FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

ZigBee Network Layer Simulation on top of IEEE 802.15.4

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Informatik

eingereicht von

Dominik Bunyai
Matrikelnummer 0625305

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.-Prof. Dr. Wolfgang Kastner
Mitwirkung: Univ.-Ass. Dipl.-Ing. Lukas Krammer

Wien, 07.12.2012

(Unterschrift Verfasser) (Unterschrift Betreuung)

Erklärung zur Verfassung der Arbeit

Dominik Bunyai
Redtenbachergasse 59/25, 1160 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Danksagung

In erster Linie möchte ich mich bei meinen Betreuern Wolfgang Kastner und Lukas Krammer bedanken. Die Möglichkeit, jederzeit mit Fragen und Diskussionen zu ihnen kommen zu können und nicht zuletzt ihr hoher Einsatz bei den abschließenden Korrekturen hat geholfen die Diplomarbeit zu einem guten Ende zu bringen.

Weiters gilt mein Dank allen Studienkollegen, die viel Abwechslung in mein Studentenleben gebracht haben. Im speziellen möchte ich Christian danken, der mir durch einige studientechnische als auch persönliche Höhen und Tiefen geholfen hat.

Für die Ermöglichung des Studiums und die vorhergehende Ausbildung möchte ich mich bei meinen Eltern, Silvia und Robert Bunyai bedanken. Ebenso konnte ich auch bei meinem Bruder Peter jederzeit auf die familiäre Unterstützung vertrauen. Zu guter Letzt gilt mein besonderer Dank meiner Freundin Susanne, die mich bereits durch mein gesamtes Studium begleitet hat und auch in lern- und arbeitsintensiven Zeiten zu mir gestanden ist.

Abstract

Wireless systems surround us, anywhere we go. Well known technologies build up the wireless infrastructure mobile phones depend on. Wireless network equipment has been introduced to private homes in the last decade. The wireless automation infrastructure started to connect sensors and actuators of commercial products a few years ago. The increase of wireless products in the area of Home and Building Automation (HBA) can be observed in recent years.

ZigBee is one of the leading technologies in HBA. This protocol standard, developed in 2003, provides the functionalities expected for a Wireless Sensor and Actuator Network (WSAN). Point-to-point connections, as well as star networks can be built using ZigBee. The routing capabilities are the main feature of ZigBee. Therefore, ZigBee devices can also be used to build mesh networks.

Whenever WSANs are used in automation, they are expected to reliably provide their functionality. However, most of the wireless protocols in this context are based on indeterministic media access mechanisms. This can lead to sudden system failures. Due to this unintended behavior, a new approach for predicting WSANs' message transfer is required.

Initiating a new concept for predicting network behavior requires detailed knowledge of the protocol and its basis. Furthermore, the concept needs to be checked for correctness. Due to the indeterminism of WSANs, the behavior of such systems in terms of scalability and timeliness cannot be evaluated with formal methods. Therefore, estimation rules are deduced to predict the behavior of ZigBee networks. An implementation of the network protocol in a simulation framework is used to produce comparable results in parallel.

The estimations for predicting the behavior of ZigBee networks presented in this thesis provide the following opportunities. The average transmission delay of ZigBee networks can be estimated and predictions about the average packet transmission delay of any ZigBee network can be made. Network downtimes due to an overload of the network can be predicted by assessing the actual channel utilization. Additionally, the efficiency of the ZigBee protocol according to specific networks can be estimated. Furthermore, the relation of network management packet payload to data packet payload becomes predictable.

Kurzfassung

Eine Vielzahl von funkbasierten Systemen wird bereits in unserer näheren Umgebung eingesetzt. Die bekannteren Systeme stellen die notwendige Infrastruktur für unsere Mobiltelefone dar. Mittlerweile haben sich funkbasierte Netzwerke auch in Privathaushalten durchgesetzt. In den letzten Jahren konnte ein Vorstoß von drahtlosen Systemen im Heim- und Gebäudeautomationssektor beobachtet werden.

ZigBee ist eines der führenden Protokolle in der Heim- und Gebäudeautomation. ZigBee unterstützt den Aufbau von Punkt-zu-Punkt Verbindungen genauso wie den Aufbau von Netzwerken in einer Stern-Struktur. Der Hauptaufgabenbereich von ZigBee ist allerdings das Routing von Paketen. Daher ist auch der Aufbau von vermaschten Netzwerken mit ZigBee-Geräten möglich. ZigBee wurde im Jahr 2003 entwickelt und erfüllt all jene Funktionalitäten, die von einem funkbasierten Sensor-Netzwerk gefordert werden.

Die hohen Erwartungen an funkbasierte industrielle Sensor-Netzwerke, setzen niedrige Ausfallraten und hohe Verfügbarkeit voraus. Allerdings basieren die Zugriffsverfahren auf das Netzwerkmedium der meisten dieser Systeme auf nicht deterministischen Methoden. Daraus resultierende Systemausfälle können plötzlich auftreten. Da ein prognostizierbares Verhalten von Sensor-Netzwerken notwendig ist, müssen neue Konzepte zur Vorhersage entwickelt werden.

Für die Entwicklung eines neuen Konzepts sind sowohl detailliertes Wissen über das zugrunde liegende Netzwerkprotokoll als auch die Kenntnis über das Grundgerüst des Protokolls selbst notwendig. In Folge muss das Konzept auf Korrektheit geprüft werden. Hierfür wurde das gesamte Netzwerkprotokoll in einer Simulationsumgebung implementiert. Die daraus resultierenden Ergebnisse können mit jenen der analytischen Konzepte verglichen werden.

Das entwickelte Konzept kann zur Abschätzung und Vorhersage des Verhaltens von ZigBee-Netzwerken verwendet werden und bietet im Detail folgende Möglichkeiten: Die durchschnittliche Übertragungszeit von Paketen innerhalb eines ZigBee-Netzwerks, sowie deren tatsächliche Auslastung und Effizienz kann abgeschätzt werden. Vorhersagen über voraussichtliche Übertragungszeiten, Ausfälle auf Grund der Überlastung eines Netzwerks und das Verhältnis von Netzwerk-Managementpaketen zu Datenpaketen bezogen auf deren tatsächliche Größe können getätigt werden.

Contents

Abbreviations

ACL	Access Control List
AES	Advanced Encryption Standard
AL	Application Layer
AODV	Ad hoc On-Demand Distance Vector
BE	Backoff Exponent
BP	Backoff Period
BTR	Broadcast Transaction Record
CAP	Contention Access Period
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM*	Counter with CBC-MAC
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CSV	Comma-separated values
DAAM	Distributed Address Assignment Mechanism
DLL	Data Link Layer
DSSS	Direct Sequence Spread Spectrum
FFD	Full Function Device
GTS	Guaranteed Time Slot
GUI	Graphical User Interface
HBA	Home and Building Automation
HERA	HiErarchical Routing Algorithm
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISA	International Society of Automation
ISO	International Organization for Standardization
IT	Information Technology
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LLC	Logical Link Control

MAC	Media Access Control
NFS	Network File System
NL	Network Layer
NLDE	Network Layer Data Entity
NLME	Network Layer Management Entity
NRD	Non-routing Device
O-QPSK	Offset-Quadrature Phase Shift Keying
OOS	Optimal On-Tree Selection Algorithm
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PAN	Private Area Network
QoS	Quality of Service
RD	Routing Device
RFD	Reduced Function Device
RPC	Remote Procedure Call
RREP	Route reply
RREQ	Route Request
RTE	Routing Table Entry
SAAM	Stochastic Address Assignment Mechanism
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
UWB	Ultra-wideband
WPAN	Wireless Private Area Network
WSAN	Wireless Sensor and Actuator Network
XML	Extensible Markup Language
ZAP	ZigBee Application Profile
ZC	ZigBee Coordinator
ZDO	ZigBee Device Object
ZED	ZigBee End Device
ZiFA-R	Reliable ZigBee Forward Node Selection Algorithm
ZiRA	ZigBee Rebroadcast Algorithm
ZR	ZigBee Router

CHAPTER 1

Introduction

1.1 Field of interest

Telecommunication networks rely on Computer Science and Electrical Engineering either. The knowledge of both sciences is combined in the Computer Engineering area. One of the topics Computer Engineering deals with is automation.

Wireless technologies are ubiquitous in various domains. Currently, Wireless Local Area Network (LAN) infrastructure is available in many public areas and almost all private homes. The increasing number of HBA systems influences the development of wireless products in this scope. Nowadays, HBA technologies including wireless modules are produced.

The last century yielded various wireless technology standards and specifications. Since all wireless products revert to the same physical premises, the handling of electro-magnetic waves is more or less the same for all of those products. Differences exist at the assemblies and applications built on top of the physical modules.

This thesis compares the most popular standards in the area of wireless technologies. Furthermore, ZigBee as one of those wireless technologies, is examined in detail.

1.2 Motivation

Wireless technologies and their behavior are hard to predict due to their indeterministic behavior. The number of applications based on wireless technologies is steadily increasing. In the automation domain, even applications requiring high availability and reliability are using wireless technologies. Therefore, it is necessary to predict the behavior of wireless networks.

ZigBee's applications in HBA imply the importance to know the existing limits in reliability and interference tolerance. Naively built ZigBee networks often have to deal with over-utilization and system breakdowns as a result. Over-utilized networks even lack in timeliness of packet reception. Therefore, analyses resulting in estimations for predicting ZigBee behavior are necessary.

Predicting the system behavior before building the network can prevent unintended behavior. Simulations can visualize the timely behavior and produce statistical data. Therefore, the implementation of typical ZigBee networks inside a simulation framework is desired.

1.3 Problem statement

The behavior of ZigBee has to be examined to check the feasibility of desired network structures in advance. Since no sufficient method for predicting ZigBee networks exists, new approaches to fulfill this task are necessary. The required density of network nodes and the maximum load a network can manage have to be considered in these concepts.

When new approaches are developed to predict a systems' behavior, validation of the concept is necessary to rely on it. Validation of system concepts can be performed by proving it by simulating the system and comparing the results. By using the results of the simulation, the correctness of the prediction approach is emphasized. Such approach can then be used to detect limiting constraints for ZigBee networks. Estimations regarding the channel utilization can help detecting bottlenecks in the network. Protocol overhead predictions reveal the necessity for management packets, that cause additional network traffic. Identified unnecessary overhead can be reduced by adapting the structure of the network. In general, the knowledge of ZigBee's limits helps optimizing networks.

Due to the unpredictability of WSANs, also the behavior of ZigBee networks can not be predicted. Since applications in automation require high dependability and reliability, timing constraints and memory assumptions are necessary in advance. Therefore, rules to estimate ZigBee networks have to be developed. The estimation rules can then be used to predict ZigBee networks and to detect limits of ZigBee networks.

1.4 Aim of this thesis

This thesis is expected to present estimations to predict the behavior of ZigBee networks according to timing constraints and network utilization. The estimations for the behavior of ZigBee networks needs to predict the maximum channel utilization as well as the average transmission delays. The estimations shall be based on the number of devices associated with the network and the number of messages transmitted in one second as parameters. The computations shall be verified by simulation. This simulation has to be created with an open simulation framework. Statistical evaluations of the simulation results shall be visualized and used to prove the estimations. The simulation framework can further be used to illustrate the feasibility of network topologies in advance, without investing into physical network devices.

1.5 Methodology

Starting with the IEEE 802.15.4 standard [18] and the ZigBee specification [20], relevant literature in the area of WSANs, routing mechanisms and simulation are studied. Based on this, an approach to estimate the behavior of ZigBee devices in different types of networks is developed.

To derive the behavior of the technology in a formal way, simple and regular network structures are examined. To verify the formal considerations, simulation is used. Thereby, different simulation methods and environments are investigated. The most fitting simulation framework is used to evaluate the estimations by comparing estimated values with simulation results. Based on this comparison, the validity of the estimations is proven.

1.6 Structure of this thesis

Introducing wireless technologies, the current state of the art is presented in Chapter 2. Starting with a description of the abstracted communication model used for Information Technology (IT)-protocols, the physical limits and standards are illustrated. Various technologies in the wireless area are presented and compared. Depending on the different attributes of the technologies the advantages and disadvantages for typical use cases are discussed. Subsequently, common simulation frameworks are presented and compared regarding their required features for examining WSANs.

Subsequently, the main chapter (Chapter 3) of this thesis deals with the analysis of the ZigBee communication mechanisms. Furthermore, the main concepts for timing predictions and channel utilization of ZigBee networks are presented. A wide range of scenarios helps to understand properties and the behavior of ZigBee networks.

In contrast, Chapter 4 presents methods to evaluate the developed approach. Based on the previous scenarios, the behavior of simulated ZigBee networks is examined. Finally, the estimated values are verified by comparing them with the simulation results.

This thesis is concluded with Chapter 5, where the presented results of this work are discussed. The essentials of alternative concepts are presented and differences, advantages and disadvantages are emphasized. The very end of this thesis proposes some ideas for further work.

CHAPTER 2

State of the art

2.1 Overview

This chapter presents the state of the art of Wireless Private Area Network (WPAN) technologies. Compared to wired networks, wireless networks provide a higher level of flexibility. Wireless devices can be arbitrarily moved, which leads to an advanced mobility for the user. Most of the standards in the field of WPAN technologies are defined by the Institute of Electrical and Electronic Engineers (IEEE).

To be able to structure network protocols, the Open Systems Interconnection (OSI) model is introduced and the layers are discussed in detail. Some protocols do not clearly use all layers of the OSI model. The subsequent examination tries to assign protocol mechanisms to the appropriate OSI model layer.

The further presented standards represent the leading protocols in the area of WPAN technologies. Therefore, a detailed description about basic concepts and their relation to the OSI model of each standard is presented.

Finally, the most widely used simulators for network simulation are presented. The appropriate sections point out the advantages and disadvantages of each simulator with respect to the requirements of the simulation used for this thesis.

2.1.1 OSI model

The OSI model, specified in [14] by the International Organization for Standardization (ISO), represents an abstraction of communication protocols used in telecommunication networks. Different abstraction level of network protocols are outlined by the seven layers of the OSI model. The following list introduces the OSI layers and the main functions are mentioned.

1. Network protocols use different transmission media that are defined by the Physical Layer. For example twisted cable, ethernet cable or fiber optic cable can be used for wired networks. In contrast, wireless networks use the ”air” as the transmission medium. The

physical layer defines which medium can be used by a network protocol and how data is encoded on the medium (e.g. modulation). For example, frequency bands and their usage belong to the physical layer of the OSI model.

2. At the Data Link Layer (DLL), establishing reliable connections and reducing errors are the main tasks. This means, the DLL has to detect and possibly correct errors which occur at the physical layer [14]. One possibility is to use checksums (i.e. Cyclic Redundancy Check (CRC)) to ensure correct data transmission. Furthermore, by using data flow control mechanisms, a reduced number of data packets needs to be discarded and additional bandwidth is gained.

3. The Network Layer (NL) is an abstraction of all the mechanisms necessary to build networking connections, "ranging from network-connections supported by point-to-point configurations to network-connections supported by complex combinations of subnetworks with different characteristics." [14] Network-wide addressing as well as relaying messages are part of the NL. Higher routing performance is aimed by Quality of Service (QoS) mechanisms (i.e. analyses and use of the wireless link quality) that are also part of the NL to evaluate the network quality. Well known protocols at the NL are Internet Protocol (IP), Internet Protocol version 6 (IPv6) and Internet Control Message Protocol (ICMP).

4. Responsible for data segmentation and congestion avoidance, the Transport Layer is based on top of the NL. Furthermore, a uniform interface is provided to the upper layers (Session, Presentation and Application).

5. Session-related services (e.g. check points) are defined by the Session Layer of the OSI model. The data stream is organized and synchronized to reduce the overhead caused by interrupted sessions, that are resumed. For example, the Remote Procedure Call (RPC) protocol is located at this layer.

6. The Presentation Layer defines global representation schemes for data to enable information exchange between systems with different codesets.

7. The Application Layer (AL) allows the applications in a system to access the network. Well known protocols at the AL are Secure Shell (SSH), Network File System (NFS), Telnet, Simple Mail Transfer Protocol (SMTP), Hypertext Transfer Protocol (HTTP), Lightweight Directory Access Protocol (LDAP) and many more.

2.2 IEEE 802.15.4

The IEEE 802.15.4 protocol standard is located at the physical layer and DLL. It is used for wireless low power network communication and was specified by the IEEE 802.15.4 working group founded in 2003. Devices are self-organizing in small networks called Private Area Networks (PANs).

IEEE 802.15.4 defines two sublayers of the DLL – Logical Link Control (LLC) and Media Access Control (MAC). The MAC sublayer defines the concurrent access of multiple devices to the common medium using appropriate protocols (e.g. Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA)), whereas the LLC coordinates the distribution of data packets to the corresponding services at the Network Layer.

The two layers included in IEEE 802.15.4 completely define the medium to be used for communication and concepts to access the medium in an appropriate way. At the physical layer, frequency bands and different scan mechanisms are specified. The MAC layer specification gives concrete mechanisms to be used for accessing the medium. Due to the shared medium, the main issue of the MAC layer is to recognize and avoid packet collisions. Additionally, a low level addressing scheme, allowing small networks, is defined by the specification.

At the physical layer of the OSI model, ZigBee rests upon the IEEE 802.15.4-2003 Standard (see Section 2.2). Using modern radio technology with Offset-Quadrature Phase Shift Keying (O-QPSK) and Direct Sequence Spread Spectrum (DSSS) enhances the signal-to-noise ratio. Avoiding collisions using CSMA/CA at the DLL reduces the possibility of self-corruption of the network. Checksums ensure data integrity. Transmission retries and acknowledges at each node reduce the possibility of packet losses. The reliability of a link is detected by considering the link quality indicator, which is calculated using signal-to-noise ratio and the receiving power.

The IEEE 802.15.4 standard defines different frequency ranges. Typically, the 2.4 GHz frequency band is used by IEEE 802.15.4 devices. Wireless LAN (IEEE 802.11, [18]) and Bluetooth (IEEE 802.15.1, [4]) networks also use the 2.4 GHz frequency band which leads to interferences. To contract, the 868/915 MHz frequency range can be used, nowadays often used by stereo headphones and temperature sensors. The 868 MHz range is only used in Europe, whereas the 915 MHz is common in North America and Australia. In addition to those two frequency ranges, the IEEE 802.15.4 working group defined the 802.15.4a standard in 2007 which includes the use of the Ultra-wideband (UWB) technology. The UWB technology uses a large bandwidth while producing energy pulses at specific time intervals. One of the advantages of UWB is the reduction of multipath propagation problems by using a bandwidth wider than 500MHz.

IEEE 802.15.4 defines two device types – Full Function Devices (FFDs) and Reduced Function Devices (RFDs). The main difference between FFDs and RFDs is that RFDs do not provide capabilities to start up a PAN. Therefore, a PAN is always started up by a FFD. Furthermore, direct connections are only possible between two FFDs or a FFD and a RFD. Two RFDs can not communicate directly. Typical network structures used in IEEE 802.15.4 PANs are later discussed in Section 3.3.1.

A PAN is a network of devices within a small range. Co-existence of multiple PANs is possible, even when using the same frequency. Those self-organizing structures provide easy maintainability and some elementary self-healing mechanisms.

Network topologies

Star networks consist of one center node (FFD) that takes part in each communication and end devices (RFDs) communicating with the center node (as illustrated in Figure 2.1). Whenever two RFDs need to communicate, the first RFD transfers the information to the center node.

Further, the center node transmits the information to the second RFD. This communication scheme causes a high utilization of the center node. Besides, the center node represents a single point of failure. If – for any reason – the center node fails, no further communication between two RFDs connected to the center node will be possible anymore.

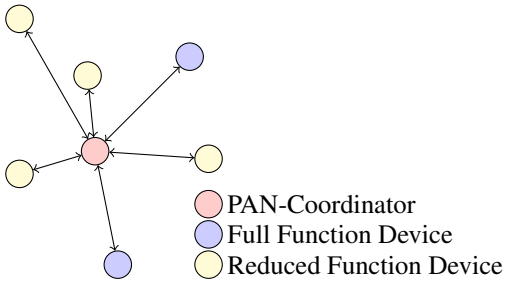

Figure 2.1: Star network topology

The second possible network topology defined by IEEE 802.15.4 is the peer-to-peer topology (as illustrated in Figure 2.2). In peer-to-peer networks, FFDs directly communicate with each other. Communication with RFDs is also possible since the FFD acts as a relay. Each single point of failure can be removed by increasing the number of direct neighbors. Still, routing along the network is not possible. The only possibility for a communication across the network are broadcasts, implying high traffic.

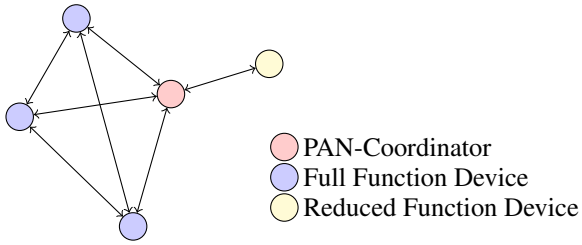

Figure 2.2: Peer-to-peer network topology

Beacons

The DLL of IEEE 802.15.4 proposes CSMA/CA as the media access strategy. Slotted and unslotted CSMA/CA mode or beacon enabled and non-beacon enabled network mode are defined, respectively. When using slotted mode, time slots are tagged by special bit combinations called beacons. The time duration between two beacons is called beacon interval. Each beacon interval consists of a given count of superframes (each lasting for a given time amount called superframe duration). The aggregate of all superframes is called Contention Access Period (CAP) (see Figure 2.3). For each time slot in the CAP, all devices having packets pending to transmit need to use the CSMA/CA mechanism to acquire the right to transmit the next pending packet in the according time slot. Additionally, the PAN coordinator can define a Contention Free Period (CFP),

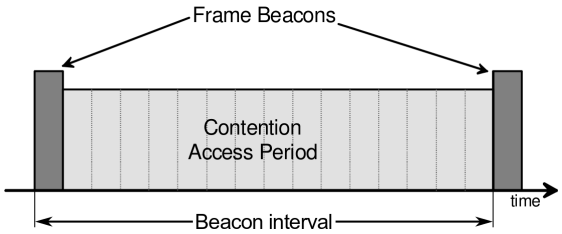

Figure 2.3: IEEE 802.15.4 Superframe structure [19]

in which Guaranteed Time Slots (GTSs) can be assigned to network devices. This GTSs can be used for real-time communication due to its deterministic behavior within the PAN. Beacon enabled network coordinator nodes periodically send beacons, including information about the network and the current transmission period. In non-beacon enabled networks, the coordinator only sends a beacon if a beacon request has been sent by any device in range. Furthermore, a device associated with a non-beacon enabled network has to acquire a time slot by using the CSMA/CA at all time to transmit a packet.

The MAC sublayer of IEEE 802.15.4 uses CSMA/CA to avoid packet collisions. CSMA/CA is a media access mechanism that avoids collisions of packets on the medium. Therefore, a device has to follow the communication on the medium and detect idle times, where the medium is not in use. Still, collisions are possible whenever two devices calculate the same point in time to start their transmission. CSMA/CA will be discussed in detail in Section 3.2.3. Using acknowledges for all unicast packets helps recognizing collisions or packets not received by the destination node.

The IEEE 802.15.4 standard defines a basic security framework for protocols at higher levels, that can be managed by using management service primitives. Symmetric cryptography can be enabled by providing appropriate keys to the DLL. Whether a packet received was encrypted is indicated by a flag in the data service primitive. Furthermore, Access Control Lists (ACLs) can be defined to restrict the reception of packets to a managed group of devices.

2.3 WirelessHART

WirelessHART is a communication standard (IEC 62591 [9]) developed by the HART Communication Foundation. In addition to the well established wired protocol standard, a compliant wireless standard was specified. Wired and wireless technologies are combined to build reliable network structures.

The lowest part of WirelessHART w.r.t. the OSI model is based on the IEEE 802.15.4 2.4 GHz physical layer. The intensively used frequency band requires robust mechanisms to reliably access the medium. Therefore, the HART Communication Foundation uses Time Division Multiple Access (TDMA) at the DLL [7]. Devices can be interconnected with Routing Devices (RDs) on behalf. Routing resides in the WirelessHART NL that provides Graph Routing, Source Routing and Superframe Routing. The AL of WirelessHART operates using HART

commands. Those HART commands consist of requests and responses specific for each device type.

The TDMA mechanism is used at the DLL of WirelessHART as the medium access strategy. TDMA uses predefined time slots to prevent collisions caused by simultaneous transmission data packets. Each of these slots can be used by the communicating devices to transfer their data. Collisions are avoided by predefining specific timestamps for each device to access the medium. This means, each device accessing a network using TDMA needs to agree on the same time base. According to [9], "devices should be built such that it does not require a keep-alive more often than once per 30 s." WirelessHART uses fixed 10ms time slots. Any transmission of a network member has to be finished within the appropriate time slot. To synchronize the internal clocks of the devices, the clocks of the sending and receiving device are compared for each packet sent and received. The difference between the expected reception and the actual reception corresponds to the current time drift of the clock at the receiving device.

WirelessHART implements Source Routing. The sending device includes information about the nodes the packet has to traverse in the message itself. Each hop the packet has to pass on its way to the destination device, can use a pointer to the Source Routing Address structure to fetch the address of the next hop. Using this routing strategy, no routing information is required at the forwarding nodes.

Graph Routing is based on predefined graphs, constructed by the network manager, that include directed edges between the devices. Each device in the network requires information about the graph and the links to its neighbors. The network manager has to construct a graph without loops, otherwise a packet would travel endlessly in such a loop. Graph Routing can coexist with Source Routing to raise the possibility of successful routing. Whenever an unknown graph is referenced by a transmitted packet, Superframe Routing is used [15].

Superframe Routing is the third routing method defined. Packets can be sent to a neighbor using the link information included in superframes. The attempt to forwarding the packet to a neighbor with a link to the current superframe is taken due to the possibility that this neighbor can successfully transmit the packet to the desired destination.

2.4 ISA100.11a

ISA100.11a "is a wireless mesh networking standard that is targeted to provide reliable and secure wireless communication and operation for process control and related application" [34]. Combining features of all current WPAN technologies, ISA100.11a aims at a reliable communication protocol in the industrial automation area.

Similar to WirelessHART (refer to Section 2.3), the physical layer of ISA100.11a is based on the IEEE 802.15.4 standard. Also the MAC layer of ISA100.11a uses mechanisms defined in IEEE 802.15.4, with some changes in the security subsystem.

ISA100.11a uses TDMA in combination with Carrier Sense Multiple Access (CSMA) and channel-hoping as the medium access mechanism [29]. Each time slot has a duration of 10ms. Interferences with other networks and protocols are reduced by CSMA as well as channel-hoping. Due to statistical methods operated on the information received by the wireless devices,

specific channels can be omitted. This method called channel-blacklisting can be used for a time interval or permanently, in case of high crowded channels.

At the DLL 16-bit addresses as defined by IEEE 802.15.4 are used. A higher number of devices is supported by the NL of ISA100.11a by using IPv6 addressing scheme with 128-bit addresses. As follows, the full IPv6 protocol header is used at the NL.

ISA100.11a uses two routing mechanisms. At the DLL, ISA100.11a divides the PAN into subnets and Graph Routing and Source Routing, similar to WirelessHART, are applied. Furthermore, Backbone Routing is implemented at NL of ISA100.11a. Therefore, the network manager has to maintain an address translation table for each device in the PAN to get a relation between 16-bit DLL addresses and 128-bit NL addresses.

2.5 6LoWPAN

The IEEE foundation published 6LoWPAN with the goal to get an efficient standard for WSANs. To provide interoperability with communication networks used in the world wide web [28], a reduced kind of IPv6 is used.

The standard only specifies the NL and above, therefore different physical layer and DLL technologies, even IEEE 802.15.4, can be used for low level communication. Similar to ISA100.11a, the NL of 6LoWPAN implements the IPv6 protocol on top of the loWPAN layer. In contrast to ISA100.11a, the IPv6 header is compressed to reduce the size of the payload and to increase the efficiency. Well-known protocols like User Datagram Protocol (UDP) and ICMP are used efficiently on top of the IPv6 layer.

In general, 6LoWPAN is based on the IEEE 802.15.4 physical layer and DLL, that is the current standard in the area of WSANs (see Section 2.2). Furthermore, Powerline technology, based on 9.6 kbit/s bandwith, emulating the CSMA/CA mechanism of IEEE 802.15.4 is supported (refer to [28] and [1]).

Since IEEE 802.15.4 (or its emulation) is used at the DLL, 16-bit addresses are used at the DLL. Additionally, the IPv6 protocol at the NL requires 128-bit addresses, that are globally unique. Therefore, address translation is necessary.

6LoWPAN uses a DLL routing strategy called ”Mesh-Under” using the 16-bit addressing of the DLL. Furthermore, ”Route-Over” at the NL is supported using the corresponding 128-bit IPv6 addresses.

2.6 ZigBee

ZigBee was first published by the ZigBee Alliance in 2002. Nowadays, many products based on ZigBee exist. Application areas include HBA, industrial automation and even traffic management systems.

[1] Watteco Power line communications, `http://www.watteco.com`

2.6.1 Technological use

Wireless Private Area Networks (WPANs) often require network structures that exceed the borders of the data link specification. Therefore, routing mechanisms are used to expand the range of communication networks. Appropriate mechanisms for low power wireless network devices are defined by the ZigBee specification.

2.6.2 ZigBee basics

ZigBee provides the following advantages to Wireless Sensor and Actuator Networks:

- high reliability,
- low-power consumption,
- high security level and
- use of an open standard [17].

Wireless networks are unreliable due to their physics. Metal, water and even human bodies may block or reflect waves. Avoiding a wireless device to receive waves transmitted by a sending device is quite easy. Therefore, ZigBee devices use many different techniques to achieve higher reliability.

Security in ZigBee is based on the Advanced Encryption Standard (AES)-128 encryption. Authentication and message integrity are provided by the encryption to ensure reliable communication between two nodes. Key distribution is based on a concept of a trust center. Typically, the ZigBee Coordinator (ZC) is the designated trust center. When the trust center concept is used, all keys used for encryption between two devices are provided by the trust center. It is also possible to set pre-installation keys using commissioning tools. Among the master key, link keys and a network key exist. Link keys are point-to-point keys only known by exactly two devices. The network key is used by all devices in the network. Management commands allow the trust center to change the network key by specifying key and sequence number, identifying when the key is changed. The CCM* security suite used by ZigBee is a low-cost implementation in terms of resources. The plaintext is scrambled into ciphertext and a message integrity code is added to the message. As follows, message integrity and sender verification are provided. [10]

ZigBee is designed and distributed as an open global standard. This provides the possibility for any vendor to implement devices according to the ZigBee specification for a very low price. Furthermore, interoperability is available as long as all vendors comply to the ZigBee specification. It is a big benefit for the customers to have that wide range of ZigBee products available.

2.6.3 ZigBee device types

All ZigBee devices are either ZigBee Routers (ZRs) or ZigBee End Devices (ZEDs). One ZR per PAN is necessary to act as the ZC. The ZC and all ZRs require to have FFD capabilities (c.f. [19]). ZEDs can be either FFDs or RFD. The differences between ZRs and ZEDs mainly

affect the communication types used to interact between two devices. ZEDs are not able to forward packets on behalf of other nodes. Therefore, a ZR is always needed to integrate a ZED into a ZigBee network. Nevertheless, many ZigBee sensors are only ZEDs due to the requirements of the lower power consumption compared to ZRs.

ZigBee devices are classified regarding the tasks assigned to the specific device type. Each ZigBee networks requires one device to initiate the network, the ZC, whereas each ZR can operate as the ZC. A point-to-point network is built by one ZED associated with one ZC. At least two ZEDs associated with a ZC can build up a star network, as illustrated in Figure 2.4. Additional ZRs provide the possibility to build tree networks (see Figure 2.5), where ZEDs can communicate by using a ZR or the ZC as a packet forwarder. More complex networks are called mesh networks, when ZRs provide more than one path through the network, due to multiple connections to neighboring nodes (see Figure 2.6).

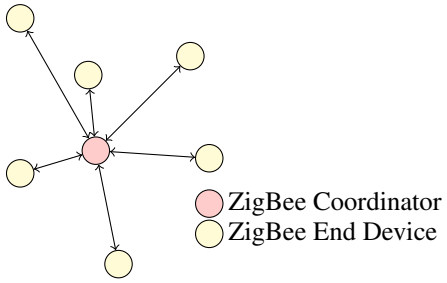

Figure 2.4: ZigBee Star network topology

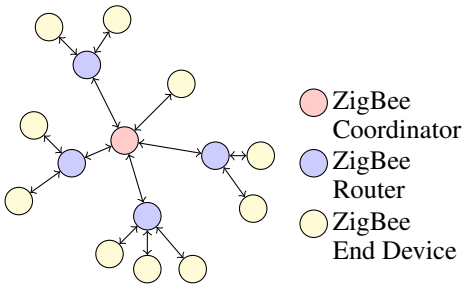

Figure 2.5: ZigBee Tree network topology

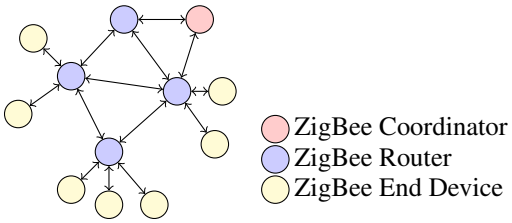

Figure 2.6: ZigBee Mesh network topology

2.6.4 ZigBee layers

Two specific layers are defined by the ZigBee specification:

- Network Layer (NL)
- Application Layer (AL)

Distributed routing, broadcasting and multicast communication are the main responsibilities of the NL. Depending on its functionalities, the NL can be split into the Network Layer Management Entity (NLME) and the Network Layer Data Entity (NLDE). The NLDE unifies the

mechanisms necessary for data exchange. In contrast, the NLME has to care about management issues required by the NLDE and mechanisms included in the NLME itself (e.g. routing, addressing).

ZigBee Device Objects (ZDOs) and ZigBee Application Profiles (ZAPs) build up the AL. The ZAPs "are agreements for messages, message formats, and processing actions" [20] to provide interoperability between multiple devices to the developer. A ZigBee device is initiated using base functionalities provided by the ZDOs (including the NL). Furthermore, management functionalities are provided by the ZDO.

Figure 2.7 illustrates a generalized view of the ZigBee layer model. The figure shows the interconnection between ZDO and ZAP as well as between NLME and NLDE. Furthermore, by using the primitive model defined by NL and AL, the two ZigBee layers can exchange information.

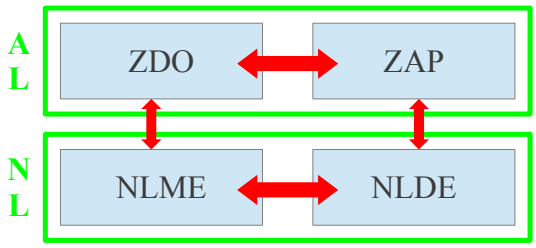

Figure 2.7: Simple ZigBee layer model

2.6.5 Addressing scheme

ZigBee uses 16-bit network addresses to uniquely identify each node in a PAN. This network addresses are automatically assigned by the coordinator or a router of the PAN. ZigBee addresses can differ from IEEE 802.15.4 addresses, since ZigBee does not rely on the IEEE 802.15.4 addressing scheme. New devices need to discover their neighborhood to find devices for association using an active scan (see Section 2.2). Association to a PAN is only possible when pairing with the ZC or a ZRs. Resulting in a list of PAN descriptors, the active scan allows the new device to choose the neighbor with the best link quality to pair. An association request indicates the purpose to join the network to the addressed ZigBee device (see Figure 2.8). The addressed ZigBee device chooses a network address and sends the response back to the new device.

Addresses are chosen using either the Distributed Address Assignment Mechanism (DAAM) or the Stochastic Address Assignment Mechanism (SAAM). When using the SAAM, addresses are chosen randomly. In contrast, when using the DAAM, each ZR (including the ZC) gets a range of addresses it may assign to its own ZR child nodes. Furthermore, an additional address range for ZEDs is assigned. Using d as the depth of the router, Cm as the maximum child count (the sum of associated ZRs and ZEDs), Rm the maximum ZR child count and Lm the maximum PAN depth, the following formula helps choosing a unique address for each child by identifying

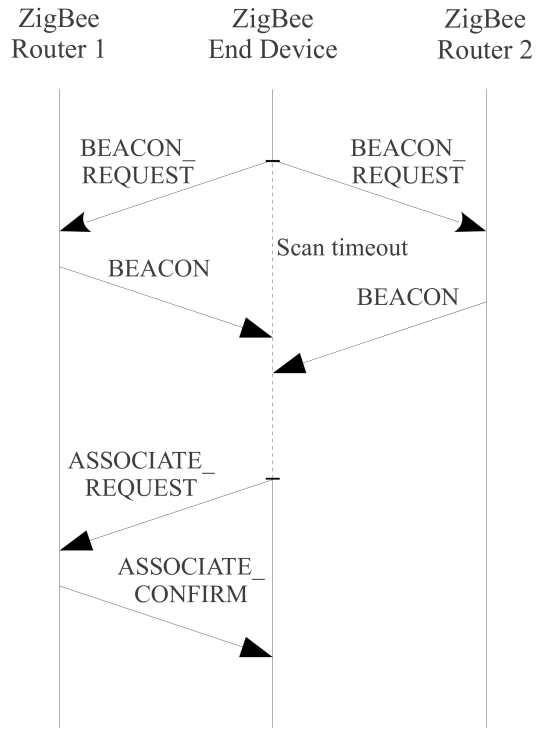

Figure 2.8: ZigBee association procedure

an address range for each ZR by defining the address interval width $Cskip(d)$ according to the depth of the router:

$$Cskip(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), & \text{if } Rm = 1 \\ \frac{1+Cm-Rm-Cm \cdot Rm^{Lm-d-1}}{1-Rm}, & \text{otherwise.} \end{cases}$$

Router addresses A_r are assigned by using the following formula where r is the count of already associated ZRs and $Cskip(d)$ as defined earlier.

$$A_r = A_{parent} + Cskip(d) \cdot r + 1$$

ZEDs get addresses A_n by using $Cskip(d)$ and Rm as defined before. The child count n of the router plus one and A_{parent} as the address of the router are used in the following calculation:

$$A_n = A_{parent} + Cskip(d) * Rm + n$$

2.6.6 Routing

The routing mechanisms of ZigBee are located at the NL. Since IEEE 802.15.4 does not have any routing capabilities, routing needs to be applied at the ZigBee NL. Data packets are encapsulated by the DLL and passed to the physical layer for transmission to a RD. This RD

decapsulates the data packet at its DLL, checks the next hop information for the dedicated destination and re-encapsulates the data packet for transmission. The destination node first has to decapsulate the packet at the DLL. Subsequently, the packet has to be decapsulated at the NL (see Figure 2.9) and finally deferred to the next higher layer.

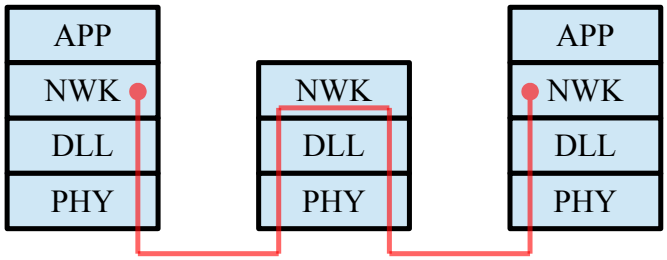

Figure 2.9: Simple ZigBee Routing model

Frames are routed by the ZC and ZRs across the PAN. ZEDs are not able to forward packets. Therefore, many ZEDs implement only RFD capabilities (refer to Section 2.2). According to different PAN structures and network requirements, different routing concepts are useful (see Chapter 3).

For static networks, Source Routing is a good possibility due to its predefined route records providing routes through the network. Source routing even shortens the time interval for a successful routing through the network.

Dynamic networks need to be capable for continued changes in the network structure. ZigBee devices can leave the range of the ZR associated with and should be able to connect to the network via another ZR. In this case, dynamic routes created by using Route Requests (RREQs) for unknown or non-working routes are much more efficient. RREQs are NL packets that are transmitted to determine the best route – w.r.t. the link available – through the network, to reach the target node. When a ZR or the ZC receives a routing request and one of its ZED child nodes or itself is the target, it shall answer with a Route reply (RREP). Each node on the way to the target increases the cumulative routing costs. Using the sum of each hop-to-hop routing cost, helps the sender to choose the most attractive path through the network. The ZigBee specification defines the Ad hoc On-Demand Distance Vector (AODV) routing algorithm. RREQs are transmitted to all devices in the network (illustrated in Figure 2.10). The corresponding device answers with a RREP packet (see Figure 2.11). Each hop that forwards the RREP packet, adds a routing information to the forwarded packet, including routing costs and next hop information. Once the RREP is received by the RREQ initiator (for the first time, or with lower route costs), the corresponding Routing Table Entry (RTE) is created or updated for further use of the route with lowest routing costs.

Transmissions across a PAN can be realized using unicast addressing, multicast addressing or broadcast addressing. Unicasts are directly destined for one device using its network address. Unicast frames are always sent by instructing the DLL layer to acknowledge the packet.

Unicasts and Broadcasts will be further discussed in Chapter 3.

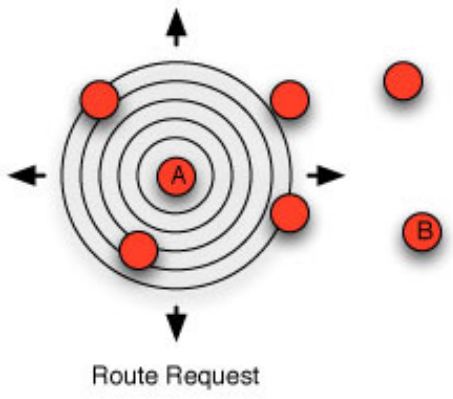

Figure 2.10: ZigBee Route Request [21]

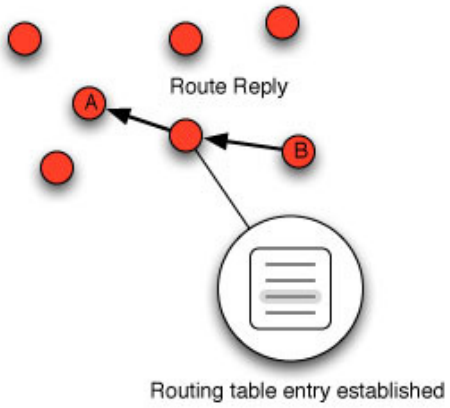

Figure 2.11: ZigBee Route reply [21]

Broadcasting

Using broadcast addressing, all nodes of the network are informed about the sent frame. Broadcast addressed frames are never acknowledged. Therefore broadcast transmissions are not reliable. The typical form of broadcasting in ZigBee networks is to flood the packet to each node in the neighborhood. Once a broadcast packet is received successfully, any further reception of the same packet is ignored. Additionally, ZigBee implements the passive acknowledgment mechanism. Whenever a broadcast is received, an entry in the broadcast transaction table is created. The broadcast transaction table stores information which neighboring device has already relayed a specific broadcast. With this knowledge, the retransmission of a broadcast is only necessary if at least one of the neighboring devices has not yet relayed the broadcast packet.

Multicast addressing means sending a frame to all members in a multicast group. Multicast groups have to be defined during network definition. Multicasts are realized by flooding the network with the packet. Each node decides whether the packet belongs to one of its child nodes or itself. In any other case, the packet is discarded.

Multicasts are a hybrid form of unicasts and broadcasts. Depending on the multicast mode used, they are relayed by all hops equally to broadcast behavior or sent by unicast until they are received by a member of the destined multicast group. Only devices being part of the destined multicast groups need to actively receive multicast packets. Receiving devices recognize multicast packets by inspecting the multicast flag in the ZigBee header. If a multicast packet is identified, the destination address is used as the multicast group ID, uniquely identifying a multicast group in the whole network. By examining its own *nwkGroupIDTable*, the receiving device decides whether it is a member of the destined multicast group. If the device is a member of the multicast group it relays the packet by broadcast and processes the contained payload. In contrast, a device not being member of the destined multicast group will relay the packet by unicast using the routing information found in its routing table. This routing information includes a next hop for the transmission to a member of the multicast group.

Most of the multicast transmissions end up in broadcast packets. Therefore, the considerations done for broadcasts in Chapter 3 can also be applied to multicasts. Regarding the transmission time, multicast packets are only transferred as fast as broadcast packets or slower, because of possible unicast transmissions between two broadcast intervals.

Many-to-one

In many use cases, there exists only one node in a PAN, that collects data of all other devices. This may be caused by having a centralized control device or one device collecting logging information. Such a collecting node is called concentrator or aggregator. Without any optimization, every device in the network needs to initiate a RREQ to get routing information towards the concentrator. Therefore, the ZigBee specification defines a routing scheme optimized for such use cases called *Many-to-one routing*. This means, the concentrator initiates a Many-to-one route discovery packet, that is broadcasted among the network. On reception of this Many-to-one route discovery packet, each RD of the network creates a RTE corresponding to the concentrator. Using this RTE, no further route discovery for the concentrator is necessary.

Similar to the Many-to-one routing concept, the concept of "island controllers" is discussed in [17]. It describes gateways (RDs) that are used to address the corresponding child devices (i.e. all devices in a hotel room or level). Routing tables of devices outside this routing areas need to store only routing information about the single gateway to be able to address all devices in the corresponding routing area.

2.7 Technology summary

Table 2.1 abstracts the presented technologies and applies them to the OSI model (as presented in Section 2.1.1). All of these protocols use IEEE 802.15.4 at least as physical layer. TDMA in combination with CSMA mechanisms as well as those mechanisms provided by IEEE 802.15.4 DLL are used to access the medium. Further differences appear on top of the DLL, where the diversity of the technologies used increases.

<table>
<tr><th>OSI Layer</th><th>WirelessHART</th><th>ISA100.11a</th><th>6LoWPAN</th><th>ZigBee</th></tr>
<tr><td>7</td><td>HART commands</td><td rowspan="3">ISA100.11a</td><td>HTTP, SSL, etc.</td><td rowspan="3">ZigBee AL</td></tr>
<tr><td>6</td><td rowspan="2">HART</td><td rowspan="2">6LoWPAN</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td><td>HART</td><td>ISA100.11a</td><td>TCP/UDP</td><td rowspan="2">ZigBee NL</td></tr>
<tr><td>3</td><td>WirelessHART</td><td>IPv6</td><td>IPv6</td></tr>
<tr><td>2</td><td>TDMA / CSMA</td><td>TDMA / CSMA</td><td rowspan="2">IEEE 802.15.4</td><td rowspan="2">IEEE 802.15.4</td></tr>
<tr><td>1</td><td>IEEE 802.15.4</td><td>IEEE 802.15.4</td></tr>
</table>

Table 2.1: Technology summary according to the OSI model

2.8 Simulation environments

Currently, there exist various different simulators for network protocols. The main key issues of those simulators are

- licensing of the source code
- user interface
- efficiency

For the simulations in the context of this thesis, it is very important to use a simulator that allows a high level of individual adjustment. Furthermore, preventing any restrictions by licenses is recommended for analyses of open standards like the IEEE standards.

2.8.1 Licensing

There exist two major types of licenses in the area of computer sciences:

- open-source licenses
- commercial licenses

Although commercial licenses save your intellectual property, restrictions in third-party support (e.g. in finding algorithms) are the consequences. That is the main argument for using an open-source license when deciding for the most suitable license for a project.

2.8.2 User interface

Simulators do not only consist of logical connections that, of course, have to be semantically correct. Some of the existing simulators provide very useful Graphical User Interfaces (GUIs) that offer many functions to improve the development of the protocol, help keeping the programming style uniformly or at least increase the overall productivity of the whole simulation process. Furthermore, statistical analyses are simplified by some functionalities of the user interfaces.

2.8.3 Efficiency

Figuring out the level of efficiency is a complex task, therefore some major points are pointed out that will be summed up to the total amount of efficiency in the end.

Implementation time: Depending on the complexity required by the input interface, the implementation time differs. It is worth if the implementation time for a model in a specific simulator framework needs more time than the implementation in a real system.

Compile time: Whenever changes are made in a simulation model, the model has to be compiled or translated again. If the compile time is high, plenty of time is left watching the compiler without being able to use the time efficiently.

Simulation time: Once the simulation model is implemented and compiled, a simulation run is required to gather information (e.g. timing, performance, a.s.o.). Simulations that last for multiples of the real-world time are not useful in most cases.

Time needed for analysis: Even the analysis of the results has to be possible in appropriate time. Simulation results that are hard to evaluate are not helpful.

2.8.4 Comparison of existing tools

With those defined key points for comparing existing tools in mind, the most interesting tools for simulation of communication protocol are discussed.

GloMoSim

The Global Mobile Information System Simulator (GloMoSim) is a discrete event-based network simulator, that was implemented using a parallel programming language called Parsec. GloMoSim was designed to simulate wireless and wired networks. [2]

Several publications concerning network simulation with GloMoSim show the possibilities given by the parallel simulation ([3], [2], [16], [33], [24]). Various protocols are implemented in GloMoSim, e.g. IEEE 802.11, IP with AODV, Transmission Control Protocol (TCP), UDP and some AL protocols.

Unfortunately, the active developement of GloMoSim has been stopped.

QualNet

QualNet is a commercial network simulator developed by SCALABLE Network Technologies, Inc. Real-time simulations as well as in-time results are provided. Up to 1000 nodes can be simulated, according to manufacturer information. [3]

In design mode, nodes can be placed in three-dimensional areas by drag & drop methods and visualizations are provided, where the current status of the simulated network is presented. The network protocols are modeled in plain text files and used as modules that can be combined to complex systems.

Simulation results are presented in table form and can be exported to Extensible Markup Language (XML) format. Therefore, analysis of the simulation results are easy-made. An example of a publication simulating IEEE 802.15.4 networks using QualNet can be found in [31].

OPNET Modeler

OPNET Modeler is a commercial simulator framework for communication networks, developed by OPNET Technologies Inc. [4] Its key features are the fast design of discrete event simulations, a huge library of already implemented protocols (including the source code), a parallel simulation engine and grid computing support for distributed simulations.

[2] http://pcl.cs.ucla.edu/projects/glomosim/

[3] http://www.scalable-networks.com/content/products/qualnet

[4] http://www.opnet.com/

Communication protocols, for example Open Shortest Path First (OSPF), IPv6 and TCP are already implemented into OPNET. The GUI provided by OPNET allows intense debugging and detailed analyses are possible. Many publications exist, using OPNET to simulate wired as well as wireless networks (e.g. [6], [25]).

The ns-2

Network Simulator 2 (ns-2) is a discrete event network simulator within a series of network simulators developed under open-source licenses. The first version of this simulator family (called ns-1) was developed in 1995 at the Lawrence Berkeley National Laboratory. In 1997 the ns-2 was released, which uses Tcl as GUI. ns-2 is available for free on the developers website. [5] Many protocols are already implemented in ns-2, which makes this simulator interesting for protocol analyses. Further work in the ns-series of network simulators is still in progress. The ns-3 was first released in October 2011, but does not provide any backward compatibility to the ns-2. Hence, all protocols already implemented to the ns-2 must be reimplemented for the ns-3.

The ns-2 has a major lack in efficiency. Each change of the implemented protocol requires a full re-compilation of the whole ns-2 binary to enable a simulation run with the new implementation. For complex protocols this is not feasible, due to the compile time.

The GUI of ns-2 as well as ns-3 consist of a Tcl/tk interface, which is hard to use for novice users.

OMNeT++

OMNeT++ is an open-source software written by András Varga and OpenSim Ltd. as module-based discrete event network simulation framework [32]. Many modules that implement different protocols are already downloadable at the developers website. [6] Its user interface is very similar to those of Eclipse [7] and therefore also provides almost all information in the main window using sidebars and subframes.

Programming the logical background of a protocol is made easy in OMNeT++ using a C++ like source code, enhanced with some OMNeT++ specific commands. Classes and class members are defined similar to C++.

As another major part of a protocol, packet structures need to be defined. For this purpose, base classes called *message* and *packet* exist globally. These two base classes represent basic packets and messages used in any protocol in telecommunication. Classes to represent specific packets or messages do not have to be written directly by the programmer. OMNeT++ provides a tool called *msgc* that compiles structures defined in Message Files (with the file extension *.msg*) into C++ classes, including functions to set and get the class members. Therefore, it is efficient to implement packet formats in OMNeT++.

Connections between different protocol layers are defined in so called Network Description Files (with the file extension *.ned*) in an easy-readable manner. Those Network Description Files can be implemented using a GUI including drag & drop functionality. As well as the connections

[5] `http://www.nsnam.org`
[6] `http://www.omnetpp.org`
[7] `http://www.eclipse.org`

between layers, multiple devices can be linked into a network using Network Description Files. For implementation details and tweaking (i.e. inserting multiple devices using loops), the source code of the Network Description Files can be edited.

OMNeT++ provides mechanisms to efficiently get statistics after each simulation run. By defining statistical variables in the source code and pushing values to those variables when necessary, scalar values as well as vectors can be analyzed. Using these statistical variables, line diagrams and histograms can be generated.

Summary

The following Table 2.2 summarizes the simulation environments presented in this chapter. Based on this comparison, the best simulation environment to simulate ZigBee networks is chosen.

Criterium	**GloMoSim**	**QualNet**	**OPNET**	**The ns-2**	**OMNeT++**
Licensing	Open-source	Commercial	Commercial	Open-souce	Open-souce
User interface	Command line	Qt – 3D	Good – 2D	Tcl/tk	Eclipse framework
Efficiency	–	High	High	Full recompile required	Only changed files recompiled
Development	stopped	ongoing	ongoing	ongoing	ongoing

Table 2.2: Summary of the simulation environments available

According to Table 2.2, the commercial simulator environments are well developed. Most of the open-source simulator environments are on a lower development level than the commercial environments. Only OMNeT++ is at the level of commercial frameworks. OMNeT++ provides the possibility to use the source of the simulation environment and all protocols already implemented. The decision to use OMNeT++ for the simulations of ZigBee networks was based on those arguments.

CHAPTER 3

ZigBee performance analysis

3.1 Scope

Network protocols can be evaluated using benchmarks revealing various limits relevant for a network protocol. ZigBee aims at building up low-power WSANs for HBA. Significant key features of network protocols are described in this chapter and ZigBee's capabilities are discussed, concerning addressing, especially limits of different addressing modes provided by the ZigBee specification.

Routing is a very complex task depending on various preconditions. These preconditions are further discussed to introduce the structure ZigBee's routing mechanisms depend on. Subsequently, influences of different addressing modes on the routing performance are presented. Starting with evaluation of limits concerning broadcast communication, channel usage and timing behavior of networks are investigated. Connection to real-world applications is established by examinations of realistic scenarios. Limits of the route discovery mechanism complete the analysis about ZigBee's routing mechanisms.

Finally, routing optimizations provided by the ZigBee specification are presented to round-up this chapter. Thereby, worst case scenarios regarding ZigBee networks are discussed and possible concepts to improve significant drawbacks of ZigBee are given.

3.2 General ZigBee issues

Whenever multiple devices have to use a single medium, an efficient access mechanism is necessary to enable high-performance concurrent communication. Therefore, CSMA/CA is used at the DLL to access the medium. Two different types of CSMA/CA, namely slotted and unslotted CSMA/CA, are defined by the DLL of the IEEE 802.15.4 standard. When using slotted CSMA/CA, information about the network, length and count of so-called slots is included in periodically sent packets (denoted as beacons, see Section 2.2). Each slot can be used by all devices. Before transmitting, each device waits for a random number of slots, allowing other

devices to access the medium. Slots that were not chosen by any devices imply idle times of the medium. Unslotted CSMA/CA forces all devices to listen to the medium to find some idle time where the medium is accessible. Idle times are only possible, when no device holds data to be sent.

Mainly influencing the interval where the medium is busy (i.e. the time when a device actively accesses the medium), the number of nodes in transmission range needs to be considered. Due to limited transmission power, the maximum range in which neighboring nodes are reachable (further denoted as ρ_T) is about 400 meters [22]. This means that the radius around a sending device where the medium is busy equals ρ_T. The network diameter of only one device equals $2 \cdot \rho_T$, in which RDs can relay packets. Therefore, the network diameter can be expanded by using multiple RDs, each increasing the range to the network by ρ_T. With a smaller distance to the first device than ρ_T, this extension is less effective. The maximum network diameter R_{max} is calculated as shown in Equation 3.1, by using d as the distance between the two devices furthest from each other (but still within the same network, interconnected by routing nodes). The maximum range can also be calculated as shown in Equation 3.2 by using n_{RD} as the number of RDs connected to the network.

$$R_{max} = d + 2 \cdot \rho_T \tag{3.1}$$

$$R_{max} = (n_{RD} + 1) \cdot \rho_T \tag{3.2}$$

Using Equation 3.2, the minimum number of RDs needed to provide a network range of R_{max} can be calculated as $n_{RD} = R_{max}/\rho_T - 1$ (see Figure 3.1).

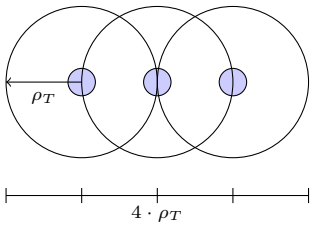

Figure 3.1: Transmission range increased by additional RDs

Table 3.1 shows the maximal possible bit rates and symbol rates of IEEE 802.15.4. The bit rate for the 2.4 GHz frequency equals 250 kBit/s, even limiting the amount of data transferable using ZigBee.

3.2.1 Network topology

Another important factor influencing the routing performance of ZigBee is the physical network topology. IEEE 802.15.4 defines star and peer-to-peer networks. Tree and mesh networks are added by the ZigBee specification.

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868-868.6	300	BPSK	20	20	Binary
	902-928	600	BPSK	40	40	Binary
2450	2400-2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

Table 3.1: Frequency bands and data rates [19]

ZigBee provides the so-called tree topology as illustrated in Figure 3.2. In contrast to the IEEE 802.15.4 peer-to-peer topology (see Section 2.2), the ZRs and the ZC build up a tree network and they are not allowed to communicate with all neighbor nodes. Direct communication is only done with their parents and all child nodes. This concept implies an easy maintainable network structure. Furthermore, unicast packets are routed by sending them to the parent node, whenever the destination node is not known as a child node. When using the DAAM, the path where a device is found is uniquely defined by its address. Similar to the star topology a single point of failure is evident. Each failing ZR causes the network to split into two independent networks. To avoid this, complex ZigBee mechanisms are able to rebuild tree structures, even when the top-most parent node is removed or fails.

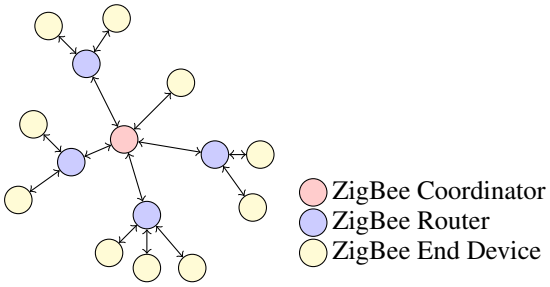

Figure 3.2: Tree network topology

ZigBee supports also mesh topologies and combines the advantages of all previously mentioned topologies (see Figure 3.3). Many ZRs and the ZC build up a network where mostly unicast communication is used. Enhancing the concept of the tree topology, each ZR can route data via all neighboring ZRs (including the ZC), not only its parent. By using route discoveries, the path providing the lowest costs, w.r.t. the link quality, is obtained as route through the network.

Routing efficiency is highly influenced by the number of nodes a packet has to tackle. Each node acquires access to the medium which influences all neighboring nodes that need access, too. The higher the number of neighboring nodes, the lower the idle times of the medium (i.e. free temporal space).

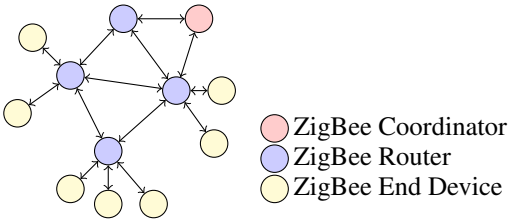

Figure 3.3: Mesh network topology

Beacons

Wireless Sensor and Actuator Networks are consisting of multiple devices that need to exchange information about the network. Furthermore, devices not associated with any network have to request information about all networks in range.

Beacons are packets sent by the MAC layer to coordinate the communication between ZR and its associated devices. According to the characteristics of a network, beacons are enabled or disabled. When discussing the limited bandwidth of wireless networks and the non-existence of synchronized clocks in the devices, the typical solution is a parent/child structure to coordinate the permissions to access the medium. The parent device sends a periodical packet to synchronize all child nodes and not yet associated devices. The access to the medium used by all devices of the network is even coordinated by these periodical packets.

Beacons include information about the network and the capabilities to associate more devices with the network. Beacons even specify time domains where child nodes need to turn on their receivers and periods where they can send data. Child nodes must not send data to their parent except time periods specified by the parent's beacon.

If beacons are not enabled, every device is allowed to send data immediately. Hence, each sending device has to ensure that all neighboring devices have their receivers enabled. This means that if some of the neighboring devices are known to turn off the receiver while idle, packets need to be transferred to those devices individually. Furthermore, each device needs to examine whether the medium is busy or access to the medium is possible.

One of the advantages of beacon enabled networks is the periodically distributed information about the network. Devices associated with beacon-enabled networks need to broadcast a beacon request packet to get information about the network. If a device wants to join the network, a beacon request packet is required to gather the necessary information.

In contrast, time is lost when each child has to wait for a beacon of its parent to send a message. Additionally, the parent submits packets to a child node only in predefined intervals. Assume for example that a network has a maximum depth of 10 – means, the root node of a tree-based network can have at most 10 successor levels – and a beacon period of one second, each RD sends one beacon per second. Thus, the transmission of one packet from the coordinator to an end device with depth 9 needs at most one second. However, the transmission of one packet from an end device at depth 9 to the coordinator takes up to 10 seconds. It is possible that the parent of each child device has sent its beacon exactly before the child receives the packet from

the node at the next lower network depth. Therefore, the child node has to wait for the next beacon to use the predefined time period for its own transmission.

Of course, when using a high number of ZEDs, the use of beacons is recommended as non-beacon enabled networks require continued polling for data by the end devices. Therefore, the network behaves more efficient if beacons are enabled for such networks.

Whenever a high number of devices can be plugged to mains or a similar supply, the use of non-beacon enabled networks results in a much higher network performance and higher throughput.

3.2.2 Addressing

Addressing is an important aspect of ZigBee as it directly influences the number of nodes that can be associated with the network. Using the same network address at two ZigBee devices results in an address conflict, that needs to be solved using additional management packets. The amount of address conflicts depends on the efficiency of the addressing mechanism. Two possible addressing mechanisms are defined by the ZigBee specification:

- Distributed Address Assignment Mechanism (DAAM),
- Stochastic Address Assignment Mechanism (SAAM).

Depending on the network size and network structure, both addressing schemes have their advantages and disadvantages. Especially the network structure can be influenced by the addressing scheme, since the network depth can be limited.

Distributed Address Assignment Mechanism

ZigBee networks using DAAM distribute address ranges to all ZR depending on their position in the hierarchical structure. Each ZR can distribute parts of the assigned address range to the child devices associated with. Therefore, addresses are split into ranges that are distributed across the ZRs in the network (see Figure 3.4) During network formation, the coordinator determines the number of routers and the maximum amount of devices that can be maintained. As long as the current tree depth is lower than the maximum depth (set by the coordinator), new child devices can be associated.

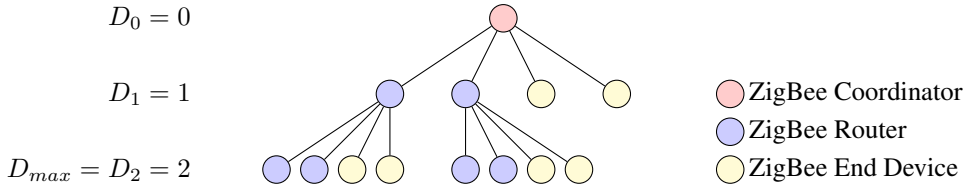

Figure 3.4: Distributed Address Assignment Mechanism

The number of addressable devices (denoted as n), can be calculated by using the depth D of a device (D in the range $[0, D_{max}]$), the number of RDs and the number of Non-routing

Devices (NRDs), n_{rd} and n_{nrd}, that can be associated with one RD. The coordinator always represents device depth $D = 0$. The total number of devices that can be associated with one node equals $n_c = n_{rd} + n_{nrd}$ (with $n_{rd}, n_{nrd} \geq 0$).

Each of the n_{rd} RDs at depth $D + 1 < D_{max}$ can also have n_c child nodes. The maximum number of addressable devices per network can be calculated using Equation 3.3.

$$\begin{aligned} n &= 1 + n_c + n_{rd} \cdot n_c + n_{rd}^2 \cdot n_c + \cdots + n_{rd}^{(D_{max}-1)} \cdot n_c \\ n &= 1 + n_c \cdot \left(1 + n_{rd} + n_{rd}^2 + \cdots + n_{rd}^{(D_{max}-1)}\right) \\ n &= \begin{cases} 1 + n_c \cdot \frac{n_{rd}^{D_{max}} - 1}{n_{rd} - 1} & \text{if } n_{rd} \neq 1 \\ 1 + n_c \cdot D_{max} & \text{else} \end{cases} \end{aligned} \tag{3.3}$$

The maximum diameter of a network (denoted as R_{max}) using the above assumptions and ρ_T as defined in Section 3.2 can be calculated with Equation 3.4. As long as the number of child nodes with routing capabilities is greater than one, the child nodes can be distributed in different directions (as illustrated in Figure 3.5), resulting in a doubled addressing range as the maximum.

$$R_{max} = \begin{cases} 2 \cdot (D_{max} + 1) \cdot \rho_T & \text{for } n_{rd} > 1 \\ (D_{max} + 1) \cdot \rho_T & \text{for } n_{rd} = 1 \\ 2 \cdot \rho_T & \text{else} \end{cases} \tag{3.4}$$

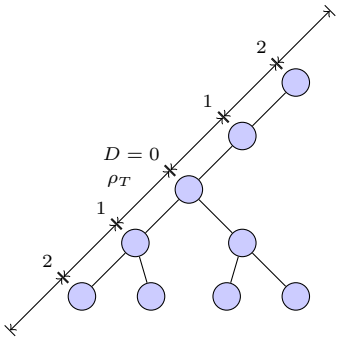

Figure 3.5: Maximum addressing range using DAAM

The following plots illustrate the possible number of addressable nodes, based on Equation 3.3. Figures 3.6, 3.7, 3.8 and 3.9 indicate the absolute number of addressable devices using a maximum depth of 3, 4, 6 and 10, respectively. The high influence of the maximum depth on the total number of addressable nodes is sketched in these plots.

Figure 3.6 shows the addressing limits assuming a maximum depth of 3. The upper bound of 2^{16} devices is only achievable if at least 40 child devices per node are allowed, all providing routing capabilities. In contrast, if 100 nodes are associated with each device, only 25 child nodes need to have routing capabilities. Allowing such a high number of overall child nodes

per device, large routing tables are required. Otherwise, a jam of the network and the following breakdown are unavoidable (according to Section 3.2.3), since route discoveries (broadcast messages) are necessary for establishing routes that are not cached in the routing table. For example, if it is assumed that $\rho_T = 400$ meters and $D_{max} = 3$, a maximum network diameter of $2 \cdot (3 + 1) \cdot 400 = 3200$ meters follows (according to Equation 3.4).

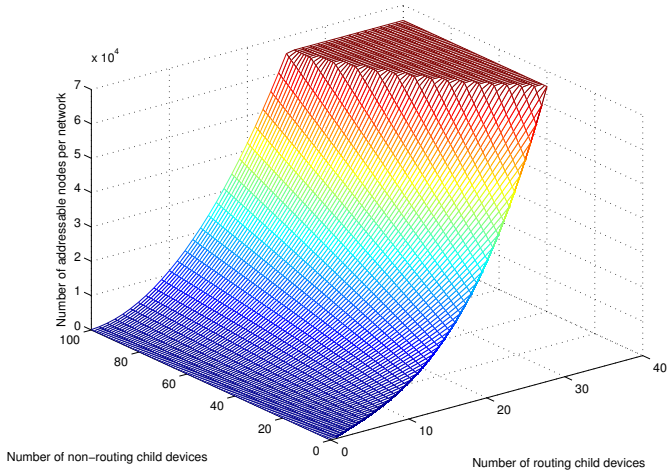

Figure 3.6: Number of addressable nodes regarding DAAM ($D_{max} = 3$)

A much faster coverage of the whole address range can be achieved by using a maximum depth of 4 (see Figure 3.7). At least 16 child nodes per device – all being routing nodes – are necessary to enable all possible addresses. Assume, for example a network with $D_{max} = 4$ and $\rho_T = 400$ meters, the network results in a maximum diameter of $2 \cdot (4 + 1) \cdot 400 = 4000$ meters by using Equation 3.4.

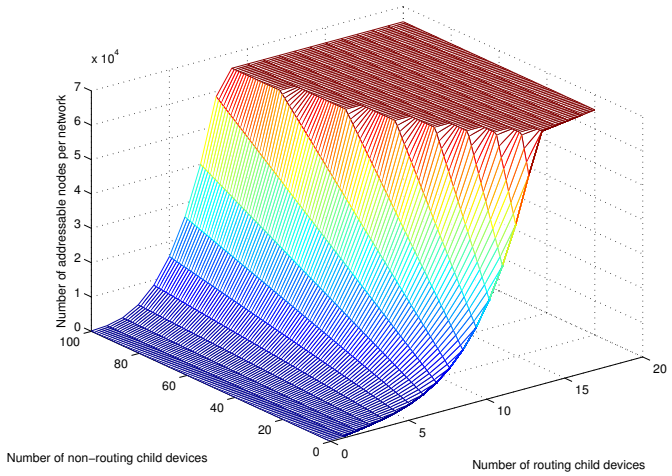

Figure 3.7: Number of addressable nodes regarding DAAM ($D_{max} = 4$)

A minimum number of 7 child nodes per device – all being routing child nodes – is needed when using a maximum depth of 6, as indicated in Figure 3.8. Still, 4 routing child nodes per device are necessary when considering 50 child nodes per device in total. According to Equation 3.4, a network with $D_{max} = 6$ and $\rho_T = 400$ meters results in a diameter of at most $2 \cdot (6 + 1) \cdot 400 = 5600$ meters. As illustrated in Figure 3.8, a low number of RDs enables the total addressing range.

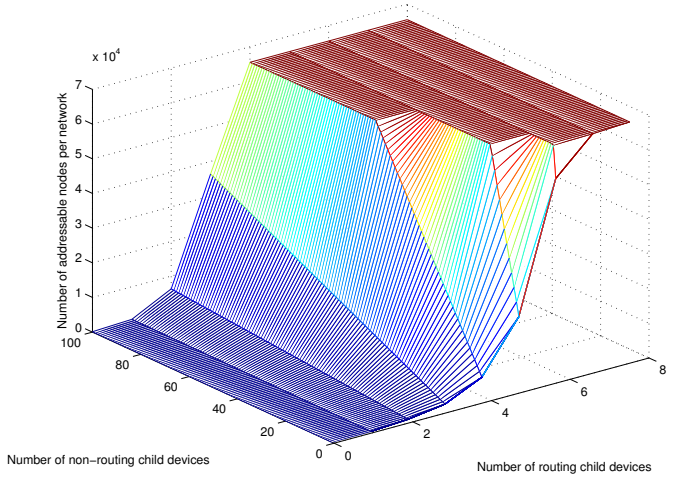

Figure 3.8: Number of addressable nodes regarding DAAM ($D_{max} = 6$)

A maximum depth of 10 provides a high number of addressable nodes by using only a low number of child nodes per device (as indicated in Figure 3.9). At least 3 child nodes per device, all being routing child nodes, are necessary to cover the whole address range. When having 100 child nodes per device, still 2 child nodes need to be routing nodes. Reconsidering the previous example having $\rho_T = 400$ meters and assuming a maximum tree depth of 10, a large network diameter of 8800 meters can be derived (according to Equation 3.4).

The transmitting range of each device is important when creating networks using most of the addressable range. Whenever the network does not perfectly connect all devices as proposed, according to DAAM defined by the ZigBee specification [20] and when using Equation 3.3, $n = n_c \cdot \left(\frac{F^{D_{max}} - 1}{F - 1} - \frac{F^d - 1}{F - 1} \right)$ devices are not able to connect to the network anymore, where d represents the depth where a connection "fault" with $d < D_{max}$ occurs. Reasons are RDs that may have reached their limit of child devices. Also blocking infrastructure as walls significantly reduce the transmission range of devices. If most of the possible addressable range is in use, the possibility to find another neighboring RD with capabilities to associate further devices converges to zero. An example as depicted in Figure 3.10 illustrates the problem where all child devices loose the connection to the main network, when the corresponding RD is no longer connected.

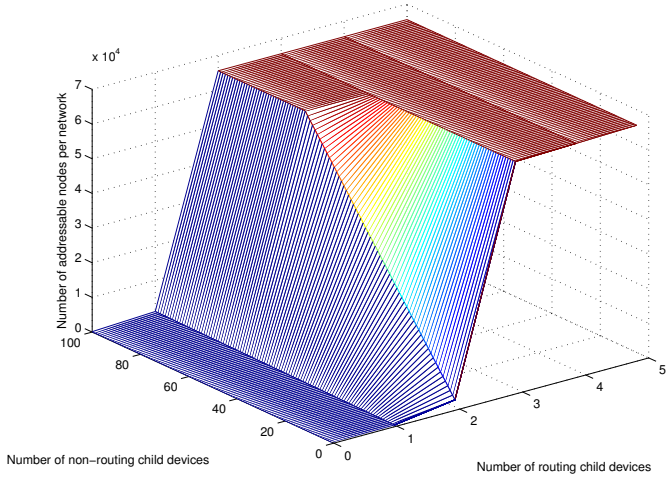

Figure 3.9: Number of addressable nodes regarding DAAM ($D_{max} = 10$)

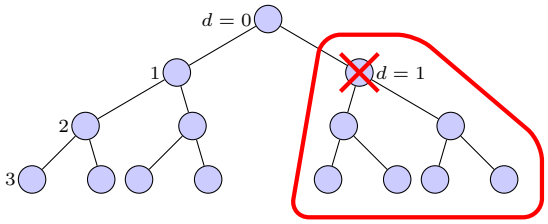

Figure 3.10: Addressing fault at depth $d = 1$

Stochastic Address Assignment Mechanism

When using SAAM, each node associated with a network obtains a randomly chosen address from its parent. Rejoining a network after moving out of the range of its former parent, causes the device to retain the previously assigned address. In case of a failure, the device chooses a random address itself. The ZC always obtains address `0x0000`, if SAAM is used. The major advantage of this addressing mechanism is the reduced processing power. In contrast, the main disadvantage of this addressing scheme is the possibility of a reuse. Address conflicts are detected by comparing the network address field in the neighbor table corresponding to the MAC address of the sender with the network address provided in each packet received. This procedure is repeated at each hop of a transmission. When an address conflict is detected, the sender is informed about its address being in conflict by broadcasting a network status command including information about the conflict. Whenever a device is informed about its own address being in conflict, the device has to obtain a new random address.

By using SAAM, all addresses are used independently of the network structure. This means that SAAM is more efficient than DAAM regarding the efficient use of the available address range. In contrast, when using SAAM, the possibility for an address conflict rises with the number of devices associated to the network. Since each associated node uses one of the available

addresses (that are 2 to the power of the bit length of the address field), the possibility for an address conflict follows as $P_{AC} = \frac{associatedNodes}{2^{addressBitLength}}$. As indicated in Figure 3.11, the possibility for an address conflict is lower than 10%, considering a network size lower than 1000 devices. For small networks, SAAM is almost as reliable as DAAM, where no address conflicts are possible on average. Furthermore, the network structure is more dynamic than in case of DAAM. Additionally, rejoining is much simpler, since no parent/child network structure has to be maintained.

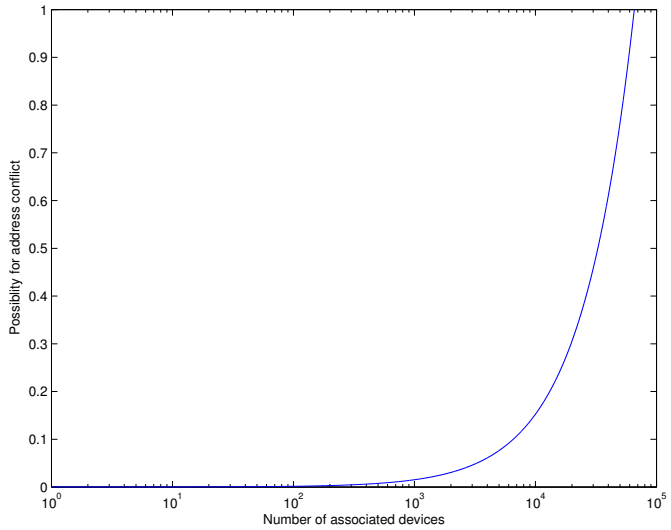

Figure 3.11: Possibility for an address conflict using SAAM

3.2.3 Broadcasting

Broadcasts are packets destined to all devices in a PAN. In general, wireless networks have bigger extents than the range of a single device pair. Thus, a special mechanism is necessary to allow communication among all devices. The easiest way to realize this is broadcasting. However, the broadcast mechanism of IEEE 802.15.4 does not support this behavior. Although, each device within transmission range of the broadcast initiator receives the packet, none of these devices retransmits the broadcast packet to ensure the transmission across the whole network. The ZigBee NL provides special addresses, intended to use as broadcast addresses. Whenever a packet with a ZigBee broadcast address is received, the ZigBee NL resends the packet to all of its neighbors, as long as the packet has not been received before.

Indicated by the ZigBee AL, the ZigBee NL of the broadcast source device receives a data request primitive including the data to be sent. Additionally, information about the methods that are used for the transmission of the packet are transferred. One of these informational arguments is the *DstAddrMode*. If this argument has a value of `0x01` and the argument *DstAddr* has a value bigger than `0xfffa`, a broadcast transmission is initiated. Depending on the *DstAddr* argument, the targets of the broadcast transmission are selected (see Table 3.2.3). The *DstAddr* argument of the NLDE primitive ends up as *DestinationAddressField* in the NL header.

Broadcast Address	Destination Group
0xffff	All devices in PAN
0xfffe	Reserved
0xfffd	*macRxOnWhenIdle* = TRUE
0xfffc	All routers and coordinators
0xfffb	Low power routers only

Table 3.2: Broadcast Addresses [20]

Subsequently, the NL sends a data request primitive containing the according payload to the MAC layer. A value of `0xffff` is used as *DstAddr* argument regarding the MAC primitive to force all receiving devices to accept the packet at the MAC layer and pass the payload to their NL. The MAC layer must not request an acknowledge for any broadcast packet initiated by ZigBee (according to [20]). Depending on the *DestinationAddressField* in the NL header, the receiving device decides whether to discard, relay and/or handle the packet.

Figure 3.12 illustrates the passive acknowledgment mechanism. The broadcast packet is initiated by NWK Neighbor 1. Since NWK Device is in transmission range of NWK Neighbor 1 the packet has to be retransmitted.

Whenever a packet is broadcasted, the source device and all relaying devices store information about the packet in the broadcast transaction table. Each Broadcast Transaction Record (BTR) includes the address of the broadcast initiating device (as *SourceAddress*) and a sequence number. Based on the comparison of *SourceAddress* and sequence number with the information included in further broadcast packets, a decision whether to relay or discard the packet can be made. Except a special case for RREQs, packets with equivalent *SourceAddress* and sequence number are relayed just once (see Section 3.2.4).

Each broadcast packet is repeated *nwkMaxBroadcastRetries* times by the source device. Before relaying a broadcast packet, a random broadcast jitter with a maximum duration of *nwkcMaxBroadcastJitter* milliseconds is inserted. Preceding each retransmission a timeout of *nwkPassiveAckTimeout* seconds is inserted. Since each device knows its neighbors (see Section 3.2.5), further retransmissions are canceled if all neighboring devices relayed the packet before the total amount of retransmissions is reached.

When a relaying device receives a broadcast packet, it decides whether it has to handle the packet. If necessary, the packet will be relayed. Similar to the procedure at the sending device, a jitter is used before retransmission. Subsequently, the packet is broadcasted *nwkMaxBroadcastRetries* times. Local handling of the packet starts after processing the timing issues for retransmission.

ZigBee denies to request an acknowledgement for broadcast packets to reduce the medium usage. Nevertheless, the successful transmission of each packet has to be proven. Therefore, ZigBee uses a so-called passive acknowledgement mechanism. Every time a broadcast packet is received, the ZigBee NL stores information about the sending device. While waiting for at most *nwkPassiveAckTimeout* seconds, the BTR corresponding to the broadcast packet is updated whenever a neighbor rebroadcasts the packet. When all neighboring devices relayed a packet,

the list of devices that relayed the packet is equivalent to the neighbor list, stored in the device (see Section 3.2.5). A retransmission is only performed, if not every neighboring device has relayed the packet after a given timeout.

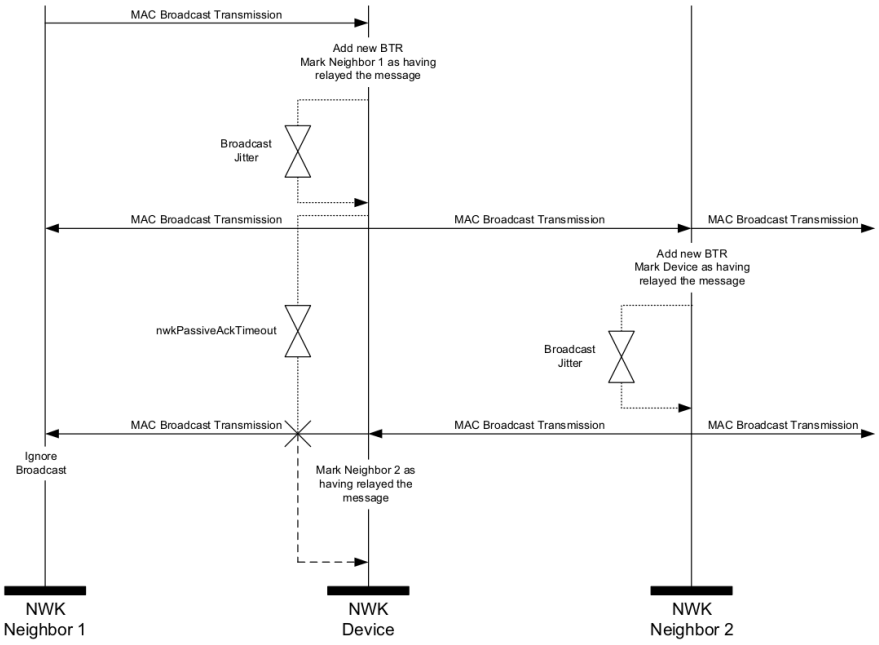

Figure 3.12: Broadcast sequence with passive acknowledgement [20]

The broadcast concept provided by the ZigBee specification defines a flooding mechanism to reach all devices associated with the PAN. This method is inefficient, since each device needs to relay the packet at least once, even if all neighboring devices already relayed the packet. Consequently, the amount of packets sent (denoted as p) to reach all devices associated to the PAN (denoted as N) using the broadcast mechanism, always results in $p \geq N$. In worst case, each device transmits the packet *nwkMaxBroadcastRetries* times (*nwkMaxBroadcastRetries* equals 3) before receiving the packet by all neighbors. As follows, the worst case of the ZigBee broadcast mechanism results in $p = 3 \cdot N$.

Broadcasts and timing constraints

Broadcasts are transmitted to all devices associated to the network. Thus, a broadcast transmission far more utilizes the medium than a unicast transmission. Hence, the efficiency of the broadcast mechanism is evaluated by examining its transmit time per broadcast packet.

The estimated transmission time per ZigBee packet (denoted as TRT) and the size of a ZigBee packet in bits (denoted as ZPS) are used to estimate the overall transmission time of

a broadcast packet. By further defining the maximum bitrate for the frequency band in use as BPS, the following relation holds:

$$TRT = ZPS \cdot \frac{1}{BPS}.$$

Using the broadcast mechanism defined by the ZigBee specification, each node has to rebroadcast the packet at least once.

As introduced in [5], assume NON as the number of neighboring nodes and the number of transmissions per node per packet (including retransmissions) as $1 \leq RBR \leq 3$. Furthermore, consider the retransmissions and the average number of neighbors, every node in the network receives $(NON + 1) \cdot RBR$ (re)broadcast messages on average (including the self-transmitted one). The average number of messages every node tries to broadcast to arbitrary receivers per second is further denoted as MPS. Since n data sources are resident in the network, the number of messages one node receives in one second can be estimated as $n \cdot MPS \cdot RBR \cdot (NON+1)$. If it is assumed that every message is received correctly and the rebroadcasts of all neighbors are received within the jitter interval, RBR can be set to 1. Subsequently, every node receives $n \cdot MPS \cdot (NON+1)$ packets per second. As follows, a channel utilization CU_{bcast} as described in Equation 3.5 is caused by broadcasts.

$$CU_{bcast} = TRT \cdot n \cdot MPS \cdot (NON + 1) \tag{3.5}$$

The previous considerations only hold, if no message has to be retransmitted, i.e., all broadcasts and relayed packets are completed within given timeouts. If a broadcast is initiated or relayed by an arbitrary node and not all known neighbors relay the packet within a given time period, the broadcast packet will be retransmitted. A reason for such a deadline miss can be a high utilization of the network. In such a case, the retransmissions cause a broadcast storm that exacerbates the already utilized network and therefore the network collapses. A network is fully utilized, if the overall duration of all messages receivable by a single node (including all protocol messages) is greater or equal than the available time (e.g., one second). For this estimation every broadcast packet has a duration of T_{bcast} on average. With these values and the previous considerations, it is possible to estimate the minimum time interval the channel is utilized. If the time interval when the channel is utilized is greater than the available time, deadlines will be missed. In following equations use the variable MPS for the messages transmitted in one second. Therefore, the channel utilization has to be lower than one to resist over-utilization. Deadline misses will further result in a chain reaction, that jams the network. Thus, if Inequation 3.6 holds, the network collapses anyway [5].

$$T_{bcast} \cdot n \cdot (NON + 1) \cdot MPS \geq 1s \tag{3.6}$$

The estimation of the transmission delay (or transmit time per packet) requires some additional parameters. First, the duration for accessing the medium using CSMA/CA has to be considered. The CSMA/CA mechanism starts with a Backoff Period (BP) to check whether the channel is idle. For unslotted CSMA/CA the BP can be computed as shown in Equation 3.7 (provided by [19]), where BE is the Backoff Exponent (BE) with $0 \leq BE \leq 5$. "Collision

avoidance is disabled during the first iteration" of the CSMA/CA algorithm, "if this value is set to 0." [19]

$$BP(BE) = (2^{BE} - 1) \cdot \text{UnitBackoffPeriod} \cdot \text{SymbolPeriod} \tag{3.7}$$

According to the IEEE 802.15.4 specification, *UnitBackoffPeriod* equals 20 symbols and *SymbolPeriod* equals $16\mu s$. Since the first BP uses $BE = 3$, the CSMA/CA mechanism needs at least $BP(3) = 2.24ms$. The probability that the channel is busy (p_{busy}) depends on the number of reachable neighbors (NON) sending MPS messages per second. Each message has a size of ZPS and uses a bitrate of BPS. Using Equation 3.6 to estimate the channel utilization, the probability for a busy medium is defined as shown in Equation 3.8, since all nodes of the network (denoted as n) are sending broadcast packets.

$$p_{busy}(n, BE) = \frac{T_{bcast} \cdot NON \cdot MPS \cdot \frac{1}{BPS} \cdot n}{1 - BP(BE)} \tag{3.8}$$

If the channel is busy, the parameter BE of the backoff duration will be increased and the CSMA/CA mechanism will initiate another BP. This procedure is repeated at most 3 times, implying a maximum value of $BE = 5$. The maximum delay results in $\sum_{i=3}^{5} BP(i) = 16.96ms$. If the CSMA/CA mechanism succeeds, the packet is transmitted. Subsequently, the channel is in use for $\frac{ZPS}{BPS}$ seconds that are added to the total transmission delay of the packet.

Since broadcast packets are not acknowledged, the time to send an acknowledge is not added to the transmission delay. In contrast, the receiving device still needs some time to compute the received packet, denoted as c. Furthermore, a variable value v is added to the transmission delay, representing a non-zero indeterministic value whenever the channel is fully utilized. Summarizing, Equation 3.9 shows the average transmit time per packet, where h is the number of hops a packet needs to travel across.

$$\begin{aligned} TRT(n) \quad = \quad & h \cdot \Big(BP(3) + p_{busy}(n,3) \cdot BP(4) \\ & + p_{busy}(n,3) \cdot p_{busy}(n,4) \cdot BP(5) \\ & + \frac{ZPS}{BPS} + c + v \Big) \end{aligned} \tag{3.9}$$

3.2.4 Routing

Routing is one of the key mechanisms of the ZigBee NL. If the routing mechanism works inefficiently, the performance of the whole network suffers. The routing mechanism and its efficiency are influenced by many factors (e.g., network size, network structure, available resources).

ZigBee uses the AODV mechanism for routing within its mesh network. The most complex part of the AODV mechanism is the route discovery. If route discovery does not cause performance losses and ready-to-use routing tables are created, further routing tasks can be handled efficiently. The following paragraphs describe the AODV route discovery mechanism and the equations used for the assessments in Section 3.3 are presented.

Route discovery is a NL mechanism to gather routing information from the network. Unknown network members, not yet existing in the routing table of a device, require RREQs to enable transmissions via additional hops.

The ZigBee specification defines route discovery as follows. Route discovery is initiated by sending a RREQ at the originating device. The RREQ packet is broadcasted to all RDs that remain their receivers on while idle. Each device that receives the RREQ packet adds the calculated path costs of the last transmission to the path cost field in the packet. Furthermore, the device searches for an entry in the route discovery table corresponding to the source address and the RREQ ID included in the packet. If there exists an appropriate entry and its forward costs are higher than the new costs, the route discovery table entry is updated and the corresponding RTE is either updated or created. A non-existing route discovery table entry will be created. Furthermore, the packet will be forwarded as a broadcast packet including the calculated new forward costs.

The final destination of the RREQ answers with a RREP. Included in this unicast packet, there are the address of the device that sent the RREQ, the RREQ ID and the path cost (equivalent to the forward cost from the RREQ sender to the destination device).

Each device on the way back to the RREQ source needs to update the path cost field by adding the calculated path cost from the sender of the RREP packet to itself. A unicast transmission is used to relay the packet to the final destination.

Memory and timing constraints

All information about routes (e.g., destination address, next hop address) can be found in the routing table. The number of routes that can be stored in the routing table of a device depends on the physical memory available for storing RTEs. It is only possible to route a packet to its destination, if the corresponding RTE for the given destination address exists. This means, whenever a requested RTE for the destination of a ZigBee packet is missing, the initiation of a RREQ is necessary.

Similar to [5], the following estimation uses the assumption that the network contains n devices with routing capabilities (including the coordinator). It is further assumed that only unicast messages are transmitted and the broadcast mechanism as well as many-to-one routing are not used for data transmissions (except for the broadcast packets for route discovery). In this scenario, every device has exactly RTE RTEs. The number of direct neighbors is denoted as NON. These neighbors do not need route discoveries to be reachable. The total number of nodes in the network is denoted as n. Therefore, the probability for a route discovery is defined by Equation 3.10.

$$P_{RD} = max(0, \frac{n - NON - RTE}{n}) \tag{3.10}$$

Furthermore, RBR is defined as the average number of retransmits of a broadcast message by a single node. Due to the passive acknowledgment mechanism, every device has to retransmit a received broadcast message at least once and therefore $RBR \geq 1$. Depending on the physical structure of a network, every node has a different number of reachable neighbors, influencing the necessity of retransmissions.

The transmission time (further denoted as TRT) is the time duration a packets needs from the sending device until it is received at the final destination. The continued CSMA/CA medium access tries at each hop are not included in TRT, as they do not influence the channel utilization. To estimate the transmission time per ZigBee packet the size of the ZigBee packet in bits

(denoted as ZPS) is used. Using the maximum bitrate for the 2.4 GHz band of 250 kbit/s as BPS results in Equation 3.11.

$$TRT = ZPS \cdot \frac{1}{BPS} \tag{3.11}$$

If no RTE corresponding to the destination device is found in the routing table, a RREQ has to be broadcast to all nodes of the PAN, using the ZigBee broadcast mechanism (see Section 3.2.3). The number of neighboring nodes (denoted as NON) is used as defined earlier in this section. The time-depending probability that the channel is busy $p_{busy}(n,t)$ (refer to Equation 3.8) is an indeterministic variable, depending on the current channel utilization, receiver power and external influences. Furthermore, the probability that one node does not rebroadcast the packet three times is defined by Equation 3.12. The total probability is a sum of three probabilities (and parts of it products of probabilities) as the packet has to be rebroadcast only once.

$$p_{NRB} = p_{busy}(n,3) \cdot \Big(1 + p_{busy}(n,4) \cdot \big(1 + p_{busy}(n,5)\big)\Big) \tag{3.12}$$

Therefore, a node rebroadcasts within the timeout with a probability of $1 - p_{NRB}$. Subsequently, the probability that at least one of the neighbors has not yet rebroadcasted the packet and therefore the packet needs to be re-sent is defined by Equation 3.13.

$$p_{RBR} = 1 - (1 - p_{NRB})^{NON} \tag{3.13}$$

If a RREQ is necessary, each neighboring node needs to rebroadcast the packet. As a result, the channel utilization (CUP) necessary for sending one unicast packet is defined by Equation 3.14.

$$CUP = TRT + p_{RD} \cdot (1 + NON) \cdot TRT \cdot (1 + p_{RBR}) \tag{3.14}$$

3.2.5 Optimizing mechanisms

Basically, ZigBee uses an abstraction of the AODV algorithm as published in [27]. The AODV algorithm is based on the transmission of RREQs that are broadcast among the network. The destination of the RREQ answers to each request with a RREP packet as long as the forward path costs included in the RREQ are smaller than the currently stored forward path costs. The destination device always overwrites the stored path costs with better ones. Each RREP is sent as unicast packet to the source device of the RREQ by using the route stored in the devices on the way. The source device stores the next hop address included in the RREP into its routing table, if the path costs are lower than the currently stored path costs. After a static timeout, the source device accepts the RTE and sends its packet to the according next hop.

As this is an inefficient routing method, the ZigBee specification defines some routing optimizations. In this section, the many-to-one concept, neighbor tables and link status messages will be described and their efficiency will be discussed. Since all of these routing optimizations require memory, the optimization level is limited.

Neighbor table

ZigBee uses neighbor tables to reduce the total amount of RREQs required. Neighbor tables include information about each neighbor to enable network joining and rejoining as well as routing

information. The routing mechanism is optimized by using the address of the corresponding device as part of a neighbor table entry. Furthermore, forward costs via according neighbor and an aging field are stored in the neighbor table entry. The neighbor tables are used to gain an overview of the immediate neighbors and to enable the self-connecting network. This means, each packet originating from a device not in the neighbor table causes the creation of a new entry in the neighbor table.

Link status messages

Building WSANs, requires considerations about the dynamic characteristic of wireless networks. They foster the ability of the devices movement. To be aware of ongoing changes in the network structure, ZCs and ZRs continuously send link status messages. Status messages are sent as one-hop broadcast (to all neighbors in range) to give them a hint of their current position and the link quality available. These non-acknowledged messages are received by each device in range and therefore help to keep the neighbor tables up-to-date. When a device does not get any link status message of one specific neighbor within *nwkRouterAgeLimit* · *nwkLinkStatusPeriod* seconds (per default $3 \cdot 15 = 45$ seconds), it resets the forward cost field of the corresponding neighbor table entry. Hence, the corresponding neighbor is not used to route packets anymore and the device can overwrite the neighbor table entry if necessary. Therefore, link status messages ensure, that all neighbor tables are up-to-date within a time period of less than one minute per default. The time period can be raised up to 65025 seconds depending on the according scenario. This sparse time period corresponds to an update of each neighbor table within about eighteen hours. In practice, it is not recommended for networks with devices installed at fixed positions.

Resources

Embedded devices do not have unlimited resources (e.g., memory) available. This holds even for most ZigBee devices. Actually, the size of the available memory for storing routing information in a ZigBee device can extensively influence the ZigBee routing capabilities. In detail, the routing table contains information to send a packet to a non-neighboring device. Whenever the RTE for a specific destination address is missing, a route discovery is initiated. As follows, the relation between the actually addressed destination addresses (AAD) and the number of RTEs that can be stored, determines whether at most AAD route discoveries are needed to infinitely communicate with all destination nodes or the number of route discoveries increases infinitely. The actual amount of route discoveries, according to different network topologies, is simulated in Section 4.3.2.

3.2.6 Worst case scenarios

Worst case scenarios represent situations where the worst parameters are used. The following subsections discuss possible worst case scenarios for ZigBee networks and try to give some possibilities to overcome those situations.

High network density

Assume, for example, a small area with a diameter of at most 100 meters like a small factory or a flat. Due to the small diameter of the network, all devices are pairwise connected, while using typical transmission power. This implies, that each transmitted packet influences the channel utilization of the whole network. In contrast, every packet can be directly delivered to its final destination without routing. Hence, the neighbor tables require much memory in the devices. Otherwise, the devices will not be able to store information about all neighboring nodes. Considering peak-load scenarios, the network will be jammed within less than one second, as each packet increases the channel utilization dramatically.

One possibility to overcome this issue is to reduce the transmission power of the devices, implying a reduced range for all devices. Another option is to choose different transmission channels for some groups of devices, i.e., using disjoint ZigBee PANs. This reduces the number of devices influencing the channel utilization by a factor of 3, but induces the need to interconnect ZigBee networks.

High amount of packets

According to the addressing scheme of ZigBee, up to 2^{16} devices per PAN are possible. This number increases when the number of possible PANs per channel is considered. The higher the number of devices is, the higher is the amount of packets sent within a PAN. Examined in Section 3.3, existing limits regarding the amount of transmittable packets are shown.

Many small packets

Consider a high number of associated devices, all of them sending many small packets (the smallest possible data frames in ZigBee have a size of 8 bytes of ZigBee information plus 15 bytes for the IEEE 802.15.4 header). Thus, each packet needs 0.736 ms to be transmitted. Assuming that 50 devices are in range, each device can send only 27 packets in one second without over-utilizing the network.

A solution for this problem is to check the protocol, whether this high number of messages is necessary. Another approach can be to reduce the number of devices in range by reducing the transmission power or by splitting the devices into networks using different channels.

Huge packets

Assuming an application protocol that needs to transmit huge packets (i.e., multimedia applications) continuously. The highest amount of ZigBee payload allowed by the MAC layer to be included in one data frame equals 25 bytes. Thus, the transmission of one packet needs $1.536ms$ and a maximum number of 651 packets can be transmitted in one second before over-utilizing the network. In total, the maximum amount of application data that can be transmitted in one second equals 16275 *bytes* ≈ 16 *KByte*.

Single line network

Of course, even networks where each device has exactly two neighbors can be built. Figure 3.13 illustrates such a single line network. The missing redundancy leads to many cases where the single point of failure (that can be each device of the network) splits the whole network into two parts. It follows that devices within one part of this split network are no longer able to communicate with any device of the other part.

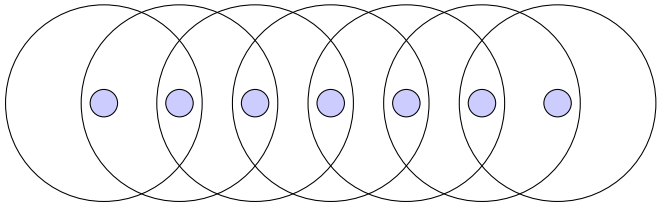

Figure 3.13: Concept for a single line network

3.3 Estimations

The previous section examined the attributes of ZigBee and its protocol structure. In this section, the performance of ZigBee and its underlying mechanisms are examined. Therefore, some generalized networks structures are defined. Furthermore, diagrams show the affect of formally defined properties and reveal the limits of ZigBee.

3.3.1 Network structures

Since each network structure is unique, abstract network structures are developed to analyze the behavior of ZigBee. Furthermore, those network structures allow to examine worst case scenarios as presented in Section 3.2.6.

The network structure influences the performance of the wireless network. Therefore, different network structures are examined. The following assumptions allow to examine the performance of ZigBee with a reduced complexity. Assuming a non-dynamic network, where no moving devices are allowed. Once all devices are connected to the network, routing conditions will not change anymore. Using this static structure, different possible concepts are examined.

Square networks as illustrated in Figure 3.14 represent typical networks as used in building automation or industrial automation. Multiple devices can communicate directly, hence many routes within the network are possible. In contrast, line networks as illustrated in Figure 3.16 exist whenever networks span across a long distance. This section examines both network structures and reveals advantages and disadvantages.

Square networks

Assuming a network of $m \times m = n$ nodes, each device (except the outer nodes) is having the same distance to four neighboring nodes. As illustrated in Figure 3.14, the distance between nodes of a square network is defined as shown by Equations 3.15 and 3.16.

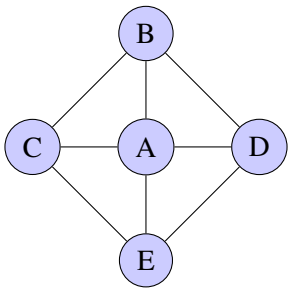

Figure 3.14: Square network structure distance concept

$$\overline{AB} = \overline{AC} = \overline{AD} = \overline{AE} = a \tag{3.15}$$

$$\overline{BC} = \overline{BD} = \overline{CE} = \overline{DE} = \sqrt{2} \cdot a \tag{3.16}$$

This structure will end up in a square structure, where all devices (except the nodes at the border) have the same number of direct neighbors. This type of network exists for any network where the associated nodes (e.g. sensors, actuators) are uniformly distributed in a room or building. Figure 3.15 illustrates the square network structure concept.

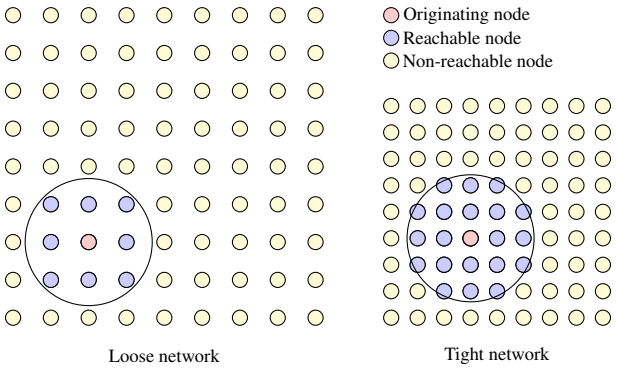

Figure 3.15: Square network structure concept

Line networks

Another possible network structure is the line structure. In this scenario, many nodes are connected to each other only in one dimension or with at most one pair node (e.g. railway, highway).

This results in a line where at least every other node needs to route all packets to enable connectivity between all associated devices. Here, each data packet transferred from one end of the line to the other end of the network needs to be relayed by at least half of all nodes in between. Depending on the distance between the nodes, a high number of transmissions is necessary. Figure 3.16 illustrates the line network structure concept.

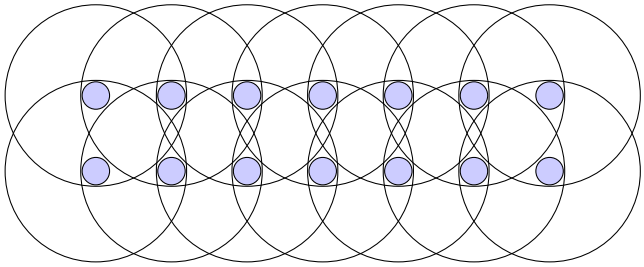

Figure 3.16: Line network structure concept

Tight and loose networks

In this chapter the terms loose and tight network are used to denote network structures, where one node has few or many connections to neighboring nodes, respectively. Few neighbors means a minimal amount of neighboring devices. In a square network, few neighbors means to have 4 or at most 8 neighboring nodes that receive each transmitted packet. A node in a square network has many neighbors, if 20 neighboring devices can be used to build up the mesh network. Loose line networks have one or at most two neighbors on each side (see Figure 3.17). In contrast, a device in a tight line network has at least 4 neighbors on each side, still having only 8 neighbors in total.

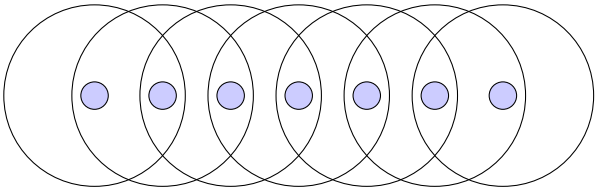

Figure 3.17: Loose line network structure concept

3.3.2 Estimated behavior of broadcasting ZigBee networks

The following subsection discusses the key features of the ZigBee broadcasting mechanism. According to Equation 3.5, the channel utilization of broadcasting ZigBee networks is examined. Broadcasting in this context means, that only broadcast packets are transmitted. Regarding the time domain, the transmission delay (as defined by Equation 3.9) is examined.

Loose square networks

Assuming a square network – as defined in Section 3.3.1 – of $m \times m = n$ nodes, having n as the PAN-wide node count and m as the number of rows and columns of nodes. Each node in the network continuously sends MPS broadcast messages in one second.

The estimated channel utilization is an indicator, how much data can be transferred using a single channel. Whenever the channel utilization has a value greater than or equal to 1, the channel is fully utilized and no further communication is possible. A network, not able to send any further packets, is said to be jammed. As the estimated channel utilization defines a fixed limit for a network, real-world networks are always built with respect to the channel utilization. Furthermore, real-world networks will never reach the estimated limit, as they are jamming earlier because of suboptimal use of the channel (due to fragmentation). Furthermore, external influences disallow a perfect use.

The following plots are generated according to Equation 3.9. As illustrated in Figure 3.18, a loose ZigBee channel (having a maximum of 8 neighboring nodes per device) is already fully utilized when having 4 members, all sending 20 broadcast messages in one second. This fact is caused by the high amount of packets needed to be transferred when sending a broadcast packet. This holds even for a low count of devices associated with the network. When the number of associated devices raises up to a number of 20, only 8 broadcast messages per second will fully utilize the channel. Follows, a network consisting of 20 nodes, all sending 8 or more broadcast messages in one second will jam the network. Hence, no further communication will be possible. Moreover, 45 associated devices, all sending 2 broadcast messages per second will lead to a network jam.

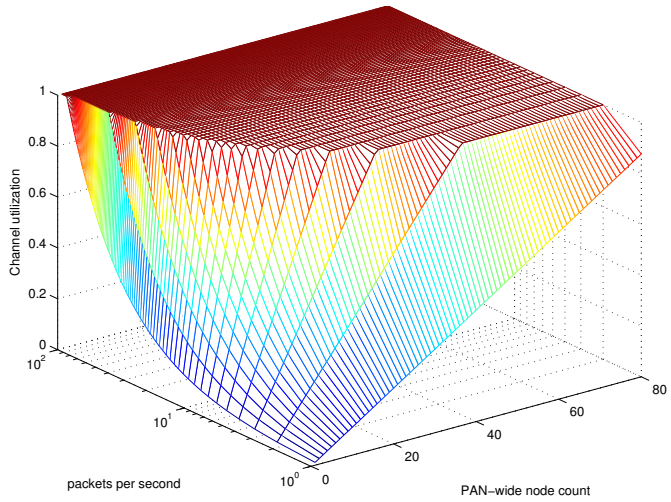

Figure 3.18: Estimated broadcast channel utilization for a loose square network

The estimated transmission delay of loose square networks is estimated and plotted in Figure 3.19. Due to the indeterminism of wireless networks, the estimations are based on worst cases. The estimated transmission time is split into two areas: exponentially raising area and in-

deterministic area. The exponentially raising area can be found where low numbers of nodes and low numbers of messages transmitted in one second are represented. In this area, the transmit time is exponentially growing, since its mainly influenced by the delay caused by the CSMA/CA mechanism. Especially an increasing number of nodes per network causes the transmit time to grow faster. The reason for this effect is the number of unsuccessful CSMA/CA BPs, which are more influenced by the number of hops (and therefore indirectly by the number of nodes) than by the number of messages transmitted in one second. The transition from the exponentially raising to the indeterministic area is constituted where the CSMA/CA mechanism is unsuccessful for more than three times. This even defines the upper bound for the CSMA/CA mechanism to use positive possibilities to provide a successful medium access. Outside these borders the medium is fully utilized. Therefore, this area is called indeterministic area. In fully utilized real-world networks it is impossible to know, whether or which device gets the possibility to access the medium. Figure 3.19 is limited to a maximum transmission time of one second, since the indeterministic area starts below this limit.

The lowest point where the earlier described transition occurs, can be found at $0.15s$ transmission time for networks having 4 members and sending 100 messages per second. A reduced number of messages transmitted in one second results in better values, where 20 nodes can send 10 messages per second while having an average transmit time of $0.3s$. Networks consisting of 80 nodes can send up to 2 messages per second with an average transmit time of $0.6s$. One message per second can be sent by 150 nodes per network, resulting in an average transmit time of $0.9s$.

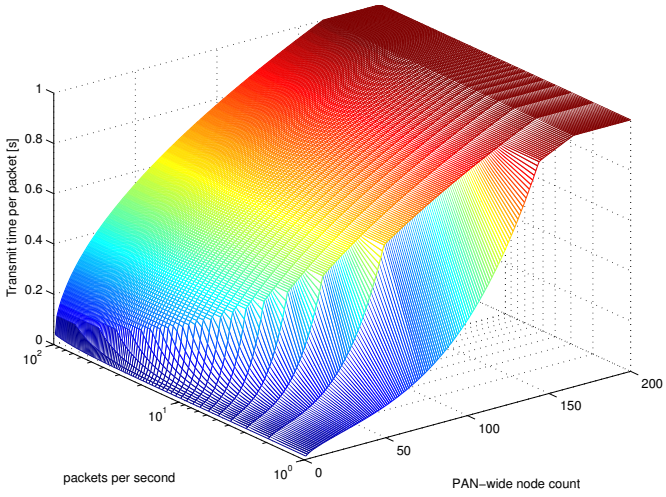

Figure 3.19: Estimated transmission delay for a loose square network

Tight square networks

Reducing the distance between the devices results in a lower amount of packets being transferred without a network jam. Assuming 20 neighboring nodes per device, the number of nodes per

square unit increases essentially. The estimations predict an earlier point, where the channel is fully utilized, as illustrated in Figure 3.20. Whereas for the loose square network, the outer limits for node count and packets transmitted in one second come to 100 both, for the tight network it results in limits of 40 and 130, respectively. A number of 20 associated nodes, all sending 2 broadcast packets per second jams the surrounding medium. Thus, no further transmissions are possible. This holds even for 10 associated nodes and 3 broadcast packets transmitted per node in one second.

Whenever the number of associated nodes is higher than 7, the most influencing factor of the channel utilization is the number of broadcast packets transmitted. Additionally, networks having a structure similar to the described and estimated square network, will almost always have a higher number of nodes. Therefore, broadcast packets have to be prevented as much as possible, since they can inhibit packets to be transmitted. Section 3.2.4 discusses the number of broadcast packets necessary for route discovery. Possibilities to prevent these broadcast packets as well as effects of wrong use of routing resources will be shown.

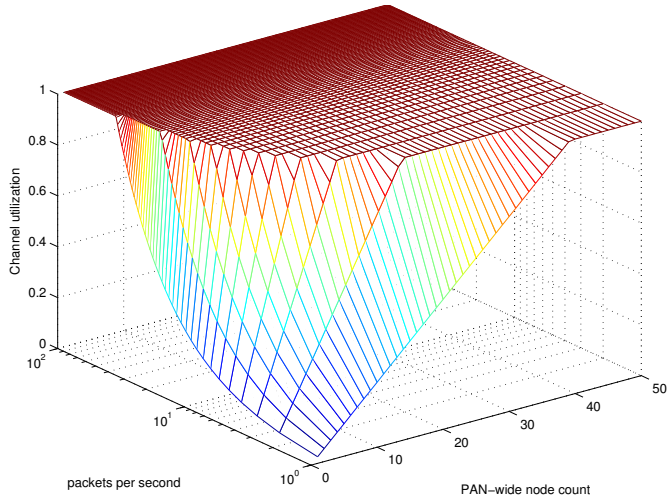

Figure 3.20: Estimated broadcast channel utilization for a tight square network

Square networks, where each node has 20 neighbors, provide a low number of deterministic transmit times, where the channel utilization is lower than one second. This means, transmit times can only be predicted for networks having less than 50 devices, all sending less then 60 messages per second, as illustrated in Figure 3.21. For networks with 50 devices, transmit times of $0.6s$ can be predicted as long as only one message is sent in one second per device. When reducing the number of nodes per network, 25 nodes can send up to 4 messages in one second, resulting in a transmit time of $0.3s$ on average. Further reduction leads to 5 nodes per network that can send up to 10 messages per second, resulting in a delay of $0.2s$ on average.

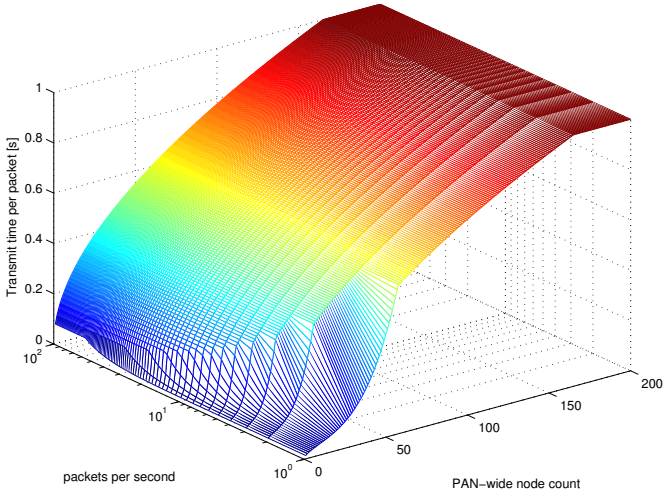

Figure 3.21: Estimated transmission delay for a tight square network

Loose line networks

Networks having a line structure – as described in Section 3.3.1 – are not differing that much to square networks. The channel utilization of a line structured network equals the channel utilization of a square network. Having an equal number of neighboring nodes, the same total number of network devices and the same amount of messages sent per node in one second, since no influencing factor differs.

Nevertheless, when having a loose line network where each device has only 4 neighboring devices, results – unrelated to square networks – occur. As illustrated in Figure 3.22, the low number of neighboring nodes allows for a higher number of network devices in total and a higher amount of messages per node in one second. On the one hand, up to 150 devices can transmit one message per second until the full utilization of the channel occurs. On the other hand, a total network size of 8 devices allows all devices to send up to 200 messages per second. Examining the results at the point where the network has a total amount of 25 devices, 6 messages per second can be sent. Figure 3.22 illustrates, that the number of messages sent per second influences the channel utilization much more, than the number of network devices. Although, the network diameter almost corresponds to the number of network devices, whereas the network diameter of square networks only corresponds to the square root of the total number of network devices.

The lower number of neighboring nodes in line networks compared to square networks allows more communication of one node in its wireless communication area. As illustrated in Figure 3.23, networks sending less than 20 messages per second are always in a deterministic state, where the CSMA/CA mechanism finishes successfully. Successful CSMA/CA medium access tries imply a higher reliability of these networks since the possibility for each packet to reach its final destination is higher than for packets that are not relayed after three unsuccessful medium access tries. Furthermore, the total transmission time – from the originator to the final destination – is shorter, if the first CSMA/CA medium access try at each hop is successful.

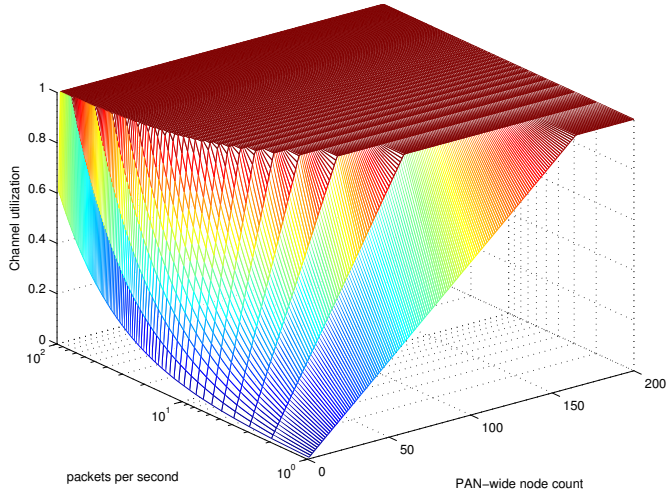

Figure 3.22: Estimated broadcast channel utilization for a loose line network

Therefore, time-critical packets have to be sent over networks and paths providing under-utilized channels only. The plot is limited to a transmit time of one second, since longer delays are irrelevant for ZigBee transmissions. One of the big points of the figure is to show that 100 network devices can send one message per second, still having transmission times lower than or equal to 1 second. Furthermore, networks built up by 40 devices can send up to 5 messages per second.

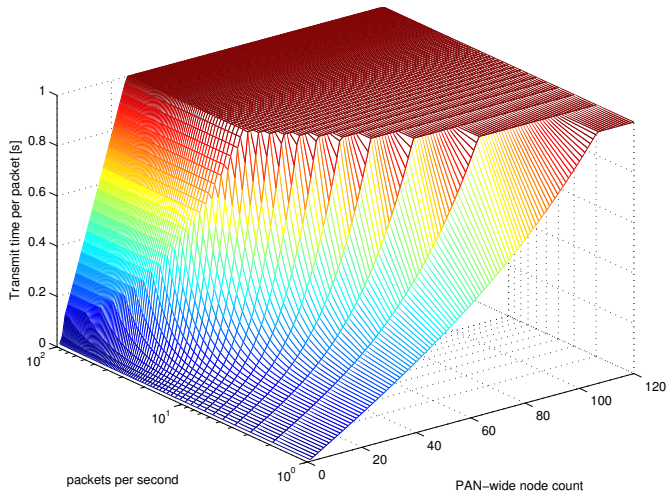

Figure 3.23: Estimated transmission delay for a loose line network

Tight line networks

As mentioned earlier in this section, independent from the structure of a network, as long as the number of neighboring nodes, the total number of network devices and the amount of messages sent per node in one second are equal, even the channel utilization equals. Therefore, all descriptions given for Figure 3.18 even hold for a network with a line structure, having 8 neighboring nodes per device.

In contrast, the transmission times can totally differ, since they mainly depend on the structure of the network. Figure 3.24 illustrates a capability of 80 devices sending one message per second with transmission delays of at most one second. As follows, all messages reach the final destination before the initiation of the next transmission. Networks of 50 devices can send 2 messages per second to get the same result. At the number of 5 messages per second sent by 25 devices, the limit for the under-utilization is reached. A further increase of messages per second would result in over-utilization of the network and subsequently would cause indeterministic transmission delays.

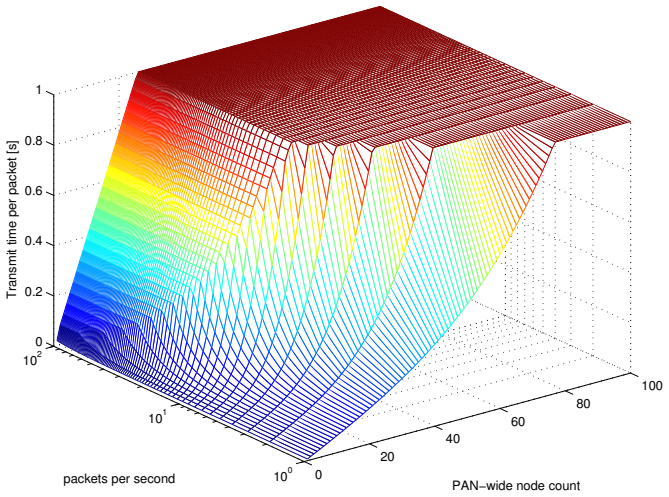

Figure 3.24: Estimated transmission delay for a tight line network

3.3.3 Estimated limits regarding Route Requests

The following section examines the behavior of the route discovery mechanism of ZigBee regarding the channel utilization. The channel utilization is estimated based on Equation 3.14. Furthermore, this section reveals the limits of ZigBee networks regarding network size and amount of messages that can be sent within one second without jamming the network.

Loose square networks

Based on Equation 3.14, network structures as defined in Section 3.3.1 are examined according to their channel utilization behavior. The number of direct addressable nodes is defined as the

sum of the number of neighbors per node and the number of RTEs storable in the routing table of the device. Estimations of the channel utilization of loose square networks, continuously sending route discoveries, indicate a good network performance as long as the number of direct addressable nodes is smaller than or equal to the total number of nodes in the network (see Figure 3.25). Whenever the number of nodes in the network is higher, performance losses follow similarly to the results of simple broadcasting networks (as discussed in Section 3.2.3).

Loose square networks with 8 neighbors per node and a routing table consisting of 10 entries result in channel utilizations where a good performance can be guaranteed as long as the number of nodes in the network is smaller or equal to 18. For these small-sized networks, more than 30 route discoveries are sent per node without a jam of the whole network. Between 18 and 40 nodes per network, the results show big performance losses, where each additional node causes a reduction of up to 10 possible route discoveries per second. For node numbers higher than 40 the performance loss shrinks. Further examinations show that networks, consisting of 100 nodes, allow only two route discoveries per second without causing the network jams.

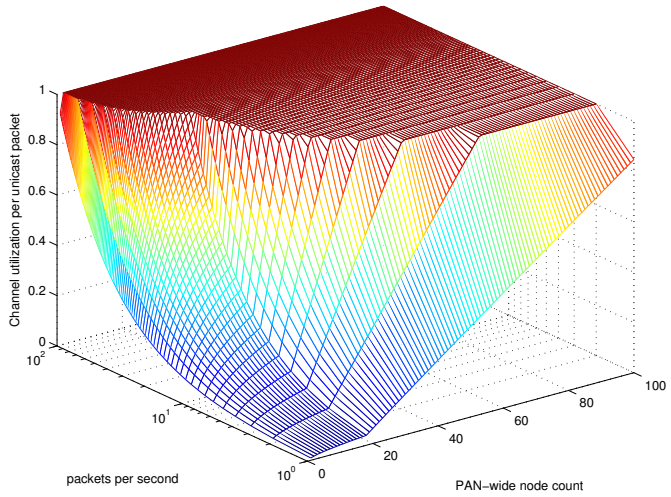

Figure 3.25: Estimated route discovery channel utilization for a loose square network

Tight square networks

When increasing the number of neighbors per node, the number of possible route discoveries reduces. As illustrated in Figure 3.26, assuming a network with 20 neighbors per node and routing tables with 10 entries, networks with a node count less than 30 can send more than 10 route discoveries in one second per node. Whenever more than 30 nodes are associated to the network, the performance reduces much faster than for loose square networks. Defining an upper limit, networks with 46 nodes can send only two route discoveries in one second per node without a network jam. More than 65 nodes per network allow less than one route discovery per node in one second.

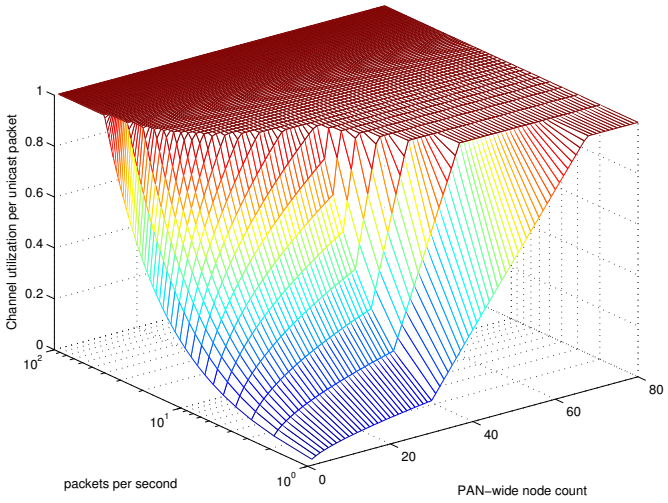

Figure 3.26: Estimated route discovery channel utilization for a tight square network

Loose line networks

Comparing square networks and line networks shows the advantages of a lower number of neighbors per node. Figure 3.27 shows the estimated route discovery channel utilization of regarding line networks having 4 neighbors per node and routing tables with 10 entries. Similar to previous results, as long as the number of nodes per network is smaller than the sum of neighbors per node and routing tables per node, the channel is minor utilized. As long as this holds, route discoveries up to a number of 100 in one second per node can be sent without jamming the network. Even the reduction of the performance, caused by a further increase of nodes per network, is smoother than for square networks. Still, the main performance loss follows between 14 nodes per network (the sum of RTEs and neighbors per node) and 70 nodes per network (where only 3 route discoveries can be transferred per node in one second). Line networks consisting of 200 nodes (not visible in the plot), already can provide a wide network range of up to $80.4km$ (according to Section 3.2.2). They can still send up to one route discovery per node in one second without causing a network jam.

Tight line networks

Line and square networks differ in many points. However, an equivalence of the main parameters, namely the number of nodes, messages per second per node and number of nodes per network causes line and square networks to behave similar. Therefore, the estimation shows the same results, due to equal parameters. Hence, nodes of line networks, where each node has 8 neighbors and 10 RTEs, can send the same amount of route discoveries per node in one second according to the total number of nodes in the network as square networks with the same parameters do (see Figure 3.25).

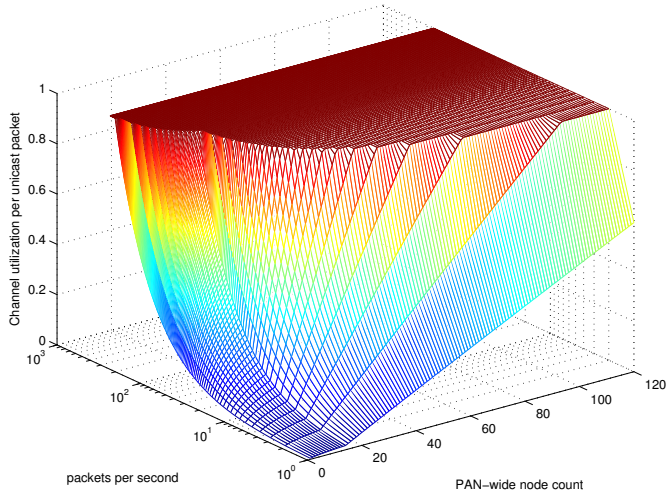

Figure 3.27: Estimated route discovery channel utilization for a loose line network

CHAPTER 4

Evaluation

4.1 General

The purpose of this chapter is to evaluate the results gained so far. Due to the indeterminism of WSANs, a complete formal proof is not feasible. Therefore, this is basically done by comparing the results of Chapter 3 by means of simulation. In this thesis, the networks structures described in Chapter 3 are simulated w.r.t the ZigBee standard, followed by a comparison of the simulation results with the analytical results of Chapter 3 to prove the concepts. The aim of this comparison is not to achieve perfectly matching results. Obviously, this is not possible due to the abstraction and simplification of the analytical approach in Chapter 3. As long as an overall match exists, the derived concepts are considered to be valid.

The decision for the simulation and against a real-world test system is provided by the apparent differences. Simulation models can be scaled dynamically, whereas physical hardware requires more effort in resources. Another argument was the higher transparency and better logging capabilities provided by a simulation framework.

To realize the simulation, OMNeT++ is used (see Section 2.8). The different scenarios are implemented using OMNeT++ network description files which are dynamically controlled using initialization files. As introduced in [8], the IEEE 802.15.4 library was used in combination with the ZigBee implementation realized as part of this master thesis.

4.2 Simulation framework

The results of Chapter 3 are validated by using a simulation model of the ZigBee stack implemented to OMNeT++. While using the implementation of the IEEE 802.15.4 physical layer offered by the INETMANET[1] bundle of OMNeT++, the IEEE 802.15.4 DLL provided by Chen [8] completes the protocol layers below the ZigBee stack. On top of the DLL, the ZigBee NL (in-

[1] http://inet.omnetpp.org/

cluding all its routing capabilities) was implemented to enable a full simulation of the resulting three-layer model (see Figure 4.1).

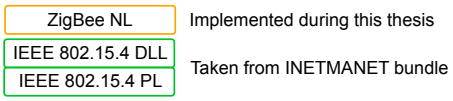

Figure 4.1: Contribution of communication layers

Multiple instances of the simulation model can be combined to build networks of ZigBee devices. The actual number of devices is controlled by the initialization files necessary for each simulation run. These initialization files control the total number of devices in the network and the location of the devices. Furthermore, transmission power per device, message rate and various additional properties can be adapted.

The values interfaced by the initialization files are input to the network objects of the simulation model. As part of the network object, multiple instances of ZigBee device objects are located within the given simulation area. Each of this ZigBee device objects includes a ZigBee stack unifying one instance each of the IEEE 802.15.4 physical layer, the IEEE 802.15.4 DLL, the ZigBee NL and an individual AL. Up to the ZigBee NL, each layer is implemented according to the appropriate specifications. The individual AL differs for the ZC, ZRs and ZEDs. Primitives according to the IEEE 802.15.4 standard and as described in the ZigBee specification are passed between layers as instances of specific message objects. Each data packet is represented as an instance of a data packet object, that is encapsulated and decapsulated between layers. Furthermore, data packet instances can be copied from a physical layer instance to each physical layer instance of the device objects in transmission range.

Since arbitrary scenarios are possible in real-world networks, suitable representative network structures are simulated. The first scenario represents a network where each device transmits only broadcast packets. With this scenario, the maximum possible throughput is tested, since only a low number of network management packets is required. In the second scenario, each device continuously transmits a unicast packet to a randomly – uniformly distributed – chosen device associated to the same network, while limiting the size of the routing table of each device to a few entries. This scenario examines the performance of the routing mechanism defined by the ZigBee specification.

Each of the scenarios is simulated for two different network structures and two network densities. The two network structures are the square structure and the line structure as defined in Section 3.3.1. The different network densities are specified by the number of neighboring devices that are directly reachable. Tight networks can reach a higher number of devices than loose networks (for detailed definitions see Section 3.3.1).

The output of the simulation is parsed to obtain the desired values. A summary of the statistical variables is produced for each simulation run. These plain text format files are parsed for the appropriate information used for the analysis. For easier computations, the results are converted to the Comma-separated values (CSV) format.

The conversion from the CSV format to 3D-plots is done using a Matlab script. Since the discrete data loaded from the CSV files represents single data points and do not cover the whole

spectrum plotted, 3D-plots of these data sets are hard to interpret. Therefore, after loading the data points into a 2D matrix, the undefined data points are interpolated using a multiquadratic method as explained in [23]. Using this interpolation, surface plots can be produced that are well suited for analyses. The number of associated nodes and the message rate are used as the significant values. The resulting 3D-plots are presented in this chapter. The complete workflow is illustrated in Figure 4.2.

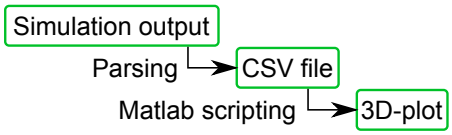

Figure 4.2: Simulation to visualization workflow

Based on the comparison of simulated and estimated results, the correctness of the analytical concepts of Chapter 3 and the simulation models are shown.

4.3 Simulation

According to the network structures used in Chapter 3, the implemented simulation networks either consist of square network structures or line network structures (see Section 3.3.1 for detailed definitions). Similarly, the definitions for tight and loose networks of Section 3.3.1 are used in this chapter. By combining different network structures and variations in the network node density, a wide spread view is presented.

This chapter examines two different applications. The first application describes ZigBee networks, where each node continuously transmits broadcast messages within a predefined interval. ZigBee devices that are simulated using the second application, continuously send unicast messages to random – uniformly distributed – destination nodes, associated to the network.

For both applications network structure, network density, size of the network and the amount of messages sent per second are varied, according to the concepts discussed in Chapter 3. Thus, it is possible to analyze various different networks using an implementation of the ZigBee stack.

4.3.1 Broadcasts

Miscellaneous applications require a high number of data packets to be sent to all devices in the network. Predictions about these networks can help to build up reliable ZigBee networks. Therefore, analysis of the behavior of ZigBee networks, where each device sends multiple broadcast packets per second, are necessary. To discover the behavior of a ZigBee network in a high-load situation, the previously described simulation model is used. The corresponding application transmits broadcast packets continuously. The normal distributed timeouts average to the according message rate.

While simulating, the average transmit time required to deliver the packets to the devices in the network is measured. The required transmit time is defined as the difference between the time where the packet is initiated and the reception of the packet at the last associated device

such that all devices associated with the network received the packet. All transmit times per packet are summed up and the average time is calculated. Since messages have to arrive at the final destination node within one second to retain the information content, the resulting value reveals whether the network is over-utilized.

Furthermore, the protocol overhead is measured. For this purpose, the total number of network management frames and the total number of data frames are compared bitwise. The relation of the data packets to the total number of packets transmitted determines the protocol overhead. The lower the required protocol overhead, the better the protocol performance. Therefore, the protocol overhead acts as indicator for the routing capacity of a ZigBee RD.

The channel efficiency of one device is the relation of actually transferred information bits (by any device in range or the device itself) to the theoretical maximum amount of transferable information bits (250 kbit/s). The overall channel efficiency is the average channel efficiency including all devices associated with the network. The channel efficiency is a measure for the medium access strategy. The better the medium access mechanism (in the case of ZigBee, CSMA/CA is used), the better the channel efficiency. Inefficient medium access mechanisms do not use the total amount of time available. For example, CSMA/CA requires time to listen on the medium whether it is idle.

Loose square networks

The following subsection concerns loose square networks. In the simulation framework, the transmission power is reduced to limit the number of neighbors per node to a constant value of 8. Only these 8 neighbors around a device receive transmitted packets.

Figure 4.3 shows the transmission delay in simulated loose square networks. Similar to the results presented in Figure 3.19, networks with more than 160 nodes cause a total broadcast transmission delay of more than one second. As a consequence of the high number of devices, the greater diameter of the network is the reason for the long transmission delays.

The figure further illustrates, that even 200 broadcast packets per device in one second do not increase the transmission delay, independent of the number of associated devices. Thus, the number of broadcast packets per second influences the transmission delay fewer than the number of devices and hence the diameter of the network does.

The result plot is limited to one second for the transmission delay axis to show more details about interesting areas. Due to given deadlines, various applications require the reception of each message within one second.

The protocol overhead of loose square networks is illustrated in Figure 4.4. Since broadcast packets do not require any network management packets, the protocol overhead produced by broadcast ZigBee networks originates from general ZigBee mechanisms (e.g., periodic network status messages).

Referring to the simulation results, the protocol overhead of loose square networks is linear. The reason for this characteristic is the missing relationship between protocol overhead and broadcast packets. Broadcast packets are data packets that require no network management packets to be transferred to all devices.

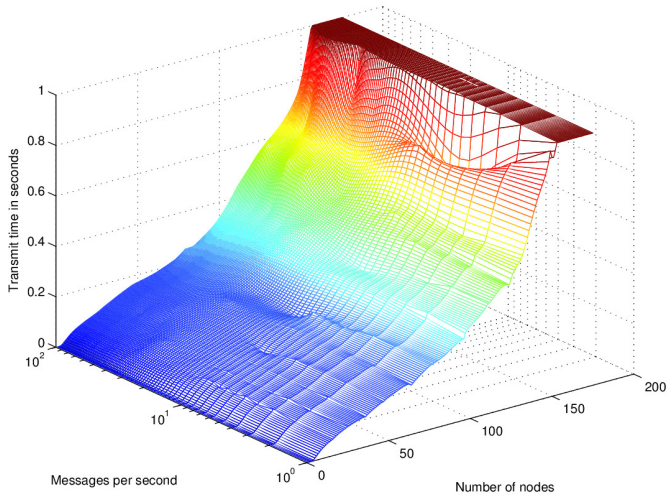

Figure 4.3: Simulated broadcast transmission delay for a loose square network

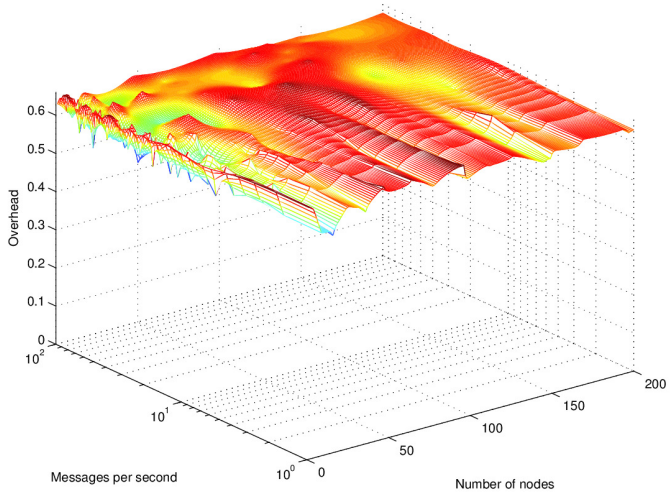

Figure 4.4: Simulated broadcast protocol overhead for a loose square network

Figure 4.5 illustrates the channel efficiency of loose square networks transmitting broadcast packets. Similar to the transmission delay, the channel efficiency does not rely on the number of messages per second, but on the number of devices.

The channel efficiency of loose square networks converges to 35%, similar to the results of loose square networks transmitting route discovery packets, presented later in this Section. Due to the lower number of messages required for transmitting broadcast packets (in relation to the required route discoveries at each hop), the channel efficiency raises slower than in networks transmitting route discoveries.

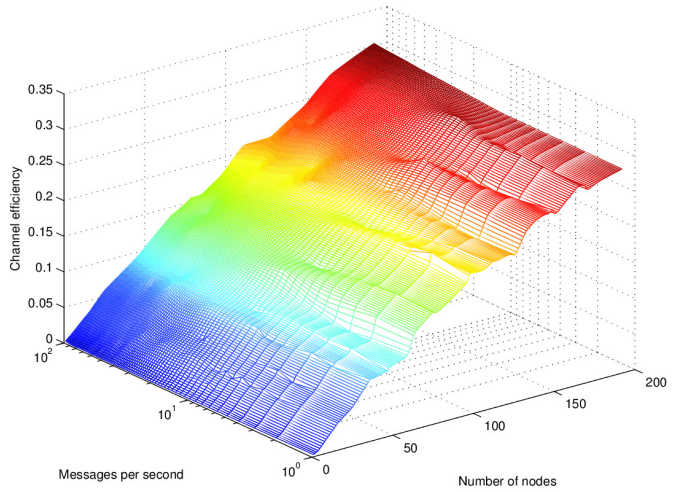

Figure 4.5: Simulated broadcast channel efficiency for a loose square network

Tight square networks

The devices in tight square networks can communicate with up to 20 direct neighbors depending on the location within the networks structure. This can be caused either by a higher transmission power or by a higher density of network devices per square meter. All devices are located with equal distances (see Figure 2.2). For a detailed definition of the square network structure see Section 3.3.1.

Figure 4.6 presents the results regarding the transmission delays of tight square networks. Similar to loose square networks, the number of messages per device in one second does not influence the transmission delay as much as the number of associated devices. Hence, the transmission delay increases according to the increasing number of associated devices. In contrast, an increase of the messages per second does not distinctly increase the transmission delay.

To point out the exponential growth of the transmission delay according to the number of associated nodes, the results are compared to Figure 4.12 which illustrates the corresponding results for tight line networks. For small numbers of associated nodes, the transmission delay of tight square networks increases slowly and transmission delays lower than 0.2 seconds are the average value for networks with 80 associated nodes. Whenever the number of associated nodes is higher than 120, the transmission delay rises fast and the one second limit is reached within a small increase of the number of associated nodes.

Due to the shorter distance between the devices, tight square networks can use a lower number of hops to reach the final destination than loose square networks. Therefore, tight square networks exhibit shorter transmission delays. Significantly, larger networks show up a bigger difference between those two network structures.

Due to the higher node density compared to loose square networks, the protocol overhead of tight square networks could be expected to vary in some areas. As illustrated in Figure 4.7, this is not the case. The protocol overhead of tight square networks equals the protocol overhead of

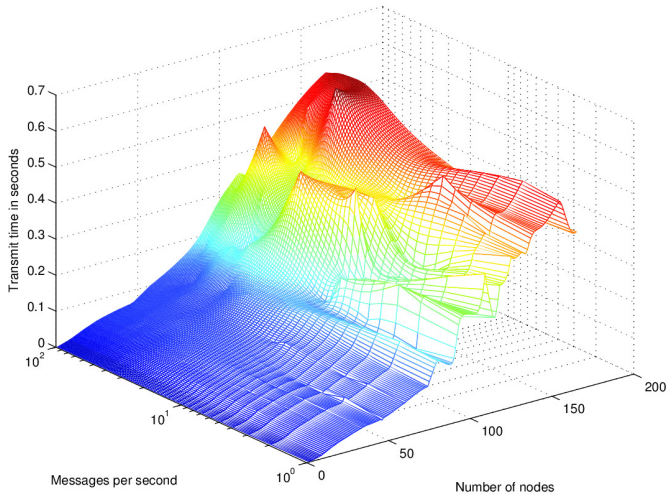

Figure 4.6: Simulated broadcast transmission delay for a tight square network

loose square networks. The average protocol overhead of 60% represents a static value, valid in the entire simulated range.

Continuously transmitted ZigBee network management packets are used to discover the node's neighborhood and to inform the neighbors about the availability state. Furthermore, static network management packets required according to the broadcast packets cause a non-changing protocol overhead behavior. Compared to the actual data packet payload, the network management packet payload dominates high-load situations as well as situations, where the medium is idle. With an increasing number of associated nodes beyond the direct neighborhood of a node, the number of network management packets increases.

Regarding the channel efficiency of broadcast-only tight square networks, the most influencing factor is the number of associated nodes. Illustrated in Figure 4.8, the number of messages transmitted per device in one second does not influence the channel efficiency up to 100 transmitted messages per second. The reason for this independence is the low influence of one device to the channel efficiency. However, the higher the number of associated devices per network is, the higher is the number of bits transmitted within the range of one device. The number of bits influencing one device is multiplied by the number of associated nodes. The diagram is not linearly increasing, since the actual additional number of messages influencing one device decreases with an increase of the network diameter.

Compared to Figure 4.5, that illustrates the channel efficiency of loose square networks, the channel efficiency of tight square networks increases faster. Due to the higher number of direct neighbors, the number of messages influencing one device increases faster. When increasing the network size, the dependence of the number of influencing messages per device on the number of neighboring devices reduces. Both, the channel efficiency of tight square networks and the channel efficiency of loose square networks converge to the same value of about 50%.

Neither the channel efficiency of tight square networks, nor the channel efficiency of loose square networks exceed the limit of 50%. However, a higher number of neighbors per device

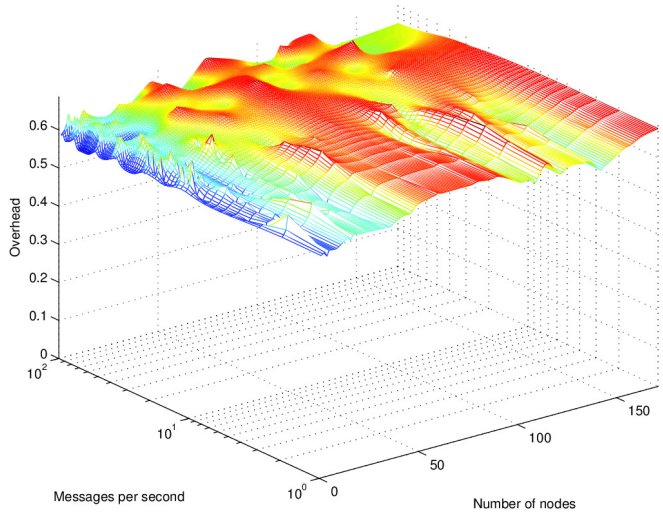

Figure 4.7: Simulated broadcast protocol overhead for a tight square network

(by increasing the transmission power or the network density) can further increase the channel efficiency, due to the higher number of messages transmitted within the range of one device.

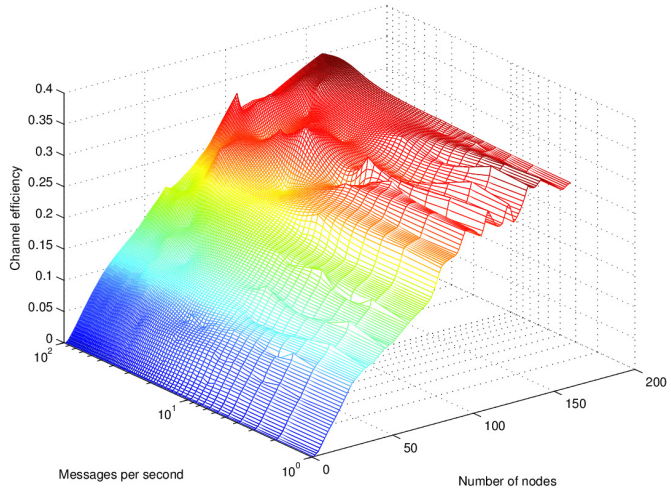

Figure 4.8: Simulated broadcast channel efficiency for a tight square network

Loose line networks

Loose line networks represent low redundancy networks with the main purpose of increasing the network space. By only using pairs of devices to provide a basic form of redundancy and the big distance between each of these device pairs, one device of each pair is necessary to forward a routed packet.

The broadcast transmission delay of loose line networks highly depends on the number of devices associated with the network. Since the network diameter and the number of device pairs are related only by factor two, even the number of hops for reaching the final destination and the average transmission delay relate on this factor.

In contrast to square networks, where the number of hops increases only in a logarithmic manner, the number of hops required to reach the final destination of a packet in a line network, increases linearly (see Figure 4.9). The reason for this linear growth is the necessity for using one device of each node pair. Depending on the linearly increasing number of hops to reach the final destination, the transmission delay also increases linearly.

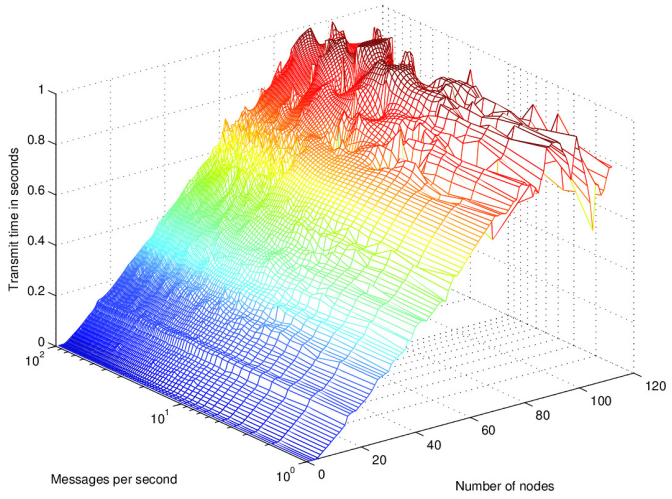

Figure 4.9: Simulated broadcast transmission delay for a loose line network

Broadcast packets always require the same amount of network management packets to be transmitted successfully. Therefore, the protocol overhead in loose line networks equals the protocol overhead of tight line networks and square networks. The network management packet payload always produced by the ZigBee protocol remains in the same relation to the data packet payload for any network size and message amount transmitted per device in one second (see Figure 4.10).

The channel efficiency for loose line networks mainly depends on the size of the network (see Figure 4.11). Since the number of devices associated with the network directly influences the network diameter, the actual number of messages influencing one node increases while enlarging the network size. The channel efficiency converges to a limiting value while the significance of messages transmitted by additional network devices influencing one node, are reduced.

The low value of the channel efficiency indicates a reduced reliability of line networks with less direct neighbors per device. The limited reliability is caused by the low physical redundancy of networks with a low number of direct neighbors. By increasing the number of direct neighbors, the reliability of the network raises, due to the higher redundancy.

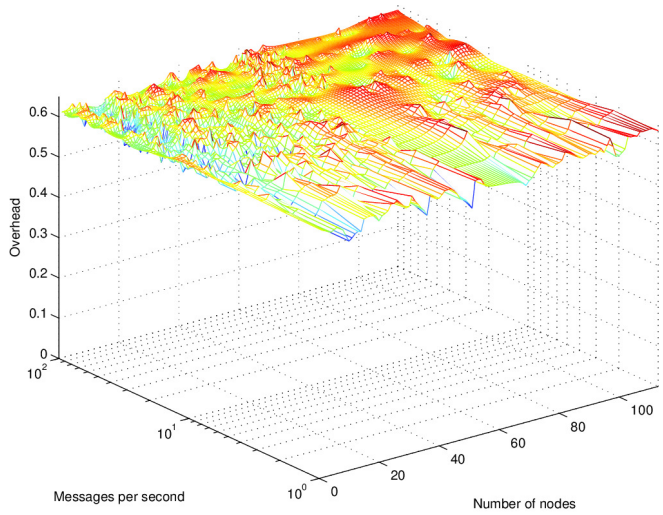

Figure 4.10: Simulated broadcast protocol overhead for a loose line network

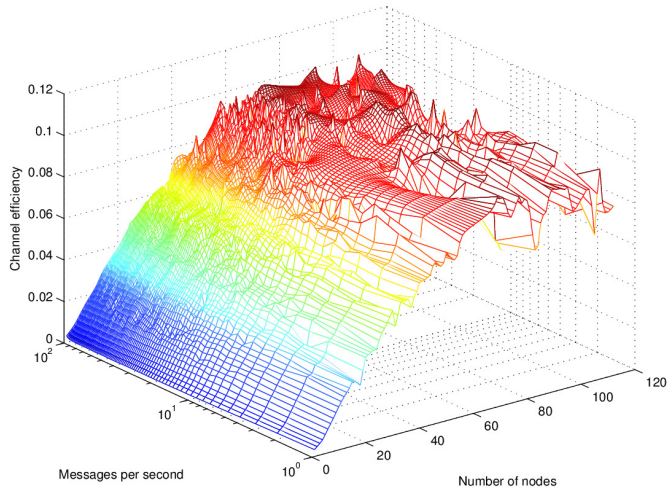

Figure 4.11: Simulated broadcast channel efficiency for a loose line network

Tight line networks

Tight line networks provide a higher redundancy than loose line networks. The higher number of device pairs in transmission range allows skipping one device pair to reach the final destination, due to reduced routing costs. One reason is a lower hop count, necessary to transmit a packet to the destined device.

Similar to loose line networks, the transmission delay of tight line networks almost linearly increases with an increasing number of devices in the network. Due to the possibility

for skipping each other device pair, the transmission delay of tight line networks is half of the transmission delay of loose line networks (see Figure 4.12).

Since the transmission delay of networks with 100 associated devices has an average value of 0.6 seconds, it is possible to further increase the number of devices in the network. Additionally, a higher number of direct neighbors allows to further increase the number of devices associated while keeping the average transmission delay below one second.

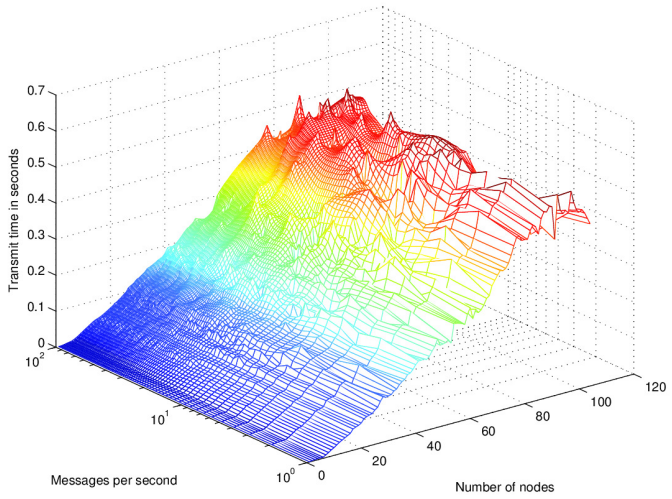

Figure 4.12: Simulated broadcast transmission delay for a tight line network

The protocol overhead of tight line networks has a static value of about 60%, similar to the protocol overhead of loose line networks. Due to the static amount of network management packets required to transmit a broadcast packet, the relation between network management packet payload and data packet payload does not vary according to the increase of associated devices or increase of the number of messages transmitted per device in one second (refer to Figure 4.13).

The channel efficiency of tight line networks is mainly influenced by the number of associated devices. The number of messages per second necessary for a significant increase of the channel efficiency is not included in the simulation results, illustrated in Figure 4.14. Due to the doubled number of devices within transmission range – compared to loose line networks – of each device, the channel efficiency is twice as high as those of loose line networks.

The low channel efficiency can be raised by further increasing the number of direct neighbors per device due to the higher number of messages using the medium. By increasing the number of direct neighbors per device, even the total number of associated devices can be increased.

4.3.2 Route discovery

The following subsections discuss networks where a higher number of devices (i.e., unicast destination addresses) than storable RTEs are addressed. To simulate a typical behavior of a

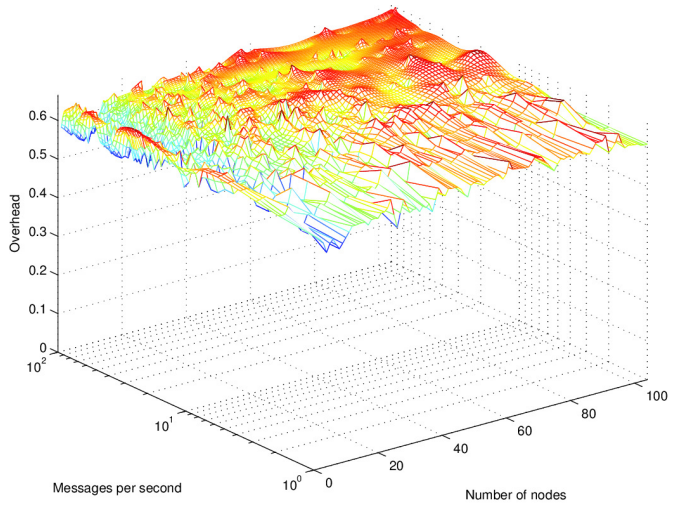

Figure 4.13: Simulated broadcast protocol overhead for a tight line network

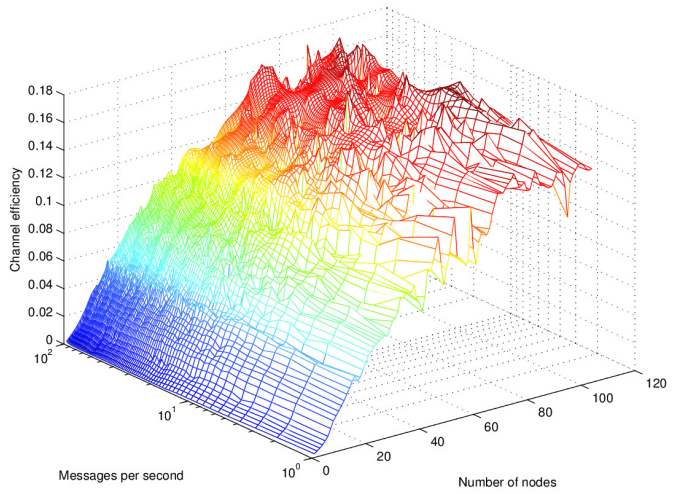

Figure 4.14: Simulated broadcast channel efficiency for a tight line network

network, randomly – uniformly distributed – selected destination addresses, out of the actually associated network devices, are addressed by unicast packets.

Similar to the previous subsections, the transmission delay, protocol overhead and channel efficiency are examined.

Loose square networks

Loose square networks are simulated with reduced transmission power to limit the number of reachable devices per node to at most 8, as illustrated in Figure 4.15. Generally, as long as only

neighboring nodes are addressed and the number of destination nodes remains below the number of storable RTEs, no RREQ is required and each transmission is completed within milliseconds.

Figure 4.15 illustrates the transmission delay of loose square networks. The transmission delay is low, as long as all associated devices are in transmission range or the number of devices associated with the network (reduced by the number of direct neighbors) is lower than the number of RTEs.

The resulting figure further illustrates a fast increase of the transmission delay, where the number of messages per device in one second is higher than 7. Combining the information of Figure 4.15, Figure 4.16 and Figure 4.17, the rising channel efficiency and the reduction of the protocol overhead at the same point are revealed. This means, ZigBee networks with no routing requirements (all devices are direct neighbors) work efficiently, whenever 5 to 8 messages are transmitted in one second by each device.

ZigBee networks consisting of devices working within the maximum designated message rate per device (between one and two messages per second), can be built up to an overall size of 35 devices. For networks consisting of more than 35 devices, transmission delays greater than one second have to be expected.

Figure 4.15 is limited to a transmission delay of 4 seconds, since longer delays of messages imply reduced informational content, depending on the appropriate application.

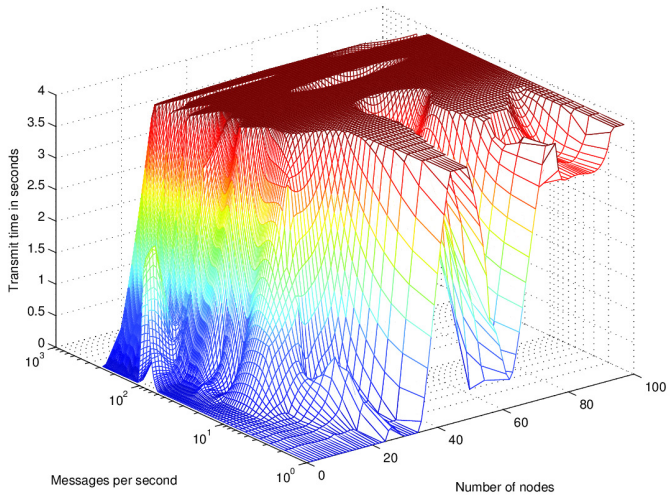

Figure 4.15: Simulated route discovery transmission delay for a loose square network

The following Figure 4.16 illustrates the protocol overhead of loose square networks. Regarding Chapter 3, ZigBee devices transmit network management packets even when being idle. For example, a network status message is transmitted every 15 seconds on average by each device. Since the protocol overhead represents the relation between the number of network management packet bits and the number of data packet bits, a low number of data packets implies an increased protocol overhead. Therefore, whenever ZigBee devices transmit less than 10 messages in one second, the protocol overhead is about 50%. For small ZigBee networks, an increased number of data packets results in a reduced protocol overhead.

Due to the limited number of RTEs per ZigBee device and since a RREQ is required for each destination device not included in the routing table, a higher number of devices associated with a ZigBee network increases the number of network management packets required to address all devices. Therefore, the number of network management packets increases with a higher number of destination addresses not included in the routing tables of the devices associated with the network.

Similar to the transmission delay and the channel efficiency, the protocol overhead increases whenever more devices are associated with the network than the sum of direct neighbors and storable RTEs per device.

Since each missing RTE requires a broadcast network management packet (i.e., RREQ), the protocol overhead converges to 100% with increasing number of associated network devices. Therefore, the number of RTEs should be greater than or equal to the number devices associated with the network.

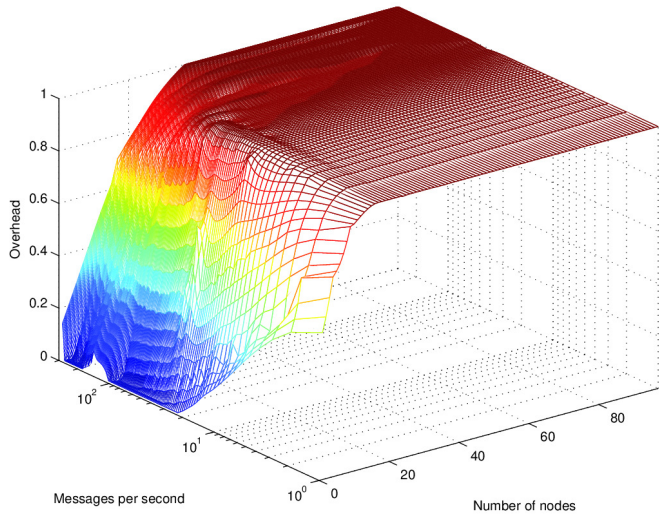

Figure 4.16: Simulated route discovery protocol overhead for a loose square network

Figure 4.17 shows the simulation results of loose square networks, regarding channel efficiency. Compared to Figure 3.25, where the estimated channel utilization is shown, the simulated channel utilization only raises up to 45% of the channel utilization. Inefficient channel usage, caused by a real medium access strategy, implies this low rate of channel utilization. Nevertheless, the overall image of the channel utilization regarding differing numbers of nodes and messages per second is similar to the estimated channel utilization.

The results show that networks consisting of 20 or less devices only use a small amount of the possible channel utilization. As long as the number of messages per device in one second is smaller than 10, channel utilizations lower than 10% can be expected. At least when each device sends more than 100 messages in one second, these small networks use the full amount of channel utilization possible, since the total number of messages in the network is too small.

As expected by the estimations of Chapter 3, networks consisting of more than 20 devices, each sending more than 10 messages in one second, use the total channel utilization possible. Further increasing the number of devices associated with the network does not increase the channel efficiency. Similarly, increasing the number of messages transmitted per device in one second does not increase the channel efficiency, since the global maximum is already reached. An increase of the channel utilization is only possible if different medium access strategies (i.e. TDMA) are used.

Whenever the channel efficiency is close to the global maximum, the duration of the periods where the network is stalling (i.e. the relation of successful data requests to the actually transmitted packets is smaller than one percent) increases. In these periods, a device has to wait for the channel to become idle. Hence, the number of devices waiting for the channel to become idle increases. Therefore, the channel load can only be reduced by decreasing the number of messages transmitted per device in one second.

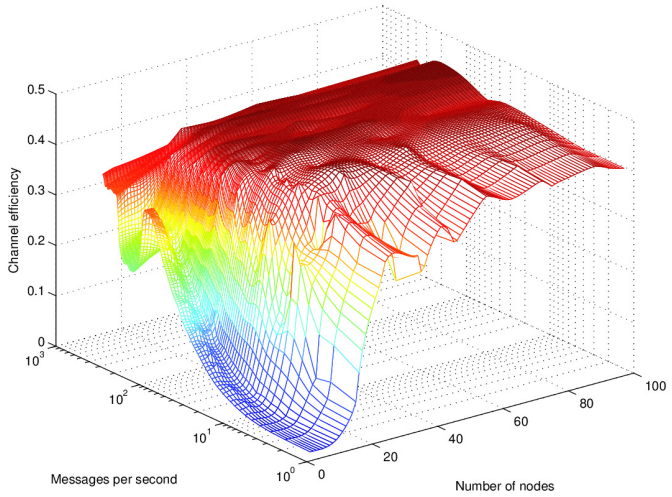

Figure 4.17: Simulated route discovery channel efficiency for a loose square network

Tight square networks

Tight square networks consist of devices aligned in a square structure. Compared to loose square networks the number of direct neighbor devices is higher. This can be the case because of either a higher transmission power used or a higher application-specific network density. In general, whenever a network consists of more devices than the sum of neighboring devices of one node and the number of RTEs, performance losses are to be expected.

Figure 4.18 illustrates the transmission delay in tight square networks. Due to the higher number of neighboring nodes, a higher number of devices is possible until an appreciably increase of the transmission delay. In contrast, the higher density of the network causes a higher channel usage. According to the message rate, the transmission delay reaches almost the same point as in loose square networks.

An increasing messages transmission rate reduces the number of devices that can be associated without increasing the transmission delay above one second. When keeping the number of messages per second below two (this is the designated maximum limit for ZigBee), about 30 devices can be associated. All of these 30 devices can continuously transmit packets that arrive within one second at the final destination. Further increasing the message rate, reduces the number of devices that can be associated.

According to Figure 3.26 and Figure 4.20, an increasing channel utilization and increasing actual channel efficiency cause the transmission delay to increase, too. Therefore, the simulation results of tight square networks correspond to the appropriate estimations.

The transmission delays presented in Figure 4.18 are limited to 4 seconds, since higher transmission delays are not interesting due to their reduced information content. ZigBee packets are designated to arrive at their final destination within at most one second. Networks with longer transmission delay are over-utilized.

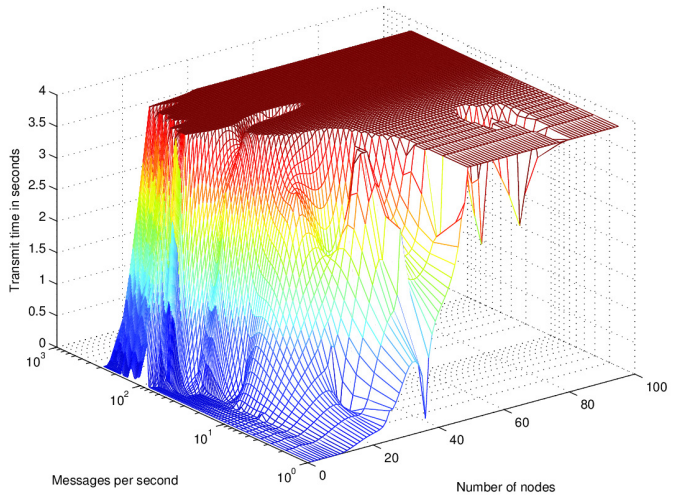

Figure 4.18: Simulated route discovery transmission delay for a tight square network

Figure 4.19 illustrates the protocol overhead for tight square networks. The higher protocol overhead in networks with less than 20 devices and message rates lower than 10 messages per second is caused by the static amount of network management packets. Therefore, the protocol overhead reduces for a higher rate of data packets. Overall, when increasing the number of messages per second, the protocol overhead will converge to a dedicated value, since there exists a maximum limit for the amount of network management packets required to transmit a data packet.

For higher numbers of devices and lower numbers of messages per second, the protocol overhead raises up to an amount of about 98%. This is caused by the high amount of RREQs required to find the next hop address for destination addresses neither included in the neighbor table nor in the routing table. Therefore, the protocol overhead increases to its global maximum.

The protocol overhead can be reduced by increasing the number of RTEs per device. Each RTE requires a memory amount of at least 5 Bytes. Since memory is limited especially con-

sidering embedded devices, this can be problematic. The results show the high requirements of memory for ZigBee devices to ensure high availability of the network.

Another possibility to reduce the protocol overhead is to increase the density of the network. Since the results illustrated in Figure 4.19 originate from tight square networks, a further increase of the network density implies longer transmission delays and earlier over-utilization of the channel.

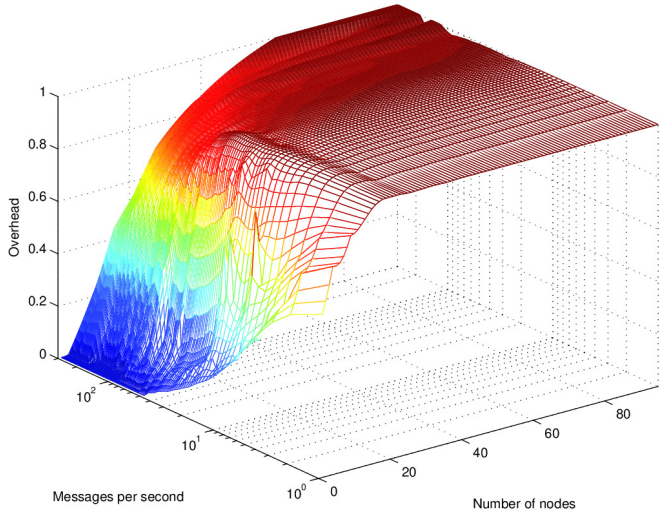

Figure 4.19: Simulated route discovery protocol overhead for a tight square network

Figure 4.20 illustrates the channel efficiency of tight square networks. A higher channel efficiency is possible with a higher number of messages, since more time is required to transmit the packets. Whenever the time required to transfer all messages requested in one second is lower than one second, the channel efficiency reduces.

According to Figure 4.20 the channel efficiency of tight square networks increases with a higher amount of associated devices as well as sent messages per second per device. As outlined before, the reason for this increase is the higher number of transferred bits in relation to the maximum number of transferable bits per second (250 kbit/s).

Corresponding to Figure 3.26, the channel efficiency increases according to an increasing number of messages per device in one second. The actual maximum for the channel efficiency is about 60% since the medium access mechanism is not optimal. A higher channel efficiency is only possible when using a more efficient medium access mechanisms (e.g. TDMA).

Compared to Figure 4.17, a higher channel efficiency is possible due to the higher network density. Furthermore, the channel efficiency of tight square networks reaches its maximum faster than in loose square networks, when increasing the number of devices associated with the network. When increasing the message rate for small networks, the local maximum channel efficiency is equal for loose and tight square networks. The reason is that all devices are direct neighbors or at least its corresponding RTEs fit into the routing table of each device.

While increasing the number of messages per device transmitted in one second, the channel efficiency increases linearly. The axis of the messages per second has a logarithmic scale, to present an overview of the whole transmission spectrum and to show low data rates in detail.

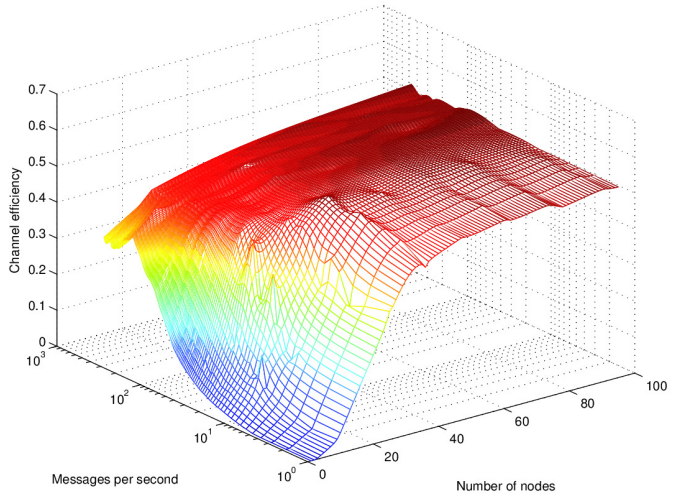

Figure 4.20: Simulated route discovery channel efficiency for a tight square network

Loose line networks

Loose line networks relate to a line network structure as defined in Section 3.3.1. Due to longer distances between the nodes, a device associated with a loose line network has a low number of direct neighbor devices.

Figure 4.21 illustrates the average transmission delay of packets transferred in loose line networks. Compared to loose square networks, the message rate can be much higher until the transmission delay rises over the one second limit. This way, more messages can be transmitted within the same amount of time, since the the transmission time is lower in general.

The protocol overhead of loose line networks transmitting unicast packets rises to more than 90% whenever the number of associated devices per network is higher than the number of storable RTEs.

An increased number of transmitted messages per device in one second reduces the value of the protocol overhead (see Figure 4.22). This holds only for networks with a lower number of devices than storable RTEs per device. The lower amount of network management packet payload in relation to the data packet payload is the reason for the reducing protocol overhead.

Figure 4.23 illustrates the channel efficiency of loose line networks. Networks with a higher number of devices than the sum of RTEs and direct neigbhors, can use the maximum channel efficency.

Small networks (with less devices than RTEs storable per device) can increase the channel efficiency since more messages per device in one second can be transmitted. Even the maximum channel efficiency of 35% can be reached.

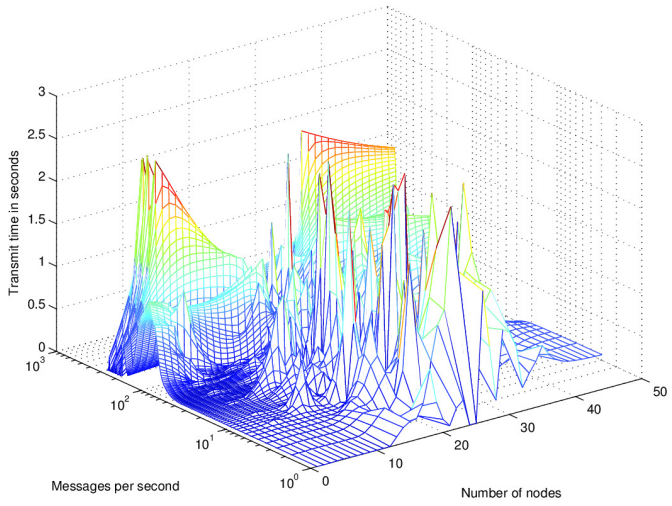

Figure 4.21: Simulated route discovery transmission delay for a loose line network

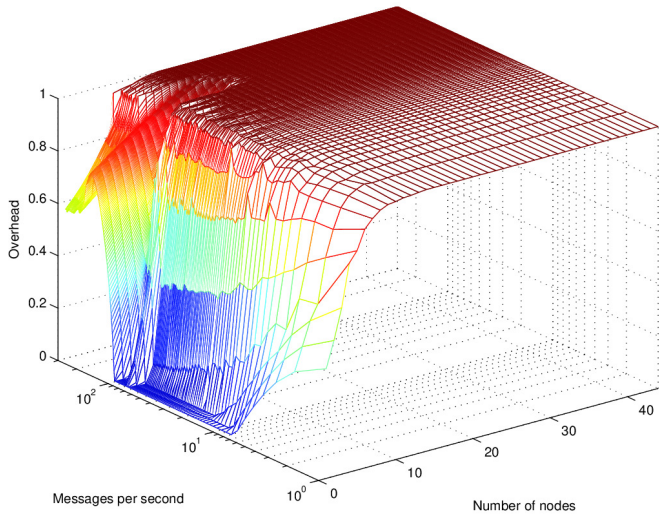

Figure 4.22: Simulated route discovery protocol overhead for a loose line network

Tight line networks

Tight line networks are characterized by a higher number of direct neighbors per device than in loose line networks. Due to the higher number of direct neighbors, tight line networks provide a higher level of redundancy. The network structure consists of device pairs that are placed side by side with other device pairs. For a detailed description of the line network structure, see Section 3.3.1.

Whenever more devices are associated to the network than direct neighbors per device, route discovery packets are necessary to reach the final destination. Since each device of the net-

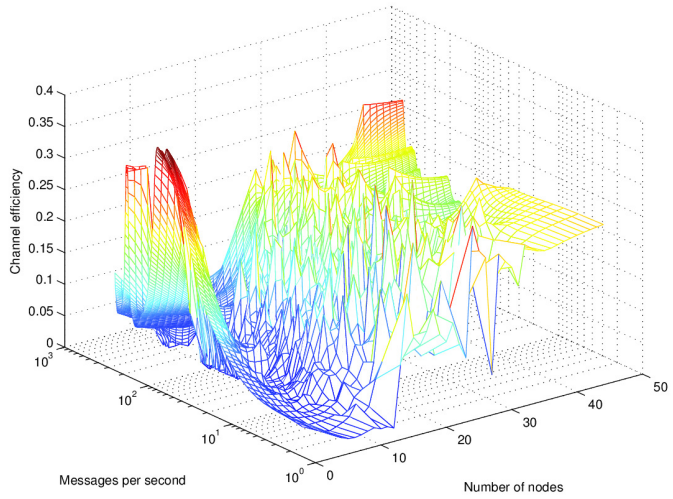

Figure 4.23: Simulated route discovery channel efficiency for a loose line network

work needs to rebroadcast the route discovery packet, the transmission delay increases (see Figure 4.24). For networks with a low network device number, where all devices are in range of each device, a message rate of 80 messages per second or higher causes the transmission delay to increase.

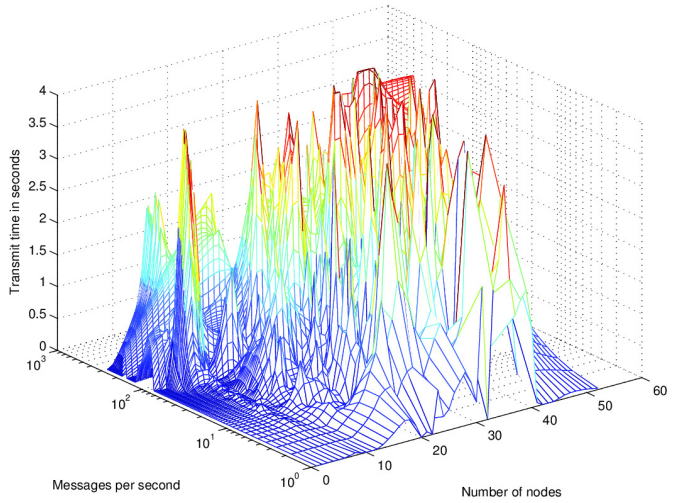

Figure 4.24: Simulated route discovery transmission delay for a tight line network

The protocol overhead of tight line networks increases to 90%, whenever route discovery packets are required continuously. This is the case, when the number of devices is higher than the sum of the number of direct neighbors per device and the number of possible RTEs per device.

For message rates lower than 10 messages per device in one second, the protocol overhead increases even for small networks. This is caused by the static network management packet payload produced by the ZigBee protocol. For higher message rates in smaller networks, the protocol overhead reduces, due to the increasing amount of data packet payload (see Figure 4.25).

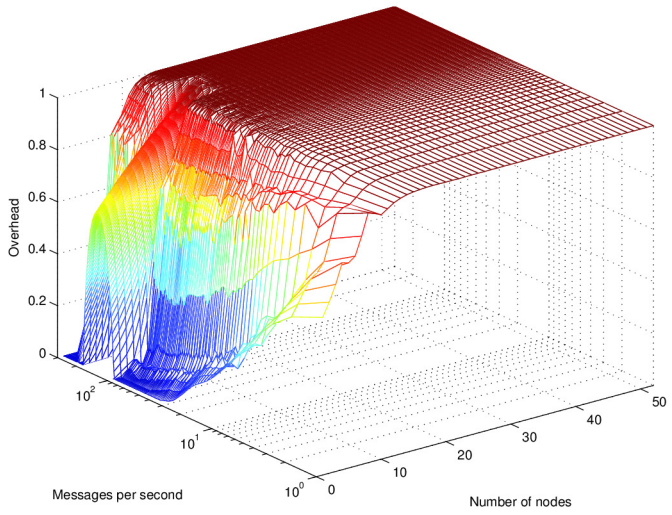

Figure 4.25: Simulated route discovery protocol overhead for a tight line network

According to Figure 4.26, the channel efficiency of tight line networks increases for small networks, when the message rate increases, due to the increasing use of the medium. The maximum channel efficiency of tight line networks equals twice the channel efficiency of loose line networks, due to the doubled number of direct neighbors per device.

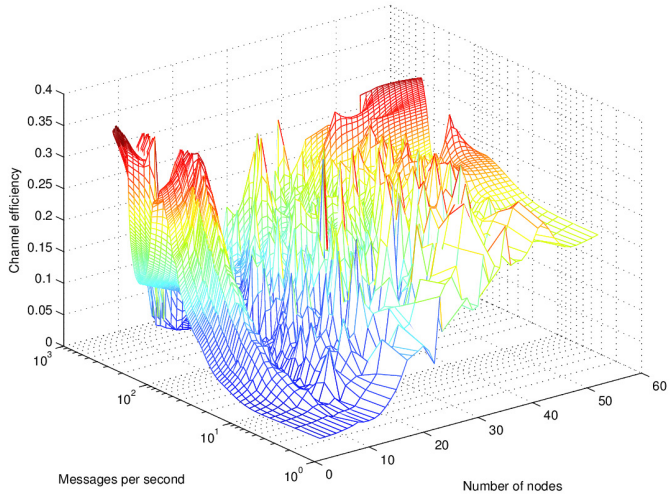

Figure 4.26: Simulated route discovery channel efficiency for a tight line network

CHAPTER 5

Conclusion & Outlook

5.1 Summary

In this thesis, abstract network structures of ZigBee were analytically analyzed. The results showed the different behavior of ZigBee networks according to different network structures. The following paragraphs summarize the results and emphasize the main differences between the examined network structures.

The simulations presented in Chapter 4 were used to prove the correctness of the estimations developed in Chapter 3. The results of the simulations and estimations illustrated the importance of redundancy in WSANs. A higher number of neighbors gives higher performance than networks with a low number of neighbors per device.

Furthermore, Chapter 3 outlined the performance of ZigBee's addressing mechanisms (DAAM and SAAM). Advantages and disadvantages of both were discussed. Tree-based and static networks have a high performance when using DAAM. In contrast, networks with a high number of devices and dynamic networks – where devices change the location – can use the the address space more efficiently when using SAAM than with DAAM.

The problem of unpredictable WSANs, as described in the introduction, is solved for simple square networks. This type of networks can be predicted using the estimations presented in Chapter 3.

The results of this thesis can be used to predict the behavior of square and line networks by using the concepts presented in Chapter 3. These different basic structures can be assembled to different kinds of networks. Square networks, as defined in Section 3.3.1, can absorb intense traffic due to the multi-directed network expansion. Therefore, the maximum values transferable in square networks were not reached by simulations. Neither for tight square networks, nor for loose square networks it was possible to increase the network size or the number of messages transmitted per device in one second up to the over-utilized of the network. Figure 4.5 and Figure 4.8 illustrate the continued increase of the channel efficiency. An increase of the number of direct neighboring devices per node can solve this issue in further simulations, since the higher number of direct neighbors implies a higher traffic on the medium used by each device. Even

an increase of the total number of devices can further raise the channel efficiency, since each additional device adds traffic to the whole network, as long as broadcast packets are transmitted or destination devices across the whole network are addressed.

When comparing loose square networks and tight square networks regarding to broadcast-only communication, the difference is not so significant. Nevertheless, due to the higher number of direct neighbors in tight square networks, the channel efficiency of tight square networks is higher. In total, the number of messages transmitted is equal, but in loose square networks the average load on the medium is reduced.

According to square networks submitting unicast packets which can also lead to route discovery packets broadcast to the network, the average transmission delay does not differ significantly. In contrast, the protocol overhead of tight square networks is significantly lower for high message transmission rates than the protocol overhead of loose square networks. This is caused by the lower number of route discoveries necessary due to the higher number of direct neighboring devices. These neighboring devices can only be addressed directly. Furthermore, the number of hops, and therefore the number of route discoveries necessary for reaching the final destination is lower.

This thesis also examined line networks as defined in Section 3.3.1. Especially two abstractions of line networks, tight and loose line networks are considered. In general, the simulations emphasized the low reliability of line networks. Especially line networks with a low number of direct neighbors should never be used for applications requiring high reliability. Any high load situation or package loss can cause the network to break down.

Regarding broadcast-only networks, the transmission delay of tight line networks is half of the transmission delay of loose line networks. Due to the doubled number of direct neighboring devices, half of the hops are necessary to transmit one broadcast packet to all devices in the networks. Furthermore, the higher number of direct neighboring devices allows to have a higher channel efficiency per device. The channel efficiency of tight line networks can not be increased up to twice of the channel efficiency of loose line networks, since the channel efficiency of loose line networks increases more linear than the channel efficiency of tight line networks.

Line networks that transmit unicast packets to random destination devices in the network and therefore have to mix unicast and broadcast packets can be jammed with a few packets. Even a low number of transmitted packets can cause the network to jam. Tight line networks provide slightly better results regarding the protocol overhead. In general, the number of direct neighbors in line networks needs to be higher than 8 neighbors per device to provide reliable behavior.

Summarizing the results of line networks and square networks, square networks behave more reliable and predictions for square networks are more exact than for line networks. Due to the low reliability of line networks, predictions are hard to make. Each unpredicted packet drop can cause a jam of the whole line network and the remaining prediction fails totally. A combination of square and line networks could be a good solution for many applications. Besides this behavior, the channel efficiency of line networks is always lower than for square networks, due to the lower number of direct neighboring devices. Nevertheless, the line network parts of the application should have groups of three devices instead of pairs, to provide a higher overall reliability.

Broadcast-only square networks have logarithmic transmission delays according to an increasing number of devices. Transmission delay diagrams of line networks, where the devices transmit only broadcast packets, have a linear characteristic. The possibility in square networks to continuously transmit the packet in all four directions simultaneously, following parallel, non-influencing transmissions at subsequent hops, is a big advantage towards line networks. Line networks can only transmit in two directions simultaneously.

Mixed packet networks also show this main difference. Transmission delays do not increase linearly for an increasing network size in square networks. In contrast, line networks show this strictly linear increasing transmission delay behavior according to an increasing network size. The reason for this main difference is the behavior of the routing concept of ZigBee. In square networks it is possible to use shorter routes, since direct connections can be used. Routes in line networks always have to use the line connections and no shorter connections are available. This redundancy and availability of different connections is one of the main advantage of square networks towards line networks.

5.2 Alternative concepts

This section deals with alternative concepts regarding ZigBee routing mechanisms. The advantages and disadvantages of these concepts are discussed.

Since AODV is based on route discovery, including broadcast packets, the flooding broadcast mechanism has to be analyzed. The minimum requirements regarding timing and memory aspects of broadcast algorithms are discussed in [13]. Furthermore, the Optimal On-Tree Selection Algorithm (OOS) is presented and a variant adapted for ZigBee is presented. The OOS is an algorithm to find an optimal solution for the forward node selection problem. Every hop on the way to the destination node needs to execute the OOS to find a subset of its one-hop neighbors usable as next hop for the transmission. The OOS solves this problem in polynomial time.

An improvement of the forward node selection algorithm is presented in [12]. Rebroadcasts are triggered by ”a non-random rebroadcast timer ... according to the number of neighbors to covered, distance and link quality [12]”. Due to the polynomial behavior of the Reliable ZigBee Forward Node Selection Algorithm (ZiFA-R) node selection algorithm, the use of this algorithm can be useful to improve the performance of the ZigBee routing concept. ZiFA-R provides a higher broadcast redundancy than ZigBee Rebroadcast Algorithm (ZiRA), whereas ZiRA shows lower transmission delays.

The base broadcasting performance of ZigBee is compared to ZiFA-R in [30]. Once more, the better performance of ZiFA-R regarding rebroadcast percentage, packet arrival and collision is shown.

Efficient algorithms, regarding broadcasting in mesh networks, are analyzed and simulated in [26]. Transmission delay and message arrival rate of networks using synchronized multi-node broadcast algorithms are analyzed. The vulnerability to node failures is considered in the literature. In consequence, an increase of redundancy caused an increase of the transmission delays.

The flooding broadcast method used by AODV reduces the performance of the networks. Therefore, optimizations regarding the routing concept of AODV are presented in the following literature.

The routing algorithm included to the ZigBee specification AODV is compared to a tree-based routing scheme, called HiErarchical Routing Algorithm (HERA), based on a hierarchical structure in [11]. End-to-end delay, packet loss and the number of RREQs are measured. The analysis resulted in a low percentage of package loss for HERA and a significantly reduced transmission delay.

The base functionality of AODV is improved by considering load balancing, link quality and successful transmission rate in [1]. The improved path selection method basis on a calculation of the expected transmission time calculated during route discovery.

5.3 Further work

This thesis examined the routing mechanisms defined by the ZigBee specification [20]. Based on the different routing concepts presented in Section 5.2, a comparison of predictability and reliability should be examined. Routing concepts not included in the ZigBee specification but used by other WSANs can result in a higher performance.

A widened view on the behavior of ZigBee networks could be delivered by examining additional network structures. At least hybrid network structures could be interesting for predicting the behavior of ZigBee networks in different applications.

In addition to the homogeneous approach, where only ZigBee devices where used, heterogeneous approaches with devices interfering the medium with other protocols operating in the same frequency band, could be used. This could lead to a better comprehension for the behavior of ZigBee networks in a real world environment.

List of Figures

List of Tables

Bibliography

[1] M.M. Ajmal, K. Mahmood, and S.A. Madani. Efficient routing in wireless mesh network by enhanced AODV. In *Proc. of the International Conference on Information and Emerging Technologies (ICIET)*, pages 1–7, 2010.

[2] R.L. Bagrodia. Parallel languages for discrete-event simulation models. *Computational Science & Engineering, IEEE*, 5(2):27–38, 1998.

[3] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. Glomosim: A scalable network simulation environment. *UCLA Computer Science Department Technical Report*, 990027:213, 1999.

[4] SIG Bluetooth. Specification of the Bluetooth system. *Core, version*, 1, 2001.

[5] Dominik Bunyai, Lukas Krammer, and Wolfgang Kastner. Limiting Constraints for ZigBee Networks. In *Proc. of the 38th Annual Conference of the IEEE Industrial Electronics Society (IECON' 12)*, pages 1–7, Montreal, Canada, October 2012.

[6] X. Chang. Network simulations with OPNET. In *Proc. of the Winter Simulation Conference*, pages 307–314, 1999.

[7] D. Chen, M. Nixon, and A. Mok. *WirelessHART: real-time mesh network for industrial automation*. Springer Publishing Company, Incorporated, 2010.

[8] F. Chen and F. Dressler. A simulation model of IEEE 802.15.4 in OMNeT++. 6. *GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Poster Session*, 2007.

[9] International Electrotechnical Commission. Industrial communication networks - Wireless communication network and communication profiles - WirelessHART, IEC 62591, 2009.

[10] Robert Cragie. ZigBee Security. Technical report, ZigBee Alliance ZARC Security Task Group, 2009.

[11] F. Cuomo, S. Della Luna, U. Monaco, and F. Melodia. Routing in ZigBee: benefits from exploiting the IEEE 802.15. 4 association tree. In *Proc. of the IEEE International Conference on Communications (ICC'07)*, pages 3271–3276, 2007.

[12] G. Ding, Z. Sahinoglu, B. Bhargava, P. Orlik, and J. Zhang. Reliable broadcast in ZigBee networks. In *Proc. of the Ann. IEEE Comm. Soc. Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks*, pages 510–520, 2005.

[13] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava. Tree-based data broadcast in IEEE 802.15. 4 and ZigBee networks. *IEEE Transactions on Mobile Computing*, 5(11):1561–1574, 2006.

[14] International Organization for Standardization (ISO). ISO/IEC 7498-1. Information technology - Open Systems Interconnection - Basic reference model: The basic model, november 1994.

[15] Thomas Fruehwirth. ZigBee WirelessHART 6LoWPAN - ein Vergleich. Technical Report 183/1-160, A-Lab @ Automation Systems Group, TU Vienna, April 2012.

[16] M. Gerla, K. Tang, and R. Bagrodia. TCP performance in wireless multi-hop networks. In *Proc. of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pages 41–50. IEEE, 1999.

[17] D. Gislason. *ZigBee Wireless Networking*. Newnes, 2008.

[18] IEEE 802.11 Working Group et al. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. 1997.

[19] IEEE Computer Society. *IEEE Standard 802.15.4*. IEEE, 2003.

[20] IEEE Computer Society. ZigBee specification. *ZigBee document 053474r17, ZigBee Standard Organization*, 2008.

[21] Daintree Networks Inc. Getting Started with ZigBee and IEEE 802.15.4. Technical report, 2010.

[22] Texas Instruments. *CC2520 Datasheet*, 2007. Datasheet.

[23] Karl Kraus. *Photogrammetrie 3. Topographische Informationssysteme*. Walter de Gruyter, Berlin, 1. edition, 2000.

[24] S.J. Lee, W. Su, and M. Gerla. Ad hoc wireless multicast with mobility prediction. In *Proc. of the Eight International Conference on Computer Communications and Networks*, pages 4–9, 1999.

[25] C. Marghescu, M. Pantazica, A. Brodeala, and P. Svasta. Simulation of a wireless sensor network using OPNET. In *Proc. of the IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 249–252, 2011.

[26] E. Modiano and A. Ephremides. Efficient algorithms for performing packet broadcasts in a mesh network. *IEEE/ACM Transactions on Networking (TON)*, 4(4):639–648, 1996.

[27] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pages 90–100, 1999.

[28] Shelby, Zach, Bormann, and Carsten. *6LoWPAN - The Wireless Embedded Internet*. John Wiley & Sons, 2011.

[29] ISA Standardization. ISA100. 11a. *Wireless Systems for Industrial Automation: Process Control and Related Applications*, 2009.

[30] T.W. Sung, T.T. Wu, C.S. Yang, and Y.M. Huang. Reliable Data Broadcast For Zigbee Wireless Sensor Networks. *International journal on smart sensing and intelligent systems*, 3(3):504–520, 2010.

[31] SL Ullo, A. Vaccaro, and G. Velotto. Performance Analysis of IEEE 802.15. 4 based Sensor Networks for smart grids Communications. *Journal of Electrical Engineering: Theory and Application*, 1(3):129–134, 2010.

[32] András Varga and OpenSim Ltd. *OMNet++ User Manual*, 2010.

[33] A. Wang, P.L. Reiher, and R. Bagrodia. A simulation evaluation of optimistic replicated filing in mobile environments. In *Proc. of the IEEE International Performance, Computing and Communications Conference (IPCCC'99)*, pages 43–51, 1999.

[34] G. Wang. Comparison and evaluation of industrial wireless sensor network standards ISA100. 11a and WirelessHART. Master's thesis, Chalmers University of Technology, 2011.