géographique pour les réseaux sans fil multi-sauts à grande échelle. Son idée principale est de maintenir des routes vers un petit nombre de « waypoints » et de les utiliser pour transmettre les paquets vers n'importe quelle destination. Un nœud divise l'espace géographique d'adressage par un processus de découpage quadtree qui crée des sous-espaces de taille décroissante. Un nœud apprend les directions vers les waypoints pour les sous-espaces en utilisant les traces partielles transportées dans les paquets. Contrairement aux méthodes similaires déjà proposés, notre protocole ne nécessite pas de routage par la source, et le volume des informations sur le routage dans un nœud reste très petit par rapport à la taille du réseau, parce que le nombre de waypoints augmente de manière logarithmique avec le nombre total de nœuds. Les nœuds bornent le volume d'information sur la trace en limitant les routes à une taille fixe configurable. De cette façon, le protocole peut passer à l'échelle. Nous avons effectué des simulations en se basant sur les hypothèses simples du graphe de disque unitaire pour valider les principes du protocole et pour les comparer avec le routage optimal du plus court chemin. Nos résultats montrent que le protocole proposé atteint une probabilité élevée de livraison des paquets, même pour les réseaux de très grande taille avec 40 000 nœuds et pour les réseaux très dynamiques ayant un taux de connexion/déconnexion élevé. Il est intéressant de noter que la distribution de la longueur des routes est similaire au routage optimal du plus court chemin ce qui montre que notre méthode permet d'obtenir des chemins presque optimaux.

8
Conclusion

Les réseaux sans fil multi-sauts spontanés promettent une architecture totalement différente de l'Internet du futur par rapport à son état actuel. De nombreux problèmes doivent être résolus avant le déploiement de tels réseaux. Les algorithmes classiques de routage doivent être remplacés par des techniques appropriées qui garantissent le passage à l'échelle et une bonne connectivité. Nous avons examiné les algorithmes de routage géographique, parce qu'ils ne nécessitent pas la connaissance globale de la topologie du réseau pour calculer les chemins : ils passent donc beaucoup mieux à l'échelle que les algorithmes topologiques. Il y a, cependant, de nombreux problèmes que nous devons résoudre. Le plus important est que les algorithmes géographiques ont une efficacité faible. Dans certains cas ils calculent des chemins plus longs que les algorithmes de routage topologique. Afin de résoudre ce problème, dans le chapitre 5, nous avons étudié le comportement du routage géographique glouton simple, l'élément de base des algorithmes géographiques. Nous avons découvert que cette mauvaise performance des algorithmes géographiques apparaît parce que le routage géographique glouton ne dispose pas d'informations sur la topologie globale et il ne suit pas correctement le graphe de connectivité du réseau. Souvent, le routage géographique glouton simple ne réussit pas à délivrer les paquets, la procédure de « face routing » devant être utilisée. Afin de résoudre ce problème, nous avons proposé les protocoles « Binary Waypoint Routing » et « Scalable Waypoint Routing ». Ces protocoles ne nécessitent pas de surcoût pour découvrir la topologie du réseau, mais analysent le trafic d'une manière passive. Ce qui est étonnant est que nous ne recueillons pas d'information topologique sur le graphe, mais une information géographique limitée qui permet de transmettre les paquets de façon que ces derniers suivent les propriétés topologiques du réseau. Dans les chapitres 6 et 7, nous présentons une étude approfondie des principes proposés analysés par les méthodes de simulation. Afin de trouver les propriétés basiques de nos algorithmes, nous avons développé un simulateur Monte-Carlo capable de simuler les réseaux jusqu'à 40 000 nœuds. Nous avons trouvé que nos algorithmes ont des propriétés très intéressantes. C'est pourquoi nous avons proposé des hybrides de Scalable Waypoint Routing et de GPSR ou GOAFR+ et nous avons étudié leur comportement dans les petits réseaux en utilisant simulateur NS-2. D'après nos simulations, ces hybrides ont des performances bien meilleures que les algorithmes originaux, car elles calculent des chemins beaucoup plus courts. Binary Waypoint Routing et Scalable Waypoint Routing peuvent être facilement étendus à trois dimensions : c'est un des nouveaux champs possibles à étudier dans l'avenir. L'algorithme peut être aussi intéressant du point de vue de l'études sur les réseaux des capteurs.

Les protocoles de routage géographique utilisent les positions des nœuds pour la transmission des paquets. Afin d'obtenir des coordonnées des nœuds, des méthodes de localisation doivent être utilisées. Les nœuds peuvent utiliser le GPS ou les techniques de la localisation par radio pour obtenir les coordonnées géographiques. C'est aussi un problème important parce que la localisation a une influence sur le routage géographique. Le premier problème est quelles sont les conditions topologiques pour localiser un nœud. Dans le chapitre 4, nous avons supposé que nous effectuons la localisation totale en utilisant les techniques de localisation par radio à partir d'un petit groupe de référence qui contient quelques nœuds avec des positions connues. Nous avons trouvé que la localisation nécessite une connectivité importante dans le réseau, avec plus de 11 voisins en moyenne afin de couvrir un grand nombre des nœuds du réseau. Dans les réseaux avec une densité en dessous de 11 voisins, l'opération peut ne pas aboutir à la localisation totale et le routage géographique des paquets est alors impossible. Nous n'avons pas étudié la précision des techniques de localisation, mais la précision a également une influence importante sur le comportement des protocoles de routage géographique, de sorte que ces questions devront être prises en compte dans les prochains travaux.

Bibliography

[1] J. Saltzer, “On the naming and binding of network destinations. RFC 1498,” 1993.

[2] P. Francis, “Addressing in Internetwork Protocols. PhD thesis, University College London, UK.” 1994.

[3] I. Castineyra, N. Chiappa, and M. Steenstrup, “The Nimrod routing architecture. RFC 1992,” 1996.

[4] M. Crawford, A. Mankin, T. Narten, I. Stewart, and L. Zhang., “Separating identifiers and locators in addresses: An analysis of the GSE proposal for IPv6. Internet draft,” 1999.

[5] R. Moskowitz and P. Nikander, “Host Identity Protocol (HIP) Architecture. RFC 4423.” 2006.

[6] S. Capkun, M. Hamdi, and J. P. Hubaux, “Gps-free positioning in mobile ad-hoc networks,” *Cluster Computing*, vol. 5, April 2002.

[7] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic Routing without Location Information,” in *Proc. MOBICOM*, 2003.

[8] R. Kleinberg, “Geographic routing using hyperbolic space,” in *Proc. INFOCOM*, 2007.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content-addressable network,” in *SIGCOMM*. San Diego, USA: ACM, August 2001.

[10] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Computer Networks*, vol. 47, no. 4, pp. 445–487, Mar. 2005.

[11] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer*

(PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std., Sep. 2006.

[12] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements. Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs(tm))*, IEEE Std., Jun. 2005.

[13] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std., Jun. 2007.

[14] *IEEE Standard for Local and metropolitan area networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std., Oct. 2004.

[15] "Specification of the bluetooth system v2.0 + edr," 2004. [Online]. Available: http://www.bluetooth.com

[16] "Specification of the bluetooth system v3.0 + hs," 2009. [Online]. Available: http://www.bluetooth.com

[17] "Specification of the bluetooth system v4.0," 2009. [Online]. Available: http://www.bluetooth.com

[18] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proceedings of the 22nd Infocom 2003*, 2003.

[19] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 5: Enhancements for Higher Throughput*, IEEE Std., Oct. 2009.

[20] C. E. Perkins, E. M. Belding Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF, RFC 3561, July 2003.

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

[22] D. B. Johnson, Y.-C. Hu, and D. A. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," IETF, RFC 4728, February 2007.

[23] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF, RFC 3626, October 2003.

[24] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. of ACM Int. Workshop DIAL M99*, August 1999, pp. 609–616.

[25] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. MOBICOM*, Boston, USA, August 2000.

[26] D. P. Bertsekas and R. G. Gallager, "Distributed asynchronous bellman-ford algorithm," in *Data Networks*. Prentice Hall, Englewood Cliffs, 1987, ch. 5.2.4, pp. 325–333.

[27] U. Black, *IP Routing Protocols: RIP, OSPF, BGP, PNNI and Cisco Routing Protocols*, 1st ed. Prentice Hall, February 2000.

[28] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. of ACM SIGCOMM*, 1999, pp. 251–262.

[29] G. Finn, "Routing and Addressing Problems in Large Metropolitan–Scale Internetworks ," 1987.

[30] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the 11th Canadian Conference on Computational Geometry*, 1999.

[31] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," in *Proc. IEEE*, 1992, pp. 1502–1517.

[32] G. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

[33] K. Gabriel and R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, pp. 259–278, 1969.

[34] Y. Kim, R. Govindan, B. Karp, and S. S., "Lazy cross-link removal for geographic routing," in *Proc. SENSYS*, 2006.

[35] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Principles of Distributed Computing (PODC)*. ACM, 2003.

[36] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information," in *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP '05)*, Boston, Massachusetts, November 2005.

[37] Y. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proc. NSDI*, 2005, pp. 112–124.

[38] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2005, pp. 31–42.

[39] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 121–132, 2004.

[40] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2002, pp. 148–159.

[41] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, “A Scalable Location Service for Geographic Ad Hoc Routing,” in *Proc. of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom ’00)*, Boston, Massachusetts, August 2000, pp. 120–130.

[42] R. Jain, A. Puri, and R. Sengupta, “Geographical Routing Using Partial Information for Wireless Ad Hoc Networks,” *IEEE Personal Comm.*, February 2001.

[43] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless Sensor Network Localization Techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, July 2007.

[44] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” in *Proc. MobiCom 2001*, Rome, Italy, 2001, pp. 166–179.

[45] A. Savvides, H. Park, and M. B. Srivastava, “The N-hop Multilateration Primitive For Node Localization Problems,” *Mobile Networks and Applications Journal*, vol. 8, number 4, pp. 443–451, August 2003.

[46] S. Nawaz and S. Jha, “Collaborative Localization for Wireless Sensor Networks,” in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07)*, Sept. 2007.

[47] S. Lanzisera, D. T. Lin, and K. S. Pister, “RF Time of Flight Ranging for Wireless Sensor Network Localization,” in *Proc. of WISES 2006*, Vienna, Austria, June 2006.

[48] T. Karalar and J. Rabaey, “An RF ToF Based Ranging Implementation for Sensor Networks,” in *Proc. IEEE ICC 2006*, vol. 7, Istanbul, Turkey, June 2006, pp. 3347–3352.

[49] F. Shin, L. Zhiwei, and L. J. Xing, “Symmetric Multi-Way Ranging for Wireless Sensor Network,” in *Proc. IEEE PIMRC 2007*, Athens, Greece, September 2007, pp. 1–5.

[50] Z. Sahinoglu and S. Gezici, “Ranging in the IEEE 802.15.4a Standard,” in *Proc. of IEEE WAMICON’06*, December 2006, pp. 1–5.

[51] D. Niculescu and B. Nath, “Ad hoc Positioning System (APS) using AOA,” in *Proceedings of INFOCOM 2003*, vol. 3, April 2003, pp. 1734–1743.

[52] P. Bergamo and G. Mazzini, “Localization in Sensor Networks with Fading and Mobility,” in *Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications(PIMRC02)*, vol. 2, Sept. 2002, pp. 750–754.

[53] P. Bahl and V. N. Padmanabhan, “RADAR: An In-building RF-based User Location and Tracking System,” in *Proceedings of IEEE INFOCOM 2000*, vol. 2, Tel-Aviv, Israel, 2000, pp. 775–784. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2000.832252

[54] S. Tanvir, E. Schiller, B. Ponsard, and A. Duda, “Propagation Protocols for Network-wide Localization Based on Two-Way Ranging,” in *Proc. IEEE WCNC*, Apr 2010.

[55] J. Dall and M. Christensen, “Random Geometric Graphs,” *Phys. Rev. E*, vol. 66, no. 1, Jul 2002.

[56] C. Bettstetter, “On the minimum node degree and connectivity of a wireless multihop network,” in *MOBIHOC*, 2002.

[57] B. Clark, C. Colbourn, and D. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[58] P. Gupta and P. Kumar, “Capacity of Wireless Networks,” *IEEE Trans. on Information Theory*, vol. IT-46, 2000.

[59] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi, “Fault Tolerant Deployment and Topology Control in Wireless Networks,” in *Proc. of MobiHoc 2003*. ACM, 2003, pp. 117–128.

[60] P. Gupta and P. Kumar, “Critical Power for Asymptitic Connectivity in Wireless Networks,” *Stochastic Analysis, Control, Optimization, and Applications*, pp. 547–566, 1998.

[61] M. D. Penrose, “On k-Connectivity for a Geometric Random Graph,” *Random Struct. Algorithms*, vol. 15, no. 2, pp. 145–164, 1999.

[62] X. Ta, G. Mao, and B. Anderson, “On the Probability of k-Hop Connection in Wireless Sensor Networks,” *IEEE Communication Letters*, 2009.

[63] ——, “On the Connectivity Properties of Wireless Multi-Hop Networks,” in *Proc. Chinacom 2008*, 2008.

[64] ——, “On the Properties of Giant Component in Wireless Multi-hop Networks,” in *Proc. 28th IEEE Infocom’09 mini-conference*, 2009.

[65] S. Mueller, R. Tsang, and D. Ghosal, “Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges,” in *Lecture Notes in Computer Science, Volume 2965*, 2004, pp. 209–234.

[66] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. R. Kumar, “Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol,” in *Proc. European Wireless*, 2002.

[67] V. Kawadia and P. Kumar, “Power Control and Clustering in Ad Hoc Networks,” in *Proc. INFOCOM*, 2003.

[68] F. Xue and P. R. Kumar, “The Number of Neighbors Needed for Connectivity of Wireless Networks,” *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[69] ——, “On the θ-Coverage and Connectivity of Large Random Networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2289–2299, 2006.

[70] D. Miorandi and E. Altman, “Coverage and Connectivity of Ad Hoc Networks in Presence of Channel Randomness,” in *INFOCOM*, 2005, pp. 491–502.

[71] J. Cartigny and D. Simplot, “Border Node Retransmission Based Probabilistic Broadcast Protocols in Ad-Hoc Networks,” *Telecommunication Systems*, vol. 22, no. 1-4, pp. 189–204, 2003.

[72] Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks,” in *Proc. of IEEE WCNC 2003*, 2003, pp. 1124–1130.

[73] Q. Zhang and P. Agrawal, Dharma, “Dynamic Probabilistic Broadcasting in MANETs,” *J. Parallel Distrib. Comput.*, vol. 65, no. 2, pp. 220–233, 2005.

[74] V. Raman and I. Gupta, “Performance Tradeoffs Among Percolation-based Broadcast Protocols in Wireless Sensor Networks,” in *Proc. Workshop on Wireless Ad hoc and Sensor Networks*, 2009.

[75] D. Centola, V. Eguíluz, and M. Macy, “Cascade Dynamics of Complex Propagation,” *Physica A*, no. 374, 2007.

[76] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-Based Ad Hoc Routing,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 479–491, 2006.

[77] C. Savarese, J. M. Rabaey, and K. Langendoen, “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks,” in *Proc. USENIX Annual Technical Conference*, 2002, pp. 317–327.

[78] J. Liu, Y. Zhang, and F. Zao, “Robust Distributed Node Localization with Error Management,” in *Proc. ACM MobiHoc 2006*, Florence, Italy, May 2006, pp. 250–261.

[79] S. McCanne and S. Floyd, *The NS2 Network Simulator, Version 2.28*, 2005. [Online]. Available: http://www.isi.edu/nsnam/ns/

[80] E. Schiller, P. Starzetz, F. Theoleyre, and A. Duda, “Properties of Greedy Geographical Routing in Spontaneous Wireless Mesh Networks,” in *Proc. IEEE GLOBECOM*, 2007.

[81] V. Beffara and V. Sidoravicius, “Percolation theory,” *Encyclopedia of Mathematical Physics*, Elsevier 2006.

[82] I. Glauche, W. Krause, R. Sollacher, and M. Greiner, “Continuum Percolation of Wireless Ad-hoc Communication Networks,” *Physica: Statistical Mechanics and its Applications*, vol. 325, no. 3-4, pp. 577–60, 2003.

[83] “Distances in bounded regions,” http://www.mathpages.com.

[84] E. Schiller, P. Starzetz, F. Rousseau, and A. Duda, “Binary Waypoint Geographical Routing in Wireless Mesh Networks,” in *Proc. ACM MSWiM*, 2008.

Publications

1 Sadaf Tanvir, Eryk Schiller, Benoît Ponsard and Andrzej Duda. Propagation Protocols for Network–Wide Localization Based on Two-Way Ranging. *In Proceedings of IEEE WCNC (Wireless Communications and Networking Conference)*. Sydney, Australia, April 18-21, 2010.

2 Fabrice Theoleyre, Eryk Schiller, Andrzej Duda. Efficient Greedy Geographical Non–Planar Routing with Reactive Deflection. *In Proceedings of IEEE ICC (International Conference on Communications)*. Dresden, Germany, June 14-18, 2009.

3 Eryk Schiller, Paul Starzetz, Franck Rousseau, Andrzej Duda. Binary Waypoint Geographical Routing in Wireless Mesh Networks. *In Proceedings of ACM MSWiM 2008*. Vancouver, Canada, 27-31 October 2008.

4 Eryk Schiller, Paul Starzetz, Fabrice Theoleyre and Andrzej Duda. Properties of Greedy Geographical Routing in Spontaneous Wireless Mesh Networks. *In Proceedings of IEEE GLOBECOM 2007*. Washington, DC, USA, November 26-30, 2007.

5 Franck Rousseau, Yan Grunenberger, Vincent Untz, Eryk Schiller, Paul Starzetz, Fabrice Theoleyre, Martin Heusse, Olivier Alphand, Andrzej Duda. An Architecture for Seamless Mobility in Spontaneous Wireless Mesh Networks. *In Proceedings of the 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (Mobiarch'07)*. Kyoto, Japan, August 27, 2007.