Self-Organizing Construction of Connected k-Hop Dominating Sets in Wireless Sensor Networks

Self-Organizing Construction of Connected k-Hop Dominating Sets in Wireless Sensor Networks

Peter Janáčik

A thesis submitted to the
Faculty of Computer Science, Electrical Engineering and Mathematics
of the
University of Paderborn
in partial fulfillment of the requirements for the degree
Dr. rer. nat.

Paderborn, Germany
March 22, 2010

HEINZ NIXDORF INSTITUTE
University of Paderborn

Main supervisor:
Prof. Dr. rer. nat. F. J. Rammig, University of Paderborn

Acknowledgements

First of all, I would like to express my gratitude to Prof. Dr. Franz J. Rammig for supervising this work. In particular, I thank him for his support, helpful advice, and the constructive directions he has given me. Despite being very busy, he always took his time to answer spontaneous questions, even if that often meant interrupting his own work. I really appreciate the great freedom I had to shape the focus of my research, as well as, to publish at, and visit many international conferences. Within his working group, I had the opportunity to become involved in various interesting research projects and to advise students on their BSc and MSc theses.

In the same way, I am grateful for the discussions and joint work with many colleagues. Particularly inspiring and fruitful were the scientific and not-so-scientific discussions and cooperations with Johannes Lessmann, Tales Heimfarth, Dalimir Orfanus, Emi Mathews, Michael Kortenjan, Fahad Bin Tariq, and Sufyan Samara. Looking back at my time in the group of Prof. Rammig, I enjoyed getting to know and to collaborate with colleagues and friends from so many different nations. Furthermore, I am also indebted to my students, especially to Alexander Kujat, Michael Karch, Lazar Lachev, Yara Khaluf, Thomas Kemmerich, and Dirk Meister, who helped shaping ideas, as well as, conducted much of the implementation work.

I am also deeply thankful to my wonderful fiancée, Tanja—a great team player, my truest friend, source of joy and comfort. Her support, encouragement, and love were an enormous help during the work on my dissertation.

Moreover, I would like to say thank you to my parents for their love, the right choices during my youth, and support thereafter.

Finally, most of all, I thank God for all of his great support and overwhelming love.

Contents

List of Figures

List of Tables

List of Abbreviations

AODV	Ad Hoc On-Demand Distance-Vector
ARPA	Advanced Research Projects Agency
ARPANET	ARPA Network
CCA	Connection Construction Agent
CDS	Connected Dominating Set
CEAs	Connection Exploration Agents
CIPA	Center Information Propagation Agent
CkDS	Connected k-hop Dominating Set
CORIE	Columbia River Estuary
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DARPA	Defense Advanced Research Projects Agency
DC	Drought Code
DMC	Duff Moisture Code
DSR	Dynamic Source Routing
EOFS	Environmental Observation and Forecasting System
FFMC	Fine Fuel Moisture Code
FWI	Fire Weather Index
ICA	Initial Construction Agent
IEA	Initial Exploration Agent
LMST	Local Minimum Spanning Tree
MAC	Medium Access Control
MIS	Maximal Independent Set
MPR	Multi-Point Relay
MPRS	Multi-Point Relay Set
NCW	Network Centric Warfare
NPL	National Physical Laboratory
PRNET	Packet Radio Network
RF	Radio Frequency
RFID	Radio-Frequency IDentification
RSSI	Received Signal Strength Indication
RTI	Returnable Transport Item
RTS	Request To Send
SOSUS	Sound Surveillance System
SURAN	Survivable, Adaptive Network

SYNC .... Synchronization
TIC .... Traffic Information Center
UGSN .... Unattended Ground Sensor Network
WCDS .... Weakly-Connected Dominating Set
WSN .... Wireless Sensor Network

1 Introduction

A wireless sensor network (WSN) typically consists of highly resource-constrained wireless nodes that collectively monitor the area in which they are deployed. Their radio consumes relatively high amounts of energy compared to other WSN node components (for a short overview of WSN hardware see Section 2.1.4). Thus, there is a need for effective approaches to cope with this challenge. One such approach that is very promising is the construction of connected k-hop dominating sets (CkDS), which can help to save energy by, for example, reducing the effects of a broadcast storm or allowing non-CkDS nodes to decrease their duty cycles. In addition to that, CkDS can be employed for various other purposes. Before expanding this discussion, a connected k-hop dominating set needs to be defined:

Definition 1 *A set of vertices $D_{Ck} \subseteq V$ in an undirected, connected graph $G = (V, E)$ is a* connected k-hop dominating set *(CkDS), if two properties are satisfied: (1) each vertex v from the set of all vertices in the graph, V, is either in D_{Ck}, or there exists a path of length m, $m \leq k$, between v and a vertex in D_{Ck}; (2) between each pair of vertices in D_{Ck}, there exists a path which consists only of vertices from D_{Ck}. I provide further definitions needed in Section 4.3.*

CkDS have various applications in WSNs: They are used to lower the amount of overhearing, idle listening, and collisions, when contention-based medium access control is used, by thinning out the topology and thus reducing the number of competing nodes. Further, a CkDS can alleviate the broadcast storm problem, by limiting the flooding of packets to a smaller subnetwork (which is the CkDS). A further application of CkDS that makes them especially useful in WSNs is area coverage. Moreover, a CkDS can be used to provide a lower-resolution image of the data acquired by the whole network. Finally, a CkDS establishes rendez-vous areas that can be utilized to efficiently implement, for instance, publish-subscribe mechanisms. For an extensive discussion of CkDS applications in WSNs, see Section 2.2.

A wide range of approaches focuses on the construction of connected 1-hop dominating sets (CDS) for ad hoc networks and WSNs, such as [11], [50], and [164], as reviewed in Section 3.1. However, often a CkDS with $k > 1$ is desired for the applications discussed above. Therefore, during the last years, several novel CkDS approaches were proposed for the considered class of networks. To my knowledge, there are four CkDS approaches published in [130], [143], [165], and [166]; they are reviewed in Section 3.2. The commonality of these approaches is that the cost incurred by them is strongly dependent on the node degree and the chosen k.

In Chapter 4 of this thesis, I propose a novel protocol which is fundamentally different from state of the art (that is [130, 143, 165, 166]): a *CkDS* is constructed mainly using random walks consisting of a sequence of unicasts, minimizing the dependence of cost incurred on node degree and eliminating such a dependence on k. It is the first protocol for the construction of connected dominating structures, including CDS and CkDS, in wireless networks to be based on this technique. The proposed self-organizing protocol is inspired by the general technique of random walks and, in particular, by the flight behavior of ovipositing Pieris rapae, inheriting some of its desirable properties. It uses two intertwined behavior blocks, the first aimed at creating a dominating set, while the second one connects this set to a CkDS. Section 4.2 provides a comprehensive summary of the proposed protocol.

I conduct an in-depth analysis of the properties of my approach and compare them to the properties of other state-of-the-art approaches. Key findings from this evaluation are summarized in Section 5.2.9.

During the design of the proposed protocol, I focused on its application in wireless sensor networks, since its biological archetype appeared to exhibit exactly the strengths desirable in this network class: a high amount of tolerance to an unreliable topology, extremely low requirements in terms of information exchange, and the ability to operate within an arbitrarily large and globally unknown system, to name the most important ones. Naturally, independent of its suitability for WSNs, the proposed protocol could be very well applied to create structures in many other areas, such as in large-scale wired networks.

After this brief introduction, Chapter 2 provides useful background information relating to the topic of this thesis. Subsequently, Chapter 3 reviews the state of the art, before Chapter 4 presents the proposed approach to CkDS construction. The results of the evaluation of the proposed approach are described and discussed in Chapter 5. Finally, Chapter 6 concludes this thesis.

2
Background

Following the brief introduction, this chapter describes the background of CkDS, forming a basis for later chapters of this thesis. It is organized as follows: The first section provides an overview of different aspects of wireless sensor networks (WSNs), such as history, applications, hardware, and communication, enabling the reader to understand the nature and the challenges of WSNs. Thereafter, the second section presents the motivation for the construction of *connected k-hop dominating sets* (CkDS).

2.1 Wireless Sensor Networks

Currently, wireless sensor networks (WSNs) are a field of extensive research. These networks typically consist of computer nodes

1. with very limited hardware resources,
2. equipped with sensing capabilities,
3. and low-power wireless radios.

WSNs have been identified by the *MIT Technology Review* as one of ten emerging technologies that will change the world [126]. *Business Week* even regards sensor networks as one of *21 ideas for the 21st century* [8], eventually covering the world like an *electronic skin*, autonomously detecting events and taking actions.

This section first looks at the history of wireless sensor networks. Thereafter, in order to provide a background for communication protocols, this section describes history, applications, hardware and communication properties of WSNs.

2.1.1 History

The background knowledge of the origins of WSNs helps to understand the motivation for developments which led to the current state of these networks and their communication protocols. This overview includes important milestones of WSN development, starting after the second world war.

2.1.1.1 Cold War

As with many other technologies, the first sensor network was developed and deployed by the military. In the aftermath of World War II, in 1949, the US Navy recognized the necessity of long range underwater acoustic, wired sensor networks in order to be able to detect hostile submarines [106]. At that time the detection focused on the low frequency sounds produced by the state-of-the-art diesel submarines. Receiving a generous funding, the first installations of the *Sound Surveillance System* (SOSUS) network included a six acoustic sensor node network in 1951 [41], followed by 80 nodes in 1960. The first sighting of a SOSUS-detected submarine occurred during the Cuban Missile Crisis on October 26, 1962 [106]. Similarly to SOSUS, a wired network of air defense radars and other sensors, aimed at defending the continental United States and Canada, was put into operation during the Cold War [41].

In the early 1960s, Paul Baran laid the foundation for routing and data collection used in many of today's WSNs, by introducing the *store-and-forward* concept, which later has become known by the name *packet switching* [13]. Remarkably, the *Hot-Potato Heuristic Routing Doctrine*, described in the same paper, already employs a *flooding*-based route discovery, as well as, routing tables including links belonging to *redundant* paths towards the destination. Baran's concept of packet switching has been taken up by the *Advanced Research Projects Agency* (ARPA) and the National Physical Laboratory (NPL) for ARPANET [124] and the network designed by NPL [51] in the mid 60s.

The first experimental *packet radio networks* (PRNETs) were put into operation between 1975 and 1978 by Robert E. Kahn and colleagues [84, 85] based on the experiences acquired through ALOHA, an early MAC protocol developed at the University of Hawaii [1]. Similar to ARPANET, given the military background, these first PRNETs focused on reliability and survivability. In order to advance PRNETs, the *Defense Advanced Research Projects Agency* (DARPA) initiated the *survivable, adaptive networks* (SURAN) program. It was aimed at developing low-cost radios, as well as, corresponding robust, adaptive, and secure algorithms for packet radio networks scaling to thousands of nodes [19].

During the course of the 1980s, besides new approaches for communication-oriented operating systems that were suited for the first generations of networks, consisting of computers equipped with sensors [120], also diverse signal-processing techniques, such as, for multi-target tracking were developed [32].

2.1.1.2 1990s

In the 1990s, the military even more embraced sensor network technology recognizing its key role in *network centric warfare* (NCW) [4]. The overall goal in network centric warfare is to transfer the intelligence from the weapons or sensors to an information infrastructure, called *infostructure*, and relocating the complexity from the platform to the network in order to perform better in conflicts.

An example for NCW besides e.g. [82] are *unattended ground sensor networks* (UGSN) for area surveillance [29] using many simple, limited-capacity, miniature wireless sensor nodes. As highlighted by the authors of [29], the advantages of such systems are their robustness (providing graceful degradation) and low price compared to using a network of fewer, more sophisticated sensor nodes. So in contrast to the traditional approach of employing computationally intensive algorithms on a single image provided by a sophisticated sensor, the UGSN uses computationally efficient algorithms on data gathered from different sensors, which—as the authors argue—simplifies the matter. Further advantages of *distributed collaborative surveillance systems* are described in [52]. The wireless nodes in an UGSN may include mobile nodes, such as, unmanned all-terrain vehicles equipped with a variety of sensors and communication equipment.

After the change of the political climate in the early 1990s, the SOSUS system, operated by the U.S. Navy, was used to monitor seismic activity [22]. Its goals were to study low magnitude oceanic earthquakes, as well as, to determine the spatial and temporal distributions of large marine cetaceans in the Atlantic Ocean by cataloging their acoustic signals. SOSUS' advantage over land-based seismographs is its capability to detect two orders of magnitude more locatable events. Similar to this system, a variety of other *environmental observation and forecasting systems* (EOFS) was brought into being during the 1990s. Besides further seismic networks [102, 137] e.g. for Tsunami reporting [102], other EOFS were put into operation. One example for such a network is the *Columbia River Estuary* (CORIE) [12, 136] consisting of 13 sensor stations located throughout the Columbia River estuary and one offshore station. Data such as salinity, temperature, water velocity, and level obtained by the sensors is streamed via a single-hop wireless link to a master node near the shore.

By the early 1990s, the development in micro computer and wireless communication technology led to an improvement in capabilities at lower prices, which resulted in the arrival of new sensor network-based solutions for already existing applications. It should also be noticed that, at this time, *radio frequency* (RF) communication gained acceptance in more and more market sectors: automotive, personal security, agriculture, manufacturing, and transportation, to name just a few [74]. Besides applications, like automatic feeding or milking of cattle, computer networks appear to have been used only to track entities by identifying them at different sensor nodes of a network, which is distributed over a geographical area. Some of the applications observed at this time are tracking of: farm animals; tools used at different machining centers to better assess their wear and danger of breakage; car bodies; railway wagons and containers.

An example for such networks is the *active badge location system*, installed in 1990 by Want and colleagues [153]. At the time, there was a need for a system determining the location of staff members in large office buildings and hospitals. Given its high cost in the early 1990s, affordable mobile telephone technology was not a viable solution. Although the problem could be addressed by a pager system, this solution suffers from several drawbacks: First, if the callee does not reply, the caller cannot know whether the callee's pager did not receive the message due to an unsuccessful transmission or simply did not reply intentionally or unintentionally (for example, by misreading the call-back number). Second, if there are several people capable of reacting in an emergency, it is not known which one of them is located the closest and is thus the most suited for being contacted.

Another alternative would be using card-key systems known from access control/logging to acquire the needed location information. For many organizations it is inappropriate to use this system, given the inconveniences incurred, such as distractions and loss of time. Moreover, the system may exhibit inaccuracies, if multiple people obtain access to adjoining zones using only one card-key.

The solution described by Want et al. [153] consists of active badges ($55 \times 55 \times 7$ mm, 40 g), carried by the personnel, and a network of sensors distributed in the host building. To enable a detection by the sensors, the active badges emit a unique code via a pulse-width modulated infrared signal for approximately a tenth of a second every 15 seconds. The badge's batteries are expected to last for about one year.

Akyldiz et al. [3]	Chong et al. [41]	Karl et al. [87]
Military	Infrastructure security	Disaster relief
Environmental	Environment and habitat monitoring	Environmental control and biodiversity
Health	Industrial sensing	Intelligent buildings
Home	Traffic control	Facility management
Other		Machine surveillance and preventive maintenance
		Precision agriculture
		Medicine and health care
		Logistics
		Telematics

Table 2.1: Different categorization schemes for WSN applications

In the mid and late 1990s, wireless mobile ad hoc networks have become an area of increased research activity, paving the way for the advent of wireless multi-hop sensor networks. New protocols like *Dynamic Source Routing* (DSR) [83] or the *Ad Hoc On-Demand Distance-Vector* (AODV) routing [115], enabling efficient multi-hop communication, were developed. The first non-military real-world evaluations of these protocols were conducted in the late 1990s with four static nodes running the associativity-based routing protocol [147, 148], as well as, five mobile car-mounted and two stationary nodes executing DSR [98].

2.1.2 Current Applications

Presently, there is a vast range of applications for WSNs. In order to provide an overview, it is helpful to examine the different categories they fall in. Different categorization schemes have been proposed in [3, 41, 87], as described in Table 2.1. Since the scheme from [87] appears to be the most comprehensive and up-to-date of the three, I chose it as a basis for an overview. The categories used in this scheme are listed in the third column of Table 2.1.

For each category, I chose to describe two example applications in the following overview:

Disaster relief applications Within this application category, the goal of a WSN is to prevent disaster or help to cope with it, when it strikes, as in the following examples:

Flood To help avoid the devastating effects of river floods, a flood detection sensor network [14] is tested along the Agun River and its channels in north-eastern Honduras. The two-tier network consists mainly of patches comprising water pressure (for obtaining the river level), rain, and temperature sensors. These patches are spaced at distances on the order of 25 km. Intra- and inter-patch communication is conducted using different frequencies (900 and 144 MHz). Figure 2.1 depicts this system. The dotted circles around the water pressure sensors illustrate the intra-patch communication, whereas the dotted lines between these sensors represent inter-patch links.

Forest fire The goals of the WSN described in [73] are the early warning about potential forest fires and the estimation of the scale and intensity of existing fires. To achieve these goals, the authors use the *Fire Weather Index* (FWI) taking into account temperature, relative

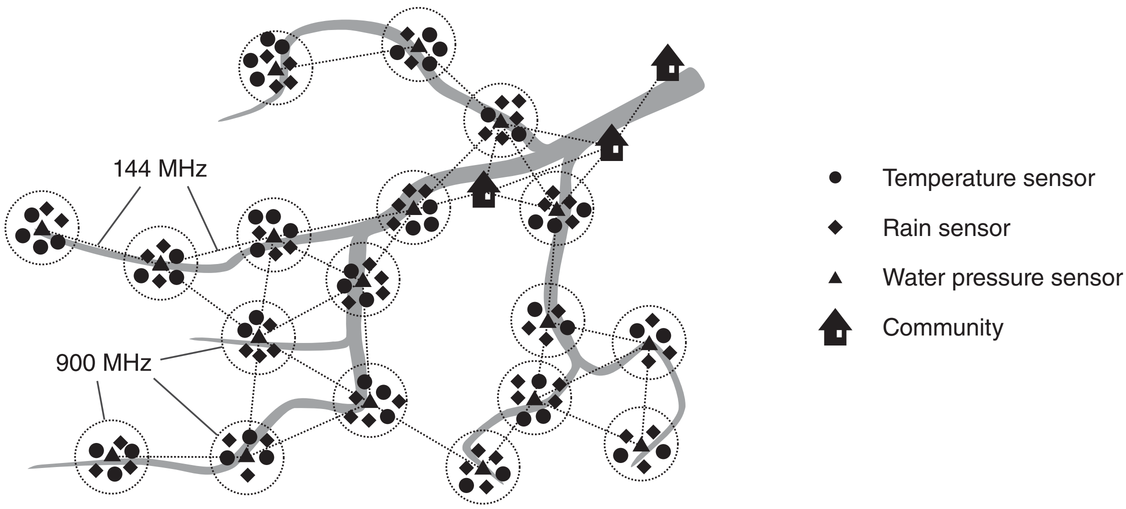

Figure 2.1: Flood warning sensor network

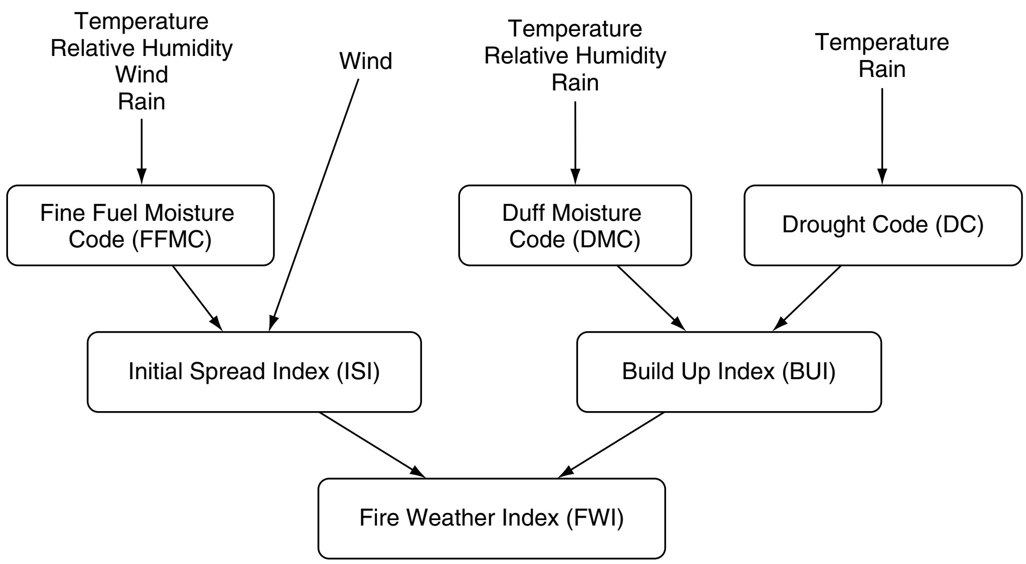

Figure 2.2: Computation of fire weather index

humidity, wind, and rain. They assume a node's sensing range of approximately 100 m yielding networks consisting of thousands of nodes. To get an impression of how these decisions are made, it is interesting to consider the process depicted graphically in Figure 2.2 employing the following codes and indices:

- *Fine Fuel Moisture Code (FFMC)*: Moisture content of litter and fine fuels. Depth: 1–2 cm. Indicates ignition probability.
- *Duff Moisture Code (DMC)*: Assesses the moisture content of loosely compacted, decomposing organic matter. Depth: 5–10 cm. Determines the probability of fire ignitions due to lightning and indicates rate of fuel consumption.
- *Drought Code (DC)*: Rates moisture content of the deep layer of compacted organic matter. Depth: 10–20 cm. Reflects long-term moisture conditions, determines a

fire's resistance to extinguishing, and indicates fuel consumption.

Further, there are three fire indices assessing the behavior of fire:

◇ *Initial Spread Index (ISI)*: Determines the rate of fire spread immediately after ignition.

◇ *Build Up Index (BUI)*: Reflects the total amount of fuel available for combustion.

◇ *Fire Weather Index (FWI)*: Indicator for fire intensity, which is defined as the energy output measured in kilowatts per meter of flame length at the head of a fire.

Environmental control and biodiversity mapping Applications within this category have the task of helping to control the environment by providing the necessary means to assess it. Examples for applications falling into this area are:

Noise pollution The authors in [128] developed a WSN consisting of nodes collecting acoustic samples. Its aim is to lower the cost for noise pollution mapping while drastically increasing its accuracy. Moreover, the network is even capable of providing vehicle counts and estimates of per-vehicle noise levels. The data can then be visualized in common map-based web interfaces like *Google Maps* [68].

Habitat monitoring An application for habitat monitoring at Great Duck Island is described in [97, 141]. The network helps to acquire data for the monitoring of the ecology of Leach's Storm Patrel using a multi-tiered network of distributed sensor patches. Its advantages over conventional (non-sensor network) methods include an inconspicuous, continuous, and long-lived operation, as well as, the possibility of management at-a-distance.

Intelligent buildings Using wireless sensor networks in a building aims, for example, at increasing the comfort level of its occupants, reducing energy consumption, and lowering wiring costs, such as in the following applications:

Light control Singhvi et al. [133] propose a light control network using a strategy with the objective to find the best trade-off between satisfying the occupant's preferences and minimizing the corresponding energy utilization. The network consists of wirelessly-connected nodes sensing for light and RFID tags [72], using this information to actuate light levels.

Motion monitoring In order to enable intelligent buildings to operate properly and adjust exactly to their occupants' behavior, in many cases, it is required to obtain further information like people's location and motion. The WSN proposed and evaluated in [21] uses, among other sensors, video cameras and laser scanners to achieve this task. The sensor data obtained is fused and can be displayed on a map of the building.

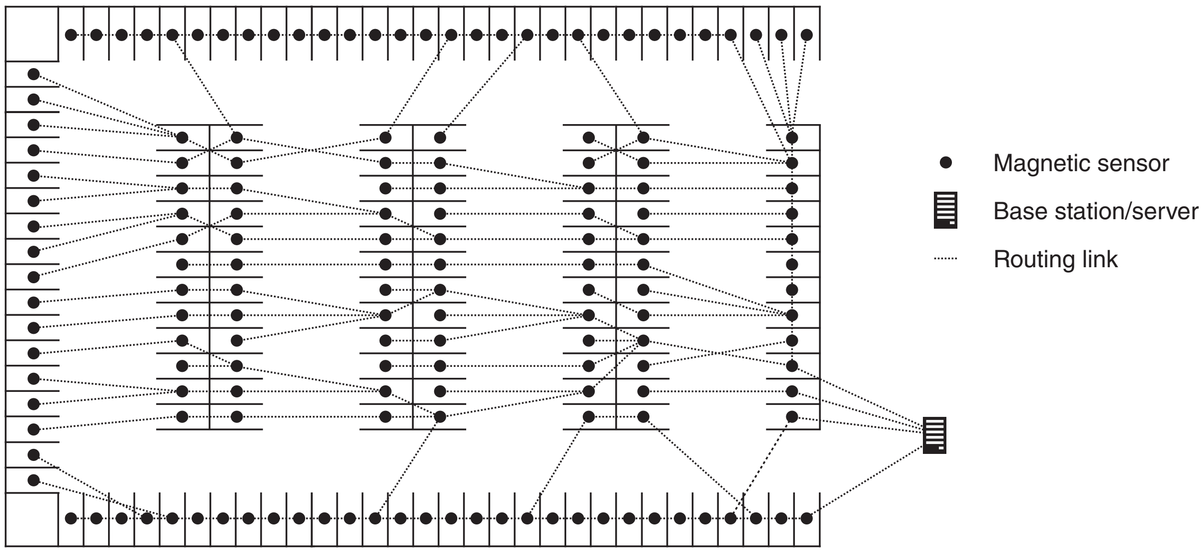

Figure 2.3: Car park management network with a possible routing topology

Facility management This class of applications usually has the task to provide services spanning over complexes of buildings, such as, access control or the monitoring of the status and motion of single entities. Further possible goals may be the detection of conditions of interest, such as, wasteful energy consumption.

Car park management The car park management network described in [17] aims at guiding a driver's choice of parking space, as well as, supporting management and planning. Nodes used in this network are equipped with magnetic sensors and situated on the ground in the center of each parking space. Occupancy information is then routed over multiple hops towards the base station, as depicted in Figure 2.3. This networks's advantages over wired systems are its substantially lower installation costs in already existing car parks and its potential for being used for on-street parking spaces.

Electrical energy consumption monitoring In order to enable facility administration to realize energy savings, the exact consumption data for a large amount of devices is measured by the WSN described in [86]. It consists of sensors placed, for example, adapter-like in power outlets or sub-distribution nodes, such as, fuse boxes. Consumption data is routed to a central unit over multi-hop paths.

Machine surveillance and preventive maintenance In industrial applications used for machine surveillance and preventive maintenance, WSNs help to lower costs for wiring, which can amount to $ 1000 per foot [146]. Moreover, WSNs can be employed and operated with less effort in hazardous, inaccessible, or restricted areas. Ota and Wright [111] provide an overview of these and further trends in this application area.

Condition-based maintenance for heating and air conditioning Tiwari et al. [146] describe a WSN installed at a heating and air conditioning plant equipped with vibration, temperature, pressure, strain, and other sensors. The data obtained from the sensors is collected at a central location and analyzed offline.

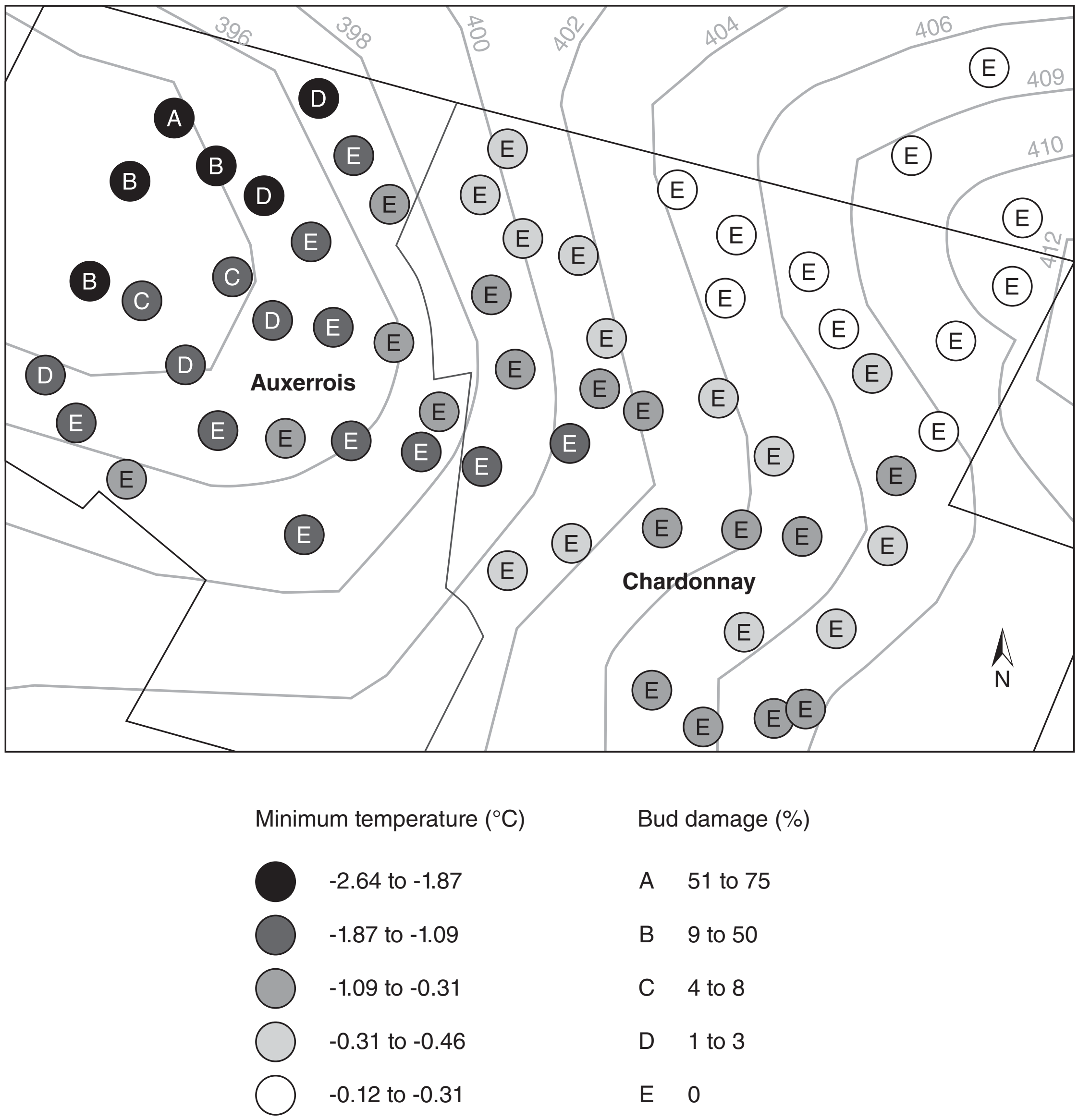

Figure 2.4: Real-world deployment of a WSN in a vineyard in Okanagan Valley, British Columbia. Data source: [16]

Shipboard machine monitoring A WSN for British Patrol ships was evaluated at *Loch Rannoch*, an 132,000 ton oil tanker [88]. Using a multi-hop network, the system copes well with the ship's high metal content, which represents a tough environment for RF signals. One-hundred-and-fifty accelerometers, attached to machinery, like the engine's pumps and motors, provide the data. Checks, which were conducted before every eight weeks manually by operators using hand-held devices, connecting to one accelerometer at a time, can now be performed automatically every 18 hours.

Precision agriculture Carrying out crop or livestock surveillance, sensor networks are effective means of status assessment, for example, the status of the animals' health conditions. Further, they offer recommendations, such as for appropriate harvest times.

Vineyard The authors in [16] describe a wireless multi-hop sensor network, which was installed and evaluated in a vineyard. Deploying the nodes densely, they found out that the mean temperature can vary by 35 % measured in heat summation units in as little as 100 meters, which is a greater difference than the 20 % between Tuscany and the Rhine region. This precise knowledge enables the winegrower to match wine grape types to planting conditions more exactly. Further, the network enables high-resolution frost warnings, allowing the winegrower to make a targeted use of irrigation systems to prevent frost damage.

Figure 2.4 shows a cutout of the vineyard, which was equipped with wireless sensors, and the obtained real-world sensor data. It helps the winegrower to better understand the climatic conditions in the vineyard, enabling him or her to increase revenues. The sensor nodes are depicted as circles. Different shades of the circles represent the minimum temperatures obtained by the nodes during the first arctic outflow during the season. The amount of bud damage due to frost is represented by the letters inside the circles. Boundaries define the areas planted in Auxerrois Blanc and Chardonnay.

First, this particular data indicates that elevation co-varies with temperature, but not perfectly. Further, it is evident that bud damage occurs only in the coldest areas. Roughly summarized, the more cold-proof Auxerrois appears to be planted correctly in the lower and colder areas. However, the temperature results indicate that the south-eastern part of the Auxerrois area seems to be warm enough for Chardonnay. This is a tangible example for information winegrowers are interested in, as the per-ton price of Chardonnay is often as high as double than that for Auxerrois [16].

Virtual fences In order to implement virtual fences, cows are equipped with collar-mounted wireless sensor nodes using GPS for location tracking [25]. When approaching a virtual fence, a node plays a naturally occurring sound that is scary to the animals (e.g. a roaring tiger). Furthermore, sensor nodes collect useful data, such as position traces, enabling the creation of grazing models and response profiles for single animals. To move virtual fences, new coordinates are propagated through the network in a multi-hop manner.

Medicine and health care In this area, the use of WSNs aims at the reduction of cost and improvement of quality in patient monitoring and care. However, some issues are controversial from the ethical point of view, such as the increased anonymity that could be incurred in certain scenarios.

Fall detection In order to detect accidental falls of patients, Tabar et al. [142] developed a WSN consisting of an accelerometer-equipped user-attached node and several wireless image sensor nodes, installed near the ceiling on a wall or directly at the ceiling. Using received signal strength measurement, position tracking is implemented. When there is no movement observed for extended periods of time, this circumstance is reported. To detect falls more promptly, a second mechanism is used: When an accelerometer-equipped node attached to the user senses a likely fall, it triggers the most suitable image sensor according to the position information. These sensors then process the data and exchange local decisions to generate a report. Both sensor types cooperate to decrease the number of false alarms.

Long-term monitoring The sensor network described in [151] enables continuous long-term monitoring of assisted- and independent-living residents. Consisting of wirelessly connected motion, body, indoor temperature, luminosity, pulse-oximeter, EKG, and other sensors, the network tracks motion proactively and may receive queries for patients' vital signs and environmental conditions. The authors stress that, since their system uses wireless communication and self-management, high costs could be avoided.

Logistics WSNs are applied in logistics, providing the possibility of monitoring the delivery of goods more precisely and thereby increasing efficiency. Further they help to lower the risk of errors by reducing the number of interactions with personnel.

Cold chain management The cold chain management network described in [123] comprises multiple tiers. At the lowest tier, it consists of sensor units, which collect temperature data and are transported with the goods. The next-higher tier contains relay units, which forward the data read from the lowest tier via a multi-hop route towards the access box, the local master of the system, on the next-higher tier. Access boxes then report to the data warehouse. This application demonstrates one of the evident strengths of WSNs: the installation of a site in a retail store required about four hours for a network of 60 nodes.

Item tracking in a warehouse Evers et al. [60] developed a WSN to support the process of transporting products from their bin locations over the expedition floor towards trailers, which ship them to the customers. The proposed network aims at reducing the amount of errors, such as wrong selection of items or item misplacement, at speeding up the handling of items, and at reducing the size of the human work force. To achieve this, each returnable transport item (RTI), such as a pallet or cart, is equipped with several wireless sensor nodes. Besides providing a means of identification, these sensors measure environmental information, necessary to preserve the quality of goods like flowers, and communicate with each other operating in a context-aware manner. Furthermore, wireless sensor nodes are placed in a grid above the expedition floor serving as a basis for localization, and providing the means of communication with a central coordinator, for instance, to acquire the target to which RTIs are to be delivered or to report their activities.

Telematics WSNs that consist of nodes built into cars or nodes embedded in the roadside are employed, for example, in order to improve safety, reduce the length of travel or fuel consumption.

Adaptive vehicle navigation Current navigational systems mainly rely on static maps and information provided via systems like TMC, which is however limited, since it has to be compiled by a central *traffic information center* lacking granularity and timeliness. Therefore, the WSN presented in [31] consists of nodes communicating via IEEE802.16 e/j [10]. They are built into vehicles that are capable of sensing speed and direction of their neighbors and are also equipped with GPS receivers [48]. When inquiring the traffic information of the roads towards the destination, a vehicle dispatches a query propagated to nodes on these roads in multi-hop manner. After the query replies arrive at the inquiring node, it determines a route based on information provided by the static map and these replies.

Application	Target network size	Deployment method	Node size
Flood warning [14]	1 per 25 km of river	manual	small tower
Forest fire detection [73]	1000s	manual	not specified
Noise pollution monitoring [128]	100s	manual	brick
Habitat monitoring [97, 141]	10s–100s	manual	matchbox
Light control [133]	1000s	manual	matchbox
Motion monitoring [21]	1000s	manual	matchbox/brick
Car park management [17]	100s–1000s	manual	matchbox
Energy consumption monitoring [86]	100s–1000s	manual	matchbox
Condition-based maintenance for heating and air conditioning [146]	1000s	manual	matchbox
Shipboard machine monitoring [88]	1000s	manual	matchbox
Vinyard monitoring [16]	100s–1000s	manual	matchbox
Virtual fences for cattle [25]	100s–1000s	manual	brick
Fall detection for persons in care [142]	$<$10 per room	manual	matchbox
Long-term patient monitoring [151]	10s per room	manual	matchbox
Cold chain management [123]	1000s	manual/automated	matchbox/brick
Item tracking in a warehouse [60]	1000s	manual/automated	not specified
Adaptive vehicle navigation [31]	millions	automated	built into car
Collision warning system [140]	10s to millions	manual	brick

Table 2.2: Comparison of WSN applications: target network size, deployment method, node size

Collision warning system To reduce the number of unexpected accidents on curved roads, Sung et al. [140] developed a multi-tier collision warning system. At the lowest tier, the network consists of nodes deployed in the center of the road lane, equipped with magnetic sensors to detect vehicles. These nodes report detections to sink nodes, which belong to the second tier and are installed at the roadside. Sink nodes propagate data in multi-hop manner towards base stations at the third tier, which aggregate and analyze it and then provide the results of this analysis to vehicles within transmission range. The motivation for situating the propagation step in the second tier are the poor communication capabilities of the sensor nodes in the lowest, first tier, given their in-road deployment.

Tables 2.2 and 2.3 provide an overview of the key properties of the application examples presented above.

2.1.2.1 Comparison of Applications

In Table 2.2, the target network and node sizes, as well as, deployment methods are presented for each application. The target network size does not represent the size of the networks used for evaluation in the cited publications, but the estimated size of the deployment as it is planned, once the networks leave the test phase. Table 2.3 provides information on mobility, network graph topology, and the degree of evaluation.

Strikingly, when looking at the target network sizes, it is evident that a majority of the applications is designed for hundreds of nodes or more when deployed, although this fact is not yet reflected by the sizes of the networks used for evaluation. Nonetheless, when many of the

Application	Mobility	Network graph topology	Degree of evaluation
Flood warning [14]	static	multi-hop/-tier	medium-scale test deployment
Forest fire detection [73]	static	multi-hop	simulation
Noise pollution monitoring [128]	static	one/multi-hop/-tier	small-scale test deployment
Habitat monitoring [97, 141]	static	multi-hop/-tier	medium-scale test deployment
Light control [133]	static	single-hop	testbed
Motion monitoring [21]	static	multi-hop	medium-scale test deployment
Car park management [17]	static	multi-hop	small-scale test deployment
Energy consumption monitoring [86]	static	multi-hop	not specified
Condition-based maintenance for heating and air conditioning [146]	static	single-hop	small-scale deployment
Shipboard machine monitoring [88]	static	multi-hop/-tier	large-scale test deployment
Vinyard monitoring [16]	static	multi-hop/-tier	medium-scale test deployment
Virtual fences for cattle [25]	cattle mobility	multi-hop	small-scale test deployment
Fall detection for persons in care [142]	static/patient mobility	single-hop	small-scale test deployment
Long-term patient monitoring [151]	static/patient mobility	single-hop, multi-tier	small-scale test deployment
Cold chain management [123]	static/good mobility	multi-hop/-tier	medium-scale test deployment
Item tracking in a warehouse [60]	static/good mobility	multi-tier	not specified
Adaptive vehicle navigation [31]	car mobility	multi-hop	numerical evaluation
Collision warning system [140]	static	multi-hop/-tier	small-scale deployment

Table 2.3: Comparison of WSN applications: mobility, network graph topology, degree of evaluation

described networks like noise pollution monitoring, virtual fences, and car park management become operational, their deployment is only reasonable at larger scales.

Given the nature of many applications, such as energy consumption or shipboard machine monitoring, it is only possible to deploy sensor nodes manually with the help of standard tools. Naturally, there is the alternative of building sensor nodes into the objects which are to be monitored, such as power outlets or ship engines. An example for such built-in sensors is provided by the adaptive vehicle navigation application.

When examining the mobility properties of the discussed applications, it becomes apparent that most of them are static. In case there is mobility, the node movement follows the entities observed, such as cattle, goods, or cars, whose mobility models differ substantially. Therefore, it is not possible to make assumptions about a *general* node mobility model for WSNs.

Similar to mobility, there is a predominant topology type: multi-hopping. There are different reasons for this. First, multi-hopping copes well with an environment which is hostile to the propagation of RF signals, such as the engine room of a tanker. Second, in many cases, multi-hopping can decrease overall energy consumption by using shorter links and reducing the amount of contention for the medium, as described in Section 2.1.5.4.

2.1.3 Envisioned Future Applications

When designing communication protocols for WSNs, it is helpful to keep in mind the current and possible future applications. Although it is a challenging task to predict future developments, still it is possible to observe current trends and use them to derive prognoses. The prognoses, which I describe further below, can be summarized as follows:

> In the future, people will be surrounded by wirelessly-networked nodes, which are embedded everywhere, in living rooms, office buildings, or natural parks. They will work to make everyday life more efficient and more convenient. In order to detect the state of the environment and to recognize the wants of their creators, these nodes will be equipped with sensors and thus constitute an extremely large-scale wireless sensor network.

Depending on the perspective of the author, different visions have been devised:

Ubiquitous computing The first vision containing the above elements stems from Marc Weiser of Xerox PARC and is called *Ubiquitous Computing* [154, 155]. Weiser emphasizes the notion of a disappearing technology, which weaves itself into the fabric of everyday life until it is indistinguishable from it. Seeing PCs, notebooks, and other devices only as transitional steps, he compares the coming age of disappearing technology to spoken language or electrical wires in today's buildings.

Pervasive computing In contrast to the academic, idealistic, and human-centered vision of ubiquitous computing, the term pervasive computing [129] was mainly coined by the industry [100]. It has the same core idea, however, with the primary aim of realizing the elements of the vision in the context of electronic commerce and web-based business-process scenarios. Some authors [57, 42] link ubiquitous and pervasive computing to the notion of *smart environments*, i.e. environments making use of ubiquitous/pervasive sensing and actuating.

Ambient intelligence There are many similarities between ubiquitous computing and ambient intelligence [54], which is promoted by the European Union. It is envisioned as a retreated, cautious technology surrounding people, placing a strong emphasis on user-friendliness and some form of artificial intelligence that controls its actions. For instance, a possible implementation could include agents negotiating with other agents on behalf of the user.

Internet of things The concept of the *Internet of things* [53] was formulated by Auto-ID Center [55], now *EPCglobal*, Inc., which consists of over 100 global companies, such as Procter & Gamble, Unilever, but also research centers like the Massachusetts Institute of Technology or the University of Cambridge. EPCglobal aims at establishing an infrastructure for RFID tags built into billions of everyday objects constituting a global Internet of things. However, the term is also used to refer to ubiquitous computing, such as in [100].

Besides the probably most prominent visions described above, there are also some less-known instances, such as:

Proactive computing More technology-centered, the vision of *proactive computing* encompasses three aspects: (1) Proactive computing systems will be physical, i.e. connected to the world around them via sensors and actuators; (2) they will be part of reality, so they will routinely respond to external stimuli; and (3) humans will "stand above" computing systems and not within, as they do now. Therefore, proactive computing appears to imply a similar set of functions as ubiquitous computing.

Data space In data space [76], the world is covered by administrative and also geometric cubes of data. An administrative data cube encapsulates an entity, such as a room, a building, a street, or a city. The cubes are populated by a number of physical objects locally producing and storing data. User queries can navigate through the data space and retrieve this data. When trying to integrate data space with one of the above visions, it is easy to imagine that this is possible: in order to be able to determine actions by processing sensor readings, they not only have to be read, but also stored and made available within the ubiquitous network.

In summary, there are different overlapping visions for the future wireless sensor network, which share a common idea, as discussed above. Concerning network topology, it is evident that all of the visions include extremely large networks which consist of mostly static nodes, as well as, some nodes with sporadic mobility.

2.1.4 Hardware

When network protocols are discussed, the underlying hardware always also needs to be considered. Therefore, in this section, a short outline of current WSN hardware is provided.

There are several different commonly used node platforms. Most of them have their roots in academia, since in recent years, WSNs have been mainly subject to basic research. Next, I provide some examples of more widely used WSN platforms:

Berkeley motes The development of *motes* [65] started in the late 1990s at the University of California at Berkeley, partially in cooperation with Intel. Successors of these first motes, such as the *MicaZ* [44] (depicted in Figure 2.5 (a)), were brought to market by Crossbow Technology, Inc. These nodes usually run *TinyOS*, an operating system also developed at Berkeley, use microcontrollers from the Atmel family, and offer the possibility to connect additional boards. Using a similar design, the Telos sensors [116] were introduced in 2005, commercially available from the Sentilla Corporation. Telos was built from scratch after the lessons learned from previous mote designs. The design goals were a reduced power consumption, as well as, an improved ease of use and robustness. Table 2.4 presents a brief overview of the Berkeley mote family.

Eyes In the context of the *Energy-efficient Sensor Network* project [61], funded by the European Union, the Eyes nodes were developed by the University of Twente. They use a Texas Instruments MSP 430 microcontroller, an Infineon TR 1001 transceiver, and connect to a PC via USB.

BTnode Developed by the ETH Zürich in the context of different research projects, these nodes [58] use Atmel ATmega 128L microcontrollers and Chipcon CC 1000 radios, sharing many similarities with Mica2 nodes.

(a) (b)

Figure 2.5: Example sensor nodes: (a) MicaZ node, (b) SunSPOT. Photo sources: [157, 158]

	Node	WeC	Ren	Dot	Mica	Mica2Dot	Mica2	Telos
Year	1998	1999	2000	2000	2001	2002	2002	2004
Microcontroller								
IC	AT90LS8535		ATmega163		ATmega128			TI MSP430
Flash ROM (kB)	8		16		128			48
RAM (kB)	0.5		1		4			10
Onboard memory								
IC	24LC256				AT45DB041B			ST M25P80
Size (kB)	32				512			1024
Radio	TR1000				TR1000	CC1000	CC1000	CC2420
Interface								
Extension pins	none	51	51	none	51	19	51	16
PC Communication	IEEE 1284/RS232							USB
Integrated sensors	no	no	no	yes	no	no	no	yes

Table 2.4: Comparison of members of the Berkeley mote family. Data source: [116]

ScatterWeb The Freie Universität Berlin developed a family of nodes [131] ranging from standard devices equipped with a Texas Instruments MSP 430 microcontroller and the RFM TR 1001 radio (see Table 2.5) to more exotic nodes embedding web servers, as well as, offering a wide range of connection possibilities, such as CAN support.

Table 2.5 shows key data of nodes from each of the above groups or families. The EYES and the ScatterWeb nodes use relatively energy-saving microcontrollers. At the same time, also the least power-consuming memory is built into the ScatterWeb node. Looking at the PC communication interfaces in both tables, 2.4 and 2.5, newer nodes tend to have more sophisticated interfaces, such as USB and Bluetooth. The parameters of the radios used by the platforms in Tables 2.4 and 2.5 are presented in Table 2.6.

Some WSN radios offer the capability to tune the transmitted power, which naturally has

	Mica2	Telos	EYES	BTnode	ESB ScatterWeb
Developer	Crossbow	UC Berkeley	Uni. Twente	ETH Zuerich	FU Berlin
Year	2002	2004	2003	2004	2005
Microcontroller					
IC	ATMega128L	MSP430F1611	MSP430F149	ATMega128L	MSP430F149
Speed (MHz)	7.37	0.4–8	5	7.37	?
Architecture	8 bit RISC	16 bit RISC	16 bit RISC	8 bit RISC	16 bit RISC
Flash ROM (kB)	128	48	60	128	60
RAM (kB)	4	10	2	4	2
Power, active (mA)	8	4	3.2	8	3.2
per 1 MHz (mA)	1.09	0.5	0.64	1.09	?
Power, sleep (μA)	15	2	1.6	15	1.6
per 1 MHz (μA)	2.04	0.25	0.32	2.04	?
Wakeup time (μS)	180	6	6	180	6
Onboard memory					
IC	AT45DB041B	ST M25P80	ST M25P40	62S2048U	MC 24LC64
Type	Flash	Flash	Flash	SRAM	EEPROM
Non-volatile	yes	yes	yes	no	yes
Size (kB)	512	1024	512	240	64
Power, idle (μW)	5	150	150	?	0.03
per 100 kB (μW)	0.98	14.65	29.3	?	0.05
Power, read (mW)	10	12	12	?	0.15
per 100 kB (mW)	1.95	1.17	2.34	?	0.23
Power, write (mW)	37	45	45	?	0.3
per 100 kB (mW)	7.23	4.39	8.79	?	0.47
Radio	CC1000	CC2420	TR1001	CC1000	TR1001
Interface					
Extension pins	51	16	14	55	24
PC Communication	RS232	USB	RS232/ JTAG	Bluetooth/ JTAG	RS232/ JTAG
Integrated sensors	no	yes	no	no	yes
Recommended OS	TinyOS	TinyOS	PeerOS	none (TinyOS)	none (TinyOS)
Size (mm $\times$ mm)	32×58	32×65	$\approx 32 \times 92$	32×58	$\approx 45 \times 54$

Table 2.5: Comparison of widely-used WSN nodes. Data source: [15]

	CC1000	CC2420	TR1000	TR1001
Manufacturer	Chipcon (now Texas Instruments)		RF Monolithics	
Max data rate (kbps)	76.8	250	115	115
RX power (mA)	9.6	18.8	3.8	3.8
TX power (mA/dBm)	27.7/10	17.4/0	12/1.5	12/0
Powerdown power (μA)	0.2	20	0.7	0.7
Modulation	FSK	O-QPSK	OOK/ASK	OOK/ASK

Table 2.6: Comparison of widely-used wireless sensor network radios. Data sources: [15, 38, 39, 116, 121, 122]

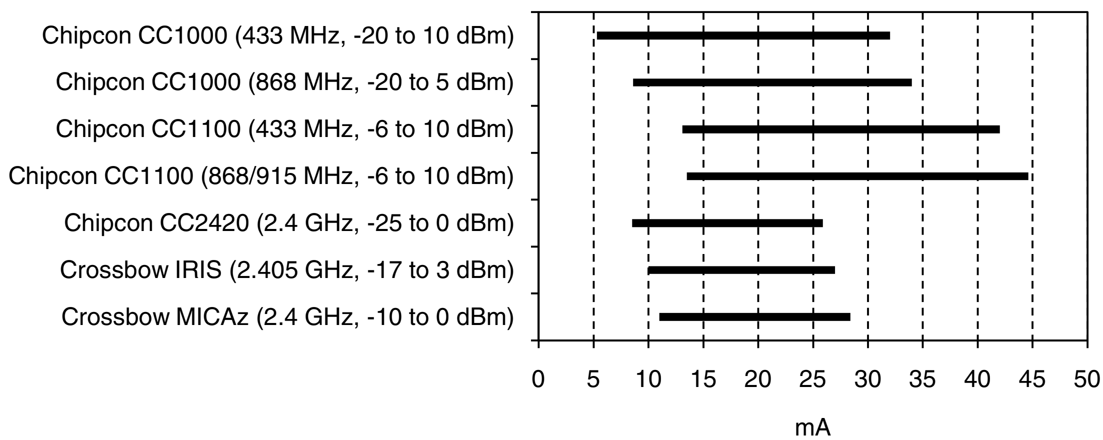

Figure 2.6: Power consumption of common tunable wireless sensor network radios. Data sources: [38, 39, 43, 44, 121, 122]

an effect on their overall energy consumption, as depicted in Figure 2.6. The horizontal bars represent the spectrum of energy consumed by the radios between (and including) the lowest and the highest output power setting, which is identified in the captions on the left.

Examining the data, it is evident that there is always a basic amount of power needed to run the radio electronics constituting a basic, always-present overhead, so that there is only a weak correlation between consumed and transmitted power. To provide an example, the Chipcon CC1000 transceiver (433 MHz) draws 5.3 mA, when tuned to 0.01 mW (i.e. -20 dBm) of output power, its lowest setting. In contrast, at its highest setting, 10 mW of output power (i.e. 10 dBm), it draws 26.7 mA. Hence, although tuned to only a thousandth of the output power in its lowest setting, the radio still consumes roughly 20 % of the power consumed in the highest setting.

Given the broad spectrum of challenges in WSNs, many different, sometimes exotic, node designs have been devised, such as the *SunSPOT* nodes (depicted in Figure 2.5 (b)). Manufactured by Sun Microsystems, they run a fully capable J2ME CLDC 1.1 Java virtual machine [139]. Other node designs are, for example, driven by the development of new operating systems, such as the SNoW [15] running the newly developed *SmartOS*. An overview of current and past WSN node designs can be found in [59].

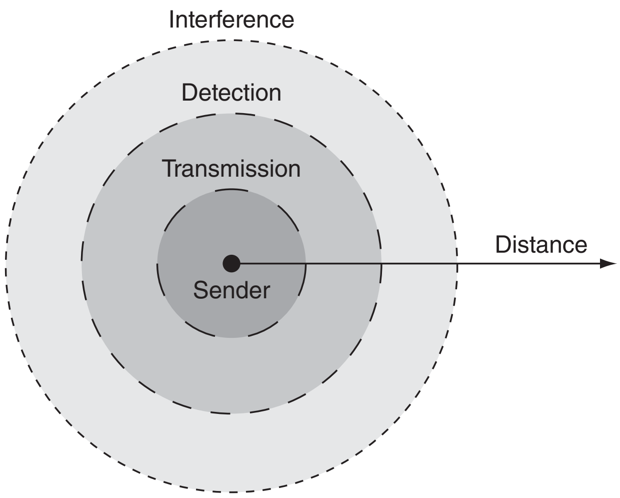

Figure 2.7: Communication zones around a sender

2.1.5 Communication

When designing protocols for WSNs, it is important to take the wireless medium into consideration, since it fundamentally differs from the wired being, for instance, much less reliable. This subsection provides a short summary of the key properties of wireless communication.

2.1.5.1 Transmission Zones

In an ideal model without obstacles, with sender and receiver situated in a vacuum, three concentric zones surround the sender, as depicted in Figure 2.7 [132]:

Transmission zone This is the innermost zone. A successful transmission to a receiver in this zone is possible. Naturally, it is detectable by other receivers in this zone and it also disturbs other signals.

Detection zone Within this zone, a receiver is able to detect the transmission originating at the sender, but the error rates are too high to receive valid data. At the same time, a transmission also disturbs other signals in this range.

Interference zone Inside this zone, a transmission originating at the sender interferes with other transmissions, although it is not detectable.

However, it is important to remark that the above model serves only well to form a rough impression of the real communication properties. To provide a more precise description, it should be mentioned that there are no clear borders between the above zones. For illustration, the model for a transmission zone of the Berkeley Mica mote is depicted in Figure 2.8, based on the data from [156]. In the model, the probability of a successful reception of a packet depends on the location of the receiver relative to the sender.

2.1.5.2 Path Loss

Assuming a sender with an ideal, omnidirectional antenna as a point in space, the signal generated by this antenna moves away from it as a wave with a spherical shape. The sphere grows as

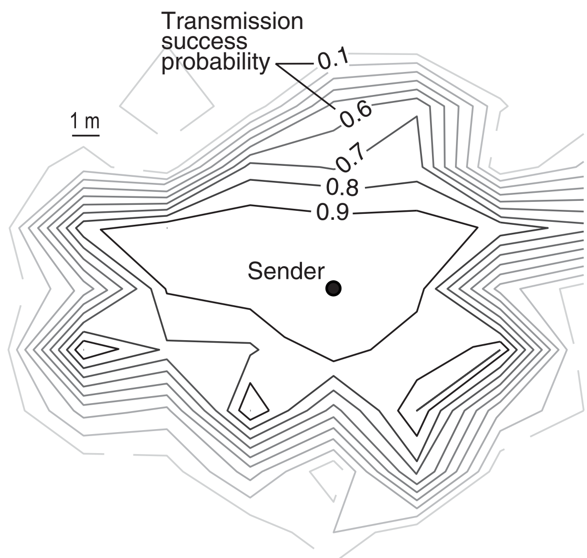

Figure 2.8: Probabilistic model of a transmission zone by Woo et al. [156]

the signal propagates. Therefore, with distance d increasing, the surface s of the sphere grows according to $s = 4\pi d^2$. The received power is even $\frac{1}{d^\gamma}$, with path loss exponent $\gamma = 2$.

In the real world, depending on the environment and the obstacles situated in the propagation area, the path loss exponent is often considerably higher. For a fuller discussion of these matters, the interested reader is referred to the article by Sohrabi et al. [134].

2.1.5.3 Environmental Effects

Depending on the properties of the area in which sender and receiver are situated, further effects can be observed, such as the following, which are also illustrated in Figure 2.9:

Blocking The propagation of a signal is blocked by an obstacle. This effect is also known as shadowing.

Reflection A signal is diverted into another direction by an object, which is large compared to the wavelength, enabling sender and receiver to communicate without line-of-sight. Typically, the object absorbs some of the signal's power.

Refraction Signal waves are refracted, since their velocity depends on the medium through which they travel. Thus, for example, waves entering a denser medium are bent towards this medium.

Scattering This effect can be observed, when the size of the object the waves collide with is in the order of a wavelength or less. Scattering results in a pattern with varying signal strength, depending on the receiver's location.

Diffraction The signal waves are deflected at an edge of a large object and propagated into different directions. Similar to scattering, diffraction also leads to varying signal strength patterns depending on the location of the receiver.

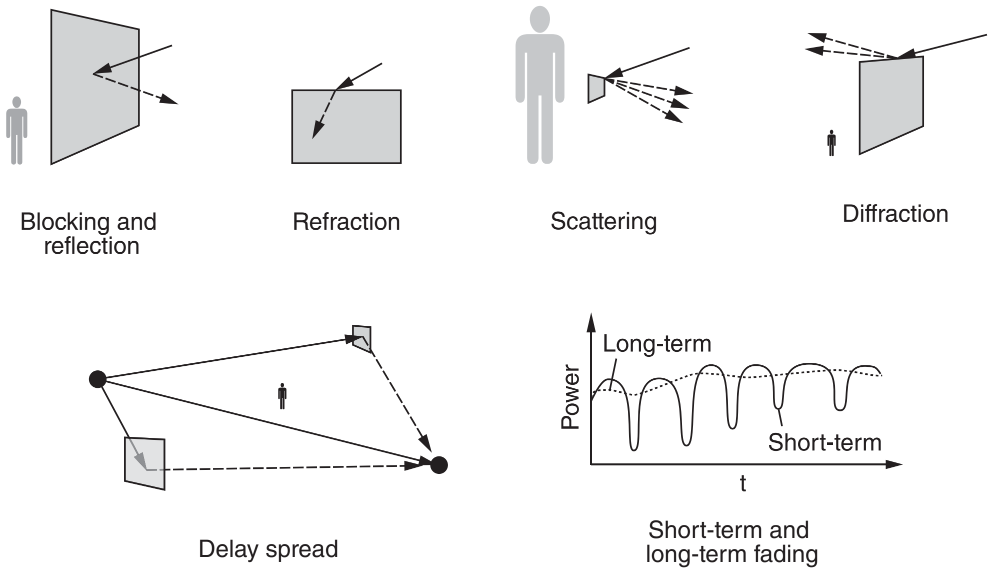

Figure 2.9: Environmental effects on signal propagation, assuming a wave length in the order of tens of cm

Delay spread Due to multi-path propagation, the original signal arrives at the receiver at different points in time. The effect may lead to *intersymbol interference*: the energy representing one symbol arrives at a time when another signal is expected to be received.

Short-term fading Fluctuations in received power caused by the fast movement of sender or receiver. The channel characteristics change rapidly, given the above effects.

Long-term fading Variations in received power caused by the slow movement of sender or receiver. The channel characteristics change gradually, given the above effects.

The above overview assumes a wavelength in the order of tens of cm, which is typical for WSNs. A comprehensive description of these and further effects on signal propagation is provided by [132].

2.1.5.4 Multi-Hopping

Given the typical application areas and hardware properties of WSNs, multi-hopping has become a widely-used technique.

In a 1-hop network, there is no need for topology control, since all nodes are members of the same clique. Therefore, it is interesting and legitimate to ask: why is a multi-hop network created, incurring the need for topology control, if it appears much more straightforward to use a 1-hop topology?

The following reasons can be cited for preferring multi-hop over single-hop WSNs:

Obstacles Real WSN application scenarios contain a vast range of possible obstacles, such as rocks, hills, dense vegetation, walls out of reinforced concrete, or machinery. Among other effects already described above, obstacles can block or attenuate the signal. Therefore, multi-hopping can be used to route communication around them.

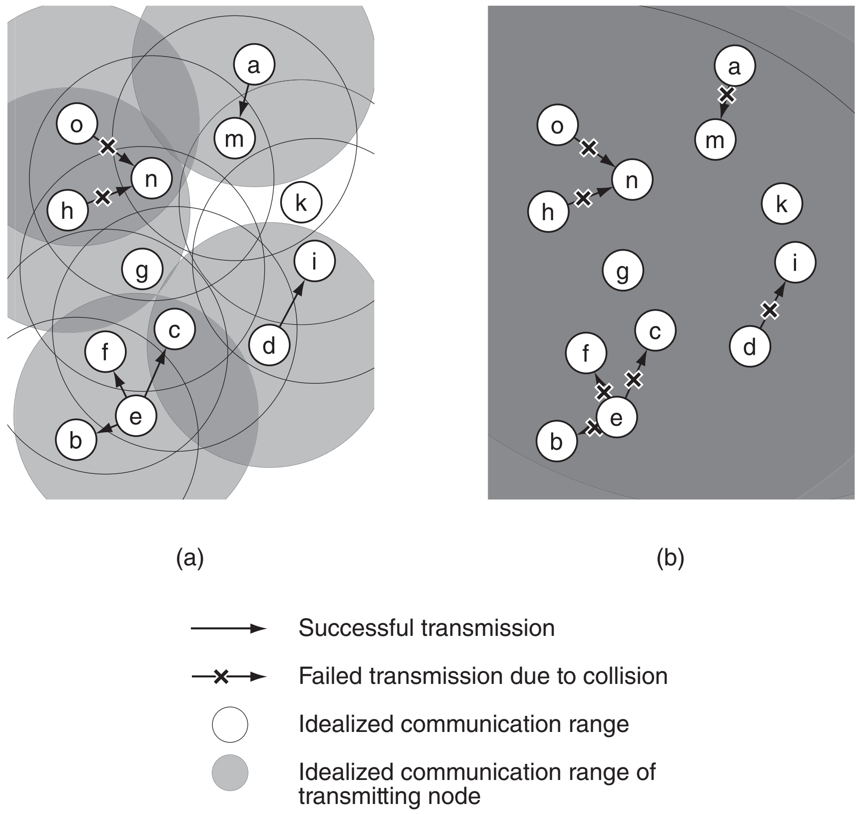

Figure 2.10: Snapshot of communication in a multi- (a) and a single-hop network (b)

Reduced contention, increased capacity Assuming a large WSN with 100s or 1000s of nodes, such as, for example, described in [9], in a 1-hop network, communication patterns typically encountered in WSNs could not be realized due to the extremely high amount of contention for the medium. In a multi-hop network, in contrast, communication can take place in parallel and only neighboring nodes within each other's transmission range compete for the medium.

Figure 2.10 illustrates this issue. It shows a multi- (a) and single-hop (b) network consisting of 13 nodes, as well as, the nodes' idealized communication ranges. Arrows between the nodes depict transmissions at the current time. In the multi-hop case (a), the transmissions from e to b,c,f, a to m, and d to i run successfully in parallel. Only the transmissions of h and o collide, since n receives both of them at the same time. In contrast, in the single-hop case depicted in Figure 2.10 (b), all of the transmissions fail due to collisions, as the entire network is a clique.

Physical properties Whether multi-hopping *by itself*—without considering the above aspects—saves, or even consumes more energy, highly depends on the distance between sender and receiver, the concrete hardware, and the properties of the environment. The popular belief that multi-hopping leads to reduced power consumption is based on the fact that signal power decreases quadratically with increasing distance from the sender—assuming the basic (and also optimistic) model in which the signal is imagined as an expanding sphere around the sender (as described in this section). In principle, this conception is correct. However, it does not take into account the non-radiated energy consumed by radios, as

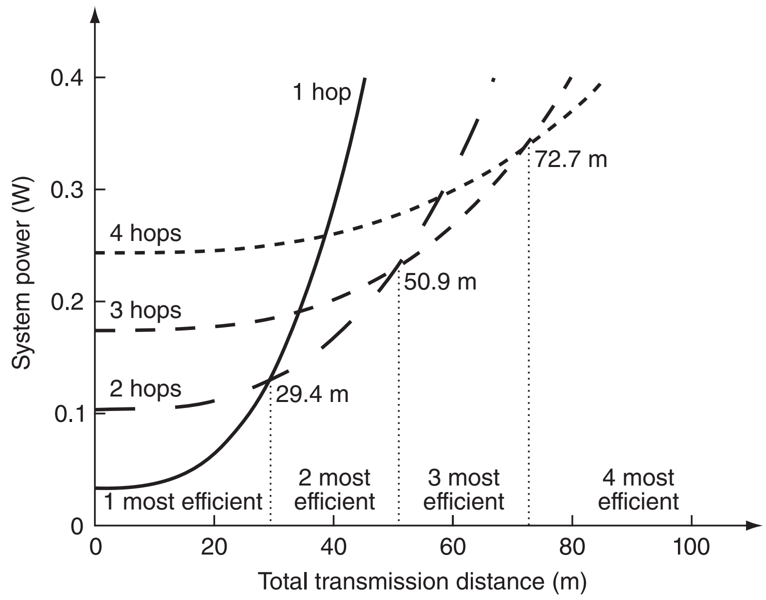

Figure 2.11: System power as a function of transmission distance for one, two, three, and four hops. Data source: [30]

depicted in Figure 2.6 (see Section 2.1.4), which supports the argument that a lower amount of hops saves energy.

In contrast, an argument *for* increasing the number of hops is that the path loss exponent encountered in the real world is considerably higher than 2. Merrill, Sohrabi, et al. [101, 134] demonstrated in real-world experiments that typical path loss exponents are rather a multiple of 2 on average for likely WSN scenarios: 3.0 in a concrete and steel parking structure on the UCLA campus (*car park management* application from Section 2.1.2), 3.5 on top of a wall made out of crushed limestone (*noise pollution* application), 3.6 in tall grassy fields with few tall bushes, and even 5.0 in a bamboo jungle [134] (both *habitat monitoring* applications).

Chandrakasan, Min, and their colleagues [30, 105] shed light on this issue, producing the data depicted in Figure 2.11. Using representative power consumption parameters for commodity of the shelf-based WSN nodes, the figure shows what transmitted and received power is needed to communicate over a particular transmission distance using one, two, three, or four hops. Obviously, a *general* statement pro or contra multi-hop can only be wrong. Until 29.4 m of distance, the 1-hop communication is the most efficient. But since increasing distance leads to an excessive path loss, the 2-hop communication becomes the most efficient beyond 29.4 m until 50.9 m. The effect repeats, so that between 50.9 m and 72.7 m, three, and beyond 72.7 m four hops are most efficient. In summary, this leads to the conclusion that there exists an optimal number of hops given a certain combination of transmission distance, radio hardware, and environmental conditions.

In [30, 87, 105] further information on the efficiency of multi-hopping can be found.

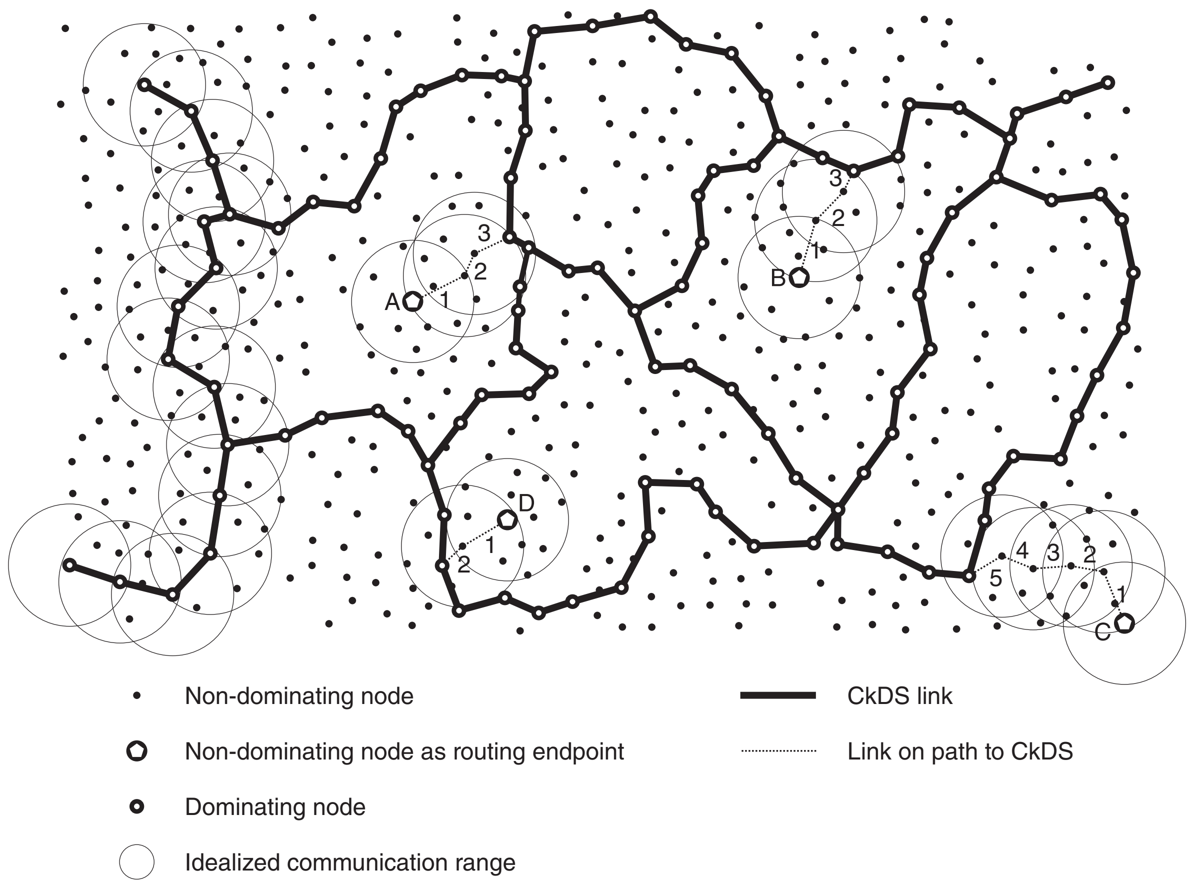

Figure 2.12: A connected 5-hop dominating set with example communication endpoints *A*, *B*, *C*, and *D*

2.2 Motivation for CkDS Formation

Figure 2.12 depicts a connected 5-hop dominating set, i.e. a CkDS with $k = 5$, in a WSN. For a definition of a CkDS refer to Chapter 1 (Definition 1 on page 1). In Figure 2.12, dominating nodes are represented by larger circles and connected using bold lines. For some of the dominating nodes (the arbitrary selection only serves the purpose of illustration) on the left side of the network, idealized communication ranges are depicted. By convention, a node that is situated in the center of an idealized communication range, is able to communicate with all other nodes whose center is within the boundaries of the same range.

The fact that the depicted dominating set in Figure 2.12 is a CkDS with $k = 5$ is illustrated using four nodes, *A*–*D*: Nodes *A* and *B* are three hops away, node *D* is two hops away from the CkDS. $k = 5$, since the shortest distance between *C* and its closest dominating node is 5, and there is no greater shortest distance between any non-dominating node and its closest dominating node in the network.

In WSNs, CkDS are constructed for various reasons, which are described in the next subsections. Finally, this section is concluded with a discussion on why to prefer CkDS over CDS.

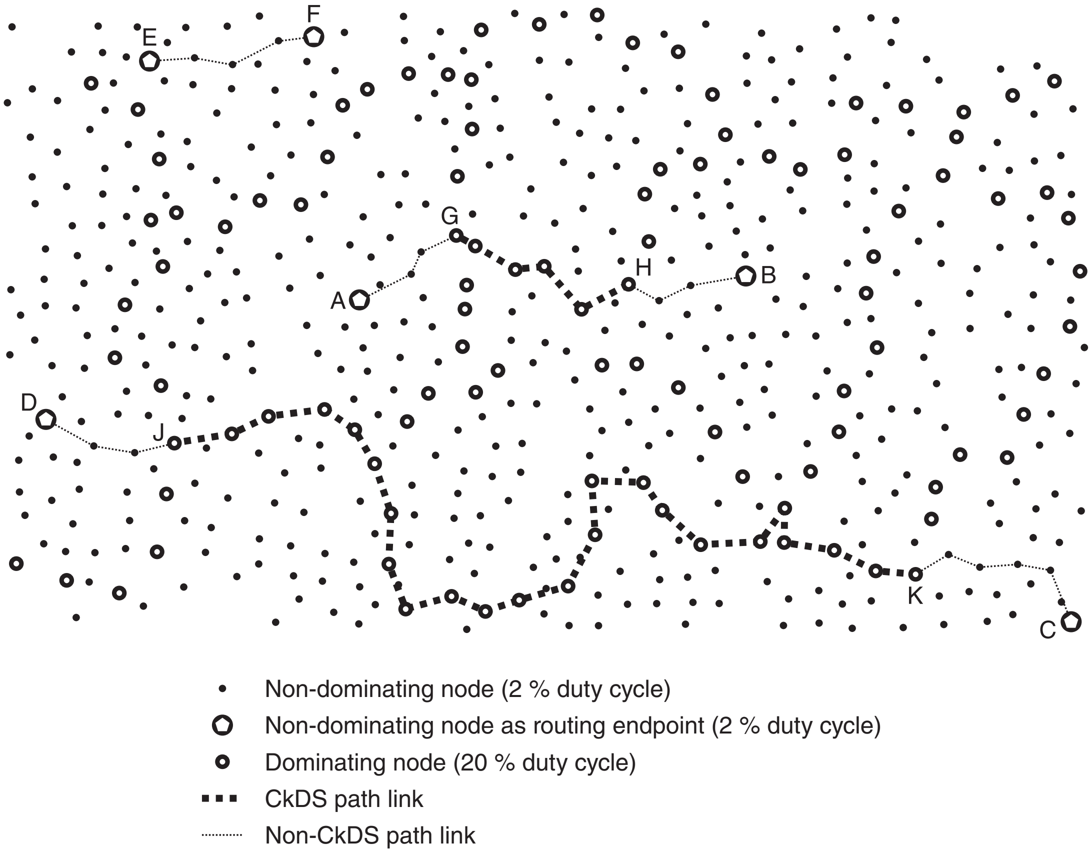

Figure 2.13: Long-range communication in a CkDS with $k = 5$ between routing endpoints A and B, C and D, E and F

2.2.1 Reduction of Overhearing, Idle Listening, and Collisions

Depending on the density of nodes in a WSN, not all of them are needed, if the purpose of the network is, for example, to ensure a certain amount or coverage or to enable long-range communication. With increasing node density, the risk of overhearing, idle listening, and collisions grows considerably, as shown in [167]. This can be explained by a higher amount of nodes competing for the medium at the same time. By creating a CkDS, as depicted in Figure 2.12, topology control eliminates nodes from the network topology which are not necessary to achieve a defined amount of connectivity but may lead to an increased occurrence of overhearing, idle listening, and collisions.

In Figure 2.13, a possible communication scenario using a CkDS is illustrated. The routing paths between node pairs $\{A,B\}$, $\{C,D\}$, and $\{E,F\}$ are depicted by dotted lines. The pair $\{E,F\}$ communicates over a short range directly, without using the CkDS. In contrast, pairs $\{A,B\}$ and $\{C,D\}$ communicate over a long-range employing CkDS links, to reduce the amount of the discussed effects.

Typically, in CkDS with larger k, such as in the example from Figure 2.12, most dominating nodes have between 2 and 3 dominating neighbors, i.e. their *dominating degree* is $\in [2,3]$. This is not only evident from the figure, but confirmed empirically in the results chapter of this thesis (see Section 5.2.7). If not using a CkDS, the degree of nodes is mostly on the order of 5 to 10 even in sparse networks, which can be observed in Figure 2.15 and was as well confirmed

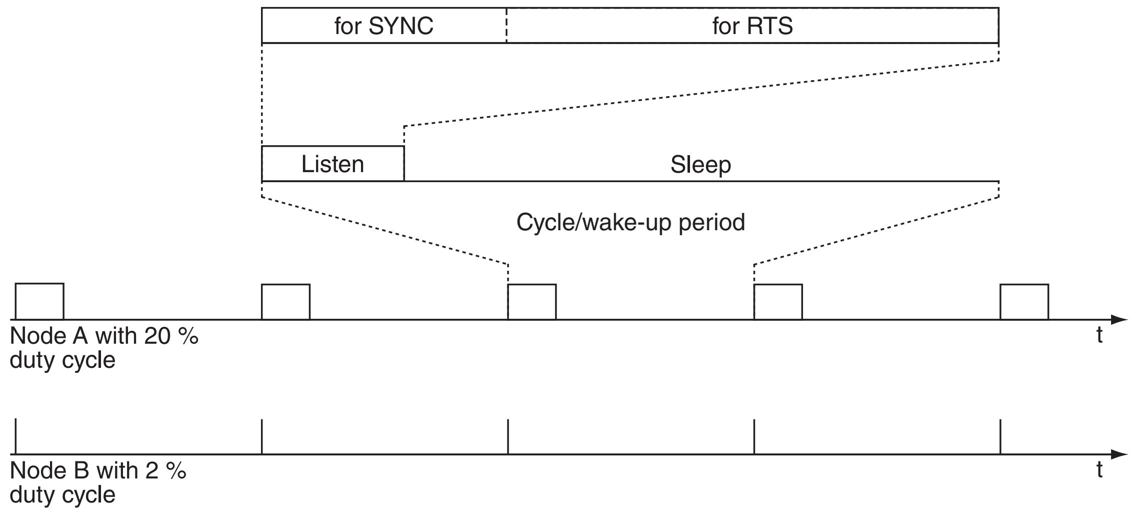

Figure 2.14: Duty cycling in S-MAC [167]

empirically. Moreover, the dominating degree is also the reason why CkDS are preferred over CDS (i.e. CkDS with $k = 1$): in CDS, the dominating degree and thus also the contention for the medium is much higher than in CkDS (with $k \geq 2$, assuming that most of the communication takes place on top of the dominating set).

2.2.2 Lower Duty Cycles and Delays

Many state-of-the-art medium access control (MAC) protocols for WSNs, such as S-MAC [167] or TRAMA [119], use some type of *duty cycling* to save energy. Each of the *cycles* or *wake-up periods* is divided into a *listen* and a *sleep period*, as depicted for S-MAC in Figure 2.14. During the sleep period, the radio changes into powerdown mode (if no communication is pending), which consumes by *several* orders of magnitude less energy, as described in Section 2.1.4 and shown in Table 2.6.

In S-MAC, for example, a listen period is further subdivided into periods in which the nodes listen to different frame types, as also shown in Figure 2.14: *synchronization* (SYNC) and *request to send* (RTS). A *duty cycle* is defined as the ratio between the length of the listen period and the length of the cycle [87]. Naturally, with lower duty cycles, the amount of energy consumed by the radio decreases drastically.

In large-scale WSNs, such as described in [9], with hundreds or thousands of nodes, long-distance communication can be expected to constitute a major portion of the overall traffic. Without using topology control, all nodes could be switched to a low duty cycle of, for instance, 3 %. However, this would incur high delays, since nodes would be sleeping for 97 % of the time (in case there is no pending communication). So at each hop, a packet would have to wait up to 97 % of the cycle (or even more, in case it e.g. does not win the competition for the medium), until being able to reach the next hop. At the same time, it has only up to 3 % of the cycle to win the medium. When routes consist of tens of hops, the per-hop delays sum up to a considerable total delay. Moreover, note that many WSN MAC protocols, including [119, 167], use synchronization to enable the nodes to wake up at the same time given a certain amount of clock drift. Evidently, the lower the duty cycles become, the greater the need for synchronization can be expected.

Employing topology control, long-distance communication can be concentrated on top of the CkDS, using an appropriate routing protocol, and non-dominating nodes can switch to much lower duty cycles to save energy. This way, only the short paths towards the CkDS incur the undesirable delays known from networks without topology control, while the paths on top of the CkDS exhibit only minimal delays. Non-dominating nodes constitute a vast majority in the network, as shown empirically in Section 5.2.6 and Figure 5.7, so that adjusting their duty cycles represents a key approach to substantial energy savings. Figure 2.14 depicts the wake-up periods of nodes A and B with duty cycles of 20 % and 2 %. Moreover, since medium to high duty cycles can be used at dominating nodes, the need to synchronize their schedules can be eliminated.

To provide a concrete example, assume that all non-dominating nodes in Figure 2.13 run at a 2 % duty cycle, while all dominating nodes run at 20 %. The node pairs $\{A,B\}$, $\{C,D\}$, and $\{E,F\}$ communicate with each other using multi-hop routes. In the case of $\{C,D\}$, the packets are first routed from the communication endpoint C towards the CkDS using non-dominating nodes. Since, on this path, nodes run at a 2 % duty cycle, the typical, undesirable delay known from networks without topology control is incurred. On the other hand, non-dominating nodes, constituting a vast majority, can sleep for 98 % of the time. Thereafter, packets travel on top of the dominating nodes from K to J. Along this section of the path, nodes run at a relatively high duty cycle, so that the delay caused by this part of the path is minimal. Finally, the packets are routed from J to D along non-dominating nodes with a higher per-hop delay. If there is no reasonable path over the CkDS, as in the case of $\{E,F\}$, the underlying routing protocol should choose a path over non-dominating nodes, as illustrated in Figure 2.13.

2.2.3 Reduction of Broadcast Storm Problem

The *broadcast storm problem* consists of the following subproblems [108]:

Collisions and Contention As a first problem, *collisions* are highly probable during flooding, since, in most cases, there is no effective backoff mechanism in MAC protocols for broadcasts. For example, *carrier sense multiple access with collision avoidance* (CSMA/CA), used in IEEE 802.11 [91], specifies that a node has to start a backoff procedure directly after transmitting a message, or, when the node intends to transmit but the medium is busy and the previous backoff has been conducted. However, this mechanism is only helpful to a limited degree during network *flooding*, i.e. the network-wide propagation of broadcasts during the route discovery of most routing protocols, including AODV [115], multi-path AODV [99], DSDV [114], or even the more exotic protocols described in [28] or [63].

For an illustration of this issue, assume a scenario in which node U broadcasts a packet m, which is heard by several of U's neighbors. If the medium in which U is situated has been quiet for a sufficient amount of time, all of U's neighbors may have passed their backoff procedures. Naturally, there is also no RTS/CTS forwarding dialog in broadcast communication. Therefore, during the typical route discovery phase of routing protocols, all of U's neighbors, constituting the set Y, with $l = |Y|$, that have not yet been involved in discovery, retransmit an altered version of packet m at approximately the same time. Obviously this leads to a collision, which presents two serious drawbacks: First, the packet has been broadcasted by l nodes without any benefit. Second, the links between the members of Y and the nodes that could have received and made

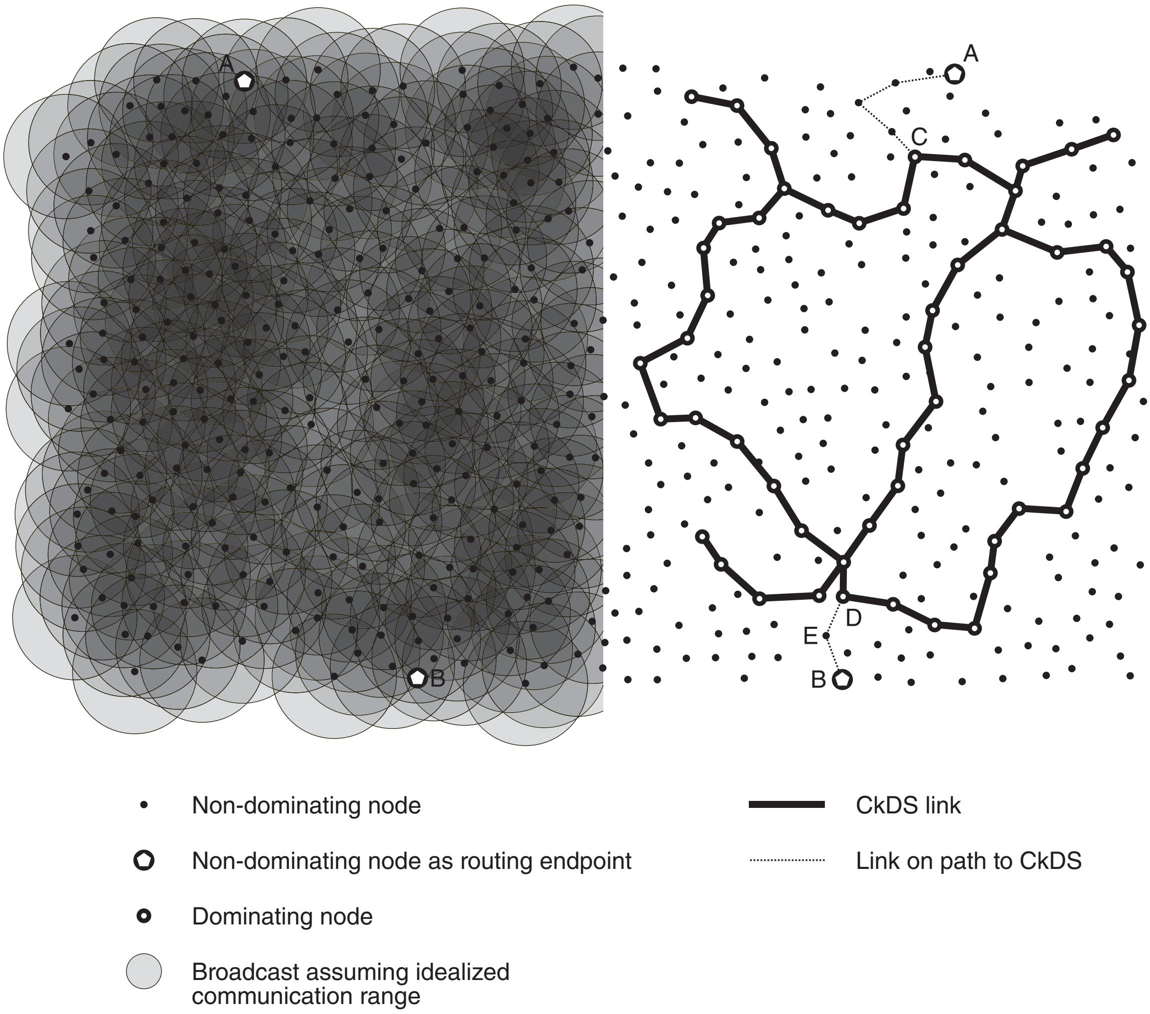

Figure 2.15: Route discovery without (left) and with topology control creating a CkDS (right)

use of their broadcasted packets are not set up. This may result in an inferior route quality or even in not finding a route towards the destination.

Redundancy and Overhead As discussed above, during flooding, all nodes in the network except the destination node send one broadcast, which is typically received by all of their neighbors (assuming ideal communication radii). In other words, let the average node degree be a and number of nodes be n, then, during flooding, the route request is transmitted $n-1$ times and received approximately $(n-1)\cdot a$ times. This is illustrated in the left half of Figure 2.15 for source A and destination B. Naturally, flooding has devastating effects on power consumption, since reception is nearly as expensive as transmission in terms of energy, as shown in Section 2.1.4 and Table 2.6.

Instead of the broadcast storm, a different approach can be chosen: Assume that each node X, willing to serve as routing endpoint, announces its existence to nearby CkDS nodes. This can be realized using, for instance, a random walk. The information maintained at the CkDS can be the path towards X. For a concrete example, consider the right half of the network depicted in Figure 2.15: Node B has initiated a random walk leading a packet towards node D. At node D, the route recorded by the packet, i.e. $\langle B, E, D \rangle$, is saved. Notice that a random walk consists only of unicasts, so that each transmitted packet is received exactly once (without considering further effects at the MAC layer).

Based on the above preparatory work, a discovery can be performed in a straightforward manner: The source node sends out a discovery packet, which reaches the CkDS using, for instance, a random walk. After arriving at the CkDS, the packet visits all of its nodes, finding the necessary information about the destination. To illustrate this approach, consider the example in the right half of Figure 2.15: Node A intends to find a route to node B. Therefore, A sends out a discovery packet on a random walk until reaching the CkDS. From this point on, the packet follows all links to other CkDS nodes, splitting appropriately. Having arrived at node D, the discovery packet notices the routing information towards B, and the discovery concludes. Using this procedure, there is not a single broadcast involved in the discovery. If assuming that 20 % of the nodes in the network belong to the CkDS, then the discovery has incurred only approximately $0.2 \cdot n + 6$ transmitted and the same number of received packets. Moreover, also the above-discussed problems concerning collisions and contention have been solved, since unicast communication is much more resistant to collisions, employing e.g. RTS/CTS, as discussed above, and much fewer nodes compete for the medium, if only CkDS nodes are used for long-distance packet propagation.

2.2.4 Approximation of Area Coverage

One of the most important and common functions of WSNs is area coverage. However, the concrete degree of area coverage desired might not be known at network setup, or it might vary depending on temporary conditions, such as mating seasons of observed animals in the habitat monitoring application (see Section 2.1.2). To cope with these challenges, the number and density of nodes needs to be adjusted during the operation time of the network. Unfortunately, it is a laborious task to change the *real* topology on-the-fly. Instead, topology control—and more concretely CkDS—can be utilized to provide the desired degree of area coverage by modifying the *virtual* topology.

Consider the example network presented in Figure 4.3. For the sake of simplicity (and without loss of generality), the nodes are positioned in a grid pattern at a diagonal distance d that equals the idealized sensing and communication range of a node. Thus, for the depicted CkDS $k = 11$. Evidently, all events with a diameter larger than approximately $2 \cdot k \cdot d$ would be detected by a node in the CkDS (assuming a convex shape of the event). The k-parameter can be adjusted according to the size of the event that needs to be identified. Moreover, tracking of mobile events that necessarily cross CkDS paths when relocating represents a further application.

2.2.5 Rendez-Vous Areas

Instead of flooding the entire network with a certain type of information, such as particular software updates, that is needed only by a few nodes, the CkDS can be employed as a rendez-vous area, in which information is disseminated or exchanged. Nodes interested in a certain type of information then only need to look on top of the CkDS. Figure 2.15 illustrates the difference between these two options, if assuming that node A is the information provider and node B the information consumer.

2.2.6 CkDS versus CDS

It should be remarked that the above arguments are not only valid for CkDS but to a certain degree also for CDS, which are basically CkDS with $k = 1$. This has been recognized early, so that many approaches for the construction of CDS, discussed in Section 3.1, have been proposed. However, the observed effects, such as a lower communication overhead and overall routing delays, are much more pronounced in CkDS, since for $k \geq 2$ they are typically much smaller than CDS. For example, there is a much lower probability of collisions in CkDS with higher k than in CDS, since the average degree of dominating nodes in a CkDS tends to be much lower than in CDS. Moreover, since traffic can be concentrated to a greater extent on dominating nodes, the duty cycles of non-dominating nodes can be scaled down to a higher degree, saving greater amounts of energy. Similarly, rendez-vous areas, whose size directly corresponds to the number of dominating nodes, can be much smaller than when employing CDS and, further, using CkDS their size can be adjusted. The same applies to the approximation of area coverage, which is easily adjustable in CkDS, but static in CDS.

For further discussions alluding to the topic of this section refer to [11, 47, 108, 130, 165, 167].

2.3 Summary

This chapter provided background information relating to the topic of this thesis. It started by discussing different aspects of WSNs and thereafter shed light on the motivation for the construction of CkDS.

The first sensor networks were deployed as early as in the 1940s. Only many decades later, wireless packet radio technology reached maturity in the 1970s and 1980s, paving the way for wireless sensor networks (WSNs) emerging in the 1990s. Currently, there is a broad range of applications for WSNs, including, for instance, flood warning, cold chain management, or vineyard monitoring. The commonality of most of these applications is that their topology is static. In the future, WSNs are expected to help realize the visions of ubiquitous computing and ambient intelligence.

A variety of WSN platforms has been devised. Characteristic of all of them is that transmitting and receiving one bit over the radio is more expensive than processing one instruction by at least an order of magnitude. Further, the properties of the wireless medium make the communication in WSNs more complex. For example, whether a transmission will be successful cannot be properly described by means of unit discs, but only using a two-dimensional reception probability landscape. To cope with the challenges presented by the wireless medium, multi-hopping is employed in many WSNs. It enables an efficient use of transmission power, helps to circumvent obstacles, and reduces the amount of contention for the medium.

There are several reasons to utilize CkDS in WSNs. First, by thinning out the topology, CkDS help reduce the amount of overhearing, idle listening, and collisions. When employing CkDS, it is further possible to concentrate communication traffic on top of the CkDS and assign lower duty cycles to non-CkDS nodes, saving substantial amounts of energy and reducing delays. Similarly, CkDS can alleviate the broadcast storm problem. A function that is of particular importance for WSNs, is area coverage. Given the ability to tune the dominating distance, CkDS even provide a flexibly adjustable amount of coverage. The CkDS can furthermore be used as a rendez-vous area, enabling an efficient implementation of publish-subscribe mechanisms. Finally, CkDS are preferred over CDS, since their size is usually smaller, amplifying the desired effects. Naturally, in contrast to CDS, their size can be adjusted in a straightforward manner.

3
State of the Art

Given the broad spectrum of applications for connected dominating sets (CDS) in wireless networks, approaches for their construction have been proposed as early as in the 1980s. While the first algorithms were predominantly centralized, with time, their successors became more and more distributed and parallel, improving scalability and structural issues, but also lowering overall cost.

The *minimum* connected dominating set problem was shown to be NP-complete in [64]. Therefore only approximations for it are feasible in larger wireless sensor networks. However, in most cases, it is not even desirable to find a minimum CDS, since such structures usually exhibit only poor reliability properties due to a lack of redundancy. On the contrary, a certain degree of redundancy is advantageous in the considered network class, so that the created structure does not become useless, when parts of it fail. Therefore, approaches for CDS construction typically aim to find some solution with considerably fewer dominating nodes than the total number of nodes in the network, while, at the same time, trying to minimize the cost incurred.

In order to enable a new range of applications, discussed in Section 2.2, in the recent years, several connected k-hop dominating set (CkDS) construction protocols have been proposed. An unsolved issue in all of the existing approaches is that their cost strongly depends on k, determining the maximum distance to the CkDS, and d, the average node degree.

This chapter is divided into two sections: the first describes the state-of-the-art CDS approaches, while the second focuses on CkDS approaches. I chose this procedure, since the approaches for the construction of CDS are the natural predecessors of the CkDS approaches. Moreover, the techniques, used for CkDS construction in the presented approaches, were borrowed from their CDS predecessors.

It should be mentioned at this point that, in the description of the state of the art in this chapter, I aimed at finding a good trade-off between readability, achieved by describing the algorithms and protocols with normal language, and precision, achieved using a more formal notation.

Note that not all of the approaches presented clearly fall into one category, since they use multiple methods at the same time, such as the greedy MPRS-based protocol from [2]. In such a case, that approach was included in a category which matches its predominant method.

3.1 Connected Dominating Sets

A connected dominating set (CDS) is defined as follows:

Definition 2 *Assuming an undirected, connected graph $G = (V, E)$ (according to Definition 13 on page 67), a CDS $D_{CDS} \subseteq G$ satisfies the following properties:*

1. *D_{CDS} is connected (using Definition 12 on page 67).*
2. *For each vertex $v \in G, \notin D_{CDS}$, there exists an edge (v, u), with $u \in G, \in D_{CDS}$.*

For instance, the set of black vertices in Figure 3.1 (e) constitutes a CDS.

In this section, I first present centralized solutions for the CDS problem, since, in many cases, the distributed approaches, presented as second, can be considered alterations of their centralized predecessors.

3.1.1 Centralized Construction

Several centralized solutions for the CDS problem have been proposed for wireless sensor and ad hoc networks. In this context, *centralized* means that the algorithm possesses the knowledge of the complete network graph. Naturally, in a real-world deployment, this implies that such a knowledge has to be gathered over multiple hops from the entire network. While this procedure may be satisfactory for small networks, it is entirely unsuitable for larger networks, since being error-prone and incurring high costs in terms of communication. Moreover, in such a case, the network relies on one node that runs the CDS algorithm based on the gathered data, representing a single point of failure. The centralized solutions discussed in this subsection subdivide into greedy, Steiner tree-, WCDS-, and pruning-based methods.

3.1.1.1 Greedy Construction

The greedy solution proposed by Guha and Khuller [70] basically works by growing a tree starting at the vertex with the maximum degree v_m. After the tree has been constructed and contains all vertices in the graph, all non-leaf vertices are selected to become members of the dominating set.

To build the tree, initially, all vertices are colored white. The algorithm starts by coloring v_m black. When a vertex is colored black, adjacent vertices would be colored gray (see Definition 15 on page 67). Then, from the set of gray vertices S_g, a node v_s with the highest *yield* is selected (operation o_1) and colored black. The yield for a node v_s is the total number of vertices which are colored gray after coloring v_s black. The algorithm stops when all vertices are colored either black or gray. Then, the set of black vertices constitutes a CDS.

The authors present a modification to this algorithm by introducing a new operation (o_2): scanning a pair of adjacent vertices v_g and v_w, such that v_g is gray and v_w white. If o_2 is chosen, this means that first v_g and then v_w are colored black. After coloring a vertex black, the modified algorithm selects either o_1 or o_2 depending on which one has the highest yield.

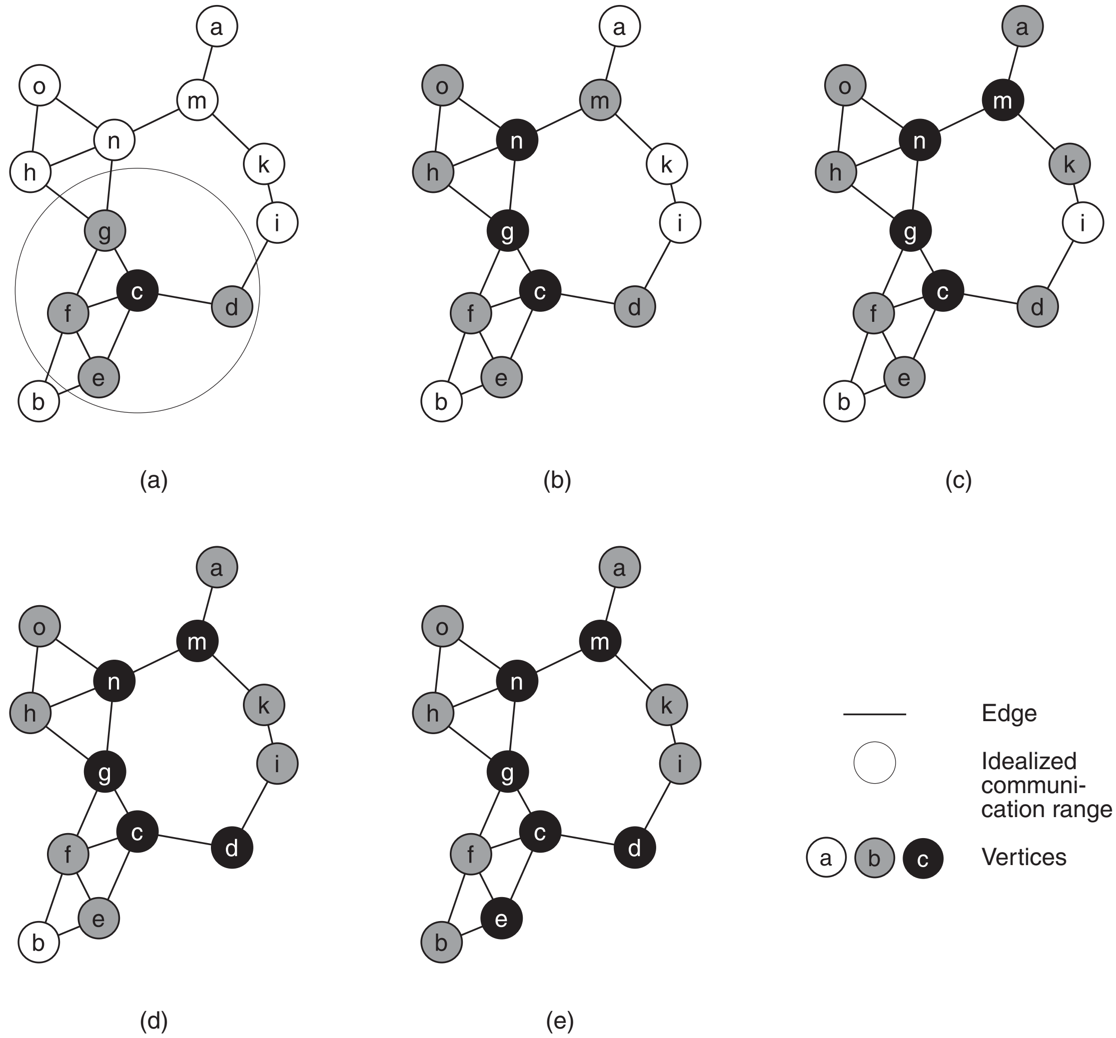

Figure 3.1: Modified centralized greedy algorithm by Guha and Khuller [70]

Example To illustrate the operation of the modified algorithm, consider the example in Figure 3.1, in which ties are broken using the lowest ID. Initially, all nodes are colored white. Given the fact that vertex c has the highest degree together with f and n but the lowest ID of $\{c, f, n\}$, it is selected as starting point for the tree construction. Consequently, in Figure 3.1 (a), c is colored black and its adjacent vertices (i.e. its 1-hop neighbors $\{d, e, f, g\}$, inside an idealized communication range) are colored gray. Considering operation o_1, the yield for d, e, and f is 1, for g 2, and for operation o_2 for (d, i) 1, for (g, h) 2, and for (g, n) 3. Therefore, o_2 is selected coloring first g and then n black, as well as, the adjacent vertices gray, as depicted in (b). Now with o_1, for d, e, and f the yield is 1 and for m 2, while, with o_2, for (d, i) or (m, a) it is 1, for (m, k) 2. As the yield with o_1 selecting m is 2 already, this operation and choice wins, resulting in (c). In (c) and (d) again o_1 is applied, selecting d first, as it has the lowest of the candidates' IDs. The final result is depicted in (e), with the CDS consisting of all nodes colored black.

In [127], a further greedy algorithm is presented. Moreover, Gupta et al. [71] propose another centralized greedy approach, which however focuses on area coverage instead of vertex coverage.

3.1.1.2 Steiner Tree-Based Construction

In [103], Min et al. propose a CDS construction algorithm, which is based on Steiner trees. In order to describe the algorithm, the following definitions are needed.

Definition 3 *In a graph* $G = (V,E)$*, a set* $S_I \subseteq V$ *is an* independent set*, if no two vertices from* S_I *are adjacent in G (using Definitions 13 and 15 on page 67).*

Definition 4 *An independent set is a* maximal independent set *(MIS)* $S_{MI} \subseteq V$ *in graph* $G = (V,E)$ *(according to Definition 13 on page 67), so that the addition of any further vertex* $v_f \in V$ *to* S_{MI} *would lead to the violation of the independence property (i.e.* S_{MI} *would cease being independent).*

Definition 5 *In a graph* $G = (V,E)$ *(according to Definition 13 on page 67), a* Steiner tree *for a given subset of vertices* $S_t \subseteq V$*, called set of terminals, is a tree with minimal size including* S_t*.*

Definition 6 *Given a graph* $G = (V,E)$ *(according to Definition 13 on page 67), a set of vertices constituting a Steiner tree* $S_{st} \subseteq V$*, and a set of vertices called terminals* $S_t \subseteq S_{st}, S_t \subseteq V$*, a vertex* v_{st} *is a* Steiner vertex*, if* $v_{st} \in S_{st}, v_{st} \notin S_t$*.*

The approach by Min et al. [103] is divided into two steps:

1. A maximal independent set S_{MI} is constructed using an existing algorithm. Note that every *maximal* independent set is always also a dominating set.
2. At the beginning of this step, all vertices in S_{MI} are colored black and all other vertices gray. The dominating set is then connected by approximating a Steiner tree in two stages:
   (a) While there is a gray vertex v_g adjacent to at least three black components, v_g changes its color to black. A *black component* is a connected component of the subgraph induced by black vertices.
   (b) While there exists a gray vertex v_g adjacent to at least two black components, v_g's color is changed to black.

Finally, all black nodes constitute the CDS, which is also a Steiner tree.

Example The operation of the above algorithm is illustrated in Figure 3.2. At the beginning of the second step, in (a), the members of a maximal independent set (MIS) are colored black, all other vertices gray. The MIS, at the same time, corresponds to the set of terminals, which are to be connected by the Steiner tree. Since o is the only gray vertex adjacent to (at least) three black components, namely d, k, and n, it is colored black, resulting in the state depicted in (b). Consequently, at this point, there are no more gray vertices which are adjacent to at least three black components. The only gray vertices adjacent to the two remaining components $\{d,k,n,o\}$ and f are c and g. As in this example the ties are broken using lowest ID, vertex

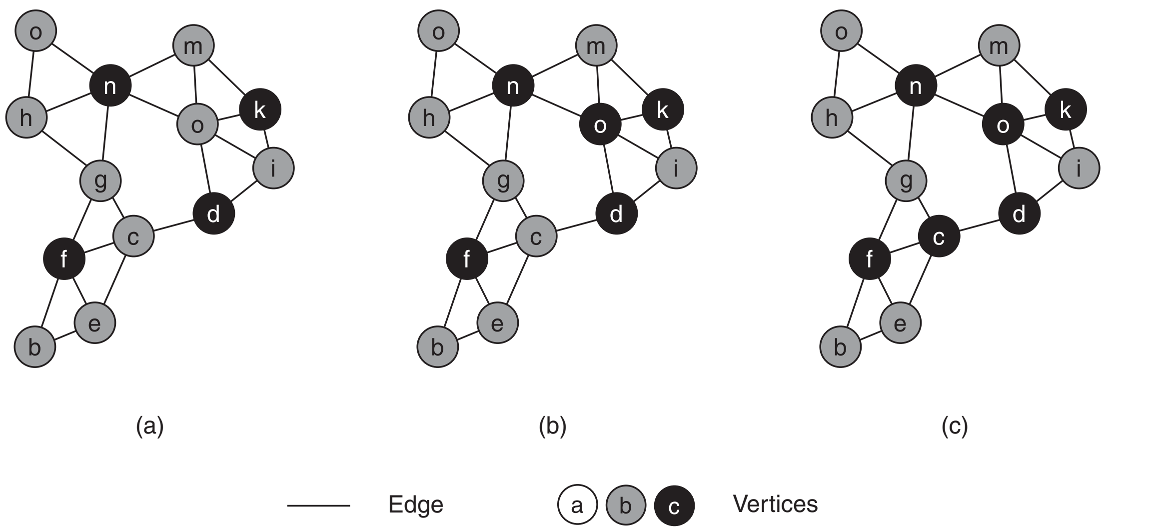

Figure 3.2: Centralized Steiner tree-based algorithm by Min et al. [103]

c is colored black, resulting in (c). Finally, no more gray vertices are adjacent to two or more black components (actually there exists only one black component $\{c,d,f,k,n,o\}$), so that the algorithm stops and yields the vertices colored black as a result in (c).

A further and partially similar Steiner-tree based centralized approach was presented by Guha and Khuller in [70]. It uses the notion of *pieces*, which are defined as single white vertices and as black connected components. In each step, the algorithm picks the node with the maximum, but nonzero, reduction of the number of pieces.

3.1.1.3 WCDS-Based Construction

The approach by Chen and Liestman [35] is based on the idea of finding a weakly-connected dominating set (WCDS), which implies a connected dominating set. A WCDS is defined as follows:

Definition 7 *Assume a set of vertices S_D in graph $G = (V,E)$, according to Definition 13 on page 67. For each vertex $v_d \in S_D$, add v_d and the set of all its adjacent vertices to the set S_{V1}. Further add all edges (u,w) with $u \in S_D$ or $w \in S_D$ to set S_{E1}. S_D is a* weakly-connected dominating set *(WCDS), if it is dominating (see Definition 16 on page 67), and if G's subgraph $G' = (S_{V1}, S_{E1})$ is connected (see Definition 12 on page 67).*

Further:

Definition 8 *G' from Definition 7 is a* weakly-induced subgraph *of S_D (from the same definition).*

Example In Figure 3.3 (e), the set of black vertices constitutes a WCDS. Further, the subgraph weakly-induced by the WCDS comprises the black and the gray vertices.

There are several arguments for using a WCDS to construct a CDS, as also discussed by the authors in [35]: Not only can the members of the WCDS be utilized as clusterheads and the other members of S_{V1} as connectors between them. In addition to that, WCDS can be smaller than CDS, and they result in fewer clusters, when employing the members of these sets as clusterheads. A drawback to this method is however that the derived CDS tends to be fairly large.

A centralized method for CDS construction, presented in [35] (algorithm 1), operates as follows: Initially all vertices are white. When the color of a vertex changes to black, the white vertices adjacent to it are colored gray. To further describe the algorithm, the notion of a *piece* needs to be introduced. A white vertex constitutes a *white piece*. In contrast, a *black piece* consists of a maximal set S_B of black vertices whose weakly-induced subgraph is connected and all gray vertices adjacent to S_B. *Improvement* for a candidate vertex v_c is defined as the number of distinct pieces that would be merged by coloring v_c black. In each iteration, the non-black vertex with the highest improvement is chosen to be colored black. The algorithm terminates when there is only one piece in the graph, or, in other words, if all vertices are either black or gray.

Example Consider the example depicted in Figure 3.3. Initially all vertices are white, as depicted in (a), so that each vertex is a piece by itself. Vertices h and a have, for instance, an improvement of 4 and 2. The vertices with the highest improvement, of 6, in the graph are vertices n and o. Breaking ties by lowest ID, vertex n is selected to be colored black, so that all adjacent white vertices are colored gray, resulting in the situation depicted in (b). As there is no more vertex with an improvement of 6, c is chosen to be colored black, since it is the lowest-ID vertex from the vertices with an improvement of 5, namely c and f; this results in (c). At this point, the highest-improvement, lowest-ID vertex is i (winning over o), with four pieces, so it is chosen to be colored black in (d). Finally, only vertices a and m remain with an improvement of 2. a, having the lower ID, wins, being colored black. The final WCDS consists of the black vertices in (e). Notice that, in the example, the obtained CDS, consisting of the weakly-induced subgraph of the WCDS, comprising the black and the gray vertices, is relatively large.

3.1.1.4 Pruning-Based Construction

Algorithms based on pruning are interesting, since they follow the opposite strategy of all previously described approaches: instead of building a set that is to become a CDS by adding nodes successively, they start with a set which includes all vertices from the graph and remove vertices consecutively until producing a CDS with the desired properties.

A centralized algorithm based on pruning is proposed in [24]. In the approach, D denotes the current CDS, while F is the set of fixed vertices. A vertex which is fixed cannot be removed from the CDS. At the start of the algorithm, all vertices are in D, and $F \subseteq D$. While not all vertices in D are fixed, the vertex v_m with the minimum degree is selected from $D \backslash F$. If the removal of v_m from D leads to the disconnection of D, v_m is added to F. Else, v_m is removed from D, and, if v_m has no adjacent vertices in F, its adjacent vertex with the highest degree v_h is added to F, in order to reduce the overall number of fixed vertices. When the algorithm terminates, $F = D$ is the resulting CDS.

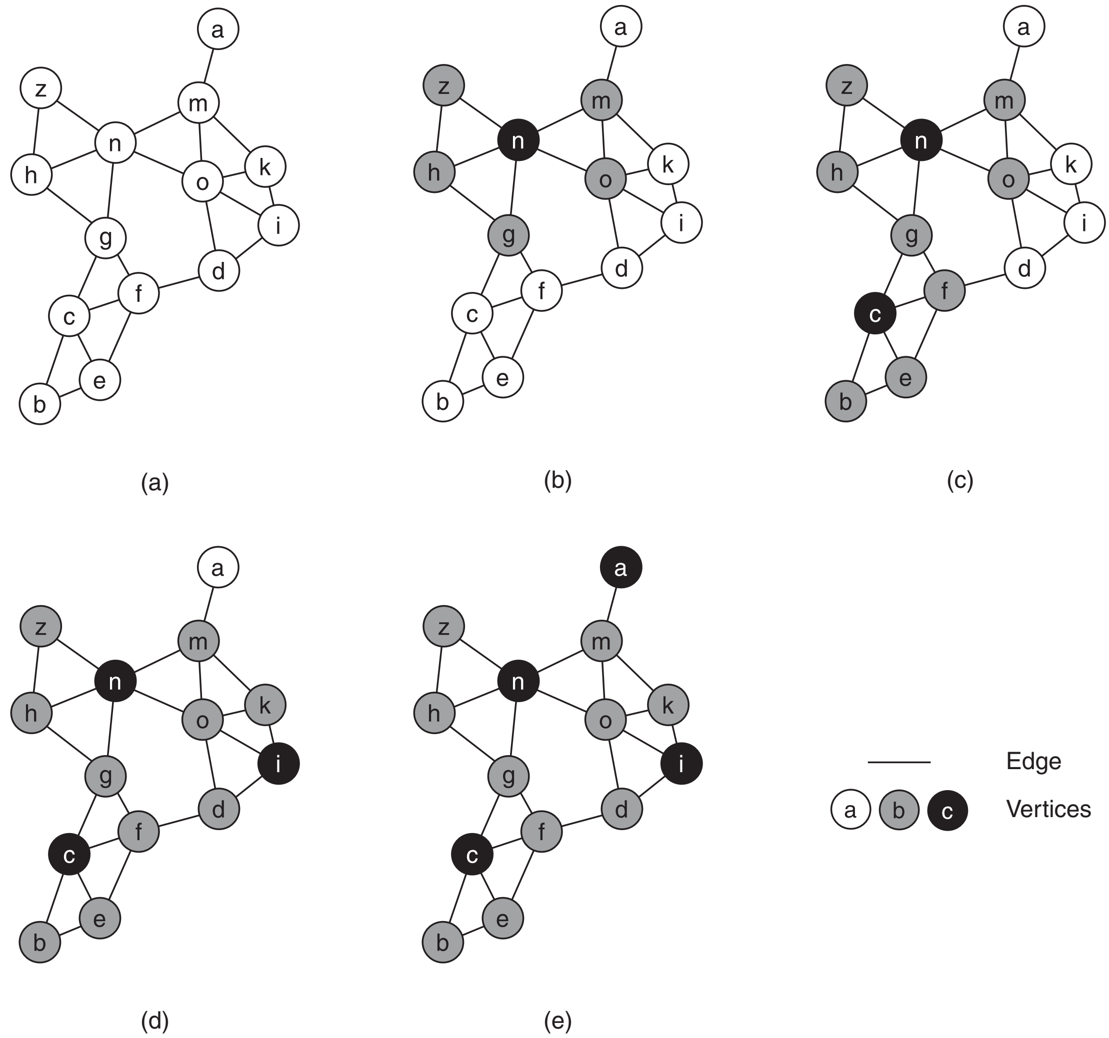

Figure 3.3: Centralized WCDS-based algorithm by Chen and Liestman [35]

3.1.2 Distributed Construction

Using centralized solutions is reasonable, when the entire network graph is known or the necessary information can be obtained cheaply by a node. However, if communication is highly expensive in terms of energy, as it is the case in WSNs, it is not feasible to transfer all needed information about the entire topology to a central point. Therefore, a wide range of approaches was proposed to solve the CDS problem in a distributed manner. Their categorization bears some resemblance to the categorization of centralized algorithms, since similar methods can be used in the distributed case. However, distribution enables but also necessitates new strategies. Therefore, besides greedy, Steiner tree-, WCDS-, and pruning-based distributed approaches, there are also maximal independent set- (MIS-), multi-point relay- (MPR-), and connected clustering-based methods.

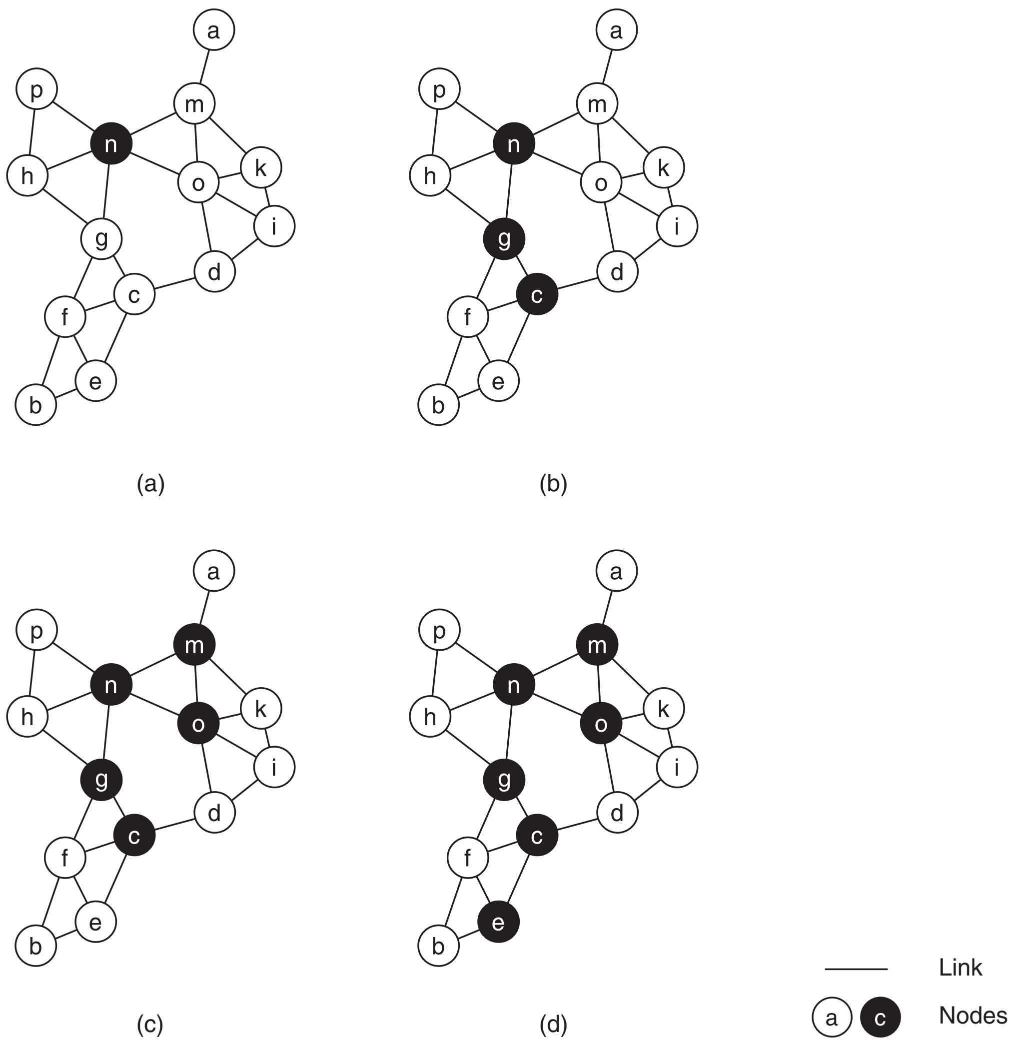

Figure 3.4: Distributed greedy protocol by Das et al. [49, 50]

3.1.2.1 Greedy Construction

The greedy approach presented by Das and colleagues [49, 50] is one of the first approaches to distributed CDS construction in wireless networks. It grows one dominating set fragment until it becomes a CDS. The process starts by adding the first node to the dominating set D, which is selected using a degree-based rating. Then, in each round, from all paths, or *extensions*, $p_1 = \langle v_d, v_1 \rangle$ and $p_2 = \langle v_d, v_1, v_2 \rangle$, with $v_d \in D$ and $v_1, v_2 \notin D$, the path with the maximum effective combined degree is added to D. For a path p, let $p.v(j)$ identify the jth node in p, $p.l$ the number of nodes in p, and $N_1^n(v)$, the non-dominating nodes in the 1-hop neighborhood of v. Then the *effective combined degree* (ECD) for a path p is defined as follows (assuming that members of p are treated as dominating):

$$ECD(p) = \Big| \bigcup_{i=2,\ldots,p.l} N_1^n(p.v(i)) \Big| \tag{3.1}$$

Example In Figure 3.4 (a), initially, node n is chosen over node o as first node to be added to D and thus colored black, given the fact that it has the highest degree in the network of 5 together with o but a lower ID than o (assuming a lower ID to break ties). At this point, depicted in (a), there are the following path candidates and ECD ratings: $\langle n,g \rangle$ (ECD of 3), $\langle n,h \rangle$ (2), $\langle n,m \rangle$ (3), $\langle n,o \rangle$ (4), $\langle n,g,c \rangle$ (4), $\langle n,g,f \rangle$ (4), $\langle n,h,p \rangle$ (1), $\langle n,p,h \rangle$ (1), $\langle n,m,a \rangle$ (2), $\langle n,m,k \rangle$ (3), $\langle n,m,o \rangle$ (4), $\langle n,o,i \rangle$ (3), $\langle n,o,k \rangle$ (3), and $\langle n,o,d \rangle$ (4). Breaking ties using lowest IDs, the path $\langle n,g,c \rangle$ wins over paths $\langle n,o \rangle$, $\langle n,g,f \rangle$, $\langle n,m,o \rangle$, and $\langle n,o,d \rangle$; nodes g and c are added to D and thus colored black in Subfigure (b). In (c), m and o were added to D, since being the lowest-ID path from the candidates with the ECD of 4, i.e. $\langle n,o \rangle$, $\langle n,m,o \rangle$, and $\langle n,o,m \rangle$. Finally, in (d), node e, with the highest ECD (of 2), was chosen, winning over f.

In [36], Cheng et al. proposed further distributed, greedy CDS construction protocols. Their first protocol grows a spanning tree greedily in a distributed manner, starting at the leader. All non-leaf nodes in the resulting tree constitute the CDS. In order to avoid leader election, their second method constructs a forest of trees, so that each tree is routed at a node which has the lowest ID among its 1-hop neighbors. Thereafter, the protocol connects the forest. The authors of [62] also propose to grow a tree to construct a CDS but using depth-first-search. A further distributed greedy protocol is presented in [71]. However, it focuses on area instead of node coverage.

3.1.2.2 Steiner Tree-Based Construction

In [103], the authors present a distributed version of their centralized Steiner Tree-based algorithm, which is discussed in Section 3.1.1.2; the Steiner tree is introduced in Definition 5 on page 36. The distributed version assumes that as starting point, a maximal independent set has been created, using an adequate state-of-the-art protocol, such as [27, 152]. Further, it requires that all nodes within a black component maintain a z-value, which serves as a common identifier for this component and initially corresponds to the nodes' IDs. A *black component* is a connected component of the subgraph induced by black nodes. In contrast, the y-value, used at the gray nodes, which are the 1-hop neighbors of all black nodes, reflects the number of black components adjacent to such a node. A gray node's *rank* is higher, if it has a larger y, while ties are broken using lower IDs. Two gray nodes are competitors, if being adjacent to each other or to the same black component.

A gray node v_w changes its color to black, if and only if ranked higher than each of its competitors. If this is the case, z-values within the new black component are updated to the lowest one of all z-values of the black components merged at this point. Further, all of v_w's competitors become competitors of every competitor of v_w. Note that, in order to realize this competition, information about the status of all gray nodes adjacent to a black component has to be exchanged within this component and the adjoining gray nodes.

3.1.2.3 WCDS-Based Construction

In [35], the authors present an approach based on their centralized algorithm described in Section 3.1.1.3. Similar to the centralized version, when a node is colored black, all its white 1-hop neighbors are colored gray. To further describe the protocol, the notion of a *piece* is used. A white node constitutes a *white piece*. In contrast, a *black piece* consists of a maximal set S_B of

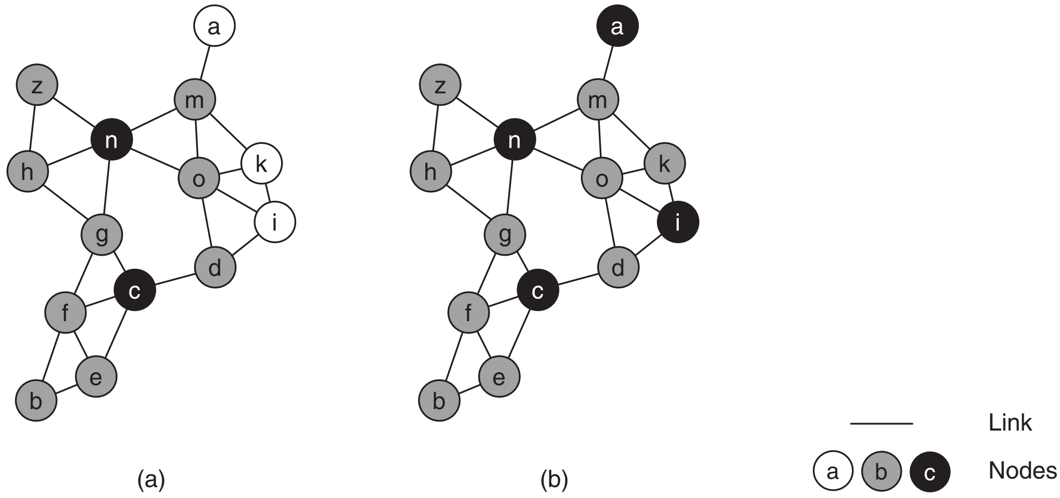

Figure 3.5: Distributed WCDS-based protocol by Chen and Liestman [35]

black nodes whose weakly-induced subgraph (see Definition 8 on page 37) is connected and all gray nodes adjacent to S_B. A node that changes its color or piece ID, informs its 1-hop neighbors. *Improvement* for a candidate node v_c is defined as the number of distinct pieces that would be merged by coloring v_c black.

The protocol starts with each node that has the highest improvement in its 1-hop neighborhood coloring itself black and setting its generation value to 1. A newly colored black node v_b

- ◇ informs its 1-hop neighbors of its new color using a status-change message, so that white 1-hop neighbors receiving this message color themselves gray;
- ◇ becomes the root of the black piece it is part of;
- ◇ floods its ID, which becomes the new piece ID, using a new-piece-ID message within its piece;
- ◇ floods a best-candidate-inquiry message through its piece, creating a broadcast tree.

Leaves of the broadcast tree rooted at v_b reply with their improvement values, which propagate towards v_b along the links of the tree. During this process, only values which are better than already encountered ones are forwarded. v_b then chooses the candidate v_c with the best improvement (ties broken adequately) and sends a please-color-black message to v_c including v_b's generation value. Upon reception of this message, v_c colors itself black and sets its generation field one greater the received value. Due to asynchrony, it is possible that multiple nodes announce becoming roots at the same time. In this case, only the highest encountered generation is further propagated. The above process continues until there is only one black piece.

Example The construction process is illustrated using the example in Figure 3.5. In (a), nodes c and n colored themselves black, since having had the highest improvement of 5 and 6 in their 1-hop neighborhoods and a lower ID than their competitors: c won over f, and n won over o. At this point, a and d exhibit an improvement of 2, while i, k, m, and o offer an improvement of 3.

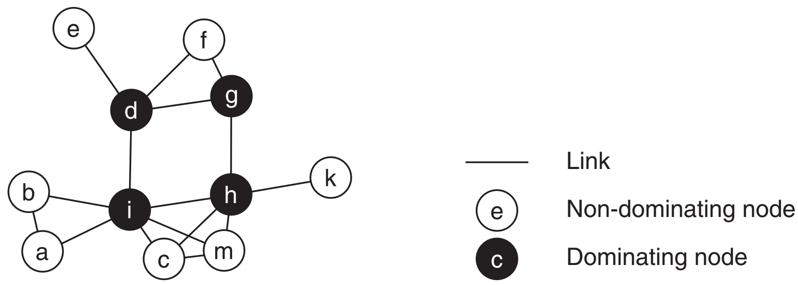

Figure 3.6: Distributed pruning-based protocol by Dai, Wu, and Li [45, 46, 163]

Breaking ties by lowest ID, in this example, first *i* colors itself black, so that *m*'s improvement drops to 2, as *k* becomes gray. Thereafter, *a* with an improvement of 2 follows, winning over *m*, resulting in (b).

Notice that this protocol [35] is somewhat similar to the distributed Steiner tree-based protocol by Min et al. [103] described above. One of the differences is that the Steiner tree-based approach uses a maximal independent set (MIS) as starting point, whereas this WCDS-based protocol first selects nodes with the highest improvement in their 1-hop neighborhood. Among other differences, the set of black nodes is augmented in a different manner: the Steiner tree-based approach only joins black components, while the WCDS-based protocol may join any combination of black and white pieces.

A further distributed method to construct CDS based on WCDS is presented in [112], focusing on minimizing the amount of collisions, latency, and redundancy while maximizing throughput.

3.1.2.4 Pruning-Based Construction

The first distributed and, at the same time, remarkably simple pruning protocol, which was proposed by Dai, Wu, and Li in [45, 46, 163], uses the following construction process, consisting of three steps:

1. *Initialization:* All nodes in the network are marked as non-dominating.

2. *Neighborhood information exchange:* Each node periodically sends a message including the set of its known 1-hop neighbors to its 1-hop neighborhood. When it receives such a message, it updates its own set accordingly.

3. *Marking:* Based on the information obtained in the previous step, each node v decides independently of its status. If v is aware of a pair of nodes u,w that is not connected, i.e. $(u,w) \notin E$ and $u,w \neq v$, v marks itself as dominating.

Example The result produced by the protocol in an example network is depicted in Figure 3.6. Node a does not mark itself as dominating, since the only pair of 1-hop neighbors, $\{b,i\}$, it knows, is connected. g however is aware of the pairs $\{d,h\}$ and $\{f,h\}$, which are not connected, and therefore marks itself as dominating.

Note that this protocol deviates from the classical notion of pruning as assumed by, for example, the centralized algorithm from [24] and the distributed CkDS protocol described in [165], since it does not start from a state at which all nodes are dominating. However, it can be considered some form of inverse pruning, in which each node does not add itself to the CDS, if it comes to the conclusion that it can be pruned (or removed) without compromising connectivity.

In [159], Wu presents an extension of the above protocol, reducing the size of the obtained CDS. To achieve this and to prolong the lifetime of the CDS, Wu and colleagues introduce two further rules in [162], considering node degree and energy levels. A similar extension is proposed in [138], which takes into account the degree of candidate nodes.

Butenko et al. present a different pruning-based distributed approach in [24], which selects nodes for removal from the CDS, after performing a test for disconnection. Ju et al. propose a protocol [81], which operates on heterogeneous networks, so that it tolerates the presence of nodes that are not capable or not intended to become members of the CDS.

3.1.2.5 Maximal Independent Set-Based Construction

Alzoubi et al. present a distributed protocol [7] based on maximal independent sets (MIS) (see Definition 4 on page 36), which is an enhanced version of their previous work described in [5, 6, 152]. Its underlying idea is motivated by the observation that a MIS is also an independent dominating set. The protocol is divided into two phases:

1. *Construction of a MIS*: If the number of (non-dominating) 1-hop candidate neighbors of node v with IDs lower than v's is 0, v becomes dominating, i.e. member of the dominating set D.

2. *Connection of the MIS to a CDS*: After each non-dominating node identifies dominating nodes in at most 2-hop distance, it broadcasts this information. Based on it, each dominating node $v_d \in D$ chooses a path of at most three hops to each dominator that is at most three hops away from itself and has a larger ID. Having made this choice, v_d notifies the members of the chosen paths, the *connectors*. All connectors constitute the set C. The resulting CDS S_{CD} is obtained by $S_{CD} = D \cup C$.

The approach works, since, by definition, any pair of nodes in a MIS is at least two hops away from each other. Moreover, any subset of a MIS is at most three hops away from the rest of the MIS. Note also that there are numerous internal data structures and message exchanges needed to implement the above protocol. However, for the sake of clarity, the above description concentrates on the underlying concept.

A further protocol, which uses a MIS to produce a CDS is proposed by Butenko et al. [23]. First, it computes a MIS and then uses a tree to connect it. Min et al. follow a similar principle in [104], by first finding a MIS with only 2-hop separations and then interconnecting it, employing a minimum spanning tree.

3.1.2.6 Multi-Point Relay-Based Construction

A further group of approaches originates from the research conducted to alleviate the broadcast storm problem (see Section 2.2.3) using multi-point relay sets (MPRS). Such a set reduces the number of nodes involved in a flooding, while still allowing the flood to reach all nodes in the network (given that it is connected). The MPRS is defined as follows:

Definition 9 *A set* $S_{MPR}(v) \subseteq N_1(v)$ *is a* multi-point relay set *(MPRS) for node v, if* $(N_2(v) - N_1(v)) \subset N_1(S_{MPR}(v))$ *(see Definition 18 on page 67).*

Figure 3.7 (a) provides an example of an MPRS: the gray nodes, m and n, constitute the MPRS for node a, also called the *owner* of that set in this document. The following definition is further needed:

Definition 10 *A neighbor* $v_{conn}(v)$ *is a* connecting neighbor *of node v, if* $v_{conn}(v) \in N_1(v)$ *and* $v, v_2 \in N_1(v_{conn}(v))$*, with* $v_2 \in (N_2(v) - N_1(v))$*. Alternatively, a node is a* connecting neighbor *of node v, if* $v \in S_{MPR}(v)$*.*

In [2], two protocols are presented using MPRS. The greedy MPRS-based protocol from that publication starts with two steps, executed for each node v in the network:

1. A node v finds all its 2-hop neighbors that have only one connecting neighbor and adds all such connecting neighbors to $S'_{MPR}(v)$.

2. Node v adds node v_a which covers the largest number of 2-hop neighbors not yet covered by any node in $S'_{MPR}(v)$ to $S'_{MPR}(v)$. This step repeats until all 2-hop neighbors are covered by $S'_{MPR}(v)$.

After the above steps, MPRS have been obtained for each node.

Example Figure 3.7 shows the MPR set $S'_{MPR}(v)$ for each node v of the example network depicted in Figure 3.8. The nodes added to these sets in the first and the second steps, as well as, the nodes causing the connections (via nodes from $S'_{MPR}(v)$) are marked. For node c, as depicted in Figure 3.7 (c), for example, $S'_{MPR}(c) = \{d, f, g\}$, since node d is the only connecting neighbor to node m, node g the only connector to h and n, and node f the only connector to k. Node $e \notin S'_{MPR}(c)$, since node b is connected via two 1-hop neighbors, namely e and f. As d, f, and g cover all 2-hop neighbors, no node is added to $S'_{MPR}(c)$ in step 2. In Subfigure (i), $S'_{MPR}(i)$ first $= \{f\}$, since it is the only connector to node e; all other 2-hop neighbors are connected via two 1-hop neighbors. Thus the first step is concluded. In the second step, since n is still uncovered by $S'_{MPR}(i)$, node g is added to $S'_{MPR}(i)$, winning over h given its lower ID. Therefore, after step 2, $S'_{MPR}(i) = \{f, g\}$.

A node v becomes member of the CDS S_{CD}, if one of the following conditions is satisfied:

i. v's ID is lower than the IDs of all its 1-hop neighbors (i.e. v has the lowest ID in $N_1(v) \cup v$).

ii. v is the multi-point relay of its neighbor with the lowest ID, i.e. $v \in S'_{MPR}(v_s)$, v_s being v's neighbor with the lowest ID in $N_1(v)$.

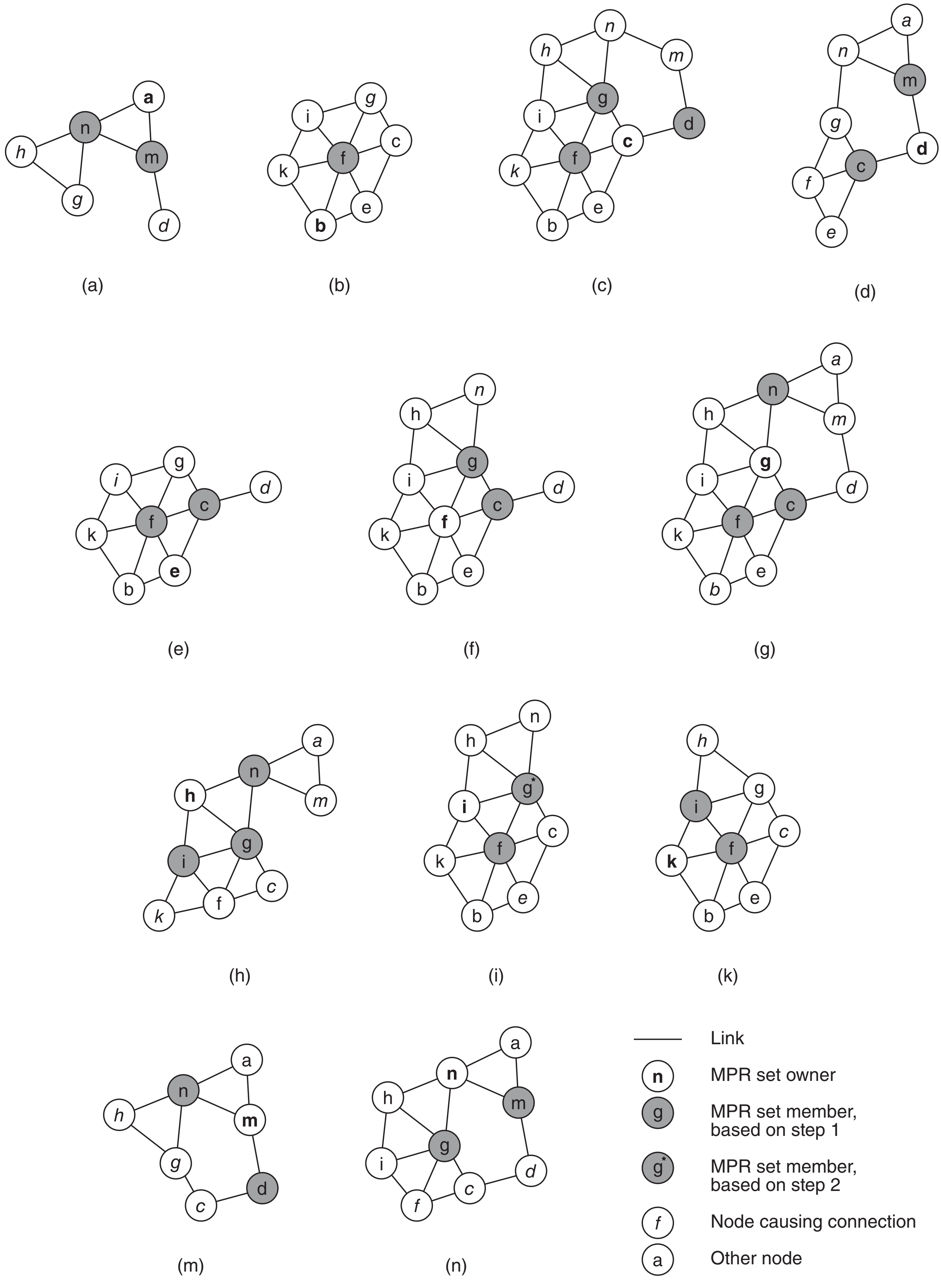

Figure 3.7: Distributed multi-point relay-based protocol by Adjih et al. [2]: MPR set construction

Figure 3.8: Distributed multi-point relay-based protocol by Adjih et al. [2]: CDS construction

Example The previous example from Figure 3.7 continues in Figure 3.8. Node a is added according to condition i, since it has the lowest ID from its 1-hop neighborhood including itself (that is m, n, and a). In contrast, node m is added based on condition ii, being multi-point relay of its lowest-ID neighbor a, as depicted in Figure 3.7 (a). Node e is not added, since it is neither the lowest-ID member of its neighborhood including itself ($\{b,c,e,f\}$) nor a multi-point relay of its lowest-ID neighbor b, as evident from Figure 3.7 (b).

In [117, 94] further methods for the selection of MPR nodes are presented. Wu et al. [160] extend the protocol of Adjih and colleagues [2] to construct smaller forwarding node sets, without incurring additional cost. Similarly, Chen and Shen [33, 34] extend the protocols by Adjih et al. [2] and Wu and colleagues [160] by using node degree instead of node ID in order to break ties. Moreover, they introduce further new rules, so that the size of forwarding sets is reduced to a greater extent.

3.1.2.7 Connected Clustering-Based Construction

Distributed approaches based on connected clustering are closely related to approaches based on WCDS and MIS (see Definition 7 on page 37 and Definition 4 on page 36). In many 1-hop *clustering* protocols, one node is selected to assume the role of a clusterhead; it can be regarded as the leader of the cluster. Connected clustering as a means to create a CDS was considered as early as 1987 in [56]. The set of clusterheads naturally constitutes a dominating set but also always a WCDS, as well as, in some cases, an independent set. Therefore finding a WCDS also translates to finding a feasible set of clusterheads and vice versa. Besides Kwon and Gerla [89], Chlamtec and Farago explicitly exploit this relationship in their approach from [40], by considering and adding only 2- and 3-hop neighbors to the set of clusterheads.

In contrast to the above protocol ([40]), the approach by Gerla et al. [66] solely considers the 2-hop neighborhood and adds nodes to the set of clusterheads independent of their distance. During the construction process, each node periodically broadcasts its state (clusterhead or non-clusterhead). A node changes its state to clusterhead, if it hears only nodes with a higher ID than its own. If the ID of node v_l is the lowest among all IDs of nodes, node v hears from, and lower than its own, v_l becomes v's clusterhead. Moreover, a node that hears two or more clusterheads changes its state to gateway. At the end of the construction process, the sets of clusterheads and gateways together constitute the CDS.

If also including a list of each node's neighbors and their states in the periodic broadcasts,

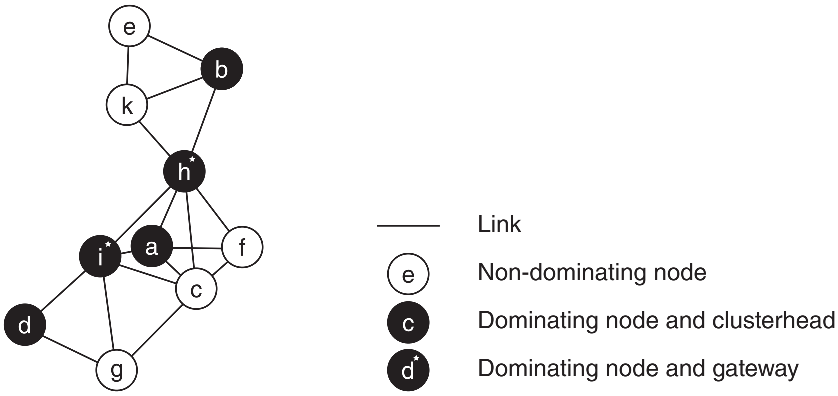

Figure 3.9: Distributed connected clustering-based protocol by Gerla and Tsai [66]

the protocol can be extended to select the most highly-connected nodes from the uncovered neighborhood.

Example Consider the example depicted in Figure 3.9. While nodes a, b, and d are clusterheads, nodes h and i are gateways. Both, clusterheads and gateways, together constitute the CDS. Node b, for example, becomes a clusterhead, as it has the lowest ID within its 1-hop neighborhood including itself ($\{b,e,h,k\}$). Node h changes its state to gateway, since hearing from two clusterheads, namely a and b.

Remotely similar approaches were proposed in [95, 96, 113]; the approach from [96] takes into account energy during clusterhead selection. In [37], the authors introduce a method connecting clusterheads by employing a multicast tree.

3.2 Connected k-Hop Dominating Sets

From the area of connected k-hop dominating sets (CkDS) (see Definition 1 on page 1), with $k > 1$, there are much fewer approaches than from the area of CDS, i.e. CkDS with $k = 1$. Using the classification scheme from previous sections, they can be categorized into connected clustering-, greedy-, and pruning-based approaches. Since there are currently only very few CkDS construction protocols (all of them are distributed), in contrast to the previous section, they are described in chronological order in this section.

3.2.1 Connected Clustering-Based Construction

The first approach to solve the CkDS problem in wireless networks was presented by Theoleyre and Valois in [143, 144].

3.2.1.1 Approach by Theoleyre and Valois

The approach [143, 144] is divided into two phases: First, a k-hop dominating set is established, based on information exchanges which need k-hop flooding. The resulting dominators are also clusterheads at the same time. Further, nodes that have changed their state to dominatee upon reception of the dominators' hello messages become members of their clusters. In the second phase, the k-hop dominating set is interconnected successively starting at a selected dominator, which notifies all $2k+1$-hop distant dominators using flooding, thereby connecting them. Upon connection, each dominator repeats this action, until the entire dominating set is connected.

In order to establish k-hop knowledge of the neighborhood in the *first phase*, each node periodically floods its k-hop neighborhood with hello messages. Additionally, such a flooding is performed after each state change. A hello message includes the *ID*, the *state*, and the *weight* of the originating node. Nodes are in one of the following four states:

Dominator The node is dominating, i.e. member of the dominating set D

Dominatee A node covered, or dominated, by a dominator

Active Participating in election

Idle At initialization

The *weight* of a node may reflect its properties in terms of, for example, energy or degree. However, details on the concrete use of the weight are not discussed in [143, 144].

The state changes performed by the protocol are based on the following rules:

Idle → active If a node is in the idle state and receives a hello message from a dominatee or another idle node, it becomes active.

Active → dominatee If a node v_c is in the active state and receives a hello message which originates at a dominator v_d, it changes its state to dominatee. v_c further becomes a member of v_d's cluster (although this does not impact the construction process).

Active → dominator If a node is active and has the highest weight of all active nodes in its k-hop neighborhood for τ time, it changes its state to dominator.

Example Consider the example in Figure 3.10 (a). Node 1 has first changed its state to dominator, flooding its $k = 5$-hop neighborhood. For illustration purposes, this is depicted in Subfigure (a) using circles to represent the broadcasts. In (b), already dominators 1, 2, 3, and 4 exist. The dotted and dashed lines in (a) and (b) represent the borders of the k-hop floodings with hello messages by the dominators. In Subfigure (b), the circles representing the broadcasts are omitted. Note that periodic k-hop flooding is not only performed by the dominators, as depicted in the figure, but also by every other node in the network.

The *second phase* of the protocol is initiated by the leader v_l, which is one of the dominators and which represents a starting point for the connection of the dominating set. v_l can be, for example, an access point, acting as a gateway to a wired network. When the second step starts by setting $v'_d = v_l$, or when another dominator v'_d becomes connected, node v'_d sends a join message to all dominators in its $2k+1$-hop neighborhood using flooding. All not yet connected dominators receiving the join message return a join-reply message, acknowledging their connection. The join-reply message follows the inverse route of the join message, setting the state of all nodes on that route to dominating. When the message exchange concludes, it yields a CkDS.

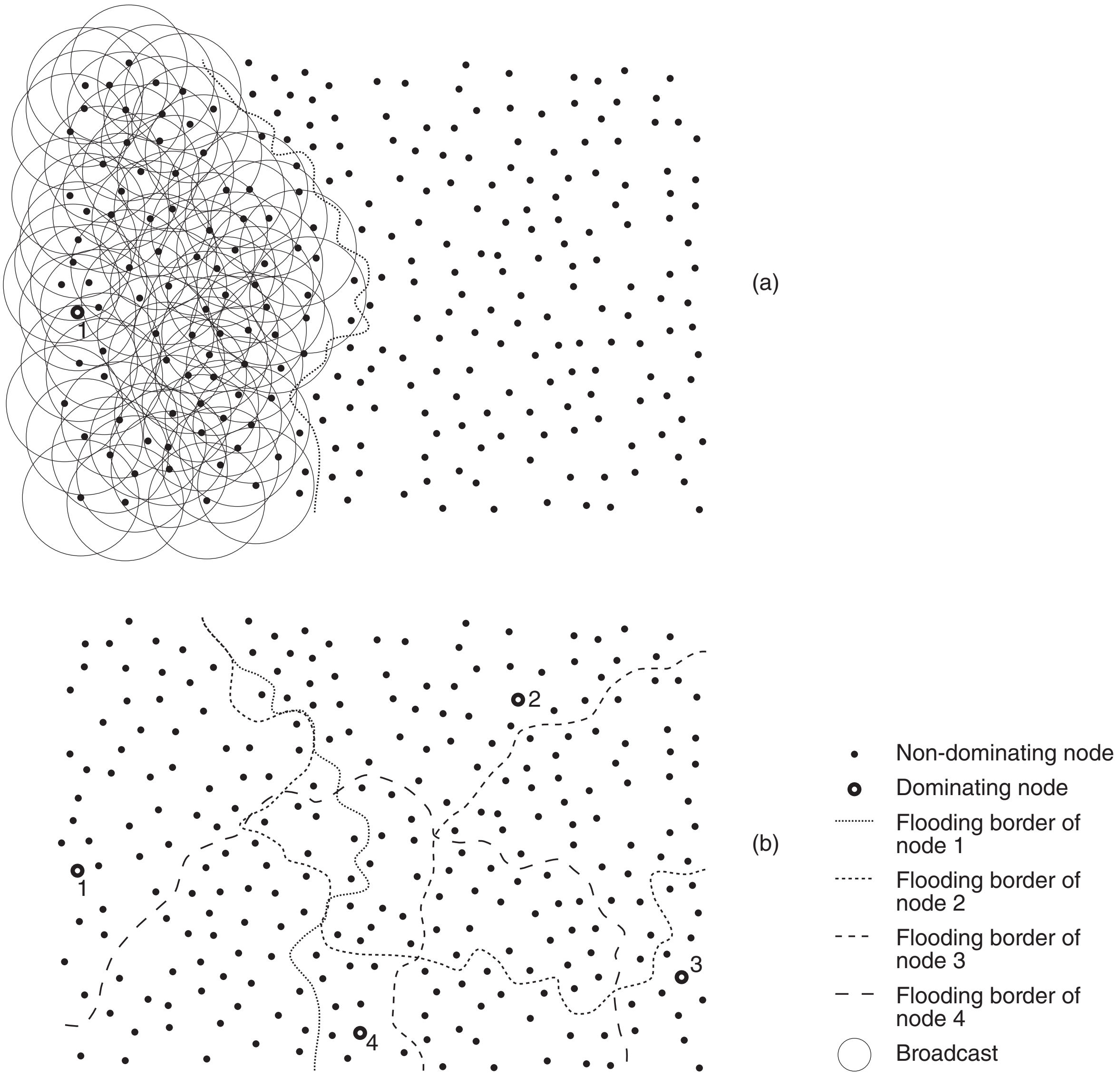

Figure 3.10: Connected clustering-based protocol by Theoleyre and Valois [143, 144]: first phase

Example Figure 3.11 illustrates the second phase of the protocol, continuing the example from the previous figure. Assume that node 1 is the leader starting the interconnection. Therefore, in Subfigure (a), it floods the join message in its $2 \cdot k + 1 = 2 \cdot 5 + 1 = 11$-hop neighborhood, so that it reaches dominators 2 and 4, which connect to it. The dotted line in this subfigure demonstrates how far the flooding reaches, while the solid black lines represent the interconnections. In (b), the dominators 2 and 4 flood their $2 \cdot k + 1 = 2 \cdot 5 + 1 = 11$-hop neighborhoods with join messages, reaching nearly the entire network. Dominator 3 first hears from dominator 4 and connects to it, so that the resulting dominating set is a CkDS with $k = 5$.

3.2.1.2 Approach by Yang, Wu, and Cao

Yang et al. propose an approach [166] for CkDS construction, which is mainly based on: (1) traditional lowest ID multi-hop clustering [95] to determine clusterheads, and (2) an altered

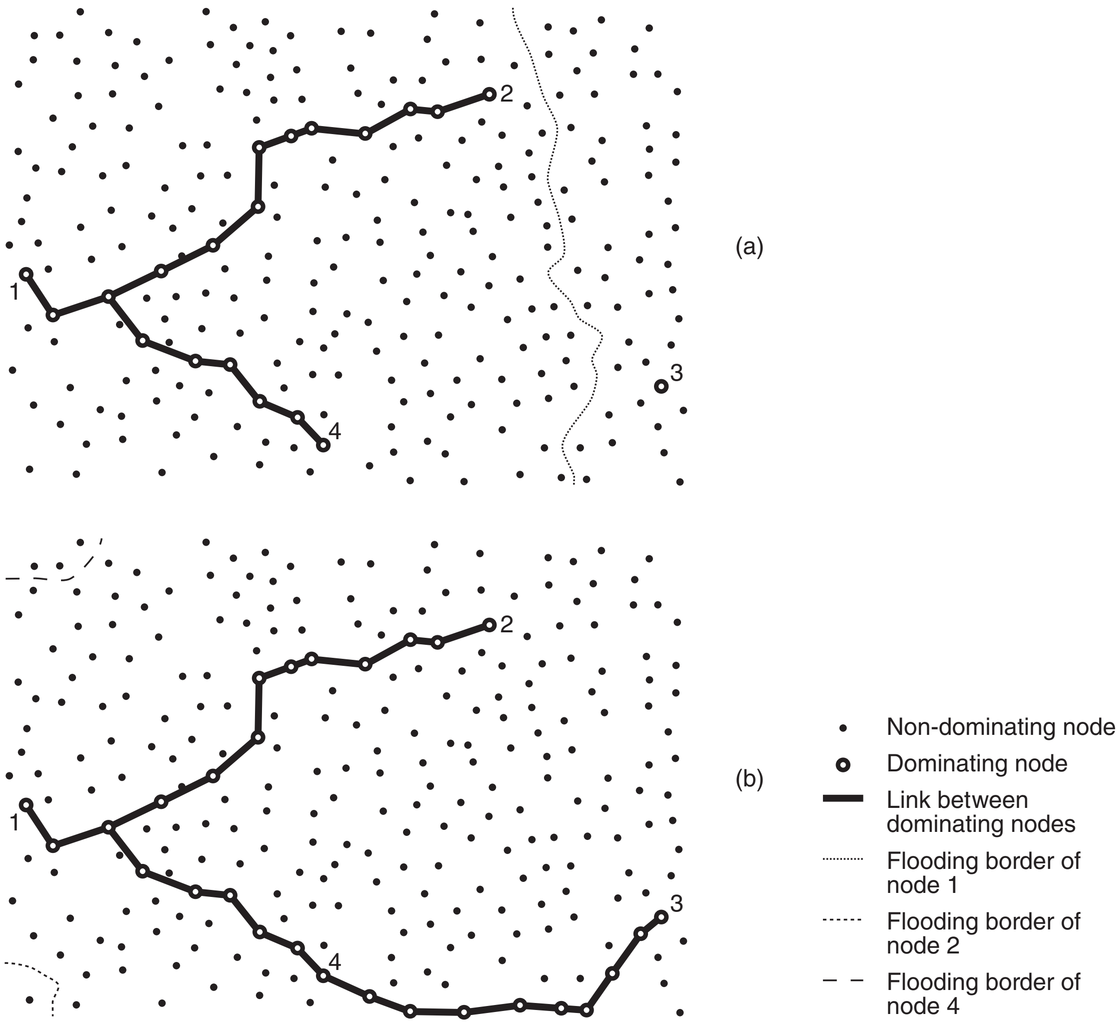

Figure 3.11: Connected clustering-based protocol by Theoleyre and Valois [143, 144]: second phase

version of the local minimum spanning tree (LMST) protocol [93] to interconnect them. It is a successor to their approach for the construction of CkDS with $k = 2$, introduced in [161].

As *first step*, the approach executes lowest-ID clustering (similar to [95]), in which

- ◇ each node floods its k-hop neighborhood announcing its existence and priority,
- ◇ nodes with the highest priority in their k-hop neighborhood declare themselves clusterheads and flood clusterhead-declaration messages within this neighborhood.

If a non-clusterhead receives one or more clusterhead-declaration messages, it selects one cluster to join as member. The selection can be based on the ID of the clusterhead, the distance to it, or the size of the cluster.

The clusterheads, marked in the first step, are connected in the *second step* using the following procedure:

1. Each clusterhead floods an announcement message over $2k+1$ hops, which includes its ID and records the path traveled.

2. A clusterhead administrates a table S of known clusterheads and the paths towards these clusterheads. If a clusterhead receives an announcement message from clusterhead v_c which does not exist in S, it adds v_c and the path towards v_c to S. Else, if an announcement message m_a from clusterhead v'_c which is already present in S arrives, and the path p_m recorded by m_a is shorter than the path p_t recorded towards v'_c in S, p_t is replaced by p_m in the table.

3. Each clusterhead floods the table S, which it administrates locally, using a distances message through the *entire* network.

4. After receiving all incoming distances messages, each clusterhead v_c* constructs a local minimum spanning tree T rooted at v_c*, using the information from the tables S included in these messages. Non-clusterhead members of T become gateways.

After the above steps, clusterheads and gateways have been determined, which together constitute the CkDS. Given the fact that the necessary computations are performed locally by each clusterhead, it is likely that two different clusterheads will obtain two different CkDS. Note that for the sake of clarity, the messages which are not named in [93], were assigned names in this description.

Example In the example depicted in Figure 3.12, clusterheads 1, 2, 3, and 4 have been selected, as shown in Subfigure (a). Given the fact that for node 1, for instance, solely nodes 2 and 4 are reachable within $2 \cdot k + 1 = 2 \cdot 5 + 1 = 11$ hops, only the depicted connections, representing the shortest paths to its clusterhead neighbors, are part of node 1's table S. In Subfigures (a) and (b), the dotted and dashed lines represent the borders of the $2 \cdot k + 1 = 2 \cdot 5 + 1 = 11$-hop floodings used. In (b), all sets S of nodes 1, 2, 3, and 4 are depicted using solid lines. Note that node 2 is aware of a different shortest path to node 4 than vice versa. This situation can occur, since each clusterhead only considers incoming announcement messages. In (c), the CkDS corresponding to the local minimum spanning tree computed by, and rooted at node 3 is depicted.

3.2.1.3 Communication Cost and Construction Time

The *communication cost* incurred by both connected clustering-based approaches [143, 144, 166] depends quadratically on k, since it is dominated by k-hop flooding in terms of k. This interdependency can be explained well, when considering in what manner the process of k-hop flooding develops over time: As the area of a circle grows quadratically with radius r ($\pi \cdot r^2$), the area covered by an omnidirectional flooding will grow in a similar manner with growing k. When assuming a certain average number of nodes per unit of area, the number of nodes involved in the flooding grows linearly with the area. Notice that in both of the approaches, k-hop flooding is initiated by each node in the network.

Moreover, the cost of the clustering-based approaches grows linearly with the average node degree d, since flooding by definition consists of broadcasts, which are received by all 1-hop neighbors of the transmitting node (if not considering effects like packet loss, etc.).

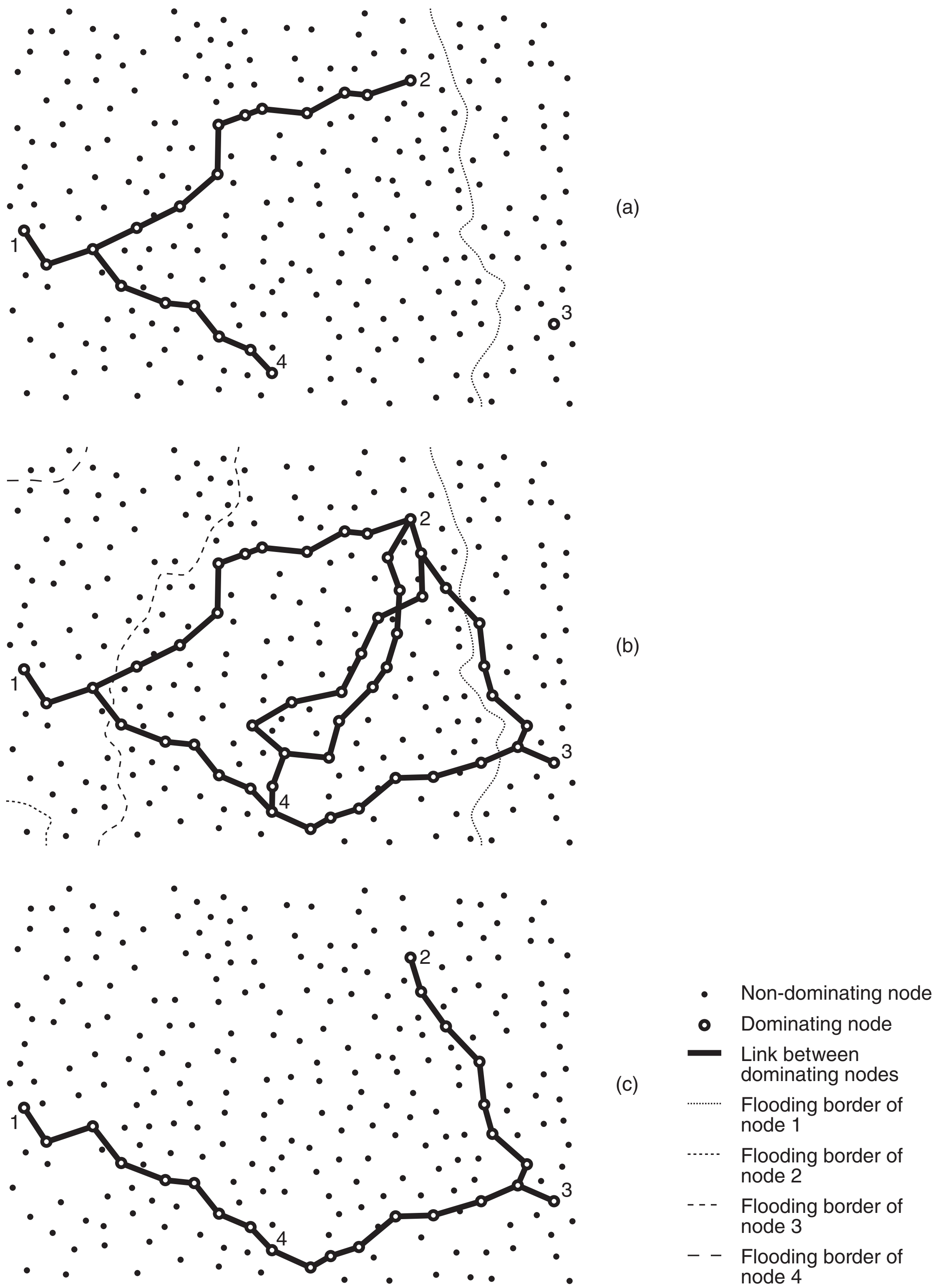

Figure 3.12: Connected clustering-based protocol by Yang et al. [166]

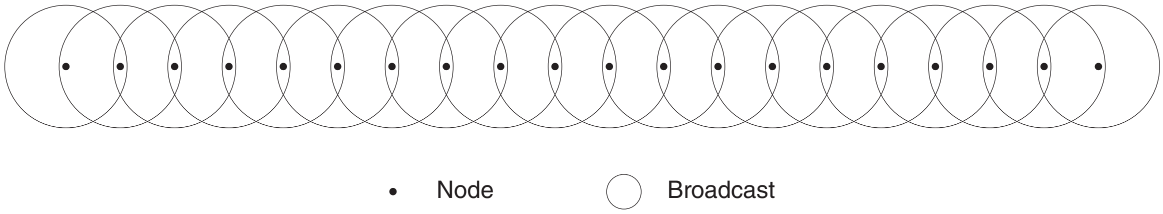

Figure 3.13: Connected network with n nodes in which $n-2$ nodes exhibit a node degree of 2, and 2 nodes a node degree of 1

Given the fact that the *construction time*, in terms of k, is dominated by k-hop flooding, it grows linearly with k, due to the flooding's parallel nature. Furthermore, the construction time of both approaches increases also linearly with the number of nodes in the network, since

- ◇ in the approach by Theoleyre and Valois [143, 144], during the interconnection phase (that is the second phase), all existing dominators in the network are connected consecutively. This translates to the following worst-case scenario: in a connected network in which $n-2$ nodes exhibit node degree 2, and 2 nodes node degree 1, as depicted in Figure 3.13, all nodes are *consecutively* reached at least once by the join message, assuming that a node with degree 1 is the leader.
- ◇ in the approach by Yang and colleagues [166], each clusterhead floods table S, which contains information on its $2k+1$-hop clusterhead neighbors, through the entire network during the interconnection step (i.e. the second step). This translates to the following worst-case scenario: in a connected network in which $n-2$ nodes exhibit node degree 2, and 2 nodes node degree 1, as depicted in Figure 3.13, all nodes are *consecutively* reached at least once by the distances message including table S, assuming one of the nodes with degree 1 to be a clusterhead.

3.2.2 Greedy Construction

Sausen et al. propose a greedy method for constructing a CkDS in [130]. The construction process starts at a node which was selected for this purpose, such as the base station. For the sake of clarity, this node is referred to as *base station* in this description. The process of construction is divided into the following phases and subphases:

Phase 1 The aim of the first phase is to collect information about the distance to the base station and the k-hop neighborhood of each node, which enables the election in phase 2. Therefore, this phase is further divided into two subphases:

a. The base station v_b broadcasts an information message (IM) m_{IM}, recording the number of hops taken since being dispatched by v_b in $m_{IM}.dbs$ (distance to base station). When a node v_r receives an IM m'_{IM}, it sets its own distance-to-base-station (dbs) field to $m'_{IM}.dbs$ and rebroadcasts m'_{IM}, if one of the following conditions is satisfied:
   - ○ v_r receives an information message for the first time, or,
   - ○ $m'_{IM}.dbs$ is lower than v_r's dbs field.

Else, m'_{IM} is discarded. This subphase translates to a flooding of the entire network.

b. In each of the k rounds, each node v broadcasts the information it has already collected about its k-hop neighborhood and the inter-node distances within this neighborhood, $N_k(v)$, using a neighborhood message (NM) (this name was assigned to the message to make the description more readable; originally, it is not named in [130]). This subphase translates to a k-fold flooding of the network.

Phase 2 In the second phase, the information gathered in the first phase is used for the local election decisions at each node. The base station is the first node to make such a decision and always elects itself to become dominating. Any other node v_e, conducting the election, selects a node v_s with the minimum distance to the base station (dbs) in v_e's k-hop neighborhood, including itself, to be dominating. Further, only nodes can be elected by v_e which are already dominating or 1-hop neighbors of dominating nodes. Ties are broken by largest degree, persisting ties by largest ID. After a node, the base station or v_e, has conducted the election, it announces its choice by flooding the election message (EM) over k hops. A node receiving an EM for the first time, conducts the election.

Note that the parameter k is called r in the paper that introduced this approach [130] by its authors.

Example Figure 3.14 illustrates the construction of a CkDS, using the approach by Sausen et al. [130]. In the example, the process starts from the base station B. Subfigures (a)–(c) show the CkDS at successive stages of development.

3.2.2.1 Communication Cost and Construction Time

The *communication cost* incurred by the approach introduced in [130] depends quadratically on k, since it is dominated by k-hop flooding, which is conducted by each node in the network, in terms of k. This interdependency is discussed in detail above (on page 52) and in Section 5.2.4, assuming a two-dimensional worst-case scenario with an uniform node density, where a flooding does not reach the borders of the network, as illustrated in Figure 5.4.

Moreover, the cost of the approach by Sausen et al. [130] grows linearly with the average node degree d, since flooding by definition consists of broadcasts, which are received by all 1-hop neighbors of the transmitting node (if not considering effects like packet loss, etc.).

Given the fact that the *construction time*, in terms of k, is dominated by k-hop flooding, it grows linearly with k, due to the flooding's parallel nature.

Furthermore, the construction time of the approach increases linearly with the number of nodes in the network, since the CkDS is constructed from the base station towards the borders of the network, as depicted in Figure 3.14. This translates to the following worst-case scenario: in a connected network in which $n-2$ nodes exhibit node degree 2, and 2 nodes node degree 1, as depicted in Figure 3.13, all nodes *consecutively* participate in the election, assuming that a node with degree 1 serves as base station.

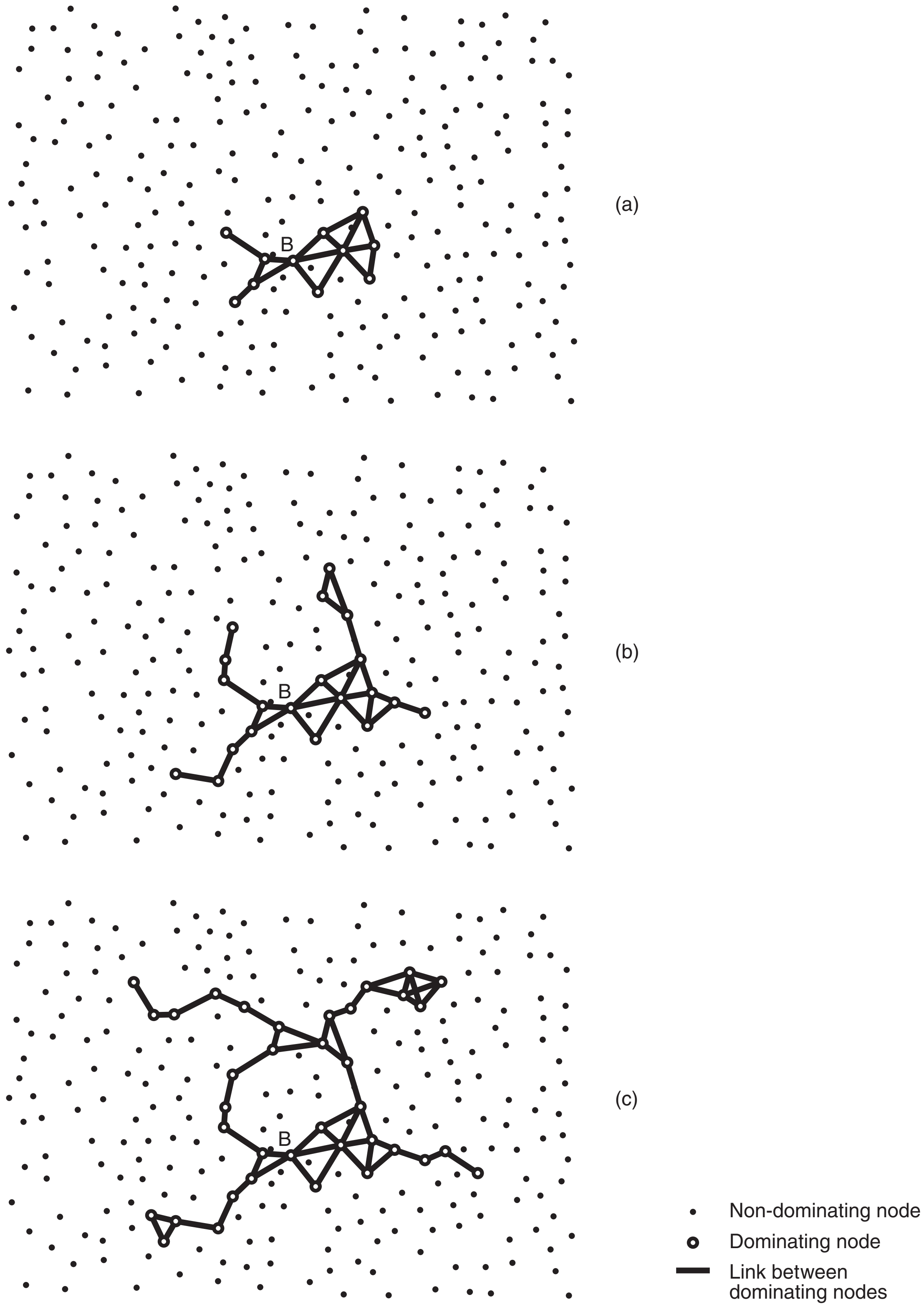

Figure 3.14: Greedy protocol by Sausen et al. [130]

3.2.3 Pruning-Based Construction

A method for the construction of a CkDS based on pruning is presented in [165] by Yang et al. The construction process is conducted in k rounds, so that in round j, with $j \in [1,k]$, a CxDS with $x = j-1$ is pruned to become a CxDS with $x = j$.

During the pruning process with the objective to construct a CkDS with a certain k, $D_k[m]$ is the connected dominating set constructed in round m, in which there exists a path of at most m hops between any node $\notin D_k[m]$ and some node $\in D_k[m]$. Note that $D_k[0]$ contains all nodes in the network.

Each node v in the network administers its ID, *v.id*, its chosen number, *v.num*, its neighborhood set encompassing all its known 1-hop neighbors, $v.neigh = N(v)$, and their status, i.e. dominating or non-dominating; *v.num* is initialized to ∞ for each node v.

To create a CkDS, in each round $j \in [1,k]$, each node v that participates in the construction process (that are all nodes at the beginning) follows the steps described next to transform $D_k[j-1]$ to $D_k[j]$:

1. v exchanges *v.id*, *v.num*, and *v.neigh* with its 1-hop neighborhood.

2. If v was pruned in the previous, $(j-1)$th, round, and v does not have any neighbors in the dominating set of the previous round, i.e. $N(v) \cap D_k[j-1] = \emptyset$, *v.num* is set to ∞. Further, v exits the construction process.

3. If

    ◇ v is part of the previous-round dominating set, that is $v \in D_k[j-1]$, and if

    ◦ all of v's neighbors in the previous-round dominating set, i.e. $N(v) \cap D_k[j-1]$, are directly connected (*first case*), or,

    ◦
    $$(N(v) - U) \cap D_k[j-1] \subseteq \bigcup_{z \in U} N(z) \cap D_k[j-1] \tag{3.2}$$

    for some connected subset U of

    $$\{z | z \in N(v) \cap D_k[j-1] \wedge z.id > v.id\} \tag{3.3}$$

    In words, a connected subset U contains all of v's 1-hop neighbors from the dominating set of the previous round with IDs *greater* than v's. Then, it is checked in Equation 3.2, whether the *lower*-ID nodes from the previous-round dominating set neighboring to v are a subset of all neighbors from the previous-round dominating set of all nodes from U (*second case*).

    then, v prunes itself from $D_k[j-1]$, so that $v \notin D_k[j]$, setting $v.num = (j-1) \cdot maxID + v.id$, with *maxID* being the maximum node ID in the network.

Finally, after k rounds, all nodes $\in D_k[k]$, and at the same time, all nodes v_d with $v_d.num = \infty$ constitute the CkDS. Given step 1, the network is flooded k times during the construction process.

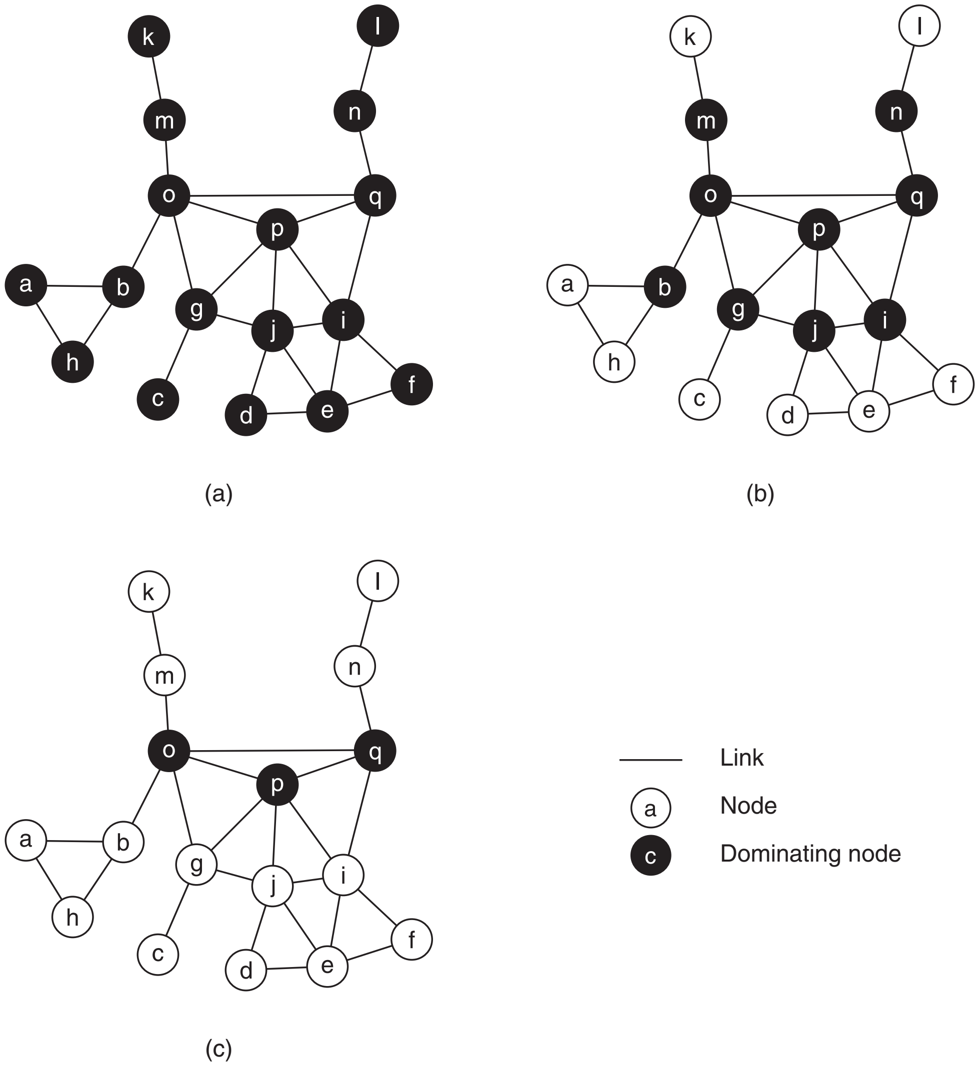

Figure 3.15: Pruning-based protocol by Yang et al. [165]

Example Consider the example in Figure 3.15. The aim is to construct a CkDS with $k = 2$. Subfigures (a), (b), and (c) show $D_2[0]$, $D_2[1]$, and $D_2[2]$. All nodes whose chosen number equals ∞ belong to the dominating set and are colored black. Subfigure (a) shows the initial state, in which all nodes belong to the dominating set, and their chosen numbers equal ∞. At this point, node h is pruned based on the first case, since all of its dominating neighbors, a and b, are directly connected. Node e, in contrast, cannot be pruned according to the first case, since multiple of its dominating neighbors (d and f, among others) are not directly connected. Nonetheless, node e is pruned based on the *second* case, since $(N(e) - \{f,i,j\}) \cap D_2[0] \subseteq (N(f) \cup N(i) \cup N(j)) \cap D_2[0]$ with the connected subset $U = \{f,i,j\} = \{z | z \in N(e) \cap D_2[0] \wedge z.id > e.id\}$. These and the other operations conducted result in the state depicted in Subfigure (b). The final CkDS, which resembles $D_2[2]$, is depicted in Subfigure (c).

3.2.3.1 Communication Cost and Construction Time

The *communication cost* incurred and *construction time* needed by the approach introduced in [165] depend linearly on k, since it employs k rounds, in which each node that participates in the construction process communicates with all of its 1-hop neighbors. Therefore, the number of rounds dominates cost and construction time in terms of k.

Moreover, the *communication cost* incurred by the approach increases quadratically with the average node degree d, since it is dominated in terms of d by: the exchange of tables containing a node's 1-hop neighborhood (*v.neigh*) by each node, performing step 1 of a round, with its 1-hop neighborhood.

In contrast, the *construction time* of the approach is basically independent of the number of nodes n in the network, since pruning is conducted in parallel in the entire network. However, the construction time depends linearly on d, since during the information exchange in step 1 of a round, the information exchange request cannot be answered by all 1-hop neighboring nodes simultaneously but only sequentially (else, there would be a collision).

3.3 Summary

This chapter reviewed the state of the art. It first discussed approaches for the construction of connected dominating sets (short CDS, which are basically CkDS with $k = 1$), before focusing on CkDS (if not further specified, $k > 1$ is assumed).

CDS have been subject to extensive research since the 1980s. Reflecting the development of computer networks, the first approaches were centralized, assuming that the entire network topology is known at a single place. In this class of approaches, I found greedy, Steiner tree-based, weakly-CDS-based, and pruning-based algorithms. One greedy approach [70], for example, works basically by growing a tree, starting at the vertex with the maximum degree in the network. After the tree has been constructed, all non-leaf vertices belong to the CDS. Naturally, the main drawbacks of these centralized algorithms are their limited scalability, the reliance on a single point of failure, and the relatively high overhead incurred, since information about the entire network topology needs to be routed to a single point.

In order to enable an efficient CDS construction in wireless networks, distributed approaches were proposed. While some of the methods were adopted from centralized algorithms, also new ones were used: this class of approaches includes greedy, Steiner tree-based, weakly-CDS-based, pruning-based, maximal independent set-based, multi-point relay-based, as well as, connected clustering-based protocols.

The above approaches serve as precursors for state-of-the-art CkDS construction protocols, which employ connected clustering-, greedy, and pruning-based methods. One of the connected clustering-based protocols [143, 144], for example, first creates k-hop clusters using k-hop flooding. Subsequently, it connects their clusterheads by employing $2k+1$-hop flooding. Finally, all clusterheads and their connectors constitute the resulting CkDS.

A concise analysis of the operation characteristics of the state-of-the-art CkDS approaches yields that the communication cost incurred by them is linearly or quadratically dependent on k and average node degree d. Moreover, the construction time needed by these approaches grows linearly with k and number of nodes n in the network. The only exception is the protocol by Yang et al., proposed in [165], which exhibits a construction time that is independent of n.

4

Self-Organizing Random Walk-Based CkDS Construction

After a review of background information and state of the art in the previous chapters, this chapter introduces a novel approach for the construction of connected k-hop dominating sets (CkDS) in wireless sensor networks (WSNs). To cope with the resource restrictions of this network class, as reviewed in Section 2.1.4, the biologically-inspired, self-organizing protocol employs methods and exhibits properties which are inherent in many biological systems. It is inspired by the general technique of random walks and, in particular, by the flight behavior of ovipositing Pieris rapae, which efficiently solves the coverage problem in nature by employing random walks. The proposed approach is the first protocol for the construction of connected dominating structures, including CDS and CkDS, to adopt random walks to wireless networks. In [78], the central contribution of this chapter was published recently.

The first section of this chapter presents an overview of the inspiration and the design considerations for the introduced CkDS construction method. Thereafter, Sections 4.2–4.7 provide a detailed description of the proposed protocol, before Section 4.8 discusses its behavior.

4.1 Inspiration and Design Considerations

The general technique of random walks, which are encountered in many biological, as well as, other natural systems, served as archetype for the first versions of the devised protocol. In subsequent stages of the design, to identify the concrete biological roots, as well as, to study further details and intricacies of this behavior pattern, I looked for specific examples from the domain of biology. As discussed in Section 4.1.1, from the candidate species, P. rapae appeared to be the most suitable to advance the development of my approach. For instance, while the first versions of the protocol operated in a sequential manner, building the dominating set from a starting point recursively towards the edges of the network, the concurrent oviposition of multiple P. rapae females inspired the parallelization of the approach, so that its construction time became independent of the size of the network. Further, for example, after observing P. rapae's behavior, rules that created bifurcations in the connected dominating structures were

Figure 4.1: Pieris rapae. Image source: [90]

removed from the protocol in favor of the long-distance random walks of P. rapae, leading to the construction of more regular dominating structures. In addition to this, also artificial elements had to be added, such as the division into two behavior blocks and further subblocks, as outlined in Section 4.1.3.

During the design of the proposed protocol, I focused on its application in wireless sensor networks, since its biological archetype appeared to exhibit exactly the strengths desirable in this network class:

- ◇ a high amount of tolerance to an unreliable topology,
- ◇ extremely low requirements in terms of information exchange, and
- ◇ the ability to operate within an arbitrarily large and globally unknown system,

to name the most important ones, which reflect the archetype's desirable properties, discussed extensively in Section 4.1.2. Naturally, independent of its suitability for WSNs, the proposed protocol could be very well applied to create structures in many other areas, such as in large-scale wired networks.

4.1.1 Behavior of Pieris Rapae

Pieris rapae, also known as Cabbage Butterfly or Small White, depicted in Figure 4.1, is native in Europe and North Africa [125] but has invaded North America, Australia, and New Zealand. Root and Kareiva have studied the flight behavior of ovipositing P. rapae extensively in [125]. They measured the flight paths of female P. rapae in terms of move lengths and turning angles, yielding a sequence of movement vectors. As test scenarios, the authors used various experimental gardens planted with different mixtures and arrangements of collards, serving as hosts for oviposition, and companion plants, as well as, diverse meadows. They obtained the

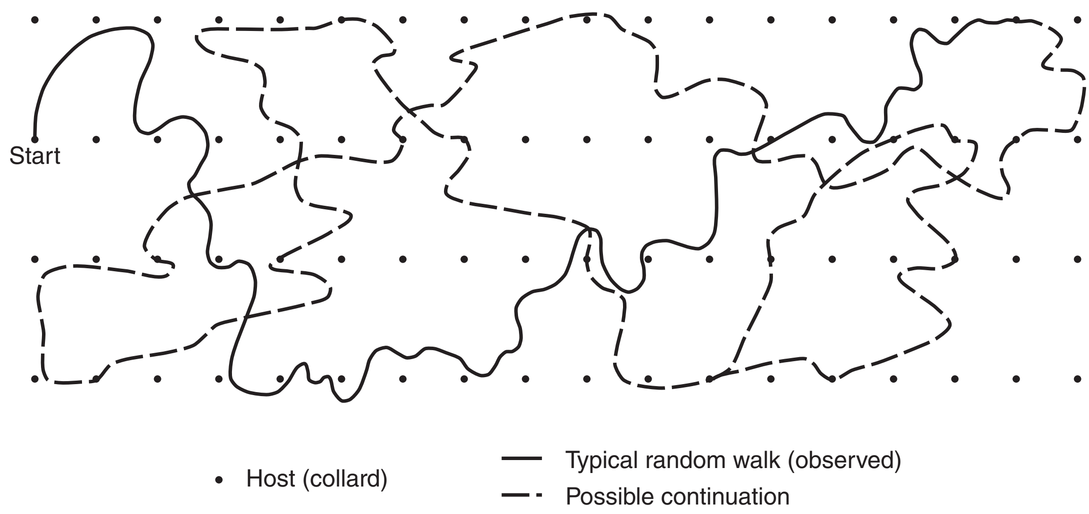

Figure 4.2: Generated pattern in the natural system

results employing two methods: observation from towers and direct observation by following the individuals keeping two to five meters behind them.

Root and Kareiva [125] conclude that, during the evolutionary process, P. rapae females are selected which average out spatial variation in survivorship, and thereby reduce the generation-to-generation variation in reproductive rates. The reduction of generation-to-generation variation in number of offspring surviving per parent has been linked to the increase of relative fitness of a genotype [67]. One of the important factors determining to what extent this aim is achieved is the flight behavior during oviposition by the fecund P. rapae adult female. She possesses the capability to recognize hosts and uses a random walk to visit a subset of them for oviposition. To my knowledge, there exist models of her flight behavior obtained empirically (as the one by Root and Kareiva), but there is little understanding about her internal cognitive processes. Her flight in between the stops is highly irregular, however, with a tendency towards linearity. A description of this flight behavior is necessarily probabilistic. Her objective is to maximize the payoff of oviposition by finding a good trade-off between two contrary intentions:

a. using egg spreading to average out variations in larval survivorship, which is highly influenced by the risks imposed by the environment, and

b. energy spent for relocation.

Typical dangers for eggs and larvae are drowning in water on leaves, being washed off by heavy rain, being stuck in sites that deteriorate after the eggs were laid, or other localized catastrophic events [125].

Although P. rapae's motivation is to maximize the payoff of oviposition, she produces another artifact: if selecting an adequate section of her flight trajectory, one obtains an s-distant connected structure, where s is a maximum distance between any point in the topology and its nearest point in the structure. An example of such a structure is depicted in Figure 4.2, drawn using a solid and dashed line: As in the studies by Root and Kareiva, a cabbage field serves as topology. A typical random walk observed by the authors in [125] is depicted using the solid line, while a likely continuation of this random walk is represented by the dashed line.

It should be noted that, although random walks can be observed in many other species, such as Caribous [18], fish [109], etc., I selected P. rapae, since Root and Kareiva's model appeared to me more useful than the others (e.g. [18, 109]) for a transfer to the considered artificial system, given the following reasons:

1. The topology used for the experiments by Root and Kareiva to develop their model resembles the considered artificial system. It is a bounded area (here, the field, network area in the artificial system) with a number of spatially distributed landing sites (here, the host plants, wireless nodes in the artificial system).

2. P. rapae's flight behavior during oviposition is adequately simple, so there appear to be no higher-level behavior elements, such as the fidelity to specific sites (as, for example, described in [18]) that are not desired in the artificial system and could make the transfer to it more difficult.

4.1.2 Desirable Properties and Mapping

Drawing inspiration from the flight behavior of ovipositing P. rapae, the protocol aims at inheriting some of its desirable properties:

- ◇ *Distribution* of a swarm of agents (i.e. ovipositing P. rapae) which translates to avoiding a central point of failure. Consider the contrary case, in which there is a central instance responsible for making decisions within the network. This instance needs to acquire knowledge about the network, which, in wireless sensor networks or other multi-hop networks, needs to be transferred along several hops. As communication is expensive, this approach scales poorly.

- ◇ *Parallelism* coupled with *asynchronity* reducing the amount of inter-dependencies. Similar to distribution, if entities of the protocol run in parallel and are highly asynchronous, this implies that the amount of information that needs to be exchanged among them is rather low, translating to a highly desirable property in WSNs.

- ◇ Use of only *local knowledge* and *lower-level interactions*, eliminating the need to obtain (i.e. transfer) non-local information. Similar to the above properties, this reduces the communication overhead needed and is highly desirable in WSNs.

- ◇ *Randomization* and *redundancy*, so that risk is spread and failures can be compensated. Failures, such as link failures or nodes exhausting their energy reserves, have to be expected to occur at different points in the WSN. Randomization provides a good means to cope with systematic, or patterned, errors and failures. At the same time, redundancy translates to being capable of choosing between different options, so that the failure of not all of them results in the availability of a certain amount of remaining functionality.

- ◇ *Scalability*, given distribution, parallelism, and local knowledge.

- ◇ Increased *robustness*, given the above properties.

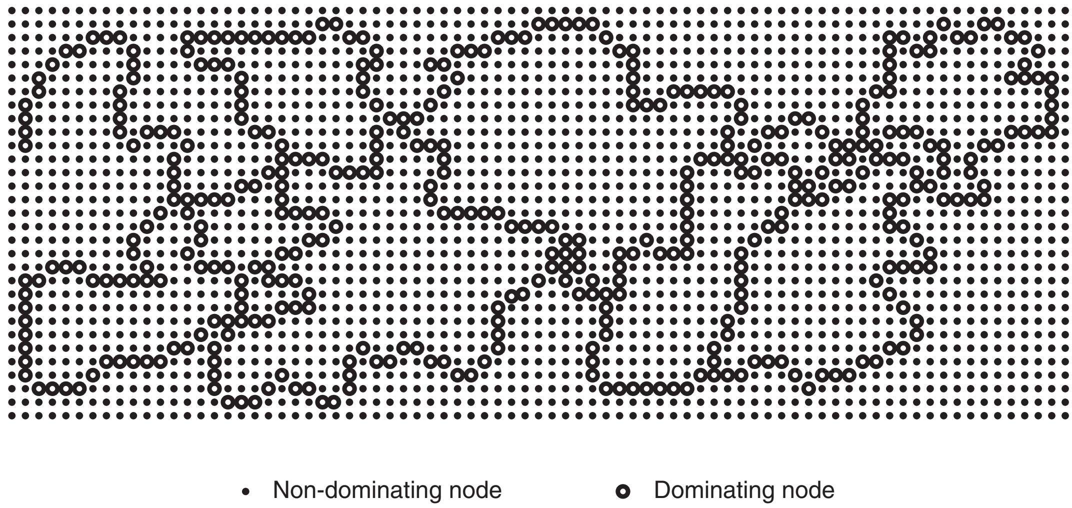

Figure 4.3: Generated pattern in the artificial system (CkDS with $k \geq 11$ and $k' = 11$)

◇ Furthermore, the process is *self-organizing*, since the resulting global-level pattern, consisting of the flight trajectories, emerges solely from numerous lower-level interactions with the environment, specified by rules executed using only local information, without reference to the global pattern (according to the definition from [26]). The implications of this property are discussed extensively in Section 4.8.

In order to transfer the behavior for producing the artifact described in Section 4.1.1 to WSNs, the ovipositing female P. rapae are modeled as artificial exploration agents, the hosts as sensor nodes, and the flight trajectories over the hosts as sequences of visits of the sensor nodes along these flight trajectories. An example result of such a modeling, a CkDS with $k = 11$, is depicted in Figure 4.3, assuming that each node which is not at the topology's edges has eight 1-hop neighbors.

4.1.3 Artificial Adaptations

Unfortunately, the early versions of the proposed protocol, inspired by the general technique of random walks and P. rapae's behavior, based on the mapping discussed above (see Section 4.1.2), yielded rather poor results. In experiments, I found out that they often created *dis*connected dominating sets with high variation in local dominating-to-all-node ratios and many dead ends (i.e. dominating nodes with only one other dominating node in their 1-hop neighborhood). As pointed out in [75], when adopting solutions from natural systems, pure mimicry often leads to failure, given the lack of a sufficient amount of adaptation to the artificial system, as experienced, for example, during early attempts to build flight apparatus closely resembling birds' wings. In addition to imitation, it needed the adaptation of the shape of the wings to the changed system properties (material weight, propulsion, etc.) by Lilienthal and the Wright brothers to succeed. Therefore there were further adaptations needed: I divided the protocol into two intertwined behavior blocks. The first one creates a dominating set, while the second connects this set to a CkDS. Further, I created rules, such as for controlling the length of the walks by the exploration agents and selection of nodes to be added to the dominating set. The resulting protocol is described in the next section.

The behavior exhibited by the proposed protocol consists of a myriad of distributed actions, executed by fully autonomous entities, which communicate via *stigmergy* and thus interact only with their local environment. For a comprehensive overview of the concept of stigmergy, the interested reader is referred to the article by Theraulaz and Bonabeau [145].

Given the above design decisions, the protocol's description is based fully on the notion of *agents*, i.e. acting entities, situated in a habitat, the wireless sensor network. These agents are only aware of their own state and the state of the node they are visiting at the current point of time. Apart from that, they are not capable of perceiving each other. Further, it is important to remark that there is no superior entity, which controls or influences any of the agents.

4.2 Outline

The proposed protocol consists of two intertwined behavior blocks: in the first, a dominating set is constructed, in the second, this set is connected to become a CkDS. Each of the blocks is further subdivided into two subblocks: exploration and construction.

Assuming a network without a dominating set, the first behavior block (Section 4.6) starts with the exploration subblock, in which exploration agents roam the network in order to find candidate segments for the addition to the dominating set. Their movement pattern is similar to the discussed movement pattern of P. rapae (Section 4.1.1). If certain conditions are satisfied, a candidate segment is added to the dominating set by agents from the construction subblock. As a result of the parallel construction operations of this block, a dominating set is produced.

While the agents in the first behavior block still explore and construct, the operations specified by the second behavior block (Section 4.7) are already executed. Agents from the exploration subblock perform random walks similar to their relatives in the previous block and in nature, but with restrictions which increase the probability to find a candidate path for the connection of two disconnected segments of the already existing dominating set. A rule which forces these agents to start only from dominating nodes is, for example, part of these restrictions. In the construction subblock, agents construct connections between dominating set segments that appear disconnected, selecting from the candidate paths found by exploration agents. To improve the quality of the solution, for instance, there are mechanisms that help to avoid the creation of redundant interconnections. As a result of these behavior blocks, a CkDS is produced.

The description of the proposed protocol is organized as follows: All necessary definitions used later in the text are introduced in Section 4.3. The local data structures and next-hop candidate ratings utilized by the protocol are specified in Sections 4.4 and 4.5. Subsequently, in Sections 4.6 and 4.7, the two behavior blocks representing the core of the protocol are described in detail.

4.3 Definitions

A connected WSN is modeled as graph, using the following definitions:

Definition 11 *An* undirected graph $G = (V,E)$ *consists of a set of vertices* V *and* E*, a set of edges* (u,v)*, where* $u,v \in V$ *and* $u \neq v$*.* (u,v) *and* (v,u) *are considered the same edge. If* $(u,v) \in E$*, then also* $(v,u) \in E$*.*

Definition 12 *A set of vertices $S \subseteq V$ in an undirected graph $G = (V,E)$ (according to Definition 11) is* connected*, if, between each pair of vertices $\{u,v\}$, with $u,v \in S$, there exists a path consisting only of vertices from S.*

The proposed protocol, similar to the related approaches, assumes bidirectional links, which are modeled as undirected edges constituting set E in $G = (V,E)$. This assumption reflects the fact that usually unicasts need to be acknowledged at MAC level, which is only possible given bidirectional links. However, real links may be unidirectional, as implied, for example, by Figure 2.8 in Section 2.1.5. To cope with this, in the real-world, the network graph is simply stripped of unidirectional links (i.e. such links are ignored).

Definition 13 *An* undirected, connected graph $G = (V,E)$ *is an undirected graph according to Definition 11 whose set of vertices V is connected under Definition 12.*

The following definitions assume an undirected, connected graph $G = (V,E)$ according to the above definition:

Definition 14 *A* path *of length l between v and u is a sequence of vertices $\langle v_0, v_1, v_2, \ldots, v_l \rangle$, such that $v = v_0$, $u = v_l$, $(v_{i-1}, v_i) \in E$, and $i = 1, 2, \ldots, l$, with $v_0, v_1, v_2, \ldots, v_l \in V$.*

Definition 15 *Two vertices $v,u \in V$ are* adjacent*, if there exists an edge $(u,v) \in E$.*

There are two *vertex states*: *dominating* and *non-dominating*.

Definition 16 *A* dominating set $D \subseteq V$ *is the set of all vertices $v \in V$ whose state is dominating.*

A connected *k*-hop dominating set (CkDS) is defined as in Definition 1 on page 1.

Definition 17 *A vertex v_c is also called a* center*, if $v_c \in$ dominating set D, and there exist three edges (v_c, v_1), (v_c, v_2), and (v_c, v_3), with $v_1, v_2, v_3 \in D$ and $v_c \neq v_1 \neq v_2 \neq v_3$.*

Different neighborhood sets are defined as follows:

Definition 18 $N_w(v)$ *contains* all *vertices in the w-hop neighborhood of vertex v.*

Definition 19 $N_w^D(v)$ *contains* all dominating *vertices in the w-hop neighborhood of vertex v.*

Definition 20 $N_w^n(v)$ *contains* all non-dominating *vertices in the w-hop neighborhood of vertex v.*

Definition 21 $N_w^c(v)$ *contains* all *vertices called* centers *in the w-hop neighborhood of vertex v.*

4.4 Local Data Structures

Each node v maintains a field $d \in \{n, D\}$, which contains the state of the node: non-dominating (n) or dominating (D). Additionally, each node administers the following tables:

◇ *Neighborhood Table* $nTab(v_n)$*:* This table includes the 1-hop neighborhood of a node. The information in this table is used for the probabilistic next-hop selection by the initial exploration and connection exploration agents (IEAs and CEAs), as described in Sections 4.6.1.3 and 4.7.1.4. An entry is associated with the neighbor v_n (identified by its address) and contains the following fields:

  ◦ s records the state of v_n, i.e. whether it is dominating or non-dominating. This information can be obtained from a received broadcast sent by a state-changing node.

  ◦ *rssi* records the received signal strength indication (RSSI) of v_n, assuming that it has been normalized adequately to reflect the approx. distance to v_n. The method for acquiring RSSI values is platform dependent.

  The addresses of 1-hop neighbors can be, for example, obtained from an underlying WSN medium access control (MAC) protocol, such as *S-MAC* [167] or *SCP-MAC* [168], which is aware of the 1-hop neighborhood, so that no additional overhead is incurred.

◇ *Interconnection Table* $iTab(v_s)$*:* In order to enable the interconnection of dominating set segments in the second behavior block, when a connectivity exploration agent (CEA) a_{CE} visits node v, a_{CE} downloads its path and center information to v's local interconnection table (see Section 4.7.1.3). When another CEA visits v, it can evaluate whether its interconnection table contains paths that lead to a dominating set segment which appears to be disconnected (see Section 4.7.1.4). For each source node v_s, the entry has the following format:

  ◦ p records the path to dominating node v_s

  ◦ *cns* contains the centers that are reachable via the dominating node v_s

  Note that $p.length$ and *p.source* return the length and the source (i.e. first) node of the path (this also applies to the path fields of agents). Further, if an entry has not been updated within t_{itu} time, it is deleted.

◇ *Next-Hop Utilization Table* $nhuTab(v_s, v_n)$*:* The utilization of next hops by connection exploration agents (CEAs) is recorded in the next-hop utilization table (see Section 4.7.1.3). After a CEA, originating from node v_s, selects its next-hop, it records its selection to this table. Thereafter, the probabilities assigned to next hops are influenced by this selection for other CEAs also originating from v_s (see Section 4.7.1.5). For a source node v_s and a next-hop neighbor v_n, the entries contain only one field:

  ◦ uf, the utilization frequency, which is initially 0.

◇ *Center Distance Table* $cdTab(v_c)$*:* This table maintains information on center nodes within $CIPA_{mh}$-hops (along dominating nodes) of a node. It serves two purposes: first, it facilitates the rating of the connectivity of a dominating node (see Section 4.7.1.2), second, it enables CEAs to recognize dominating set segments that appear to be disconnected (see Section 4.7.1.6). The entries of the center distance table, associated with a center v_c, contain only one field

  ○ d, recording the distance to v_c.

Note that not all tables are needed on all nodes. The center distance table is only maintained on dominating nodes, for instance. Moreover, most of the tables serve several purposes and may be utilized in combination: for example, a CEA evaluates the neighborhood, the next-hop utilization, and the interconnection tables to select its next hop (see Section 4.7.1.4).

In order to refer to elements of these tables, this document employs several simple notations. Here are some examples for typical usage:

◇ $v_a.nTab(v_b).rssi$ for the *rssi* field associated with neighboring node v_b in the neighborhood table of node v_a

◇ $v_a.iTab.v_s$: all source nodes in the interconnection table at node v_a

◇ $v_a.iTab.cns$: all *cns* fields in the interconnection table at node v_a

4.5 Next-Hop Candidate Ratings

In contrast to the existing CkDS approaches, reviewed in Section 3.2, the proposed approach offers a seamless integration of next-hop candidate ratings, colluding with its probabilistic next-hop selection process. The ratings take into account the quality of a link (Section 4.5.1) towards a potential next hop and its utilization properties (Section 4.5.2). Therefore, the proposed protocol achieves randomized connected coverage, while considering the quality and utilization of next-hop candidates.

4.5.1 Link Quality Rating

The link quality rating has the following objectives: First, links that are expected to lead to successful transmissions more often should be preferred over links that are expected to yield lower transmission success rates. Second, from the links which are expected to lead to high transmission success rates, the ones should be preferred that cross as much distance as possible, so that fewer hops are needed to cover a certain area of the network.

In other words, the link rating aims at maximizing the additional amount of coverage of each link in a random walk, while, at the same time, it avoids links with low transmission success rates. Since a successful transmission implies a successful reception and vice versa, I will use the term *reception success* to describe a successful transmission and reception of a frame or packet, as it is more common in the community.

Before introducing the proposed link rating, related, preparatory work by other authors needs to be reviewed briefly: As described in [156] and [169], two correlations can be observed:

First, there is a positive correlation between received signal strength and reception success rate. Naturally, at the same time, a negative correlation between received signal strength and distance can be found. Note that for the sake of brevity, I will frequently use the abbreviation RSSI, short for *received signal strength indication*, to refer to received signal strength.

The relationship between distance and reception success rate is depicted in Figure 4.4, based on the data from [156] for Berkeley Mica motes. In the figure, the rate of reception success is plotted as a function of distance. To obtain the data, the authors positioned a grid of nodes in an open tennis court at two feet (60.96 cm) distance in both dimensions. The nodes transmitted 200 packets at a rate of eight packets per second, with only one transmitter active at a given time and the remaining nodes receiving. It is evident from the figure (a and b are marked in the chart) that the distances can be divided into three categories:

0 to a In this region, the probability that a frame/packet will be received successfully is very high. Thus this region should be preferred.

a to b Within these limits, there is an acceptable probability that a packet will be received successfully.

b to ∞ In this area, the transmitted packet is likely not to be received. Therefore it should be avoided.

From the description above, it is evident that

a represents a distance threshold below which the reception success rates are excellent. Since there is a negative correlation between distance and signal strength, the RSSI value corresponding to this distance can be regarded as a threshold, labeled r_{pr}, above which the reception success can be expected to be excellent and therefore links with RSSI values greater r_{pr} should be *preferred*.

b identifies a distance threshold below which the reception success rates are acceptable. As there is a negative correlation between distance and signal strength, the RSSI value corresponding to this distance can be regarded as a threshold, labeled r_{ac}, above which the reception success can be expected to be acceptable and therefore links with RSSI values greater r_{ac} should be *accepted*.

The proposed link quality rating takes into account these considerations by categorizing links into three rating classes according to their RSSI values r and the thresholds r_{ac} and r_{pr}. Depending on this categorization, the links are rated in a different manner. More concretely, I propose the following *link quality rating*:

$$qr(v_n) = \begin{cases} \max((\frac{r_{pr}}{r})^\gamma, \alpha) & \text{if} \quad r \geq r_{pr} \\ \beta & \text{if} \quad r_{pr} > r \geq r_{ac} \\ 0 & \text{else} \end{cases} \tag{4.1}$$

using the RSSI value $r = nTab(v_n).rssi$ of the link to node v_n, $\gamma > 0$ to adjust the steepness of the function, and r_{pr}/r_{ac} as RSSI thresholds for links which are preferred/accepted, as they can be expected to exhibit high/acceptable reception success rates. The influence of the different parameters (α, γ, r_{pr}, r_{ac}, and β) on the rating is illustrated in Figure 4.5.

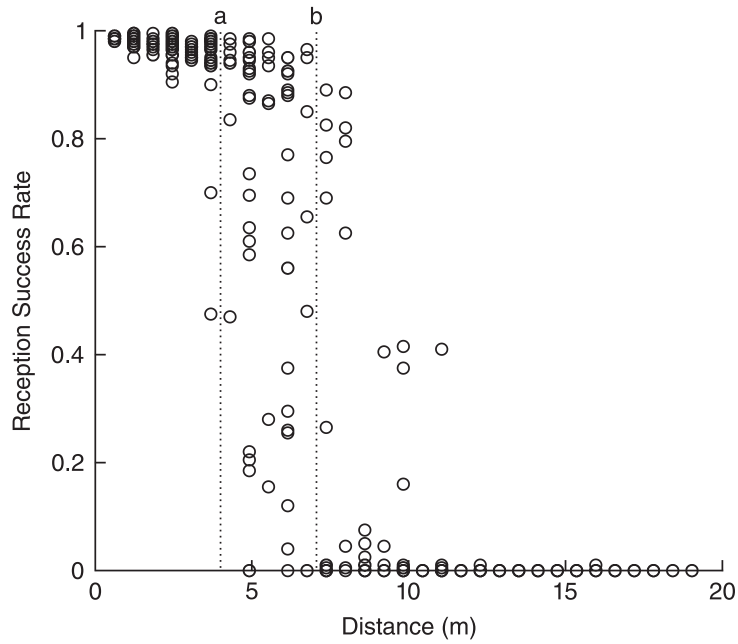

Figure 4.4: Reception success rate as a function of distance. Data source: [156]

As long as $r \geq r_{pr}$, the underlying idea is to favor links with lower RSSI, as it is likely that they bridge longer distances. The minimum value produced by the rating, as long as $r \geq r_{pr}$, is $\alpha \in (\beta, 1]$, to always assess them better than more unreliable links with $r < r_{pr}$. r_{pr} is represented by the dotted line, marked with a, in Figure 4.4.

r_{ac} represents a threshold above which lower, however, still to a certain extent acceptable reception success rates can be expected, so that a lower constant value is assigned ($\beta \in [0, 1)$). In Figure 4.4, r_{ac} is depicted using the dotted line marked with b.

All non-acceptable links, for which it can be expected that they yield too low reception success rates to be useful, are assigned 0 as rating.

When looking at the observations from [156] and [169], given the nature of the communication channel and the relative weakness of the correlation, it is clear that the above rating can only approximate the actual link properties.

4.5.2 Utilization Rating

In the second behavior block, described in Section 4.7, the dominating set segments created by the first behavior block are interconnected. To realize this, exploration agents are dispatched from dominating nodes in order to search for other dominating segments that appear to be disjoint, so that these can be connected subsequently. An example of a state prior to interconnection is depicted in Figure 4.8 (a).

As long as an exploration agent from the second behavior block does not find a trace towards a dominating set segment that it considers disconnected from the dominated set segment it originated from, it uses a random walk to move through the network. However, to reduce the number of explorations needed in the second behavior block, the agent's strategy aims at spreading the random walks more evenly over the network topology, by reducing the probability of repeating previous choices (see Section 4.7.1.5).

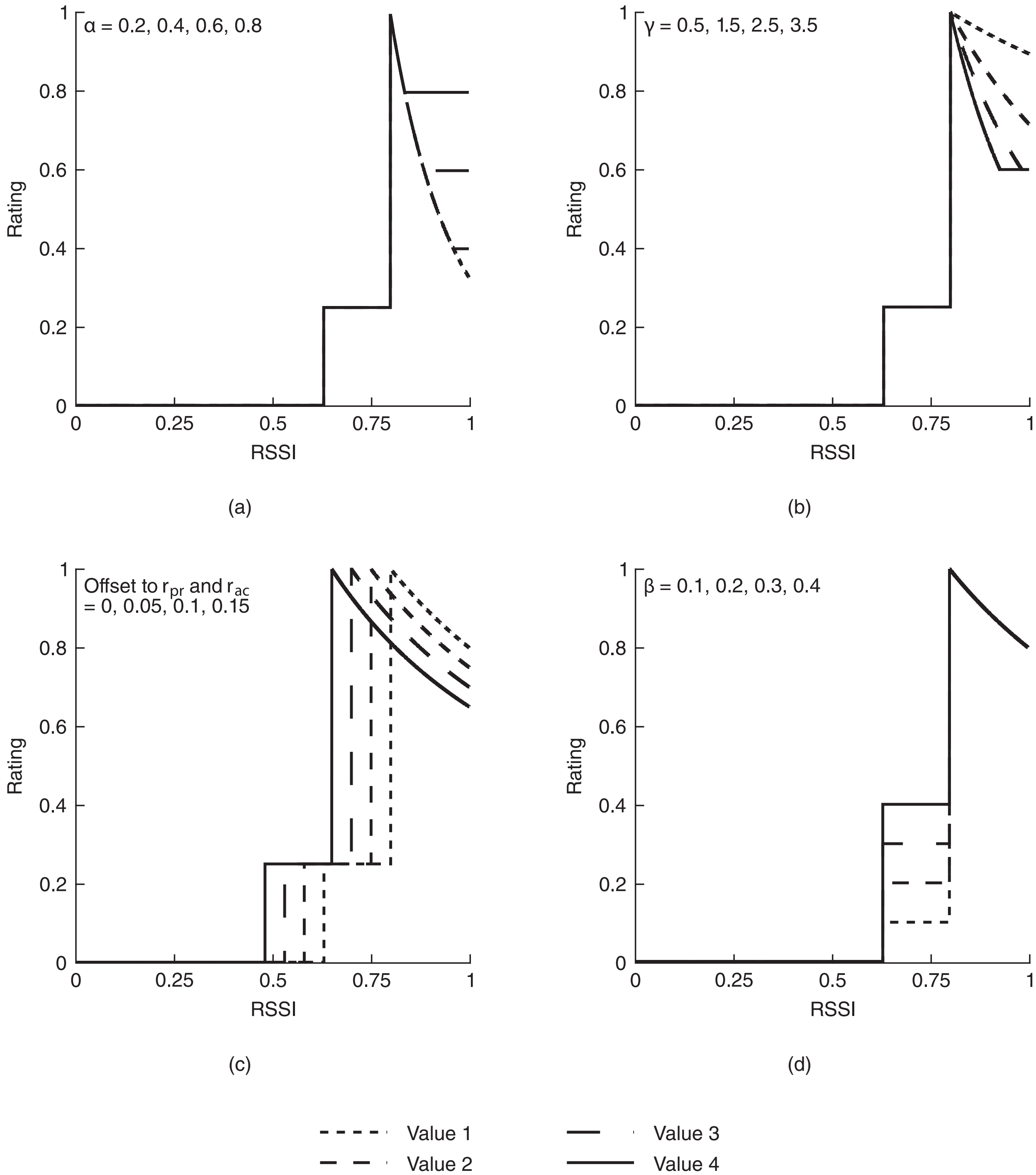

Figure 4.5: Rating results as a function of normalized RSSI for different parameters: (a) α, (b) γ, (c) offset to r_{pr} and r_{ac}, and (d) β

Consider the situation depicted in Figure 4.8 (b): the exploration agent originating from node 13 has now arrived at node 14. Assume that the candidates for next-hop selection are nodes 15, 16, 17, and 18. If, for example, node 17 has been used as next hop three times and all of the other nodes only once, the underlying idea of the utilization rating is to increase the probability of choosing one of the less-often selected next-hop candidates.

In order to realize the above idea, the *utilization rating* is defined as follows:

$$ur(v_s, v_n) = 1 - \frac{v.nhuTab(v_s, v_n).uf}{\sum_{v_i \in S_c} v.nhuTab(v_s, v_i).uf} \tag{4.2}$$

with

- ◇ v, the current node visited by the agent,
- ◇ v_n, the rated next-hop candidate,
- ◇ v_s, the node at which the agent was generated,
- ◇ S_c as defined in Equation 4.6 (see Section 4.6.1.3). It represents the neighborhood of a node from which, if possible, nodes that would lead an agent towards previously visited regions were removed.

The probability to select v_n consequently declines with higher previous utilization. v_s has to be included in the rating, in order to take into account the fact that exploration agents in behavior block II may originate from any dominating node. Not doing so would lead to highly non-linear walks, since the trajectory of an agent would be influenced by the utilization traces ($v.nhuTab$) of agents that approached the current node from a different direction.

4.5.2.1 Integration with Link Quality Rating

To be applicable, the utilization rating is integrated with the link quality rating to an *extended rating* for a link from v to v_n, considered by an agent generated at v_s:

$$er(v_s, v_n) = qr(v_n)^{\omega} \cdot ur(v_s, v_n)^{\varpi} \tag{4.3}$$

with ω and ϖ used for tuning the influence of qr and ur.

4.6 Behavior Block I: Initial Dominating Set Construction

The first behavior block is divided into two behavior subblocks: exploration and construction (Sections 4.6.1 and 4.6.2). It needs to be emphasized that as behavior blocks I and II work in parallel, also their subblocks, exploration and construction, are executed in parallel.

In the first subblock, in order to enable the protocol to select nodes for the dominating set, first, a swarm of agents explores the network area. Exploration agents start in a probabilistic manner from different nodes (Section 4.6.1.1). The swarm of these agents determines paths, from which it selects some to be added to the dominating set. For this exploration method, the proposed approach draws inspiration from the flight behavior of ovipositing P. rapae. It imitates its random walks by employing a probabilistic next-hop selection function (Section

4.6.1.3). Further, a tendency towards linearity similar to P. rapae's is achieved by integrating a multi-hop path straightening method (Section 4.6.1.3).

To adapt the imitated behavior to the properties of the artificial system, i.e. the WSN, further rules to the proposed behavior are needed: To use long-range links with high reception success rates, a link quality rating (Section 4.5.1) is included in the next-hop selection process. Further, rules that are only necessary in the artificial system determine under which conditions and how nodes are added to the dominating set (Section 4.6.2.1). Finally, in order to enable the second behavior block to connect dominating set segments that were added by this block and are considered disjoint, the description specifies how the necessary information is provided (Section 4.6.2.3).

4.6.1 Exploration

Within this subblock, agents explore the network using random walks, thereby defining paths, which serve as candidates for the addition to the dominating set. There are two advantages to considering entire paths as candidates instead of single nodes like in the approaches from state of the art [130, 143, 165, 166]:

- ◇ When looking at a CkDS, such as the one depicted in Figure 2.12, one can intuitively interpret the structure as an accumulation of numerous intersecting paths. The proposed protocol exploits this observation by deciding whether to add entire paths instead of single nodes to the dominating set. Since each path typically consists of multiple nodes, this design choice aims to reduce the number of marking decisions and thereby the overall cost of the process. It is also a point at which the devised protocol closely resembles its natural archetype, since each path can be regarded as a random walk by a P. rapae female.

- ◇ Naturally, there must be rules to select which candidate nodes to add to the dominating set. If a protocol operates on a per-node basis, the vicinity of a node considered as candidate within a certain number of hops, reflecting the dominating distance k, needs to be known in order to provide enough information for these rules. In contrast, when paths are utilized as candidates, the length of the paths already implies distance information, so that it does not have to be obtained explicitly. In other words, the length of a path can be made use of to decide, whether to add a set of candidate nodes constituting the path to the dominating set—*without* knowing their multi-hop neighborhood. Thus only a minimum amount of information is required for this decision, since paths are established through a random walk consisting of unicasts, which translates to low costs in terms of communication and thus less energy consumed. Moreover, it makes the cost of candidate selection virtually independent of the node degree and the desired dominating distance, which is also confirmed by the simulation results in Chapter 5.

To produce the swarm, an *initial exploration agent* (IEA) a_{IE} is generated at each node v immediately after the protocol starts its operation. The role of each of the agents is to create information by modifying its own state and the state of visited nodes, as well as, to evaluate information present at nodes to draw conclusions from it. By this, agents do not communicate with each other directly but using stigmergy.

4.6.1.1 IEA Departure Time

In order to enable an evaluation of useful information, the agents' activities need to be dispersed over time. Else, if agents roamed the network exactly at the same time, there would not be enough information existing already that could be made use of. Therefore, agents determine their departure time from the node of their creation, v, using the function

$$t_{IEA} = \begin{cases} t_c + random() \cdot t_{IEAmd} & \text{if} \quad random() \leq p_{dIEA} \\ \infty & \text{else} \end{cases} \tag{4.4}$$

with

- ◇ t_c: the current time
- ◇ $random() \in [0,1]$: a function generating random numbers
- ◇ p_{dIEA}: the departure probability
- ◇ $t_{IEAmd} > 0$: a maximum delay

$t_{IEA} = \infty$ corresponds to the death of the agent. According to the above function, the agent departs at a random time between now and t_{IEAmd} with probability p_{dIEA}. The probability p_{dIEA} was introduced, since it became evident, after experiments, that it was sufficient to start IEAs from only a subset of all nodes.

An IEA a_{IE} has the fields $\langle p, ts \rangle$:

- ◇ p: recording the path traveled, so that every visited node is added to p.
- ◇ ts: denotes the tabu set, which consists of the IDs of nodes in the 1-hop neighborhood of the agent's previous hops, as well as, the previous hops themselves, added every time before leaving a node. ts_{mnn} and ts_{mph} specify the maximum size of ts in number of nodes, $ts.nn$, and previous hops, $ts.ph$. Thus, for example, with $ts_{mph} = 2$, $a_{IE}.ts$ of IEA a_{IE} that visited the sequence of nodes v_a, v_b, v_c, v_d, after leaving v_d, will contain $N_1(v_c) \cup N_1(v_d) \cup v_c \cup v_d$, assuming a sufficiently large $ts_{mnn} \geq |N_1(v_c) \cup N_1(v_d) \cup v_c \cup v_d|$. If $ts.nn > ts_{mnn}$ or $ts.ph > ts_{mph}$, nodes are deleted from ns in the order of their insertion.

4.6.1.2 IEA Departure Procedure and Structure

Before leaving a node v, an IEA a_{IE} checks, if any other IEA has left v within the last t_{mw} time, i.e. the maximum expected walk time. If the condition is satisfied, a_{IE} sleeps for t_{mw}, wakes up, and then the check and subsequent actions repeat.

Further, there is a second check: before leaving v, a_{IE} checks if $v \in D$ (that is, if v is dominating). If this is true, it dies (i.e. is deleted), in order to decrease the risk of creating a too dense dominating set, which exhibits only a low coverage per dominating node, in this area.

The above mechanisms and Equation 4.4 are important and aim at reducing the amount of concurrency during the exploration process, thereby lowering communication cost and improving the overall result quality. Note that they are related to the second rule in the construction subblock (see Section 4.6.2.1).

4.6.1.3 IEA Next-Hop Selection

If $|N_1^D(v)| = 0$, so that none of v's neighbors is dominating, an IEA a_{IE} selects its next hop v_x at node v with the probability

$$p(v_x) = \frac{qr(v_x)}{\sum_{v_i \in S_c} qr(v_i)} \tag{4.5}$$

using the link quality rating qr from Equation 4.1 in Section 4.5.1. Else, if $|N_1^D(v)| > 0$ (at least one of v's neighbors is dominating), a_{IE} randomly selects a node from the set of dominating 1-hop neighbors, $N_1^D(v)$, as next hop. This **sticky behavior** aims at avoiding the creation of parallel segments of the dominating set, since they provide only little additional coverage but increase its size considerably.

The set of next-hop candidates S_c is computed at node v using the following function, assuming tabu set $S_t = a_{IE}.ts$, $N_1(v)$ obtained from $v.nTab.v_n$, previous hop v_p from $a_{IE}.p$:

$$S_c = \begin{cases} (N_1(v) \setminus S_t) \setminus v_p & \text{if} \quad |(N_1(v) \setminus S_t) \setminus v_p| \geq \eta \\ N_1(v) \setminus v_p & \text{else if} \quad |N_1(v) \setminus v_p| \geq \eta \\ N_1(v) & \text{else} \end{cases} \tag{4.6}$$

The function increases the size of the selection depending on whether there are enough ($\geq \eta$) candidates. If $N_1(v) = \emptyset$ for any reasons, such as the failure of the underlying MAC protocol to provide neighborhood information, a_{IE} dies.

Example The example in Figure 4.6 illustrates the exploration process of behavior block I. In Figure 4.6 (a), IEA a_1 has started from node 1, being now after the second hop. Shortly after that, IEA a_{31} has started from node 31, now after its first hop. Both IEAs are fully unaware of each other, and the only interaction takes place, if they find hints of each other's visits, i.e. by employing stigmergy.

Figure 4.6 (b) shows the state of the exploration after another two hops. IEA a_2 started from node 2, now being after its second hop. IEAs a_1 and a_{31} are now after their fourth and third hops.

Five hops later, Figure 4.6 (c) depicts the current state of the exploration process with IEA a_{31} after its eighth and IEA a_1 after its ninth hop, arriving at node 3. IEA a_2, after its seventh hop, arrives at node 7, which was visited by a_1 within the last t_{mw}. Therefore, a_2 starts sleeping for t_{mw}, since it can be expected that a_1 triggers a construction for the path it traveled, starting from node 1 until arriving at node 3. Notice that instead of communicating directly, the agents a_1 and a_2 communicate indirectly via stigmergy.

4.6.2 Construction

In the exploration subblock, random walks were used by the agents to create candidate paths for the addition to the dominating set. Within the construction subblock, agents apply rules to decide which candidate path to select and subsequently conduct the addition of the selected candidate to the dominating set.

Figure 4.6: Exploration in behavior block I

4.6.2.1 Trigger, ICA Generation and Structure

The decision to add nodes to the dominating set D is produced autonomously by an IEA using only local information. An IEA a_{IE}, visiting node v, decides to construct a dominating set segment, if one of the following conditions is satisfied

1. $a_{IE}.p.length \geq IEA_{wsl}$, with the walk segment length IEA_{wsl}
2. $v \in D$ and $a_{IE}.p.length \geq IEA_{mdcbd}$, with the minimum dominating contact build distance $IEA_{mdcbd} \leq IEA_{wsl}$.

If one of these conditions is satisfied, an *initial construction agent* (ICA) a_{IC} is generated. The initial construction agent a_{IC} contains a path field p, which is initialized by setting it equal to a_{IE}'s path field ($a_{IC}.p = a_{IE}.p$). If v is not dominating ($v \notin D$), a_{IE} continues its walk after its path field has been cleared. Else, it dies.

The information that is available to the agent includes its path and the neighborhood of the currently visited node. Thus intuitively, it suggests itself to utilize this information to decide which candidate path to add to the dominating set. However, such a decision cannot be made after any arbitrary number of hops. In order to understand why, the technique for adding nodes to the dominating set should be discussed first.

Assume that an IEA a has visited the path $a.p = \langle v_0, v_1, \ldots, v_l \rangle$, with $a.p.length = l$. If it decides to add the nodes in $a.p$ to the dominating set, a very simple, yet effective solution is that it creates an ICA a' which is responsible for visiting the sequence of nodes $\langle v_l, v_{l-1}, \ldots, v_0 \rangle$ and marking each node $\in a'.p = a.p$ as dominating. However, as the length of $a.p$ increases, the probability also grows that a or a' will disappear due to a communication error, for example, if the reception of the frame containing a or a' fails after encountering an uncorrectable amount of flipped bits. Therefore, to reduce the probability of failure, the underlying idea of the *first* rule is to split up the walk of a into segments of the length IEA_{wsl}. In other words, after a walked IEA_{wsl} hops, it creates ICA a' to walk back $a'.p = a.p$ and to add all visited nodes to the dominating set, while a clears its path field and continues its walk. As discussed extensively at the end of this chapter (see Section 4.8), the choice of IEA_{wsl} has implications on the effective maximum dominating distance (k', see Definition 22 on page 101), achieved after the construction of the CkDS concludes.

After motivating the first rule, the *second* one needs to be considered. It complements the rules specified in the exploration subblock which delay the departure of an IEA from a node, if it has been visited by another IEA within a certain amount of time, or let the IEA die, in case the visited node is dominating (see Section 4.6.1.2). Both of these rules aim to curb the amount of redundancy of dominating nodes in an area, by stopping the walk of an IEA, in case it visits a node that is dominating or a candidate for being added to the dominating set. However, they do not specify what happens to an IEA a, visiting a dominating node, before it dies, in other words, how its recorded path $a.p$ is utilized. It is important to make use of $a.p$, since else the cost incurred by a would not yield much benefit. Here the second rule is applied to realize this. When the path walked by a which currently visits a dominating node exceeds a certain length, that is if $a.p.length \geq IEA_{mdcbd}$, it creates ICA a', setting $a'.p = a.p$, before dying. Subsequently a' walks along $a'.p$ adding all visited nodes to the dominating set. Similar to IEA_{wsl}, the choice of IEA_{mdcbd} has implications on the dominating distance (k', see Definition 22 on page 101) of the final CkDS, as discussed at the end of this chapter (see Section 4.8).

4.6.2.2 Addition of Nodes to the Dominating Set by ICA

a_{IC}, whose creation and initialization was described above (see Section 4.6.2.1), follows the node sequence stored in its path field, $a_{IC}.p$, marking each node on its path (including v) as dominating, before it dies. If, visiting node v_i, the next hop in the path $a_{IC}.p$ after v_i does not exist in $v_i.nTab$, a_{IC} dies immediately.

Each node marked as dominating announces its new state to its 1-hop neighbors using broadcast. This allows its neighbors to realize the sticky behavior described further above (see Section 4.6.1.3). Note that the size of the broadcast packet is minimal, since only the ID of the sender and its new state have to be announced, as well as, that this is the *only* point at which a broadcast is employed in this protocol. Therefore, the *total* number of broadcasts used by the protocol equals exactly the size of the dominating set D.

Example Five hops after Figure 4.6 (b), Figure 4.6 (c) depicts the current state of the exploration process with IEA a_{31} after its eighth and IEA a_1 after its ninth hop, arriving at node 3. In the example, the IEAs' walk segment length $IEA_{wsl} = 9$ and the minimum dominating contact build distance $IEA_{mdcbd} = 6$.

In Figure 4.6 (c), the length of the path traveled by a_1 is equal to the walk segment length, $a_1.p.length = IEA_{wsl}$, at node 3 (condition 1 satisfied). Thus node 3 generates an ICA a_3, initializing it with a_1's path, setting $a_3.p = a_1.p$ (i.e. $a_3.p$ assumes the value of $a_1.p$). Accordingly, ICA a_3 starts following $a_3.p$ towards 1, marking all visited nodes as dominating. At the same time, a_1 continues its walk after clearing its path field, with $a_1.p = \langle\rangle$, from node 3.

The result of the operations described above is shown in Figure 4.7 (a) after three additional hops. ICA a_3 has now marked four nodes as dominating, arriving at node 5. Similarly, IEA a_{31}, which started at node 31 and is now at node 36, reached its walk segment length at node 32 two hops ago and triggered an ICA a_{32}, now at node 33 heading towards 31.

Figure 4.7 (b) depicts the scenario after six additional hops. ICA a_3, dispatched at node 3, now has reached node 1, adding all nodes on the visited path to the dominating set. Similarly, ICA a_{32} is now at node 34 and has only one additional hop to follow. The IEAs a_{31} and a_1 reached nodes 37 and 8. a_1 starts sleeping, since IEA a_2 visited node 8 within the last t_{mw}. Notice that while the construction is already in progress, new IEAs may start, such as IEA a_{35} from node 35, now after its third hop.

Being at *dominating* node 7 in Figure 4.7 (b), IEA a_2 wakes up after sleeping for t_{mw} and compares its path length to the minimum dominating contact build distance. Since $a_2.p.length \geq IEA_{mdcbd}$ (condition 2 satisfied), node 7 creates an ICA a_7 copying the path from IEA a_2 ($a_7.p = a_2.p$). Thereafter, a_2 dies, since being at a dominating node, and ICA a_7 starts its walk towards node 2 according $a_7.p$.

Seven hops later, the example scenario is depicted in Figure 4.7 (c). ICA a_7, now at node 2, followed its path $a_7.p$, starting at node 7 and marking all visited nodes as dominating; a_7 dies, since it achieved its objective. ICAs a_{42} and a_{11}, now at nodes 40 and 41, were created at nodes 42 (condition 1) and 11 (condition 2). They currently follow their paths, marking visited nodes as dominating. Moreover, the actions by ICAs a_7 and a_{11} lead to the emergence of centers 7 and 11.

In Figure 4.7 (c), at node 42, IEA a_{31} continued its walk after clearing its path field. When IEA a_{31} migrates from node 38 to node 39, for the purpose of illustration, a communication error occurs due to e.g. an uncorrectable amount of flipped bits, leading to the death of a_{31}

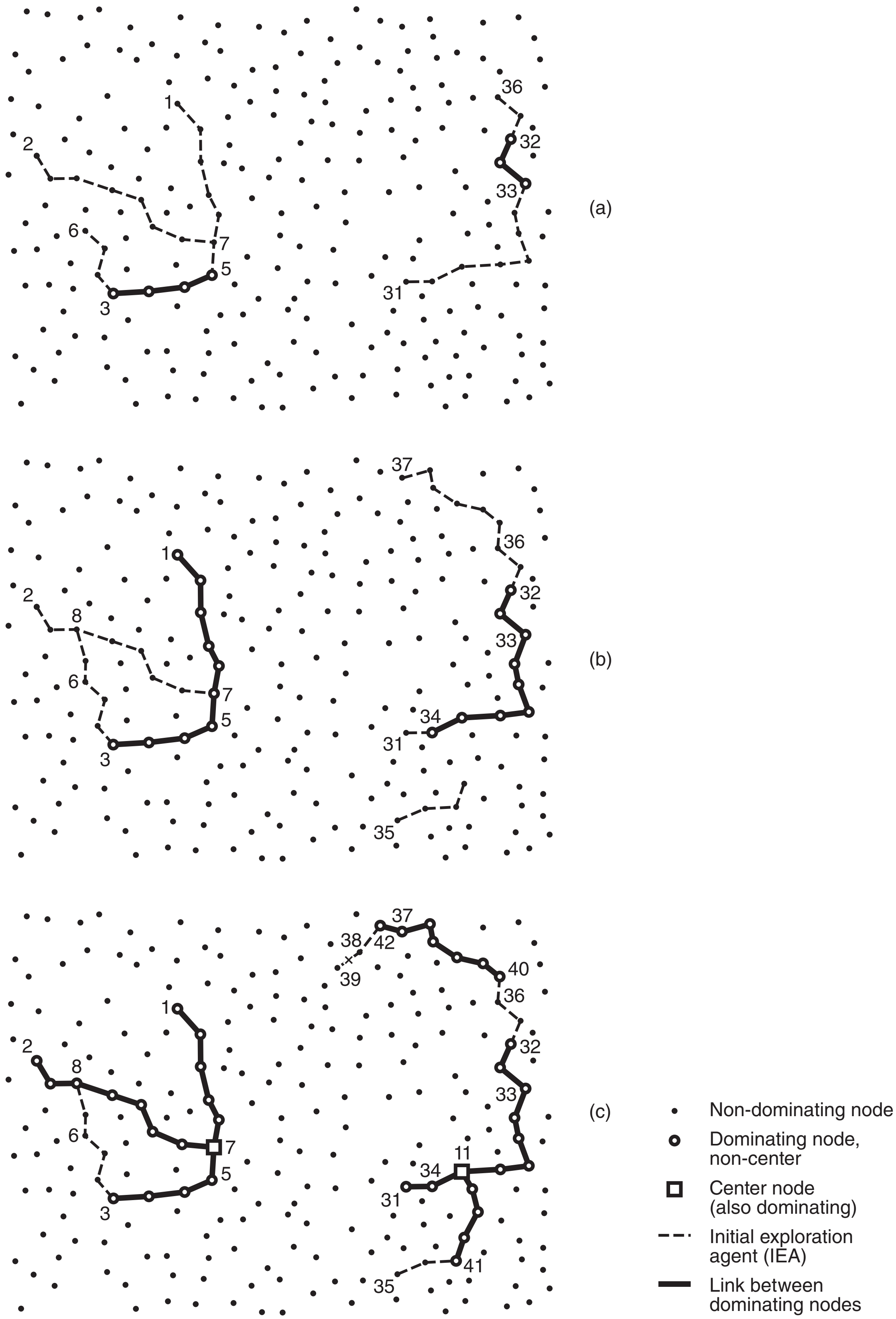

Figure 4.7: Exploration and construction in behavior block I

(as evident from the figure, this is fully tolerated by the protocol). At node 8, IEA a_1 wakes up after sleeping for t_{mw}, but the length of the path it traveled is lower than the minimum dominating contact build distance, i.e. $a_1.p.length < IEA_{mdcbd}$: it dies without triggering any further actions. Assuming that no new agents are created in behavior block I, the construction efforts from Figure 4.7 (c) lead to the situation depicted in Figure 4.8 (a). Notice that this clear separation between behavior blocks serves only the purpose of illustration and that, in a simulated or real-world execution, behavior blocks I and II operate completely in parallel.

4.6.2.3 Propagation of Center Information by CIPA

If a node v_c has become a center, it generates a *center information propagation agent* (CIPA) a_{CIP}. Its objective is to propagate center information within $CIPA_{mh}$ distance along dominating nodes. This information is used in the second behavior block to assess the local connectivity and to locally recognize disconnected segments of D (see Sections 4.7.1.2 and 4.7.1.4). a_{CIP} consists of the fields: $\langle src, pre, h \rangle$, which represent the source center v_c, the previous hop of a_{CIP}, and a hop counter, initialized to 0.

Arriving at node v_d, if $a_{CIP}.h > CIPA_{mh}$ or $a_{CIP}.src$ exists in $v_d.cdTab$, a_{CIP} dies. Else and upon its creation, being at node v_d, it selects all nodes from $N_1^D(v_d) \setminus a_{CIP}.pre$ as next hops, replicating adequately. Further it leaves a copy of itself a_{CIPs} sleeping at the current node. If $|N_1^D(v_d)|$ of a node v_d increases, a_{CIPs} wakes up and sends a copy of itself, named a_{CIP}, to the new member of $N_1^D(v_d)$, before going to sleep again. At each node v_d that a_{CIP} visits, $v_d.cdTab(a_{CIP}.src).d = a_{CIP}.h$.

Example Assuming $CIPA_{mh} = 10$, in Figure 4.7 (c), a CIPA a_7', generated by the new center 7, reaches all dominating nodes within its connected dominating segment, including, for example, nodes 1 and 3.

It should be remarked that at this point no precise statement can be made about the state of the construction, since both behavior blocks and their subblocks are fully executed in parallel. Therefore I discuss the properties of the produced structure after the construction process concludes, in Section 4.8, at the end of this chapter.

4.7 Behavior Block II: Transformation to a Connected k-Hop Dominating Set

The second behavior block, which is intertwined with the first one, connects the dominating set segments added in the first block to a CkDS. Similar to the first behavior block, the second is subdivided into two subblocks, which are executed in parallel, realizing exploration and construction.

In the first subblock (Section 4.7.1), a swarm of agents explores the network area to find remote, disconnected segments of the dominating set. The agents, drawing inspiration from the flight behavior of ovipositing P. rapae, move through the network similar to exploration agents from the first behavior block and their natural counterparts, employing a probabilistic next-hop selection, given certain conditions (Section 4.7.1.5). Nevertheless, since the objective is, in contrast to the first behavior block, to connect already existing segments of the dominating set as efficiently as possible, there are four main adjustments, which will be described in full detail after the following overview:

1. Exploration agents start and conclude their walks only at dominating nodes, since their primary goal is to find connections between disconnected dominating set segments (see Section 4.7.1.1).

2. Their generation rate (see Section 4.7.1.1) depends on local connectivity properties, which are rated to assess the necessity of generation. For this, center nodes (nodes with more than two dominating 1-hop neighbors) are regarded as landmarks (see Section 4.7.1.2). Based on the landmarks' number and distance to the current node, the complexity of the surrounding dominating set segment is estimated. The method aims at reducing cost, where it is adequate, by decreasing the number of agents generated.

3. The exploration agents' random walks exhibit a preference for nodes that were previously selected less often as next hops (see Section 4.7.1.5). This strategy has two aims: reduction of the total number of walks and therefore also the reduction of communication cost.

4. Next-hop selection includes a greedy component that lets exploration agents walk directly towards dominating set segments that are considered disjoint (see Section 4.7.1.6), to further save communication cost. The information necessary to recognize and approach segments that appear to be disjoint is supplied using stigmergy (see Section 4.7.1.3; for an overview of stigmergy refer to [145]).

In order to enable the use of stigmergy in points (3) and (4), agents need to carry and deposit additional connectivity information on nodes they visit (see Section 4.7.1.3).

Finally, for the second subblock (construction, see Section 4.7.2), to obtain a CkDS based on the above preparatory work, the necessary rules are specified on how nodes are selected for addition to the dominating set and in what way this is carried out.

4.7.1 Exploration

Exploration is realized using *connection exploration agents* (CEAs), which have the objective to explore connections between disconnected segments of the dominating set D.

4.7.1.1 Generation Times of CEAs

For the first time, at node v_d, a CEA is generated t_{CEA} time after v_d has become member of D. The intervals between the first and further generations of a CEA at v_d are defined as follows:

$$t_{CEA} = \begin{cases} t_{CEA} \cdot (\beta_1 + \beta_2 \cdot cr(S_c')^{\sigma}) & \text{if} \quad t_{CEA} \leq t_{CEAmg} \\ \infty & \text{else} \end{cases} \tag{4.7}$$

with,

◇ $t_{CEA} > 1$: the current interval between two successive CEA generations. To compute t_{CEA}, its previous value is multiplied by a factor which is greater than 1.

- ◇ $\beta_1 > 1$: a basic summand, representing the value that is at least used for multiplication with the previous value of t_{CEA}.
- ◇ S'_c: the set of centers from v_d's center distance table ($S'_c = v_d.cdTab.v_c$, see Section 4.4)
- ◇ cr: connectivity rating used to assess the amount of connectivity and thus also the need for further connections, defined in Equation 4.8 in Section 4.7.1.2.
- ◇ $\beta_2, \sigma \geq 0$: weights for cr
- ◇ t_{CEAmg}: the maximum generation time of a CEA

The design of the function is based on the following ideas and considerations: The connection efforts can only start after a node has become dominating. Therefore, the first CEA is generated t_{CEA} after this event. t_{CEA}'s initial value is defined before the start of the protocol. It should be much greater than the typical time needed by an agent to migrate between nodes, so that a certain amount of structure is already present at the time of the first agent generation, and centers, serving as *landmarks* in the connectivity rating, are likely to be already established.

During the evolution of the protocol, it became clear that usually a single CEA per dominating node is not sufficient. Thus CEAs are generated one or multiple times at each dominating node and, in Equation 4.7, t_{CEA} is updated after each generation. The function prolongs the intervals between subsequent generations in the described manner for two reasons:

- ◇ To enable a rapid interconnection in the beginning.
- ◇ To spread connectivity efforts over the available time period, so that information which has been deposited in the network in the meantime can be utilized through stigmergy to create interconnections, if this is still necessary at a later time.

Finally, the time between agent generations also depends on the evaluation of the surrounding dominating set segment by the connectivity rating, which is described next.

4.7.1.2 Connectivity Rating and CEA Structure

In order to estimate the connectivity, the protocol uses the following *connectivity rating* at node $v_d \in D$:

$$cr(S'_c) = \sum_{c_i \in S'_c} \max\left(CIPA_{mh} - v_d.cdTab(c_i).d, 0\right) \tag{4.8}$$

As discussed in Section 4.6.2.3, $CIPA_{mh}$ determines the maximum number of hops a center information propagation agent (CIPA) may travel. Therefore, this constant also defines the size of the dominating environment in which centers are known, or visible. The maximum function in Equation 4.8 has the purpose to avoid negative summands in case, for example, due to changes in the parameters, a center is believed to be more than $CIPA_{mh}$ hops away from the current node v_d.

The underlying idea of this function is to employ centers as **landmarks**, which provide cues about the connectivity of the dominating neighborhood: the more and the closer to v_d the centers in S'_c are, the denser the dominating set in v_d's vicinity, and thereby the lower the local need for connection exploration is estimated. In other words, the constellation of centers, i.e. their distance and number, around v_d is interpreted as indication for the degree of connectivity in v_d's neighborhood and necessity, as well as, stimulus for further exploration.

A CEA a_{CE}, generated at node v_d, contains the fields $\langle p, ck, ts \rangle$:

- ◇ p: the path traveled by the CEA, initialized by setting $a_{CE}.p = \langle\rangle$.
- ◇ ck: the centers known at the generation node v_d, initialized by $a_{CE}.ck = v_d.cdTab.v_c$. This information is central to enable the local recognition of disjoint dominating set segments, outlined in Section 4.7.1.4.
- ◇ ts: the tabu set

After being generated and initialized, CEA a_{CE} leaves node v_d as described further below (see Section 4.7.1.4).

4.7.1.3 Preparatory Work for Connectivity Exploration

During their walk through the network, CEAs maintain their own data structures but also data structures at the visited nodes, i.e. the interconnection and the next-hop utilization tables, in order to enable succeeding agents to exploit this information through stigmergy mainly for the following purposes:

- ◇ The local recognition of disjoint dominating set segments, which is important to trigger the greedy part of a CEA's behavior (see Section 4.7.1.6).
- ◇ To enable a CEA to move towards a dominating set segment locally recognized as disjoint, when behaving in a greedy manner (see Section 4.7.1.6).
- ◇ To allow the probabilistic part of a CEA's behavior to balance the utilization of next-hop candidates (see Section 4.7.1.5).

In order to conduct the preparatory work for the above functionality, which will be described further below (see Section 4.7.1.4), on every visited node v, a_{CE} adds v to its path field $a_{CE}.p$, recording the path for later use. It further adds/updates the interconnection table entry $v.iTab(a_{CE}.p.source)$ using the information it carries (i.e. $a_{CE}.p$, $a_{CE}.ck$). This makes it possible for succeeding agents to recognize a center in the interconnection table as unknown and to obtain a path towards it.

Further, when CEA a_{CE} leaves node v to migrate to next-hop node v_n, it adds the $v.nhuTab$ $(a_{CE}.p.source, v_n)$ entry, initializing $v.nhuTab(a_{CE}.p.source, v_n).uf = 1$ or, if already existing, sets $v.nhuTab(a_{CE}.p.source, v_n).uf = v.nhuTab(a_{CE}.p.source, v_n).uf + 1$. This information provides the basis for the probabilistic and deterministic next-hop selection methods (described below in Section 4.7.1.4) employed by CEAs.

4.7.1.4 Outline of Next-Hop Selection during Connectivity Exploration

For next-hop selection, a CEA distinguishes the following two cases, which are discussed in more detail further below:

1. A disjoint dominating set segment *is not* recognized based on the interconnection table and the agent's center information, or the agent is before its first hop (see Section 4.7.1.5). In this case, it follows a random walk taking into account link quality and utilization.

2. A disjoint dominating set segment *is locally* recognized based on the interconnection table and the agent's center information, and the agent is not before its first hop (see Section 4.7.1.6). Thus, it randomly selects from the locally recognized disjoint dominating set segments that appear to be the closest.

To recognize a dominating set segment as locally disconnected, a CEA a_{CE} basically compares its field containing known centers, $a_{CE}.ck$ (see Section 4.7.1.2), with the information available in a node v's interconnection table, *v.iTab.cns* (see Section 4.4), deposited by agents previously visiting node v (see Section 4.7.1.3). If a_{CE} finds centers in *v.iTab.cns* that it does not know, it assumes that they will probably belong to a dominating set segment that is not connected to the dominating set segment which it departed from. Since, however, all information available is local, i.e. describing state only within the neighborhood of a limited number of hops, this recognition is not reliable and therefore called a *local* recognition. In other words, although a dominating set segment is considered (or apparently) disjoint, there is no guarantee for this property. However, that does not compromise the operation of the proposed protocol and is an inherent part of its nature. Before further discussing this issue in Section 4.7.1.7, the recognition mechanism, as well as, the probabilistic and the greedy parts of a CEA's behavior, triggered depending on the recognition, are described in the following paragraphs.

4.7.1.5 Next-Hop Selection during Connectivity Exploration, Case 1: Disjoint Dominating Set Segment not Recognized or before First Hop

If visiting a non-dominating node $v \notin D$ and $v.iTab.cns \setminus a_{CE}.ck = \emptyset$, i.e. a disjoint dominating set segment is *not* recognized, or if $a_{CE}.p = \langle \rangle$, i.e. a_{CE} is before its first hop, the CEA a_{CE} proceeds as follows: it conducts a random walk similar to the IEA, but using the extended link rating *er* from Equation 4.3 (see Section 4.5.2.1) instead of link quality rating *qr* (see Section 4.5.1) during the *probabilistic* next-hop selection (see Equation 4.5 in Section 4.6.1.3). The extended link rating *er* integrates the link quality rating *qr* with the utilization rating *ur*, described in Equation 4.2 in Section 4.5.2. This way, the utilization of the next-hop links is taken into account, so that next-hop choices are spread more evenly—or in other words—there is a (tunable amount of) preference for new solution candidates.

Example Figure 4.8 (a) shows the result of the operations conducted in Figure 4.7 (c), continuing the example. In Figure 4.8 (b), node 13 created a CEA a_{13}, which now arrives at node 14, after visiting nodes 23 and 22, with the following path and center-related information: $a_{13}.p = \langle 13, 23, 22, 14 \rangle$ and $a_{13}.ck = \{11\}$, assuming that node 13 is aware of center 11, provided $CIPA_{mh} = 10$. Now a_{13} adds the information it carries to an empty interconnection table, $14.iTab$, so that: $14.iTab(13).cns = \{11\}$, $14.iTab(13).p = \langle 13, 23, 22, 14 \rangle$.

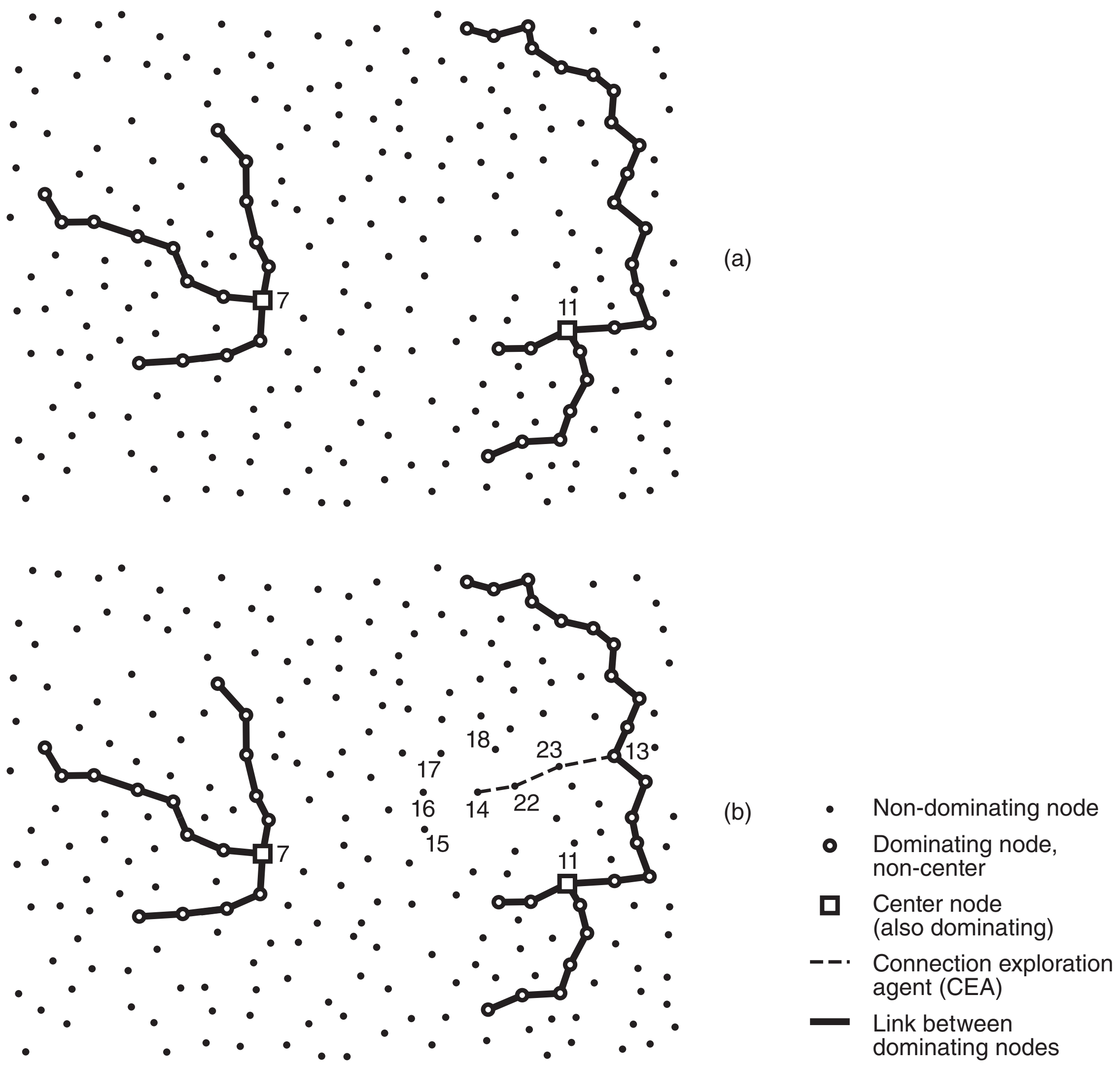

Figure 4.8: Exploration in behavior block II

In order to allow succeeding CEAs to balance their next-hop choices among the available candidates, the next-hop utilization table needs to be maintained by CEA a_{13}. For the purpose of illustration, assume first that the next-hop utilization table of node 14 is empty. If, in Figure 4.8 (b), CEA a_{13}, which is at node 14, selects node 18 as next hop, it sets $14.nhuTab(13,18).uf = 1$.

Consider a different situation: after the events above, further CEAs stop over at node 14, which results in $14.nhuTab(13,15).uf = 2$, $14.nhuTab(13,16).uf = 3$, $14.nhuTab(13,17).uf = 3$, and $14.nhuTab(13,18).uf = 2$. At this point, when a CEA at node 14 decides probabilistically for a next hop, it employs the extended rating *er*, which includes the utilization rating *ur*, described in Equation 4.2 in Section 4.5.2: in this case, the utilization rating $ur(13,16) = 1 - \frac{3}{2+3+3+2} = 0.7$ at node 14 for a CEA originating from node 13.

4.7.1.6 Next-Hop Selection during Connectivity Exploration, Case 2: Disjoint Dominating Set Segment Locally Recognized and not before First Hop

If a_{CE} arrives at a non-dominating node $v \notin D$ and $v.iTab.cns \setminus a_{CE}.ck \neq \emptyset$ (this way a disjoint dominating set segment is locally recognized), it selects a center from $v.iTab.cns \setminus a_{CE}.ck$ (i.e. one of the centers believed to belong to a disjoint segment) which is associated with the shortest path to some dominating node v_d. Ties are broken using node IDs. a_{CE} then chooses the next hop obtained from $v.iTab(v_d).p$.

Note that the above recognition is not reliable, since it is based only on the local knowledge of a CEA. However, this very part of the protocol does not have any negative influence on the protocol's behavior, as discussed further below, in Section 4.7.1.7.

This *deterministic* selection method is used in order to reduce the time and the communication cost of exploration. Notice that it is based on *iTab* entries that were modified by CEAs which were selecting their next hops probabilistically (see Sections 4.7.1.3 and 4.7.1.5). Therefore, the final parts of the solution, the connections, are obtained as a result of a *non*-deterministic behavior.

In addition to the above methods, after the second hop, a CEA exhibits the same *sticky behavior* as an IEA (see Section 4.6.1.3). It also handles failures similar to an IEA.

Example In Figure 4.9 (a), the example from Figure 4.8 continues. Assume that there were further CEAs originating from nodes 19, 20, and 21 that have visited node 14, adding the entries $14.iTab(19).cns, 14.iTab(20).cns = \{7\}$ and $14.iTab(21).cns = \{11\}$. The paths traveled by these agents until reaching node 14 are depicted using dotted lines and were also added to $14.iTab$. CEA a_{13}, currently at node 14, does not know center 7, as $14.iTab.cns \setminus a_{13}.ck = \{7,11\} \setminus \{11\} = \{7\} \neq \emptyset$. Since the path towards node 19 is the shortest known path to the segment considered disjoint, i.e. $14.iTab(19).p.length < 14.iTab(20).p.length$, CEA a_{13} will choose greedily 16 as next hop, as depicted in Figure 4.9 (b).

4.7.1.7 Implications of Local Recognition Mechanism

Although a CEA may locally recognize a dominating set part B as disjoint (see Section 4.7.1.6), B may nonetheless be connected to the dominating set part A, the agent originates from. This situation occurs, when the center information propagation ranges, corresponding to $CIPA_{mh}$, of the centers identifying A and B do not overlap at the point of evaluation ($CIPA_{mh}$ represents the maximum number of hops a center information propagation agent is allowed to walk, described in Section 4.6.2.3). However, given the design decision to only use local knowledge in order to generate the positive effects associated with self-organization, there is no means to provide a guarantee for disjointness from an agent's local point of view.

Example Figure 4.10 depicts a situation in which CEA a_{13} started at node 13 and now arrived at node 14. Since $CIPA_{mh} = 10$, a_{13} is only aware of center 11, i.e. $a_{13}.ck = \{11\}$. At node 14, a_{13} finds the interconnection information $14.iTab(19).cns = \{7\}$ because at node 19 (point of evaluation), from which an agent, depositing this information at node 14, originated, only center 7 is known. As $14.iTab.cns \setminus a_{13}.ck = \{7\} \setminus \{11\} \neq \emptyset$, CEA a_{13} assumes to have recognized a disjoint dominating set segment, although this is evidently not the case.

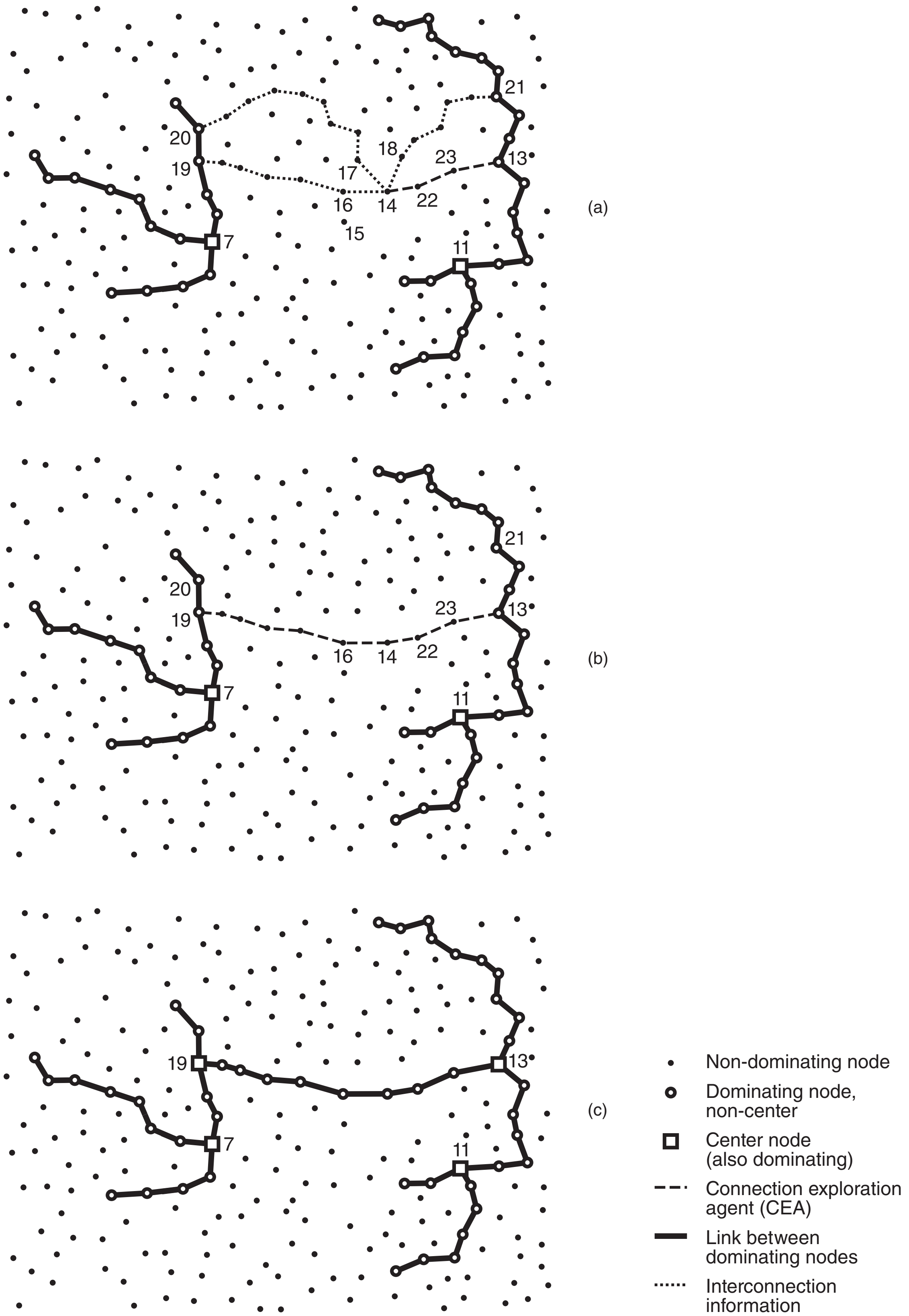

Figure 4.9: Exploration and construction in behavior block II

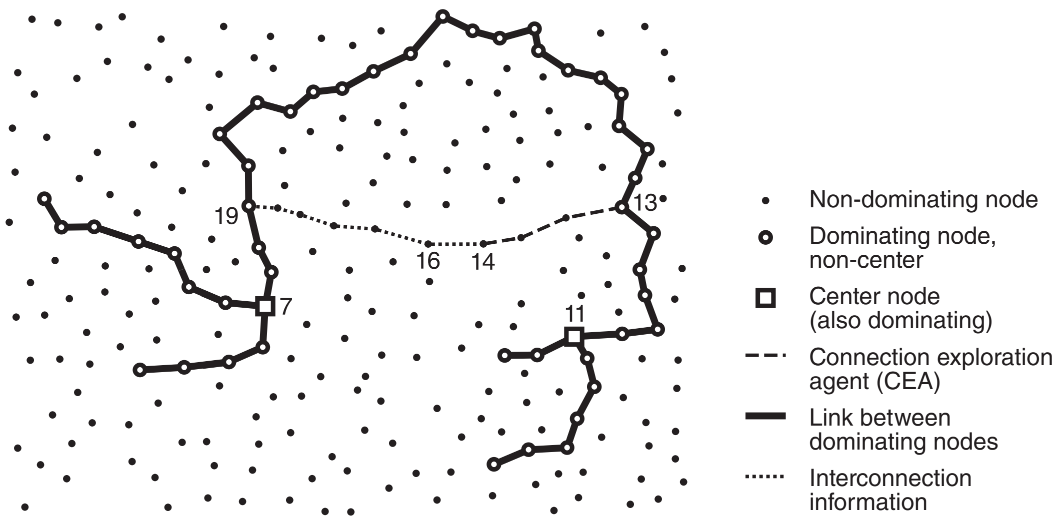

Figure 4.10: Disjoint dominating set part locally recognized

As it becomes clear from the example, the probability for the discussed situation to occur could be reduced by increasing $CIPA_{mh}$, which might consequently incur higher costs. Then again, such situations can be considered parts of the overall process that contribute to a construction of a CkDS with a particular dominating distance. Note that the connection in Figure 4.10 between nodes 13 and 19 would not result in an undesirable amount of redundancy of dominating nodes. On the contrary, in the situation depicted in Figure 4.11, a connection between nodes 13 and 19 can even be considered useful, for instance, to shorten path lengths, when a routing protocol is executed on top of the CkDS. At the same time, an interconnection that highly increases the amount of redundancy, such as in Figure 4.12, would be avoided by recognizing the existing connections. Consequently, the $CIPA_{mh}$ parameter can be employed to adjust the amount of redundancy incurred by interconnections of distant parts of the dominating set, as well as, the achieved dominating distance, which will be discussed further below, in Section 4.8.

4.7.2 Construction

If CEA a_{CE} arrives at a dominating node $v_d \in D$, it checks if $v_d.cdTab.v_c \setminus a_{CE}.ck \neq \emptyset$, that is whether v_d appears to belong to a disjoint dominating set segment. If this condition is satisfied, it generates a *connection construction agent* (CCA) a_{CC}, containing only the field p and dies, after setting $a_{CC}.p = a_{CE}.p$ ($a_{CC}.p$ assumes the value of $a_{CE}.p$). Else ($v_d.cdTab.v_c \setminus a_{CE}.ck = \emptyset$) a_{CE} dies without any actions.

After its generation and initialization, CCA a_{CC} follows $a_{CC}.p$, marking all visited nodes as dominating, thus creating a connection, and dies afterwards. It uses the same failure handling as an ICA (see Section 4.6.2.2).

Example Assume that CEA a_{13} from the above examples greedily followed the path from node 14 via node 16 towards node 19, as depicted in Figure 4.9 (b). Next, a_{13} checks whether the centers known by it, in this case center 11, are present in the center distance table of node 19, that is whether $19.cdTab.v_c \setminus a_{13}.ck \neq \emptyset$, in this example whether $\{7\} \setminus \{11\} \neq \emptyset$. This

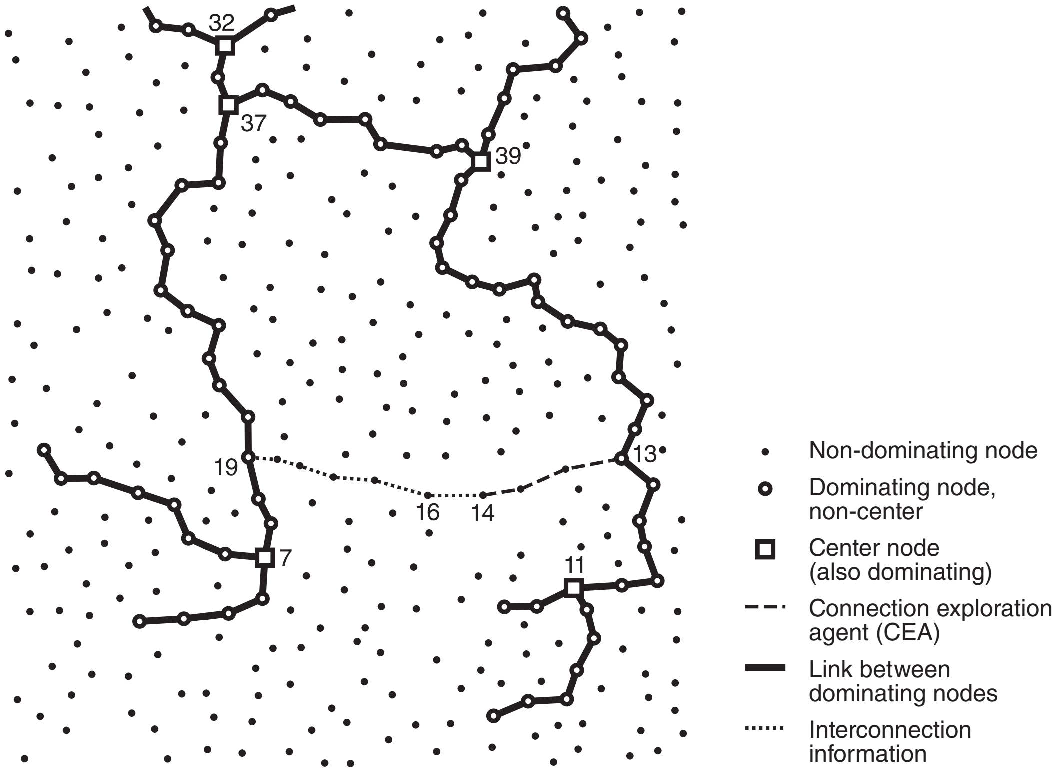

Figure 4.11: Disjoint dominating set part locally recognized in extended network

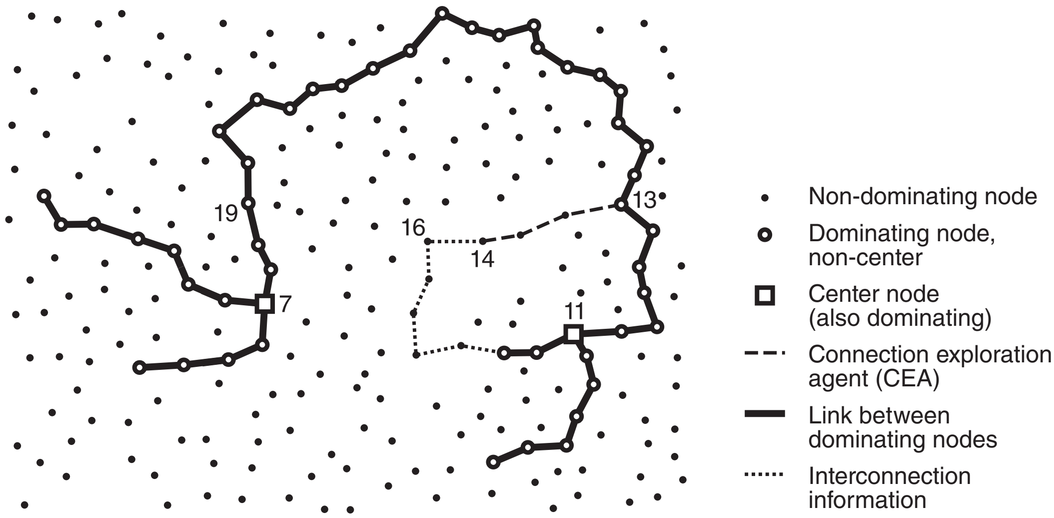

Figure 4.12: Disjoint dominating set part *not* recognized

condition is satisfied, so that node 19 creates a new CCA a_{19}, initializes it with a_{13}'s path, setting $a_{19}.p = a_{13}.p$, and lets a_{13} die thereafter. Finally, a_{19} follows $a_{19}.p$, marking all visited nodes as dominating, before it dies. This creates two new centers, 13 and 19, and results in the final state depicted in Figure 4.9 (c).

4.8 Discussion

The proposed CkDS construction process is self-organizing according to the definition by Camazine, Deneubourg, and colleagues from their seminal book on self-organization in biological systems [26]:

> Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.

This definition carries several implications:

1. Self-organization yields a global-level result. In the proposed process it is a CkDS.
2. The utilization of only lower-level interactions and local information translates to low requirements in terms of information exchange. Since the amount of information exchanged highly influences energy consumption (see Section 2.1.4), this is an especially desirable property in wireless sensor networks.
3. There is no reference between the execution based on local information at a lower level and the global-level result. This implies that there is no parameter directly corresponding to k in the proposed process.

In summary, self-organization promises to efficiently yield the desired result, however, there is no direct link between lower levels and the global level. Therefore, it is not possible to set a k at a lower level and obtain a CkDS with this exact k at the global level. Similarly, it cannot be guaranteed that the solution is connected or that two parts of the dominating set can be reliably recognized as disjoint. This missing reference between the lower levels and the global level is a property that the proposed protocol shares with many other related, self-organizing protocols. Ant colony routing [20] can well serve as an example: ants are confronted with the challenge of navigating through complex, unknown terrain or vast labyrinths within their formicaries. Their self-organizing protocol can be examined according to the above points:

1. Using self-organization, they produce a structure consisting of the shortest paths between their points of interest, such as food sources and their formicaries' food storage facilities.
2. Ants solely use lower-level interactions and local information by depositing and perceiving pheromone, as well as, other ants in their direct vicinity.

3. There is no reference between the rules the ants execute based on local information at a lower level and producing a structure consisting of the shortest paths between a network of points of interest.

However, naturally self-organization also entails that ants do not always find a structure consisting of the shortest paths between their points of interest. In fact, Goss and colleagues showed for the Argentine ant (Iridomyrmex humilis) [69] that, for example, given two alternatives, when the ratio of the long path's to the short path's length was 2, only in 11 out of 14 experiments more than 80 % of the total traffic used the short path. In general, they conclude that the colonies' probability of selecting the shortest path increases with the difference between two alternative paths.

Given the above considerations, it is evident that the proposed approach—since it is self-organizing—cannot guarantee connectivity or an exact dominating distance k. However, in simulations used for evaluation with up to 3000 nodes, the fraction of disconnected dominating sets was extremely close to 0. Moreover, as the results in Chapter 5 indicate, there is a strong correlation between the parameters IEA_{wsl}, IEA_{mdcbd}, $CIPA_{mh}$ and the desired dominating distance. Therefore, for example, in a real-world application of the CkDS, a desired k can be achieved in a straightforward manner by adjusting IEA_{wsl}, IEA_{mdcbd}, and $CIPA_{mh}$ appropriately, as discussed in Section 5.2.5.

Within this context, the nature of the definition of a CkDS should be reexamined. A CkDS with $k = u$ is always also a CkDS with $k \geq u$ according to Definition 1 on page 1. In other words, a CkDS with $k = 11$ is always also a CkDS with $k = 1000$. Given this imprecision, I introduce a precise distance definition (22 on page 101), which implies that for k':

1. There is no node v which is further away from the dominating node closest to v than k' hops.

2. There is a node v' which is k' hops distant from the dominating node closest to v'.

Therefore basically, after setting a certain k, the result produced by most related approaches is a Ck'DS with $k' \leq k$, when assuming perfect, or lossless, communication channels. The discrepancy between k and k' depends on the exact process of CkDS construction. For example, the connected clustering-based approaches [143, 144, 166] can be expected to be especially prone to this phenomenon, since they first produce a k-hop dominating set, which they connect subsequently. In contrast, this phenomenon is non-existent in the proposed approach.

In addition to the above issue, there is a second phenomenon to be considered, when non-perfect, or lossy, communication channels are used. This is especially the case in wireless sensor networks, in which the links between nodes can be expected to be highly unreliable (see Section 2.1.5). Therefore, in practice, the resulting k' of a CkDS may be even *greater* than the specified k in all related approaches [130, 143, 144, 165, 166], as shown in Section 5.2.5 for the reference approach [130]. This is due to, for example, election messages not arriving at the recipient after a transmission failure, which leads to a node not participating in an election—a violation of the protocol specifications. In contrast, the proposed protocol exhibited a high amount of predictability in terms of k' even in the unreliable wireless network employed for simulations (see Section 5.2.5).

In summary, the protocols from state of the art [130, 143, 144, 165, 166] allow the user to specify the desired dominating distance k in the CkDS to be constructed. However, it is evident that none of them can guarantee to produce a *connected* kDS with the desired k in the presence of an unreliable medium, which has been demonstrated for the reference approach [130] in the results chapter. The protocol proposed in this document does not allow the user to specify a desired k, given its self-organizing nature. Instead, it employs parameters which highly correlate with k and k'. They can be adjusted to reliably achieve the desired k/k' in a CkDS even in the presence of a highly unreliable medium, as shown in the next chapter and Section 5.2.5 in particular.

4.9 Summary

This chapter presented the proposed approach to CkDS construction, after discussing its inspiration and the considerations during its design, as well as, providing definitions needed in the subsequent sections.

During oviposition, fecund Pieris rapae females aim to balance the necessity of egg spreading, utilized to average out variations in larval survivorship, and energy spent for relocation. Therefore, they employ random walks to visit a subset of hosts for oviposition. When considering the area used for these activities, an adequate section of the flight trajectory of P. rapae represents an s-distant connected structure. The basic idea of the protocol, proposed in this chapter, is to imitate and adapt P. rapae's behavior to create CkDS in wireless sensor networks, while inheriting several of its desirable properties: distribution, parallelism, asynchronity, use of only local knowledge and lower-level interactions, randomization, redundancy, scalability, robustness, as well as, self-organization.

The proposed protocol is self-organizing, since a global-level pattern, the CkDS, emerges solely from numerous lower-level interactions, specified by rules executed using only local information, without reference to the global pattern. It consists of two intertwined behavior blocks, which are both essentially based on random walks: the first is responsible for the construction of a dominating set, while the second connects the existing segments of dominating nodes to a connected k-hop dominating set. The proposed approach is the first CkDS (and also CDS) construction protocol for wireless networks to employ random walks.

In order to avoid redundancies within this chapter, I refer the reader to Section 4.2, which provides a comprehensive summary of the proposed protocol.

5

Results

This chapter presents the results of extensive network simulations, which I used to evaluate the proposed self-organizing connected k-hop dominating set (CkDS) construction approach, introduced in the previous chapter, as well as, the results of a comparison to the state of the art. The first section of this chapter describes ShoX, the network simulator employed for the evaluation, which I co-initiated and which was also co-developed by me. Subsequently, the second section begins by providing an overview of the setup of the simulations. But more importantly, it further presents and discusses the results obtained in the simulation-based part of the evaluation of the proposed protocol and compares them to the findings for the state of the art.

5.1 Simulation Environment: ShoX

For the evaluation, I used the ShoX network simulator [92, 135], which I co-initiated and co-developed. There were various reasons for creating a novel simulator, such as that existing network simulators, like J-Sim [77], ns-2 [107], Opnet [110], and OMNeT++ [150] were initially designed to support wired networks. Their extensions for wireless networks followed only many years later, when this network class emerged, which had several implications: For example, as the architectures of the existing simulators were tailored to wired networks, some concepts inherent to wireless networks, not compatible with them, were integrated in an incoherent way, often in the form of makeshift solutions. Further, the generality of the set of functions in the existing simulators makes it hard to identify functions relevant for wireless networks, or, many times, some of these functions are missing entirely.

5.1.1 Main Concepts

In contrast to the existing simulators, the design of the ShoX architecture focused on adhering to the inherent properties of wireless networks. A second goal was to enable a straightforward integration of new protocols and models into the simulator. At the same time, a good usability was also important. The following concepts were used to achieve these desired goals:

1. ShoX is solely based on Java and XML. Java is used to implement protocols and models, while XML is employed for their configuration. In contrast to Tcl, utilized in J-Sim [77], OTcl (in ns-2 [107]), NED (in OMNeT++ [150]), Proto-C (in Opnet [110]), and Parsec [149] (in GloMoSim/Qualnet [118]), both of the languages used in ShoX are well known. Therefore, for most users, this eliminates the need to learn a new language. Furthermore, Java projects are easily expandable, enabling straightforward integration and interoperation of/among different simulated entities.

2. Using Java eliminates many issues associated with C/C++, which is employed in ns-2, OMNet++, and Opnet, such as segmentation faults, memory allocation, and deallocation. At the same time, XML allows the utilization of a myriad of standard tools for parsing and transformation.

3. Everything is a class. All layers, including protocols, packets, as well as, mobility, signal propagation, and traffic models are defined as classes. A new protocol or model is added by subclassing and the implementation of several methods. There is a broad range of ready-to-use superclasses for the simulation of various aspects and entities inherent to wireless networks, such as routing and MAC protocols, as well as, physical layer, mobility, or interaction models.

4. ShoX has a strong GUI support for the different facets of simulations, ranging from scenario configuration, over network visualization, to statics processing. Therefore, it is not necessary to edit XML configuration files manually. At the same time, the GUI ensures the validity of configuration input.

5. ShoX comes in a single package. It is not necessary to manually configure different packages or the environment in a special way.

5.1.2 Architecture

Similar to ns-2 or OMNeT, ShoX is a discrete-event simulator: it uses a global event queue, into which all events, such as packets, internal messages, timers, and node movement events, are inserted. Each event contains an unique ID, an addressee, and delivery time, at which it is to be removed from the queue and delivered to the addressee. The global event queue is sorted by delivery time, so that the next event to be processed can be retrieved at low cost.

The central entity in ShoX is the *simulation manager* acting as scheduler, which manages the event queue and which is responsible for dequeuing and delivering the first element of the queue. At the time of delivery, the current time is set to the delivery time of the delivered event, so that parallelism can be simulated.

To illustrate the operation of ShoX, consider the example in Figure 5.1. Among other parts of the simulation, it shows the global event queue with three events ordered by delivery time. Besides delivery time, these events include information about the type and the addressee, for instance, WakeUpCall for the LLC layer at node 78 (second event in the queue). Node 261 just enqueued a packet for the MAC layer of its own network stack, providing an example for intra-node communication. At the current time, the scheduler dequeues the next-earliest event, which is a network-layer packet, sets the current simulation time to the packet's delivery time and finally hands it over to the network layer of node 286.

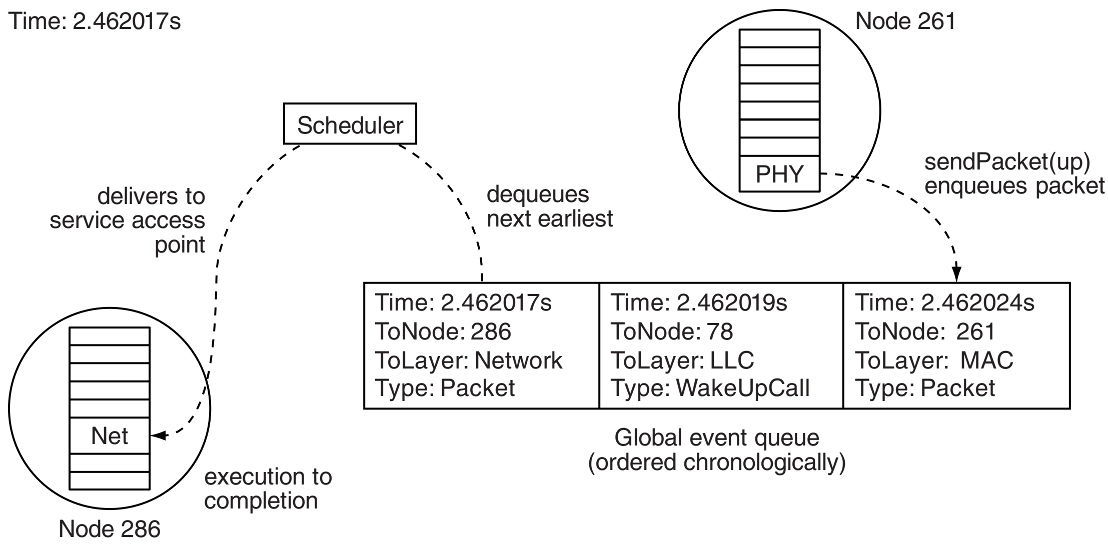

Figure 5.1: Discrete event simulation in ShoX—an example

ShoX is fully object oriented. All concepts and models, like OSI layers, packets, failure, and traffic models, are represented by abstract classes providing the required interfaces. This makes the definition of own concepts and models very straightforward. The architecture is also safe, since ShoX does not provide access to other layers or models directly. Instead, the given and the individually defined components of ShoX rely on message exchange via the global event queue to communicate with each other.

The packet class is a subclass of the event class. Packets, defined by a protocol at a certain layer, such as AODV at the network layer, are again subclasses of the general packet class. ShoX conforms to the OSI layer model. Nevertheless, it is also possible to introduce new layers at any point in the layer stack. Below the physical layer, there is the AirModule. It manages the state of a node's radio, i.e. transmitting, receiving, idle listening, sleeping, power off, and also, for instance, monitors any potential interferences. Upon the arrival of a packet, the AirModule notifies the interference handler about occurring interferences, which subsequently decides about the handling of the packet: for example, the packet can be ignored, errors can be inserted into it, or it can be discarded.

As a further example, signal propagation effects are modeled using the abstract class PhysicalModel. Based on the knowledge of node positions and signal strength, an instance of a subclass of PhysicalModel computes whether two nodes can reach each other or whether there is any interference. If the receiver is in at least interference distance, a concrete PhysicalModel further returns the strength of received signals. Using this approach, a broad spectrum of signal propagation effects can be simulated.

Finally, for instance, it is not only very straightforward to implement communication traffic models but also to record traffic. Independent of the chosen model, ShoX automatically generates traffic traces, which can be later used to conduct new or repeated simulations. The simulation based on traffic traces can save computation overhead, when complex models are used.

To provide a more comprehensive overview of ShoX, it should be added that ShoX includes diverse tools to support the protocol developer by, for example, improving the efficiency of the implementation and debugging process. These tools enable him or her, among other functions, to visualize the network scenario or to create and process statistics. However, given the focus of this document, they are not described here in detail. The interested reader is referred to [79, 80, 92].

5.1.3 Configuration

Before executing a simulation, the scenario to be simulated must be specified. This includes information, such as the number of nodes, the size of the deployment area, the layer stack of the nodes, and, for instance, the signal propagation and traffic models which are to be used. In ShoX, this is done using a GUI. The deployment area is specified, by providing width and height explicitly or by loading a Scalable Vector Graphics (SVG) file which contains a map of the area. The remaining fields in the configuration panel are to input the Java classes that implement the required models and layers. For every field, there are default (dummy) classes available, which are used automatically in case the user leaves a field empty.

Most user-implemented layers and models require some kind of initialization to work correctly. Specifically, they have certain class variables which need to be set by the user. In ShoX, this is realized in a very straightforward way. Any variable within a class that needs external initialization by the user, before the simulation is started, is annotated, for example, as follows:

```
@ShoXParameter(description = ”The walk segment length of initial exploration
agents (IEA_WSL)”, required = true)
private int iEAWalkSegmentLength;
```

An annotation automatically exports the associated variable to the GUI. The variable itself can be used like any other variable within the class. Variables are not confined to primitive data types: objects of any kind can also be exported. Therefore, the GUI provides recursive dialogs for the initialization of complex objects.

5.2 Evaluation

The simulation results which served as basis for the evaluation, presented in this section, were obtained using the ShoX network simulator [92, 135], described in the previous section. Both, the reference and the proposed approach, were simulated in topologies of different sizes and node densities (see Section 5.2.2 for the reference approach).

5.2.1 Outline

In order to produce a comprehensive evaluation, I considered the following aspects:

Efficiency The efficiency of both, the reference and the proposed protocol was evaluated in terms of total communication cost in MBit, based on the number of sent and received data, as a function of effective maximum dominating distance k' (see Definition 22 on page 101), average node degree d, and number of nodes n. See Section 5.2.4 for further details.

Offset Reflects the degree of discrepancy between the adjusted parameters, influencing the effective maximum dominating distance k', and the produced results. The offset and its standard deviation are evaluated for different k' in Section 5.2.5.

Solution Size The larger the CkDS, the fewer nodes can switch their radio to powerdown mode or to lower duty cycles (see Sections 2.1.4 and 2.2.2). On the other hand, larger CkDS can be expected to offer more redundancy and thereby more failure resilience. This aspect is examined for different effective maximum dominating distances k', average node degrees d, as well as, for different numbers of nodes n in Section 5.2.6.

Dominating Degree Denotes the number of dominating 1-hop neighbors of dominating nodes, and its average has implications for the shape of the CkDS. Besides the average dominating degree, Section 5.2.7 also examines its standard deviation within each run, which allows conclusions about the regularity of the produced CkDS structure, as a function of effective maximum dominating distance k'.

Construction Time When constructing a CkDS, not only the efficiency and quality of the solution are of importance but also the construction time. In Section 5.2.8, the construction time for different effective maximum dominating distances k' and network sizes n is examined. Moreover, the subsection also sheds light on the development of the construction process.

It should be remarked that the evaluation of all of the above aspects alludes to the topic of scalability, as well. For instance, Section 5.2.4 also examines how the communication cost incurred scales with different node numbers n or average node degrees d. Similarly, Sections 5.2.6 and 5.2.8 evaluate how the CkDS sizes scale with different effective maximum dominating distances k' and construction time.

The next sections describe the simulation setup and parameters used. Subsequently, Sections 5.2.4–5.2.8 present the simulation-based part of the evaluation, focusing on the proposed and the reference approach. Finally, the evaluation is concluded by a comparison of the proposed approach to all known state-of-the-art approaches in Section 5.2.9.

5.2.2 Setup

To obtain the results, the ShoX wireless network simulator [92, 135] was used with the default wireless medium implementation, simulating packet loss and collisions, as well as, the implemented IEEE 802.11g MAC.

At the beginning of the implementations, from the relevant state-of-the-art protocols up to now known to me (that are [130, 143, 165, 166]), only [130, 143], and [166] were published. I selected the work from Sausen et al. [130] as the *reference approach* (see Section 3.2.2 for an overview), since it was the most-recently proposed, and as it shares an important characteristic with the other two (i.e. [143, 166]): its cost strongly depends on k and the average node degree d.

5.2.3 Parameters

The simulations were conducted utilizing rectangular topologies with aspect ratio 1:1.5 with

$$n = 500, 1000, 1500, 2000, 2500, 3000 \tag{5.1}$$

randomly placed nodes with average node degree

$$d = 7, 9, 11, 13, 15, 17 \tag{5.2}$$

For each chart, a subset of all possible combinations was used. For instance, for the chart in Figure 5.2, the following combinations were needed: $(n = 2000, d = 9)$, $(n = 2000, d = 15)$, and $(n = 3000, d = 9)$.

The differently-sized topologies with different average node degrees were employed, in order to evaluate the scalability of the approaches. At the same time, in the light of scenarios such as ubiquitous computing (see Section 2.1.3), the networks had to be sufficiently large and dense in order to examine, how well the protocols also cope in these kinds of scenarios.

All charts in this section present the averaged results obtained from a series of 50 runs for each combination of approach, average node degree d, number of nodes n, and parameter setting identified using par, which is defined differently for the reference and the proposed approach, as described next.

For the reference approach

$$par = r = 2, 3, \ldots, 9 \tag{5.3}$$

r is used in the reference approach to set the desired k in the CkDS which is constructed.

For the proposed approach

$$par = IEA_{wsl} = 2, 3, \ldots, 37 \tag{5.4}$$

further $CIPA_{mh} = \lfloor 0.5 \cdot IEA_{wsl} \rfloor$, and $IEA_{mdcbd} = \lfloor 0.7 \cdot IEA_{wsl} \rfloor$, with

$\mathbf{IEA_{wsl}}$ the walk segment length of an initial exploration agent (IEA), introduced in Section 4.6.2.1.

$\mathbf{IEA_{mdcbd}}$ the minimum dominating contact build distance of an initial exploration agent (IEA), introduced in Section 4.6.2.1.

$\mathbf{CIPA_{mh}}$ the maximum number of hops a center information propagation agent (CIPA) may travel. It was introduced in Section 4.6.2.3.

IEA_{wsl}, IEA_{mdcbd}, and $CIPA_{mh}$ are tied together, as described above, since after initial experiments, this constellation appeared to correlate best with the effective maximum dominating distance k' (see Definition 22 on page 101).

The time needed to run simulations turned out to be a limiting factor, since a *single* simulation of the reference approach at $r = 9$, $d = 9$, and $n = 3000$, for instance, required approximately half a day of time on a single machine. Therefore, simulations needed to be distributed over a pool of computers.

5.2.3.1 Significance

For the data points in Figures 5.2, 5.3, and 5.5–5.8, representing averaged values resulting from the simulations, the following confidence intervals were computed, confirming the statistical significance:

- ◇ 95 % confidence intervals (level of significance $\alpha = 0.05$) for the *proposed approach*. Let $\Gamma = \frac{y-l}{y} = \frac{u-y}{y}$, with y, the value represented by a data point, and further, l and u, the lower and upper endpoints of the corresponding confidence interval. Then, for 87 % of the data points, $\Gamma \leq 0.025$, for 11 % of the data points, $\Gamma \in (0.025, 0.05]$, and, for 2 % of the data points, $\Gamma \in (0.05, 0.075]$ (there were exactly 100 data points for this approach). In other words, e.g. for 87 % of the values y represented by data points, the endpoints of their corresponding 95 % confidence intervals are within ± 2.5 % of y.

  For example, the lower and upper endpoints of the 95 % confidence interval of the data point of the proposed approach in Figure 5.2 with $d = 15, n = 2000, k' = 4$ would be at $y = 27.496$ MBit ± 0.530 %, that is $l = 27.351$ and $u = 27.642$ MBit.

- ◇ 90 % confidence intervals (level of significance $\alpha = 0.1$) for the *reference approach*, with $\Gamma \leq 0.05$ for 69.32 % of the data points, $\Gamma \in (0.05, 0.1]$ for 23.86 % of the data points, $\Gamma \in (0.1, 0.15]$ for 5.68 % of the data points, and $\Gamma > 0.15$ for 1.14 % of the data points.

The significance difference in the results is mainly related to the cost and offset properties of the considered approaches (see Sections 5.2.4 and 5.2.5). As discussed in Sections 3.2.2.1 and 5.2.4, the cost of the *reference* approach grows approximately quadratically with the adjusted dominating distance r. However, as evident in Section 5.2.5, a particular adjusted r may lead to different effective maximum dominating distances k' (see Definition 22). Therefore, the cost incurred by the construction of a solution with a specific k' may vary extremely, which is the central cause for the differences in the significance. In contrast, the *proposed* approach yields highly similar costs—independent of the adjusted parameters and the produced effective maximum dominating distances k', which translates to better significance values—and implies a higher amount of reliability.

5.2.3.2 Effective Maximum Dominating Distance

In order to enable a more accurate evaluation of the results of both protocols, the *effective maximum dominating distance* k' needs to be introduced:

Definition 22 $k' = \max_{v_i \in V}(length(\zeta(v_i)))$*, with* $\zeta(v_a)$*, the shortest path between node* v_a *and the node* $v_d \in D$ *closest to* v_a*. Further,* $length(\zeta(v_i))$ *returns the length of* $\zeta(v_i)$*.*

This implies that for k':

1. There is no node v which is further away from the dominating node closest to v than k' hops.
2. There is a node v' which is k' hops distant from the dominating node closest to v'.

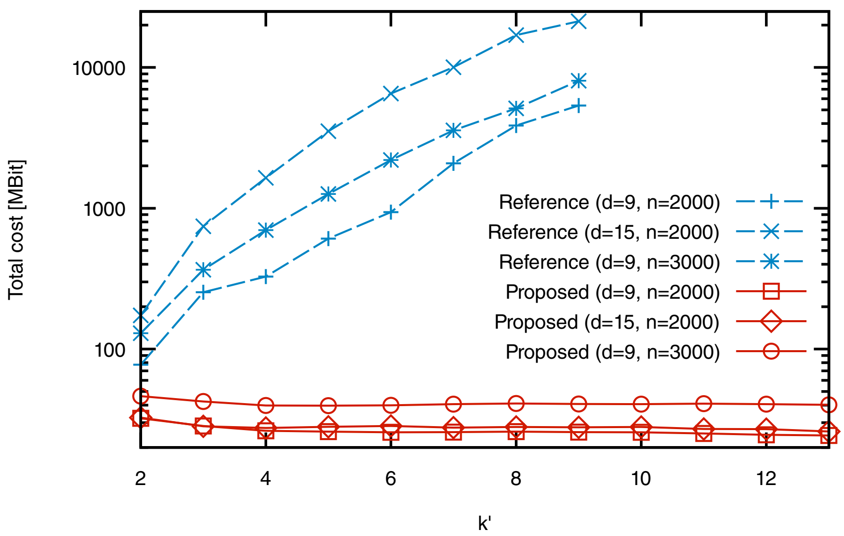

Figure 5.2: Cost with varying k'

The effective maximum dominating distance k' is used to classify and compare the solutions produced, as it is more precise than k: A CkDS with $k = q$ is also a CkDS with $k > q$, but a CkDS with $k' = u$ is *not* also a CkDS with $k' > u$. In other words, a CkDS with $k = 11$ is always also a CkDS with $k = 1000$, as illustrated in Figure 4.3. The classification is necessary to be able to compare the same classes of solutions, since both approaches use different input parameters which moreover do not yield deterministic results, as discussed in Section 5.2.5.

For the sake of brevity, this document will refer to the effective maximum dominating distance only as k' from this point on.

5.2.4 Efficiency

To evaluate the efficiency, the total cost in MBit (sent and received) of each of the approaches for a different k' for $d = 9, 15$ and $n = 2000, 3000$ is compared in Figure 5.2.

It is evident that, while the cost incurred by the reference approach grows quadratically with increasing k', the cost of the proposed approach is practically uninfluenced by k' (for $k' > 3$). This behavior can be explained with the strong dependency of the reference approach on k-hop flooding: As the area of a circle grows quadratically with radius r ($\pi \cdot r^2$), the area covered by the flooding will grow in a similar manner with growing k. Since there is a certain average number of nodes per unit of area, the number of nodes involved in the flooding grows linearly with the area. This is illustrated in Figure 5.4, where a six-hop flooding is depicted, for which the node colors represent different hops. The ideal communication radii, which provide the basis for this example, are depicted in gray. Naturally, in a bounded area, the growth will slow down with the number of nodes involved in k-hop flooding approaching n, which is evident in Figure 5.2. Imagine the flooding in Figure 5.4 to reach more than seven hops: evidently, the number of nodes newly involved in flooding at each hop would decrease with the number of hops, as the border of the flooding approaches the borders of the network area.

Notice that this evaluation can only present data for the reference approach up to $k' = 9$, since with increasing communication cost, the time needed to run a single simulation reached approximately half a day.

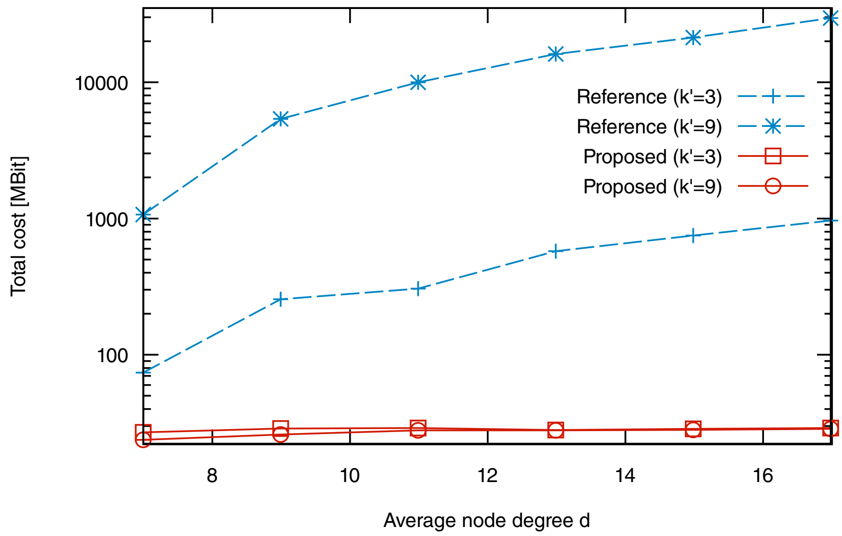

Figure 5.3: Cost with varying degree d, $n = 2000$

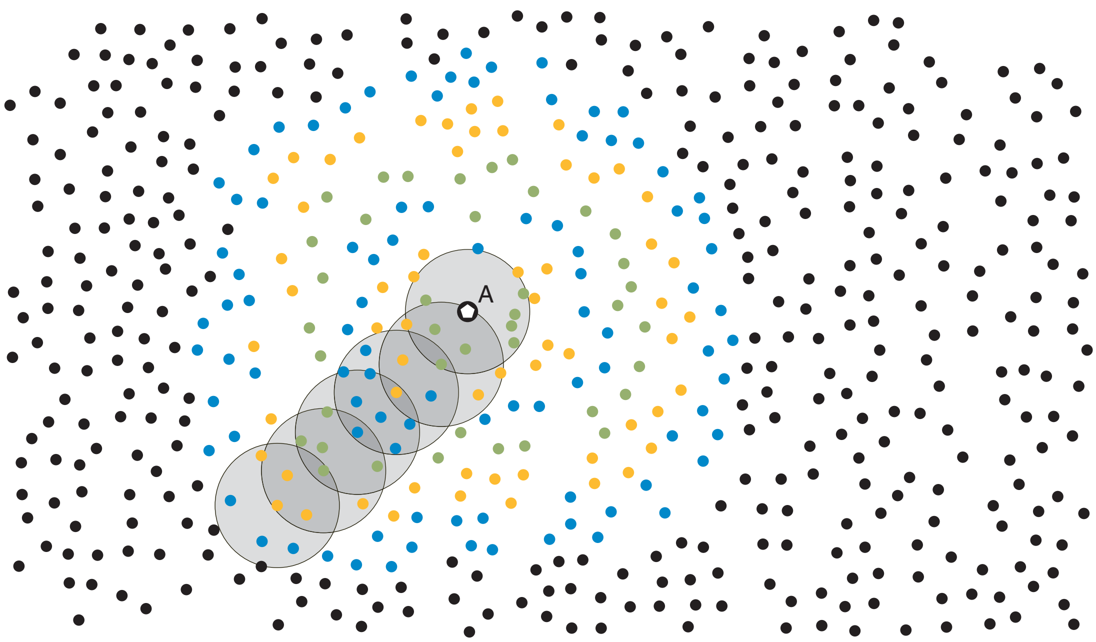

Figure 5.4: A six-hop flooding by node A

In Figure 5.3, the cost with varying d and $n = 2000$ is plotted. The lowest value for d is 7, since it was not able to obtain a randomly generated topology for $d = 5$, as in networks of this size, with decreasing *average* node degree d, the probability for a connected topology drops rapidly. On the other end of the range, 17 is the maximum value that is used for d, since for $d = 19$, the run time of single simulations already reached an unacceptably high level.

Looking at the data in Figure 5.3, the cost of the reference approach grows linearly with d, as each broadcast within the flooding is received by approximately $d - 1$ nodes on average. The proposed approach exhibits also a linear cost increase, although it is minimal compared to the reference approach. This is due to the fact that the number of broadcasts employed by the proposed approach is extremely low, since it equals the number of dominating nodes.

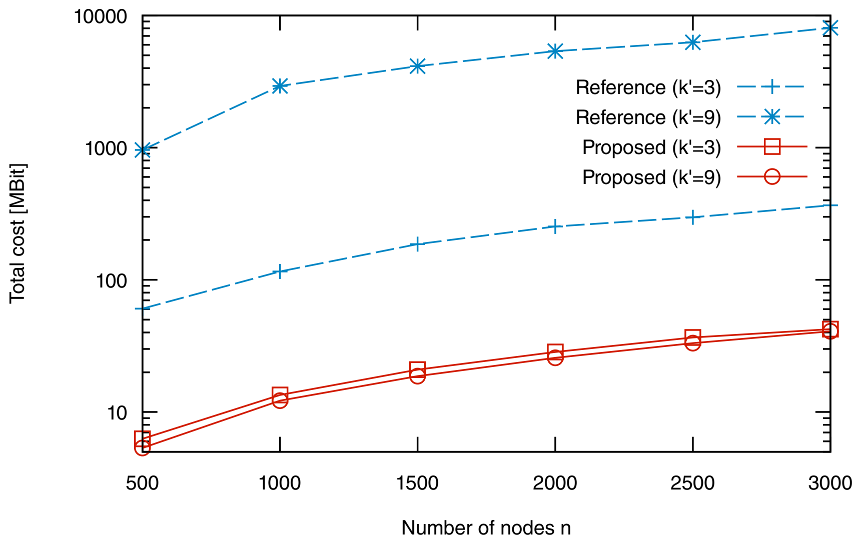

Figure 5.5: Cost with varying node number n, $d = 9$

Varying n in Figure 5.5 with $d = 9$, it can be observed that the cost for both approaches grows approximately in a linear manner with node number. Moreover, as indicated in the figure, the proposed approach with $k' = 3$ is slightly more expensive than with $k' = 9$, since for $k' = 3$ the number of dominating nodes and therefore the number of broadcasts is significantly higher.

5.2.5 Offset

The *offset* is defined as

$$o = \frac{k'}{par} \tag{5.5}$$

with $par = r$ in the reference, and $par = IEA_{wsl}$ in the proposed approach. It measures to what extent k' differs from a parameter that is employed to adjust k'. As depicted in Figure 5.6 using $d = 9$ and $n = 2000$, both approaches tend to achieve offsets, which are $\neq 1$. I assume the offset in the reference approach to be mainly determined by packet loss, when, for example, election messages are not delivered, and therefore nodes do not participate in the election. In contrast, the offset exhibited by the proposed approach is part of its nature, as there is no direct link between IEA_{wsl}, or any other parameter, and k', i.e. there is no parameter that corresponds to k'.

When comparing the offsets in Figure 5.6, the offset of the proposed approach is approximately equal for all k', while the offset for the reference approach differs considerably for different k'. The standard deviation of offsets achieved in a series of runs for the proposed approach is equally independent of k' and significantly lower than for the reference approach, making the proposed approach more predictable. If a specific minimum or maximum k' needs to be achieved, both r and IEA_{wsl} have to be selected in a conservative manner. This means, for example, that in order to achieve a minimum $k' = 5$ with a high probability, one needs to set $r = 8$ and $IEA_{wsl} = 18$ (and $CIPA_{mh}, IEA_{mdcbd}$ as in Section 5.2.3) in the reference and the proposed approach.

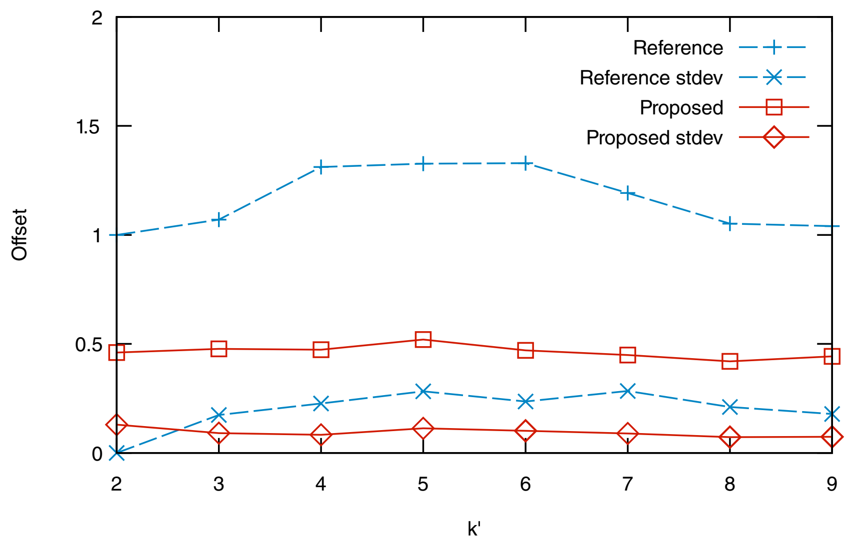

Figure 5.6: Offset, $d = 9$, $n = 2000$

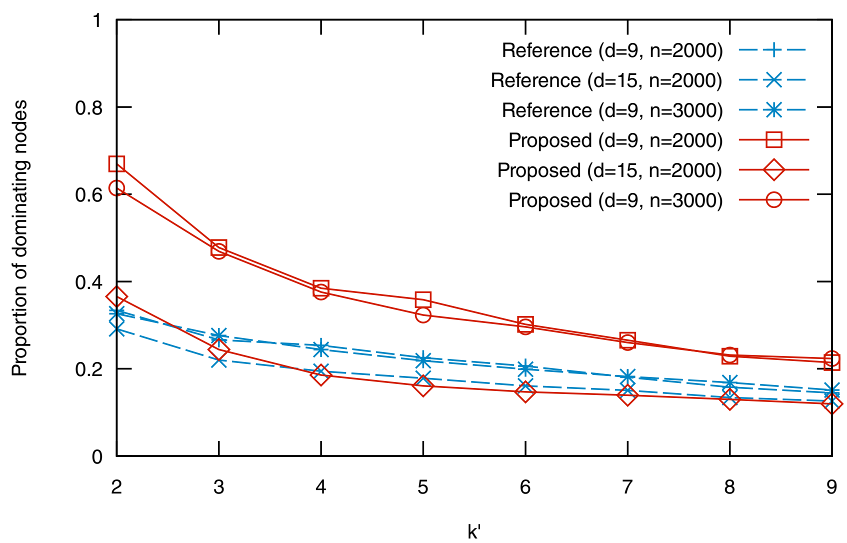

Figure 5.7: Solution size

5.2.6 Solution Size

The *size* of the solutions produced by both approaches is depicted in Figure 5.7. For $d = 9$, the solution provided by the proposed approach is considerably larger, when k' is very low. With growing k', however, the CkDS size decreases reaching levels only slightly higher than for the reference approach. For $d = 15$ and $k' \geq 4$, the solution provided by the proposed approach is always smaller than any of the solutions produced by the reference approach. As discussed in Section 5.2.1, depending on the application of a CkDS and the considered scenario, a smaller- or larger-sized solution is preferred.

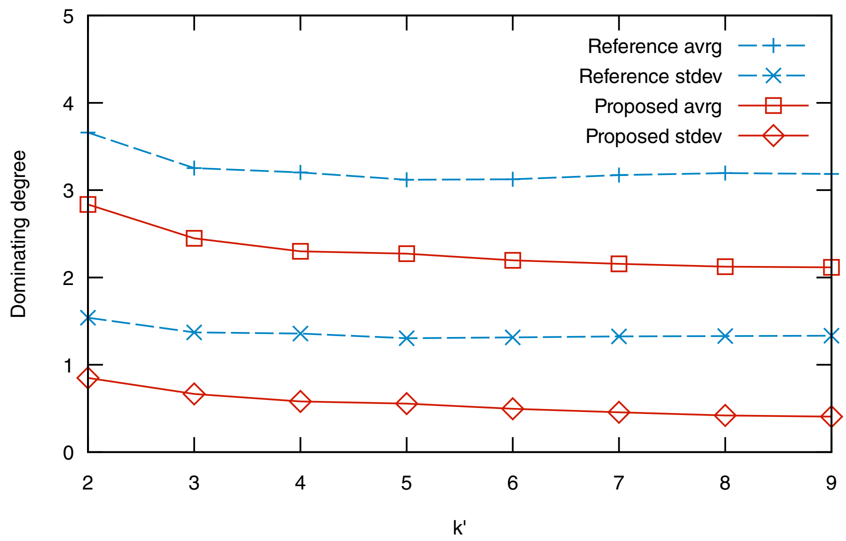

Figure 5.8: Dominating degree, $d = 9$, $n = 2000$

5.2.7 Dominating Degree

Next, the *dominating degree* is examined, which is defined as $|N_1^D(v)|$ for $v \in V$ ($N_1^D(v)$ consists of all dominating 1-hop neighbors of node v, see Definition 19 on page 67). In some applications, for instance, when the CkDS is used as a backbone for routing, it is not favorable to have dominating nodes $v_d \in D$ with large $|N_1^D(v_d)|$, since they can be expected to be overburdened much faster than dominating nodes $v'_d \in D$ with lower $|N_1^D(v'_d)|$. At the same time, for these applications, the degree should be evenly distributed, so that dominating nodes share the burden of a certain function more equally. Therefore, I use the *averages* and *standard deviations* of dominating degrees of *dominating* nodes within each of the runs as metrics. When looking at Figure 5.8 with $d = 9$ and $n = 2000$, it is evident that the results for both metrics are considerably lower for the proposed compared to the reference approach.

5.2.8 Construction Time

The *construction time* needed by both approaches to obtain a full (that is complete) and half of the solution, measured in number of nodes, is depicted in Figure 5.9 as a function of k'. It is defined as the length of the time span between the start of the protocol and the last modification of the dominating set. For the sake of clarity, all time values in Figures 5.9, 5.10, and 5.11 are normalized to the longest construction time (which is the reference approach, $k' = 9$, $d = 9$, $n = 3000$) encountered.

It is evident from Figure 5.9 that, while the construction time of the reference approach grows linearly with the number of nodes in the network, n, the construction time of the proposed approach is practically uninfluenced by it. This can be explained by the fact that the proposed approach works in an entirely parallel manner. In contrast, the reference approach grows one piece of the dominating set consecutively from the base station towards the borders of the network. During this process, nodes are added concurrently at the edge of this piece, since each node performs the election independently of others.

Further, it can be observed in Figure 5.9 that, while the construction time of the proposed

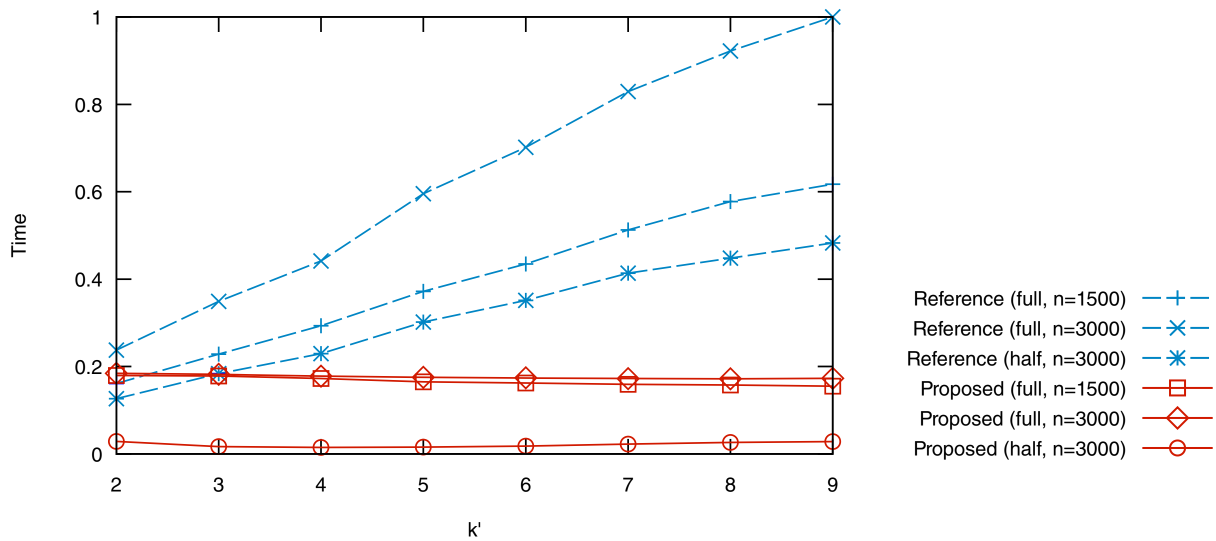

Figure 5.9: Construction time, $d = 9$

approach is practically uninfluenced by k', the construction time of the reference approach grows linearly with it. This is due to the fact that each node needs to exchange information within its r-hop neighborhood in the reference approach, and r strongly correlates with k' (as depicted in Figure 5.6).

Finally, the data in Figure 5.9 indicates that a major portion of the solution, half of it, is provided much earlier by the proposed than by the reference approach. To understand the *construction progress* over time, the proportion of the full solution that was completed, measured in number of nodes, is plotted as a function of time in Figures 5.10 and 5.11 for $k' = 3$ and $k' = 9$. In the figures, it can be observed that at the beginning of the construction, the pace of the proposed approach is significantly higher compared to the reference approach, before it gradually slows down towards the end. This is a result of the self-regulation exhibited by the proposed protocol: at the beginning, much area is uncovered, so explored paths that are candidates for the dominating set satisfy the given conditions (see Sections 4.6.2.1 and 4.7.2) very frequently; towards the end, as only little additional coverage is needed and can be achieved, fewer and fewer explored candidates satisfy these conditions.

In contrast, the construction progress produced by the reference approach is represented and can be described by much flatter, s-shaped curves. This is due to the manner the reference approach constructs the dominating set: At the beginning, only few nodes can be added per unit of time, as the edge of the dominating set, growing around the base station, is very short. During the course of the construction it grows larger, and so the number of nodes that can be added per unit of time follows this growth. When the edge of the dominating set reaches the borders of the network, the number of nodes that can be added starts to decline accordingly.

5.2.9 Proposed Approach versus State of the Art

Up to this point, the evaluation concentrated on the properties of the proposed protocol and a comparison to the corresponding properties of the reference approach. However, the question arises as to how the proposed approach compares to the other state-of-the-art approaches. To answer this question, all comparable results that were obtained in this thesis, concerning the

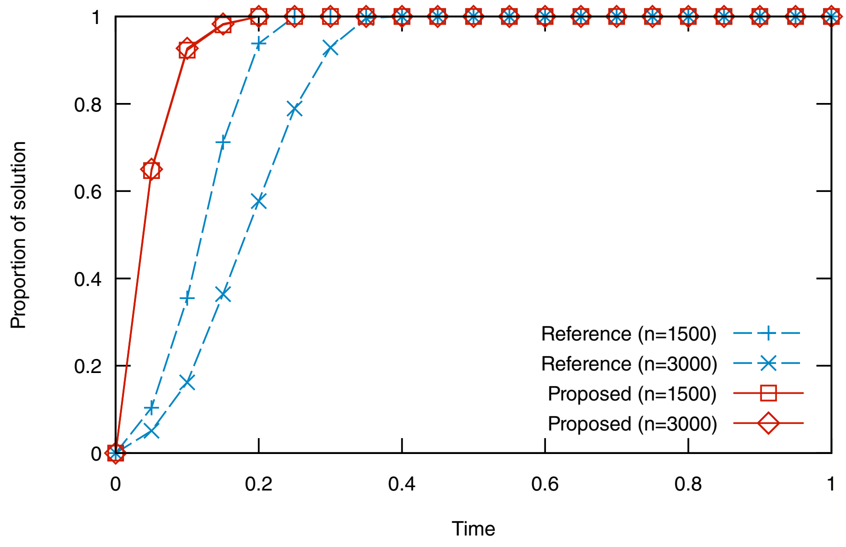

Figure 5.10: Construction progress, $k' = 3$, $d = 9$

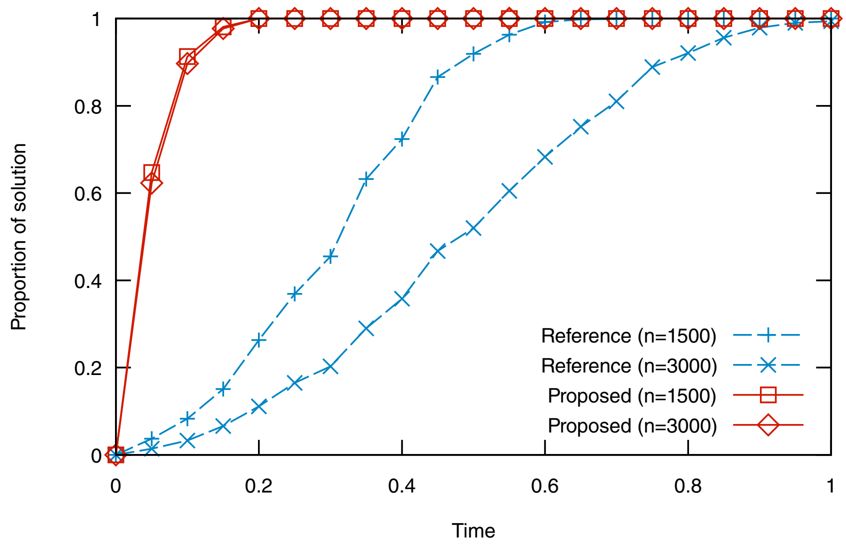

Figure 5.11: Construction progress, $k' = 9$, $d = 9$

state-of-the-art and the proposed approaches, are summarized in Table 5.1. The table shows the dependencies of the communication cost incurred and construction time needed by the considered protocols on different parameters:

k maximum dominating distance

k′ effective maximum dominating distance (see Definition 22 on page 101)

d average node degree

n number of nodes

Findings from the concise analyses conducted in Section 3.2 and this section are marked by ^{a}, while all results yielded by means of simulation are labeled with ^{s}. Note that the table

Parameter	Communication cost k,k'	d	Construction time k,k'	n
Theoleyre and Valois [143, 144]	quadratic^{a}	**linear**^{a}	linear^{a}	linear^{a}
Yang, Wu, and Cao [166]	quadratic^{a}	**linear**^{a}	linear^{a}	linear^{a}
Sausen et al. [130]	quadratic^{as}	**linear**^{as}	linear^{as}	linear^{as}
Yang, Lin, and Tsai [165]	linear^{a}	quadratic^{a}	linear^{a}	**independent**^{a}
Proposed	**independent**^{s}	**linear**^{as}	**independent**^{s}	**independent**^{as}

Table 5.1: Comparison of proposed approach with state of the art

omits the dependencies of communication cost on parameter n, as well as, construction time on parameter d, since, except for one case, for all of the approaches, they can be clearly considered as linear and independent. The exception is the approach by Yang, Lin, and Tsai [165], whose construction time depends linearly on d, since during the information exchange in step 1 of a round, the information exchange request cannot be answered by all neighboring nodes simultaneously but only sequentially.

It should be remarked that the table includes both, k and k', since the authors of the state-of-the-art approaches employ k in the description of their protocols in order to adjust the desired k'. From the definitions of the protocols and also from the simulation results (see Section 5.2.5), it is evident that the offset (see Equation 5.5 in Section 5.2.5) can be expected not to equal 1 in most cases. This is due to, for instance, packet loss, as observed in the approach by Sausen and colleagues [130], or the change of the dominating distance after the connection of disjoint dominating set parts, like in the approach by Yang, Wu, and Cao [166]. However, a high amount of correlation between the adjusted k and the resulting k' can be assumed, based on the definitions of the state-of-the-art protocols, which is also confirmed by the simulation results in Figure 5.6 for the approach by Sausen and colleagues [130].

First of all, when looking at the results in Table 5.1, it is evident that the simulations confirm the analysis findings for the approach by Sausen and colleagues [130].

Comparing the results in Table 5.1 yields that the communication cost incurred and construction time needed by the proposed approach are independent of k and k'. In contrast, the cost and construction time of the state-of-the-art approaches depend linearly or quadratically on these parameters. This behavior of the proposed approach and one of the state-of-the-art approaches from Sausen and colleagues [130] can also be observed in the simulation results from Figures 5.2 and 5.9.

Based on reasoning, the communication cost incurred by all approaches except [165] evidently depends linearly on the average node degree d, since it is dominated by the use of broadcasts, which reach all of a transmitting node's neighbors in the ideal case, in terms of d. However, when comparing the quantities of the broadcasts used, there are great differences.

In the first three approaches from the table [130, 143, 144, 166], each node initiates at least one k-hop flooding, which itself consists of numerous broadcasts and for which the number of nodes involved grows quadratically with k. In contrast, the proposed approach utilizes by far fewer broadcasts: each node that becomes dominating transmits exactly one broadcast, which does not include any information except the indication of the state change and the ID of the sending node. That is the only point in the protocol where broadcasts are used.

The difference in the quantities of the broadcasts is evident, for example, from Figure 5.3 for the approach by Sausen et al. [130] and the proposed approach. While the cost grows extremely with average node degree d for the reference approach, the cost incurred by the proposed approach increases only slightly.

Finally, when considering the construction time, it is evident that the proposed approach is completely independent of the number of nodes n, since all its actions are executed in parallel. This property, shared with only one further protocol from state of the art (i.e. [165]), can also be observed in the simulation results from Figure 5.9.

5.3 Summary

This chapter presented the results from the evaluation of the proposed approach, which was based on simulations. After providing a concise overview of the simulation environment, the setup, and the parameters used, the chapter focused on the evaluation of the proposed approach, examining the following aspects: efficiency, offset, solution size, dominating degree, construction time, as well as, scalability. Moreover, for each of the aspects, the proposed approach was compared to the recently devised reference protocol by Sausen and colleagues [130]. Finally, at the end of the chapter, the comparison was extended to all state-of-the-art approaches for the key aspects.

For the sake of brevity, the following overview of the results was limited to the most important findings of the evaluation.

Looking at the efficiency, the evaluation shows that the communication cost incurred by the reference approach grows quadratically with increasing effective maximum dominating distance k' (see Definition 22 on page 101) and linearly with average node degree d. In contrast, the cost incurred by the proposed approach remains independent of k'. It increases linearly with d, but at a far lower rate than for the reference approach. Comparing the proposed approach to all state-of-the-art approaches, it is the only one whose cost is independent of k'.

The evaluation moreover considered the quality of the produced solutions. For instance, the figures indicate that the reference and the proposed approaches produce CkDS structures of approximately the same size for reasonably high k' (that is > 3). CkDS created by the proposed approach further tend to be more regular, since exhibiting a lower standard deviation of the dominating degrees of dominating nodes within a simulation run. The dominating degree of a node v is defined as the number of dominating neighbors of v.

When studying the evolution of the construction process, it is evident from the results that the amount of time, needed by the proposed approach to produce a solution, is independent of k', as well as, of the number of nodes n. In contrast, the time required for this by the reference approach grows linearly with k' and n. Comparing the proposed approach to all state-of-the-art approaches, it is the only one whose construction time is independent of k'.

Table 5.1 summarizes the key findings.

6 Conclusion

In this thesis, I propose a self-organizing random walk-based protocol for connected k-hop dominating set (CkDS) construction in wireless sensor networks, which draws inspiration from the general technique of random walks and, in particular, from the flight behavior of ovipositing Pieris rapae. The protocol, sharing many properties with the observed behavior of P. rapae, is solely based on local information, being used by locally-interacting distributed agents, constructing the global solution in a parallel manner. Moreover, it is the first protocol for the construction of connected dominating structures, including CDS and CkDS, in wireless networks to be based on random walks.

The effects of the proposed protocol's design, which differs fundamentally from state of the art, become evident in the evaluation, which includes a comparison with existing CkDS construction approaches: While the communication cost incurred during the construction of a CkDS by all state-of-the-art approaches depends linearly or quadratically on the effective maximum dominating distance k', the cost incurred by the proposed protocol is independent of it. The time needed by the proposed approach to construct a CkDS is similarly independent of k', whereas the construction time required by all state-of-the-art approaches grows in a linear manner with this parameter. Furthermore, the construction time needed by the proposed protocol is independent of the number of nodes n in the network, which is a property that it has in common only with one additional approach from state of the art (all others are linearly dependent).

Compared to the state-of-the-art approach employed as reference in the simulation-based part of the evaluation, which shares many common properties, such as the dependence of its cost on k and d, with other existing CkDS construction protocols, the proposed approach is by up to multiple orders of magnitude more efficient. In addition, it exhibits further favorable characteristics, for instance, that it produces highly regular solutions.

Bibliography

[1] Norman Abramson. The Aloha system – another alternative for computer communication. Technical Report AFOSR 70-1686 TR, Air Force Office of Scientific Research (SRMA), 1970.

[2] Cedric Adjih, Philippe Jacquet, and Laurent Viennot. Computing connected dominated sets with multipoint relays. *Ad Hoc & Sensor Wireless Networks*, 1(1-2), 2005. ocpscience.

[3] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002. Elsevier.

[4] David S. Alberts, John J. Garstka, and Frederick P. Stein. *Network Centric Warfare: Developing and Leveraging Information Superiority*. CCRP, 1999. http://www.dodccrp.org/files/Alberts_NCW.pdf, accessed on September 22, 2008.

[5] K. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, volume 9, pages 3849–3855, Hawaii, USA, January 7–10, 2002. IEEE Computer Society.

[6] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *IEEE ComSoc/KICS Journal on Communication Networks*, 4(1):22–29, March 2002. IEEE.

[7] Khaled M. Alzoubi, Peng-Jun Wan, and Ophir Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02)*, pages 157–164, Lausanne, Switzerland, June 9–11, 2002. ACM.

[8] Larry Armstrong, Stephen Baker, John A. Byrne, Otis Port, Diane Brady, Karen Pennar, John Carey, Steven V. Brull, Paul Raeburn, Catherine Arnst, Neil Gross, Peter Coy, Richard S. Dunham, Richard A. Melcher, Diane Brady, and Evelyn L. Wright. 21 ideas for the 21st century. *Business Week*, pages 28–167, August 30, 1999.

[9] Anish Arora, Rajiv Ramnath, Emre Ertin, Prasun Sinha, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Hui Cao, Mukundan Sridharan, Santosh Kumar, Nick Seddon, Chris Anderson, Ted Herman, Nishank Trivedi, Chen Zhang, Mikhail Nesterenko, Romil Shah, Sandeep Kulkarni, Mahesh Aramugam, Limin Wang, Mohamed Gouda, Young ri Choi, David Culler, Prabal Dutta, Cory Sharp, Gilman Tolle, Mike Grimmer, Bill Ferriera, and Ken Parker. ExScal: Elements of an extreme scale wireless sensor network. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005)*, pages 102–108, Hong Kong, China, August 17–19, 2005. IEEE.

[10] IEEE Standards Association. IEEE 802.16 Broadband Wireless Metropolitan Area Network (WirelessMAN). Online: http://standards.ieee.org/getieee802/802.16.html. Accessed on February 23, 2009.

[11] Lichun Bao and J. J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, pages 129–140, Annapolis, Maryland, USA, June 1–3, 2003. ACM.

[12] Antonio M. Baptista, Michael Wilkin, Phillip Pearson, Paul Turner, Cole McCandlish, and Philip Barrett. Coastal and estuarine forecast systems. *Earth System Monitor*, 9(3):1–5, March 1999. National Oceanographic Data Center (NODC).

[13] Paul Baran. On distributed communications: I. introduction to distributed communications networks. Memorandum RM-3420-PR, RAND Corporation, August 1964.

[14] Elizabeth Basha and Daniela Rus. Design of early warning flood detection systems for developing countries. In *Proceedings of the IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD 2007)*, pages 1–10, Bangalore, India, December 15–16, 2007. IEEE.

[15] Marcel Baunach, Reiner Kolla, and Clemens Mühlberger. SNoW^5: A versatile ultra low power modular node for wireless ad hoc sensor networking. Technical Report TR 424, Universität Würzburg, Lehrstuhl für Informatik V, May 2007.

[16] Richard Beckwith, Dan Teibel, and Pat Bowen. Report from the field: Results from an agricultural wireless sensor network. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 471–478, Tampa, Florida, USA, November 16–18, 2004. IEEE Computer Society.

[17] J. P. Benson, T. ODonovan, P. OSullivan, U. Roedig, and C. Sreenan. Car-park management using wireless sensor networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN'06)*, pages 588–595, Tampa, Florida, USA, November 14–16, 2006. IEEE.

[18] Carita M. Bergman, James A. Schaefer, and S. N. Luttich. Caribou movement as a correlated random walk. *Oecologia*, 123(3):364–374, May 2000. Springer.

[19] David A. Beyer. Accomplishments of the DARPA SURAN program. In *Proceedings of the IEEE Military Communications Conference (MILCOM '90)*, volume 2, pages 855–862, Monterey, CA, USA, September 1990. IEEE.

[20] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA, 1999.

[21] Alex Brooks, Alexei Makarenko, Tobias Kaupp, Stefan B. Williams, and Hugh F. Durrant-Whyte. Implementation of an indoor active sensor network. In *Experimental Robotics IX*, Springer Tracts in Advanced Robotics, pages 397–406. Springer Berlin / Heidelberg, 2004.

[22] C. J. Bryan and C. E. Nishimura. Monitoring oceanic earthquakes with SOSUS: an example from the caribbean. *Oceanography*, 8(1):4–10, 1995.

[23] Sergiy Butenko, Xiuzhen Cheng, Ding-Zhu Du, and Panos M. Pardalos. *Cooperative Control: Models, Applications and Algorithms*, chapter On the construction of virtual backbone for ad hoc wireless network, pages 43–54. Kluwer Academic Publishers, 2003.

[24] Sergiy Butenko, Xiuzhen Cheng, Carlos A. S. Oliveira, and P. M. Pardalos. *Recent Developments in Cooperative Control and Optimization*, chapter 4: A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks, pages 61–73. Kluwer Academic Publishers, 2004.

[25] Zack J. Butler, Peter I. Corke, Ronald A. Peterson, and Daniela Rus. Virtual fences for controlling cows. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4429–4436, New Orleans, LA, USA, April 26–May 1, 2004. IEEE.

[26] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. *Self-Organization in Biological Systems: Princeton Studies in Complexity*. Princeton University Press, 2003.

[27] Mihaela Cardei, Xiaoyan Cheng, Xiuzhen Cheng, and Ding zhu Du. Connected domination in multihop ad hoc wireless networks. In *Proceedings of the 6th International Conference on Computer Science and Informatics (CS&I 2002)*, Athens, Greece, June 25–28, 2002.

[28] Gianni Di Caro, Frederick Ducatelle, and Luca Maria Gambardella. AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 461–470, Birmingham, UK, September 18–22, 2004.

[29] Yvonne Carts-Powell. Unattended ground sensors stop and analyze the roses. SPIE Newsroom, number 196, online, April 2000. http://spie.org/x18996.xml, accessed September 24, 2008.

[30] Anantha Chandrakasan, Rex Min, Manish Bhardwaj, Seong-Hwan Cho, and Alice Wang. Power aware wireless microsensor systems. In *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC)*, pages 47–54, Firenze, Italy, September 24–26, 2002.

[31] Ben-Jye Chang, Bo-Jhang Huang, and Ying-Hsin Liang. Wireless sensor network-based adaptive vehicle navigation in multihop-relay WiMAX networks. In *Proceedings of IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 56–63, GinoWan, Okinawa, Japan, March 25–28, 2008. IEEE.

[32] Kuo-Chu Chang, Shozo Mori, and Chee-Yee Chong. Performance evaluation of a multiple-hypothesis multi-target tracking algorithm. In *Proceedings of the 29th IEEE Conference on Decision and Control*, volume 4, pages 2258–2263, Honolulu, HI, USA, December 1990. IEEE.

[33] Xiao Chen and Jian Shen. Reducing connected dominating set size with multipoint relays in ad hoc wireless networks. In *Proceedings of 7th International Symposium on Parallel Architectures, Algorithms and Networks*, pages 539–543, May 10–12, 2004.

[34] Xiao Chen and Jian Shen. Improved schemes for power-efficient broadcast in ad hoc networks. *Int. J. High Perform. Comput. Netw.*, 4(3/4):198–206, 2006.

[35] Yuanzhu Peter Chen and Arthur L. Liestman. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02)*, pages 165–172, Lausanne, Switzerland, June 9–11, 2002. ACM.

[36] Xiuzhen Cheng, Min Ding, David Hongwei Du, and Xiaohua Jia. Virtual backbone construction in multihop ad hoc wireless networks. *Wireless Communications and Mobile Computing*, 6(1):183–190, February 2006. John Wiley & Sons.

[37] Ching-Chuan Chiang and Mario Gerla. Routing and multicast in multihop, mobile wireless networks. In *Conference Record of IEEE 6th International Conference on Universal Personal Communications*, volume 2, pages 546–551, San Diego, CA, USA, October 12–16, 1997. IEEE.

[38] Texas Instruments Chipcon Products. CC1000 – single chip very low power RF transceiver. Online: http://focus.ti.com/lit/ds/symlink/cc1000.pdf. Accessed on November 13, 2008.

[39] Texas Instruments Chipcon Products. CC2420 – 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Online: http://inst.eecs.berkeley.edu/˜cs150/Documents/CC2420.pdf. Accessed on February 19, 2009.

[40] Imrich Chlamtac and András Faragó. A new approach to the design and analysis of peer-to-peer mobile networks. *Wirel. Netw.*, 5(3):149–156, 1999. Springer US.

[41] Chee-Yee Chong and Srikanta P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003. IEEE.

[42] Vlad Coroama, Jürgen Bohn, and Friedemann Mattern. Living in a smart environment – implications for the coming ubiquitous information society. In *SMC*, volume 6, pages 5633–5638, 2004.

[43] Crossbow Technology, Inc. IRIS Wireless Measurement System. Online: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/IRIS_OEM_Datasheet.pdf. Accessed on February 23, 2009.

[44] Crossbow Technology, Inc. MICAz Wireless Measurement System. Online: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf. Accessed on February 23, 2009.

[45] Fei Dai and Jie Wu. Distributed dominant pruning in ad hoc networks. In *Proceedings of IEEE International Conference on Communications (ICC '03)*, volume 1, pages 353–357, Anchorage, Alaska, USA, May 11-15, 2003. IEEE.

[46] Fei Dai and Jie Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(10):908–920, October 2004. IEEE.

[47] Fei Dai and Jie Wu. On constructing k-connected k-dominating set in wireless networks. In *Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS 2005)*, Denver, CA, USA, April 4–8, 2005.

[48] P. Daly. Navstar GPS and GLONASS: global satellite navigation systems. *Electronics & Communication Engineering Journal*, 5(6):349–357, December 1993. IET.

[49] Bevan Das and Vaduvur Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 1, pages 376–380, Montreal, Quebec, Canada, June 8–12, 1997. IEEE.

[50] Bevan Das, Raghupathy Sivakumar, and Vaduvur Bharghavan. Routing in ad hoc networks using a spine. In *Proceedings of Sixth International Conference on Computer Communications and Networks*, pages 34–39, Las Vegas, NV, USA, September 22–25, 1997. IEEE.

[51] D. W. Davies, K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. A digital communication network for computers giving rapid response at remote terminals. In *Proceedings of the First ACM Symposium on Operating System Principles (SOSP '67)*, pages 2.1–2.17, New York, NY, USA, 1967. ACM.

[52] Christopher P. Diehl, Mahesh Saptharishia, John B. Hampshire, and Pradeep K. Khosla. Collaborative surveillance using both fixed and mobile unattended ground sensors. *Proceedings of the SPIE*, 3713:178–185, July 1999.

[53] Sean Dodson. The internet of things. *The Guardian*, October 9, 2003.

[54] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. Scenarios for ambient intelligence in 2010. Technical report, Institute for Prospective Technological Studies, Seville, February 2001.

[55] EPCglobal Inc. EPCglobal Homepage. Online: http://www.epcglobalinc.org/home/. Accessed February 23, 2009.

[56] Anthony Ephremides, Jeffrey E. Wieselthier, and Dennis J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. In *Proceedings of the IEEE*, volume 75, pages 56–73. IEEE, January 1987.

[57] Irfan A. Essa. Ubiquitous sensing for smart and aware environments. *IEEE Personal Communications*, 7(5):47–49, October 2000. IEEE.

[58] ETH Zurich. BTnodes – A Distributed Environment for Prototyping Ad Hoc Networks. Online: http://www.btnode.ethz.ch/. Accessed on February 23, 2009.

[59] ETH Zurich. The Sensor Network Museum: Main – Home Page. Online: http://www.snm.ethz.ch/Main/Home Page. Accessed on February 23, 2009.

[60] L. Evers, M. J. J. Bijl, M. Marin-Perianu, R. Marin-Perianu, and P. J. M. Havinga. Wireless sensor networks and beyond: A case study on transport and logistics. In *Proceedings of the International Workshop on Wireless Ad-hoc Networks (IWWAN 2005)*, London, UK, May 2005.

[61] eyes Consortium. eyes – energy efficient sensor networks. Online: http://www.eyes.eu.org/. Accessed on February 23, 2009.

[62] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, pages 222–232, Annapolis, Maryland, USA, June 1–3, 2003. ACM.

[63] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.

[64] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[65] David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesC language: A holistic approach to networked embedded systems. In *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI '03)*, pages 1–11, New York, NY, USA, 2003. ACM.

[66] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 1(3):255–265, September 1995. Springer US.

[67] J. H. Gillespie. Natural selection for within-generation variance in offspring number. *Genetics*, 76:601–608, 1974. Genetics Society of America.

[68] Google. Google maps. Online: http://maps.google.com/. Accessed on February 20, 2009.

[69] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76:579–581, 1989.

[70] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998. Springer New York.

[71] Himanshu Gupta, Zongheng Zhou, Samir R. Das, and Quinyi Gu. Connected sensor cover: self-organization of sensor networks for efficient query execution. *IEEE/ACM Transactions on Networking*, 14(1):55–67, February 2006. IEEE.

[72] Dirk Haehnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with RFID technology. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 1015–1020. IEEE, April 26–May 1, 2004.

[73] Mohamed Hefeeda and Majid Bagheri. Wireless sensor networks for early detection of forest fires. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 1–6, Pisa, Italy, October 8–11, 2007. IEEE.

[74] P. Hewkin. Smart tags – the distributed-memory revolution. *IEE Review*, 35:203–206, June 1989.

[75] Michael G. Hinchey and Roy Sterritt. 99% (biological) inspiration ... In *Proceedings of the 1st IFIP International Conference on Biologically Inspired Computing (BICC 2006)*, pages 7–20, Santiago, Chile, August 21–24, 2006. Springer.

[76] Tomasz Imielinski and Samir Goel. Dataspace: querying and monitoring deeply networked collections in physical space. *IEEE Personal Communications*, 7(5):4–9, October 2000. IEEE.

[77] J-Sim. Online: http://www.j-sim.org. Accessed on February 17, 2009.

[78] Peter Janacik and Alexander Kujat. Biologically-inspired construction of connected k-hop dominating sets in wireless sensor networks. In *Proceedings of the Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2009)*, San Francisco, CA, USA, September 14–18, 2009. IEEE.

[79] Peter Janacik, Johannes Lessmann, and Michael Karch. Distributed simulation environment for the ShoX network simulator. In *Proceedings of the Sixth International Conference on Networking and Services (ICNS)*. IEEE Computer Society Press, March 2010.

[80] Peter Janacik, Johannes Lessmann, and Michael Karch. Multi-view communication visualization for wireless network simulations. In *Proceedings of the International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Liverpool, UK, January 27–29, 2010. IEEE Computer Society Press.

[81] Laura Huei jiun Ju, Izhak Rubin, Kevin Ni, and Christopher Wu. A distributed mobile backbone formation algorithm for wireless ad hoc networks. In *Proceedings of the 1st International Conference on Broadband Networks (BROADNETS)*, pages 661–670, San José, CA, USA, October 25–29, 2004. IEEE.

[82] John Hopkins APL Technical Digest. The cooperative engagement capability. Online: http://techdigest.jhuapl.edu/td1604/APLteam.pdf, 1995. Accessed on September 22, 2008.

[83] David B. Johnson and David A. Maltz. *Mobile Computing: Dynamic Source Routing in Ad Hoc Wireless Networks*, volume 353, chapter 6, pages 153–181. Springer US, 1996.

[84] Robert E. Kahn. The organization of computer resources into a packet radio network. *IEEE Transactions on Communications*, 25(1):169–178, January 1977. IEEE.

[85] Robert E. Kahn and Jerry Burchfiel. Advances in packet radio technology. *Proceedings of the IEEE*, 66(11):1468–1496, November 1978. IEEE.

[86] Cornelia Kappler and Georg Riegel. A real-world, simple wireless sensor network for monitoring electrical energy consumption. In *Proceedings of the First European Workshop on Wireless Sensor Networks*, Lecture Notes in Computer Science, pages 339–352, Berlin, Germany, January 19–21, 2004. Springer.

[87] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley & Sons, 2005.

[88] Tom Kevan. Shipboard machine monitoring for predictive maintenance. *Sensors Online (http://www.sensorsmag.com/)*, February 2006. Published online.

[89] Taek Jin Kwon and Mario Gerla. Efficient flooding with passive clustering (PC) in ad hoc networks. *SIGCOMM Comput. Commun. Rev.*, 32(1):44–56, 2002. ACM.

[90] Laitche. Pieris rapae. Online image source: http://en.wikipedia.org/wiki/File:Small_White_November_2007.jpg. Accessed on March 27, 2009. Released into the public domain by its author.

[91] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11-1997, wireless LAN medium access control (MAC) and physical (PHY) layer specifications, Novemeber 1997.

[92] Johannes Lessmann, Tales Heimfarth, and Peter Janacik. ShoX: An easy to use simulation platform for wireless networks. In *Proceedings of Tenth International Conference on Computer Modeling and Simulation, 2008*, pages 410–415, Cambridge, UK, 1–3 April 2008. IEEE.

[93] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an MST-based topology control algorithm. In *IEEE Transactions on Wireless Communications*, volume 4, pages 1195–1206. IEEE, May 2005.

[94] Hyojun Lim and Chongkwon Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000)*, pages 61–68, Boston, MA, USA, August 11, 2000. ACM.

[95] Chunhung Richard Lin and Mario Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, September 1997. IEEE.

[96] Haitao Liu and Rajiv Gupta. Selective backbone construction for topology control in ad hoc networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 41–50, Fort Lauderdale, FL, USA, October 25–27, 2004. IEEE.

[97] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 88–97, New York, NY, USA, 2002. ACM.

[98] D. A. Maltz, J. Broch, and D. B. Johnson. Quantitative lessons from a full-scale multi-hop wireless ad hoc network testbed. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 992–997, Chicago, IL, USA, September 23–28, 2000. IEEE.

[99] Mahesh K. Marina and Samir R. Das. On-demand multi path distance vector routing in ad hoc networks. In *Proceedings of the Ninth International Conference on Network Protocols (ICNP '01)*, Riverside, CA, USA, November 11–14, 2001. IEEE Computer Society.

[100] Friedemann Mattern. Pervasive/Ubiquitous Computing – Aktuelles Schlagwort. *Informatik Spektrum*, 24(3):145–147, 2001. Springer.

[101] W. M. Merrill, H. L. N. Liu, J. Leong, K. Sohrabi, and G. J. Pottie. Quantifying short-range surface-to-surface communications links. *IEEE Antennas and Propagation Magazine*, 46(3):36–46, June 2004. IEEE.

[102] Hugh B. Milburn, Alex I. Nakamura, and Frank I. Gonzalez. Real-time Tsunami reporting from the deep ocean. Online: http://nctr.pmel.noaa.gov/milburn1996.html. Accessed on February 19, 2009.

[103] Manki Min, Hongwei Du, Xiaohua Jia, Christina Xiao Huang, Scott C.-H. Huang, and Weili Wu. Improving construction for connected dominating set with Steiner tree in wireless sensor networks. *J. of Global Optimization*, 35(1):111–119, 2006. Springer US.

[104] Manki Min, Feng Wang, Ding-Zhu Du, and Panos M. Pardalos. A reliable virtual backbone scheme in mobile ad-hoc networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, pages 60–69, Fort Lauderdale, FL, USA, October 25–27, 2004. IEEE.

[105] Rex Min and Anantha Chandrakasan. Top five myths about the energy consumption of wireless communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(1):65–67, 2003. ACM.

[106] U.S. Navy. History of IUSS. Online: http://www.cus.navy.mil/timeline.htm. Accessed on September 17, 2008.

[107] DARPA/NSF. The network simulator ns 2. Online: http://www.isi.edu/nsnam/ns/. Accessed on February 1, 2009.

[108] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, pages 151–162, Seattle, WA, USA, August 17–19, 1999. ACM.

[109] Hiro-Sato Niwa. Random-walk dynamics of exploited fish populations. *ICES Journal of Marine Science*, 64:496–502, 2007. International Council for the Exploration of the Sea, Oxford Journals.

[110] Opnet Technologies, Inc. Opnet. Online: http://www.opnet.com, accessed on October 12, 2008.

[111] Nathan Ota and Paul Wright. Trends in wireless sensor networks for manufacturing. *IJMR*, 1(1):3–17, 2006.

[112] Srinivasan Parthasarathy and Rajiv Gandhi. Fast distributed well connected dominating sets for ad hoc networks. Technical Report CS-TR-4559, Department of Computer Science, University of Maryland, 2002.

[113] B. Paul and S.V. Rao. *Distributed Computing and Internet Technology*, chapter Distributed Clustering Algorithm for Finding Virtual Backbone in Ad Hoc Networks, pages 50–55. Springer Berlin / Heidelberg, 2005.

[114] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pages 234–244, London, UK, August 31–September 2, 1994. ACM.

[115] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, volume 2, New Orleans, LA, USA, February 25–26, 1999. IEEE Computer Society.

[116] Joseph Polastre, Robert Szewczyk, and David E. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05)*, pages 364–369, Los Angeles, CA, USA, April 25–27, 2005.

[117] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, pages 3866–3875, Big Island, HI, USA, January 7–10, 2002. IEEE Computer Society.

[118] QualNet. Online: http://www.qualnet.com. Accessed on October 15th, 2009.

[119] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 181–192, Los Angeles, CA, USA, November 5–7, 2003. ACM.

[120] Richard F. Rashid and George G. Robertson. Accent: A communication oriented network operating system kernel. In *Proceedings of the Eighth ACM Symposium on Operating Systems Principles (SOSP '81)*, pages 64–75, Pacific Grove, CA, USA, December 14–16, 1981. ACM.

[121] RF Monolithics, Inc. TR1000 – 916.50 MHz Hybrid Transceiver. Online: http://www.rfm.com/products/data/tr1000.pdf. Accessed on February 19, 2009.

[122] RF Monolithics, Inc. TR1001 – 868.35 MHz Hybrid Transceiver. Online: http://www.rfm.com/products/data/tr1001.pdf. Accessed on February 19, 2009.

[123] Ruud Riem-Vis. Cold chain management using an ultra low power wireless sensor network. In *Proceedings of the ACM MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, Boston, MA, USA, June 6, 2004. ACM.

[124] Lawrence G. Roberts. Multiple computer networks and intercomputer communication. In *Proceedings of the First ACM Symposium on Operating System Principles (SOSP '67)*, pages 3.1–3.6, New York, NY, USA, 1967. ACM.

[125] Richard B. Root and Peter M. Kareiva. The search for resources by cabbage butterflies (pieris rapae): Ecological consequences and adaptive significance of markovian movements in a patchy environment. *Ecology*, 65(1):147–165, 1984. Ecological Society of America.

[126] Wade Roush, Alexandra M. Goho, Eric Scigilano, David Talbot, M. Mitchell Waldrop, Gregory T. Huang, Peter Fairley, Erika Jonietz, Jon Cohen, and Herb Brody. 10 emerging technologies that will change the world. *MIT Technology Review*, 106(1):33–49, February 2003.

[127] Lu Ruan, Hongwei Du, Xiaohua Jia, Weili Wu, Yingshu Li, and Ker-I Ko. A greedy approximation for minimum connected dominating sets. *Theor. Comput. Sci.*, 329(1–3):325–330, 2004. Elsevier.

[128] Silvia Santini, Benedikt Ostermaier, and Andrea Vitaletti. First experiences using wireless sensor networks for noise pollution monitoring. In *Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks (REALWSN '08)*, pages 61–65, New York, NY, USA, Glasgow, Scotland, UK, 2008. ACM.

[129] M. Satyanarayanan. Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17, August 2001. IEEE.

[130] Paulo Sausen, Marco Aurlio Spohn, Antonio Marcus Nogueira de Lima, and Angelo Perkusich. Bounded-distance multi-coverage backbones in wireless sensor networks. In *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC)*, pages 203–208, Seoul, Korea, March 11–15, 2007. ACM.

[131] ScatterWeb GmbH. ScatterWeb – wireless network solutions. Online: http://www.scatterweb.com/. Accessed on February 23, 2009.

[132] Jochen Schiller. *Mobile Communications*. Addison Wesley, 2nd edition, August 2003.

[133] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett, Jr., and H. Scott Matthews. Intelligent light control using sensor networks. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 218–229, San Diego, CA, USA, November 2–4, 2005. ACM.

[134] Katayoun Sohrabi, Bertha Manriquez, and Gregory J. Pottie. Near ground wideband channel measurement in 800–1000 MHz. In *Proceedings of the IEEE 49th Vehicular Technology Conference*, volume 1, pages 571–574, Houston, TX, USA, May 16–20, 1999. IEEE.

[135] Sourceforge.net. ShoX – a scalable ad hoc network simulator. Online: http://shox.sourceforge.net/. Accessed on March 10, 2009.

[136] David C. Steere, António M. Baptista, Dylan McNamee, Calton Pu, and Jonathan Walpole. Research challenges in environmental observation and forecasting systems. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 292–299, Boston, MA, USA, August 6–11, 2000. ACM.

[137] R. A. Stephen, J. A. Collins, A. Hildebrand, J. Orcutt, K. R. Peal, F. N. Spiess, and F. L. Vernon. The ocean seismic network pilot experiment. Online: http://msg.whoi.edu/osn/EOS/EOS_paper_additional_jul_23_99.html. Accessed on October 21, 2008.

[138] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, January 2002. IEEE.

[139] Sun Microsystems, Inc. Sunspotworld – project sun spot products. Online: http://www.sunspotworld.com/products/. Accessed on February 23, 2009.

[140] Kyungbok Sung, Jaejun Yoo, and Dohyun Kim. Collision warning system on a curved road using wireless sensor networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, pages 1942–1946, Baltimore, MD, USA, September 30–October 3, 2007. IEEE.

[141] Robert Szewczyk, Joseph Polastre, Alan M. Mainwaring, and David E. Culler. Lessons from a sensor network expedition. In *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN)*, volume 2920 of *Lecture Notes in Computer Science*, pages 307–322, Berlin, Germany, January 19–21, 2004. Springer.

[142] Ali Maleki Tabar, Arezou Keshavarz, and Hamid Aghajan. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the 4th ACM International Workshop on Video Surveillance & Sensor Networks (VSSN '06)*, pages 145–154, Santa Barbara, CA, USA, October 27, 2006. ACM.

[143] Fabrice Theoleyre and Fabrice Valois. A virtual structure for hybrid networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 1040–1045, Atlanta, GA, USA, March 21–25, 2004. IEEE.

[144] Fabrice Theoleyre and Fabrice Valois. *Self-Stabilizing Systems*, volume 3764, chapter Self-stabilization of a Virtual Topology for Self-organization in Ad Hoc Networks, pages 214–228. Springer Berlin / Heidelberg, October 2005.

[145] Guy Theraulaz and Eric Bonabeau. A brief history of stigmergy. *Artificial Life*, 5(2):97–116, Spring 1999. MIT Press Journals.

[146] A. Tiwari, F. L. Lewis, and S. S. Ge. Wireless sensor network for machine condition based maintenance. In *Proceedings of the 8th Control, Automation, Robotics and Vision Conference (ICARCV)*, volume 1, pages 461–467, Kunming, China, December 6–9, 2004. IEEE.

[147] Chai-Keong Toh. Associativity-based routing for ad hoc mobile networks. *Wirel. Pers. Commun.*, 4(2):103–139, 1997. IEEE.

[148] Chai-Keong Toh, Richard Chen, Minar Delwar, and Donald Allen. Experimenting with an ad hoc wireless network on campus: insights and experiences. *SIGMETRICS Performance Evaluation Review*, 28(3):21–29, 2000. ACM.

[149] University of California at Los Angeles, Parallel Computing Laboratory. Parsec – parallel simulation environment for complex systems. Online at http://pcl.cs.ucla.edu/projects/parsec/, accessed September 27, 2008.

[150] A. Vargas. Omnet++ – discrete event simulation system. Online: http://www.omnetpp.org. Accessed on February 10, 2009.

[151] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic. An assisted living oriented information system based on a residential wireless sensor network. In *Proceedings of 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference*, pages 95–100, Arlington, VA, USA, April 2–4, 2006. IEEE.

[152] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):141–149, 2004. Kluwer Academic Publishers.

[153] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992. ACM.

[154] Mark Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–100, 1991.

[155] Mark Weiser and John Seely Brown. *Beyond Calculation: the Next Fifty Years*, chapter The Coming Age of Calm Technology, pages 75–85. Copernicus, 1997.

[156] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 14–27, Los Angeles, CA, USA, November 5–7, 2003. ACM.

[157] Wouterh. Micaz. Online: http://www.flickr.com/photos/wouterh/2409251427/, license: http://creativecommons.org/licenses/by-nc-sa/2.0/deed.en. Accessed on February 20, 2009.

[158] Wouterh. Sun spot. Online: http://www.flickr.com/photos/wouterh/2409251097/, license: http://creativecommons.org/licenses/by-nc-sa/2.0/. Accessed on February 19, 2009.

[159] Jie Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):866–881, September 2002. IEEE.

[160] Jie Wu. An enhanced approach to determine a small forward node set based on multipoint relays. In *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC 2003-Fall)*, volume 4, pages 2774–2777, Orlando, FL, USA, October 6–9, 2003. IEEE.

[161] Jie Wu, Mihaela Cardei, Fei Dai, and Shuhui Yang. Extended dominating set in ad hoc networks using co-operative communication. In *Proceedings of the IFIP Networking Conference*, Waterloo, Ontario, Canada, May 2–6, 2005. IFIP. Poster Session.

[162] Jie Wu, Fei Dai, Ming Gao, and Ivan Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *Journal of Communication and Networks*, 4(1):59–70, March 2002. Korea Information and Communications Society.

[163] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M 1999)*, pages 7–14, Seattle, WA, USA, August 20, 1999. ACM.

[164] Jie Wu and Hailan Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1–3):13–36, 2001. Springer US.

[165] Hong-Yen Yang, Chia-Hung Lin, and Ming-Jer Tsai. Distributed algorithm for efficient construction and maintenance of connected k-hop dominating sets in mobile ad hoc networks. *IEEE Transactions on Mobile Computing (TMC)*, 7(4):444–457, April 2008. IEEE.

[166] Shuhui Yang, Jie Wu, and Jiannong Cao. Connected k-hop clustering in ad hoc networks. In *Proceedings of the International Conference on Parallel Processing (ICPP)*, pages 373–380, June 2005.

[167] Wei Ye, John S. Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, USA, June 23–27, 2002. IEEE.

[168] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys '06)*, pages 321–334, Boulder, CO, USA, October 31–November 3, 2006. ACM.

[169] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 1–13, Los Angeles, CA, USA, November 5–7, 2003. ACM.