Fakultät für Informatik und Mathematik

Universität Passau, Germany

Energy-Efficient and Timely Event Reporting Using Wireless Sensor Networks

Patrick Wüchner

Supervisor: Hermann de Meer

A thesis submitted for

Doctoral Degree

August 2013

1. Reviewer: Prof. Hermann de Meer
   Professor of Computer Networks and Communications
   University of Passau
   Innstraße 43
   94032 Passau, Germany
   Email: demeer@uni-passau.de
   Web: http://www.net.fim.uni-passau.de

2. Reviewer: Prof. Ian F. Akyildiz
   Ken Byers Chair Professor in Telecommunications
   Georgia Institute of Technology
   Centergy One, Room 5170
   75 5th St. NW
   Atlanta, GA 30308, USA
   Email: ian@ece.gatech.edu
   Web: http://www.ece.gatech.edu/research/labs/bwn/IFA/

Abstract

This thesis investigates the suitability of state-of-the-art protocols for large-scale and long-term environmental event monitoring using wireless sensor networks based on the application scenario of early forest fire detection. By suitable combination of energy-efficient protocol mechanisms a novel communication protocol, referred to as *cross-layer message-merging protocol* (XLMMP), is developed. Qualitative and quantitative protocol analyses are carried out to confirm that XLMMP is particularly suitable for this application area. The quantitative analysis is mainly based on finite-source retrial queues with multiple unreliable servers. While this queueing model is widely applicable in various research areas even beyond communication networks, this thesis is the first to determine the distribution of the response time in this model. The model evaluation is mainly carried out using Markovian analysis and the method of phases. The obtained quantitative results show that XLMMP is a feasible basis to design scalable wireless sensor networks that (1) may comprise hundreds of thousands of tiny sensor nodes with reduced node complexity, (2) are suitable to monitor an area of tens of square kilometers, (3) achieve a lifetime of several years. The deduced quantifiable relationships between key network parameters—e.g., node size, node density, size of the monitored area, aspired lifetime, and the maximum end-to-end communication delay—enable application-specific optimization of the protocol.

Acknowledgements

Sincere gratitude is due to my primary supervisor, Prof. Hermann de Meer. He not only let me work self-dependently while showing appreciated patience and stressing the importance of methodological impact, but also provided me with the opportunity to collaborate with a wide range of international partners in interesting and manifold research projects. As a representative example, I would like to mention the EU-funded Network of Excellence *EuroNF* (FP7, ICT-2007-1-216366) here.

I also want to cordially thank Prof. Ian F. Akyildiz for kindly agreeing to serve as second reviewer.

I feel sincerely indebted to former and current colleagues, who made my time at the research group and at the University of Passau a pleasant experience. In particular, I want to thank Andreas Fischer, Dr. Andreas Berl, Ralph Herkenhöhner, Gergő Lovász, and Dr. Ulrich Zukowski for their support, inspiration, and fruitful discussions on various research-related and research-unrelated issues. I thank my office mate Dr. Richard Holzer for his kind and expert help, in particular on resolving mathematical knottiness.

I need to thank late Acad. Director Dr. Gunter Bolch who far too early left this world in May 2008. Gunter drew my attention to the research area of performance modeling and supported—together with Dr. Jörg Barner and Prof. Khalid Al-Begain—the first steps of my academic career.

During my frequent research visits to the University of Debrecen, I was able to focus on the topic of retrial queues and I am grateful to Prof. János Sztrik for serving as a pleasant host, competent contact, and enduring co-author. I appreciate the corresponding bilateral travel grants (436 UNG 113/180/0-1, 436 UNG 113/197/0-1) of the German and Hungarian Science Foundations (DFG and MTA/HAS).

Likewise, I want to cordially thank Prof. Vicente Casares-Giner for valuable discussions in the field of wireless sensor networks and performance modeling.

I also need to address thanks to the members of my two bands (*Texor* and *Chord Dump*) and many friends—too numerous to list individually—for cheering me up in hard times.

I am permanently thankful to my parents and my sister for their enduring support—in particular in non-academic matters.

Last but definitely not least, I want to express my sincere gratefulness to my beloved girlfriend Steffi for guarding my back and for frequently having to endure the company of my body without its mind.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

If you have the right attitude, interesting problems will find you.

— Eric S. Raymond

Modern sensor and communication technology is subject to ongoing miniaturization. For wireless sensor networks (WSNs)*, this development steadily enlarges the areas of imaginable applications. Since the requirements imposed on WSNs are highly application-specific, the use of generic solutions is often rendered infeasible. Hence, for many application areas, a plethora of open research questions still needs to be solved to enable the design and implementation of actually practical solutions.

1.1 WSN-Based Environmental Event Monitoring

One of many possible application areas of WSNs is environmental monitoring, which includes, e.g., forest fire or flood detection, volcano or habitat monitoring, precision agriculture, or pollution studies as specific applications (cf. [3, Sec. 2.2], [221]). The main advantage of using WSNs for environmental monitoring is the increased spatial resolution in comparison to classical monitoring approaches that rely on significantly fewer but more complex and more costly sensors, which usually demand additional infrastructure for power supply and communication. Particular challenges of this application area arise from the relatively large area to be monitored, the usually large number of sensor nodes needed due to their limited sensing range, the nodes' environmental compatibility, and the relatively long operation time aspired.

For example, based on the application of early forest fire detection, this thesis shows that it is advisable to think further the vision of deploying WSNs with hundreds of thousands of sensor nodes to monitor areas of tens of square kilometers. The large number of sensor nodes has several implications:

- The price of each sensor node needs to be as low as possible.
- For a low environmental impact, each sensor node needs to be as small as possible.
- The price and size constraints render the use of powerful technology (e.g., long-range radio transceivers, fast processing units, or high-capacity batteries) infeasible.

* A list of all abbreviations used in this thesis is provided in Appendix Section A.1.

- Deterministic deployment is infeasible. Instead, the sensor nodes need to be deployed randomly, e.g., by dropping or spraying them from aerial vehicles. Immediately after random deployment, the position of each sensor node is hence unknown. Some localization method is needed to make the sensor nodes location-aware. The location of the sensed event can then be determined and communicated to the network operator.

- Regular maintenance of the sensor nodes is infeasible. This includes manual recharging* or replacing the batteries. Hence, despite the small size of the sensor nodes, a lifetime in the order of years should be aspired. To achieve this goal, the WSN needs to be designed such that it operates in a highly energy-efficient manner.

- For reducing the sensor nodes' energy consumption, their radio transceivers need to be switched to sleep mode as often as possible, because these are usually responsible for the sensor nodes' main power drain (see, e.g., [3, Sec. 3.7] and the references therein).

- The limited radio range and large size of the monitored area necessitates multi-hop communication of sensor data from the source sensor node to a destination node. Consequently, each sensor node does not only serve as a data source but also as a data router. This additional duty results in additional power consumption.

A prominent destination node of data messages communicated by a WSN is the *sink node*. The sink node usually is more powerful than the tiny sensor nodes and serves as a gateway between the WSN and a more capable external communication infrastructure, which takes care of sending sensor data from the WSN to the network operator and network management data in the opposite direction.

An important property of WSNs that are applied for safety-critical applications like forest fire detection is the time needed to communicate a detected event—i.e., fire—from the detecting source node to the sink. In this thesis, this time is referred to as *detection-to-notification delay*. For example, the size of a forest fire may—depending on the environmental conditions†—spread very quickly. Therefore, early intervention of fire fighters is of utmost importance. Hence, the detection-to-notification delay should lie in the order of less than only a few minutes (cf. Section 3.3). Clearly, there is a trade-off between this delay and the durations of the active and sleep periods that are adhered to by the nodes' transceivers and determine the nodes' lifetime. That is, the higher the probability that the transceivers are sleeping, the longer messages need to traverse the WSN.

The challenge addressed by this thesis is hence to find a scalable and energy-efficient communication protocol that is suitable for long-term environmental event monitoring. In particular, the protocol must be able to meet lifetime requirements on the one hand and detection-to-notification delay requirements on the other hand. Since an optimal protocol cannot be found, the design of an improved protocol is projected and the novel protocol's performance requires evaluation to proof its theoretical feasibility. Furthermore, the evaluation approach should be generic enough to allow for application-specific optimization of main WSN and protocol parameters.

* As discussed in Section 4.2.8, also the use of power harvesting methods seems impracticable in the investigated forest fire scenario.

† Including, e.g., fuel type, fuel moisture, wind speed, and air temperature.

1.2 Solution Approach

Several solutions for environmental monitoring in general (see Section 2.2) and solutions for WSN-based early forest fire detection in particular (see Section 2.1) have been proposed in related work. However, an investigation of these solutions reveals that real deployments so far comprise only up to hundreds of nodes, while this thesis exemplarily shows for the forest fire scenario that it seems advisable to foresee networks of hundreds of thousands of nodes. Proposed WSN-based solutions that explicitly address forest fire monitoring lack explicit lifetime estimations that consider the constraints following from network size, node size and detection-to-notification delay requirements. After reviewing solutions that seem feasible for large-scale and long-term environmental monitoring—in particular approaches that are based on cross-layer communication protocols that are based on receiver-based routing (cf. Section 2.2)—this thesis comes to the conclusion that there are still unexploited optimization opportunities to design a novel WSN protocol that is especially suitable for long-term environmental event monitoring where report-worthy events (e.g., an upcoming forest fire indicated by an exceeded temperature threshold) occur rarely.

This thesis hence designs a novel communication protocol, which is especially optimized for WSN-based monitoring of rare events in large-scale environmental areas. In the following, the protocol is referred to as *cross-layer message-merging protocol* (XLMMP). XLMMP is designed to comprise the following main properties, whose combination establishes the novelty of XLMMP:

Cross-layering. A cross-layer approach is chosen to increase the application-awareness of different protocol services (e.g., routing and medium access). Despite imminent disadvantages including decreased modularity, robustness, reusability, stability, and maintainability (cf. [166, 198]), cross-layer approaches bear a large potential to achieve significantly higher energy efficiency in comparison to layered protocol stacks (cf. [3, Ch. 10] and the references therein).

Message-merging. The protocol fosters in-network data message aggregation by allowing for application data-aware routing decisions. The resulting decrease of the number of messages further increases the protocols energy efficiency.

ID-less, unsynchronized, and self-localizing. In contrast to most protocols previously proposed in related work for environmental monitoring and WSN-based early forest fire detection, XLMMP does not rely on external services that provide unique node IDs or addresses, time synchronization, and node localization due to the following reasons:

- For unique node IDs and addresses, dedicated hardware (ID chips) or address negotiation protocols would be needed. In XLMMP, node IDs are completely avoided in order to reduce hardware requirements, energy consumption, and communication overhead. It should be noted that the lack of node IDs also naturally follows the paradigm shift from node-centric to data-centric networking (cf. [143, Ch. 12], [82, Ch. 11], [3, Sec. 7.2]).

- Regular time synchronization would require external time sources (e.g., the satellite-based Global Positioning System (GPS) or terrestrial beacon nodes) or dedicated time synchronization protocols that need relatively frequent communication. Similar to node

IDs, XLMMP avoids the need for time synchronization to reduce the corresponding hardware, energy, and/or communication overhead, in particular since it would usually need to be maintained even during long event-less periods to be useful.

- Node localization is often assumed to be achieved by the help of some satellite-based global navigation system (e.g., GPS), a dedicated localization protocol, or deterministic deployment. The drawback of these approaches is again an increased hardware, energy, and/or communication overhead. In contrast to node IDs and time synchronization, however, node localization is indispensable for achieving event localization. Hence, XLMMP provides node localization by itself with no hardware and only moderate additional communication and computation overhead.

Unclustered. In contrast to common ambitions to promote clustering in WSNs, this thesis refrains from following this trend. The decision is based on the following main reasons:

- In case of static cluster heads, these usually need to be provided with more powerful hardware. The resulting node heterogeneity, however, complicates deployment and introduces single points of failures.
- If the cluster heads are chosen dynamically, communication overhead is introduced for cluster head election, cluster formation, and cluster maintenance. This permanent overhead seems disproportional in comparison to the traffic caused by rare application events.

Receiver-based routing. To achieve high energy efficiency, the sensor nodes are subject to active/sleep cycles with significantly dominating sleep periods. To still keep the event-to-detection delay low, to foster message merging, to facilitate scalability and robustness, and to enable energy-aware routing, XLMMP applies receiver-based routing (cf. [6, 139, 198, 320], [3, p. 225]). Here, a sender node is not aware of its neighbor nodes that may serve as suitable next hops to forward the data message towards the sink. Instead, a sender broadcasts the message to its neighborhood by exploiting the broadcast character of wireless communication. The message-receiving neighbors then negotiate which of them signs responsible to forward the message further. If no next hop can be found immediately, the sender retries to find a suitable neighbor by resending the message.

The quantitative evaluation of XLMMP carried out in this thesis aims at investigating the influence of the WSN's and environmental parameters on the detection-to-notification delay. For deriving realistic parameters, the investigation focuses on a practical forest fire scenario. Still, it should be noted that this thesis does not yet strive for a perfect representation of the investigated WSN and XLMMP in form of detailed and large-scale simulations or testbed implementations. Instead, its main goals are to grasp the behavioral properties, principles, and dependencies of the underlying protocol mechanisms and to analyze the results' sensitivity on changes of the system parameters. In future work, the derived insights can then be used to focus on the most decisive factors when building more detailed and refined models.

A further advantage of this approach is that the proposed queueing models and the corresponding evaluation approaches are, to a large extend, application-independent and can be applied in a much broader context—even beyond the research area of communication networks.

The evaluation of the WSN is based on a model which is split into two main parts: the single-hop model and the multi-hop model. The *single-hop model* aims at reflecting the time needed by a message to traverse one hop of its multi-hop path to the sink node. The model in particular copes with the retrial behavior of the message-sending node and the active/sleep periods of the potential next-hop nodes. Also, the model reflects the limited number of distinct event messages, owing to the merging of similar messages. This is achieved by adopting the modeling formalism of retrial queues. More specifically, a finite-source retrial queue with homogeneous, unreliable servers is applied to deduce the single-hop response time and further performance measures. In contrast to related work on numerical evaluation of finite-source retrial queues with unreliable servers (see Section 2.4), this thesis not only derives mean performance measures (using numerical Markovian analysis provided by the tools MOSEL-2 and SPNP)* but also the distribution of the waiting and response times using the Method of Phases (cf. [169, Part II]). Additionally, an approximation is proposed that allows to increase the scalability of the evaluation method in case of large state spaces. The approximation's derivation relies on symbolic regression (cf. [263]) provided by the tool Eureqa. The proposed single-hop modeling and evaluation process is verified† by comparing the results to outcomes of related work and of a suitable discrete-event simulation (DES) which developed in this thesis and is implemented using CPN Tools.

For the evaluation of the protocol's *multi-hop* detection-to-notification delay, the sensor nodes' hop count‡ distribution is investigated by developing an multi-agent simulation model. This model is implemented using the MASON toolkit. The combination of the single-hop response time distribution with the hop count distribution is implemented in form of a numerical approach based on a Monte Carlo experiment and allows to calculate the aspired detection-to-notification delay distribution. Based on this result, the thesis in particular is able to answer the following interrelated questions:

- For a given set of system parameters, which upper bound of the detection-to-notification delay is met by a given fraction of events?
- For a given set of system parameters, what is the fraction of events that meet a given upper bound of the detection-to-notification delay?

The obtained quantitative results show that XLMMP works well under given assumptions and allow to identify the model parameters for which key WSN performance measures show high sensitivity.

* The most important software and analysis tools applied are described in more detail within the thesis (in particular in Sections 5.3.4, 5.4.3.1, and 5.5.2).

† Here, *verification* refers to the process of getting confidence in the statement *"the models and their evaluation are implemented correctly"*. For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

‡ Throughout this thesis, the term *hop count* refers to the distance of a node to the closest sink, measured in the number of required communication hops.

1.3 Main Contributions

The main contributions of this thesis can be summarized as follows:

1. The thesis combines the research fields of environmental monitoring (with a focus on the application of forest fire detection), wireless sensor networks, and retrial queues. State-of-the-art forest fire detection methods are classified by differing human-based and technical methods, while the latter are further classified according to their location with respect to the forest (cf. Section 3.3). For WSN-based forest fire detection, the thesis identifies the lack of tangible lifetime estimations as a major open research issue (cf. Section 2.1). The thesis proposes a classification of cross-layer WSN protocols that apply receiver-based routing. The classification considers the type of packet (data or control) that is initially transmitted by a sender node and captures whether the protocols execute receiver initiative determination and/or a receiver contention mechanism (cf. Sections 2.2 and B.1). A basic classification of the literature on retrial queueing models is proposed based on the number of servers, their reliability, and the number of sources (cf. Section 2.4). Previous works that use retrial queues to model WSNs are shown to lack investigation of the detection-to-notification delay distribution (cf. Section 2.3).

2. This thesis designs a novel communication protocol, referred to as *cross-layer message-merging protocol* (XLMMP). The protocol's behavior (cf. Section 4.1) and message format (cf. Section 4.3.6) is tailored towards the timely and energy-efficient communication of events within large-scale WSNs applied for long-term environmental event monitoring. It is shown how the desired functionality, including node localization and data aggregation, can be achieved without relying on node synchronization, unique node addresses, and dedicated localization hardware.

3. To proof the feasibility of XLMMP by *quantitative* evaluation, a variety of evaluation models is specified. To a large extent, the proposed models are application-independent and can be applied in a much broader context. This particularly holds for the mathematical models that are based on retrial queues. For evaluation, all models are implemented—if available, based on suitable software tools. Different evaluation process variants are compared, in particular according to their scalability with respect to the size of the underlying state space. The proposed models and their addressed aspects can be briefly summarized as follows:

   - A *hop count simulation model* is developed to simulate, investigate, and visualize the random deployment of sensor nodes, the coverage and connectivity achieved, the formation of hop count rings around the sinks, and the distribution of the nodes' hop count to the sinks (cf., e.g., Sections 4.3.2.2, 5.7.1, 5.8.2, 5.30, and Figs. 4.14 and 4.18).
   - A *retrial queueing model* is obtained to investigate single-hop mean steady-state performance measures by applying Markovian analysis to the underlying three-dimensional irreducible continuous-time Markov chain (cf. Sections 5.2 and 5.3).
   - Additionally, a three-dimensional *reducible continuous-time Markov chain* variant is constructed that allows to derive the distribution of the waiting time of retrying jobs which is shown to be phase-type distributed under the given assumptions (cf. Sections 5.4 and B.2).

- An *approximate model* of the retrying jobs' waiting time distribution is developed to increase the numerical evaluation's scalability in case of large state spaces.
- A *single-hop DES model* based on colored Petri nets is built to verify the correctness of the numerical single-hop models' implementation and to serve as input for developing the approximate model in parameter ranges where numerical analysis fails due to state-space explosion (cf. Sections 5.3.4, 5.4.3.2, 5.5.2 and B.7).
- A multi-hop model is built and evaluated using a *Monte Carlo experiment*. It allows to find the multi-hop detection-to-notification delay distribution based on the single-hop results and the hop count distribution (cf. Section 5.7).

4. XLMMP is *qualitatively* evaluated (cf. Section 4.4.1) based on design principles stated in related work (cf. [143, Sec. 3.3]) and by comparing its behavior to the related protocol XLP (cf. [6, 198, 320] or [3, Sec. 10.3])*.

5. The thesis proposes and discusses several approaches to estimate the nodes' position within their hop count ring. These approaches can be used to improve the accuracy and performance of hop count-based localization and routing methods, respectively (cf. Sections 4.3.2.5 and 4.3.2.6).

6. The thesis proposes a hex grid-based approach that provides short addresses for ranges of locations within the monitored area. This keeps the message size small, helps to identify spatial correlation between data packets, and fosters the merging of messages. Transformations between the hop count, physical, and grid locations are provided (cf. Sections 4.3.5 and B.4).

7. The thesis fosters the practical applicability of the proposed protocol and the applied evaluation methods by attaching importance to deriving and working with realistic environmental, WSN, and protocol parameters. For example, this is reflected by the following measures:

   - The protocol's design and evaluation is embedded in a realistic forest scenario which is based on the Neuburg Forest located near Passau, Germany, (cf. Sections 4.2.2 and following).
   - The power consumption of typical state-of-the-art radio transceiver hardware is collected from related work and used for the derivation of lifetime estimations (cf. Section 4.2.8).
   - State-of-the-art data storage technologies are briefly investigated to estimate realistic volumetric storage densities (cf. Section B.3).

* Note that [6] proposes the *cross-layer module* (XLM). According to [320], the publication [6] is a "preliminary version" of [320], which proposes the *cross-layer protocol* (XLP). Since the basic mechanisms of XLM and XLP coincide, publications [6] and [320] are jointly discussed in this thesis. Publications [198] and [3, Sec. 10.3] are examples of secondary literature that discuss XLM and XLP, respectively.

1.4 Thesis Structure

Related work on forest fire detection with WSNs, WSN protocols, and finite-source retrial queues is discussed in Chapter 2.

In Chapter 3, the scenario of early forest fire detection is investigated as a typical example of environmental event monitoring. Application-specific background is provided and requirements for early forest fire detection methods are discussed on the basis of state-of-the art solutions. The chapter also helps to embed the rather theoretical contributions of later chapters into an accessible application scenario. This further motivates the theoretical contributions by showing their practical relevance and provides a basis for narrowing the range of investigated system parameters to relevant scales.

Based on the forest fire scenario, Chapter 4 first introduces the basic behavior of XLMMP. It then presents essential assumptions and basic design decisions the protocol is based on, before discussing more technical details and carrying out a qualitative evaluation of XLMMP.

Chapter 5 focuses on the quantitative evaluation of the proposed protocol. The single-hop model is treated in Sections 5.2 to 5.6. The multi-hop is presented in Section 5.7. The quantitative results are discussed in Section 5.8 and allow to infer suggestions regarding protocol parameterization and model refinements.

Finally, Chapter 6 concludes the thesis by providing a summary and directions for future work including a sketch of further research steps towards a testbed implementation of the proposed forest fire-detecting WSN and a brief discussion of the thesis's impact on the research landscape.

The thesis is complemented by two appendices. Appendix A summarizes the abbreviations, mathematical notation, and parameters used in this thesis. Finally, bulky illustrations, lengthy derivations, auxiliary calculations, and secondary discussions are provided in Appendix B.

A *technical report* summarizing the experience with software tools gathered while working on this thesis and providing technical details on the model implementations is projected. The interested reader is referred to `http://wuechner.eu/thesis/` for details. Future research and developments related to this thesis are also planned to be made available at this Web address.

Chapter 2

Related Work

If I have seen further it is by standing on ye sholders of Giants.

— Isaac Newton

The most relevant related publications with respect to this thesis's main contributions are summarized in this chapter. The thesis is concerned with the long-term *environmental monitoring* and event communication using *WSNs*. In particular, the application area of early *forest fire detection* is considered. For evaluating the proposed WSN protocol, stochastic models based on finite-source *retrial queues* are developed and applied. The relationship of these topics is illustrated in Fig. 2.1.

As a rough indicator of the attention each topic received by the research community, the approximate number of Google™ Scholar* hits is provided in Fig. 2.1 for each topic.

For the four topics highlighted by a star symbol in Fig. 2.1, related work is discussed in detail in Sections 2.1 to 2.4. The presentation roughly follows the order in which the topics are tackled in Chapters 3 to 5 (from left to right in Fig. 2.1).

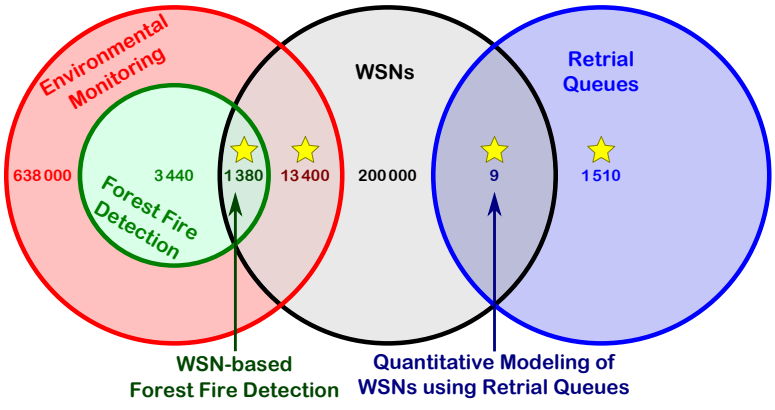


Figure 2.1: Relationship of research topics broached in this thesis.

* `http://scholar.google.de/` (last accessed: 27 May 2013). Hit numbers given as of 11 April 2013. Search terms used (from left to right): "environmental monitoring"; "forest fire detection"; "forest fire detection" "wireless sensor networks"; "environmental monitoring" "wireless sensor networks"; "wireless sensor networks"; "retrial queues" "wireless sensor networks"; "retrial queues". Note that not all hits refer to publications that actually focus primarily on the respective topic. That is, publications that just mention the respective topics are also included.

Table 2.1: Sensor types suitable for WSN-based forest fire detection

Sensor Type	Example References
Temperature	[25, 37, 81, 156, 158, 184, 216, 284]
Relative humidity	[25, 37, 81, 158, 184]
Smoke particles	[37, 81, 158, 184, 284]
Optical radiation (visible or IR)	[37, 158, 184, 284]
Barometric pressure	[25, 37, 81]
Carbon dioxide	[158, 184]
Carbon monoxide	[158]
Methyl chloride	[158]
Wind speed	[158]
Passive microwave	[158]

2.1 Forest-Fire Detection with WSNs

The detection of forest fires is an application example that is mentioned frequently in literature on WSNs (e.g., in [4], [360, p. 297], [143, p. 3], [305], [318, p. 17], [52], [3, p. 21]). In the following, concrete proposals in this direction are surveyed by focusing on WSNs with immobile sensor nodes. Alternative forest fire detection methods are briefly discussed in Chapter 3, which is dedicated to motivating and introducing the application scenario of forest fires and early forest fire detection.

A significant number* of publications explicitly tackles the topic of wild- and forest fire detection and monitoring using WSNs. Surveys of this research are provided in [31, Sec. 3] (up to year 2008), [24, Sec. 2], [26, Sec. 3.2.1], and [25, Sec. 2] (all up to year 2010). More recent publications include [37, 38, 69, 70, 134, 157, 176, 186, 216, 284, 306].

In [17, 37, 38, 55, 69, 70, 81, 115, 156, 184, 216, 284], the basic applicability of WSNs for forest fire detection is shown by outdoor experiments with a relatively small number of nodes†. Most commonly, the detection is based on temperature and humidity sensors. A list of more sensor types used or suggested for forest fire detection and monitoring is provided in Table 2.1. The sensor nodes had direct contact to flames in [17, 81, 216]. These works indicate that a trade-off is needed between investing effort to insulate the nodes (cf. [17]) and considering them as dispensable (cf. [216]). The former approach increases the nodes' size, production complexity, and cost while decreasing the nodes' sensitivity and consequently its sensing range. Hence, it likely increases the required node density. When following the latter approach, the nodes need to be cheap, environment-friendly or even compostable/biodegradable, and easily (re-)deployable. In any case, due to the relatively small sensing range and resulting large number of nodes required, the practicability of approaches based on high-end sensors that are laboriously installed manually in tree crowns (as proposed in, e.g., [184]), stands to reason.

Some publications, including [24, 25, 37, 119, 186, 213, 356], also tackle networking aspects, i.e., the formation of a network topology and the communication of sensor readings or detected forest fire events to the operator. None of these publications, however, give ex-

* At least 50 publications in English between 2003 and April 2013.
† Up to 44 nodes in [156].

plicit lifetime estimations considering the trade-off between node size, communication delay, and energy efficiency. In particular, most solutions assume—often implicitly—that all nodes have unique IDs, are time-synchronized, and their location is known to themselves and/or the operator. However, the hardware and energy requirements for achieving these properties are usually not discussed in these works. A significant portion of publications (e.g., [37, 55, 186, 187, 285, 356, 357]) use, or suggest to use, a combination of the IEEE 802.15.4 (cf. [127]) and ZigBee (cf. [364]) protocol standards, stressing their energy efficiency. In [213], flat and clustered communication topologies are briefly compared in the context of forest fire-detecting WSNs. The authors conclude that clustering should be preferred because in flat topologies, nodes closer to the sink deplete their energy more quickly. Several publications on fire-detecting WSNs (e.g., [24, 25, 119, 134, 186, 306, 353, 356, 357]) also prefer a clustered topology.

In contrast to related work, this thesis avoids the IEEE 802.15.4 standard mainly due to the following two reasons.

- IEEE 802.15.4 relies on *node addresses*. Each device has a globally unique 64 bit address. Depending on the topology, a shorter, locally unique 16 bit address may be negotiated for each device. As discussed in Section 4.2.15, using such addresses would add significant overhead in terms of message length and address maintenance* in the scenario discussed by this thesis. Instead, this thesis follows the paradigm shift from *node-centric* (i.e., address-based) to *data-centric* networking (cf. [143, Ch. 12], [82, Ch. 11], [3, Sec. 7.2]) by fully avoiding the need for node IDs or addresses.
- IEEE 802.15.4 aims at duty cycles down to approximately 1% (see [130, p. 205], [360, p. 68], [318, p. 127f], [3, p. 114]), i.e., the nodes' transceivers are assumed to be active more than 1% of time. As discussed in Section 4.2.8, this thesis aims at duty cycles in the order of 0.04%.

The approach chosen by this thesis also avoids the use of clustering in the investigated application scenario in particular due to the following reasons. Because of their additional duties, cluster heads require more energy than non-head nodes (cf. [318, p. 251]). If the cluster heads are chosen statically, they need to be provided with significantly more power and probably also additional processing, radio, and memory resources. This introduces node heterogeneity, complicates deployment, and increases the chance of single points of failures. In particular, it should be easier to replace depleted nodes that are located close to a usually deterministically placed sink than re-deploying cluster head nodes at random locations within the monitored area. If, on the other hand, the cluster heads are chosen dynamically, the overhead of cluster head election, cluster formation, and cluster maintenance is considered disproportional to the low rate of events that need to be reported in the investigated application scenario.

In summary, this thesis investigates—in contrast to related work on WSN-based early forest fire detection—protocol approaches that are applicable for large-scale networks where the nodes are not provided with unique IDs, no effort is invested for keeping the network time synchronized or clustered, and node localization is achieved by the protocol itself. Here, the

* For example, the negotiation of locally unique addresses and the resolution of address conflicts, the hardware and energy costs for providing and using globally unique addresses, and the computation needed by sensors to maintain lists of addresses, like routing tables.

lifetime of the network, the size and cost of the nodes, and the achieved communication delay are considered as the major design parameters in the application scenario of early forest fire detection.

2.2 WSN Communication Protocols for Environmental Event Monitoring

The literature on wireless sensor networks is extensive, diverse, and rich. Up to today*, the two early surveys by Ian F. Akyildiz et al., [4] and [5], received more than 10100 and 10180 citations†, respectively. There are several scholarly books (including [3, 82, 143, 147, 203, 237, 293, 318, 354, 360]) and PhD theses (e.g., [108, 125, 226, 328, 340, 348]) that focus on WSNs. Further surveys on WSNs in general include [9, 35, 97, 142, 233, 309, 324, 350]. A significant number of surveys address specific aspects of WSNs, e.g., applications (e.g., [8, 18, 26, 31, 85, 221]), multimedia (e.g., [272]), transport protocols (e.g., [323]), addressing (e.g., [316, 349]), routing and clustering (e.g., [1, 7, 84, 112, 352]), medium access control (MAC) protocols (e.g., [30, 73, 78, 124]), time/clock synchronization (e.g., [168, 239, 282, 296]), in-network processing (e.g., [162, 358]), connectivity (e.g., [363]), coverage (e.g., [212, 322, 363]), sensor selection (e.g., [251]), localization‡ (e.g., [15, 32, 48, 62, 71, 85, 110, 178, 225, 287, 325]), lifetime (e.g., [75]), energy sources (e.g., [250, 289, 294]), security (e.g., [60, 118, 190, 211, 273, 287, 321, 351]), fault tolerance (e.g., [208]), congestion control (e.g., [136]), debugging (e.g., [266]), real deployments (e.g., [52]), node mobility (e.g., [15, 78]), operating systems (e.g., [79]), or self-organization (e.g., [83]). In particular, the authors of [221] survey WSN solutions proposed for the application of environmental monitoring. They emphasize that real deployments so far consist of tens to hundreds of nodes. To achieve the required network lifetime, the involved communication protocol architecture needs to be optimized with respect to energy efficiency. Recent research, including [3, Ch. 10] and the references therein, concludes that cross-layer designs of application-aware communication protocols—despite usually also having disadvantages, like decreased modularity, robustness, reusability, stability, and maintainability (cf. [166, 198])—are able to achieve a significantly higher energy efficiency than layered approaches. Based on this insight, several cross-layer communication protocols have been proposed for WSNs in the last years. Prominent ones are collected, surveyed, and classified in, e.g., [198], [166, Sec. 4.2], [96], and [199].

Receiver-based routing (cf. [6, 139, 198, 320], [3, p. 225]) is such a cross-layer approach. It combines (at least) the MAC and network layer of the classical seven-layer ISO/OSI (cf. [304, p. 28]) or five-layer Internet (cf. [165, p. 48]) protocol stacks. Receiver-based routing is also called *receiver-oriented routing/forwarding* (cf. [90]), *opportunistic routing/forwarding* (cf. [29, 42, 148, 149, 150, 260, 328, 329, 347, 362]), *passive routing* (cf. [283]), *(MAC-layer) anycast routing/forwarding* (cf. [63, 89, 148, 149, 150, 151, 163, 183, 223, 328, 329, 347], [3, p. 225]), or *anypath routing* (cf. [260]). Note that there are (mostly author-specific) differences in interpreting the terminology. while all papers roughly follow the same idea, the properties of the actually used mechanisms vary. In contrast to classical protocols that are tailored towards wired communication, receiver-based routing exploits the broadcast character of the wireless

* April 2013.

† According to Google™ Scholar (`http://scholar.google.de/`, last accessed: 27 May 2013).

‡ Node localization is discussed in more detail in Section 4.2.12, where also different techniques proposed in related work are discussed with respect to their suitability in this thesis's application scenario.

air interface. Its strength is that a sender does not need to be aware of the recipient before initiating the connection. Instead, it broadcasts the message (or first control packet of the handshake) to all of its neighbors. The neighbors then negotiate who will act as recipient. This negotiation is often achieved in two steps. First, each neighbor decides whether it is able to suitably contribute to the communication at all. This binary decision is usually denoted as *(receiver) initiative determination* or *assessment* (cf. [6, 90, 198, 320]). Often, the decision is mainly based on the receiver's location*, but it may also include further threshold metrics, like the neighbor's remaining energy, message buffer and traffic status, or signal-to-noise ratio (SNR) of the received packet (cf. [6, 198, 320]). In a second step, all neighbors that consider themselves as suitable receiver candidates perform a *receiver contention* mechanism† (cf. [6, 90, 139, 198, 283, 320, 347, 365, 366], [3, p. 225]), which is usually implemented in form of a back-off mechanism. That is, the receiver candidates respond after some delay whose duration is calculated by each receiver candidate based on its local situation (like location, energy status, signal-to-noise ratio (SNR), or utilization). The neighbor that responds first is then chosen as recipient. The approach deals well with unstable network topologies, which change frequently due to, e.g., the nodes' active/sleep periods (cf. [149]) and variable radio conditions (cf. [362]).

In Tables B.1 to B.3 of Appendix Section B.1, an extensive overview of previously proposed receiver contention-based, anycast, and/or opportunistic routing protocols is provided. A more compact representation is provided by Table 2.2.

Table 2.2: Related work on protocols based on receiver-based routing (• =yes, ○ =no; RID=receiver initiative determination, RCM=receiver contention mechanism)

References	Year	Initial Sender Packet	Class	Receivers Perform	
				RID	RCM
[206]	2001	DATA	3	○	•
[44]	2003	CONTROL	1	•	•
[365, 366]	2003	CONTROL or DATA	1 or 3	○	•
[63]	2004	not discussed	?	○	○
[283]	2004	CONTROL	1	○	•
[42]	2005	DATA	3	•	•
[6, 198, 320], [3, Sec. 10.3]‡	2006–2010	CONTROL	1	•	•
[163]	2006	CONTROL with optional application data	1 or 2	•	•
[89]	2007	CONTROL with aggregation support information	2	•	•
[183]	2007	CONTROL	1	•	•
[27, 28, 29]	2008–2010	DATA (CONTROL as variant)	3 (1)	○	•
[148, 149, 150, 151]	2008–2011	CONTROL	1	•	○
[223]	2008	CONTROL	1	•	○
[260]	2009	DATA	3	•	•
[139]	2010	CONTROL	1	○	•
[347]	2010	CONTROL	1	•	•
[328, 329]	2011	CONTROL	1	•	○
[90]	2012	CONTROL	1	•	•

* That is, the neighbor only acts as receiver if it is closer to the destination than the sender. This decision can be based on geographical or topological location of the nodes.

† More seldom, *receiver contention* is also referred to as *relay self-selection* (cf. [29]).

‡ Note that [6] proposes the *cross-layer module* (XLM). According to [320], publication [6] is a "preliminary version" of [320], which proposes the *cross-layer protocol* (XLP). Since the basic mechanisms of XLM and XLP coincide, [6] and [320] are jointly discussed in this thesis. Publications [198] and [3, Sec. 10.3] constitute secondary literature that discusses XLM and XLP, respectively.

This thesis partitions the listed protocols into three classes:

1. *Class-1 protocols* where the sender initially sends a *control packet* that is not foreseen to contain application layer data (e.g., [44]; [63]; [283]; [6, 198, 320], [3, Sec. 10.3]; [183]; [148, 149, 150, 151]; [223]; [139]; [347]; [328, 329]; [90]).

2. *Class-2 protocols* where the sender initially sends a *control packet* that may be enriched with some application layer data (e.g., [163]; [89]).

3. *Class-3 protocols* where the sender initially may send a *data packet* that comprises all relevant application layer data (e.g., [206]; [365, 366]; [42]; [27, 28, 29]; [260]).

Class-1 and Class-2 protocols determine the next-hop node *before* sending the data packet. The data packet is then sent to the selected next-hop node in a unicast manner, expecting that the radio conditions do not change significantly and the selected node is still reachable. For this unicasting, the selected receiver needs to be uniquely identifiable—either by providing (locally) unique addresses or by knowing its location with sufficiently fine granularity, i.e., such that two nodes do not share the same location. The control packets (usually denoted as *request-to-send* (RTS) and *clear-to-send* (CTS) packets) employed by these protocols are also used to tackle the hidden-node problem (cf. [3, p. 81]).

When using a Class-3 protocol, the receiver is selected *after* the data packet has been sent. Hence, only receiver candidates that actually received the data packet correctly will offer themselves as next hop. Additionally, Class-3 protocols in principle allow the set of receiver candidates to additionally consider the full application layer data when calculating their individual back-off delay. Hence, receiver candidates that already hold similar packets can advertise themselves (by keeping their back-off delay short), since they are able to merge the packets by aggregating correlated data. The decrease in the number of data packets that results from this aggregation can lead to a significant reduction in traffic and energy consumption. This advantage, however, is not exploited by most Class-3 protocols. The given Class-3 protocols show the following properties.

- The authors of [206] focus on developing custom tools and models for evaluating high-density, energy-conscious sensor networks. The address-free multi-hop protocol used in the evaluations is described roughly only. Although the full application data is available to the set of potential receivers, the receiver contention is only based on the receivers' distance to the sink. In particular, aggregation of packets is not foreseen. Moreover, the evaluation focuses on the sensor network's energy efficiency. The end-to-end delay is not discussed.

- In [365, 366], the receiver contention-based *geographic random forwarding* (GeRaF) is proposed. While the authors also foresee the use of using data packets only, their main discussion is focused on an RTS/CTS-based (Class-1) approach, which—as discussed above—needs some means for uniquely addressing the selected next hop in the data packet. Including application layer data when calculating the back-off delay is not foreseen. Instead, the selection of the next hop is solely based on its geographic location in relation to the sender and the destination node, whose location always needs to be included in the packets. The authors additionally suggest to use dual-radio transceivers to prevent packet collisions by sending additional busy tones. This, however comes with significant energy and hardware costs.

- The *ExOR* protocol proposed in [42] is tailored towards maximizing the throughput. While the receivers content for being the next hop, their preferred order is specified in a priority list sent by the source node together with the data. This priority list is constructed by the source node based on the involved links' delivery probability. Hence, all nodes need to be uniquely identifiable and aware of the complete set of inter-node loss rates. Communicating this link state information to all nodes and keeping it up to date comes with significant overhead. Additionally, the number of potential receivers needs to be limited to keep the length of the priority list and the costs for agreement communication low. Moreover, packets are sent in batches and the receiver contention mechanism is only used for 90% of a batch's packets. The remaining 10% are sent using traditional routing. Hence, the full functionality of traditional routing (including unique addresses, routing tables, etc.) needs to be provided in addition to the ExOR functionality. Furthermore, [42] does not consider data packet aggregation.

- In the protocol proposed in [27, 28, 29], all receivers start to answer simultaneously after the sender broadcasted the data to its neighbors. The answer consists of an individual signaling burst that encodes the rank (most significant bit first) of the corresponding neighbor. The neighbor that sends the first unique bit is determined as receiver. The rank of the neighbors is determined based on the nodes' location or distance progress only. Aggregation of data packets is not foreseen.

- The approach chosen by [260] is quite similar to *ExOR* (cf. [42]). Again, the sender sends a priority list of uniquely identifiable receivers within the data packet. In contrast to [42], the focus of [260] is set on energy-efficient data gathering instead of throughput improvement. In [260], however, all potential receivers are expected to acknowledge all data packets. The sender again needs to be aware of its uniquely identifiable neighbors and the transmission costs associated to them. Data packet aggregation is not considered.

In this thesis, to design the novel energy-efficient WSN communication protocol XLMMP suitable for long-term and timely environmental monitoring, a single-radio cross-layer approach is chosen, where the employed receiver-based routing applies initiative determination *and* receiver contention (comparable to [44]; [42]; [6, 198, 320] and [3, Sec. 10.3]; [163]; [89]; [183]; [260]; [347]; [90]). Similar to all of these works, the neighbors' location plays a major role in determining the receiver initiative or receiver contention back-off delay in XLMMP. However, in contrast most of these works, XLMMP uses topological locations* instead of geographical locations. This avoids the situation where messages are routed into dead ends, where a sender cannot find a neighbor which is geographically closer to the sink. Topological locations are also used by [42, 163, 260]. More precisely, [42] and [260] use an expected transmission count metric which includes the number of hops to the sink and the expected number of retransmissions. This, however, requires all nodes to be aware of the inter-node loss rates of the network which requires frequent flooding of link state information updates. Therefore, similar to [163], only hop count information is used by XLMMP. In contrast to [163], however, XLMMP does not rely on transmission power control, node IDs, and RTS/CTS control packets. Moreover, XLMMP uses the receiver contention's back-off delay to favor nodes with mergeable data, more remaining energy, and less buffer utilization. Further information can be

* That is, XLMMP uses the nodes' position in the WSN's communication topology which is reflected in the nodes' hop count value.

included, for example, more fine-grained information on the receiver's location, if available, and event priorities or event dimensions.

Similar to the Class-3 protocols, the sender in XLMMP immediately sends a data packet to provide the necessary data to the neighbors before receiver contention is carried out. While this approach is susceptible to the hidden-node problem, it is shown in Section 4.3.6 that the size of the data packet used in this thesis is very small (less than 70 bit). Hence, a colliding data packet does not result in significantly more overhead than a colliding control packet. On the contrary, due to the reduction of overhead by avoiding these control packets, the proposed protocol utilizes the wireless channel less and the packet collision probability can hence be expected to decrease. By immediately sending the data packet, the overall delay decreases in comparison to the RTS/CTS-based Class-1 and Class-2 protocols.*

Finally, destination addresses or locations are used explicitly in, e.g., [6, 198, 283, 320, 365, 366] and [3, Sec. 10.3]. In the application scenario tackled by this thesis, however, application data is always addressed to any sink node. Hence, additional overhead can be avoided by refraining to add destination addresses or locations to the data message headers of XLMMP.

2.3 Quantitative Modeling of WSNs Using Retrial Queues

Due to the unreliability of the wireless channel and due to the uncoordinated duty cycles of the nodes in combination with the receiver contention-based communication protocol, a sender likely fails to forward a packet immediately. Instead, a sender usually needs to conduct several retrials before a next-hop node is found. Moreover, a sender is not immediately aware of an awakening next-hop node. Retrial queues hence appear to be a self-evident formalism for modeling this mechanism. However, there is only a small number of publications that actually apply retrial queues for the modeling of WSNs. These are discussed in this section.

To the best knowledge of the author, [335]$^{\circlearrowleft}$ † is the first publication that explicitly mentions the possible application of retrial queues for modeling WSNs. The paper focuses on the investigation of finite-source multi-server retrial queues with balking, impatience, and orbital search. It, however, does not conduct an evaluation of WSNs by itself.

The workshop paper [336]$^{\circlearrowleft}$ appears to be the first publication that actually evaluates a receiver contention-based WSN scenario by the help of the finite-source retrial queue modeling formalism. The paper focuses on a single hop within the multi-hop WSN only. Moreover, only mean performance measures are obtained using Markovian analysis provided by the MOSEL-2 tool‡. It additionally considers the case where the orbit is unreliable (i.e., a node that holds a data packet might switch to sleep mode) and a configurable probability that a node gets informed if a next-hop node changes its state from busy to idle (orbital search).

The author of [76] presents an infinite-source priority retrial queueing model that comprises an ordinary (first-come-first-served) queue that is used by high-priority jobs, an orbit that is used by low-priority jobs, and a single server with vacations. Using this model, mean performance measures are derived. Referring to [336]$^{\circlearrowleft}$, the author considers WSNs to be a possible application which can be evaluated by the proposed model.

---

* This can be shown by comparing the message sequence charts of the proposed XLMMP and, e.g., XLP (cf. [6, 198, 320], [3, Sec. 10.3]), which is a RTS/CTS-based Class-1 cross-layer receiver contention protocol. The comparison is provided in Section 4.4.1.2.

† References to publications (co-)authored by the author of this thesis are marked with $\circlearrowleft$.

‡ More information on the MOSEL-2 tool is provided in Section 5.4.3.1.

The authors of [51] adopt and partly merge the ideas presented in [336]° and [76]. While still considering a single-server priority retrial queue and focusing on mean performance measures (obtained using the MOSEL tool), a finite number of sources is assumed.

Without explicitly referring to the concept of retrial queues, Wang (cf. [328] and the companion paper [329]) recently proposes a generic framework for the modeling and evaluation of a large class of WSN MAC and routing protocols. Wang also uses this framework to exemplarily investigate an anycast protocol variant. The inclusion of additional protocol variations, however, appears to involve significant effort, since it requires to respecify large parts of the model. The single-hop models used by Wang are based on discrete-time Markov chains (DTMCs). One drawback of the DTMC-based approach is the need for selecting a suitable time unit. For this, a trade-off between model complexity and accuracy is needed. The smaller the time unit is chosen, the more states are needed to model the same duration. In both works, the aggregation of messages is not foreseen. This is reflected in two ways. First, an infinite-source model is applied, i.e., the arrival process of distinct messages is not dependent on the number of messages already known to the node under investigation, as long as the node is not full. Second, the investigated anycast protocol uses beacon messages to inform next-hop nodes of a sender's desire to forward data. These beacon message do not contain information that supports aggregation. The considered retrials are node-centric, i.e., at each point in time, each node maintains a single retrial process only. Hence, the overall rate of retrials generated by a node is independent of the number of messages stored at the node. The retrials are reflected by a phase-type (PH) service process. This requires to set an upper limit of retrials in order to limit the size of the model's state space. While the model's scalability is not discussed in detail, a maximum of three retrials is considered as typical in [328, p. 69]. The duty cycles are considered together with other factors (like channel noise and packet collisions) in form of a transmission success probability parameter. A two-dimensional Markov chain is then sufficient to describe the underlying stochastic process, where one dimension is used to model the number of messages in the node and the second dimension models the state of the PH service process. Calculating the transmission success probability, however, turns out to be complex (cf. [328, Sec. 3.6.3]). In particular, it depends on steady-state results of the model, i.e., the model needs to be evaluated iteratively until the results converge. The obtainable results include the distribution of the single-hop delay. In contrast to [336]°, [76], and [51], Wang additionally investigates the multi-hop end-to-end delay distribution of communicating the data between two nodes of the network. For this, Wang assumes that each node maintains a probabilistic routing table, i.e., the packets are forwarded to the neighbors according to the probabilities specified in the routing table. In case of random deployment, Wang uses fine-grained location information to identify the nodes. For the anycast protocol, the routing probabilities are calculated by considering each neighbor's probability to be available when the sender transmits its request. The end-to-end delay distribution from any source node to the sink is obtained in dependence of the node's distance from the sink via iterative convolution (cf. [328, Eq. (3.37)]) of the single-hop delay distribution. The scalability of this approach is not discussed in detail. For a setup with approximately five hops from source to sink, the calculation of the probability that a certain end-to-end delay bound is met takes up to approximately 20 seconds for each combination of parameter values and delay bound. Wang additionally distinguishes between end-to-end delay and event detection delay, where the latter assumes that several messages need to be received by the sink to allow the detection of the event.

This thesis quantitatively evaluates the proposed protocol (XLMMP) in a large-scale application scenario, where 10^5 nodes communicate to multiple sinks via up to approximately 80 hops. Similar to [328], this thesis splits the investigation into a single-hop and a multi-hop model. For the single-hop model, this thesis uses models based on finite-source retrial queues with unreliable servers. In contrast to [336]$^{\circlearrowleft}$, [76], and [51], this thesis considers not only the mean performance measures but also the distribution of the mean response time. The derivation of the distribution of the multi-hop end-to-end delay in the presence of additional mechanisms (unreliable orbit and orbital search) discussed in [336]$^{\circlearrowleft}$ is considered as future work in this thesis. Moreover, the models proposed in [76] and [51] are not applicable, since they focus on single-server retrial queues. Further related work on finite-source retrial queues with unreliable servers that is not tailored towards the WSN application scenario is discussed in Section 2.4. In contrast to the DTMC-based single-hop model suggested in [328], this thesis uses a continuous time parameter. Hence, the size of the state space is independent of the selected time unit. By using the retrial queue approach, the size of the state space is also independent of the number of retrials. In further contrast to [328], the retrials considered in this thesis are message-centric, i.e., each message conducts its individual retrial process. The model presented in this thesis explicitly models the duty cycles of the sensor nodes. Hence, an iterative process to obtain transmission success probabilities, as followed by [328], is not needed. To obtain the distribution of the multi-hop end-to-end delay in face of the application scenario's large scale, this thesis, in contrast to [328], refrains from using iterative convolution for scalability reasons. Instead, a numerical approach based on a Monte Carlo experiment is applied.

2.4 Retrial Queues

The single-hop model presented in this thesis is based on the modeling formalism of retrial queues. In the last two decades, significant effort was invested by the research community to investigate this class of queueing models—not only because of their non-triviality, but also because of their broad applicability, e.g., for the evaluation of communication networks. Overviews of literature on the theory of retrial queues and their practical application is provided in the books [23, 88], the bibliographies [20, 21, 22], and the surveys mentioned therein. The literature on retrial queues can be classified in a large variety of dimensions. Figure 2.2

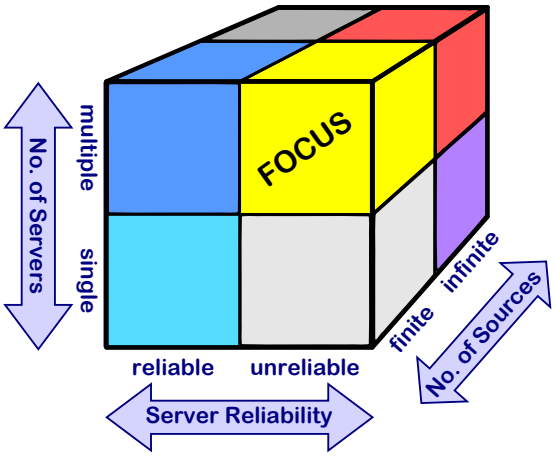


Figure 2.2: Rough classification of literature on retrial queues.

illustrates the three dimensions that are considered as the most relevant in this thesis. As motivated in detail in Section 5.2, this thesis focuses on the discussion of finite-source retrial queues with multiple, unreliable servers. There are several related papers that fall into this category, including [45, 47, 99, 100, 101, 102, 246, 247, 300, 301] and [333, 336]. These papers focus on different additional variants of the model, e.g., including heterogeneous servers with different server selection strategies, simultaneous breakdowns, unreliable orbit, or orbital search. All these papers, however, only investigate mean performance measures.

The approach chosen in this thesis additionally aims at investigating the *distribution* of the response time. For this, the distribution of the waiting time of orbit-visiting jobs is derived. The chosen approach involves the Method of Phases (cf. [169, Part II]) and can be seen as a generalization of the approach applied in [88, Sec. 4.3], where the distribution of the waiting time of a finite-source multi-server retrial queue is obtained under the assumption that the servers are reliable. Furthermore, this thesis proposes a closed-form approximation of the waiting time distribution, which can be calculated based solely on model parameters and mean performance measures. In case of large state spaces, the approximation reduces the computational effort significantly. To the best knowledge of the author, this approach has not been taken before.

Chapter 3

Early Forest Fire Detection

> *[...] climate is changing and [...] these changes are in large part caused by human activities.*
>
> — America's Climate Choices [12, p. 1]

This thesis proposes and evaluates a WSN protocol that is tailored towards the long-term environmental monitoring of rare events. As an illustrative application example of such a WSN, the scenario of *early forest fire detection* is selected. Early forest fire detection is considered to be a suitable application scenario for the investigated class of WSNs since it requires long-term and wide-area surveillance, tiny nodes, and hence energy-efficient implementation, the fire events to be monitored occur (hopefully) rarely, and the operators expect a timely detection and reporting of those events. Forest fire detection also is of ongoing interest and is one of several application examples of WSNs that is mentioned frequently in WSN-related literature. However, as discussed in Section 2.1, large-scale and durable WSN-based implementations are not yet available. Hence, this thesis also contributes to answering the question whether WSNs are indeed suitable for carrying out forest fire detection. Based on the selected application scenario, practical and realistic parameters—e.g., size of covered area, sensor range, transmission range, sensor size, energy storage capacity, mean number of hops—can be determined in later chapters. The protocol's evaluation focuses then on this parameter range.

The aim of this chapter is to provide basic background information on forest fires and an overview on state-of-the-art detection methods.

3.1 Motivational Background

Forests are considered to be among the most important natural biospheres and resources (cf. [134, 186]). They not only constitute an essential factor for regulating the global ecological balance (cf. [134]), but also represent the basis of existence of a large variety of floral and faunal species (cf. [69, 306]). In general*, forest fires harbor the risk of high ecologic, economic, and social damage. They do not only pose a direct and indirect† thread to the fauna and flora resid-

* In specific cases, the disturbance caused by forest fires may also feature positive aspects, for example, rejuvenation of the population, promoting species that have adopted to fires, and fostering biodiversity (cf. [49, 113, 123, 196, 319]).

† For example, a forest that is weakened by fire is also more susceptible to pests (cf. [123, 161]).

ing in the forest, but also endanger fire fighters and humans living in the forest's vicinity*, may destroy properties, wood resources, recreational areas, and harvests, and may lead to decreased water quality and tourism, increased erosion and floods, and significant monetary cost for fire detection, fire fighting, salvage, and site rehabilitation (cf. [49, 113, 123, 186, 196, 216]).

Currently, wildfires are "most prominent in savannas, Mediterranean woodlands, and boreal forests" [113, p. 21]. In 2010, the most affected European countries were Portugal, Spain, France, Italy, and Greece (see [264, p. 8]). In 2010 and 2011, Germany experienced *only*† 780 and 888 forest fires, resulting in 522 ha and 214 ha of forest burned and an estimated economic damage of 1.2 and 0.9 million Euro, according to [264, p. 38] and [265, p. 38–39], respectively.

However, the number and severity of forest fires as well as their regional distribution might change significantly in future due to global warming. The authors of [12] state that "climate change is occurring, is caused largely by human activities, and poses significant risks for—and in many cases is already affecting—a broad range of human and natural systems. [...] Most of the warming over the last several decades can be attributed to human activities that release carbon dioxide (CO2) and other heat-trapping greenhouse gases (GHGs) into the atmosphere" [12, p. 3]. Forest fires are assumed to be correlated to global warming in two ways. First, it is expected that the number and severity of forest fires increase with global warming (see, e.g., [229], [120], [113], [49], [254], [339]). Vice versa, experts suppose that the emissions caused by forest fires abet global warming (see, e.g., [49], [120], [113], [238], [269]). The self-reinforcing process resulting from this positive feedback loop indicates that countermeasures against (unplanned) forest and wildfires are of increasing interest in future.

The European Commission collaborates with national authorities to reduce the impact of forest fires by "launching yearly awareness-raising campaigns, using early fire detection tools, carrying out forest fire prevention measures [...], forecasting and assessing fire danger, and last but not least, by developing further fire research activities" [264, p. 7]. But, "basic forest fire prevention measures [...], including awareness-raising campaigns and training of those involved in the forest fire issue, are [...] not always as effective as they could be, in particular in Mediterranean high risk areas" [264, p. 7]. Thus, early fire detection methods remain an important tool to curtail forest fires. Starting with Section 3.3, this thesis focuses on these.

3.2 Causes and Classes of Forest Fires

To be able to decide on suitability of detection methods, some basic understanding of forest fire properties is needed. There are, in principle, three main factors that determine the occurrence and evolution of forest fires: weather/climate, fuels, and ignitions (see, e.g., [50, 234]).

For selected European countries and year 2011, Fig. 3.1 illustrates the percentage‡ of the main causes for forest fire ignitions. The data used for the figure is extracted from [265] and classified into unknown causes, natural causes (includes lightning), and human-caused fires initiated by arsonists or negligence§. Unfortunately, the causes are unknown for the majority

* Note that the decrease of air quality caused by severe fires may increase the mortality of weak humans—e.g., elderly, ill, unborn, and infants—in a relatively large area (cf. [123, 185]).

† In comparison to, e.g., Portugal (2010: 22 026 fires (133 090 ha); 2011: 25 221 fires (73 813 ha)) and Spain (2010: 11 475 fires (54 770 ha); 2011: 16 028 fires (84 490 ha)), according to [264, 265].

‡ With respect to the number (not size) of fires.

§ Includes accidental fires, e.g., caused by agricultural and forestry activities, hikers' fire places, cigarettes, transport systems, power lines, etc.

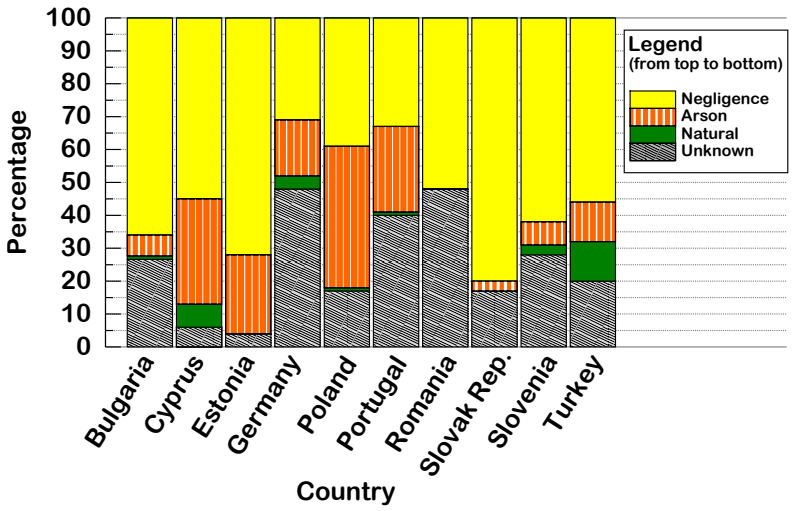


Figure 3.1: Proportionate fire causes for selected European countries in 2011.

of fires. According to [265, p. 77], the lack of this knowledge is still hampering the detailed comprehension of forest fires. Figure 3.1 shows that most fires for which the cause is known are caused by negligence. In Europe, only a very small fraction of fires is attributed to natural causes, like lightning.

There are basically three classes of forest fires (cf. [271, p. 3], [123, p. 400–403], [114], [161, Sec. 1.1], [295, Sec. 2.1]): *surface fires*, *crown fires*, and *ground fires*. Most human-caused fires start as *surface fires*, where dry fine fuels* are usually ignited first. According to [249, p. 29] and [72, p. 6], those fine fuels are also the primary reason for the fire's spread.

Crown fires are less common than surface fires, since they usually require a severe surface fire for ignition† and support (cf. [271, p. 4]). Without surface fire support, crown fires usually spread only given enough aridity, terrain slope, and wind (cf. [271, p. 4]). Crown fires, however, are usually far more difficult to control than surface fires (cf. [271, p. 1]).

Ground fires‡ are usually invisible fires that smolder slowly below the surface, making them hard to detect and control (cf. [295, Sec. 2.1], [256]). They may damage tree roots (cf. [295, Sec. 2.1]) and re-ignite surface fires even after long periods§. Like crown fires, they are usually caused by an active surface fire (cf. [271, p. 3]).

From the point of view of the behavior of forest fires, it is hence advisable to sense the environment close to the ground to achieve an early detection.

* Like needles, moss, thin twigs (cf. [72, p. 5]).

† Via so-called *ladder fuels*—like bushes, shrubs, dead branches, needle drape, and understory trees—that allow the fire to cross the gap between ground level fuels and the canopy (see, e.g., [271, p. 6], [106, p. 7,10]).

‡ For a clearer distinction from *surface fires*, *ground fires* are also referred to as *subsurface fires* or *peat fires* (cf. [114, 123, 161]).

§ Months or even years in the presence of coal (cf. [295, Sec. 2.1.2]).

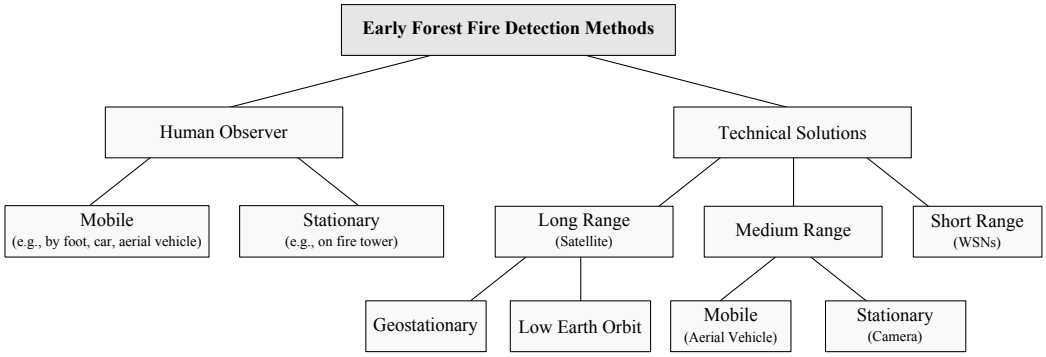


Figure 3.2: Classification of early forest fire detection methods.

3.3 Early Forest Fire Detection Methods

Tackling these requirements, several early fire detection methods have been investigated and partly established in the past. This thesis suggests a classification of the most prominent methods following Figure 3.2. Starting with the traditional approach involving human observers, these methods are discussed in Sections 3.3.2 and 3.3.3.

3.3.1 Basic Requirements

Usually, extinguishing forest fires would be easiest in their earliest stage when they are still small. Hence, the main goal of early fire detection methods is detecting, by investing reasonable effort, a starting fire and its location as early as possible.

To achieve this goal and to make the detection method attractive to land owners, the following properties are primarily significant:

- *Spatial Resolution:* This thesis treats two aspects of spatial resolution. The *detection resolution* refers to the minimum fire size that is required (in the mean) for the observer* being able to detect the fire event. On the other hand, the term *reporting resolution* refers to the accuracy of event location information provided to the operator.

- *Delay:* The time between the start of the fire and the notification of the operators (e.g., firefighters) is referred to as *event-to-notification* delay (T_{e2n}) in the following. This delay should lie in the order of a few minutes† and can be split into the *event-to-detection* delay (T_{e2d}; time between start of fire and detection of the fire by the first observer) and the *detection-to-notification* delay (T_{d2n}; time needed to communicate the detected event to the operator). While the latter delay mainly depends on the communication infrastructure and protocols used, the event-to-detection delay is mainly determined by the sensors' location, spatial resolution, and sampling frequency. More precisely, the

* In the following, two roles are differentiated: the *observer* and the *operator*. The *observer* is the entity (human or technical sensing device) that detects the event (or at least the relevant information that allows the *operator* to decide whether an event happened) and reports it to the *operator*. The *operator* (e.g., the responsible fire department or land owner) coordinates the monitoring and, in case of a fire event, initiates appropriate counter measures.

† According to [24], the National Oceanic and Atmospheric Administration of the United States Department of Commerce suggests a maximum of 6 min.

event-to-detection delay can be seen as being composed of an *event-to-detectable* delay (T_{e2a}; the event grows and/or moves such that it can be detected by some observer) and an *detectable-to-detection* delay (T_{a2d}; a observer samples the environment and actually detects the event).

- *Cost:* Trivial solutions* that meet the given spatial resolution and delay requirements are rendered infeasible by additional constraints regarding the effort and monetary cost for salaries, hardware, software, deployment, and long-term maintenance.
- *Environmental Compatibility:* Especially when hardware components are distributed into nature, they should be as non-intrusive as possible—both from a visual and biological/biochemical point of view.

There are several immediate implications and interrelationships of these requirements. For example, the spatial resolution highly depends on the observer type and the number of observers used. Likewise, for reducing the event-to-detection delay, each observer needs a high sensitivity or a large number of sensors is needed. High sensitivity increases the chance of false alarms. A large probability of false alarms again increases the cost, either for providing additional alarm verification mechanisms or for initiating inappropriate fire fighting measures. The more observers are required, the less expensive and environmentally intrusive a single observer should be. For technical solutions, these two requirements can partly be mapped to the observers' size. Also, to keep the maintenance cost low, each observer should have a long lifetime, especially when a large number of technical observers is involved. Since forest fire detection methods promise the service of detecting forest fires, which is a safety-critical service considering the threads caused by undetected fires (cf. Section 3.1), their reliability—and related properties, e.g., resilience and security— is also a very important aspect.

3.3.2 Human Observer

On many sites, fire detection still relies on human observers (see [291], [274, Sec. 3.1], [264], [265], [290]). For example, according to [264], Cyprus, Poland, and Turkey operated 27, 646, and 776 fire lookout stations, respectively, in 2010. Often, those are supported by further mobile forces patrolling jeopardized zones.

The most promising human sense for early fire detection is the visible-range light detection provided by the human eye†. It is able to identify fire and smoke from a distance of several kilometers. Moreover, the sense can rather easily be enhanced by, e.g., binoculars (cf. [290, Sec. 1]). Employing human observers, however, has several disadvantages, for example,

- *high operating costs*, because the area covered by a single person is limited (cf. [290, Sec. 1]); hence, a large number of persons is needed;
- *limited reliability*, because event rarity reduces attention of observers; moreover, observers might be temporary unavailable (cf. [312, Sec. 1], [291, Sec. 1], [92, Sec. 7.2]);

* Like dense deployment of human observers equipped with fire extinguishers and mobile phones throughout the forest for continuous monitoring and fast response.

† Clearly, several human senses are suitable for detecting fire in principle. For example, smoke can be smelled, temperature can be felt, and the crackle of burning wood can be heard. However, these senses have a rather short range and human observers need to abide some minimum distance from the fire for safety and also the overall number of human observers needs to be minimized for reducing costs.

- *limited quality of reported data* since data collected by humans may be subjective (cf. [274, Sec. 1]) and often is not sufficient for making fire fighting decisions (cf. [291, Sec. 1]),
- without additional equipment, there is a *lack of (digital) data*; therefore, there are *no automatic documentation* (e.g., no video recording) for post-fire analysis (cf. [290, Sec. 1]) and *no means for automatic detection*;
- *restriction to visible-range video* makes it often hard to differentiate smoke from other visible aerosols like mist, fog, clouds, or dust;
- *line of sight needed*; small fires and their smoke can easily be hidden by the trees' canopy; smoke may hide further fires;
- *basic infrastructure needed* to accommodate, supply, and substitute the human observer; often, an elevated lookout post (e.g., fire tower) is needed to increase the line of sight and communication infrastructure is provided for reporting incidents to the operator.

To tackle these disadvantages, human observers more and more get supported or replaced by technical sensor solutions, like optical cameras. In the following, the thesis focuses on these technical solutions.

3.3.3 Technical Solutions

In this thesis, technical fire detection methods are split into three main classes for further investigation: *long-range detection* (e.g., earth-orbiting satellites equipped with optical sensors), *medium-range detection* (e.g., fire towers equipped with video cameras or manned/unmanned aerial vehicles that patrol periled areas), and *short-range detection* (e.g., small, usually immobile wireless sensor nodes that may be equipped with several different sensors). These classes are illustrated in Fig. 3.3 and briefly discussed in the following.

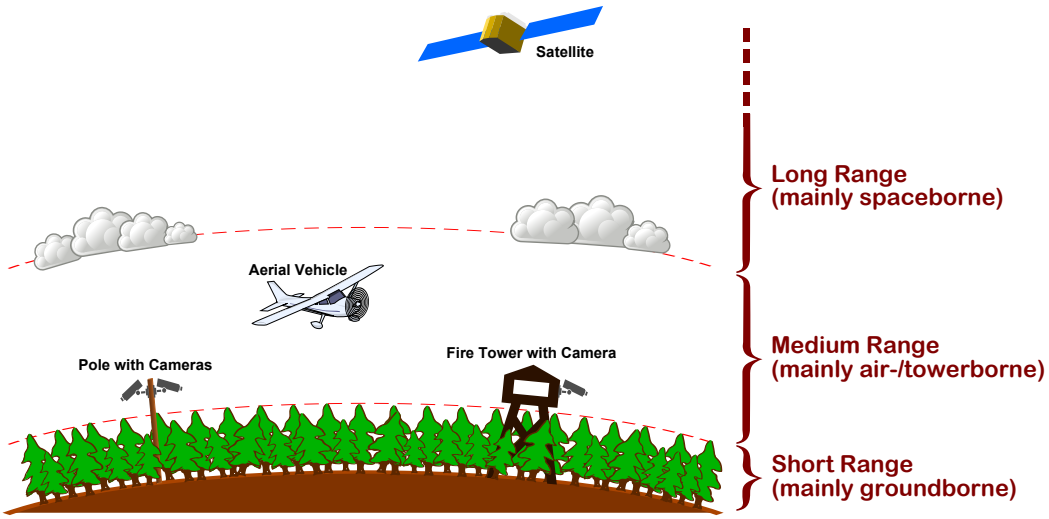


Figure 3.3: Illustration of detection range classification.

In literature, the term *remote sensing* (cf. [295, Sec. 2.2]) is frequently used to refer to sensing methods classified as *medium* and *long range* in this thesis.

3.3.3.1 Long-Range Detection Using Satellites

Earth-orbiting satellites (cf. [87, 180]) used for remote sensing are usually located in either *geostationary* (GEO, 35 800 km over ground) or *low earth orbit* (LEO, typically 300 to 1400 km over ground).

Due to their synchronicity with the Earth's surface, GEO satellites achieve—in principle—full temporal coverage of the monitored area. However, due to the large area and large amount of data they need to process, usually 15–30 min pass between the availability of two subsequent recordings of the same area (see, e.g., [174], [252, p. 374]). Due to their large distance from the Earth's surface, they additionally suffer from reduced spatial resolution (currently approx. 1–4 km^2 Earth surface per pixel in the best case; cf. [175], [174]). Note that already an area of about 1 ha (i.e., $10^4\,m^2$) needs to burn in order to identify a fire in a pixel of 1 km^2 size (see [175] or [245]).

Compared to GEO satellites, LEO satellites achieve significantly higher spatial resolutions of 1 m and less per pixel (see [87]). However, LEO satellites have a limited temporal resolution. For cycling the earth, they need about 100 min (see [87]). The number of visits per time interval to the same area depends on various factors, including the number of satellites carrying the relevant sensor instrument and the location of their orbits. Usually, the time between two visits takes several hours.

For example, the Aqua and Terra LEO satellites, each carrying a *moderate-resolution imaging spectroradiometer* (MODIS, [87]) instrument, often cited in the context of remote fire sensing (see, e.g., [120, 175, 245, 264, 269, 291]), together revisit the same area two to four times per day (cf. [175]), i.e., roughly every six hours in the best case, and have a maximum spatial resolution of 250 m per pixel (cf. [87, 264]). For MODIS, the authors of [205, Sec. 3] additionally describes an offset error of 2–3 km between the detected and actual location of the fire due to imprecise spacecraft location, software tolerances, and operational errors.

These spatial and temporal resolutions are too coarse for early fire detection. Satellite-based active-fire and burned-area monitoring is therefore mainly used for long-term and regional- to global-scale fire documentation (cf. [113]). The obtained data can then be used, e.g., for building and improving fire prediction models (cf. [111, 232]). Moreover, satellite-based remote sensing of active forest fires suffers from unfavorable weather conditions. For example, clouds and smoke might block the satellites' view, and sunlight—reflected by ground, clouds, and water bodies—may lead to false alarms (cf. [241, Sec. 1 & 5], [144], [245], [269, Sec. 1]). Also industrial sites may lead to false alarms (cf. [241, Sec. 5], [144], [245]).

For additional and recent research on satellite-based forest fire detection, the interested reader is also referred to [189, 231, 253, 326, 343] and the references therein.

3.3.3.2 Medium-Range Detection

This class can be subdivided into stationary and mobile medium-range detection methods. A prominent example of *stationary* medium-range detection is a set of video cameras installed on fire towers which often were used previously by human observers. A further innovative (though less established) stationary method is based on the use of acoustic signals. A typical example for *mobile* medium-range detection is the use of aerial vehicles. In the following, the three examples are discussed in more detail.

Camera-based detection. Cameras used for early forest fire detection typically operate in the visible (cf., [290], [312]) and/or infrared (IR, e.g., [311], [291], [19]) range of the electromagnetic spectrum. Visible-range cameras mainly aim at sensing the fire's flame (in particular at night) and smoke (in particular at daytime). IR cameras usually detect the fire's heat.

Generally, the cameras are installed such that they are provided with ample power and communication resources. The recorded video data can then be transmitted reliably and quite unrestrictedly to the operators. Often, the operators are able to remote control (pan, tilt, zoom) the camera manually to assess suspicious areas in more detail (cf. [290, 291]). In any case, the events need to be identified within the provided video data. Pattern recognition algorithms and similar techniques got established that allow automatic identification of fire or smoke within the video data and the research on improved algorithms is still ongoing (cf. [80, 98, 104, 291, 311, 312, 314]). Usually, the generated alert is subsequently verified by a human operator before local fire fighters are actually invoked (cf. [267]).

Several implementations of camera-based techniques for early detection of forest fires are commercially available and in operation, e.g., in form of FireWatch®,*, Firehawk Forest Watch™,†, EVS Forest Watch®,‡, iForestFire®,§, CICLOPE™,¶, AlarmEye®AE Forest‖, and ForestVu™,**. The interested reader is referred to [192, 193, 267], which provide a comparison of the FireWatch® and EVS Forest Watch® products.

Camera-based solutions share several disadvantages with the human observer. They also need line of sight and fail to detect small fires that are hidden below the forest's canopy. Line of sight is also decreased by hilly terrain and disadvantageous weather conditions (cf. [286]). Especially when they are restricted to visible-range video, they also frequently suffer from false alarms caused by, e.g., dust, mist, fog, clouds. While IR-based solutions are less prone to false alarms, they are significantly more expensive (cf. [291, 315]). The cameras also need basic infrastructure for elevated installation, power supply, and communication. It also should be noted that a single camera usually has a restricted angle of view. To observe the environment to the full 360° around the camera's location, the camera needs to be rotated, which takes six to twelve minutes according to [267]. To increase the temporal resolution, the number of cameras needs to be increased. This, on the other hand multiplies the costs.

Acoustic forest fire detection. Another stationary medium-range approach is taken in [257], where a radio-acoustic sounding system (RASS) is proposed. The system aims at the detection of crown and surface fires by measuring air temperature profiles with fine temporal and spatial resolution. For this, the authors exploit the fact that the propagation of sound waves is highly sensitive to air temperature and movement. The proposed system comprises sound sources (i.e., speakers) installed in the forest and a radar center that is used to measure the speed of the emitted sound just above the treetops. Analyzing the detected sound speeds allows to estimate a thermal map of the monitored area. For each speaker, a coverage of up to approximately

* http://www.fire-watch.de/ (last accessed: 18 March 2013)
† http://www.firehawk.co.za/ (last accessed: 18 March 2013)
‡ http://www.evsolutions.biz/Products/ForestWatch (last accessed: 18 March 2013)
§ http://iforestfire.fesb.hr/ (last accessed: 18 March 2013)
¶ http://www.inov.pt/pages_e/monitoring/tele_florestal_e.php (last accessed: 18 March 2013)
‖ http://www.innosys-ind.com/product_detail/innosys-ind2/index.php?Product_SN=114436&Company_SN=19641&Product_Site_Classify_SN=23299 (last accessed: 18 March 2013)
** http://www.firevu.co.uk/Forestvu.htm (last accessed: 18 March 2013)

$30\,km^2$ is foreseen. The main problem of this approach is that the audio signals emitted by the sound sources are in the audible range of humans and animals.

Airborne forest fire detection. An example for *mobile* medium-range detection is the use of aerial vehicles. Traditionally, these are manned and the crew members act as human observers that are usually particularly trained (cf. [228, p. 181 ff.]). During the last two decades, equipping manned aerial vehicles additionally with technical sensors —like cameras that operate in the visual or IR range, or gas sensors—gets more common (cf. [13, 56, 107, 155, 218, 227, 236, 344]). More recently, also the development of unmanned aerial vehicles (UAVs; cf. [331]) and their application for forest fire detection and monitoring makes significant progress (cf. [11, 57, 155, 159, 191, 200, 201, 202, 242, 268]). The authors of [145] give a survey of the large variety of airborne sensing techniques, which can also be used in other application scenarios.

Due to the increased personnel and/or fuel cost when keeping aerial vehicles in action, they are usually only deployed on-demand (for surveillance of an ongoing, already detected fire) and during high-risk periods. Hence, in contrast to continuous monitoring solutions, aerial vehicles fail to detect fires in assumedly non-critical periods (cf. [228, p. 188–189]).

Like long-range detection methods and immobile cameras, most airborne solutions are mainly based on optical detection, and hence, need line of sight. Therefore, they are mainly restricted to observe the environment from above the trees' canopy and usually fail to detect small fires.

In addition to optical detection methods, further medium-range sensors are applicable for aerial vehicles in principle. Airborne gas detection, for instance, comes with its own, additional research challenges. Some of these are discussed in [13, 159, 338]. For example, the detected concentration of gas and smoke significantly depends on the vehicle's speed (cf. [159]), but lowering the speed increases the re-visit time.

3.3.3.3 Short-Range Detection

The great advantage of short-range detection methods lies in their ability to monitor the situation close to the ground where most of the fires start. Due to their closeness to the event, they allow larger variety of sensor types, including non-optical sensors (cf. Table 2.1) that are less susceptible to line-of-sight obstructions. The combination of different sensor types can additional help to reduce the false-alarm probability (cf. [31]).

The majority of short-range methods is covered by solutions based on immobile WSNs. These are considered as related work and discussed in more detail in Section 2.1.

Only few solutions for early forest fire detection have been proposed that are based on WSNs with mobile nodes. For example, [256] proposes attaching wireless sensor nodes to animals. The sensor nodes send information on environmental conditions (like temperature) and their current location to a central computer. In addition to the temperature, the movement of panicking animals can then serve as an indicator of the presence of fire. As a side effect, the same WSN could be used for research on animal behavior. However, involving animals has obvious disadvantages. For example, they need to be available or caught and equipped with the sensors. The lifetime of the sensor nodes is not discussed in detail in [256]. While a maximum lifetime of three years is stated for a sensor with a volume of approximately $10\,cm^3$, it remains unclear how this value is obtained and whether it considers the energy needed for

localization (via GPS) and for the foreseen direct communication to satellites or the access points (deployed every 3-4 km). Further, the animals' mobility hampers the recharging of the nodes. In [173], a GPS-less on-demand localization approach for mobile, animal-attached nodes is proposed, which might be used to replace GPS in [256]. The authors of [33] discuss the case of controlled node mobility in WSN for monitoring forest fire risk. They propose a model of node mobility, which is based on a Bayesian network approach (cf. [137]) and aims at optimizing the spatial coverage to match the dynamics of the fire risk. According to the authors, the model can, in principle, also be used for the application of fire detection. Unfortunately, basic questions like the achieved lifetime in the face of node mobility, GPS-based node localization, and communicating the required metadata to the central evaluation entity of the optimization model remain unanswered in the paper.

In summary, the application of mobile sensor nodes bears a significant number of additional open research questions. Therefore, this thesis focuses on the application of immobile WSNs.

3.4 Chapter Summary

This chapter illustrates the ongoing need for early forest fire detection methods. Basic requirements for effective solutions and their implications on main properties are summarized. Available detection methods are classified and their shortcomings are briefly discussed. Since all methods show specific shortcomings, a single optimal solution for early forest-fire detection is not available. But, a combination of methods might lead to significant advantages. To assess the situation close to the ground, where most fires are initiated, short-range detection methods—like WSN-based solutions—can be used. The following chapters contribute to answering the question how such a WSN-based solution can be implemented.

Chapter 4

WSN Protocol Design

Success isn't permanent, and failure isn't fatal.

— Mike Ditka

As shown in Section 2.1, the application of WSNs to early forest fire detection still offers a plethora of open research issues. Also in this thesis, not all of the constraints following from the basic requirements (cf. Section 3.3.1)—like security and resilience requirements— can be tackled sufficiently. Instead, the thesis focuses on the problem area arising from the trade-off between node density, node size, network lifetime, and detection-to-notification delay. Section 2.1 indicates that this trade-off is not adequately taken into account by available solution proposals for WSN-based early forest fire detection. To meet these requirements, a novel communication protocol for WSNs is proposed in the current chapter. The proposed protocol is named *cross-layer message-merging protocol* (XLMMP) and explicitly tailored towards the energy-efficient and timely reporting of rare events in large monitored areas.

XLMMP is first described in brief in Section 4.1 to anticipate the main mechanisms. In Section 4.2, essential scenario-specific assumptions, estimations, and basic design decisions are provided that led to this protocol design. Based on these, Section 4.3 discusses further details of the protocol, including, e.g., the protocol's resilience in the face of message loss and node failure, details on the hop count-based routing and localization, and the message format. Section 4.4 concludes this chapter.

4.1 Brief Protocol Overview

This thesis chooses a customized cross-layer approach instead of a universal multi-layer protocol stack. On the one hand, this cross-layer design provides means for extensive optimization tailored towards the addressed application. On the other hand, however, many different aspects of the communication protocol require elaboration.

As illustrated in Fig. 4.1, the WSN consists of two types of nodes: a large number* of immobile wireless sensor nodes (in the following referred to as "sensor nodes" or just "nodes" as long as there is no ambiguity) that are randomly deployed† in the monitored area and a small number of reliable, immobile sink nodes (just called "sinks" in the following). The sinks

* In the order of $1 \cdot 10^5$, as discussed in more detail in Section 4.2.3.

† The stochastic distribution underlying the random node locations significantly depends on the deployment method chosen. More details are provided in Section 4.2.4.

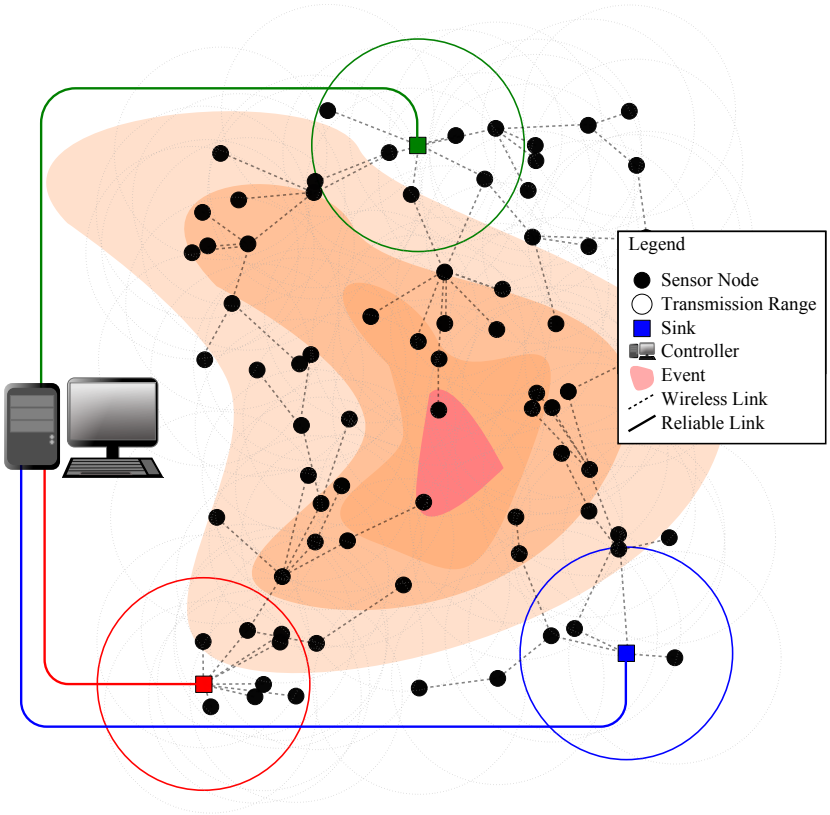


Figure 4.1: Illustration of monitored area, wireless sensor nodes, sinks, and their transmission ranges.

are deployed deterministically and serve as gateway between the WSN and a reliable network architecture (e.g., the Internet) for communication with the reliable controller (e.g., the fire fighter department).

In the following, this thesis assumes that the sinks are not energy constrained, for example, by being connected to power lines or equipped with sufficiently dimensioned rechargeable batteries and solar panels, and always available. For using the sinks as anchors in a hop count-based localization method, (at least*) three sinks are assumed to be deployed close to the edge of the monitored area and their geographical position is known. Due to the relatively small number of required sink nodes, the effort for fulfilling these assumptions seems acceptable. Additionally, the sinks have a short unique ID to be used within the WSN. From now on, the three sinks are denoted by *Sink 0*, *Sink 1*, and *Sink 2*.

Similar to the sinks, the controller has virtually (i.e., in comparison to the sensor nodes) unlimited resources, i.e., sufficient processing power, memory, and reliability, as well as negligible response time. Hence, the bottleneck of the architecture and the focus of this thesis is the communication between the sinks and the sensor nodes. Therefore, the sink and controller entities are usually subsumed under the term "sink" in the following. That is, if the thesis

* For conducting trilateration, which is discussed in more detail in Section 4.3.4.1, at least three sinks are needed, which are positioned in different locations. For preserving conciseness, the reminder of this thesis is based on the assumption that there are exactly three sinks. The extension of XLMMP to work with more than three sinks is rather straightforward.

Table 4.1: Overview of message types used by XLMMP.

	Sink Message	Event Message	Next Hop Found Message
Abbreviation	SMSG	EVM	NHF
Purpose	network management	event reporting	stopping receiver contention locally
Source	sinks	sensor node	sensor node
Destination	all sensor nodes	any sink	neighbor nodes
Main Content	commands (cf. Sec. 4.3.6.2)	event and location information	event and location information

states that *information is provided by a sink* or a *decision is made by a sink*, this might actually be done by the controller automatically using some technical device/algorithm or with human interaction.

Due to the sensor nodes' limited transmission range* and the large size of the monitored area†, all messages are sent in a multi-hop fashion.

The sensor nodes utilize one or several sensors to monitor their environment. Based on the sensor readings and threshold information provided by the sinks, the nodes decide whether a sensor reading corresponds to a report-worthy event, and consequently, whether an *event message* (EVM) needs to be sent to the sink. The EVMs' purpose is to communicate the detected events to any sink.

For message routing and for localization of sensor nodes and events, all nodes need to be aware of their distance to each of the three sinks measured in the number of communication hops. This can be achieved by hop count updates implemented by counters contained in *sink messages* (SMSGs) that are broadcast by the sinks and incremented by each next-hop sensor node. The SMSGs also serve as interest notifications and end-to-end acknowledgments (ACKs) as described in Section 4.3.6.2 in more detail. Hence, the SMSGs are mainly used for network management and control.

Next hop found messages (NHFs) are introduced to enable sensor nodes to inform their neighbors‡ that another neighbor already agreed to serve as next hop. Hence, the neighbors may discard their copy of the EVM. This process is described in more detail later in this section.

The three messages types are summarized in Table 4.1 and their format is discussed in more detail in Section 4.3.6.

Apart from their hop distances to the sinks and their current local situation (comprising sensor readings, stored EVMs, and battery level), the sensor nodes are indistinguishable, in

* In the order of 100 m, as discussed in Section 4.2.6.

† In the order of 70 km^2; see Section 4.2.2.

‡ A node's *neighborhood* comprises all its *neighbors*, i.e., all nodes that are located within its transmission range. In this thesis, symmetrical connectivity is assumed. Hence, neighborships are considered to be mutual.

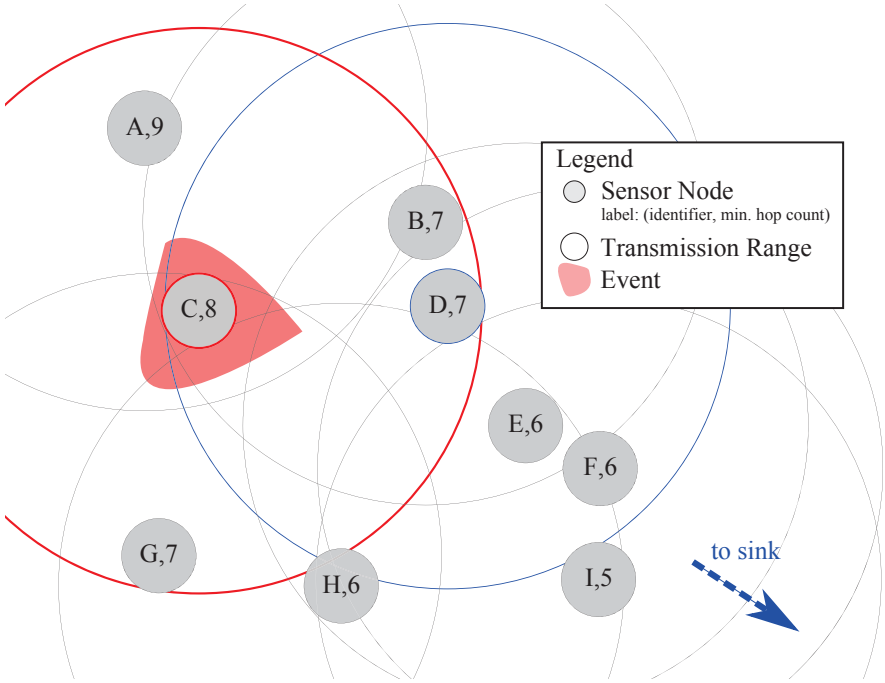


Figure 4.2: Detail of Fig. 4.1: Sensor nodes and their transmission ranges.

particular due to the lack of unique IDs*.

It is assumed, for now, that all sensor nodes are aware of their hop distance to each sink and the nodes calculated their position based on this information. Moreover, each node is supplied with a list of reporting thresholds that was provided before deployment and possibly updated by SMSGs.

Figure 4.2 shows a detail of Fig. 4.1. It illustrates an example node placement and aids as reference during the description of XLMMP's behavior provided in the following. The shown sensor nodes are labeled with an identifier and the number of hops to the next sink.

All sensing nodes compare their sensor readings (e.g., a temperature) to their list of thresholds in order to decide whether the sensor readings correspond to report-worthy events (e.g., temperature exceeds 60 °C). If an event needs to be reported (as in the case of node C), the sensor node becomes the source node of a new EVM. The EVM contains the source node's hop-distance (8) to the closest sink, an identification of the sensed event and its location†, and redundant bits (e.g., parity bit or checksum) for transmission error detection. Since EVMs are always directed to any sink, no destination address needs to be added explicitly.

Then, the source node (C) activates its transceiver and tries to get medium access using a contention-based method that can be seen as a variant of *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA, see [65], [3, p. 80] without RTS/CTS‡ as discussed in

* The decision to refrain from providing a unique ID to each sensor node is motivated in Section 4.2.15. Throughout this thesis, whenever identifiers are provided for sensor nodes in figures, they are only used for referencing the nodes during the figures' discussion. They are unknown to the WSN.

† The location of an event is determined by the location of the sensor nodes that sense the event. The localization of the sensor nodes is described in more detail in Sections 4.2.12 and 4.3.4.1. The compact addressing of location ranges is discussed in Section 4.3.5.

‡ Request To Send / Clear To Send.

Section 4.2.11). For this, the source node (C) listens to the wireless channel and checks whether it is free or busy. If the wireless channel is free, the source node (C) immediately transmits the EVM. If the channel is busy, the source node (C) continues listening to the channel.* When the wireless channel gets free, the source node (C) waits for an additional random delay and re-checks the wireless channel. The additional delay avoids synchronized transmission and resulting collisions with further previously waiting nodes in the neighborhood. The process is repeated until the wireless channel is free after the random delay and the message can be sent. All transmissions mentioned in the following adhere to this basic random-access method.

After sending the EVM, the source node (C) listens to the wireless channel and waits for an ACK by some neighboring node that is closer to any sink. Due to the broadcast character of the wireless air interface, all neighbors of the source node (C) are able, in principle, to receive the EVM. However, nodes are able to put their transceivers to sleep mode for saving energy. Therefore, only a small subset (if any) of neighbors are actually able to pick the EVM up. In the scenario of Fig. 4.2, neighbors A, B, D, and G are able to receive the EVM correctly.

All message-receiving neighbors with the same or higher hop-count than the source node (like node A) ignore the EVM. All other message-receiving neighbors (nodes B, D, and G) store the message locally and negotiate which of them should be the next hop and sign responsible for relaying the message further toward the sink. The negotiation is done via *receiver contention*, i.e., upon receiving the message correctly, each neighbor calculates an individual back-off delay based on the message's content and the node's local situation and waits for its expiration in listening mode.

As soon as its back-off delay expires, the receiver with the shortest back-off delay (say, node D) retransmits the EVM first, with its own, decreased hop count (7). From now on, this neighbor (node D) is responsible for forwarding the EVM further toward the sink. Note that a sink will always respond without delay. All other neighbors that overhear the updated message (from node D) can then stop their receiver contention process for this message by canceling the back-off timer and discarding the EVM (like node B).

The source node (C) likely also receives the updated message (from D) and interprets it as ACK. On reception, it sends out an NHF such that all of its neighbors (like node G which is not in the transmission range of node D) can also stop their receiver contention process for this EVM. Moreover, all non-responsible nodes, including the source node (C), can then discard the original EVM.

Assuming that there are no transmission errors and negligible delays caused by medium access and processing, Fig. 4.3 illustrates the sequence of messages transmitted in the scenario of Fig. 4.2 between nodes G, C, D, and E. In the figure, the boxes referring to EVMs and NHFs are complemented by the sending node's hop count to the next sink.

In the face of the described behavior of XLMMP, each node maintains two pools† of EVMs. In the *incoming-EVM pool* (inPool), it stores the yet unacknowledged EVMs for which the individual back-off timers are running and receiver contention is still carried out. In the *outgoing-EVM pool* (outPool), it stores the EVMs for which it signed responsible for relaying. Both pools have finite capacity. The current utilization of the outPool is considered in the calculation of the back-off delay. When the inPool is full, the node does not accept further incoming EVMs unless these can be merged with already stored ones.

* It continues listening to probably overhear EVMs that can be merged with its own EVM. The concept of message merging is described in more detail later in this section.

† The term *queue* is avoided here to prevent *first-come-first-served* connotations.

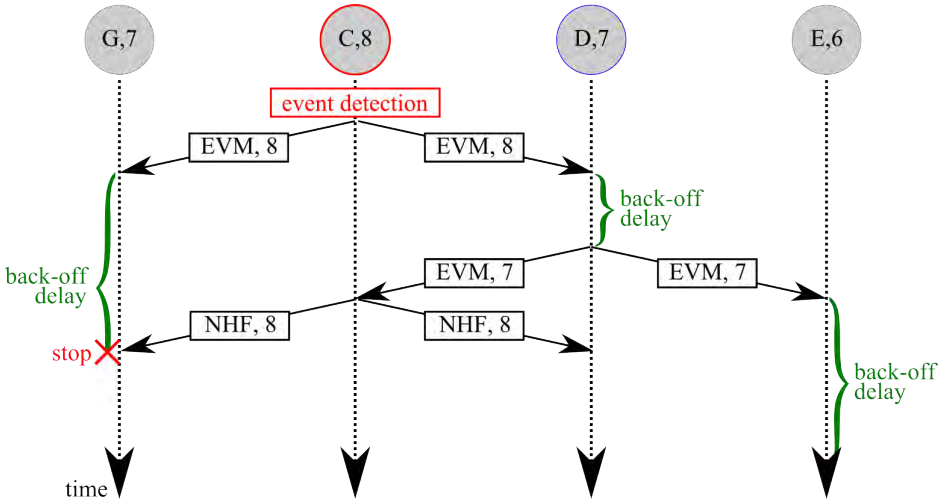


Figure 4.3: Message sequence chart

Figure 4.4 shows an activity diagram of XLMMP's behavior with respect to the handling of received EVMs and NHFs and the generation of these messages at a tagged node. The diagram already comprises protocol characteristics that are presented in more detail in the upcoming sections. It does not include the handling of SMSGs which are treated in Section 4.3.6.2.

The unique and novel strength of XLMMP is that the receivers can take into account the event data included in the incoming EVM already while doing receiver contention. In particular, it allows a receiver to prefer responsibility for EVMs that can be merged with EVMs that are stored in its outPool. The property could also be exploited to provide faster forwarding to important events by calculating the back-off delay based on event priorities.

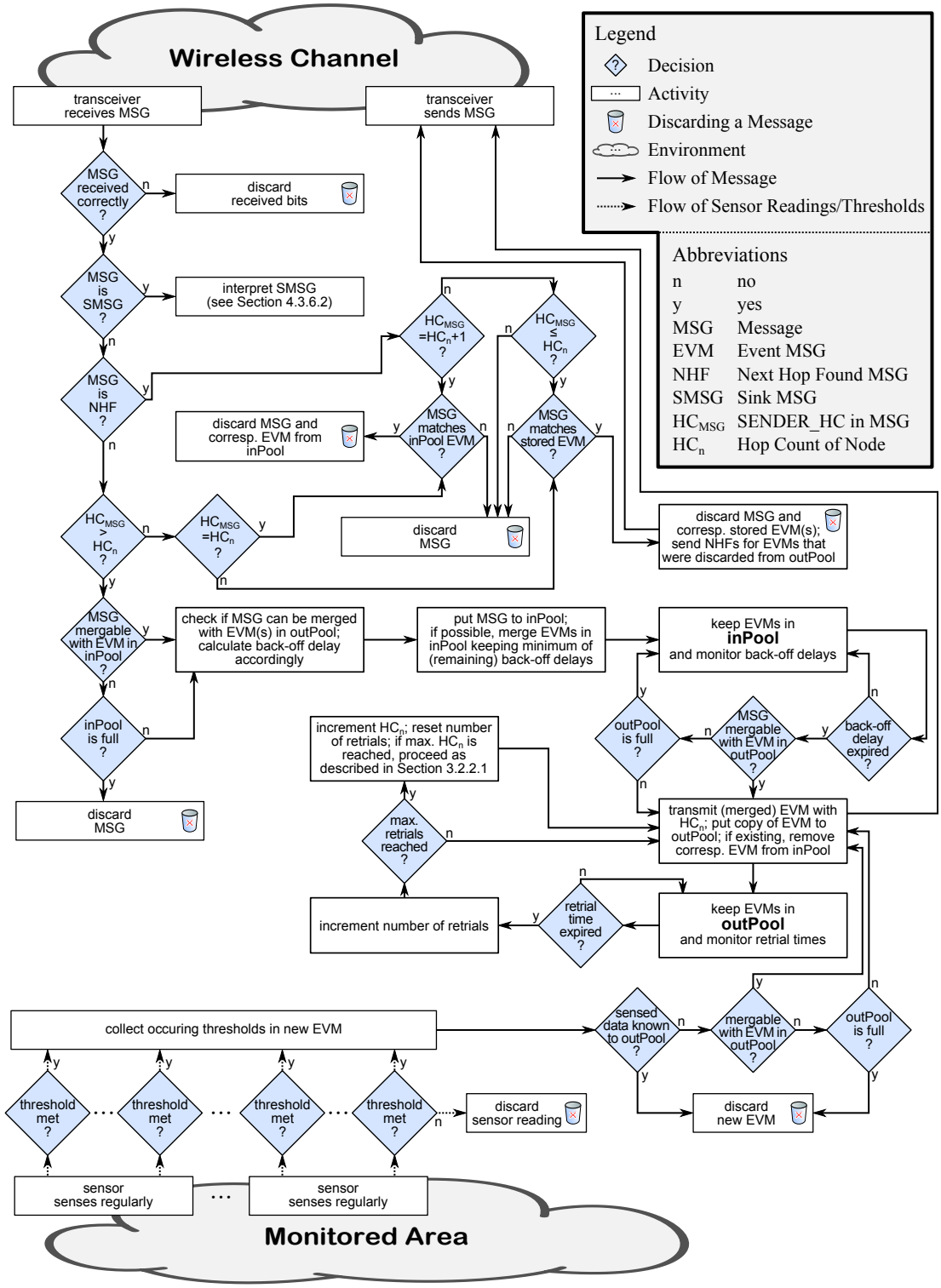


Figure 4.4: Activity diagram for the generation and handling of EVMs and NHFs.

4.2 Basic Scenario-Specific Assumptions, Estimations, and Design Decisions

The design of XLMMP is based on a set of essential assumptions, estimations, and design decisions. These are motivated in this section on the basis of the forest fire application scenario presented in Chapter 3. In particular, rough but reasonable estimates for main system and environmental parameters are derived. In Chapter 5, the proposed protocol's performance is also investigated for variations of these parameters by quantitative evaluation.

4.2.1 Sensing Range

According to the discussions presented in Section 3.2, most forest fires start at the ground. Hence, the sensor nodes should be placed at ground level to achieve early detection of fires before these reach the treetops. This also significantly simplifies deployment. However, it is then very likely that the field of view of each sensor is severely obstructed by obstacles like trees, undergrowth, foliage, or the terrain itself. Hence, the sensing range of a single sensor node is quite limited. The sensing range is highly dependent on the actual sensor type used and on the environmental conditions. A detailed analysis of such physical/chemical relationships lies outside the focus of this thesis. Instead, it is assumed in the following that a report-worthy fire can be detected early enough if it starts less than $d_S = 20\,\mathrm{m}$ away from the next sensor node.

With d_S given, the size of the area $A_S = d_s^2 \mathring{\pi} \approx 1257\,\mathrm{m}^2$ which is "sensed" by a sensor node can be calculated.*

4.2.2 Area of Interest

Sizes of forests vary significantly and relying on the mean forest size per owner (approximately 2.3 ha=0.023 km^2 in Bavaria†) seems infeasible. Instead, this thesis chooses a compact, medium-sized part of the "Neuburg Forest", located at the south-west of the city of Passau, as an illustrative reference. The forest is depicted in Fig. 4.5 and covers an area of approximately $A_\mathrm{F} = 70\,\mathrm{km}^2$. From the figure, the maximum linear distance between two sensor nodes (12.5 km) and the maximum optimal length of the communication path between two sensor nodes (13.0 km) can be derived. Based on these measures, the monitored area's maximum diameter can be estimated‡ as $d_\mathrm{p}^{(\max)} \approx 13\,\mathrm{km}$.

4.2.3 Number of Nodes and Node Density

Based on the area covered per node and area of interest, the overall number of nodes needed can be estimated. Assuming optimal deployment of nodes, i.e., without overlapping sensor ranges, at least $N_S^{(opt)} = \frac{A_F}{A_S} \approx \frac{70\,\mathrm{km}^2}{1257\,\mathrm{m}^2} \approx 56000$ nodes are needed to achieve full coverage. This results in a node density of $\overline{N}_d^{(opt)} = \frac{N_S^{(opt)}}{A_F} \approx 800\,\mathrm{km}^{-2}$.

---

* Throughout this thesis, $\mathring{\pi}$ refers to the mathematical constant *pi* (3.14159...), since the symbol π is used for steady-state probabilities in later sections.

† According to statistics of the "Bayerischer Waldbesitzerverband e.V." available at `http://www.bayer-waldbesitzerverband.de/waldstat.html` (last accessed: 10 May 2012).

‡ For comparison, note that in a perfect square of size 70 km^2 the diagonal's length is approximately 11.7 km and the minimum diameter for this size is 9.4 km and achieved by a disc-shaped area.

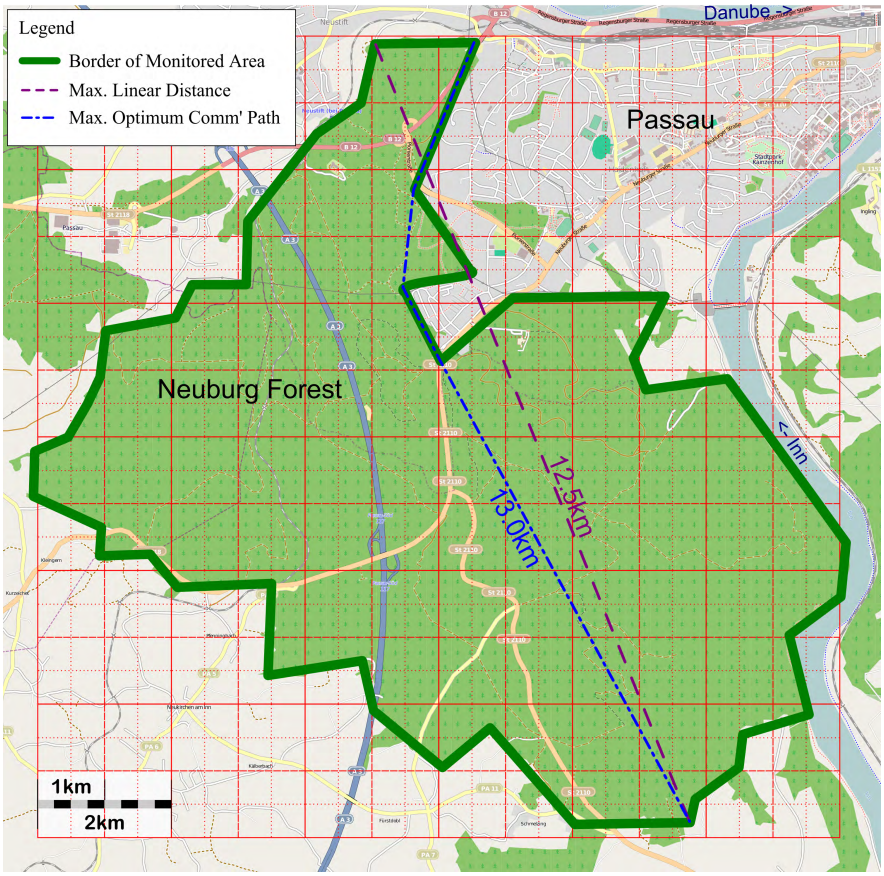


Figure 4.5: Example forest area for illustration (underlying map © OpenStreetMap contributors, CC-BY-SA[§]).

But, since the two-dimensional monitored area cannot be fully covered by circular sensing ranges without overlap, due to non-deterministic deployment (cf. Section 4.2.4), and since redundant deployment is intended to increase reliability, the number of nodes N_S needs to be chosen significantly higher. A more detailed analysis of this problem is out of the focus of this thesis. The interested reader is referred to [212, 322, 363] and [143, Sec. 13.2.2].

In the following, an overall network size in the order of $N_S = 100\,000$ sensor nodes is chosen to cover the area of size $A_F = 70\,\text{km}^2$, resulting in a mean node density of $\overline{N}_d = \frac{N_S}{A_F} \approx 1400\,\text{km}^{-2}$.

4.2.4 Random Deployment

Due to the large number of nodes needed, manual deterministic deployment is infeasible. Instead, nodes are typically scattered randomly over the area, e.g., by dropping the sensors from an airplane (as mentioned in, e.g., [8, 74, 86]) or using autonomous UAVs as proposed in [66].

The deployment method has significant influence on the sensor node distribution. Note, however, that XLMMP allows to regularly determine the current coverage (cf. Section 4.3.4.3).

[§] `http://www.openstreetmap.org/`, `http://creativecommons.org/licenses/by-sa/2.0/` (last accessed: 14 May 2012)

Hence, redeployment actions can be taken to bring out additional nodes in uncovered areas.

In the following, it is assumed that the nodes are uniformly distributed in the forest with node density $\overline{N}_d$.

4.2.5 Node Size

Due to the high node density, each node needs to be as small as possible. This makes the sensor network less invasive (cf. [8]), often less expensive (cf. [140]), and simplifies deployment (cf. [140]).

The authors of [140] foresee *Smart Dust* consisting of sensor nodes having a volume of a mere cubic millimeter each. In 2002, the concept of a $16\,\mathrm{mm}^3$ Smart Dust node implementation has been demonstrated in [330]. The *Centre for Speckled Computing*[*] aims at developing SpeckNet, a wireless ad-hoc sensor network with thousands of tiny nodes (called *specks*; with a volume of approximately one cubic millimeter each) that can be sprayed or scattered on surfaces or persons (cf. [125]). Further research (e.g., [2]) even predicts *nanosensors* with sizes down to $10\text{–}100\,\mu\mathrm{m}^3$. It stands to reason how fast these visions will become reality (cf. [8]), but the reduction of size will surely broaden the application area of WSNs.

Since lifetime plays an important role in WSNs, the major part of a node's maximum volume likely is used by the battery (cf. [2, 140], see also Section 4.2.8).

This thesis strives for a node size in the order of approximately $2\,\mathrm{cm}^3$ (plus antenna[†]). In case of a cube-shaped node, this corresponds to an edge length of approximately 1.26 cm.

4.2.6 Transmission Range

Current WSN implementations achieve maximum transmission ranges of up to several hundred meters (see, e.g., [3, p. 39],[340, p. 29]). This range heavily depends on the transceiver and antenna design as well as environmental conditions (see, e.g., [138] for near-ground measurements in a forest environment). A thorough discussion of these topics lies beyond the scope of this thesis and refraining from going into physical details of wireless channels (as discussed in [143, Ch. 4], [3, Ch. 4], [340, Ch. 3]), the transmission range is roughly estimated to be in the order of $d_\mathrm{T} = 100\,\mathrm{m}$.

In the investigated forest scenario, it is rather unlikely to achieve higher transmission ranges due to the high path loss coefficients expected (approximately in the order of 4 to 6, cf. [143, p. 93, 108], [3, p. 53]) since the nodes' antennas are rather small, located close to the ground, are likely to be covered by foliage and obstructed by undergrowth, etc.

4.2.7 Number of Neighbors and Connectivity

The number of neighbors of a sensor node is the number of sensors that are located within its transmission range and hence estimated[‡] by $\overline{N}_\mathrm{N} \approx A_\mathrm{T}\overline{N}_d \approx 44$, where $A_\mathrm{T} = \pi d_\mathrm{T}^2 \approx 31\,400\,\mathrm{m}^2$ is the size of the area covered by the transmission range $d_\mathrm{T} = 100\,\mathrm{m}$ of a single node and $\overline{N}_d \approx 1400\,\mathrm{km}^{-2}$ is the node density.

[*] `http://www.specknet.org` (last accessed: 10 Jan. 2012).

[†] The length of an antenna is depending on the radio frequency. For 2.4 GHz radio frequency, the length of an appropriate antenna would be approximately 3 cm (cf. [307, p. 60]).

[‡] Ignoring border effects caused by the bounded monitored area.

For achieving connectivity of all nodes to the sink and vice-versa, there must not be an isolated node, which is a node that has no neighbors. According to [40, Eq. (7)], the probability p_i that a tagged node is isolated given that the nodes are uniformly distributed can be calculated approximately* via

$$p_\mathrm{i} \approx e^{-\overline{N}_d \pi d_\mathrm{T}^2} \approx 7.92 \cdot 10^{-20} .$$

Given $N_S = 100\,000$ nodes, the probability that at least one of them is isolated can be approximated† by

$$1 - (1 - p_\mathrm{i})^{N_S} \approx 7.92 \cdot 10^{-15} ,$$

that is, the probability that the network comprises at least one isolated node is approximately zero.

However, the non-existence of isolated nodes is only a necessary but insufficient condition for the network's connectivity, as also realized in [40, Sec. 5]. For example, consider two nodes that are in mutual transmission range but disconnected from the remainder of the network. In this network, no node is isolated, but the network is still disconnected.

Note, however, that literature on connectivity in wireless networks of uniformly distributed nodes (e.g., [154, 217, 302, 303]) shows that given an average number of at least approximately five to ten neighbors, network connectivity is given with high probability. Since in the investigated scenario the expected number of neighbors is $\overline{N}_\mathrm{N} \approx 44 > 10$, connectivity can be assumed‡.

4.2.8 Lifetime

Since the wireless sensor nodes are scattered in a large area with high density, replacing or recharging them is a tedious (if not infeasible) task. Therefore their lifetime should be relatively high, i.e., in the order of years. This thesis investigates the feasibility of WSN designs with a lifetime of at least three years and preferably beyond.

The main factor that limits the nodes' lifetime is the restricted amount of energy available to the nodes. There are three main approaches for supplying sensor nodes with electrical power. First, there are energy harvesting techniques (see [250, Sec. 3.5], [294]) that allow the node to scavenge the environment. Second, small-scale power sources can be applied (see [250, Sec. 3.3]), which include electrochemical batteries, ultracapacitors, micro-scale fuel cells and heat engines, radioactive power sources, etc. Third, nodes could be supplied by electromagnetic, wired, acoustic, or optical power distribution (cf. [250, Sec. 3.4]).

While some of these techniques may further increase the applicability of small-scale wireless sensor nodes in various environments, this thesis focuses on primary (i.e., non-rechargeable) batteries as the only power source, since they are more established and according to [289, Sec. 9.2] they are cheap, "require little to no environmental calibration, do not influence the placement of sensor nodes [...], do not place exotic packaging requirements on sensor nodes, do not depend on environmental variables, have predictable behavior, and require less energy conversion overhead".

* Again, ignoring border effects.

† Like [40], by assuming independence of two different nodes' p_i.

‡ This assumption is implicitly validated by the MASON simulation results presented in Sections 4.3.2, 4.3.4.1, 5.7.1, and 5.8. In all simulation runs, each sensor node achieved a finite hop count to each sink.

In particular, note that typical energy harvesting techniques are infeasible in the investigated application scenario. For example, photovoltaics and wind energy probably fails since the nodes are small and likely covered by foliage. There are no significant temperature differences in the forest that could be exploited.* Neither are the nodes attached to frequently moving entities. Hence, obtaining sufficient energy from node acceleration or vibration is also unlikely.

The volumetric energy density of primary battery technologies currently lie in the order of approximately $4000\,\mathrm{J\,cm^{-3}}$ for zinc–air batteries and $3000\,\mathrm{J\,cm^{-3}}$ for lithium batteries (see [250, Tab. 3.1]). Since the lifetime of zinc–air batteries is limited to approximately half a year (see [250, Fig. 3.1]), lithium batteries should be preferred. Assuming that about 80 % of the aspired node size of $2\,\mathrm{cm^3}$ is used by the battery, the battery's volume is approximately $1.6\,\mathrm{cm^3}$. Consequently, assuming a volumetric energy density of $3000\,\mathrm{J\,cm^{-3}}$, approximately $E^{(\max)} \approx 4800\,\mathrm{J}$ are available to the sensor node during its lifetime. Note that more recent and future developments in the area of battery technology can be exploited to further increase the WSNs lifetime, reduce the price per node, reduce the nodes' size, or increase the nodes' biodegradability.

It can be shown that usually the power consumption of a sensor node is dominated by the transceiver (see, e.g., [3, Sec. 3.7]). Without going into more detail, this thesis assumes that approximately half of the available energy (i.e., $E_{\mathrm{trans}} \approx 2400\,\mathrm{J}$) can be used by the (partly sleeping) transceiver before the node's battery is depleted. That is, in the same time, approximately 2400 J are available to the processor, memory, and sensors.

Typically, a transceiver has four main modes (cf. [3, Fig. 3.2]): transmission (TX), reception (RX), idle listening, and sleeping. The former three modes usually consume approximately the same power P_{active} and are referred to as *active modes* in the following. In general, the sleeping mode requires significantly less power (P_{sleep}) than an active mode. For several typical low-power state-of-the-art transceivers, the power consumption of each mode is provided in Table 4.2.† The table reveals that transceivers consume power roughly in the order of $P_{\mathrm{active}} \approx 50\,\mathrm{mW}$ in active mode and $P_{\mathrm{sleep}} \approx 5\,\mu\mathrm{W}$ in sleep mode. The upcoming discussions are based on these values.

In the following, p_{sleep} denotes the probability that a transceiver is in sleep mode. The mean power consumption of a transceiver‡ is then given by

$$\begin{aligned}\overline{P_{\mathrm{trans}}} &\approx p_{\mathrm{sleep}} P_{\mathrm{sleep}} + (1 - p_{\mathrm{sleep}}) P_{\mathrm{active}} \\ &= p_{\mathrm{sleep}} \left(P_{\mathrm{sleep}} - P_{\mathrm{active}}\right) + P_{\mathrm{active}} .\end{aligned}$$

* While temperature differences exist in the case of a fire event, it is unlikely that all nodes needed to communicate the event are exposed to these temperatures. Moreover, for network maintenance, nodes should also be able to communicate without the presence of an actual event.

† The utilized references are provided in the table. If a transceiver offers several transmission powers, the maximum power is considered. Likewise, the maximum reception and idle powers are considered. All of the listed transceivers accept a supply voltage of 3.0 V, which is used to calculate the corresponding power based on the current that is usually provided in the documentation. The power consumption in idle listening mode is usually not provided explicitly. However, following [3, Fig. 3.2], it seems fair to assume that it is identical to the RX mode's power consumption.

‡ A closer inspection of the influence of the power consumed due to the switching between modes is postponed to future work.

Table 4.2: Power consumption of typical low-power transceivers

Transceiver	Power			Source
	TX (mW)	RX/Idle (mW)	Sleep (mW)	
TR1000	36	13.5	$15 \cdot 10^{-3}$	[3, Tab. 4.1]
TDA5250	35.7	28.5	$27 \cdot 10^{-3}$	[3, Tab. 4.1]
CC1000	80.1	28.8	$3 \cdot 10^{-3}$	[3, Tab. 4.1]
CC2420	52.2	56.4	$0.06 \cdot 10^{-3}$	[3, Tab. 4.1]
MPR300CB	36	5.4	$3 \cdot 10^{-3}$	[292, Tab. II]
XE1205	99	56	$0.6 \cdot 10^{-3}$	[292, Tab. II]
CC1101	45	44.1	$0.6 \cdot 10^{-3}$	[292, Tab. II]
CC1020	59.7	59.7	$0.6 \cdot 10^{-3}$	[292, Tab. II]
nRF2401	30	66	$1.2 \cdot 10^{-3}$	[292, Tab. II]
MC13191	102	111	$3 \cdot 10^{-3}$	[292, Tab. II]
mean	57.57	46.94	$5.41 \cdot 10^{-3}$	

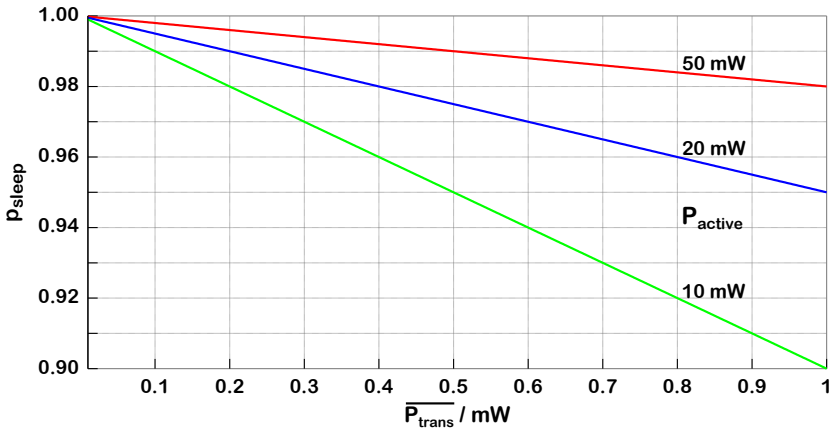


Figure 4.6: Influence of $\overline{P_{\text{trans}}}$ (x-axis) and P_{active} (curves) on p_{sleep} (y-axis), for $P_{\text{sleep}} = 5\,\mu\text{W}$.

For $P_{\text{sleep}} = 5\,\mu\text{W}$, this relationship between $\overline{P_{\text{trans}}}$, P_{active}, and P_{sleep} is shown graphically in Fig. 4.6. A node's expected lifetime $\overline{T_{\text{life}}}$ can then be given by

$$\overline{T_{\text{life}}} \approx \frac{E_{\text{trans}}}{\overline{P_{\text{trans}}}} \approx \frac{E_{\text{trans}}}{p_{\text{sleep}} \left(P_{\text{sleep}} - P_{\text{active}}\right) + P_{\text{active}}} . \tag{4.1}$$

Also note that

$$\overline{P_{\text{trans}}} \approx \frac{E_{\text{trans}}}{\overline{T_{\text{life}}}} ,$$

and

$$p_{\text{sleep}} \approx \frac{P_{\text{active}} - \overline{P_{\text{trans}}}}{P_{\text{active}} - P_{\text{sleep}}} \approx \frac{P_{\text{active}} - \frac{E_{\text{trans}}}{\overline{T_{\text{life}}}}}{P_{\text{active}} - P_{\text{sleep}}} .$$

To enable a node with $E_{\text{trans}} \approx 2400\,\text{J}$ to achieve a lifetime of at least three years (i.e., $\overline{T_{\text{life}}} \approx 95 \cdot 10^6\,\text{s}$), its transceiver (with $P_{\text{active}} \approx 50\,\text{mW}$) needs to sleep at least with a probability of $p_{\text{sleep}} \approx \frac{50\,\text{mW} - \frac{2400\,\text{J}}{95 \cdot 10^6\,\text{s}}}{50\,\text{mW} - 5\,\mu\text{W}} \approx 0.9996$, i.e., more than approximately 99.96% of the time. Hence, the node may be active up to approximately 3.5 h per year.

Remember that E_{trans} is determined by the technology of the power source (e.g., type and size of a primary battery). The value of P_{active} is determined by the chosen transceiver technology. Hence, future development in both fields might decrease the necessary p_{sleep}, increase the achievable lifetime $\overline{T_{\text{life}}}$, or may allow to decrease the nodes' size.

The influence of p_{sleep} on the system performance (in particular on the detection-to-notification delay) is one of the main foci in Chapter 5.

4.2.9 Detection-to-Notification Delay

Following the discussions in Sections 3.3 and 4.2.1, this thesis aims at a maximum detection-to-notification delay in the order of 5 min. Note that the National Oceanic and Atmospheric Administration of the United States Department of Commerce suggests a maximum of 6 min (cf. [24]) and most state-of-the-art long- and medium-range detection methods discussed in Sections 3.3.3.1 and 3.3.3.2 also fail to perform significantly better. Nonetheless, it should be noted that this value is only used as a rough point of reference. The combination of model and evaluation method proposed in this thesis can be applied to estimate which system parameters can be modified to achieve significantly lower detection-to-notification delays.

Additionally, this thesis deems fine-granular information on the absolute time of detection to be dispensable. That is, when an operator gets the notification of a detected event, it is sufficient to know that the notification was generated, with high probability, within the maximum detection-to-notification delay (i.e., within the last 5 min). From the application perspective, providing the detection time in higher resolution does not give significant additional value. In particular, note that the event-to-detection delay already varies significantly, since it depends, e.g., on the distance between the event's starting point and the closest sensor and on the fire's speed of spread. Therefore, the proposed design refrains from including the absolute event detection time in the event message. This approach not only reduces the messages' size, but also saves the sensors from keeping their timers synchronized to some global reference. Avoiding this synchronization effort results in significant advantages that are discussed in Section 4.2.16 in more detail.

4.2.10 Message Transmission and Propagation Delay Estimation

As discussed in more detail in Section 4.2.14, a data rate of 100 kbps is assumed. In Section 4.3.6, the maximum lengths of the communication messages is derived as 86 bit. Hence, the expected maximum transmission delay (without collisions and retransmissions) is $T_{\text{tran}} \approx 860\,\mu\text{s}$.

According to [67, Eq. (3.30)], the velocity of radio waves in forested areas can be given by $v_{\text{forest}} = v_{\text{air}} \varepsilon_r^{-0.5} \approx 2.9 \cdot 10^8\,\text{m}\,\text{s}^{-1}$, where $v_{\text{air}} \approx 3.0 \cdot 10^8\,\text{m}\,\text{s}^{-1}$ is the speed of light in air (see [160, p. 343]) and $\varepsilon_r \approx 1.06$ is the permittivity of a medium-dense forest (see [67, Tab. 2]). Given the transmission range $d_{\text{T}} = 100\,\text{m}$, the maximum propagation delay between two nodes is approximately $T_{\text{prop}} \approx 0.34\,\mu\text{s}$. Hence, in comparison to the expected maximum transmission delay $T_{\text{tran}} \approx 860\,\mu\text{s}$, the propagation delay can be neglected.

4.2.11 Lack of RTS/CTS Mechanisms

Similar to the IEEE 802.15.4/ZigBee standard (cf. [127]) no RTS/CTS mechanism (cf. [3, p. 82], [143, p. 118]) is foreseen in XLMMP. Due to the very short data messages in XLMMP, the overhead of additional RTS and CTS packets is considered too high. The advantage of repeating a collided RTS message instead of a data message is negligible in the investigated scenario.

4.2.12 Node Localization

In many WSN scenarios, including the investigated one, it is necessary to derive the position of the sensor nodes within the monitored area. This process is called *localization**. In this thesis, node locations are required for deducing the location of the sensed events as described in Section 4.3.5. Moreover, the information is needed for location-based routing discussed in Section 4.3.2.

Localization methods for WSNs have received significant research interest lately. Literature surveys are provided by, e.g., [15, 32, 48, 62, 110, 146, 178, 225, 287, 325]. There are basically five different techniques to obtain localization information on each sensor node: deterministic deployment, relying on global navigation satellite systems, methods based on environmental monitoring, range-based methods, and range-free methods. Additionally, there are hybrid approaches combining different methods. These techniques are briefly discussed in this section to select one that suits the discussed application scenario. The realization of the selected technique within XLMMP is described in Section 4.3.4.1.

When *deploying nodes deterministically*, their location can be programmed into the node before or during deployment (cf. [36]). However, deterministic deployment is rather impractical in the given scenario, as already discussed in Section 4.2.4.

The major drawbacks of solutions based on *global navigation satellite systems* like GPS† are the price (in relation to the overall node price), power consumption, and size of the additional hardware needed for each node (cf. [36, 224]). Moreover, satellite-based localization is infeasible in indoor, underground, underwater, and other concealed environments (cf. [143, p. 207]). In particular, the authors of [361] show that GPS performance is significantly degraded by forest canopies.

Localization methods based on *environmental monitoring* are also referred to as *scene analysis* (cf. [122]). Nodes monitor certain phenomena of their surroundings, e.g., by recording video data (see [248, Ch. 8]), and use this data to deduce their location. The method requires suitable sensors and a-priory knowledge of the environment which is not given in most cases.

Most *range-based methods* have, similar to GPS, increased hardware requirements to measure angles or distances between sending and receiving nodes (cf. [36]). They can be based on measurements of the angle of arrival (AOA), time of arrival (TOA), time difference of arrival (TDOA), interferometry, use received signal strength (RSS) measurements, or combine a subset of these measures to assess the location by tri-/multi-angulation or tri-/multi-lateration (cf. [230]). In particular, AOA would require rotating antennas or antenna arrays (cf. [248, Ch. 8], [258]), TOA requires exact time synchronization of the nodes, TDOA usually requires communication using two different technologies (like ultrasound and RF; see [235]), and interferometry requires either two transceivers per node or nanosecond-precise timing, precise

* Some authors prefer the term *positioning* instead.

† Global Positioning System.

phase control at the transmitter, and precise phase detection at the receiver (cf. [224, 299]). RSS-based methods (surveyed in [71]) only require hardware that is capable of measuring the RSS. This usually requires some support by the transceiver and a further channel of the analog-to-digital converter (ADC). Since in XLMMP an ADC is also needed for converting the analog sensor data and battery level, the additional hardware effort is relatively small. Moreover, the RSS needs to be measured anyway for using the foreseen CSMA method. Still, using the RSS for fine-grained localization comes with further challenges. According to [71], these include interference and measurement errors due to multi-path effects, shadowing, transmitter and receiver variability, diversity of antenna orientation, and atmospheric conditions. All these issues are relevant in the forest scenario.

Range-free localization methods usually do not need any specific hardware. Instead of doing physical range measurements, they rely on connectivity information only. Examples of such methods include DV-Hop [219], LAEP [327], MDS-MAP [278, 279], CPE [77], APIT [117], Amorphous [214, 215], centroid-based methods [54, 59], directional beacon-based methods [171, 172], RAW [327], DHL [342], PDM [179], CrMCs [346], and SLA [16]. However, not all of these methods render feasible in the investigated scenario.

For example, methods based on centroids and directional beacons* as well as, for example, APIT and SLA, require direct signaling from beacons to sensor nodes. When the same frequency is used for this signaling and for data communication, it has to be ensured that the signals do not interfere with ongoing communications within the WSN. If different frequencies are used, each node needs an additional receiver which again requires additional hardware. Moreover, relatively strong signals or a relatively high beacon density is needed to cover the monitored area.

Some methods (like MDS-MAP, CPE, SLA), require to collect global connectivity information and carry out the calculations at a central entity. This is not favorable due to the large number of nodes and the resulting communication overhead. Moreover, these solutions usually require unique node IDs†.

The localization method used by XLMMP mainly falls into the category of distributed, multi-hop, range-free localization methods. These methods aim at enabling the nodes to estimate their distance to at least three beacon nodes by providing them with the number of communication hops needed to reach each beacon and with an estimate of the mean distance traveled by one hop‡. The nodes are then able to calculate their approximate distance to each beacon by multiplying the number of hops with the mean hop distance. Using tri- or multilateration based on the distances to the beacons and the known beacon locations, the nodes are then able to calculate their own position.

In particular, XLMMP is similar to the DV-Hop protocol which does not rely on node IDs. Further related protocols are RAW, Amorphous, and PDM. In principle, also DHL, CrMCs, and LAEP fall into this category, but their algorithms require all nodes to determine their actual number of neighbors (or even maintain a list of neighbors), which cannot be easily done without

* Beacons (sometimes also referred to as anchors, locator nodes, seed nodes, or landmarks) are (usually few) nodes that are aware of their position. This information is provided by other means, e.g., GPS or deterministic deployment. Beacons provide location information to the other nodes in the WSN. In this thesis, the sinks take the role of beacons.

† Unfortunately, the requirement of node IDs is not a classification criterion in any available survey of localization protocols.

‡ In Section 4.3.4.1, XLMMP's localization method is described in more detail. The mean hop distance is denoted by $\overline{d}_{\mathrm{w}}$ (mean ring width) there.

node IDs and due to the nodes' unsynchronized active/sleep periods.

The main difference between RAW, DV-Hop, Amorphous, and PDM is their approach to calculate the mean hop distance. RAW assumes that the mean hop distance is equal to the transmission range, which overestimates the distance and only holds for very dense sensor networks. DV-Hop calculates the mean hop distance based on the number of hops and physical distance actually measured between the sinks. This tends to result in higher errors when the node density varies significantly within the sensor network, but is reported to perform better than RAW. When using Amorphous, each sensor calculates the mean hop distance based on the mean node density and transmission range using a closed-form equation (Eq. (5) of [215]). An improvement to DV-Hop is achieved by additional smoothing done by the algorithm. The smoothing, however, relies on information collected from all neighbors and hence cannot be easily implemented without node IDs. PDM uses a truncated singular value decomposition (SVD, [105, Sec. 2.5.3], [116, Sec. 11.10]) technique to calculate the node positions. While it in principle uses the same information as DV-Hop, it reportedly achieves better results by retaining as much topological information as possible and by reducing the effect of measurement noise. The downside is a slightly increased communication overhead and the relatively high computational complexity of SVD.

An example for *hybrid localization methods* is the approach of [259], which uses the coarse results of a range-free localization method (similar to DV-Hop) to initialize a more exact refinement that utilizes range-based methods between neighboring nodes. Another example is RDV-bop proposed in [308], which estimates the distance between beacons and their one-hop neighbors using RSS and DV-Hop for all other distances.

The localization method of XLMMP is designed similar to the DV-Hop protocol, which does not require node IDs, works in a decentralized fashion, and does not require direct (i.e., single-hop) signal or packet exchange between the beacon(s) and sensor nodes. However, if, for example, RSS measurements are additionally used to estimate the node's position within their ring*, it can be seen as a hybrid approach.

Finally, it should be noted that several extensions to DV-Hop have been proposed by various researchers, e.g., [61, 177, 181, 182, 220, 281, 337, 345]. The extensions aim at improving different aspects of DV-Hop's performance. While a detailed investigation of their applicability within the scope of XLMMP is not in the focus of this thesis, the extensions offer starting points for future improvements.

4.2.13 Memory Constraints

Wireless sensor nodes are usually equipped with up to three types of memory (see [143, p. 21], [82, p. 77], [167], [3, p. 2–5, 39]): (1) fast but volatile random access memory (RAM) to store temporary data, (2) non-volatile memory, like read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), for storing unchanging program code, and (3) non-volatile flash memory that can be used to support the first two memory types at the expense of speed and power.

In particular, the memory available for storing event messages (see Sections 4.3.6 and 5.2.2) is of interest, i.e., the expected size of RAM and flash memory. Even in the case of RAM, current developments (as of mid 2012) achieve a volumetric storage density (bit per volume) of approximately $10\,\mathrm{Mbit/mm^3}$ (see Appendix Section B.3). It is shown in Section 5.2.2 that due

* This and alternative improvements are discussed in more detail in Section 4.3.2.5.

to the small message size, 10 Mbit are utterly sufficient to store an adequate number of event messages. Hence, the size of the memory hardware could even be decreased below 1 mm^3, and from this point of view, message memory does not constitute a significant constraint.

4.2.14 Data Rates

Several state-of-the-art low-power and small-sized transceiver technologies are surveyed in [359, Sec. 2] and [125, Sec. 2.2.3]. The surveys show that data rates in the order of 100 kbps can be achieved with relatively small (approximately* 2.5 mm^3), simple (using OOK or FSK modulation†), and power-saving (less than 50 mW) transceivers.

While the selection of the optimum transceiver with respect to, e.g., data rate, size, power consumption, and baseband frequency (influencing path loss and antenna size) needs to be done in future work, a data rate of 100 kbps is assumed in the following.

4.2.15 Lack of Unique IDs

XLMMP is designed such that unique node IDs and addresses are not needed. This decision is motivated in the following.

Unique IDs increase the heterogeneity of the nodes. This makes node development more complex. The IDs could be chosen globally unique, unique within the WSN, or locally unique within a neighborhood of nodes (cf. [143, Sec. 7.1.3]). In principle, providing nodes with unique IDs can be done before or after deployment.

One of the latter option's advantages is that nodes are able to negotiate their IDs adaptively. This allows to keep the address space short even in the face of node failures and redeployments. On the other hand, the nodes then need to execute suitable algorithms for ID (and/or address) management.‡ Usually, these algorithms are distributed and comprise significant communication overhead. Hence, they are quite energy consuming.

IDs provided before deployment are usually hard-coded and hence globally unique. For instance, an extra chip (like the DS2401 or DS2411 silicon serial number chips§ might be used that hold a globally unique ID. However, to achieve global uniqueness, the IDs need to be quite long¶. Therefore, using the IDs in communication would significantly increase the message size in the present scenario. Moreover, the chips themselves cause additional cost with respect to price, energy consumption, and volume.

Alternatively, a customized ID range could be provided for a specific deployment site such that the IDs are unique within the WSN only. In the application scenario of this thesis, at least 17 bit are needed to identify 100000 nodes. Such an ID would still constitute a good portion of the message length. Furthermore, special care has to be taken when redeploying nodes. On the one hand, the ID range is limited. On the other hand, IDs still in use should not be duplicated. Hence, this approach is less adaptive, e.g., when the node density or size of the monitored area should be increased significantly.

---

* When assuming a circuit thickness of approximately 100 µm; see Section B.3.

† OOK: On-Off Keying; FSK: Frequency Shift Keying

‡ Surveys of such methods are provided by, e.g., [143, Sec. 7.2–7.4], [316, 349].

§ The DS2401 (cf. [194]) is used by, e.g., for Mica2 (cf. [68]) and MicaZ (cf. [68, 262]) sensor nodes. The DS2411 (cf. [195]) is used by, e.g., Telos-B (cf. [209]), tmote sky (cf. [210]), and Scatterweb MSB430 (cf. [58]) nodes.

¶ 48 bit in case of DS2401 and DS2411.

Finally, providing and sending IDs only makes sense if they are actually used at the receiver. For this, they need to be mapped to more meaningful node attributes like node locations. The maintenance of suitable mapping tables at the nodes or at the sinks would involve further communication overhead.

Therefore, it is investigated in this thesis how the desired functionality can be achieved without unique node IDs and addresses.

4.2.16 Hardware Clock and Lack of Time Synchronization

Sensor nodes are usually provided with crystal oscillators that provide pulses with a specific frequency. These pulses can be used for basic timing tasks, e.g., to provide clock signals for the node's processor and the baseband frequency to the transceiver.

In principle, the clock signals can be used to maintain a node-local time. Moreover, if the node is informed of the global time once, the node is able to map its local time to the global time which then can be added to messages (e.g., to report exact occurrence times of events) and for synchronized interaction between nodes (e.g., using synchronized duty cycles or medium access methods like time division multiple access (TDMA)).

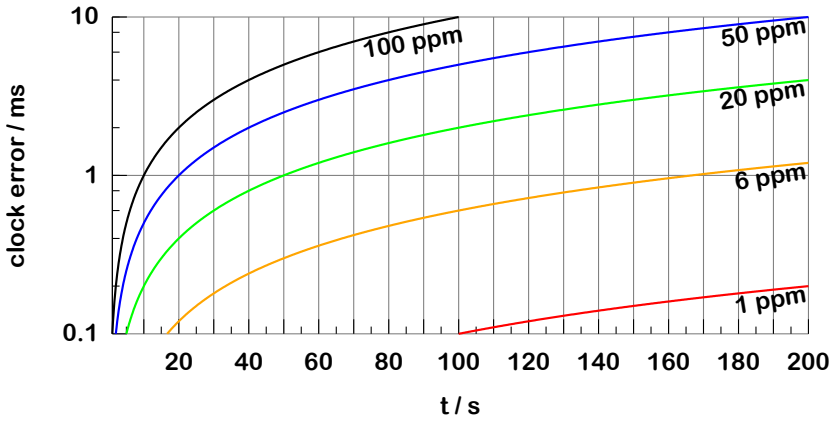


Figure 4.7: Estimation of clock error (y-axis, logarithmic) in dependence of drift rate (curves) and time since last synchronization (x-axis).

Unfortunately however, the crystal oscillators are subject to frequency drift caused by manufacturing imprecision, aging, node acceleration, and environmental conditions* (cf. [143, Sec. 8.1.2], [261]). Hence, when the system is built such that each node needs to know the global time up to some specific precision, the nodes' clocks need to be resynchronized frequently. The required resynchronization frequency is determined by the quality of the oscillators and by the required precision (see [143, p. 204]). Following [143, p. 204], Fig. 4.7 gives a rough estimation of the expected clock error (y-axis, logarithmic) after time t (x-axis) for different values of the oscillator drift rates† (curves). For example, the *iSense CM30x* wireless nodes feature a high-accuracy real-time clock with a drift rate of only 6 ppm (cf. [64, p. 4]). If a clock

* Like temperature, electric and magnetic fields, particle radiation, pressure, and humidity, while the effects of the latter two can be significantly reduced by proper packaging (cf. [261]).

† Drift rates are usually specified in *parts per million* (ppm), which is a dimensionless unit. It describes the difference from a million oscillations of the actual number of oscillations achieved by the oscillator during the time nominally needed for a million oscillations (cf. [143, p. 204]). For typical, inexpensive crystals used in wireless sensor nodes, the drift rates lie in the order of 1–100 ppm (cf. [143, p. 204], [261]).

accuracy of 1 ms needs to be maintained*, the nodes still need to be resynchronized approximately every 170 s. If nodes are aware of their own clock drift, situation improves since they can correct their clock accordingly. But since the drift is not constant, frequent synchronization is still required.

Nodes can be synchronized via external time sources or via time synchronization protocols (cf. [143, p. 207]). External time sources usually require additional hardware. For example, nodes could be synchronized using GPS. This is considered infeasible as already discussed in Section 4.2.12. Also the use of available long-range radio time signals† would require additional hardware and energy. Alternatively, dedicated time signals could be provided by the WSN's operator via the sinks and/or additional time beacon nodes. However, this shares the same drawbacks (interference with ongoing communication or additional hardware) discussed in Section 4.2.12 for localization signals. Moreover, any time a sensor node expects a time signal, it needs to switch to reception mode.

An alternative to external time sources is the application of time synchronization protocols. Examples of these protocols are surveyed in, e.g., [168, 239, 244, 282, 296], [143, Sec. 8.2–8.4], and [3, Sec. 11.4–11.10]. However, most protocols seem to rely on unique node IDs.‡ Moreover, and even worse, all synchronization protocols share the same main disadvantage. During each synchronization, all nodes have to at least receive one synchronization message. This implies that in each neighborhood at least one node needs to send a message. In the present application scenario, due to the limited transmission range and large area to be monitored, this would require more than $\frac{A_F}{A_T} \approx 2300$ messages for each time synchronization. The synchronization protocols hence induce a significant communication overhead, in particular in WSNs tailored towards the long-term detection of rare events.

Therefore, this thesis investigates whether synchronization is dispensable. As a result, no event times are included in the event messages§, no TOA ranging can be used for localization, TDMA cannot be used as medium access protocol, and the active/sleep periods of the nodes are unsynchronized.

4.2.17 Summary and Scope for Development

Based on the discussions in Sections 4.2.1 to 4.2.16, partially rather high demands and severe restrictions are posed on the WSN protocol. The main challenges are provoked by the lack of node IDs and time synchronization. Both are attributed to keeping the overhead low in the face of the aspired large lifetime despite the nodes' hardware constraints.

Fortunately, the application scenario also offers several opportunities that simplify the protocol design. For example:

- While the duplication of messages should be avoided to save energy and bandwidth, it can be tolerated from an application point of view.
- The sequence of messages is not viable. Hence, out-of-order messages do not pose a thread.

* This duration roughly reflects the time needed to transmit a message. See Section 4.2.10 for details.

† Like WWV in the United States or DCF77 in Germany.

‡ Unfortunately, the requirement of IDs is not a protocol property which is checked explicitly in the given surveys. A more detailed investigation and identification of synchronization protocols that might work even without node IDs is postponed to future work.

§ As discussed in Section 4.2.9, this is not critical in the investigated application scenario.

- Due to the small message size, no segmentation is necessary.
- There is only a single application. Hence, no multiplexing of application data (as provided by the transport layer in the Internet) is needed.
- The WSN is located at the edge of the Internet and it is optimized for carrying out its main application-specific task. Offering communication services to other applications is not foreseen. Hence, no support of Internet protocols (like TCP or IP) is necessary.
- No strict end-to-end reliability of single messages is needed. As long as report-worthy (i.e., to some degree persistent) events are reported reliably to any sink, the loss of single EVMs is (and needs to be) acceptable.

Finally, note that XLMMP relies on a few further basic mechanisms a node needs to provide that are not discussed in full detail in this thesis. These include the following.

- Since XLMMP is designed to be energy-aware, a possibility to measure the remaining energy level of the battery needs to be provided. An approximation of the energy level can be obtained by measuring the battery's voltage and providing this analog information to the processor via an additional ADC channel.
- A random number source is needed, for example for the random back-off during medium access. For this, sensor noise may be processed by a randomness extractor.
- The employed transceiver is expected to provide basic physical-layer services (like de-/modulation and channel coding).

4.3 Protocol Details

In this section, details of XLMMP are discussed.

4.3.1 Special-Case Protocol Behavior

A variety of situations can occur where the basic behavior of XLMMP, as described in Section 4.1, is challenged. A selection of these situations and their remedies are described in the following. The identification of further situations is one of several important tasks for future work discussed in Section 4.4.

4.3.1.1 No Next Hop Available

A sending node assumes that there is no next hop available if it does not receive an EVM with lower hop count before the maximum back-off delay* is reached. There are several situations that make this impression to the sending node. These include the following.

- Situation 1: The EVM collides on the wireless channel.
- Situation 2: There is no operational next-hop node (i.e., the sending node's hop counter is outdated).

* The *maximum back-off delay* is introduced in Section 4.3.3.

- Situation 3: There are operational next-hop nodes, but all of them are currently sleeping.
- Situation 4: There are operational and active next-hop nodes, but their inPool is full.
- Situation 5: There is at least one active next-hop node that received the EVM and sent an ACK (i.e., updated EVM), but no ACK reached the sending node due to collision on the wireless channel or asymmetric connectivity*.

Since the sending node is not able to differ between these situations, it treats them equivalently as follows. When the maximum back-off delay is reached, the sending node resends the EVM until the number of retrials reaches an upper limit, which is yet to be determined. Then, the sending node increases its own hop count by one and restarts the retrial process. This allows more neighbors to be involved in the receiver contention. When the maximum hop count is reached, the sending node assumes to be isolated, discards all EVM in its inPool and outPool, periodically listens to the wireless channel, and refrains from generating new EVMs until it is able to overhear messages from other, potentially redeployed, nodes.

Situation 5 likely generates EVM duplicates, which can be reunified when meeting on their way toward the sink (or ultimately at the sink). However, duplicates should still be avoided in order to reduce energy consumption.

4.3.1.2 Failure of Responsible Node and Loss of EVMs

EVMs may get lost, for example, because a node that is responsible for one or several EVMs (kept in its outPool) fails due to battery depletion or other hardware failures. However, as discussed in Section 4.2.17, the loss of single EVMs an be tolerated up to some degree as long as report-worthy, non-transient events are reported reliably to the sinks. Still, even if the loss is partly acceptable, it might significantly increase the event reporting delay and the energy consumption due to futile previous communication effort.

To increase the event reporting reliability, each source node repeats its EVM until the event disappears or the sinks revoke their interest in the event by updating the event reporting thresholds accordingly using SMSGs. These threshold updates can hence be interpreted as end-to-end ACKs. Selecting the frequency of EVM repetition at source nodes allows to trade-off the WSNs energy efficiency against its end-to-end reliability. The investigation of suitable repetition frequencies is postponed to Chapter 5.

4.3.1.3 NHF not Received

If a node that is in receiver contention mode does not receive the NHF message of the source node in time, it might relay the EVM, even if another node relayed the EVM earlier. This also results in a duplication of the EVM. Similar to Section 4.3.1.1, an early merging of the duplicates is desirable.

* Currently, the protocol allows nodes to directly accept EVMs sent from nodes with arbitrarily higher hop count to minimize the number of hops required by each EVM. Alternatively, to avoid asymmetric connectivity, only EVMs whose SENDER_HC is greater by exactly one than the receiver's hop count could be considered for reception. The investigation which of both approaches actually performs better is postponed to future work.

4.3.2 Hop Count-Based Routing

Before presenting the calculation of the back-off delay in Section 4.3.3, the behavior of the hop count-based routing is investigated in more detail. While doing so, is can be shown that relying solely on the hop count as information on the position within the WSN's communication topology leads to an effect that may significantly increase the energy consumption and end-to-end delay.

4.3.2.1 Location-Based Routing Methods

There are two types of node locations: *virtual location* of the nodes within the WSN's communication topology and *physical location* of the nodes within the monitored area. In literature, virtual locations are sometimes also referred to as *logical* (cf. [280]) or *relative* (cf. [36, 121, 258]) locations. Physical locations are also called *absolute* (see [121, 258, 259]), *geographic* (see [36, 280]), *global* (see [258, 259]), or *true* (see [240]) locations. Knowing the nodes' physical position often is vital to deduce the location of the sensed phenomenon (see [259]).

XLMMP exploits the correlation between virtual and physical locations in two ways. On the one hand, the correlation allows to estimate the nodes' physical location based on their virtual location. This physical localization is described in more detail in Section 4.3.4.1. On the other hand, the sensor data of physically neighboring nodes tends to be highly correlated in many application scenarios of WSNs, including the early forest fire detection. Hence, if physically neighboring nodes also have similar virtual locations in the communication topology, their messages can be merged more easily. This is exploited in Section 4.3.7.

Both types of locations can be used for *direction-based routing* (also known as position-based routing, location-based routing, or location-aware routing; cf. [82, 103]). Direction-based routing that is based on physical node locations is commonly known as *geographic routing* (cf. [240]). Geographic routing algorithms, in particular its greedy forwarding variants, suffer from dead ends, since not every sensor node might have a neighbor that is physically closer to the destination than the node itself. Other variants of geographic routing algorithms are not loop-free (cf. [103, 240]). Since in contrast to physical locations, virtual locations are based on connectivity information, they do not share these problems (cf. [240]) and hence are preferred in this thesis.

4.3.2.2 Hop Count Update

After random deployment, the sensor nodes are unaware of their position within the monitored area and also within the WSN's topology. For the hop count-based routing and localization, all sensor nodes need to get informed of their distance to the three sinks, measured in the number of communication hops to the sinks. For this, each node maintains a vector $\mathbf{d}_\mathrm{h} = \langle d_\mathrm{h}^{(0)}, d_\mathrm{h}^{(1)}, d_\mathrm{h}^{(2)} \rangle$ of the three distance estimates. The vector can be seen as a node's coordinate* in the hop count-based coordinate system. It is initially† chosen as $\mathbf{d}_\mathrm{h} = \langle N_\mathrm{h}^{(\mathrm{upper})}, N_\mathrm{h}^{(\mathrm{upper})}, N_\mathrm{h}^{(\mathrm{upper})} \rangle$, where

* In this thesis, angle brackets are always referring to coordinates. If the type of the coordinate (e.g., hop count-based (h), physical (p), or grid (g)) cannot be recognized by its (e.g., numerical) elements, the clarifying subscript is added to the right-hand side bracket, e.g., $\langle 0,0,0 \rangle_\mathrm{h}$.

† The nodes' $\mathbf{d}_\mathrm{h}$ can also be reset to $\langle N_\mathrm{h}^{(\mathrm{upper})}, N_\mathrm{h}^{(\mathrm{upper})}, N_\mathrm{h}^{(\mathrm{upper})} \rangle$ by using specific SMSGs. See Section 4.3.6.2 for details.

$N_{\mathrm{h}}^{(\mathrm{upper})}$ is an upper bound of the maximum number of communication hops between nodes and sinks and is estimated in Section 4.3.6.1.

The hop count information is included in hop count update SMSGs, which are broadcast by the sinks. The following detailed discussion exemplarily focuses on a single sink, say Sink 1. Sink 1 initiates the hop count update by sending a corresponding SMSG with hop count $d_{\mathrm{h}}^{(1)} = 0$. All awake direct neighbors of the sink receive this SMSG, increase $d_{\mathrm{h}}^{(1)}$ by one, store $d_{\mathrm{h}}^{(1)} = 1$ in their local hop count vector $\mathbf{d}_{\mathrm{h}}$, and repeat the SMSG with increased hop count. All recipients of the updated SMSG compare the SMSG's hop count to their own, locally stored one. When their own hop count is greater than the received one, they store and resend the received hop count increased by one. Otherwise, i.e., if a node receives a hop count update SMSG containing a hop count that is greater than or equal to its own, it ignores the SMSG.

If a node wants to send a hop count update SMSG but the wireless channel is busy, it changes to sleeping mode and retries after a random time. This additional delay decreases the node's time spent in energy-hungry receiving mode and increases the chance that previously sleeping nodes are now awake and can be informed of their hop count. Still, due to the defensive duty cycle, the sink might have to initiate the hop count update several times before all nodes are aware of their hop count. On the other hand, hop counts are included in all messages (EVMs, NHFs, and SMSGs) traversing the WSN. Hence, all of these messages can be used by the nodes to update their hop count.

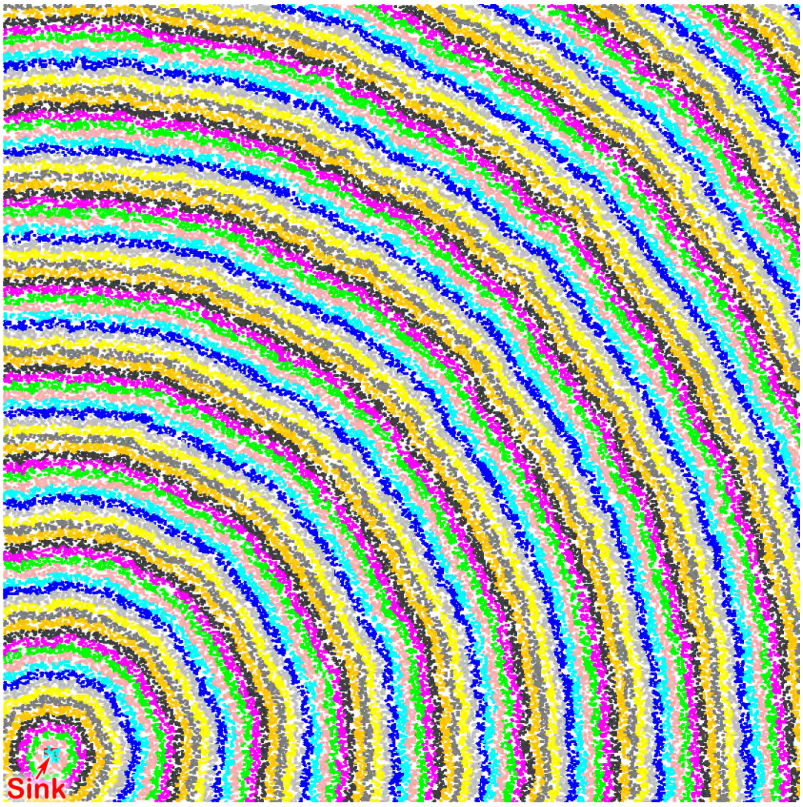


Figure 4.8: MASON simulation shows sensor nodes forming rings with same hop count.

Figure 4.8 illustrates MASON* simulation results for random uniform node deployment. The figure has a resolution of 837×837 pixels, each pixel referring to 10 m in reality. Hence, a square area of approximately $70\,\mathrm{km}^2$ is simulated. The $N_S = 100\,000$ shown colored dots have a diameter of 4 pixels and illustrate the nodes' sensing ranges with radius $d_S = 20\,\mathrm{m}$. Hence, the figure also allows to exemplarily assess the coverage achieved with a node density of $\overline{N}_d \approx 1400\,\mathrm{km}^{-2}$. The transmission range of all sensor nodes is $d_\mathrm{T} = 100\,\mathrm{m}$. For the investigation, the focus is on a single sink, which is located close to the lower left corner of the figure. The colors of the nodes' sensing ranges reflect their hop distance to the sink after convergence of the hop count update. The colors are reused every ten hops.

It can be seen that sensor nodes with the same hop distance to the sink form rings around the sink. In the following, *Ring 0* denotes the sink itself, *Ring 1* denotes the area comprising the neighbors of the sink, *Ring 2* is the area containing the nodes with a hop count of two, and so forth. Provided sufficient node density and/or transmission range, the width of each ring is approximately equal to the transmission range d_T and the ring borders can be approximated by concentric circles around the sink. Note that the width of the rings actually is always smaller than the (idealized) transmission range†, especially for smaller node densities. This error accumulates and needs to be taken into account when estimating the maximum number of hops in the WSN and when relying on localization based on hop counts, as considered in Sections 4.3.6.1 and 4.3.4.1, respectively.

4.3.2.3 Number of Next-Hop Neighbors of an Arbitrary Node

Numerical evaluation of the proposed protocol's performance done in Chapter 5 identifies the number of a node's next-hop neighbors as an important system parameter. This number is estimated in the following for a randomly chosen node, called the *tagged node*.

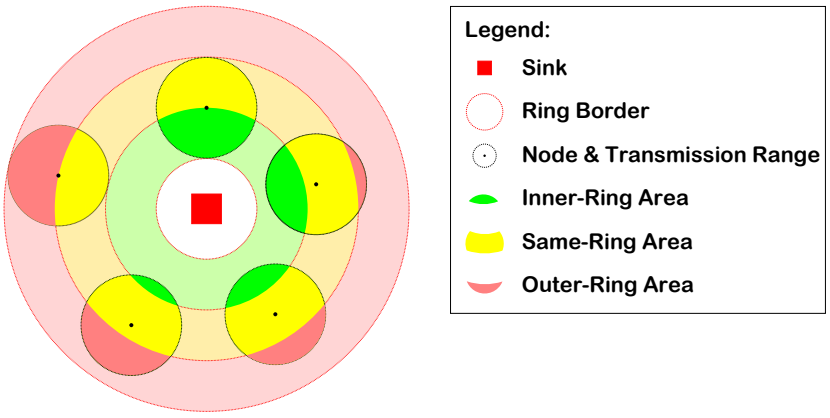


Figure 4.9: Influence of node location within its ring on the overlap of its transmission range with adjacent rings.

* The multi-agent simulation toolkit MASON is a discrete-event simulation library for the Java programming language. MASON and its documentation is available at `http://cs.gmu.edu/~eclab/projects/mason/` (last accessed: 21 Jan. 2013). This thesis applies Version 15 of the toolkit. It is used to simulate and visualize the random deployment of sensor nodes in a two-dimensional area, the coverage and connectivity achieved by the sensor nodes and sinks (assuming a simple connectivity model where two nodes are able to communicate when they are less than transmission range d_T apart), and the formation of hop-count rings using SMSGs.

† A more detailed discussion is provided in Section 4.3.2.4.

Neighbors of a tagged node fall into three classes: (1) *outer-ring neighbors* (i.e., previous-hop nodes for EVMs), which are exactly one hop count farther away from the sink than the tagged node, (2) *same-ring neighbors*, which have the same hop count as the tagged node, and (3) *inner-ring neighbors* (i.e., next-hop nodes for EVMs), which are exactly one hop closer to the sink than the tagged node. As shown in Fig. 4.9, the tagged node's location within its ring significantly affects how much of its transmission range overlaps with inner-ring, same-ring, and outer-ring areas.

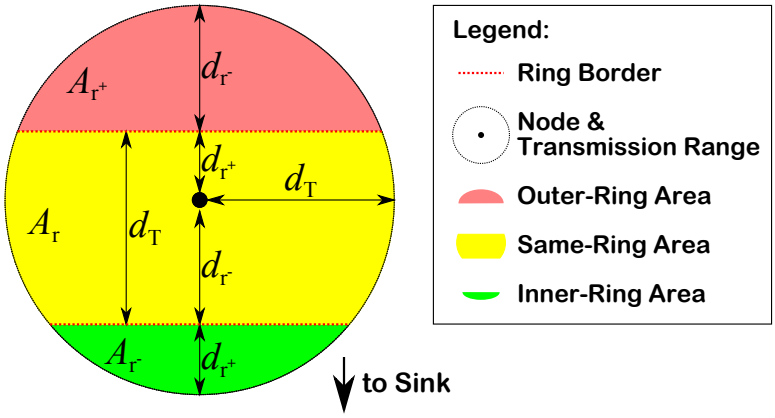


Figure 4.10: Estimating the overlap of transmission range with adjacent rings.

To estimate the mean area sizes of these overlaps, the tagged node's transmission range is assumed being much smaller than the radius of the ring it is located in. This especially holds true for rings that are less close to the sink and significantly simplifies the estimation because the borders between rings can then be approximated by straight lines within the transmission range of the tagged node, as illustrated in Fig. 4.10. This approximation avoids calculating tedious surface integrals when deriving the means.* When the tagged node's distance to the next inner ring is equal to d_{r^-} (with $0 \le d_{r^-} \le d_T$), the ring borders divide the transmission range area A_T into the outer-ring segment of height d_{r^-}, the inner-ring segment of height $d_{r^+} = d_T - d_{r^-}$, and the same-ring area of height d_T in between.

The size A_{r^+} of the outer-ring segment can be calculated via the formula

$$A_{r^+} = d_T^2 \arccos\left(\frac{d_{r^+}}{d_T}\right) - d_{r^+}\sqrt{d_T^2 - d_{r^+}^2}\,, \tag{4.2}$$

well-known from mathematical geometry. For additional simplification, d_{r^+} of a randomly selected node is assumed to be uniformly distributed between 0 m and d_T. For $d_T = 100\,\text{m}$ and $0\,\text{m} \le d_{r^+} \le d_T$, the mean outer-ring segment size $\overline{A}_{r^+}$ is then given by†

$$\overline{A}_{r^+} = \frac{1}{d_T}\int_{0\,\text{m}}^{d_T} A_{r^+}\,\mathrm{d}d_{r^+} = \frac{2}{3}d_T^2 \approx 6667\,\text{m}^2\,. \tag{4.3}$$

* A closer investigation of the accuracy of this approximation is omitted at this point. It should be reconsidered when including more realistic radio propagation models into the protocol's investigation.

† The derivation is shown in more detail in Appendix Section B.11.

Due to symmetry, the mean inner-ring segment size is also given by $\overline{A}_{\mathrm{r}^-} = \overline{A}_{\mathrm{r}^+} \approx 6667\,\mathrm{m}^2$, and consequently, the mean same-ring area can be calculated via $\overline{A}_{\mathrm{r}} = A_{\mathrm{T}} - \overline{A}_{\mathrm{r}^+} - \overline{A}_{\mathrm{r}^-} \approx 31\,400\,\mathrm{m}^2 - 2 \cdot 6667\,\mathrm{m}^2 \approx 18\,083\,\mathrm{m}^2$.

Given the node density $\overline{N}_d \approx 1400\,\mathrm{km}^{-2}$, the mean number of inner-ring neighbors of a randomly selected sensor node can therefore be approximated as

$$\overline{N}_{\mathrm{r}^-} = \overline{A}_{\mathrm{r}^-} \cdot \overline{N}_d \approx 9\,, \tag{4.4}$$

and equivalently, its mean number of outer-ring neighbors is approximately

$$\overline{N}_{\mathrm{r}^+} = \overline{A}_{\mathrm{r}^+} \cdot \overline{N}_d \approx 9\,.$$

At this point, a more rigorous validation of the approximate result given in Eq. (4.4) could be performed. Here, the term *validation* refers to the process of getting confidence in the statement that the derived approximation of $\overline{N}_{\mathrm{r}^-}$ is suitable, i.e., the approximate result $\overline{N}_{\mathrm{r}^-} \approx 9$ is sufficiently close to the number of inner-ring neighbors that can be expected in a real-world implementation and deployment of the aspired WSN. However, this validation is not in the focus of this thesis, since the approximate result of $\overline{N}_{\mathrm{r}^-}$ serves as a rough point of reference only. The evaluation methods proposed in Chapter 5 are developed such that they are feasible for a wide range* of $\overline{N}_{\mathrm{r}^-}$ and the influence of changes of $\overline{N}_{\mathrm{r}^-}$ on the WSNs performance is investigated in Section 5.8.8. An initial step towards proper validation could be carried out by deriving the distribution of the number of inner-ring neighbors from the multi-agent simulation model developed for this thesis and used, e.g., in Section 4.3.2.2. For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

4.3.2.4 Number of Next-Hop Neighbors of an EVM-Relaying Node

In Section 4.3.2.3, it is assumed that d_{r^+} of a (randomly selected) sensor node is uniformly distributed between 0 m and d_{T}. However, without modification of the proposed hop count-based protocol, this assumption is too optimistic for nodes that are actually responsible for relaying EVMs in the WSN, because messages tend to be routed via nodes located at the outer edges of the hop count rings. Unfortunately, these nodes have a significantly reduced number of inner-ring neighbors.

Figure 4.11 provides a schematic illustration of this effect by showing the behavior of two protocol variants in the upper and lower half of the figure, respectively. For simplified illustration, the investigation in reduced to a scenario where the active sensor nodes that potentially contribute to the communication are assumed to be randomly† located on a straight line between the EVM's source node and the EVM-receiving sink. Identically for both protocol variants shown in the figure, this straight line of active sensor nodes is represented by a horizontal line with dots, where the dots represent the active sensor nodes. Inactive sensor nodes

* For example, for being able to calculate the second moment of the waiting time using the method of phases and default parameters defined as *RealSet* in Table 5.2, $\overline{N}_{\mathrm{r}^-}$ may take values between 1 and approximately 2500. The limit decreases significantly for increasing other model parameters (in particular N_{c} introduced in Section 5.2.2). The limit can again be increased by applying the approximation proposed in this thesis (see Section 5.5) or using DES. The parameterization and scalability of the different evaluation methods proposed and applied in this thesis are discussed in much more detail in Chapter 5. In particular, Section 5.6.1 provides a summarizing comparison of the different methods.

† Uniformly distributed and identical for both protocol variants discussed in Figure 4.11.

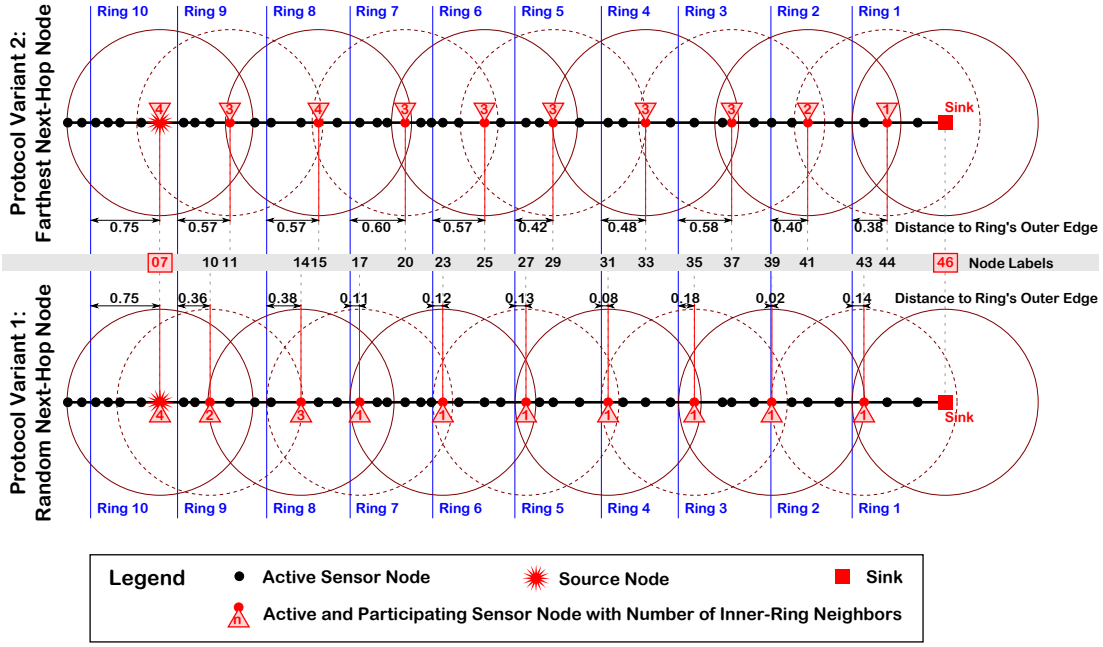


Figure 4.11: Schematic illustration of EVM accumulation at the rings' edges.

that might be located off the straight line are not shown for reducing graphical complexity. For reference, the shown active nodes are labeled (from 01 to 46). Those labels are provided in the middle of the figure for all nodes actually participating in communication. The sink is the rightmost node with highlighted label 46. The source node, which initially holds the EVM that needs to be communicated to the sink, is highlighted with label 07. Sensor nodes that actually participate in forwarding the EVM are additionally highlighted by triangles, their transmission ranges are shown*, and their distance† to the outer (i.e., left) edge of their ring is given in the middle of the figure. The triangles additionally contain the participating nodes' number of next-hop, i.e., inner-ring neighbors.

For example, there are four nodes in Ring 5 with labels 27–30. The transmission range of the leftmost node in Ring 5 (Node 27)‡ reaches almost to Node 22 to its left. This location marks the outer edge of Ring 6.

The discussion starts with focusing on the lower half of Fig. 4.11 labeled with "Protocol Variant 1: Random Next-Hop Node". This protocol variant is identical to the behavior described in Section 4.1, where any inner-ring neighbor of an EVM-sending node might serve as (equally probable) next hop, regardless of its location within its ring.

First, Node 07 of Ring 10 transmits the EVM to its neighborhood. Four of its neighbors (Nodes 08–11) are members of the next inner ring (Ring 9) and conduct receiver contention. Any of these inner-ring nodes may serve as next hop. Here, Node 10 is randomly selected, i.e., it responds first. It can be seen that the distance of Node 10 to the outer edge of Ring 9 ($0.36 d_T$) decreases significantly when compared to the distance of Node 07 to the outer edge of Ring 10 ($0.75 d_T$). Since Node 10 is closer to its ring's outer edge, its inner-ring neighbors

* The transmission ranges are dashed for participating nodes located in odd-numbered rings.
† Measured in fractions of the transmission range d_T.
‡ The (dashed) transmission range of Node 27 is shown in the lower half of the figure.

in Ring 8 are fewer (only two) and also located close to their ring's outer edge. This trend is followed in each hop. Consequently, after a few hops, all EVM forwarding nodes tend to be the ones closest to their rings' outer edge. These nodes likely have only one next-hop neighbor which is again the one closest to its ring's outer edge.

XLMMP already has a countermeasure for tackling these situations, as explained in Section 4.3.1.1. Consider an EVM-holding node that sits at the outer edge of its hop count ring and does not find an active inner-ring neighbor. This node retries transmission several times and eventually increases its own hop count. Consequently, it changes its topological location from the outer edge of Ring i to the inner edge of Ring $i+1$. All of its previous same-ring neighbors are now inner-ring neighbors, would respond to its EVM transmission, and are likely having more success in forwarding the message further towards the sink.

However, the countermeasure's significant disadvantage in this scenario is the increased delay and energy consumption for forwarding the message across the ring, since the EVM-holding node first conducts several retrials, and after increasing the node's hop count, at least two successful communications are needed to pass the ring. Moreover, by increasing the hop count of the EVM-holding node, the width of the next-outer ring is increased. Hence, it is made harder for subsequent messages to cross the next-outer ring.

4.3.2.5 Protocol Improvement

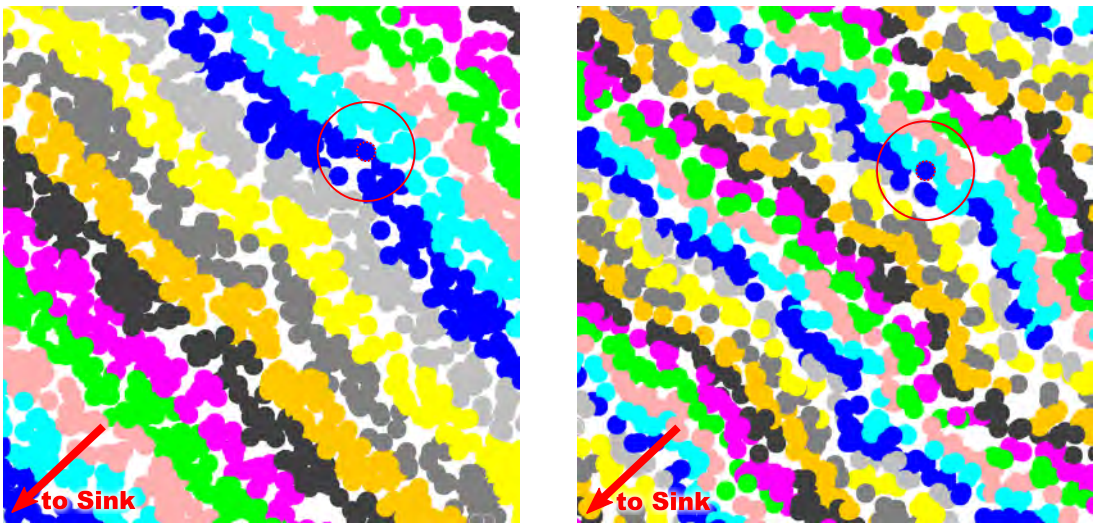
The performance can be improved when nodes are made aware of their location within their ring. This is exemplarily shown in the upper half of Fig. 4.11, which is labeled with "Protocol Variant 2: Farthest Next-Hop Node". Here, those inner-ring neighbors are preferred that have the farthest distance from their ring's outer edge. Following this rule, message-routing nodes stay more distant to their ring's outer edge and the number of their inner-ring neighbors stays higher. Note that depending on the nodes' arrangement, the distance from the participating nodes to their outer edge might even increase. This can be observed in the figure for the transitions from Rings 8 to 7, 5 to 4, and 4 to 3.

However, to follow this protocol variant, the nodes need to know their location within the ring, which is not yet the case when applying the proposed protocol, but can be achieved by one or by a combination of several of the mechanisms explained in the following.

Decreasing the transmission range for SMSG transmissions. When the hop count update SMSGs are sent with reduced transmission range (e.g., half of the transmission range d_T of EVMs), the granularity of hop counts can be increased. Consequently, the overlap of an EVM-sending nodes' transition range with at least one inner ring increases as well.

This is illustrated in Figs. 4.12(a) and (b), which show example MASON simulation results for SMSG transmission ranges of d_T and $0.5d_\mathrm{T}$, respectively. In both figures, a sensor node which is close to its ring's outer edge is tagged and the larger circle shows its EVM transmission range d_T. It can be seen that in Fig. 4.12a, which describes the original approach with the same transmission range for all messages, the tagged node has only a very limited number (one or two) of gray inner-ring neighbors in its transmission range. In Fig. 4.12b, the tagged node has an increased number (three or four) of inner-ring neighbors plus a further neighbor whose hop count is two less than the tagged node's. If the latter node receives the EVM, it should try to answer first, since one communication hop can be saved.

There are two main drawbacks of using this approach to obtain more information on the



(a) Transmission Range for SMSGs is d_T.

(b) Transmission Range for SMSGs is $0.5d_T$.

Figure 4.12: MASON simulation results with different transmission ranges for hop count update SMSGs (8× zoom into the center of the overall WSN area shown in Fig. 4.8).

nodes' location. First, the nodes need to support at least two different transmission powers which requires support by its hardware and contradicts the minimal hardware design principle chosen in this thesis. Second, the time until a hop count update process converges increases significantly. While a more detailed investigation is postponed to future work, first MASON simulations suggest that the convergence time is doubled when the transmission range of hop count update SMSGs is halved. Moreover, EVMs and NHFs can no longer be used easily for hop count updates, since their transmission ranges do not fit.

Estimating distance to sending node using RSS measurements. Provided the necessary hardware, a receiving node could determine the RSS of the EVM. Based on the RSS, the receiving node can estimate its distance to the sending node and this information could be used as an indicator for the receiving node's distance to the outer-edge of its ring. That is, the lower the received EVM's RRS, the sooner the receiving node responds to the message since it assumes that is farther away from the sending node and consequently from the outer edge of its ring. However, even if the RSS was a good measure for distances between sender and receiver (see discussion in Section 4.2.12), the distance might be misleading. This is illustrated in Fig. 4.13. Node 1 transmits the EVM and Node 2 receives it with a higher RSS than Node 3. Due to the lower RSS, Node 3 responds first but in the given scenario, Node 2 would have been the better choice, since Node 2 is farther from its ring's outer edge.

Keeping the rings narrow by limiting the number of hop count updates allowed per node. While observing the simulated hop count update visualized using MASON, it can be noticed that during the ring forming process the rings grow in width. This insight suggests that an alternative method for keeping the rings narrower can be implemented by limiting the number of local hop count updates per sensor node. In the following, let $N_{\text{hcu}}^{(\max)}$ denote the maximum number of hop count improvements that are allowed per sensor node.

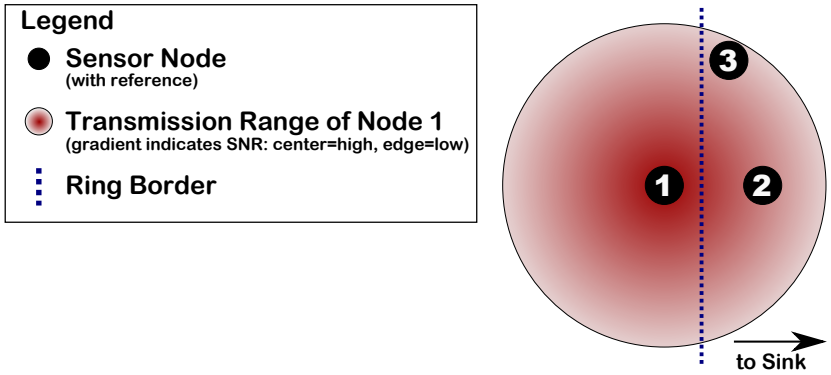


Figure 4.13: Failure of using the distance between sender and receiver for estimating the distance to the ring border.

As a first evaluation of this effect, the influence of $N_{\text{hcu}}^{(\max)}$ on the rings' form and width is investigated by MASON simulation. The simulation is carried out for eleven different values of $N_{\text{hcu}}^{(\max)}$ between 1 through 50. All other simulation parameters are chosen equivalently to the ones used for Fig. 4.8 in Section 4.3.2.2.* Note that for $N_{\text{hcu}}^{(\max)} > 50$ no further changes of the hop count rings are discovered by the simulation. Exemplary graphical simulation results are shown in Figs. 4.14 to 4.17 for $N_{\text{hcu}}^{(\max)} = 1, 10, 30$, and 50. The (a)-labeled sub-figures show the full simulated area. The (hardly visible) sink is again located close to the lower left of the area. The (b)-labeled sub-figures show the center of the simulated area in more detail.

In the (b)-labeled sub-figures, one node is highlighted by two circles. In the following, this node is referred to as *Center Node*. The smaller, dashed circle represents the *Center Node*'s sensing range. The larger circle corresponds to its transmission range. The *Center Node*'s simulated distance from the sink is 5.22 km.

In addition to the *Center Node*, a second node is investigated exemplarily in more detail in the following. This node is denoted as *Farthest Node*. It is the upper- and rightmost node within the simulated area. It has the farthest distance (11.12 km) to the sink.

For all investigated $N_{\text{hcu}}^{(\max)}$, the hop counts perceived by these two nodes are given in Table 4.3. In the following, Figs. 4.14 to 4.17 and Table 4.3 are discussed in more detail. Figure 4.14(a) illustrates that for $N_{\text{hcu}}^{(\max)} = 1$ (i.e., each node is allowed to update its hop count only once) no observable rings† are formed. Still, the whole simulated area is covered by colored nodes. This indicates that the network is connected and each node achieved the single hop count update. Also the more detailed Fig. 4.14(b) shows that no readily identifiable rings have formed in the vicinity of the *Center Node*.

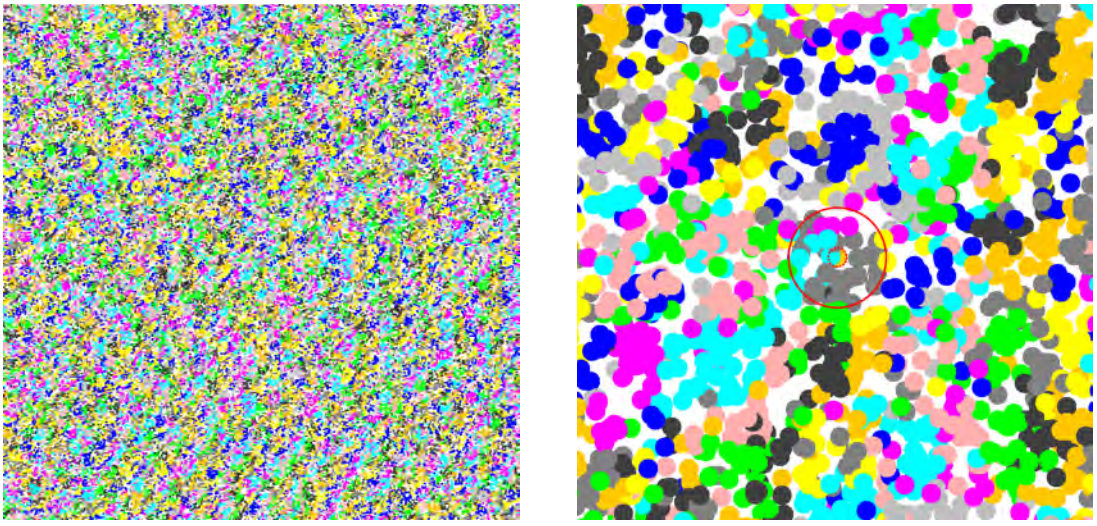
For $N_{\text{hcu}}^{(\max)} = 1$, the numerical MASON simulation results summarized in Table 4.3 show that the hop count perceived by the *Center Node* is 115. Knowing the *Center Node*'s distance from the sink, the mean ring width can be estimated by $\frac{5.22\,\text{km}}{115-0.5} \approx 45.6\,\text{m} \approx 0.456 d_{\text{T}}$. The mean ring width estimated by using the hop count 232 perceived by the *Farthest Node* can be calculated equivalently as approximately $0.480 d_{\text{T}}$. For both nodes, the estimated mean ring widths are also collected in Table 4.3.

* In particular, a square area of approximately $70\,\text{km}^2$ is simulated with $N_S = 100\,000$ uniformly distributed sensor nodes having a transmission range of $d_{\text{T}} = 100\,\text{m}$ each.

† The only ring that can be hardly imagined is the small Ring 1 which already formed in the sink's neighborhood at the bottom left.

Table 4.3: Mean ring widths for different $N_{\text{hcu}}^{(\max)}$

$N_{\text{hcu}}^{(\max)}$	Farthest Node		Center Node		Fig.
	Hop Count	Mean Ring Width*	Hop Count	Mean Ring Width*	
1	232	0.480	115	0.456	4.14
2	231	0.482	113	0.464	
3	224	0.497	112	0.468	
5	222	0.502	107	0.490	
7	219	0.509	105	0.500	
10	213	0.523	95	0.553	4.15
15	194	0.575	79	0.665	
20	181	0.616	64	0.822	
30	151	0.739	58	0.908	4.16
40	129	0.865	58	0.908	
50	122	0.915	58	0.908	4.17
∞	122	0.915	58	0.908	



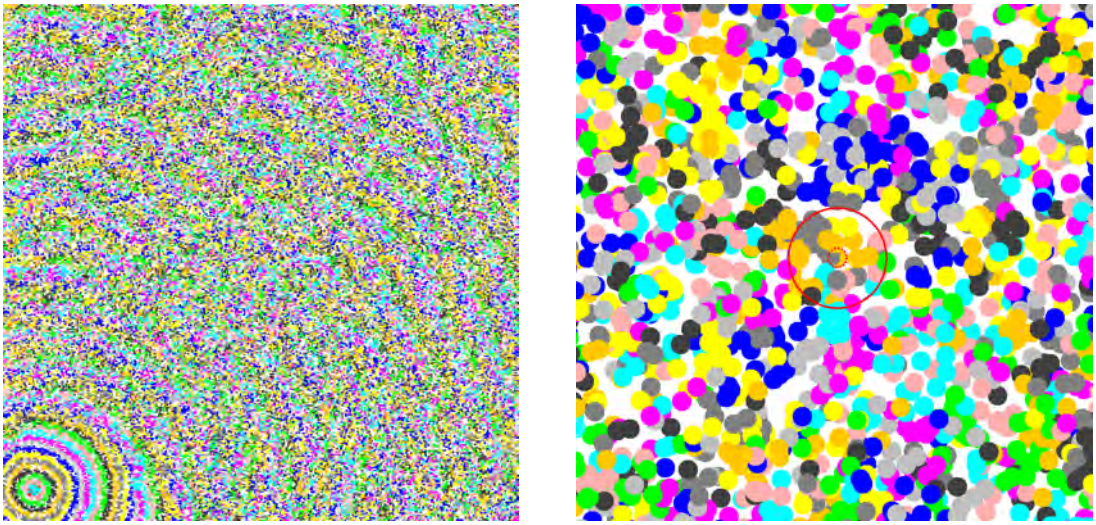
(a) Whole simulated area.

(b) 8× zoom into the center of the simulated area.

Figure 4.14: MASON simulation results for $N_{\text{hcu}}^{(\max)} = 1$.

For $N_{\text{hcu}}^{(\max)} = 10$, the graphical MASON simulation results are shown in Fig. 4.15. In Fig. 4.15(a), the first approximately 15 rings are now clearly visible. In the remainder of the simulation area, the rings are still very diffuse, but in contrast to Fig. 4.14(a), they seem to emerge. Figure 4.15(b) again does not hint at significant ring formation in the center of simulated area.

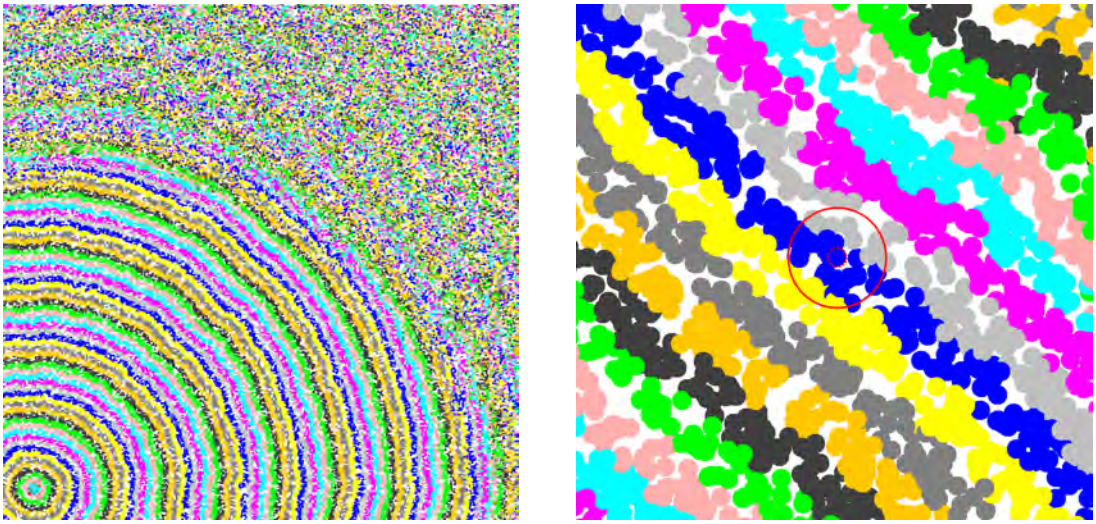
* Given in fractions of transmission range d_{T}.



(a) Whole simulated area.

(b) 8× zoom into the center of the simulated area.

Figure 4.15: MASON simulation results for $N_{\text{hcu}}^{(\max)} = 10$.



(a) Whole simulated area.

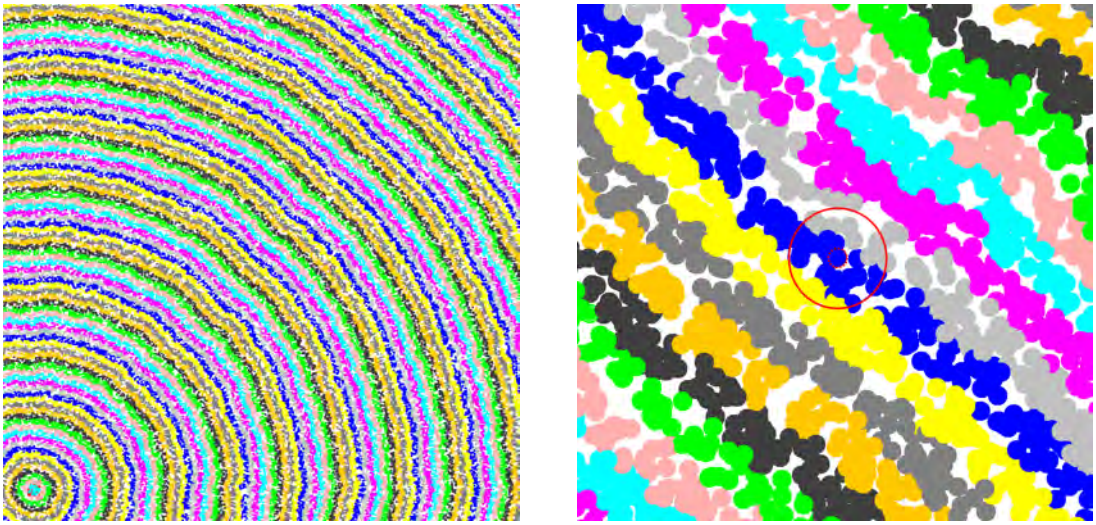
(b) 8× zoom into the center of the simulated area.

Figure 4.16: MASON simulation results for $N_{\text{hcu}}^{(\max)} = 30$.

As shown in Table 4.3, the hop counts of the *Farthest* and the *Center Node* decreased to 213 and 95, respectively, due to the increase of $N_{\text{hcu}}^{(\max)}$. The corresponding estimations of the mean ring width consequently increased to $0.523 d_{\text{T}}$ and $0.553 d_{\text{T}}$, respectively.

Figure 4.16 illustrates the resulting rings for $N_{\text{hcu}}^{(\max)} = 30$. It can be seen in Fig. 4.16(a) that distinct rings reach the center of the simulated area. This is confirmed by Fig. 4.16(b).

The hop counts perceived by the *Farthest* and the *Center Node* further decrease to 151 and 58, leading to a mean ring width estimate of approximately $0.739 d_{\text{T}}$ and $0.908 d_{\text{T}}$, respectively.



(a) Whole simulated area.

(b) 8× zoom into the center of the simulated area.

Figure 4.17: MASON simulation results for $N_{\mathrm{hcu}}^{(\max)} = 50$.

In can be seen from Table 4.3 that a further increase of $N_{\mathrm{hcu}}^{(\max)}$ beyond 30 does not lead to a further decrease of the *Center Node*'s hop count below 58. Consequently, the *Center Node* reaches is minimum hop count for some $N_{\mathrm{hcu}}^{(\max)}$ between 21 and 30. This indicates that for the same $N_{\mathrm{hcu}}^{(\max)}$ Rings 1 to 58 reach their maximal width of approximately $0.908 d_{\mathrm{T}}$ each in the mean. On the other hand, it can be seen in Fig. 4.17(a) that the formation for rings that are farther away from the sink is not completed.

According to Table 4.3, the *Farthest Node* reaches its minimal hop count for some $N_{\mathrm{hcu}}^{(\max)}$ between 41 and 50. For this $N_{\mathrm{hcu}}^{(\max)}$ and beyond, all rings in the simulated area reach their maximal width as shown in Figure 4.17. Based on the *Farthest Node*'s hop count, the mean ring width can be estimated as $0.915 d_{\mathrm{T}}$, which is close to the value $0.908 d_{\mathrm{T}}$ observed by the *Center Node*.

In summary, the exemplary simulation result provided in Figs 4.14 to 4.17 and Table 4.3 allow to draw the following preliminary conclusions which, however, should be affirmed by more detailed and more reliable analyses postponed to future work.

Reducing the maximum number $N_{\mathrm{hcu}}^{(\max)}$ of hop count updates per node allows to reduce the width of the rings. Hence, by limiting $N_{\mathrm{hcu}}^{(\max)}$, a finer granularity of hop count rings can be achieved.

However, the influence of $N_{\mathrm{hcu}}^{(\max)}$ on the ring width strongly depends on the ring's distance to the sink, since rings close to the sink converge much faster than rings far from the sink. Consequently, when applying this approach to decrease the rings' width, $N_{\mathrm{hcu}}^{(\max)}$ should be set in dependence of the nodes' preliminarily perceived hop count to achieve balanced ring widths and forms for any distance from the sink.

For finding such a dependency, the MASON model introduced in this thesis can be used to obtain the number of updates needed by a significant number of random nodes (located in

different rings) to achieve their final hop count. Applying, e.g., symbolic regression* on this dataset then allows to derive an approximate equation that enables calculation of the mean number of updates needed by the nodes that are members of a specific ring. The nodes could then use this equation and their current hop count estimate to deduce the number of updates they probably need to achieve the exact hop count. Limiting their own number of updates to the estimate minus a predefined reduction parameter would probably lead to narrow and balanced rings. However, the explicit derivation of the function and the reduction parameter as well as the investigation of the influences of node density, transmission range, and shape of the simulated area are postponed to future work.

Recording the ratio of messages received from inner-ring and outer-ring neighbors. As shown in Fig. 4.9, the node's location within its ring influences the overlap of its transmission range with adjacent rings and hence the number of outer-ring neighbors and the number of inner-ring neighbors. Vice versa, if a node was able to estimate the ratio between outer- and inner-ring neighbors, it could estimate its position in the ring.

Counting the number of neighbors is not a trivial task in the proposed WSN, since nodes lack unique IDs and the duty cycles are unsynchronized. However, a node could rather easily keep track of the number of messages (EVMs, NHFs, and SMSGs) it receives with higher and lower hop count and take the ratio of both numbers as an indicator for its location within its ring.

4.3.2.6 Discussion

The solutions presented in Section 4.3.2.5 show different ways how a node may get more fine-grained location information. This information can be used to prefer, as next hops, nodes with seemingly more inner-ring neighbors to nodes with less inner-ring neighbors.

The main advantage of the latter two methods is that they do not require additional hardware (as long as their moderate additional CPU cycles and memory usage can be considered negligible).

A more detailed, quantitative evaluation and comparison of the solutions' efficiency is not in the focus of this thesis and postponed to future work. This also holds for investigating the added value when combining two or more solutions.

The remainder of this thesis assumes that a suitable solution is identified and implemented. The sensor nodes are hence able to estimate their location within their ring and adapt their back-off delay accordingly. Assuming that nodes farther from their ring's outer edge are consequently able to attract EVMs to such extend that assuming uniform distribution of EVM-relaying nodes within a ring is fair, the estimations carried out in Section 4.3.2.3 also hold sufficiently for EVM-relaying nodes.

4.3.3 Back-Off Delay Calculation

Remember that a node joins the receiver contention for an incoming EVM only if it received the message correctly, its own hop count towards the next sink is less than the senders hop count specified in the EVM, and there is space in the inPool or the incoming EVM can be merged

* For example, by using the Eureqa tool which is also applied more extensively in Section 5.5.2.

with inPool EVMs. In this case, it puts the EVM into its inPool and starts the corresponding back-off timer. Otherwise, the EVM is ignored.

Let T_b denote the back-off delay, with $0 < T_b < T_b^{(\max)}$. The higher the node chooses its individual T_b on reception of an EVM, the more reluctant it is to become the responsible relay node.

The maximum back-off delay $T_b^{(\max)}$ is a system parameter that is defined at design time, but in principle could also be updated by SMSGs. According to Section 4.2.9, the aspired maximum detection-to-notification delay is 5 min, and according to Section 4.3.6.1, there are up to approximately 145 hops needed. Consequently, a communication hop should be passed in a maximum of $\frac{5\,\text{min}}{145} \approx 2\,\text{s}$. Hence, this value would be an upper bound of $T_b^{(\max)}$. A lower bound of $T_b^{(\max)}$ mainly depends on the granularity aspired for the back-off delays calculated by the nodes participating in receiver contention. Moreover, the back-off delays should not be dominated by the time needed to transmit the ACK*. Postponing a more detailed analysis of the optimum maximum back-off delay $T_b^{(\max)}$ to future work, this thesis roughly assumes that providing a maximum back-off delay of at least 100 ms should be foreseen.

For calculating its individual back-off delay T_b, a node can involve its knowledge on its own situation and information drawn from the EVM. For example, the more remaining energy a node has, the shorter it should choose T_b. And if a node is able to merge the received EVM with another EVM waiting in its outPool, it should also choose a short T_b. On the other hand, if the nodes' outPool is quite full, it should refrain from attracting further unmergeable EVMs by increasing the back-off delay to avoid congestion.

A possible calculation of a node's back-off delay T_b could be realized by

$$T_b = f_b \cdot T_b^{(\max)},$$

where f_b is called the back-off delay function taking values between zero and one. The function f_b depends on the node's remaining energy† $E_r \leq E^{(\max)}$, on the current number $N_{\text{out}} < N_{\text{out}}^{(\max)}$ of EVMs in the node's outPool, on the EVMs mergeability‡ reflected by indicator I_{mm} given by

$$I_{\text{mm}} = \begin{cases} 1 & \text{if incoming EVM can be merged with queueing EVM,} \\ 0 & \text{otherwise,} \end{cases}$$

and on the node's estimated normalized location $d_{\text{r}^-}^{\text{norm}}$ within its ring, where $0 \leq d_{\text{r}^-}^{\text{norm}} \leq 1$ is dimensionless and $d_{\text{r}^-}^{\text{norm}} = 0$ and $d_{\text{r}^-}^{\text{norm}} = 1$ refer to the cases when the node is located at the inner and outer edge of its ring, respectively.

Defining f_b to be the convex combination§

$$f_b = C_{b,E_r} \cdot \frac{E_r}{E^{(\max)}} + C_{b,N_q} \cdot \frac{N_q}{N_q^{(\max)}} + C_{b,I_{\text{mm}}} \cdot (1 - I_{\text{mm}}) + C_{b,d_{\text{r}^-}^{\text{norm}}} \cdot d_{\text{r}^-}^{\text{norm}},$$

the constants C_{b,E_r}, $C_{b,N_{\text{out}}}$, $C_{b,I_{\text{mm}}}$, and $C_{b,d_{\text{r}^-}^{\text{norm}}}$ can be used to control the influence of the remaining energy, outPool level, EVM mergeability, and location within the ring on f_b, respectively.

* That is, approximately $\frac{70\,\text{bit}}{100\,\text{kbps}} \approx 0.7\,\text{ms}$, according to Sections 4.2.9 and 4.3.6.

† Here, the remaining usable energy E_r and its (initial) maximum $E^{(\max)}$ are normalized such that a node stops working when $E_r = 0$, even if physically some unusable energy remains.

‡ EVM mergeability is discussed in Section 4.3.7.

§ That is, $C_{b,E_r} + C_{b,N_q} + C_{b,I_{\text{mm}}} + C_{b,d_{\text{r}^-}^{\text{norm}}} = 1$.

As mentioned in Section 4.1, further information can be included in the calculation of the back-off delay in principle, like reducing the delay for events that have higher priority or larger spatial coverage. A more detailed investigation of these options is postponed to future work.

4.3.4 Network Management

In this section, several aspects of the WSN management are discussed. These include the derivation on the nodes' location based on the hop count, the update of reporting thresholds, and coverage monitoring.

4.3.4.1 Hop Count-Based Localization

In the following, it is assumed that all sensor nodes obtained their distances to the sinks measured in the number of hops, i.e., their hop count coordinates $\mathbf{d}_\mathrm{h} = \langle d_\mathrm{h}^{(0)}, d_\mathrm{h}^{(1)}, d_\mathrm{h}^{(2)} \rangle$, using the procedure described in Section 4.3.2.2. In the proposed protocol, these distances constitute the nodes' *virtual location* within the sensor network's communication topology.

Additionally, the nodes' *physical locations* within the monitored area need to be known to deduce the location of the sensed events. In this thesis, a node's physical location is defined by its physical coordinate $\mathbf{d}_\mathrm{p} = \langle x_p, y_p \rangle$ in a two-dimensional Cartesian coordinate system, where x_p and y_p are the abscissa and the ordinate, respectively. The physical location of Sink i ($0 \leq i \leq 2$) is denoted by $\mathbf{d}_\mathrm{p}^{(i)} = \langle x_p^{(i)}, y_p^{(i)} \rangle$. For the sake of conciseness but without loss of generality, the physical coordinate system is chosen such that Sink 0 is located at the origin (i.e., $\mathbf{d}_\mathrm{p}^{(0)} = \langle 0, 0 \rangle_\mathrm{p}$), Sink 1 is located on the positive ordinate (i.e., $\mathbf{d}_\mathrm{p}^{(1)} = \langle 0, y_p^{(1)} \rangle$, with $y_p^{(1)} > 0$), and Sink 2 is located in the positive half-plane (i.e., $\mathbf{d}_\mathrm{p}^{(2)} = \langle x_p^{(2)}, y_p^{(2)} \rangle$, with $x_p^{(2)} > 0$ and $y_p^{(2)} \in \mathbb{R}$).

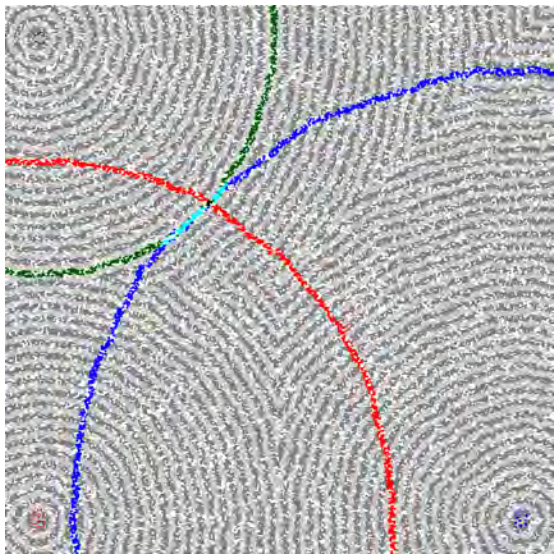


Figure 4.18: MASON simulation of monitored area with distances 60, 40, and 75 hops to Sinks 0 (lower left), 1 (upper left), and 2 (lower right), respectively, highlighted.

Knowing a node's hop count to a sink allows to reduce the node's possible physical locations to a certain ring around this sink (as illustrated in Fig. 4.8). If the hop counts to three sinks are combined, i.e., by performing hop count-based trilateration, the physical location of

the node can be estimated up to a small field*, as shown in Fig. 4.18. All black nodes that are located in the field where the three highlighted rings overlap have the same hop count coordinate $\mathbf{d}_\mathrm{h} = \langle 60, 40, 75 \rangle_\mathrm{h}$.

In the following, it is shown that each node is able to approximately calculate its own physical distance to each sink and its physical location within the monitored area as long as the sinks' physical locations and the mean width of the hop count rings around each sink are communicated to all nodes.

The mean width of the hop count rings around Sink i ($0 \leq i \leq 2$) is denoted by $\overline{d}_\mathrm{w}^{(i)}$ in the following. For each Sink i, the corresponding mean ring width $\overline{d}_\mathrm{w}^{(i)}$ can be obtained as follows. Since the sinks are deterministically deployed, their physical coordinates $\mathbf{d}_\mathrm{p}^{(i)} = \langle x_p^{(i)}, y_p^{(i)} \rangle$ are known. Hence, each sink can easily be made aware of the physical coordinates of all sinks and calculate the physical distances between them. After convergence of the hop count update, each sink also knows the number of hops that are needed by SMSGs from the other sinks to reach it. The external network can be used to share this information, and hence, all sinks are able to calculate for each Sink i the mean ring width $\overline{d}_\mathrm{w}^{(i)}$. Using SMSGs, the mean ring widths as well as the physical sink locations can then be communicated to the sensor nodes.

Based on this information, each sensor is then able to calculate its physical distance $d_\mathrm{s}^{(i)}$ to each Sink i ($0 \leq i \leq 2$) as follows. A node with hop distance $d_\mathrm{h}^{(i)}$ to Sink i is located somewhere within Ring $d_\mathrm{h}^{(i)}$ around Sink i. The physical distance of the node to Sink i can then be given by

$$d_\mathrm{s}^{(i)} = \left(d_\mathrm{h}^{(i)} - 1 \right) \overline{d}_\mathrm{w}^{(i)} + d_\mathrm{r^-}^{\mathrm{norm}}(i) \cdot \overline{d}_\mathrm{w}^{(i)} , \tag{4.5}$$

where $d_\mathrm{h}^{(i)}$ is the node's hop count to Sink i, $\overline{d}_\mathrm{w}^{(i)}$ is the mean width of the rings around Sink i, and $d_\mathrm{r^-}^{\mathrm{norm}}(i)$ (with $0 \leq d_\mathrm{r^-}^{\mathrm{norm}}(i) \leq 1$) is the node's estimated normalized location within its Sink-i ring as described in Section 4.3.3. Hence, the first summand of Eq. (4.5) reflects the sum of the estimated widths of Rings 1 through $d_\mathrm{h}^{(i)} - 1$, and hence, describes the estimated physical distance between Sink i and the inner-ring border of Ring $d_\mathrm{h}^{(i)}$. The second summand of Eq. (4.5) reflects the estimated physical distance of the node to the inner-ring border of Ring $d_\mathrm{h}^{(i)}$. Note that $d_\mathrm{r^-}^{\mathrm{norm}}(i)$ is approximated by 0.5 in case no better estimation of $d_\mathrm{r^-}^{\mathrm{norm}}(i)$ is available†.

Consequently, after a sensor node received the physical coordinates $\mathbf{d}_\mathrm{p}^{(i)}$ and the mean ring widths $\overline{d}_\mathrm{w}^{(i)}$ of each Sink i, it is able to calculate its distance to each sink using Eq. (4.5). The resulting coordinate is denoted as *physical sink distance coordinate* $\mathbf{d}_\mathrm{s} = \langle d_\mathrm{s}^{(0)}, d_\mathrm{s}^{(1)}, d_\mathrm{s}^{(2)} \rangle_\mathrm{s}$.

From $\mathbf{d}_\mathrm{s}$, the node can trilaterate its physical coordinate $\mathbf{d}_\mathrm{p} = \langle x_\mathrm{p}, y_\mathrm{p} \rangle$ via (see [109])

$$x_\mathrm{p} = \frac{1}{2C_3} \left(C_1 \left(y_\mathrm{p}^{(0)} - y_\mathrm{p}^{(2)} \right) - C_2 \left(y_\mathrm{p}^{(0)} - y_\mathrm{p}^{(1)} \right) \right) ,$$

* In case of knowing the hop distance to two sinks, there are, in general, two such small fields caused by the two intersections of the two rings. Without further information, a node cannot decide which of the two fields it is located in. However, in some situations, like the one shown in Fig. 4.8 for the highlighted rings formed by Sinks 1 and 2, two rings might be tangent, i.e., they intersect such that only a single but relatively large field results. Providing a third sink resolves these issues. Still, deploying only one or two sinks might be sufficient in special cases where ambiguities of the node locations are avoided by the field's form or by restricting the possible sink locations with respect to the sensor nodes.

† For example, by applying one of the approaches discussed in Section 4.3.2.5.

and

$$y_\mathrm{p} = \frac{1}{2C_3}\left(C_1\left(x_\mathrm{p}^{(0)} - x_\mathrm{p}^{(1)}\right) - C_2\left(x_\mathrm{p}^{(0)} - x_\mathrm{p}^{(2)}\right)\right),$$

where

$$C_1 = \left(d_\mathrm{s}^{(1)}\right)^2 - \left(d_\mathrm{s}^{(0)}\right)^2 - \left(x_\mathrm{p}^{(1)}\right)^2 + \left(x_\mathrm{p}^{(0)}\right)^2 - \left(y_\mathrm{p}^{(1)}\right)^2 + \left(y_\mathrm{p}^{(0)}\right)^2,$$

$$C_2 = \left(d_\mathrm{s}^{(2)}\right)^2 - \left(d_\mathrm{s}^{(0)}\right)^2 - \left(x_\mathrm{p}^{(2)}\right)^2 + \left(x_\mathrm{p}^{(0)}\right)^2 - \left(y_\mathrm{p}^{(2)}\right)^2 + \left(y_\mathrm{p}^{(0)}\right)^2,$$

and

$$C_3 = \left(x_\mathrm{p}^{(0)} - x_\mathrm{p}^{(1)}\right)\left(y_\mathrm{p}^{(0)} - y_\mathrm{p}^{(2)}\right) - \left(x_\mathrm{p}^{(0)} - x_\mathrm{p}^{(2)}\right)\left(y_\mathrm{p}^{(0)} - y_\mathrm{p}^{(1)}\right).$$

The accuracy of this hop count-based localization approach depends on various factors, in particular on the arrangement of the sinks. A detailed discussion of the accuracy lies not within the focus of this thesis but is related to the discussion of the size of hop count fields carried out in Section 4.3.5.2.

Note that, on first sight, nodes do not have to know their physical coordinate, since the hop count to the closest sink is sufficient for the hop count-based routing and the calculation of the event's physical localization could be done at the sink when the hop count coordinates are contained in the EVMs. However, the use of (three-dimensional) hop count coordinates has two major drawbacks. First, they are infeasible to describe neighborhood relations conveniently, and second, in the investigated scenario, communicating them needs more bits (and hence energy) than communicating coordinates based on the two-dimensional physical coordinates. Neighborhood relations and message lengths are discussed in more detail in Sections 4.3.5.1 and 4.3.6.1, respectively.

4.3.4.2 Reporting Threshold Update

For each type of events (like temperature value), a finite set of thresholds is defined (e.g., temperature exceeds 25 °C, 50 °C, ..., 150 °C). One of these thresholds is the (current) reporting threshold (say 50 °C), i.e., if a sensed value exceeds this reporting threshold, the event should be reported to the sink. As described in Section 4.3.1.2, the sensor node repeats the reporting regularly as long as the event exists. The sinks may switch off the repetitions for a known event, which also disburdens the sensor nodes and increases the network lifetime, by modifying the reporting thresholds using SMSGs. Such an increase of the reporting threshold could be seen as an end-to-end ACK sent by the sinks.

For being able to switch off the reporting for some kind of event completely, there should be a *maximum threshold* that is high enough to be never exceeded by actual sensor readings (e.g., a temperature that lies above the range of the thermometer). If this maximum threshold is announced as reporting threshold by the sinks, the sensor nodes stop reporting the corresponding event.

Note that thresholds can be set differently for different regions within the monitored area.

4.3.4.3 Coverage Monitoring

Also coverage monitoring can be achieved by updating the reporting thresholds. For this, a very low threshold needs to be provided (e.g., temperature exceeds −50 °C). When this *minimum threshold* is enabled by the sinks, all sensor nodes start so send EVMs. By evaluating the

messages, the sinks can assess the coverage. Again, this process can (and should) be initiated separately for restricted regions to avoid excessive congestion.

4.3.5 Neighborhood, Grid, and Location Ranges

Before discussing the message format and the merging of messages in detail in Sections 4.3.6 and 4.3.7, respectively, neighborhood relations within the monitored area are defined. On the one hand, these neighborhood relations are needed to identify spatial correlation of event data that can be exploited for merging EVMs. On the other hand, the addressing scheme chosen for event locations and their neighborhoods has a significant effect on the messages' format and data length.

4.3.5.1 Hop Count-Based Field Neighborhood

Neighborhood relations based on hop count-based localization are discussed first. It is shown that these neighborhood relations cannot be used conveniently for message merging.

The trilateration based on hop counts, introduced in Section 4.3.4.1, provides physical location information to the sensor nodes. The trilateration is illustrated schematically in Fig. 4.19, where concentric circles mark the edges of the hop count rings around three sinks. That is, within each of the white fields in between the circles, the nodes share the same hop count coordinate $\mathbf{d}_\mathrm{h}$. The figure shows that the overlapping rings form Moiré-like patterns (cf. [14]). Due to this effect, the field sizes vary significantly. They depend on various factors including their physical location, the rings' widths, and also on the physical location of the sinks—in particular relatively to each other.

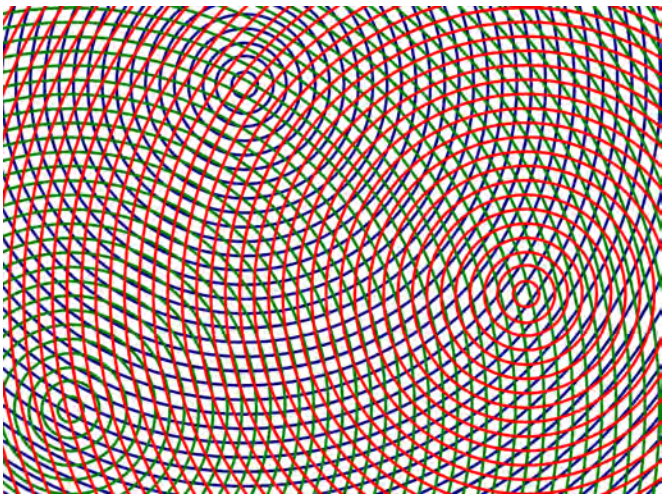


Figure 4.19: Moiré effect caused by overlapping concentric rings around three sinks.

Obviously, the situation would be worst if all sinks shared the same position or if there was only a single sink. Then, all nodes located within each ring would share the same $\mathbf{d}_\mathrm{h}$. The situation significantly improves when the three sinks are located at the outer border of the monitored area, ideally forming the largest possible equilateral triangle. Usually, the fields are then smaller than the squared ring widths. As exemplarily shown soon, the minimum field

size is zero. Related work with more detailed discussion of the accuracy of hop count-based localization is given in Section 4.2.12.

The variance of field sizes on the one hand influences the accuracy of determining the physical node locations. On the other hand, it also makes it harder to identify neighborhood relations of fields. In particular, fields might be too small to host any sensors. Figure 4.20 shows a detail of Fig. 4.19 and allows to investigate the field neighborhoods obtained by hop count trilateration in more detail.

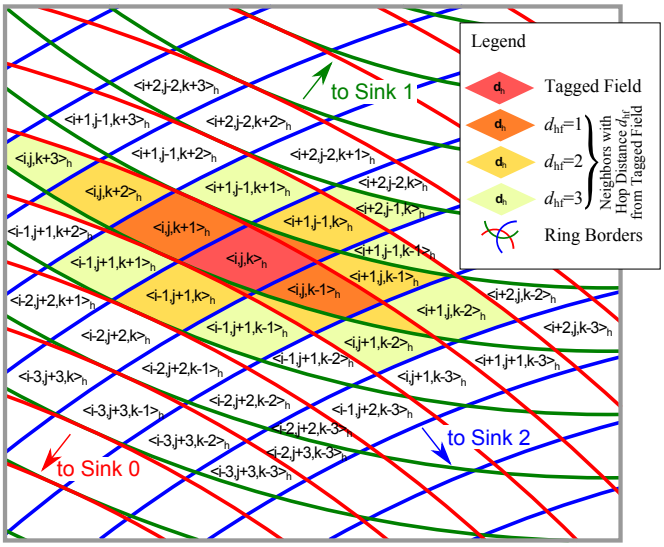


Figure 4.20: Field neighborhoods and hop distance.

Given two arbitrary fields denoted by m and n, which have hop coordinates $\mathbf{d}_\mathrm{h}(m) = \langle d_\mathrm{h}^{(0)}(m), d_\mathrm{h}^{(1)}(m), d_\mathrm{h}^{(2)}(m)\rangle$ and $\mathbf{d}_\mathrm{h}(n) = \langle d_\mathrm{h}^{(0)}(n), d_\mathrm{h}^{(1)}(n), d_\mathrm{h}^{(2)}(n)\rangle$, respectively, the *hop distance* $d_\mathrm{hf}(m,n)$ can be defined by the 1-norm (taxicab norm)

$$d_\mathrm{hf}(m,n) = |d_\mathrm{h}^{(0)}(m) - d_\mathrm{h}^{(0)}(n)| + |d_\mathrm{h}^{(1)}(m) - d_\mathrm{h}^{(1)}(n)| + |d_\mathrm{h}^{(2)}(m) - d_\mathrm{h}^{(2)}(n)|\,,$$

which gives the (minimum) number of hops across ring borders to reach field m from field n, or vice-versa.

In Fig. 4.20, it can be seen that the tagged field with hop coordinate $\langle i,j,k\rangle_\mathrm{h}$ has the four 1-hop field neighbors $\langle i,j,k+1\rangle_\mathrm{h}$ to its upper left, $\langle i,j,k-1\rangle_\mathrm{h}$ to its lower right, $\langle i+1,j,k\rangle_\mathrm{h}$ to its upper right (unlabeled due to its small size), and $\langle i,j+1,k\rangle_\mathrm{h}$ to its lower left (also unlabeled). Some theoretical 1-hop neighbors of the tagged field (like $\langle i-1,j,k\rangle_\mathrm{h}$) do not exist* and very small neighbor fields (e.g., like the unlabeled $\langle i,j+1,k\rangle_\mathrm{h}$) might not hold any sensor node. Hence, the tagged field should consider, e.g., field $\langle i-1,j+1,k\rangle_\mathrm{h}$ as one of its direct neighbors instead, although its hop distance is two. Such decisions, however, cannot be easily done by the nodes, especially if they are not even part of the corresponding neighborhood. Still, all nodes should be able to merge messages originating from neighboring regions.

As an alternative to the 1-norm, the *field distance* $d_\mathrm{f}(m,n)$ of the two fields m and n can be defined by applying the 2-norm (Euclidean distance) as follows:

$$d_\mathrm{f}(m,n) = \sqrt{|d_\mathrm{h}^{(0)}(m) - d_\mathrm{h}^{(0)}(n)|^2 + |d_\mathrm{h}^{(1)}(m) - d_\mathrm{h}^{(1)}(n)|^2 + |d_\mathrm{h}^{(2)}(m) - d_\mathrm{h}^{(2)}(n)|^2}\,.$$

* Hence, no address space should actually be reserved for them.

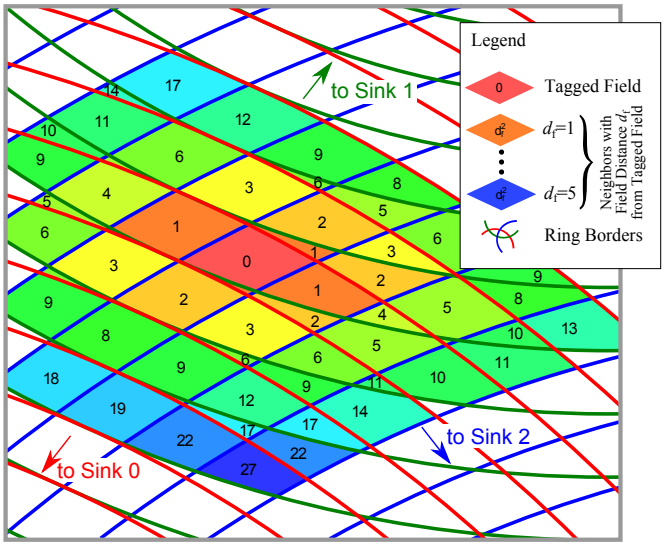


Figure 4.21: Euclidean field distances.

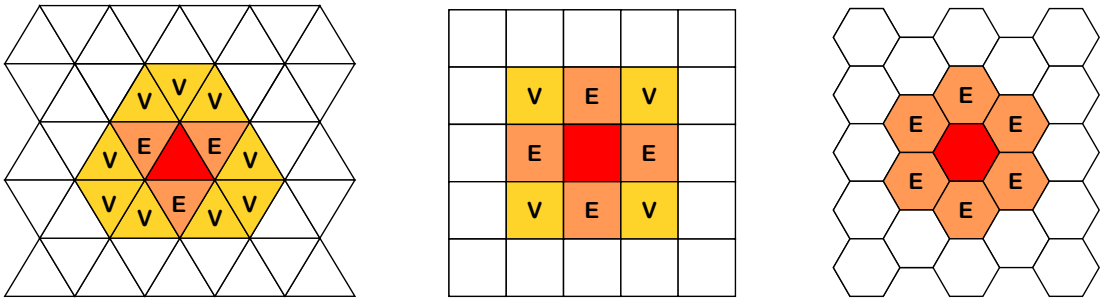
For the same scenario as used in Fig. 4.20, Fig. 4.21 shows the squared field distance (d_f^2) of each cell to the tagged cell (which has $d_f = 0$). The field distance is also reflected in the color of the cells. It can be seen from the figure that the field distance also does not reflect well the physical distance of fields. For example, consider the field with $d_f^2 = 27$ to the south of the tagged node. Its centroid's physical distance to the tagged field's centroid is approximately $3.4\overline{d}_w$, where $\overline{d}_w$ is the mean ring width. This conflicts with the situation of the field with $d_f^2 = 9$ to the east of the tagged node (just below the legend), which has a physical distance of $4.3\overline{d}_w$.

Consequently, the irregular fields built by the hop count rings seem unsuitable for directly identifying (and addressing) neighboring regions within the monitored area.

4.3.5.2 Regular Hexagonal Grid

To circumvent the problem of identifying neighboring fields, an additional, regular grid of hexagonal cells is introduced in the following. Based on their estimated physical location, the nodes are able to determine to which cell they belong and the neighborhood relations of these regular cells can then be identified and described much more conveniently when compared to the field neighborhoods.

Alternatives of regular grids. Regular tessellations of the two-dimensional plane are gap- and overlap-free tilings that are based on identical shapes (monohedral tessellations) while these shapes are regular (i.e., equiangular and equilateral) polygons. Such tessellations can only be achieved with equilateral triangles, squares, or regular hexagons (see [275, Sec. 7.4]). These three types of tessellations are illustrated in Fig. 4.22. In the following, the (triangular, square, or hexagonal) tiles induced by the tessellations are called *cells*. The neighbor cells of a tagged cell can be assigned to two classes: neighbors that are connected with the tagged cell by an edge (labeled with "E" in Fig. 4.22) and neighbors that only share a vertex ("V") with the tagged node.



(a) Equilateral triangles. **(b)** Squares. **(c)** Regular hexagons.

Figure 4.22: Regular tessellations of the two-dimensional plane.

The main advantage of the tiling based on regular hexagons (shown in Fig. 4.22c) is that each neighbor cell shares an edge with the tagged cell and the centers of all neighbors have the same distance to the tagged cell's center. Therefore, hexagonal tiling is used in the following as a basis for the regular grid. Without loss of generality, this thesis focuses on the pointy-side variant of the hexagonal tessellation, as depicted in Fig. 4.22c. Mutatis mutandis, the flat-sides variant* would lead to the same results.

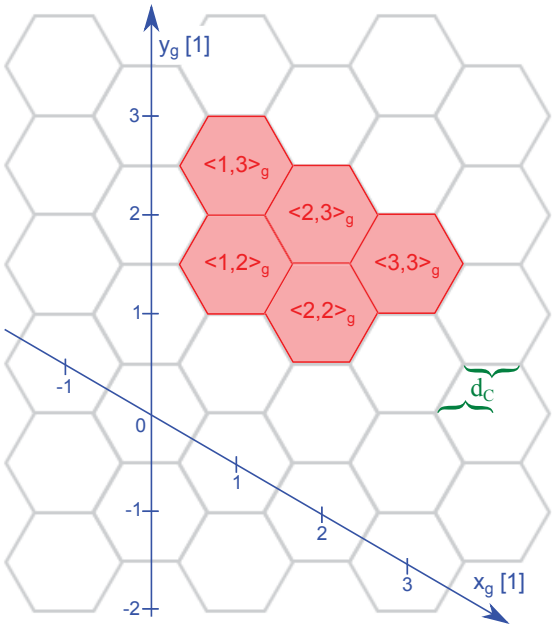


Figure 4.23: Hexagonal grid and coordinate system.

Grid coordinates. For addressing the individual cells of the hexagonal grid, a suitable coordinate system is needed. The chosen one is illustrated in Fig. 4.23. The corresponding coordinates are referred to as *grid coordinates* in the following and denoted by $\mathbf{d}_g = \langle x_g, y_g \rangle$. Similar to the physical coordinate system used in Section 4.3.4.1, the sinks' grid coordinates

* The flat-sides variant has pointy tops and bottoms, i.e., constitutes the pointy-side variant rotated by 30°.

are chosen as $\mathbf{d}_{\mathrm{g}}^{(0)} = \langle 0,0 \rangle_{\mathrm{g}}$, $\mathbf{d}_{\mathrm{g}}^{(1)} = \langle 0, y_g^{(1)} \rangle$ with $y_g^{(1)} > 0$, and $\mathbf{d}_{\mathrm{g}}^{(2)} = \langle x_g^{(2)}, y_g^{(2)} \rangle$ with $x_g^{(2)} > 0$ and $y_g^{(2)} \in \mathbb{R}$.

An important parameter of the grid is the physical cell radius d_{C} (in meters), which is equivalent to the cell's side length (see Fig. 4.23). It determines the size of the cells and consequently can be used to trade off the spatial event reporting resolution against the number of bits needed to represent the cells' coordinates and consequently the energy needed to transmit them.

In the present application scenario of early forest fire detection, an event localization granularity of 100 m seems sufficient, i.e., each hexagonal cell of the virtual localization grid has a radius of $d_{\mathrm{C}} = 100\,\mathrm{m}$. With this radius, fire fighters that are directed to the center of such a cell should be able to locate a starting fire located within the cell after a sufficiently limited time. The resulting cell size is $A_{\mathrm{C}} = \frac{3\sqrt{3}}{2} d_{\mathrm{C}}^2 \approx 2.6 \cdot 10^4\,\mathrm{m}^2 \approx 0.026\,\mathrm{km}^2$. Consequently, the overall number of cells that can be differentiated and localized within the area of interest can be estimated by $N_{\mathrm{C}} \approx \frac{A_{\mathrm{F}}}{A_{\mathrm{C}}} \approx \frac{70\,\mathrm{km}^2}{0.026\,\mathrm{km}^2} \approx 2700$.

Coordinate transformation. A node can estimate its grid coordinate $\mathbf{d}_{\mathrm{g}} = \langle x_{\mathrm{g}}, y_{\mathrm{g}} \rangle$ based on its estimated physical coordinate $\mathbf{d}_{\mathrm{p}} = \langle x_{\mathrm{p}}, y_{\mathrm{p}} \rangle$ for a specific grid cell radius d_{C} via the following transformation*:

$$x_g = \begin{cases} x_{\mathrm{g}}^{(\mathrm{pre})}, & \text{if } -\sqrt{3}\left(x_{\mathrm{p}}^{(\mathrm{norm})} + d_{\mathrm{C}}\right) \leq y_{\mathrm{p}}^{(\mathrm{norm})} \leq \sqrt{3}\left(x_{\mathrm{p}}^{(\mathrm{norm})} + d_{\mathrm{C}}\right), \\ x_{\mathrm{g}}^{(\mathrm{pre})} - 1, & \text{otherwise}, \end{cases} \tag{4.6}$$

and

$$y_g = \begin{cases} y_{\mathrm{g}}^{(\mathrm{pre})} - 1, & \text{if } y_{\mathrm{p}}^{(\mathrm{norm})} < -\sqrt{3}\left(x_{\mathrm{p}}^{(\mathrm{norm})} + d_{\mathrm{C}}\right), \\ y_{\mathrm{g}}^{(\mathrm{pre})}, & \text{otherwise}, \end{cases} \tag{4.7}$$

where

$$x_{\mathrm{g}}^{(\mathrm{norm})} = x_p - \left(1.5 d_{\mathrm{C}} x_{\mathrm{g}}^{(\mathrm{pre})}\right), \quad y_{\mathrm{g}}^{(\mathrm{norm})} = y_p - \left(0.5\sqrt{3} d_{\mathrm{C}} (2 y_{\mathrm{g}}^{(\mathrm{pre})} - x_{\mathrm{g}}^{(\mathrm{pre})})\right),$$

$$x_{\mathrm{g}}^{(\mathrm{pre})} = \left\lfloor \frac{x_p + d_{\mathrm{C}}}{1.5 d_{\mathrm{C}}} \right\rfloor, \quad y_{\mathrm{g}}^{(\mathrm{pre})} = \left\lfloor \frac{y_p}{\sqrt{3} d_{\mathrm{C}}} + \frac{x_{\mathrm{g}}^{(\mathrm{pre})} + 1}{2} \right\rfloor.$$

Data length of grid coordinates. To determine the number of bits needed to describe (and transmit) a grid coordinate, first note that for an optimal addressing of $N_{\mathrm{C}} \approx 2700$ cells, $\lceil \mathrm{ld}(N_{\mathrm{C}}) \rceil = 12\,\mathrm{bit}$ are needed in principle. However, the sensor nodes are not aware of all cells (and their coordinates) within the WSN. Therefore, they are not able to do an optimal distribution of the address space across all cells. Instead, enough address space must be provided to map both grid coordinates (x_g and y_g) up to the maximum diameter $d_p^{\max} \approx 13\,\mathrm{km}$ of the monitored area. To determine the number of grid cells needed to span the maximum diameter, it should be noted that the distance between two cells' centers in physical y_p direction is $\sqrt{3} d_{\mathrm{C}} \approx 1.73 d_{\mathrm{C}}$, while in physical x_p direction, it is $1.5 d_{\mathrm{C}}$.† Hence, in the estimated

* The derivation of this transformation is an adaption of the idea presented in [133] and provided in Appendix Section B.4.

† This can be easily seen from Fig. B.11 given in the Appendix, page 183.

Table 4.4: Grid-based location ranges

Location Range Type	Description	Range Parameter (length/bit)		Figure(s)
1	single cell	N/A	(0)	4.24(a)
2	cell-centered disc	radius	(7)	4.24(b), (c)
3	left vertex-centered disc	radius	(7)	4.24(d)
4	right vertex-centered disc	radius	(7)	4.24(e), (f)
5	rhomboid	2nd reference cell	(16)	4.24(g)
6	rectangle	2nd reference cell	(16)	4.24(h)

worst case, $N_{\mathrm{g}}^{(\max)} \approx \left\lceil \frac{d_p^{\max}}{1.5 d_{\mathrm{C}}} \right\rceil \approx 87$ grid cells are needed to span $d_p^{\max}$. For both grid dimensions (x_g and y_g), an extra bit allows for negative coordinates. Therefore, $\left\lceil \mathrm{ld}\left(N_{\mathrm{g}}^{(\max)}\right) \right\rceil + 1 = 8\,\mathrm{bit}$ need to be reserved for each dimension in the given scenario.* Hence, 16 bit are reserved in messages for grid coordinates $\langle x, y \rangle_{\mathrm{g}}$.

For comparison, note that the transmission of a hop coordinate $\mathbf{d}_{\mathrm{h}}$ would require $3 \cdot 8\,\mathrm{bit} = 24\,\mathrm{bit}$.

4.3.5.3 Grid-Based Location Ranges

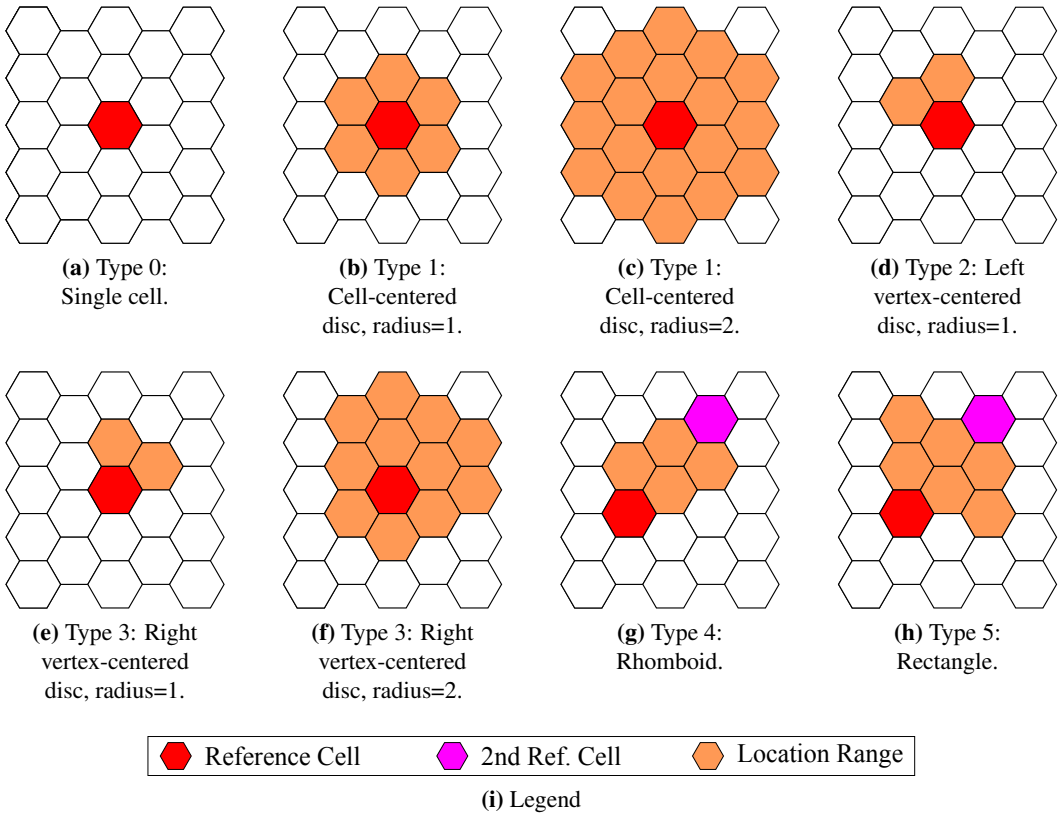
For merging messages based on the neighborhood relations of events, a compact addressing scheme for location ranges based on the hexagonal grid is also required. Such a scheme is proposed in Table 4.4 which refers to Fig. 4.24 for graphical illustration.

Following this addressing scheme, a location range can be identified by three values: the type of the location range (3 bit needed), the grid coordinates of the reference cell (16 bit), and a range parameter that is depending on the type. If the location range is a single cell (Type 1), the range parameter is empty (0 bit). If the location range takes the form of a disc, the range parameter denotes the radius measured in grid cells. For radii, 7 bit are reserved, since a radius cannot exceed the monitored area's maximum diameter $d_p^{\max}$. In case of rhomboid or rectangle ranges, the second parameter is the coordinate of a second reference cell (16 bit).

4.3.6 Message Format

Remember that there are two main types of messages: *sink messages* (SMSGs) originating from a sink and messages initiated by some wireless sensor node. The latter are either *event messages* (EVMs) or *next hop found* (NHF) messages. All messages share the same basic format illustrated in Fig. 4.25.

* Actually, if the locations of the sinks are known in advance (and they are not modified later), estimating the positions of the grid coordinate system with respect to the shape of the monitored area is possible. This would allow to optimize the number of bits needed for each dimension. Still, getting better than $\lceil \mathrm{ld}(N_{\mathrm{C}}) \rceil$ is impossible.



(a) Type 0: Single cell.
(b) Type 1: Cell-centered disc, radius=1.
(c) Type 1: Cell-centered disc, radius=2.
(d) Type 2: Left vertex-centered disc, radius=1.
(e) Type 3: Right vertex-centered disc, radius=1.
(f) Type 3: Right vertex-centered disc, radius=2.
(g) Type 4: Rhomboid.
(h) Type 5: Rectangle.
(i) Legend

Figure 4.24: Location ranges in the hexagonal grid.

Figure 4.25: Basic XLMMP message format (length: 5–88 bit).

The first bit of each message declares the message type, i.e., whether the message is an SMSG (MSG_TYPE=1) or not (MSG_TYPE=0). Then several DATA bits follow that contain message type-dependent data as explained in Sections 4.3.6.1 and 4.3.6.2. Each message ends with at least a parity bit that allows to detect transmission errors (ERR_DET). The extension of the parity bit to more reliable (but more costly) error detection methods (like checksum or cyclic redundancy check (CRC; cf. [165, p. 428]) is straightforward in principle. However, finding the optimum trade-off between reliable error detection (or even forward error control) and its corresponding overhead is deferred to future work.

It should be noted that while messages of different type and content may have different length, a dedicated field that carries explicitly the actual message length is not foreseen to save the corresponding bits. However, the message length can relatively easily be determined by evaluating a small set of selected message fields (MSG_TYPE, LOC_TYPE, EVENT_IDS, and COMMAND; described in more detail in the sequel).

The cross-layer nature of XLMMP suggests to meld the terms *message*, *segment*, *packet* (or *datagram*), and *frame*, which are used in the classical layered protocol stack on application,

transport, network, and link layer, respectively (cf. [304, p. 339], [165, p. 50–51]). In particular, the size of all messages is small compared to the message sizes of classical wired and wireless networks. Hence, a fragmentation of messages is not foreseen. Moreover, there is only a single application and the routing scheme is determined by the message type and based on hop counts only. Additionally, due to the small message size, using one frame per message is possible and an extra frame control sequence (FCS) is unnecessary due to the already foreseen ERR_DET field. Furthermore, XLMMP does not need explicit link-layer addresses (beyond the hop count) for forwarding messages. In the following, the term *message* is used to refer to XLMMP's cross-layer data unit.

While this thesis does not investigate link and physical layer issues in much more detail, note that on link and physical layer, frames are usually provided with additional bit sequences, e.g., for enabling receivers to detect the beginning and the end of a frame and for synchronizing the receiver's clock with the sender's clock.

For detecting the start end end of a frame, one of several methods or combinations can be chosen. According to [304, p. 179], these include *character count*, *starting and ending characters* (with character stuffing*), *starting and ending flags* (with bit stuffing), and *physical layer coding violations*†.

For example, Ethernet (IEEE 802.3, [128, p. 126]) uses a flag of one octet (8 bit) length called *start frame delimiter* (SFD) to mark the start of a frame. The frame's end is marked by an *end of transmission delimiter* (idle signal (IDL) for the duration of 2 bit) that turns off the transmitter. In WLANs (IEEE 802.11, [126, p. 152, p. 196], [129, p. 1588]) with frequency-hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS), the SFD has a length of 16 bit and is accompanied by an additional field (called *PSDU*‡ *length word* (PLW)) of 12 bit (FHSS) or 16 bit (DSSS) length that specifies the number of octets contained in the data unit. A similar approach is chosen for LR-WPAN (IEEE 802.15.4, [127, p. 43]), where the SFD's length is one octet and the size of the frame length field is 7 bit.

On physical layer, usually a *preamble SYNC field* is prefixed to the frame and facilitates, e.g., the clock synchronization. Typical lengths of preamble SYNC fields lie in the order of 56 bit (Ethernet, IEEE 802.3, [128, p. 126]; LR-WPAN, IEEE 802.15.4, [127, p. 43]), 72 bit (Bluetooth Version 4.0, [43, Vol. 2, p. 110]), 80 bit (WLAN, IEEE 802.11, with FHSS, [126, p. 152]), or 128 bit (WLAN, IEEE 802.11, with DSSS, [126, p. 196]).

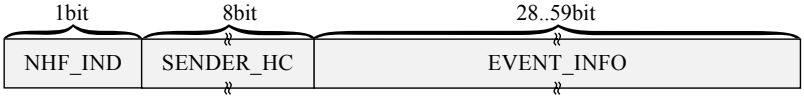
In XLMMP, the overhead induced by these link-layer and physical-layer measures in relation to the small payload can get relatively high. While a more rigorous investigation needs to be done in future work after selecting a suitable communication protocol on physical layer, the estimated effect on the overhead is accommodated in a defensively chosen, low data rate on message level in Section 4.2.14.

For each type of messages, i.e., EVM/NHFs and SMSGs, the format of the DATA field is explained in more detail in the following.

* *Stuffing* refers to a technique where the sender adds additional masking characters or bits into the payload if it contains sequences identical to the flags used by the link or physical layers. The stuffed characters or bits are removed at the receiver side before passing the payload to the higher layers.

† This method is only applicable if the data encoding on the physical layer provides additional redundancy.

‡ Physical layer convergence protocol service data unit.



(a) Format of DATA field (length: 37–68 bit).

(b) Format of EVENT_INFO field (length: 28–59 bit).

Figure 4.26: DATA field format of EVMs and NHFs.

4.3.6.1 DATA Field of EVMs and NHFs

The DATA field's format of EVMs and NHFs is shown in Fig. 4.26(a). It starts with the single-bit field (NHF_IND) that indicates whether the message is an NHF (NHF_IND=1) or an EVM (NHF_IND=0).

Then, the sender's hop count (SENDER_HC) distance to its closest sink is provided, which is needed for the hop count-based routing of messages within the WSN. The distance is given by $\min\left(d_{\mathrm{h}}^{(0)}, d_{\mathrm{h}}^{(1)}, d_{\mathrm{h}}^{(2)}\right)$. The necessary length of field SENDER_HC can be derived as follows. As discussed in Section 4.2.2, the monitored area's maximum diameter is approximately $d_p^{\max} = 13\,\mathrm{km}$. From the last row of Table 4.3, a rough estimate of the mean ring width $\overline{d}_{\mathrm{w}} \approx 0.9\,d_{\mathrm{T}}$ can be derived. Consequently, the expected maximum number of hops between any sink and sensor node is $N_{\mathrm{h}}^{(\max)} \approx \left\lceil \frac{d_p^{\max}}{\overline{d}_{\mathrm{w}}} \right\rceil \approx \left\lceil \frac{13\,\mathrm{km}}{90\,\mathrm{m}} \right\rceil \approx 145$. For transmitting this integer, 8 bit are needed. Since the maximum integer representable by 8 bit is 255, $N_{\mathrm{h}}^{(\mathrm{upper})} = 255$ is a suitable upper bound for the maximum number of communication hops in the given scenario.

The remaining bits of an EVM's and NHF's DATA field (EVENT_INFO) are used for the description of the communicated event(s). The format of the EVENT_INFO field is shown in Fig. 4.26(b).

The first three fields of EVENT_INFO describe the corresponding location range based on its type (LOC_TYPE, 3 bit), the reference cell's coordinate (LOC_ADDR, 16 bit), and the location range parameter (LOC_PAR, 0–16 bit) as defined in Section 4.3.5.3.

The last two fields of EVENT_INFO bear the information on the events. Field EVENT_ID indicates the types of the events that are reported. For example, each sensor node may be able to sense three different environmental properties (e.g., temperature, humidity, and pressure; cp. Table 2.1). For each property, the node may monitor, e.g., the absolute value and a gradient. This results in six different *event types*. For each event type, a bit is reserved in field EVENT_ID. If the bit is set to one, a corresponding event threshold is reported in field EVENTS. Consequently, EVENTS contains a set of one to six sub-fields in the order of reported events types as specified by field EVENT_ID. Each provided subfield contains an identification of the applicable threshold. This thesis assumes that six proper reporting thresholds are sufficient for each event type. Together with the maximum threshold for disabling reporting (see Section 4.3.4.2) and the minimum threshold for coverage monitoring (see Section 4.3.4.3), the WSN needs to support eight thresholds, and hence, 3 bit are needed for each subfield.

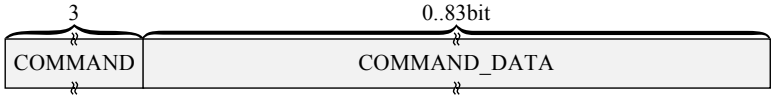


Figure 4.27: SMSG format (length: 3–86 bit).

Table 4.5: SMSG commands

COMMAND	Description	COMMAND_DATA (length/bit)	
000	hop count reset	N/A	(0)
001	hop count update	$\mathbf{d}_\mathrm{h}$	(24)
010	reporting threshold update	EVENT_INFO	(28–59)
011	combined hop count and threshold update	$\mathbf{d}_\mathrm{h}$, EVENT_INFO	(52–83)
100	physical sink location update and hop count reset	sink ID, $\mathbf{d}_\mathrm{p}^{\text{sink ID}}$	(30)
101	mean ring widths update	$\overline{d}_\mathrm{w}^{(0)}, \overline{d}_\mathrm{w}^{(1)}, \overline{d}_\mathrm{w}^{(2)}$	(24)

4.3.6.2 DATA Field of SMSGs

As shown in Fig. 4.27, the DATA field of SMSGs has the two main fields COMMAND and COMMAND_DATA. Field COMMAND has a length of 3 bit, capable of referring to up to eight different commands. Five command examples are shown in Table 4.5. They provide means for updating the hop counts, reporting thresholds, and physical sink locations in case the sinks need to be moved. Additional commands that could be discussed in future work may include a *network hibernation command* that allows to reduce the sensing frequency (or to switch sensing off completely) and increase the nodes' sleep periods.

The format and length of field COMMAND_DATA is command specific, as shown in Table 4.5. The minimum length is 0 bit, the maximum length is 83 bit.

4.3.7 Message Merging

If two EVMs meet at some sensor node on their way to the sink, merging them to a single message would benefit energy efficiency by cutting the number of subsequent transmissions by half. However, to keep the message size of the resulting EVM small, the merging should go well beyond a mere concatenation of the two original EVM. Hence, the merging should always exploit correlations between the two EVMs.

Events can be correlated in three dimensions: (1) type of the event, (2) event location, and (3) event time. As discussed in Section 4.2.9, the age of an EVM should stay below about 5 min. Hence, the event times of two meeting EVMs always correlate and do not pose any additional constraint on their mergeability, in particular since more detailed event time information is not included explicitly in the EVMs as discussed in Section 4.2.16.

Readings of different sensors hosted by a single sensor node and close-by sensor nodes are

Table 4.6: Possible cases when merging two EVMs

Location Range	Sensor Data	
	Identical	Different
Identical	*Case A*	*Case B*
Mergeable	*Case C*	*Case D*

likely correlated* since they measure the same environment. Hence, providing the possibility to include more than one event type into a single EVM promises to significantly reduce the number of messages by the expense of only a few additional bits per message, as shown in Section 4.3.6.1. There, it is also demonstrated that encoding the event's location requires a good portion of an EVM's data length, i.e., communicating a location or location range requires more bits than communicating an event's type and effective threshold. Hence, merging spatially correlated EVMs, i.e., EVMs that apply to identical or similar (i.e., neighboring) locations, while providing a compact representation of the corresponding resulting location range (as proposed in Section 4.3.5.3) bears a high potential for reducing the number and length of EVMs and consequently increases energy efficiency.

Therefore, the current version of the XLMMP draft is designed such that EVMs are able to communicate several events for a single location or location range, but for separate, non-neighboring locations, separate EVMs are needed. When two EVMs meet at one node, a precondition for their mergeability thus is that their corresponding locations or location ranges are identical or can be merged. Two or more locations can be merged when they can be jointly represented† by one of the location ranges introduced in Section 4.3.5.3.

If so, there are still four different cases—defined in Table 4.6—that need to be treated differently as described in the following.

Case A. The two meeting EVMs are identical. Hence, one of them can be discarded and the remaining one can be considered as the merged EVM.

Case B. If the location range is identical and the reported sensor data thresholds differ, then for each reported event type, the largest reported threshold needs to be included in the merged message.

Case C. If the location ranges of the two EVMs are mergeable and the reported event types and their thresholds are identical, the merged EVM contains the merged location range and exactly these thresholds.

Case D. If both the location ranges and the event types or their threshold differ, a straightforward merging of the two EVMs is not possible, since including the sensor data of one location

* For a more rigorous investigation of correlation coefficients, physical and chemical features of forest fires, like fire spread models (see, e.g., [93]) and sensor characteristics, need to be investigated in more detail. These topics, however, are beyond the scope of this thesis.

† The development of an actual algorithm that identifies the mergeability and selection of the optimum range type(s) is not within the focus of this thesis. Methods of pattern recognition can possibly be applied here.

range must not imply false conclusions on the other location range. In principle, an additional, third EVM could be generated that includes the commonalities (i.e., the minima of the reported thresholds). As far as possible, the common information could then be removed from the two original EVMs. This process would probably increase the future mergeability of all three messages. However, if the increased mergeability outweighs the additional message cannot be estimated easily. This thesis refrains from investigating these issues in more detail but assumes that no merging is done in *Case D*.

4.3.8 Capacity of EVM Pools

Remember from Section 4.1 that each sensor node maintains two pools of EVMs: the inPool, which contains the unacknowledged EVMs with individual back-off timers, and the outPool, which stores the EVMs that the node accepted to relay. In this section, the storage capacity of the pools is discusses, i.e., the number of EVMs that can be buffered in them. These capacities have significant influence on the size of the state space-based models introduced in Chapter 5 (in particular, see Sections 5.2.2 to 5.2.3 and 5.3.2).

First, the influence of memory constraints on the maximum total number N_m of EVMs that can be stored on a sensor node is investigated. N_m by can be estimated by

$$N_m \approx \frac{D_{\text{EVM,n}}}{D_{\text{EVM}}} \approx 147\,000\,,$$

where $D_{\text{EVM}} = 68\,\text{bit}$ is the maximum length of an EVM's DATA field (cf. Fig. 4.26(a)) and $D_{\text{EVM,n}}$ is the estimated total memory available for EVMs. According to Section 4.2.13, $D_{\text{EVM,n}} \approx 10\,\text{Mbit}$, even if the volume of the memory hardware used for storing EVMs is only about $1\,\text{mm}^3$.

Remember that EVMs that origin from the same cell can be easily merged (*Case A* and *Case B* of Section 4.3.7). Hence, the maximum number of distinct, i.e., unmergeable EVMs in the investigated scenario is equal to the number of grid cells $N_C \approx 2700$. Since this is far less than N_m, the memory by itself does not pose a severe limitation.

However, note that, in addition to direct memory consumption, storing a large number of EVMs also implies further challenges to the node. For example, the node needs to maintain a back-off and retransmission timer for each EVM stored in the inPool and outPool, respectively. Additionally, each incoming EVM needs to be compared to all stored EVMs to check mergeability and to calculate the back-off delay. Obviously, a node would also like to keep the number of stored EVMs small to be able to return to the sleep mode early, and hence, to reduce its own energy consumption. On the other hand, acting too egoistic might increase the total energy consumption within the node's neighborhood. Moreover, the higher the number of EVMs stored in the inPool, the more likely the node's situation (like energy level, number of nodes in the outPool) changes until the back-off delay of a given EVM expires. By then, the back-off delay might no longer reflect the node's situation well anymore. Keeping the inPool small also keeps EVMs longer at previous-hop nodes. This may increase the chance that EVMs are merged earlier on their path to the sink.

The qualitative arguments given so far are mostly in favor of a small capacity of the pools, in particular of the inPool. However, also increasing the capacities has several advantages. For example, the probability that a previous-hop node needs to repeat an EVM is expected to decrease when the EVMs are more likely accepted by next-hop nodes due to larger inPools.

This also decreases overall detection-to-notification delay. Larger inPools also increase the chance that inPool messages can be merged with outPool messages that are to be sent.

Investigating the details and finding the optimum capacity parameters is an interesting subproblem which needs to be solved by future work when the initial performance investigation of the XLMMP draft carried out in Chapter 5 leads to promising results. This investigation is limited to the exploration of the case where the capacity of the inPool is one EVM. An increased capacity of the inPool would lead to significantly increased model complexity as discussed in Section 5.2.3. The capacity of the outPool is assumed to be virtually unrestricted. See Section 5.2.2 for details.

4.4 Chapter Summary

This chapter is summarized by a qualitative evaluation of XLMMP in Section 4.4.1 and by a discussion of protocol-related open research issues in Section 4.4.2.

4.4.1 Qualitative Evaluation

Here, a qualitative evaluation of XLMMP is conducted by discussing whether XLMMP follows design principles proposed for WSNs and by comparing its basic behavior to XLP.

A quantitative evaluation of XLMMP is carried out in Chapter 5 with a focus on the trade-off between energy efficiency and end-to-end delay.

4.4.1.1 Qualitative Evaluation Based on Design Principles

The authors of [143] state that considering the specific needs of a concrete application is more important than following generic design suggestions. Still, in [143, Sec. 3.3], design principles for WSNs are suggested. This section briefly discusses whether XLMMP follows these principles.

Distributed organization. Distributed organization should be preferred to improve scalability and robustness, e.g. by avoiding single points of failures. XLMMP's organization is highly distributed since most important decisions (like medium access, routing, and localization) are done locally at the nodes in a self-organized manner. Robustness and scalability is also improved by providing more than a single sink.

However, following [143], it should be noted that a complete decentralization might also lead to disadvantages. Therefore, a compromise is usually aimed for by applying centralization in a locally restricted scope, e.g., by maintaining dynamic hierarchies like node clusters. In XLMMP, a form of hierarchy is built by the hop count rings which are used for localization and routing. Apart from this structure, XLMMP refrains from building more fine-granular hierarchies or clusters since the maintenance overhead is considered disproportional.

In-network processing. In a distributedly organized network, the nodes should be involved in taking decisions. While such in-network processing may take manifold forms, [143] mentions a few example techniques. These are related to XLMMP in the following.

Aggregation is one of the simplest examples of in-network techniques. In XLMMP, aggregation is performed due to the message merging property.

Distributed source coding and distributed compression is not implemented explicitly in XLMMP. Spatial and temporal correlation of the event data is exploited by message merging. Due to the relatively small amount of data that is contained in EVMs, a further reduction of bits by distributed source coding or compression is not foreseen in XLMMP.

Distributed and collaborative signal processing is very basically done at each sensor node, which compares its sensors' signals to the current reporting thresholds for deciding whether the event needs to be reported. More sophisticated signal processing (like beamforming or target tracking) is not intended by XLMMP, since it is not considered essential for the investigated application.

Mobile code/Agent-based networking is avoided in XLMMP since the corresponding communication overhead is considered significant and the added value stands to reason due to the quite restricted scope of the WSN's duties.

Adaptive fidelity and accuracy. XLMMP addresses adaptive fidelity by its energy awareness. A decrease in available energy in the network results in a graceful degradation of the detection-to-notification delay due to the increasing back-off delays chosen by the sensor nodes during receiver contention. A node's back-off delay is also increased with increasing load. The resulting increase of the detection-to-notification delay increases the chance of message merging and hence decreases the traffic load within the network. Moreover, the controller is able to control the traffic load adaptively by modifying the reporting thresholds. An additional adaptivity of fidelity and accuracy is not yet provided by XLMMP. However, by providing additional SMSG commands, the operator could influence the cell sizes, sensing rates, the maximum back-off delay, active/sleep ratios, and EVM repetition frequencies.

Data centricity. XLMMP is inherently data centric. Individual sensor nodes cannot be addressed due to the lack of node IDs.

Exploit location information. In XLMMP, location information is one of the main aspects of event data and for message merging. Moreover, the location information is strongly intertwined with the network topology (hop count rings) used for routing the EVMs to any sink.

Exploit activity patterns. XLMMP is tailored towards the monitoring of rare events. In particular, XLMMP tries to keep the communication at a minimum when there is no event by avoiding to maintain a fine-granular hierarchy of nodes and by refraining from keeping the nodes time synchronized.

Exploit heterogeneity. The authors of [143] differ between heterogeneity by construction and heterogeneity by evolution. XLMMP only provides a very limited form of heterogeneity by construction by employing two kind of nodes: wireless sensor nodes and sink nodes. A further construction-based heterogeneity within the WSN is not foreseen, since it complicates deployment and increases the chance of single points of failures. Heterogeneity by evolution can be seen as being directly related to the nodes' energy-aware adaptivity. Nodes that run out of energy are less actively participate in routing tasks.

Component-based protocol stacks and cross-layer optimization. XLMMP is a cross-layer protocol that merges application, transport, network, and link layer functionalities. Components are not yet defined explicitly. However, when implementing the protocol in future work, it might be helpful to split the functionality into different building blocks. For example, the handling of EVMs (event reporting) could be tackled in another component than the handling of SMSGs (network management, localization, etc.). Another orthogonal component could cover communication security aspects.

4.4.1.2 Comparison to XLP

In addition to the discussion in Section 2.2, Fig. 4.28 directly compares the message sequence charts of XLMMP and XLP (cf. [320]). The figure illustrates that XLMMP is able to transmit the event data from Node (C,8) to Node (E,6) with less communication overhead. Assuming similar back-off delays and radio technology, XLMMP can hence be expected to outperform XLP in the investigated scenario where the data packets (EVMs) are very short.

4.4.2 Open Research Issues

A more detailed discussion of XLMMP's behavior is postponed to future work and should address, for example,

- investigation of the protocol behavior in face of (wanted or unwanted) node mobility,
- finding the optimum frequency of SMSG generation for hop count resets and updates,
- deriving the optimum frequency of EVM repetition at the sensing nodes,

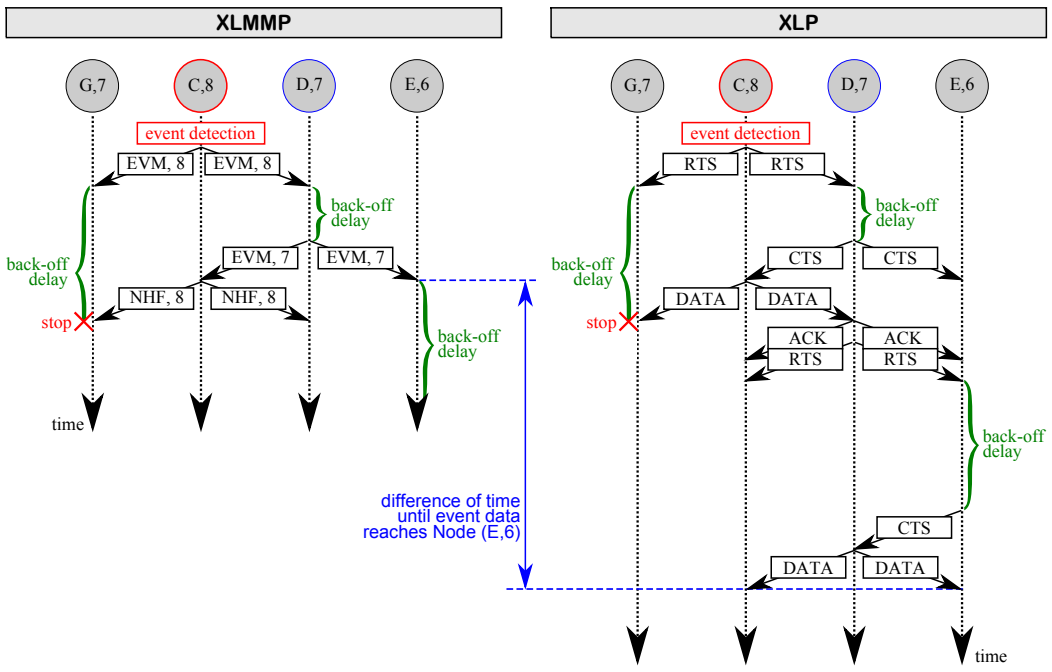


Figure 4.28: Message sequence chart comparison of XLMMP and XLP.

- finding the optimum cell size for localization and message merging,
- deriving the optimum C_{b,E_r}, C_{b,N_q}, and $C_{b,I_{\text{mm}}}$ parameters for calculating the back-off delay and investigation of alternatives,
- identifying the optimum maximum back-off delay $T_b^{(\max)}$
- searching for the optimum maximum number of retrials before a node increases its hop count,
- investigating the distribution of energy consumption within one ring in dependence of the chosen protocol improvement (cf. 4.3.2.5),
- investigation of security aspects*,
- examining the protocol behavior in face of hardware failures†,
- investigation of the influence of different node distributions on the protocol behavior‡,
- analyzing and improving the broadcasting method for SMSGs (e.g., by applying one of the improvements surveyed in [324, Sec. V.B]),
- in particular, analyzing and improving the convergence of hop count and threshold updates§,
- discussing the advantages and disadvantages of *late message merging*, i.e., nodes keep a limited history of recently forwarded EVMs with which arriving EVMs could still be merged,
- exploring the optimum number of sinks (increased localization accuracy and decreased communication path lengths versus message size and sink cost),

* WSN-specific security threads and possible counter measures are surveyed in, e.g., [60, 190, 273, 321]. In the application scenario of forest fire detection, especially arsonists may tamper with the WSN. Obvious attacks include pretending to be a sink to attract EVMs (called *black hole attack* or *sink hole attack*) or frequently triggering hop-count updates to drain the WSN's energy (called *energy drain attack* or *resource exhaustion attack*).

† For example, if a node is not able to detect its hardware failure by local mechanisms and reports events erroneously, SMSGs might be used to switch this reporting off. Since there are no node IDs, the SMSG cannot be addressed directly to the faulty node. Instead, the command should state, e.g., "If your location is in range x and your current value for sensor y exceeds threshold z, then deactivate sensor y since it produces false events." This only works, if the node is still able to receive and carry out the command. Hence, a similar SMSG might be constructed that enables the operator to switch off routing for specific events at all nodes. This, however, requires all nodes to be able to maintain some sort of blacklist. In any case, the operator is responsible to ensure that the reported value is indeed incorrect.

‡ In [41], one possible, generic way for relaxing the assumption of uniform node distributions is described. Also, a node distribution model that describes the deployment of nodes from aerial vehicles more realistically should be developed and included in the investigations.

§ Since the increase of the reporting threshold using SMSGs could be seen as an end-to-end ACK sent by the sinks and modification of the reporting thresholds enable the sinks to control the traffic load in the network, some minimum broadcasting performance likely needs to be maintained here. Moreover, it might be worthwhile to think about mechanisms that allow to reduce the network load, e.g., by assigning some validity period to thresholds before the nodes automatically return to the default threshold.

- investigating the power consumption of suitable sensor, processor, and memory implementations*,
- identification of mergeable locations and optimum representation of location ranges.

Note that the current lack of these protocol details hampers a rigorous quantitative comparison to other WSN protocols. Their discussion require representations of XLMMP with a higher level of resolution than aspired in this thesis, for example by implementing more realistic simulations and/or testbed implementations of the protocol (see also Section 6.2). However, before investing more research effort into tackling these issues, the basic behavioral mechanisms of XLMMP are evaluated first. This is carried out in Chapter 5 of this thesis.

* In particular, the assumption that the transceiver dominates the power consumption might not hold any more if $p_{\text{sleep}} \approx 1$.

Chapter 5

Quantitative Protocol Evaluation

Not everything that can be counted counts, and not everything that counts can be counted.

— Albert Einstein

This chapter is devoted to the numerical evaluation of XLMMP. On the one hand, it aims at estimating the feasibility of XLMMP with respect to the lifetime versus response time trade-off. On the other hand, this chapter presents novel research in the field of finite-source retrial queues, in particular regarding the distribution of the waiting time in face of unreliable servers. Since the evaluation also allows to investigate the sensitivity of the quantitative results on changes to individual model parameters, the evaluation also points to aspects of the WSN that should preferably be modeled and investigated in more detail in future work.

5.1 Model Planning

The WSNs operation based on XLMMP can be divided into three phases. The *setup/update phase* is dominated by the communication of SMSGs, which is costly, due to the broadcast character. However, the phase is initiated only rarely by the sinks. In the majority of time, the (event-less) *monitoring phase* is active. Hence, energy consumption in this phase is critical for the lifetime of the network. Since no communication is going on in this phase, the energy consumption is mainly determined by the sensing and the transceivers' active/sleep periods, i.e., by p_{sleep} introduced in Section 4.2.8. But, this p_{sleep} has a significant influence on the performance of the WSN in the *event reporting phase*. In this phase, the detection-to-notification delay is critical.

The main focus of the evaluation presented in this thesis is therefore the case of an upcoming fire event and the investigation of the influence of the active/sleep periods on the detection-to-notification delay.

5.1.1 Aspired Quantitative Results

XLMMP is based on multi-hop communication from the sensor nodes to the sink. Its evaluation is done in two steps. First, a single hop within the communication path is investigated. Then, the obtained results are used to evaluate the multi-hop behavior. For both models, the scopes of interest within the communication topology are exemplarily illustrated in Fig. 5.1.

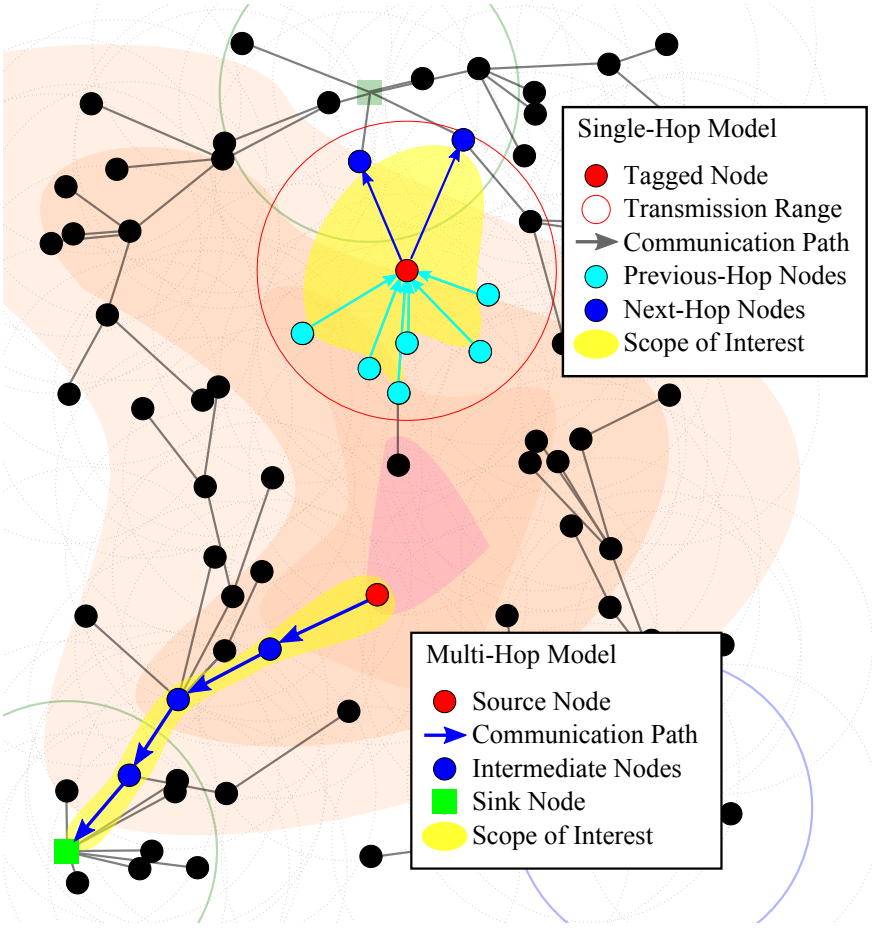


Figure 5.1: Scopes of interest of single- and multi-hop model within the communication topology. (Based on Fig. 4.1.)

The single-hop model is presented and investigated in Sections 5.2 to 5.5. It is tailored towards the discussion of the time needed by an EVM to pass one hop in dependence of the sensor nodes' active/sleep ratio. The active sensor node that happens to forward the EVM over the investigated hop is referred to as *tagged node* in the following. The tagged node receives its EVMs from some previous-hop nodes and sends them to some next-hop nodes towards the sink. More precisely, the time between the tagged node sending the updated, possibly merged EVM with decreased hop count* to its neighbors and receiving the corresponding ACK by one of its next-hop nodes is investigated. This time is equivalent to the time the EVM spends in the tagged node's outPool and referred to as *(single-hop) response time* in the following.

* Remember that the updated EVM serves as ACK to the previous-hop nodes.

Remember that this thesis aims at answering the following related questions in dependence of the system parameters.

Question 1: Which upper bound of the detection-to-notification delay is met by a given fraction of events?

Question 2: What is the fraction of events that meet a given upper bound of the detection-to-notification delay?

Consequently, these questions need to be answered on a per-hop basis first. The obtained results serve as input to the multi-hop model presented in Section 5.7, which then discusses the resulting (end-to-end, i.e., source-to-sink) detection-to-notification delay.

5.1.2 General Abstractions

The single- and multi-hop models need to abstract from several features of the investigated scenario for keeping the models practical and their evaluation by numerical analysis mathematically tractable. In particular, the following simplifications are considered.

While XLMMP provides mechanisms to deal with message collisions and loss on the wireless channel as well as node failures, the models discussed in the following do not yet evaluate these mechanisms to conserve conciseness. Still, it is needless to say that resilience is an important property of any security-relevant technical (monitoring) system. Since the results derived in the following appear to be promising, it seems reasonable to invest future research effort on addressing these issues.

For mathematical tractability, all time parameters (i.e., generation times, service times, active and sleep durations, retrial times) are assumed to be exponentially (i.e., continuous and memoryless) distributed. However, since the obtained allow to investigate the sensitivity of the performance measures on changes to the model parameters, this thesis points towards the parameters for which the assumption of exponential distribution should be relaxed first by model generalizations in future work. Note that, in general, this assumption does not imply that the performance results (response times, waiting times) are also exponentially distributed.

In the investigated state of a single upcoming fire, it is assumed there is no significant cross-traffic generated by other fires. Additionally, it is assumed that no communication of SMSGs is ongoing. The investigation of the influence of concurrent traffic as well as of high-load scenarios is postponed to future work.

As announced in Section 4.3.8, the thesis restricts itself to the case where the capacity of the inPool is one EVM. From a modeling point of view, an increase of the inPool's capacity would lead to much more complex models. In particular, it is shown in Section 5.3.2 that for an inPool capacity of one, the system state of the single-hop model can be described sufficiently by the number of sleeping next-hop nodes, the number of EVM-processing next-hop nodes, and the number of EVMs currently processed at the investigated node. If the inPool capacity is increased, the state space gets an additional dimension for each next-hop node. Also note that approximating an increased inPool capacity by just increasing the number of servers in the single-hop model* does not lead to comparable results in general (see Appendix Section B.5). A closer investigation of this issue also needs to be postponed to future work.

* That is, several servers are provided for each next-hop node without considering them failing in groups when the corresponding neighbor falls asleep.

5.2 Single-Hop Model: Retrial Queue Representation

In classical $M/M/m-FCFS$ queueing systems (see Fig. 5.2)* *jobs*† arrive to the queueing system according to a Poisson process with arrival rate λ. Each job gets served by one of m servers before leaving the queueing system, where $m = N_\mu$ in this thesis' notation. The servers are identical and have an exponentially distributed service time with rate μ each. If all servers are busy, arriving jobs enter the queue of infinite capacity. As soon as one of the servers finishes its service, the processed job departs the queueing system and the server it takes a waiting job from the queue according to the first-come-first-served queueing discipline.

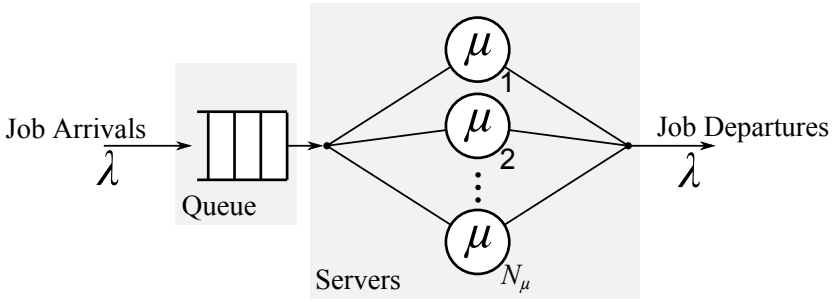


Figure 5.2: A classical $M/M/m-FCFS$ queueing system.

Several features of the WSN scenario, however, render the modeling using $M/M/m-FCFS$ queueing systems infeasible. First, the active/sleep periods likely have a significant effect on the performance measures. Second, the number of distinct, non-mergeable event messages is limited. Third, sensor nodes are not immediately aware of idle next-hop nodes.

To tackle these features, this work applies models based on $-/M/m$ *finite-source retrial queueing systems with unreliable servers*. A graphical representation of such a queueing system is shown in Fig. 5.3. The states of each unreliable server are illustrated in Fig. 5.4‡.

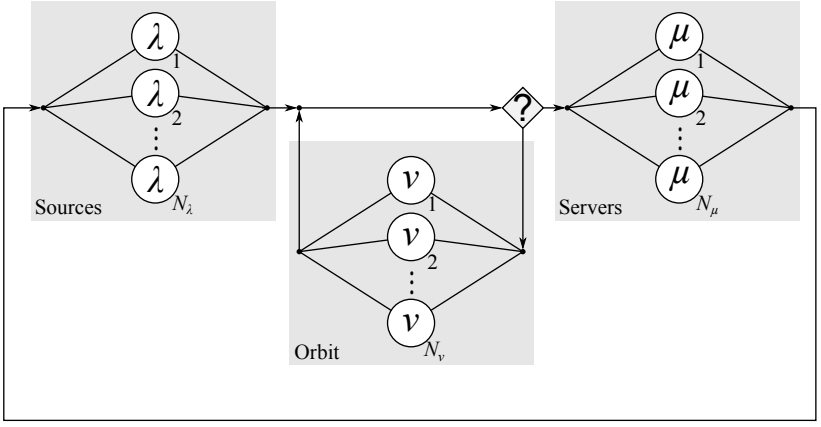


Figure 5.3: A $-/M/m$ finite-source retrial queueing system.

* For an introduction to queueing systems and Kendall's notation, see, e.g., [153] or [46, Sec. 6].
† Depending on the application, these jobs are also called *customers*, *calls*, *tokens*, or *workload*.
‡ A GSPN representation which combines Figs. 5.3 and 5.4 is presented in Section 5.3.1.

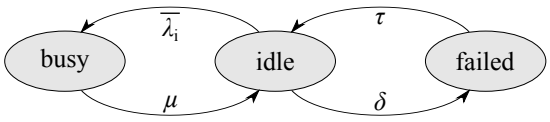


Figure 5.4: State diagram of each unreliable server.

Using the terminology of queueing theory, the behavior of the investigated model can be described as follows.* There are N_λ *sources*, N_μ *servers*, and an *orbit* of capacity N_ν. The servers and the orbit are subsumed under the term *system* in the following. The system has a finite capacity of N_c. A server that currently processes a job is called *busy*. The number of busy servers is denoted by $N_{\mu b}$. The number of jobs in the orbit is denoted by $N_{\nu b}$. Consequently, the number of jobs in the system is given by $N_{\mu b} + N_{\nu b}$ and the system is full if $N_{\mu b} + N_{\nu b} = N_c$. Each active source generates a job with generation rate λ. Unless the system is full, the generated job enters the system at generation instant. A system-entering job is called *arriving job* or just *arrival* in the following. If the system is full, a generated job is *blocked* and does not lead to an arrival. Due to this blocking†, the job stays at its source. The corresponding blocked source immediately restarts the job's generation—again with generation rate λ. An arriving job directly enters an idle server (if any), which then becomes busy. Such a job is called *direct job* in the following. If none of the N_μ servers is idle, the arriving job enters the orbit instead and is called *orbit-visiting job* from now on. Each orbiting job retries to get access to an idle server with retrial rate ν and re-joins the orbit if still no idle server is available. Each busy server processes its job with a service rate of μ. After service, the job returns to the sources where its generation is restarted. Idle servers fail with a failure rate of δ. Each failed server is repaired with a repair rate of τ. Note that the applied model does not provide for the failure of busy servers. An idle server gets busy with rate $\overline{\lambda_i}$, which describes the mean arrival rate of jobs at an idle server. Note that $\overline{\lambda_i}$ is not a model parameter but a performance measure, since it is determined by the set of model parameters. An expression for calculating $\overline{\lambda_i}$ is given by Eq. (5.11) in Section 5.3.3, where also further performance measures are derived.

The described finite source retrial queue with unreliable servers is used to model a single hop of XLMMP shortly after the appearance of a previously unknown fire event. Roughly speaking, the jobs refer to the EVMs, sources refer to EVM generation and EVM relaying by the previous-hop nodes, the orbit represents the tagged node, and the servers model the next-hop nodes.

More precisely, each job represents a distinct EVM that cannot be merged with messages that are already handled by the tagged node. Let N_λ denote the maximum number of these distinct EVMs. From the point of view of the tagged node, these distinct EVMs can be in different states as illustrated in Fig. 5.5. The illustration also describes how the different states are mapped to the model.

The jobs located at the sources represent the subset of EVMs for which the tagged node is currently not responsible. All of these events are currently unknown to the tagged node and

* Numerical ranges of all introduced model parameters are derived in Sections 5.2.1 to 5.2.7 and summarized in Section 5.2.8.

† Note that the notion of *blocking* chosen in this thesis is comparable to the one used in [46, Sec. 10.8]. An alternative notion is to call each arriving job *blocked* unless it is not able to join a server immediately. Instead, the latter is called *orbit-visiting* job in this thesis.

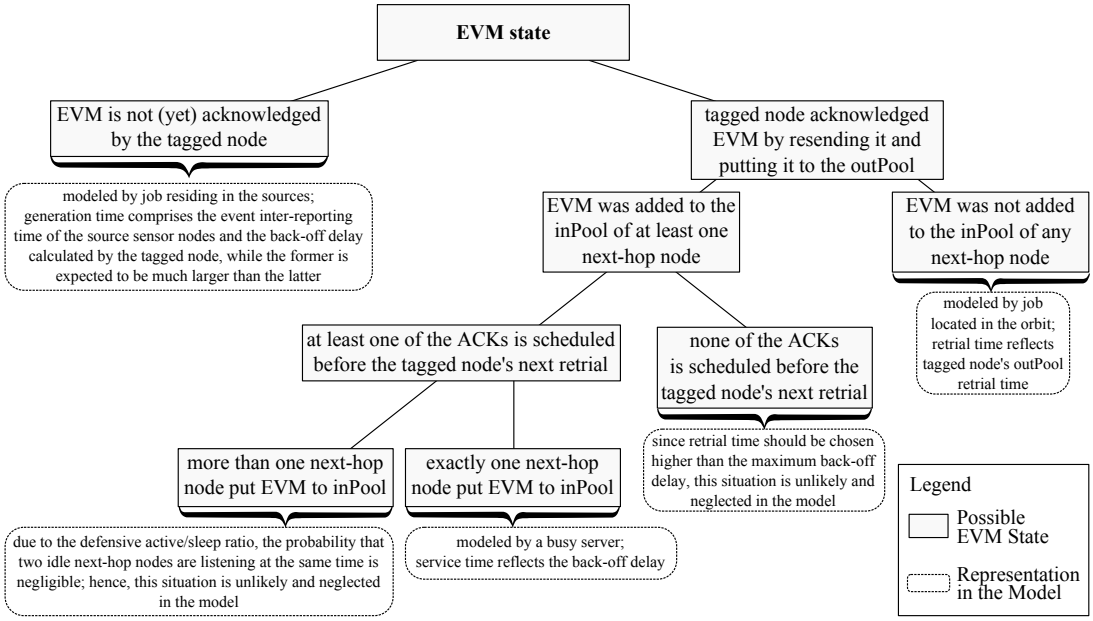


Figure 5.5: EVM states and their mapping to the single-hop model.

result in an unmergeable EVM sent to the tagged node with rate λ. In short, λ relates to the generation rate of distinct EVMs at the event-detecting source nodes.

The service time (with mean μ^{-1}) mainly* models the *back-off delay* T_b of a next-hop node. Note that the back-off delay is only relevant for EVMs that successfully reached a next-hop node, i.e., for EVMs that are eventually acknowledged by a next-hop node. Otherwise, the time spent in retrial state at the tagged node (i.e., in the orbit) applies. The number of servers N_μ resembles the overall number of EVMs that can be stored in the inPools of all next-hop nodes.

The jobs located at the orbit of size N_ν represent EVMs for which the tagged node currently feels responsible. More precisely, these EVMs have not yet reached a next-hop node or will not be acknowledged by a next-hop node before the EVM's next retrial. Each of these EVMs is resent by the tagged node with rate ν (i.e., after a timeout with duration ν^{-1}).

In total, the node may have to store a maximum of $N_c = N_\nu + N_\mu$ EVMs. Only mergeable incoming EVMs are accepted by the tagged node while capacity N_c is reached.

As long as the node takes care of an EVM (modeled by a job staying at the system, i.e., at the orbit or servers), correlated EVMs can be merged immediately, do not imply additional communication, and do not significantly contribute to the time the event is reported initially. However, the tagged node resets its message-specific knowledge on a forwarded EVM as soon as it receives the ACK by the next hop (job leaves the server and returns to the sources). Consequently, an identical EVM may then be generated by previous-hop nodes and accepted by the tagged node with rate λ.

In XLMMP, next-hop nodes might reject incoming EVMs based on their local circumstances, e.g., their energy level or memory status. In particular, next-hop nodes reject all messages if they are in sleep mode. The rejection case is represented by a server's failure state,

* The mapping of the model parameters to the corresponding system properties is detailed further in Sections 5.2.1 to 5.2.7.

entered by idle servers with rate δ each. A failed server returns to idle state with rate τ. Since an EVM-holding sensor node maintains a running back-off delay timer and needs to listen to the channel to infer whether one of its neighbors responds earlier and to receive mergeable EVMs, EVM-holding nodes to not switch to sleep mode. Consequently, a busy server does not fail. Hence, the probability that a server is in state *failed* of Fig. 5.4 is different from p_{sleep}, since the latter is determined for the event-less monitoring phase where no traffic keeps the sensor nodes from switching to sleep mode.

In the following Sections 5.2.1 to 5.2.7, the model parameters are now discussed in more detail and their numerical values are estimated. These estimations are mainly based on the application scenario and WSN protocol introduced in Chapter 3 and 4, respectively. Section 5.2.8 then summarizes the obtained parameter ranges.

It should be noted that several parameter interdependencies are identified in the following sections. Some (but far from all) of these dependencies are exploited in Sections 5.3 and following to reduce the proposed models' complexity. Section 5.2.8 also outlines which interdependencies can be relaxed without having to change these models.

5.2.1 Estimating the Number of Sources N_λ

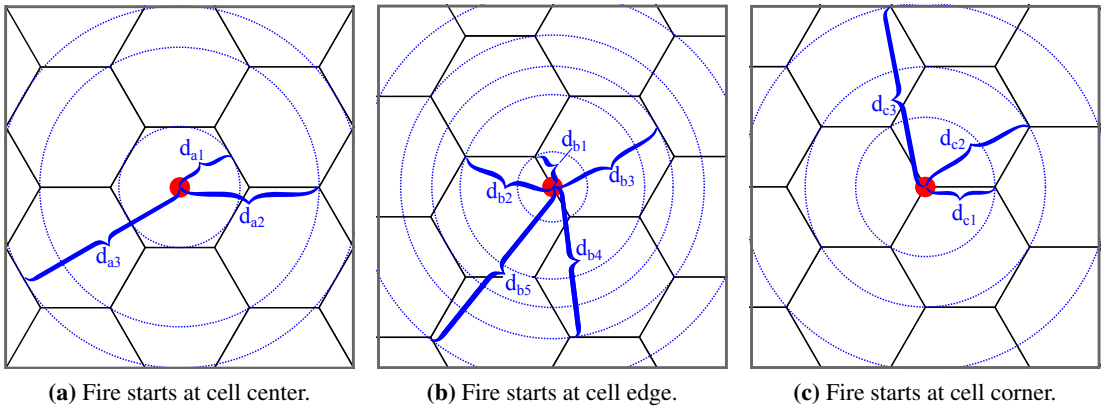
The sources represent the overall number of distinct (i.e., unmergeable) EVMs that potentially try to pass through the tagged node simultaneously. To estimate the number of sources, the situation of a single, upcoming fire event is investigated, as discussed at the beginning of Section 5. Initially, only a single cell might be affected by the starting fire. Remember that EVMs that origin from the same cell can be easily merged (*Case A* and *Case B* of Section 4.3.7). If this is not already done by other nodes, it can be done by the tagged node. However, if the fire is starting close to the cell's edge, one or two additional, neighboring cells might be affected immediately. The corresponding EVMs can only be merged if they contain the same event types (*Case C* of Section 4.3.7). How fast further cells are affected significantly depends on the cells' size, the starting location of the fire within the grid, and on the speed of the fire's growth. The latter highly depends on environmental parameters like fuel and weather conditions.

Assuming an omnidirectional spread of the fire, the influence of the fire's location and size on the number of affected cells is illustrated in Fig. 5.6 for three selected fire starting locations. The fire radii d_{a1} to d_{c3} are summarized in Table 5.1. Additionally, the dependency is shown quantitatively in Fig. 5.7.

For example, if the fire starts close to the center of the edge between two cells (Fig. 5.6(b) and Curve (b) in Fig. 5.7), it immediately affects the two adjacent cells and hence might be reported by nodes located in both cells. If it radially grows by $0.5\,d_C = 50\,\text{m}$, it affects two more cells, i.e., four cells in total.

Still assuming omnidirectional fire spread, it can be deduced from Fig. 5.7 that the maximum number of affected cells that can be expected for fire radii of $0.5\,d_C = 50\,\text{m}$, $d_C = 100\,\text{m}$, $1.5\,d_C = 150\,\text{m}$, and $2\,d_C = 200\,\text{m}$, is 4, 7, 8, and 13, respectively.

In the following, the fire is assumed to not grow larger than 100 m in the first few minutes until the sinks get initially aware of the fire, and hence, a sensor node might get up to seven distinct messages during this time. Shortly after, the operator is expected to initiate secondary monitoring measures (if available) and/or send fire relief units. Moreover, the operator starts to increase the reporting thresholds within the affected region to reduce network traffic.



(a) Fire starts at cell center. **(b)** Fire starts at cell edge. **(c)** Fire starts at cell corner.

(d) Legend.

Figure 5.6: Influence of a fire's starting location and size on the number of affected cells.

Table 5.1: Example fire radii of Fig. 5.6

Radius	Value	Number of Affected Cells
d_{a0}	$0\,d_C$	1
d_{a1}	$\frac{\sqrt{3}}{2}\,d_C \approx 0.87\,d_C$	7
d_{a2}	$2\,d_C$	13
d_{a3}	$\frac{3}{2}\sqrt{3}\,d_C \approx 2.60\,d_C$	19
d_{b0}	$0\,d_C$	2
d_{b1}	$0.5\,d_C$	4
d_{b2}	$\frac{\sqrt{7}}{2}\,d_C \approx 1.32\,d_C$	8
d_{b3}	$\sqrt{3}\,d_C \approx 1.73$	10
d_{b4}	$\frac{\sqrt{19}}{2}\,d_C \approx 2.18\,d_C$	14
d_{b5}	$\frac{\sqrt{31}}{2}\,d_C \approx 2.78\,d_C$	18
d_{c0}	$0\,d_C$	3
d_{c1}	$1\,d_C$	6
d_{c2}	$\sqrt{3}\,d_C \approx 1.73\,d_C$	12
d_{c3}	$\sqrt{7}\,d_C \approx 2.65\,d_C$	18

A worst-case upper bound of the number of distinct messages appearing almost simultaneously at a sensor node is the overall number of cells in the WSN $N_C \approx 2700$ (see Section 4.3.5.2). However, note that when having three sinks, the WSN is divided into three communication trees. The number of cells covered by each communication tree depends on the placement of sinks within the monitored area. For reasonable placements, the largest of the three communication trees should cover between about $\frac{N_C}{3} \approx 900$ to $\frac{N_C}{2} \approx 1350$ cells. Hence, in the unrealistic scenario of a sudden ignition of all cells within the largest communication tree, nodes that are located close to the corresponding sink might have to deal with up to 1350

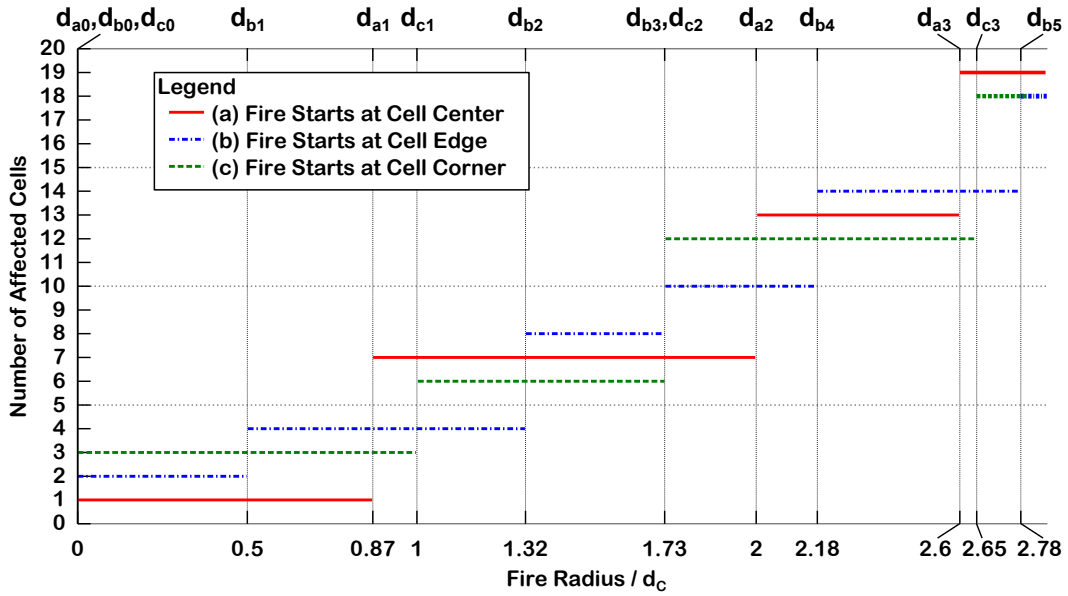


Figure 5.7: Number of affected cells versus fire size for three different fire starting locations.

distinct EVMs, unless these EVMs can be merged further. Moreover, the effort of forwarding the distinct EVMs is usually shared between all active nodes within the same-ring neighborhood of the tagged node. According to Section 4.3.2.3, the same-ring neighborhood comprises approximately $\overline{N}_\mathrm{r} \approx \overline{A}_\mathrm{r} \cdot \overline{N}_d \approx 15\,300\,\mathrm{m}^2 \cdot 1400\,\mathrm{km}^{-2} \approx 21$ nodes, including the tagged node. These neighbors, however, are partly sleeping. The actual number $\overline{N}_\mathrm{Nar} = p_\mathrm{active}\overline{N}_\mathrm{r}$ of same-ring neighbors that are currently active further depends on the probability p_active that a node is active which, e.g., depends on the yet unknown active/sleep cycle parameters τ and δ.

To summarize, on the one hand, for investigating the more realistic scenario, the generation rate is set to $N_\lambda = 7$ by focusing on a single, upcoming fire event and by defining the tagged nodes as one of the nodes that are actually involved in the initial communication of the event.

On the other hand, the derived models and in particular the evaluation methods should be suitable for significantly higher N_λ. This allows to apply them for estimating the network's performance in the face of several fires, more mature fires, or smaller cell radius d_C and hence increased number of cells per affected area. In this thesis, the scalability of the evaluation methods is investigated for N_λ up to at least 100.

5.2.2 Estimating the System Capacity N_c and Orbit Size N_ν

The system capacity N_c refers to the maximum number of jobs that can be located at the orbit (not more than N_ν) or the servers (not more than N_μ), i.e., $N_\mathrm{c} \leq N_\nu + N_\mu$. Remember that all these jobs refer to EVMs that are already acknowledged by the tagged node, i.e., the EVM is located in the node's outPool (cf. Section 4.1).

From a modeling point of view, there is no need to provide a system capacity that is higher than N_λ, since the number of sources N_λ dictates the maximum number of jobs that may enter the orbit and servers simultaneously. Moreover, based on the discussion carried out in Section 4.3.8, it seems feasible to assume that a sufficient outPool capacity can easily be provided for $N_\lambda \leq 100$. Consequently, it is safe to focus on $N_\mathrm{c} = N_\lambda$ in the following.

The same argumentation induces $N_\nu \leq N_c = N_\lambda$. Moreover, since all next hops might reject EVMs at the same time, all stored EVMs might have to be retransmitted, i.e., all jobs may reside in the orbit. Therefore, $N_\nu = N_c = N_\lambda$ in the following.

5.2.3 Estimating the Number of Servers N_μ

Jobs located in the servers refer to EVMs that are stored in next-hop neighbors' inPools. As already illustrated in Fig. 5.5, it is assumed in the following that due to the defensive active/sleep ratio, the probability that more than a single next-hop node is available at a time is very low. Hence, the model is based on the assumption that each EVM is stored in the inPool of at most one next-hop node.

Based on the results of Section 4.3.2.3, the mean number of next-hop neighbors can be approximated by $\overline{N}_{r^-} \approx 9$. According to Section 4.3.8, the inPool capacity of each node is one EVM, that is, each node provides one memory slot that is able to hold a single EVM. Consequently, the overall number of EVMs that can be handled simultaneously by the inPools of the next-hop neighbors is given by $N_\mu = \overline{N}_{r^-} \approx 9$. Assuming that in the investigated scenario of an upcoming fire there is no significant cross-traffic from other nodes to the tagged node's next-hop neighbors, all of these N_μ next-hop inPool memory slots are usable by the tagged node as long as the corresponding next-hop nodes are active.

For keeping the evaluation models concise, it is assumed that $N_\mu \geq N_c = N_\lambda$. That is, the maximum number $N_c = N_\lambda$ of EVMs located in the outPool of the tagged node is less than the total number N_μ of inPool memory slots available at next-hop nodes.

Note that N_μ depends on the node density (linearly), on the number of inPool memory slots per node (linearly), and the transmission range (quadratically). This thesis investigates the scalability of the evaluation methods for N_μ up to 110, since $N_\mu \geq N_c = N_\lambda \leq 100$. That is, for checking the scalability, N_μ may be chosen in the same order of magnitude as N_c, and even in the case of the maximum N_c, it might exceed N_c by approximately 10%.

5.2.4 Estimating the Generation Rate λ

The generation rate λ is the reciprocal of the mean time after which an EVM re-appears at the tagged node after it has been removed from the outPool and hence cannot be merged at the tagged node anymore.

The value of λ depends on several factors, including the speed of the fire's growth, the node density, the sensors' sensing rate, the number of nodes in the affected area, the source nodes' repetition rate of EVMs, and the probability of message merging at previous hop nodes. To roughly curtail the range of λ, the worst (largest λ) and best (smallest λ) cases are estimated.

As already assumed in Section 5.2.1, the radius of the fire-affected area stays below 100 m during the investigated initial phase of the fire. Up to approximately $(100\,\mathrm{m})^2 \mathring{\pi} \overline{N}_d \approx 44$ nodes are affected by this fire. In the mean, the tagged node might receive up to 4.4 EVMs per second, if each node samples its environment every ten seconds*, the repetition of EVMs is not

* Finding the optimum sensing rate is postponed to future work since it requires knowledge about the sensor hardware used. The sensing rate is governed by a trade-off between the energy consumption needed for sensing the environment and the event-to-detection delay, while the latter is mostly dominated by the distance between the fire's starting location and the sensor. But also for calculating meaningful gradients, high sensing rates might be required.

bounded at the source nodes, and message merging is not possible on the path to the tagged node.

On the other hand, it seems to make sense to limit the source nodes' repetition rate of identical EVMs as discussed in Section 4.3.1.2. When following a relatively restrictive approach, source nodes may wait for the maximum time an EVM is expected to reach the sink and an SMSG containing a corresponding threshold update to return to the source node, i.e., in the order of a few minutes, say 10 min. If additionally only a single node was affected or all correlated messages could be merged before reaching the tagged node, only one message would arrive at the tagged node within 10 min.

Consequently, the protocol performance is investigated for a relatively broad range of λ between 0.001 and 5 arrivals per second.

5.2.5 Estimating the Retrial Rate ν

The retrial rate ν determines the rate at which EVMs located in the tagged node's outPool are repeated, since no ACK was received from next-hop nodes, yet.

Similar to the discussion of the arrival rate λ in Section 5.2.4, the range of ν is restricted by investigating rough upper and lower bounds.

An upper bound of ν is given due to the maximum radio transmission rate of EVMs. According to Section 4.2.14, data rates lie in the order of 100 kbps. Together with the EVM size of 39 to 70 bit (see Section 4.3.6), this results in a maximum EVM transmission rate of approximately* 1400 to $2600\,\mathrm{s}^{-1}$. Note, however, that sending with these rates would likely take up the full bandwidth of the wireless channel and significantly increase the message collision probability. Consequently, the retrial rate probably needs to be chosen significantly smaller. Accordingly, before resending the EVM, a node should give the next-hop nodes the chance to acknowledge its receipt, i.e., wait for approximately the maximum back-off delay $T_b^{\max}$. Following Section 4.3.3, $T_b^{\max}$ can be chosen between 1 ms and 2 s. Hence, an upper bound of the retrial rate is $\frac{1}{1\,\mathrm{ms}} = 1000\,\mathrm{s}^{-1}$.

Following the argumentation of the maximum back-off delay in Section 4.3.3 allows to deduce a lower bound of the retrial rate. That is, to meet the detection-to-notification delay requirement, the mean retrial time ν^{-1} should be chosen smaller than 2 s, i.e., $\nu > 0.5\,\mathrm{s}^{-1}$.

5.2.6 Estimating the Service Rate μ

The service time directly reflects the back-off delay plus the delays needed to transmit the message and the ACK between the tagged node and the next hop with the smallest back-off delay. Hence, an upper bound of the service time is the maximum back-off delay, i.e., $\mu^{-1} \leq 2\,\mathrm{s}$. A lower bound is twice the transmission delay of an EVM, i.e., $\mu^{-1} \geq 2 \cdot \frac{39\,\mathrm{bit}}{100\,\mathrm{kbps}} = 0.78\,\mathrm{ms}$.

Therefore, the system is investigated for a service rate μ between $0.5\,\mathrm{s}^{-1}$ and $1300\,\mathrm{s}^{-1}$.

5.2.7 Estimating the Repair Rate τ and Failure Rate δ

Remember from the beginning of Section 5.1 that the WSN spends the majority of time in the event-less monitoring phase. Hence, this thesis assumes that the sensor nodes' lifetime is dominated by the power consumed during this phase. For calculating the lifetime, focus is

* Following Section 4.2.9, the propagation delay is negligible.

therefore given to this phase. Note that, in this phase, the nodes' listen/sleep periods are not influenced by message traffic that keeps the nodes active. Then, the probability p_{sleep} that a node is in sleep mode, provided that there is no network traffic, is determined solely by the repair rate τ and failure rate δ. Since τ^{-1} and δ^{-1} model the duration of the sleep and active modes, respectively, p_{sleep} can be calculated as

$$p_{\text{sleep}} = \frac{\delta}{\delta + \tau}. \tag{5.1}$$

Consequently,

$$\delta = \frac{\tau}{1 - p_{\text{sleep}}}. \tag{5.2}$$

Based on Eqs. (4.1) and (5.1), a node's expected lifetime $\overline{T_{\text{life}}}$ can hence be calculated for a given δ and τ as

$$\overline{T_{\text{life}}} \approx \frac{E_{\text{trans}}}{\frac{\delta}{\delta+\tau}\left(P_{\text{sleep}} - P_{\text{active}}\right) + P_{\text{active}}}, \tag{5.3}$$

where $E_{\text{trans}} = 2400\,\text{J}$, $P_{\text{sleep}} = 5\,\mu\text{W}$, and $P_{\text{active}} = 50\,\text{mW}$ are hardware parameters as described in Section 4.2.8.

Remember that achieving a lifetime of at least three years demands that $p_{\text{sleep}} > 0.9996$, as motivated in Section 4.2.8. For this, Eq. (5.2) requires that δ is at least $2500 \cdot \tau$. Without further constraints, however, this requirement does not allow to estimate the absolute values of τ or δ.

A first upper bound of the repair rate can be determined by the wake-up time needed to reactivate the transceiver, which depends on the transceiver technology used. Typical state-of-the-art low-power transceivers (e.g., CC1000 [188, Tab. 2], CC2420 [307, Tab. 6.6–6.8], TR1000 [243, p. 7]) are subject to wake-up times in the order of approximately 1–5 ms, resulting in a maximum repair rate of $1000\,\text{s}^{-1}$. Consequently, the corresponding maximum failure rate amounts to $2.5 \cdot 10^6\,\text{s}^{-1}$, i.e., a minimum mean active mode duration of $0.4\,\mu\text{s}$.

However, the active mode should last at least long enough to enable a node to sense the carrier. For example, IEEE 802.15.4 suggests a carrier sensing time* equal to eight symbol periods (see [127, p. 54]). In the case of a CC2420 transceiver, this translates to a sensing time† of 128 µs (see [307, p. 11]). This duration appears quite long and defensive, for example, when compared to the carrier sensing time‡ of approximately 4 to 27 µs (depending on the selected physical layer specification) foreseen in IEEE 802.11 (see [129, p. 1487, 1503, 1514, 1532, 1632, 1663, 1761]). Assuming a carrier sensing time of roughly up to 30 µs in the following, this results in a lower bound $\delta^{-1} \leq 30\,\mu\text{s}$ of a node's active time. That is, the failure rate δ should be smaller than approximately $33.3 \cdot 10^3\,\text{s}^{-1}$. Consequently, the repair rate τ should be smaller than approximately $13.3\,\text{s}^{-1}$, that is, the minimum time spent in the sleep mode is roughly $\tau^{-1} \geq 75\,\text{ms}$.

In principle, the repair rate τ could be chosen arbitrarily small to maximize the energy efficiency. However, a very small τ can be expected to significantly increase the response time. In the remainder of this chapter, the influence is investigated quantitatively for τ down to approximately $0.5\,\text{s}^{-1}$. The corresponding lower bound of the failure rate δ is $1250\,\text{s}^{-1}$.

* [127] refers to the carrier sensing time as *clear channel assessment* (CCA) *time*.

† [307] refers to the carrier sensing time as *RSSI average time*.

‡ Referred to as *aCCATime* in [129].

Table 5.2: Parameter summary of single-hop model

Parameter Description		Values			
Model Terminology	Scenario Terminology	Range	*RealSet*	*VerSet*	*ScalSet*
N_λ Number of sources	Number of distinct EVMs that potentially pass through the tagged node	$1 \ldots 100$	7	10	100
N_c System capacity	Maximum number of EVMs that can be stored at the tagged node	$1 \ldots N_\lambda$	7	5	100
N_ν Orbit size	Maximum number of EVMs in tagged node's outPool	N_c	7	5	100
N_μ Number of servers	Maximum number of EVMs in the inner-ring neighbors's inPools	$N_c \ldots 110$	9	5	110
λ Generation rate	EVM reappearance rate at tagged node	$1 \cdot 10^{-3} \ldots 5\,s^{-1}$	$0.1\,s^{-1}$	$5\,s^{-1}$	$0.1\,s^{-1}$
ν Retrial rate	EVM retransmission rate	$0.5 \ldots 1 \cdot 10^3\,s^{-1}$	$5\,s^{-1}$	$5\,s^{-1}$	$0.1\,s^{-1}$
μ Service rate	ACK rate	$0.5 \ldots 1.3 \cdot 10^3\,s^{-1}$	$10\,s^{-1}$	$1\,s^{-1}$	$0.2\,s^{-1}$
τ Repair rate	Rate of leaving the sleep mode	$0.5 \ldots 20\,s^{-1}$	$1\,s^{-1}$	$1\,s^{-1}$	$1\,s^{-1}$
δ Failure rate	Rate of entering the sleep mode	$\tau \ldots 25 \cdot 10^3\,\tau$	$2.5 \cdot 10^3\,s^{-1}$	$5\,s^{-1}$	$100\,s^{-1}$

5.2.8 Parameter Summary

Table 5.2 summarizes the model parameters. The set of the realistic parameter values in column *RealSet* are based on the estimations carried out in Sections 5.2.1 to 5.2.7. These values are used by default in later sections unless explicitly specified otherwise. Note that rates ν, τ, and δ are used as the main control knobs for optimizing the "energy consumption versus response time" trade-off in later sections.

Two additional parameter sets are provided in the table. The set in column *VerSet* is mainly used to verify* the correctness of the implementation of the model and its evaluation by comparing evaluation results using different evaluation methods. The set defined by column *ScalSet* is used to check the scalability of the model evaluation approach. To further motivate the existence of *VerSet* and *ScalSet*, it should be noted that during carrying out the research this thesis is based on, these sets were investigated first to check the models' implementability and the implementations' correctness and scalability. Due to the positive experience gathered with *VerSet* and *RealSet*, more effort was then invested to derive the more realistic model parameters of

* For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

RealSet. In particular, to verify the correctness of the models' implementation and evaluation by comparing their results to existing literature, it is convenient to have $\mu = 1\,\mathrm{s}^{-1}$ in *VerSet*. In contrast to *RealSet* and *ScalSet*, *VerSet* also allows to investigate the models' behavior and the implementations' correctness for $N_\lambda > N_\mathrm{c}$. The large values of N_λ and N_c in the *ScalSet* induce a large state space, which allows to investigate the implementation's scalability. Hence, *VerSet* and *ScalSet* can be used to check the approaches' feasibility beyond *RealSet*, which is chosen to reflect the investigated WSN scenario.

It is shown in Sections 5.3.2 that parameters N_c and N_μ influence the size of the models state space. Hence, for large values* of N_c and/or N_μ, the models proposed in the following might no longer be solvable. Additionally, to avoid unattractive distinctions of cases, the application of the method of phases in Section 5.4 restricts to the case where $N_\mathrm{c} \leq N_\mu$. This case appears to be sufficient to describe the investigated scenario. A generalization that includes the case $N_\mathrm{c} > N_\mu$ is postponed to future work. Likewise, all model implementations exploit that $N_\nu = N_\mathrm{c}$ $N_\mathrm{c} \leq N_\lambda$ and for conciseness.

Apart from these restrictions, the models presented in the following can readily be applied for other parameter ranges than the ones described in Tab. 5.2. In particular, note that for the models and their evaluability, the actually chosen values of the rate parameters are less critical. That is, the models can be expected to be applicable without the need for major modifications even when choosing significantly different rate parameters.

5.3 Single-Hop Model: Mean Steady-State Performance Measures

This section is devoted to the numerical steady-state analysis of the single-hop model. The evaluation is based on Markovian analysis techniques.

5.3.1 GSPN Representation of the Single-Hop Model

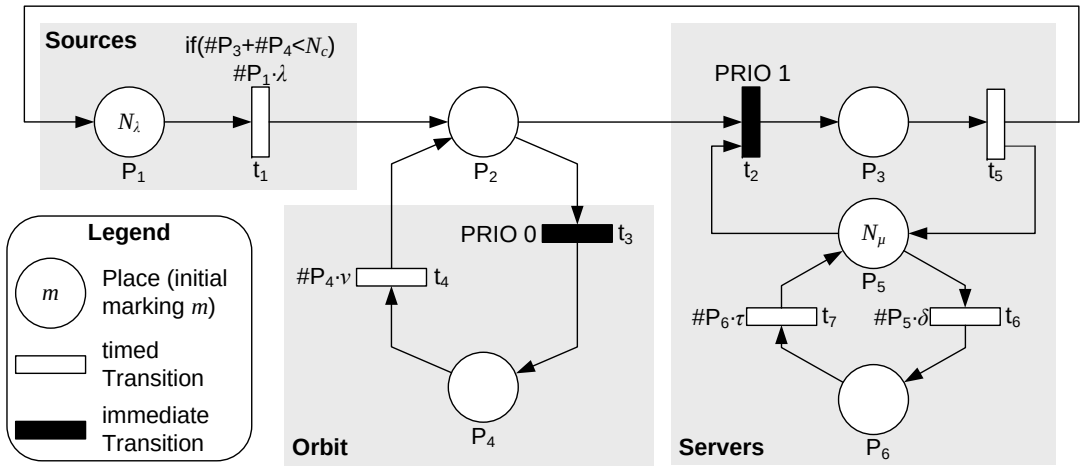


Figure 5.8: GSPN representation of the single-hop model.

* Experience on the models' scalability is gained in particular in Section 5.4.3.1 and summarized in Section 5.6.1.

As a first step to formalize the single-hop model further, the finite-source retrial queue with unreliable servers is provided graphically as generalized stochastic Petri net (GSPN; see [46, Sec. 2.3.2–2.3.3]) in Fig. 5.8.

5.3.2 Underlying Irreducible Markov Chain

The GSPN presented in Section 5.3.1 can be transformed into a CTMC (see [46, Sec. 2.3.4]). The resulting CTMC is three-dimensional and denoted by $X(t) = (N_{\mu\mathrm{f}}(t), N_{\mu\mathrm{b}}(t), N_{\nu\mathrm{b}}(t))$, where $N_{\mu\mathrm{f}}(t)$, $N_{\mu\mathrm{b}}(t)$, and $N_{\nu\mathrm{b}}(t)$ refer to the number of failed servers ($0 \leq N_{\mu\mathrm{f}}(t) \leq N_\mu$), the number of busy servers ($0 \leq N_{\mu\mathrm{b}}(t) \leq N_\mu$), and the number of jobs in the orbit ($0 \leq N_{\nu\mathrm{b}}(t) \leq N_\nu$), respectively. The CTMC's state space is denoted by $\mathbb{X}$. Note that the number $N_{\lambda\mathrm{b}}(t)$ of job-generating sources at time t is determined by $N_{\lambda\mathrm{b}}(t) = N_\lambda - N_{\mu\mathrm{b}}(t) - N_{\nu\mathrm{b}}(t)$ with $0 \leq N_{\lambda\mathrm{b}}(t) \leq N_\lambda$. The number $N_{\mu\mathrm{i}}(t)$ of idle servers at time t is given by $N_{\mu\mathrm{i}}(t) = N_\mu - N_{\mu\mathrm{b}}(t) - N_{\mu\mathrm{f}}(t)$ with $0 \leq N_{\mu\mathrm{i}}(t) \leq N_\mu$. Moreover, all involved inter-event times are exponentially distributed, and hence, obey the memoryless property. Consequently, $X(t)$ sufficiently describes the system's state at any time t.

In this thesis, the first dimension of the CTMC (represented by state variable $N_{\mu\mathrm{f}}(t)$) is called the *Layer*. Following the terminology commonly used when describing structured Markov chains, the second dimension ($N_{\mu\mathrm{b}}(t)$) is called the *Level* and the third dimension ($N_{\nu\mathrm{b}}(t)$) is called the *Phase*.

Since the state space $\mathbb{X}$ is finite and the CTMC is irreducible for all reasonable (i.e., positive) values of the involved rates, the (homogeneous) CTMC is ergodic and unique steady-state probabilities and performance measures can be derived using Markovian analysis (cf. [46, Sec. 2.1.2.2]).

A graphical representation of the CTMC and the derivation of its infinitesimal generator matrix $\mathbf{Q}$ are omitted at this point due to their complexity.*

The size $|\mathbb{X}|$ of the state space $\mathbb{X}$ can be given by

$$|\mathbb{X}| = \sum_{n_{\mu\mathrm{f}}=0}^{N_\mu} \sum_{n_{\mu\mathrm{b}}=0}^{n_{\mu\mathrm{b}}^{(n_{\mu\mathrm{f}})}} \left(N_\mathrm{c} - n_{\mu\mathrm{b}} + 1\right), \tag{5.4}$$

where

$$n_{\mu\mathrm{b}}^{(n_{\mu\mathrm{f}})} = \min(N_\mu - n_{\mu\mathrm{f}}, N_\mathrm{c}), \tag{5.5}$$

denotes the maximum number of busy servers when there are $n_{\mu\mathrm{f}}$ failed servers. The first sum of Eq. (5.4) considers the varying number of failed servers (layers), the second sum considers the number of remaining operational servers that may be busy (levels), and the summation term considers the resulting possible states of the orbit (phases), including being empty.

Figure 5.9 shows a heatmap of $|\mathbb{X}|$ for different values of N_μ and N_c, i.e., smaller, medium, and higher state space sizes are given (from top left to bottom right) on green, yellow, and red background, respectively. For the *RealSet* (i.e., $N_\mathrm{c} = N_\lambda = 7$ and $N_\mu = 9$) and *VerSet* (i.e., $N_\mathrm{c} = 5$ and $N_\mu = 5$), the size of the state space is $|\mathbb{X}| = 276$ and $|\mathbb{X}| = 91$, respectively. For the *ScalSet*, i.e., $N_\mathrm{c} = N_\lambda = 100$ and $N_\mu = 110$, the CTMC has approximately $4{\cdot}10^5$ states (highlighted by ellipse in Fig. 5.9).

* Note, however, that a reducible variant of the CTMC is discussed in more detail in Section 5.4.1.

N_c \ N_μ	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
10	506	1 166	1 826	2 486	3 146	3 806	4 466	5 126	5 786	6 446	7 106	7 766	8 426	9 086	9 746
20	1 166	3 311	5 621	7 931	10 241	12 551	14 861	17 171	19 481	21 791	24 101	26 411	28 721	31 031	33 341
30	1 826	5 621	10 416	15 376	20 336	25 296	30 256	35 216	40 176	45 136	50 096	55 056	60 016	64 976	69 936
40	2 486	7 931	15 376	23 821	32 431	41 041	49 651	58 261	66 871	75 481	84 091	92 701	101 311	109 921	118 531
50	3 146	10 241	20 336	32 431	45 526	58 786	72 046	85 306	98 566	111 826	125 086	138 346	151 606	164 866	178 126
60	3 806	12 551	25 296	41 041	58 786	77 531	96 441	115 351	134 261	153 171	172 081	190 991	209 901	228 811	247 721
70	4 466	14 861	30 256	49 651	72 046	96 441	121 836	147 396	172 956	198 516	224 076	249 636	275 196	300 756	326 316
80	5 126	17 171	35 216	58 261	85 306	115 351	147 396	180 441	213 651	246 861	280 071	313 281	346 491	379 701	412 911
90	5 786	19 481	40 176	66 871	98 566	134 261	172 956	213 651	255 346	297 206	339 066	380 926	422 786	464 646	506 506
100	6 446	21 791	45 136	75 481	111 826	153 171	198 516	246 861	297 206	348 551	400 061	451 571	503 081	554 591	606 101
110	7 106	24 101	50 096	84 091	125 086	172 081	224 076	280 071	339 066	400 061	462 056	524 216	586 376	648 536	710 696
120	7 766	26 411	55 056	92 701	138 346	190 991	249 636	313 281	380 926	451 571	524 216	597 861	671 671	745 481	819 291
130	8 426	28 721	60 016	101 311	151 606	209 901	275 196	346 491	422 786	503 081	586 376	671 671	757 966	844 426	930 886
140	9 086	31 031	64 976	109 921	164 866	228 811	300 756	379 701	464 646	554 591	648 536	745 481	844 426	944 371	1 044 481
150	9 746	33 341	69 936	118 531	178 126	247 721	326 316	412 911	506 506	606 101	710 696	819 291	930 886	1 044 481	1 159 076
160	10 406	35 651	74 896	127 141	191 386	266 631	351 876	446 121	548 366	657 611	772 856	893 101	1 017 346	1 144 591	1 273 836
170	11 066	37 961	79 856	135 751	204 646	285 541	377 436	479 331	590 226	709 121	835 016	966 911	1 103 806	1 244 701	1 388 596
180	11 726	40 271	84 816	144 361	217 906	304 451	402 996	512 541	632 086	760 631	897 176	1 040 721	1 190 266	1 344 811	1 503 356
190	12 386	42 581	89 776	152 971	231 166	323 361	428 556	545 751	673 946	812 141	959 336	1 114 531	1 276 726	1 444 921	1 618 116
200	13 046	44 891	94 736	161 581	244 426	342 271	454 116	578 961	715 806	863 651	1 021 496	1 188 341	1 363 186	1 545 031	1 732 876

Figure 5.9: Size $|\mathbb{X}|$ of state space $\mathbb{X}$ for different numbers N_μ of servers and system capacities N_c.

5.3.3 Steady-State Performance Measures

For obtaining the steady-state performance measures, Markovian steady-state analysis (cf. [46, Ch. 3]) is applied.

5.3.3.1 State Probabilities as Seen by an Outside Observer

As a first step, the state probabilities of the CTMC (as seen by an outside observer) need to be determined in steady state. They are defined by

$$\pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) = \lim_{t\to\infty} P(N_{\mu\mathrm{f}}(t) = n_{\mu\mathrm{f}}, N_{\mu\mathrm{b}}(t) = n_{\mu\mathrm{b}}, N_{\nu\mathrm{b}}(t) = n_{\nu\mathrm{b}}),$$

and collected in the steady-state probability vector

$$\begin{aligned}\boldsymbol{\pi} = (&\pi(0,0,0), \pi(0,0,1), \pi(0,0,2), \ldots,\\ &\pi(0,1,0), \pi(0,1,1), \pi(0,1,2), \ldots,\\ &\pi(1,0,0), \pi(1,0,1), \pi(1,0,2), \ldots,\\ &\pi(N_\mu, 0, N_\mathrm{c}))\,,\end{aligned}$$

in which the steady-state probabilities are sorted lexicographically by layers, levels, and phases. In general (see, e.g., [46, Ch. 3]), the steady-state probabilities are obtained by solving the set of linear equations provided by the global balance equations

$$\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}\,, \tag{5.6}$$

where the entries of $\mathbf{Q}$ are sorted to match $\boldsymbol{\pi}$, together with the normalization condition

$$\boldsymbol{\pi}\mathbf{1} = 1\,. \tag{5.7}$$

In the present case, the solution can be derived conveniently by modeling and evaluating the GSPN presented in Section 5.3.1 using the MOSEL-2 tool*.

* More information on MOSEL-2 is provided in Section 5.4.3.1, which covers the implementation of the proposed approach.

5.3.3.2 Derivation of Basic Performance Measures

Once the steady-state probabilities $\pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})$ are known for all states $(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) \in \mathbb{X}$ by solving Eq. (5.6) with Eq. (5.7), basic performance measures can be obtained as follows.

- Mean number of failed servers ($\overline{N_{\mu\mathrm{f}}}$):

$$\overline{N_{\mu\mathrm{f}}} = \sum_{(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) \in \mathbb{X}} n_{\mu\mathrm{f}}\, \pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})\,.$$

- Probability that all servers are failed (p_{failall}):

$$p_{\mathrm{failall}} = \sum_{n_{\nu\mathrm{b}}=0}^{N_\nu} \pi(N_\mu, 0, n_{\nu\mathrm{b}})\,.$$

- Mean number of operational servers ($\overline{N_{\mu\mathrm{o}}}$):

$$\overline{N_{\mu\mathrm{o}}} = N_\mu - \overline{N_{\mu\mathrm{f}}}\,.$$

- Mean number of busy servers ($\overline{N_{\mu\mathrm{b}}}$):

$$\overline{N_{\mu\mathrm{b}}} = \sum_{(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) \in \mathbb{X}} n_{\mu\mathrm{b}}\, \pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})\,.$$

- Utilization of servers (ρ):

$$\rho = \frac{\overline{N_{\mu\mathrm{b}}}}{N_\mu}\,.$$

- Mean number of idle servers ($\overline{N_{\mu\mathrm{i}}}$):

$$\overline{N_{\mu\mathrm{i}}} = \overline{N_{\mu\mathrm{o}}} - \overline{N_{\mu\mathrm{b}}}\,.$$

- Mean number of jobs in the orbit ($\overline{N_{\nu\mathrm{b}}}$):

$$\overline{N_{\nu\mathrm{b}}} = \sum_{(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) \in \mathbb{X}} n_{\nu\mathrm{b}}\, \pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})\,.$$

- Mean number of jobs in the system, i.e., at the orbit or servers ($\overline{N_{\nu\mu}}$):

$$\overline{N_{\nu\mu}} = \overline{N_{\nu\mathrm{b}}} + \overline{N_{\mu\mathrm{b}}}\,.$$

- Mean number of job-generating sources ($\overline{N_{\lambda\mathrm{b}}}$):

$$\overline{N_{\lambda\mathrm{b}}} = \sum_{(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) \in \mathbb{X}} (N_\lambda - n_{\mu\mathrm{b}} - n_{\nu\mathrm{b}})\pi(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) = N_\lambda - \overline{N_{\nu\mu}}\,. \tag{5.8}$$

- Mean overall job generation rate ($\widetilde{\lambda}$):

$$\widetilde{\lambda} = \lambda \overline{N_{\lambda\mathrm{b}}}\,. \tag{5.9}$$

- Mean number of non-blocked job-generating sources ($\overline{N_{\lambda \mathrm{bn}}}$):

$$\overline{N_{\lambda \mathrm{bn}}} = \sum_{\substack{(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \in \mathbb{X} \\ n_{\mu \mathrm{b}} + n_{\nu \mathrm{b}} < N_{\mathrm{c}}}} (N_\lambda - n_{\mu \mathrm{b}} - n_{\nu \mathrm{b}}) \pi(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \,.$$

- Mean overall job arrival rate to the queueing system, i.e., mean system throughput ($\overline{\lambda}$):

$$\overline{\lambda} = \lambda \overline{N_{\lambda \mathrm{bn}}} \,, \tag{5.10}$$

  and

$$\overline{\lambda} = \rho N_\mu \mu = \overline{N_\mu} \mu \,.$$

  This relation can be applied to partly verify the correctness of the model evaluation's implementation.

- Mean job arrival rate to an idle server ($\overline{\lambda_{\mathrm{i}}}$):

$$\overline{\lambda_{\mathrm{i}}} = \sum_{\substack{(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \in \mathbb{X} \\ n_{\mu \mathrm{b}} + n_{\nu \mathrm{b}} < N_{\mathrm{c}} \,\wedge\, n_{\mu \mathrm{f}} + n_{\mu \mathrm{b}} < N_\mu}} \frac{\lambda \left(N_\lambda - n_{\mu \mathrm{b}} - n_{\nu \mathrm{b}}\right) + \nu n_{\nu \mathrm{b}}}{N_\mu - n_{\mu \mathrm{f}} - n_{\mu \mathrm{b}}} \pi(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \,. \tag{5.11}$$

- Mean service time a job spends at the server ($\overline{T_\mu}$):

$$\overline{T_\mu} = \frac{1}{\mu} \,.$$

- Mean waiting time a job spends at the orbit ($\overline{T_\nu}$):

$$\overline{T_\nu} = \frac{\overline{N_{\nu \mathrm{b}}}}{\overline{\lambda}} \,. \tag{5.12}$$

  This derivation relies on Little's Law (see, e.g., [46, p. 245], [335, Eq. (9)]$^{\circlearrowleft}$).

- Mean response time a job spends at the orbit and server ($\overline{T_{\nu\mu}}$):

$$\overline{T_{\nu\mu}} = \frac{\overline{N_{\nu\mu}}}{\overline{\lambda}} \,. \tag{5.13}$$

  This derivation also relies on Little's Law.

- (Mean) number of job-generating sources when system is full ($\overline{N_{\lambda \mathrm{bb}}}$):

$$\overline{N_{\lambda \mathrm{bb}}} = N_{\lambda \mathrm{bb}} = N_\lambda - N_{\mathrm{c}} \,.$$

- Mean number of retrials before service ($\overline{N_{\mathrm{ret}}}$):

$$\overline{N_{\mathrm{ret}}} = \overline{T_\nu} \nu \,.$$

- Probability that the system is full (p_{full}):

$$p_{\mathrm{full}} = \sum_{\substack{(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \in \mathbb{X} \\ n_{\mu \mathrm{b}} + n_{\nu \mathrm{b}} = N_{\mathrm{c}}}} \pi(n_{\mu \mathrm{f}}, n_{\mu \mathrm{b}}, n_{\nu \mathrm{b}}) \,. \tag{5.14}$$

5.3.3.3 State Probabilities as Seen by a Generated Job

Since in the finite-source case the arrival process is non-Poisson, the *Poisson arrivals see time averages* (PASTA) theorem introduced in [341] is not generally applicable here. Hence, the state probabilities seen by a generated job are different from the ones seen by an outside observer. Also note that the arrival theorem* for finite birth-and-death–type queueing systems does not hold in the case of finite-source retrial queues (cf. [88, p. 274]). Hence, the state probabilities seen by a generated job are also different from the ones experienced by an outside observer looking at a finite system with one job less.

Instead, to derive the state probabilities seen by a generated job, the approach provided by [88, p. 274] is followed in this section while investigating the more general case including *unreliable servers* and a *finite capacity* of $N_c \leq N_\lambda$. Here, since the system can be full with probability p_{full} given by Eq. (5.14), not all generated jobs also lead to an arrival.

When assuming that all jobs are distinguishable, each system state $(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$ with known state probability $\pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$ can be decomposed into its micro-states, which allow to assess the state of each individual job. Each job can be seen to be in one of three states: (1) it is currently in generation at its source, (2) it is currently in service by a server, or (3) it is located at the orbit. Consequently, each system state $(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$ consists of $|(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})| = \binom{N_\lambda}{n_{\mu\text{b}}}\binom{N_\lambda - n_{\mu\text{b}}}{n_{\nu\text{b}}}$ micro-states, where the first binomial coefficient of this multinomial coefficient (cf. [91, p. 37]) chooses $n_{\mu\text{b}}$ jobs in service out of N_λ jobs in total and the second binomial coefficient selects $n_{\nu\text{b}}$ orbiting jobs out of $(N_\lambda - n_{\mu\text{b}})$ remaining jobs that are not in service. All other jobs are located at the sources.

Due to symmetry, all micro-states of some system state $(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$ have the same probability which is therefore given by

$$p^*(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}) = \frac{\pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})}{|(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})|} = \frac{\pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})}{\binom{N_\lambda}{n_{\mu\text{b}}}\binom{N_\lambda - n_{\mu\text{b}}}{n_{\nu\text{b}}}}. \tag{5.15}$$

Now consider a tagged job and denote by $p'(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$ the probability that the tagged job is located at some source and, at the same time, the system's state is $(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})$. Then, the number of micro-states obeying this condition is given by the multinomial coefficient $|(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})'| = \binom{N_\lambda - 1}{n_{\mu\text{b}}}\binom{N_\lambda - 1 - n_{\mu\text{b}}}{n_{\nu\text{b}}}$, where the first binomial coefficient again selects $n_{\mu\text{b}}$ jobs in service out of $N_\lambda - 1$ jobs in total (without the tagged job) and the second chooses $n_{\nu\text{b}}$ orbiting jobs out of $(N_\lambda - 1 - n_{\mu\text{b}})$ remaining jobs (except the tagged one). Consequently,

$$\begin{aligned}
p'(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}) &= |(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})'| p^*(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}) \\
&\overset{\text{Eq. (5.15)}}{=} \frac{|(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})'|}{|(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}})|} \pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}) \\
&= \frac{\binom{N_\lambda - 1}{n_{\mu\text{b}}}\binom{N_\lambda - 1 - n_{\mu\text{b}}}{n_{\nu\text{b}}}}{\binom{N_\lambda}{n_{\mu\text{b}}}\binom{N_\lambda - n_{\mu\text{b}}}{n_{\nu\text{b}}}} \pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}) \\
&= \frac{N_\lambda - n_{\mu\text{b}} - n_{\nu\text{b}}}{N_\lambda} \pi(n_{\mu\text{f}}, n_{\mu\text{b}}, n_{\nu\text{b}}).
\end{aligned} \tag{5.16}$$

* Here, the term *arrival theorem* refers to the *theorem of the distribution at arrival time* in closed queueing networks, as discussed in, e.g., [46, p. 384], [170], [276].

The steady-state probability $\hat{p}$ that the tagged job—irrespectively of the system state—is located at some source, and hence, the corresponding source is able to generate a new (blocked or unblocked) job, can then be given by

$$\begin{aligned}
\hat{p} &= \sum_{(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\in\mathbb{X}} p'(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}}) \\
&\overset{\text{Eq. (5.16)}}{=} \frac{1}{N_\lambda} \sum_{(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\in\mathbb{X}} (N_\lambda - n_{\mu\mathrm{b}} - n_{\nu\mathrm{b}})\pi(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}}) \\
&\overset{\text{Eq. (5.8)}}{=} \frac{1}{N_\lambda}\overline{N_{\lambda\mathrm{b}}} \\
&\overset{\text{Eq. (5.9)}}{=} \frac{\widetilde{\lambda}}{N_\lambda \lambda}\,.
\end{aligned}$$

The generated job's distribution, i.e., the conditional distribution of the system's state, provided that the corresponding source is able to generate it, can then be given by

$$\begin{aligned}
p_\lambda(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}}) &= \frac{p'(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})}{\hat{p}} \\
&= \frac{N_\lambda - n_{\mu\mathrm{b}} - n_{\nu\mathrm{b}}}{N_\lambda}\pi(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\,\frac{N_\lambda\lambda}{\widetilde{\lambda}} \\
&= \frac{\left(N_\lambda - n_{\mu\mathrm{b}} - n_{\nu\mathrm{b}}\right)\lambda}{\widetilde{\lambda}}\pi(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\,. \qquad (5.17)
\end{aligned}$$

Note that while this is actually the outside observer's distribution of the system state if the corresponding source is holding the tagged job, the result coincides with the generated job's distribution, since the source generates its job with exponentially distributed generation time (with rate λ), and hence, the PASTA property holds when investigating this job-generating source.

5.3.3.4 Derivation of Further Performance Measures

Once the generated job's distribution $p_\lambda(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})$ is obtained by applying Eq. (5.17), further interesting performance measures can be obtained as follows.

- Blocking probability, i.e., probability that a tagged job experiences, at generation instant, a full system and, hence, is blocked from arrival (p_{block}):

$$p_{\mathrm{block}} = \sum_{\substack{(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\in\mathbb{X} \\ n_{\mu\mathrm{b}}+n_{\nu\mathrm{b}}=N_\mathrm{c}}} p_\lambda(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\,.$$

  Note that $p_{\mathrm{block}} \neq p_{\mathrm{full}}$ in general, where p_{full} (see Eq. (5.14)) is seen by the outside observer and p_{block} is seen by a generated job. Note, for example, that $p_{\mathrm{full}} > 0$ but $p_{\mathrm{block}} = 0$ if $N_\lambda = N_\mathrm{c}$.

- Arrival probability, i.e., steady-state probability that a tagged generated job can enter the system (p_{arrival}):

$$p_{\mathrm{arrival}} = \sum_{\substack{(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}})\in\mathbb{X} \\ n_{\mu\mathrm{b}}+n_{\nu\mathrm{b}}<N_\mathrm{c}}} p_\lambda(n_{\mu\mathrm{f}},n_{\mu\mathrm{b}},n_{\nu\mathrm{b}}) = 1 - p_{\mathrm{block}}\,.$$

- Arriving job's distribution, i.e., the conditional probability that a tagged, unblocked job experiences, at arrival instant, $n_{\mu\mathrm{f}}$ failed servers, $n_{\mu\mathrm{b}}$ busy servers, and $n_{\nu\mathrm{b}}$ jobs in the orbit ($p_{\overline{\lambda}}(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})$)

$$p_{\overline{\lambda}}(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}}) = \begin{cases} \frac{p_{\lambda}(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})}{p_{\mathrm{arrival}}} & \text{if } n_{\mu\mathrm{b}} + n_{\nu\mathrm{b}} < N_{\mathrm{c}}\,, \\ 0 & \text{if } n_{\mu\mathrm{b}} + n_{\nu\mathrm{b}} = N_{\mathrm{c}}\,. \end{cases} \tag{5.18}$$

  This is again the outside observer's distribution of the system state when the tagged source generates an arrival. However, when the system is not full, the tagged source generates arrivals with exponential inter-arrival time distribution (rate λ). Hence, the PASTA property holds.

- Retrial probability, i.e., probability that a tagged, unblocked job experiences, at arrival instant, no idle servers and, hence, needs to join the orbit and conduct at least one retrial (p_{ν}):

$$p_{\nu} = \sum_{\substack{(n_{\mu\mathrm{f}}, N_{\mu} - n_{\mu\mathrm{f}}, n_{\nu\mathrm{b}}) \in \mathbb{X} \\ N_{\mu} - n_{\mu\mathrm{f}} + n_{\nu\mathrm{b}} < N_{\mathrm{c}}}} p_{\overline{\lambda}}(n_{\mu\mathrm{f}}, N_{\mu} - n_{\mu\mathrm{f}}, n_{\nu\mathrm{b}})\,. \tag{5.19}$$

- Mean number of retrials conducted by an orbit-visiting job ($\overline{N_{\mathrm{ret,o}}}$):

$$\overline{N_{\mathrm{ret,o}}} = \frac{\overline{N_{\mathrm{ret}}}}{p_{\nu}}\,.$$

5.3.4 Implementation and Verification

The model presented in Sections 5.3.1 to 5.3.3 (in the following referred to as the *proposed model*) can be implemented conveniently using the MOSEL-2 tool*. Based on this implementation, numerical results of the basic performance measures introduced in Section 5.3.3.2 can be obtained. The state probabilities seen by generated and arriving jobs as well as the performance measures derived from them as described in Section 5.3.3.3 can be obtained by extending the SPNP model generated by MOSEL-2.

The current section aims at verifying the implementation of the proposed model and the implementation of the model's evaluation. Here, *verification* refers to the process of getting confidence in the statement "the model and its evaluation are implemented correctly".† This confidence is achieved by comparing a selection of numerical model results to results presented in related work (e.g., [10], [333]$^{\circlearrowleft}$, and [88]) and to results obtained by a discrete-event simulation (DES) of the model using CPN Tools‡. This comparison also cross-verifies the implementation of the DES model.

In [10] and [333]$^{\circlearrowleft}$, unreliable finite-source retrial queues are modeled with single and multiple servers, respectively. However, the papers do present numerical results only for the

* The tool is described in more detail in Section 5.4.3.1.

† For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

‡ CPN Tools is a DES tool for modeling and simulating colored Petri nets, which may be timed (see limitations discussed in Appendix Section B.6). The tool is available at `http://cpntools.org/` (last accessed: 21 Jan. 2013). This thesis is based on Version 3.4.0 of the tool. A graphical representation of the CPN Tools model developed for this thesis is provided in Appendix Section B.7.

Table 5.3: Mean performance measures (reliable case)

Performance Measure	Result by [88]
$\overline{T_{v\mu}}$	1.8731
$\overline{T_v}$	$8.7310 \cdot 10^{-1}$
$\overline{N_v}$	4.2116
$\widetilde{\lambda} = \overline{\lambda}$	4.8237

case where servers may fail even if they are busy. Additionally, [10] restricts itself to the case where generations are stopped when all servers are failed. Hence, these results are not directly applicable for comparison.

In [333, Tab. 1]$^{\circlearrowleft}$, results are also presented for the case of reliable, homogeneous* servers. The results are obtained by applying the algorithm provided by Falin and Templeton in [88, Sec. 4.3] (referred to as *Falin–Templeton algorithm* in the following) and by numerical analysis using a MOSEL model. Using the same parameters ($N_\mu = 4, N_\lambda = 20, N_c = 24, \lambda = 0.1\,s^{-1}, \mu = 1\,s^{-1}, \nu = 1.2\,s^{-1}, \delta = 10^{-25}\,s^{-1}, \tau = 10^{25}\,s^{-1}$), the proposed model gives the results

- mean waiting time: $\overline{T_v} = 0.106\,50\,s$,
- mean number of busy servers: $\overline{N_{\mu b}} = 1.8008$,
- mean number of orbiting jobs: $\overline{N_{vb}} = 0.19177$,

which are identical to the ones presented in [333, Tab. 1]$^{\circlearrowleft}$.

The Falin–Templeton algorithm tackles the reliable case without server failures and assumes a normalization such that $\mu = 1$. For the *VerSet* (cf. Table 5.2) but $N_c = N_\nu = N_\lambda = 10$, $\delta = 10^{-25}\,s^{-1}$, and $\tau = 10^{25}\,s^{-1}$, example mean performance measures obtained by the Falin–Templeton algorithm are provided in Table 5.3. The results obtained by the proposed model are identical up to all six digits provided by MOSEL-2. Note that since the Falin–Templeton algorithm assumes sufficient system capacity, generations are never blocked, and hence the overall generation rate $\widetilde{\lambda}$ (Eq. (5.9)) equals the overall arrival rate $\overline{\lambda}$ (Eq. (5.10)).

In addition, the algorithm provides results of the arriving job's distribution, as shown exemplarily for selected states of the same parameter set in Table 5.4, and hence, facilitates further verification of the proposed model's implementation. Since arrivals are never blocked, the generated job's distribution coincides with the arriving job's distribution. Also these results are identical to the ones obtained by the proposed model and verify the correctness of the implementation and evaluation of the proposed model in the reliable case.

Using a DES model allows verification of the proposed model's evaluation in the unreliable case and in the case where $N_c < N_\lambda$. This model is implemented using CPN Tools.

For the *VerSet* but varying failure rates δ (x-axis), the result $p_{\overline{\lambda}}(1,3,1)$ is exemplarily shown in Fig. 5.10 as obtained by DES† and using the proposed model (numerical analysis with MOSEL-2 and application of Eq. (5.18)).

The DES results are provided as mean values with confidence intervals (CIs) of confidence level 99% (significance level $\alpha' = 1 - 0.99 = 0.01$) obtained from $n = 5$ independent simulation

* That is, all servers share the same characteristics, like service, failure, and repair rates.

† Here, the simulation results are obtained using a *precision factor* of $1 \cdot 10^4$. See Appendix Section B.6 for details.

Table 5.4: Example state probabilities as seen by an arriving job (reliable case)

$n_{\mu\mathrm{f}}$	$n_{\mu\mathrm{b}}$	$n_{\nu\mathrm{b}}$	$p_{\overline{\lambda}}(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})$
0	0	0	$5.3535 \cdot 10^{-9}$
0	1	4	$1.0807 \cdot 10^{-4}$
0	2	4	$1.9421 \cdot 10^{-3}$
0	3	5	$1.2769 \cdot 10^{-2}$
0	4	5	$5.8907 \cdot 10^{-2}$
0	5	3	$2.5144 \cdot 10^{-1}$
0	5	4	$3.2889 \cdot 10^{-1}$
0	5	5	0.0000

runs with approximately $2.8 \cdot 10^6$ job generations each. The DES takes up to 20 min per run*, depending on the model parameters. It can be seen in Fig. 5.10 that the relatively small number of runs is sufficient to get narrow CIs for *VerSet* results.† Due to the small n, the $t_{(1-\frac{\alpha'}{2},n-1)}$ quantile of the Student's-t distribution is applied (see [313, p. 660]), which according to [313, p. 790] amounts to $t_{(0.995,4)} = 4.604$. The CIs are then calculated via (cf. [313, p. 661])

$$\left[\overline{X} \pm t_{(0.995,4)} \frac{s}{\sqrt{n}}\right],$$

where in the present case, $\overline{X} \approx p_{\overline{\lambda}}(1,3,1)$ is the sample mean of the samples X_i ($i = 1 \ldots n$) and s is the samples' standard deviation given by (cf. [313, p. 641])

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}.$$

Figure 5.10 shows that the values of $p_{\overline{\lambda}}(1,3,1)$ obtained by numerical analysis match very well with the simulation results.

Additionally, a selection of mean performance measures as obtained from simulation (99% CIs; 5 runs) and by the proposed model are compared in Table 5.5 for *VerSet* ($\delta = 5\,\mathrm{s}^{-1}$). Again, the results of the proposed model match well with the simulation results. Hence, the implementation of the models and their evaluation can be considered to be correct.

5.4 Single-Hop Model: Derivation of Waiting Time Distribution Using Method of Phases

To answer the two questions formulated in Section 5.1.1, the distribution of the response time $T_{\nu\mu}$ needs to be characterized. The response time distribution is composed of the waiting time distribution and the service time distribution. While the service time is exponentially distributed, the waiting time distribution is yet unknown. A random variable can usually be

* On a computer with Windows XP (32 bit) operating system, one dual-core CPU (Intel® Core™ 2 Duo E6400, 2130 MHz clock speed), and 3.5 GiB of RAM.

† Also note that there is no noteworthy outlier in the 50 simulation runs shown in Fig. 5.10.

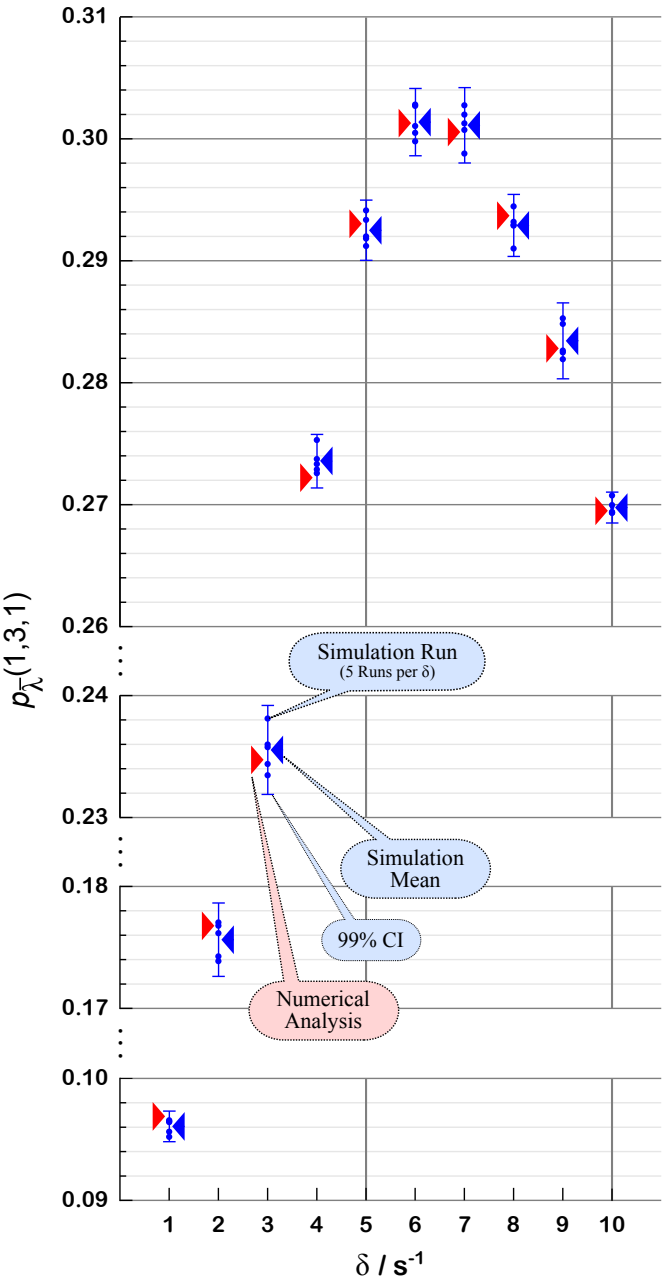


Figure 5.10: Comparing numerical to simulation results of $p_{\overline{\lambda}}(1,3,1)$ using *VerSet* (unreliable case).

characterized by its moments*. The first moment of the waiting time is given by $\overline{T_v}$ and derivable via Eq. (5.12). This section focuses on the derivation of the second and higher moments of the waiting time. To derive these moments, this thesis chooses the *Method of Phases* (cf. [169, Ch. 2]).

In the following, an EVM that is just acknowledged by the tagged node is investigated.

* If not stated explicitly otherwise, the term *moment* refers to the *moment about zero* (also known as *raw moment*) in this thesis.

Table 5.5: Comparing numerical to simulation results of selected mean performance measures (*VerSet*, unreliable case)

	Verification Results		
Performance Measure	Simulation Model (±CI/2)	Proposed Model	Unit
$\overline{T_\nu}$	$2.3397 \cdot 10^{-1} \pm 1.33 \cdot 10^{-3}$	$2.3354 \cdot 10^{-1}$	s
$\overline{T_{\nu\mu}}$	$1.2348 \pm 2.65 \cdot 10^{-3}$	1.2335	s
$\overline{N_\lambda}$	$5.1418 \pm 9.99 \cdot 10^{-4}$	5.1417	
$\overline{N_\nu}$	$9.2055 \cdot 10^{-1} \pm 6.03 \cdot 10^{-3}$	$9.1979 \cdot 10^{-1}$	
$\overline{N_\mu}$	$3.9376 \pm 6.22 \cdot 10^{-3}$	3.9385	
p_{arrival}	$1.5300 \cdot 10^{-1} \pm 5.02 \cdot 10^{-4}$	$1.5320 \cdot 10^{-1}$	

This EVM is modeled by an unblocked job that is just generated by its source. This arriving job is referred to as *tagged job* in the following. With probability $1 - p_\nu$, the tagged job is a direct job, i.e., it directly joins the servers without experiencing waiting time in the orbit, where the retrial probability p_ν is given by Eq. (5.19). With probability p_ν, the tagged job is an orbit-visiting job, i.e., it enters the orbit and conducts at least one retrial before entering a server.

The Markov process until the tagged job leaves the orbit again can be described as a *phase-type* (PH) distribution where the absorbing state refers to the state when the tagged job ultimately joins a previously idle server. In other words, the corresponding time to absorption (TTA) models the waiting time T_ν of the tagged job and is PH distributed. Therefore, the approach taken in this section is referred to as *PH approach* in thesis.

5.4.1 Underlying Reducible Markov Chain

The reducible three-dimensional CTMC $\widetilde{X}(t) = (N_{\mu\mathrm{f}}(t), N_{\mu\mathrm{b}}(t), N_{\nu\mathrm{b}}(t))$ underlying the corresponding PH distribution can be described as follows. All transient states refer to the case when there is at least the tagged job in the orbit, i.e., $N_{\nu\mathrm{b}}(t) \geq 1$. All states that refer to the case when the tagged job left the orbit are subsumed in a single absorbing state. This absorbing state is entered with retrial rate ν (successful retrial of the tagged job) from all transient states in which there is at least one idle server.

Graphical representations of this three-dimensional structured CTMC are deferred to Appendix Section B.2, due to their size. As already mentioned in Section 5.2.3, this thesis in the following limits itself to the case where $N_\mu \geq N_\mathrm{c}$ (i.e., focusing on the CTMC representations illustrated in Figs. B.6 to B.10). In particular, this avoids unattractive distinctions of cases and conserves conciseness. Note, however, that in principle the approach taken can also be applied when $N_\mu < N_\mathrm{c}$.

Following [169, Sec. 2.3], the infinitesimal generator matrix $\widetilde{\mathbf{Q}}$ of the CTMC $\widetilde{X}(t)$ can be given by

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{q} & \mathbf{T} \end{pmatrix},$$

where,

- the column vector $\mathbf{q}$ contains the transition rates from transient states to the absorbing state,
- $\mathbf{0}$ is a row vector with all elements equal to zero, i.e., there are—of course—no transitions out of the absorbing state,
- and square submatrix $\mathbf{T}$ contains the transition rates between transient states.

The size of column vector $\mathbf{q}$ is $h \times 1$, the size of the row vector $\mathbf{0}$ is $1 \times h$, and the size of the square matrix $\mathbf{T}$ is $h \times h$, where in the investigated scenario,

$$h = \sum_{n_{\mu\mathrm{f}}=0}^{N_\mu} \sum_{n_{\mu\mathrm{b}}=0}^{n_{\mu\mathrm{b}}^{(n_{\mu\mathrm{f}})}} \left(N_\mathrm{c} - n_{\mu\mathrm{b}}\right) , \tag{5.20}$$

where $n_{\mu\mathrm{b}}^{(n_{\mu\mathrm{f}})}$ is given by Eq. (5.5). Equation (5.20) is comparable to Eq. (5.4) but accounts for the tagged job residing in the orbit, i.e., the orbit must not be empty. For *RealSet*, *VerSet*, and *ScalSet*, h is 224, 70, and 393900, respectively.

Matrix $\mathbf{T}$ can be structured into $(N_\mu + 1) \times (N_\mu + 1)$ submatrices $\mathbf{A}_{ij}$ referring to the transitions from Layer i to Layer j, with $0 \leq i \leq N_\mu$, $0 \leq j \leq N_\mu$ (cp. first sum of Eq. (5.20)). That is,

$$\mathbf{T} = \begin{pmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{10} & \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{N_\mu-1,N_\mu-2} & \mathbf{A}_{N_\mu-1,N_\mu-1} & \mathbf{A}_{N_\mu-1,N_\mu} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{N_\mu,N_\mu-1} & \mathbf{A}_{N_\mu,N_\mu} \end{pmatrix} . \tag{5.21}$$

Each layer matrix A_{ij} can again be structured into $(n_{\mu\mathrm{b}}^{(i)} + 1) \times (n_{\mu\mathrm{b}}^{(j)} + 1)$ submatrices $\mathbf{B}_{ij}^{(kl)}$ containing the rates of the transitions from Level k of Layer i to Level l of Layer j, with $0 \leq k \leq n_{\mu\mathrm{b}}^{(i)}$ and $0 \leq l \leq n_{\mu\mathrm{b}}^{(j)}$ (cp. second sum of Eq. (5.20)), i.e.,

$$\mathbf{A}_{ij} = \begin{pmatrix} \mathbf{B}_{ij}^{(00)} & \mathbf{B}_{ij}^{(01)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{ij}^{(10)} & \mathbf{B}_{ij}^{(11)} & \mathbf{B}_{ij}^{(12)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{ij}^{(21)} & \mathbf{B}_{ij}^{(22)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_{ij}^{(n_{\mu\mathrm{b}}^{(i)}-1,n_{\mu\mathrm{b}}^{(j)}-2)} & \mathbf{B}_{ij}^{(n_{\mu\mathrm{b}}^{(i)}-1,n_{\mu\mathrm{b}}^{(j)}-1)} & \mathbf{B}_{ij}^{(n_{\mu\mathrm{b}}^{(i)}-1,n_{\mu\mathrm{b}}^{(j)})} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{ij}^{(n_{\mu\mathrm{b}}^{(i)},n_{\mu\mathrm{b}}^{(j)}-1)} & \mathbf{B}_{ij}^{(n_{\mu\mathrm{b}}^{(i)},n_{\mu\mathrm{b}}^{(j)})} \end{pmatrix} .$$

Each level matrix $\mathbf{B}_{ij}^{(kl)}$ is of size $(N_\mathrm{c} - k) \times (N_\mathrm{c} - j)$, which is comparable to the summation term of Eq. (5.20). The elements of a level matrix $\mathbf{B}_{ij}^{(kl)}$ are denoted with $b_{ij}^{(kl)}(m,n)$ and refer to the transition rates from state (i,k,m) to state (j,l,n).

Ignoring the absorbing state, the transient part of $\widetilde{X}(t)$ can be called skip-free in both directions of all three dimensions (levels, phases, and layers). Hence, the transient part of $\widetilde{X}(t)$ is a three-dimensional variant of a finite quasi-birth–death process (QBD, see, e.g., [169, Ch. 6], [169, Ch. 10], [46, Sec. 3.2.2]). Consequently, for $m > 0, n > 0$, there are no transitions between any states (i,k,m) and (j,l,n) if $|i-j| > 1$, $|k-l| > 1$, or $|m-n| > 1$, and hence, the corresponding rates $b_{ij}^{(kl)}(m,n)$ are equal to zero, matrices $\mathbf{T}$ and $\mathbf{A}_{ii}$ are block tridiagonal, and the matrices $\mathbf{B}_{ij}^{(kl)}$ are tridiagonal.

More precisely, all $b_{ij}^{(kl)}(m,n)$ are equal to zero except the ones detailed in the following. Since $\widetilde{\mathbf{Q}}$ is an infinitesimal generator, all main diagonal elements ($j = i, l = k, n = m$) are the negative row sums, i.e.,

$$b_{ii}^{(kk)}(m,m) = -\left(\sum_{\substack{j',l',n' \in \mathbb{X} \\ j' \neq i \vee l' \neq k \vee n \neq m}} b_{ij'}^{(kl')}(m,n') \right) - q_{i,k,m},$$

where $q_{i,k,m}$ are the elements of column vector $\mathbf{q}$, which denote the rates of successful retrials by the tagged job, i.e., the transition rates to the absorbing state, and are given by

$$q_{i,k,m} = \begin{cases} \nu & \text{if } i+k < N_\mu, \\ 0 & \text{if } i+k = N_\mu. \end{cases}$$

Note that $\mathbf{q}$ is structured similar to $\mathbf{T}$, i.e., into layers, levels, and phases.

Based on the graphical representations provided in Fig. B.6 to B.10, the transitions between the transient states covered by $\mathbf{T}$ and their rates can be summarized as follows:

Arrival to orbit:	(i,k,m)	$\xrightarrow[=(N_\lambda-k-m)\lambda]{b_{ii}^{(kk)}(m,m+1)}$	$(i,k,m+1)$	if $k+m < N_c, i+k = N_\mu$;
Arrival to server:	(i,k,m)	$\xrightarrow[=(N_\lambda-k-m)\lambda]{b_{ii}^{(k,k+1)}(m,m)}$	$(i,k+1,m)$	if $k+m < N_c, i+k < N_\mu$;
Service:	(i,k,m)	$\xrightarrow[=k\mu]{b_{ii}^{(k,k-1)}(m,m)}$	$(i,k-1,m)$	if $k > 0$;
Retrial to server*:	(i,k,m)	$\xrightarrow[=(m-1)\nu]{b_{ii}^{(k,k+1)}(m,m-1)}$	$(i,k+1,m-1)$	if $m > 1, i+k < N_\mu$;
Failure:	(i,k,m)	$\xrightarrow[=(N_\mu-i-k)\delta]{b_{i,i+1}^{(kk)}(m,m)}$	$(i+1,k,m)$	if $i+k < N_\mu$;
Repair:	(i,k,m)	$\xrightarrow[=i\tau]{b_{i,i-1}^{(kk)}(m,m)}$	$(i-1,k,m)$	if $i > 0$.

For deriving transient results, the initial probability vector $(\hat{\pi}_0, \hat{\boldsymbol{\pi}})$ is also needed to fully specify the investigated Markov process with CTMC $\widetilde{X}(t)$. The row vector $(\hat{\pi}_0, \hat{\boldsymbol{\pi}})$ is structured similar to $\widetilde{\mathbf{Q}}$. Its first element $\hat{\pi}_0$ is the probability $1 - p_\nu$ that the tagged job finds an idle server on arrival, and consequently, it does not experience any waiting time, i.e., the investigated

* Successful retrials of all orbiting jobs but the tagged one.

process $\widetilde{X}(t)$ starts in the absorbing state. All other h elements collected in row vector $\hat{\boldsymbol{\pi}}$ are denoted by $\hat{\pi}_{i,k,m}$ (with $(i,k,m) \in \mathbb{X}$ and $m > 0$) and given by

$$\hat{\pi}_{i,k,m} = \begin{cases} 0 & \text{if } i+k = N_\mu\,, \\ p_\lambda(i,k,m-1) & \text{if } i+k < N_\mu\,, \end{cases} \tag{5.22}$$

where $p_\lambda(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})$ is given by Eq. (5.17). Finally, note that $\mathbf{q} = -\mathbf{T}\mathbf{1}$, $\hat{\boldsymbol{\pi}}\mathbf{1} = p_\nu$, and $\hat{\pi}_0 = 1 - \hat{\boldsymbol{\pi}}\mathbf{1} = 1 - p_\nu$.

5.4.2 Orbit Time Distribution of Tagged Job

Based on matrix $\mathbf{T}$ and the row vector $\hat{\boldsymbol{\pi}}$, which are both derived in Section 5.4.1, the distribution function of the waiting time $T_\nu \sim PH(\hat{\boldsymbol{\pi}}, \mathbf{T})$ of the tagged job in the orbit can be given by (see [169, p. 41], which also provides a proof)

$$F(t_\nu) = 1 - \hat{\boldsymbol{\pi}} e^{\mathbf{T}t_\nu}\mathbf{1} \quad \text{for } t_\nu \geq 0\,,$$

and its density function by

$$f(t_\nu) = \hat{\boldsymbol{\pi}} e^{\mathbf{T}t_\nu}\mathbf{q} \quad \text{for } t_\nu \geq 0\,,$$

where the matrix exponential $e^{\mathbf{T}t_\nu}$ is defined by

$$e^{\mathbf{T}t_\nu} = \sum_{n=0}^{\infty} \frac{1}{n!}(\mathbf{T}t_\nu)^n = \mathbf{I} + (\mathbf{T}t_\nu) + \frac{(\mathbf{T}t_\nu)^2}{2!} + \frac{(\mathbf{T}t_\nu)^3}{3!} + \ldots\,.$$

The Laplace–Stieltjes transform (LST) of T_ν can then be given by (see [169, p. 44])

$$T_\nu^{\sim}(s) = \hat{\pi}_0 + \hat{\boldsymbol{\pi}}(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{q}\,.$$

The k-th moment of T_ν can then be obtained by calculating the k-th derivation of $T_\nu^{\sim}(s)$, i.e.,

$$E[T_\nu^k] = k!\hat{\boldsymbol{\pi}}\left(-\mathbf{T}^{-1}\right)^k\mathbf{1}\,, \tag{5.23}$$

and, in particular, the mean value $\overline{T_\nu}$ of the waiting time is

$$\overline{T_\nu} = E[T_\nu] = -\hat{\boldsymbol{\pi}}\mathbf{T}^{-1}\mathbf{1} = -\hat{\boldsymbol{\pi}}(\mathbf{T}\backslash\mathbf{1})\,, \tag{5.24}$$

and its second moment is*

$$\overline{T_\nu^2} = E[T_\nu^2] = 2\hat{\boldsymbol{\pi}}(\mathbf{T}^2)^{-1}\mathbf{1} = 2\hat{\boldsymbol{\pi}}(\mathbf{T}^2\backslash\mathbf{1})\,. \tag{5.25}$$

Here, the $\backslash$ operator refers to the *matrix left division*†, which is much more efficient‡ than inverting (powers of) $\mathbf{T}$. A more detailed discussion of the method's scalability is provided in Section 5.4.3.1.

* Remember that the absorbing state of $\widetilde{\mathbf{Q}}$ can be reached directly from each transient state of $\widetilde{\mathbf{Q}}$ with rate $\nu > 0$. Hence, from any transient state, the absorbing state can be reached almost surely after a finite time. According to [169, p. 43, Theorem 2.4.3], this implies that matrix $\mathbf{T}$ is non-singular. Since $\mathbf{T}$ is non-singular, it is invertible (cf. [53, p. 242]). Invertible matrices form a group, which is closed under the operations of matrix multiplication and inversion. This group is usually called the *general linear group* denoted with $GL(n;K)$ (cf. [94, p. 142]). For any element $\mathbf{G} \in GL(n;K)$ it can be shown that $\left(\mathbf{G}^{-1}\right)^k = \mathbf{G}^{-1}\cdot\ldots\cdot\mathbf{G}^{-1} = (\mathbf{G}\cdot\ldots\cdot\mathbf{G})^{-1} = \left(\mathbf{G}^k\right)^{-1}$ (cf. [94, p. 43], [53, p. 244]). Since $\mathbf{T} \in GL(n;K)$, $\left(\mathbf{T}^{-1}\right)^k = \left(\mathbf{T}^k\right)^{-1}$, and for $k = 2$, $\left(\mathbf{T}^{-1}\right)^2 = \left(\mathbf{T}^2\right)^{-1}$.

† Called *mldivide* operator in MATLAB/Octave.

‡ In terms of computation speed and memory consumption.

5.4.3 Implementation and Verification

The approach to derive the moments of the PH-distributed waiting time introduced in Sections 5.4.1 to 5.4.2, which is based on the method of phases, is just referred to as *PH approach* in the following. In this section, its implementation is described in detail and verified. Here, *verification* refers to the process of getting confidence in the statement "the PH approach is implemented correctly".* This confidence is achieved by comparing a selection of numerical first moment ($\overline{T_V}$) results obtained by applying the PH approach to results of the Markovian analysis applied in Section 5.3 and by comparing selected PH approach-based results of second moment ($\overline{T_V^2}$) to results of DES. Moreover, the scalability of the implementation is discussed.

5.4.3.1 Implementation

The PH approach is implemented via a set of MOSEL-2, SPNP, Perl, and Octave models, scripts, and functions. These tools are selected and employed based on their individual strengths:

- MOSEL-2 provides a convenient high-level model description language and translators to the *C-based SPN language* (CSPL, SPNP's model description language)†. MOSEL-2 is available at `http://mosel2.net.fim.uni-passau.de/` (last accessed: 21 Jan. 2013). Version 2.13 of MOSEL-2 is used in this thesis.

- SPNP comprises mature implementations of methods for generating the CTMC underlying the modeled GSPN and for numerically solving the corresponding system of global-balance equations‡. More information on SPNP and its license is available at `http://people.ee.duke.edu/~kst/software_packages.html` (last accessed: 21 Jan. 2013). This thesis applies Version 6.1 of SPNP.

- Octave is a free, mathematical software that is based on a MATLAB-like interpreted language tailored towards numerical computation. It subsumes powerful methods for matrix operations which makes it attractive for this thesis. Octave can be obtained from `http://www.gnu.org/software/octave/` (last accessed: 21 Jan. 2013). Version 3.6 is used here.

- Perl is a preferred programming language for parsing text files using regular expressions. In this thesis, Perl (Version 5) is used to link the other tools. It particularly helps to convert the output of one tool into input suitable for the next tool in the process. Perl is available at `http://www.perl.org/` (last accessed: 21 Jan. 2013).

Table 5.6 gives an overview on the overall process of deriving the waiting time's mean and second moment given by Eqs. (5.24) and (5.25), respectively. The approximate durations are given for *VerSet* and *ScalSet* in the latter two columns. The *RealSet* performs almost identically to the *VerSet*. The durations are measured on a computer with Ubuntu Linux (Desktop version 12.04 LTS, 64 bit) operating system, two 6-core CPUs (AMD® Opteron™4180, 2600 MHz

* For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

† Further model description languages are supported by MOSEL-2 but not used in this thesis. See [332]° for details.

‡ For example, the Gauss–Seidel method and successive over-relaxation (SOR). See [46, Sec. 3.5.4–3.5.5].

Table 5.6: Workflow overview of the modeling and evaluation process

Step	Description	Durations *VerSet*	Durations *ScalSet*
1	Basis: GSPN model (Fig. 5.8)	—	
2	↓ Manual conversion	(manually)	
3	MOSEL-2 model	—	
4	↓ Automatic conversion by MOSEL-2	< 1 s	
5	SPNP model	—	
6	↓ Manual extension	(manually)	
7	Extended SPNP model	—	
8	↓ Compilation of SPNP model	6 s	
9	Binary SPNP model executable	—	
10	↓ SPNP model execution	< 1 s	8 min
11	SPNP evaluation results	—	
12	↓ Parsing SPNP results (with Perl script)	< 1 s	40 s
13	Octave script and binary $\hat{\boldsymbol{\pi}}$-data file	—	
14	↓ Executing Octave script	< 1 s	*(fails after* 16 min*)*
15	Evaluation results (Table 5.7)	—	

clock speed)*, and 11.67 GiB of RAM†. In the following, the evaluation steps are discussed in more detail.

The evaluation process is based on the GSPN presented in Fig. 5.8, which represents the finite-source retrial queue with unreliable servers as introduced in Section 5.2.

The graphical model is manually converted into a textual MOSEL-2 model (Step 2) which can be converted into an SPNP model by the MOSEL-2 tool automatically (Step 4). Due to its more compact and intuitive syntax, it is usually more convenient to specify a MOSEL-2 model than specifying an SPNP model in CSPL.

* Note that the applied software is mostly unable to utilize more than one core simultaneously.
† Memory swapping to the HDD is allowed up to additional 12 GiB.

The generated SPNP model is manually extended (Step 6) for the following reasons.

1. To minimize SPNP's switching between methods for solving iteratively the linear system of global-balance equations (Eq. (5.6) with Eq. (5.7)), the Gauss–Seidel method is given priority over SOR, the desired precision is decreased from the default 10^{-6} to 10^{-5} , and the maximum number of iterations is increased from the default $2{\cdot}10^3$ to 10^4.

2. Requesting the steady-state probabilities of each single state explicitly (e.g., by using MOSEL-2's loop pre-processor) turned out to be non-scalable. Therefore, the steady-state probabilities are requested from SPNP (in form of *.prb* and *.rg* result files).

3. SPNP is instructed to ask the user for the model parameters at runtime. Therefore, Steps 1–8 only have to be done once and not for each parameter set that needs to be investigated.

4. For simplifying subsequent automation, the model parameters are restated in SPNP's result file (*.out*).

After compiling the SPNP model (Step 8), the model binary can be executed (Step 10) with varying parameter sets. The execution time depends on the provided model parameters, in particular on N_μ and N_c, which influence the size of the state space as shown in Fig. 5.9. The binary generates the CTMC underlying the described GSPN following an algorithm similar to [46, Sec. 2.3.4]. Then, it solves the global balance equations for $\boldsymbol{\pi}$ using an iterative method, cf. [46, Sec. 3.5]. This requires the majority of execution time. Finally, the binary calculates the requested basic performance measures. The results are provided in three files. The *.rg*-file describes the reachability graph and allows to map state $(n_{\mu\mathrm{f}}, n_{\mu\mathrm{b}}, n_{\nu\mathrm{b}})$ to its integer state ID, which is chosen by SPNP during CTMC generation. The *.prb*-file contains a list of state IDs together with their steady-state probabilities. The *.out*-file comprises the requested basic performance measures.

In Step 12, a Perl script is used to process the three output files. In particular, it calculates $\hat{\boldsymbol{\pi}}$ based on $\boldsymbol{\pi}$ and saves it in a binary file that can be read efficiently by Octave. Moreover, it generates an Octave script that comprises the current parameter set's data and can be directly executed using the Octave tool.

Provided with this script and $\hat{\boldsymbol{\pi}}$, Octave in Step 14 generates matrix $\mathbf{T}$ according to Eq. (5.21) and tries to calculate $\overline{T_\nu}$ and higher moments of T_ν according to Eqs. (5.23) to (5.25). However, due to the large size of $\mathbf{T}$ (393900×393900 in the *ScalSet*), this step can be considered as the evaluation method's "bottleneck" with respect to scalability. Scalability is significantly improved by treating $\mathbf{T}$ as a sparse matrix* and, as already mentioned at the end of Section 5.4.2, by avoiding inversion of $\mathbf{T}$ by all means†. Still, the current (as of August 2012) stable version (3.6.2) of Octave is using 32 bit indexing‡ and cannot address (non-sparse) arrays with more than $2.15{\cdot}10^9$ elements§. Also note that powers of sparse block-tridiagonal matrices are usually denser¶ than the original matrix. So, even if Octave is able to calculate $\mathbf{T}\backslash\mathbf{1}$, the calculation of $\mathbf{T}^2\backslash\mathbf{1}$ and beyond might still fail. In Table 5.7, an overview is given

* See densities given in Table 5.7.

† In all investigated parameter sets, the inverse's density turned out to be close to one.

‡ Switching to the (currently experimental) 64 bit indexing is postponed to future work.

§ According to Octave's `sizemax()` command.

¶ By an approximate factor of three for $\mathbf{T}^2$ according to Table 5.7.

Table 5.7: Scalability of PH approach

Param.		#Rows		Densities		Result Values		Durations	
N_μ	N_c	**Q**	$\mathbf{T},\mathbf{T}^2$	**T**	$\mathbf{T}^2$	$\overline{T_V}$ in s	$\overline{T_V^2}$ in s^2	$\overline{T_V}$	$\overline{T_V^2}$
110	100	400061	393900	$2{\cdot}10^{-5}$	$5{\cdot}10^{-5}$	*(fails)*	*N/A*	*(*16 min*)*	*N/A*
100	90	297206	292110	$2{\cdot}10^{-5}$	$6{\cdot}10^{-5}$	*(fails)*	*N/A*	*(*15 min*)*	*N/A*
90	80	213651	209520	$3{\cdot}10^{-5}$	$9{\cdot}10^{-5}$	9.35	*(fails)*	16 min	*(*6 min*)*
70	60	96441	93940	$6{\cdot}10^{-5}$	$2{\cdot}10^{-4}$	$1.25{\cdot}10^{1}$	*(fails)*	4.5 min	*(*4 min*)*
80	50	85306	82450	$7{\cdot}10^{-5}$	$2{\cdot}10^{-4}$	9.97	$3.99{\cdot}10^{2}$	2 min	5.5 min
70	50	72046	69700	$8{\cdot}10^{-5}$	$3{\cdot}10^{-4}$	$1.21{\cdot}10^{1}$	$5.33{\cdot}10^{2}$	3 min	4 min
60	50	58786	56950	$1{\cdot}10^{-4}$	$3{\cdot}10^{-4}$	$1.49{\cdot}10^{1}$	$7.44{\cdot}10^{2}$	2 min	3.5 min
50	40	32431	31160	$2{\cdot}10^{-4}$	$6{\cdot}10^{-4}$	$1.82{\cdot}10^{1}$	$1.03{\cdot}10^{3}$	25 s	46 s
50	30	20336	19220	$3{\cdot}10^{-4}$	$9{\cdot}10^{-4}$	$1.75{\cdot}10^{1}$	$9.64{\cdot}10^{2}$	3 s	13 s
30	20	5621	5180	$1{\cdot}10^{-3}$	$3{\cdot}10^{-3}$	$3.12{\cdot}10^{1}$	$2.58{\cdot}10^{3}$	0 s	2 s
10	10	506	440	$1{\cdot}10^{-2}$	$3{\cdot}10^{-2}$	$9.97{\cdot}10^{1}$	$2.19{\cdot}10^{4}$	0 s	0 s
9	7	276	224	$2{\cdot}10^{-2}$	$6{\cdot}10^{-2}$	$5.56{\cdot}10^{1}$	$6.21{\cdot}10^{3}$	0 s	0 s
5	5	91	70	$6{\cdot}10^{-2}$	$2{\cdot}10^{-1}$	$2.34{\cdot}10^{-1}$	$5.17{\cdot}10^{-1}$	0 s	0 s

on different parameter sets and their evaluation success. The model parameters N_μ and N_c are provided in the table. With exception of the last two rows of the table, all other parameters are chosen according to the *ScalSet*, where $N_\lambda = N_\nu = N_c$. The last two rows treat the *RealSet* ($N_\mu = 9$, $N_\lambda = N_\nu = N_c = 7$) and *VerSet* ($N_\lambda = 10$, $N_\mu = N_c = N_\nu = 5$) with the corresponding set's rate parameters. It can be seen that the second moment of T_V can be derived up to a maximum of approximately $8.5{\cdot}10^3$ states in the irreducible CTMC.

To derive the moments of T_V for larger models, an approximation is proposed in Section 5.5. Since the approximation's derivation also relies on results obtained using the PH approach described in here, the current section concludes with a verification of the PH approach's correct implementation.

5.4.3.2 Verification

The implementation of the PH approach is now verified* in two ways. First, Eq. (5.24) and (5.12) both provide means for calculating the waiting time $\overline{T_V}$. The former is based on Markovian analysis (here provided by the tools MOSEL-2 and SPNP) and Little's Law. The latter is based on the PH approach. Comparing the results allows to cross-verify the two approaches. Second, DES is applied using CPN Tools. The corresponding CPN Tools model of the finite-source retrial queue with unreliable servers is illustrated in Appendix Section B.7.

The result comparison is shown in Table 5.8 ($\overline{T_V}$) and Table 5.9 ($\overline{T_V^2}$) for the same parameter sets as chosen for Table 5.7. The simulation results are obtained similarly† to the ones presented in Section 5.3.4. The tables show that, in general, the numerical results are very close to the

* Here, the term *verification* refers to the process of getting confidence in the statement "the PH approach is implemented correctly". Section 6.2 provides more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis.

† Note that due to the high rates (e.g., $\delta = 2500$) involved in the *RealSet* (second to last row; $N_\mu = 9$, $N_c = 7$), the simulation's precision factor is increased to 10^5 for this parameter set. In return, the number of simulation runs is increased to ten to achieve narrow CIs. The interested reader is referred to the more detailed discussion on CPN Tools' precision of the time parameter provided in Appendix Section B.6.

Table 5.8: Verification of $\overline{T_v}$ obtained by applying the PH approach

Param.		$\overline{T_v}$ (in s)			
N_μ	N_c	Eq. (5.12)	Eq. (5.24)	Simulation (±CI/2)	
110	100	7.2960	N/A	7.3262	$\pm$ $4.20 \cdot 10^{-2}$
100	90	8.2205	N/A	8.2485	$\pm$ $3.48 \cdot 10^{-2}$
90	80	9.3469	9.3469	9.3745	$\pm$ $6.19 \cdot 10^{-2}$
70	60	$1.2542 \cdot 10^{1}$	$1.2542 \cdot 10^{1}$	$1.2561 \cdot 10^{1}$	$\pm$ $4.26 \cdot 10^{-2}$
80	50	9.9722	9.9722	$1.0002 \cdot 10^{1}$	$\pm$ $7.04 \cdot 10^{-2}$
70	50	$1.2075 \cdot 10^{1}$	$1.2075 \cdot 10^{1}$	$1.2111 \cdot 10^{1}$	$\pm$ $7.71 \cdot 10^{-2}$
60	50	$1.4918 \cdot 10^{1}$	$1.4918 \cdot 10^{1}$	$1.4951 \cdot 10^{1}$	$\pm$ $7.51 \cdot 10^{-2}$
50	40	$1.8220 \cdot 10^{1}$	$1.8220 \cdot 10^{1}$	$1.8305 \cdot 10^{1}$	$\pm$ $8.38 \cdot 10^{-2}$
50	30	$1.7505 \cdot 10^{1}$	$1.7505 \cdot 10^{1}$	$1.7514 \cdot 10^{1}$	$\pm$ $1.99 \cdot 10^{-1}$
30	20	$3.1246 \cdot 10^{1}$	$3.1246 \cdot 10^{1}$	$3.1435 \cdot 10^{1}$	$\pm$ $3.02 \cdot 10^{-1}$
10	10	$9.9735 \cdot 10^{1}$	$9.9735 \cdot 10^{1}$	$9.9545 \cdot 10^{1}$	$\pm$ 3.42
9	7	$5.5632 \cdot 10^{1}$	$5.5631 \cdot 10^{1}$	$5.6870 \cdot 10^{1}$	$\pm$ 1.79
5	5	$2.3354 \cdot 10^{-1}$	$2.3354 \cdot 10^{-1}$	$2.3385 \cdot 10^{-1}$	$\pm$ $8.00 \cdot 10^{-4}$

Table 5.9: Verification of $\overline{T_v^2}$ obtained by applying the PH approach

Param.		$\overline{T_v^2}$ (in s^2)		
N_μ	N_c	Eq. (5.25)	Simulation (±CI/2)	
110	100	N/A	$2.5467 \cdot 10^{2}$	$\pm$ 1.75
100	90	N/A	$3.0251 \cdot 10^{2}$	$\pm$ 3.15
90	80	N/A	$3.6405 \cdot 10^{2}$	$\pm$ 4.94
70	60	N/A	$5.6725 \cdot 10^{2}$	$\pm$ 4.14
80	50	$3.9856 \cdot 10^{2}$	$4.0132 \cdot 10^{2}$	$\pm$ 6.93
70	50	$5.3344 \cdot 10^{2}$	$5.3751 \cdot 10^{2}$	$\pm$ 7.94
60	50	$7.4397 \cdot 10^{2}$	$7.4741 \cdot 10^{2}$	$\pm$ 7.50
50	40	$1.0291 \cdot 10^{3}$	$1.0407 \cdot 10^{3}$	$\pm$ $1.64 \cdot 10^{1}$
50	30	$9.6362 \cdot 10^{2}$	$9.6316 \cdot 10^{2}$	$\pm$ $2.25 \cdot 10^{1}$
30	20	$2.5797 \cdot 10^{3}$	$2.6037 \cdot 10^{3}$	$\pm$ $1.47 \cdot 10^{1}$
10	10	$2.1911 \cdot 10^{4}$	$2.2112 \cdot 10^{4}$	$\pm$ $1.47 \cdot 10^{3}$
9	7	$6.2122 \cdot 10^{3}$	$6.4793 \cdot 10^{3}$	$\pm$ $4.84 \cdot 10^{2}$
5	5	$5.1668 \cdot 10^{-1}$	$5.1826 \cdot 10^{-1}$	$\pm$ $6.09 \cdot 10^{-3}$

simulation results and lie within the corresponding CI.

The above discussions show that for numerical analysis it is difficult to get results of $\overline{T_v^2}$ for finite-source unreliable retrial queues with a high number of states (i.e., high N_μ and/or N_c). When applying DES (using version 3.4.0 of CPN Tools), special care has to be taken if there are high transition rates. Even if a more suitable implementation of the DES can be achieved in future work, there is little hope for getting fully rid of problems caused by rare events (see e.g., [277]). These are usually pending whenever rate parameters differ by several magnitudes, as given in the investigated scenario.

To tackle these issues, a further approach is proposed in Section 5.5. It aims at providing an approximation of the waiting time distribution, which serves as an alternative to DES and to the PH approach especially for models where high transition rates meet a large state space.

5.5 Single-Hop Model: Derivation of Waiting Time Distribution Using Gamma Approximation

Section 5.4 shows that the derivation of the second and higher moments of the waiting time T_v based on numerical analysis and the PH approach suffers from state space explosion for large N_μ and N_c. Additionally, while DES (using CPN Tools) is insusceptible to state space explosion, the results are less reliable for high transition rates due to the rounding error induced to the sampled sojourn times and bounded simulation length and in the presence of rare events.

Therefore, an approximation of the waiting time distribution is proposed in this section, which is based solely on mean steady-state performance measures obtainable by the standard numerical analysis shown in Section 5.3.3. That is, the goal is to be able to obtain a feasible approximation already in Step 12 of the evaluation workflow described in Table 5.6.

5.5.1 Selecting Candidate Distributions

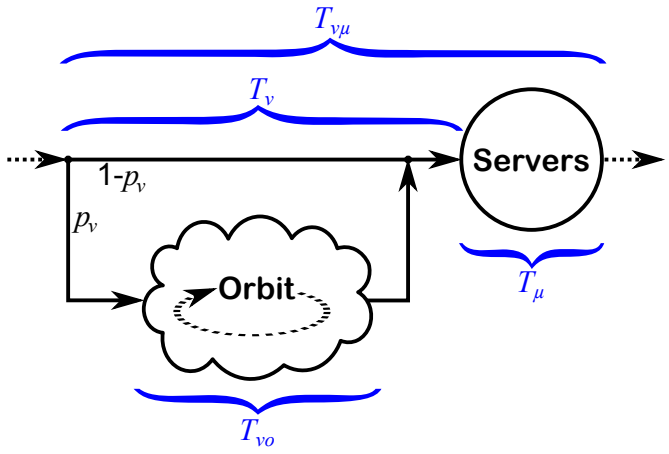


Figure 5.11: Schematic illustration of arriving jobs' behavior.

In Fig. 5.11, the basic behavior of arriving jobs is illustrated schematically. With retrial probability p_v given by Eq. (5.19), an arriving job finds no idle server, and hence, is forced to join the orbit. Such an *orbit-visiting job* experiences some random waiting time T_{vo} with mean $\overline{T_{vo}}$.

Table 5.10: Model parameters of *ExtraSet*

Parameter	N_λ	N_c	N_ν	N_μ	λ	ν	μ	τ	δ
Value	100	10	10	10	$1\,s^{-1}$	$0.5\,s^{-1}$	$1\,s^{-1}$	$1\,s^{-1}$	$1\,s^{-1}$

With probability $1-p_\nu$, arriving jobs directly join a server without having to wait. These *direct jobs* do not experience any waiting time $T_{\nu d} = \overline{T_{\nu d}} = 0$.

The mean waiting time $\overline{T_\nu}$ of *all* arriving jobs can consequently be given by

$$\overline{T_\nu} = (1-p_\nu)\,\overline{T_{\nu d}} + p_\nu \overline{T_{\nu o}} = p_\nu \overline{T_{\nu o}}\,. \tag{5.26}$$

Using results of research on finite mixture densities (see, e.g., [95, p. 11]), higher moments of T_ν can be obtained similarly to Eq. (5.26), i.e.,

$$\overline{T_\nu^i} = (1-p_\nu)\,\overline{T_{\nu d}^i} + p_\nu \overline{T_{\nu o}^i} = p_\nu \overline{T_{\nu o}^i}\,, \tag{5.27}$$

where $i \in \mathbb{N}$. That is, if the moments of the orbit-visiting jobs' waiting time distribution can be derived, also the moments of the waiting time distribution of all arriving jobs can be calculated. Deriving the distribution of $T_{\nu o}$ is hence the focus in the following.

Figure 5.12 shows histograms of $T_{\nu o}$ for different simulation scenarios. For each histogram, the number of histogram classes is set to the (rounded) square root of the number of samples, as suggested in [34, p. 327]. Heavy tails are truncated to show the more interesting parts in higher detail.

Figures 5.12(a), (b), and (c), show the histograms for a single simulation run of parameter sets *RealSet*, *ScalSet*, and *VerSet*, respectively. Remember that these parameter sets have been defined in Section 5.2.8. Figure 5.12(e) examines an extra parameter set (*ExtraSet*), which is defined in Table 5.10.*

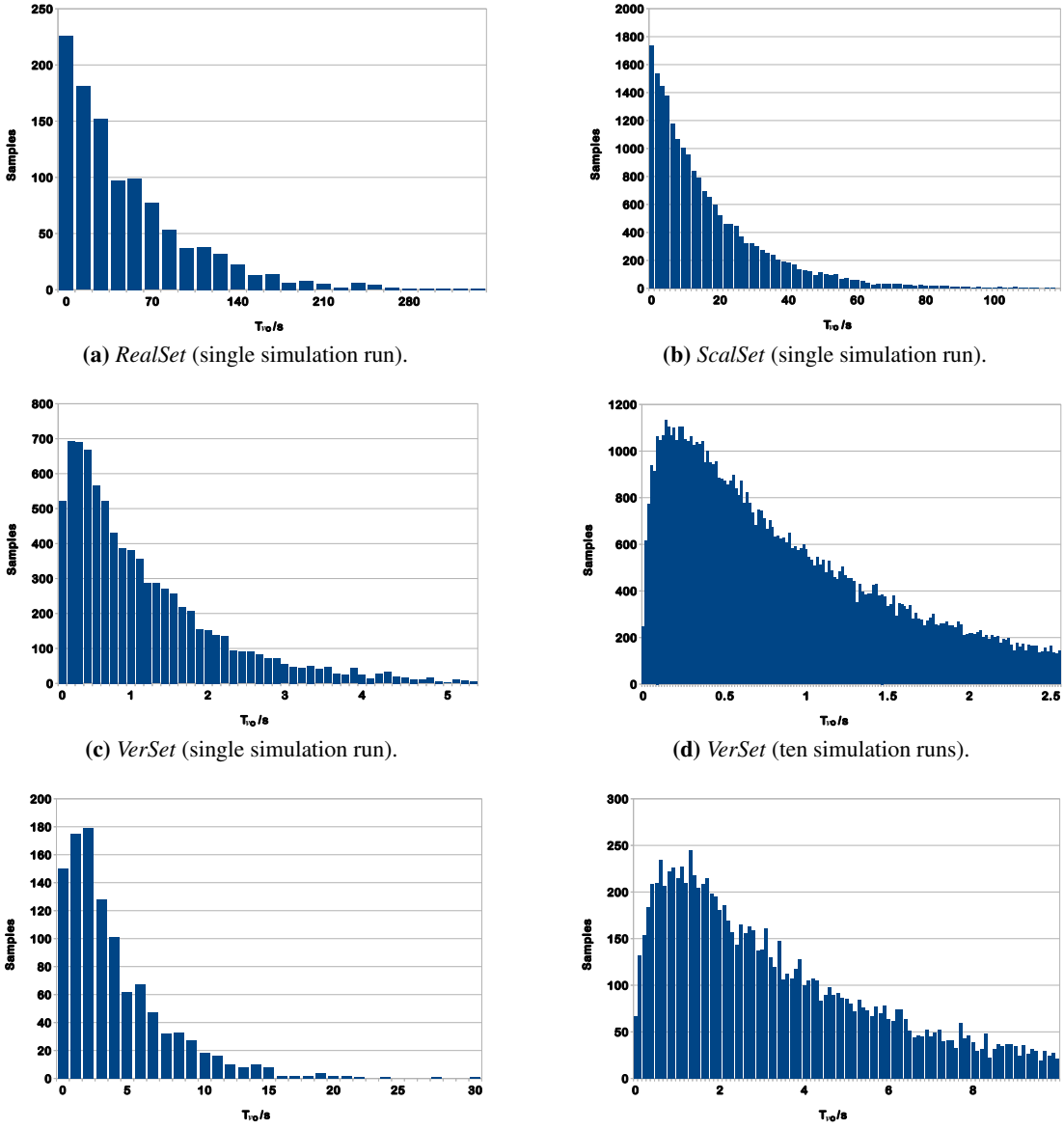
The histograms in Figs. 5.12(a) and (b) seem to point to the probability density function (PDF) of an exponential distribution. The histograms in Figs. 5.12(c) and (e) hint towards a more gamma distribution-like shape. This is further confirmed by Figs. 5.12(d), and (f), which are based on the $T_{\nu o}$ data obtained by ten independent simulation runs of *VerSet* and the *ExtraSet*, respectively, and allow a closer look at the histograms' slope close to $T_{\nu o} = 0$.

In the following, a stochastic distribution is sought after that allows to approximate the distribution of $T_{\nu o}$. There is quite a number of alternative continuous, unimodal, right-skewed distributions† that are defined on the interval $[0,\infty)$ (i.e., left bounded) and whose PDF may take the form suggested by Figs. 5.12(c) to (f).‡ Out of these, the Erlang distribution appears to be attractive due to its stochastic interpretation, which is related to the PH model of Section 5.4 as described in the following.

* This *ExtraSet* turns out to exhibit a comparably large value of the value α_γ, which is discussed in Section 5.5.2.

† Graphical representations of PDFs are given for a large set of probability distributions in, e.g., [197, 317].

‡ For example, the Burr (and its special case Fisk/log-logistic), gamma (and its special cases Erlang, exponential, and chi-squared), Dagum/inverse-Burr/kappa, F/Fisher–Snedecor, fatigue life, folded-normal, inverse-Gaussian/inverse-normal, log-gamma, log-Laplace, log-normal/Cobb–Douglas, Nakagami (and its special cases chi, half-normal and Rayleigh), Pearson, and Weibull/Fréchet (which also includes Exponential and Rayleigh as special cases) distributions.



(a) *RealSet* (single simulation run).

(b) *ScalSet* (single simulation run).

(c) *VerSet* (single simulation run).

(d) *VerSet* (ten simulation runs).

(e) *ExtraSet* (single simulation run).

(f) *ExtraSet* (ten simulation runs).

Figure 5.12: Histograms of $T_{\nu o}$ obtained by DES.

In principle, an approximate representation of the three-dimensional PH process introduced in Section 5.4.1 by a one-dimensional PH process is striven for. Each orbit-visiting job conducts an integer number of retrials before it finds an idle server. The time between two consecutive retrials of a tagged job is exponentially distributed with mean $1/\nu$. Hence, the waiting time of the tagged job is the sum of several exponential distributions. Consequently, if one knew the actual number of retrials $N_{\text{ret,o}}$ of a tagged, orbit-visiting job, its waiting time could be expected being Erlang* distributed with parameters ν (rate per phase) and $k = N_{\text{ret,o}}$ (number

* The chosen representation of the Erlang distribution follows [46, p. 23], i.e., k phases and a rate of $k\mu$ per phase lead to an overall mean of $1/\mu$ and a variance of $1/(k\mu^2)$.

of phases). However, the actual number of retrials of a tagged job is not constant but depends on various factors, including the current number of available servers and the retrial order of the currently orbiting jobs.

The next step is to approximately derive the parameters of the one-dimensional PH process. For this, discussion is switched from the Erlang distribution (with requires integer shape parameter k) to the gamma distribution. The gamma distribution can be seen as a generalization of the Erlang distribution by allowing for a real-valued shape parameter, which is denoted by α_γ from now on. This provides more flexibility when fitting the parameter.

The assumption of a gamma distribution is also supported by fitting actual samples of the orbit-visiting jobs' waiting time obtained by DES to common distributions and comparing the quality of their fit, e.g., based on the Bayesian information criterion (BIC, see [270]). This can be carried out conveniently, e.g., with the MATLAB function ALLFITDIST()*, which identifies the gamma distribution to fit best (in comparison to the continuous distributions beta, Birnbaum–Saunders, exponential, extreme value, generalized extreme value, generalized Pareto, inverse Gaussian, logistic, log-logistic, log-normal, Nakagami, normal, Rayleigh, Rician, t location-scale, and Weibull).

5.5.2 Approximating the Parameters of Gamma-Distributed Waiting Time

According to the discussions in Section 5.5.1, the suitability of modeling the waiting time of orbit-visiting jobs by a gamma distribution is investigated in more detail in the following. In this thesis, the gamma distribution's representation follows [46, p. 25], i.e., its PDF is given for $x > 0$ by

$$f_\gamma(x) = \frac{\alpha_\gamma \mu_\gamma (\alpha_\gamma \mu_\gamma x)^{\alpha_\gamma - 1}}{\Gamma(\alpha_\gamma)} e^{-\alpha_\gamma \mu_\gamma x},$$

where $\Gamma\left(\alpha_\gamma\right)$ is the gamma function defined for $\alpha_\gamma > 0$ by

$$\Gamma\left(\alpha_\gamma\right) = \int_0^\infty y^{\alpha_\gamma - 1} e^{-y} \, \mathrm{d}y. \tag{5.28}$$

This gamma distribution has rate parameter $\mu_\gamma > 0$ (i.e., mean $1/\mu_\gamma$) and shape parameter $\alpha_\gamma < 0$ (i.e., variance $1/(\alpha_\gamma \mu_\gamma^2)$).

The rate parameter μ_γ can be estimated via

$$\mu_\gamma = \frac{1}{\overline{T_{v\mathrm{o}}}} \overset{\text{Eq. (5.26)}}{=} \left(\frac{\overline{T_v}}{p_v}\right)^{-1} = \frac{p_v}{\overline{T_v}}, \tag{5.29}$$

where $\overline{T_v}$ can be obtained quite easily by numerical analysis (see Eq. (5.12)).

The shape parameter α_γ can, in principle, be estimated via (see [46, p. 25])

$$\alpha_\gamma = \frac{1}{c^2_{T_{v\mathrm{o}}}}, \tag{5.30}$$

* Provided by Mike Sheppard at `http://www.mathworks.com/matlabcentral/fileexchange/34943-fit-all-valid-parametric-probability-distributions-to-data` (Last accessed on 9 December 2012) in February 2012; uses BIC by default.

where $c^2_{T_{vo}}$ is the squared coefficient of variation of an orbit-visiting job's waiting time and can be derived via (see Eqs. (1.19) and (1.20) of [46, p. 19])

$$\begin{aligned} c^2_{T_{vo}} &= \frac{\mathrm{Var}(T_{vo})}{\overline{T_{vo}}^2} = \frac{\overline{T_{vo}^2} - \overline{T_{vo}}^2}{\overline{T_{vo}}^2} = \frac{\overline{T_{vo}^2}}{\overline{T_{vo}}^2} - 1 \\ &\overset{\text{Eq. (5.26)}}{=} \frac{\overline{T_{vo}^2}}{\left(\frac{\overline{T_v}}{p_v}\right)^2} - 1 = \frac{p_v^2 \overline{T_{vo}^2}}{\overline{T_v}^2} - 1 \\ &\overset{\text{Eq. (5.27)}}{=} \frac{p_v^2 \frac{\overline{T_v^2}}{p_v}}{\overline{T_v}^2} - 1 = \frac{p_v \overline{T_v^2}}{\overline{T_v}^2} - 1 = \frac{p_v \overline{T_v^2} - \overline{T_v}^2}{\overline{T_v}^2}. \end{aligned} \tag{5.31}$$

Putting $c^2_{T_{vo}}$ of Eq. (5.31) into Eq. (5.30) results in

$$\alpha_\gamma = \frac{1}{c^2_{T_{vo}}} = \frac{\overline{T_v}^2}{p_v \overline{T_v^2} - \overline{T_v}^2}. \tag{5.32}$$

Using Eq. (5.32), α_γ can be estimated based on p_v and $\overline{T_v}$, which are available by numerical analysis, and on the second moment $\overline{T_v^2}$ of the waiting time, which can be obtained from the PH approach, i.e., by applying Eq. (5.25), or from DES.

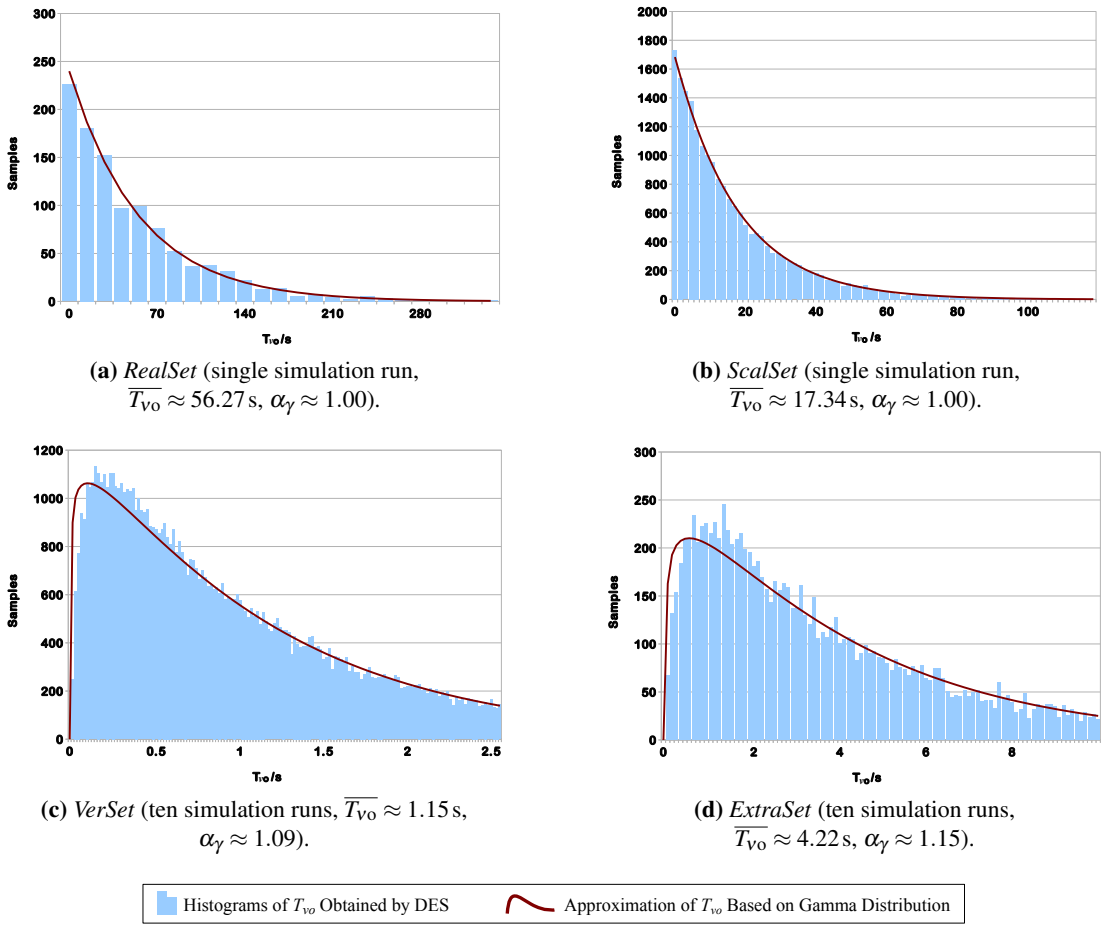
In Fig. 5.13, the resulting gamma distributions are compared to the histograms already discussed in Fig. 5.12. It can be seen that the gamma distribution (solid lines) reflects the corresponding histograms (bar graphs) quite well for each of the four investigated parameter sets as illustrated in Figs. 5.13 (a) to (d) . A rigorous, quantitative investigation of the approximation's quality and limits is omitted at this point. Within the scope of this thesis, the obtained approximation is considered as exact enough since other assumptions (like exponentiality of retrial time and service time; see Section 5.1.2) likely have more effect on the validity of the quantitative results, i.e., on their closeness to the expected results of a real-world implementation of the proposed WSN. The corresponding required validation, which aims at getting confidence in the statement that the single-hop model is sufficiently reflecting the aspired real system, is not in the focus of this thesis, since more detailed representations of the proposed WSN are not available for comparison yet.* In any case, a more rigorous investigation of the approximation's quality is recommended when applying the approximation in a more general context or under a broader range of parameters.

Let $\alpha_\gamma^{(num)}$ be the α_γ that can be obtained by deriving $\overline{T_v}$ and $\overline{T_v^2}$ by applying the PH approach and using these as input to Eq. (5.32). In Fig. 5.14, $\alpha_\gamma^{(num)}$ is shown for varying failure rate δ in the three different parameter sets *RealSet*, *VerSet*, and *ExtraSet*.

It can be seen that $\alpha_\gamma^{(num)}$ tends to one, i.e., towards an exponential distribution for large δ, in each parameter set. However, for small δ, assuming $\alpha_\gamma = 1$ does not reflect the waiting time distribution well.

To get a good approximation of α_γ based on just model parameters and basic mean steady-state performance measures, but without having to follow the PH approach to derive higher

* See also Section 6.2, which gives more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis.



(a) *RealSet* (single simulation run, $\overline{T_{vo}} \approx 56.27\,\text{s}$, $\alpha_\gamma \approx 1.00$).

(b) *ScalSet* (single simulation run, $\overline{T_{vo}} \approx 17.34\,\text{s}$, $\alpha_\gamma \approx 1.00$).

(c) *VerSet* (ten simulation runs, $\overline{T_{vo}} \approx 1.15\,\text{s}$, $\alpha_\gamma \approx 1.09$).

(d) *ExtraSet* (ten simulation runs, $\overline{T_{vo}} \approx 4.22\,\text{s}$, $\alpha_\gamma \approx 1.15$).

(e) Legend.

Figure 5.13: Comparing histograms of T_{vo} obtained by DES with gamma distributed T_{vo}.

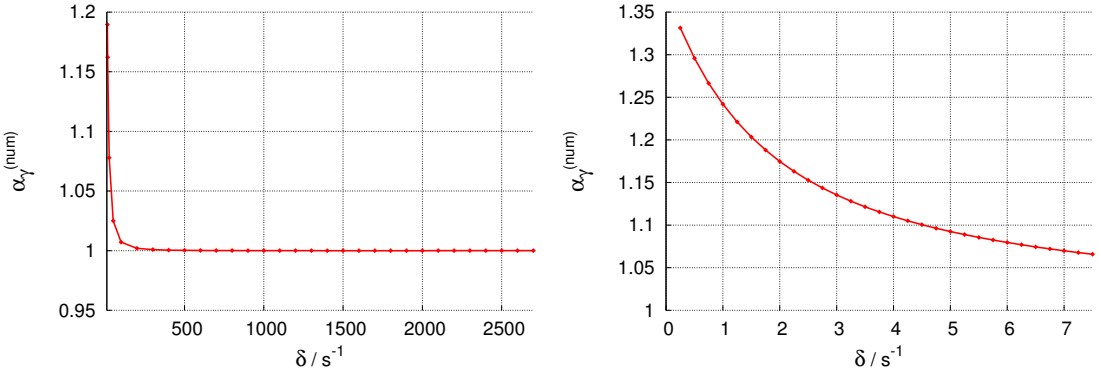
moments of the waiting time, symbolic regression is carried out by applying the tool Eureqa*. For this, the model is evaluated for 710 parameter sets, of which 606 are numerically analyzed and 104 are simulated using CPN Tools. The total parameter ranges of these sets are summarized in Table 5.11. They cover the ranges foreseen in Table 5.2 and beyond.

The evaluations provide mean performance measures and $\overline{T_v^2}$. This allows to derive α_γ based on Eq. (5.32). The derived α_γ lie in the range of approximately 1.0 to 1.7. For each parameter set, all model parameters, a selection of mean performance measures†, and α_γ serve as input to Eureqa which then searches for dependencies of α_γ on the other provided values by applying symbolic regression.

Eureqa is able to find a large set of possible approximations of α_γ, which have different complexity and accuracy. Symbolic regression not only searches for the *parameters* of the

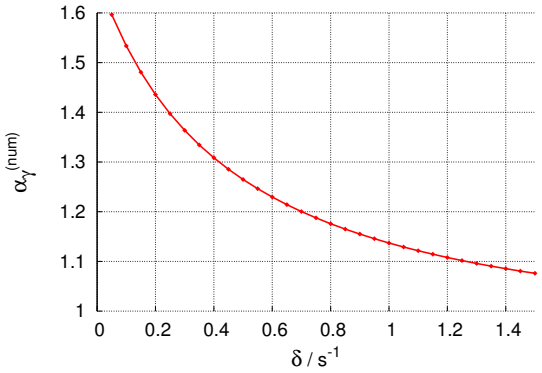
* Eureqa is a software tool that implements genetic programming-based symbolic regression (see [263]) and allows to conveniently search for mathematical relationships in the provided data. The tool is available at `http://creativemachines.cornell.edu/eureqa` (last accessed: 21 Jan. 2013). This thesis uses version 0.98.1 Beta (also known as *Eureqa II* or *Formulize*).

† $\overline{N_{\mu b}}$, $\overline{N_{vb}}$, $\overline{N_{\mu f}}$, $\overline{N_{\lambda bn}}$, $\overline{\lambda}$, p_{full}, $\hat{p}$, p_v, p_{arrival}, $\overline{T_v}$, and $\overline{N_{\text{ret,o}}}$.



(a) $\alpha_\gamma^{(num)}$ in *RealSet*.

(b) $\alpha_\gamma^{(num)}$ in *VerSet*.



(c) $\alpha_\gamma^{(num)}$ in *ExtraSet*.

Figure 5.14: Investigation of $\alpha_\gamma^{(num)}$ for varying δ.

Table 5.11: Parameter ranges covered by the input to Eureqa

Parameter	N_λ	N_c	N_ν
Range	$1 \ldots 100$	$1 \ldots 100$	$1 \ldots 100$
Parameter	N_μ	λ	ν
Range	$1 \ldots 110$	$1{\cdot}10^{-3} \ldots 10\,\mathrm{s}^{-1}$	$0.1 \ldots 1{\cdot}10^{3}\,\mathrm{s}^{-1}$
Parameter	μ	τ	δ
Range	$0.1 \ldots 1.3{\cdot}10^{3}\,\mathrm{s}^{-1}$	$0.01 \ldots 1000\,\mathrm{s}^{-1}$	$0.01\tau \ldots 25{\cdot}10^{3}\tau$

approximation that minimizes the chosen error metric, but also for the approximation's *form* itself. The solution's error is calculated for an automatically retained subset of data sets not used for training but only for validation. The complexity of a solution is measured by the number of elements (operators, parameters, and constants) used, where for each element type an integer complexity can be assigned.*

* This thesis uses Eureqa's default complexity assignments.

Table 5.12: Overview of conducted Eureqa searches

Search	Objective (Error Metric)	Number of Generations	Number of Formula Evaluations	Stability	Maturity
(a)	Mean Absolute Error	$2.1 \cdot 10^8$	$1.8 \cdot 10^{13}$	33 %	92 %
(b)	Worst Case; Maximum Error	$2.3 \cdot 10^8$	$4.8 \cdot 10^{12}$	37 %	99 %
(c)	Hybrid Correlation/Error Metric	$2.4 \cdot 10^8$	$4.9 \cdot 10^{12}$	81 %	88 %

In this thesis, Eureqa is applied to search for solutions using three different search objectives*, i.e., by minimizing three different error metrics. The chosen objectives are listed in Table 5.12 together with their achieved performance indicators. In particular†, the *Number of Generations* is the total number of candidate solutions generated before (manually) stopping the search, and the *Number of Formula Evaluations* is the "total number of times any solution has been compared to a data point" before manually stopping the search. *Stability* and *Maturity* are "experimental statistics" that allow to estimate the solution's confidence. Usually, "[w]hen both are high (>50–90%) it is a strong indication of confidence in the current results." In the present setup, high *Maturity* values can be achieved consistently. The *Stability*, however, fails to achieve and maintain high values. The *Stability* values shown for each search in Table 5.12 are the highest ones observed in the final third of all generations. Note that, while Search (c) achieved a high *Stability*, its results are not distinctly better than the other two, reportedly less stable searches' results. Consequently, the expressiveness of the experimental *Stability* value stands to reason.

Figures 5.15(a) and (b) plot the mean absolute and maximum errors (y-axis), respectively, vs. complexity (x-axis) of the best solutions ("Pareto frontiers", cf. [207, p. 141] or [222, p. 122]) found by the three searches listed in Tab. 5.12. However, the complexity of the solutions is not a major selection criterion in this thesis, since any closed-form approximation of α_γ can be assumed to be computationally less complex than the PH approach or DES. To select the best solution with respect to the mean absolute and maximum error, Fig. 5.16 plots the maximum error (y-axis) vs. the mean absolute error (x-axis) of the found solutions. The data point size reflects the solution's complexity. From the solutions, five examples (denoted with A1 to A5) are selected, as indicated in Figs. 5.15 and 5.16. For these examples, more details are shown in Table 5.13. Their corresponding equations approximating α_γ are given by

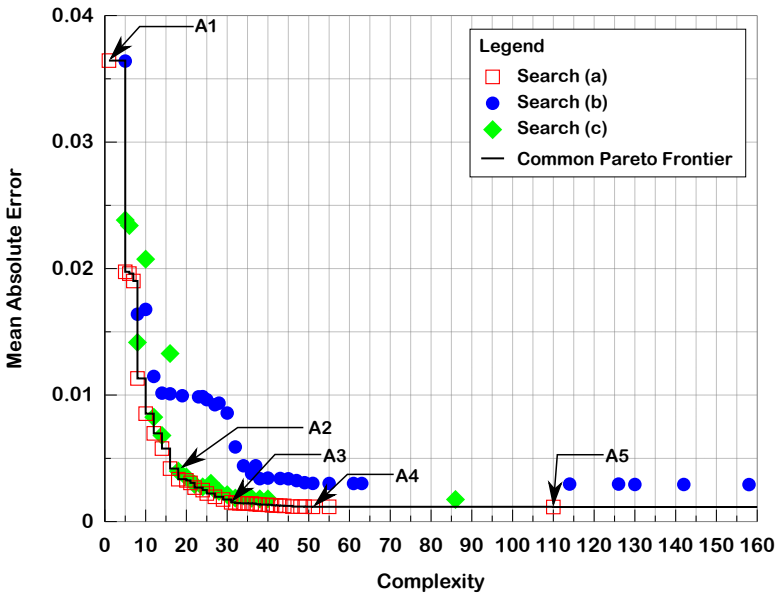
$$\alpha_\gamma^{(\mathrm{A1})} = 1\,, \tag{5.33}$$

$$\alpha_\gamma^{(\mathrm{A2})} = 1 + \frac{C_{21} - p_\nu}{C_{00}\tau + C_{22}\overline{N_{\mathrm{ret,o}}} + C_{23}\overline{N_{\mu\mathrm{b}}}\,\overline{N_{\mu\mathrm{f}}}}\,, \tag{5.34}$$

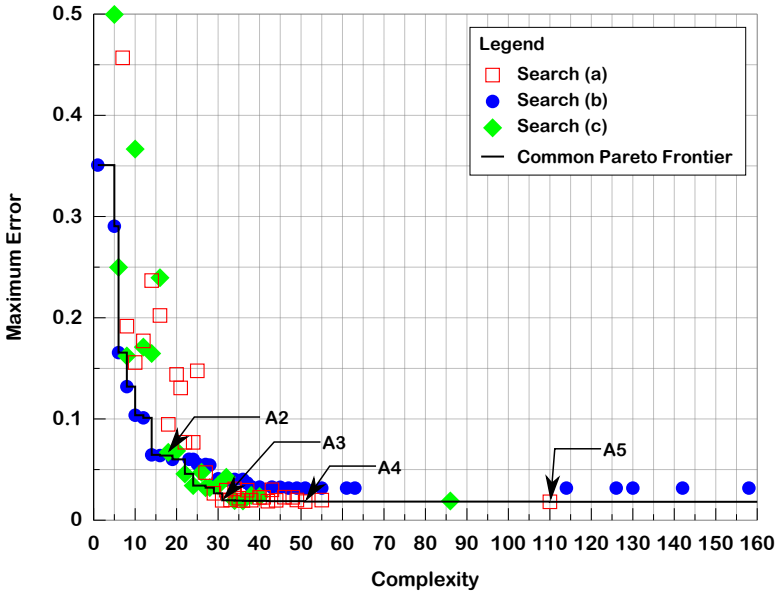
$$\alpha_\gamma^{(\mathrm{A3})} = 1 + \frac{C_{31}}{C_{00}\tau + C_{32}\overline{N_{\mathrm{ret,o}}} + C_{00}\delta\overline{N_{\nu\mathrm{b}}} + \overline{N_{\mu\mathrm{b}}}\,\overline{N_{\mu\mathrm{f}}}\hat{p} + \overline{N_{\mu\mathrm{b}}}\min\left(C_{33} - p_{\mathrm{arrival}}, \overline{N_{\mu\mathrm{f}}}\right)}\,, \tag{5.35}$$

* The three search objectives have been selected based on the performance of initial shorter runs of all search objectives available in Eureqa.

† The information is collected from Eureqa's online user guide available at `http://formulize.nutonian.com/` (last accessed: 18 Dec. 2012) and insights provided by the developer *Michael Schmidt* via the Eureqa Google Group at `https://groups.google.com/forum/?fromgroups=#!forum/eureqa-group` (last accessed: 18 Dec. 2012). Statements of the developer are quoted.



(a) Mean absolute error.



(b) Maximum error (A1 is out of range).

Figure 5.15: Common Pareto frontiers of results found by three Eureqa searches with different objectives.

$$\alpha_{\gamma}^{(\mathrm{A4})} = 1 + C_{41} / \Big(C_{00}\tau + C_{42}\overline{N_{\mathrm{ret,o}}} + C_{00}\delta\overline{N_{\nu\mathrm{b}}} + \left(\overline{N_{\mu\mathrm{b}}}\,\overline{N_{\mu\mathrm{f}}}\right)^{\hat{p}} + \lfloor C_{43} + \overline{N_{\mu\mathrm{b}}} \rfloor \min\left(C_{44} - p_{\mathrm{arrival}}, \hat{p}\left(\sin\left(\overline{N_{\mu\mathrm{f}}}\right) \bmod C_{45}\right)\right)\Big), \tag{5.36}$$

and

$$\alpha_{\gamma}^{(\mathrm{A5})} = 1 + 0.9423 / \Big(C_{00}\tau + 0.3141\overline{N_{\mathrm{ret,o}}} + C_{00}\delta\overline{N_{\nu\mathrm{b}}} + \left(\overline{N_{\mu\mathrm{b}}}\,\overline{N_{\mu\mathrm{f}}}\right)^{\min(0.9423,\hat{p})} + \lfloor \overline{\lambda} \rfloor \min\left(0.4892 - p_{\mathrm{arrival}}, \hat{p}\left(\sin\left(\overline{N_{\mu\mathrm{f}}}\right) \bmod 0.8297\right)\right)\Big). \tag{5.37}$$

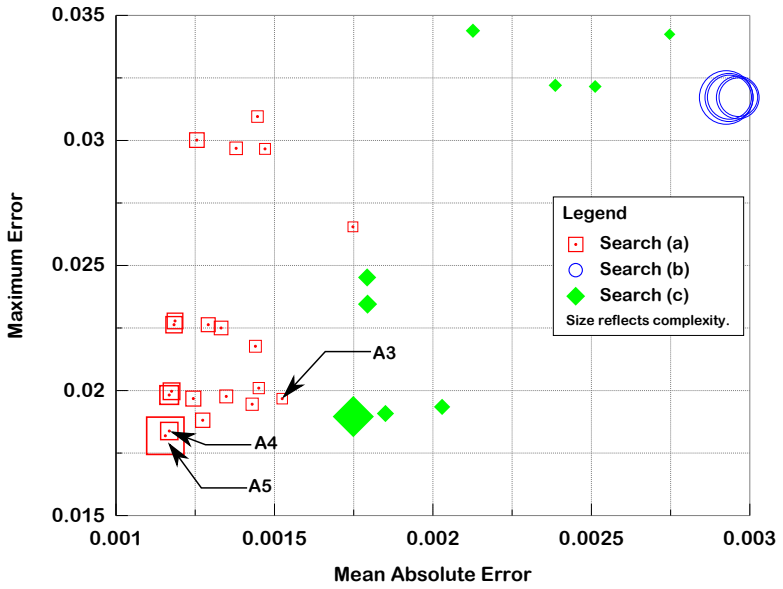


Figure 5.16: Maximum error versus mean absolute error of best results found by the three Eureqa searches (A1 and A2 are out of range).

Table 5.13: Properties of selected Eureqa solutions

Approximation		Found by Search	Complexity	Mean Absolute Error	Maximum Error
A1	Eq. (5.33)	(a)	1	$3.6442 \cdot 10^{-2}$	$6.8337 \cdot 10^{-1}$
A2	Eq. (5.34)	(c)	18	$3.9939 \cdot 10^{-3}$	$6.7023 \cdot 10^{-2}$
A3	Eq. (5.35)	(a)	31	$1.5244 \cdot 10^{-3}$	$1.9675 \cdot 10^{-2}$
A4	Eq. (5.36)	(a)	51	$1.1682 \cdot 10^{-3}$	$1.8382 \cdot 10^{-2}$
A5	Eq. (5.36)	(a)	110	$1.1558 \cdot 10^{-3}$	$1.8190 \cdot 10^{-2}$

Table 5.14: Coefficients of α_γ approximations (Eqs. (5.33)–(5.37))

Coefficient	C_{00}	C_{21}	C_{22}	C_{23}	
Value	1 s	0.9743	0.3450	1.627	
Coefficient	C_{31}	C_{32}	C_{33}		
Value	0.9825	0.3648	0.4708		
Coefficient	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}
Value	0.9476	0.3186	0.043 14	0.4840	0.8297
Coefficient	C_{51}	C_{52}	C_{53}	C_{54}	C_{55}
Value	0.9423	0.3141	0.9423	0.4892	0.8297

where the coefficients C_{00} to C_{58} are given in Table 5.14*.

When investigating these approximations, the following issues can be noticed. (1) For a

* The coefficient C_{00} (with unit *seconds*) is introduced to make α_γ unit-less.

complexity of one, approximation A1 is the best found with respect to the mean absolute error. Consequently, for a significant part of the parameter sets, assuming $\alpha_\gamma = 1$, i.e., an exponentially distributed waiting time of orbit-visiting jobs, gets close to the actual distribution. (2) Giving a feasible stochastic interpretation of the approximations is difficult. In particular, it is surprising that trigonometric functions can help (in A4 and A5) to approximate α_γ. Also quite unexpectedly, the product of $\overline{N_{\mu\mathrm{b}}}$ and $\overline{N_{\mu\mathrm{f}}}$ provides a helpful contribution to the approximation (in A2 to A5). The same holds for the minimum (in A3 to A5), floor (A4 and A5), and modulo (A4 and A5) operators. (3) The actual additional complexity of Eq. (5.37) in comparison to Eq. (5.36) is marginal. Hence, Eureqa's complexity measure, which suggests complexity 51 for A4 and 110 for A5, is somewhat misleading.

5.6 Single-Hop Model: Summary

In Sections 5.3 to 5.5, a total of four different approaches are presented that allow to derive performance measures of the finite-source retrial queue with unreliable servers. These approaches are summarized in Section 5.6.1.

Finally, the results are applied in Section 5.6.2 to deduce the distribution of the single-hop response time $T_{\nu\mu}^{(\max)}$. This complements the research on the single-hop model presented in this thesis. The single-hop model then serves as input to the multi-hop model discussed in Section 5.7.

5.6.1 Comparison of Approaches

Section 5.3 is concerned with the numerical *Markovian steady-state analysis* of the irreducible CTMC underlying the single-hop model. The numerical analysis is carried out by applying the tools MOSEL-2 and SPNP. This approach is feasible for models up to approximately 10^6 states (cf. [334, Tab. 4]$^{\circlearrowleft}$) and, under the given model assumptions, achieves an accuracy up to the numerical error* configured in the iterative numerical solvers of the global-balance equations. The steady-state analysis, however, is not able to derive the distribution of the waiting time by itself.

The *PH approach* presented in Section 5.4 is based on the Method of Phases. It is shown that, under the given assumptions, the waiting time distribution can be exactly described by a $PH(\hat{\boldsymbol{\pi}}, \mathbf{T})$ distribution, where $\hat{\boldsymbol{\pi}}$ is given by Eq. (5.22) and $\mathbf{T}$ is given by Eq. (5.21). While $\mathbf{T}$ can be generated automatically and relatively efficiently based on the model parameters, the derivation of $\hat{\boldsymbol{\pi}}$ requires knowing the steady-state probabilities obtained by Markovian steady-state analysis. Using the PH approach, in principle all moments of the waiting time can then be obtained exactly† by applying Eq. (5.23). Unfortunately, however, it is shown in Section 5.4.3.1 that this approach does not scale well for increasing sizes of the single-hop model's state space. For example, using currently accessible software tools and hardware, the second moment $\overline{T_\nu^2}$ of the waiting time cannot be obtained for models exceeding approximately $9{\cdot}10^4$ states.

Alternatively, mean performance measures and also the waiting time distribution can be obtained by *DES*. This is carried out using CPN Tools in, e.g., Sections 5.3.4 and 5.4.3.2 for verifying the implementations of the Markovian steady-state analysis and PH approach, respectively, and in Section 5.5.1 to derive histograms of the waiting time of orbit-visiting jobs.

* Remember that a precision of 10^{-5} is selected for SPNP in Section 5.4.3.1.

† Again, up to the numerical errors of the underlying iterative algorithms.

DES does not need to span and maintain the full state space underlying the model, and hence, it is not severely limited by large state spaces. Additionally, it can be easily extended to non-exponentially distributed generation, service, and retrial times in future work. Therefore, even if the simulation model is based on the same assumptions as the analytical models, it still has attractive advantages. However, DES usually comes with some general disadvantages. For example, it requires long runtimes to get steady-state results that are independent from the initial state. Also a large number of runs might be needed to achieve narrow CIs. Both drawbacks are worsened in the presence of rare events caused, e.g., by strongly diverse rate parameters. Moreover, the specific implementation of the DES model using CPN Tools chosen in this thesis turns out to be susceptible to round-off errors caused by high rate parameters (like $\delta = 2500\tau$) combined with CPN Tools' bounded simulation time as discussed in Appendix Section B.6.

Due to the drawbacks of the PH approach and DES, this thesis proposes an alternative *approximation method*, which aims at the approximate derivation of the waiting time distribution for models with large state space *and* high or strongly diverse rate parameters. The method is based on the observation that the waiting time distribution of an orbit-visiting job can be approximated by a gamma distribution with parameters μ_γ and α_γ. Parameter μ_γ can be obtained quite conveniently from steady-state analysis results using Eq. (5.29). For deriving parameter α_γ, Eq. (5.32) can be applied when $\overline{T_v^2}$ is known from the PH approach or DES. Alternatively, with Eq. (5.37), this thesis also provides an approximation of α_γ in closed form, which is derived using symbolic regression. Applying the approximation allows to calculate α_γ solely based on model parameters and steady-state performance measures. Hence, neither the PH approach nor DES need to be performed to approximately obtain the waiting time's distribution.

Note that it indeed has advantages to refrain from applying the PH approach for calculating the exact distribution of the waiting time (or a high number of moments) while still using it to derive α_γ, since for calculating the latter only the second moment of the waiting time is needed. In contrast to higher moments, the second moment often can be determined for larger state spaces. Moreover, the gamma distribution can be used more easily than the $PH(\hat{\boldsymbol{\pi}}, \mathbf{T})$ distribution in subsequent calculations, this is exploited in the multi-hop model presented in Section 5.7.

In summary, based on the experience gained in Section 5.4.3.1, the following suggestions can be provided for deriving the waiting time distribution in finite-source retrial queues with unreliable servers.

- For models with *small state space* (few hundred states*), steady-state performance measures can be easily (and almost exactly) obtained using Markovian steady-state analysis, and the PH approach can be used to obtain the distribution or density function of the waiting time up to high accuracy or a significant number of the waiting time's moments. Still, using the gamma approximation might significantly simplify subsequent calculations.

- For *medium-sized state spaces* (up to approximately $9{\cdot}10^4$ states), Markovian steady-state analysis can still be applied. The second moment $\overline{T_v^2}$ and parameter α_γ are still obtainable (almost exactly) using the PH approach. Hence, the gamma approximation

* Note that the given maximum sizes of the state space are rough estimations which are based on the hardware and software tools applied in Section 5.4.3.1. The actual values can be expected to depend significantly on the available computational resources.

can be applied based on the results provided by both methods. This approach is chosen in Section 5.6.2 and following.

- Models with *large state spaces* (up to approximately 10^6 states) can still be evaluated using Markovian steady-state analysis. However, $\overline{T_v^2}$ can no longer be derived using the PH approach. Hence, the gamma approximation can only be applied based on the approximate α_γ provided by Eq. 5.32 or when $\overline{T_v^2}$ is obtained by applying DES.
- For *very large state spaces* (beyond approximately 10^6 states), Markovian steady-state analysis is no longer feasible and DES seems to be the only remaining option to derive steady-state performance measures and waiting time distributions.

5.6.2 Distribution of the Single-Hop Response Time

Focus is now given to the question *"What is the probability that an arriving job meets a given upper bound $T_{v\mu}^{(max)}$ of the single-hop response time $T_{v\mu}$?"* To answer this question, the stochastic distribution $F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) = P\left(T_{v\mu} \leq T_{v\mu}^{(\max)}\right)$ of the single-hop response time $T_{v\mu}$ is described based on the intermediate results obtained in Section 5.5.2. It can be seen in Fig. 5.11 that $T_{v\mu} = T_v + T_\mu$, and hence, $P\left(T_{v\mu} \leq T_{v\mu}^{(\max)}\right) = P\left(T_v + T_\mu \leq T_{v\mu}^{(\max)}\right)$. Then, it can be shown* that

$$F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) = \int_0^{T_{v\mu}^{(\max)}} \left(1 - e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu}\right) p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t}\, \mathrm{d}t\,, \tag{5.38}$$

where the gamma function $\Gamma\left(\alpha_\gamma\right)$ is given by Eq. (5.28).

With this result, the single-hop model discussed in Sections 5.2 to 5.5 is completed. The single-hop results constitute a major component of the multi-hop model discussed in Section 5.7.

5.7 Multi-Hop Model

The multi-hop model and its evaluation aim at investigating the detection-to-notification delay T_{d2n} achieved by the WSN in dependence on various WSN and protocol parameters. T_{d2n} mainly depends on two factors. First, on the single-hop response time $T_{v\mu}$, whose mean value is given by Eq. (5.13) and whose CDF can be approximated by Eq. (5.38). The second factor is the number of hops an EVM needs to travel from its source node to the closest sink.

5.7.1 Distribution of Number of Communication Hops

Still assuming three sinks, the number of hops between a tagged source node and its closest sink is given by $d_{\mathrm{h}} = \min\left(d_{\mathrm{h}}^{(0)}, d_{\mathrm{h}}^{(1)}, d_{\mathrm{h}}^{(2)}\right)$, which is just referred to as *hop count* in the following. The hop count distribution $P(d_{\mathrm{h}} = h)$ describes the probability that a source node's hop distance to the closest sink(s) is h ($h \in \mathbb{N}^+$). $P(d_{\mathrm{h}} = h)$ depends significantly on various

* The full derivation of $F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right)$ is provided in Appendix Section B.9, Eq. (B.14).

Table 5.15: Summary of assumptions for MASON simulation of multi-hop scenario

Description	Assumption/Value
Size of monitored area (A_F)	approx. $70\,\mathrm{km}^2$
Number of sensor nodes (N_S)	10^5
Resulting node density ($\overline{N}_d$)	approx. $1400\,\mathrm{km}^{-2}$
Shape of monitored area	square, without obstacles
Sensor node deployment	random uniform
Number of sinks	3
Sink deployment	deterministic, large triangle
Distance of sinks to border of monitored area	500 m minimum
Transmission range of sensor nodes and sinks	100 m
Antenna radiation pattern	omnidirectional

Table 5.16: Excerpt of $P(d_\mathrm{h} = h)$ lookup table corresponding to Fig. 5.19

h	1	2	...	39	40	...	86	87	88	...
$P(d_\mathrm{h} = h) \cdot 10^3$	1.33	3.40	...	28.53	29.19	...	0.01	0.00	0.00	...

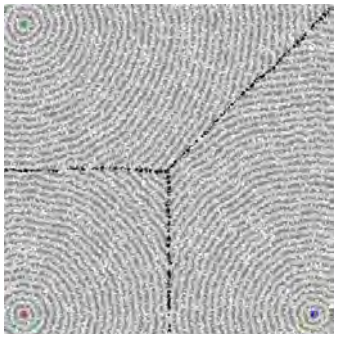
factors, including the number of sinks, their placement within the monitored area, and the monitored area's shape. Hence, giving an exact, generally applicable description of $P(d_\mathrm{h} = h)$ is tedious and not the focus of this thesis.

Nevertheless, for getting a feeling for $P(d_\mathrm{h} = h)$, it is exemplarily investigated for the six sink placements shown in Fig. 5.17. The figure illustrates MASON simulation results based on assumptions derived in Chapter 4 and summarized in Table 5.15. Sensor nodes with odd and even d_h are shown as dark- and light-gray dots, respectively. Sensor nodes whose d_h refers to their distance to at least two sinks are given in black color. These black-colored nodes consequently form the edges of a Voronoi diagram whose generating points are the sinks.

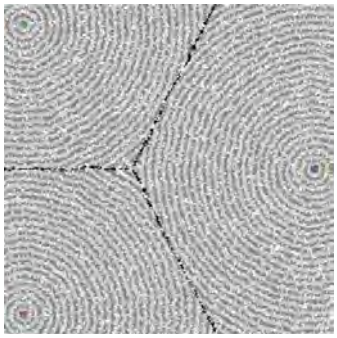
The resulting hop count distributions are shown in Fig. 5.18. Each graph of the figure shows the mean of five MASON simulation runs* of the corresponding sink placement. For each run, the random placement of the sensor nodes is resampled. Averaging over all sink placements, the relative frequency of hop counts is shown in Fig. 5.19 with 99% CIs.

The mean simulation results shown in Fig. 5.19 can be provided explicitly to subsequent calculations via a lookup table (or array), which allows to retrieve $P(d_\mathrm{h} = h)$ for any $h \in \mathbb{N}^+$. Table 5.16 shows an excerpt of the lookup table corresponding to Fig. 5.19. Remember, however, that this lookup table is based on the assumptions summarized in Table 5.15. If the values given in this table need to be adjusted to achieve the desired performance, the lookup table needs to be regenerated accordingly.

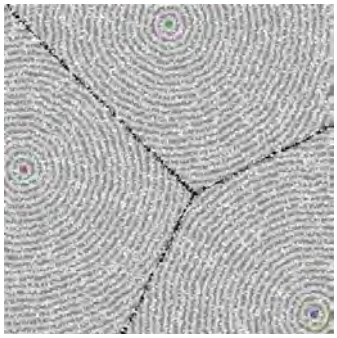
* 99% CIs are obtained but not shown in Fig. 5.18 to preserve clarity. Their maximum half-width is approximately 259 nodes for Placement F at 41 hops.



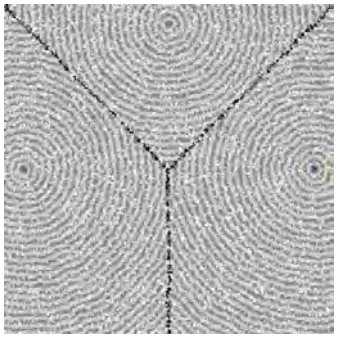
(a) Placement A (3 corners, 0 edges).



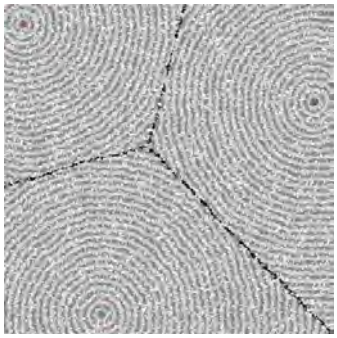
(b) Placement B (2 corners, 1 edge).



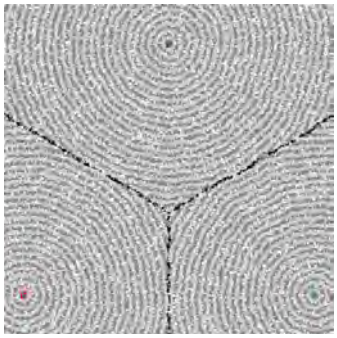
(c) Placement C (1 corner, 2 edges).



(d) Placement D (0 corners, 3 edges).



(e) Placement E (equilateral triangle).



(f) Placement F (rotated equilateral triangle).

Figure 5.17: Example sink placements and resulting hop count rings.

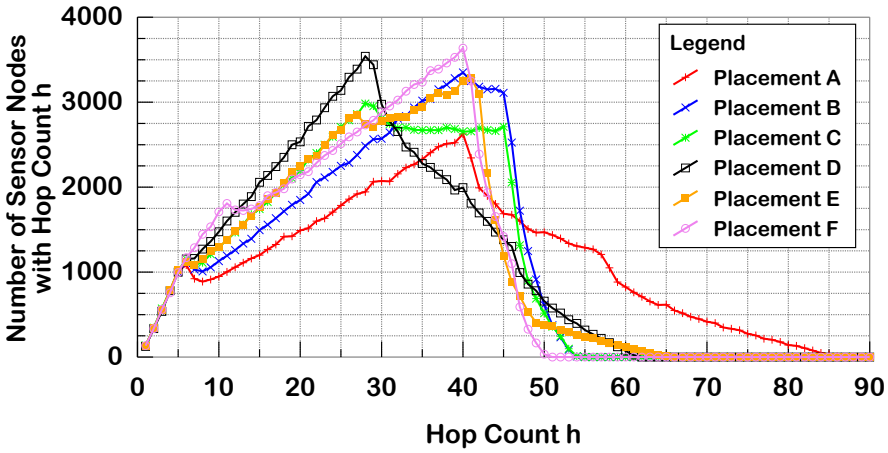


Figure 5.18: Hop count distributions obtained by simulating sink placements defined in Fig. 5.17.

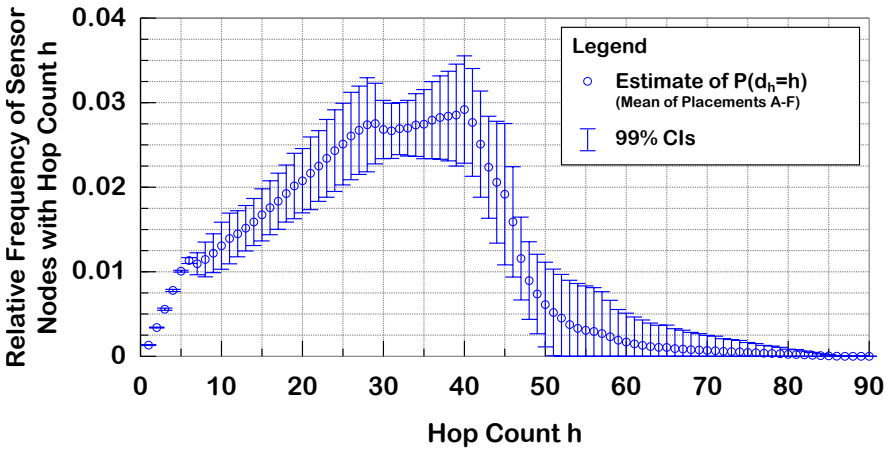


Figure 5.19: Joint hop count distribution of all sink placements.

5.7.2 Distribution of Detection-to-Notification Delay

This thesis aims at deriving the probability $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ that the detection-to-notification delay T_{d2n} lies below a given upper bound $T_{\mathrm{d2n}}^{(\max)}$. Still assuming uniform distributions of node and event locations within the monitored area, each node equally likely senses an upcoming event. $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ can then be given by

$$P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right) = \sum_{h=1}^{d_{\mathrm{h}}^{(\max)}} P(d_{\mathrm{h}} = h)\, P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = h\right), \tag{5.39}$$

where is $d_{\mathrm{h}}^{(\max)} = 100$ is an upper bound* of the hop count, $P(d_{\mathrm{h}} = h)$ is given by Table 5.16, and the remainder of this section is devoted to the derivation of $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = h\right)$.

Consider a tagged source node with hop count $d_{\mathrm{h}} = h$ that just generated an EVM and let $T_{v\mu}^{(i)}$, $1 \leq i \leq h$, describe the single-hop delay experienced by the EVM during transition from Ring i to Ring $(i-1)$ on its way to the sink. Since the sink is always active and responds to all EVMs immediately (i.e., without back-off delay), $T_{v\mu}^{(1)} \approx 0\,\mathrm{s}$. All other $(h-1)$ single-hop delays (i.e., $T_{v\mu}^{(i)}$, $2 \leq i \leq h$) experienced by the EVM are independent and identically distributed (iid) random variables, which follow the distribution given by Eq. (5.38). The sum of the single-hop delays is hence a sum of $(h-1)$ iid random variables.

There are different ways for calculating this sum. On the one hand, it can be calculated using the $(h-1)$-fold convolution of $F_{T_{v\mu}}(x)$ with itself. However, there is not much hope that this approach can be implemented efficiently in the given scenario. A brief discussion of the convolution approach is given in Appendix Section B.10.

Instead, this thesis uses a numerical approach based on a Monte Carlo experiment. Each run of the experiment carries out a configurable number of trials, where each trial simulates

* According to Section 4.3.6.1, an upper bound of the maximum number of hops from any sensor node to any sink is estimated as 145 in the investigated scenario. On the other hand, the simulation results shown in Fig. 5.18 indicate that 90 is a sufficient upper bound for the hop distance to the closest sink when the sinks are reasonably placed. In the following, $d_{\mathrm{h}}^{(\max)} = 100$ is chosen defensively. If the number and location of sinks, the size of the monitored area, or the nodes' transmission range is altered, the value of $d_{\mathrm{h}}^{(\max)} = 100$ needs to be reconsidered.

how many hops can be crossed by an EVM for given $T_{\mathrm{d2n}}^{(\max)}$, μ, α_γ, μ_γ, and p_ν. For this, each trial samples T_μ and $T_{\nu\mathrm{o}}$ from their exponential (with rate μ) and gamma (with parameters α_γ and μ_γ) distribution, respectively. Additionally, to calculate T_ν from $T_{\nu\mathrm{o}}$, a Bernoulli trial with parameter p_ν decides whether the EVM visits the orbit. The obtained samples of T_μ and T_ν allow to calculate a sample of the single-hop response time $T_{\nu\mu} = T_\mu + T_\nu$. The number of such $T_{\nu\mu}$ samples that fit into $T_{\mathrm{d2n}}^{(\max)}$ describe the number of hops that can be crossed by the EVM. The result is incremented by one to account for the delay-less hop between the first ring and the sink. Based on the trials' results and $P(d_\mathrm{h} = h)$, an estimate of $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ can be obtained for each run. Several runs are conducted and the overall means and corresponding confidence intervals are derived. The results for $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ presented in Section 5.8 are based on ten runs with 1000 trials each and 99% confidence intervals are provided.

At this point, the implementation of the multi-hop model in form of the described Monte Carlo experiment could be verified more rigorously. Here, *verification* refers to the process of getting confidence in the statement "the experiment is implemented correctly".* This verification, however, is not in the focus on this thesis, since more importance is given to the single-hop model due to its more general applicability. An initial verification could check, whether the results for the detection-to-notification delay of nodes located in the second ring are identical to the single-hop results. As a next step, the mean value of the detection-to-notification delay of nodes located in Ring h could be compared to the $(h-1)$-fold mean value of the single-hop delay. Additionally, the distribution results of the Monte Carlo experiment could be compared to the $(h-1)$-fold convolution of $F_{T_{\nu\mu}}(x)$ with itself (see Section B.10) for small h and a limited amount of result values.

5.8 Numerical Results and Discussion

This section applies the multi-hop model, which is derived in Section 5.7 based on the single-hop models introduced in Sections 5.2 to 5.5, to discuss the performance of the proposed WSN setup in particular with respect to the trade-off between lifetime and detection-to-notification delay.

This thesis aspires a setup where at least 99% of all nodes are able to communicate an EVM to the closest sink within five minutes or less (i.e., $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s}) \geq 0.99$). Several scenarios are investigated in the following for discussing whether and how this goal can be achieved and which system parameters turn out to have the main influence. A brief overview of the scenarios investigated in Sections 5.8.1 to 5.8.9 is provided in Table 5.17. Additionally, the table already provides a rating of the varied parameters' influence on T_{d2n}. This rating is based on the scenario results discussed in more detail in the following.

All scenarios are based on the *RealSet*. According to Table 5.7, the reducible CTMC underlying the *RealSet* has a relatively small size of 276 states. Therefore, the α_γ parameter of the waiting time's distribution is obtained using the PH approach of Section 5.4 and Eq. 5.32.

* For more details on the interpretation of the terms *verification* and *validation* used in the scope of this thesis, see Section 6.2.

Table 5.17: Scenario overview

Scenario	Description	Influence on T_{d2n}
1	Original *RealSet* and multi-hop parameters	—
2	Increasing the number of sinks	medium
3	Changing the retrial rate ν	strong
4	Varying the service rate μ	weak
5	Altering the repair rate τ while keeping $\delta = 2500\tau$	weak
6	Modifying ratio of δ to τ	strong
7	Increasing the traffic load λ	weak
8	Varying the node density $\overline{N}_d$	strong
9	Modifying the transmission range d_{T}	strong

5.8.1 Scenario 1: Original Model

Scenario 1 investigates the situation where the single-hop model follows the original *RealSet* parameters given in Table 5.2 and the multi-hop model is based on the assumptions summarized in Table 5.15. Remember that in the *RealSet* the active/sleep ratio governed by the parameters τ and $\delta = 2500\tau$ is chosen such that the aspired lifetime of three years is achieved.

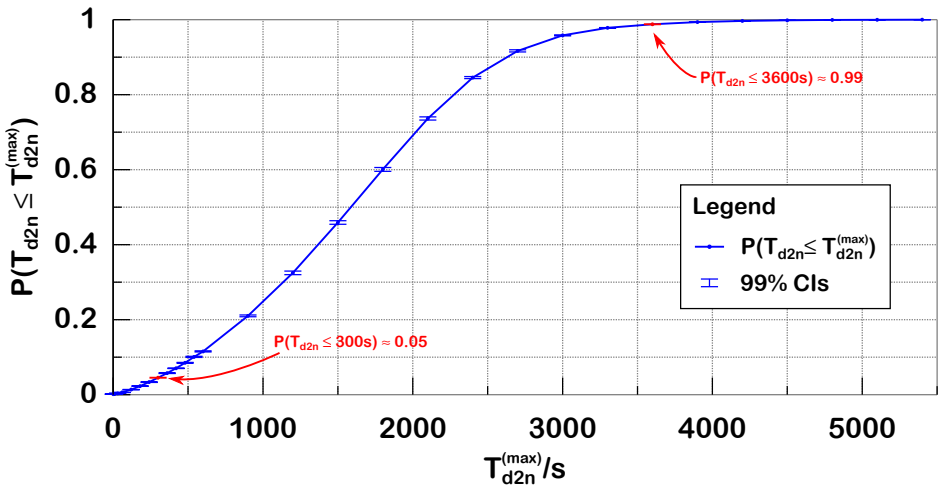


Figure 5.20: *Scenario 1*: Distribution of detection-to-notification delay.

For this scenario, $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ is shown versus $T_{\mathrm{d2n}}^{(\max)}$ in Fig. 5.20. The figure reveals that only approximately 5% of the sensor nodes are able to achieve the desired T_{d2n} of five minutes (i.e., 300 s) or less. Approximately 1% of the sensor nodes need even more than 3600 s (i.e., 1 h) to communicate a detected event to the next sink.

Hence, the required performance is not reached by *Scenario 1*. The scenarios provided in the following discuss different modifications to *Scenario 1* tailored towards achieving the aspired performance.

5.8.2 Scenario 2: Increasing the Number of Sinks

As a first, obvious approach to improve *Scenario 1*, *Scenario 2* aims at reducing the number of hops between the sensor nodes and the sinks. This can be achieved by increasing the number of sinks, reducing the size of the monitored area, or increasing the nodes' transmission range. Assuming that the size of the monitored area is non-negotiable and that the transmission range cannot be increased significantly due to hardware constraints, increasing the number of sinks seems the only viable option for *Scenario 2*.

For *Scenario 2*, it is assumed that there are five sinks. Four of these are located in the corners of the monitored area. The fifth sink is located in the area's center. All other assumptions are chosen identical to *Scenario 1* (i.e., Table 5.15). Similar to Section 5.7.1, an approximation of the hop count distribution $P(d_{\mathrm{h}} = h)$ can be obtained by simulating *Scenario 2* using MASON.

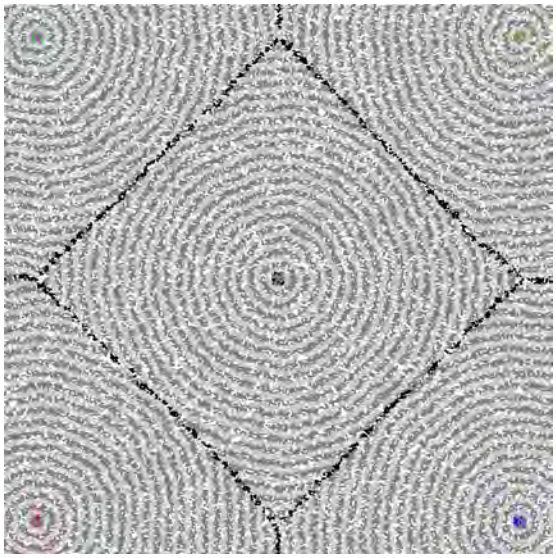


Figure 5.21: *Scenario 2*: Placement of five sinks and resulting hop count rings.

Figure 5.21 illustrates an exemplary MASON simulation result of *Scenario 2* and demonstrates the placement of the five sinks and the corresponding hop count rings formed by the sensor nodes.

Figure 5.22 shows the estimated hop count distribution and the 99% CIs obtained from five MASON simulation runs of *Scenario 2*.

In Fig. 5.23, the resulting detection-to-notification delay distribution $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ is provided for *Scenario 2* as obtained from the multi-hop model. It can be seen that in *Scenario 2* approximately 7% of the source nodes are now able to achieve the desired T_{d2n} and approximately 1% of the source nodes still need more than 2400 s (i.e., 40 min) to communicate an EVM to the next sink. While in comparison to *Scenario 1* the situation improved notably, the result is not yet satisfactory.

Whether the number of sinks can be increased even further or their placement can be optimized depends on several factors including the sinks' price, the monitored area's terrain, and the availability of the required infrastructure. Hence, before deciding to increase the number of sinks further, other options for improving the situation are investigated in the following.

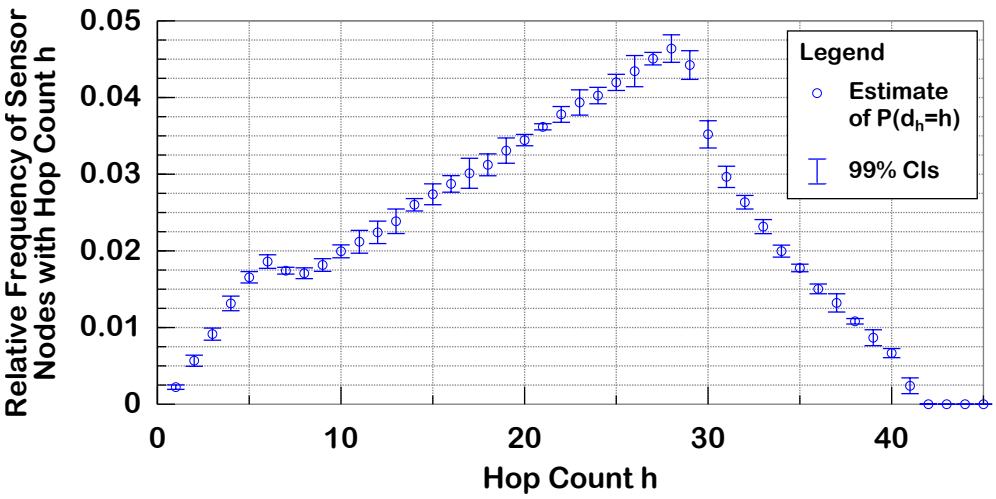


Figure 5.22: *Scenario 2*: Hop count distributions obtained by simulating.

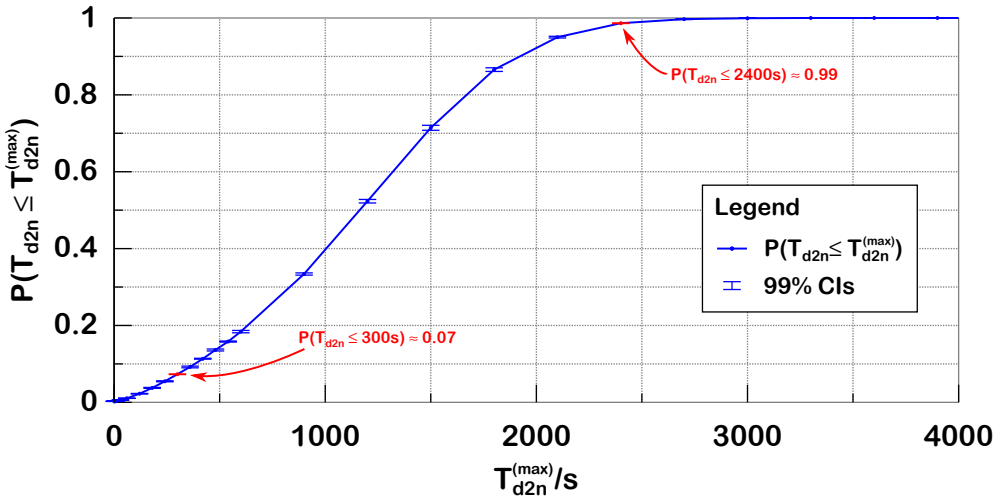


Figure 5.23: *Scenario 2*: Distribution of detection-to-notification delay.

5.8.3 Scenario 3: Increasing the Retrial Rate

Scenario 3 is similar to *Scenario 1* except that the retrial rate ν is varied. By increasing ν an EVM-holding node should be able to catch a next-hop node faster and the overall detection-to-notification delay should be decreased.

In Fig. 5.24, $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ (y-axis) is plotted versus ν (x-axis). The figure shows that T_{d2n}

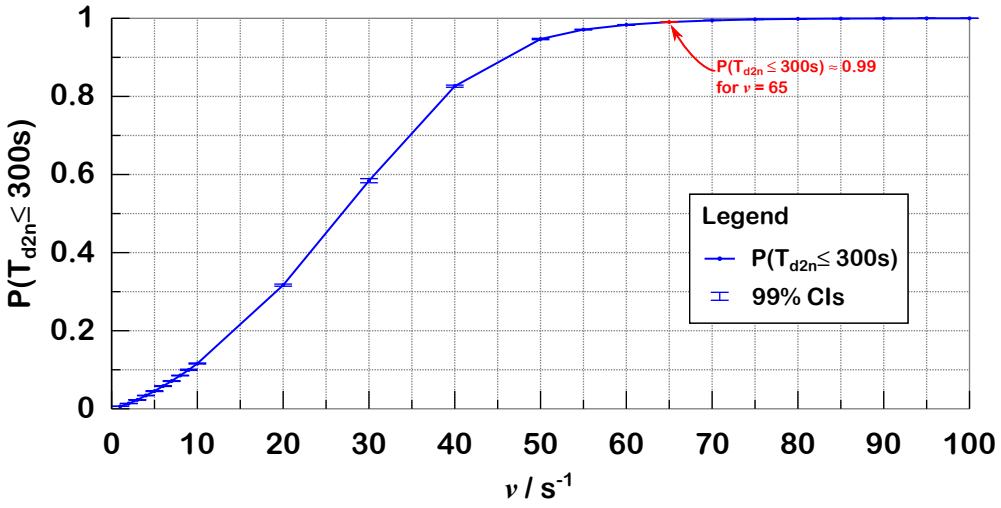


Figure 5.24: *Scenario 3*: $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ in dependence of ν.

indeed depends significantly on the retrial rate ν. This result justifies the decision to model the proposed protocol using retrial queues.

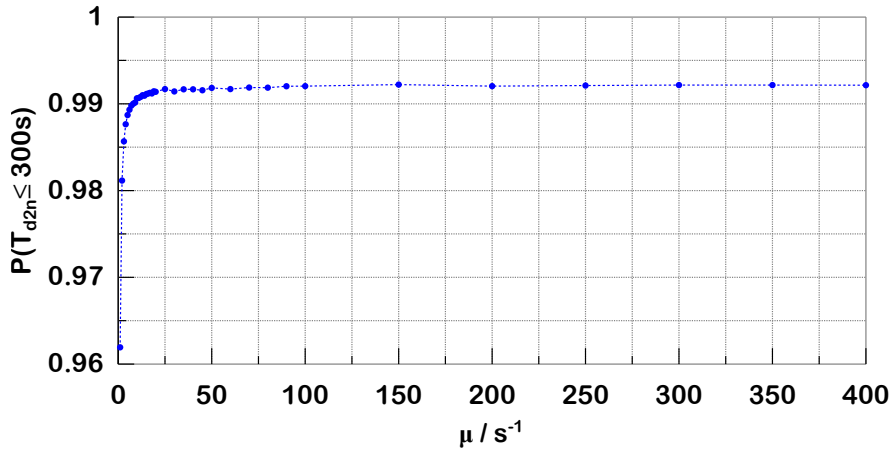
By increasing ν from $5\,\mathrm{s}^{-1}$ to at least $65\,\mathrm{s}^{-1}$, the desired performance can be reached. The advantage of this solution is that no additional hardware is required. Remember that according to Section 5.2.5, ν models the rate at which a sensor node repeats the EVM if no ACK was received from one of the next-hop neighbors. The rate $\nu = 65\,\mathrm{s}^{-1}$ still lies well beyond the upper bound of $1000\,\mathrm{s}^{-1}$ derived in Section 5.2.5. It implies that the EVM needs to be re-sent approximately every $\nu^{-1} = 15\,\mathrm{ms}$. Depending on the EVM's size, this uses approximately 2.5 to 4.6 % of the radio bandwidth within the node's transmission (or rather interference) range.

In addition to this increased but still seemingly acceptable bandwidth consumption, the increased ν also (on first sight) leads to an increased energy consumption of the sensor node, since the EVM needs to be sent more frequently. However, the contrary can be expected. First, remember that the power needed for transmitting is not much higher than the power needed for idle listening (see Section 4.2.8). Second, the shorter the EVM stays in the outPool of the sensor node (i.e., the shorter the single-hop delay $T_{\nu\mu}$), the faster the node can switch from idle listening back to sleep mode.

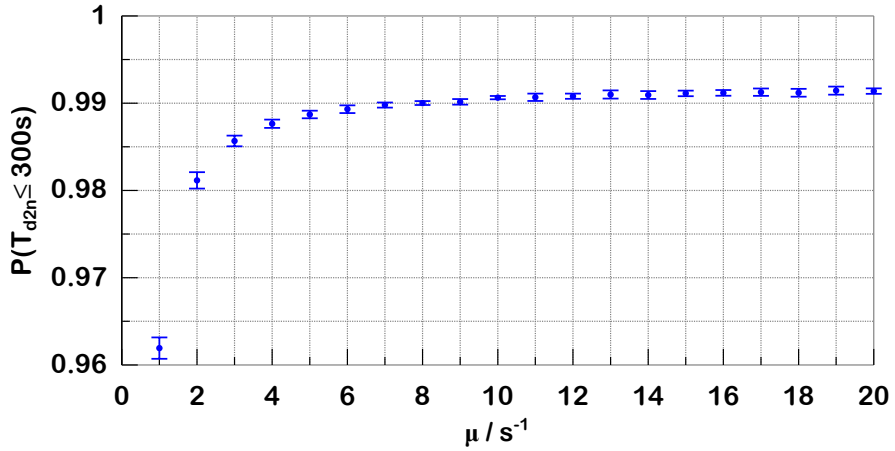
Also remember from Section 5.2.5 that, in order to reduce the number of unnecessary retransmissions, $T_b^{(\max)}$ should be chosen smaller than ν^{-1}. *Scenario 4* is devoted to the investigation of variations of $T_b^{(\max)}$.

5.8.4 Scenario 4: Varying the Service Rate

The maximum back-off delay $T_b^{(\max)}$ is modeled via the service rate μ, i.e., $T_b^{(\max)} \approx \overline{T_\mu} = \mu^{-1}$. *Scenario 4* is devoted to investigating the sensibility of the results obtained for *Scenario 3* on variations of μ. This allows to assess the need for more accurate models of $T_b^{(\max)}$. The model parameters are chosen similar to *Scenario 1* and *Scenario 3*, except that $\nu = 65\,\mathrm{s}^{-1}$ and μ is variable.



(a) $1 \leq \mu \leq 400$, without CIs.



(b) $1 \leq \mu \leq 20$, with 99% CIs.

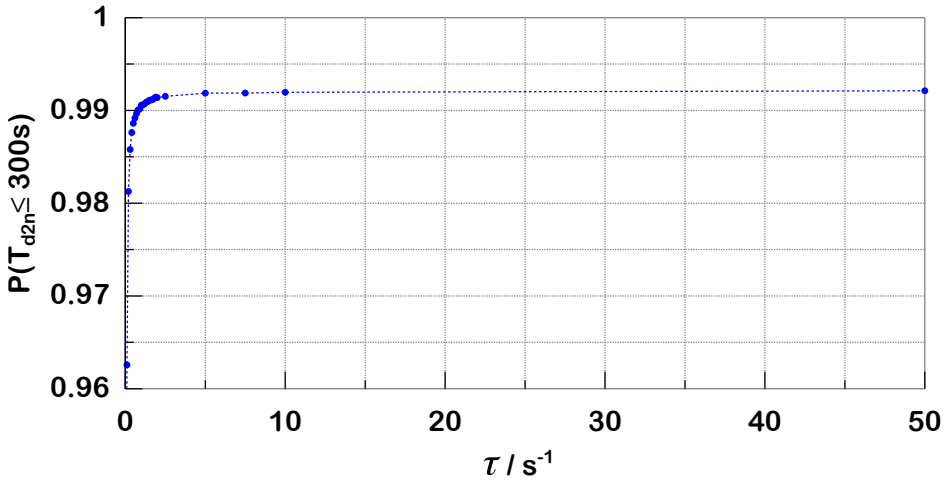
Figure 5.25: *Scenario 4*: $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ in dependence of μ.

Figure 5.25 illustrates $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ for various values of μ. In particular, Fig. 5.25(a) covers $1 \leq \mu \leq 400$. Figure 5.25(b) shows $1 \leq \mu \leq 20$ in more detail and includes the 99% CIs. The figures indicate that $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s}) \leq 99\%$ for approximately $\mu \geq 8$. For $\mu \geq 8$, the actual influence of μ on $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ is relatively insignificant and the server performance appears to be dominated by the active/sleep periods instead. This conjecture is investigated in more detail in *Scenarios 5* and *6*.

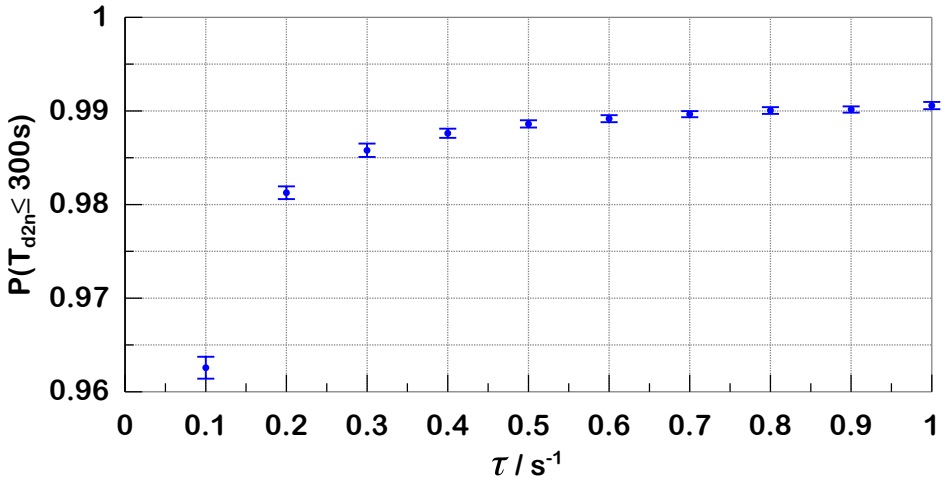
Remember that $\mu^{-1} = T_b^{(\max)}$ should be chosen smaller than ν^{-1}. According to the results of *Scenario 3*, this implies $\mu > \nu = 65\,\mathrm{s}$. The results of *Scenario 4* show that this requirement does not pose an actual drawback with respect to T_{d2n}.

5.8.5 Scenario 5: Varying the Wake-Up Rate

The wake-up rate (repair rate) τ is varied in this scenario while keeping the active/sleep ratio constant, i.e., $\delta = 2500\tau$. Consequently, the expected lifetime of the nodes is not (significantly*) altered. Again, $\nu = 65$ and all other parameters are equivalent to *Scenario 1*. Figure 5.26(a) shows $P(T_{\text{d2n}} \leq 300\,\text{s})$ over τ for $0.1 \leq \tau \leq 50$.



(a) $0.1 \leq \tau \leq 50$, without CIs.



(b) $0.1 \leq \tau \leq 1$, with 99% CIs.

Figure 5.26: *Scenario 5*: $P(T_{\text{d2n}} \leq 300\,\text{s})$ in dependence of τ.

Figure 5.26(b) provides the 99% CIs of $P(T_{\text{d2n}} \leq 300\,\text{s})$ for $0.1 \leq \tau \leq 10$. The figures indicate that in the present scenario, τ should be greater than approximately 0.8 to ensure that $P(T_{\text{d2n}} \leq 300\,\text{s}) > 99\%$. As long as this requirement is fulfilled, the absolute values of τ and $\delta = 2500\tau$ have only little influence on $P(T_{\text{d2n}} \leq 300\,\text{s})$.

* Remember from Section 4.2.8 that the power consumption during mode switching is considered negligible in this thesis.

5.8.6 Scenario 6: Altering the Active/Sleep Ratio

In *Scenarios 1* to *5* the active/sleep ratio is kept constant at $\delta = 2500\tau$. *Scenario 6* investigates the impact of changes to this ratio, i.e., whether it is possible to significantly increase the lifetime without significant increase of the delay. All model parameters except $\nu = 65$ and the variable δ are chosen equivalently to *Scenario 1*. In particular, $\tau = 1\,\mathrm{s}^{-1}$ is fixed. Remember that δ is the rate which determines how fast an idle node enters the sleep mode. Hence, an increase of δ increases the nodes' lifetime further.

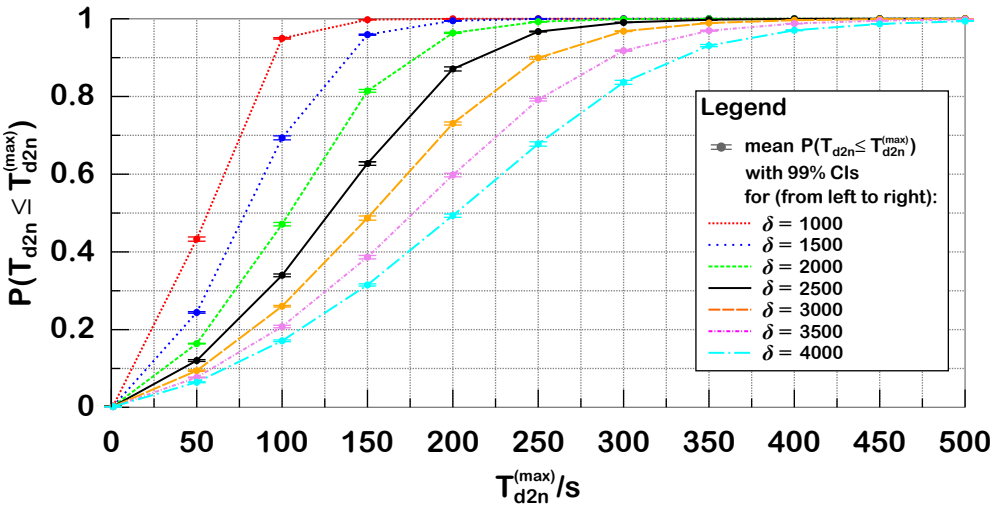


Figure 5.27: *Scenario 6*: Distribution of detection-to-notification delay for various δ.

Figure 5.27 illustrates $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ (y-axis) versus $T_{\mathrm{d2n}}^{(\max)}$ (x-axis) for various values of δ (curves). The figure demonstrates that the T_{d2n} is strongly coupled to the active/sleep ratio, and hence to the nodes' lifetime. Therefore, when keeping all other parameters unmodified, increasing the nodes' lifetime by increasing δ always results in an increase of T_{d2n}.* For example, increasing δ from $2500\,\mathrm{s}^{-1}$ to $4000\,\mathrm{s}^{-1}$ would increase the nodes' lifetime from approximately three years to approximately five years.† At the same time, however, T_{d2n} is increased such that 1% of the events can be expected to be reported after 8 min (instead of the aspired 5 min) or later. Increasing ν might help to decrease T_{d2n} again, as described in *Scenario 3*.

Vice versa, when T_{d2n} needs to be decreased and increasing ν is no longer an option, this can be achieved by modifying the active/sleep ratio via increasing τ and/or decreasing δ, i.e., at the expense of the nodes' lifetime. For example, if 99% of all events should be reported after 3 min (instead of 5 min) or earlier, then δ needs to be set to approximately $1200\,\mathrm{s}^{-1}$. This reduces the nodes' lifetime to approximately 1.5 years.

* Due to the results of *Scenario 5*, similar can be expected when decreasing τ.

† Under the assumptions discussed in Sections 4.2.8 and 5.2.7. In particular, $E_{\mathrm{trans}} \approx 2400\,\mathrm{J}$, $P_{\mathrm{sleep}} = 5\,\mu\mathrm{W}$, $P_{\mathrm{active}} \approx 50\,\mathrm{mW}$, and using Eq. (5.3) to calculate $\overline{T_{\mathrm{life}}}$.

5.8.7 Scenario 7: Increasing the Traffic Load

Scenario 7 investigates the traffic load's influence on T_{d2n}. The traffic load is governed by the two parameters N_λ (the number of sources) and λ (the generation rate of each source). The retrial rate is again set to $\nu = 65\,\mathrm{s}^{-1}$ in this scenario. All other parameters are equivalent to *Scenario 1*.

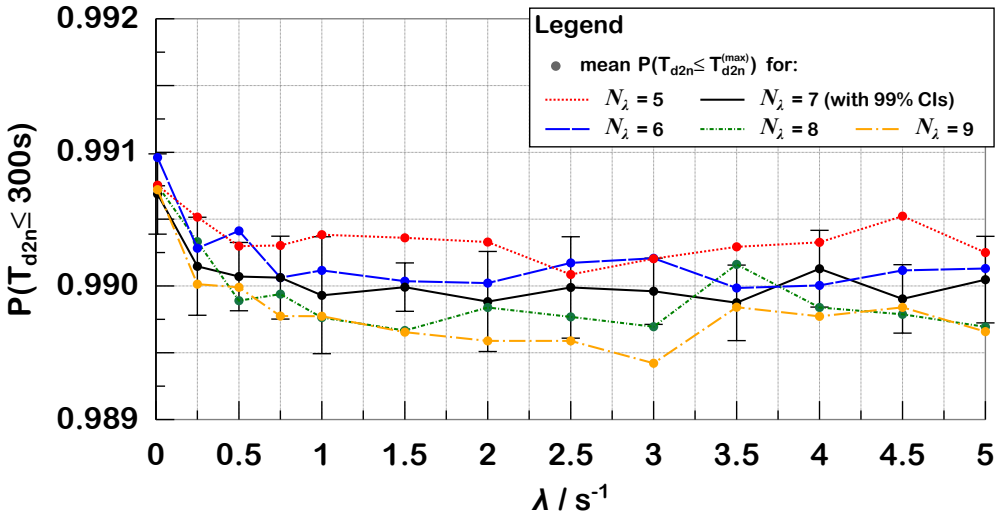


Figure 5.28: *Scenario 7*: $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ in dependence of λ and N_λ.

In Figure 5.28, the influence of λ (x-axis) and N_λ (curves) on $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ (y-axis) is indicated. The 99% CIs are exemplarily provided for $N_\lambda = 7$. While the figure appears confusing on first sight, interesting conclusion can be inferred from it. In particular, mind the fine-granular scale of the y-axis. Apparently, both parameters have only small influence on $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$. The variations of $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ lie in the order of the 99% CIs' width. Still, two weak trends can be identified. First, for very small $\lambda \leq 0.5$, $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ seems to increase a bit, while for larger λ, $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ does not drop significantly below 99%. Second, $P(T_{\mathrm{d2n}} \leq 300\,\mathrm{s})$ decreases slightly for increasing N_λ.

5.8.8 Scenario 8: Varying the Node Density

According to Section 5.2.3, the node density is directly related to the number of servers in the model. In particular, remember that N_μ is equal to the expected number of next-hop neighbors $\overline{N}_{\mathrm{r}^-}$ of any node, which linearly* depends on the node density $\overline{N}_d$. *Scenario 8* investigates the influence of N_μ on $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ while keeping $\nu = 65\,\mathrm{s}^{-1}$ and all other parameters identical to *Scenario 1*. The result is shown in Fig. 5.29 for $N_\mu \in \{7, 9, 11, 15, 20\}$.

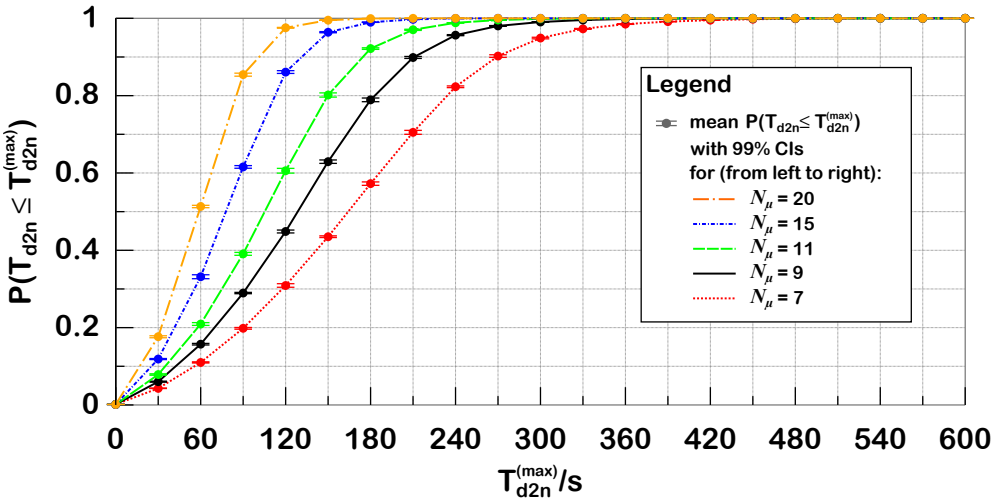


Figure 5.29: *Scenario 8*: Distribution of detection-to-notification delay for various N_μ.

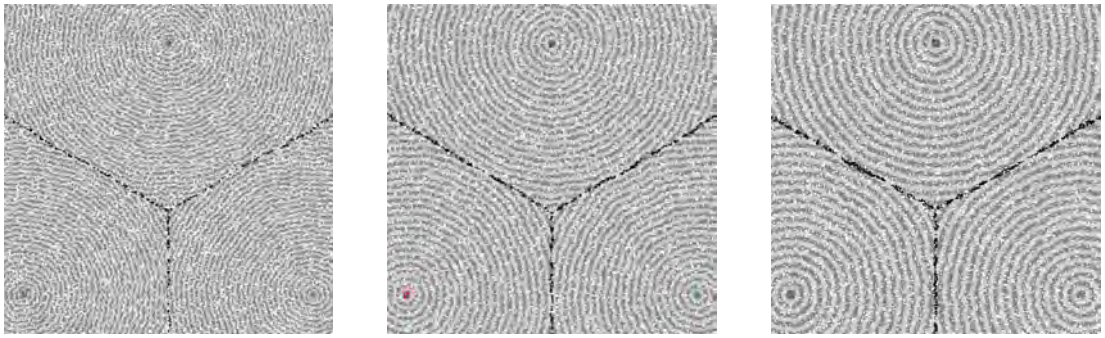
It can be seen that N_μ has a significant influence on $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$. For example, if N_μ is increased from 9 to 20, i.e., the node density is approximately doubled, the detection-to-notification delay that is met by at least 99% can be approximately halved from about 300 s to about 150 s. On the other hand, only about 95% of all events can be expected to be reported after approximately 300 s or less if the expected number of next-hop neighbors reduces from nine to seven. Hence, it is important to ensure sufficient node density during the WSN's operation period to fulfill the aspired detection-to-notification delay requirement. An initial over-provisioning of nodes might help to avoid early re-deployments.

* According to Eq. (4.4).

5.8.9 Scenario 9: Varying the Transmission Range

Scenario 8 indicates that the node density $\overline{N}_d$ influences the expected number of next-hop neighbors $\overline{N}_{\mathrm{r}^-} = N_\mu$ and shows that $\overline{N}_{\mathrm{r}^-}$ significantly influences $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$. The value of $\overline{N}_{\mathrm{r}^-}$, however, does not only depend linearly on $\overline{N}_d$, but also quadratically on the nodes' transmission range d_{T}.* Moreover, changing d_{T} also significantly influences the hop count distribution $P(d_{\mathrm{h}} = h)$, which also effects $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ via Eq. (5.39).

The influence of d_{T} on $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ is the main focus of *Scenario 9*. For the sake of conciseness, the sink placement is assumed to follow Fig. 5.17(f)†. The retrial rate ν is set to $65\,\mathrm{s}^{-1}$ and $d_{\mathrm{T}} \in \{75\,\mathrm{m}, 100\,\mathrm{m}, 125\,\mathrm{m}\}$ is varied. Note that for d_{T} equal to 75 m, 100 m, and 125 m, N_μ approximately takes the values 5, 9, and 15, respectively. N_λ is set to five.‡ All other parameters are chosen similarly to *Scenario 1*.



(a) $d_{\mathrm{T}} = 75\,\mathrm{m}$. **(b)** $d_{\mathrm{T}} = 100\,\mathrm{m}$. **(c)** $d_{\mathrm{T}} = 125\,\mathrm{m}$.

Figure 5.30: *Scenario 9*: Simulated hop count rings for different transmission ranges.

Figures 5.30(a), (b), and (c) show exemplary MASON simulation results of the hop-count rings for transmission range d_{T} equal to 75 m, 100 m, and 125 m, respectively.

The corresponding hop-count distributions are provided in Fig. 5.31§. Based on these results, $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ is derived for each d_{T} and illustrated in Fig. 5.32. Under the given assumptions, the results show that for $d_{\mathrm{T}} = 100\,\mathrm{m}$ the WSN is able to report 99% of the detected events within $T_{\mathrm{d2n}}^{(\max)} \approx 240\,\mathrm{s}$. For $d_{\mathrm{T}} = 125\,\mathrm{m}$, the corresponding $T_{\mathrm{d2n}}^{(\max)}$ even reduces to approximately 120 s. For $d_{\mathrm{T}} = 75\,\mathrm{m}$, however, the $T_{\mathrm{d2n}}^{(\max)}$ needed to report at least 99% of all events increases to approximately 600 s. Only about 48% of the events can be expected to be reported within the aspired 300 s. These results highlight the significant influence of the transmission range on the detection-to-notification delay.

* See Eqs. (4.4) and (4.3) of Section 4.3.2.3.

† Placement F; rotated equilateral triangle.

‡ To allow $N_\mu \geq N_\lambda$ for all investigated d_{T}. According to *Scenario 8*, this change has no significant influence on $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$. Moreover, *Scenario 9* primarily focuses on comparing the different results obtained for $N_\lambda = 5$.

§ Results are obtained similar to the process described in Section 5.7.1. Again, the means with 99% CIs of five simulation runs each are shown. Note that curve $d_{\mathrm{T}} = 100\,\mathrm{m}$ of Fig. 5.31 corresponds to curve *Placement F* of Fig. 5.18.

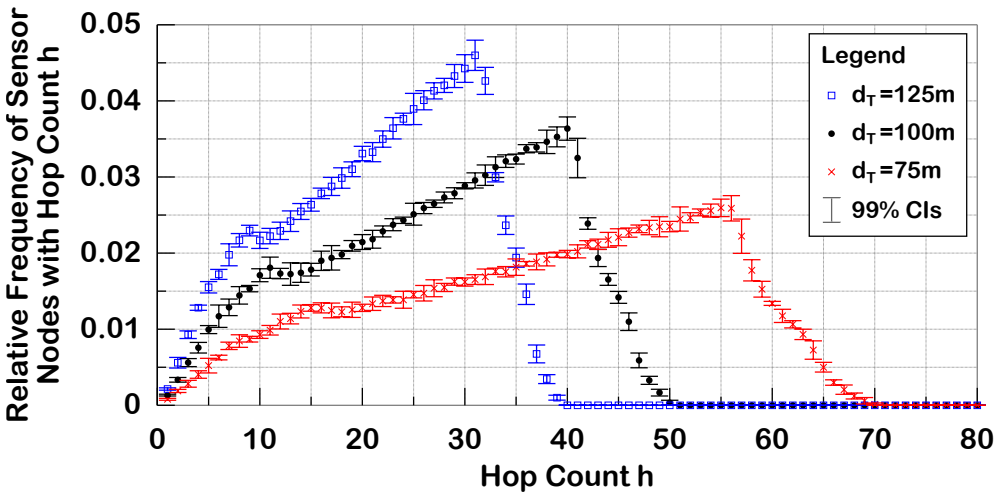


Figure 5.31: *Scenario 9*: Simulated hop count distributions for different transmission ranges.

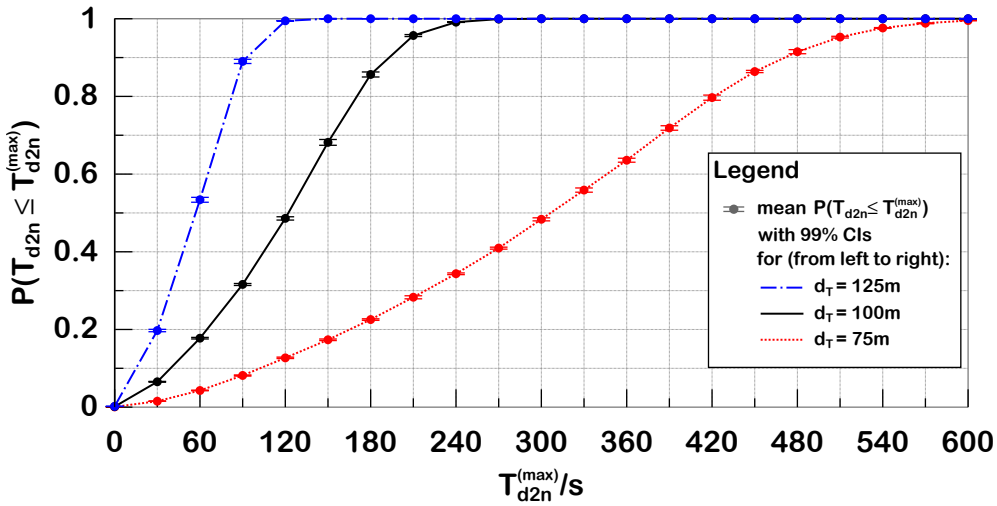


Figure 5.32: *Scenario 9*: Distribution of detection-to-notification delay for various d_T.

5.9 Chapter Summary

Chapter 5 is devoted to the quantitative evaluation of XLMMP. It focuses on investigating the influence of the WSN's and environmental parameters on the detection-to-notification delay in the face of an upcoming fire event. Of particular interest are the parameters τ (repair rate) and δ (failure rate), which govern the nodes' active/sleep period and hence their lifetime.

For the quantitative evaluation, two models are presented: the single-hop model and the multi-hop model. The *single-hop model* is tailored towards obtaining the single-hop response time, i.e., the time needed by an EVM to pass one hop of the multi-hop communication, and its distribution. The model is based on a variant of finite-source retrial queues that comprises unreliable servers and a limited capacity. It can be evaluated by using the four approaches summarized in Section 5.6.1.

The *multi-hop model* addresses the source-to-sink detection-to-notification delay and its distribution. The corresponding results are obtained using Monte Carlo experiments. These experiments are based on the results of the single-hop model and on the distribution of the hop count, which is derived from an agent-based simulation.

Interesting results are obtained and discussed. They show that, under the given model assumptions, the detection-to-notification delay and lifetime requirements can be met by XLMMP (and related protocols), as long as relevant parameters are chosen appropriately. Investing further effort in a more detailed investigation of XLMMP seems appropriate.

The results also allow to identify the parameters that mainly influence the detection-to-notification delay within the investigated scope. In particular, the retrial rate, the active/sleep ratio, the node density, and the transmission range have a strong effect. These insights on the high sensitivity of the detection-to-notification delay on these parameters point towards the aspects of the system that should be modeled in more detail as a next step.

For example, the high sensitivity of the numerical results with respect to the transmission range stresses the need for the inclusion of more realistic models of radio propagation in forest environments. Since also the retrial rate has a significant effect on the results, the influence of changing the retrial time distribution should be investigated in future work.

Chapter 6

Conclusion

> *Think outside the limitations of existing systems—imagine what might be possible; but then do the hard work of figuring out how to get there from the current state of affairs. [...] It may take decades to implement this, mission by mission, but to paraphrase: "A man's reach should extend his grasp, or what are the heavens for?"*
>
> — Vinton G. Cerf [165, p. 415]

This thesis presents the design of a novel communication protocol for long-term and large-scale environmental event monitoring using WSNs. The protocol is referred to as *cross-layer message-merging protocol* (XLMMP) and tailored towards the energy-efficient and timely communication of detected events. The thesis also makes a considerable contribution to the research field of retrial queues by deriving the waiting time distribution in finite-source retrial queues with unreliable servers. Additionally, an approximation of this distribution is developed that allows to efficiently estimate the waiting time distribution in case of large state spaces.

This chapter summarizes the design of XLMMP and its qualitative and quantitative evaluation. Additionally, it sketches further steps towards a testbed implementation of XLMMP. Finally, the implications of this thesis on the research landscape is discussed.

6.1 Summary of Protocol Design, Evaluation, and Results

This thesis investigates the application of WSNs for long-term and large-scale environmental monitoring. For the scenario of early forest fire detection, the spatial resolution, event-to-notification delay, cost, and environmental compatibility are identified to be among the most important properties. It is shown that WSN-based fire detection methods have several advantages (e.g., increased spacial and temporal resolution by placing them closer to the ground, where fires usually start) in comparison to classical forest fire detection methods (e.g., human observers and technical solutions including satellite- or camera-based methods). While related work already shows that, in principle, working WSN-based forest fire detection methods can be implemented using off-the-shelf WSN hardware, these prototypical solutions fail to address important issues, including the required node density, resulting network size, node size limitations, and the resulting trade-off between network lifetime and detection-to-notification delay. These aspects are explicitly addressed in this thesis. Based on a typical forest scenario

inspired by the Neuburg Forest (size of approx. $70\,\mathrm{km}^2$) near Passau, Germany, and state-of-the-art wireless sensor technology, the thesis discusses realistic boundary conditions that influence key network parameters, including the number of nodes (approx. 10^5), node density (approx. $1400\,\mathrm{km}^{-2}$), transmission range (approx. 100 m), energy available for communication (approx. 2400 J), aspired network lifetime (about three years), or the required sleep ratio of the transceivers (approx. 99.96%).

The constraints determined by these parameters suggest the selection of a set of basic communication protocol mechanisms and properties whose combination lead to the design of the novel protocol XLMMP. XLMMP follows a cross-layer approach, fosters message-merging, is ID- and address-less, avoids the need for time-synchronization, provides means for node localization, is unclustered, and uses receiver-based routing. By comparison of the design of XLMMP with general design principles of WSNs (cf. [143, Sec. 3.3]), this thesis shows that XLMMP achieves distributed organization, in-network processing, adaptive fidelity and accuracy, and data centricity. It exploits location information, application-layer activity patterns, and cross-layer optimization. The suggestion to exploit node heterogeneity is followed in a limited form. This approach eases deployment and avoids single points of failure.

The strength of the protocol is that, as long as there is no event, no communication—apart from infrequent, manually triggered network management operations— is needed after an initial setup phase, during which hop count information is distributed. This significantly increases the network's lifetime, since event-less periods dominate in the investigated class of applications. In particular, the proposed WSN architecture does not require frequent time synchronization or regular updates of some clustered topology. The manually triggered network management operations include, e.g., checking the WSN's coverage or updating the hop count information. It should be noted, however, that the network operation is adaptive to outdated hop count information.

The thesis illustrates that the nodes, which likely will be deployed randomly (e.g., by dropping them from a plane) due to their large number, can be made aware of their physical locations using a range-free, hop count-based localization method. The same hop count information is exploited in receiver-based routing. Moreover, the thesis offers a solution to represent the events' identification and location by a minimum amount of information using event thresholds and a tessellation of the monitored area based on a hexagonal grid. Beyond that, a scheme is presented that allows to efficiently address location ranges within the grid-based coordinate system. This scheme fosters the in-network data aggregation by allowing for merging messages whose event locations can be mapped to the same location range. The necessary transformations between the hop count-based virtual node locations within the WSNs topology, the physical node locations within the monitored area, and the grid-based locations are developed and provided in the thesis. A modification of the grid's cell size, which can even be done at runtime using operator-triggered sink messages (SMSGs), allows to trade-off granularity of reported event locations against the degree of message merging.

The thesis shows that the protocol's main purpose is achieved with a very small size (up to 70 bit) of event data messages (EVMs). This allows XLMMP to work without RTS/CTS control packets, similar to the IEEE 802.15.4 (cf. [127]) standard. The thesis also shows that this has two major advantages. First, application-layer data can be used for calculating the receivers' back-off delay, which is needed for receiver contention executed by receiver-based routing. Second, the communication can be sped up significantly—especially in the low-traffic case, which is the dominant case in long-term monitoring of rare environmental events. The

thesis illustrates the systematic speedup by comparing the message sequence charts of XLMMP with XLP (cf. [320]), which is a related cross-layer protocol that also applies receiver-based routing but uses RTS/CTS mechanisms.

A significant part of the thesis is devoted to the quantitative analysis of XLMMP. Here, the thesis focuses on the case of an upcoming fire event and the investigation of the influence of the active/sleep periods on the detection-to-notification delay. The evaluation model is divided into a *single-hop model* and a *multi-hop model*.

The *single-hop model* focuses on determining the time between a node sending an EVM to its neighbors and receiving the corresponding ACK (i.e., updated EVM with decreased hop count) by one of its potential next-hop nodes. To exploit the strengths of different evaluation methods (Markovian steady-state analysis, method of phases, discrete-event simulation) and to allow cross-verification* of the model implementations, the single-hop model is constructed using different modeling formalisms (retrial queue, generalized stochastic Petri net, colored Petri net, continuous-time Markov chain). For implementing these models, the thesis relies on established software tools (e.g., MOSEL-2, SPNP, CPN Tools, Octave). In particular, this thesis is the first work that does not only discuss mean performance measures but, using the method of phases, also derives the distribution of an orbit-visiting customer's waiting time for finite-source retrial queues with unreliable servers. The thesis investigates the scalability and accuracy of the different evaluation methods with respect to the underlying state space and involved rate parameters (see Section 5.6.1). It can be shown that using Markovian analysis for getting mean performance measures is feasible up to approximately 10^6 states† which translates to models that consider, for example, approximately 200 sources (distinct events) and 110 servers (next-hop nodes) or 140 sources and 150 servers. The thesis shows that the maximum size of evaluable state spaces reduces to approximately $9{\cdot}10^4$ states for the method of phases‡. To bridge this scalability gap (10^6 versus $9{\cdot}10^4$ states) between Markovian analysis and the method of phases, this thesis shows that the distribution of an orbit-visiting customer's waiting time can be approximated by a gamma distribution. The parameters of this gamma distribution can be approximately determined by closed-form expressions. The latter are based on model parameters and mean performance measures derivable from Markovian analysis. They are derived in this thesis by applying symbolic regression provided by the Eureqa tool. Solving these closed-form expressions and working with the gamma distribution significantly reduces the computational effort compared to applying the method of phases and working with the corresponding phase-type distribution. In the context of this thesis, the obtained approximation is considered as being sufficiently exact. Especially when planning to use the approximation in a more general context or under an even broader range of parameters, a more rigorous analysis of the its quality and limits is suggested, though.

The *multi-hop model* investigates the distribution of the (end-to-end, i.e., source-to-sink) detection-to-notification delay. For this, the model combines the distribution of the single-hop response time, which is obtained from the single-hop model, with the hop count distribution. The latter is obtained for different scenarios (with variable number of sinks, sink placements, and transmission ranges) from a multi-agent simulation model implemented in this thesis using the MASON toolkit. The simulation results show that the maximum distance between any sink

* See also Section 6.2.

† The absolute numbers depend on implementation and hardware details. However, the given numbers derived based on the setup used in the thesis (see Section 5.4.3.1) should suffice as a rough reference.

‡ When trying to derive the second moment of the underlying phase-type distribution.

and any node lies in the order of 90 hops in the investigated scenario. Since the application of the 90-fold convolution of the single-hop response time distribution with itself seems infeasible, this thesis implements the multi-hop model in form of a Monte Carlo experiment.

Quantitative results are obtained for a set of different scenarios where several system parameters (number of sinks, retrial rate, maximum back-off delay, wake-up rate, active/sleep ratio, traffic load, node density, transmission range) are varied. The results illustrate that, when choosing the system parameters appropriately, the aspired maximum detection-to-notification delay of five minutes (cf. Section 4.2.9) can be achieved by XLMMP for more than 99% of events—despite of the nodes' low activity (0.04%, cf. Section 4.2.8), which allows a lifetime of approximately three years. Hence, the quantitative results show the feasibility of XLMMP in the investigated scenario.

However, as discussed in Section 5.1.2, the applied models are subject to general abstractions (e.g., requiring exponentiality of the involved time parameters) for keeping the models mathematically tractable. Since this thesis does not yet aim at providing a detailed, more realistic implementation of XLMMP, e.g., in form of realistic simulations including radio models and fire spread models or in form of testbed implementations of the protocol, a validation of the proposed models against these unavailable protocol implementations is not yet possible.* However, the thesis assesses the sensitivity of key performance measures on modifications of the model parameters. Therefore, the thesis is able to suggest model parameters for which the corresponding assumption should primarily be relaxed by model refinements in future work. More concretely, the results show that the detection-to-notification delay is mainly influenced by the following parameters: senders' retrial rate, nodes' active/sleep ratio, node density, and nodes' transmission range. Compared to the effort of providing them, the number of sinks only has medium effect on the WSN's performance. Only weak influence is caused by changes of the absolute values of the durations of the active and sleep periods as long as their ratio is kept constant. Moderate variations of the traffic load and of the duration of the nodes' back-off delay used for receiver contention also show only little influence. The high sensitivity of the results on changes of the transmission range stresses the importance to include more realistic radio models in future, more detailed investigation. The significant effect of the retrial rate suggests to investigate changes of the retrial time's distribution in future work. In general, relaxing the assumptions comes with significant additional effort†, some assumptions can likely be relaxed by relatively simple generalizations of the proposed models‡.

6.2 Towards Implementation and Validation

Before briefly discussing future research steps towards a testbed implementation of the proposed forest fire-detecting WSN, the interpretation of the terms *validation* and *verification* are recapitulated in this section. In this thesis, following [135, p. 413, 420], [355, p. 367], and

* See also Section 6.2.

† For example, the inclusion of realistic random node deployment models or radio propagation models. The suitable implementation of the latter are an open issue for many simulation tools that address WSNs (see, e.g., [288]).

‡ For example, by using phase-type instead of exponential distributions for timed parameters, the exponentiality assumption can be relaxed at the expense of the Markovian models' scalability. Alternatively, non-Markovian models can be tackled by modifying the DES implementation (in CPN Tools) of the model proposed in this thesis.

[34, p. 367], *validation* refers to the process of getting confidence in the statement that *something suitable is implemented*, while *verification* refers to the process of getting confidence in the statement that *something is implemented correctly*. In the context of this thesis, this *something* can refer to the proposed WSN architecture (including the proposed WSN protocol) as well as to the models built to evaluate the proposed WSN architecture. Hence, there are four combinations that finally require observation:

- The *validation of the proposed WSN architecture* aims at demonstrating that the architecture is designed suitably by showing that it meets the application-specific requirements.
- The *verification of the proposed WSN architecture* aims at showing that the designed architecture is implemented correctly.
- The *validation of the applied models* checks whether the models sufficiently reflect the aspired real system.
- The *verification of the applied models* investigates whether the models and their evaluation are implemented correctly.

It should be noted that there are different methods to achieve the sought confidence. For example, there are formal (e.g., using formal proofs of correctness) and non-formal (e.g., using tests) verification methods (cf. [355, p. 367]).

Figure 6.1 graphically summarizes the different aspects (i.e., specifications, models, evaluations, as well as validation and verification processes) addressed in this thesis (left half of the figure) and sketches the further steps towards a testbed implementation and ultimately a customer-ready product (right half of the figure). The figure shows six main groups (or layers) of objects and methods (from top to bottom): the *Application Requirement Specification* (labeled with A.1), the *WSN Architecture Specification* (A.2), the *Models and Implementations* (A.3), the *Evaluation Methods* (A.4), the *Evaluation Results* (A.5), and the *Validation* (A.6). The arrows labeled with B.* illustrate the interactions and interdependencies between these layers with respect to the specification, design, modeling, and evaluation process. Arrows labeled with C.* represent the input and output of validation, while the dashed arrows labeled with D (between A.5.1 to A.5.3 and A.4.1 to A.4.3) illustrate the cross-verification of different evaluation methods applied to the single-hop model (A.3.1.1). The E-labeled stippled arrow indicates that successful validation (A.6.2) of the WSN's design (A.2) motivates further steps towards a testbed implementation of the proposed WSN.

In the thesis, the application is investigated first (Chapter 3) and application-specific requirements are deduced (Section 4.2). These requirements are reflected by A.1 and lay the basis (via B.1) for the design of the proposed WSN architecture and protocol (A.2) described in Chapter 4. Several models (in A.3) are defined in Chapter 5 that address different aspects of the WSN. The modeling process itself fosters refinement (B.3) of the WSN's specification. The evaluation of the models (using the methods stated in A.4) leads to various quantitative results (A.5). These evaluation results mainly serve as input to the validation process (A.6). In particular, by comparing quantitative multi-hop results (A.5.6 via C.6) with the specified application requirements (A.1 via C.1), a first validation of the proposed WSN architecture and protocol (A.6.2) can be achieved. On the one hand, this validation provides hints for refining the WSN's specification (A.2 via C.2)—for example, by increasing the retrial rate ν, the aspired detection-to-notification delay can be achieved (see Section 5.8.3). On the other hand,

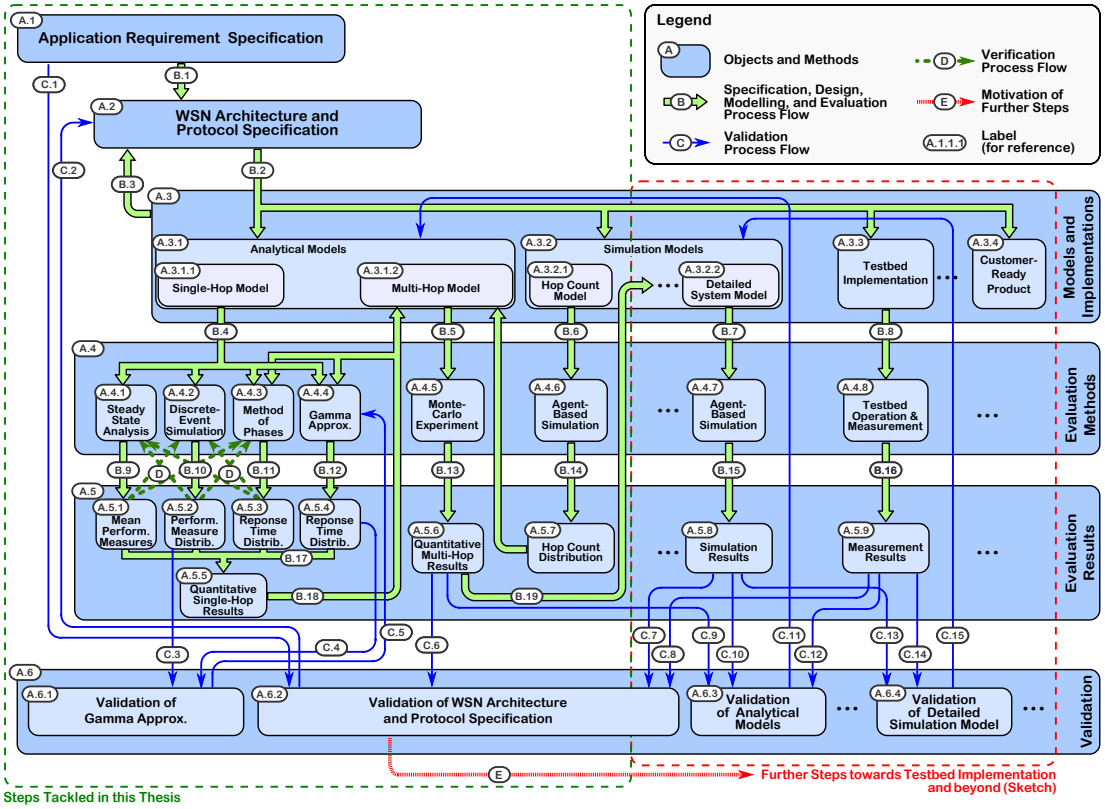


Figure 6.1: Summary of the system design and development process towards a testbed implementation and beyond.

the subsequently achieved successful validation based on the quantitative results (A.5.6) of the multi-hop model (A.3.1.2) increases the confidence in the statement that the proposed WSN (A.2) is able to meet important requirements (in particular lifetime and communication delay constraints) of the addressed application (A.1) and hence motivates (E) following further steps (right half of Fig. 6.1) towards a testbed implementation (A.3.3) of the proposed protocol (A.2) and probably further steps (indicated by "...") towards a customer-ready product (A.3.4).

In this thesis, emphasis is placed on the analytical single-hop model (A.3.1.1, Sections 5.2 to 5.6), since a more general applicability of the model beyond the scope of this thesis is foreseen. Moreover, its quantitative results (A.5.5) serve as important input (via B.18) to the multi-hop model (A.3.1.2). The single-hop model's evaluation is carried out using different methods (A.4.1 to A.4.4) having different advantages and disadvantages (cf. Section 5.6.1). The implementations of methods A.4.1 to A.4.3 can be (cross-)verified, i.e., the confidence in their correctness can be increased, by showing that their results (A.5.1 to A.5.3) coincide (indicated by the dashed verification arrows labeled with D). Additionally, and not reflected in Fig. 6.1 to reduce complexity, related work that proposes comparable models is used for verification (Section 5.3.4). The suitability of the single-hop model's gamma approximation (A.4.4, Section 5.5) can be validated by comparing its results (A.5.4) to DES (A.4.2) results of the single-hop model (A.5.2) carried out in Fig. 5.13 of Section 5.5.2 as indicated by A.6.1 and C.3 to C.5 in Fig. 6.1.

A rigorous *validation* (A.6.3) of the analytical models (A.3.1 via C.11), i.e., increasing the confidence that the analytical models appropriately reflect the real-world scenario and the as-

pired real-world implementation of the WSN, is not the focus of this thesis due to the lack of more detailed representations, e.g., in form of more general analytical models, more detailed simulation models (A.3.2.2), or testbeds (A.3.3). A detailed simulation model (A.3.2.2) could be developed by extending the hop count simulation model (A.3.2.1) developed in this thesis based on the MASON multi-agent simulation toolkit (cf. Section 4.3.2.2). Endeavors aiming at a testbed implementation (A.3.3) of the proposed WSN should start with a survey of current sensor node hardware platforms that allow for modifications of low-layer protocols within their communication protocol stack. This thesis provides sufficient motivation (E) to invest more effort into such models and implementations that provide a higher level of resolution. These would then allow to perform *cross-model validation* (see [355, p. 388]), which is sketched in Fig. 6.1 by A.6.3, A.6.4, and C.9 to C.15. Involving further researchers and research groups into this process (including *face validation*, cf. [46, p. 14]) will foster further optimization of the proposed WSN architecture and protocol.

In an academic environment that addresses computer science, the development and evaluation of a testbed implementation (A.3.3, A.4.8, A.5.9) and exploiting the resulting possibilities (C.8, C.12, C.14) of further validating the analytical and simulation models as well as further refinement and validation the WSN design likely are the final steps conducted towards the customer-ready product (A.3.4). Clearly, producing the latter requires further actions by the industry—including, e.g., market analysis, prototype development, and product manufacturing. The product's lifecycle management (cf. [255]) would then also include marketing, procurement, and customer support (sales, delivery, service, and maintenance).

6.3 Implications on Research Landscape

This thesis further paves the way towards the design and development of viable solutions that exploit the unique strengths of WSNs for environmental monitoring. While the thesis exemplarily focuses on the application scenario of forest fire detection for clarity, the proposed WSN architecture comprising the XLMMP protocol appears to be applicable in comparable scenarios, e.g., pollution detection. The presented quantitative evaluation methods allow to assess the protocols feasibility in a wide range of application-specific environment and network parameters and can be used to optimize network and protocol parameters correspondingly.

The quantitative evaluation carried out in this thesis is based on the modeling formalism of retrial queues. In various related work (cf. Section 2.4), the broad applicability of this type of models is shown. The novel results (exact results and more scalable gamma approximation) obtained in this thesis for the distribution of the waiting time in finite-source retrial queues with unreliable servers further increases the attractiveness of this modeling formalism. Vice-versa, generalizations that have been applied to retrial queueing models (e.g., orbital search or impatient customers) in related work suggest further possible generalizations of the evaluation approaches chosen in this thesis.

The event-centric operation of XLMMP is an important mechanism to achieve high energy efficiency. However, it also limits the protocol's field of application. In particular, the protocol requires the sensor nodes to be able to identify report-worthy events based on the environmental conditions measured by a single node. Hence, the protocol is not directly suitable for applications where the identification of a report-worthy events can only be achieved by combining the sensor data of several nodes. Moreover, report-worthy events need to occur relatively rarely. Applications that foresee the reporting of frequent events benefit likely from

keeping the nodes' active/sleep periods synchronized. A break-even analysis model that allows to estimates the maximum event frequency for which unsynchronized active/sleep periods are still beneficial is an interesting direction of future work.

As summarized in Section 6.2, the implementation of the proposed WSN architecture and protocol in form of a customer-ready product still requires considerable research, design, and development. Open research issues with respect to the protocol are identified in Section 4.4.2. Among the most important issues are the investigation of further resilience and suitable security mechanisms as well as refinement and validation of the design based on more detailed simulations and testbed implementations.

Similar to related work, this thesis suffers from the fact that WSN research still lacks a convenient and unbiased framework for efficient comparison of different WSN protocols. While WSN-targeted simulators and testbeds achieve this to some degree, their use for comparing different protocols require significant implementation effort, in particular due to the lack of a standard procedure. Moreover, the scalability of these approaches is as limited well. For example, the multi-agent DES (implemented using MASON) applied in this thesis allows to simulate the hop count update process of a 10^5-node network with adequate performance. Yet, there is little hope that this scale can be sustained when further aspects of the protocol (e.g., medium access, routing) or application-layer mechanisms (e.g., fire spread) are included in the simulation. While testbeds can, in principle, be implemented arbitrarily close to the tackled real-world scenario, their implementability and scalability strongly depends on the availability of suitable hardware. For example, the scenario and cross-layer protocol design tackled in this thesis would require WSN node hardware that allows modifications to low-level hardware and software mechanisms, including the nodes' low-layer communication protocol implementation. Manifold interesting research questions arise from this problem area.

Addressing an even more general context, this thesis shows that application-specific design is important to achieve optimized WSNs. An important tool for the application-specific design of WSN communication protocols is cross-layering. Cross-layering enables all protocol layers to benefit from application-layer information. In this thesis, this benefit gets apparent, e.g., by the fact that message merging in XLMMP can be fostered by making application-layer data available to the receiver-based routing mechanism. Vice-versa, some application-layer services can be realized more efficiently based on information provided by lower layers. For example, XLMMP achieves node localization without dedicated hardware or localization protocols by using hop count-based topology information. Still, this thesis definitely does not suggest to completely ignore the concept of layered communication due to its manifold advantages. For example, a layered approach helps to achieve modularity, robustness, reusability, stability, and maintainability (cf. [166, 198]). Especially the modularity allows scientists and developers more naturally to focus on subproblems (like localization, routing, medium access, or channel coding). This simplifies the design of development of novel solutions that address these subproblems. Nevertheless, this thesis indicates that, if one does not cling to layered architectures too desperately, reasoned combination of mechanisms that cross the layers' borders can lead to significant improvements.

Appendix A

Abbreviations and Mathematical Notation

A.1 Abbreviations

ACK	acknowledgment
ADC	analog-to-digital converter
BIC	Bayesian information criterion
CCA	clear channel assessment
CDF	cumulative distribution function
CPN	colored Petri net
CPU	central processing unit
CRC	cyclic redundancy check
CSPL	C-based SPN language
CTMC	continuous-time Markov chain
CTS	clear to send
DES	discrete-event simulation
DSSS	direct sequence spread spectrum
DTMC	discrete-time Markov chain
EEPROM	electrically erasable programmable read-only memory
EVM	event message
ExtraSet	extra parameter set defined in Table 5.10
FEC	forward error control

FFS	fastest free server
FHSS	frequency-hopping spread spectrum
FIFO	first in, first out
FSK	frequency shift keying (modulation technique)
FSRQ	finite-source retrial queue
GPS	Global Positioning System
GSPN	generalized stochastic Petri net
HDD	hard disk drive
ID	identifier
IDL	idle signal
IEEE	Institute of Electrical and Electronics Engineers
iff	if and only if
iid	independent and identically distributed
inPool	incoming-EVM pool
IT	information technology
LAN	local-area network
LR-WPAN	low-rate wireless personal area network
LST	Laplace–Stieltjes transform
MAC	medium access control
MSG	message
MTTA	mean time to absorption
NAK	negative acknowledgment
NHF	next hop found message
N/A	not available; not applicable
OOK	on-off keying (modulation technique)
OS	operating system
outPool	outgoing-EVM pool
PAN	personal area network

PASTA	Poisson arrivals see time averages
PDF	probability density function
PH	phase-type (distribution)
PLW	PSDU length word
PFQN	product-form queueing network
ppm	parts per million
PSDU	physical layer convergence protocol service data unit
P2P	peer-to-peer
QBD	quasi-birth–death process
RAM	random-access memory
RCM	receiver contention mechanism
RED	random early detection
RID	receiver initiative determination
ROM	read-only memory
RS	random selection
RealSet	realistic parameter set, i.e., parameter set that is realistic with respect to the application scenario (see Table 5.2)
RSS	received signal strength
RSSI	RSS indicator
RTS	request to send
RX	receiver/reception
ScalSet	scalability parameter set; used for checking the scalability of the evaluation methods (see Table 5.2)
SFD	start frame delimiter
SMSG	sink message
SN	sensor network
SNR	signal-to-noise ratio
SOR	successive over-relaxation
SOS	self-organizing system

SPN	stochastic Petri net
SRN	stochastic reward net
SVD	singular value decomposition
TDMA	time division multiple access
TTA	time to absorption
TX	transmitter/transmission
UAV	unmanned aerial vehicle
VerSet	verification parameter set; used for verification of the models and their evaluation (see Table 5.2)
WLAN	wireless LAN
WPAN	wireless personal area network
WSN	wireless sensor network
WSAN	wireless sensor and actor network
XLM	cross-layer module (cf. [6])
XLMMP	cross-layer message-merging protocol
XLP	cross-layer protocol (cf. [320])

A.2 Notation

The following lists summarize the used symbols for convenient reference. The lists are sorted alphabetically.

A.2.1 General Notation

In general, the following rules are applied (x and X are used as wildcards in this list).

- The letters h, i, j, k, l, m, and n as well as their primed variants (like i', i'', i''', etc.) are flexibly used as counters and may frequently change their interpretation throughout the text.
- Small Greek letters are usually used for rates (in events per second, s^{-1}), with exception of π_x for steady-state probabilities, $\mathring{\pi}$ for the mathematical constant pi, and ρ_x for utilizations.
- Bold letters usually refer to row vectors (small letters) or matrices (capital letters).
- $\mathbf{0}$ denotes a matrix of appropriate size with all elements equal to zero.
- $\mathbf{1}$ denotes a column vector of appropriate size with all elements equal to one.

- **I** denotes an identity matrix of appropriate size with all elements on the main diagonal equal to one and all other elements equal to zero.
- "Blackboard bold" capital letters, like $\mathbb{X}$, refer to sets.
- $|x|$ refers to the absolute value of a scalar x.
- $|\mathbb{X}|$ refers to the cardinality of set $\mathbb{X}$.
- $X(t)$ is the value of (random) variable X at some time instant t.
- $X^{\sim}(s)$ is the LST of random variable X
- $\overline{X}$ refers to the mean-value of random variable X.
- $var(X)$ is the variance of random variable X.

A.2.2 Additional Symbols and Operators

Additionally, the following symbols and operators are used.

$\sim$	$X \sim Y$ means that the random variables X and Y have the same distribution.
$\approx$	$X \approx Y$ means that the value of X is approximately equal to the value of Y.
$*$	convolution operator, i.e., $X * Y(z) = \int_{-\infty}^{\infty} X(z-t)\,\mathrm{d}Y(t)$
∞	infinity
$\wedge$	logical AND
$\vee$	logical OR
d	used in differential operator, e.g., $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$
$e^{\mathbf{X}}$	matrix exponential of matrix $\mathbf{X}$ defined as $e^{\mathbf{X}} = \sum_{n=0}^{\infty} \frac{1}{n!}\mathbf{X}^n$
$E[X^k]$	the k-th moment of random variable X, i.e., $E[X] = \overline{X}$
$f_X(x)$	the PDF of random variable X, i.e., $f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x}$
$F_X(x)$	the CDF of random variable X, i.e., $F_X(x) = P(X \leq x)$
$\mathrm{ld}\,(x)$	binary logarithm (logarithmus dualis) of variable x
mod	modulo operator
$P(X)$	the probability of event X
$\mathrm{Var}(X)$	the variance of random variable X
↺	self-citation; (co-)authored publication

A.2.3 Units

The following units are used.

bit	bit
B	byte ($1\,\mathrm{KiB} = 1024\,\mathrm{B}$; $1\,\mathrm{kB} = 1000\,\mathrm{B}$)
°C	degree Celsius
ha	hectare ($1\,\mathrm{ha} = 10^4\,\mathrm{m}^2$)
Hz	Hertz (s^{-1})
h	hour
J	Joule ($\mathrm{W\,s}$)
kbps	kilobit per second ($\mathrm{kbit\,s}^{-1}$)
m	meter
min	minute
s	second
W	Watt ($\mathrm{J\,s}^{-1}$)

A.2.4 List of Main Variables and Parameters

For convenient reference, the following list summarizes all variables and parameters used in this thesis ($\star$ is used as wildcard here) in alphabetical order. Greek letters are sorted according to their modern Greek transliteration (following ISO 843:1997).

$A_\star$	size of a spatial area (in square meters, m^2)
A_C	size of the area covered by one cell of the hexagonal localization grid (in square meters, m^2)
A_F	size of the monitored area of interest, i.e., of the forest (in square meters, m^2)
A_r	size of the part of the tagged sensor node's transmission range that lies within the same ring (in square meters, m^2)
A_{r^-}	size of the transmission ranges' segment that lies within the next inner ring (in square meters, m^2)
A_{r^+}	size of the transmission ranges' segment that lies within the next outer ring (in square meters, m^2)
A_S	size of the area monitored by a single sensor node (in square meters, m^2)

A_{T}	size of the area covered by the transmission range of a single sensor node (in square meters, m^2)
$\mathbf{A}_{ij}$	layer submatrix of $\mathbf{T}$ referring to transitions from layer i to j
α_γ	shape parameter of gamma distribution
$\mathbf{B}_{ij}^{(kl)}$	level submatrix of $\mathbf{A}_{ij}$ referring to transitions from level k of layer i to level l of layer j
$C_{\star\star}$	coefficients of the α_γ approximations
$C_{b,\star}$	back-off delay constant for parameter $\star$
$d_\star$	a spatial distance or range (in meters, m, or number of communication hops)
d_{C}	radius of a cell of the hexagonal localization grid (in meters, m)
$d_{\mathrm{f}}(m,n)$	field distance between two sensor fields m and n
$d_{\mathrm{h}}^{(x)}$	a sensor node's distance to Sink x, $x \in \{0,1,2\}$, measured in number of communication hops
d_{h}	a sensor node's hop count, i.e., the minimum distance $\min\left(d_{\mathrm{h}}^{(0)}, d_{\mathrm{h}}^{(1)}, d_{\mathrm{h}}^{(2)}\right)$ to any Sink x, $x \in \{0,1,2\}$, measured in number of communication hops
$d_{\mathrm{hf}}(m,n)$	hop distance between two sensor fields m and n
$\mathbf{d}_\star$	a row vector of distances denoting a coordinate in a specific coordinate system identified by $\star$: g (hexagonal grid coordinate; two dimensions), h (hop count distances to sinks, virtual coordinate; three dimensions), p (physical coordinate; two dimensions), s (physical distances to sinks; three dimensions); alternative notations: $\mathbf{d}_\star = \langle x_\star, y_\star \rangle = \langle x, y \rangle_\star$ (equivalently for three dimensions); example: $\mathbf{d}_{\mathrm{h}} = \langle d_{\mathrm{h}}^{(0)}, d_{\mathrm{h}}^{(1)}, d_{\mathrm{h}}^{(2)} \rangle$
$\mathbf{d}_{\mathrm{h}}$	vector of a sensor node's distances to the three sinks measured in number of communication hops, i.e., $\mathbf{d}_{\mathrm{h}} = \langle d_{\mathrm{h}}^{(0)}, d_{\mathrm{h}}^{(1)}, d_{\mathrm{h}}^{(2)} \rangle$
$d_{\mathrm{p}}^{(x)}$	a sensor node's physical distance to Sink x (in meters, m)
$d_{\mathrm{p}}^{(\max)}$	the maximum physical distance between two sensor nodes; diameter of the monitored area (in meters, m)
d_{r^-}	distance of tagged sensor node to next inner ring (in meters, m)
$d_{\mathrm{r}^-}^{\mathrm{norm}}$	estimated normalized distance of tagged sensor node to next inner ring (dimensionless, $0 \leq d_{\mathrm{r}^-}^{\mathrm{norm}} \leq 1$)
d_{r^+}	distance of tagged sensor node to next outer ring (in meters, m)
d_{S}	sensing range (in meters, m)

d_{T}	transmission range (in meters, m)
$\overline{d}_{\mathrm{w}}^{(x)}$	mean width of hop count rings around Sink x (in meters, m)
$\overline{d}_{\mathrm{w}}$	overall mean width of hop count rings, i.e., mean of all $\overline{d}_{\mathrm{w}}^{(x)}$ (in meters, m)
$D_{\star}$	a data length; a number of bits (in bits (bit), or Bytes (B); 1 B=8 bit).
D_{m}	length of incident message (in bits, bit)
D_{n}	memory capacity of sensor node (in bits, bit)
δ_{idle}	failure rate of idle servers of a queueing system (in failures per second, $1/\mathrm{s}$)
δ	failure rate of idle servers of a queueing system (in failures per second, $1/\mathrm{s}$)
E_r	remaining energy of a wireless sensor node (in Joule, J)
$E^{(\max)}$	maximum energy that can be used by a wireless sensor node (in Joule, J)
E_{trans}	maximum energy that can be used by a wireless sensor node's transceiver (in Joule, J)
ε_r	relative permittivity of transmission medium
I_{mm}	indicator function of event message mergeability
λ	arrival rate of an infinite-source queueing system; per-source generation rate of a finite-source queueing system (in arrivals per second, $1/\mathrm{s}$)
$\widetilde{\lambda}$	overall generation rate evoked by the job-generating sources of a finite-source queueing system (in arrivals per second, $1/\mathrm{s}$)
$\overline{\lambda}$	mean overall arrival rate to a finite-source queueing system (in arrivals per second, $1/\mathrm{s}$)
$\overline{\lambda_i}$	mean overall arrival rate to an idle server (in arrivals per second, $1/\mathrm{s}$)
μ	service rate of a single-server queueing system; per-server service rate of a multi-server queueing system (in services per second, $1/\mathrm{s}$)
μ_γ	rate parameter of gamma distribution (unit depending on described random variable)
$n_{\mu\mathrm{b}}^{(n_{\mu\mathrm{f}})}$	maximum number of busy servers when there are $n_{\mu\mathrm{f}}$ failed servers
$\mathbb{N} = \{0, 1, \ldots\}$	set of non-negative integer numbers
$\mathbb{N}^+ = \{1, 2, \ldots\}$	set of positive integer numbers
$N_{\star}$	refers to some real or natural unit-less number
N_{c}	number of EVMs that can be simultaneously handled/stored by a sensor node and that are acknowledged by this sensor node; capacity of retrial queueing system

N_{C}	number of hexagonal location grid cells in the area of interest
N_d	number of sensor nodes per unit area; sensor node density (in nodes per square kilometer, $1/\mathrm{km}^2$)
$N_{\mathrm{g}}^{(\max)}$	estimated maximum number of grid cells between two sensor nodes
$N_{\mathrm{g}}^{(\mathrm{upper})}$	upper bound of the maximum number of grid cells between two sensor nodes
$N_{\mathrm{h}}^{(\max)}$	estimated maximum number of communication hops between sensor nodes and sinks
$N_{\mathrm{h}}^{(\mathrm{upper})}$	upper bound of the maximum number of communication hops between sensor nodes and sinks
$N_{\mathrm{hcu}}^{(\max)}$	maximum number of local hop count updates per sensor node
N_{in}	current number of EVMs currently queued in a sensor node's inPool
$N_{\mathrm{in}}^{(\max)}$	capacity of a sensor node's inPool
N_λ	number of sources in a finite-source queueing system
$N_{\lambda\mathrm{b}}$	number of busy (i.e., job-generating) sources in a finite-source queueing system
$N_{\lambda\mathrm{bb}}$	number of busy (i.e., job-generating) but blocked sources in a finite-source queueing system
$N_{\lambda\mathrm{bb}}$	number of busy (i.e., job-generating) and non-blocked sources in a (not full) finite-source queueing system
N_m	maximum number of EVMs storable at a sensor node
N_μ	number of servers in a multi-server queueing system
$N_{\mu\mathrm{b}}$	number of busy servers in a multi-server queueing system
$N_{\mu\mathrm{f}}$	number of failed servers in a multi-server queueing system
$N_{\mu\mathrm{i}}$	number of idle servers in a multi-server queueing system
$N_{\mu\mathrm{o}}$	number of operational servers in a multi-server queueing system
$N_{\nu\mathrm{b}}$	number of busy orbit entities (i.e., number of orbiting customers) in a retrial queueing system
N_{N}	number of neighbors of a tagged sensor node
N_{Na}	number of active neighbors of a tagged sensor node
N_{Nar}	number of active same-ring neighbors of a tagged sensor node

N_{Nar^-}	number of active inner-ring neighbors of a tagged sensor node
N_{Nar^+}	number of active outer-ring neighbors of a tagged sensor node
N_ν	maximum number of jobs in the orbit of a retrial queueing system
$N_{\nu\mu}$	number of jobs at the queueing system (in orbit or service)
N_{out}	current number of EVMs currently queued in a sensor node's outPool
$N_{\text{out}}^{(\max)}$	capacity of a sensor node's outPool
N_{r}	number of same-ring neighbors of a tagged sensor node
N_{r^-}	number of inner-ring neighbors of a tagged sensor node
N_{r^+}	number of outer-ring neighbors of a tagged sensor node
N_{ret}	number of retrials before an arriving job enters server of a retrial queueing system
$N_{\text{ret,o}}$	number of retrials before an orbit-visiting job enters server of a retrial queueing system
N_S	overall number of sensors deployed in the area of interest
ν	retrial rate of a retrial queueing system (in retrials per second, $1/\text{s}$)
$p_\star \in (0,1)$	some probability
p_{block}	blocking probability, i.e., probability that an arriving job experiences a full queueing system
p_{i}	probability that a tagged node is isolated, i.e., the node has no neighbors
p_{failall}	probability that all servers of an unreliable queueing system are failed
p_{full}	probability that the queueing system is full and blocks all arrivals
$p_\lambda(n_{\mu\text{b}}, n_{\nu\text{b}}, n_{\mu\text{f}})$	probability that an arriving job sees the retrial queueing system in state $(n_{\mu\text{b}}, n_{\mu\text{b}}, n_{\mu\text{f}})$
p_{oper}	probability that a server of an unreliable queueing system is operational, i.e., in busy or idle state (not failed)
p_ν	retrial probability, i.e., probability that an incoming job needs to enter the orbit because none of the servers is idle
p_{sleep}	probability that a selected transceiver is in sleep mode
P_{active}	estimated power consumption of an active transceiver (in mode TX/RX/idle)
$\overline{P}_{\text{trans}}$	mean overall power consumption of a transceiver

P_{sleep}	estimated power consumption of a transceiver in sleep mode
$\pi \in [0,1]$	a steady-state probability
$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots)$	a vector of steady-state probabilities
$\mathring{\pi} = 3.1416\ldots$	the mathematical constant *pi*
$\mathbf{Q}$	an infinitesimal generator matrix (rate matrix) of a CTMC
$\mathbb{R} = (-\infty, \infty)$	set of real numbers
$\mathbb{R}^+ = (0, \infty)$	set of positive real numbers
$\mathbb{R}_0^+ = [0, \infty)$	set of non-negative real numbers
ρ	server utilization in a queueing system
$t \in \mathbb{R}_0^+$	instant of time
$T \in \mathbb{R}_0^+$	an amount of time, duration, or delay (in seconds, s)
T_{a2d}	detectable-to-detection delay (in seconds, s)
T_b	back-off delay used in receiver contention (in seconds, s)
$T_b^{(\max)}$	maximum back-off delay (in seconds, s)
T_{d2n}	detection-to-notification delay (in seconds, s)
$T_{\text{d2n}}^{(\max)}$	investigated upper bound of detection-to-notification delay (in seconds, s)
T_{e2a}	event-to-detectable delay (in seconds, s)
$T_{\text{e2d}} = T_{\text{e2a}} + T_{\text{a2d}}$	event-to-detection delay (in seconds, s)
$T_{\text{e2n}} = T_{\text{e2d}} + T_{\text{d2n}}$	event-to-notification delay (in seconds, s)
T_{life}	lifetime of a node (in seconds, s)
T_μ	service time, i.e., time a job spends at the server of a queueing system (in seconds, s)
T_ν	waiting time, i.e., time a job spends at the orbit of a retrial queueing system (in seconds, s)
$T_{\nu\text{d}}$	waiting time of a job that directly joins a free server on arrival; usually, $T_{\nu\text{d}} = 0\,\text{s}$ (in seconds, s)
$T_{\nu\mu}$	response time, i.e., time a job spends at the orbit and server(s) of a retrial queueing system (in seconds, s)
$T_{\nu\text{o}}$	waiting time of an orbit-visiting job (in seconds, s)
T_{prop}	estimated maximum propagation delay of radio signal between two wireless sensor nodes (in seconds, s)

T_{trans}	estimated maximum transmission delay of a message (in seconds, s)
$\mathbf{T}$	submatrix of infinitesimal generator matrix $\widetilde{\mathbf{Q}}$ referring to the transitions between transient states
τ_{idle}	repair rate of idle servers in a queueing system (in repairs per second, $1/\text{s}$)
τ	repair rate of idle servers in a queueing system (in repairs per second, $1/\text{s}$)
v_c	the speed of light in vacuum (in meters per second, $\text{m}\,\text{s}^{-1}$)
v_{air}	the speed of light in air (in meters per second, $\text{m}\,\text{s}^{-1}$)
v_{forest}	the speed of radio waves in forested areas (in meters per second, $\text{m}\,\text{s}^{-1}$)
$X(t) \in \mathbb{X}$	stochastic process
$\mathbb{X}$	state space of stochastic process

Appendix B

Elaborated Investigations and Derivations

B.1 Related Protocols

Tables B.1 to B.3 summarize related work on protocols that are based on (MAC-layer) anycast, opportunistic routing, and/or receiver contention. A more compact version of the tables is provided by Table 2.2. The protocols are sorted chronologically according to the oldest corresponding reference found by the involved authors. In the tables, the terms *receiver initiative determination* and *receiver contention mechanism* are abbreviated as RID and RCM, respectively.

Note that [6] proposes the *cross-layer module* (XLM). According to [320], the publication [6] is a "preliminary version" of [320], which proposes the *cross-layer protocol* (XLP). Since the basic mechanisms of XLM and XLP coincide, publications [6] and [320] are jointly discussed in Tables B.1. Publications [198] and [3, Sec. 10.3] are examples of secondary literature that discuss XLM and XLP, respectively.

Table B.1: Related work on protocols based on anycast, opportunistic routing, and/or receiver contention

Ref.	Initial Sender Packet	Next-Hop Selection	Decision Based on	Details
[206]	○ DATA; contains metric of distance between sender and base station	○ RCM ○ first next-hop DATA packet forward cancels other next-hop candidates	○ RCM: distance metric or RSSI	○ idea is only roughly presented ○ nodes need to be able to estimate their distance to the sink or measure the RSSI ○ no end-to-end delay investigated ○ no support of aggregation foreseen
[44]	○ CONTROL: Open RTS (ORTS); contains location of sender	○ RID and RCM ○ first neighbor that sends CTS is selected	○ RID: location ○ RCM: distance progress, remaining energy, random value	○ all candidate nodes need to be in mutual transmission range ○ nodes need to know their location (fine-grained) ○ node IDs/addresses needed (to explicitly address the selected receiver) ○ no support of aggregation foreseen
[365] [366]	○ indefinite; either CONTROL (RTS) or DATA packet ○ packets must contain own location and destination's location ○ solution with RTS and dual radio is investigated in more detail	○ RCM ○ first neighbor that sends CTS is selected	○ RCM: based on location/distance progress only	○ nodes need to know their own and the destination's location (fine-grained) ○ uses "broadcast addresses" ○ suggests to use two radios for busy tones ○ node IDs/addresses needed (to explicitly address the selected receiver) ○ no support of aggregation foreseen
[63]	○ links to the anycast group (set of candidate neighbors) are probed sequentially ○ format of (control/data) packets needed for probing is not discussed	○ anycast group is selected by sender's routing layer ○ instantaneous link probing: links are probed link-by-link ○ order and number of retransmissions are selected by sender ○ first successful link determines next hop	○ order and number of retransmissions are selected by the sender based on the neighbors' recent behavior, e.g., responsiveness to previous transmissions, overheard transmissions, or a-priory knowledge of their sleep schedules	○ sender-based anycast ○ no RCM ○ node IDs/addresses needed (to explicitly address the receivers sequentially) ○ no support of aggregation foreseen
[283]	○ CONTROL: RTS; contains sender's and destination's location	○ RCM ○ first neighbor that sends CTS is selected	○ RCM: only distance progress is investigated (to simplify analysis); further parameters foreseen (e.g., remaining energy, link quality)	○ nodes need to know their own and the destination's location ○ node IDs/addresses or fine-grained localization needed (to explicitly address the selected receiver) ○ no support of aggregation foreseen
[42]	○ DATA; contains forwarder priority list, unique batch ID, and a batch map;	○ RID and RCM ○ neighbor with highest priority sends DATA packet further and updates other potential receivers	○ RID: a priority list of selected receivers is maintained by the sender (based on corresponding links' delivery probability) and communicated to all receivers together with the DATA packet; ○ RCM: active receivers answer in the order of this list	○ all source nodes need to know the complete set of inter-node loss rates (requires regular flooding of link state information) ○ only 90% of packets sent opportunistically, the remaining 10% by traditional routing ○ focus on throughput (batches of packets) ○ node IDs/addresses needed (for list maintenance and for traditional routing) ○ no support of aggregation foreseen
[6] [198] [3, Sec. 10.3] [320]	○ CONTROL: RTS; contains location of sender and sink	○ RID and RCM ○ first neighbor that sends CTS is selected	○ RID: SNR, traffic limit, buffer limit, energy, feasible region ○ RCM: distance progress; [320] includes angle-based routing	○ nodes need to know their own location (fine-grained) ○ no IDs needed; locations used instead ○ no support of aggregation foreseen

Table B.2: Related work on protocols based on anycast, opportunistic routing, and/or receiver contention (cont'd)

Ref.	Initial Sender Packet	Next-Hop Selection	Decision Based on	Details
[163]	◦ CONTROL: Request to Relay (RTR); contains one-hop random MAC address, tier ID, NAV entry (indicating length of data packet; optional), optional data packet information	◦ RID and RCM ◦ first neighbor that sends clear-to-relay packet (CTR) is selected; CTR contains one-hop MAC address (random) of sender and relay and relay's tier ID	◦ RID: tier ID ◦ RCM: only random backoff to avoid CTR collisions	◦ transmission power control needed ◦ temporary short random IDs needed ◦ aggregation not discussed (but optional data packet information in the RTR could be used to estimate the mergeability)
[89]	◦ CONTROL: RTS; contains location of sender and aggregation ID (AID) which describes the content	◦ RID and RCM ◦ first neighbor that sends CTS is selected	◦ RID: no CTS sent if neighbor is farther from destination than sender and no aggregation is possible ◦ RCM: 3 priority classes based on aggregateability (AID) and distance progress ◦ all CTS are additionally randomly delayed (for avoiding CTS collision)	◦ nodes need to know own and sink's locations (fine-grained) ◦ nodes need to be time-synchronized (optional, when aggregation based on geographic location suffices) ◦ node IDs/addresses or fine-grained localization needed (to explicitly address the selected receiver) ◦ additionally proposes random waiting to increase chance for aggregation
[183]	◦ CONTROL: RTS bursts; packet size 44 bytes; content not discussed	◦ RID and RCM ◦ first neighbor that sends CTS is selected for an extended period (switching from anycast to unicast)	◦ initiative and RCM: based on arbitrary routing metric (in the paper, geographic distance is chosen)	◦ node IDs/addresses or fine-grained localization needed (to explicitly address the selected receiver) ◦ no support of aggregation foreseen
[27] [28] [29]	◦ DATA; no additional information ◦ (using CONTROL packet (RTS) mentioned as variant)	◦ RCM ◦ first neighbor that sends unique bit in signal burst forwards the DATA packet further	◦ RCM: based on location / distance progress	◦ each recipient calculates individual ACK signal burst ◦ each recipient sends individual ACK signal burst until difference detected ◦ all nodes need to know their (relatively fine-grained) location to calculate their rank ◦ no support of aggregation foreseen
[148] [149] [150] [151]	◦ CONTROL: Beacon signal with sender ID	◦ RID ◦ from receivers that answer beacon signal with ACK, sender chooses optimal one	◦ RID: each receiver that considers itself as next-hop node sends ACK directly after Beacon/ID (collisions are treated in [148]) ◦ sender chooses optimal receiver based on estimated delay values of the next-hop nodes	◦ sender needs to know delay values of its neighbors (estimations of the neighbors' end-to-end delay to the sink); for this, a distributed value-iteration algorithm needs to be solved, preferably in synchronized steps; this involves significant communication and computational overhead, especially when the delay values need to be recalculated frequently due to topology changes. ◦ each node needs to maintain a list of previous-hop nodes ◦ IDs needed ◦ no RCM ◦ no support of aggregation foreseen
[223]	◦ CONTROL: Small control packet with tier ID	◦ RID mechanism ◦ neighbor that sends first ACK triggers sender to send data packet	◦ RID: tier ID	◦ relatively fine-granular location information needed, since the tier width should be less than half of the transmission range ◦ contention and ACK collision between several receivers not discussed ◦ data packet transmission error not discussed ◦ no RCM ◦ no support of aggregation foreseen
[260]	◦ DATA; contains receiver priority list and own cost to reach the sink	◦ RID and RCM ◦ neighbor that sends first ACK forwards the DATA packet further	◦ RID: a priority list of selected receivers is maintained by the sender (based on corresponding links' delivery probability) and communicated to all receivers together with the DATA packet; ◦ RCM: active receivers answer in the order of this list	◦ approach similar to [42], but additionally considers energy efficiency ◦ each sender needs information on the transmission costs (here expected transmission count metric) to each of its neighbors to select forwarding set ◦ IDs needed ◦ each receiver needs to acknowledge DATA packet ◦ no aggregation considered

Table B.3: Related work on protocols based on anycast, opportunistic routing, and/or receiver contention (cont'd)

Ref.	Initial Sender Packet	Next-Hop Selection	Decision Based on	Details
[139]	◦ CONTROL: BRTS (format/-content not discussed)	◦ RCM ◦ first neighbor that sends BCTS is selected	◦ RCM: actual calculation of backoff delay not discussed; mentions the following example parameters: location (hop count or geographic coordinate), residual energy, aggregation criteria	◦ node IDs/addresses or fine-grained localization needed (to explicitly address the selected receiver) ◦ nodes need to know their location or hop count for routing the packets to the sink ◦ aggregation foreseen, but no details given; in particular, the required application layer data provided in the BRTS packets is not discussed
[347]	◦ CONTROL: Preamble and RTS	◦ RID and RCM ◦ first neighbor that sends CTS is selected	◦ RID: location ◦ RCM: greedy forwarding metric	◦ nodes need to know their location (fine-grained) ◦ Nodes need to calculate their priority (using the forwarding metric that considers the achievable hop distance and packet reception probabilities from a channel model) ◦ IDs needed ◦ no aggregation considered
[329] [328]	◦ CONTROL: short beacon messages (format and content not discussed)	◦ RID ◦ first neighbor that sends CTS is selected	◦ RID: location, SNR	◦ neighbors need to know location of sender; i.e., either the sender's location is included in the beacon message or IDs are provided and location tables are maintained by other means ◦ node IDs/addresses or fine-grained localization also needed (to explicitly address the selected receiver) ◦ no RCM; collisions of CTS considered negligible ◦ no aggregation considered
[90]	◦ CONTROL: RTS; contains location	◦ RID and RCM ◦ first neighbor that sends CTS is selected	◦ RID: SNR, remaining energy, input traffic rate, buffer occupancy, channel utilization, ◦ RCM (normal mode): closer to sink, remaining energy, distance vs. hop count tradeoff ◦ RCM (angle-based routing mode): uses angle instead of distance	◦ approach similar to [320] ◦ nodes are assumed to be location-aware (fine-grained) ◦ node IDs/addresses or fine-grained localization also needed (to explicitly address the selected receiver) ◦ no aggregation considered

B.2 Single-Hop Model: Reducible CTMC of Tagged Customer

Figures B.1 to B.5 and Fig. B.6 to B.10 illustrate the three-dimensional, reducible, structured CTMC discussed in Section 5.4.1 for the cases $N_\mu \leq N_\mathrm{c}$ and $N_\mu \geq N_\mathrm{c}$, respectively. More precisely, for each case, five two-dimensional cuts through the three-dimensional CTMC are shown. Each of the first four cuts shows the levels and phases of one layer. The fifth cut shows the layers and levels of one phase. Consequently, the first four cuts are parallel, and orthogonal to the fifth.

Note that, for the sake of clarity, only the transient states are shown as oval nodes. The three values given in each node from top to bottom are the state variables $n_{\mu\mathrm{f}}$ (layer), $n_{\mu\mathrm{b}}$ (level), and $n_{\nu\mathrm{b}}$ (phase). The arcs and their labels correspond to the transitions and their rates, respectively.

Ordinary states are depicted by nodes with gray background and single-line border. White nodes refer to states where the system is full, i.e., $n_{\mu\mathrm{b}} + n_{\nu\mathrm{b}} = N_\mathrm{c}$. No arrivals occur in these states. Nodes with double-line border refer to states where there is no idle server, i.e., $n_{\mu\mathrm{b}} + n_{\mu\mathrm{f}} = N_\mu$. In these states, arrivals (if any) go to the orbit, neither server failures nor retrials are possible, and, in particular, there is no transition to the absorbing state.

The absorbing state and the transitions to it are not graphically represented to preserve clarity. From all transient states that are depicted with single-line border, the absorbing state can be reached with rate ν.

At the top left of Fig. B.6, state $(N_\mathrm{c}-1, 1, n_{\mu\mathrm{f}})$ is depicted with a dashed double border. It should be interpreted as a white single-line node (full system) for Layer $n_{\mu\mathrm{f}} < (N_\mu - N_\mathrm{c} + 1)$ and as a white double-line node (full system with all servers busy of failed) for Layer $n_{\mu\mathrm{f}} = (N_\mu - N_\mathrm{c} + 1)$.

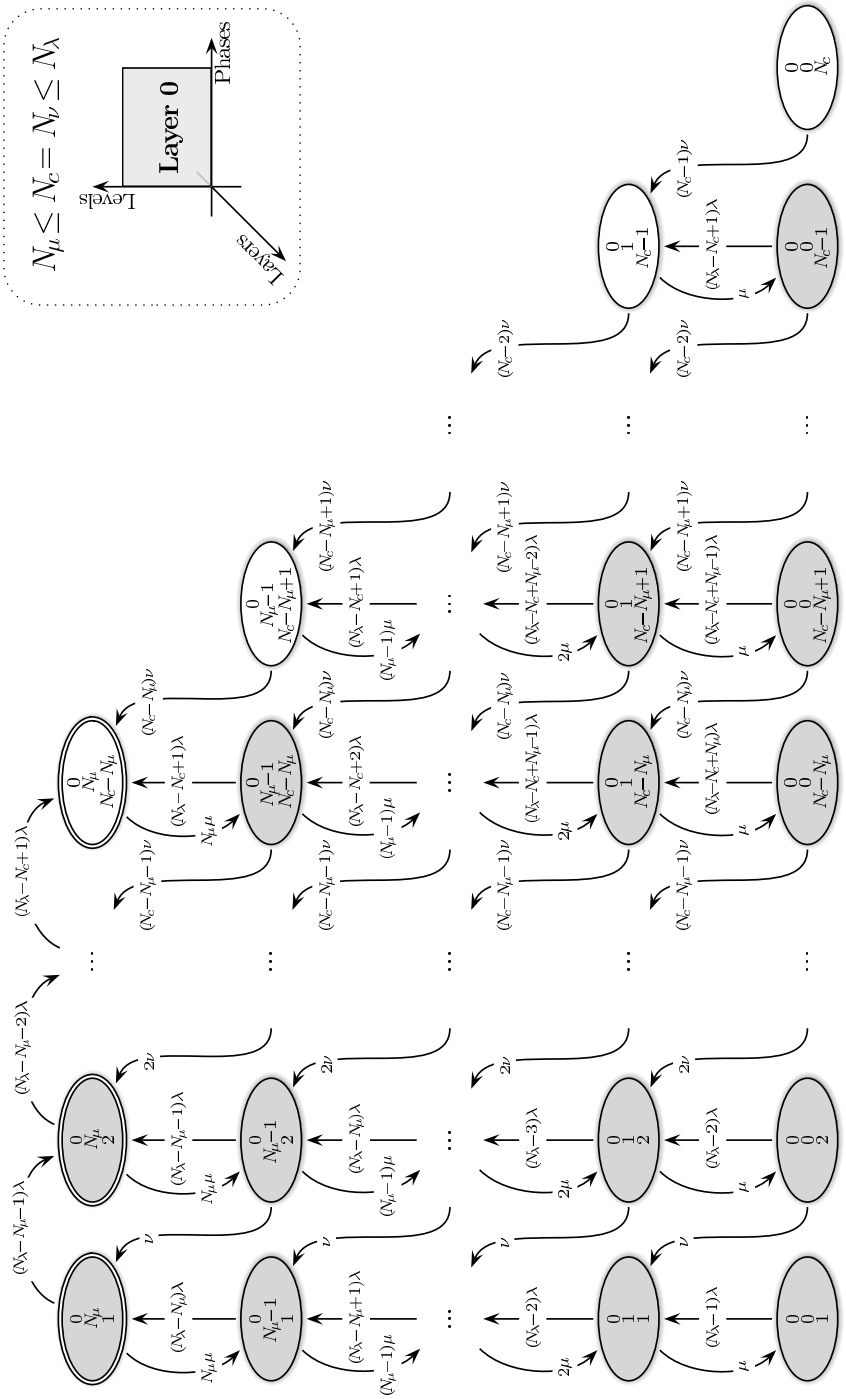


Figure B.1: Diagram of transient states in Layer $n_{\mu\mathrm{f}} = 0$ of CTMC $\widetilde{X}(t)$ for $N_\mu \leq N_c$.

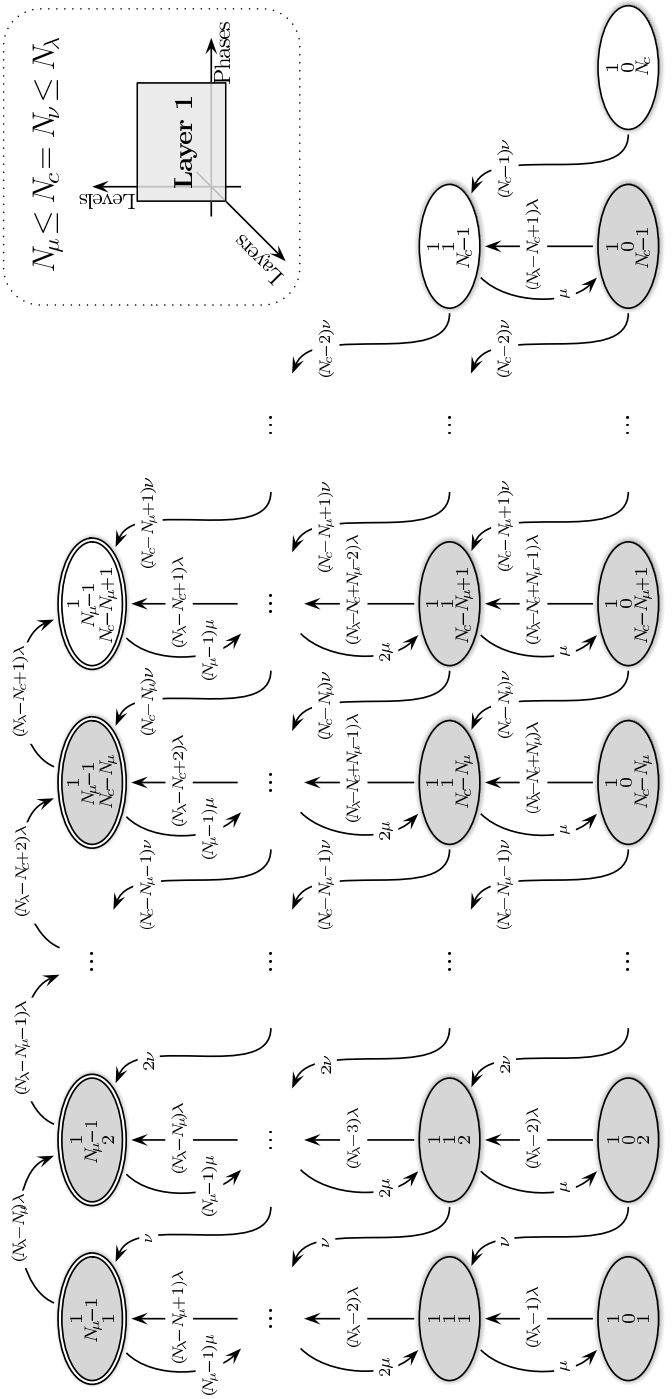


Figure B.2: Diagram of transient states in Layer $n_{\mu f} = 1$ of CTMC $\widetilde{X}(t)$ for $N_\mu \leq N_c$.

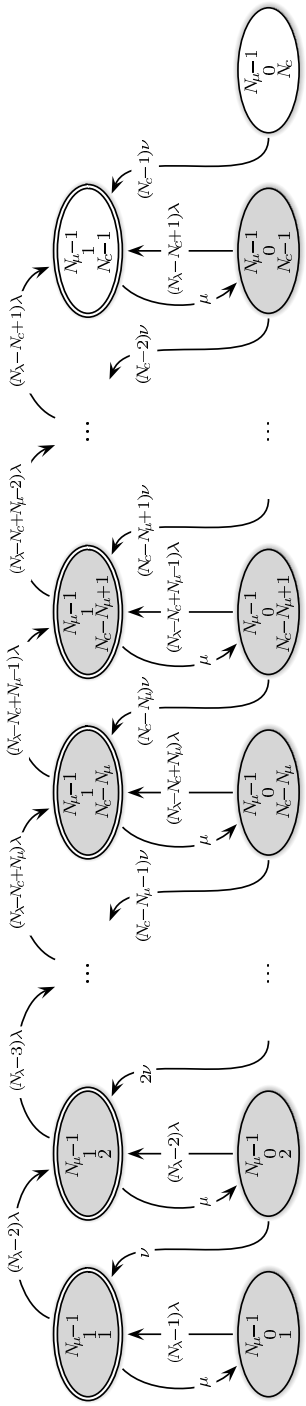


Figure B.3: Diagram of transient states in Layer $n_{\mu\mathrm{f}} = N_\mu - 1$ of CTMC $\widetilde{X}(t)$ for $N_\mu \leq N_\mathrm{c}$.

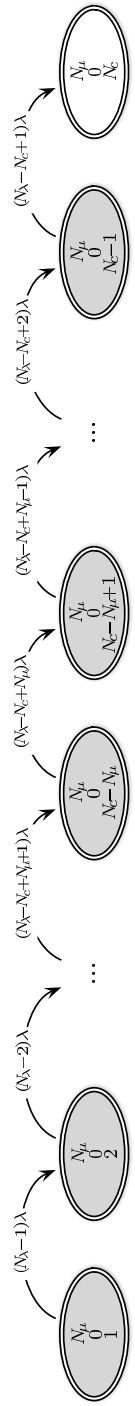


Figure B.4: Diagram of transient states in Layer $n_{\mu\mathrm{f}} = N_\mu$ of CTMC $\widetilde{X}(t)$ for $N_\mu \leq N_\mathrm{c}$.

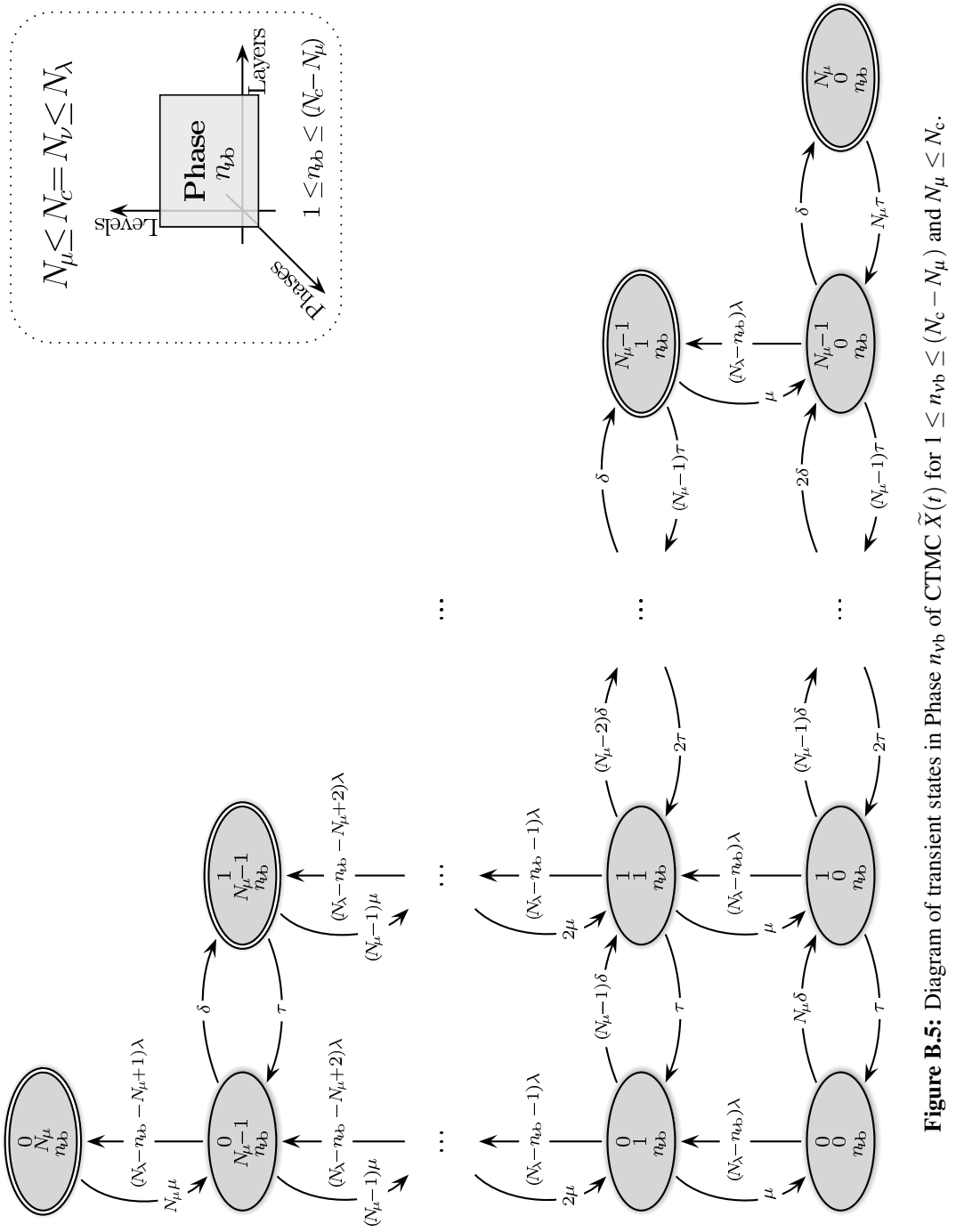


Figure B.5: Diagram of transient states in Phase $n_{\nu\mathrm{b}}$ of CTMC $\widetilde{X}(t)$ for $1 \leq n_{\nu\mathrm{b}} \leq (N_\mathrm{c} - N_\mu)$ and $N_\mu \leq N_\mathrm{c}$.

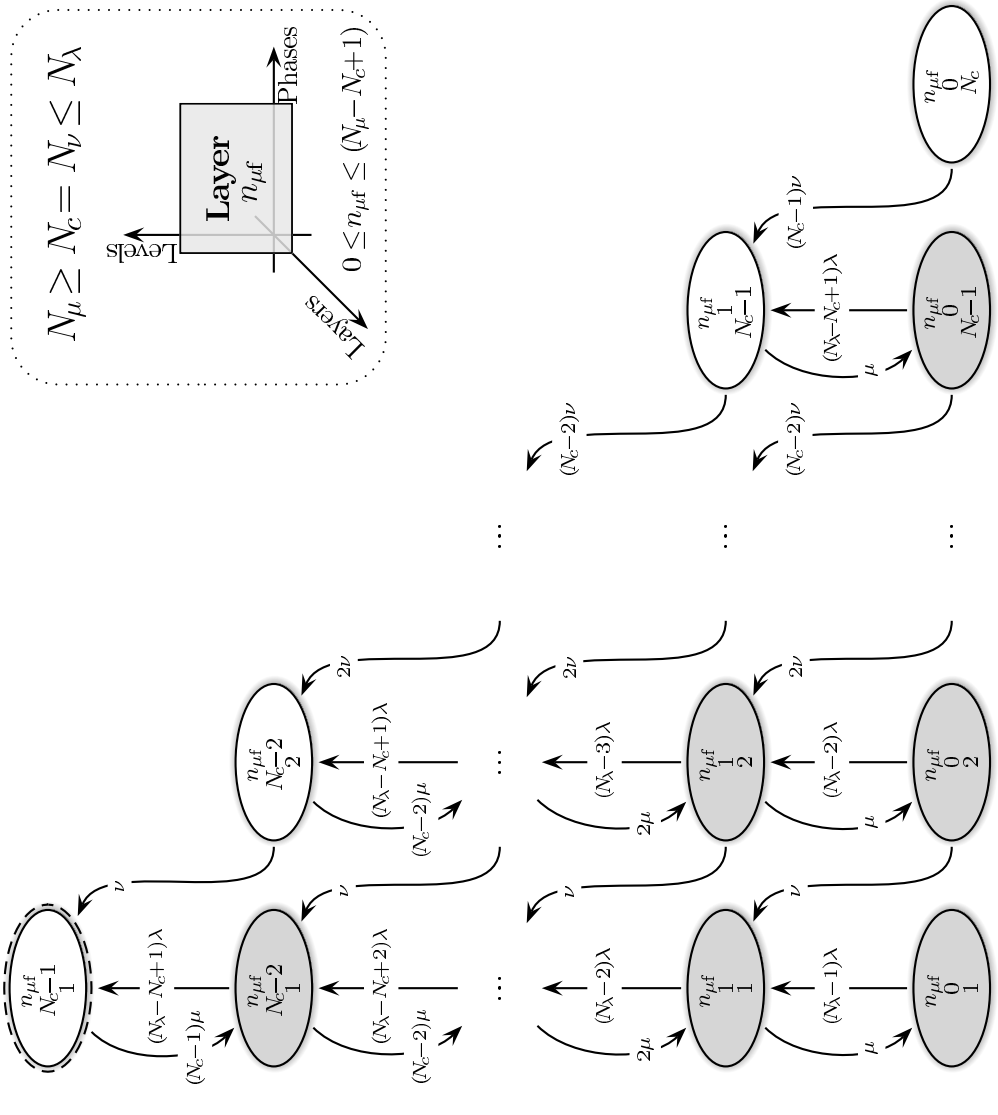


Figure B.6: Diagram of transient states in Layer $n_{\mu f}$ of CTMC $\widetilde{X}(t)$ for $0 \leq n_{\mu f} \leq (N_\mu - N_c + 1)$ and $N_\mu \geq N_c$.

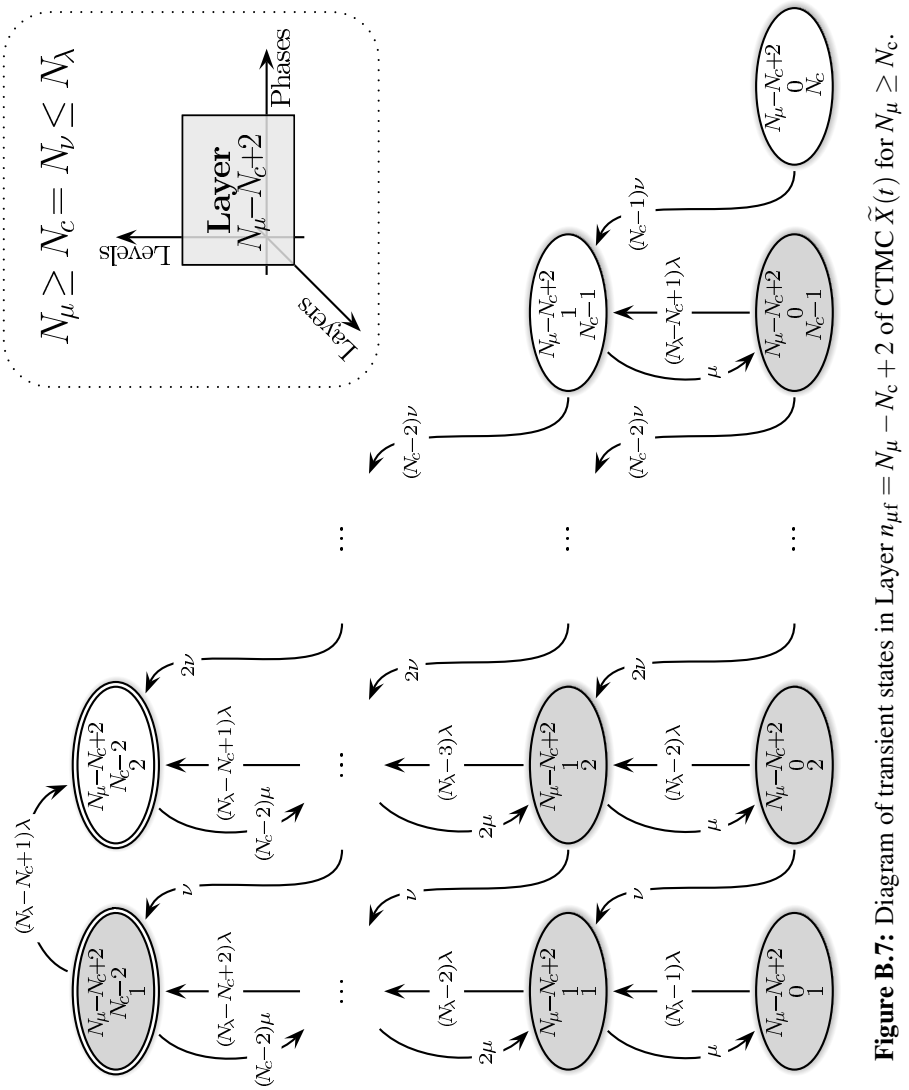


Figure B.7: Diagram of transient states in Layer $n_{\mu f} = N_\mu - N_c + 2$ of CTMC $\widetilde{X}(t)$ for $N_\mu \geq N_c$.

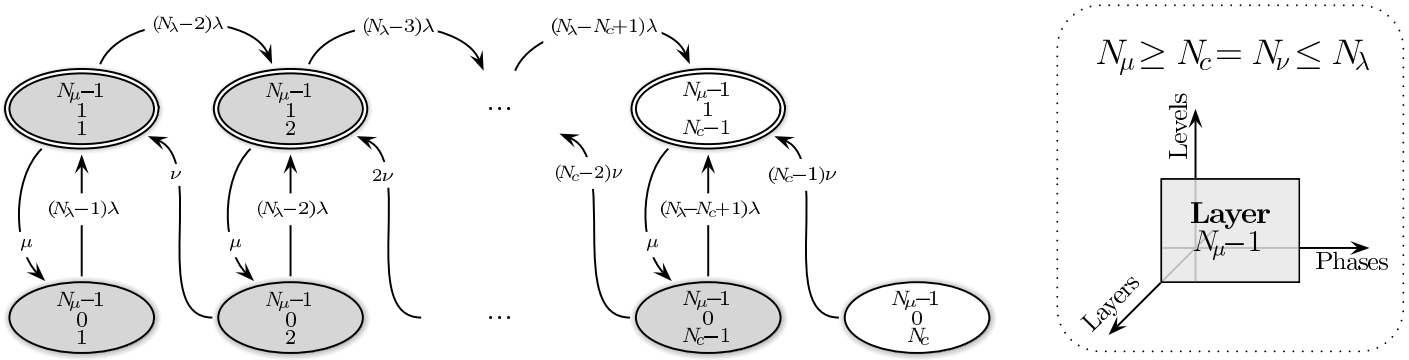


Figure B.8: Diagram of transient states in Layer $n_{\mu\mathrm{f}} = N_\mu - 1$ of CTMC $\widetilde{X}(t)$ for $N_\mu \geq N_\mathrm{c}$.

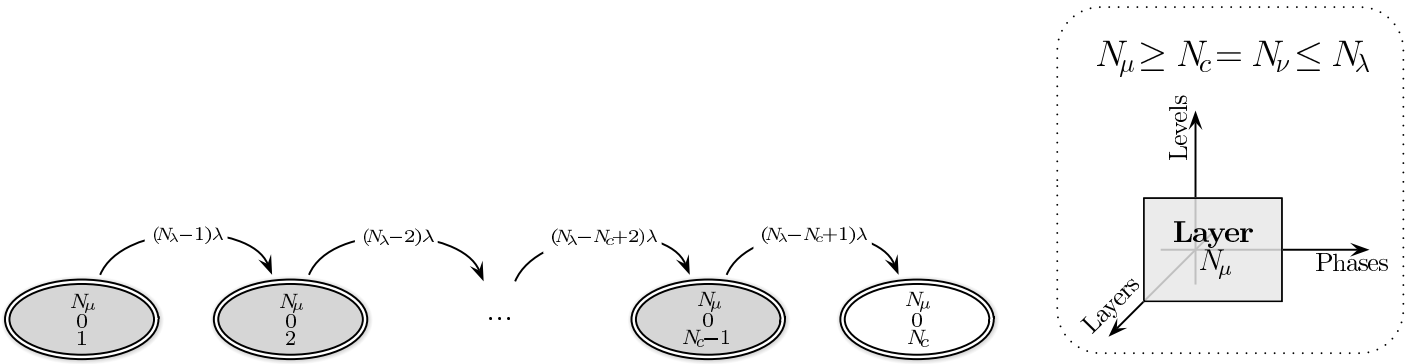


Figure B.9: Diagram of transient states in Layer $n_{\mu\mathrm{f}} = N_\mu$ of CTMC $\widetilde{X}(t)$ for $N_\mu \geq N_\mathrm{c}$.

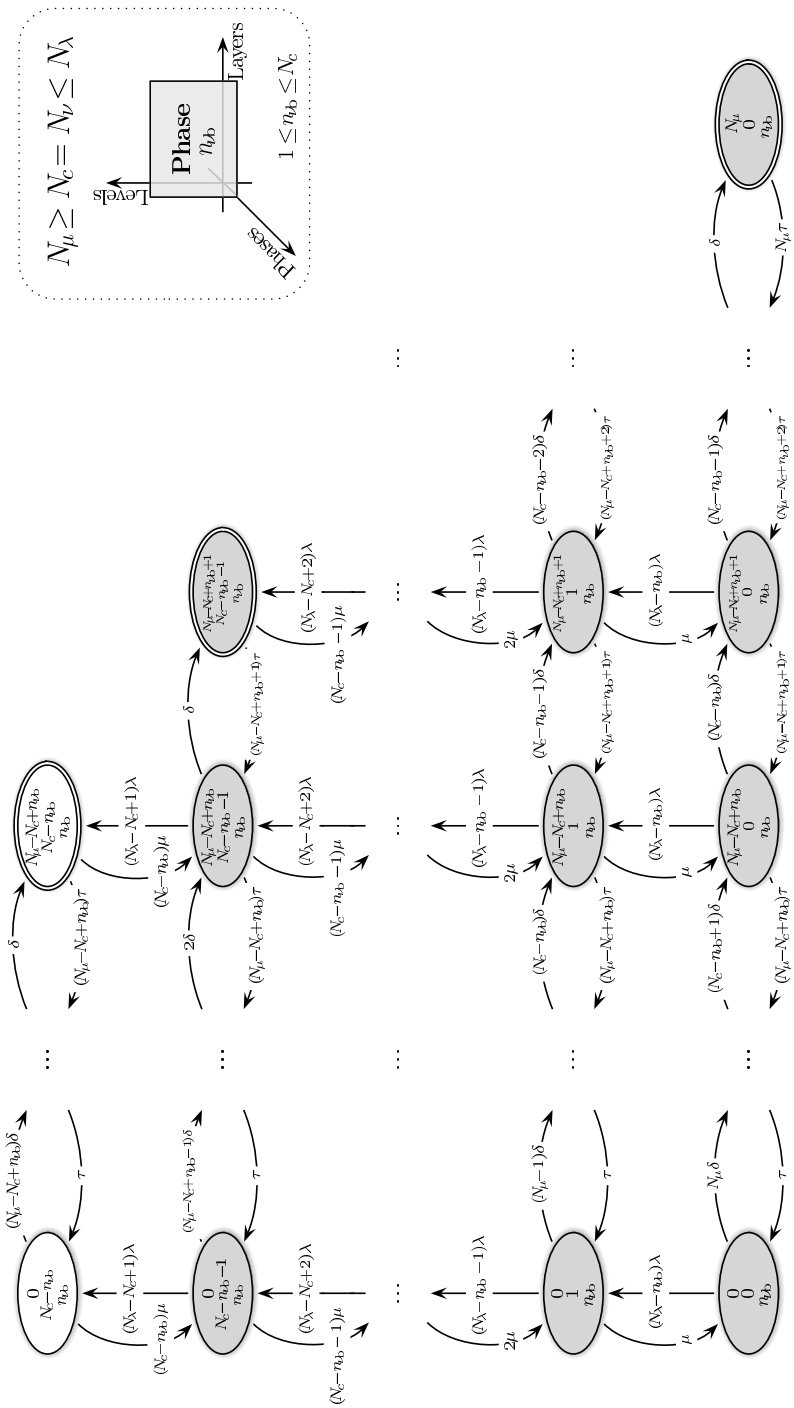


Figure B.10: Diagram of transient states in Phase $n_{\nu b}$ of CTMC $\widetilde{X}(t)$ for $1 \leq n_{\nu b} \leq (N_c - N_\mu)$ and $N_\mu \geq N_c$.

B.3 Estimation of Volumetric Storage Density

Here, a rough estimation is provided of the volumetric storage density (i.e., number of bits that can be stored per unit volume) of current state-of-the-art (as of mid 2012*) storage technologies that could be applied in tiny (see Section 4.2.5) wireless sensor nodes.

B.3.1 Integrated Circuit Thickness

According to [132, Ch. Assembly&Packaging, p. 12], a total circuit thickness of less than 100 µm can already be achieved in high volume manufacturing of integrated circuits. A further reduction of thickness is foreseen (see [132, Ch. Assembly&Packaging, p. 12, 53]). For keeping production costs low, a circuit thickness of approximately 100 µm is assumed in the following.

B.3.2 Volumetric Storage Density of SRAM

In September 2009, [131] announced the successful implementation of static random-access memory (SRAM) cells of a minimum area of approximately $0.1\,\mu m^2$. That is, 10 Mbit can be stored in an area of $1\,mm^2$. Assuming a thickness of 0.1 mm following Section B.3.1, approximately 10 Mbit can be stored in a volume of $1\,mm^3$.

B.3.3 Volumetric Storage Density of Flash Memory

In February 2012, [310] announced fitting 128 Gbit on an area of $170\,mm^2$. This results in 753 Mbit in an area of $1\,mm^2$. Assuming again a thickness of 0.1 mm results in a volumetric storage density of approximately $7.5\,Gbit/mm^3$.

B.4 Coordinate Transformation between Physical and Grid Coordinates

Transforming Cartesian coordinates to coordinates of a hexagonal grid (and vice-versa) is a common problem in game programming. The following transformation is inspired by [133] and includes the following modifications:

- allow for real-valued Cartesian coordinates (physical coordinates) instead of integer pixel coordinates,
- allow for negative-valued coordinates,

* Further improvements are under way (see, e.g., [152]).

- adaption to the hexagonal coordinate system illustrated in Figure B.11 (to match Section 4.3.5.2) with pointy sides and an angle of 120° between the x_g and y_g axes,
- simplification by reducing approach to a single "section type" by shifting odd-numbered columns of grid cells.

B.4.1 Transformation of Grid Coordinates to Physical Coordinates

First, the transformation of the grid coordinate of hexagonal cell $\mathbf{d}_\mathrm{g} = \langle x_\mathrm{g}, y_\mathrm{g} \rangle$ to the physical coordinate $\mathbf{d}_\mathrm{p}^{(\mathrm{c})} = \langle x_\mathrm{p}^{(\mathrm{c})}, y_\mathrm{p}^{(\mathrm{c})} \rangle$ of the cell's center is performed. This case is simpler than the reverse case and its results can be reused later.

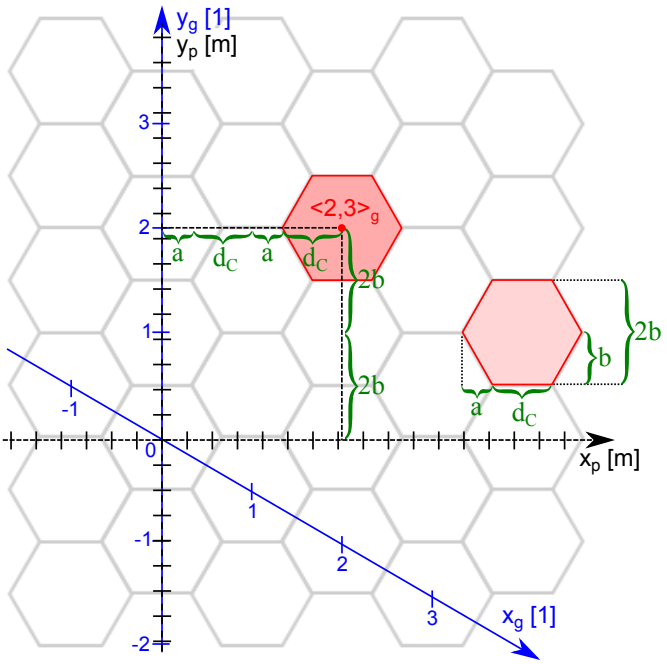


Figure B.11: Hexagonal grid with physical and grid coordinate systems.

A supporting illustration is shown in Figure B.11, where the physical radius of a cell is d_C (in meters) and two auxiliary parameters a and b (both in meters) are defined, which can be calculated via

$$a = d_\mathrm{C} \sin(30^\circ) = 0.5 \cdot d_\mathrm{C},$$

and

$$b \overset{\text{Pythagoras}}{=} \sqrt{d_\mathrm{C}^2 - a^2} = \sqrt{3}a = \sqrt{3} \cdot 0.5 \cdot d_\mathrm{C},$$

respectively.

It can be seen rather easily now from Figure B.11 that the cell's center's $x_\mathrm{p}^{(\mathrm{c})}$-coordinate is given by

$$x_\mathrm{p}^{(\mathrm{c})} = x_\mathrm{g} \, (a + d_\mathrm{C}) = x_\mathrm{g} \cdot 1.5 \, d_\mathrm{C}, \tag{B.1}$$

and its $y_p^{(c)}$-coordinate by

$$\begin{aligned} y_p^{(c)} &= y_g \cdot 2b - x_g b = b(2y_g - x_g) \\ &= \sqrt{3} \cdot 0.5 \cdot d_C(2y_g - x_g) \approx 0.866 \cdot d_C(2y_g - x_g)\,. \end{aligned} \tag{B.2}$$

B.4.2 Transformation of Physical Coordinates to Grid Coordinates

The transformation of a node's physical coordinate $\mathbf{d}_p = \langle x_p, y_p \rangle$ to its corresponding grid cell coordinate $\mathbf{d}_g = \langle x_g, y_g \rangle$ is done in three steps:

1. *Approximate hexagons by rectangles.* This step is only done for deriving the transformation. The nodes only need to apply the final equations which are exact.

2. *Decide which rectangle the physical coordinate is located in.* This leads to a preliminary grid coordinate based on rectangles.

3. *Correct preliminary grid coordinate if necessary.* This leads to the final and exact grid coordinate of the given physical coordinate.

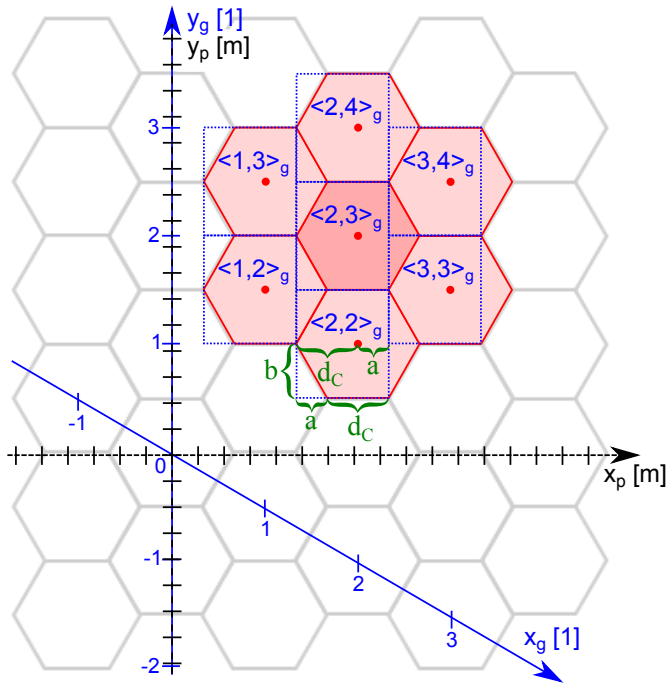


Figure B.12: Approximation of hexagonal grid by rectangles.

Step 1: Approximate hexagons by rectangles. In Figure B.12, the hexagonal grid is approximated by rectangles with physical width

$$w = a + d_C = 1.5 \cdot d_C\,,$$

and physical height

$$h = 2b = \sqrt{3} \cdot d_{\mathrm{C}} \,.$$

In the following, the rectangles are addressed by using the grid coordinate of the corresponding hexagonal grid cell, i.e., of the cell whose center is located within the rectangle.

Step 2: Decide which rectangle the physical coordinate is located in. It can be seen in Figure B.12 that the rectangles horizontally spread from $\left(x_{\mathrm{p}}^{(\mathrm{c})} - d_{\mathrm{C}}\right)$ to $\left(x_{\mathrm{p}}^{(\mathrm{c})} + a\right)$. Equivalently, the rectangles spread vertically from $\left(y_{\mathrm{p}}^{(\mathrm{c})} - b\right)$ to $\left(y_{\mathrm{p}}^{(\mathrm{c})} + b\right)$.

Based on these considerations, physical coordinate's preliminary grid coordinates can be given by

$$x_{\mathrm{g}}^{(\mathrm{pre})} = \left\lfloor \frac{x_{\mathrm{p}} + d_{\mathrm{C}}}{w} \right\rfloor ,$$

and

$$y_{\mathrm{g}}^{(\mathrm{pre})} = \left\lfloor \frac{y_{\mathrm{p}} + \left(x_{\mathrm{g}}^{(\mathrm{pre})} + 1\right) b}{h} \right\rfloor = \left\lfloor \frac{y_{\mathrm{p}}}{h} + \frac{x_{\mathrm{g}}^{(\mathrm{pre})} + 1}{2} \right\rfloor .$$

Step 3: Correct preliminary grid coordinate if necessary. For the final correction of the preliminary grid coordinate, the physical coordinate's position within the rectangle needs to be considered. For convenience, the investigation is reduced to the hexagon at the origin by normalizing the physical coordinate as follows:

$$x_{\mathrm{p}}^{(\mathrm{norm})} = x_{\mathrm{p}} - x_{\mathrm{p}}^{(\mathrm{c})} ,$$

and

$$y_{\mathrm{p}}^{(\mathrm{norm})} = y_{\mathrm{p}} - y_{\mathrm{p}}^{(\mathrm{c})} ,$$

where $x_{\mathrm{p}}^{(\mathrm{c})}$ and $y_{\mathrm{p}}^{(\mathrm{c})}$ are the physical coordinates corresponding to the center of the of preliminary grid cell $\langle x_{\mathrm{g}}^{(\mathrm{pre})}, y_{\mathrm{g}}^{(\mathrm{pre})} \rangle$ and can be derived with Eqs. (B.1) and (B.2), respectively.

Consequently, it is sufficient to investigate the scenario shown in Figure B.13. After normalization to $\langle x_{\mathrm{p}}^{(\mathrm{norm})}, y_{\mathrm{p}}^{(\mathrm{norm})} \rangle$, the physical coordinate $\mathbf{d}_{\mathrm{p}} = \langle x_{\mathrm{p}}, y_{\mathrm{p}} \rangle$ lies somewhere within the rectangle $\langle 0,0 \rangle_{\mathrm{g}}^{(\mathrm{pre})}$. For all normalized physical coordinates that do not fall into one of the two green triangles at the left-hand side of the rectangle, $\langle x_{\mathrm{g}}^{(\mathrm{pre})}, y_{\mathrm{g}}^{(\mathrm{pre})} \rangle$ is already the correct and final grid coordinate Otherwise, the preliminary grid coordinate has to be adapted as described in the following.

The slope of the border between the upper green triangle and the hexagonal cell is given by

$$m^{+} = \frac{b}{a} = \frac{\sqrt{3} \cdot 0.5 \cdot d_{\mathrm{C}}}{0.5 \cdot d_{\mathrm{C}}} = \sqrt{3} .$$

Similarly, the border slope of the lower green triangle is

$$m^{-} = \frac{-b}{a} = -\sqrt{3}$$

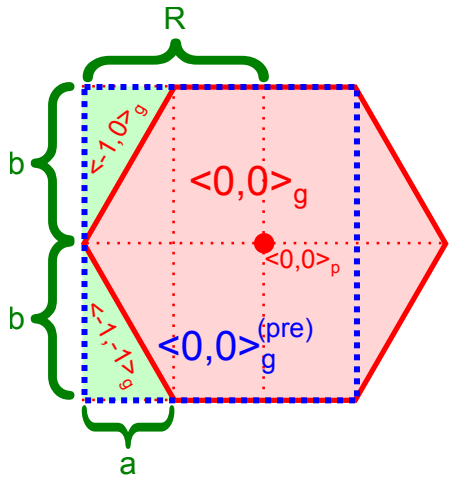


Figure B.13: Normalized grid cell for correction of preliminary grid coordinates.

Hence, if $y_p^{(norm)} > \sqrt{3}\left(x_p^{(norm)} + d_C\right)$, the physical coordinate lies within the upper green triangle and its final grid coordinate is given by $\mathbf{d}_g = \langle x_g^{(pre)} - 1, y_g^{(pre)} \rangle$. If, on the other hand, $y_p^{(norm)} < -\sqrt{3}\left(x_p^{(norm)} + d_C\right)$, the physical coordinate lies within the lower green triangle and its final grid coordinate is given by $\mathbf{d}_g = \langle x_g^{(pre)} - 1, y_g^{(pre)} - 1 \rangle$. Remember that the final grid coordinate is identical to the preliminary one, i.e., $\mathbf{d}_g = \langle x_g^{(pre)}, y_g^{(pre)} \rangle$, if the physical coordinate lies outside any triangle.

Summary. Wrapping up the equations of Steps 1 to 3 leads to the compact solution presented in Eqs. (4.6) and (4.7). These can be used by sensor nodes to determine their grid coordinate $\mathbf{d}_g = \langle x_g, y_g \rangle$ based on their physical coordinate $\mathbf{d}_g = \langle x_p, y_p \rangle$.

B.5 Impact of Group Failure

This section investigates the difference between two variants of finite-source retrial queues with multiple homogeneous and unreliable servers. In the first variant, the servers fail independently of each other. This is the variant which is investigated more frequently in the literature. In the second variant, the servers are failing in groups. The first variant can be seen as a special case of the second variant when there is only a single server per group. An incoming or retrying job randomly selects one of the available servers. If no server is available, it (re-)joins the orbit. To reflect the behavior of the protocol proposed in Chapter 4, a group may only fail if all servers of the group are idle.

In Figs. B.14 and B.15, the mean response time (y-axis) is shown for varying number of servers per group (spg) and generation rate λ (x-axis) below and above 1, respectively. The results are obtained with MOSEL-2 using the parameters summarized in Table B.4. The case spg $= 1$ coincides with the model where each server fails individually. The results show that the number of servers per group have a notable influence on the mean response time. Additionally, this influence is load dependent. Consequently, there is little hope that systems that exhibit group failure can be approximated well by models that treat the server failures individually.

Table B.4: Parameters used for discussing influence of group failure

Parameter	Value
generation rate	$\lambda = 0.01, \ldots, 10$ (x-axis)
number of servers per group	spg $\in \{1, 2, 5, 10\}$ (curves)
number of sources	$K = 5$
number of servers	$c = 10$
service rate	$\mu = 1$
retrial rate	$\nu = 5$
failure rate	$\delta = 5$
repair rate	$\tau = 1$

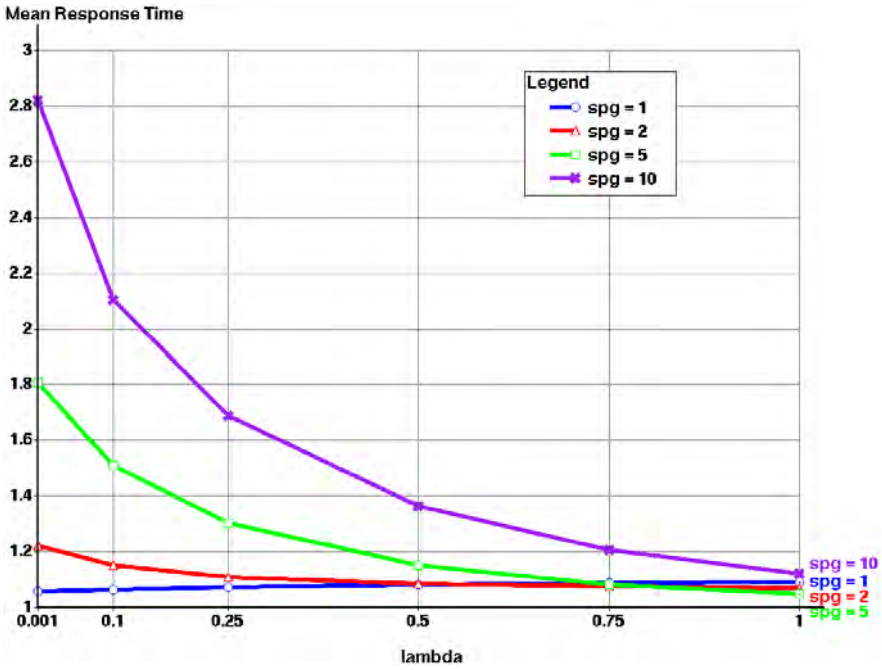


Figure B.14: Mean response time of finite source retrial queue with group failure ($\lambda \leq 1$).

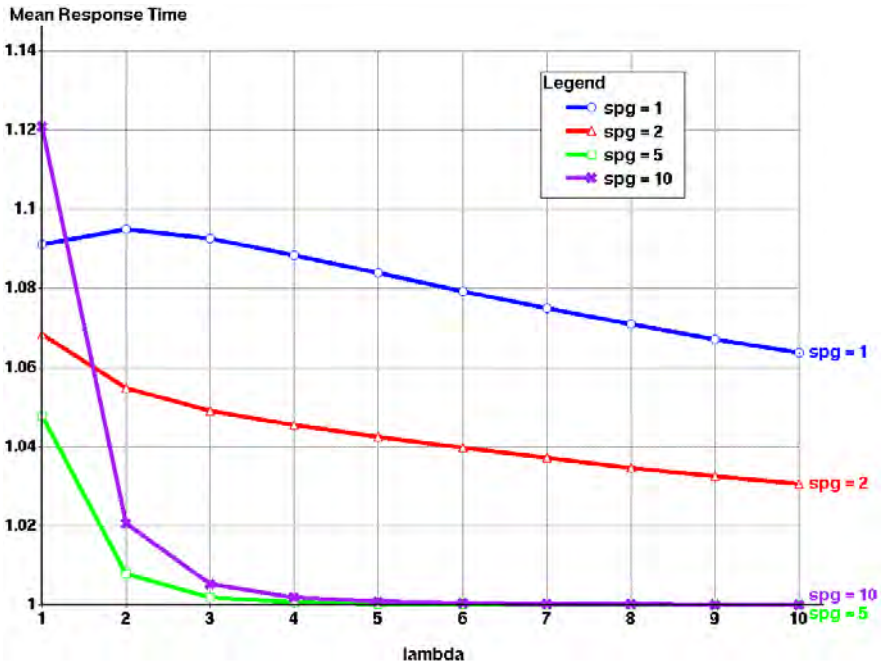


Figure B.15: Mean response time of finite source retrial queue with group failure ($\lambda \geq 1$).

B.6 CPN Tools' Precision of Time Values

This thesis focuses on exponentially distributed random time variables. While CPN Tools allows to sample random variables with floating-point precision, the obtained values need to be rounded to integers* for being able to use them as time values. This section discusses the consequence of this rounding and proposes a suitable workaround.

Let $T(\lambda) \sim \text{Exp}(\lambda)$ be an exponentially distributed random variable with mean $\overline{T}(\lambda) = \lambda^{-1}$, where λ is called the distribution's rate parameter. The expected value of the rounded random variable is denoted as $\overline{T}_r(\lambda) = E(\text{round}\, T)$ in the following. The derivation of $\overline{T}_r(\lambda)$ is provided in Eq. (B.5) given in Appendix Section B.8. Variable $\overline{D}(\lambda) = \overline{T}_r(\lambda) - \overline{T}(\lambda)$ describes the mean deviation of the two expected value. Similarly, the relative deviation can be given as $\overline{D_r}(\lambda) = \frac{\overline{T}_r(\lambda) - \overline{T}(\lambda)}{\overline{T}(\lambda)}$.

Both deviations are plotted against rate λ in Fig. B.16. The figure illustrates that for keeping the relative error $|D_r(\lambda)|$ smaller than 10%, rate λ needs to be chosen smaller than approximately 1.6.

Hence, for dealing with rates up to at least 2500 (see, e.g., δ in *RealSet* of Tab. 5.2), a suitable workaround to this issue is needed before being able to apply CPN Tools.

The proposed workaround is to rescale all rate parameters used in simulation and rescale the timed results (e.g., sojourn and response times) accordingly. The untimed results (like mean number of jobs in a place) are not affected by the rescaling. The rescaling can be seen as a modification of the time unit during simulation. The rescaling uses a parameter that is denoted as *precision factor*. All sampled times are multiplied with this precision factor before

* While during the preparation of this thesis, the support of real-valued time parameters has been (partly) implemented into CPN Tools, a first trial to adapt the model accordingly unfortunately failed. Therefore, the decision was taken to postpone this effort to future work.

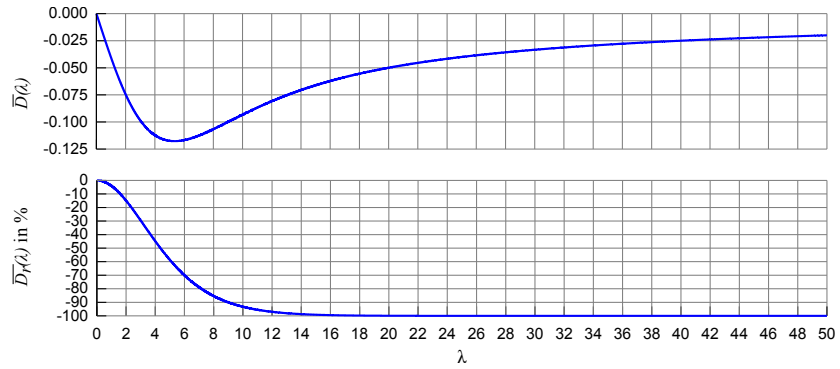


Figure B.16: Deviation introduced by rounding an exponentially distributed random variable with rate λ.

getting rounded. To compensate for this modification, all timed result values are divided by the precision factor.

Unfortunately, the precision factor cannot be chosen arbitrarily high. This is because (positive) time values can take a maximum value of approximately $1.0700 \cdot 10^9$ time units (corresponding to 30 bit) in CPN Tools. This thesis focuses on steady-state evaluation of the system behavior. Therefore, the simulation length is always chosen close to the maximum simulation length of $1.0700 \cdot 10^9$ time units.

Multiplying the sampled times with the precision factor is equivalent to dividing all rate parameters by the precision factor. Consequently, the number of events (called *simulation steps* in CPN Tools) that are generated during the simulation is reduced approximately by the same factor. That is, the higher the precision factor is chosen the less events are generated and the wider the CIs get.

These considerations are supported by the simulation results presented in Fig. B.17 for various precision factors. The results are obtained using the model described in Section 5.2 and the *RealSet* of Tab. 5.2. Each given mean value and its CI correspond to ten simulation runs. For comparison, results obtained by numerical analysis (approaches described in Sections 5.3 and 5.4) are also provided and obviously independent of the simulations precision factor. Exemplarily, also the overall number of generation and retrial events (i.e., tokens that pass through place P_2 of Fig. 5.8) is also provided. As expected, it decreases with increasing precision factor.

The figure shows that for higher precision factors (and reduced rounding errors), the simulation results get closer to the results obtained by numerical analysis. On the other hand, for higher precision factors, the CIs get wider.

Hence, to improve the simulation results, the rates should be kept low by applying a high precision factor. Moreover, since the maximum length of a single simulation run is bounded, the number of runs needs to be increased when increasing the precision factor to keep the CIs narrow. This not necessarily increases the time needed for simulation, since the number of events per run is smaller. However, this approach increases the effect of the initial state on the results which is in contrast to our goal to carry out steady-state evaluation.

The influence of the simulation length (in number of steps) on the results is exemplarily shown for $\overline{T_V}$ in Fig. B.18. It can be seen that the simulation length may be held responsible only to a small degree for the discrepancies of the mean result values shown in Fig. B.17.

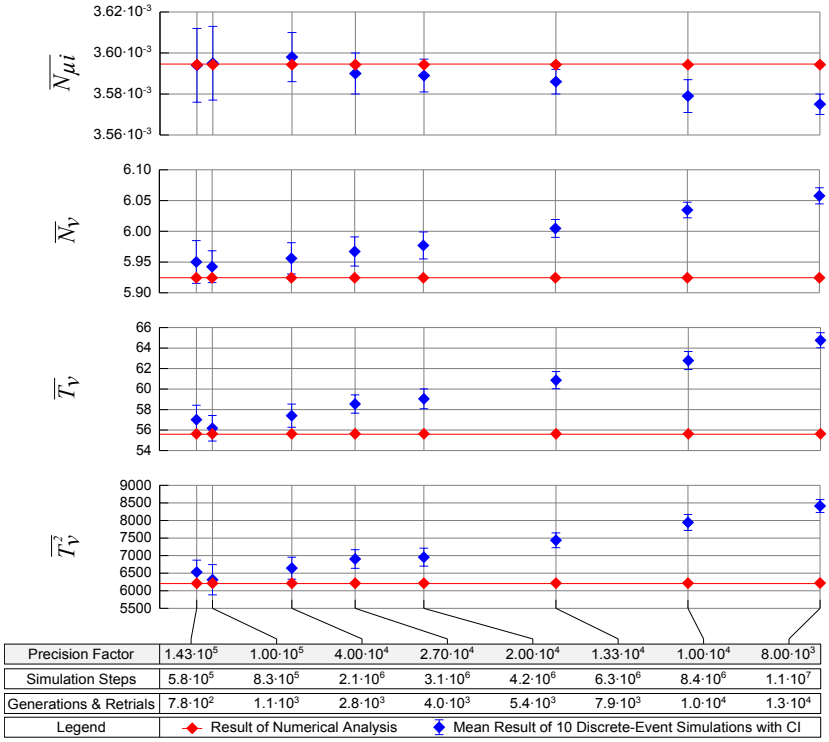


Figure B.17: Influence of time unit factor on simulation result.

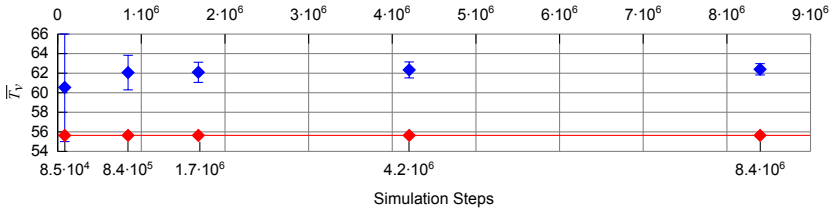


Figure B.18: Influence of simulation length on simulation result of $\overline{T_V}$.

Hence, the main influence can indeed be attributed to the rounding issue which is sensible to the chosen magnitude of the model's rate parameters. However, it can again be seen in Fig. B.18 that reducing the number of simulation steps widens the CIs.

B.7 CPN Tools Model of Finite-Source Retrial Queue with Unreliable Servers

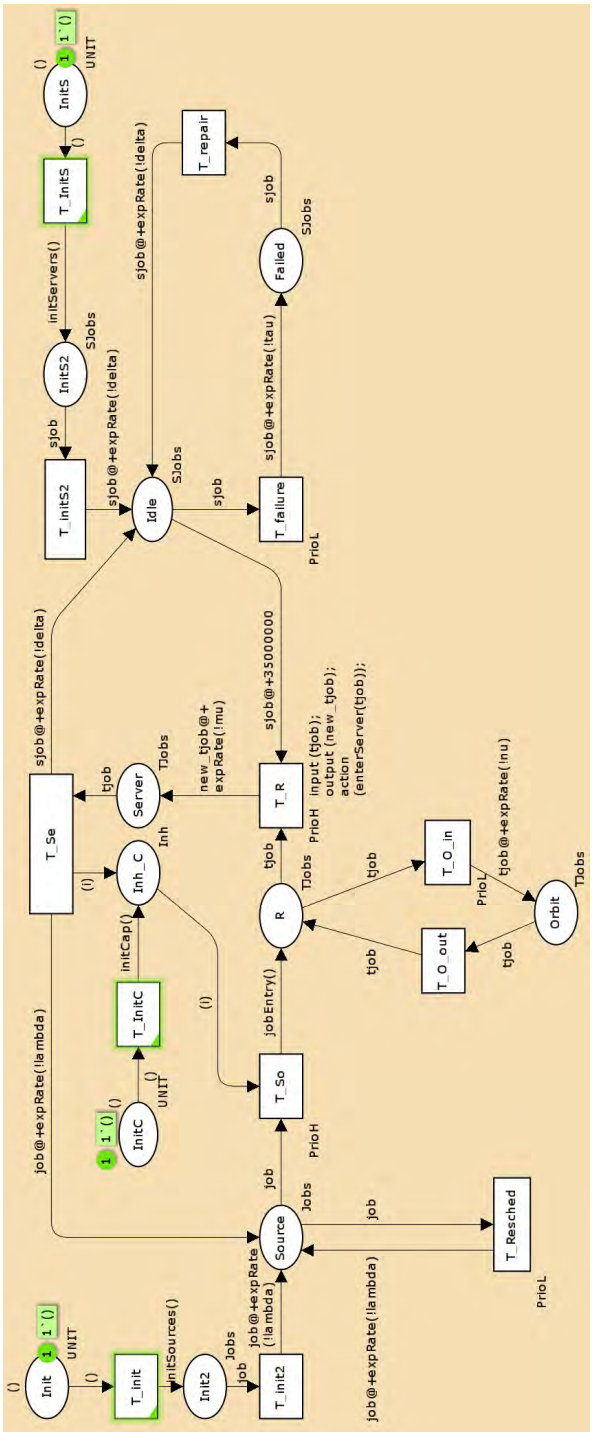


Figure B.19: Finite-source retrial queue with unreliable servers modeled in CPN Tools.

B.8 The Expected Value of a Rounded Exponentially Distributed Random Variable

In this section, the expected value $E(\text{round}(T))$ of the exponentially distributed random variable T $\text{Exp}(\lambda)$ (with $T \geq 0$, $\text{round}(T) \geq 0$, and $\overline{T} = \frac{1}{\lambda}$) is derived. The random variable $\text{round}(T)$ has a discrete distribution. Hence,

$$\begin{aligned} E(\text{round}(T)) &= \sum_{n\in\mathbb{N}_0} nP(\text{round}(T)=n) \\ &\overset{\text{Fig. B.20}}{=} 0\cdot P_0(\lambda)+1\cdot(P_1(\lambda)+P_2(\lambda))+2\cdot(P_3(\lambda)+P_4(\lambda))+\ldots \\ &= \sum_{n\in\mathbb{N}} n\left(P_{2n-1}(\lambda)+P_{2n}(\lambda)\right), \end{aligned} \tag{B.3}$$

where $P_k(\lambda)$ ($k \in \mathbb{N}_0$) is the probability that the sampled time falls into the k-th interval and given by

$$\begin{aligned} P_k(\lambda) = P\left(\frac{1}{2}k \leq t \leq \frac{1}{2}(k+1)\right) &= \int_{\frac{1}{2}k}^{\frac{1}{2}(k+1)} \lambda e^{-\lambda t}\,\mathrm{d}t \\ &= \lambda\left[-\frac{1}{\lambda}e^{-\lambda t}\right]_{\frac{1}{2}k}^{\frac{1}{2}(k+1)} = -e^{-\frac{1}{2}\lambda(k+1)}+e^{-\frac{1}{2}\lambda k} \\ &= e^{-\frac{1}{2}\lambda k}-e^{-\frac{1}{2}\lambda k-\frac{1}{2}\lambda} = e^{-\frac{1}{2}\lambda k}\left(1-e^{-\frac{1}{2}\lambda}\right). \end{aligned} \tag{B.4}$$

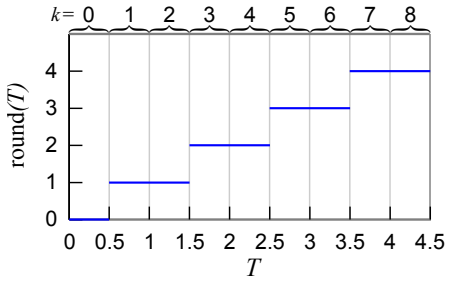


Figure B.20: Illustration of rounding function and definition of intervals $k \in \mathbb{N}_0$.

Consequently, Eq. (B.3) can be continued as

$$\begin{aligned} E(\text{round}(T)) &= \sum_{n\in\mathbb{N}} n\left(P_{2n-1}(\lambda)+P_{2n}(\lambda)\right) \\ &\overset{\text{Eq. (B.4)}}{=} \left(1-e^{-\frac{1}{2}\lambda}\right)\left(\sum_{n\in\mathbb{N}} ne^{-\frac{1}{2}\lambda(2n-1)}+\sum_{n\in\mathbb{N}} ne^{-\frac{1}{2}\lambda 2n}\right) \\ &= \left(1-e^{-\frac{1}{2}\lambda}\right)\left(\sum_{n\in\mathbb{N}} ne^{-\lambda n}e^{\frac{1}{2}\lambda}+\sum_{n\in\mathbb{N}} ne^{-\lambda n}\right) \\ &= \left(1-e^{-\frac{1}{2}\lambda}\right)\left(e^{\frac{1}{2}\lambda}+1\right)\sum_{n\in\mathbb{N}} ne^{-\lambda n} \end{aligned} \tag{B.5}$$

$$= \left(e^{\frac{1}{2}\lambda} - e^{-\frac{1}{2}\lambda}\right) \sum_{n\in\mathbb{N}} n\left(e^{-\lambda}\right)^n$$
$$\overset{\text{Eq. (B.6)}}{=} \left(e^{\frac{1}{2}\lambda} - e^{-\frac{1}{2}\lambda}\right) \frac{e^{-\lambda}}{\left(1-e^{-\lambda}\right)^2}.$$

The final step of Eq. (B.5) uses the relation

$$\sum_{n\in\mathbb{N}_0} nx^n = \frac{x}{(1-x)^2}, \tag{B.6}$$

which can be derived as follows. Defining

$$S = \sum_{n=0}^{\infty} nx^n = \sum_{n=1}^{\infty} nx^n, \tag{B.7}$$

leads to

$$Sx = \sum_{n=0}^{\infty} nx^{(n+1)} = \sum_{n=1}^{\infty} (n-1)x^n,$$

and hence,

$$S - Sx = \sum_{n=1}^{\infty} nx^n - \sum_{n=1}^{\infty} (n-1)x^n = \sum_{n=1}^{\infty} x^n \tag{B.8}$$
$$= -1 + \sum_{n=0}^{\infty} x^n \overset{\text{(geom. series)}}{=} -1 + \frac{1}{1-x} = \frac{x}{1-x}.$$

From Eq. (B.8) one directly gets

$$S(1-x) = \frac{x}{1-x},$$

and hence,

$$S = \frac{x}{(1-x)^2} \overset{\text{Eq. (B.7)}}{=} \sum_{n=0}^{\infty} nx^n.$$

B.9 Distribution of Single-Hop Mean Response Time

In this section, the cumulative distribution function (CDF) $F_{T_{v\mu}}(T_{v\mu}^{(\max)})$ of the single-hop mean response time $T_{v\mu}$ is derived*. First note that

$$F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) = P\left(T_{v\mu} \le T_{v\mu}^{(\max)}\right) = P\left(T_v + T_\mu \le T_{v\mu}^{(\max)}\right) = F_{T_v+T_\mu}\left(T_{v\mu}^{(\max)}\right).$$

* The time unit in this section is *seconds*. For conserving conciseness, the time unit is not explicitly included in the derivations.

Moreover, note that a tagged job's distributions of T_ν and T_μ are mutually independent. Hence, the sum of the two random variables with CDFs F_{T_ν} and F_{T_μ} can be represented by the convolution of the two CDFs (see, e.g., [39, p. 64], [141, p. 37] or [164, p. 14]), i.e.,

$$\begin{aligned} F_{T_{\nu\mu}}\left(T_{\nu\mu}^{(\max)}\right) &= F_{T_\nu+T_\mu}\left(T_{\nu\mu}^{(\max)}\right) = F_{T_\nu} * F_{T_\mu}\left(T_{\nu\mu}^{(\max)}\right) \\ &= \int_{-\infty}^{\infty} F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) \mathrm{d}F_{T_\nu}(t) \\ &= \int_{-\infty}^{\infty} F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) \frac{\mathrm{d}F_{T_\nu}(t)}{\mathrm{d}t}\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty} F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) f_{T_\nu}(t)\,\mathrm{d}t\,. \end{aligned} \tag{B.9}$$

Remember that T_μ is exponentially distributed, i.e., its CDF can be given by (cf. [46, p. 20])

$$\begin{aligned} F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) &= \begin{cases} 1 - e^{-\frac{T_{\nu\mu}^{(\max)} - t}{T_\mu}}, & \text{if } \left(T_{\nu\mu}^{(\max)} - t\right) \geq 0\,, \\ 0\,, & \text{otherwise}\,, \end{cases} \\ &= \begin{cases} 1 - e^{-\left(T_{\nu\mu}^{(\max)} - t\right)\mu}, & \text{if } \left(T_{\nu\mu}^{(\max)} - t\right) \geq 0\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \end{aligned} \tag{B.10}$$

Since $F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) = 0$ for $t > T_{\nu\mu}^{(\max)}$ and $f_{T_\nu}(t) = 0$ for $t < 0$, the bounds of integration in Eq. (B.9) can be narrowed accordingly, i.e.,

$$F_{T_{\nu\mu}}\left(T_{\nu\mu}^{(\max)}\right) = \int_{0}^{T_{\nu\mu}^{(\max)}} F_{T_\mu}\left(T_{\nu\mu}^{(\max)} - t\right) f_{T_\nu}(t)\,\mathrm{d}t\,. \tag{B.11}$$

The PDF $f_{T_\nu}(t)$ of the waiting time T_ν can be rephrased as

$$\begin{aligned} f_{T_\nu}(t) &= \frac{\mathrm{d}F_{T_\nu}(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}P(T_\nu \leq t) = \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - P(T_\nu > t)\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - p_\nu P(T_{\nu\mathrm{o}} > t)\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - p_\nu\left(1 - F_{T_{\nu\mathrm{o}}}(t)\right)\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - p_\nu + p_\nu F_{T_{\nu\mathrm{o}}}(t)\right) = p_\nu \frac{\mathrm{d}}{\mathrm{d}t}F_{T_{\nu\mathrm{o}}}(t) = p_\nu\, f_{T_{\nu\mathrm{o}}}(t)\,. \end{aligned} \tag{B.12}$$

Since the waiting time $T_{\nu\mathrm{o}}$ of orbit-visiting customers can be approximated by a gamma distribution with parameters μ_γ and α_γ (see Section 5.5.2), Eq. (B.12) can be continued as (see, e.g., [46, p. 25])

$$f_{T_\nu}(t) = p_\nu\, f_{T_{\nu\mathrm{o}}}(t) \approx p_\nu \frac{\alpha_\gamma\mu_\gamma\left(\alpha_\gamma\mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma\mu_\gamma t}\,, \tag{B.13}$$

where $\Gamma\left(\alpha_\gamma\right)$ is the gamma function defined in Eq. (5.28).

Consequently, by inserting Eqs. (B.10) and (B.13) into Eq. (B.11), $F_{T_{v\mu}}(T_{v\mu}^{(\max)})$ is given by

$$F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) = \int_0^{T_{v\mu}^{(\max)}} \left(1 - e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu}\right) p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t} \, \mathrm{d}t \,. \tag{B.14}$$

For the discussion presented in Appendix Section B.10, additionally the PDF $f_{T_{v\mu}}(x)$ of the single-hop response time $T_{v\mu}$ is required. It is derived from the CDF $F_{T_{v\mu}}(x)$ given in the form of Eq. (B.14) via

$$\begin{aligned} f_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) &= \frac{\mathrm{d}}{\mathrm{d}T_{v\mu}^{(\max)}} F_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}T_{v\mu}^{(\max)}} \int_0^{T_{v\mu}^{(\max)}} \underbrace{\left(1 - e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu}\right) p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t}}_{\text{denoted by } f(T_{v\mu}^{(\max)}, t) \text{ in the following}} \mathrm{d}t \,. \end{aligned} \tag{B.15}$$

At this point, the derivation uses the Leibniz rule for parameter integrals (see [204, p. 432]), which states that, if functions $g(x)$ and $h(x)$ are continuously differentiable, then

$$\frac{\mathrm{d}}{\mathrm{d}x} \int_{g(x)}^{h(x)} f(x,y) \, \mathrm{d}y = \int_{g(x)}^{h(x)} \frac{\mathrm{d}f(x,y)}{\mathrm{d}x} \, \mathrm{d}y + f(x,h(x)) \frac{\mathrm{d}h(x)}{\mathrm{d}x} - f(x,g(x)) \frac{\mathrm{d}g(x)}{\mathrm{d}x} \,, \tag{B.16}$$

where, in the case of Eq. (B.15), $h(x) = x = T_{v\mu}^{(\max)}$, $\frac{\mathrm{d}h(x)}{\mathrm{d}x} = 1$, $g(x) = \frac{\mathrm{d}g(x)}{\mathrm{d}x} = 0$ (which renders $f(x,g(x))$ irrelevant), $y = t$,

$$\begin{aligned} f(x,h(x)) &= f(T_{v\mu}^{(\max)}, T_{v\mu}^{(\max)}) \\ &= \left(1 - e^{-\left(T_{v\mu}^{(\max)} - T_{v\mu}^{(\max)}\right)\mu}\right) p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma T_{v\mu}^{(\max)}\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma T_{v\mu}^{(\max)}} \\ &= 0 \,, \end{aligned}$$

and

$$\begin{aligned} \frac{\mathrm{d}f(x,y)}{\mathrm{d}x} &= \frac{\mathrm{d}f(T_{v\mu}^{(\max)}, t)}{\mathrm{d}T_{v\mu}^{(\max)}} \\ &= \frac{\mathrm{d}}{\mathrm{d}T_{v\mu}^{(\max)}} \left(\left(1 - e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu}\right) p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t} \right) \\ &= \left(p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t} \right) \cdot \frac{\mathrm{d}}{\mathrm{d}T_{v\mu}^{(\max)}} \left(-e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu} \right) \\ &= \left(p_v \frac{\alpha_\gamma \mu_\gamma \left(\alpha_\gamma \mu_\gamma t\right)^{\alpha_\gamma - 1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma \mu_\gamma t} \right) \cdot \left(\mu e^{-\left(T_{v\mu}^{(\max)} - t\right)\mu} \right) . \end{aligned} \tag{B.17}$$

Consequently, Eq. (B.15) can be continued based on Eqs. (B.16) to (B.17) via

$$f_{T_{v\mu}}\left(T_{v\mu}^{(\max)}\right) = \int_0^{T_{v\mu}^{(\max)}} \frac{\mathrm{d}f(T_{v\mu}^{(\max)},t)}{\mathrm{d}T_{v\mu}^{(\max)}}\,\mathrm{d}t \tag{B.18}$$
$$= \int_0^{T_{v\mu}^{(\max)}} \left(p_v \frac{\alpha_\gamma\mu_\gamma\left(\alpha_\gamma\mu_\gamma t\right)^{\alpha_\gamma-1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma\mu_\gamma t}\right) \cdot \left(\mu e^{-\left(T_{v\mu}^{(\max)}-t\right)\mu}\right)\mathrm{d}t$$

B.10 Discussing the Use of Convolution to Calculate the Multi-Hop Delay Distribution

As discussed in Section 5.7.2, the detection-to-notification delay T_{d2n} is a sum of $n=(h-1)$ single-hop delays, i.e., a sum of n iid. random variables. In principle, it could therefore be calculated using the n-fold convolution of $F_{T_{v\mu}}(x)$ with itself. Such a convolution usually is denoted as n-fold convolution power $F_{T_{v\mu}}^{(n)*}(x) = F_{T_{v\mu}} * \ldots * F_{T_{v\mu}}(x)$ (see, e.g., [298, p. 9] or [39, p. 66]) and can be calculated recursively via

$$F_{T_{v\mu}}^{(n)*}(x) = \int_{-\infty}^{\infty} F_{T_{v\mu}}^{(n-1)*}(x-t)\,\mathrm{d}F_{T_{v\mu}}(t) \tag{B.19}$$
$$= \int_{-\infty}^{\infty} F_{T_{v\mu}}^{(n-1)*}(x-t)\frac{\mathrm{d}F_{T_{v\mu}}(t)}{\mathrm{d}t}\,\mathrm{d}t$$
$$= \int_{-\infty}^{\infty} F_{T_{v\mu}}^{(n-1)*}(x-t)\,f_{T_{v\mu}}(t)\,\mathrm{d}t\,,$$

where the PDF $f_{T_{v\mu}}(t)$ of the single-hop response time $T_{v\mu}$ is given by*

$$f_{T_{v\mu}}(t) = \int_0^t \left(p_v \frac{\alpha_\gamma\mu_\gamma\left(\alpha_\gamma\mu_\gamma t'\right)^{\alpha_\gamma-1}}{\Gamma\left(\alpha_\gamma\right)} e^{-\alpha_\gamma\mu_\gamma t'}\right) \cdot \left(\mu e^{-(t-t')\mu}\right)\mathrm{d}t'\,. \tag{B.20}$$

Since for calculating Eq. 5.39 the n needs to be varied from 1 to $d_{\mathrm{h}}^{(\max)} = 100$, it makes sense to *iteratively* calculate $P\left(T_{\mathrm{d2n}} \le T_{\mathrm{d2n}}^{(\max)}\right)$ based on a given $T_{\mathrm{d2n}}^{(\max)} > 0$.†

A possible iterative approach comprises the following steps, where $1 \le h \le d_{\mathrm{h}}^{(\max)} = 100$ is used as an index variable counting the number of hops.

- ***Step 1:***
  For initialization, set $h=1$ and $P\left(T_{\mathrm{d2n}} \le T_{\mathrm{d2n}}^{(\max)}\right) = 0$.
- ***Step 2:***
  Get $P(d_{\mathrm{h}} = h)$ from Table 5.16

* The derivation of $f_{T_{v\mu}}(t)$ is given in Appendix Section B.9, Eq. (B.18).

† The authors of [297] reach a similar conclusion. Hence, there is little hope that using transforms achieve significantly better performance in this case.

- ***Step 3:***
  - If $h = 1$, set $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = 1\right) = 1$, since EVMs sent by 1st-Ring nodes do not experience any delay. Continue with *Step 5*.
  - Otherwise, if $h = 2$, define function $F_{\mathrm{conv}}(h,x) = F_{T_{v\mu}}(x)$, as given by Eq. (5.38). Set $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = 2\right) = F_{T_{v\mu}}\left(T_{\mathrm{d2n}}^{(\max)}\right)$. Continue with *Step 5*.
  - Otherwise, if $h \geq 3$, continue with *Step 4*.
- ***Step 4:***
  Obtain $F_{\mathrm{conv}}(h,x) = \int_{-\infty}^{\infty} F_{\mathrm{conv}}(h-1,x)\, f_{T_{v\mu}}(t)\, \mathrm{d}t$, similar to Eq. (B.19) and using Eq. (B.20). Depending on the algorithm's implementation, $F_{\mathrm{conv}}(h-1,x)$ may be discarded at this point. Set $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = h\right) = F_{\mathrm{conv}}(h, T_{\mathrm{d2n}}^{(\max)})$.
- ***Step 5:***
  Recalculate
  $$P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right) = P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right) + P(d_{\mathrm{h}} = h)\, P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)} | d_{\mathrm{h}} = h\right)$$
  .
- ***Step 6:***
  Increment h.
- ***Step 7:***
  - If $h > d_{\mathrm{h}}^{(\max)}$, stop the algorithm. $P\left(T_{\mathrm{d2n}} \leq T_{\mathrm{d2n}}^{(\max)}\right)$ contains the result.
  - Otherwise, if $h \leq d_{\mathrm{h}}^{(\max)}$, continue with *Step 2*.

The main problem of this approach is the complexity hidden in ***Step 4*** (and Eq. (B.20)). Transcribing Eq. (B.20) to a closed form seems infeasible and its solution by calculating value tables for each iteration is likely even more cumbersome and prone to numerical inaccuracies that the Monte Carlo approach chosen in Section 5.7.2.

B.11 Calculation of the Mean Outer-Ring Segment Size

In more detail, the derivation of Eq. (4.3) is given by*

$$\begin{aligned}
\overline{A}_{\mathrm{r}^+} &= \frac{1}{d_{\mathrm{T}}} \int_0^{d_{\mathrm{T}}} A_{\mathrm{r}^+}\, \mathrm{d}d_{\mathrm{r}^+} \\
&\overset{\text{Eq. (4.2)}}{=} \frac{1}{d_{\mathrm{T}}} \int_0^{d_{\mathrm{T}}} d_{\mathrm{T}}^2 \arccos\left(\frac{d_{\mathrm{r}^+}}{d_{\mathrm{T}}}\right) - d_{\mathrm{r}^+} \sqrt{d_{\mathrm{T}}^2 - d_{\mathrm{r}^+}^2}\, \mathrm{d}d_{\mathrm{r}^+} \\
&= \frac{1}{d_{\mathrm{T}}} \left(d_{\mathrm{T}}^2 \int_0^{d_{\mathrm{T}}} \arccos\left(\frac{d_{\mathrm{r}^+}}{d_{\mathrm{T}}}\right) \mathrm{d}d_{\mathrm{r}^+} - \int_0^{d_{\mathrm{T}}} d_{\mathrm{r}^+} \sqrt{d_{\mathrm{T}}^2 - d_{\mathrm{r}^+}^2}\, \mathrm{d}d_{\mathrm{r}^+} \right)
\end{aligned}$$

* The unit of length in this section is *meter*. For conserving conciseness, the unit of length is not explicitly included in the derivations.

$$
\begin{aligned}
&\overset{\text{see note}}{=} && d_{\mathrm{T}}\left[d_{\mathrm{r}^+}\arccos\left(\frac{d_{\mathrm{r}^+}}{d_{\mathrm{T}}}\right)-\sqrt{d_{\mathrm{T}}^2-d_{\mathrm{r}^+}^2}\right]_0^{d_{\mathrm{T}}}-\frac{1}{d_{\mathrm{T}}}\left[-\frac{1}{3}\sqrt{\left(d_{\mathrm{T}}^2-d_{\mathrm{r}^+}^2\right)^3}\right]_0^{d_{\mathrm{T}}}\\
&= && d_{\mathrm{T}}\left[\left(d_{\mathrm{T}}\arccos 1-0\right)-\left(0-\sqrt{d_{\mathrm{T}}^2}\right)\right]-\frac{1}{d_{\mathrm{T}}}\left[0-\left(-\frac{1}{3}\sqrt{\left(d_{\mathrm{T}}^2\right)^3}\right)\right]\\
&= && d_{\mathrm{T}}\left[d_{\mathrm{T}}\right]-\frac{1}{d_{\mathrm{T}}}\left[\frac{1}{3}d_{\mathrm{T}}^3\right]\\
&= && d_{\mathrm{T}}^2-\frac{1}{3}d_{\mathrm{T}}^2=\frac{2}{3}d_{\mathrm{T}}^2.
\end{aligned}
$$

Note: The primitive of the minuend is derived based on [53, p. 990, n. 493]. The primitive of the subtrahend is derived based on [53, p. 966, n. 158].

Bibliography

[1] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005. ISSN 1570-8705. doi: 10.1016/j.adhoc.2003.09.010. 12

[2] Ian F. Akyildiz and Josep Miquel Jornet. Electromagnetic wireless nanosensor networks. *Nano Communication Networks*, 1(1):3–19, 2010. ISSN 1878-7789. doi: 10.1016/j.nancom.2010.04.001. 40

[3] Ian F. Akyildiz and Mehmet C. Vuran. *Wireless Sensor Networks*. John Wiley & Sons, July 2010. ISBN 978-0-470-03601-3. doi: 10.1002/9780470515181. 1, 2, 3, 4, 7, 10, 11, 12, 13, 14, 15, 16, 34, 40, 42, 43, 45, 47, 50, 169, 170

[4] Ian F. Akyildiz, Weilan Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002. doi: 10.1016/S1389-1286(01)00302-4. 10, 12

[5] Ian F. Akyildiz, Weilan Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40:102–114, 2002. doi: 10.1109/MCOM.2002.1024422. 12

[6] Ian F. Akyildiz, Mehmet C. Vuran, and Özgür B. Akan. A cross-layer protocol for wireless sensor networks. In *Proc. of the 40th Annual Conf. on Information Sciences and Systems (CISS 2006)*, pages 1102–1107, 2006. doi: 10.1109/CISS.2006.286630. 4, 7, 12, 13, 14, 15, 16, 160, 169, 170

[7] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6):6–28, 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1368893. 12

[8] Cesare Alippi, Romolo Camplani, Cristian Galperti, and Manuel Roveri. From labs to real environments: The dark side of WSNs. In Subhas C. Mukhopadhyay, Gourab S. Gupta, and Ray Y.-M. Huang, editors, *Recent Advances in Sensing Technology*, volume 49 of *Lecture Notes in Electrical Engineering (LNEE)*, pages 143–168. Springer Verlag, 2009. ISBN 978-3-642-00578-7. doi: 10.1007/978-3-642-00578-7_9. 12, 39, 40

[9] Ahmad A. A. Alkhatib, Gurvinder S. Baicher, and Waleed K. Darwish. Wireless sensor network—an advanced survey. *Int'l Journal of Engineering and Innovative Technology*, 2(7):355–369, 2013. ISSN 2277-3754. 12

[10] Béla Almási, János Roszik, and János Sztrik. Homogeneous finite-source retrial queues with server subject to breakdowns and repairs. *Mathematical and Computer Modelling*, 42:673–682, 2005. doi: 10.1016/j.mcm.2004.02.046. 107, 108

[11] Vincent G. Ambrosia, Steven S. Wegener, Donald V. Sullivan, Sally W. Buechel, Stephen E. Dunagan, James A. Brass, and Jay Stoneburner. Demonstrating UAV-acquired real-time thermal data over fires. *Photogrammetric Engineering and Remote Sensing (PE&RS)*, 69(4):391–402, 2003. URL `https://eserv.asprs.org/PERS/2003journal/apr/2003_apr_391-402.pdf`. Available online (last accessed: 25 June 2013). 29

[12] America's Climate Choices: Panel on Advancing the Science of Climate Change; National Research Council. *Advancing the Science of Climate Change*. The National Academies Press, 2010. ISBN 978-0-309-14588-6. URL `http://www.nap.edu/openbook.php?record_id=12782`. Available online (last accessed: 7 May 2012). 21, 22

[13] Stefania Amici, Martin J. Wooster, and Alessandro Piscini. Multi-resolution spectral analysis of wildfire potassium emission signatures using laboratory, airborne and spaceborne remote sensing. *Remote Sensing of Environment*, 115:1811–1823, 2011. doi: 10.1016/j.rse.2011.02.022. 29

[14] Isaac Amidror. *The Theory of the Moiré Phenomenon: Volume I: Periodic Layers*, volume 1 of *Computational Imaging and Vision*. Springer Verlag, 2nd edition, 2009. ISBN 978-1-84882-180-4. 70

[15] Isaac Amundson and Xenofon D. Koutsoukos. A survey on localization for mobile wireless sensor networks. In Richard Fuller and Xenofon D. Koutsoukos, editors, *Proc. of the 2nd Int'l Workshop on Mobile Entity Localization and Tracking in GPS-less Environnments (MELT 2009)*, volume 5801 of *Lecture Notes in Computer Science (LNCS)*, pages 235–254, 2009. ISBN 978-3-642-04378-9. doi: 10.1007/978-3-642-04385-7_16. 12, 45

[16] Farooq Anjum, Santosh Pandey, and Prathima Agrawal. Secure localization in sensor networks using transmission range variation. In *Proc. of the IEEE Int'l Conf. on Mobile Adhoc and Sensor Systems Conference (MASS 2005)*, 2005. ISBN 0-7803-9466-6/05. doi: 10.1109/MAHSS.2005.1542800. 46

[17] Thierry Antoine-Santoni, Jean-Francois Santucci, Emmanuelle de Gentili, Xavier Silvani, and Frederic Morandini. Performance of a protected wireless sensor network in a fire. analysis of fire spread and data transmission. *Sensors*, 9:5878–5893, 2009. ISSN 1424-8220. doi: 10.3390/s90805878. 10

[18] Theodoros D. Arampatzis, John A. Lygeros, and Stamatis A. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005*, pages 719–724, 2005. doi: 10.1109/.2005.1467103. 12

[19] Begoña C. Arrue, Aníbal Ollero, and J. Ramiro Martinez de Dios. An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intelligent Systems Magazine*, 15(3):64–73, 2000. ISSN 1094-7167. doi: 10.1109/5254.846287. 28

[20] Jesus R. Artalejo. Accessible bibliography on retrial queues. *Mathematical and Computer Modelling*, 30:1–6, 1999. doi: 10.1016/S0895-7177(99)00128-4. 18

[21] Jesus R. Artalejo. A classified bibliography of research on retrial queues: Progress in 1990–1999. *TOP*, 7:187–211, 1999. doi: 10.1007/BF02564721. 18

[22] Jesus R. Artalejo. Accessible bibliography on retrial queues: Progress in 2000–2009. *Mathematical and Computer Modelling*, 51:1071–1081, May 2010. ISSN 0895-7177. doi: 10.1016/j.mcm.2009.12.011. 18

[23] Jesus R. Artalejo and Antonio Gómez-Corral. *Retrial Queueing Systems: A Computational Approach*. Springer Verlag, 2008. ISBN 978-3-540-78724-2. 18

[24] Yunus E. Aslan. A framework for the use of wireless sensor networks in forest fire detection and monitoring. Master's thesis, Bilkent University, August 2010. URL `http://www.cs.bilkent.edu.tr/tech-reports/2010/BU-CE-1014.pdf`. Available online (last accessed: 25 June 2013). 10, 11, 24, 44

[25] Yunus E. Aslan, Ibrahim Korpeoglu, and Özgür Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36(6):614–625, November 2012. ISSN 0198-9715. doi: 10.1016/j.compenvurbsys.2012.03.002. Special Issue: Advances in Geocomputation. 10, 11

[26] Mohammad A. Azim, Fatemeh M. Kiaie, and Mohamed H. Ahmed. Environmental forest monitoring using wireless sensor networks: A survey. In Shafiullah Khan, Al-Sakib K. Pathan, and Nabil A. Alrajeh, editors, *Wireless Sensor Networks: Current Status and Future Trends*. CRC Press, 2012. ISBN 978-1-4665-0606-0. 10, 12

[27] Francois Baccelli, Bartlomej Blaszczyszyn, Erwan Ermel, and Paul Mühlethaler. An optimized relay self selection technique for opportunistic routing in mobile ad hoc networks. In *Proc. of the 14th European Wireless Conf. (EW 2008)*, pages 1–7, 2008. doi: 10.1109/EW.2008.4623843. 13, 14, 15, 171

[28] Francois Baccelli, Bartlomej Blaszczyszyn, and Paul Mühlethaler. On the performance of time-space opportunistic routing in multihop mobile ad hoc networks. In *Proc. of the 6th Int'l Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops (WiOPT 2008)*, pages 307–316, 2008. doi: 10.1109/WIOPT.2008.4586083. 13, 14, 15, 171

[29] Francois Baccelli, Bartlomiej Blaszczyszyn, and Paul Mühlethaler. Time–space opportunistic routing in wireless ad hoc networks: Algorithms and performance optimization by stochastic geometry. *The Computer Journal*, 53(5):592–609, June 2010. ISSN 1460-2067. doi: 10.1093/comjnl/bxp049. 12, 13, 14, 15, 171

[30] Abdelmalik Bachir, Mischa Dohler, Thomas Watteyne, and Kin K. Leung. MAC essentials for wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 12(2): 222–248, 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.020510.00058. 12

[31] Majid Bahrepour, Nirvana Meratnia, and Paul J. M. Havinga. Automatic fire detection: A survey from wireless sensor network perspective. Technical report TR-CTIT-08-73, Centre for Telematics and Information Technology, University of Twente, 2008. URL `http://eprints.eemcs.utwente.nl/14624/`. Available online (last accessed: 25 June 2013). 10, 12, 29

[32] Mert Bal, Min Liu, Weiming Shen, and Hamada Ghenniwa. Localization in cooperative wireless sensor networks: A review. In *Proc. of the 13th Int'l Conf. on Computer Supported Cooperative Work in Design (CSCWD 2009)*, pages 438–443, 2009. ISBN 978-1-4244-3535-7. doi: 10.1109/CSCWD.2009.4968098. 12, 45

[33] Daniela Ballari, Monica Wachowicz, Arnold K. Bregt, and Miguel Manso-Callejo. A mobility constraint model to infer sensor behaviour in forest fire risk monitoring. *Computers, Environment and Urban Systems*, 36(1):81–95, 2012. ISSN 0198-9715. doi: 10.1016/j.compenvurbsys.2011.06.004. 30

[34] Jerry Banks, John S. Carson II, Barry L. Nelson, and David M. Nicol. *Discrete-Event System Simulation*. Prentice-Hall, 3rd edition, 2001. ISBN 978-0-13-088702-3. 121, 153

[35] Paolo Baronti, Prashant Pillai, Vince W. C. Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications*, 30(7):1655–1695, May 2007. ISSN 0140-3664. doi: 10.1016/j.comcom.2006.12.020. 12

[36] Ferruccio Barsi, Alan A. Bertossi, Christian Lavault, Alfredo Navarra, Stephan Olariu, M. Cristina Pinotti, and Vlady Ravelomanana. Efficient location training protocols for heterogeneous sensor and actor networks. *IEEE Transactions on Mobile Computing*, 10 (3):377–391, 2011. ISSN 1536-1233. doi: 10.1109/TMC.2010.168. 45, 53

[37] Alberto Bayo, Diego Antolín, Nicolás Medrano, Belén Calvo, and Santiago Celma. Development of a wireless sensor network system for early forest fire detection. In *Proc. of the European Workshop on Smart Objects: Systems, Technologies and Applications (RFID Systech 2010)*, 2010. ISBN 978-3-8007-3282-1. 10, 11

[38] Ronald Beaubrun and Yacine Kraimia. Experiments for fire detection using a wireless sensor network. In *Proc. of the 6th Int'l Conf. on Sensor Technologies and Applications (SENSORCOMM 2012)*, 2012. ISBN 978-1-61208-207-3. 10

[39] Frank Beichelt. *Stochastic Processes in Science, Engineering and Finance*. Taylor and Francis Group, 2006. ISBN 978-1-58488-493-4. 194, 196

[40] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proc. of the 3rd ACM Int'l Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2002)*, pages 80–91, 2002. ISBN 978-1-58113-501-5. doi: 10.1145/513800.513811. 41

[41] Christian Bettstetter, Michael Gyarmati, and Udo Schilcher. An inhomogeneous spatial node distribution and its stochastic properties. In *Proc. of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM 2007)*, pages 400–404. ACM, 2007. ISBN 978-1-59593-851-0. doi: 10.1145/1298126.1298195. 85

[42] Sanjit Biswas and Robert Morris. ExOR: opportunistic multi-hop routing for wireless networks. *Computer Communication Review*, 35(4):133–144, August 2005. ISSN 0146-4833. doi: 10.1145/1090191.1080108. 12, 13, 14, 15, 170, 171

[43] Bluetooth SIG. Bluetooth specification, version 4.0. Standard, 2010. URL `http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737`. Available online (last accessed: 25 June 2013). 77

[44] Brain M. Blum, Tian He, Sang Son, and John A. Stankovic. IGF: a state-free robust communication protocol for wireless sensor networks. Technical report CS-2003-11, Department of Computer Science, School of Engineering and Applied Science, University of Virginia, Annapolis, MD, USA, 2003. URL `http://trove.nla.gov.au/version/13238044`. Available online (last accessed: 25 June 2013). 13, 14, 15, 170

[45] Gunter Bolch, János Roszik, and János Sztrik. Modeling finite-source retrial queueing systems with heterogeneous non-reliable servers by MOSEL. Technical report TR-I4-2005-01, University of Erlangen-Nürnberg, Germany, 2005. URL `http://irh.inf.unideb.hu/~jsztrik/publications/resrep/MTA_DFG05paper.pdf`. Available online (last accessed: 25 June 2013). 19

[46] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi. *Queueing Networks and Markov Chains*. John Wiley & Sons, New York, 2nd edition, 2006. ISBN 978-0-471-56525-3. 90, 91, 101, 102, 104, 105, 113, 115, 117, 122, 123, 124, 155, 194

[47] Gunter Bolch, János Roszik, János Sztrik, and Patrick Wüchner. Modeling finite-source retrial queueing systems with unreliable heterogeneous servers and different service policies using MOSEL. Technical report MIP-0611, Fakultät für Informatik und Mathematik, Universität Passau, 2006. URL `http://www.fim.uni-passau.de/forschung/mip-berichte/MIP-0611.pdf`. Available online (last accessed: 25 June 2013). 19

[48] Azzedine Boukerche, Horacio A. B. F. Oliveira, Eduardo F. Nakamura, and Antonio A. F. Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications Magazine*, 14(6):6–12, 2007. ISSN 1536-1284. doi: 10.1109/MWC.2007.4407221. 12, 45

[49] David M. J. S. Bowman, Jennifer K. Balch, Paulo Artaxo, William J. Bond, Jean M. Carlson, Mark A. Cochrane, Carla M. D'Antonio, Ruth S. DeFries, John C. Doyle, Sandy P. Harrison, Fay H. Johnston, Jon E. Keeley, Meg A. Krawchuk, Christian A. Kull, J. Brad Marston, Max A. Moritz, I. Colin Prentice, Christopher I. Roos, Andrew C. Scott, Thomas W. Swetnam, Guido R. van der Werf, and Stephen J. Pyne. Fire in the earth system. *Science*, 324(5926):481–484, 2009. doi: 10.1126/science.1163886. 21, 22

[50] Ross A. Bradstock. A biogeographic model of fire regimes in Australia: Current and future implications. *Global Ecology and Biogeography*, 19:145–158, 2010. ISSN 1466-8238. doi: 10.1111/j.1466-8238.2009.00512.x. 22

[51] Tamás Bérczes, János Sztrik, Péter Orosz, Pascal Moyal, Nicolaos Limnios, and Stylianos Georgiadis. Tool supported modeling of sensor communication networks by using finite-source priority retrial queues. *Carpathian Journal of Electronic and Computer Engineering*, 5:13–18, 2012. ISSN 1844-9689. 17, 18

[52] Diana Bri, Miguel Garcia, Jaime Lloret, and Petre Dini. Real deployments of wireless sensor networks. In *Proc. of the 3rd Int'l Conf. on Sensor Technologies and Applications (SENSORCOMM 2009)*, 2009. ISBN 978-0-7695-3669-9. doi: 10.1109/SENSORCOMM.2009.69. 10, 12

[53] Ilja N. Bronstein, Konstantin A. Semendjajew, Gerhard Musiol, and Heiner Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch Verlag, 3rd edition, 1997. ISBN 978-3-8171-2003-1. 114, 198

[54] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000. ISSN 1070-9916. doi: 10.1109/98.878533. 46

[55] Stephen S. O. Burgess, Mark L. Kranz, N. E. Turner, Rachel Cardell-Oliver, and Todd E. Dawson. Harnessing wireless sensor technologies to advance forest ecology and agricultural research. *Agricultural and Forest Meteorology*, 150(1):30–37, 2010. ISSN 0168-1923. doi: 10.1016/j.agrformet.2009.08.002. 10, 11

[56] Doug Campbell, Wally G. Born, Judi Beck, Bill Bereska, Kurt Frederick, and Sun Hua. Airborne Wildfire Intelligence System: A decision support tool for wildland fire managers in Alberta. In Xavier P. Maldague and Andres E. Rozlosnik, editors, *Proc. of the 24th Annual Thermal Infrared Applications Conference (ThermoSense XXIV)*, volume 4710 of *SPIE Proceedings*, pages 159–170, 2002. doi: 10.1117/12.459563. 29

[57] David W. Casbeer, Derek B. Kingston, Randal W. Beard, and Timothy W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *Int'l Journal of System Science*, 37(6):351–360, 2006. doi: 10.1080/00207720500438480. 29

[58] Colin Chaballier. SensLAB node hardware. Deliverable D1.1a, SensLAB Consortium, July 2008. URL `http://senslab.inrialpes.fr/wp-content/uploads/2010/07/senslab-deliverable-d1.1a.pdf`. Available online (last accessed: 25 June 2013). 48

[59] Hongyang Chen, Pei Huang, Marcelo Martins, Hing Cheung So, and Kaoru Sezaki. Novel centroid localization algorithm for three-dimensional wireless sensor networks. In *Proc. of the 4th Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCOM 2008)*, 2008. ISBN 978-1-4244-2108-4. doi: 10.1109/WiCom.2008.841. 46

[60] Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou. Sensor network security: A survey. *IEEE Communications Surveys and Tutorials*, 11(2):52–73, 2009. ISSN 1553-877X. doi: 10.1109/SURV.2009.090205. 12, 85

[61] Xiao Chen and Benliang Zhang. Improved DV-hop node localization algorithm in wireless sensor networks. *Int'l Journal of Distributed Sensor Networks*, 2012:213980:1–7, 2012. doi: 10.1155/2012/213980. 47

[62] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. A survey of localization in wireless sensor network. *Int'l Journal of Distributed Sensor Networks*, 2012:1–12, 2012. doi: 10.1155/2012/962523. 12, 45

[63] Romit R. Choudhury and Nitin H. Vaidya. MAC-layer anycasting in ad hoc networks. *Computer Communication Review*, 34(1):75–80, January 2004. ISSN 0146-4833. doi: 10.1145/972374.972388. 12, 13, 14, 170

[64] Coalesenses GmbH. *iSense Core Module 3 Datasheet—CM30I, CM30U, CM30HP*, August 2012. URL `http://www.coalesenses.com/download/data_sheets/DS_CM30X_1v3.pdf`. Version 1.3. Available online (last accessed: 24 June 2013). 49

[65] Alan Colvin. CSMA with collision avoidance. *Computer Communications*, 6(5):227–235, 1983. ISSN 0140-3664. doi: 10.1016/0140-3664(83)90084-1. 34

[66] Peter I. Corke, Stefan E. Hrabar, Ron Peterson, Daniela Rus, Srikanth Saripalli, and Gaurav S. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA 2004)*, volume 4, pages 3602–3608, May 2004. doi: 10.1109/ROBOT.2004.1308811. 39

[67] Zachary M. Crane. Short distance ground wave propagation modeling in irregular and forested environments. Master's thesis, University of Tennessee, 2011. URL `http://trace.tennessee.edu/utk_gradthes/1063`. Available online (last accessed: 24 June 2013). 44

[68] Crossbow Technology, Inc. *MPR-MIB Users Manual*, June 2007. URL `http://bullseye.xbow.com:81/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf`. Revision A (PN: 7430-0021-08). Available online (last accessed: 24 June 2013). 48

[69] Osman S. da Penha, Jr. and Eduardo F. Nakamura. Fusing light and temperature data for fire detection. In *Proc. of the IEEE Symposium on Computers and Communications (ISCC 2010)*, 2010. ISBN 978-1-4244-7754-8. doi: 10.1109/ISCC.2010.5546519. 10, 21

[70] Arnoldo Díaz-Ramírez, Luis A. Tafoya, Jorge A. Atempa, and Pedro Mejía-Alvarez. Wireless sensor networks and fusion information methods for forest fire detection. In *Proc. of the Iberoamerican Conf. on Electronics Engineering and Computer Science (CIIECC 2012)*, volume 3 of *Procedia Technology*, pages 69–79, 2012. doi: 10.1016/j.protcy.2012.03.008. 10

[71] Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen, Maarten Weyn, and Jerry Bracke. Localization in wireless sensor networks. Master thesis paper (e-lab 2008-2009), Artesis University College of Antwerp, 2008. URL http://wsnlocalizationscala.googlecode.com/svn/trunk/Portfolio/Other/Papers/Surveypaper.pdf. Available online (last accessed: 25 June 2013). 12, 46

[72] William J. De Groot. Interpreting the canadian forest fire weather index (FWI) system. In *Proc. of the 4th Central Regional Fire Weather Committee Scientific and Technical Seminar*, 1987. URL http://www.dnr.state.mi.us/WWW/FMD/WEATHER/Reference/FWI_Background.pdf. Available online (last accessed: 25 June 2013). 23

[73] Ilker Demirkol, Cem Ersoy, and Fatih Alagoz. MAC protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 44(4):115–121, 2006. ISSN 0163-6804. doi: 10.1109/MCOM.2006.1632658. 12

[74] Santpal S. Dhillon and Krishnendu Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC 2003)*, volume 3, pages 1609–614, March 2003. doi: 10.1109/WCNC.2003.1200627. 39

[75] Isabel Dietrich and Falko Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(1):5:1–5:39, February 2009. doi: 10.1145/1464420.1464425. 12

[76] Ioannis Dimitriou. Analysis of a priority retrial queue with dependent vacation scheme and application to power saving in wireless communication systems. *The Computer Journal*, Advance Access:1–18, September 2012. ISSN 1460-2067. doi: 10.1093/comjnl/bxs125. 16, 17, 18

[77] Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2001)*, volume 3, pages 1655–1663, 2001. doi: 10.1109/INFCOM.2001.916662. 46

[78] Qian Dong and W. Dargie. A survey on mobility and mobility-aware MAC protocols in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 15(1):88–100, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.013012.00051. 12

[79] Wei Dong, Chun Chen, Xue Liu, and Jiajun Bu. Providing OS support for wireless sensor networks: Challenges and approaches. *IEEE Communications Surveys and Tutorials*, 12(4):519–530, 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.032610.00045. 12

[80] Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti. Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation. *IEEE Transactions on Systems, Man, and Cybernetics*, 43(4):1003–1012, 2012. ISSN 2168-2216. doi: 10.1109/TSMCA.2012.2224335. 28

[81] David M. Doolin and Nicholas Sitar. Wireless sensors for wildfire monitoring. In Masayoshi Tomizuka, editor, *Proc. of Smart Structures and Materials 2005: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, volume 5765 of *SPIE Proceedings*, 2005. ISBN 978-0-8194-5746-2. doi: 10.1117/12.605655. 10

[82] Falko Dressler. *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons, 2007. ISBN 978-0-470-02820-9. 3, 11, 12, 47, 53

[83] Falko Dressler. A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications*, 31(13):3018–3029, 2008. doi: 10.1016/j.comcom.2008.02.001. 12

[84] Samina Ehsan and Bechir Hamdaoui. A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks. *IEEE Communications Surveys and Tutorials*, 14(2):265–278, 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.020211.00058. 12

[85] Melike Erol-Kantarci, Hussein T. Mouftah, and Sema Oktug. A survey of architectures and localization techniques for underwater acoustic sensor networks. *IEEE Communications Surveys and Tutorials*, 13(3):487–502, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.020211.00035. 12

[86] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. of the ACM/IEEE Int'l Conf. on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999. ACM. doi: 10.1145/313451.313556. 39

[87] European Space Agency (ESA) and Committee on Earth Observation Satellites (CEOS). The earth observation handbook – climate change special edition. ESA SP-1315, November 2011. URL `http://www.eohandbook.com/`. Available online (last accessed: 14 Nov. 2011). 27

[88] Gennadi I. Falin and James G. C. Templeton. *Retrial Queues*. Chapman & Hall, 1997. ISBN 978-0-412-78550-4. 18, 19, 105, 107, 108

[89] Kai-Wei Fan, Sha Liu, and Prasun Sinha. Structure-free data aggregation in sensor networks. *IEEE Transactions on Mobile Computing*, 6(8):929–942, 2007. ISSN 1536-1233. doi: 10.1109/TMC.2007.1011. 12, 13, 14, 15, 171

[90] Weiwei Fang, Zhen Liu, and Feng Liu. A cross-layer protocol for reliable and efficient communication in wireless sensor networks. *Int'l Journal of Innovative Computing, Information and Control*, 8(10):7185–7198, 2012. ISSN 1349-4198. URL `http://www.ijicic.org/ijicic-11-09047.pdf`. Available online (last accessed: 25 June 2013). 12, 13, 14, 15, 172

[91] William Feller. *An introduction to probability theory and its applications*, volume 1. John Wiley & Sons, 3rd edition, 1968. ISBN 978-0-471-25708-0. 105

[92] Jorge Fernandez-Berni, Ricardo Carmona-Galan, and Angel Rodriguez-Vazquez. Case study: Early detection of forest fires. In *Low-Power Smart Imagers for Vision-Enabled Sensor Networks*, chapter 7, pages 127–146. Springer Verlag, 2012. ISBN 978-1-4614-2391-1. doi: 10.1007/978-1-4614-2392-8_7. 25

[93] Mark A. Finney, Jack D. Cohen, Sara S. McAllister, and W. Matt Jolly. On the need for a theory of wildland fire spread. *Int'l Journal of Wildland Fire*, 22(1):25–36, 2012. doi: 10.1071/WF11117. 80

[94] Gerd Fischer. *Lineare Algebra*, volume 17 of *vieweg studium - Grundkurs Mathematik*. Vieweg Verlag, 11th edition, 1997. ISBN 978-3-528-77217-8. 114

[95] Sylvia Frühwirth-Schnatter. *Finite mixture and Markov switching models*. Springer Verlag, 2006. ISBN 978-0-387-35768-3. doi: 10.1007/978-0-387-35768-3. 121

[96] Sachin H. Gajjar, Shrikant N. Pradhan, and Kankar S. Dasgupta. Cross layer architectural approaches for wireless sensor networks. In *Proc. of the IEEE Conf. on Recent Advances in Intelligent Computational Systems (RAICS 2011)*, pages 557–562, 2011. doi: 10.1109/RAICS.2011.6069374. 12

[97] Carlos F. García-Hernández, Pablo H. Ibarguengoytia-Gonzalez, Joaquín García-Hernández, and Jesús A. Pérez-Díaz. Wireless sensor networks and applications: A survey. *IJCSNS International Journal of Computer Science and Network Security*, 7 (3):264–273, 2007. URL `http://paper.ijcsns.org/07_book/200703/20070338.pdf`. Available online (last accessed: 25 June 2013). 12

[98] Angelo Genovese, Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti. Wildfire smoke detection using computational intelligence techniques. In *Proc. of the IEEE Int'l Conf. on Computational Intelligence for Measurement Systems and Applications (CIMSA 2011)*, 2011. doi: 10.1109/CIMSA.2011.6059930. 28

[99] Nawel Gharbi. Modeling and performance evaluation of small cell wireless networks with base station channels breakdowns. In *Proc. of the 8th Int'l Conf. on Wireless and Mobile Communications (ICWMC 2012)*, pages 42–48, 2012. ISBN 978-1-61208-203-5. URL `http://www.thinkmind.org/index.php?view=article&articleid=icwmc_2012_3_10_20237`. Available online (last accessed: 25 June 2013). 19

[100] Nawel Gharbi. A numerical approach for performance evaluation of cellular mobile networks with channels breakdowns. In *Proc. of the 12th Int'l Conf. on Networks (ICN 2013)*, pages 190–196, January 2013. ISBN 978-1-61208-245-5. URL `http://www.thinkmind.org/index.php?view=article&articleid=icn_2013_8_40_10115`. Available online (last accessed: 25 June 2013). 19

[101] Nawel Gharbi and Claude Dutheillet. An algorithmic approach for analysis of finite-source retrial systems with unreliable servers. *Computers & Mathematics with Applications*, 62(6):2535–2546, September 2011. ISSN 0898-1221. doi: 10.1016/j.camwa.2011.03.109. 19

[102] Nawel Gharbi and Malika Ioualalen. GSPN analysis of retrial systems with servers breakdowns and repairs. *Applied Mathematics and Computation*, 174:1151–1168, 2006. doi: 10.1016/j.amc.2005.06.005. 19

[103] Silvia Giordano, Ivan Stojmenovic, and Ljubica Blazevic. Position based routing algorithms for ad hoc networks: A taxonomy. In Xiuzhen Cheng, Xiao Huang, and Ding-Zhu Du, editors, *Ad Hoc Wireless Networking*, Network Theory and Applications, pages 103–136. Kluwer Academic Publishers, 2003. ISBN 978-1-4020-7712-8. 53

[104] Osman Günay, Behçet U. Töreyin, Kivanç Köse, and Aydin E. Çetin. Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video. *IEEE Transactions on Image Processing*, 21(5):2853–2865, 2012. ISSN 1057-7149. doi: 10.1109/TIP.2012.2183141. 28

[105] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996. ISBN 978-0-8018-5414-9. 47

[106] Ross W. Gorte and Kelsi Bracmort. Forest fire/wildfire protection. CRS report for congress RL30755, Congressional Research Service, March 2012. URL `http://www.fas.org/sgp/crs/misc/RL30755.pdf`. Available online (last accessed: 24 June 2013). 23

[107] Paul H. Greenfield, Woodrow Smith, and David C. Chamberlain. PHOENIX—The new forest service airborne infrared fire detection and mapping system. In *Proc. of the 2nd Int'l Wildland Fire Ecology and Fire Management Congress and the 5th Symposium on Fire and Forest Meteorology*, 2003. URL `https://ams.confex.com/ams/pdfpapers/66675.pdf`. Available online (last accessed: 25 June 2013). 29

[108] Malka N. Halgamuge. *Performance Evaluation and Enhancement of Mobile and Sensor Networks*. Phd thesis, University of Melbourne, 2006. URL `http://repository.unimelb.edu.au/10187/1637`. Available online (last accessed: 25 June 2013). 12

[109] Guangjie Han, Deokjai Choi, and Wontaek Lim. A novel reference node selection algorithm based on trilateration for indoor sensor networks. In *Proc. of the 7th IEEE Int'l Conf. on Computer and Information Technology (CIT 2007)*, 2007. doi: 10.1109/CIT.2007.15. 68

[110] Guangjie Han, Huihui Xu, Trung Q. Duong, Jinfang Jiang, and Takahiro Hara. Localization algorithms of wireless sensor networks: A survey. *Telecommunication Systems*, online:1–18, 2011. ISSN 1018-4864. doi: 10.1007/s11235-011-9564-7. 12, 45

[111] Stijn Hantson, Marc Padilla, Dante Corti, and Emilio Chuvieco. Strengths and weaknesses of MODIS hotspots to characterize global fire occurrence. *Remote Sensing of Environment*, 131:152–159, 2013. ISSN 0034-4257. doi: 10.1016/j.rse.2012.12.004. 27

[112] Jie Hao, Baoxian Zhang, and Hussein T. Mouftah. Routing protocols for duty cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, 50(12):116–123, 2012. ISSN 0163-6804. doi: 10.1109/MCOM.2012.6384460. 12

[113] Sandy P. Harrison, Jennifer R. Marlon, and Patrick J. Bartlein. Fire in the earth system. In John Dodson, Eduardo F. J. Mulder, and Edward Derbyshire, editors, *Changing Climates, Earth Systems and Society*, International Year of Planet Earth, pages 21–48. Springer Verlag, 2010. ISBN 978-90-481-8716-4. doi: 10.1007/978-90-481-8716-4_3. 21, 22, 27

[114] Roberta A. Hartford and William H. Frandsen. When it's hot, it's hot ... or maybe it's not! (Surface flaming may not portend extensive soil heating). *Int'l Journal of Wildland Fire*, 2(3):139–144, 1992. ISSN 1049-8001. doi: 10.1071/WF9920139. 23

[115] Carl Hartung, Richard Han, Carl Seielstad, and Saxon Holbrook. FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *Proc. of the 4th Int'l Conf. on Mobile Systems, Applications and Services (MobiSys 2006)*, 2006. ISBN 978-1-59593-195-5. doi: 10.1145/1134680.1134685. 10

[116] Simon S. Haykin. *Adaptive filter theory*. Prentice-Hall information and system sciences series. Prentice-Hall, 3rd edition, 1996. ISBN 978-0-13-322760-4. 47

[117] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proc. of the 9th Annual Int'l Conf. on Mobile Computing and Networking (MobiCom 2003)*, pages 81–95. ACM, 2003. ISBN 1-58113-753-2. doi: 10.1145/938985.938995. 46

[118] Xiaobing He, Michael Niedermeier, and Hermann de Meer. Dynamic key management in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 36(2):611–622, 2013. ISSN 1084-8045. doi: 10.1016/j.jnca.2012.12.010. 12

[119] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using-wireless sensor networks. *Ad Hoc and Sensor Wireless Networks*, 7:169–224, 2009. doi: 10.1109/MOBHOC.2007.4428702. 10, 11

[120] Amy E. Hessl. Pathways for climate change effects on fire: Models, data, and uncertainties. *Progress in Physical Geography*, 35(3):393–407, 2011. doi: 10.1177/0309133311407654. 22, 27

[121] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer Magazine*, 34(8):57–66, August 2001. ISSN 0018-9162. doi: 10.1109/2.940014. 53

[122] Jeffrey Hightower and Gaetano Borriello. Location sensing techniques. Technical report UW CSE 01-07-01, University of Washington, Department of Computer Science and Engineering, Seattle, WA, July 2001. URL `http://www.hightowerweb.org/pubs/hightower2001techniques/hightower2001techniques.pdf`. Available online (last accessed: 25 June 2013). 45

[123] Thea K. Hincks, Bruce D. Malamud, R. Stephen J. Sparks, Martin J. Wooster, and Timothy J. Lynham. Risk assessment and management of wildfires. In *Risk and Uncertainty Assessment for Natural Hazards*, chapter 12. Cambridge University Press, 2013. ISBN 978-1-107-00619-5. 21, 22, 23

[124] Pei Huang, Li Xiao, Soroor Soltani, Matt W. Mutka, and Ning Xi. The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 15(1):101–120, 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.040412.00105. 12

[125] Chi J. Hwang. *Ultra-low power radio transceiver for wireless sensor networks*. Phd thesis, University of Glasgow, 2010. URL http://theses.gla.ac.uk/1690/. Available online (last accessed: 25 June 2013). 12, 40, 48

[126] IEEE. ANSI/IEEE Std 802.11-1999, IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Standard, 1999. 77

[127] IEEE. IEEE Std 802.15.4-2003, IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS). Standard, 2003. 11, 45, 77, 98, 150

[128] IEEE. IEEE Std 802.15.3-2008, IEEE Standard: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Standard, 2008. 77

[129] IEEE. IEEE Std 802.11-2012, IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Standard, 2012. 77, 98

[130] IEEE. IEEE Std 802.15.4e-2012, IEEE Standard: Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC sublayer. Standard, 2012. 11

[131] Intel Corporation. Intel developer forum 22nm news facts. News Fact Sheet, September 2009. URL http://www.intel.de/content/www/de/de/pdf-pages/22nm-technology-idf2009-factsheet.html. Available online (last accessed: 16 Oct. 2012). 182

[132] ITRS Organization. International technology roadmap for semiconductors – 2009 edition, 2009. URL http://www.itrs.net/Links/2009ITRS/Home2009.htm. Available online (last accessed: 16 Oct. 2012). 182

[133] Thomas Jahn. Coordinates in hexagon-based tile maps, April 2002. URL http://www.gamedev.net/page/resources/_/technical/game-programming/coordinates-in-hexagon-based-tile-maps-r1800. Available online (last accessed: 14 Feb. 2012). 74, 182

[134] Ankit K. Jain, Ankit Khare, and Kaushlendra K. Pandey. Developing an efficient framework for real time monitoring of forest fire using wireless sensor network. In *Proc. of the 2nd IEEE Int'l Conf. on Parallel, Distributed and Grid Computing (PDGC-2012)*, 2012. ISBN 978-1-4673-2922-4. doi: 10.1109/PDGC.2012.6449927. 10, 11, 21

[135] Raj Jain. *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1991. 152

[136] D. F. Jenolin Flora, V. Kavitha, and M. Muthuselvi. A survey on congestion control techniques in wireless sensor networks. In *Proc. of the Int'l Conf. on Emerging Trends in Electrical and Computer Technology (ICETECT 2011)*, pages 1146–1149, 2011. ISBN 978-1-4244-7923-8. doi: 10.1109/ICETECT.2011.5760292. 12

[137] Finn V. Jensen and Thomas D. Nielsen. *Bayesian Networks and Decision Graphs*. Information Science and Statistics Series. Springer Verlag, 2nd edition, 2007. ISBN 978-0-387-68282-2. 30

[138] Gaurav G. Joshi, Carl B. Dietrich, Christopher R. Anderson, William G. Newhall, William A. Davis, James C. Isaacs, and Gregory Barnett. Near-ground channel measurements over line-of-sight and forested paths. *IEE Proceedings Microwaves, Antennas and Propagation*, 152(6):589–596, 2005. doi: 10.1049/ip-map:20050013. 40

[139] Adrian F. Kacsó and Roland Wismüller. Comparison of unicast and receiver-based routing services in wireless sensor networks. In *Proc. of the 6th Int'l Conf. on Wireless and Mobile Communications (ICWMC 2010)*, pages 343–348. IEEE, September 2010. ISBN 978-1-4244-8021-0. doi: 10.1109/ICWMC.2010.23. 4, 12, 13, 14, 172

[140] Joseph M. Kahn, Randy H. Katz, and Kristofer S. J. Pister. Emerging challenges: Mobile networking for "Smart Dust". *Journal of Communications and Networks*, 2(3):188–196, 2000. ISSN 1976-5541. 40

[141] Mark P. Kaminskiy. *Reliability Models for Engineers and Scientists*. Taylor and Francis Group, 2013. ISBN 1-4665-6592-6. 194

[142] Holger Karl and Andreas Willig. A short survey of wireless sensor networks. Technical report TKN-03-018, TU Berlin, Telecommunication Networks Group, 2003. URL `http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/TechReport_03_018.pdf`. Available online (last accessed: 25 June 2013). 12

[143] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005. ISBN 978-0-470-09510-2. 3, 7, 10, 11, 12, 39, 40, 45, 47, 48, 49, 50, 82, 83, 150

[144] Väinö Kelhä, Yrjö Rauste, and Alessandra Buongiorno. Forest fire detection by satellites for fire control. *Automation Technology Review*, 2000:34–40, 2000. URL `http://virtual.vtt.fi/virtual/space/firealert/forest_fire_detection.pdf`. VTT Automation. Available online (last accessed: 25 June 2013). 27

[145] Norman Kerle, Stephan Heuel, and Norbert Pfeiffer. Real-time data collection and information generation using airborne sensors. In *Geospatial information technology for emergency response*, volume 6 of *ISPRS Book Series*. Taylor and Francis Group, 2008. ISBN 978-0-415-42247-5. 29

[146] Manijeh Keshtgary, Masood Fasihy, and Zahra Ronaghi. Performance evaluation of hop-based range-free localization methods in wireless sensor networks. *ISRN Communications and Networking*, 2011:485486:1–6, 2011. doi: 10.5402/2011/485486. 45

[147] Shafiullah Khan, Al-Sakib Khan Pathan, and Nabil Ali Alrajeh, editors. *Wireless Sensor Networks: Current Status and Future Trends*. CRC Press, 2012. ISBN 978-1-4665-0606-0. 12

[148] Joohwan Kim, Xiaojun Lin, Ness B. Shroff, and Prasun Sinha. On maximizing the lifetime of delay-sensitive wireless sensor networks with anycast. In *Proc. of the 27th*

IEEE Conf. on Computer Communications (INFOCOM 2008), pages 807–815, 2008. doi: 10.1109/INFOCOM.2008.132. 12, 13, 14, 171

[149] Joohwan Kim, Xiaojun Lin, Ness B. Shroff, and Prasun Sinha. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. Technical report, Purdue University, IN, USA, 2008. URL `http://www.ifp.illinois.edu/~joohwan/Kim08tech2.pdf`. Available online (last accessed: 25 June 2013). 12, 13, 14, 171

[150] Joohwan Kim, Xiaojun Lin, Ness B. Shroff, and Prasun Sinha. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Transactions on Networking*, 18(2):515–528, 2010. ISSN 1063-6692. doi: 10.1109/TNET.2009.2032294. 12, 13, 14, 171

[151] Joohwan Kim, Xiaojun Lin, and Ness B. Shroff. Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks. *IEEE/ACM Transactions on Networking*, 19(2):484–497, 2011. ISSN 1063-6692. doi: 10.1109/TNET.2010.2072515. 12, 13, 14, 171

[152] Kinam Kim and Gwan-Hyeob Koh. Future memory technology including emerging new memories. In *Proc. of the 24th Int'l Conf. on Microelectronics (MIEL 2004)*, volume 1, pages 377–384. IEEE, 2004. doi: 10.1109/ICMEL.2004.1314646. 182

[153] Leonard Kleinrock. *Queueing Systems*, volume 1: Theory. John Wiley & Sons, 1975. ISBN 978-0-471-49110-1. 90

[154] Leonard Kleinrock and John Silvester. Optimum transmission radii for packet radio networks or why Six is a magic number. In *Proc. of the IEEE National Telecommunications Conf. (NTC 1978), Conference Record*, volume 1, 1978. URL `www.lk.cs.ucla.edu/data/files/Kleinrock/OptimumTransmissionRadiiforPacketRadio.pdf`. Available online (last accessed: 25 June 2013). 41

[155] Charalabos Kontoes, Iphigenia Keramitsoglou, Nicolaos Sifakis, and Pavlos Konstantinidis. SITHON: An airborne fire detection system compliant with operational tactical requirements. *Sensors*, 9:1204–1220, 2009. ISSN 1424-8220. doi: 10.3390/s90201204. 29

[156] Bilgin Kosucu, Kerem Irgan, Gurhan Kucuk, and Sebnem Baydere. FireSenseTB: A wireless sensor networks testbed for forest fire detection. In *Proc. of the Int'l Conf. on Wireless Communications and Mobile Computing (IWCMC 2009)*, 2009. ISBN 978-1-60558-569-7. doi: 10.1145/1582379.1582636. 10

[157] Georgios Koutitas, Fotini-Niovi Pavlidou, and Ljiljana Jankovic. A comparative study of two alternative wildfire models, with applications to WSN topology control. In Giovanni Perona and Carlos A. Brebbia, editors, *Modelling, Monitoring and Management of Forest Fires II*, volume 137 of *WIT Transactions on Ecology and the Environment*. WIT Press Publishing, 2010. ISBN 978-1-84564-453-6. doi: 10.2495/FIVA100031. 10

[158] Robert Kremens, Jason Faulring, Ann Gallagher, Adolph Seema, and Anthony Vodacek. Autonomous field-deployable wildland fire sensors. *Int'l Journal of Wildland Fire*, 12 (2):237–244, 2003. ISSN 1049-8001. doi: 10.1071/WF02055. 10

[159] Wolfgang Krüll, Robert Tobera, Ingolf Willms, Helmut Essen, and Nora von Wahl. Early forest fire detection and verification using optical smoke, gas and microwave sensors. In *Proc. of the Int'l Symposium on Safety Science and Technology (ISSST 2012)*, volume 45 of *Procedia Engineering*, pages 584–594, 2012. doi: 10.1016/j.proeng.2012.08.208. 29

[160] Horst Kuchling. *Taschenbuch der Physik*. Harri Deutsch Verlag, 13th edition, 1991. ISBN 3-8171-1020-0. 44

[161] Elena G. Kulikova. Phytosanitary risks. In Michel Arbez, Yves Birot, and Jean-Michel Camus, editors, *Proc. of the 8th Annual EFI Conf. on Risk Management and Sustainable Forestry*, volume 45 of *EFI Proceedings*, pages 43–53. European Forest Institute, 2002. ISBN 952-9844-93-X. 21, 23

[162] Raghavendra V. Kulkarni, Anna Forster, and Ganesh K. Venayagamoorthy. Computational intelligence in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 13(1):68–96, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.040310.00002. 12

[163] Sunil Kulkarni, Aravind Iyer, and Catherine Rosenberg. An address-light, integrated MAC and routing protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(4):793–806, 2006. ISSN 1063-6692. doi: 10.1109/TNET.2006.880163. 12, 13, 14, 15, 171

[164] Way Kuo and Ming J. Zuo. *Optimal Reliability Modeling: Principles and Applications*. John Wiley & Sons, 2003. ISBN 978-0-471-39761-8. 194

[165] James F. Kurose and Keith W. Ross. *Computer Networking*. Pearson Education, 3rd edition, 2005. ISBN 978-0-321-22735-5. 12, 76, 77, 149

[166] Andreas J. Lachenmann. *A Cross-Layer Framework for Sensor Networks*. Phd thesis, Institut für Parallele und Verteilte Systeme (IPVS), Universität Stuttgart, Germany, 2008. URL `http://elib.uni-stuttgart.de/opus/volltexte/2008/3568/`. Available online (last accessed: 25 June 2013). 3, 12, 156

[167] Andreas J. Lachenmann, Pedro J. Marrón, Matthias Gauger, Daniel Minder, Olga Saukh, and Kurt Rothermel. Removing the memory limitations of sensor networks with flash-based virtual memory. In *Proc. of the European Conf. on Computer Systems (EuroSys 2007)*. ACM, 2007. doi: 10.1145/1272996.1273012. 47

[168] Sami M. Lasassmeh and James M. Conrad. Time synchronization in wireless sensor networks: A survey. In *Proc. of the Annual IEEE Region 3 Technical, Professional, and Student Conf. (SoutheastCon 2010)*, pages 242–245, 2010. doi: 10.1109/SECON.2010.5453878. 12, 50

[169] Guy Latouche and Vaidyanathan Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Series on Statistics and Applied Probability. ASA-SIAM, Philadelphia, PA, 1999. ISBN 978-0-89871-425-8. 5, 19, 110, 111, 113, 114

[170] Stephen S. Lavenberg and Martin Reiser. Stationary state probabilities at arrival instants for closed queueing networks with multiple types of customers. *Journal of Applied Probability*, 17(4):1048–1061, 1980. ISSN 0021-9002. 105

[171] Loukas Lazos and Radha Poovendran. SeRLoc: Secure range-independent localization for wireless sensor networks. In *Proc. of the 3rd ACM Workshop on Wireless Security (WiSe 2004)*, pages 21–30, 2004. ISBN 1-58113-925-X. doi: 10.1145/1023646.1023650. 46

[172] Loukas Lazos and Radha Poovendran. HiRLoc: High-resolution robust localization for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(2): 233–246, 2006. ISSN 0733-8716. doi: 10.1109/JSAC.2005.861381. 46

[173] Thu N. Le, Peter H. J. Chong, Xue J. Li, and Wai Y. Leong. A simple grid-based localization technique in wireless sensor networks for forest fire detection. In *Proc. of the 2nd Int'l Conf. on Communication Software and Networks (ICCSN 2010)*, 2010. ISBN 978-1-4244-5727-4. doi: 10.1109/ICCSN.2010.108. 30

[174] Richard Legeckis, Christopher W. Brown, and Paul S. Chang. Geostationary satellites reveal motions of ocean surface fronts. *Journal of Marine Systems*, 37:3–15, 2002. doi: 10.1016/S0924-7963(02)00192-6. 27

[175] Leigh B. Lentile, Zachary A. Holden, Alistair M. S. Smith, Michael J. Falkowski, Andrew T. Hudak, Penelope Morgan, Sarah A. Lewis, Paul E. Gessler, and Nate C. Benson. Remote sensing techniques to assess active fire characteristics and post-fire effects. *International Journal of Wildland Fire*, 15(3):319–345, 2006. URL `http://www.treesearch.fs.fed.us/pubs/24613`. Available online (last accessed: 25 June 2013). 27

[176] Guanghui Li, Hong Li, Hailin Feng, Zhi Wang, and Ziwei Xuan. The effect of wireless sensor nodes deployment density in forest fire monitoring quality evaluation. *Journal of Networks*, 7(7):1116–1122, 2012. doi: 10.4304/jnw.7.7.1116-1122. 10

[177] Jian Li, Jianmin Zhang, and Liu Xiande. A weighted DV-hop localization scheme for wireless sensor networks. In *Proc. of the Int'l Conf. on Scalable Computing and Communications; 8th Int'l Conf. on Embedded Computing (SCALCOM-EMBEDDEDCOM 2009)*, pages 269–272, 2009. ISBN 978-0-7695-3825-9. doi: 10.1109/EmbeddedComScalCom.2009.55. 47

[178] Zhetao Li, Renfa Li, Yehua Wei, and Tingrui Pei. Survey of localization techniques in wireless sensor networks. *Information Technology Journal*, 9(8):1754–1757, 2010. ISSN 1812-5638. doi: 10.3923/itj.2010.1754.1757. 12, 45

[179] Hyuk Lim and Jennifer C. Hou. Distributed localization for anisotropic sensor networks. *ACM Transactions on Sensor Networks*, 5(2):11:1–15, 2009. doi: 10.1145/1498915.1498917. 46

[180] Alan D. Little, Doreen O. Neil, Glen W. Sachse, Jack Fishman, and Arlin J. Krueger. Remote sensing from geostationary orbit: GEO TROPSAT, a new concept for atmospheric remote sensing. In Hiroyuki Fujisada, editor, *Proc. of the Conf. on Sensors, Systems, and Next-Generation Satellites*, volume 3221 of *SPIE*, pages 480–488, 1997. doi: 10.1117/12.298116. 27

[181] Dongxiao Liu, Yujun Kuang, and Wei Wei. Research and improvement of DVHOP localization algorithm in wireless sensor networks. In *Proc. of the Int'l Conf. on Computational Problem-Solving (ICCP 2010)*, pages 47–50, 2010. ISBN 978-1-4244-8654-0. 47

[182] Kezhong Liu, Xinping Yan, and Fuping Hu. A modified DV-hop localization algorithm for wireless sensor networks. In *Proc. of IEEE Int'l Conf. on Intelligent Computing and Intelligent Systems (ICIS 2009)*, pages 511–514, 2009. ISBN 978-1-4244-4738-1/09. 47

[183] Sha Liu, Kai-Wei Fan, and Prasun Sinha. CMAC: an energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. In *Proc. of the 4th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007)*, pages 11–20, 2007. doi: 10.1109/SAHCN.2007.4292813. 12, 13, 14, 15, 171

[184] Jaime Lloret, Miguel Garcia, Diana Bri, and Sandra Sendra. A wireless sensor network deployment for rural and forest fire detection and verification. *Sensors*, 9:8722–8747, 2009. ISSN 1424-8220. doi: 10.3390/s91108722. 10

[185] David J. Lohman, David Bickford, and Navjot S. Sodhi. The burning issue. *Science*, 316 (5823):376–376, April 2007. ISSN 1095-9203. doi: 10.1126/science.1140278. PMID: 17446376. 22

[186] Carlos A. Lozano and Oscar J. Rodriguez. Design of forest fire early detection system using wireless sensor networks. *Online Journal on Electronics and Electrical Engineering*, 3(2):402–405, 2011. URL `http://infomesr.org/attachments/28-097.pdf`. Available online (last accessed: 25 June 2013). 10, 11, 21, 22

[187] Guang Lu and Wei Xue. Adaptive weighted fusion algorithm for monitoring system of forest fire based on wireless sensor networks. In *Proc. of the 2nd Int'l Conf. on Computer Modeling and Simulation (ICCMS 2010)*, 2010. ISBN 978-0-7695-3941-6. doi: 10.1109/ICCMS.2010.274. 11

[188] Rajagopal Maheswar and Raman Jayaparvathy. Performance analysis of cluster based sensor networks using N-policy M/G/1 queueing model. *Eur. J. Scientific Research*, 58(2):177–188, August 2011. ISSN 1450-216X. URL `http://www.europeanjournalofscientificresearch.com/ISSUES/EJSR_58_2_05.pdf`. Available online (last accessed: 13 March 2013). 98

[189] Stefan W. Maier, Jeremy Russell-Smith, Andrew C. Edwards, and Cameron Yates. Sensitivity of the MODIS fire detection algorithm (MOD14) in the savanna region of the Northern Territory, Australia. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:11–16, 2013. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2012.11.005. 27

[190] M. Yasir Malik. An outline of security in wireless sensor networks: Threats, countermeasures and implementations. In Noor Zaman, Khaled Ragab, and Azween Bin Abdullah, editors, *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, pages 24.01–24.20. IGI Global, 2012. ISBN 978-1-4666-0101-7. doi: 10.4018/978-1-4666-0101-7.ch024. 12, 85

[191] J. Ramiro Martinez de Dios, Luis Merino, and Aníbal Ollero. Fire detection using autonomous aerial vehicles with infrared and visual cameras. In *Proc. of the 16th World Congress of the Int'l Federation of Automatic Control (IFAC 2005)*, 2005. doi: 10.3182/20050703-6-CZ-1902.01380. 29

[192] Stuart Matthews, Andrew Sullivan, Jim Gould, Richard Hurley, Peter Ellis, and John Larmour. Evaluation of three fire detection systems. Technical report CSE-BDA-002, Bushfire Cooperative Research Centre, 2010. URL `http://www.emknowledge.gov.au/resource/?id=3119`. Available online (last accessed: 25 June 2013). 28

[193] Stuart Matthews, Andrew Sullivan, Jim Gould, Richard Hurley, Peter Ellis, and John Larmour. Field evaluation of two image-based wildland fire detection systems. *Fire Safety Journal*, 47(1):54–61, 2012. ISSN 0379-7112. doi: 10.1016/j.firesaf.2011.11.001. 28

[194] Maxim Integrated. DS2401—Silicon Serial Number. Data sheet (Rev. 3), 2011. URL `http://datasheets.maximintegrated.com/en/ds/DS2401.pdf`. Available online (last accessed: 24 June 2013). 48

[195] Maxim Integrated. DS2411—Silicon Serial Number with V_{CC} Input. Data sheet (Rev. 3), 2011. URL `http://datasheets.maximintegrated.com/en/ds/DS2411.pdf`. Available online (last accessed: 24 June 2013). 48

[196] Paul McFarlane, Grant Pearce, and John Moore. Introduction: Forestry and risk management—New Zealand in a global context. In Michel Arbez, Yves Birot, and Jean-Michel Camus, editors, *Proc. of the 8th Annual EFI Conf. on Risk Management and Sustainable Forestry*, volume 45 of *EFI Proceedings*, pages 5–19. European Forest Institute, 2002. ISBN 952-9844-93-X. 21, 22

[197] Michael P. McLaughlin. A compendium of common probability distributions. Regress+ Appendix A (Version 2.3), 2001. URL `http://www.causascientia.org/math_stat/Dists/Compendium.html`. Available online (last accessed: 27 Oct. 2012). 121

[198] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili. The state of the art in cross-layer design for wireless sensor networks. In *Proc. of the 2nd Int'l Conf. on Wireless Systems and Network Architectures in Next Generation Internet (EURO-NGI 2005)*, volume 3883 of *Lecture Notes in Computer Science (LNCS)*, pages 78–92, 2006. ISBN 978-3-540-34025-6. doi: 10.1007/11750673_7. 3, 4, 7, 12, 13, 14, 15, 16, 156, 169, 170

[199] Lucas D. P. Mendes and Joel J. P. C. Rodrigues. A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications*, 34(2): 523–534, March 2011. ISSN 1084-8045. doi: 10.1016/j.jnca.2010.11.009. 12

[200] Luis Merino, Fernando Caballero, J. Ramiro Martinez de Dios, Ivan Maza, and Aníbal Ollero. Automatic forest fire monitoring and measurement using unmanned aerial vehicles. In *Proc. of the 6th Int'l Conf. on Forest Fire Research (ICFFR 2010)*, 2010. URL `http://www.upo.es/isa/lmercab/publications/b2hd-icffr10monit.html`. Available online (last accessed: 25 June 2013). 29

[201] Luis Merino, Fernando Caballero, J. Ramiro Martinez de Dios, Iván Maza, and Aníbal Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent and Robotic Systems*, 65:533–548, 2012. ISSN 0921-0296. doi: 10.1007/s10846-011-9560-x. 29

[202] Peter W. Merlin. *Ikhana Unmanned Aircraft System : Western States fire missions*, volume 44 of *Monographs In Aerospace History*. National Aeronautics and Space Administration (NASA), 2009. URL `http://www.nasa.gov/pdf/470839main_ikhana_monograph.pdf`. SP-2009-4544. Available online (last accessed: 25 June 2013). 29

[203] Geoff V. Merrett and Yen Kheng Tan, editors. *Wireless Sensor Networks: Application-Centric Design*. InTech—Open Access Publisher, 2010. doi: 10.5772/658. 12

[204] Kurt Meyberg and Peter Vachenauer. *Höhere Mathematik 1*, volume 6 of *Springer-Lehrbuch*. Springer Verlag, 2001. ISBN 978-3-540-41850-4. 195

[205] Jerry Miller, Kirk Borne, Brian Thomas, Huang Zhenping, and Yuechen Chi. Automated wildfire detection through artificial neural networks. In *Proc. of the 2005 EastFIRE Conf. (EastFIRE 2005)*, 2005. URL `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050180456_2005177531.pdf`. Available online (last accessed: 25 June 2013). 27

[206] Rex Min and Anantha Chandrakasan. Energy-efficient communication for ad-hoc wireless sensor networks. In *Proc. of the 35th Asilomar Conf. on Signals, Systems and Computers (SS&C 2001)*, volume 1, pages 139–143, 2001. doi: 10.1109/ACSSC.2001.986894. 13, 14, 170

[207] Sam Mishra. *Strategic Case Analysis*. Franteractive Incorporated, 2007. ISBN 978-0-9798354-0-7. 127

[208] Luciana Moreira Sá de Souza, Harald Vogt, and Michael Beigl. A survey on fault tolerance in wireless sensor networks. Internal report 22, Fakultät für Informatik, Universität Karlsruhe, 2007. URL `http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/11824`. Available online (last accessed: 25 June 2013). 12

[209] Moteiv Corporation. Telos—Ultra low power IEEE 802.15.4 compliant wireless sensor module. Data sheet, 2004. URL `http://moss.csc.ncsu.edu/~mueller/rt/rt11/readings/projects/g4/datasheet.pdf`. Revision B. Available online (last accessed: 25 June 2013). 48

[210] Moteiv Corporation. tmote sky—Ultra low power IEEE 802.15.4 compliant wireless sensor module. Data sheet, 2006. URL `http://www.snm.ethz.ch/snmwiki/pub/uploads/Projects/tmote_sky_datasheet.pdf`. Available online (last accessed: 25 June 2013). 48

[211] Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. A survey on jamming attacks and countermeasures in WSNs. *IEEE Communications Surveys and Tutorials*, 11(4):42–56, 2009. ISSN 1553-877X. doi: 10.1109/SURV.2009.090404. 12

[212] Raymond Mulligan and Habib M. Ammari. Coverage in wireless sensor networks: A survey. *Network Protocols and Algorithms*, 2(2):27–53, 2010. ISSN 1943-3581. URL `http://www.macrothink.org/journal/index.php/npa/article/view/276`. Available online (last accessed: 25 June 2013). 12, 39

[213] P. Muralidhar and Chunduri B. Rama Rao. Reconfigurable wireless sensor network node based on NIOS core. In *Proc. of the 4th Int'l Conf. on Wireless Communication and Sensor Networks (WCSN 2008)*, 2008. ISBN 978-1-4244-3328-5. doi: 10.1109/WCSN.2008.4772684. 10, 11

[214] Radhika Nagpal. Organizing a global coordinate system from local information on an amorphous computer. A.I. Memo AIM-1666, MIT Computer Science and Artificial Intelligence Laboratory, December 1999. URL `http://dspace.mit.edu/bitstream/handle/1721.1/5926/AIM-1666.pdf`. Available online (last accessed: 25 June 2013). 46

[215] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proc. of the 2nd Int'l Conf. on Information Processing in Sensor Networks (IPSN 2003)*, pages 333–348, 2003. ISBN 3-540-02111-6. 46, 47

[216] Björn Neubauer, Johan Sidén, Christopher Olofsson, Mikael Gulliksson, Andrei Koptyug, Hans-Erik Nilsson, and Magnus Norgren. A new thermally activated battery cell-based forest fire detection and monitoring system. In Carlos A. Brebbia and Giovanni Perona, editors, *Modelling, Monitoring and Management of Forest Fires III*, volume 158 of *WIT Transactions on Ecology and the Environment*. WIT Press Publishing, 2012. ISBN 978-1-84564-585-4. doi: 10.2495/FIVA120101. 10, 22

[217] Jian Ni and S. A. G. Chandler. Connectivity properties of a random radio network. *IEE Proceedings Communications*, 141(4):289–296, 1994. ISSN 1350-2425. doi: 10.1049/ip-com:19941197. 41

[218] Joseph D. Nichols. Firefly system concept. In Bjorn F. Andresen, Marija Scholl, and Irving J. Spiro, editors, *Proc. of the 17th Conf. on Infrared Technology and Applications (Infrared Technology XVII)*, volume 1540 of *SPIE Proceedings*, pages 202–206, 1991. doi: 10.1117/12.48731. 29

[219] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *Proc. of IEEE Global Telecommunications Conf. (GLOBECOM 2001)*, pages 2926–2931, 2001. ISBN 0-7803-7206-9. doi: 10.1109/GLOCOM.2001.965964. 46

[220] Yanchao Niu, Sidong Zhang, Xiaoyu Xu, Hongwei Huo, and Shuai Gao. An enhanced DV-hop localization algorithm for irregularly shaped sensor networks. In *Proc. of the 3rd Int'l Conf. on Mobile Ad-Hoc and Sensor Networks (MSN 2007)*, volume 4864 of *Lecture Notes in Computer Science (LNCS)*, pages 694–704, 2007. ISBN 978-3-540-77023-7. doi: 10.1007/978-3-540-77024-4_63. 47

[221] Luís M. Oliveira and Joel J. P. C. Rodrigues. Wireless sensor networks: A survey on environmental monitoring. *Journal of Communications*, 6(2):143–151, April 2011. ISSN 1796-2021. doi: 10.4304/jcm.6.2.143-151. 1, 12

[222] Guillermo Owen. *Game Theory*. Emerald Group Publishing Ltd., 3rd edition, 1995. ISBN 978-0-12-531151-9. 127

[223] Wooguil Pak, Jin-Ghoo Choi, and Saewoong Bahk. Tier based anycast to achieve maximum lifetime by duty cycle control in wireless sensor networks. In *Proc. of the Int'l Wireless Communications and Mobile Computing Conference (IWCMC 2008)*, pages 123–128, 2008. doi: 10.1109/IWCMC.2008.22. 12, 13, 14, 171

[224] Amitangshu Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms*, 2(1):45–74, 2010. ISSN 1943-3581. doi: 10.5296/npa.v2i1.279. 45, 46

[225] Santosh Pandey and Prathima Agrawal. A survey on localization techniques for wireless networks. *Journal of the Chinese Institute of Engineers*, 29(7):1125–1148, 2006. ISSN 0253-3839. URL `http://www.dissertations.se/dissertation/34ee64ee6d/`. Available online (last accessed: 25 June 2013). 12, 45

[226] Pangun Park. *Modeling, Analysis, and Design of Wireless Sensor Network Protocols*. Phd thesis, KTH School of Electrical Engineering, Stockholm, Sweden, 2011. 12

[227] Enric Pastor, Marc Sole, Juan Lopez, Pablo Royo, and Cristina Barrado. Helicopter-based wildfire monitoring system software architecture. In *Proc. of the IEEE Aerospace Conf. (IEEEAC 2010)*, 2010. ISBN 978-1-4244-3888-4. doi: 10.1109/AERO.2010.5446807. 29

[228] Steffen T. Patzelt. *Waldbrandprognose und Waldbrandbekämpfung in Deutschland – zukunftsorientierte Strategien und Konzepte unter besonderer Berücksichtigung der Brandbekämpfung aus der Luft*. Phd thesis, Johannes Gutenberg-Universität, Mainz, 2008. URL `http://d-nb.info/988951649/34`. Available online (last accessed: 25 June 2013). 29

[229] Olga Pechony and Drew T. Shindell. Driving forces of global wildfires over the past millennium and the forthcoming century. *Proceedings of the National Academy of Sciences of the U.S.A. (PNAS)*, 107(45):19167–19170, 2010. doi: 10.1073/pnas.1003669107. 22

[230] Lina M. Pestana Leão de Brito and Laura M. Rodríguez Peralta. An analysis of localization problems and solutions in wireless sensor networks. *Tékhne - Revista de Estudos Politécnicos*, 6(9):1–27, 2008. ISSN 1645-9911. 45

[231] David Peterson and Jun Wang. A sub-pixel-based calculation of fire radiative power from MODIS observations: 2. Sensitivity analysis and potential fire weather application. *Remote Sensing of Environment*, 129:231–249, 2013. ISSN 0034-4257. doi: 10.1016/j.rse.2012.10.020. 27

[232] David Peterson, Edward Hyer, and Jun Wang. A short-term predictor of satellite-observed fire activity in the North American boreal forest: Toward improving the prediction of smoke emissions. *Atmospheric Environment*, (to appear):N/A, 2013. doi: 10.1016/j.atmosenv.2013.01.052. 27

[233] Vidyasagar Potdar, Atif Sharif, and Elizabeth Chang. Wireless sensor networks: A survey. In *Proc. of the Int'l Conf. on Advanced Information Networking and Applications Workshops (WAINA 2009)*, pages 636–641. IEEE, May 2009. ISBN 978-1-4244-3999-7. doi: 10.1109/WAINA.2009.192. 12

[234] Owen F. Price, Rittick Borah, and Stefan W. Maier. Role of weather and fuel in stopping fire spread in tropical savannas. *Austral Ecology*, (in press):1–10, 2013. doi: 10.1111/aec.12021. 22

[235] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proc. of the 6th ACM Int'l Conf. on Mobile Computing and Networking (MobiCom 2000)*, 2000. doi: 10.1145/345910.345917. 45

[236] Lawrence F. Radke, Terry L. Clark, Janice L. Coen, Craig A. Walther, Robert N. Lockwood, Phillip J. Riggan, James A. Brass, and Robert G. Higgins. The wildfire experiment (WIFE): observations with airborne remote sensors. *Canadian Journal of Remote Sensing*, 26(5):406–417, 2000. URL `http://nldr.library.ucar.edu/repository/collections/OSGC-000-000-008-302`. Available online (last accessed: 25 June 2013). 29

[237] Cauligi S. Raghavendra, Krishna M. Sivalingam, and Taieb Znati, editors. *Wireless Sensor Networks*. Kluwer Academic Publishers, 2004. 12

[238] Veerabhadran Ramanathan and Gregory R. Carmichael. Global and regional climate changes due to black carbon. *Nature Geoscience*, 1:221–227, April 2008. doi: 10.1038/ngeo156. 22

[239] Prakash Ranganathan and Kendall Nygard. Time synchronization in wireless sensor networks: A survey. *Int'l Journal of Ubiquitous Computing*, 1(2):92–102, April 2010. ISSN 0975-8992. doi: 10.5121/iju.2010.1206. 12, 50

[240] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. Geographic routing without location information. In *Proc. of the 9th Annual Int'l Conf. on Mobile Computing and Networking (MobiCom 2003)*, pages 96–108. ACM, 2003. ISBN 1-58113-753-2. doi: 10.1145/938985.938996. 53

[241] Yrjö Rauste. Forest fire detection with satellites for fire control. *Int'l Archives of Photogrammetry and Remote Sensing*, XXXI(Part 7B):584–588, 1996. URL `http://www.isprs.org/proceedings/XXXI/congress/part7/584_XXXI-part7.pdf`. Available online (last accessed: 25 June 2013). 27

[242] Agoston Restas. Wildfire management supported by UAV based air reconnaissance: Experiments and results at the Szendro fire department, Hungary. In *Proc. of the 1st Int'l Workshop on Fire Management*, 2006. URL `http://www.fire.uni-freiburg.de/GlobalNetworks/Caribbean/SIMFOR2006/Restas_hu_1.pdf`. Available online (last accessed: 25 June 2013). 29

[243] RF Monolithics Incorporated. TR1000—916.50 MHz hybrid transceiver. Data sheet, April 2008. URL `http://www.rfm.com/products/data/tr1000.pdf`. Available online (last accessed: 25 June 2013). 98

[244] Ill-Keun Rhee, Jaehan Lee, Jangsub Kim, Erchin Serpedin, and Yik-Chung Wu. Clock synchronization in wireless sensor networks: An overview. *Sensors*, 9(1):56–85, 2009. ISSN 1424-8220. doi: 10.3390/s90100056. 50

[245] Jennifer M. Robinson. Fire from space: Global fire evaluation using infrared remote sensing. *International Journal on Remote Sensing*, 12(1):3–24, 1991. doi: 10.1080/01431169108929628. 27

[246] Janos Roszik. *Retrial Queues and their Application in Performance Modelling of Communication Networks*. Phd thesis, University of Debrecen, 2006. URL `http://ganymedes.lib.unideb.hu:8080/dea/bitstream/2437/91167/4/ertekezes_angol.pdf`. Available online (last accessed: 25 June 2013). 19

[247] Janos Roszik and Janos Sztrik. Performance analysis of finite-source retrial queues with non-reliable heterogeneous servers. In *XXIV-th Seminar on Stability Problems of Stochastic Models*, Jurmala, Latvia, 2004. URL `http://irh.inf.unideb.hu/~jsztrik/publications/papers/Stabil04szjrev.pdf`. Available online (last accessed: 25 June 2013). 19

[248] Jörg Roth. *Mobile Computing – Grundlagen, Technik, Konzepte*. dpunkt.verlag, 2005. ISBN 978-3-89864-366-5. 45

[249] Richard C. Rothermel. A mathematical model for predicting fire spread in wildland fuels. Research paper INT-115, USDA Forest Service, Intermountain Forest and Range Experiment Station, 1972. URL `http://www.fs.fed.us/rm/pubs_int/int_rp115.pdf`. Available online (last accessed: 25 June 2013). 23

[250] Shad Roundy and Luc Frechette. Energy scavenging and nontraditional power sources for wireless sensor networks. In Ivan Stojmenovic, editor, *Handbook of Sensor Networks—Algorithms and Architectures*, pages 75–105. John Wiley & Sons, 2005. ISBN 978-0-471-68472-5. 12, 41, 42

[251] Hosam Rowaihy, Sharanya Eswaran, Matthew Johnson, Dinesh Verma, Amotz Bar-Noy, Theodore Brown, and Thomas La Porta. A survey of sensor selection schemes in wireless sensor networks. In *Proc. of the 9th SPIE Defense and Security Symposium Conf. on Unattended Ground, Sea, and Air Sensor Technologies and Applications*, 2007. doi: 10.1117/12.723514. 12

[252] Partha S. Roy. Forest fire and degradation assessment using satellite remote sensing and geographic information system. In *Proc. of the Training Workshop on Satellite remote sensing and GIS applications in agricultural meteorology*, 2004. URL `http://www.preventionweb.net/english/professional/publications/v.php?id=1682`. Available online (last accessed: 25 June 2013). 27

[253] Gernot Ruecker, Eckehard Lorenz, Anja A. Hoffmann, Dieter Oertel, Joachim Tiemann, and Winfried Halle. Upcoming and prospective fire monitoring missions based on the heritage of the BIRD (bi-spectral infrared detection) satellite. In *Proc. of the IEEE Int'l Geoscience and Remote Sensing Symposium (IGARSS 2012)*, pages 225–228, 2012. doi: 10.1109/IGARSS.2012.6351597. 27

[254] Steven W. Running. Is global warming causing more, larger wildfires? *Science*, 313 (5789):927–928, 2006. doi: 10.1126/science.1130370. 22

[255] Antti Saaksvuori and Anselmi Immonen. *Product Lifecycle Management*. Springer Verlag, 3rd edition, 2008. ISBN 978-3-540-78172-1. doi: 10.1007/978-3-540-78172-1. 155

[256] Yasar G. Sahin. Animals as mobile biological sensors for forest fire detection. *Sensors*, 7:3084–3099, 2007. ISSN 1424-8220. doi: 10.3390/s7123084. 23, 29, 30

[257] Yasar G. Sahin and Turker Ince. Early forest fire detection using radio-acoustic sounding system. *Sensors*, 9:1485–1498, 2009. ISSN 1424-8220. doi: 10.3390/s90301485. 28

[258] Chris Savarese, Jan M. Rabaey, and Jan Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *Proc. of the IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, volume 4, pages 2037–2040, 2001. doi: 10.1109/ICASSP.2001.940391. 45, 53

[259] Chris Savarese, Jan M. Rabaey, and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proc. of the USENIX Annual Technical Conf. (USENIX 2002)*, pages 317–327, 2002. ISBN 1-880446-00-6. URL `http://dl.acm.org/citation.cfm?id=647057.713854`. Available online (last accessed: 25 June 2013). 47, 53

[260] Gunnar Schaefer, François Ingelrest, and Martin Vetterli. Potentials of opportunistic routing in energy-constrained wireless sensor networks. In Utz Roedig and Cormac J. Sreenan, editors, *Proc. of the 6th European Conf. on Wireless Sensor Networks (EWSN 2009)*, volume 5432 of *Lecture Notes in Computer Science (LNCS)*, pages 118–133. Springer Verlag, January 2009. ISBN 978-3-642-00223-6. doi: 10.1007/978-3-642-00224-3_8. 12, 13, 14, 15, 171

[261] Thomas Schmid, Jonathan Friedman, Zainul Charbiwala, Young H. Cho, and Mani B. Srivastava. XCXO: An ultra-low cost ultra-high accuracy clock system for wireless sensor networks in harsh remote outdoor environments. Technical report TR-UCLA-NESL-200802-02, Networked & Embedded Systems Laboratory, University of California, Los Angeles, February 2008. URL `http://nesl.ee.ucla.edu/document/show/258`. Available online (last accessed: 16 Jan. 2013). 49

[262] Daniel Schmidt, Matthias Berning, and Norbert Wehn. Error correction in single-hop wireless sensor networks—A case study. In *Proc. of the Conf. and Exhibition on Design, Automation and Test (DATE 2009)*, pages 1296–1301, April 2009. doi: 10.1109/DATE.2009.5090865. 48

[263] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, April 2009. ISSN 1095-9203. doi: 10.1126/science.1165893. 5, 125

[264] Guido Schmuck, Jesús San-Miguel-Ayanz, Andrea Camia, Tracy Durrant, Sandra Santos de Oliveira, Roberto Boca, Ceri Whitmore, Cristiano Giovando, Giorgio Libertà, Paolo Corti, and Ernst Schulte. Forest fires in Europe 2010. Technical report JRC

66167, EUR 24910 EN, European Commission, Joint Research Centre, Institute for Environment and Sustainability, 2011. 22, 25, 27

[265] Guido Schmuck, Jesús San-Miguel-Ayanz, Andrea Camia, Tracy Durrant, Roberto Boca, Ceri Whitmore, Giorgio Libertà, Paolo Corti, and Ernst Schulte. Forest fires in Europe, Middle East and North Africa 2011. Technical report JRC 74152, EUR 25483 EN, European Commission, Joint Research Centre, Institute for Environment and Sustainability, 2012. 22, 23, 25

[266] Anthony Schoofs, Gregory M. P. O'Hare, and Antonio G. Ruzzelli. Debugging low-power and lossy wireless networks: A survey. *IEEE Communications Surveys and Tutorials*, 14(2):311–321, 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.021111.00098. 12

[267] Dave Schroeder. Evaluation of three wildfire smoke detection systems. *Advantage*, 5 (24):1–8, 2004. ISSN 1493-3381. 28

[268] Malte Schultjan. Towards the deployment of UAVs for fire surveillance. Master's thesis, Hamburg University of Technology, 2012. URL `http://www.sts.tu-harburg.de/pw-and-m-theses/2012/schult12.pdf`. Available online (last accessed: 25 June 2013). 29

[269] Martin G. Schultz, Angelika Heil, Judith J. Hoelzemann, Allan Spessa, Kirsten Thonicke, Johann G. Goldammer, Alexander C. Held, Jose M. C. Pereira, and Maarten van het Bolscher. Global wildland fire emissions from 1960 to 2000. *Global Biogeochemical Cycles*, 22:1–17, 2008. doi: 10.1029/2007GB003031. 22, 27

[270] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978. doi: doi:10.1214/aos/1176344136. 123

[271] Joe H. Scott and Elizabeth D. Reinhardt. Assessing crown fire potential by linking models of surface and crown fire behavior. Research paper RMRS-RP-29, USDA Forest Service, Rocky Mountain Research Station, 2001. URL `http://www.fire.org/downloads/nexus/2.0.0/rp29.pdf`. Available online (last accessed: 25 June 2013). 23

[272] Adolph Seema and Martin Reisslein. Towards efficient wireless video sensor networks: A survey of existing node architectures and proposal for a flexi-WVSNP design. *IEEE Communications Surveys and Tutorials*, 13(3):462–486, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.102910.00098. 12

[273] Jaydip Sen. A survey on wireless sensor network security. *Int'l Journal of Communication Networks and Information Security*, 1(2):55–78, 2009. URL `http://arxiv.org/abs/1011.1529`. Available online (last accessed: 25 June 2013). 12, 85

[274] Ljiljana Seric, Darko Stipanicev, and Maja Stula. Agent based sensor and data fusion in forest fire observer. In Nada Milisavljevic, editor, *Sensor and Data Fusion*, pages 365–378. InTech—Open Access Publisher, 2009. doi: 10.5772/6587. 25, 26

[275] Michael Serra. *Discovering Geometry: An Investigative Approach*. Key Curriculum Press, 4th edition, 2008. ISBN 978-1-55953-882-4. 72

[276] Kenneth C. Sevcik and Isi Mitrani. The distribution of queuing network states at input and output instants. *Journal of the Association for Computing Machinery*, 28(2):358–371, 1981. doi: 0.1145/322248.322257. 105

[277] Perwez Shahabuddin. Rare event simulation. In Boudewijn R. Haverkort, Raymond Marie, Gerardo Rubino, and Kishor Trivedi, editors, *Performability Modeling – Techniques and Tools*, pages 163–178. John Wiley & Sons, 2001. ISBN 0-471-49195-0. 120

[278] Yi Shang and Wheeler Ruml. Improved MDS-based localization. In *Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2004)*, volume 4, pages 2640–2651, 2004. ISBN 0-7803-8355-9. doi: 10.1109/INFCOM.2004.1354683. 46

[279] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz. Localization from mere connectivity. In *Proc. of the 4th ACM Int'l Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2003)*, pages 201–212, 2003. ISBN 1-58113-684-6. doi: 10.1145/778415.778439. 46

[280] Jang-Ping Sheu, Yu-Chia Chang, and Gang-Hua Song. Logical coordinate assignment for geographic routing in wireless sensor networks. *Int'l Journal of Pervasive Computing and Communications*, 3(3):274–288, 2008. ISSN 1742-7371. doi: 10.1108/17427370710856237. 53

[281] Jian Shu, Linlan Liu, Yubin Chen, and Haitao Hu. A novel three-dimensional localization algorithm in wireless sensor networks. In *Proc. of the 4th Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCOM 2008)*, 2008. ISBN 978-1-4244-2108-4. doi: 10.1109/WiCom.2008.945. 47

[282] Fikret Sivrikaya and Bülent Yener. Time synchronization in sensor networks: A survey. *IEEE Network Magazine*, 18(4):45–50, 2004. ISSN 0890-8044. doi: 10.1109/MNET.2004.1316761. 12, 50

[283] Primoz Skraba, Hamid Aghajan, and Ahmad Bahai. Cross-layer optimization for high density sensor networks: Distributed passive routing decisions. In Ioanis Nikolaidis, Michel Barbeau, and Evangelos Kranakis, editors, *Proc. of the 3rd Int'l Conf. on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW 2004)*, volume 3158 of *Lecture Notes in Computer Science (LNCS)*, pages 266–279. Springer Verlag, January 2004. ISBN 978-3-540-28634-9. doi: 10.1007/978-3-540-28634-9_21. 12, 13, 14, 16, 170

[284] Hamdy Soliman, Komal Sudan, and Ashish Mishra. A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks. In *Proc. of the 9th IEEE Conf. on Sensors (SENSORS 2010)*, 2010. ISBN 978-1-4244-8168-2. doi: 10.1109/ICSENS.2010.5690033. 10

[285] Byungrak Son, Yong-Sork Her, and Jung-Gyu Kim. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. *Int'l Journal of Computer Science and Network Security*, 6(9B):124–130, 2006. ISSN 1738-7906. 11

[286] Athanasios Spyridakos, Yannis Psaromiligkos, Lazaros Vryzidis, Maria Litsardaki, and Ioannis Salmon. Survey forest fire detection systems via multicriteria disaggregation-aggregation approach: The case of Lycabettus Hill. *Int'l Journal of Multicriteria Decision Making*, 2(1):29–46, 2012. doi: 10.1504/IJMCDM.2012.045081. 28

[287] Avinash Srinivasan and Jie Wu. A survey on secure localization in wireless sensor networks. In Borko Furht, editor, *Encyclopedia of Wireless and Mobile Communications*. Taylor and Francis Group, 2007. ISBN 1-4200-5562-3. doi: 10.1081/E-EWMC-120043801. 12, 45

[288] Martin Stehlík. Comparison of simulators for wireless sensor networks. Master's thesis, Masaryk University, Brno, Czech Republic, 2011. URL `http://is.muni.cz/th/172616/fi_m/master_thesis.pdf`. Available online (last accessed: 25 June 2013). 152

[289] Dan Steingart. Power sources for wireless sensor networks. In Shashank Priya and Daniel J. Inman, editors, *Energy Harvesting Technologies*, pages 267–286. Springer Verlag, 2009. ISBN 978-0-387-76463-4. doi: 10.1007/978-0-387-76464-1_9. 12, 41

[290] Darko Stipanicev, Tomislav Vuko, Damir Krstinic, Maja Stula, and Ljiljana Bodrozic. Forest fire protection by advanced video detection system—Croatian experiences. In *Proc. of the 3rd TIEMS Workshop on Improvement of Disaster Management System*, 2006. URL `http://bib.irb.hr/datoteka/279548.TIEMS_-_Stipanicev_i_ostali.pdf`. Available online (last accessed: 25 June 2013). 25, 26, 28

[291] Darko Stipanicev, Maja Stula, Damir Krstinic, Ljiljana Seric, Toni Jakovcevic, and Marin Bugaric. Advanced automatic wildfire surveillance and monitoring network. In *Proc. of the 6th Int'l Conf. on Forest Fire Research*, 2010. URL `http://laris.fesb.hr/PDF/Advancedautomaticwildfiresurveillanceandmonitoringnetwork.pdf`. Available online (last accessed: 25 June 2013). 25, 26, 27, 28

[292] Mile K. Stojcev, Mirko R. Kosanovic, and Ljubisa R. Golubovic. Power management and energy harvesting techniques for wireless sensor nodes. In *Proc. of the 9th Int'l Conf. on Telecommunication in Modern Satellite, Cable, and Broadcasting Services (TELSIKS 2009)*, pages 65–72, 2009. ISBN 978-1-4244-4383-3. 43

[293] Ivan Stojmenovic, editor. *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, 2005. ISBN 978-0-471-68472-5. 12

[294] Sujesha Sudevalayam and Purushottam Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys and Tutorials*, 13(3):443–461, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.060710.00094. 12, 41

[295] Ruandha Agung Sugardiman. *Spaceborne Radar Monitoring of Forest Fires and Forest Cover Change—A case study in Kalimantan*. Phd thesis, Wageningen University, The Netherlands, 2007. URL `http://library.wur.nl/WebQuery/wda/1839900`. Available online (last accessed: 25 June 2013). 23, 26

[296] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks*, 3(3):281–323, 2005. ISSN 1570-8705. doi: 10.1016/j.adhoc.2005.01.002. 12, 50

[297] Bjorn Sundt and David C. M. Dickson. Comparison of methods for evaluation of the n-fold convolution of an arithmetic distribution. Research paper 65, University of Melbourne, Centre for Actuarial Studies, August 1998. URL `http://www.economics.unimelb.edu.au/ACT/html/no65.pdf`. Available online (last accessed: 25 June 2013). 196

[298] Bjorn Sundt and Raluca Vernic. *Recursions for Convolutions and Compound Distributions with Insurance Applications*. Lecture Notes of the European Actuarial Academy GmbH (EAA Series). Springer Verlag, 2009. ISBN 978-3-540-92899-7. 196

[299] Sandor Szilvasi, Peter Volgyesi, Janos Sallai, Akos Ledeczi, and Miklos Maroti. Interferometry in wireless sensor networks. In Ivan Padron, editor, *Interferometry – Research and Applications in Science and Technology*, pages 437–462. InTech—Open Access Publisher, 2012. ISBN 978-953-51-0403-2. 46

[300] Janos Sztrik and Janos Roszik. Finite-source retrial queueing systems with heterogeneous non-reliable servers and different service policies. In *Proc. of the 26th Hungarian Operational Research Conf.*, Gyor, Hungary, 2004. URL `http://irh.inf.unideb.hu/~jsztrik/publications/papers/retrial-hom-ms-diffservOpKut04.pdf`. Available online (last accessed: 25 June 2013). 19

[301] János Sztrik and János Roszik. Performance analysis of finite-source retrial queueing systems with heterogeneous non-reliable servers and different service policies. Technical report 2004/6 (Preprints No. 316), Institute of Informatics, University of Debrecen, Hungary, 2004. URL `http://irh.inf.unideb.hu/~jsztrik/publications/resrep/tr-retrial-hom-ms-diffserv.pdf`. Available online (last accessed: 25 June 2013). 19

[302] Xiaoyuan Ta, Guoqiang Mao, and Brian D. O. Anderson. On the connectivity properties of wireless multi-hop networks. In *Proc. of the 3rd Int'l Conf. on Communications and Networking in China (ChinaCom 2008)*, pages 959–963, 2008. ISBN 978-1-4244-2374-3. doi: 10.1109/CHINACOM.2008.4685183. 41

[303] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984. ISSN 0090-6778. doi: 10.1109/TCOM.1984.1096061. 41

[304] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, 1996. ISBN 978-0-13-349945-2. 12, 77

[305] Andrew S. Tanenbaum, Chandana Gamage, and Bruno Crispo. Taking sensor networks from the lab to the jungle. *IEEE Computer Magazine*, 39(8):98–100, 2006. ISSN 0018-9162. doi: 10.1109/MC.2006.280. 10

[306] Sumit Kumar Tetarave and Ashish Kumar Srivastava. FOREST GUARD: A complete safety for wildlife using mobile agents and sensor clouds in WSN. *Int'l Journal of Computer Science Issues*, 9(6):310–314, 2012. ISSN 1694-0814. 10, 11, 21

[307] Texas Instruments Incorporated. CC2420—2.4 GHz IEEE 802.15.4 / ZigBee-ready RF transceiver. Data sheet, November 2010. URL http://www.ti.com/lit/ds/symlink/cc2420.pdf. Available online (last accessed: 24 June 2013). 40, 98

[308] Shuang Tian, Xinming Zhang, Pengxi Liu, Peng Sun, and Xinguo Wang. A RSSI-based DV-hop algorithm for wireless sensor networks. In *Proc. of the Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pages 2555–2558, 2007. ISBN 1-4244-1312-5/07. doi: 10.1109/WICOM.2007.636. 47

[309] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(2):28–36, April 2002. ISSN 1559-1662. doi: 10.1145/565702.565708. 12

[310] Toshiba Corporation. Toshiba develops manufactures 19nm generation NAND flash memory with world's largest density and smallest die size. Press release, February 2012. URL http://www.toshiba-components.com/prpdf/6193A.pdf. Available online (last accessed: 16 Oct. 2012). 182

[311] B. Ugur Töreyin. *Fire Detection Algorithms Using Multimodal Signal and Image Analysis*. Phd thesis, Bilkent University, Ankara, Turkey, January 2009. URL http://theses.eurasip.org/theses/224/fire-detection-algorithms-using-multimodal-signal/. Available online (last accessed: 25 June 2013). 28

[312] B. Ugur Töreyin and A. Enis Cetin. Wildfire detection using LMS based active learning. In *Proc. of the IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pages 1461–1464, April 2009. doi: 10.1109/ICASSP.2009.4959870. 25, 28

[313] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, 2nd edition, November 2001. ISBN 0-471-33341-7. 109

[314] Vassileios Tsetsos, Odysseas Sekkas, George Tsoublekas, Stathes Hadjieythymiades, and Evangelos Zervas. A forest fire detection system: The Meleager approach. In *Proc. of the 16th Panhellenic Conf. on Informatics (PCI 2012)*, pages 320–326, 2012. doi: 10.1109/PCi.2012.62. 28

[315] Georgios Tsiourlis, Stamatis Andreadakis, and Pavlos Konstantinidis. SITHON: A wireless network of in situ optical cameras applied to the early detection-notification-monitoring of forest fires. *Sensors*, 9:4465–4482, 2009. ISSN 1424-8220. doi: 10.3390/s90604465. 28

[316] Md. Yusuf S. Uddin and Md. Mostofa Akbar. Addressing techniques in wireless sensor networks: A short survey. In *Proc. of the 4th Int'l Conf. on Electrical and Computer Engineering (ICECE 2006)*, pages 581–584, December 2006. ISBN 98432-3814-1. doi: 10.1109/ICECE.2006.355698. 12, 48

[317] Michael Van Hauwermeiren and David Vose. A compendium of distributions, 2009. URL http://www.vosesoftware.com/content/ebookmr4.pdf. Available online (last accessed: 27 Oct. 2012). 121

[318] Roberto Verdone, Davide Dardari, Gianluca Mazzini, and Andrea Conti. *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Academic Press, 2008. ISBN 978-0-12-372539-4. 10, 11, 12

[319] Ricardo Vérlez. Causes of forest fires in the mediterranean basin. In Michel Arbez, Yves Birot, and Jean-Michel Camus, editors, *Proc. of the 8th Annual EFI Conf. on Risk Management and Sustainable Forestry*, volume 45 of *EFI Proceedings*, pages 35–42. European Forest Institute, 2002. ISBN 952-9844-93-X. 21

[320] Mehmet C. Vuran and Ian F. Akyildiz. XLP: A cross layer protocol for efficient communication in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(11): 1578–1591, 2010. doi: 10.1109/TMC.2010.125. 4, 7, 12, 13, 14, 15, 16, 84, 151, 160, 169, 170, 172

[321] John P. Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. Wireless sensor network security: A survey. In Yang Xiao, editor, *Security in Distributed, Grid, Mobile, and Pervasive Computing*, pages 367–410. Auerbach Publications, 2007. ISBN 978-0-8493-7921-5. 12, 85

[322] Bang Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys*, 43(4):32:1–32:53, October 2011. ISSN 0360-0300. doi: 10.1145/1978802.1978811. 12, 39

[323] Chonggang Wang, Kazem Sohraby, Bo Li, Mahmoud Daneshmand, and Yueming Hu. A survey of transport protocols for wireless sensor networks. *IEEE Network Magazine*, 20(3):34–40, 2006. ISSN 0890-8044. doi: 10.1109/MNET.2006.1637930. 12

[324] Feng Wang and Jiangchuan Liu. Networked wireless sensor data collection: Issues, challenges, and approaches. *IEEE Communications Surveys and Tutorials*, 13(4):673–687, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.060710.00066. 12, 85

[325] Jing Wang, Ratan K. Ghosh, and Sajal K. Das. A survey on sensor localization. *Journal of Control Theory and Applications*, 8(1):2–11, 2010. doi: 10.1007/s11768-010-9187-7. 12, 45

[326] Shi-Dong Wang, L. L. Miao, and Guang-Xiong Peng. An improved algorithm for forest fire detection using HJ data. In *Proc. of the 18th Biennial ISEM Conf. on Ecological Modelling for Global Change and Coupled Human and Natural System*, volume 13 of *Procedia Environmental Sciences*, pages 140–150, 2012. doi: 10.1016/j.proenv.2012.01.014. 27

[327] Yun Wang, Xiaodong Wang, Demin Wang, and Dharma P. Agrawal. Range-free localization using expected hop progress in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(10):1540–1552, 2009. doi: 10.1109/TPDS.2008.239. 46

[328] Yunbo Wang. *Probabilistic QoS Analysis in Wireless Sensor Networks*. Phd thesis, University of Nebraska, Lincoln, USA, 2012. URL `http://digitalcommons.unl.edu/computerscidiss/40/`. Available online (last accessed: 25 June 2013). 12, 13, 14, 17, 18, 172

[329] Yunbo Wang, Mehmet C. Vuran, and Steve Goddard. Analysis of event detection delay in wireless sensor networks. In *Proc. of the 30th IEEE Int'l Conf. on Computer Communications (INFOCOM 2011)*, pages 1296–1304, 2011. doi: 10.1109/INFCOM.2011.5934912. 12, 13, 14, 17, 172

[330] Brett A. Warneke and Kristofer S. J. Pister. Exploring the limits of system integration with smart dust. In *Proc. of the 2002 ASME Int'l Mechanical Engineering Congress & Exposition (IMECE 2002)*, 2002. URL `http://old.shahed.ac.ir/references/[13]imece2002-34360.pdf`. Available online (last accessed: 25 June 2013). 40

[331] Adam C. Watts, Vincent G. Ambrosia, and Everett A. Hinkley. Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 4(6):1671–1692, 2012. ISSN 2072-4292. doi: 10.3390/rs4061671. 29

[332] Patrick Wüchner, Hermann de Meer, Jörg Barner, and Gunter Bolch. A brief introduction to MOSEL-2. In Reinhard German and Armin Heindl, editors, *Proc. of the 13th GI/ITG Conf. on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2006)*, pages 473–476. GI/ITG/MMB, University of Erlangen-Nürnberg, VDE Verlag, 2006. ISBN 978-3-8007-2945-6. URL `http://www.net.fim.uni-passau.de/pdf/Wuechner2006c.pdf`. Available online (last accessed: 25 June 2013). 115

[333] Patrick Wüchner, Hermann de Meer, Gunter Bolch, János Roszik, and János Sztrik. Modeling finite-source retrial queueing systems with unreliable heterogeneous servers and different service policies using MOSEL. In Khalid Al-Begain, Armin Heindl, and Miklos Telek, editors, *ASMTA 2007 Conference*, pages 75–80, Prague, Czech Republic, June 2007. URL `http://www.net.fim.uni-passau.de/pdf/Wuechner2007c.pdf`. Available online (last accessed: 25 June 2013). 19, 107, 108

[334] Patrick Wüchner, János Sztrik, and Hermann de Meer. Homogeneous finite-source retrial queues with search of customers from the orbit. In *Proc. of the 14th GI/ITG Conf. on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2008)*, Dortmund, Germany, March 2008. URL `http://www.net.fim.uni-passau.de/pdf/Wuechner2008c.pdf`. Available online (last accessed: 25 June 2013). 130

[335] Patrick Wüchner, János Sztrik, and Hermann de Meer. Finite-source M/M/S retrial queue with search for balking and impatient customers from the orbit. *Computer Networks*, 53(8):1264–1273, June 2009. doi: 10.1016/j.comnet.2009.02.015. 16, 104

[336] Patrick Wüchner, János Sztrik, and Hermann de Meer. Modeling wireless sensor networks using finite-source retrial queues with unreliable orbit. In *Proc. of the Workshop on Performance Evaluation of Computer and Communication Systems (PERFORM 2010)*, volume 6821 of *Lecture Notes in Computer Science (LNCS)*, pages 73–86. Springer Verlag, 2011. doi: 10.1007/978-3-642-25575-5_7. 16, 17, 18, 19

[337] Quanrui Wei, Jiuqiang Han, Dexing Zhong, and Ruiling Liu. An improved multihop distance estimation for DV-hop localization algorithm in wireless sensor networks. In

In. Proc. of the IEEE Vehicular Technology Conf. (VTC Fall 2012), 2012. ISBN 978-1-4673-1879-2. doi: 10.1109/VTCFall.2012.6399209. 47

[338] Aaron Weiner. *A Systems Level Characterization and Tradespace Evaluation of a Simulated Airborne Fourier Transform Infrared Spectrometer for Gas Detection*. Phd thesis, Rochester Institute of Technology, 2010. URL `https://ritdml.rit.edu/handle/1850/12708`. Available online (last accessed: 25 June 2013). 29

[339] Anthony L. Westerling, Hugo G. Hidalgo, Daniel R. Cayan, and Thomas W. Swetnam. Warming and earlier spring increase western u.s. forest wildfire activity. *Science*, 313 (5789):940–943, 2006. doi: 10.1126/science.1128834. 22

[340] Simon L. Willis. *Investigation into long-range wireless sensor networks*. Phd thesis, James Cook University, 2007. URL `http://eprints.jcu.edu.au/2034/`. Available online (last accessed: 25 June 2013). 12, 40

[341] Ronald W. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2):223–231, 1982. doi: 10.1287/opre.30.2.223. 105

[342] Sau Y. Wong, Joo G. Lim, S. V. Rao, and Winston K. G. Seah. Density-aware hop-count localization (DHL) in wireless sensor networks with variable density. In *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC 2005)*, volume 3, pages 1848–1853, 2005. ISBN 0-7803-8966-2. doi: 10.1109/WCNC.2005.1424793. 46

[343] Martin J. Wooster, Weidong Xu, and Timothy J. Nightingale. Sentinel-3 SLSTR active fire detection and FRP product: Pre-launch algorithm development and performance evaluation using MODIS and ASTER datasets. *Remote Sensing of Environment*, 120: 236–254, 2012. ISSN 0034-4257. doi: 10.1016/j.rse.2011.09.033. 27

[344] D. Bruce Wright, Tohru Yotsumata, and Naser El-Sheimy. Real time identification and location of forest fire hotspots from geo-referenced thermal images. In *Proc. of the 10th Congress of the Int'l Society for Programmetry and Remote Sensing (ISPRS 2004), Technical Commission II*, volume XXXV B2 of *ISPRS Archives*, pages 13–18, 2004. URL `http://www.isprs.org/proceedings/XXXV/congress/comm2/papers/90.pdf`. Available online (last accessed: 25 June 2013). 29

[345] Lingfei Wu, Max Q.-H. Meng, Zhenzhong Dong, and Huawei Liang. An empirical study of DV-hop localization algorithm in random sensor network. In *Proc. of the 2nd Int'l Conf. on Intelligent Computation Technology and Automation (ICICTA 2009)*, pages 41–44, 2009. ISBN 978-0-7695-3804-4/09. doi: 10.1109/ICICTA.2009.726. 47

[346] Qingjun Xiao, Bin Xiao, Jiannong Cao, and Jianping Wang. Multihop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme. *IEEE Transactions on Mobile Computing*, 9(11):1592–1607, 2010. doi: 10.1109/TMC.2010.129. 46

[347] Yuyan Xue, Mehmet C. Vuran, and Byrav Ramamurthy. Cost efficiency of anycast-based forwarding in duty-cycled WSNs with lossy channel. In *Proc. of the 7th Annual IEEE Communications Society Conf. on Sensor Mesh and Ad Hoc Communications and Networks (SECON 2010)*, pages 1–9, 2010. doi: 10.1109/SECON.2010.5508263. 12, 13, 14, 15, 172

[348] Huanjia Yang. *Integrated ZigBee RFID Sensor Networks for Resource Tracking and Monitoring in Logistics Management*. Phd thesis, Loughborough University, Leicestershire, UK, 2010. URL `https://dspace.lboro.ac.uk/2134/7295`. Available online (last accessed: 25 June 2013). 12

[349] Fujun Ye and Ruifang Pan. A survey of addressing algorithms for wireless sensor networks. In *Proc. of the 5th Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCom 2009)*, pages 1 –7, September 2009. ISBN 978-1-4244-3693-4. doi: 10.1109/WICOM.2009.5304082. 12, 48

[350] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, August 2008. ISSN 13891286. doi: 10.1016/j.comnet.2008.04.002. 12

[351] Maxwell Young and Raouf Boutaba. Overcoming adversaries in sensor networks: A survey of theoretical models and algorithmic approaches for tolerating malicious interference. *IEEE Communications Surveys and Tutorials*, 13(4):617–641, 2011. ISSN 1553-877X. doi: 10.1109/SURV.2011.041311.00156. 12

[352] Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network Magazine*, 20(3):20–25, 2006. ISSN 0890-8044. doi: 10.1109/MNET.2006.1637928. 12

[353] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Proc. of the Int'l Conf. on Wireless Communications, Networking and Mobile Computing (WiCOM 2005)*, 2005. ISBN 0-7803-9335-X. doi: 10.1109/WCNM.2005.1544272. 11

[354] Noor Zaman, Khaled Ragab, and Azween Bin Abdullah, editors. *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*. IGI Global, 2012. ISBN 978-1-4666-0101-7. doi: 10.4018/978-1-4666-0101-7. 12

[355] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory Of Modeling and Simulation*. Academic Press, 2nd edition, January 2000. ISBN 0127784551. 152, 153, 155

[356] Junguo Zhang, Wenbin Li, Ning Han, and Jiangming Kan. Forest fire detection system based on a ZigBee wireless sensor network. *Frontiers of Forestry in China*, 3(3):369–374, 2008. ISSN 1673-3517. doi: 10.1007/s11461-008-0054-3. 10, 11

[357] Junguo Zhang, Wenbin Li, Zhongxing Yin, Shengbo Liu, and Xiaolin Guo. Forest fire detection system based on wireless sensor network. In *Proc. of the 4th IEEE Conf. on Industrial Electronics and Applications (ICIEA 2009)*, 2009. ISBN 978-1-4244-2800-7. doi: 10.1109/ICIEA.2009.5138260. 11

[358] Yang Zhang, Nirvana Meratnia, and Paul Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 12 (2):159–170, 2010. ISSN 1553-877X. doi: 10.1109/SURV.2010.021510.00088. 12

[359] Bo Zhao and Huazhong Yang. Design of radio-frequency transceivers for wireless sensor networks. In Geoff V. Merrett and Yen K. Tan, editors, *Wireless Sensor Networks: Application-Centric Design*, pages 249–269. InTech—Open Access Publisher, 2010. ISBN 978-953-307-321-7. doi: 10.5772/658. 48

[360] Feng Zhao and J. Guibas, Leonidas. *Wireless Sensor Networks: An Information Processing Approach*. The Morgan Kaufmann Series in Networking. Elsevier Science Ltd., 2004. ISBN 978-1-55860-914-3. 10, 11, 12

[361] Jianyang Zheng, Yinhai Wang, and Nancy L. Nihan. Quantitative evaluation of GPS performance under forest canopies. In *Proc. of the 2nd IEEE Int'l Conf. on Networking, Sensing and Control*, pages 777–782, 2005. ISBN 0-7803-8812-7. doi: 10.1109/ICNSC.2005.1461289. 45

[362] Rong Zheng and Chengzhi Li. How good is opportunistic routing? – A reality check under Rayleigh fading channels. In *Proc. the 11th Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2008)*, pages 260–267. ACM, 2008. ISBN 978-1-60558-235-1. doi: 10.1145/1454503.1454549. 12, 13

[363] Chuan Zhu, Chunlin Zheng, Lei Shu, and Guangjie Han. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632, 2012. ISSN 1084-8045. doi: 10.1016/j.jnca.2011.11.016. 12, 39

[364] ZigBee Alliance, Inc. Zigbee specification. Document 053474r17, 2007. URL `http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Specification.aspx`. Available online (last accessed: 25 June 2013). 11

[365] Michele Zorzi and Ramesh R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337–348, 2003. ISSN 1536-1233. doi: 10.1109/TMC.2003.1255648. 13, 14, 16, 170

[366] Michele Zorzi and Ramesh R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003. ISSN 1536-1233. doi: 10.1109/TMC.2003.1255650. 13, 14, 16, 170