

# **Quantifying and Reducing the Cost of Cooperative Relaying in Wireless Multi-Hop Networks**

**Dissertation**

Hermann Simon Lichte

zur Erlangung des akademischen Grades

*Doktor der Naturwissenschaften (Dr. rer. nat.)*

im Fach Informatik

eingereicht an der

Fakultät für Elektrotechnik, Informatik und Mathematik

an der

Universität Paderborn

Gutachter:

Prof. Dr. rer. nat. Holger Karl (Universität Paderborn)

Prof. Dr.-Ing. habil. Falko Dressler (Universität Innsbruck)

Eingereicht am: 15. März 2011

Tag der mündlichen Prüfung: 17. Juni 2011

Für meine Tochter Paulina Margarete



# Abstract

Fast and reliable data transmission in wireless networks is hard to achieve due to interference and fading. Interference limits spatial reuse in the network and fading leads to high error rates for transmissions. These error rates can be exponentially reduced by cooperative relaying. With this approach, neighboring nodes can help the transmission by repeating the signal via independent channels. In large wireless multi-hop networks, this poses two major problems.

First, the additional transmission of relays causes additional interference. To achieve a desired outage capacity, receivers span guard zones to mitigate interference and thereby affect spatial reuse by consuming area – the so-called *spatial consumption*. Second, retransmission requires additional channel resources which typically reduces data rate – the so-called *multiplexing loss*. Without careful studies it is not obvious when these costs pay off.

Therefore, I study how the additional interference due to relaying affects the network's performance in two steps: First, I analyze the spatial consumption of the cooperative transmission using geometry. Second, I combine this analysis with outage capacity expressions. With the resulting analytical framework, I show that although relays require additional space by blocking neighbors, the diversity gain compensates for interference. I identify the operating regions of cooperative relaying and show that, especially at high robustness requirements, cooperative relaying even leads to better spatial reuse than conventional transmission approaches.

To reduce multiplexing loss for unicasts, I develop a routing-informed Medium Access Control (MAC) protocol that schedules a single relay retransmission such that it assists in two successive point-to-point transmissions, so-called *two-for-one cooperation*. I show how to integrate the proposed protocol into the IEEE 802.11 Wireless Local Area Network (WLAN) standard by modifying an open-source Linux WLAN driver. Measurement results show the efficiency of two-for-one cooperation in multi-hop networks.

Contrary to unicasts, multiplexing loss is not a cost anymore for broadcasts as in this case retransmissions happen anyway. Thus, to make broadcasts robust against fading, cooperative relaying can be used without sacrificing data rate. To broadcast as quickly as possible (i.e., with low latency), only a subset of nodes should retransmit messages. I find that existing heuristics for determining this so-called *broadcast set* have poor delivery ratios with fading channels. I identify the static unit disk model of transmission ranges to be the source of the problem and propose a probabilistic model as a replacement. I additionally exploit cooperation diversity during construction of the broadcast set to gain improved reliability while still keeping the size of the set and hence the latency low. Simulation results

---

show that the proposed Probabilistic Cooperation Diversity Broadcast (PCDB) decreases the time for broadcasts while still distributing packets to all nodes with high probability.

The simulation and measurement results in this thesis require implementations of the same MAC protocols on different platforms. To untie this complex development process, I propose to automate its most error-prone parts: Implementation of MAC automata, analysis, and code generation. To do so, I formalize cooperative MAC protocols by an easy-to-use specification language and propose strategies to construct compilers to automatically analyze validity and performance of the specification and to translate the specified protocols into program code for implementation.

# Zusammenfassung

Schnelle und zuverlässige Datenübertragung in drahtlosen Netzen ist aufgrund von Interferenz und Fading schwer zu erreichen. Interferenz beschränkt die räumliche Wiederverwendbarkeit in dem Netz und Fading führt zu hohen Fehlerraten für Übertragungen. Diese Fehlerraten können durch kooperatives Relaying exponentiell gesenkt werden. Bei diesem Verfahren unterstützen Nachbarknoten eine Übertragung, indem sie das Signal über unabhängige Kanäle wiederholen. Dies führt zu zwei gewichtigen Problemen in großen drahtlosen Multi-Hop-Netzen.

Erstens erzeugt die zusätzliche Übertragung des Relays zusätzliche Interferenz. Um eine gewünschte Ausfallkapazität zu erreichen, spannen Empfänger Schutzzonen auf, um die Interferenz zu beschränken. Die dadurch verbrauchte Fläche beeinträchtigt jedoch die räumliche Wiederverwendbarkeit. Zweitens benötigt die Übertragungswiederholung durch den Relay zusätzliche Kanalressourcen, die typischerweise die Datenrate reduzieren; ein Effekt, der als Multiplexingverlust bekannt ist. Ohne sorgfältige Studien ist nicht ersichtlich, wann sich diese Kosten amortisieren.

Aus diesem Grund analysiere ich in zwei Schritten, wie die zusätzliche Interferenz durch die Relay-Übertragung die Netzleistung beeinflusst. Zuerst verwende ich Geometrie, um den Flächenverbrauch der kooperativen Übertragung zu beschreiben. Anschließend kombiniere ich diese Analyse mit den Ausfallkapazitäts-Gleichungen. Mit dem daraus resultierenden analytischen Werkzeug zeige ich, dass trotz des zusätzlichen Flächenverbrauchs durch die Relays der dabei entstehende Diversitätsgewinn die Interferenz kompensiert. Ich identifiziere die Arbeitsbereiche von kooperativem Relaying und zeige, dass besonders bei hohen Robustheitsanforderungen kooperatives Relaying sogar zu einer besseren räumlichen Wiederverwendbarkeit führt als herkömmliche Übertraungsverfahren.

Um den Multiplexingverlust für Unicasts zu verringern, entwickle ich ein routinginformiertes Medienzugriffsprotokoll, das eine einzelne Relay-Übertragung so ausführt, dass diese zwei aufeinander folgende Punkt-zu-Punkt-Übertragungen unterstützt, so genannte *zwei-für-eins Kooperation*. Ich zeige durch Modifikation eines quelloffenen Linux-Gerätetreibers für WLAN Karten, wie man das vorgeschlagene Protokoll in den IEEE 802.11 WLAN Standard integrieren kann. Messergebnisse belegen die Effizienz der zwei-für-eins Kooperation in Multi-Hop-Netzen.

Im Gegensatz zu Unicasts stellt der Multiplexingverlust keine Kosten im Broadcast-Fall dar, weil dort in jedem Fall Übertragungswiederholungen stattfinden. Man kann deshalb kooperatives Relaying ohne Einbußen in der Datenrate einsetzen, um die Robustheit von Broadcasts gegenüber Fading zu erhöhen. Um Broadcasts so schnell wie möglich durchzuführen (also mit niedriger Latenz) reicht es, wenn nur eine Teilmenge der Knoten Nach-

---

richten wiederholt. Ich stelle fest, dass existierende Heuristiken, mit denen solche so genannten Broadcast-Mengen gefunden werden können, schlechte Auslieferungsraten bei Fading haben. Als Ursache des Problems identifiziere ich das statische Kreismodell von Übertragungsbereichen und schlage ein probabilistisches Modell als Ersatz vor. Dabei nutze ich zusätzlich kooperative Diversität aus, um während der Konstruktion der Broadcast-Menge von erhöhter Robustheit zu profitieren und gleichzeitig die Größe dieser Menge und damit die Latenz niedrig zu halten. Simulationen zeigen, dass der vorgeschlagene PCDB die für Broadcasts benötigte Zeit verringert und dennoch mit hoher Wahrscheinlichkeit die Pakete zu allen Knoten ausliefert.

Die Simulations- und Messergebnisse in dieser Arbeit machen die Implementierung desselben Medienzugriffsprotokolls auf unterschiedlichen Plattformen erforderlich. Um diesen komplexen Entwicklungsprozess zu vereinfachen und zu beschleunigen, schlage ich ein Verfahren zur Automatisierung der fehleranfälligen Teile vor, nämlich Implementierung des Automaten, Analyse und Codeerzeugung. Dazu entwerfe ich eine leicht zu benutzende Spezifikationssprache zur Formalisierung kooperativer Medienzugriffsprotokolle und zeige Ansätze, mit denen man Compiler erzeugen kann, die automatisch die Leistung des spezifizierten Protokolls im Vorfeld analysieren und den erforderlichen Code für die Implementierung generieren.

# Acknowledgments

*Meinen Eltern Roswitha und Hans-Hermann, die mir immer ermöglichten, meine Ziele zu verfolgen. Meiner Frau Corina, die auch in schweren Zeiten zu mir hielt und die ich dafür umso mehr liebe. Und meiner Tochter Paulina, die mich daran erinnerte, wieder mit dem unbedarften Blick eines Kindes zu sehen. Ihr habt diese Dissertation mehr beeinflusst, als euch bewusst ist. Danke.*

I carried out this research during my four years at the Computer Networks Group under the guidance of Holger Karl from the University of Paderborn. I sincerely thank him for supervising my research, keeping me focused, and always asking the right questions. I thank Falko Dressler from the University of Innsbruck for vice-supervising my work.

I am grateful to my colleagues from the Computer Networks Group who made these four years not only a productive but also a pleasant time of my life. I thank Stefan Valentin for the many publications that we accomplished together and for his valuable comments that improved the quality of this thesis. I thank Hannes Frey for supervising the FLYNET project group with me and for helping me write my first MobiHoc paper. I thank Tobias Volkhausen for sharing his MATLAB simulator with me and for bearing with me while finalizing this thesis. I thank Dereje Woldegebreal for his assistance whenever I had to solve deep electrical engineering problems. I thank Matthias Andree for sharing his broad knowledge about computer administration that helped me out whenever our technical support failed to do so quickly. I thank Tanja Langen for taking care of any non-computer administrative tasks that made working in a public institution easy. I thank Rafael Funke and Thorsten Biermann for their assistance in building a SORBAS prototype. I thank our lab technician Hans-Joachim Kraus for all the good handicraft work, without forgetting Michael Sessinghaus, Azeem Khan, Matthias Herrlich, Christian Dannewitz, and Tanja Bürger.

I am also grateful to Jörg Widmer, Imad Aad, and Luis Loyola for a very productive and uncomplicated industrial collaboration. Finally, I thank my students for their contributions of code – Jannis Weide for implementing obstacles in the Mobility Framework for OM-NeT++, and Kornelius Dridger for integrating my two-for-one cooperation protocol into an open-source Atheros device driver for Linux.

These were really good times.

Paderborn, March 2011.



# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Thesis contributions . . . . .	3
<b>2. Basics of cooperation and spatial reuse</b>	<b>7</b>
2.1. Channel model . . . . .	7
2.1.1. Outage probability . . . . .	9
2.2. Cooperative diversity . . . . .	10
2.2.1. Linear diversity combining . . . . .	10
2.2.2. Diversity order . . . . .	11
2.2.3. Outage capacity . . . . .	14
2.3. Characterizing spatial reuse . . . . .	14
2.3.1. Poisson point process . . . . .	15
2.3.2. Spatial consumption by guard zones . . . . .	16
<b>3. Spatial consumption of cooperative relaying</b>	<b>19</b>
3.1. Expected interference in random wireless networks with guard zones . . .	21
3.1.1. Distribution of $k$ -nearest neighbor's distance . . . . .	22
3.1.2. Interference from $k$ th neighbor . . . . .	23
3.1.3. Aggregate expected interference for $\alpha > 2$ . . . . .	24
3.1.4. Verification by simulation . . . . .	25
3.2. Linking spatial consumption with outage capacity . . . . .	29
3.2.1. Direct transmission . . . . .	31
3.2.2. Non-Cooperative Relaying (NCR) . . . . .	32
3.2.3. Cooperative Triangle (CTR) . . . . .	33
3.2.4. Weak Full Diamond (WFD) . . . . .	34
3.2.5. Strong Full Diamond (SFD) . . . . .	34
3.3. The effect of diversity order on spatial consumption . . . . .	37
3.4. Conclusions . . . . .	40
<b>4. Mechanisms for cooperative unicasting</b>	<b>43</b>
4.1. Rate-per-link adaptation for cooperative relaying . . . . .	44
4.1.1. Upper bound on the data rate for cooperative relaying . . . . .	45
4.1.2. Cooperative rate adaptation algorithm for $N$ relays . . . . .	46
4.1.3. Simulations in a fading scenario . . . . .	51

4.2.	Designing a routing-informed cooperative MAC protocol . . . . .	54
4.2.1.	Simulations in a fading scenario . . . . .	59
4.2.2.	Measurements using a modified IEEE 802.11a/g device driver . . . . .	63
4.3.	Conclusions . . . . .	69
<b>5.</b>	<b>Mechanisms for cooperative broadcasting</b>	<b>71</b>
5.1.	Problem statement . . . . .	72
5.2.	Flexible framework for broadcast set construction . . . . .	73
5.2.1.	Taxonomy of broadcast set construction heuristics . . . . .	73
5.2.2.	A unified framework for analysis and simulation . . . . .	77
5.3.	Fading-resistant low-latency broadcast sets . . . . .	83
5.3.1.	Probabilistic Cooperation Diversity Broadcast (PCDB) . . . . .	86
5.3.2.	Simulations in a fading scenario . . . . .	90
5.4.	Conclusions . . . . .	94
<b>6.</b>	<b>Rapid prototyping for cooperative MAC protocols</b>	<b>97</b>
6.1.	Describing MAC protocols by patterns . . . . .	98
6.1.1.	MAC patterns: Intuition and formal definition . . . . .	98
6.1.2.	Example: Non-cooperative patterns . . . . .	100
6.1.3.	Example: Cooperative patterns . . . . .	101
6.2.	Automating MAC development by patterns . . . . .	103
6.2.1.	Step 1: From MAC patterns to trees . . . . .	103
6.2.2.	Step 2: From trees to a pattern-based language . . . . .	106
6.3.	Compiler backends . . . . .	108
6.3.1.	Synthesis of patterns . . . . .	111
6.3.2.	Analysis backend . . . . .	114
6.3.3.	Simulator and SDR backends . . . . .	118
6.4.	Conclusions . . . . .	120
<b>7.</b>	<b>Conclusions and future work</b>	<b>123</b>
<b>A.</b>	<b>Mathematical derivations</b>	<b>127</b>
A.1.	Area of two circle's intersection . . . . .	127
A.2.	Area of three circle's intersection . . . . .	128
<b>B.</b>	<b>Occurrence probabilities of diamonds in random networks</b>	<b>131</b>
B.1.	Modeling cooperative triangles and diamonds . . . . .	131
B.2.	Counting in unobstructed and Manhattan grid scenarios . . . . .	132
<b>C.</b>	<b>NP-hardness of Minimum Latency Cooperative Broadcast</b>	<b>135</b>
C.1.	NP-hardness for the exponential distribution . . . . .	141

# List of Figures

1.1. Comparing a MISO system and CTR . . . . .	2
2.1. System model of the wireless channel . . . . .	7
2.2. The four relay configurations studied in this thesis . . . . .	12
2.3. Outage probabilities and diversity orders in a symmetric diamond . . . . .	13
2.4. Poisson process with node density $\lambda = 1/2$ in a square area of $8 \times 8$ . . . . .	16
2.5. Guard zone mitigating interference around receiver . . . . .	17
3.1. Structure of guard zones for three protocol classes . . . . .	20
3.2. Annulus capturing the $k$ -nearest neighbor . . . . .	22
3.3. Poisson node distribution in an independent scenario . . . . .	26
3.4. Poisson node distribution in a clustered scenario . . . . .	27
3.5. Closed-form expected interference verified by simulations . . . . .	28
3.6. Standard deviation of expected interference . . . . .	29
3.7. Direct transmission from source to destination . . . . .	32
3.8. NCR from source via relay to destination . . . . .	32
3.9. CTR rooted at source via relay to destination . . . . .	33
3.10. WFD rooted at source via two relays to destination . . . . .	35
3.11. SFD rooted at source via two relays to destination . . . . .	36
3.12. Spatial consumption with optimal guard zones at low robustness . . . . .	38
3.13. Spatial consumption with optimal guard zones at high robustness . . . . .	39
3.14. Operating regions with respect to spatial consumption . . . . .	41
4.1. Hop-wise and routing-informed cooperative relaying . . . . .	43
4.2. Outage capacities for direct transmission, (a)symmetric, and chain CTR . . . . .	47
4.3. BER of BPSK for uncoded transmission and convolutional coding . . . . .	48
4.4. Theoretical gain of rate-per-link over link-equal adaptation . . . . .	50
4.5. Received rate vs. transmission power, asymmetric configuration, $v=1$ m/s . . . . .	52
4.6. Received rate vs. transmission power, asymmetric configuration, $v=50$ m/s . . . . .	53
4.7. Received rate vs. transmission power, symmetric configuration, $v=1$ m/s . . . . .	53
4.8. Received rate vs. transmission power, chain configuration, $v=1$ m/s . . . . .	54
4.9. Message sequence chart for Non-Cooperative Relaying (NCR) behavior . . . . .	57
4.10. Message sequence chart of two-for-one cooperation with iACK . . . . .	58
4.11. Message sequence chart of two-for-one cooperation with fACK . . . . .	59
4.12. PER of (non-)cooperative relaying using ideal signaling . . . . .	61

4.13. Received rate of (non-)cooperative relaying using ideal signaling . . . . .	62
4.14. Architecture of <i>ath5k/mac80211</i> Linux WLAN driver . . . . .	63
4.15. Structure of a complete IEEE 802.11 MAC frame . . . . .	64
4.16. Symmetric diamond consisting of four notebooks with AR2414 cards . . .	67
4.17. Received rate vs. transmission mode for two-for-one cooperation . . . . .	68
5.1. Cooperative diversity gains without throughput penalty for broadcasts . . .	72
5.2. Optimistically constructing a CDS . . . . .	74
5.3. Extended graph model used by Wu et al. [95] . . . . .	78
5.4. Total Effective Contribution (TEC) of E-MCDS . . . . .	82
5.5. Broadcasting on an optimistically constructed CDS with fading . . . . .	84
5.6. Delivery ratios deteriorate with fading using static unit disk graphs . . . .	85
5.7. Approximating the instantaneous transmission ranges with fading . . . . .	85
5.8. Probabilistically constructing a CDS . . . . .	88
5.9. Constructing a PCDB set . . . . .	89
5.10. Delivery ratios improve using the probabilistic model with fading . . . . .	92
5.11. PCDB achieves the lowest latency with full delivery . . . . .	93
5.12. Denser broadcast sets become resistant against fading . . . . .	93
5.13. Evolution of backbone heuristics . . . . .	94
6.1. The frame compound describes a frame exchange between two users . . . .	99
6.2. Example of two non-cooperative patterns . . . . .	100
6.3. Example of a simple two-phase cooperative protocol . . . . .	102
6.4. Attributes of frame compounds used in abstract program trees . . . . .	104
6.5. Patterns of the simple two-phase cooperative protocol . . . . .	105
6.6. Abstract program tree for the simple two-phase cooperative protocol . . . .	105
6.7. Example of a parse tree for a code fragment . . . . .	112
6.8. Annotating trees with outage probabilities . . . . .	115
6.9. Annotated tree for the simple two-phase cooperative protocol . . . . .	116
6.10. Transformation of a linked pair of frame compounds . . . . .	118
6.11. Flow diagram of an MRC decider . . . . .	120
6.12. In-radio compilation for software radios . . . . .	121
A.1. Intersection of two circles with equal radius . . . . .	128
A.2. Intersection of three circles with equal radius . . . . .	129
B.1. Occurrence probability vs. average number of neighbors . . . . .	133
C.1. Polynomial time reduction example . . . . .	137

# List of Tables

- 2.1. Outage capacities of direct transmission and cooperative relaying . . . . . 15
- 3.1. Spatial consumption in the symmetric diamond configuration . . . . . 37
- 4.1. Outage capacities of symmetric, asymmetric, and chain CTRs . . . . . 46
- 4.2. Transmission modes of the IEEE 802.11a physical layer [39] . . . . . 49
- 5.1. Taxonomy of broadcast set construction algorithms . . . . . 76
- 5.2. Transmission rates with their maximum transmission ranges . . . . . 78
- 5.3. Summary of simulation parameters used for broadcasting . . . . . 91



# List of Abbreviations

<b>ACK</b>	Acknowledgment
<b>AF</b>	Amplify-and-Forward
<b>AMC</b>	Adaptive Modulation and Coding
<b>AODV</b>	Ad Hoc On-Demand Distance Vector
<b>AP</b>	Abstract Protocol
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>AWGN</b>	Additive White Gaussian Noise
<b>BER</b>	Bit Error Rate
<b>BPSK</b>	Binary Phase Shift Keying
<b>BSSID</b>	Basic Service Set Identifier
<b>CCA</b>	Clear Channel Assessment
<b>CDF</b>	Cumulative Distribution Function
<b>CDMA</b>	Code Division Multiple Access
<b>CDS</b>	Connected Dominating Set
<b>CoMP</b>	Coordinated Multi-Point
<b>CRC</b>	Cyclic Redundancy Check
<b>CSI</b>	Channel State Information
<b>CTR</b>	Cooperative Triangle
<b>CTS</b>	Clear-to-Send
<b>DCF</b>	Distributed Coordination Function
<b>DF</b>	Decode-and-Forward

- ECDS** Extended Connected Dominating Set
- EDS** Extended Dominating Set
- E-MCDS** Extended Minimum CDS
- fACK** faked ACK
- FEC** Forward Error Correction
- GK** Guha and Khuller's heuristic
- iACK** intermediate ACK
- IFS** Inter-Frame Space
- i.i.d.** independently and identically distributed
- LOS** Line of Sight
- LTL** Linear Temporal Logic
- MAC** Medium Access Control
- MCDS** Minimum Connected Dominated Set
- MISO** Multiple Input Single Output
- MPDL** MAC Pattern Description Language
- MRC** Maximal Ratio Combining
- NAV** Network Allocation Vector
- NCR** Non-Cooperative Relaying
- NLOS** Non-Line of Sight
- PCDB** Probabilistic Cooperation Diversity Broadcast
- PDF** Probability Density Function
- PER** Packet Error Rate
- PHY** Physical Layer
- PLCP** Physical Layer Convergence Procedure
- PSC** Packet Selection Combining

- RTS** Request-to-Send
- SBMRC** Soft-Bit Maximum Ratio Combining
- SDF** Selection Decode-and-Forward
- SDR** Software-Defined Radio
- SFD** Strong Full Diamond
- SIFS** Short Inter-Frame Space
- SNR** Signal-to-Noise Ratio
- SINR** Signal-to-Interference-plus-Noise Ratio
- SIR** Signal-to-Interference Ratio
- TDMA** Time Division Multiple Access
- TEC** Total Effective Contribution
- TO** Time-out
- WFD** Weak Full Diamond
- WLAN** Wireless Local Area Network



# 1. Introduction

Wireless transmissions are inherently error-prone due to the deteriorating nature of the wireless channel, in particular *path loss*, *shadowing*, and *fading*. These effects impair the propagation of the electromagnetic waves and thus the strength of the received signal. From these phenomena, *fading* – resulting from movements in a multi-path propagation environment – is the most challenging one to deal with [81]. With fading, the received signal strength fluctuates in time and frequency so that sudden and severe drops in signal strength occur frequently. These so-called *deep fades* make it impossible to reconstruct the transmitted message at the receiver.

Diversity techniques help to mitigate the detrimental effects of fading [84] and are widely used today. Diversity arises when the same signal arrives at the receiver via *independent* channels so that the signals are not correlated. For example, if the channel changes from one time slot to another, transmitting the same signal in both time slots exploits *temporal diversity*. If the signal, received during the first time slot, is too weak to be correctly decoded, the signal of the second time slot may be sufficient. Similarly, if the channel changes on different propagation paths, i.e., the paths are uncorrelated, transmitting the same signal on both paths exploits *spatial diversity*. These redundant but independent channel uses are called diversity branches. Advanced diversity techniques combine the signals from different diversity branches and can thus aggregate and restore the original information even if all transmissions are partly wrong. Such schemes are described in Section 2.2.

Multiple Input Single Output (MISO) systems employ multiple antennas at a single wireless device as shown in Figure 1.1(a); they already exploit spatial diversity and therefore achieve high capacity gains [24]. Unfortunately, MISO systems are infeasible if the wireless devices are too small to allow for multiple antennas per node (which nowadays only pertains to sensor motes) or if the diversity branches are still correlated due to insufficient spacing between the antennas. In this thesis, I therefore focus on *cooperative diversity* that arises when a neighboring node retransmits an overheard message. The retransmitting node is called a *relay* and techniques that achieve cooperative diversity by relaying are called *cooperative relaying* for short.

The idea of cooperative relaying dates back as early as 1971 when van der Meulen [89] introduced the *relay channel*. In the relay channel, a source broadcasts a message to both a destination and a relay as shown in Figure 1.1(b). The relay forwards the received message to the destination and hence the destination receives the same message via two independent channels. Cooperative relaying comprises both *sequential* forwarding (from source via relay to destination) as well as *multi-reception* at the destination (from source *and* relay).

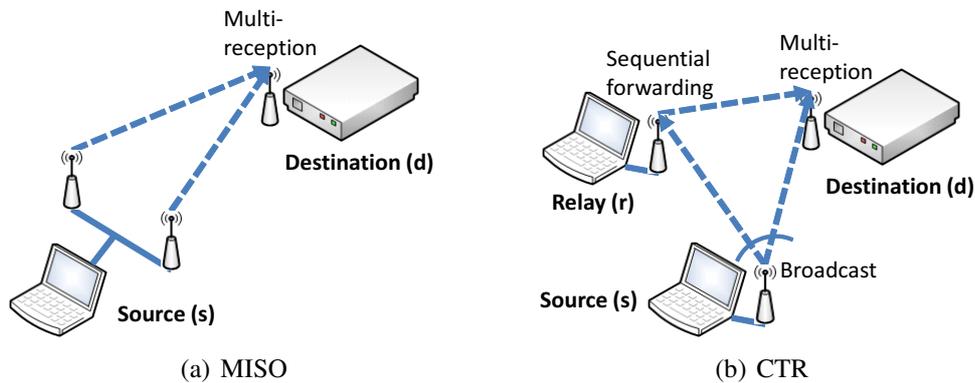


Figure 1.1.: Comparing a Multiple Input Single Output (MISO) system and Cooperative Triangle (CTR) with error-prone inter-user link

It is different from MISO in that the source-relay channel is error-prone. Additionally, with half-duplex constraints, the relay can only retransmit after it has overheard the complete message. This sequential forwarding is a major cost of cooperative relaying as it reduces the maximum achievable data rate by a factor of one half – the so-called *multiplexing loss*.

Still, the relay channel is theoretically interesting as it provides multiple gains at the same time, particularly power gains by the relay’s retransmission and diversity gains by exploiting independent channels. The seminal work of Cover and El Gamal [18] for the first time provided an *upper bound* on the capacity of the relay channel using information-theoretic arguments, but its exact capacity is still unknown today. However, when one simplifies the problem by neglecting the channel from source to destination, one can compute the capacity exactly. This simplified case is the *degraded* relay channel.

Since the late 1990’s the idea of cooperative communication lead to heavy research on the topic. Driven by the demands for higher data rate in cellular networks, Sendonaris et al. [76] first proposed how to exploit cooperative communication in Code Division Multiple Access (CDMA) networks. Laneman et al. [50] characterized fundamental relaying protocols in terms of their diversity order and outage probabilities. It has been convincingly argued that cooperative relaying, in theory, provides appealing new tradeoffs between rate and power that next-generation wireless networks could exploit to extend either coverage or data rate [68].

The appealing theoretical results were supported by practical experiments to find out how closely the theoretical gains can be reached in realistic scenarios. The experiments showed that practical issues such as limited and erroneous feedback (often assumed to be perfect in theoretical studies) and Medium Access Control (MAC) protocol overhead drastically decreased these gains [85]. However, in systems operating at low Signal-to-Noise Ratio (SNR), even practical implementations showed that the remaining gains were significant enough. Consequently, cooperative relaying can support next generation wireless systems [37, 85]. For example, next-generation cellular networks (4G) use it

to mitigate inter-cell interference. The future cellular standard LTE-Advanced may include cooperative communication for avoiding interference, known as Coordinated Multi-Point (CoMP) [75].

Wireless multi-hop networks pose new challenges that do not occur when only focusing on the three-node relay channel. An important example is the *relay blocking* problem [41]. To assist in a cooperative transmission, the relay needs to overhear the source's message, i.e., receive and correctly decode the message. But this is only possible by mitigating or avoiding interference at the relay. To avoid interference during this transmission, nodes close to the relay should not transmit simultaneously with the source. Consequently, the spatial reuse of a cooperative network decreases. All in all, cooperation consumes space. This effect cannot be studied in isolated cooperative networks where possible interferers are ignored.

Focusing on the three-node configuration is, even from a practical perspective, insufficient and might be misleading when generalizing to larger networks. On the one hand, costs that appear to be small in a three-node configuration increase considerably with size of the network and higher traffic load with multiple flows [96]. For example, if the source node selects the relay according to Channel State Information (CSI), all potential relays must signal their CSI to the source node. This signaling introduces additional overhead. With increasing number of neighboring nodes that could possibly act as relays, this overhead also increases. On the other hand, the interaction with routing in multi-hop wireless networks offer additional potential for optimization that does not exist in the three-node configuration [6]. While a simple configuration was necessary to develop a basic understanding, the journey must not stop there.

## 1.1. Thesis contributions

This thesis shows how cooperative relaying can be efficiently exploited in wireless multi-hop networks. Here, source and destination may be more than one hop apart and concurrent transmissions can take place due to spatial reuse. This is unlike most previous work on cooperative relaying that focuses on basic configurations consisting of source, destination, and cooperating relays.

For these networks, I characterize and compare cooperative relaying approaches in terms of their *spatial* consumption. Knowing the spatial consumption is essential for characterizing the efficiency of spatial reuse for large wireless networks. It also helps to narrow the list of possible cooperative relaying approaches to the spatially most efficient one. To increase data rate, I propose and analyze improvements that naturally arise in multi-hop networks both for unicast and broadcast transmissions. The techniques proposed in this thesis decrease multiplexing loss as the major cost known from the basic configuration. My cooperation protocol mitigates the multiplexing loss for unicast transmissions in multi-hop networks by exploiting information from the routing layer. The strength of my protocol is that it only requires read access to routing tables; it does not require modifi-

cations of the routing protocol itself. Thus, cooperative relaying can still be implemented at the physical layer and MAC sub-layer only. For broadcasting, multiplexing loss is not a penalty anymore, making broadcasting the ideal scenario for exploiting cooperation diversity. However, this is only true as long as an efficient subset of nodes retransmits the broadcast. To do so, static unit disk models as commonly used by broadcast set construction heuristics prevent them from achieving best performance, so I propose an alternative model and show that it is more suited for building broadcast sets that are robust in fading scenarios and deliver messages in short time.

I now summarize the contributions of this thesis and list the papers in which I first published them.

- **Derivation of the closed-form expected interference in random wireless networks with guard zones** (Section 3.1) – Interference limits the performance of wireless transmissions. To improve performance, nodes in the vicinity of a receiver must not transmit while it is receiving. To achieve this, MAC protocols can use the Request-to-Send (RTS) and Clear-to-Send (CTS) handshake. Analytically, a circular *guard zone* around the receiver models the area in which nodes must not transmit. While guard zones improve the performance of a *single* transmission, with increasing size they adversely affect spatial reuse and may compromise the performance of the *entire* network. To characterize spatial reuse of cooperative relaying, one first needs to capture interference analytically. For this, I derive an upper bound on the expected interference in random networks that employ guard zones around the receiver of a transmission [57].

📄 H. S. Lichte, S. Valentin, and H. Karl. Closed-form expected interference in wireless networks using a geometric path loss model. *IEEE Communications Letters*, 14(2):1–3, February 2010.

- **Integrating spatial consumption into outage capacity analysis** (Section 3.2) – To understand the impact that cooperative relaying has on the spatial reuse of a wireless network I determine the spatial consumption of cooperative and non-cooperative transmission at a fixed outage capacity [54]. Taking spatial consumption into account, using as many relays as possible is not necessarily desirable. It all depends on the required robustness and the target capacity at which the network should operate. My framework makes it possible to quantify the operating regions of cooperative and non-cooperative transmission, allowing engineers to choose the spatially most efficient approach for their network.

📄 H. S. Lichte, S. Valentin, H. Karl, I. Aad, and J. Widmer. Analyzing space/capacity tradeoffs of cooperative wireless networks using a probabilistic model of interference. In *Proc. 12th ACM Int. Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2009.

- **Analysis of SNR-based rate adaptation for cooperative relaying** (Section 4.1) – In non-cooperative networks, rate adaptation is an efficient mechanism to improve the network’s data rate by trading off data rate versus robustness [34]. As such, it is appealing to apply it to cooperative relaying. I systematically analyze the basic three-node configuration where three links have to be jointly considered instead of only one as in direct transmission. I propose an algorithm to choose suitable rates and I analyze the maximum possible gain. I find that, for practical systems with discrete rates to choose from, rate adaptation does not provide enough performance gains to mitigate the multiplexing loss of cooperative relaying [55].

📄 H. S. Lichte, S. Valentin, H. von Malm, H. Karl, A. Bin Sediq, and I. Aad. Rate-per-link adaptation in cooperative wireless networks with multi-rate combining. In *Proc. IEEE Int. Conf. Communications (ICC)*, June 2009.

- **Design of a routing-informed MAC protocol** (Section 4.2) – The major cost of cooperative relaying for unicast transmissions is its multiplexing loss due to the relay’s retransmission. I show that by exploiting routing information at the MAC sub-layer, a cooperative MAC protocol can reduce this multiplexing loss by avoiding retransmissions. This is possible in a multi-hop network by letting the relay support *two* point-to-point transmissions with a single retransmission – so-called *two-for-one* cooperation [52].

📄 H. S. Lichte, S. Valentin, H. Karl, I. Aad, L. Loyola, and J. Widmer. Design and evaluation of a routing-informed cooperative MAC protocol for ad hoc networks. In *Proc. 27th IEEE Conf. Computer Communications (INFOCOM)*, pages 1858–1866, April 2008.

- **Heuristic for constructing fast and reliable broadcast sets** (Chapter 5) – For broadcasts, cooperative relaying does not incur any multiplexing loss as retransmissions in the network happen anyway, making the use of cooperative diversity particularly appealing. By cleverly choosing only a subset of nodes for broadcasting cooperatively, this reduces the time to distribute the message in the entire network. I show that previous broadcast set construction heuristics, even those that already exploit cooperation diversity, severely suffer from fading, leading to packet delivery ratios well below 90 %. I identify the static unit disk model to be the source of the problem and suggest an alternative *probabilistic* model instead. With the probabilistic model, finding a subset of nodes as forwarders is an NP-complete problem. Therefore, I propose a heuristic that tries to find a set of nodes which achieve high delivery ratios and small latencies at the same time [56].

📄 H. S. Lichte, H. Frey, and H. Karl. Fading-resistant low-latency broadcasts in wireless multihop networks: The probabilistic cooperation diversity approach. In *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.

- **Design of a specification language for rapid prototyping of MAC protocols** (Chapter 6) – Since the process of implementing cooperative relaying at the MAC sub-layer is a time-consuming and error-prone task, I identify and exploit regularities between MAC frame sequences, so-called *patterns*, and the corresponding MAC protocol automaton to develop a code generator for such an automaton. By designing a suitable specification language, I illustrate how to simplify the process of implementing MAC protocols for analysis, simulations, and prototyping [51, 53].

📄 H. S. Lichte, S. Valentin, and H. Karl. Automated development of cooperative MAC protocols: A compiler-assisted approach. *Mobile Networks and Applications*, 15(6):769–785, 2009.

📄 H. S. Lichte and S. Valentin. Implementing MAC protocols for cooperative relaying: A compiler-assisted approach. In *Proc. 1st Int. Conf. Simulation Tools and Techniques for Communications, Networks and Systems (SIMU-Tools)*, March 2008. **Best paper award.**

## 2. Basics of cooperation and spatial reuse

*I introduce the information-theoretic models that form the basis for the analyses in this thesis. Based on these models, I derive metrics used for performance evaluation throughout this thesis. I briefly recapitulate fundamental results about cooperation diversity that allow the reader to understand the results of the following chapters in context.*

### 2.1. Channel model

Figure 2.1 shows the model used for wireless transmissions. The sending user  $i$  transmits the complex signal  $x_{ij}$  to the receiving user  $j$  over a wireless channel. The channel multiplies the signal with a complex channel coefficient  $h_{ij}$ , which captures path loss and fading and it adds white Gaussian noise  $n_i$ .

Path loss is the attenuation of the signal due to the distance between sender and receiver. Numerous deterministic models use the Euclidean distance  $s_{ij}$  between two nodes  $i$  and  $j$  to determine the *mean* SNR  $\bar{\gamma}_{ij}$ . The general empirical model uses path loss at a reference distance  $s_0$  [66]; for simplicity I assume that  $s_0 = 1$  and hence

$$\bar{\gamma}_{ij} = \frac{P_i}{N_0} \mathbb{E} [|h_{ij}|^2] = \frac{P_i s_{ij}^{-\alpha}}{N_0} \quad (2.1)$$

where  $P_i$  denotes the transmission power used by node  $i$  and  $\alpha$  denotes the path loss exponent. Typical values for the path loss exponent are between 2 and 5.  $N_0$  is the power spectral density of the band-passed noise at the receiver.

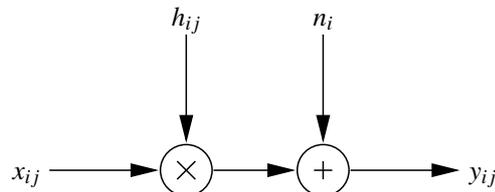


Figure 2.1.: System model of the wireless channel

## 2. Basics of cooperation and spatial reuse

---

As common in theoretical studies, I account for noise and average transmission power by a *reference* SNR  $\Gamma$  and express channel-related effects as scaling factors to this reference [85].  $\Gamma$  is given by

$$\Gamma = \frac{P}{N_0} \quad (2.2)$$

where I use a global transmission power  $P$  for all nodes in the network. Hence, the mean SNR  $\bar{\gamma}_{ij}$  for a transmission from node  $i$  received at node  $j$  can be written as

$$\bar{\gamma}_{ij} = \Gamma_{ij}\Gamma. \quad (2.3)$$

A propagation environment with a large number of equally distributed scatterers and Non-Line of Sight (NLOS) between sender and receiver subjects the signal to Rayleigh fading. This means that the magnitude of the channel coefficient  $|h_{ij}|$  is Rayleigh distributed and consequently the magnitude of the power coefficient  $|h_{ij}|^2$  is exponentially distributed. The parameter of the exponential distribution is given by the *mean* SNR as  $1/\overline{\text{gamma}}_{ij}$ . With this so-called Rayleigh-fading model, the *instantaneous* SNR  $\gamma_{ij}$  then follows the Probability Density Function (PDF) [78]

$$f_{\gamma_{ij}}(\gamma) = \frac{1}{\bar{\gamma}_{ij}} e^{-\gamma/\bar{\gamma}_{ij}}. \quad (2.4)$$

Rayleigh fading is the most severe form of fading; it typically occurs in a large scattering environment, e.g., indoors. It is the dominant model used in analytical studies due to the simplicity of the PDF.

The Signal-to-Interference-plus-Noise Ratio (SINR) is crucial to study wireless networks. I adopt the commonly used model [31] that defines the SINR for a receiver  $j$  by

$$\text{SINR} = \frac{|h_{ij}|^2 P_i s_{ij}^{-\alpha}}{N_0 + \sum_{k \neq i, j} |h_{kj}|^2 P_k s_{kj}^{-\alpha}}. \quad (2.5)$$

In dense networks with steady traffic, interference dominates noise. Thus, in these cases, noise is negligible. For example, at unit transmission power and considering only path loss, a sender must be at least  $1/s^\alpha = N_0 W \Leftrightarrow s = \sqrt[\alpha]{1/(N_0 W)}$  away such that the received power is equal or below thermal noise; at ambient temperature  $40^\circ\text{C}$ , a bandwidth  $W$  of 20 MHz, and path loss exponent  $\alpha = 3$  it follows that  $s = 22.6$  km. Practical networks have significantly shorter inter-node distances. Since this thesis focuses on wireless multi-hop networks, they are subject to interference due to concurrent transmissions that arise from spatial reuse. Consequently, I use Signal-to-Interference Ratio (SIR) for my analytical work instead of SNR. SIR is given by

$$\text{SIR} = \frac{|h_{ij}|^2 P_i s_{ij}^{-\alpha}}{\sum_{k \neq i, j} |h_{kj}|^2 P_k s_{kj}^{-\alpha}}. \quad (2.6)$$

Next I define the criteria for which a receiver can recover the sent data from a wireless transmission, i.e., the transmission is successful. For this, I use the popular  $E_b/N_0$ -threshold model [74]. In this model, the SNR *per bit*,  $E_b/N_0$ , must exceed a fixed threshold. This threshold depends on the modulation and coding scheme used. In this thesis, I modify the denominator to either account for noise and interference (SINR) or interference only (SIR).

### 2.1.1. Outage probability

The *channel capacity*  $C$  is the maximum number of bits per seconds per Hertz that a sender can transmit over an AWGN channel. If the data rate  $R$  at the transmitter (also called *spectral efficiency*) does not exceed this capacity, there exists a channel code that makes the probability of error arbitrarily small [84]. For data rates exceeding the channel capacity, no such code exists. Shannon showed that the capacity of an Additive White Gaussian Noise (AWGN) channel depends on the SNR at the receiver [77] according to

$$C(\gamma) = \log_2(1 + \gamma). \quad (2.7)$$

*Outage probability* is a common performance measure for slow fading channels and extensively used in theoretical analyses of wireless communication [38, 50, 84]. An outage event occurs when the wireless channel does not support the transmitter's data rate. The outage probability of this event is given by

$$P_{\text{Direct}}^{\text{out}} = \Pr\{C(\gamma) < R\} = \Pr\{\gamma < 2^R - 1\}. \quad (2.8)$$

Evaluating (2.8) using (2.4) yields the outage probability for a single connection with Rayleigh fading, given by

$$P_{\text{Direct}}^{\text{out}} = \int_0^{2^R-1} f_{\gamma_{ij}}(\gamma) d\gamma = 1 - e^{-(2^R-1)/\bar{\gamma}_{ij}} \approx \frac{1}{\Gamma_{ij}} \frac{2^R - 1}{\Gamma}. \quad (2.9)$$

The approximation is valid for high SNR only.

Theoretical studies typically assume *quasi-static* fading (also called *block fading*) to simplify analysis. Quasi-static fading models the wireless channel over time as a sequence of blocks. Each block has an equal duration. During such a block, the instantaneous SNR is constant, distributed according to (2.4), all blocks are independently and identically distributed (i.i.d.), and units of data under consideration (e.g., a packet) fit into a single block. Quasi-static fading can be used to model uncorrelated fading. As a simplification, uncorrelated fading is a suitable assumption for communication systems that apply interleaving to alleviate burst errors.

## 2.2. Cooperative diversity

Figure 1.1(b) shows the minimal configuration for cooperative diversity, namely the Cooperative Triangle (CTR). The source  $s$  transmits a message via a broadcast to both relay  $r$  and destination  $d$ . The relay  $r$  retransmits the message to the destination so that the destination  $d$  receives *both* messages from  $s$  and  $r$ , so-called *cooperative relaying*, via an orthogonal multiple access channel. For all analyses I assume a Time Division Multiple Access (TDMA) MAC protocol that enforces the orthogonality of the source's and the relay's transmissions in the temporal domain. TDMA is the natural choice as the relay needs to receive the source's message first before retransmitting it to the destination. Moreover, the popular IEEE 802.11 wireless network standard specifies a TDMA MAC to share the channel between multiple stations [39].

The relay, having received the source's message, needs to decide *when* and *how* to retransmit the message. Using *Amplify-and-Forward (AF)* the relay amplifies the signal received from the source before retransmitting it [50]. While AF is non-trivial to implement (due to buffering, amplifying, and retransmitting in the analog domain) and its performance at low SNR is inferior (due to amplification and retransmission of receiver noise), it is still suitable for *analyzing* cooperative relaying due to its simplicity [68].

A practicable alternative is *Decode-and-Forward (DF)* where the relay first demodulates the received signal from the source and then decodes it to obtain an estimate of the source's message. Then, the relay encodes the message again and modulates it before retransmitting. Re-encoding is regenerative and the relay may use the same codebook (repetition coding) or a different codebook for that. To avoid error propagation at the relay, it only retransmits the message if it has received it without errors, e.g., determined by Cyclic Redundancy Check (CRC). This so-called *Selection Decode-and-Forward (SDF)* achieves diversity gains exponentially in the number of relays [50].

### 2.2.1. Linear diversity combining

When two or more signals arrive at the destination carrying the same information, the destination can use a linear combining technique to improve the signal quality [12]. In general, if node  $j$  receives  $L$  signals, pertaining to the same data, from nodes in the set  $\mathbf{L}$ , a linear combiner operating at node  $j$  constructs the resulting signal  $y_{\mathbf{L}j}$  according to a weighted sum

$$y_{\mathbf{L}j} = \sum_{i \in \mathbf{L}} a_i y_{ij}, \quad (2.10)$$

where  $a_i$  is a combining coefficient to weight the signal from node  $i$ . The optimum combiner is a maximal ratio combiner that weights the signal proportional to its strength (requiring an equalizer to compensate for the phase shift in the channel beforehand). To do so, the receiver needs to know the channel factor  $h_{ij}$  (and the corresponding phase shifts) perfectly which, in practice, are only estimates at the receiver, e.g., determined using a preamble [74]. Thus, results derived using a model of Maximal Ratio Combining (MRC)

in analysis and simulations must be regarded as upper bounds on what could be achieved if MRC were implemented in a practical system.

When MRC is used, the instantaneous SNR of the combined signal is the sum of instantaneous SNRs of all individual signals, i.e.,

$$\gamma_{L,j} = \sum_{i \in L} \gamma_{ij}. \quad (2.11)$$

With Rayleigh fading, all  $\gamma_{ij}$  are exponentially distributed with mean SNRs  $\bar{\gamma}_{ij}$ . The PDF of the resulting random variable  $\gamma_{L,j}$  can be numerically computed using convolution, but even closed-form solutions for the Cumulative Distribution Function (CDF) exist [3, 21, 47] with varying complexity.

Valentin et al. [88] showed by simulations and measurements that for slow fading environments, e.g., indoor or urban scenarios, combining on packet level is as efficient as MRC. For such slow fading, the channel is unlikely to change during a single packet and hence there is no need in combining symbols with MRC. Instead, it suffices for the destination to choose simply the first correctly decoded packet, so-called Packet Selection Combining (PSC). PSC is advantageous in that it does not depend on channel estimation quality and can be implemented solely at the link layer [88].

### 2.2.2. Diversity order

The *diversity order* is the number of independent diversity branches, i.e., the number of channels having independent channel coefficients over which redundant information is transmitted, contributing to the reception of a signal. Figure 2.2 shows the four relay configurations studied in this thesis. This section discusses the diversity order that each of the configurations achieves based on the results in [86].

Non-Cooperative Relaying (NCR) is a per-hop relaying scheme where messages are received only via a single path. Figure 2.2(a) shows two-hop NCR, namely from source  $a$  via relay  $b$  to destination  $d$ . Both links  $a \rightarrow b$  and  $b \rightarrow d$  are direct and, thus, only achieve a diversity order of one. The outage probability of NCR is

$$P_{\text{NCR}}^{\text{out}} = \frac{\Gamma_{ab} + \Gamma_{bd}}{\Gamma_{ab}\Gamma_{bd}} \left( \frac{2^{2R} - 1}{\Gamma} \right). \quad (2.12)$$

Although NCR does not exploit cooperative diversity, it can still provide a benefit over the direct transmission  $a \rightarrow d$ . If a long link is split into two shorter links, the non-linearity of path loss can, even for a total power constraint, still offer power savings. In contrast, any additional hop requires another time slot for transmission, incurring a *multiplexing loss* proportional to the number of hops. Here, the multiplexing loss reduces data rate by a factor of 1/2.

If the destination  $d$  is also able to exploit transmissions received directly from the source  $a$ , node  $d$  can reach a diversity order of two. Figure 2.2(b) shows the Coopera-

## 2. Basics of cooperation and spatial reuse

---

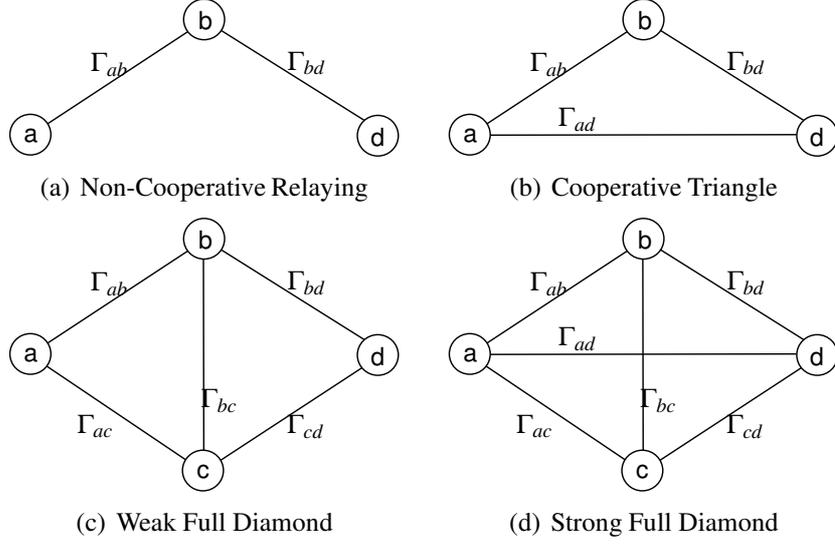


Figure 2.2.: The four relay configurations studied in this thesis with the channel-specific scaling factors to the reference SNR  $\Gamma$ .

tive Triangle (CTR), the simplest configuration for cooperative relaying. Looking at its outage probability,

$$P_{\text{CTR}}^{\text{out}} = \frac{\Gamma_{ab} + \Gamma_{bd}}{2\Gamma_{ab}\Gamma_{bd}\Gamma_{ad}} \left( \frac{2^{2R} - 1}{\Gamma} \right)^2, \quad (2.13)$$

the exponent 2 reflects the diversity order of two. Analytical studies often compare NCR and CTR to show the impact of cooperative diversity.

Adding another relay  $c$  augments the CTR to a diamond-like configuration. In this thesis, I consider two types of diamonds, namely the Weak Full Diamond (WFD) where the destination ignores whatever it receives from the source, and the Strong Full Diamond (SFD) where the destination tries to exploit whatever it receives from the source [86]. In both cases, the relays overhear each other. Diamonds are more costly due to their increased multiplexing loss of  $1/3$ .

In terms of diversity order, these two diamonds are different. Without using the direct channel from source  $s$  to destination  $d$  in the WFD, the destination only reaches a diversity order of two; refer to the exponent in WFD's outage probability equation

$$P_{\text{WFD}}^{\text{out}} = \frac{\Gamma_{ab}\Gamma_{ac} + \Gamma_{bd}\Gamma_{cd}}{2\Gamma_{ab}\Gamma_{ac}\Gamma_{bd}\Gamma_{cd}} \left( \frac{2^{3R} - 1}{\Gamma} \right)^2. \quad (2.14)$$

Hence, the WFD is not better than NCR in terms of diversity order (although it is better in terms of throughput efficiency as Section 4.2 will show).

Additionally exploiting the direct channel at  $d$  in the SFD improves the diversity order to three. Again, the exponent 3 reflects the diversity order in SFD's outage probability

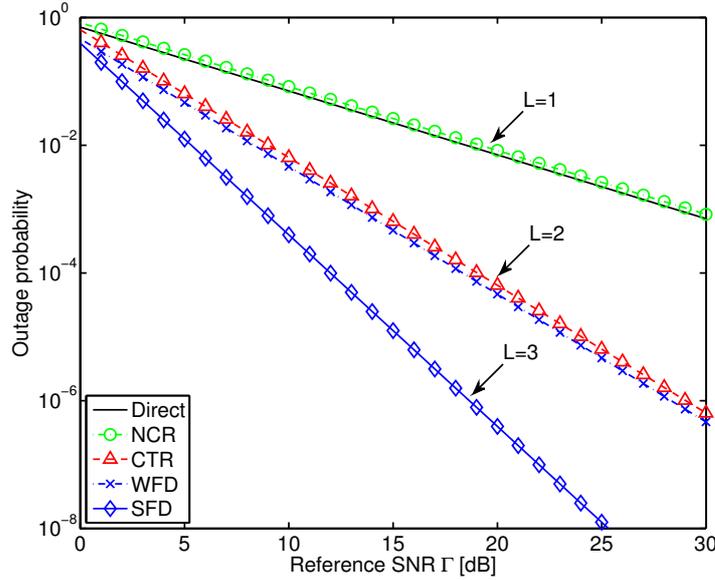


Figure 2.3.: Outage probability vs. reference SNR for  $R = 1/4$  bits/s/Hz, path loss exponent  $\alpha = 2.4$ , and nodes placed according to a symmetric diamond configuration.  $L$  denotes the diversity order.

equation

$$P_{\text{SFD}}^{\text{out}} = \frac{\Gamma_{ab}\Gamma_{ac} + \Gamma_{bd}\Gamma_{cd}}{6\Gamma_{ab}\Gamma_{ac}\Gamma_{bd}\Gamma_{cd}\Gamma_{ad}} \left( \frac{2^{3R} - 1}{\Gamma} \right)^3. \quad (2.15)$$

For arbitrary configurations consisting of any number of relays, the outage probability using cooperative relaying can be determined by cut set analysis, leading to the general equation [11]

$$P_{\text{any}}^{\text{out}} = \frac{1}{L!} \Theta \cdot \left( \frac{2^{KR} - 1}{\Gamma} \right)^L. \quad (2.16)$$

Here,  $K$  denotes the number of transmission phases (due to one source and  $K - 1$  relays retransmitting) and therefore quantifies the multiplexing loss;  $L$  denotes the diversity order. The diversity order depends on the employed channels that the configuration-dependent parameter  $\Theta$  describes.  $\Theta$  consists of the channel-related scaling factors, thereby reflecting the position of the nodes relative to each other. Boyer et al. [11] describe an approach based on cut set analysis to determine  $\Theta$  for arbitrary cooperative networks with  $K$  relays. This approach has been applied to derive the outage probability equations above. The interested reader finds the detailed derivation in [85].

To illustrate the substantial effect of the diversity order, Figure 2.3 shows the outage probabilities of the configurations in Figure 2.2 and direct transmission for different SNRs. For all approaches, outage probabilities decay with increasing SNR. The important fact

to observe is that outage probabilities decay exponentially in diversity order. Thus, from all configurations the SFD has the lowest outage probability. For decreasing error rates, diversity order has the highest effect. However, a higher diversity order always comes with high multiplexing loss and hence costs data rate. This fact is known as the diversity-multiplexing tradeoff [98]. As Chapter 3 will show in detail, the multiplexing loss also decreases the network's spatial reuse and, hence, needs to be mitigated to improve the network's overall data rate.

### 2.2.3. Outage capacity

Outage capacity is a common metric for describing slow fading channels [5, 84]. It denotes the spectral efficiency at which outage occurs with a probability of at most  $\varepsilon$ . It can be easily obtained by setting the corresponding outage probability equation  $P_{\text{out}} = \varepsilon$  and solving for  $R$ . The parameter  $\varepsilon$  is also referred to as the error rate bound. Unlike Shannon capacity, outage capacity takes into account the error rate bound as an important design criteria of wireless communication systems. Formally, it is defined as follows.

**Definition 1.** *Outage capacity  $C^{\text{out}}$  is the highest rate such that  $P^{\text{out}} \leq \varepsilon$ . The parameter  $\varepsilon$  describes the maximum outage probability that can be tolerated with  $0 \leq \varepsilon \leq 1$ .*

$$C^{\text{out}} := \sup_{P^{\text{out}} \leq \varepsilon} R \quad (2.17)$$

Table 2.1 summarizes the outage capacities of direct transmission and the relay configurations of Figure 2.2. One obtains these outage capacities by solving each of the corresponding outage probability equations for the spectral efficiency. For an arbitrary configuration with  $K$  relays using SDF, the general outage capacity is given by rearranging (2.16) as

$$C^{\text{out}} = \frac{1}{K} \log_2 \left( \Gamma \sqrt[L]{\frac{L! \varepsilon}{\Theta}} + 1 \right). \quad (2.18)$$

For simplicity, the reference SNR  $\Gamma$  has been factored out, leaving only path-loss-dependent factors for the involved channels in  $\Theta$ .

## 2.3. Characterizing spatial reuse

The performance of wireless multi-hop networks with spatial multiplexing is limited by interference. This limitation is not expressed by SNR but by SINR. Interference is a function of the network's geometry due to the distance-dependent path loss. *Stochastic geometry* emerged as a new tool in the analysis of large wireless networks where numerous spatial realizations of the network characterize its average performance [31]. Nodes are placed according to some probability distribution. Here, I focus on the important Poisson distribution and review the corresponding spatial Poisson point process in Section 2.3.1.

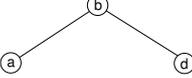
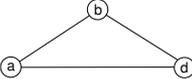
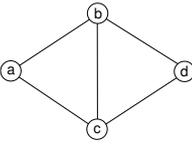
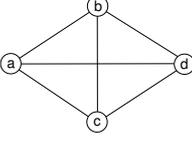
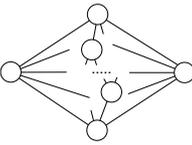
Approach	Configuration	Outage capacity
Direct		$\log_2 (\varepsilon \Gamma_{ad} \Gamma + 1)$
NCR		$\frac{1}{2} \log_2 \left( \varepsilon \left( \frac{1}{\Gamma_{ab}} + \frac{1}{\Gamma_{bd}} \right)^{-1} \Gamma + 1 \right)$
CTR		$\frac{1}{2} \log_2 \left( \sqrt{2\varepsilon \left( \frac{1}{\Gamma_{ad}} \left( \frac{1}{\Gamma_{ab}} + \frac{1}{\Gamma_{bd}} \right) \right)^{-1} \Gamma + 1} \right)$
WFD		$\frac{1}{3} \log_2 \left( \sqrt{2\varepsilon \left( \frac{1}{\Gamma_{ab}\Gamma_{ac}} + \frac{1}{\Gamma_{bd}\Gamma_{cd}} \right)^{-1} \Gamma + 1} \right)$
SFD		$\frac{1}{3} \log_2 \left( \sqrt[3]{6\varepsilon \left( \frac{1}{\Gamma_{ad}} \left( \frac{1}{\Gamma_{ab}\Gamma_{ac}} + \frac{1}{\Gamma_{bd}\Gamma_{cd}} \right) \right)^{-1} \Gamma + 1} \right)$
Arbitrary		$\frac{1}{K} \log_2 \left( \sqrt[L]{\frac{L! \varepsilon}{\Theta} \Gamma + 1} \right)$

Table 2.1.: Outage capacities of direct transmission and the four cooperative relaying strategies studied in this thesis as well as the arbitrary configuration; obtained by solving the corresponding outage probability equations for the spectral efficiency.

Guard zones around receivers control the interference in a wireless network. The size of the guard zone impacts the capacity of both the point-to-point transmission as well as the entire network. Section 2.3.2 introduces the concept of guard zones and discusses how protocols can practically implement it.

### 2.3.1. Poisson point process

The behavior of nodes in a large wireless ad hoc network is often impossible to specify deterministically. If nodes in a wireless network independently join and leave the network, a suitable model for the distribution of its nodes is the *Poisson point process* whose characteristic feature is stochastic independence [48]. The Poisson point process is commonly used in analytical studies of large wireless networks [20, 23, 72, 73] not only for its suitability but also for its simplicity. Figure 2.4 exemplarily shows a realization of a homogeneous Poisson process with node density  $\lambda = 1/2$  in a square area of  $A = 8^2$ . In a

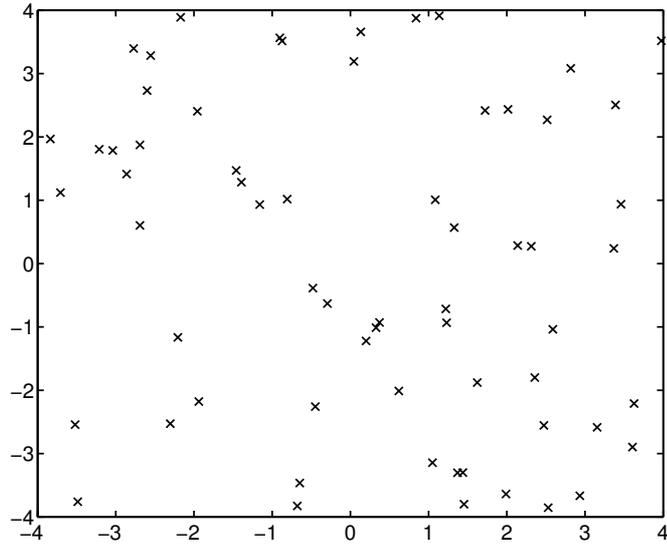


Figure 2.4.: Poisson process with node density  $\lambda = 1/2$  in a square area of  $8 \times 8$ .

Poisson process, the probability to find exactly  $k$  nodes in an area  $A$  is given by

$$\mathbb{P}\{k \text{ nodes in area } A\} = \frac{(\lambda A)^k}{k!} e^{-\lambda A}. \quad (2.19)$$

To place nodes according to a Poisson point process, one first has to determine the number of nodes in the particular area  $A$  using (2.19). Second, one distributes the nodes independently over the area by drawing  $(x, y)$ -coordinates from the uniform distribution.

### 2.3.2. Spatial consumption by guard zones

Receivers in wireless networks can employ guard zones in which interferers cannot be present to control the received SINR [90]. Hasan and Andrews [32] showed that in CDMA networks guard zones exist that maximize network capacity. For the link  $a \rightarrow d$ , Figure 2.5 shows the destination's guard zone with radius  $g$ . Nodes 1 and 2 are inside the guard zone and, hence, inhibited to transmit, while the remaining nodes 3 and 4 are allowed to transmit as they reside outside the guard zone. If nodes 3 and 4 were indeed transmitting simultaneously (which is a worst-case assumption), they would interfere at  $d$  and decrease the SINR for the transmission from source to destination.

For direct transmission (Figure 2.5) it suffices to express spatial consumption simply as the area of the guard zone around the destination in units of  $\text{m}^2$  as only one transmission takes place. In general, a transmission approach can comprise more than one phase with different senders and receivers. For example, CTR consists of two phases where relay

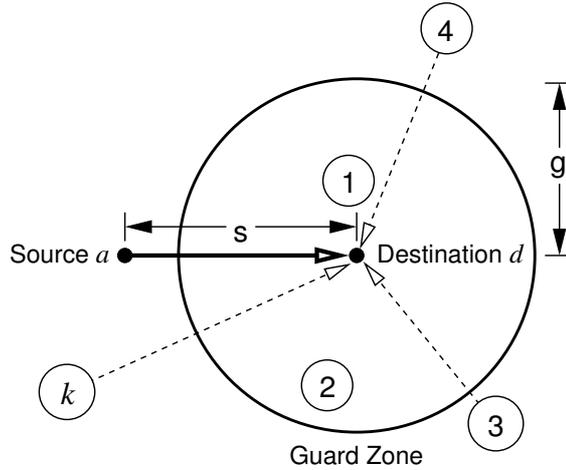


Figure 2.5.: A transmission between source and destination, which are a distance of  $s$  apart, is protected by a circular guard zone with radius  $g$  around the destination. All nodes outside the guard zone transmit simultaneously and hence create interference at the destination.

and destination span a combined guard zone during the first phase and only the destination spans a guard zone during the second phase. In this case, simply considering the area of the spanned guard zones does not suffice anymore as it ignores the temporal aspect, namely the duration of a phase during which the nodes span their guard zones.

Therefore, I extend the spatial consumption such that it takes the time into account during which receivers span their guard zones. The spatial consumption that I refer to in the remainder of this thesis is always defined as follows (where I use  $\zeta$  instead of  $A$  to clearly distinguish it from mere area).

**Definition 2.** *The spatial consumption  $\zeta_X$  of a transmission approach  $X$  with  $n$  transmission phases is*

$$\zeta_X = \frac{1}{n} \sum_{i=1}^n A_{X,i}$$

where  $A_{X,i}$  denotes the area of the guard zones spanned by receiving nodes during transmission phase  $i$  of  $X$ . The unit of  $\zeta_X$  is  $m^2s$ .

This definition assumes that all transmission phases have the same duration, which is the case for all transmission approaches I consider in this thesis, but it easily generalizes to transmission phases with arbitrary durations.

Practically, a MAC protocol can realize guard zones by announcing medium reservations. For example, the popular IEEE 802.11 MAC announces the duration of a complete transmission cycle in every MAC frame [39]. Nodes that are not involved in the transmission will, by overhearing these frames, update an internal timer called the Network Allocation Vector (NAV). This vector indicates the time that the medium is expected to

## 2. Basics of cooperation and spatial reuse

---

be busy. Nodes will refrain from transmitting if they find the vector to be non-zero. This mechanism is called *virtual carrier sensing* and complements the physical Clear Channel Assessment (CCA) procedure.

Using virtual carrier sensing, the transmission power used for sending frames with announcements determines the size of the guard zone. Using a higher transmission power, the frames can, on average, be received over a larger distance. Alternatively, the data rate for sending the frames also impacts the size of the guard zone as higher rates require a higher SINR at the receiver for successful decoding.

The size of a guard zone affects the throughput of nodes in the network since nodes inside the guard zone do not transmit. Therefore, the guard zone dictates the amount of spatial reuse in the network. With large guard zones, interference at a single receiver is mitigated and the received data rate increases. At the same time, fewer simultaneous transmissions can take place in the vicinity of the receiver spanning the guard zone, such that the overall throughput of the network decreases. Thus, the guard zone is an essential parameter to control the tradeoff between the network's throughput and the link-wise throughput. In the following chapter, I will develop a framework for analyzing this important tradeoff to shed light on the spatial reuse of cooperative relaying.

### 3. Spatial consumption of cooperative relaying

*Cooperative relaying requires at least one additional node for retransmission. Although the resulting cooperative diversity gains can improve a single transmission in the network, it is still unknown how it affects the entire network's capacity. The relay's transmission blocks the nodes in its vicinity, thus reducing spatial reuse. I integrate the spatial consumption of cooperative relaying into outage capacity analysis and show analytically that, in relevant scenarios, cooperative relaying is spatially more efficient than non-cooperative transmissions.*

In a real network deployment, nodes transmitting in the vicinity of receiving nodes adversely affect the SINR. For direct communication, this only pertains to nodes around the destination, which should abstain from sending while the destination receives. In this sense, communication not only consumes time and power, but also space by spanning a guard zone around the destination as Section 2.3.2 explained. Cooperative relaying requires more space, since in this case, also nodes around the relay have to be silent. Otherwise the relay cannot receive the message and thus cannot assist the source; no or only limited performance gains would be achievable. A first analysis of this problem by Marchenko et al. [60] in two simple toy configurations shows that cooperative relaying, although beneficial for a single transmission, decreases the total network performance. However, their results may be misleading. First, performance depends on how the error rates on the involved links behave with fading which the authors have not included in their analysis. Second, they deliberately constructed specific configurations where the problem occurs and it remains unclear how often these configurations occur in realistic wireless networks. In this chapter, I tackle these two points since they directly affect performance of the entire network.

In a subsequent study, Marchenko et al. [61] suggest that a cooperative MAC protocol should select a relay that blocks the minimum number of additional nodes in its neighborhood. This allows for more concurrent transmissions, possibly benefiting the overall network capacity. The authors compare different selection schemes in terms of probability of successful contention and number of nodes blocked by the selected relay. Although the authors evaluate the selection schemes for random wireless networks, they do not quantify the spatial consumption that their selection schemes achieve. Thus, it still remains unclear to which extent relay selection schemes, even if they reduce the number of blocked nodes, can benefit the network's capacity. Nevertheless, their research shows that spatial reuse of

### 3. Spatial consumption of cooperative relaying

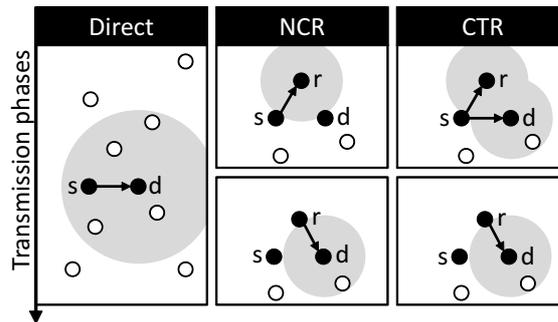


Figure 3.1.: Structure of guard zones for three different protocol classes. Shaded areas denote guard zones around receivers. Dark circles indicate transmitting interferers, light circles show nodes hindered from transmitting.

cooperative relaying is crucial for a network's performance and needs further study. I will now develop a framework for characterizing this spatial consumption analytically.

Figure 3.1 illustrates guard zones for direct transmission, NCR, and CTR. Each approach has its individual guard zone that affects spatial reuse. For example, CTR achieves higher diversity order than the other approaches, but it also consumes more space in the first phase as more neighboring nodes must not transmit. While the capacity for a particular link increases, the number of concurrent transmissions may decrease, thereby decreasing the network's overall capacity. However, the cooperation diversity gains may be partially transformed into smaller guard zones to improve the capacity of the network at the cost of the capacity of point-to-point transmissions.

This important tradeoff between outage capacity and guard zone for cooperative diversity schemes has not been characterized so far. Currently, a framework is missing that allows to normalize outage capacity by the area consumed by spanning guard zones. To develop such a framework, one needs to relate the interference in a wireless network to the size of the guard zone. To avoid specific assumptions on the network's topology, assume that interferers are randomly located in the network and independently from each other.

This aggregate interference is usually considered harmful as it severely limits throughput in large wireless networks [29]. In order to achieve better capacity scaling, the best approach is to eliminate interference altogether. Eliminating interference is possible by joint transmissions in cellular systems [43, 44]. This so-called CoMP works by letting base stations cooperate and transmit simultaneously. Using CSI feedback from the terminals, each cooperating base station computes the waveforms to transmit such that only one signal arrives at the intended terminal. This is only possible because a reliable backhaul (e.g., optical fibre) inter-connects base stations, thereby allowing them to reliably exchange CSI and data, and to synchronize their transmissions. In contrast, the scenarios of interest in this thesis are purely wireless, so the source-to-relay link is error-prone and does neither allow reliable feedback nor sufficiently tight synchronization as needed for CoMP.

Özgür et al. [70] proposed another theoretical approach for eliminating interference, which avoids interference by design: by establishing cooperative clusters of nodes that transmit exclusively one after the other in the entire network, these transmissions are not subject to interference. All nodes in a cluster transmit simultaneously to another cluster using spatial multiplexing, thereby achieving a sum-rate scaling proportional in the number of nodes. To distribute the message within a cluster, the same scheme is applied recursively within that particular cluster. Since transmissions are subject to path loss, intra-cluster communication exploits spatial reuse. Using TDMA, clusters concurrently distribute the message, before they successively send it to the destination clusters. At the destination clusters, the bits received during cooperative transmissions must now be distributed to all nodes, again using 9-TDMA for spatial reuse. Recursively applying this approach leads to a *linear* scaling of the network's total capacity (which further improves the scaling shown by Gupta and Kumar [29] for interference-limited networks and the slightly improved bound by Dousse et al. [20]). However, such a technique is only theoretically interesting due to its inherently centralized nature and overhead.

Instead of eliminating interference, one can try to only mitigate it and this is where the concept of guard zones comes in handy. I capture the concept of guard zones analytically and relate it to the interference in wireless networks in Section 3.1. This enables to characterize the expected interference depending on the size of the guard zone. I then use this result in Section 3.2 to derive a new analytical framework that links outage capacity and spatial consumption. In Section 3.3 I show how to apply the framework to shed light on how diversity order impacts this tradeoff. Finally, I provide an outlook on the utility of this new framework in Section 3.4.

### **3.1. Expected interference in random wireless networks with guard zones**

Characterizing the interference in wireless networks has always been of utmost importance to better design such networks. The problem can be approached in two different ways: First, placing nodes on some regular structure simplifies the problem at hand. For example, an approximation exists for CDMA systems [64] where interfering nodes reside on concentric circles around the receiver (so-called inner and outer tier boundaries). While an assumption of regularity can greatly simplify the math, the solution only describes a specific scenario.

Alternatively, the second approach is to use *random* networks as a realistic model for practical wireless networks [42] and then to use probabilistic arguments to describe the interference. This approach avoids any specific structure and instead captures numerous spatial incarnations of the network, thereby allowing to derive more general results about *average* cases. Node locations in a wireless network are then described by a Poisson point process (Section 2.3.1).

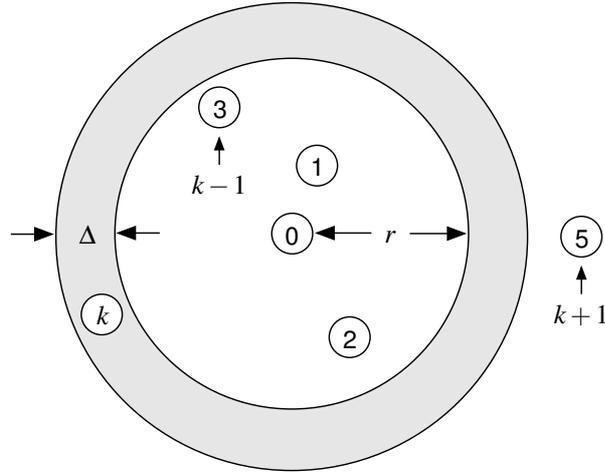


Figure 3.2.: Capturing the  $k$ -nearest neighbor of node 0 in an annulus with radius  $r$  and thickness  $\Delta$  (shown for  $k = 4$ ).

Mathar and Mattfeldt [63] use a Poisson point process for placing interfering nodes. In their model, nodes up to certain radius  $M$  have Line of Sight (LOS) to the receiver. As a consequence, they use the Rician PDF to model the interference power up to a distance of  $M$ , from there on they use the Rayleigh PDF. Additionally, they also model shadowing effects by letting the transmitter power be lognormal distributed. Due to the model's complexity they cannot present a closed-form solution to the resulting integral equation so far and only proceed numerically.

To the best of my knowledge, all related work on characterizing the interference in random wireless networks ignores the fact that receivers span guard zones to mitigate interference. Hence, I provide an alternative derivation for the interference distribution that takes the circular guard zone around the receiver into account. With this distribution, the *average* interference in a random wireless network can be characterized.

### 3.1.1. Distribution of $k$ -nearest neighbor's distance

Since the interference depends on the geometry of the wireless network, it is important to characterize the node's distances. I now derive the distribution for a node's distance to its  $k$ -nearest neighbor for a random wireless network in the plane. Haenggi [30] provides a similar derivation for an arbitrary number of dimensions.

Let  $D_k$  denote the random variable representing a node's distance to its  $k$ -nearest neighbor. Figure 3.2 exemplarily shows a random network where the fourth neighbor ( $k = 4$ ) is inside an annulus with radius  $r$  and thickness  $\Delta$ . The probability to find the  $k$ -nearest neighbor in this annulus is  $P(r \leq D_k < r + \Delta)$ . For  $r \leq D_k < r + \Delta$  to hold, there must be exactly one node in the annulus with inner radius  $r$  and outer radius  $r + \Delta$  (for  $\Delta \rightarrow 0$  one can ignore the  $k + 1, k + 2, \dots$  neighbors) and there must be exactly  $k - 1$  nodes in

the circle with radius  $r$ . Both are independent events and their probabilities follow directly from the spatial Poisson process (2.19). Hence,

$$\mathbb{P} \left\{ 1 \text{ node in area } \pi (r + \Delta)^2 - \pi r^2 \right\} = \lambda \pi (2r\Delta + \Delta^2) e^{-\lambda \pi (2r\Delta + \Delta^2)} \quad (3.1)$$

and

$$\mathbb{P} \{k - 1 \text{ nodes in circle with radius } r\} = \frac{(\lambda \pi r^2)^{k-1}}{(k-1)!} e^{-\lambda \pi r^2}. \quad (3.2)$$

The probability that exactly one node falls inside the annulus (3.1) *and* exactly  $k - 1$  nodes fall inside its interior (3.2) evaluates to

$$\mathbb{P} \{r \leq D_k < r + \Delta\} = \lambda \pi (2r\Delta + \Delta^2) e^{-\lambda \pi (2r\Delta + \Delta^2)} \frac{(\lambda \pi r^2)^{k-1}}{(k-1)!} e^{-\lambda \pi r^2}. \quad (3.3)$$

To obtain the PDF of  $D_k$ , divide (3.3) by  $\Delta$  and let  $\Delta \rightarrow 0$ , leading to

$$\begin{aligned} f_{D_k}(r) &= \lim_{\Delta \rightarrow 0} P(r \leq D_k < r + \Delta) / \Delta \\ &= \lim_{\Delta \rightarrow 0} \frac{(\lambda \pi r^2)^k e^{-\lambda \pi (r+\Delta)^2} (2r + \Delta)}{r^2 (k-1)!} = 2 \frac{(\lambda \pi r^2)^k}{(k-1)! r} e^{-\lambda \pi r^2}. \end{aligned} \quad (3.4)$$

With this PDF, one can now proceed to characterize the power received from the  $k$ th neighbor as it depends on the distance according to a power law as in (2.1).

### 3.1.2. Interference from $k$ th neighbor

The random variable  $I'_k = P/D_k^\alpha$  denotes the interference power received from the  $k$ th neighbor in absence of any guard zone (hence the prime), assuming only distance-dependent path loss. I use the letter  $I$  to indicate that this power is regarded as interference. I now look at the probability that the power  $I'_k$  takes on values within a small interval of size  $\delta$ , starting at  $x$ .

$$\begin{aligned} \mathbb{P} \{x \leq I'_k < x + \delta\} &= \mathbb{P} \{x \leq P/D_k^\alpha < x + \delta\} = \\ &= \mathbb{P} \left\{ \sqrt[\alpha]{P/(x + \delta)} \leq D_k < \sqrt[\alpha]{P/x} \right\} = \int_{\sqrt[\alpha]{P/(x+\delta)}}^{\sqrt[\alpha]{P/x}} f_{D_k}(r) dr \end{aligned} \quad (3.5)$$

### 3. Spatial consumption of cooperative relaying

---

Note that  $\sqrt[\alpha]{P/(x+\delta)} < \sqrt[\alpha]{P/x}$  for  $\delta > 0$ . Dividing (3.5) by  $\delta$  and letting  $\delta \rightarrow 0$  results in the density of the  $k$ th neighbor's interfering power  $I'_k$  in absence of any guard zone as

$$\begin{aligned} f_{I'_k}(x) &= \lim_{\delta \rightarrow 0} \frac{2}{\delta} \int_{\sqrt[\alpha]{P/(x+\delta)}}^{\sqrt[\alpha]{P/x}} \frac{(\lambda \pi r^2)^k}{r^{(k-1)!}} e^{-\lambda \pi r^2} dr \\ &= 2 \frac{\left( \lambda \pi \sqrt[\alpha]{(P/x)^2} \right)^k}{\alpha x (k-1)!} e^{-\lambda \pi \sqrt[\alpha]{(P/x)^2}} = \frac{2\Psi(x)^k e^{-\Psi(x)}}{\alpha x (k-1)!} \end{aligned} \quad (3.6)$$

where I define  $\Psi(x) := \lambda \pi \sqrt[\alpha]{(P/x)^2}$  for better readability.

The random variable  $I'_k$  does not take any guard zone into account. With a guard zone of radius  $g$  around the receiver, the random variable  $I_k$  describes the interference assuming a suitable MAC protocol, where

$$I_k = \begin{cases} I'_k & \text{if } D_k > g \\ 0 & \text{if } D_k \leq g \end{cases} . \quad (3.7)$$

#### 3.1.3. Aggregate expected interference for $\alpha > 2$

Determining the *total* interference power received at the origin requires to aggregate the expected power from all neighbors. First look at the  $k$ th neighbor and ask for the expected power. With  $\alpha > 2$ , this expected interference power using a guard zone at the origin can be written as

$$\mathbb{E}[I_k] = \underbrace{\int_0^\eta x f_{I_k}(x) dx}_{\text{Outside guard zone}} + \underbrace{\int_\eta^\infty x f_{I_k}(x) dx}_{\text{Inside guard zone}} . \quad (3.8)$$

For the special case  $\alpha = 2$  the expected interference is infinite as already noted by Haenggi [30]. A maximum interference of  $\eta$  occurs on the border of the guard zone. Larger interference is not possible since, according to the path loss depending on the node's distance, the node had to transmit *inside* the guard zone (i.e.,  $x > \eta \Leftrightarrow r < g$ ). Thus, even though nodes may be inside the guard zone due to the Poisson assumption, they do not contribute interference power in this case, allowing to simplify (3.8) to

$$\mathbb{E}[I_k] = \int_0^\eta \frac{2\Psi(x)^k e^{-\Psi(x)}}{\alpha (k-1)!} dx + \int_\eta^\infty 0 dx = \int_0^\eta \frac{2\Psi(x)^k e^{-\Psi(x)}}{\alpha (k-1)!} dx. \quad (3.9)$$

For the aggregate expected interference, one needs to compute the limit of the infinite series obtained by adding (3.9) for all  $k \geq 1$ . Since the integrand in (3.9) is continuous on  $[0, \eta]$ , swapping integration and summation is possible. Factoring out the exponential

function as well as  $\Psi(x)$  results in

$$\begin{aligned}\mathbb{E}[I] &= \sum_{k=1}^{\infty} \mathbb{E}[I_k] = \sum_{k=1}^{\infty} \int_{x=0}^{\eta} \frac{2\Psi(x)^k e^{-\Psi(x)}}{\alpha(k-1)!} dx \\ &= \frac{2}{\alpha} \int_{x=0}^{\eta} \frac{\Psi(x)}{e^{\Psi(x)}} \sum_{k=1}^{\infty} \frac{\Psi(x)^{k-1}}{(k-1)!} dx = \frac{2}{\alpha} \int_{x=0}^{\eta} \frac{\Psi(x)}{e^{\Psi(x)}} \sum_{k=0}^{\infty} \frac{\Psi(x)^k}{k!} dx.\end{aligned}\quad (3.10)$$

The infinite sum in (3.10) is an exponential series that is identical to the exponential function already occurring in the denominator of the integrand, which hence cancel out. Substituting the definition of  $\Psi(x)$ , (3.10) simplifies to

$$\mathbb{E}[I] = \frac{2\lambda\pi}{\alpha} \int_{x=0}^{\eta} \sqrt[\alpha]{(P/x)^2} dx = \frac{2\lambda\pi\eta}{\alpha-2} \left(\frac{P}{\eta}\right)^{2/\alpha}.\quad (3.11)$$

Substituting  $\eta$  in (3.11) yields the aggregate expected interference

$$\mathbb{E}[I] = \frac{2\lambda\pi P}{(\alpha-2)g^{\alpha-2}}\quad (3.12)$$

which only depends on the path loss exponent  $\alpha > 2$ , node density  $\lambda$ , transmission power per node  $P$ , and guard radius  $g$ .

### 3.1.4. Verification by simulation

I now verify the derivation of (3.12) by simulation. For this, the simulator places nodes on a circular playground with radius  $R = 1,000$  m according to a spatial Poisson process with node density  $\lambda$ , as shown exemplarily in Figure 3.3. The receiver resides at the origin and spans a guard zone. Then, for every node outside the guard zone, the simulator computes the power received at the origin and, finally, takes the sum of all these powers to obtain the aggregate interference.

To assess the quality of my solution when nodes are no longer placed according to the analytical model, I also determine the mean interference for a clustered scenario, as exemplarily shown in Figure 3.4. Here, nodes are randomly placed in  $n$  clusters with radius  $R_c < R$  and the clusters' centers are randomly placed in a circle of radius  $R - R_c$ . In both scenarios, the maximum distance that a node can have to the origin is  $R$ .

For fair comparison, I require the mean number of nodes to be equal in both scenarios, leading to a different node density  $\lambda_c > \lambda$  for nodes in the clusters. I require that

$$\lambda\pi R^2 = n\lambda_c\pi R_c^2 \Rightarrow \lambda_c = \frac{\lambda}{n} \left(\frac{R}{R_c}\right)^2,\quad (3.13)$$

i.e., the average number of nodes in Figure 3.3 should be roughly equal to the average

### 3. Spatial consumption of cooperative relaying

---

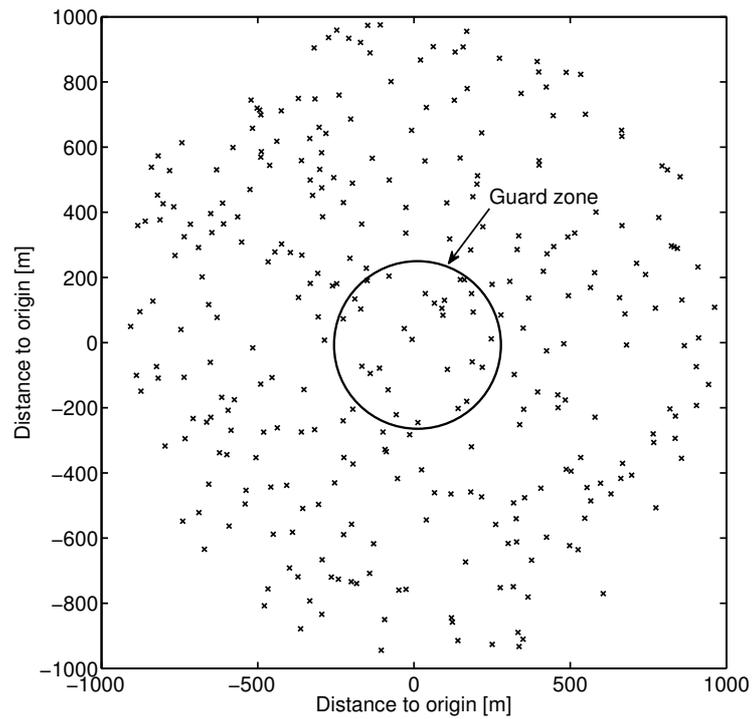


Figure 3.3.: Nodes are placed according to a spatial Poisson process with density  $\lambda$  in a circle of radius  $R$ , reflecting the analytical model. The simulator varies the radius  $g$  of the guard zone at the origin.

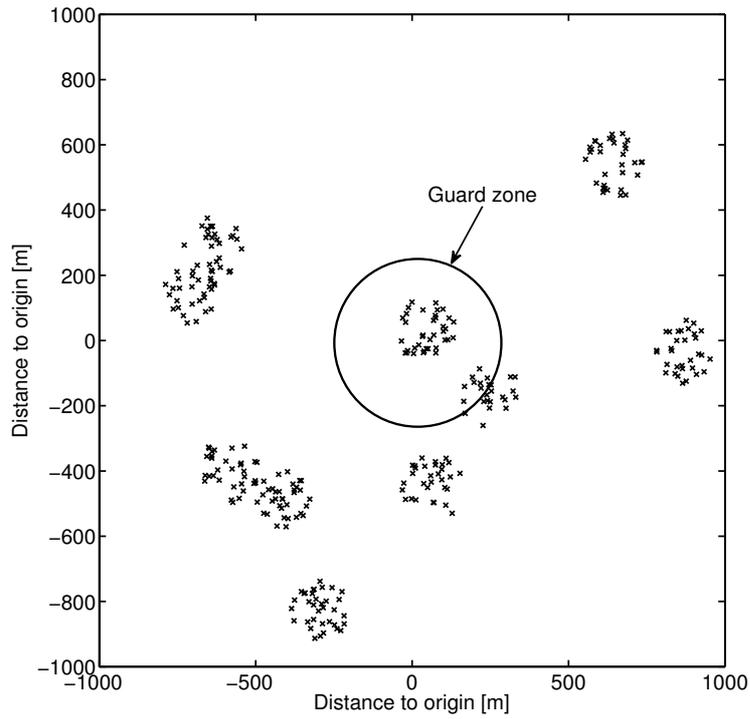


Figure 3.4.: Nodes are clustered in circles of radius  $R_c < R$  and placed according to a spatial Poisson process with density  $\lambda_c$ , violating the analytical model. The simulator varies the radius  $g$  of the guard zone at the origin.

### 3. Spatial consumption of cooperative relaying

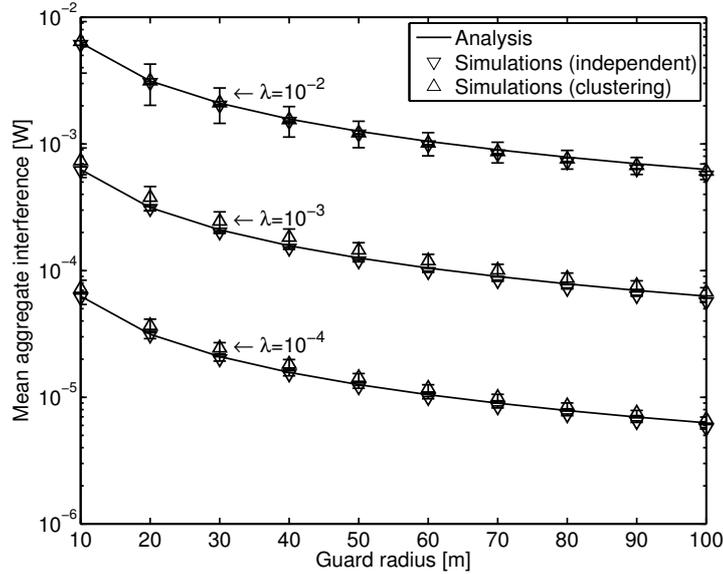


Figure 3.5.: Simulations indicate the correctness of (3.12) for three node densities  $\lambda$ . The plot shows guard radius vs. expected interference (analysis) and mean aggregated interference (simulations) for a path loss exponent of  $\alpha = 3$ , playground radius  $R = 1,000$  m, and unit transmission power. Confidence intervals shown for 99 % confidence level.

number of nodes in Figure 3.4 for a fixed number of clusters  $n$ . In every run, the simulator randomly places  $n = 10$  clusters of radius  $R_c = 100$  m and then it places the nodes within the cluster according to a spatial Poisson process with density  $\lambda_c$  given by (3.13). I consider three different node densities, namely  $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ , and for each node density the simulator does not perform more than 2,000 simulation runs (confidence intervals are shown for 99 % confidence).

Figure 3.5 shows the mean aggregate interference for both independent and clustered scenarios, as well as the corresponding analytical results using (3.12), for varying guard radius  $g$ . The simulation results closely match the analytical results, albeit a slight offset exists in both scenarios. For the independent scenario in Figure 3.3, the simulation results always stay below the analytical results. This can be explained by recalling that (3.12) has been derived assuming an *infinite* number of interfering nodes, whereas the simulator uses a *finite* area resulting in only a finite number of interfering nodes which altogether contribute less power. For the clustered scenario in Figure 3.4, the simulated average interference slightly exceeds the analytical one if node densities are small. But in both cases, the difference is small enough for (3.12) to be a good approximation.

Figure 3.6 shows the corresponding standard deviation of the aggregate interference for both independent and clustered scenarios, assuming a low node density of  $\lambda = 10^{-4}$ . Even

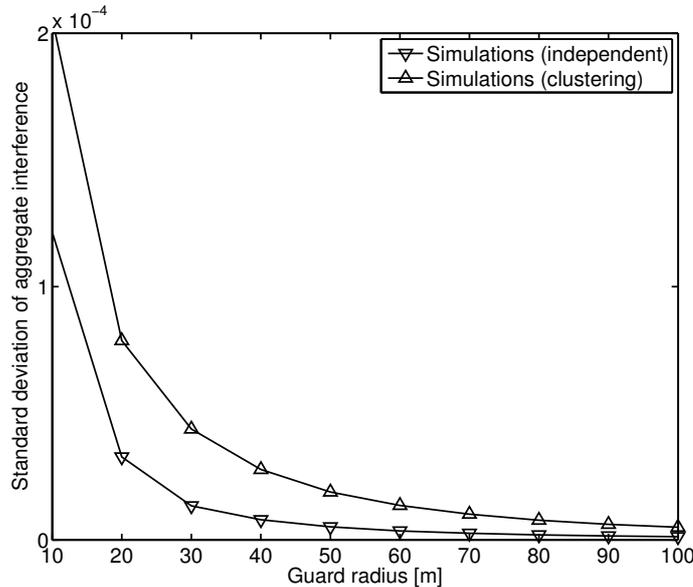


Figure 3.6.: Standard deviation of expected interference for the independent and clustered scenarios assuming a low node density of  $\lambda = 10^{-4}$ , path loss exponent of  $\alpha = 3$ , playground radius  $R = 1,000$  m, and unit transmission power.

though both scenarios achieve the same mean interference over a large number of network realizations, the clustered scenario shows a larger standard deviation. This indicates that the derived closed-form approximation is accurate for predicting the expected interference for a large set of network realizations, but for *particular* networks, its approximation is best only if all nodes are independently distributed from each other without clustering. However, for larger guard radii, the quality of the approximation improves.

I conclude that the analytical result in (3.12) establishes an upper bound on the expected interference in a wireless network. Simulations show that this bound can be approached closely when the guard radius is small compared to the area on which nodes are placed.

### 3.2. Linking spatial consumption with outage capacity

With the expected interference as a function of guard radius at hand, I now use it to link the spatial consumption (Section 2.3.2) of various transmission approaches (Table 2.1) with their outage capacities. To achieve this, I show in detail how to derive the outage capacities of the transmission approaches in Table 2.1 as a function of the guard zone. For the considered approaches, I derive their spatial consumption using simple geometric arguments depending on how the nodes span their guard zones and how these guard zones overlap during a transmission cycle. I then reformulate, for every approach, the

### 3. Spatial consumption of cooperative relaying

---

corresponding outage capacity equations summarized in Table 2.1 so that they match the same node configuration for fair comparison. The guard zone allows to relate the outage capacity of a transmission approach with its spatial consumption. One can then compare the approaches in terms of their spatial consumption at equal outage capacity by setting the appropriate guard radii per approach (Section 3.3).

I base the following discussion on the *symmetric* diamond where all nodes, except for source  $a$  and destination  $d$ , are a distance of  $s$  apart. This configuration is a pessimistic case since all links but the direct  $a \rightarrow d$  link have the same length. In this case, none of the links is favored. Assuming that all nodes use unit transmission power, only path loss determines the received *average* signal power.

$$\mathbb{E}[S] = \frac{1}{s^\alpha} \quad (3.14)$$

To express the *average* interference power, I use my result (3.12), again with unit transmission power.

$$\mathbb{E}[I] = \frac{2\lambda\pi}{(\alpha-2)g^{\alpha-2}} \quad (3.15)$$

Together, (3.14) and (3.15) give the SIR.

$$\frac{\mathbb{E}[S]}{\mathbb{E}[I]} = \frac{(\alpha-2)g^{\alpha-2}}{2\lambda\pi s^\alpha} \quad (3.16)$$

For convenience, I factor out everything that is not related to the sender-receiver distance  $s$  and use it as *reference SIR*  $\Gamma$ .

$$\Gamma := \frac{1}{\mathbb{E}[I]} = \frac{(\alpha-2)g^{\alpha-2}}{2\lambda\pi} \quad (3.17)$$

Using the reference SIR, (3.16) simplifies to

$$\frac{\mathbb{E}[S]}{\mathbb{E}[I]} = \frac{\Gamma}{s^\alpha}. \quad (3.18)$$

Due to the diamond's symmetry it holds for the mean SIRs that

$$\bar{\gamma}_{ab} = \bar{\gamma}_{ac} = \bar{\gamma}_{bc} = \bar{\gamma}_{bd} = \bar{\gamma}_{cd} = \frac{\Gamma}{s^\alpha}. \quad (3.19)$$

Only the distance between  $a$  and  $d$  is, for geometrical reasons,  $s\sqrt{3}$ , thus

$$\bar{\gamma}_{ad} = \frac{\Gamma}{(s\sqrt{3})^\alpha}. \quad (3.20)$$

I now integrate the receiver's guard zone into the outage capacity equations, thereby averaging outage capacity over numerous spatial realizations of the randomly distributed interferers with density  $\lambda$  per unit area. The guard zone's size will then determine the outage capacity.

The general outage capacity for the symmetric diamond builds upon (2.18) where I factor out the path loss  $s^\alpha$  as well as the SIR  $\Gamma$ . The parameter  $\omega$  gathers the remaining factors, hence leading to the general form

$$C^{\text{out}}(g, s, \varepsilon, \alpha, \lambda) = \frac{1}{K} \log_2 \left( \sqrt[L]{\frac{L! \varepsilon \Gamma(g, \alpha, \lambda)}{\omega}} \frac{1}{s^\alpha} + 1 \right). \quad (3.21)$$

The parameters  $K$ ,  $L$ , and  $\omega$  are specific to the transmission approach and fixed.

The essential feature now is that the outage capacity  $C^{\text{out}}$  is a function of  $g$ . For every transmission approach, one can also compute the spatial consumption using the guard radius  $g$ . Thus, parameter  $g$  links both capacity  $C^{\text{out}}(g, \dots)$  and spatial consumption  $\zeta(g)$  and allows to relate the two. Rearranging (3.21) for the guard radius  $g$  gives the necessary guard radius to reach a target capacity  $C^{\text{out}}$ .

$$g = \sqrt[\alpha-2]{\sqrt[L]{\frac{\omega}{L! \varepsilon}} \frac{2\lambda \pi s^\alpha}{\alpha-2} (2^{C^{\text{out}}K} - 1)} \quad (3.22)$$

In the following sections, I determine  $\omega$  for every transmission approach in the symmetric diamond configuration and I derive the necessary equations to compute the corresponding spatial consumption.

Further, I assume for every transmission approach the following global constraints to assure a fair comparison of the approaches. Each transmission cycle lasts unit time. With NCR and CTR, source and relay transmit which splits a cycle into two transmission phases each of duration  $1/2$ . As two nodes transmit at equal power  $P$  but each node employs only  $1/2$  time, the injected energy is always  $P$  independent of the number of transmitters (naturally, this generalizes to a higher number of relays, e.g., two relays for SFD). Thus, I compare direct transmission and all relaying protocols at equal energy. Without loss of generality, the following derivations always assume that all nodes use the same, unit transmission power.

### 3.2.1. Direct transmission

The outage capacity for direct transmission is given by

$$C_{\text{Dir}}^{\text{out}} = \log_2 (\varepsilon \Gamma_{ad} \Gamma + 1) = \log_2 \left( \frac{\varepsilon}{\sqrt{3}^\alpha} \frac{\Gamma}{s^\alpha} + 1 \right), \quad (3.23)$$

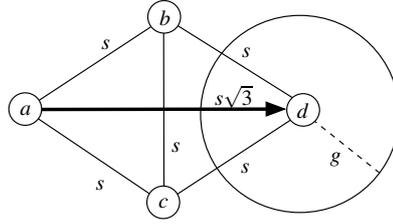


Figure 3.7.: Direct transmission from source  $a$  to destination  $d$

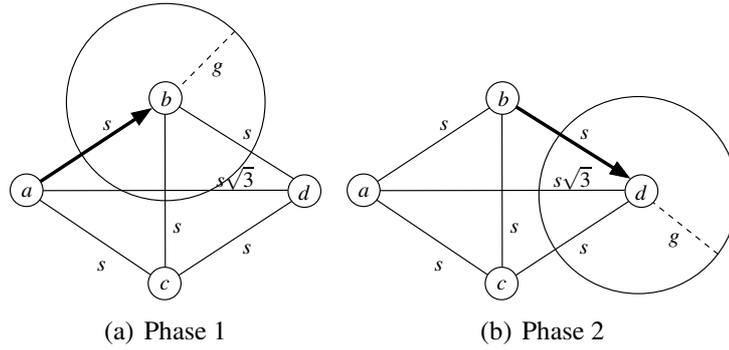


Figure 3.8.: Non-Cooperative Relaying (NCR) from source  $a$  via relay  $b$  (or  $c$ , interchangeably) to destination  $d$

thus  $K = L = 1$  and  $\omega = \sqrt{3}^\alpha$ . Figure 3.7 shows that only the destination  $d$  spans a single guard zone with radius  $g$ ; the potential relays  $b$  and  $c$  are not involved. The consumption is trivially given by the area of a circle around the destination for the entire unit transmission cycle,

$$\zeta_{\text{Dir}} = \pi g^2. \quad (3.24)$$

### 3.2.2. Non-Cooperative Relaying (NCR)

The outage capacity for NCR is given by

$$C_{\text{NCR}}^{\text{out}} = \frac{1}{2} \log_2 \left( \varepsilon \left( \frac{1}{\Gamma_{ab}} + \frac{1}{\Gamma_{bd}} \right)^{-1} \Gamma + 1 \right) = \frac{1}{2} \log_2 \left( \frac{\varepsilon \Gamma}{2 s^\alpha} + 1 \right), \quad (3.25)$$

thus  $K = 2$ ,  $L = 1$ , and  $\omega = 2$ . The transmission is split into two phases, namely from source  $a$  to relay  $b$  (or, interchangeably,  $c$ ) as shown in Figure 3.8(a), and from  $b$  to the destination  $d$  as shown in Figure 3.8(b). The parameter  $K$  captures this multiplexing loss of two. When computing the consumption, the guard zones for the first and second phase each hold for half the transmission cycle of unit time. Since both guard radii are equal,

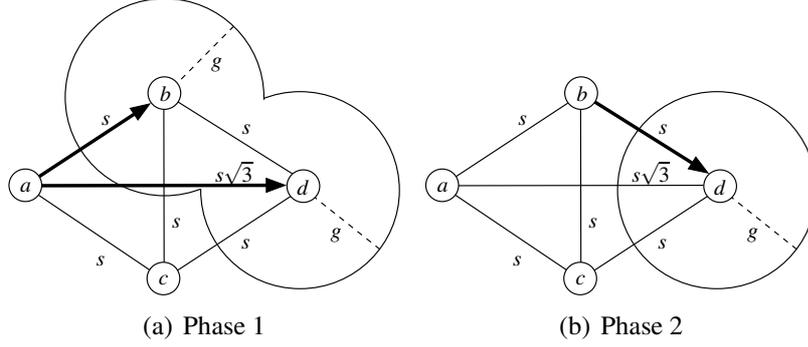


Figure 3.9.: Cooperative Triangle (CTR) rooted at source  $a$  via relay  $b$  (or  $c$ , interchangeably) to destination  $d$

one arrives at the same consumption as for direct transmission, namely

$$\zeta_{\text{NCR}} = \frac{1}{2} \left( \underbrace{\pi g^2}_{\text{Phase 1}} + \underbrace{\pi g^2}_{\text{Phase 2}} \right) = \pi g^2. \quad (3.26)$$

### 3.2.3. Cooperative Triangle (CTR)

Like NCR, the transmission is split into two phases. In the first phase, the source  $a$  transmits to relay  $b$  (or, interchangeably,  $c$ ) but, unlike NCR, the destination  $d$  also tries to receive the transmission in the first phase. Thus, not only does the relay span a guard zone, but the destination as well as shown in Figure 3.9(a).

The derivation of the average interference  $\mathbb{E}[I]$  in [57] assumes circular guard zones. Figure 3.9(a) shows that the first phase of CTR violates this assumption. I deal with this problem by approximation. For the SIRs  $\Gamma_{ab}$  and  $\Gamma_{ad}$ , I simply ignore the guard zone around the destination  $d$  and the relay  $b$ , respectively. Then, (3.16) still applies, but I *overestimate* the interference. When used to compute outage capacities, this will result in a lower capacity. As a consequence, all capacities in this section are *lower bounds*. The second phase shown in Figure 3.9(b) is identical to that of NCR.

This lower bound for the outage capacity of CTR is given by

$$C_{\text{CTR}}^{\text{out}} = \frac{1}{2} \log_2 \left( \sqrt{2\varepsilon \left( \frac{1}{\Gamma_{ad}} \left( \frac{1}{\Gamma_{ab}} + \frac{1}{\Gamma_{bd}} \right) \right)^{-1} \Gamma + 1} \right) = \frac{1}{2} \log_2 \left( \sqrt{\frac{\varepsilon}{\sqrt{3}^\alpha} \frac{\Gamma}{s^\alpha} + 1} \right), \quad (3.27)$$

thus  $K = 2$ ,  $L = 2$ , and  $\omega = 2\sqrt{3}^\alpha$ . The parameter  $L$  captures the diversity order of the transmission approach. Since  $d$  receives *two* independent copies of the same packet, CTR achieves  $L = 2$ .

Although the above outage capacity derivation underestimates the capacity, I do com-

### 3. Spatial consumption of cooperative relaying

---

pute the consumption *exactly*. For this, I need to determine the area of two overlapping circles in the first phase. I briefly review the required geometry in Appendix A.1, and I only present the resulting equation here. For the first phase it holds that

$$A_{\text{CTR},1} = 2\pi g^2 - A_{\text{IS}} = 2\pi g^2 - 2g^2 \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} + \frac{s}{2} \sqrt{4g^2 - s^2}, \quad (3.28)$$

while the second phase is identical to the second phase of NCR, leading to

$$\zeta_{\text{CTR}} = \frac{1}{2} (A_{\text{CTR},1} + \pi g^2) = g^2 \left( \frac{3}{2} \pi - \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \right) + \frac{s}{4} \sqrt{4g^2 - s^2}. \quad (3.29)$$

#### 3.2.4. Weak Full Diamond (WFD)

The outage capacity of the Weak Full Diamond (WFD) is given by

$$C_{\text{WFD}}^{\text{out}} = \frac{1}{3} \log_2 \left( \sqrt{2\varepsilon \left( \frac{1}{\Gamma_{ab}\Gamma_{ac}} + \frac{1}{\Gamma_{bd}\Gamma_{cd}} \right)^{-1} \Gamma + 1} \right) = \frac{1}{3} \log_2 \left( \sqrt{\varepsilon \frac{\Gamma}{s^\alpha} + 1} \right), \quad (3.30)$$

thus  $K = 3$ ,  $L = 3$ , and  $\omega = 2$ . Here, I again overestimate the interference by assuming circular guard zones with radius  $g$  around the nodes in phase 1 and phase 2. Since  $d$  only receives two independent copies of the same packet, WFD achieves  $L = 2$ .

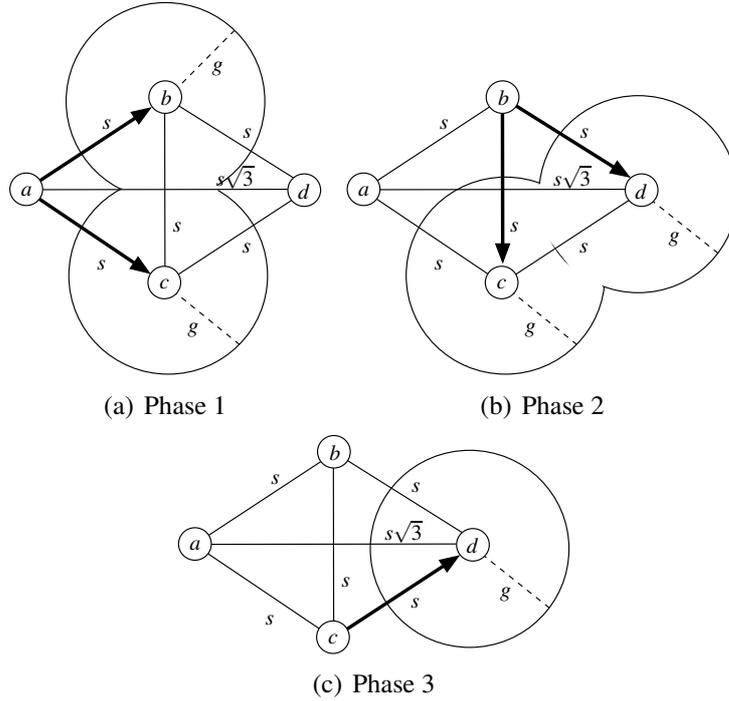
Figure 3.10 illustrates all three transmission phases of WFD. In the first phase, the source  $s$  broadcasts its message to both relays  $b$  and  $c$ . Hence, a combined guard zone of two overlapping circles arises. Due to symmetry, this combined guard zone is identical to the guard zone during the first phase of CTR. In the second phase, relay  $b$  sends the message to both  $c$  and  $d$ , again using the same combined guard zone. In the final third phase, only the destination spans a circular guard zone. Taking together, the consumption for WFD is given as

$$\zeta_{\text{WFD}} = \frac{1}{3} (2A_{\text{CTR},1} + \pi g^2) = g^2 \left( \frac{5}{3} \pi - \frac{4}{3} \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \right) + \frac{s}{3} \sqrt{4g^2 - s^2}. \quad (3.31)$$

#### 3.2.5. Strong Full Diamond (SFD)

The outage capacity of the Strong Full Diamond (SFD) is given by

$$C_{\text{SFD}}^{\text{out}} = \frac{1}{3} \log_2 \left( \sqrt[3]{6\varepsilon \left( \frac{1}{\Gamma_{ad}} \left( \frac{1}{\Gamma_{bd}\Gamma_{cd}} + \frac{1}{\Gamma_{ab}\Gamma_{ac}} \right) \right)^{-1} \Gamma + 1} \right) = \frac{1}{3} \log_2 \left( \sqrt[3]{\frac{3\varepsilon}{\sqrt{3}^\alpha} \frac{\Gamma}{s^\alpha} + 1} \right), \quad (3.32)$$


 Figure 3.10.: Weak Full Diamond (WFD) rooted at source  $a$  via relays  $b, c$  to destination  $d$ 

thus  $K = 3, L = 3$ , and  $\omega = 2\sqrt{3}^\alpha$ . Here, I again overestimate the interference by assuming circular guard zones with radius  $g$  around the nodes in phase 1 and phase 2. Since  $d$  receives *three* independent copies of the same packet, SFD achieves  $L = 3$ .

Figure 3.11 illustrates all three transmission phases of SFD. In the first phase, the source  $s$  broadcasts its message, and the destination  $d$  as well as both relays  $b$  and  $c$  try to receive it. Hence, a combined guard zone of three overlapping circles arises as shown in 3.11(b). Since the center points of all three guard zones are exactly  $s$  apart, their intersection forms an equilateral circular triangle whose area  $A_{\text{ECT}}$  is given by a closed-form expression (refer to Eq. 23 in [22]) which I briefly review in the Appendix A.2. Then, the combined area of the three circles as shown in Figure 3.11(b) is given by

$$A_{\text{SFD},1} = 3\pi g^2 - 3A_{\text{IS}} + A_{\text{ECT}}. \quad (3.33)$$

Adding the area of three circles exceeds their combined area and hence subtracting the intersection  $A_{\text{IS}}$  of two circles two times compensates for this excess. For illustration, refer to Figure 3.11(a). If subtracted a third time, the area of the equilateral circular triangle is missing. Adding it back again gives the combined area in (3.33). The second and third phase of SFD resemble the first and second phase of CTR, respectively. Thus, using

$$A_{\text{SFD},2} = 2\pi g^2 - A_{\text{IS}} \text{ and } A_{\text{SFD},3} = \pi g^2 \quad (3.34)$$

### 3. Spatial consumption of cooperative relaying

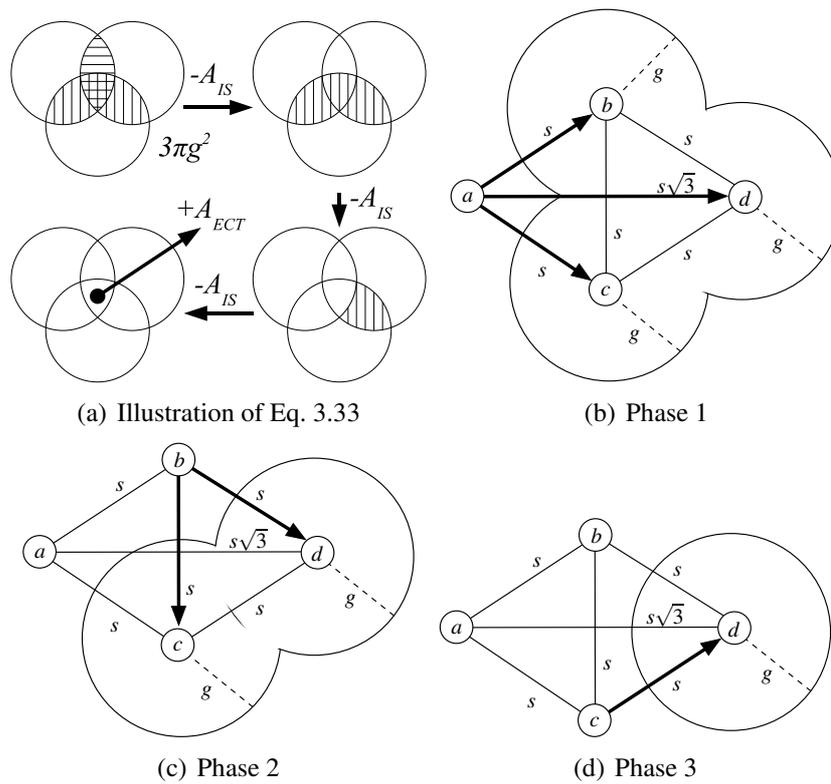


Figure 3.11.: Strong Full Diamond (SFD) rooted at source  $a$  via relay  $b$  and  $c$  to destination  $d$ ; shows how to compute the area of three overlapping circles

### 3.3. The effect of diversity order on spatial consumption

Table 3.1.: Number of phases  $K$ , diversity order  $L$  at destination  $d$ , geometric parameter  $\omega$ , and spatial consumption of the transmission approaches in the symmetric diamond configuration.

	$K$	$L$ (at $d$ )	$\omega$	Spatial consumption $\zeta$
Direct	1	1	$\sqrt{3}^\alpha$	$\pi g^2$
NCR	2	1	2	$\pi g^2$
Cooperative Triangle	2	2	$2\sqrt{3}^\alpha$	$g^2 \left( \frac{3}{2}\pi - \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \right) + \frac{s}{4} \sqrt{4g^2 - s^2}$
Weak Full Diamond	3	2	2	$g^2 \left( \frac{5}{3}\pi - \frac{4}{3} \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \right) + \frac{s}{3} \sqrt{4g^2 - s^2}$
Strong Full Diamond	3	3	$2\sqrt{3}^\alpha$	$g^2 \left( 2\pi - \frac{8}{3} \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} + \sin^{-1} \frac{c}{2g} \right) +$ $\frac{c^2}{4\sqrt{3}} - \frac{c}{4} \sqrt{4g^2 - c^2} - \frac{2s}{3} \sqrt{4g^2 - s^2}$ where $c^2 := 3g^2 - \frac{s^2}{2} - s\sqrt{3g^2 - \frac{3s^2}{4}}$

one obtains for the area occupied during an entire SFD transmission cycle

$$\zeta_{\text{SFD}} = \frac{1}{3} (6\pi g^2 - 4A_{\text{IS}} + A_{\text{ECT}}). \quad (3.35)$$

Refer to Table 3.1 for its full expression.

### 3.3. The effect of diversity order on spatial consumption

This section compares the spatial consumption of the transmission approaches in Table 2.1 at a desired outage capacity  $C^{\text{out}}$  and robustness  $\varepsilon$ . For fair comparison at a desired outage capacity  $C^{\text{out}}$ , all approaches set their guard radii  $g$  such that they achieve the outage capacity  $C^{\text{out}}$ . I do not vary any other parameter. These individual guard radii are then used in the corresponding equations for the spatial consumption, allowing me to directly relate the target capacity with the consumed area. I do this by inserting the appropriate  $K$ ,  $L$ , and  $\omega$  into (3.22), leading to the guard radius that achieves capacity  $C^{\text{out}}$ . One can now express spatial consumption as a function of  $C^{\text{out}}$  by inserting (3.22) into the corresponding equation of the spatial consumption listed in Table 3.1.

I visualize the newly established function  $A(C^{\text{out}})$  to compare the different transmission approaches in terms of their spatial consumption. I choose a path loss exponent of  $\alpha = 3$  to

### 3. Spatial consumption of cooperative relaying

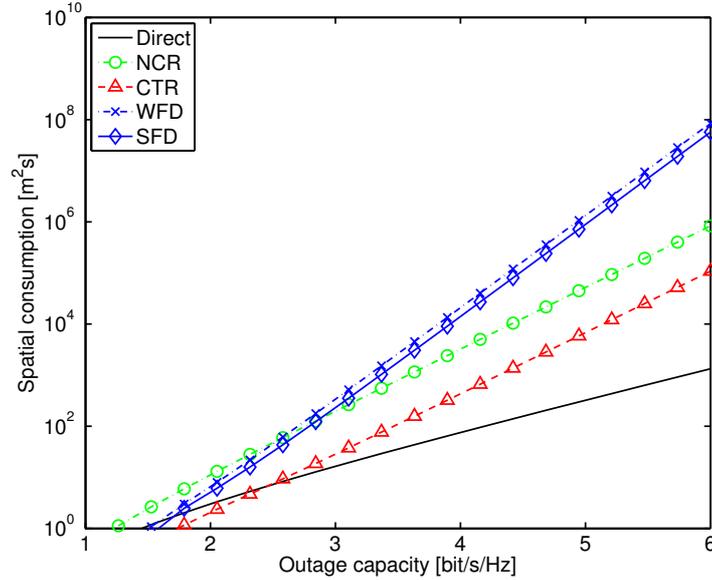


Figure 3.12.: Spatial consumption with optimal guard zones for different demanded outage capacities at low robustness at  $\varepsilon = 10^{-1}$  (max. 10 % outage) and path loss exponent  $\alpha = 3$ . A node density of  $\lambda = 10^{-3}$  has been chosen such that nodes experience a mean SIR of 32 dB (nodes transmit with unit transmission power). Note the logarithmic scale of the y-axis.

reflect a propagation environment with numerous obstacles, resulting in multi-path propagation. Fading arises due to movement of objects within the propagation environment or due to movement of the wireless nodes themselves. Note that the outage capacity approximations from Table 2.1 are only valid for high SNRs. Thus, I choose the node density such that the mean SIR is fixed at 32 dB (occurring at  $\lambda = 10^{-3}$  nodes per unit area) with all nodes operating at unit transmission power. Figure 3.12 and Figure 3.13 show the results for two characteristic outage requirements. On the one hand,  $\varepsilon = 10^{-1}$  represents a typical Wireless Local Area Network (WLAN) scenario with low robustness and, on the other hand,  $\varepsilon = 10^{-3}$  represents high robustness where retransmissions are to be avoided, e.g., for any kind of real-time traffic such as voice or video.

If the required robustness is low (Figure 3.12), then the spatial consumption of CTR and SFD is only less than direct transmission for low target capacities, i.e., beyond a target capacity of 2.5 bit/s/Hz, direct transmission has the least spatial consumption. In this case, the diversity gain is not needed and the spatial cost associated with it does not pay off. Since all approaches tolerate a higher amount of interference due to the large value of  $\varepsilon$ , direct transmission profits the most since it only spans a single guard zone that can be made small.

The situation reverses for high robustness as shown in Figure 3.13. To mitigate in-

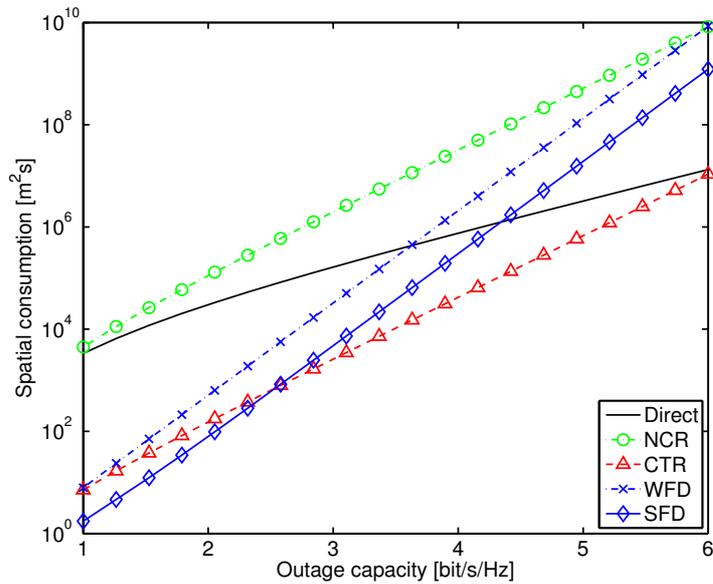


Figure 3.13.: Spatial consumption with optimal guard zones for different demanded outage capacities at high robustness at  $\varepsilon = 10^{-3}$  (max. 0.1 % outage) and path loss exponent  $\alpha = 3$ . A node density of  $\lambda = 10^{-3}$  has been chosen such that nodes experience a mean SIR of 32 dB (nodes transmit with unit transmission power). Note the logarithmic scale of the y-axis.

terference, the guard radius must increase. Since direct transmission and NCR do not exploit cooperative diversity, increasing their guard radius is the *only* way that these approaches can improve their outage capacity. CTR, WFD, and SFD, on the other hand, exploit cooperation diversity and, therefore, a substantial improvement in capacity can already be gained from it so the guard radius need not be increased as much. Even though more nodes need to span guard zones (the destination *and* all relays), taken together their spatial consumption still differs from direct transmission and NCR in two orders of magnitude for practically relevant capacities (e.g., the highest IEEE 802.11g transmission rate of 54 Mbit/s corresponds to a target capacity of 2.7 bit/s/Hz).

While the increase from no diversity (or equivalently diversity order one) to diversity order two, at an outage capacity of 2 bit/s/Hz, improves the spatial consumption by *two orders of magnitude*, increasing the diversity order from two to three does not improve spatial consumption in the same quantity further. The break-even point of SFD with CTR is already reached at a capacity of 2.5 bit/s/Hz. For high outage capacities at high robustness, a diversity order of two suffices for efficient spatial reuse. Thus, depending on the desired target outage capacity, higher diversity orders may not pay off anymore if the network should also be efficient in terms of spatial reuse.

To generalize the observations made for two specific values for  $\epsilon$ , Figure 3.14 identifies the operation regions of direct transmission, CTR, and SFD. The figure shows the break-even outage capacities of the pair SFD and CTR as well as the pair CTR and direct transmission. For the break-even outage capacity, both approaches of a pair have the same spatial consumption. For example, at a required robustness of  $\epsilon = 10^{-3}$ , the SFD occupies less space for an outage capacity of up to 2.5 bit/s/Hz. Since the break-even point is at that capacity, for outage capacities above 2.5 bit/s/Hz the situation reverses and the CTR occupies less space. At 6.1 bit/s/Hz both CTR and direct transmission have the same spatial consumption. For outage capacities beyond 6.1 bit/s/Hz, direct transmission is the spatially most efficient approach.

## 3.4. Conclusions

In this chapter I derived a closed-form solution for the expected interference in wireless ad hoc networks with homogeneous node placement and path loss exponent  $\alpha > 2$ . I then used the derived expected interference with guard zone to establish a link between outage capacity and spatial consumption. My analysis revealed how the diversity order impacts the spatial reuse of a network. My analysis leads to the following conclusions.

1. Although cooperative relaying requires an additional guard zone at the relay, this does not imply that cooperative relaying blocks more nodes. In fact, the spatial consumption of WFD (with diversity order two) is even less than that of direct transmission to reach the same target outage capacity. However, how significant the spatial improvement is depends on the required robustness. The more robustness is

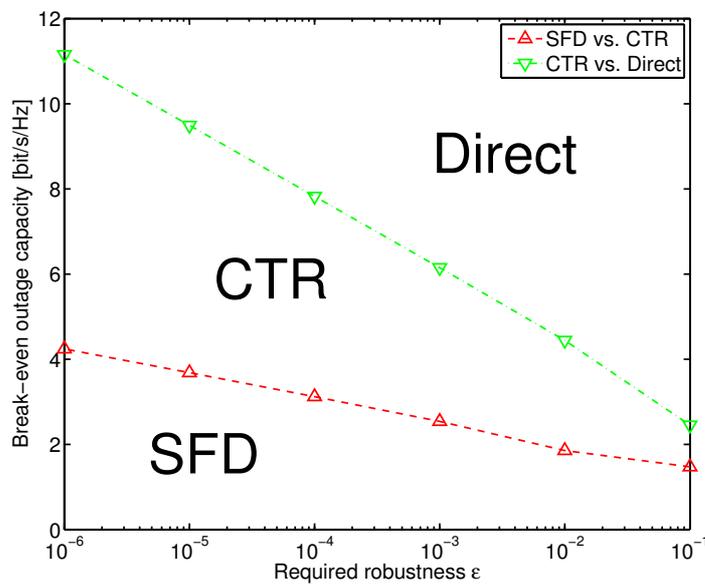


Figure 3.14.: Operating regions of direct transmission, CTR, and SFD for required robustness from  $10^{-6} \leq \epsilon \leq 10^{-1}$  with respect to their spatial consumption and path loss exponent  $\alpha = 3$ . A node density of  $\lambda = 10^{-3}$  has been chosen such that nodes experience a mean SIR of 32 dB (nodes transmit with unit transmission power). Note the logarithmic scale of the y-axis.

### 3. Spatial consumption of cooperative relaying

---

required, the better the spatial improvement is and hence the spatial reuse. Thus, I showed that cooperative relaying is even spatially more efficient than direct transmission or non-cooperative relaying where combined guard zones do not occur. This important result shows that the diversity gains not only benefit a *single* transmission in the network but also the *entire* network's capacity. For practically relevant data rates and robustness demands (e.g., voice and video traffic in cellular networks) cooperative relaying even consumes less space than conventional transmissions at the same capacity. As a consequence, the entire network capacity can be increased without compromising single transmissions.

2. When it comes to spatial consumption, a diversity order as high as possible is not necessarily desirable. It all depends on the required robustness  $\varepsilon$  and the target capacity  $C^{\text{out}}$  at which the network should operate. As a solution, network engineers can use the framework derived in this thesis to quantify the operating regions of cooperative relaying (Figure 3.14). If CTR is spatially more efficient than SFD for the network's desired  $\varepsilon$  and  $C^{\text{out}}$ , it suffices to implement cooperative relaying such that it merely uses a diversity order of two. This greatly simplifies the design and implementation of the required MAC protocol as the next chapter will show in more detail.

My results describe average networks when interfering nodes are randomly located. My results assume the relays at fixed positions and hence do not assume any sophisticated relay selection scheme. In fact, by choosing a relay that minimizes the number of blocked nodes, the performance of cooperative relaying may even increase. My results indicate that cooperative relaying is even spatially efficient. Although the problem can indeed be worse for specifically constructed configurations, the results for the spatial consumption of the average case are much better than one would anticipate from the results in [60].

All in all, the framework that I developed in this chapter is a powerful tool to evaluate the potential spatial reuse of non-cooperative and cooperative transmission approaches in wireless networks.

## 4. Mechanisms for cooperative unicasting

*Cooperative relaying requires to allocate the wireless channel to at least three nodes. Since the multiplexing loss severely reduces the achievable throughput using cooperative relaying, I analyze how to compensate for this loss. I first design a rate adaptation algorithm for cooperative relaying that optimizes the throughput per link. Simulations indicate that such link-wise adaptation is only beneficial when one of the links operates using the most robust transmission mode. Second, along multi-hop paths, multiplexing loss can be efficiently mitigated by using information from the routing layer to improve the MAC layer's performance. I design a routing-informed MAC protocol for efficient cooperation and demonstrate its performance by simulations as well as experiments.*

In large wireless networks it is likely that source and destination can only be reached via intermediate nodes. It is the task of the routing protocol to find a path connecting source and destination; the special case where source and destination are direct neighbors is a point-to-point transmission.

When applying cooperative relaying to improve performance in spite of fading in such wireless multi-hop networks, it is straightforward to apply cooperative relaying on every hop along the route. Figure 4.1(a) shows this idea. Along the path from source  $a$  via the intermediate nodes  $b$ ,  $c$ , and  $d$  to the final destination  $e$ , the relays  $s$ ,  $t$ ,  $v$ , and  $w$  establish a CTR on every hop. This hop-wise approach provides a diversity order of two at any

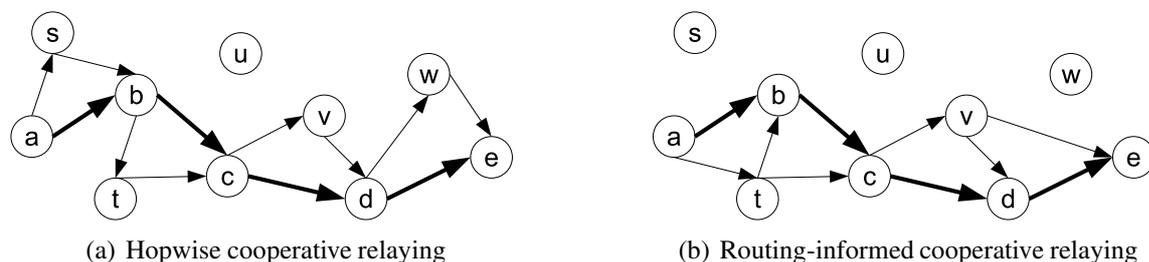


Figure 4.1.: In wireless multi-hop networks, cooperative relaying can either be applied per hop when only MAC information is available; with routing information at hand, cooperative relaying can be organized more efficiently such that a single cooperative retransmission can provide cooperative diversity at two hops.

hop, associated with a multiplexing loss of one half at every hop as well. If only one-hop neighborhood information is at hand (which is the case from a *pure* MAC perspective), cooperative relaying becomes more efficient by either using an *adaptive* strategy, trying to apply cooperative relaying only when needed and suitable relays are available, or by using *rate adaptation* to better adapt throughput to the wireless channel. I discuss whether rate adaptation for cooperative relaying mitigates the multiplexing loss in Section 4.1.

Figure 4.1(a) shows that node  $t$  is not only in the vicinity of  $a$  and  $b$  but also in the vicinity of the receiver  $c$  of the subsequent hop. If the MAC protocol had chosen node  $t$  for assisting the transmission  $a \rightarrow b$ , its retransmission would have also benefited the future hop  $b \rightarrow c$ . Since the MAC protocol only has one-hop neighborhood information at its disposal, it is unable to make such a foreseeing decision. Knowing the route would enable the MAC to choose a more efficient cooperative relay whose *single* retransmission could then assist *two* successive unicast transmissions along the route, providing a diversity order of two at both hops. Figure 4.1(b) shows an example in the same network, where relay  $t$  and  $v$  cooperate more efficiently. I call this cooperation scheme *two-for-one* cooperation as it mitigates the multiplexing loss of cooperative relaying in multi-hop networks by sparing one transmission, but it still achieves a diversity order of two at all nodes. In Section 4.2, I develop a routing-informed cooperative MAC protocol to exploit two-for-one cooperation in practical systems. I show that my proposed protocol is more efficient not only by simulation, but also by implementing an actual prototype using a Software-Defined Radio (SDR).

### 4.1. Rate-per-link adaptation for cooperative relaying

*Rate adaptation* – also called Adaptive Modulation and Coding (AMC) – trades off the transmitter’s data rate and robustness to improve the performance for varying channel conditions [97], e.g., by using the observed link state for adapting the coding rate and modulation type. It chooses higher-rate yet less robust modulations and Forward Error Correction (FEC) codes when channel conditions should still allow meeting the error bound. I base the following discussion on SNR as it provides an adequate compromise between practical relevance and theoretical tractability [97].

Integrating cooperative relaying into IEEE 802.11a WLANs shows a significant increase in throughput for low transmission powers [87] where diversity gains lead to exponentially decreased error rates in the order of retransmitting relays. At these low powers, despite the multiplexing loss, the received data rate<sup>1</sup> outperforms that of direct transmission. At lower error rates, rate adaptation may switch to higher transmission rates to increase throughput. So the question is whether the *combination* of rate adaptation and cooperative relaying can mitigate multiplexing loss and improve the overall throughput. However, with higher rates, the robustness of transmissions decreases and it is unclear to what extent the additional

---

<sup>1</sup>In this thesis, *received rate* refers to the rate of correctly received data bits.

diversity gain from cooperative relaying justifies the higher rates chosen by rate adaptation.

Lin et al. [58] first analyzed rate adaptation in the CTR, but due to a wrong path loss model, unrealistic power gains distorted the performance results. Nechiporenko et al. [65] reported better results for cooperative networks using adaptive  $M$ -QAM transmissions and AF relaying. Source and relay always use the same modulation and the destination applies MRC on symbol level. They observed that the channel's capacity could be approached within 5 dB when rates were continuously adapted, with a 1.5 dB penalty using discrete rates. Unfortunately, requiring the *same* modulation on both source and relay uplinks, which in the presence of fading may experience different channel conditions, reduces the spectral efficiency. For this reason, Bin Sediq and Yanikomeroğlu [7] proposed a combining scheme for different modulations that, *after* separately demodulating the signals from source and relay, weights and adds the resulting soft-bit symbols. This is unlike MRC which weights and combines signals *before* demodulation. The so-called Soft-Bit Maximum Ratio Combining (SBMRC) is close to the optimal yet infeasible maximum likelihood detector and outperforms selection combining by almost 2 dB. SBMRC makes it possible to use arbitrary pairs of modulations for cooperative relaying, facilitating the development of more efficient cooperative rate-adaptation algorithms, but to the best of my knowledge no such adaptation exists for cooperative networks yet that could mitigate the multiplexing loss.

#### 4.1.1. Upper bound on the data rate for cooperative relaying

I first establish an upper bound on the data rate using the outage capacity  $C^{\text{out}}$ . For an  $\varepsilon$ -constrained system,  $C^{\text{out}}$  provides the theoretical maximum for the received data rate, i.e., the maximum goodput it can achieve for the given error rate constraint and with ideal channel knowledge and coding. No system with an arbitrary rate adaptation scheme can reach a higher performance than  $C^{\text{out}}$ . I focus on the CTR and choose three different relay locations. In the symmetric CTR, all node distances are equal. Consequently, on the average all channels have the same error rate for fair comparison. In the asymmetric CTR, the relay is closer to the source, which increases its probability of successful decoding required for cooperative relaying using SDF compared to the symmetric CTR. Both source and relay uplinks have the same distance to the destination, so that the average error rate of the relay's retransmission alone is equal to the original transmission from source to destination. Finally, in the chain CTR, the relay resides half way in between source and destination so that both source-relay and relay-destination channels experience a gain through reduced path loss, benefiting the overall performance. Table 4.1 summarizes the outage capacities for these three scenarios as well as for direct transmission. One can derive the outage capacities by inserting the appropriate path losses for the channel gains  $\Gamma_{xy}$  and factoring out the reference SNR  $\Gamma$  which captures only transmission power and noise level.

Solving the resulting outage capacity equations for different values of  $\Gamma$ , a path loss exponent  $\alpha = 3$ , and for  $\varepsilon = 0.1$  (i.e., a WLAN's 10 % error rate bound) yields the numerical results in Figure 4.2. In the symmetric CTR, exploiting cooperative diversity only gives a

#### 4. Mechanisms for cooperative unicasting

Approach	Configuration	Outage capacity
CTR symmetric		$\frac{1}{2} \log_2 \left( \frac{\sqrt{\epsilon}}{2^\alpha} \Gamma + 1 \right)$
CTR asymmetric		$\frac{1}{2} \log_2 \left( \frac{\sqrt{\epsilon}}{\sqrt{2^{\alpha-1}(1+2^\alpha)}} \Gamma + 1 \right)$
CTR chain		$\frac{1}{2} \log_2 \left( \frac{\sqrt{\epsilon}}{\sqrt{2^\alpha}} \Gamma + 1 \right)$
Direct		$\log_2 \left( \frac{\epsilon}{2^\alpha} \Gamma + 1 \right)$

Table 4.1.: Outage capacities of symmetric, asymmetric, and chain Cooperative Triangles (CTRs) as well as direct transmission.

negligible gain compared to direct transmission. Here, even ideal rate adaptation can only slightly improve throughput. This is a result of the *linear* multiplexing loss  $1/K$  which dominates  $C^{\text{out}}$  compared to the merely *logarithmic* effect of the diversity order  $L$ . With a weaker direct channel, however, capacity gains increase. The chain configuration offers the largest potential for rate adaptation where the direct channel is the weakest compared to *both* relay links. In the asymmetric configuration, significant throughput gains in between those of the symmetric and the chain configuration are achievable.

#### 4.1.2. Cooperative rate adaptation algorithm for $N$ relays

In an arbitrary cooperative configuration such as the one in Table 2.1, rate adaptation must operate on  $2N + 1$  channels, i.e., one direct channel,  $N$  inter-user channels, and  $N$  uplink channels. It is not clear which set of transmission modes a rate adaptation algorithm should choose for cooperative relaying to maximize the received rate. To answer this question, I describe an algorithm that iterates over the entire range of discrete SNR values that occur for all channels involved in a practical system to find the appropriate transmission modes. This algorithm, being part of the design phase, runs offline. At run-time, nodes in the network only use its results to look up the most suitable rates for adaptive transmission in

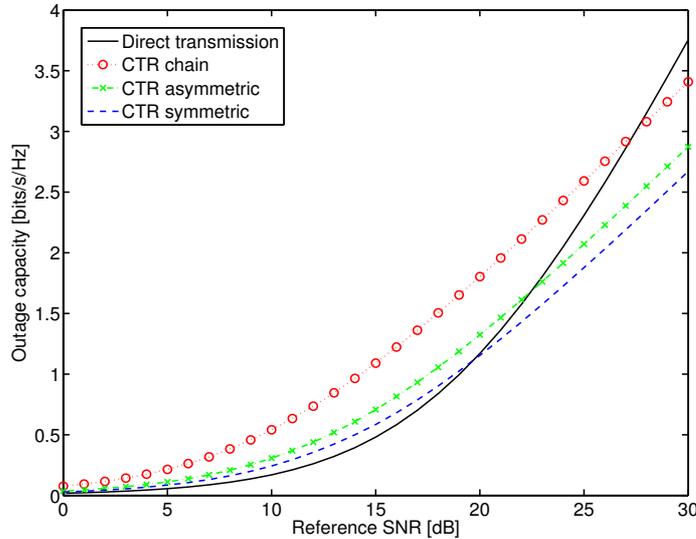


Figure 4.2.: Theoretical performance bounds: Outage capacity  $C^{\text{out}}$  in bit/s/Hz for  $\epsilon = 0.1$ ,  $\alpha = 3$ , direct transmission, and SDF cooperation (single relay) for the three configurations in Table 4.1.

the CTR. Algorithm 4.1 states the algorithm in its general form for  $N$  relays.

The developer must choose the practically relevant range of SNR values that a receiver can observe and choose a suitable resolution to obtain a finite set of discrete values. Then, for any particular SNR tuple (Line 1), the algorithm determines the transmission modes used by source  $s$  and relays  $r_i$  that maximize the received rate  $R^{\text{rx}}$ . To do so, the algorithm must be able to compute the Bit Error Rate (BER) of a transmission for a particular transmission mode  $\tau$  taking both modulation and code into account, given the SNR  $\gamma$ . BER results for direct transmission can be derived analytically [74], and for cooperative transmission with MRC and SBMRC at the destination, I use the numerical results found in [7]. Since these BER results are for *uncoded* transmissions, I still need to add the coding gains for the individual transmission modes. I numerically determine the relevant coding gains  $R_c := \{1/2, 2/3, 3/4\}$  by using reference transmissions.

Figure 4.3 shows the necessary reference transmissions for Binary Phase Shift Keying (BPSK) and the code rates of the set  $R_c$ . To obtain the BER with coding, I first determine the  $E_s/N_0$  of an uncoded reference transmission. For example, assume the modulation used is BPSK and the uncoded BER is 10%. The uncoded BPSK reference transmission achieves this BER at an  $E_s/N_0$  of  $-1$  dB. At this  $E_s/N_0$ , a BPSK transmission using a code with rate  $1/2$  on top achieves a BER of 0.5% and a code with rate  $3/4$  achieves a BER of 31%. Using this two-step translation, any uncoded BER can be transformed into the corresponding BER with coding. I created all reference transmissions by simulating an AWGN channel. The coded reference transmissions use a convolutional code with

#### 4. Mechanisms for cooperative unicasting

**Algorithm 4.1** Offline algorithm for determining SNR thresholds and transmission modes for SDF cooperative relaying with rate adaptation.

```

1: for  $\gamma_{s,d}, \gamma_{s,r_1}, \dots, \gamma_{s,r_N}, \gamma_{r_1,d}, \dots, \gamma_{r_N,d}$  do
2:    $R_{\max}^{\text{rx}} \leftarrow 0$  and  $\tau_i \leftarrow 0$  for all  $1 \leq i \leq N$ 
3:   for  $\tau_s, \tau_{r_1}, \dots, \tau_{r_N}$  do
4:      $S_r \leftarrow \prod_{i=1}^N [1 - P_{\text{dir}}^{\text{BER}}(\tau_{r_i}, \gamma_{s,r_i})]$ 
5:      $S_d \leftarrow [1 - P_{\text{coop}}^{\text{BER}}(\tau_s, \tau_{r_1}, \dots, \tau_{r_N}, \gamma_{s,d}, \gamma_{r_1,d}, \dots, \gamma_{r_N,d})]$ 
6:      $R^{\text{rx}} \leftarrow S_r S_d / [1/R^{\text{tx}}(s) + \sum_{i=1}^N 1/R^{\text{tx}}(r_i)]$ 
7:     if  $R^{\text{rx}} > R_{\max}^{\text{rx}}$  then
8:        $R_{\max}^{\text{rx}} \leftarrow R^{\text{rx}}$  and  $\tau_{\max} \leftarrow (\tau_s, \tau_{r_1}, \dots, \tau_{r_N})$ 
9:     end if
10:  end for
11:   $A[\gamma_{s,d}; \gamma_{s,r_1}; \dots; \gamma_{s,r_N}; \gamma_{r_1,d}; \dots; \gamma_{r_N,d}] \leftarrow \tau_{\max}$ 
12: end for

```

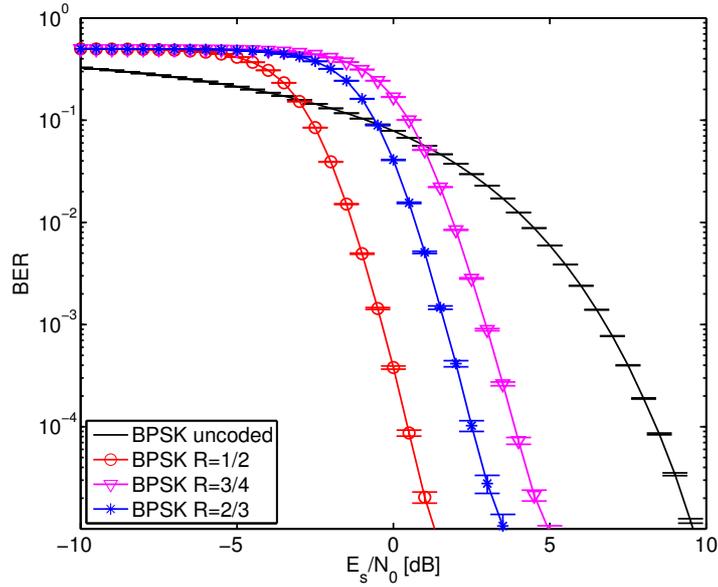


Figure 4.3.: Bit Error Rate (BER) of Binary Phase Shift Keying (BPSK) for uncoded transmission and convolutional coding with rates 1/2, 3/4, and 2/3 obtained by simulating transmissions over an AWGN channel. Confidence intervals shown for 95 % confidence level.

Table 4.2.: Transmission modes of the IEEE 802.11a physical layer [39]

Mode $\tau$	Modulation	Coding rate $R_c$	Coded bits	Data bits	Data rate [Mbit/s] per OFDM symbol
1	BPSK	1/2	48	24	6
2	BPSK	3/4	48	36	9
3	QPSK	1/2	96	48	12
4	QPSK	3/4	96	72	18
5	16-QAM	1/2	192	96	24
6	16-QAM	3/4	192	144	36
7	64-QAM	2/3	288	192	48
8	64-QAM	3/4	288	216	54

constraint length 7 and generator polynomial [171, 133] according to the IEEE 802.11a standard [39].

I denote the BER for *direct* transmission from source to a particular relay as  $P_{\text{dir}}^{\text{BER}}(\tau, \gamma)$ , assuming that mode  $\tau$  is used with an SNR  $\gamma$ . Thus,  $P_{\text{dir}}^{\text{BER}}(\tau, \gamma)$  must take both modulation and code into account. Due to the assumption of quasi-static fading,  $\gamma$  holds for the entire packet. Correspondingly, the BER achieved using a *cooperative* transmission from all users to the destination is  $P_{\text{coop}}^{\text{BER}}(\tau_s, \tau_{r_1}, \dots, \tau_{r_N}, \gamma_{s,d}, \gamma_{r_1,d}, \dots, \gamma_{r_N,d})$  where each user may use an individual transmission mode with all uplink SNRs given. Again, this BER must reflect the modulations and codes used. The results for SBMRC already contain all possible combinations of modulations. By arbitrarily combining the code rates in  $R_c$  for two transmissions, the code rate of the combined transmission can then be any value in  $R_{c,\text{coop}} := \{1/2, 7/12, 5/8, 2/3, 17/24, 3/4\}$ , thus requiring additional reference transmissions for looking up the corresponding coding gains.

Due to the SDF protocol, transmissions to all relays must succeed for them to cooperate (Line 4) and  $S_r$  denotes the probability for that<sup>2</sup>. Similarly,  $S_d$  denotes the probability of successful reception at the destination. The rate  $R^{\text{rx}}$  accounts for the correctly received bits (Line 6) during a cooperative transmission cycle, where  $R^{\text{rx}}(\tau)$  denotes the rate that a transmitter achieves with mode  $\tau$ . In the analysis that follows I use the modes and rates summarized in Table 4.2. For all possible combinations of transmission modes (Line 3), the algorithm computes the received rate and stores the rate-maximizing modes (Line 8) in a  $(2N + 1)$ -dimensional matrix  $A$  (Line 11). It is important to note that Algorithm 4.1 is an offline algorithm for computing static look-up tables that, for a particular SNR tuple, give the appropriate transmission mode. During network operation, the wireless nodes constantly measure the signal strength of received packets to estimate the SNRs on the in-

<sup>2</sup>In this section, this static strategy is feasible as I only consider the CTR and thus  $N = 1$ . If more relays are available, it suffices if only one of them receives the message to cooperate. Note that a modification of Line 4 also requires to modify Line 5 accordingly to reflect the relay selection scheme used.

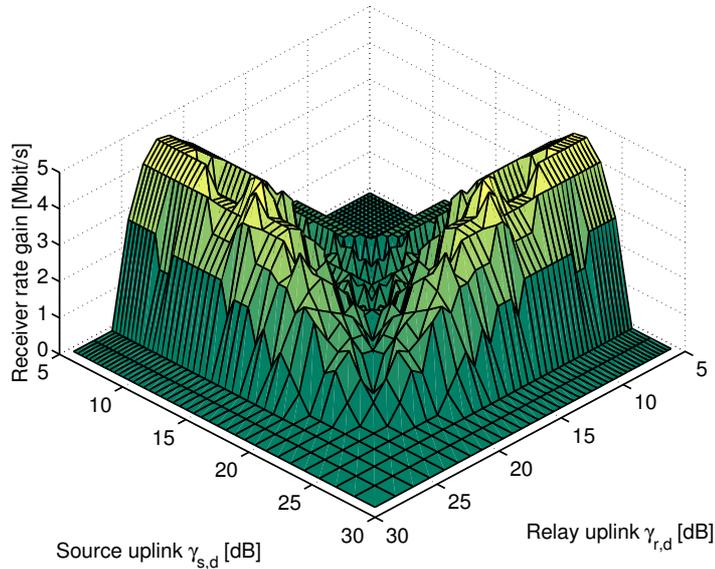


Figure 4.4.: Combining with rate-per-link adaptation theoretically increases the achievable received rate up to 5 Mbit/s for an IEEE 802.11a WLAN compared to link-equal adaptation. The channel between  $s$  and  $r$  is assumed to be perfect.

volved channels. Nodes use these pre-computed look-up tables online to find the appropriate transmission mode to use for the next packet to be transmitted. Thus, the transmission modes are also estimates and may be inappropriate since the channel state may change between the time of measurement and the time of transmission. Thus, Section 4.1.3 assesses the quality of this table-based approach by simulations in a time-correlated fading scenario where measurements may become obsolete.

Conventional MRC requires all transmitters to use the same modulation and code. In this case, the algorithm can only apply *equal* modes in Line 3, i.e.,  $\tau_s = \tau_1 = \dots = \tau_k$ , severely reducing the degrees of freedom for rate adaptation. In the case of IEEE 802.11a, only 8 out of  $8^{N+1}$  possibilities remain. The limitation of equal rates is no longer required for SBMRC. I identify the additional gain that this more flexible combining brings in Figure 4.4 for the CTR ( $N = 1$  relay). The figure shows the difference in received rate for the single-relay case between an unrestricted run of the algorithm, allowing rate-per-link adaptation, and a restricted run with  $\tau_s = \tau_r$ , only offering link-equal adaptation. A gain in rate occurs whenever there exist some  $\tau_s \neq \tau_r$  that achieve a higher received rate using SBMRC than the best  $\tau$  used on both source and relay uplink for the same transmission. For SNR tuples where no gain exists, it holds that  $\tau_s = \tau_r = \tau$ , i.e., the same transmission mode achieves the maximum received rate for both rate-per-link and link-equal adaptation. Figure 4.4 illustrates two important points. First, a notable difference in received rate of up to 5 Mbit/s exists for an IEEE 802.11a physical layer. This is the potential for improvement

that rate adaptation can bring for this specific wireless system. Second, to achieve a gain of 4 Mbit/s or more, the SNRs of both uplinks must differ by at least 8 dB. The smaller the SNR difference becomes, the smaller the gains in rate.

These results indicate that, for cooperative relaying, adapting individually *per link* offers new performance enhancements that might mitigate the multiplexing loss. The proposed algorithm is generic enough so that one can apply it to study the *potential* gains in different systems.

### 4.1.3. Simulations in a fading scenario

To characterize the performance that a combination of cooperative relaying and rate adaptation brings, I compare it in terms of received rate with the following schemes:

**Direct transmission** The transmission on the  $(s, d)$  channel does not have multiplexing losses and achieves the best received rate for high SNR, but will be more susceptible to fading effects for low SNR.

**Cooperative Triangle (CTR)** This is cooperative relaying in the three-node configuration using the SDF protocol with diversity combining at the destination. For rate adaptation, I use the transmission modes found with Algorithm 4.1 for the link-equal ( $\tau_s = \tau_r$ ) case with MRC and the per-link case with the more general SBMRC. I also provide results for a static choice of BPSK 1/2, as it is the most robust transmission mode and no rate adaptation scheme can adapt below it.

**Non-Cooperative Relaying (NCR)** Here, the destination processes only packets received from the relay which, in turn, forwards the source's correctly decoded packets. Direct rate adaptation operates independently on the  $(s, r)$  and  $(r, d)$  channels for maximum spectral efficiency. Comparing CTR with this scheme allows to characterize the cooperation diversity gain.

I study the configurations shown in Table 4.1 with path loss exponent  $\alpha = 3$ . I assume signaling to be perfect in the sense that it is error-free but it takes time to be exchanged between the nodes. During that time, the wireless channel may change depending on the relative speed  $v$ , leading to prediction errors and hence inappropriate transmission modes. Figure 4.5 shows the received rate for the asymmetric configuration and  $v = 1$  m/s, corresponding to, e.g., an indoor scenario. Operating the CTR using BPSK at code rate 1/2 as the most robust static transmission mode achieves an increase of up to 1 Mbit/s over direct transmission for low transmission powers due to the cooperative diversity gains. Direct transmission with rate adaptation performs worse as it cannot switch below BPSK 1/2 to compensate for errors, i.e., it cannot become more robust. NCR can adapt individually to the  $(s, r)$  and  $(r, d)$  channels with the same multiplexing loss like CTR, but it does not exploit cooperative diversity. Its performance is even worse than direct transmission because

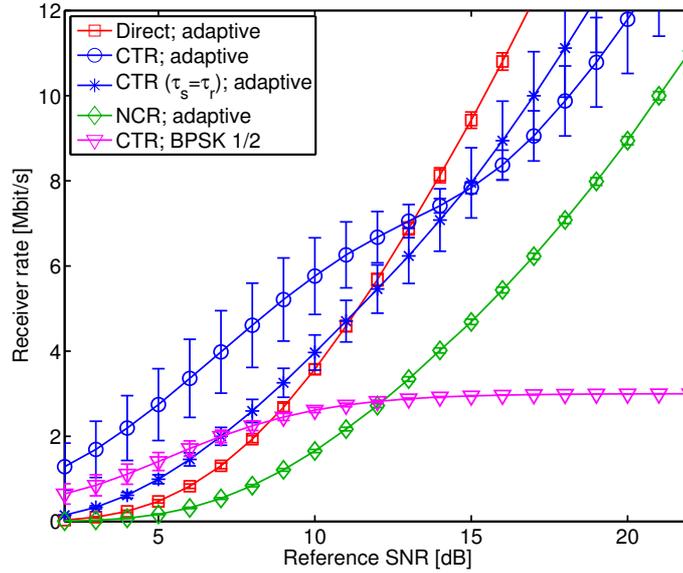


Figure 4.5.: Received rate vs. transmission power, asymmetric configuration,  $v=1$  m/s. Confidence intervals shown for 95 % confidence level.

of the additional channel from source to relay which may fail independently from the up-link. CTR with rate-per-link adaptation yields a maximum gain of roughly 2.5 Mbit/s for low transmission powers and approaches the received rate of NCR for increasing power. Restricting CTR to link-equal adaptation severely restricts the possible gains; the received rate even falls below that of the static CTR for low powers while still outperforming direct transmission.

Since rate adaptation is susceptible to the channel's coherence time due to prediction errors, I also study the effect of increasing the relative speed  $v$  depicted in Figure 4.6. For  $v = 50$  m/s, corresponding to, e.g., a moving train, the channel state decorrelates, making it impossible for rate adaptation to predict the channel state. Consequently, the static choice of BPSK 1/2 achieves the best received rate for low transmission power as it always uses the most robust rate and does not erroneously choose higher, less robust transmission rates. Adaptation in the CTR performs worse than direct transmission because now two channels are predicted separately and may suffer from prediction errors instead of only one channel as with direct transmission. Cooperation *with* rate adaptation is not feasible at high speeds, so I focus on  $v = 1$  m/s for the remaining two configurations.

At this low speed, for the symmetric configuration, direct transmission with rate adaptation outperforms any other scheme as the multiplexing loss dominates the diversity gains (Figure 4.7). If, however, the relay resides half way in between source and destination, the reduced distances benefit from the cubically reduced path loss (refer to Figure 4.8). Adaptation in the CTR achieves a maximum gain of 4 Mbit/s; although adaptive NCR

#### 4.1. Rate-per-link adaptation for cooperative relaying

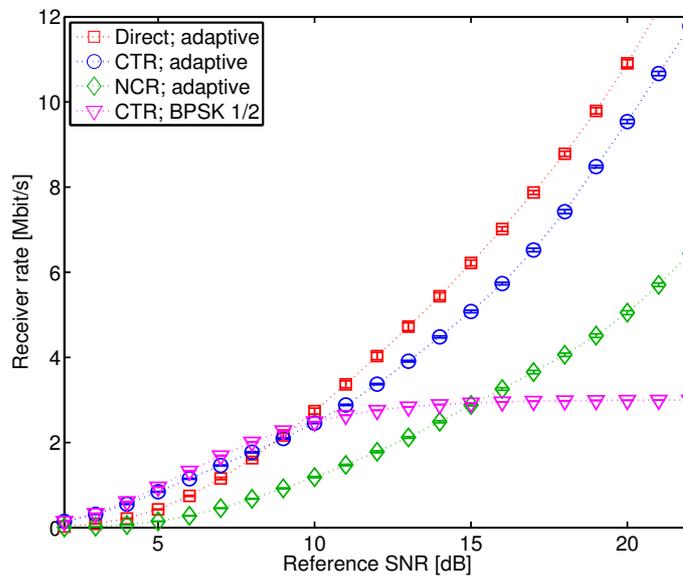


Figure 4.6.: Received rate vs. transmission power, asymmetric configuration,  $v=50$  m/s. Confidence intervals shown for 95 % confidence level.

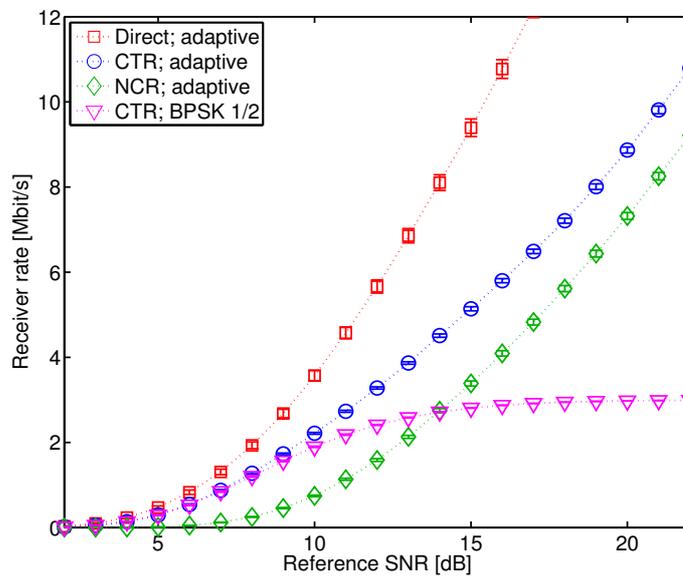


Figure 4.7.: Received rate vs. transmission power, symmetric configuration,  $v=1$  m/s. Confidence intervals shown for 95 % confidence level.

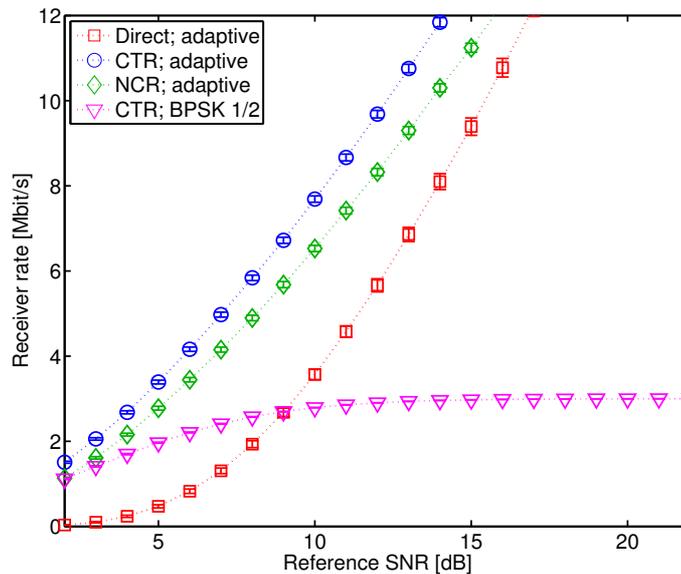


Figure 4.8.: Received rate vs. transmission power, chain configuration,  $v=1$  m/s. Confidence intervals shown for 95 % confidence level.

outperforms direct transmission as well, the difference to adapting in the CTR is with a maximum of roughly 1.8 Mbit/s still worthwhile.

Rate-per-link adaptation is only suitable to mitigate the multiplexing loss of cooperative relaying for low transmission powers and thus low SNRs, where direct rate adaptation cannot become more robust to avoid transmission errors. Here, diversity gains help and rate-per-link adaptation mitigates the multiplexing loss. When the relay always adds redundant transmissions, even with sophisticated rate adaptation, the multiplexing loss prevails. Thus, rate-per-link adaptation alone is not efficient enough. Thus, in the next section, instead of adapting the transmission rates, I mitigate multiplexing loss by *avoiding* cooperative retransmissions to some extent without compromising the diversity order at receivers. This is only possible in multi-hop networks where source and destination connect via intermediate nodes, determined by some ad hoc routing protocol.

## 4.2. Designing a routing-informed cooperative MAC protocol

While rate-per-link adaptation can make more efficient use of the available bandwidth by adapting channels individually, the multiplexing loss of cooperative relaying still remains. Therefore, I suggest a different approach to address the multiplexing loss of cooperative relaying in multi-hop networks by reducing the number of re-transmissions. As Figure 4.1(a)

shows, a hop-focused cooperation scheme squanders a crucial benefit in such multi-hop networks: the fact that a packet travels along a multi-hop route means that not only the current addressee of a packet might be in need of help, but also that a following hop exists whose receiving node could benefit from such a helping transmission.

Figure 4.1(b) illustrates this situation: Node  $a$  sends to  $e$  via some intermediate nodes. During the first hop, the packet is transmitted to  $b$ ; node  $t$  could assist here by retransmitting  $a$ 's packet, which it has overheard, to achieve a diversity gain at  $b$ . I motivate my approach by observing that node  $c$  could *as well* benefit from this broadcast transmission when, later on, it will receive the packet from node  $b$ . Thus, the *single* retransmission of redundancy can improve the success probabilities of *two* ordinary transmissions. I call this *two-for-one* cooperation; it is specific to the diamond configuration that frequently occurs in multi-hop networks. Refer to Appendix B for a simulation-based study that shows the occurrence probability of WFD and SFD for random node placements.

Opportunistic relaying [9] and opportunistic routing [8] are similar approaches that also combat fading effects. At the MAC sub-layer, opportunistic relaying chooses the best relay before it transmits. For deciding upon the best relay, both opportunistic approaches require precise CSI at the transmitters [85]. Bletsas et al. propose to use the existing RTS/CTS frame exchange of IEEE 802.11 protocols. Potential relays that overhear an RTS/CTS exchange start a timer inversely proportional to a channel estimate  $\gamma$ . This estimate is based on the received signal strength of both packets that indicate the quality of the inter-node channel  $\gamma_{si}$  (the instantaneous SNR between source  $s$  and relay  $i$ ) and the relay's uplink channel  $\gamma_{id}$  (the instantaneous SNR between relay  $i$  and destination  $d$ ). When the timer of a potential relay expires, the potential relay first senses the medium. If it cannot detect a signal, e.g., using CCA, it broadcasts a packet to announce its participation. The time-out serves as a backoff in which the node with the earliest time-out becomes the cooperating relay. Since all nodes must sense the channel before making an announcement, only one node can win the contention assuming the announcement is of sufficient duration. If potential relays may be hidden from each other, source and destination must announce the winner of the time-out period through a flag packet, therefore causing additional signaling overhead.

My approach does not require precise CSI at the transmitters since it does not try to choose the best relay beforehand. Instead, it provides the destination with two copies of the same message and lets the destination choose the best symbols in case of MRC or packets in case of PSC *after* the transmission. Although my approach does not require CSI at the transmitters, a relay must know whether the receivers of the first and second hop are its neighbors. Unlike precise CSI, this information is easier to obtain and needs to be less frequently updated.

Opportunistic routing moves this *a priori* selection of the best relay to the routing layer [8]. It does not specify a particular next hop as the receiver. Instead, it exploits multi-user diversity by letting all nodes in the vicinity receive and from those that correctly received the message, only the one closest to the destination retransmits. Again, this requires CSI to decide upon the best node before transmission. Unlike opportunistic routing,

my approach can be implemented solely at the physical layer and the MAC sub-layer as long as the routing protocol provides information about the first and second hop receivers. In this case, the routing protocol need not be replaced and two-for-one cooperation can be more efficiently implemented. Section 4.2.2 describes an implementation only at driver level for the Linux operating system and Atheros WLAN hardware.

I published my work on two-for-one cooperation in 2008 [52]. In 2009, Adam et al. [1] also recognized the potential benefit of routing-informed cooperation. One year later, they published a relay selection scheme that exploits routing information to choose a neighbor along a path that can span a diamond and provide the two-for-one benefit. Unlike my work, Adam et al. do not design and implement a routing-informed MAC protocol. They only describe the principle of a cooperative transmission phase and leave it open how to actually implement it and which layers of the protocol stack an implementation involves.

Two-for-one cooperation requires a careful design of the MAC protocol which tells a relay when to listen and when to forward the packet despite not being on the actual route. This MAC should suppress interfering transmissions around the relay; it also has to instruct the receiver that there is additional data incoming to decode a transmission.

I propose a protocol that uses information from the routing layer to recognize WFDs along a given route. Forming a diamond not only requires a relay for assistance in the first hop, but the *same* relay must first be able to assist in the following hop as well *and* second know this fact already when assisting in the first hop. Only in this case, the following hop's receiving node  $d$  can benefit from cooperation at no additional bandwidth consumption. In the remainder, I assume a relay-initiated form of cooperation, where a node overhearing a transmission decides to cooperate based on its own neighborhood table and the following hop of a packet (there is no need to know the entire route).

The WFD setting in Table 2.1 forms the basis of the following discussion, and I assume that  $a \rightarrow c \rightarrow d$  is part of the route. As depicted in Figure 4.9, node  $a$  initiates the transmission  $a \rightarrow c$  using a  $\text{RTS}(a, c, d)$ , to which the intermediate hop  $c$  replies with a  $\text{CTS}(a)$ . Unlike IEEE 802.11, my protocol not only includes the receiving node's MAC address in the RTS, but it also includes the MAC address of the future hop's receiving node  $d$  (available from, e.g., source routing or extended routing headers).

In case node  $d$  does decode such an RTS, it knows that it will eventually receive a packet. It can, thus, opportunistically buffer even a corrupt version of the packet, overheard when sent from node  $a$  to node  $c$ , for later combining with other copies of the same packet. Such a case is plausible even if direct exchange of data frames between  $a$  and  $d$  is not possible – if it were, the routing protocol would have preferred  $a \rightarrow d$  over  $a \rightarrow c \rightarrow d$  in the first place – but node  $d$  might receive the shorter, typically slowly transmitted, and thus more robust management frames correctly. This may increase diversity at negligible cost; if node  $d$  cannot decode node  $a$ 's RTS, no harm is done.

Node  $a$  then transmits its data frame to  $c$ , typically using a higher, less robust data rate than that of the RTS. If  $c$  can correctly decode the data, it acknowledges this to  $a$ . The intermediate node  $c$  would, in the same way, try to convey the received data to  $d$ . Without combining, this would resemble Non-Cooperative Relaying (NCR), but with  $d$  having

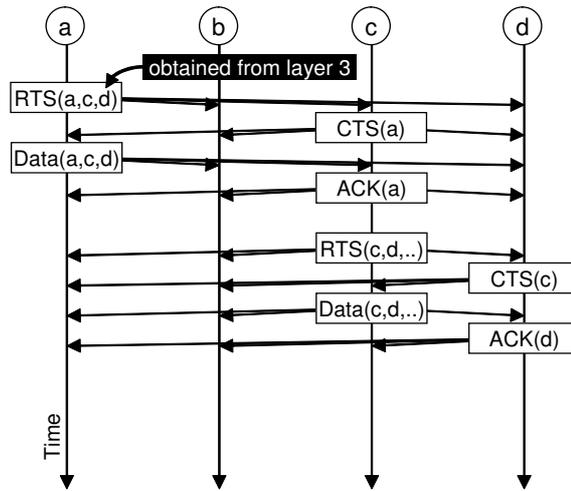


Figure 4.9.: If all data transmissions with their associated ACKs succeed, the protocol resembles Non-Cooperative Relaying (NCR).

already buffered the data frame, the second hop can benefit from cooperative diversity. In other words, node  $d$  tries to decode the data received over the second hop  $c \rightarrow d$  or, if that fails, decode the data by combining it with the buffered frame received opportunistically over the first hop  $a \rightarrow c$ .

Alternatively, if  $c$  cannot decode the data, it buffers the corrupted frame in the hope of later reconstructing it, based on additional redundancy received from a relay. A neighboring node  $b$  that happens to overhear node  $a$ 's data transmission may become a potential relay if it does not overhear the scheduled ACK packet  $c \rightarrow a$ , but it must also be eligible for relaying. Assuming SDF,  $b$  is only eligible if it has correctly decoded  $a$ 's data packet (at a high, less robust rate). Additionally, node  $b$  must complement the setting to a WFD, for which two conditions must hold. First, node  $b$  must be able to exchange at least management frames with node  $c$  (successful exchange of data frames is *not* necessary). Second, node  $b$  examines the additional routing information provided in the data frame, indicating the future hop  $c \rightarrow d$ . The relay  $b$  must know the future hop's receiver  $d$  (i.e., have an appropriate entry in its neighborhood table) and, again, exchanging at least management frames with that node  $d$  is necessary. The relay  $b$  retransmits  $a$ 's data packet after a time-out to assist node  $c$  in decoding it, thereby forming a CTR. Node  $c$  tries to decode the data from the retransmitted frame or, if that fails, decode the data by combining it with the buffered frame. The lack of an ACK makes node  $a$  increase its contention window, which statistically gives higher priority to node  $b$ 's transmission. This priority of  $b$  over  $a$  can be optionally further increased by either enlarging the ACK time-out at  $a$  or by assigning higher priority to  $b$  at the MAC sub-layer.

The CTR can be augmented by the retransmission of node  $b$  when routing information is available. Figure 4.10 shows that the relay  $b$  sends a modified  $\text{RTS}(b, d, c)$  that is addressed

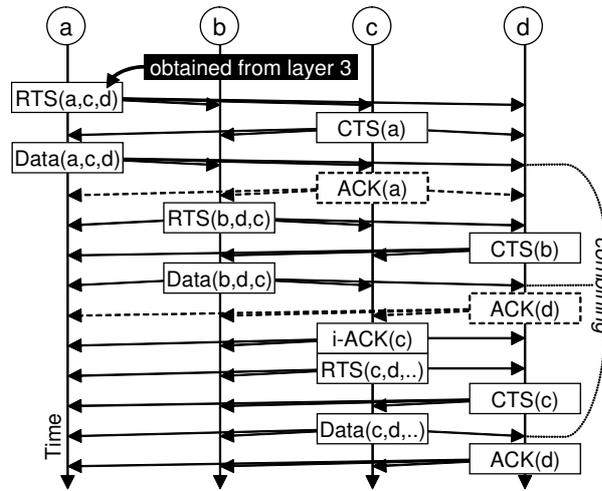


Figure 4.10.: Message sequence chart of the proposed two-for-one cooperative MAC using intermediate ACKs (iACKs); dashed lines indicate missed ACKs.

to *two* nodes, namely node *c* (forming a CTR) *and* the receiver of the following hop, node *d* (forming a WFD). By inspecting the routing information, relay *b* knows that *d* needs to get the frame eventually. If *d* replies with a CTS(*b*), the relay *b* transmits the data packet to *d*, but it is still overheard by node *c* and used for decoding – triggered by the reception of the RTS(*b*,*d*,*c*). Thus, the main difference to cooperation in triangles is that the retransmission of the relay in my protocol is exploited both at the *current and future* hop receivers *c* and *d*. One can distinguish the following two cases:

1. *Node d cannot decode the frame sent by node b.* Then, node *d* buffers the corrupt frame since an additional copy (to help in combining) may still arrive over the *c* → *d* link. The intermediate node *c* will also try to decode the data by combining the overheard frame received over *b* → *c* with the buffered one from *a* → *c*. Two sub-cases need to be distinguished:

- a) *Node c successfully decodes the frame.* Figure 4.10 shows that node *c* must send an intermediate ACK (iACK)<sup>3</sup> to node *a* so that *a* refrains from an unnecessary retransmission. Afterward, node *c* initiates a transmission to *d* as if it had received the data correctly in the first place. However, the data arrives at node *d* for at least the second time, since it was previously received over *b* → *d*, and maybe even over the direct link *a* → *d* when *d* successfully decoded the initial RTS(*a*,*c*,*d*). Therefore, node *d* can either decode the data directly or combine it with the previous, already buffered copy sent by node *b*. Thus, the single transmission by node *b* is again exploited for packet combining, realiz-

<sup>3</sup>Note that iACK and fACK (defined later) frame formats are identical to ACK; the names are only introduced here to ease description.

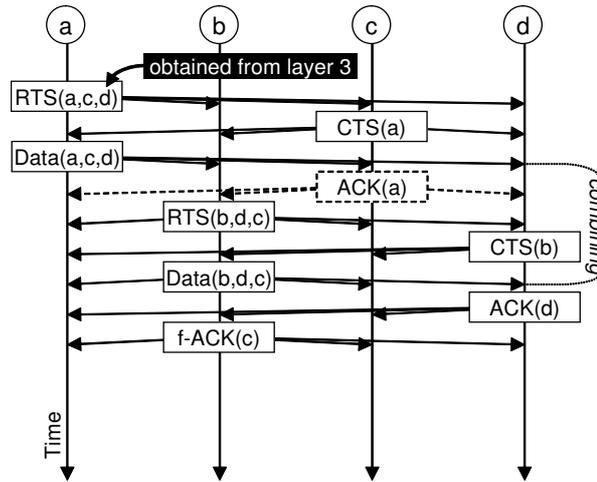


Figure 4.11.: Message sequence chart the proposed two-for-one cooperative MAC using faked ACKs (fACKs); dashed lines indicate missed ACKs.

ing the benefit of a two-for-one cooperation scheme that reaches diversity order two in the WFD.

- b) If node  $c$  still cannot decode the frame, no iACK is sent to  $a$  and the MAC time-out at  $a$  will cause the transmission to be repeated.
2. Node  $d$  correctly decodes the frame sent by node  $b$ , possibly after combination with the frame received directly from  $a$ . As a consequence,  $d$  acknowledges it, as depicted in Figure 4.11. The originally intended transmission  $c \rightarrow d$  has become superfluous now, and would reduce the received rate were its transmission not prevented. Therefore, node  $b$  sends a faked ACK (fACK) to node  $a$  which is still awaiting an ACK from  $c$ , as it does not know about the relay  $b$ . The fACK prevents  $a$  from retransmitting when it is not necessary. It can also be used to inform node  $c$  that the frame which it has buffered from  $a$  has become obsolete.

Finally, the protocol requires the following priorities to work efficiently: After the source has transmitted its data frame, the intermediate hop  $c$  must have highest priority to suppress the potential relay  $b$  when the data was correctly decoded. The relay must have a higher priority when contending with other stations to retransmit without significant delay. The time-out of the source  $a$  must be large enough to accommodate the transmission of the relay and to recognize faked and interim ACKs. As common in the IEEE 802.11 MAC protocol, these priorities can be established using appropriate Inter-Frame Spaces (IFSs).

#### 4.2.1. Simulations in a fading scenario

I first show the gain in received data rate that can be achieved by two-for-one cooperation in the symmetric diamond configuration. For this, I implement static transmission cy-

cles, assuming that all nodes are known and channel knowledge at the transmitter is ideal. Ideal implies that this channel knowledge is always error-free and does not cause overhead (which an actual protocol cannot avoid).

All channel gains experience Rayleigh fading assuming a relative speed of 1 m/s. All nodes use BPSK for modulation with a symbol time of 4  $\mu$ s and convolutional coding at rate 1/2 with generator polynomial [171, 133] according to the IEEE 802.11a standard [39] to encode a random bit string of 1,024 bytes. Nodes combine packets received over different channels using MRC. I compare the following protocol cycles:

**Direct transmission** Since direct transmission does not suffer from multiplexing loss, at sufficiently high SNR it always achieves the highest goodput. It is therefore an important comparison to identify the operating region in which (cooperative) relaying approaches are superior over direct transmission.

**Non-Cooperative Relaying (NCR)** Due to the non-linearity of path loss and regeneration of the packet at the relay, NCR is also an important comparison case. Implementations using existing MAC protocols are straightforward.

**Cooperative Triangle (CTR)** In absence of routing information, a MAC protocol can only apply the CTR twice, namely first for the transmission  $a \rightarrow b$  with relay  $c$  and second for the transmission  $b \rightarrow d$ , once again using relay  $c$ .

**Weak Full Diamond (WFD)** In presence of routing information, the relay  $c$  can efficiently cooperate by transmitting only *once* and exploiting its transmission at both  $b$  and  $d$ . The destination reaches a diversity order of two.

Figure 4.12 shows the positive effect that the additional diversity gain of the cooperative relaying approaches brings in terms of improved Packet Error Rate (PER) for increasing SNR in a symmetric diamond configuration. The increasing slope of CTR and WFD results from the higher diversity order. Both approaches achieve similar PER performance. Figure 4.13 shows the received rate for increasing SNR to identify the operating region of cooperative relaying and the impact of the multiplexing loss. At high SNR, the multiplexing loss severely limits the received rate. Here, cooperation diversity gains cannot compensate for the multiplexing loss anymore as it is not needed to deal with errors. With enough transmission power available, nodes should always prefer direct transmission if they demand high received rate alone. In practice, however, resources are often limited. For example, to deploy a limited number of nodes over an area as large as possible without losing connectivity, inter-node distances need to be as large as possible and hence the network operates at low SNR. In this case, cooperative relaying achieves more received rate than direct transmission or NCR. Since multiplexing loss is a dominant cost for all relaying schemes, using two-for-one cooperation achieves the same diversity order at less cost. While CTR requires a multiplexing loss of one fourth, WFD improves the received rate by its reduced multiplexing loss of one third. To obtain this improvement, the MAC protocol only needs access to routing information.

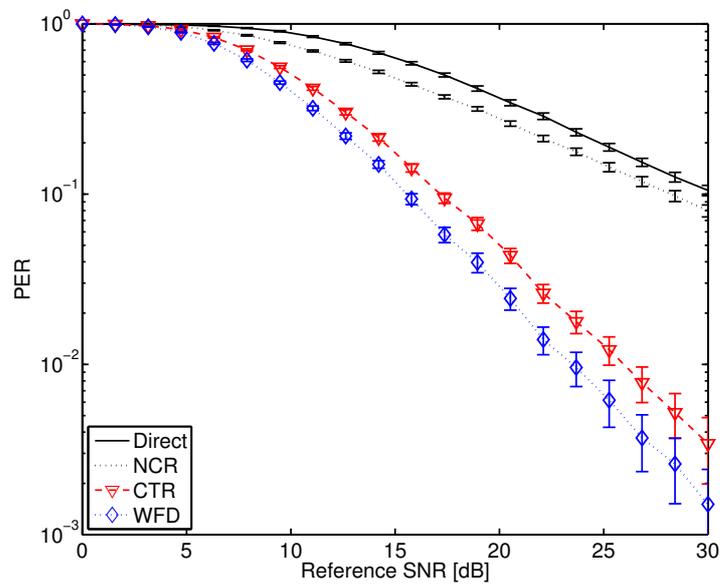


Figure 4.12.: Packet Error Rate (PER) of direct transmission, non-cooperative relaying, and cooperative relaying without and with routing information in presence of Rayleigh fading; all approaches use ideal signaling. Confidence intervals shown for 95 % confidence level.

#### 4. Mechanisms for cooperative unicasting

---

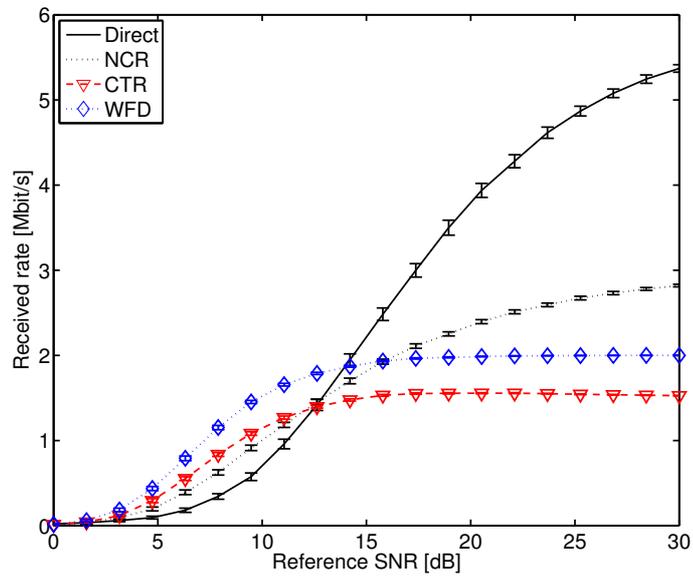
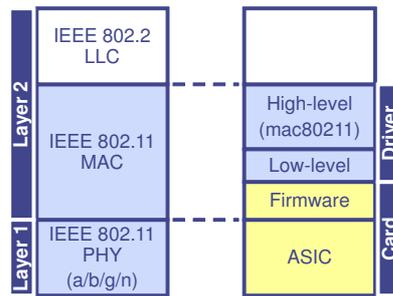


Figure 4.13.: Received rate of direct transmission, non-cooperative relaying, and cooperative relaying without and with routing information in presence of Rayleigh fading; all approaches use ideal signaling. Confidence intervals shown for 95 % confidence level.

Figure 4.14.: Architecture of *ath5k/mac80211* Linux WLAN driver

#### 4.2.2. Measurements using a modified IEEE 802.11a/g device driver

Based on the promising results with PSC in indoor environments (Section 2.2.1), I decided to modify an open-source Linux device driver for WLAN hardware with Atheros chipset. This is to show that cooperation diversity can be readily exploited with cheap consumer hardware available today.

Implementing cooperative relaying by modifying an open-source Linux WLAN driver has already been accomplished for basic three-node configurations by Korakis et al. [49]. Their proposed CoopMAC protocol exploits the multi-rate capability of the IEEE 802.11 standard. CoopMAC prefers a relay transmission over direct transmission if the achievable data rate via the alternative path using the relay is higher than direct transmission. To do so, nodes inspect all packets that they overhear to populate a cooperative neighborhood table. This CoopTable contains the rates used by the neighboring nodes' transmissions which serve as an estimate for the channel quality. A node that wants to send a packet uses this table to decide whether to transmit directly or via a relay. The work of Korakis et al. demonstrated that it is possible to implement cooperative relaying on cheap customer hardware, so I decided to implement my two-for-one cooperation protocol for multi-hop networks using commodity hardware as well.

Figure 4.14 shows the architecture of the *ath5k* Linux WLAN driver that I modified with the help of a student assistant. The *ath5k* consists of a hardware-specific driver that is tailored to the specific manufacturer's chipset, e.g., the Atheros AR2414 used for my experiments, and a hardware-independent driver that implements most of the IEEE 802.11 MAC automaton and management which works for all chipsets. Since most parts of the MAC protocol are implemented in the *mac80211* driver running on a machine that the WLAN card is connected to, the protocol can be easily modified to address an additional relay node and to retransmit overheard packets if the relay address matches. Unfortunately, some (time-critical) MAC functions cannot be changed as they reside in proprietary firmware that is closely tied with the underlying Application-Specific Integrated Circuit (ASIC) that encapsulates the physical layer. For this reason, it was impossible to modify RTS, CTS, and ACK. Especially, the ACK is automatically generated and always sent if the intended receiver correctly receives a data frame. We chose to implement a simplified version of the

#### 4. Mechanisms for cooperative unicasting

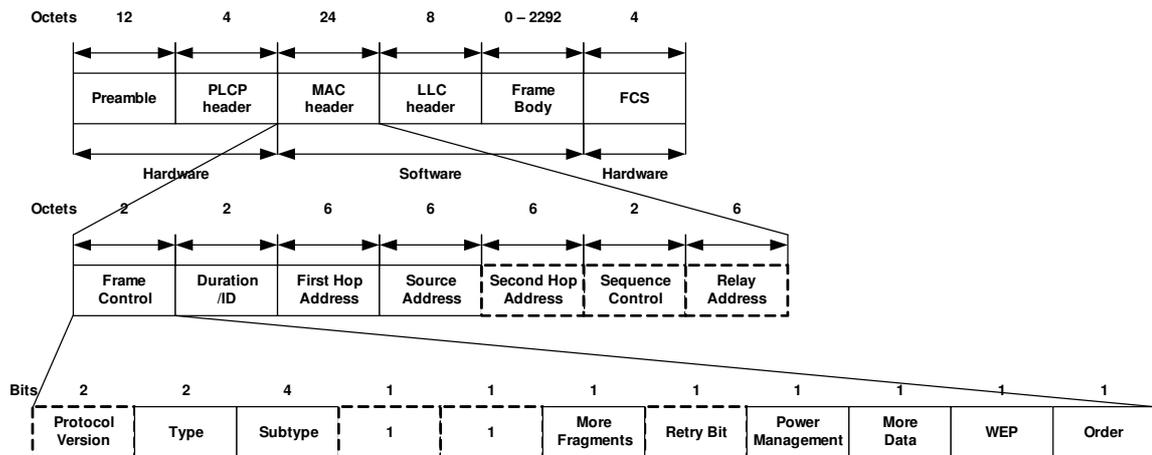


Figure 4.15.: Structure of a complete IEEE 802.11 MAC frame. Dashed lines indicate fields that needed to be modified for the implementation of two-for-one cooperation in the *ath5k* driver.

two-for-one cooperation protocol without modified handshakes as the generation of RTS, CTS, and ACK frames is not implemented in software and hence cannot be modified.

#### Integrating two-for-one cooperation into the Atheros AR2414 driver

To implement two-for-one cooperation we modified both the hardware-specific (*ath5k*) driver and the hardware-independent (*mac80211*) driver. Whenever the operating system has a packet to transmit, it will pass this packet to the *mac80211* driver encapsulated in a *socket buffer*. The socket buffer is an internal data structure of the network subsystem of the Linux kernel that represents a single packet [17]. The *mac80211* driver receives a link-layer packet with an 802.3 header that needs to be transformed into an 802.11 packet by the driver. This is the starting point for the first modification. We modify the 802.11 header by changing the default protocol version to a reserved version number, i.e., from 00 to 01, to distinguish conventional 802.11 packets from our cooperative packets. Normally, the hardware must drop all frames with a protocol version other than the default, but this can be easily disabled. Furthermore, we use the fourth address field in the 802.11 header to store the *relay address*. To do so, we pretend to use the wireless distribution system by setting both From DS and To DS fields to one. Initially, the relay address field is set to the broadcast address (all ones) because information about relays is gathered only at the hardware-specific *ath5k* driver for efficiency. Upon reception of a packet, *ath5k* learns about other nodes in the vicinity. By discarding unwanted packets without passing them to *mac80211* we avoid that the reception buffer unnecessarily fills up. Thus, only *ath5k* implements the so-called *relay table* which is similar to the CoopTable of CoopMAC [49].

After converting the 802.3 header to an 802.11 header for the packet to be transmitted, the packet passes through the remaining *mac80211* just as non-modified packets would.

Cooperative relaying requires most of the changes at the hardware-specific *ath5k*. Usually, the hardware automatically assigns the sequence number to outgoing packets. Since the relayed packets need to have the same sequence number as the original packet, we do not want the hardware to assign new sequence numbers to relayed packets. Since it is impossible to disable the sequence number assignment for specific frames only, in this case the relayed packets, we disabled the automatic assignment and use a software implementation in *ath5k* instead. Here, it is easily possible to skip relayed frames by checking the protocol version. This enables the relay to retransmit overheard packets without altering the packet and to detect duplicates at the destination.

For two-for-one cooperation, we do not want corrupt packets to be retransmitted by the source, so we set the retry counter such that packets are transmitted exactly once. For retransmission, our relay table lists suitable relays for each destination known. Although this is not needed in a static four-node configuration, it is required in a dynamic environment where users may join and leave the network at any time. When the relay table does not contain a suitable relay for the intended destination, we use the broadcast address as the relay address. Then, every node except for the destination that receives the packet, can set the relay address to its own MAC address and retransmit the packet. If the source of the packet overhears the relayed packet, it associates the relay address with the destination address using the relay table for later transmissions. We do not modify the source address to not interfere with hardware ACKs. Upon correct reception of a data frame, the hardware immediately acknowledges the frame by transmitting an ACK to the source. If the destination only receives the relayed packet, it still must send the ACK to the source and not to the relay. Since we cannot modify addresses, we use another reserved protocol version for relayed packets, i.e., 10. This helps us to distinguish at the destination whether the packet originated from the source (protocol version is 01) or the relay (protocol version is 10).

For the relay to overhear packets, two approaches are possible. First, the driver can be configured to work in *promiscuous mode*. In this case, the hardware does not filter any packets using the destination address and passes any packet to the driver. In this case, we can use the Basic Service Set Identifier (BSSID) field to store the address of the second hop that needs to be obtained from the routing protocol. Unfortunately, enabling promiscuous mode results in large overhead as all packets even from interfering users of different networks need to be processed at the driver. For efficiency, a second approach can be used instead. The hardware does not filter packets that are sent to the broadcast address. Thus, by storing the destination address in the BSSID field that, in the original implementation, contains the network's name, we can use the broadcast address as destination address instead. Upon reception, we restore the original destination address from the BSSID field. Since the hardware does not allow to use arbitrary MAC headers, with this approach, we must store the second hop address as the first six bytes of the payload, shifting the payload accordingly. For the experiments that follow, we avoid promiscuous mode for performance reasons.

The receiver path is far more complex than the transmission path within both drivers. We must make sure that the relay only retransmits packets when the relay address in the

#### 4. Mechanisms for cooperative unicasting

---

packet matches its MAC address or if the relay address is set to the broadcast address (i.e., the source does not know a suitable relay yet). According to SDF, we do not retransmit corrupt packets indicated either by a failed CRC or corrupt Physical Layer Convergence Procedure (PLCP) header. These packets are dropped right away. We use the existing code in *ath5k* to pass on the packet (again contained in a socket buffer) from *ath5k* to *mac80211*. The *ath5k* uses the retry bit to signal the *mac80211* whether to further pass on the packet to the operating system (retry bit unset) or whether to put it into the existing transmission queue for relaying (relay bit set). Right before we can put the packet into the queue, we need to make sure that *ath5k* will accept the packet as an outgoing packet. This is needed because the packet does not pass through the normal transmission path and therefore lacks required information, e.g., the transmission rate. Currently, we use a static transmission rate that can be set over the Linux *proc* interface (a configuration facility integrated into the file system that allows user applications to configure parameters of device drivers). Since the hardware automatically computes and appends the CRC to any frame to be transmitted, the relay's *ath5k* must remove the CRC of the overheard packet. Normally, the CRC is removed later on the receive path within *mac80211*.

Since the destination can now receive packets from source and relay, we need to drop duplicates. Like the relay, the destination does not accept corrupt packets. Whenever the destination receives a packet correctly, i.e., both PLCP header and CRC are correct, we check the protocol version to identify the packet's origin. If the packet originates from the source, we accept the packet and continue as normal after having stored the packet's sequence number and the time of its arrival. Thus, whenever we receive a packet from a relay, we compare its sequence number and time of arrival with the stored values to decide whether the corresponding packet from the source has already arrived correctly in which case the relay's packet can be dropped to avoid duplicates.

For the four-node configuration to be measured, we statically select the nodes so first and second hop are always known whereas in a dynamic scenario the routing protocol must provide this information to the MAC protocol. Here, I want to show which gains are actually possible with two-for-one cooperation using commodity hardware and therefore I neglect dynamic node selection.

#### Laboratory setup and measurement results

Figure 4.16 shows the experimental setup in the lab. Four notebooks are placed such that they form a symmetric diamond with the inter-node distances given in meters. In front of the destination, a disc with a diameter of 0.6 m rotates with a rotational speed of 30 rpm corresponding to a linear speed of 0.92 m/s. Half of the disc is shielded by a metal grid that also diffracts the electromagnetic waves. Due to the rotation, the propagation environment constantly changes and hence induces fading for the duration of the experiment.

Due to the lack of official documentation for the Atheros AR2414 chipset and the experimental *ath5k*, it was unclear which transmission power was used. To have at least a meaningful estimate, I decided to modify a WLAN card such that it can be connected to a

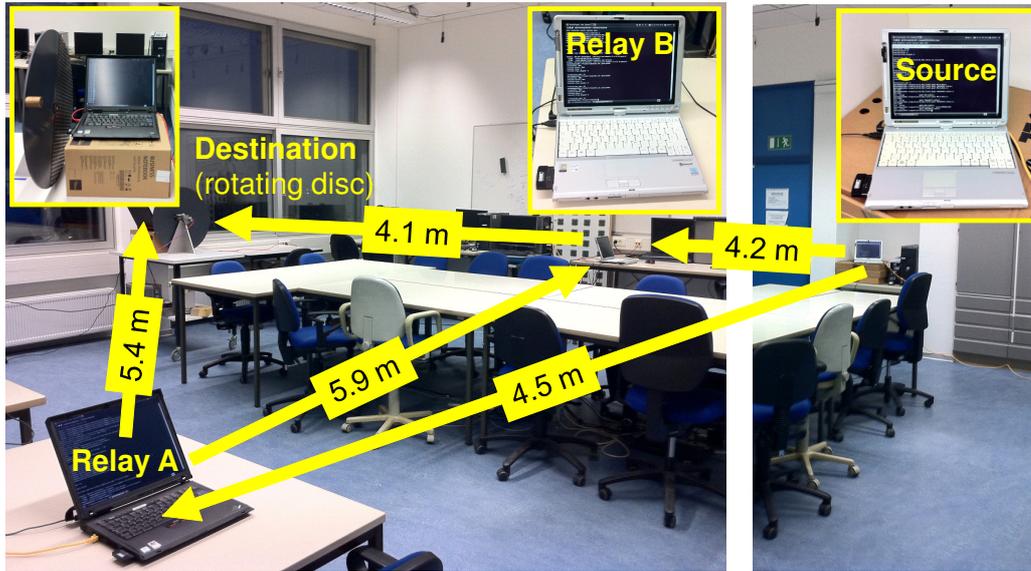


Figure 4.16.: Symmetric diamond consisting of four notebooks with Atheros WLAN cards (AR2414 chipset) and a rotating disc in front of the destination to emulate fading. Packets received on the direct path are ignored.

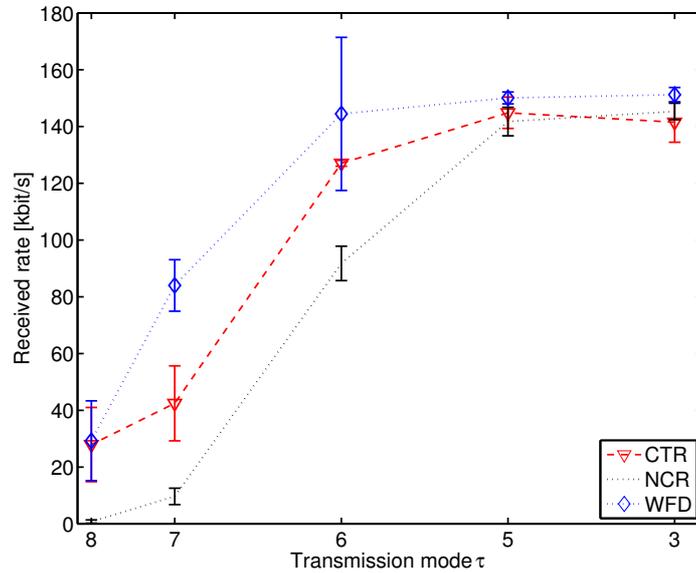
spectrum analyzer for calibration. For the measurements, I use a power setting where the transmit power (excluding antenna gains) was measured to be approximately  $-8$  dBm. Such a low transmission power is desirable for indoor measurements with inter-node distances between 4 m and 6 m.

Instead of varying a reference SNR, here I chose to modify the modulation and code rate used for transmission. Higher modulation and code rates require better SNR at the receiver for successful decoding. Since the mean SNR does not change when varying the transmission mode, a higher transmission rate is comparable to a lower SNR since the required SNR for reception increases. To assure static rate selection, I disabled the rate adaptation algorithm used in the driver.

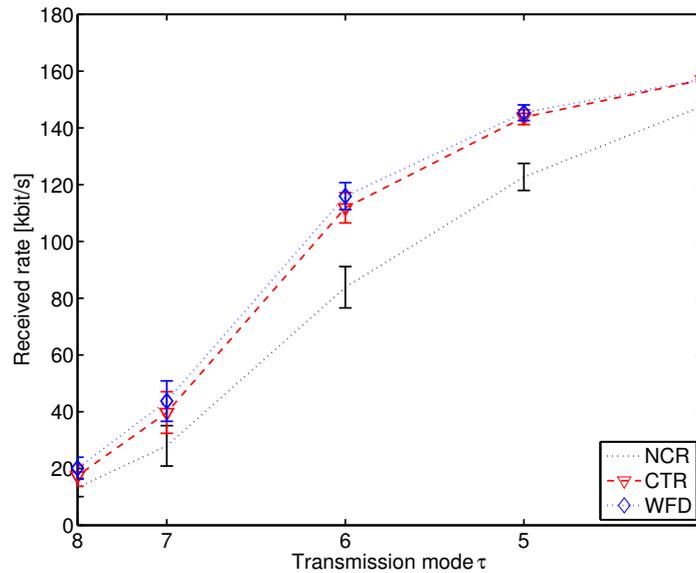
I measure NCR, CTR, and WFD for the transmission modes 3, 5, 6, 7, and 8 (corresponding to transmission rates 12, 24, 36, 48, and 54 Mbit/s) as summarized in Table 4.2 by constantly transmitting 50,000 packets containing 200 bytes payload every 0.01 s, resulting in a data rate of 160 kbit/s as an example for voice-like traffic. Figure 4.17(a) shows the measured data rate at the receiver for varying transmission modes (and hence varying modulation and code rate as shown in Table 4.2). Due to this low rate traffic model, the network never operates in saturated conditions.<sup>4</sup> Therefore, all approaches achieve the same data rate at the lowest transmission rate, corresponding to the highest SNR. Note that this

<sup>4</sup>The choice of a low transmitter data rate was also motivated by a practical reason. The modified *ath5k* driver stops working as soon as its transmit buffers fill up, which does not occur at low rates such as the one chosen for the experiment.

#### 4. Mechanisms for cooperative unicasting



(a) Experimental results



(b) Simulation results

Figure 4.17.: Experimental and simulation results for received rate vs. transmission rate for the voice-like load model comparing NCR to the proposed MAC protocol with four-nodes setups. Confidence intervals shown for 95 % confidence level.

does not restrict the usefulness of the results in Figure 4.17(a) since one would not cooperate at high SNR anyway. Both CTR and WFD outperform NCR due to their increased diversity order of two at all nodes. The performance of CTR is comparable to WFD except for transmission mode 7. While in saturated conditions, WFD indeed performs better than CTR, the simulation results depicted in Figure 4.17(b) for the same traffic model show that the data rates achieved by CTR and WFD at the receiver should be similar for all transmission modes.

The simulation results show the same characteristic behavior of NCR, CTR, and WFD also observed in the experiments. Both CTR and WFD outperform NCR due to their increased diversity order. Although both of them achieve the same data rate at the receiver, it is important to understand that WFD spares an entire relay's transmission compared to CTR at the same diversity order. In conclusion, WFD consumes less space during cooperative relaying than CTR and hence causes less interference. This makes WFD superior to CTR when it comes to improve spatial reuse in the network. The experimental results also demonstrate that two-for-one cooperation can already be built with commodity hardware available today.

### 4.3. Conclusions

This chapter proposed two orthogonal techniques that mitigate multiplexing loss to improve the efficiency of cooperative relaying. First, the performance evaluation of rate adaptation in the CTR leads to the following conclusions.

1. In symmetric scenarios, rate adaptation cannot compensate for relaying's multiplexing loss. Thus, with symmetric links, even with perfect cooperation and rate adaptation, only insignificant throughput gains are possible.
2. Forcing the relay to use the same rate as the source has adverse effects on the received rate for two reasons: On the one hand, if the relay's channel to the destination is better than that of the source, using the source's low adapted rate causes an unnecessarily long retransmission. If, on the other hand, the relay experiences a worse channel than the source, then using the source's adapted rate also at the relay causes an unnecessarily error-prone retransmission. Both cases are costly and limit the achievable throughput. Thus, especially for low speeds and slow fading, rate-per-link adaptation should be used with cooperative relaying.
3. The configuration in which rate adaptation is performed greatly impacts the overall performance. Direct transmission with rate adaptation is sufficient for symmetric configurations, but as soon as the source-relay link is favored, cooperative relaying with rate-per-link adaptation delivers substantial gains in rate for low transmission powers. For relay deployment, a chain configuration should be used as combined

#### 4. Mechanisms for cooperative unicasting

---

diversity and power gains are more significant and hold for a larger range of transmission powers.

4. For higher speeds, combining rate adaptation with cooperative relaying is not beneficial anymore as prediction errors are more costly. Note that for fast fading the fading channel approaches an AWGN channel where cooperative relaying itself becomes superfluous.

Second, two-for-one cooperation applies cooperative relaying in a more efficient way to multi-hop networks than simple strategies that use cooperative relaying for every hop with local knowledge only. The conclusions are:

5. Without routing information, potential relays cannot decide whether they can form a diamond configuration, nor can nodes two hops away decide whether they should buffer data frames in advance. Therefore, routing information is the key to mitigate the multiplexing loss that restricts gains even at low SNR.
6. I proposed a routing-informed MAC protocol for two-for-one cooperation. The proposed protocol shows increased performance both in simulations and experiments on real WLAN hardware that is already available to end-users today at low cost. At a given PER, cooperation diversity gains either help to reduce transmit power, which improves spatial reuse and therefore network capacity, or to increase the data rate of individual connections. Using two-for-one cooperation, the relay's transmission can be more efficiently exploited, thereby improving spatial reuse further.

All in all, wireless multi-hop networks can apply cooperative relaying to make unicasts more efficient. In the next chapter, I show that for broadcasts, multiplexing loss is no longer a penalty anymore, making broadcasts the ideal scenario for cooperative relaying.

## 5. Mechanisms for cooperative broadcasting

*Present broadcast approaches for wireless multi-hop networks distribute packets quickly to all nodes by constructing small broadcast sets, thereby reducing the number of forwarding transmissions. While these sets are appropriate in non-fading environments, I show that they have a low delivery rate with fading. As a solution, I incorporate the Rayleigh fading model directly into broadcast set construction to re-obtain complete delivery with high probability. To still achieve low delivery time, individual nodes combine transmissions to exploit cooperation diversity.*

In wireless multi-hop networks, broadcasting a packet to all nodes in the network is frequently used. Ad hoc routing protocols rely on broadcasts for discovering routes between network nodes. For example, the Ad Hoc On-Demand Distance Vector (AODV) routing protocol lets a node broadcast a route request packet whenever the node does not have a route to the intended destination in its routing tables [71]. A node that receives such route request responds with a route reply if it either is the intended destination or if it knows a route to the destination. These responses are unicast replies that a node sends to where it previously received the route request from.

Broadcasting can easily be implemented by *blind flooding*: any node that receives the message will retransmit it. Unfortunately, blind flooding leads to redundant transmissions that increase contention and the number of collisions in a network – the so-called *broadcast storm* problem [67]. Broadcasting becomes more efficient by determining a small subset of connected nodes as forwarders. Figure 5.1(a) shows an example of a broadcast set for a small network. Thick edges connect nodes that belong to the broadcast set.

It is characteristic for these broadcast sets that its nodes can reach all remaining nodes in the network [79, 80]. Letting only nodes of the broadcast set retransmit broadcast packets limits the number of retransmissions and, as a consequence, reduces the time required until all nodes in the network have received the broadcast message – the time to do so is the *broadcast latency*. A good broadcast set distributes the broadcast packet to all nodes in the network, so-called *full delivery*, with as low a broadcast latency as possible.

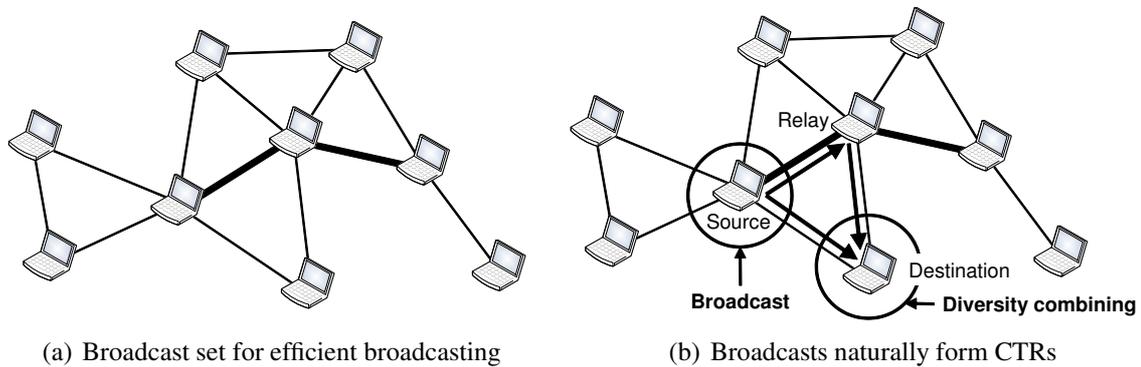


Figure 5.1.: A broadcast set reduces the number of retransmitting nodes for broadcasting in a wireless network. During retransmissions, Cooperative Triangles (CTRs) form naturally.

## 5.1. Problem statement

In Chapter 2 I described that multiplexing loss is the major cost of cooperative relaying and in Chapter 4 I proposed techniques to mitigate this loss for unicasts in multi-hop networks. By moving from unicasts to broadcasts, multiplexing loss does not need to be mitigated anymore. Figure 5.1(b) shows how a CTR naturally forms while the broadcast packet travels along the broadcast set. Since nodes of the broadcast set need to retransmit packets anyway to deliver the broadcast packet in the entire network, the same packet may naturally arrive multiple times at particular nodes. In this case, a cooperative diversity gain arises that, by suitable protocol design, does not prolong the broadcast latency, not even at high SNRs. To put it in a nutshell, not exploiting cooperative diversity in broadcast networks wastes cooperative diversity gains. It is the goal of this chapter to develop an algorithm for finding broadcast sets that, by harnessing cooperative diversity, make broadcasts more reliable, yet keeping broadcast latency as low as possible.

Since literature on broadcast sets for multi-hop networks is vast, I first summarize the most important related work in Section 5.2. I then propose in Section 5.3 a new heuristic that, by incorporating a probabilistic model into node selection, avoids operating links of broadcast sets at their limits. This allows to transform the robustness gained from cooperation diversity into smaller broadcast sets with low latency. I show by simulations that neither a probabilistic nor a cooperative approach alone achieves low latency broadcasts with full delivery. Only their *combination*, the Probabilistic Cooperation Diversity Broadcast (PCDB), shows to be fading-resistant with low latency.

## 5.2. Flexible framework for broadcast set construction

For arbitrary networks, finding a broadcast set in a wireless network that optimizes some network metric such as data rate, broadcast latency, or energy consumption is an NP-complete or NP-hard problem. I first give an overview of related work in the field of broadcast sets, concluding with a taxonomy of current approaches. This overview helps in selecting important approaches for comparison and developing a unified framework for simulation that allows for a fair comparison of these approaches with the ones I will develop in Section 5.3.1.

### 5.2.1. Taxonomy of broadcast set construction heuristics

If one models a network as a graph, an edge between two nodes indicates that these nodes can successfully communicate with each other. A *dominating set* is a broadcast set where any node that is *not* in the dominating set connects to at least one node in the dominating set. Assuming perfect links, this property assures that when all nodes in the dominating set have transmitted, *all* nodes of the network (even those not in the dominating set) will have received the transmission. Furthermore, a dominating set should also be *connected*, i.e., there exists a path from any node to any other node in the set. Connectivity hence avoids isolated subsets. Additionally, it does not matter which node initiates the broadcast to achieve full delivery. Finally, to achieve a small broadcast latency, the number of nodes in the Connected Dominating Set (CDS) that retransmit the broadcast packet should be small. Ideally, only the CDS with the minimum number of nodes should be used as the broadcast set, a so-called Minimum Connected Dominated Set (MCDS). The seminal work of Guha and Khuller [28] showed that constructing an MCDS is an NP-complete problem for which a simple yet powerful graph coloring heuristic exists.

Such a graph-oriented view originates from wired networks where the reliability of particular links is much higher. The broadcast channels of wireless networks cause interference which is hard to model in a graph, but to some extent a graph also applies to wireless networks. However, as I will show in this chapter, delivery ratios strongly depend on the model of the wireless channel that determines whether an edge connects two nodes in the graph. The simplest such model is the unit disk graph [16]. In this model, a packet is correctly received if the receiver is within a circle centered at the transmitter of some radius. This circle denotes the transmission range as its radius depends on the required SNR at the receiver to correctly decode the packet. In wireless networks, the only deterministic factor is path loss due to free space propagation as it only depends on distance. Therefore, the transmission range is the maximum distance where nodes can at least reach an SNR above a fixed decoding threshold  $\gamma_{\text{th}}$ . Figure 5.2 shows an example how to construct a CDS using Guha and Khuller's heuristic (GK) with the unit disk graph model.

The GK heuristic colors nodes in a graph using one of three colors.

**White** A white node is *uninformed*, i.e., the node does not possess the message to be

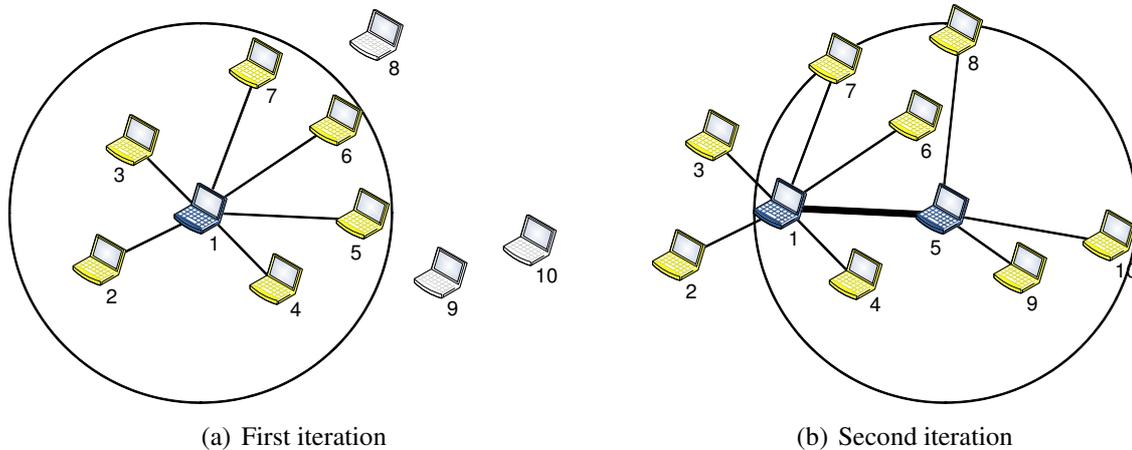


Figure 5.2.: Optimistically constructing a CDS

broadcast.

**Gray** A gray node is *informed*, i.e., it possesses the message to be broadcast. Gray also indicates that the node is not part of the broadcast set. The heuristic only elects nodes from the set of gray nodes into the broadcast set because only informed nodes can retransmit the message.

**Black** A black node is informed and it belongs to the broadcast set. Ideally, the set of black nodes should be minimal.

Initially, all nodes are white except for the start node  $s$  which is assumed to initiate the broadcast and hence is colored gray. Any node inside the set of gray nodes may retransmit the message. Thus, in every iteration of the heuristic, the set of gray nodes is searched for a particular node  $v$ . This node  $v$  should have the most uninformed white nodes within its transmission range. This greedy choice of  $v$  should, in every iteration, inform the largest possible subset of white nodes, hopefully leading to a small broadcast set. At the end of an iteration,  $v$  changes its color to black and all its white neighbors within transmission range become gray. The heuristic repeats for as long as there are still white nodes left.

Figure 5.2(a) illustrates the first iteration with node 1 being the start node of the network graph. Nodes 2 to 7 are within the unit disk transmission range and become informed. Therefore, the heuristic colors these nodes gray, making them potential candidates for the broadcast set. Nodes 8, 9, and 10 are outside the transmission range of node 1. They cannot correctly decode the message and remain uninformed; their white color does not change. At the end of the first iteration, the heuristic chooses node 1 into the broadcast set by coloring it black.

In the second iteration, the heuristic picks among the set of gray nodes (i.e., informed but not part of the broadcast set) that node with the most white (i.e., uninformed) neighbors.

This is node 5 in the example shown in Figure 5.2(a). The remaining nodes 8, 9, and 10 become gray (i.e., informed) and node 5 becomes part of the broadcast set; its color changes to black. Since no more white nodes are left, the heuristic terminates after two iterations.

In the toy example of Figure 5.2, two iterations suffice to color the entire graph. The CDS consists of nodes 1 and 5. During network operation, only nodes part of the CDS retransmit a broadcast packet. Indeed, this small subset suffices to inform the entire network. Assume, at some point in time, node 7 initiates a broadcast. Since edges in the unit disk graph model are symmetric (the distance between two nodes is always the same no matter from which node one starts measuring), node 1, among others, receives the broadcast, but only node 1 retransmits it. After this transmission, nodes 1 to 7 have received the packet and, finally, node 5 of the CDS retransmits the broadcast to the remaining uninformed nodes.

For this example, a CDS of size two suffices to achieve full delivery using the unit disk graph model, so at most three transmissions inform the entire network. In the case of blind flooding, the number of transmissions is ten, since any informed node will retransmit the message once. Consequently, the use of a CDS for broadcasting spares transmissions, making more efficient use of the wireless channel.

Throughout this chapter, I will get back to this toy example and use it to illustrate different approaches. Additionally, I will verify the observations of these toy examples for large random networks by simulations.

Chou et al. [15] propose an efficient heuristic for constructing CDSs in wireless networks with multi-rate links (e.g., IEEE 802.11a/b/g). Starting at a broadcast initiator, the algorithm constructs a broadcast set as follows. As long as the broadcast has not covered all nodes yet, the algorithm incrementally selects an already covered node  $v$  and rate  $r$  which maximize the product of that rate  $r$  and the number of additional nodes covered by  $v$  (assuming node  $v$  transmits the broadcast at rate  $r$ ). The algorithm then adds the additionally covered nodes to the set of covered nodes. The algorithm then extends the broadcast set by a link from  $v$  to the newly added nodes. Once the broadcast set is constructed, the number of transmissions that a node has to perform and the schedule of these transmissions are determined next.

Multi-rate CDSs benefit from the fact that the first transmission is done at a high rate to quickly enable spatial reuse in some part of the network. Then, the same node lowers its rate to disseminate the broadcast in the other parts of the network that cannot be reached at the high rate. Chou et al. propose in [14] how to schedule the same node to transmit at different rates.

In multi-rate wireless networks, a conflicting tradeoff exists. Letting users in the CDS broadcast with high rates reduces the time required for a *single* transmission, but comes at the cost of smaller coverage. Thus, the number of retransmissions required increases and so does the size of the broadcast set which, in turn, increases the time required for *all* broadcasts. Since broadcast sets should be small to minimize broadcast latency, one is faced with mutually conflicting goals [59]. Exploiting cooperation diversity reduces

the size of broadcast sets further by transforming diversity gains into larger transmission ranges.

Wu et al. [95] first extended the definition of dominating sets to capture cooperation diversity. They define an Extended Dominating Set (EDS) as a subgraph of some network graph. Nodes are either in the EDS or they are neighbors of nodes in the EDS (which also applies to CDS) or, additionally,  $k$ -quasineighbors of nodes in the EDS (which captures cooperation diversity). A node  $v$  is a quasineighbor of  $u$  if it can partially receive packets from  $u$ , and  $v$  is a  $k$ -quasineighbor (with  $k > 1$ ) if it needs  $k$  partial receptions of the same packet from  $k$  distinct nodes to reconstruct the packet. The EDS is called *weakly connected* if there exists some node in the EDS whose packet transmission causes all other nodes in the network to receive that packet. The EDS is called *strongly connected* if for all nodes in the EDS their packet transmission causes all other nodes in the network to receive that packet. Wu et al. showed that constructing EDSs is NP-complete and they proposed heuristics to construct Extended Connected Dominating Sets (ECDs) for *single-rate networks*. While Wu et al. use a binary decision model, I use a continuous energy model to decide upon correct receptions instead, where energies may accumulate per node as in [62]. In the binary model, transmissions either succeed (for the non-cooperative case, when a node is in neighbor range or, for the cooperative case, when  $k$  nodes are in quasineighbor range) or fail.

To the best of my knowledge, current approaches for broadcasting or broadcast set construction do not use the probability of successful communication based on combining transmissions. The Extended Minimum CDS (E-MCDS) approach by Wu et al. [95] uses the average aggregated energy as opposed to its random distribution. Ideas like  $m$ -covered nodes [83] come closer to my work, but only consider a simple form of selection combining, rather than the provably superior MRC. With selection combining, there is no need to keep track of the aggregated energy for a packet. In a sense, multi-rate heuristics for broadcast sets do keep track of signal levels, but again only do a crude form of selection combining at multiple rates [15] or do not consider combining at all [92]. Cooperative schemes focusing on power control exist (e.g., by Hong and Scaglione [36]), but focus on different performance metrics.

### 5.2.2. A unified framework for analysis and simulation

Since I want to compare my heuristics with related work, I need a single, unified framework that allows me to consistently analyze and simulate the related approaches along with those in Section 5.3.1. For this, I first propose a suitable unified model and show how the existing models can be mapped to it. Second, with Algorithm 5.1 I present a generic algorithm for constructing broadcast sets that can be easily modified to implement specific heuristics.

Table 5.1.: Taxonomy of broadcast set construction algorithms in wireless ad hoc networks

Reference	Set	Rates	Coop.	Scope	Metric
[28]	CDS	single	no	centralized	number of nodes
[93]	Tree	single	no	centralized	total power
[40]	Tree	single	no	distributed	total power
[15]	CDS	multiple	no	centralized	latency
[92]	CDS	multiple	no	distributed	latency
[95]	ECDS	single	yes	both	number of nodes
[62]	Schedule	single	yes	both	total power, network lifetime

### Feasible parameters for simulating the physical layer

Guha and Khuller [28] assume unit disk graphs to decide which nodes are neighbors. Chou et al. [15] also adopt unit disk graphs, but since they use multiple rates, they have distinct unit disks for every rate (I return to this later in the context of Table 5.2). Cooperation in [95] introduces an extended unit disk graph whose radius depends on the number of cooperating nodes. These three models are all based on unit disk graphs and, therefore, allow to construct broadcast sets as subgraphs of a network.

The ratio of signal energy per bit to noise power density per Hertz is given by [82]

$$\frac{E_b}{N_0} = \frac{P_r}{N_0 R} \quad (5.1)$$

where  $P_r$  is the received power,  $R$  the transmission rate in bits/Hz, and  $N_0$ , the amount of thermal noise, depending on the temperature (e.g.,  $N_0 = 4.32 \cdot 10^{-21}$  J at 40°C).

I adopt the two-ray propagation model used in [15, 91] for ease of comparison, where the received power is given by [33]

$$P_r = P_t G_t G_r \left( \frac{h_t h_r}{d^2} \right)^2 \quad (5.2)$$

where  $P_t$  denotes the transmit power,  $G_t$  ( $G_r$ ) denotes the transmit (receive) antenna gain and  $h_t$  ( $h_r$ ) the corresponding antenna height, and  $d$  is the distance between transmitter and receiver. Substituting (5.2) into (5.1) and, for simplification, assuming  $G_t = G_r = 1$  and  $h_t = h_r = 1$  m yields the following relationship:

$$\frac{E_b}{N_0} = \frac{P_t}{N_0 R d^4} \quad (5.3)$$

This equation allows us to determine the minimum  $E_b/N_0$  threshold required for successful reception. Again, for ease of comparison, we adopt the IEEE 802.11b model used in

## 5. Mechanisms for cooperative broadcasting

Table 5.2.: Transmission rates with their maximum transmission ranges

Rate [Mbits/s]	Old range [m] (as in [15, 91])	$E_b/N_0$ [dB]	New range [m] ( $\frac{E_b}{N_0} = 25.9$ dB)	Coop. range [m] ( $q(2) = 1.19$ )
1.0	483.0	26.3	486.8	579.0
2.0	370.0	27.9	409.4	486.8
5.5	351.0	24.4	317.9	378.1
11.0	283.0	25.2	267.3	317.9

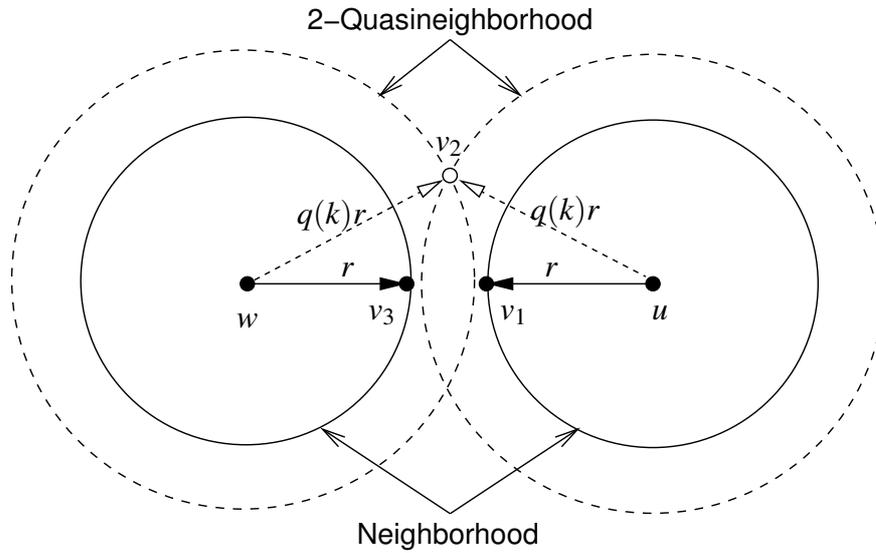


Figure 5.3.: Extended graph model used by Wu et al. [95]

[15, 91], which I reproduce in Table 5.2. For the transmission ranges specified in [15, 91], I compute the associated  $E_b/N_0$  thresholds using (5.3). I need a single  $E_b/N_0$  threshold to be able to accumulate energy from several transmissions with different rates, therefore I use the mean of the  $E_b/N_0$  thresholds from Table 5.2 (which is 25.9 dB). This, of course, yields new transmission ranges (found by rearranging (5.3) for  $d$ ) that I also state in Table 5.2 for completeness.

However, in real systems there is no single  $E_b/N_0$  threshold as most systems adapt their modulation and code rate according to the current channel state (Section 4.1). Unfortunately, different models are used in [15, 40, 62, 93, 95] for single-rate vs. multi-rate and non-cooperative vs. cooperative approaches. Assuming a single threshold is a simplification that allows one to analyze these different approaches with the *same* framework.

For their EDS using cooperation diversity, Wu et al. [95] use an extended geometric graph model, which is illustrated in Figure 5.3. As usual, two nodes  $u$  and  $v_1$  (or similarly  $w$  and  $v_3$ ) can receive each other's packets successfully (they are said to be neighbors

then) as long as they are within normal transmission range  $r$ , i.e.,  $d(u, v_1) \leq r$ . Outside the normal transmission range, signals may not be decoded on their own, but still they are detectable and they can be buffered for later combining, i.e., cooperation. The authors now assume an extended transmission range  $rq(k)$  with  $1 < q(k) \leq 2$  where a packet can still be successfully received at a so-called  $k$ -quasineighbor  $v_2$  if the packet has been transmitted  $k$  times.  $q(k) > 1$  is the *cooperative stretch* of the unit disk graph depending on the number of cooperating quasineighbors. Figure 5.3 shows an example for  $k = 2$  where nodes  $u$  and  $w$  cooperatively reach  $v_2$  and  $r < d(u, v_2) \leq rq(2)$  and  $r < d(w, v_2) \leq rq(2)$ . The factor  $q$  depends on the number of cooperating nodes, and it is monotonically increasing in  $k$ . Wu et al. assume the existence of the cooperative stretch  $q$  although they do not explicitly name it. In their entire discussion, they assume  $k = 2$  and  $q(2) = 2$ . While  $k = 2$  is a feasible assumption, the choice of  $q(2) = 2$  seems quite arbitrary. In fact, I now show that for  $k = 2$  using the two-ray propagation model,  $q(2) = 1.19$ , and  $q(k) = 2$  holds not until  $k = 16$  (in the same way, it can be shown that  $q(2) = 1.41$  for free space propagation which still is not as nice as  $q(k) = 2$ , in free space requiring  $k = 4$  quasineighbors). This is important because I later want to do an unbiased comparison and hence I need to use a realistic value of  $q$  instead of an arbitrary guess.

Assuming MRC, cooperative transmission is modeled as the superposition of received energy ratios from every quasineighbor  $u_i$  at the destination node  $v$ , thus (using (5.3) for the individual transmissions)

$$\frac{E_b}{N_0} = \sum_{i=1}^k \frac{P_{t_i}}{N_0 R_i d(u_i, v)^4}. \quad (5.4)$$

Assuming equidistant nodes, same power and rate allocation, (5.4) simplifies to

$$\frac{E_b}{N_0} = \frac{k P_t}{N_0 R (q(k) r)^4}. \quad (5.5)$$

Solving (5.5) for  $q$  and expressing  $r$  using (5.3) (simply solve for  $d$  and insert into  $r$ ) leads to the simple relation  $q(k) = \sqrt[4]{k}$ , which for  $k = 2$  gives the aforementioned  $q(2) = 1.19$  (for free space, the same derivation holds, albeit with different path loss exponent, leading to  $q(k) = \sqrt{k}$ ). The last column in Table 5.2 states the transmission ranges of the 2-quasineighborhood for completeness. All in all, these computations justify the choice of physical layer parameters summarized in Table 5.3.

### Skeleton for constructing broadcast sets

I adopt the idea of node sets analogously to GK. Nodes can be in one of three possible states: *white* for uninformed nodes, *gray* for informed nodes not in the broadcast set, and *black* for informed nodes in the set. Initially, all nodes are white except for the start node that initiates the broadcast, which starts off gray. Heuristics for broadcast set construction only differ in which of the gray nodes they promote into the set in each iteration and which

## 5. Mechanisms for cooperative broadcasting

---

metric they use for deciding this. Thus, all heuristics in this section share the skeleton given by Algorithm 5.1.

---

### Algorithm 5.1 Skeleton for broadcast set construction heuristics

---

**Require:** A node set  $V$ , a start node  $s$ , and non-negative random variables  $X_{uv}$  for all pairwise distinct  $u, v \in V$ .

**Ensure:**  $V = V_{\text{black}} \dot{\cup} V_{\text{gray}} \dot{\cup} V_{\text{white}}$ .

- 1:  $V_{\text{black}} \leftarrow \emptyset$
  - 2:  $V_{\text{gray}} \leftarrow \{s\}$
  - 3:  $V_{\text{white}} \leftarrow V \setminus \{s\}$
  - 4: **repeat**
  - 5:    $v_{\text{best}} \leftarrow \arg \max_{v \in V_{\text{gray}}} \{\text{metric}(v, V_{\text{white}})\}$
  - 6:    $V_{\text{gray}} \leftarrow V_{\text{gray}} \setminus \{v_{\text{best}}\}$
  - 7:    $V_{\text{black}} \leftarrow V_{\text{black}} \dot{\cup} \{v_{\text{best}}\}$
  - 8:    $V_{\text{promoted}} \leftarrow \text{promoteWhite}(v_{\text{best}}, V_{\text{white}}, V_{\text{black}})$
  - 9:    $V_{\text{white}} \leftarrow V_{\text{white}} \setminus V_{\text{promoted}}$
  - 10:    $V_{\text{gray}} \leftarrow V_{\text{gray}} \dot{\cup} V_{\text{promoted}}$
  - 11: **until**  $V_{\text{white}} = \emptyset$  or  $V_{\text{gray}} = \emptyset$
- 

This skeleton describes a family of greedy heuristics. Its  $\text{metric}(\cdot)$  function as well as its white-node promotion strategy  $\text{promoteWhite}(\cdot)$  define its particular behavior and allow customization. In line 5, the metric determines which gray node to adopt into the broadcast set. The heuristic assumes that this chosen node transmits, possibly resulting in more informed nodes in the neighborhood. The second function  $\text{promoteWhite}(\cdot)$  determines which uninformed neighbors become informed; the heuristic changes their color from white to gray.

I now define both metric and promotion strategy for the GK and E-MCDS heuristics. To ease notation, I define the set of *uninformed* direct neighbors of some node  $v$  as

$$N(v) := \{w \in V_{\text{white}} : \Gamma_{v,w} \geq \gamma_{\text{th}}\}, \quad (5.6)$$

and the set of *uninformed*  $k$ -quasineighbors of  $v$  as

$$\text{QN}(k, v) := \{w \in V_{\text{white}} : \gamma_{\text{th}} > \Gamma_{v,w} \geq q(k)^{-\alpha} \gamma_{\text{th}}\}. \quad (5.7)$$

The cooperative stretch  $q(k) > 1$  captures the extended range of  $k$ -quasineighbors. This parameter needs to match the propagation environment and the combining strategy [95]. As discussed above, if  $q$  is too large, the heuristic assumes nodes to aggregate more energy than is actually available.

Algorithm 5.2 shows the functions for GK [28]. GK always selects the node with the most uninformed neighbors, i.e., the cardinality of the node set given by (5.6). Since  $N(v)$  only compares the *mean* SNR  $\Gamma$  to the threshold, GK estimates correct reception solely

via path loss. At this time, the heuristics do not know the behavior of the MAC layer yet and therefore they do not know how the MAC will schedule packets. For this reason, GK indeed uses SNR (instead of SINR) and neglects interference while constructing the broadcast set. In the simulations of the actual broadcast that follow, nodes are subject to interference. Interference may then affect the protocol's performance, e.g., during MAC contention or collisions.

---

**Algorithm 5.2** Functions for Guha and Khuller's heuristic

---

**Function**  $\text{metric}(v, V_{\text{white}})$ :

1: **return**  $|\mathbf{N}(v)|$

**Function**  $\text{promoteWhite}(v_{\text{best}}, V_{\text{white}}, V_{\text{black}})$ :

2: **return**  $\mathbf{N}(v_{\text{best}})$

---

E-MCDS [95] is a cooperative variant of GK and also defines neighbors and quasineighbors using average path loss. Each broadcast node contributes 1 to all its neighbors – nodes  $v$  and  $w$  are neighbors of node  $u$  in Figure 5.4(a) – and  $1/2$  to its 2-quasineighbors – nodes  $x$  and  $z$  are 2-quasineighbors of node  $u$ . The *effective contribution* of  $u$  to some other node  $v$  is  $u$ 's contribution to  $v$  before the signal energy of  $v$  reaches 1 [95]. Consider Figure 5.4(b) for an example. To determine the effective contribution of  $v$  one needs to inspect only its white neighbors and quasineighbors as their signal energy is still below 1. In the example, the effective contribution of  $v$  to  $x$  is  $1/2$  because node  $x$  has already reached a signal energy of  $1/2$  due to  $u$ 's previous transmission in Figure 5.4(a). The effective contribution of  $v$  to  $y$  is  $1/2$  because node  $y$  is a 2-quasineighbor of  $v$  and has previously not received any signal energy as it is neither a neighbor nor a quasineighbor of  $u$ . The Total Effective Contribution (TEC) of node  $v$  is the sum of the effective contribution of  $v$  to its neighbors and quasineighbors [95], which is 1 in the case of Figure 5.4(b). Alternatively, the TEC of node  $z$  according to Figure 5.4(c) is 0 as it does not have any white neighbors or quasineighbors.

The metric in Algorithm 5.3 denotes the TEC of signal energy to all white neighbors and quasineighbors of some node  $v$ . Two steps are necessary to determine the TEC. First, consider the effective contribution to all white neighbors of  $v$ , i.e.,  $\mathbf{N}(v)$ . Assume  $u \in \mathbf{N}(v)$ , then node  $u$  cannot be a neighbor of a black node; if it were, node  $u$  would have been informed already which contradicts  $u \in \mathbf{N}(v)$ . Thus,  $u$  can only have black quasineighbors which might have already contributed to  $u$ 's signal energy. For convenience, I define

$$B(k, u) := |\{b \in V_{\text{black}} : u \in \text{QN}(k, b)\}| \quad (5.8)$$

as the number of black nodes that have  $u$  as their  $k$ -quasineighbor (and from which  $u$  receives a contribution). Since white nodes may already have aggregated energy in multiples of  $1/k$ , the fraction  $(k - B(u))/k$  denotes the missing energy until node  $u$  becomes

## 5. Mechanisms for cooperative broadcasting

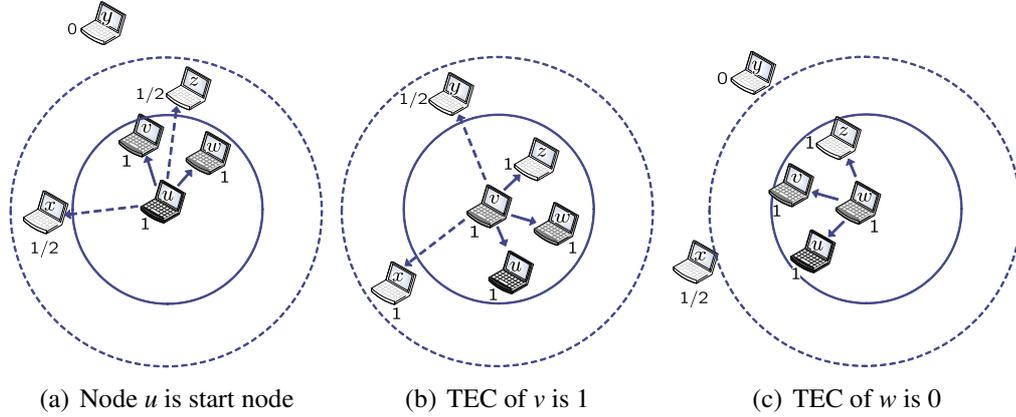


Figure 5.4.: Total Effective Contribution (TEC) of nodes  $v$  and  $w$  after start node  $u$  has been chosen into the broadcast set. Solid lines represent the direct neighborhood and dashed lines represent the 2-quasineighborhood. Nodes are annotated with their contribution.

informed. Hence, the TEC of  $v$  to all its white neighbors is

$$\sum_{u \in N(v)} \frac{k - B(k, u)}{k}. \quad (5.9)$$

Second, consider the effective contribution to all white  $k$ -quasineighbors of  $v$ , i.e.,  $QN(k, v)$ . For any  $u \in QN(k, v)$  it must hold that at least  $1/k$  is missing as it is the smallest fraction of signal energy that a node can receive. Hence, the TEC of  $v$  to all its white quasineighbors is

$$\sum_{u \in QN(k, v)} \frac{1}{k}. \quad (5.10)$$

Since E-MCDS greedily chooses that gray node into the broadcast set that achieves the maximum TEC, the metric to use in Algorithm 5.3 is the sum of (5.9) and (5.10).

---

### Algorithm 5.3 Functions for Extended Minimum CDS

---

**Function**  $\text{metric}(v, V_{\text{white}})$ :

1: **return**  $\sum_{u \in N(v)} \frac{k - B(k, u)}{k} + \frac{1}{k} |QN(k, v)|$

**Function**  $\text{promoteWhite}(v_{\text{best}}, V_{\text{white}}, V_{\text{black}})$ :

2: **return**  $N(v_{\text{best}}) \dot{\cup} \{v \in QN(v_{\text{best}}) : B(v) = k - 1\}$

---

E-MCDS promotes all white neighbors of the chosen node  $v_{\text{best}}$  as well as all of its quasineighbors  $v \in QN(v_{\text{best}})$  that reach a signal energy of 1. For such a quasineighbor  $v$  only the fraction  $1/k$  is missing. Hence,  $v$  must have received already  $k - 1$  transmissions of informed quasineighbors, hence  $B(v) = k - 1$ .

E-MCDS is an important heuristic for comparison to identify the effect of exploiting cooperation diversity with the same deterministic model of transmission ranges as used in GK. The following section shows why the deterministic model used both in GK and E-MCDS leads to poor packet delivery ratios and proposed an probabilistic model of transmission ranges instead to compensate for packet losses.

### 5.3. Fading-resistant low-latency broadcast sets

GK constructs the broadcast set by iteratively growing it from the source node. Nodes in the broadcast set inform their neighbors. In each iteration, one informed node is adopted into the set; GK greedily chooses that informed node for adoption with the most uninformed neighbors. Thus, the number of transmissions is kept small; on the other hand, informed nodes with many uninformed neighbors are typically as far away as the transmission range (if they were closer, their neighbors would likely be informed already). Hence, links in the broadcast set operate close to the minimum required SINR for successful communication. In such conditions, links become susceptible to fading and even slight drops in received energy might render a transmission useless, in which case the intended neighbors remain uninformed and the broadcast does not reach the entire network. As nodes of the broadcast set tend to be far apart, other transmissions from that set cannot compensate for this failure.

The detrimental effects of fading can be mitigated by cooperation diversity. Chapter 4 showed that for unicasting, the required retransmissions are costly as they need additional time slots and the additional robustness gained from cooperative relaying might not always justify these costs. Since retransmissions occur anyway during broadcasting, exploiting cooperation diversity comes without the usual costs and hence suggests itself to combat fading in this scenario.

ECDSs exploit the additional robustness gained from cooperation diversity to cover even larger groups of uninformed nodes. As a consequence, the distance between nodes in the broadcast set tends to grow even further, leading to yet smaller sets. In a wireless network without fading, this is desirable; with fading, however, these broadcast sets lack robustness. The problem is that the robustness gain from cooperation diversity is used to further thin out the broadcast set by letting links from quasineighbors to uninformed nodes operate close to their SINR limits. This is severe as small drops in received signal strength will stop the broadcast from reaching the entire network.

Figure 5.5 illustrates this problem in the toy example known from Figure 5.2. After having constructed a CDS using the deterministic unit disk graph model, where connectivity is based on distance-related path loss as an *average* link quality metric, the network operates on this set in a fading scenario. With fluctuations in signal strength, the circle is not an adequate representation of the transmission range anymore. Instead, the transmission range is an irregular dithering structure that even changes over time. Figure 5.5 exemplifies an *instantaneous* realization of this structure.

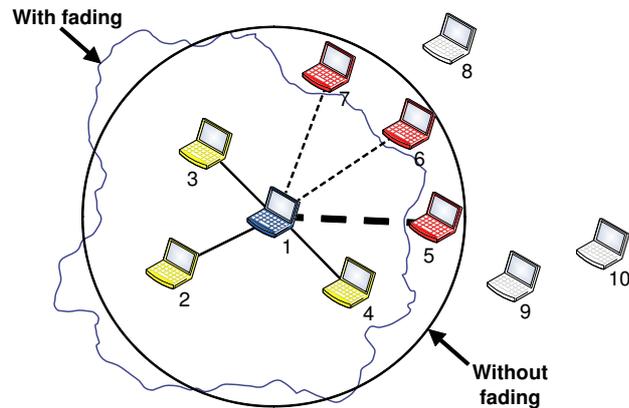


Figure 5.5.: Broadcasting on an optimistically constructed CDS in a fading scenario may already fail after the first transmission.

Assume that at this point in time, node 1 broadcasts its packet. Now only nodes 2, 3, and 4 correctly receive the packet, while nodes 5, 6, and 7 are outside the instantaneous transmission range. Thus, they do not receive the packet. What is even worse is that node 5 is part of the CDS and, since only nodes of the CDS retransmit, the broadcast dies prematurely. Nodes 8, 9, and 10 also remain uninformed as node 5 never broadcasts the packet. The same CDS that achieves full delivery without fading now fails to deliver the broadcast to the majority of the network.

To substantiate the intuition gained from the toy example, Figure 5.6 shows the delivery ratio – expected percentage of nodes receiving the broadcast packet – of the GK and the E-MCDS heuristics without and with fading for many realizations of random networks with hundreds of nodes.<sup>1</sup> As expected, without fading both heuristics achieve a close to perfect delivery ratio (some nodes are not informed because of interference between concurrent packet transmissions). In a fading environment, however, the small broadcast sets break apart when links fade below the SINR margin for successful packet delivery. Thus, small backbones are dangerous in fading environments; rather, such environments require judiciously constructed broadcast sets that achieve a high delivery ratio while staying as small as possible to keep broadcast latency down.

The problem is that the unit disk graph model used when constructing broadcast sets ignores the effects of fading. Transmission ranges are overly optimistic when their distance depends on the decoding threshold. Although more pessimistic thresholds shrink the size of unit disks, the resulting broadcast set may still not be optimal, simply because transmission ranges are not unit disks in fading scenarios. Recall the instantaneous transmission range from Figure 5.5. For a broadcast set construction heuristic, it is impossible to know any instantaneous transmission range, but, using a suitable model for fading, it can con-

<sup>1</sup>The scenario parameters are detailed in Section 5.3.2 where I evaluate my alternative approach by experiments.

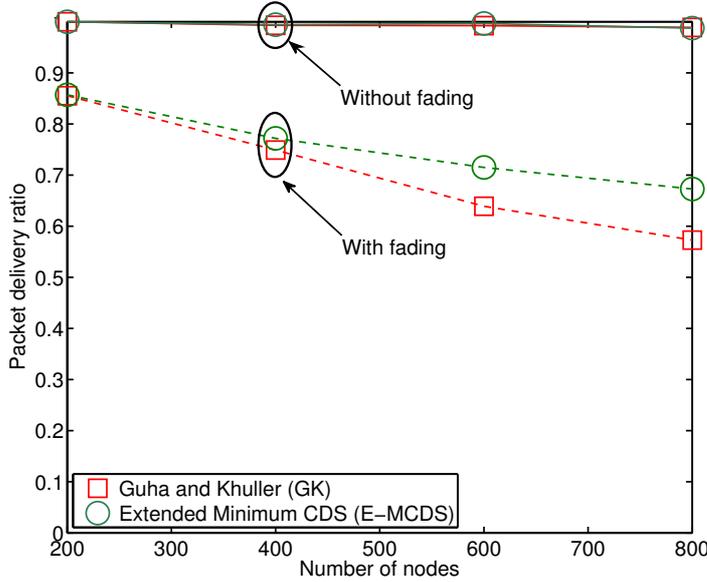


Figure 5.6.: Simulations show that delivery ratios deteriorate in fading scenarios, where a unit disk graph is no longer an appropriate model of transmission ranges in wireless networks.

struct the broadcast set for *averaged* transmission ranges with fading. Figure 5.7 illustrates what an averaged transmission range is. The figure overlays different instantaneous transmission ranges (drawn using thin lines). The idea is to approximate all these instantaneous ranges by a long-term average (drawn using a thick line). Since fading is a stochastic process, I use a probabilistic argument for this approximation. Instead of looking at the deterministic transmission range  $\bar{\gamma}_{uv} \geq \gamma_{th}$  where the SNR  $\bar{\gamma}_{uv}$  between nodes  $u$  and  $v$  consists of distance-dependent path loss only,<sup>2</sup> I want the outage probability  $\mathbb{P}\{X_{uv} < \gamma_{th}\}$  to stay *below* some robustness constraint  $\varepsilon$ . Here, the instantaneous SNR between nodes  $u$  and  $v$  is – according to the Rayleigh fading model – an exponentially distributed random variable  $X_{uv}$ . To still reflect the network’s geometry, the distance-dependent path loss determines the mean of this random variable. Using this model, the heuristic *regards* a node  $v$  as informed by node  $u$  if

$$\mathbb{P}\{X_{uv} < \gamma_{th}\} < \varepsilon. \quad (5.11)$$

If receiving nodes employ MRC for combining multiple transmissions (Section 2.2.1), the

<sup>2</sup>The threshold  $\gamma_{th}$  depends on the desired spectral efficiency as explained in Section 2.1.1. Since I consider a single rate for all nodes in this section, I use  $\gamma_{th}$  for convenience.

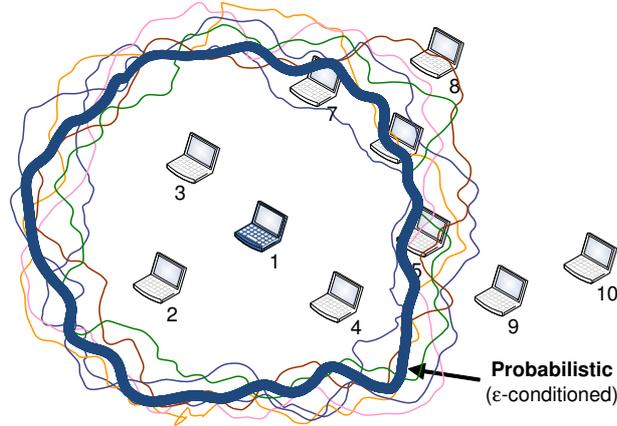


Figure 5.7.: Instead of using a unit disk model for constructing broadcast sets, I propose to use an approximation of the irregular, dithering transmission range with fading instead. Thin lines denote this range at an instantaneous point in time, whereas the thick line denotes the corresponding probabilistic approximation for a certain robustness constraint  $\varepsilon$ .

heuristic regards a node  $v$  as informed by combining transmissions from  $u_1, \dots, u_n$  if

$$\mathbb{P} \left\{ \sum_{i=1}^n X_{u_i v} < \gamma_{\text{th}} \right\} < \varepsilon. \quad (5.12)$$

Constructing fading-resistant backbones exploiting cooperation diversity with the probabilistic model is NP-hard. This justifies the heuristic approach taken in this section. Appendix C states the problem as elementary as possible and shows that, in its simplified form, it is already NP-hard. For extended problem formulations, NP-hardness then follows from the elementary one. Furthermore, the computational complexity depends on the complexity of cumulative distribution functions used in the problem formulation. Appendix C also shows that the problem remains NP-hard even if restricted to the exponential distribution.

### 5.3.1. Probabilistic Cooperation Diversity Broadcast (PCDB)

Based on the motivation in Figure 5.7, I first design a *probabilistic* heuristic in Algorithm 5.4 that achieves a high delivery ratio even in fading scenarios. For this, I incorporate the fading model into GK by modifying its node promotion; simply relying on the existence of a link defined by its *mean* SNR is not good enough in a fading environment. Instead, the heuristic regards a node as informed once its SNR exceeds  $\gamma_{\text{th}}$  with probability larger than  $1 - \varepsilon$ .

I illustrate how the probabilistic heuristic works, again using the toy topology from

---

**Algorithm 5.4** Functions for the *probabilistic* heuristic

---

**Function**  $\text{metric}(v, V_{\text{white}})$ 1: **return**  $|\mathbf{N}(v)|$ **Function**  $\text{promoteWhite}(v_{\text{best}}, V_{\text{white}}, V_{\text{black}})$ :2: **return**  $\{v \in V_{\text{white}} : \mathbb{P}\{X_{v_{\text{best}}v} < \gamma_{\text{th}}\} < \varepsilon\}$ 

---

Section 5.2.1. Figure 5.8 shows that in this topology five iterations suffice to construct the CDS. The probabilistic transmission ranges shown in this figure are not the instantaneous ones but those obtained using (5.11). Apart from that, the heuristic works like GK. The heuristic starts at node 1 in the first step and all nodes within its probabilistic transmission range are colored gray, these are nodes 2, 3, and 4. To determine these nodes, the heuristic must evaluate (5.11) for all  $X_{1,u}$  and  $u \in V$ . Whenever (5.11) holds for some node  $u$ , the heuristic considers node  $u$  as informed if its outage probability is below  $\varepsilon$ . In the second step, the heuristic chooses the node with the most uninformed (white) neighbors into the broadcast set, again using (5.11). In the example, this is node 4. The heuristic colors the newly elected set node black and all its neighbors inside its transmission range gray, node 5 in the example. This proceeds until no more white nodes remain.

Next, I combine fading-awareness using the probabilistic model *and* cooperation diversity to evaluate their *joint* benefit. I call this combined approach Probabilistic Cooperation Diversity Broadcast (PCDB), shown in Algorithm 5.5. In this approach, a node becomes informed once the *sum* of its received SNRs exceeds  $\gamma_{\text{th}}$  with probability larger than  $1 - \varepsilon$ , accounting for the benefits of MRC at the receiving nodes. Thus, the heuristic needs to evaluate (5.12).

---

**Algorithm 5.5** Functions for the *combined* heuristic

---

**Function**  $\text{metric}(v, V_{\text{white}})$ 1: **return**  $|\mathbf{N}(v)|$ **Function**  $\text{promoteWhite}(v_{\text{best}}, V_{\text{white}}, V_{\text{black}})$ :2: **return**  $\{v \in V_{\text{white}} : \mathbb{P}\{\sum_{u \in V_{\text{black}}} X_{uv} < \gamma_{\text{th}}\} < \varepsilon\}$ 

---

Figure 5.9 illustrates how the PCDB heuristic works. During the first step, the PCDB heuristic informs the same nodes as the probabilistic heuristic since the transmission range is the same and only a single node has transmitted so far. Nodes that are outside this transmission range cannot receive the broadcast packet correctly but still they may receive something. I assume that nodes can buffer corrupt broadcast messages that they have received for later combining. For illustration, progress bars beside the nodes in Figure 5.9 indicate the signal energy that a node has already received. This is merely for illustration; an implementation would not have an equivalent of these progress bars. There, the benefit comes from the implementation of MRC. After decoding, the CRC indicates whether the broadcast packet is correct or corrupt, but it does not indicate how much signal energy

## 5. Mechanisms for cooperative broadcasting

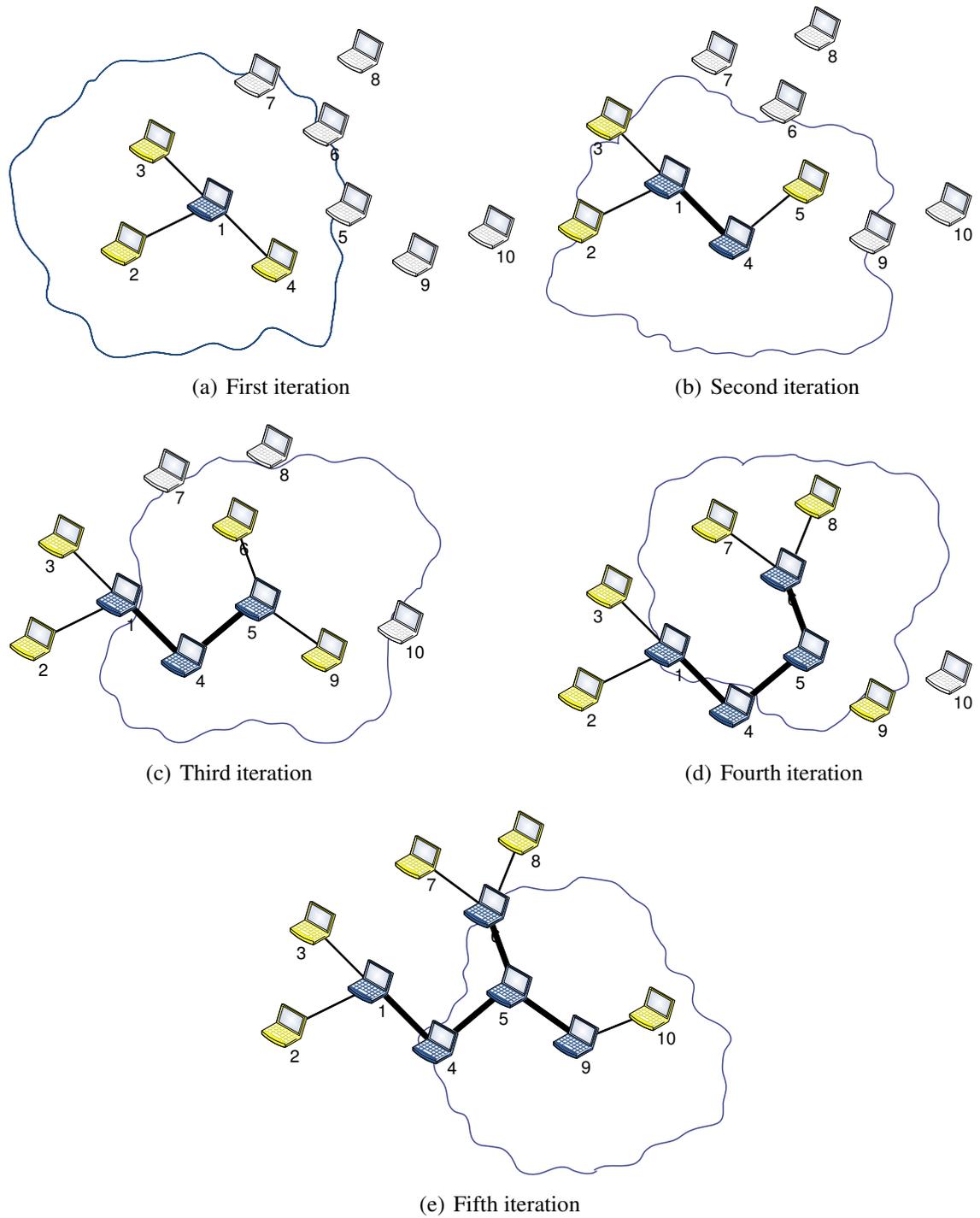


Figure 5.8.: Probabilistically constructing a CDS

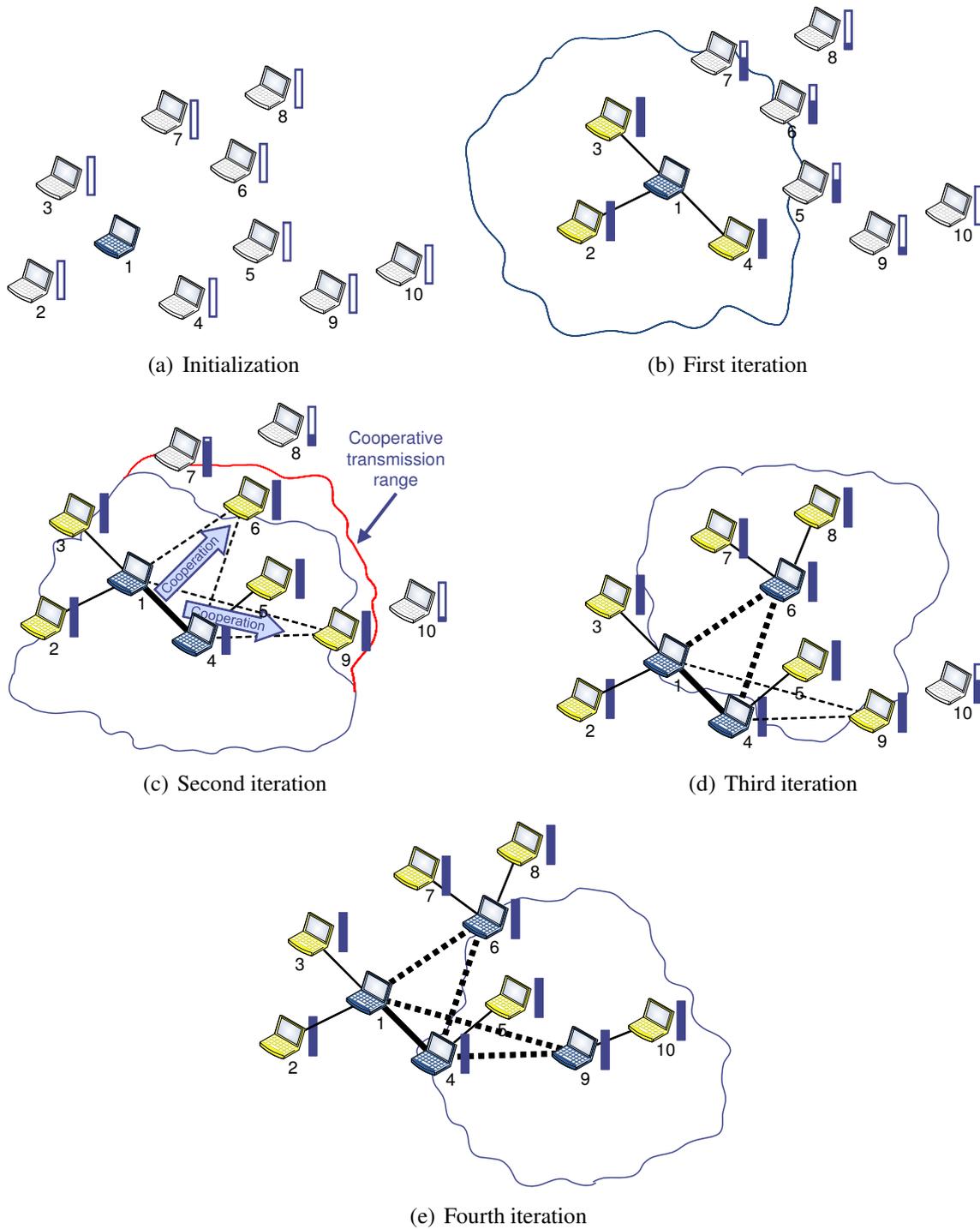


Figure 5.9.: Constructing a Probabilistic Cooperation Diversity Broadcast (PCDB) set

a node already has accumulated. After the first step, nodes 5 to 9 have all buffered the corrupt broadcast packet. If the received signal is too low for coherent reception, a node cannot buffer a packet at all in which case there is no contributed energy and the progress bar does not change. In the first step, this is the case for node 10.

The second step of the example illustrates how the probabilistic model incorporates cooperation diversity and which impact this has on the broadcast set. Without exploiting cooperation diversity (i.e., without buffering and combining broadcasts) the transmission range corresponds exactly to that of the probabilistic approach where only node 5 becomes informed in the second step. Nodes 6 and 9 do not meet (5.11), i.e., the probabilities  $\mathbb{P}\{X_{4,6} < \gamma_{\text{th}}\}$  and  $\mathbb{P}\{X_{4,9} < \gamma_{\text{th}}\}$  are above the required robustness. The PCDB heuristic instead assumes that nodes 6 and 9 have previously received information from node 1, therefore it evaluates the outage probability equations differently, i.e.,  $\mathbb{P}\{X_{1,6} + X_{4,6} < \gamma_{\text{th}}\}$  and  $\mathbb{P}\{X_{4,6} + X_{4,9} < \gamma_{\text{th}}\}$ , and this time the probability to meet the required robustness is higher.

Figure 5.9(c) illustrates how the addition of random variables reflecting the instantaneous channel SNRs impacts the transmission range spanned by node 4. The cooperative diversity gain increases this transmission range. In fact, nodes 1 and 4 *jointly* span an extended transmission range. In this step, two CTRs form where node 1 takes the role of the source and node 4 takes the role of the relay and the destination nodes are 6 and 9 for both CTRs. Node 6, reached cooperatively, now has the most uninformed neighbors and will become part of the broadcast set in the next step. In the third iteration, the heuristic informs nodes 7 and 8 and, finally, it chooses node 9 into the set in the last iteration. The resulting broadcast set consists only of nodes 1, 4, 6, and 9, whereas with the probabilistic heuristic, additionally node 5 is part of the set. This example illustrates that exploiting cooperation diversity can reduce the size of the broadcast set. To answer the question how the delivery ratios compare between the heuristics, I present the results of simulations in the next section.

### 5.3.2. Simulations in a fading scenario

The simulation consists of two important parts that are cleanly separated. In the first part, the simulation executes the heuristic on a random placement of nodes. For this, the heuristic needs to know the *mean* SNRs on all links to evaluate the exponential distribution, i.e., the heuristic requires global knowledge which is easily at hand in a simulator. The path-loss dependent distance between the nodes suffices to compute the mean SNRs and, thus, allows to compute the probability that a broadcast transmission to a certain node succeeds for a certain robustness constraint. Note that this is only a probability; it does not say anything about what, at any particular moment in time, might happen with fading. The result of the first phase is a set of nodes that forms the broadcast set for the simulation of the actual broadcast in the second step.

In the second step, I simulate a broadcast using the broadcast set determined in the first step using a time-correlated fading model. Unlike the first phase, broadcasts are now sub-

Table 5.3.: Summary of simulation parameters used for broadcasting

	Parameter	Value
<i>Channel</i>	Carrier frequency	2.4 GHz
	Path loss exponent $\alpha$	4
	Number of fading paths	20
	Doppler shift	8 Hz ( $\Rightarrow v = 1$ m/s)
<i>Nodes</i>	Transmission power	20 dBm
	Noise power density $N_0$	$4.32 \cdot 10^{-21}$ W/Hz
	Node density $\lambda$	$4 \cdot 10^{-4}$
<i>PHY</i>	Data rate	1 Mbits/s (most robust)
	Rx/Tx turnaround time	$5 \mu\text{s}$
	Distributed inter-frame space	$50 \mu\text{s}$
	Decoding threshold $\gamma_{\text{th}}$	25.9 dB (Section 5.2.2)
	Cooperative stretch $q$ for E-MCDS	1.19 (Section 5.2.2)
<i>MAC</i>	Contention window	8 slots
	Slot duration	10 ms
	Packet size	32 bytes

ject to instantaneous channel states and may also interfere with each other in case nodes transmit simultaneously. Thus, in this second step the simulator uses SINR instead of SNR. To mitigate interference, all nodes in the simulation employ a non-persistent carrier sensing MAC protocol. I obtain delivery ratios and broadcast latencies only from the second step in the simulation. To get statistically significant results, I generated 20 independent placements and averaged the computed metrics. The following plots show the results with 95 % confidence.

I start the broadcast with the same node  $s$ , determined at random, as used to compute the broadcast set. Once a set node has correctly received a message, it competes for access to the medium. Competition is needed in case several nodes in the broadcast set become informed at once and would hence collide when transmitting right away. Every node sends the broadcast message at most once; non-set nodes do not send at all. Since contention introduces another source of randomness, I repeat the broadcast simulation 100 times (per node placement). For every transmission from a broadcast node  $i$  to any other node  $j$  in the network, I properly consider interference of other set nodes  $k \neq i$ . If  $k$  and  $i$  are sufficiently spaced apart, they may transmit simultaneously due to spatial reuse. At  $j$ , the transmission from  $k$  contributes to the denominator of the SINR  $\gamma_{ij}$  at the receiver according to (2.5). The concrete parameters for the Physical Layer (PHY) and MAC sub-layer are motivated by an IEEE 802.11b system (mostly because of the CSMA/CA MAC assumption) and model a pedestrian mobility environment. Table 5.3 summarizes the simulation parameters.

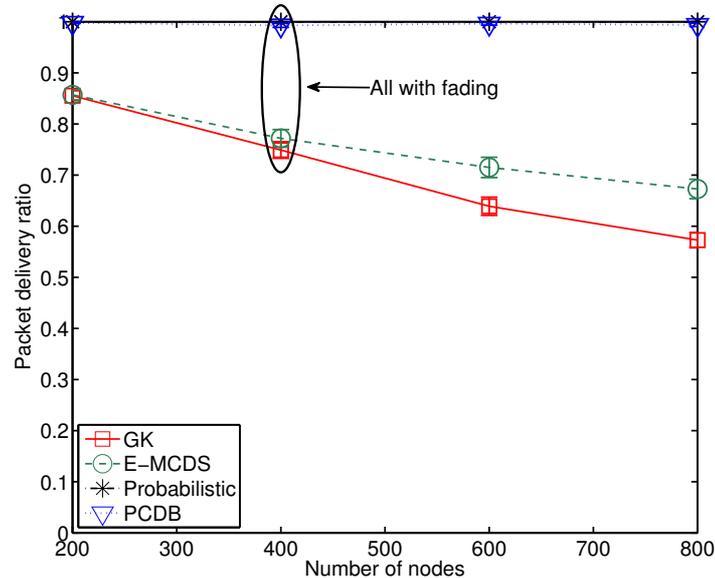


Figure 5.10.: Both probabilistic and combined approaches re-obtain full delivery with fading regardless the network size. Confidence intervals shown for 95 % confidence level.

Figure 5.10 shows the delivery ratios for varying network size. Similar to Figure 5.6, GK and E-MCDS fail to deliver the broadcast to all nodes in the network; they are *fading-susceptible*. If the network grows and broadcast sets become larger, the probability increases that links in the set fail. By incorporating the fading model into GK (probabilistic approach and PCDB), full delivery can be achieved even with fading. Furthermore, with these *fading-resistant* heuristics, the growth of the network does not cause delivery ratios to collapse. With respect to delivery ratio, both probabilistic heuristic and PCDB perform equally well and substantially outperform existing heuristics.

Since I not only strive for full but also for *fast* delivery, I next study the broadcast latencies in Figure 5.11. Small latencies do not imply that all nodes are informed by the broadcast (e.g., if no-one sends, the latency is minimized but so is the delivery ratio). Thus, I categorize the results into two groups. The fading-susceptible approaches fail to deliver the message in the entire network (e.g., for 1,000 nodes no more than 60 % can be informed). If a transmission between two nodes of the broadcast set fails in the event of fading, the set becomes disconnected and the broadcast ends prematurely. The fading-resistant approaches, on the other hand, succeed in informing more than 99 % percent of all nodes in the network. Not only are their latencies meaningful measures, they can also be directly compared as their delivery ratios are similar. The combined approach outperforms the mere probabilistic approach by exploiting cooperation diversity; the reason for this can be found in Figure 5.12.

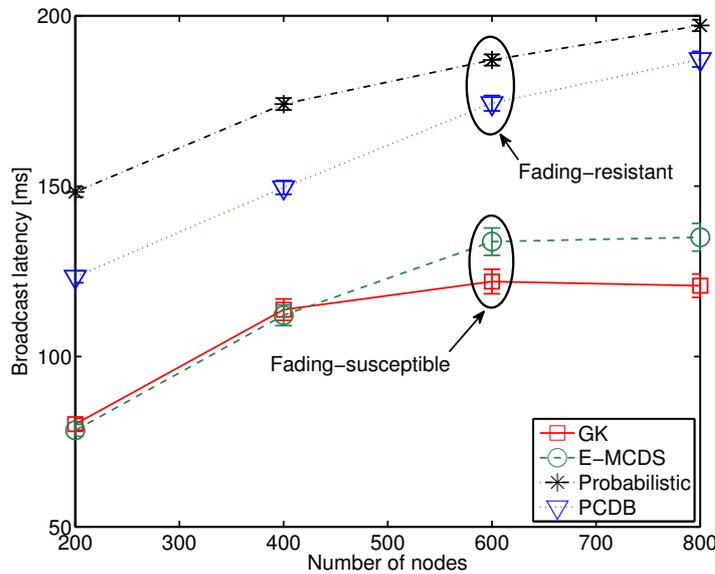


Figure 5.11.: PCDB achieves the lowest latency with full delivery (fading-resistant); the smaller latencies of GK and E-MCDS are meaningless as full delivery fails (fading-susceptible). Confidence intervals shown for 95 % confidence level.

Figure 5.12 shows the average size of the broadcast set computed by the various heuristics.<sup>3</sup> By only taking path loss into account, the fading-susceptible approaches optimistically generate the smallest broadcast sets. The distance between neighboring set nodes is large, thus on average the received signal energy is close to the decoding threshold. As a result, small drops in received energy already cause decoding errors. By incorporating the fading model into the heuristic, the density of set nodes and hence the size of the set significantly increases. Looking only at the probabilistic heuristic, neighbors receive energy on average far above the threshold. This leaves sufficient energy to compensate for losses.

By exploiting cooperation diversity, PCDB can reduce the size of broadcast sets. Now less energy is required as nodes become capable of “accumulating” energy by buffering and combining received packets and hence variance is reduced. Thus, any previous transmission reduces the gap in energy until the decoding threshold is reached. In case of a deep fade, even little additional received energy may be sufficient. The gain from cooperation diversity can now be safely transformed into smaller broadcast sets. Since smaller sets involve less retransmissions, as a result, latency can be further reduced.

<sup>3</sup>Note that these numbers are determined *only* from step 1 of my evaluation; the discrete event simulator does not play a role here. Hence, these results immediately carry over to different PHY and MAC protocols.

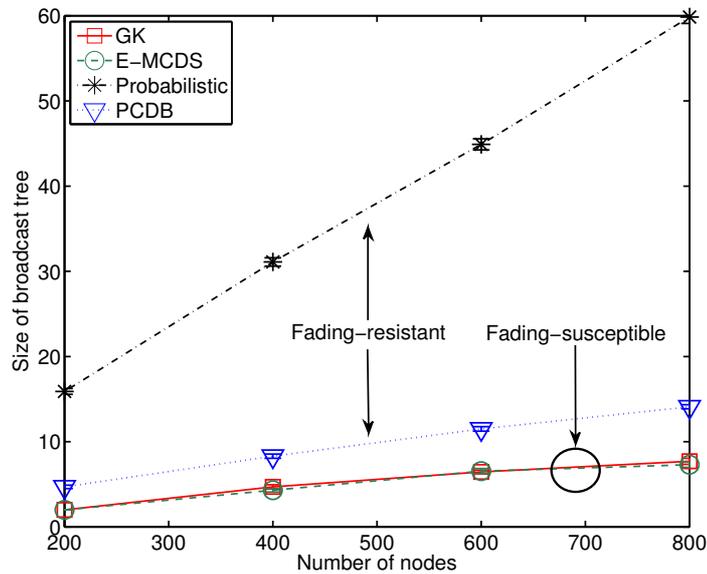


Figure 5.12.: Denser broadcast sets become resistant against fading; cooperation diversity gains can be transformed into smaller sets without sacrificing full delivery. Confidence intervals shown for 95 % confidence level.

## 5.4. Conclusions

I developed heuristics for fading-aware broadcast sets by explicitly considering the distribution function of the received signal strength. This improves delivery ratio in fading environments by constructing larger broadcast sets, but at the same time also increases the broadcast latency as more nodes retransmit the broadcast. To obtain low broadcast latency at the same time, nodes must exploit cooperation diversity on top of it. Neither cooperation diversity nor the probabilistic model on their own can achieve high delivery ratios with low latency, making their *combination* the essential technique for broadcasts in wireless networks.

Figure 5.13 summarizes the four heuristics discussed in Section 5.3 and shows how PCDB naturally evolves from them. Exploiting cooperation diversity and using a probabilistic model are orthogonal techniques. While the probabilistic model, with a suitable choice of  $\epsilon$ , builds larger broadcast sets to increase delivery ratios, exploiting cooperation diversity gains for additional robustness can thin out the set again, possibly decreasing broadcast latency.

Using the probabilistic model, the required robustness  $\epsilon$  is a parameter to trade off delivery ratio versus size of the broadcast set. Smaller values of  $\epsilon$  lead to larger broadcast sets and, thus, increase latency, but at the same time also increase delivery ratios. Ideally, one needs the largest  $\epsilon$  that achieves full delivery with high probability.

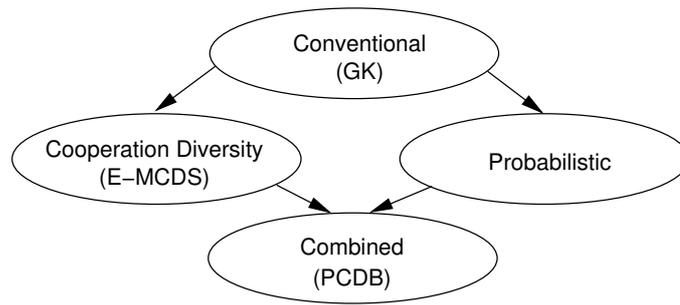


Figure 5.13.: Evolution of backbone heuristics

I did not discuss differences between centralized and distributed heuristics because the problem of low delivery rates in fading scenarios stems from an inadequate model of transmission ranges on which heuristics operate; it does not stem from a local versus global view of the network. My probabilistic heuristic as well as the PCDB heuristic can be integrated into many other schemes for broadcast set construction, opening up a wide range for future research.



## 6. Rapid prototyping for cooperative MAC protocols

*A MAC protocol developer has to cope with various, heavily distributed protocol functions that are tedious to implement and to debug. To untie this complex development process, I propose to automate its most error-prone parts: Implementation of MAC automata, analysis, and code generation. To do so, I formalize cooperative MAC protocols by a new, easy-to-use specification language. For the proposed language, I show how to construct various compiler backends that automatically analyze validity and performance of the specification and that translate the specified protocols into program code for different targets, e.g. simulators or SDRs.*

As shown in Chapter 4, integrating cooperative relaying into IEEE 802.11 WLANs requires modifications at the MAC sub-layer to include the relays in the transmission cycle. Depending on the particular approach even coding and combining extensions at the PHY may be needed. Simulations or prototype-based measurements are obligatory before manufacturers can decide which overall system design will eventually constitute their next-generation systems. Unfortunately, implementing MAC protocols for cooperative relaying is an error-prone task whose complexity leads to long development times. This is even more severe if the *same* protocol needs to be implemented on different targets, e.g., simulator and SDR. Thus, when system designs must be evaluated by simulations and prototypes, the *process* of implementing cooperation must be accelerated.

Formal specification of network protocols arose first to avoid the ambiguity of natural languages and is now a standard approach for their verification [25]. Due to the concurrent nature of network protocols, verification requires to check all possible states that communicating nodes may be in, possibly leading to a plethora of states for which symbolic model checking has been proposed [13]. Important properties of a protocol can then be stated as Linear Temporal Logic (LTL) predicates, e.g., that some state of the protocol is eventually reached (liveness), and verified in the sense that a predicate either holds surely or it might not be true. SPIN is one such model checker [35] whose specification language is similar to Gouda's Abstract Protocol (AP) notation [26].

This prior work has mainly focused on the important question whether the specified protocol does what it is intended to do. For the development of *efficient* network protocols, correctness is without doubt indispensable, but not sufficient. The reason is that correct behavior can be specified in a multitude of ways, each one having its own characteristic performance. Assume that one has a set of protocols all of which have been verified

to be correct. One then needs to find a protocol in this set that achieves the best performance according to a metric, e.g., throughput. Since performance can only be measured using suitable implementation, methods are needed to correctly and efficiently transform specifications into implementations. This is where the work in this chapter comes in.

In this chapter, I describe a compiler-assisted approach for automating the development of cooperative MAC protocols [51], comprising automated analysis and cross-platform implementation. For this, the following three steps are necessary. First, automation requires to formalize the design of cooperative relaying systems. I introduce the notion of patterns and use them to describe some non-cooperative and cooperative MAC protocols in Section 6.1. Second, I identify the domain-specific knowledge that is required to further process the formalized attributes of patterns by a machine. I derive a new specification language as a basis for my automated development approach and provide its grammar for constructing a compiler frontend in Section 6.2. Finally, in Section 6.3 I give details on how to construct compiler backends for analysis and implementation. By replacing backends, a single specification of a protocol suffices to derive analytic performance bounds and implementations for simulators or SDRs.

### 6.1. Describing MAC protocols by patterns

Since I want a framework for the analysis and implementation of MAC protocols, I first need a formal and, at best, intuitive description of MAC protocols. For this, I now introduce the concept of *patterns*. To avoid confusion, I use the term *user* to refer to nodes in a network, whereas I use the term *node* to refer to nodes in trees.

#### 6.1.1. MAC patterns: Intuition and formal definition

From a MAC perspective, users in a wireless network dynamically take roles out of a predefined set of roles. For example, two users A and B may take the roles source  $s$  and destination  $d$ . A MAC pattern describes a convention in exchanging frames that users must obey to successfully convey data from a source to a destination. The pattern does not define the roles that particular users take. Users may take *any* role defined by the pattern. Consider the two users A and B again. At some time, A may be source and B its destination; later, these roles may be swapped. A pattern is said to be *instantiated* as soon as all roles are mapped to particular users. Although some roles can be determined *a priori* (i.e., before the instantiation of the pattern has begun), the entire mapping cannot be done *a priori* as parts of the mapping depend on control information in frames. For example, the destination is determined by comparing its MAC address to that specified in the destination address field of the received frame. Patterns for cooperative MAC protocols are another example where the role of the relay  $r$  can either be determined *a priori* or while already processing the pattern. The latter case applies when the selection of a suitable relay requires information that must first be exchanged between users (e.g., to estimate channel

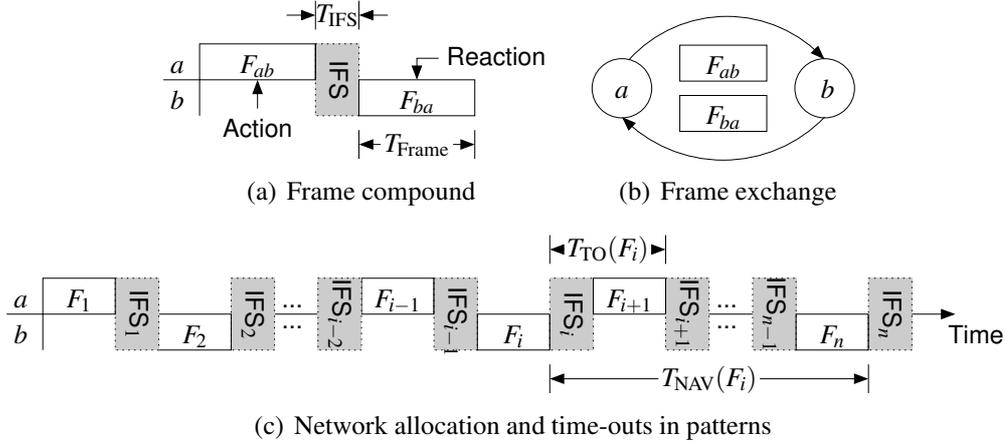


Figure 6.1.: The *frame compound* is a fundamental structure of any pattern that describes a single frame exchange between two users.

states) and, hence, relies on a frame exchange. This intuitive notion of MAC patterns can also be defined formally as follows:

**Definition 3.** Let  $\mathcal{R}$  be the set of roles. A MAC pattern  $\Pi$  is a finite, time-ordered sequence of frames  $F_i$  and IFSs  $I_i$ , i.e.,  $\Pi = (F_1, I_1, F_2, I_2, \dots, F_n, I_n)$ . For every frame in the pattern, a single transmitter role must be defined, i.e.,  $\text{Tx}(F_i) \in \mathcal{R}$ , and one or more receiver roles, i.e.,  $\text{Rx}(F_i) \subseteq \mathcal{R} \setminus \text{Tx}(F_i)$ .

I first identify the *frame compound* shown in Figure 6.1(a) as a fundamental structure of patterns. Although the frame compound is not an atomic component (such as a single frame or a single IFS), it is the smallest *compound* component that sufficiently describes a frame exchange between two users as depicted in Figure 6.1(b). A frame exchange between  $a$  and  $b$  can be uniquely described by the frame that  $a$  sends to  $b$ ,  $F_{ab}$ , the frame that  $b$  then sends in reply to  $a$ ,  $F_{ba}$ , if any, and the IFS that separates both frames. Such exchange takes time, which is important for virtual carrier sensing and error detection. Virtual carrier sensing, as typically done in IEEE 802.11-based systems [69], requires knowledge about the time that the medium is expected to be busy. Errors are detected when an anticipated frame, say  $F_{ba}$ , does not arrive after a specified time. In order to later quantify these times, I now define  $T_{\text{Frame}}(F_i)$  as the time required for transmitting frame  $F_i$ , and  $T_{\text{IFS}}(F_i)$  as the duration of the IFS  $I_i$  that follows  $F_i$ . Figure 6.1 exemplarily shows  $T_{\text{IFS}}(F_{ab})$  and  $T_{\text{Frame}}(F_{ba})$ .

The NAV and Time-out (TO) values that occur in patterns are shown in Figure 6.1(c). The NAV is a mechanism at the MAC sub-layer to realize a guard zone for reducing possible interference by other users in the vicinity as discussed in Chapter 3. The duration field of any frame in the pattern states the time span until the transmission of the last frame of the pattern completes. Given a particular frame of the pattern, the MAC implementation computes this time span by adding all transmission times of the successive frames as well

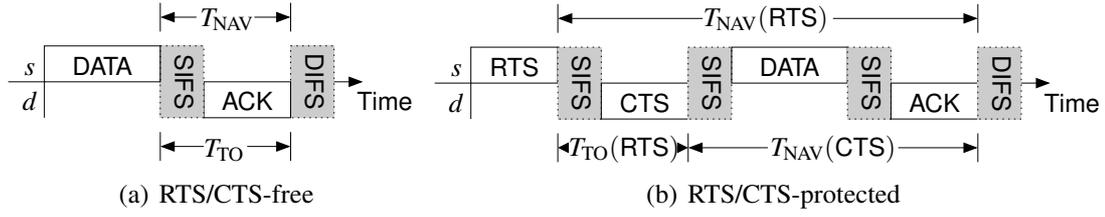


Figure 6.2.: Example of two non-cooperative patterns

as all IFSs, except for the final one. In other words, one can think of frames of a pattern as a chain of frame compounds as shown in Figure 6.1(c). Given a particular frame  $F_i$  in a pattern  $\Pi$ , the value  $T_{\text{NAV}}(F_i)$  of its duration field equals

$$T_{\text{NAV}}(F_i) = \sum_{k=i}^{n-1} T_{\text{IFS}}(F_k) + \sum_{k=i+1}^n T_{\text{Frame}}(F_k). \quad (6.1)$$

Except for the last frame in the pattern, which does not require a reply (or it would not be the last frame), any frame is associated with a time-out. This time-out indicates how long a user should wait until it considers the successive frame as lost. When the anticipated frame has not arrived until the time-out expires, alternative actions must (usually) be performed. Given a particular frame  $F_i$  in a pattern  $\Pi$  with  $i \neq n$ , its time-out  $T_{\text{TO}}(F_i)$  corresponds (at most) to the end of transmission of the successive frame, thus yielding

$$T_{\text{TO}}(F_i) = T_{\text{IFS}}(F_i) + T_{\text{Frame}}(F_{i+1}). \quad (6.2)$$

A MAC protocol comprises one or more patterns that at least have the initial frame  $F$  in common, and it can be formally defined as follows:

**Definition 4.** A MAC protocol  $\mathcal{P}$  is a finite set of patterns, i.e.,  $\mathcal{P} = \{\Pi_1, \dots, \Pi_k\}$  with  $\Pi_i = (F, I_1^i, F_2^i, I_2^i, \dots, F_{n_i}^i, I_{n_i}^i)$  for all  $1 \leq i \leq k$ .

A simple protocol that only involves the roles source and destination may be described by a single pattern, which is discussed in more detail in Section 6.1.2, but several patterns become necessary when a protocol provides alternatives. For example, a cooperative MAC protocol might initiate a retransmission through a relay if a particular frame is not acknowledged. Such a protocol could be described using two patterns, one for the normal case and one for the erroneous case that resorts to retransmission as an alternative. Patterns for cooperative MAC protocols are discussed in Section 6.1.3.

### 6.1.2. Example: Non-cooperative patterns

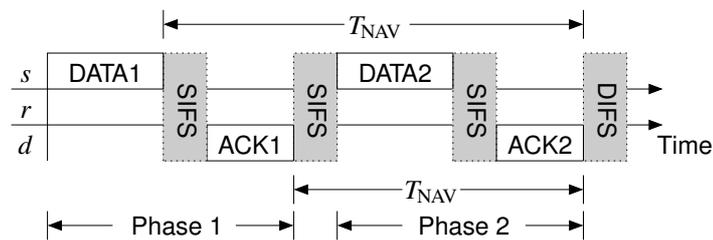
Figure 6.2(a) shows the pattern of a simple direct transmission involving two users. The pattern consists of DATA and ACK frames, where the destination sends the ACK frame in

response to a DATA frame received from the source. In absence of the ACK frame, the source assumes the transmission of the DATA frame to have failed and may either retry the transmission or signal an error to the upper layer. The first frame of any pattern is always triggered from the upper layer when it has a packet to transmit, whereas successive frames are triggered either by reception of a frame or by an event, e.g., a time-out. The progression of the pattern can be described by a finite automaton, and Section 6.3.3 later explains how to derive the implementation of such an automaton from a pattern. The pattern depicted in Figure 6.2(a) involves two roles, source and destination, which can only be determined during run-time. Once the role of a user is set, it will keep this role for the instance of the pattern. In the example, the criterion for taking the source role is a transmit request from the upper layer, whereas a user becomes the destination when its MAC address matches the address of the recipient specified in the DATA frame.

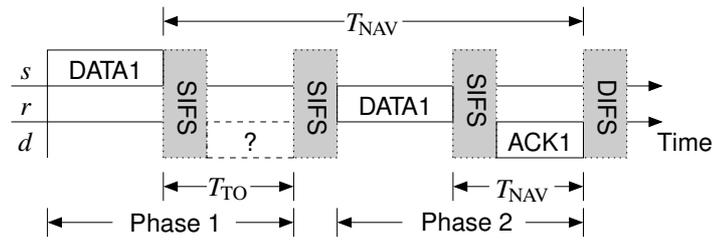
Figure 6.2(b) shows an alternative and slightly more complex pattern. It protects the DATA exchange by an RTS/CTS sequence to tackle the hidden terminal problem. Both RTS and CTS frames carry the duration for which the wireless medium is expected to be allocated, indicating how long a user must refrain from transmitting. In Figure 6.2(b), the duration field of the RTS frame covers all frames and IFSs that follow the RTS. The same is true for the CTS. Assuming that the transmission parameters of all frames are known, the pattern itself provides the required information (i.e., which frames and how many IFSs follow) to sufficiently compute the duration field of any frame. Similarly, TOs can be computed. After a frame has been sent, the recipient is required to respond within a certain time span. As an example consider the pattern in Figure 6.2(b) again. After the source has sent the RTS (i.e., the action of  $s$ ), it anticipates the CTS from the destination (i.e., the reaction of  $d$ ). The time span TO that  $s$  has to wait for a reply after it has sent its frame equals the IFS and the transmission time of the successive frame. Again, the pattern itself suffices to compute TO when the transmission parameters are known. Only the final frame is exceptional as it terminates the pattern and, thus, does not require a time-out to be set.

### 6.1.3. Example: Cooperative patterns

Figure 6.3 shows an example of a simple cooperative protocol involving three roles, namely source  $s$ , destination  $d$ , and relay  $r$ . The protocol consists of two alternative patterns. Both patterns comprise two phases where the second phase is reserved for use by the relay for retransmitting data of the first phase. Figure 6.3(a) shows the pattern that specifies the behavior when the source receives the acknowledgment of the first phase correctly. The source starts the frame exchange by transmitting a DATA frame to  $d$ . The duration field of the DATA frame is set such that overhearing users update their NAV to refrain from transmitting for the entire duration of the pattern (comprising both phases). Assuming that  $d$  can successfully decode the DATA frame, it replies with an ACK frame to indicate successful reception to  $s$  after a Short Inter-Frame Space (SIFS) time, thereby finishing the first phase of the protocol. Although retransmission is not required, note that the wireless



(a) Case 1: When the DATA frame was successfully received and acknowledged in the first phase, the source  $s$  can send its successive DATA frame in the second phase.



(b) Case 2: When the DATA frame was not acknowledged in the first phase, the relay  $r$  must retransmit the overheard DATA frame in the second phase.

Figure 6.3.: Example of a simple two-phase cooperative protocol that consists of patterns for two alternative cases.

medium has already been reserved for the duration of the entire pattern. This is because in the beginning it is not known whether the source's DATA of the first phase will be correctly received or not. User  $s$  has two alternatives. It could either remain silent during the second phase or send the next data frame in its transmit queue. In the latter case,  $d$  must again acknowledge its reception with an ACK frame after a SIFS.

Now suppose that  $d$  could not correctly decode the DATA frame sent by  $s$  in the first phase. Thus, the ACK frame anticipated by the source never arrives which  $s$  and  $r$  detect after the time-out associated with the DATA frame has elapsed. The simple two-phase cooperative protocol specifies an alternative pattern in Figure 6.3(b) that handles this case. Assuming that a user in the vicinity of  $s$  has overheard and correctly decoded the DATA frame sent by  $s$ , it must retransmit this DATA frame in the second phase of the protocol on behalf of  $s$ . I neglect the problem of finding a suitable relay here (i.e., assigning the role of being a relay to some user). One possibility would be that  $s$  permanently monitors users in its vicinity and elects one as a relay  $r$  according to some metric before starting to transmit the pattern (e.g., refer to [9]). Then,  $s$  can specify the address of  $r$  in the first phase's DATA frame, and – from the MAC protocol's view – the criterion for becoming a relay is merely a comparison of MAC addresses, like already used for determining the destination. To keep pattern's specifications simple, criteria for role election are not part of the pattern, and must be amended manually after code generation. For the specification of a pattern it does not matter which roles *particular* users take.

## 6.2. Automating MAC development by patterns

For automating the development of cooperative MAC protocols, I now develop MAC Pattern Description Language (MPDL) as a new specification language for MAC patterns. The compiler is the tool for generating both an analytical model and an implementation of the MAC automaton for *any* correct specification of a pattern. I base my work on compilers that perform their task by evaluating *abstract program trees*. When this internal data structure is used, semantic analysis and transformation can be implemented using standard methods [27]. To do so, I first need a way to represent patterns as trees.

### 6.2.1. Step 1: From MAC patterns to trees

A single pattern corresponds to a particular path in the abstract program tree whose nodes represent the frame compounds of the pattern. An alternative path in the abstract program tree is possible whenever a frame is anticipated and may or may not arrive. This notion requires NAV and TO as important attributes of the abstract program tree's nodes. Figure 6.4 shows the structure of such a node, which is an internal representation of the frame compound depicted in Figure 6.1. Action and reaction of the frame compound appear in the node as  $F_{ab}$  and  $F_{ba}$  with their associated roles  $b$  and  $a$ , respectively. The associated roles inform the compiler that a user in role  $b$  can handle the reception of a frame (denoted

## 6. Rapid prototyping for cooperative MAC protocols

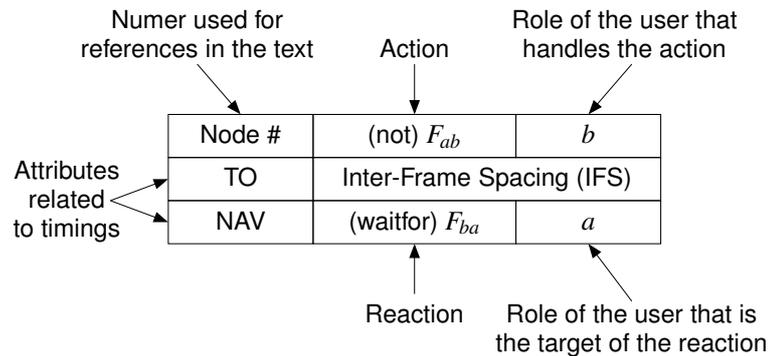


Figure 6.4.: Necessary attributes of any frame compound in form of an abstract program tree's node that is used in the compiler.

as " $F_{ab}$ " in nodes), or the absence of that frame after a time-out (denoted as " $not F_{ab}$ " in nodes). The handling (or reaction) defined by frame compounds is to send another frame in reply. Therefore, the node also states which frame  $F_{ba}$  to send after an optional IFS, and the role  $a$  of its recipient. Although a reply is the most common reaction, others are also feasible. For cooperative MAC protocols, for example, a relay that overhears a DATA frame may wait for an ACK to decide whether to retransmit or not. Thus, another reaction to a DATA frame is to wait for an ACK and to perform alternative actions when it does not arrive. Finally, a node includes the two fundamental properties NAV and TO that indicate the network allocation for the remaining pattern and the time-out for the reply, respectively. Initially, NAV equals TO, but the compiler updates NAV during semantic analysis according to an attribute grammar as discussed later in Section 6.3.

I now illustrate this notation by exemplarily constructing an abstract program tree for the two-phase cooperative protocol introduced in Section 6.1.3. Figure 6.5 annotates the frame compounds of both the primary and alternative pattern with their corresponding tree nodes. The abstract program tree corresponding to these two patterns is shown in Figure 6.6. The root of the tree corresponds to the declaration of the pattern, followed by the first frame compound that is part of the primary and the secondary pattern. The root does not correspond to a frame compound since the first frame is triggered by a transmit request from the upper layer. Thus, the action is an event rather than a frame, whereas the reaction indeed is a frame and allows for linking to other frame compounds.

This tree must provide an alternative path because the ACK frame of the first phase may not arrive at  $s$ , causing the relay  $r$  to assist. Therefore, node #2 of the alternative path defines that a user acting as the relay shall react upon the first DATA frame by waiting for the associated ACK frame. If the ACK never arrives at the relay (i.e., a " $not ACK$ "-action occurs), the relay reacts by retransmitting the DATA frame to the destination; otherwise it remains silent for the rest of the NAV period.

The compiler uses an attribute evaluator to compute NAV and TO values by performing a depth-first left-to-right walk through the abstract program tree. TO can be computed on the

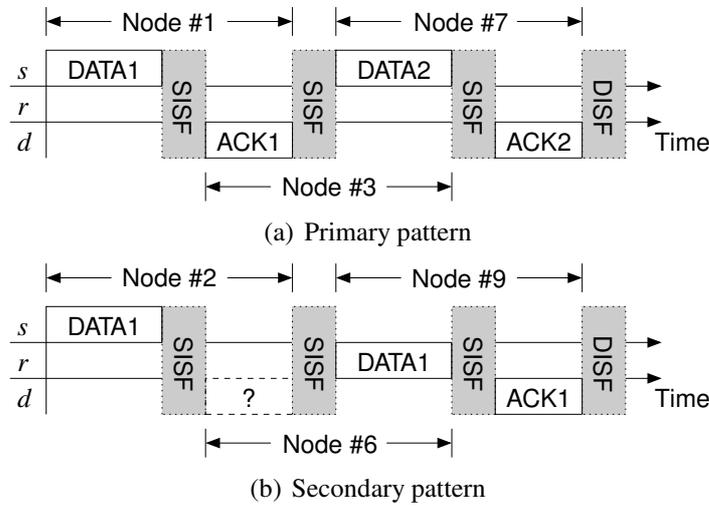


Figure 6.5.: Patterns of the simple two-phase cooperative protocol introduced in Figure 6.3. Each pattern corresponds to a path in the abstract program tree.

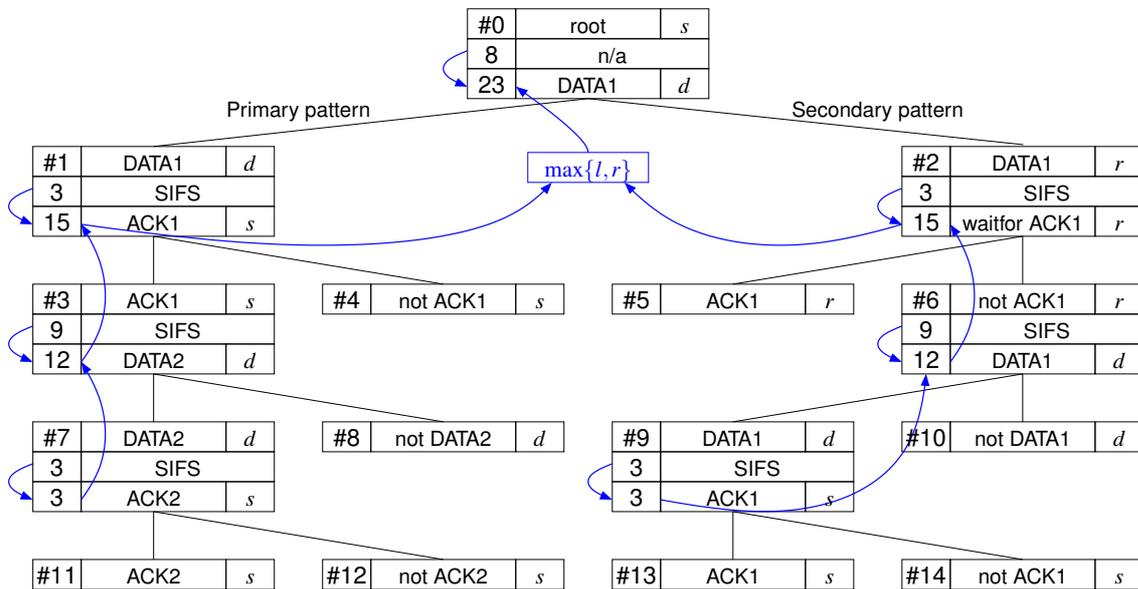


Figure 6.6.: This abstract program tree corresponds to the simple two-phase cooperative protocol shown in Figure 6.3. The computations for NAV and TO in the tree assume  $T_{\text{DATA}} = 8$  ms,  $T_{\text{ACK}} = 2$  ms, and  $T_{\text{SIFS}} = 1$  ms as an example.

first visit of a node according to (6.2). A node that describes a frame compound provides information about the IFS and the successive frame. The attribute evaluator computes the sum and stores it in the TO attribute of the node that it currently visits. For the last frame compound encountered in a particular branch, NAV and TO values are equal (refer to Figure 6.2(a) for an example); thus, the attribute evaluator sets NAV accordingly. When the attribute evaluator visits nodes again on the bottom-up pass, it updates NAV according to (6.1). It adds the NAV value of the child node to the TO value of the current node, and stores the result as the NAV value of the current node.

Figure 6.6 illustrates this computation for nodes #3 and #7 (left branch). As an example, assume that DATA frames take 8 ms to transmit, ACK frames take 2 ms, and a SIFS equals 1 ms. Node #7 describes the last frame compound of the primary pattern in Figure 6.5(a), where  $d$  sends the ACK of the second phase (2 ms) in response to the second DATA frame from  $s$  after a SIFS time (1 ms). Thus, the source should wait for 3 ms before it can assume that no ACK has arrived. The same value holds for the NAV, since the network is busy for the SIFS and the transmission time of the ACK frame as well. The parent node #3 describes the frame compound where the source sends the second DATA frame in response to the ACK of the first phase. Again, the TO value is the sum of SIFS (1 ms) and the successive frame. For this compound, the response is a DATA frame (8 ms), so TO equals 9 ms. In contrast to node #7, the NAV differs from the TO value since not only the DATA frame but also the remaining compounds that constitute the rest of the pattern must be protected. The attribute evaluator uses the NAV value of the child node (3 ms) to compute a NAV that is valid until the end of the pattern, namely  $3\text{ ms} + 9\text{ ms} = 12\text{ ms}$ . If a node has several descendants, the largest NAV of its descendants must be used to protect the longest pattern.

In practice, however, computing NAV and TO depends on additional parameters such as code rate, modulation type, or length of a frame's payload (e.g., that of a DATA frame). If these parameters are not known *a priori*, the attribute evaluator cannot compute final values for NAV and TO, and must use worst-case values instead. In this case, the generated code contains function calls for parameters that are evaluated at run-time instead of constant numeric values.

### 6.2.2. Step 2: From trees to a pattern-based language

In the following, I propose MPDL as a specification language for MAC patterns. Patterns consist of frame compounds which can be linked by overlapping. Consider the non-cooperative pattern in Figure 6.2(b) again. The first frame compound describes the RTS/CTS-exchange between source  $s$  and destination  $d$ , where  $d$  acts upon the source's RTS by sending back a CTS. In turn, a received CTS at  $s$  is the action that  $s$  reacts upon by transmitting its data to  $d$ . Thus, the RTS/CTS exchange and the CTS/DATA exchange can be linked since the frames that constitute action and reaction of both compounds are identical. In other words, the reaction of the first frame exchange becomes the action of the second. Before I can describe a frame compound formally in MPDL, I must first establish the notation of roles, IFS, frames, and handlers in MPDL. All roles that users may take

must be declared using the **role** keyword before they can be used in the specification of the pattern.

**role**  $r_1, r_2, \dots, r_n$

Similarly, all IFSs of the pattern must be declared using the **spacing** keyword before being used.

**spacing**  $I_i T_{IFS}$

The frames of the pattern must be declared using the **frame** keyword before being used.

**frame**  $F_i$  { length=...[ bits | bytes ]  
 [ , payload=...[ bits | bytes ] ]  
 [ , combining=[MRC|... ] ] }

Unlike roles, the declaration of frames involves a list of properties (*property=value*) about the fixed length of the frame, either specified in bits or bytes, the worst-case approximation to use for the variable part of the frame (i.e., the payload), and which combining algorithm to use for frames of type  $F_i$ . The length property is mandatory, but payload and combining may be omitted. Then, the frame does not have a variable part and is not considered for combining, respectively. Further PHY parameters may be supplied as properties only when they are always known *a priori*.

The **on** handler defines how a user having role  $b$  reacts to the successful reception of frame  $F_{ab}$ .

**on**  $F_{ab}$  **at**  $b$  **statement** {...} [**else** **statement** {...}]

The reaction must be specified as a *statement*. MPDL provides several statements that reflect typical user actions, e.g., sending frames, waiting to overhear a frame, or retrying a transmission. These statements cause predefined code templates to be inserted into the generated code. Further handlers can be stated in between the curly brackets for linking frame compounds and defining alternatives. The optional **else** part states what needs to be done in absence of  $F_{ab}$  after its time-out has elapsed. Now, a frame compound can be described with a handler using the **send** statement as follows.

**on**  $F_{ab}$  **at**  $b$  **send**  $F_{ba}$  **to**  $a$  [**after**  $I_{ab}$ ] {...}

Here, a user having role  $b$  reacts by sending a frame  $F_{ba}$  to another user having role  $a$  after an IFS. The specification of an IFS is optional. When omitted, the compiler assumes an IFS of zero, i.e., send immediately. The combination of **on** handler and **send** statement formally describes the frame compound depicted in Figure 6.1. Only the first frame deserves special treatment since it initiates a pattern, so it is defined using the **pattern** keyword.

**pattern** FOO **at**  $a$  **send**  $F_{ab}$  **to**  $b$  {...}

The **pattern** keyword declares a pattern called FOO, and it implicitly states the criterion for assuming role  $a$  which is any user whose MAC sub-layer receives a transmit request from the upper layer. The only feasible reaction here is to send the initial frame. I assume

that *all* users wanting to send an initial frame contend using the IEEE 802.11 Distributed Coordination Function (DCF) [69] but MPDL can be extended to use different contention mechanisms as well. Successive **send** statements are protected by the NAV and, thus, cause their frames to be sent without contention.

I now explain how to link frame compounds by using Algorithm 6.1 as an example that formally describes the RTS/CTS-protected pattern shown in Figure 6.2(b). After having specified a single frame compound using a combination of **on** handler and **send** statement, further handlers can be specified in the statement's body (given by curly brackets following the handler's statement). The body of a handler creates another level in the abstract program tree and must contain a handler for the last frame sent. For example, after the CTS is sent to *s*, the body *must* provide a handler for a CTS at *s*. The compiler enforces such handler and, if it is missing, will reject the specification as incomplete.

Further handlers may be specified in a body. For each additional handler, the compiler creates an alternative branch in the tree. Algorithm 6.2 shows a specification of the simple two-phase cooperative pattern introduced in Figure 6.3. The body of the first **send** statement specifies two handlers. The first one (DATA1 at *d*) links the next frame compound for the primary pattern. The second one (DATA1 at *r*) specifies the behavior of a relaying user *r*. After overhearing the frame DATA1, the relay waits for the required ACK1 frame. The body of the **waitfor** statement defines what to do when the ACK1 arrives (namely reset the automaton) and when it does not (namely retransmit DATA1).

In the simple two-phase protocol, frames of the first phase may be retransmitted in the second phase, and a combining scheme such as MRC should be used (Section 2.2.1). The declaration of DATA1 allows to specify a combining strategy (combining=MRC). The compiler then knows that it must generate code for buffering DATA1 frames at the destination, and code for combining DATA1 frames.

A compiler frontend for the above language definition can be automatically generated using powerful tools such as Eli [27], a publicly available tool chain for compiler construction. In the following section, I show how to use Eli to generate compiler backends for different targets.

### 6.3. Compiler backends

Although MPDL helps the protocol designer to quickly specify even cooperative MAC protocols, it is only the first step in automating the development process. To further support the developer, in a second step, the protocol description has to be automatically analyzed and the program code has to be generated. I describe how to build backends that automatically analyze a protocol specification and generate code for simulators and prototypes.

---

**Algorithm 6.1** MPDL specification of the RTS/CTS-protected pattern shown in Figure 6.2(b)

---

```
role S, D;

spacing SIFS 10us;
spacing DIFS 16us;

frame RTS {length=160bits};
frame CTS {length=112bits};
frame DATA {length=224bits , payload=2312bytes};
frame ACK {length=112bits };

pattern SIMPLE at S send RTS to D {
  on RTS at D send CTS to S after SIFS {
    on CTS at S send DATA to D after SIFS {
      on DATA at D send ACK to S after SIFS {
        on ACK at S done after DIFS else retry
      } else reset # reset when no DATA
    } else retry # retry when no CTS
  }
}
```

---

---

**Algorithm 6.2** MPDL specification of the simple two-phase cooperative pattern shown in Figure 6.3

---

```
role S, D, R;

spacing SIFS 10us;
spacing DIFS 16us;

frame DATA1 {length=272bits , payload=2312bytes ,
  combining=MRC};
frame DATA2 {length=272bits , payload=2312bytes};
frame ACK1 {length=112bits};
frame ACK2 like ACK1;

pattern COOP2PHASE at S send DATA1 to D {
  on DATA1 at D send ACK1 to S after SIFS {
    on ACK1 at S send DATA2 to D after SIFS {
      on DATA2 at D send ACK2 to S after SIFS {
        on ACK2 at S done after DIFS else retry
      } else reset # reset when no DATA2
    } else waitfor DATA1 # wait for the relay
  };

  on DATA1 at R waitfor ACK1 { # alternative
    on ACK1 at R reset else send DATA1 to D {
      on DATA1 at D send ACK2 to S after SIFS {
        on ACK2 at S done after DIFS else retry
      } else reset; # reset when still no DATA1
      on ACK2 at R reset else reset
    }
  }
}
}
```

---

### 6.3.1. Synthesis of patterns

During syntactic analysis the compiler tries to derive the supplied specification from the start symbol of the corresponding context-free grammar. For this, it applies the productions of the grammar to the non-terminal symbols. This process can be visualized in the form of a parse tree whose interior nodes represent productions, and whose leaves consist of terminal and non-terminal symbols [2]. Unlike the abstract program tree which is a concise representation of the specification, the parse tree closely resembles the productions of the context-free grammar. Figure 6.7 shows a part of such a parse tree for the specification in Algorithm 6.2. A compiler generated by the Eli system constructs a parse tree during

---

**Algorithm 6.3** Set of LIDO rules required for computing NAV and TO values in the parse tree of Figure 6.7.

---

**ATTR** NAV, TO: **int**;

**ATTR** Duration: **int**;

**RULE:** Declaration ::= 'spacing' DefSpacingIdent Duration **COMPUTE**  
 ResetDuration(DefSpacingIdent.Key, Duration);

**END;**

**RULE:** Declaration ::= 'frame' DefFrameIdent FrameBody **COMPUTE**  
 FrameBody.Key = DefFrameIdent.Key;  
 ResetDuration(DefFrameIdent.Key, **DIV** (GetLength(DefFrameIdent.Key, 0),  
 GetRate(GetMode(DefFrameIdent.Key, NoKey), 0))) <-  
 FrameBody.NumProperties;

**END;**

**RULE:** After ::= 'after' UseSpacingIdent **COMPUTE**  
 After.Duration = GetDuration(UseSpacingIdent.Key, 0);

**END;**

**RULE:** SendStmt ::= 'send' UseFrameIdent 'to' UseRoleIdent After HBody  
**COMPUTE**  
 SendStmt.TO = **ADD**(GetDuration(UseFrameIdent.Key, 0), After.Duration);  
 SendStmt.NAV = **ADD**(SendStmt.TO, **CONSTITUENTS** SendStmt.NAV  
**WITH** (**int**, Maximum, **IDENTICAL**, **ZERO**));

**END;**

---

syntactic analysis. Subsequent compiler stages like semantic analysis and code generation traverse the parse tree to compute relevant attributes (such as NAV and TO) and emit target code as the nodes are visited. Since both semantic analysis and code generation depend on computations in the parse tree, I use NAV and TO as examples for explaining how to specify computations formally and how the backend evaluates them. Note that the

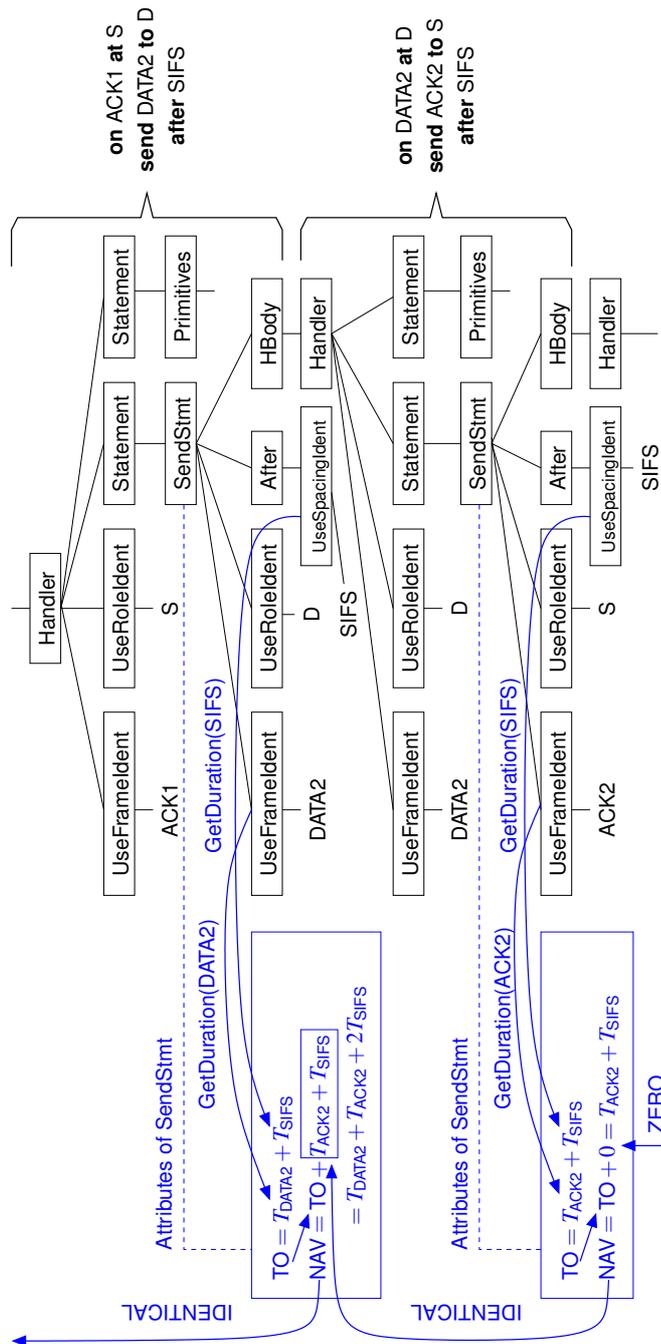


Figure 6.7.: Parse tree for the code fragment “on ACK1 at S send DATA2 to D after SIFS { on DATA2 at D send ACK2 to S after SIFS” as it appears in the primary pattern of Algorithm 6.2. The inner nodes correspond to productions of the context-free grammar, whereas the leaf nodes correspond to applied occurrences of identifiers. Computations are indicated by arrows, whose associated LIDO rules are shown in Algorithm 6.3.

Eli tool chain automatically generates the implementation of semantic analysis and the target code generation of the backend from this formal specification. Eli provides LIDO, a language for the specification of computations in trees. The interested reader is referred to the excellent documentation of the Eli tool chain [46].

Algorithm 6.3 summarizes the rules that are required to compute NAV and TO as explained in Section 6.2. Rules correspond to productions of the context-free grammar, at which computations are to be performed. The attribute evaluator, upon a tree walk, performs the computation when visiting a node. For example, when the evaluator visits the declaration of an IFS (i.e., the production containing the **spacing** terminal symbol on the right-hand side), the corresponding LIDO rule tells the evaluator to fetch the value of this IFS, represented by the leaf node `Duration`, and to store it in a property table. These tables are important data structures of the compiler and, as the name suggests, hold properties that are bound to identifiers (like `SIFS` in Algorithm 6.2). Identifiers are assigned unique keys for indexing these property tables, where identical identifiers have identical keys. For spacings, `DefSpacingIdent` represents the *defining* identifier whose key is used to store properties that are parsed from the declaration (e.g., `Duration` of an IFS). When the identifier is later on *used* in some statement, indicated by the symbol `UseSpacingIdent`, the compiler retrieves all properties needed for the computation using the key associated with it. It must be assured that all properties are stored (i.e., all declarations have been evaluated) *before* they are used in computations by appropriate dependencies. Dependencies state the order in which computations need to be evaluated, allowing Eli to deduce an efficient tree walk strategy from it. Eli recognizes trivial dependencies automatically, but in some cases they need to be specified explicitly.

When the evaluator visits the `Declaration` node of a frame, it computes the duration of the frame by dividing its length by the data rate used for transmitting it. This computation requires both rate and length properties to have been parsed and stored in the property table already, so the sub-tree rooted at `FrameBody` needs to be evaluated first. To enforce this, the computation depends on the `NumProperties` attribute of `FrameBody` (indicated by the “< –” operator in Algorithm 6.3). This attribute is used for counting the number of properties in the `FrameBody` sub-tree and is only available after its evaluation. Not having to implement the evaluator by hand eases the development of a compiler considerably.

TO and NAV are attributes of `SendStmt` nodes. According to (6.2), TO is computed by adding the IFS and the duration of the following frame, both of which are supplied in a **send** statement. Looking at the `SendStmt` production, `UseFrameIdent` is the applied occurrence of a frame identifier whose key the evaluator uses to look up the duration from the property table. The same is done for the spacing in a separate rule associated with the `After` production (Algorithm 6.3). The resulting value is stored in the `SendStmt`'s TO attribute, where it can be read from later on during code generation.

The computation of the NAV attribute is slightly more involved as this requires the summation of *all* TO values in the lower context of the tree as explained in Section 6.2.1. Since the sub-tree of `SendStmt` may contain several instances of the NAV attribute (one for each `SendStmt` occurring), the evaluator must be advised how to combine several NAV

values into one. For this, LIDO provides the `CONSTITUENTS` construct with its associated `WITH` clause that takes four arguments – namely *type name*, *combine-function name*, *single-function name*, and *null-function name* – which I briefly explain here. *Type name* specifies the data type of the attributes to be accessed and combined. *Combine-function name* specifies the function to use for combining two values into one. As an example where this is used, consider the root node in Figure 6.6. The two sub-trees are spanned by the two handlers following the initial `SendStmt` in Algorithm 6.2 on the same level. Since I want the maximum of both NAV values to be propagated, I specify the `Maximum` function to be used for combining. The evaluator will call this function on both NAV values whose result is then added to the `TO` value of the initial `SendStmt`. If there are more than two sub-trees, only the first two of them are evaluated and the resulting value is combined into one. This value is then used for combination with the following sub-tree, following a left-to-right order. If there is only one sub-tree, e.g., node #3 in Figure 6.6, the *single-function name* specifies how to process a single value from the lower context. In this example, I merely need to pass it on, thus I specify the pre-defined `IDENTICAL` function that returns whatever value is given to it. Figure 6.7 shows how the values are propagated through the tree and the functions used for it. Finally, *null-function name* specifies the value to be returned at the lowest context where the attribute NAV occurs. For example, the deepest `send` statement does not have any more children with NAV attributes, thus an initial value for NAV must be generated here. I use the `ZERO` function for this that always returns the constant value zero.

Once NAV and TO values have been computed for all tree nodes, they can be used in code generation. Just like semantic analysis, during code generation the compiler traverses the tree and outputs a pre-defined text fragment for every node visited. These text fragments typically contain placeholders that are substituted with the computed attributes. In summary, the main task in developing a compiler backend consists of writing suitable computations (with their dependencies) and text fragments. The interested reader is referred to the manual of the pattern-based text generator for more details on how to specify text fragments and output them while traversing the tree.

### 6.3.2. Analysis backend

Determining the fundamental performance limits of cooperative MAC protocols can help to identify promising designs right from the start and to sort out inefficient variants *before* implementing them. The analysis backend derive these limits from the specification by applying outage analysis (Section 2.2) to patterns using conditional probabilities.

I denote the outage probability of direct transmission between source and destination by  $P_{\text{Direct}}^{s,d}(R)$  and of SDF transmission between source, relay, and destination by  $P_{\text{SDF}}^{s,r,d}(R)$ . I assume that these outage probability terms are a function of the transmission rate  $R$  which may be selected for a particular frame and defines its robustness. Typically, management frames are encoded using the lowest, i.e., most robust, rate  $R_m$ , whereas frames containing

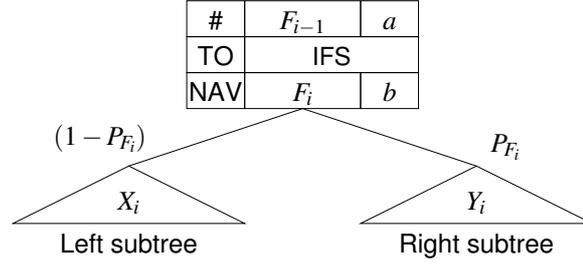


Figure 6.8.: When a frame  $F_i$  of the pattern is transmitted, it can either be decoded correctly at the receiver with probability  $(1 - P_{F_i})$ , causing progression of the left subtree  $X_i$ , or it is in outage with probability  $P_{F_i}$ , causing the time-out to be handled in the right subtree  $Y_i$ .

payload can have higher rate  $R_d$ , e.g., determined by rate adaptation.

For my analysis, I assume that *all* frames of an individual pattern must be successfully transmitted for the conveyed data packet to be received correctly. If the transmission of a single frame  $F_i$  of the pattern is in outage, indicated by the outage probability  $P_{F_i}$ , the entire pattern is in outage. If combining is used, a directly transmitted frame may be in outage in one pattern but the corresponding retransmitted frame in an alternative pattern must then not be in outage for the protocol to succeed. So if a protocol comprises several patterns (such as the one in Figure 6.3), the outage event of a specific pattern  $\Pi_i \in \mathcal{P}$  does not imply the outage event of the entire protocol  $\mathcal{P}$ . This only occurs if *all* patterns  $\Pi_1, \dots, \Pi_n$ , including those exploiting cooperation diversity, are in outage.

If the reception of a frame  $F_{i-1}$  triggers the transmission of a frame  $F_i$ , two cases can occur as depicted in Figure 6.8. If the frame is correctly decoded at the receiver with probability  $(1 - P_{F_i})$ , subtree  $X_i$  defines the following actions. Otherwise, the frame is in outage, observed when the time-out expires, and, hence, the action defined in subtree  $Y_i$  is executed. The overall probability  $P_{O_i}$  comprising both subtrees is given by the following equation.

$$P_{O_i} = (1 - P_{F_i})P_{X_i} + P_{F_i}P_{Y_i} = 1 \quad (6.3)$$

This must be a sure event since a frame either arrives correctly or it does not. Subtrees themselves can recursively consist of the tree in Figure 6.8, allowing to substitute  $P_{X_i}$  and  $P_{Y_i}$  with the appropriate  $P_O$ -values for the subtrees.  $P_{X_i}$  or  $P_{Y_i}$  may be one, if the action defined in the subtree does not cause a frame to be sent, for which, obviously, another binary decision must be defined. Since all patterns terminate eventually, there will be leaf nodes as substitutes for some  $X_i$  or  $Y_i$ . Computing  $P_O$  for the root must yield one, since the root covers the entire sample space. A path  $J_i$  from the root node to some leaf in the tree corresponds to a particular event when transmitting a pattern of the protocol, resulting either in success or outage. The probability  $P_{J_i}$  of a path  $J_i$  being taken can be obtained by expanding the recursive function given in (6.3) for the root of the tree. The expansion is a sum of *all*  $P_{J_i}$ . The outage probability of the protocol can then be determined either by

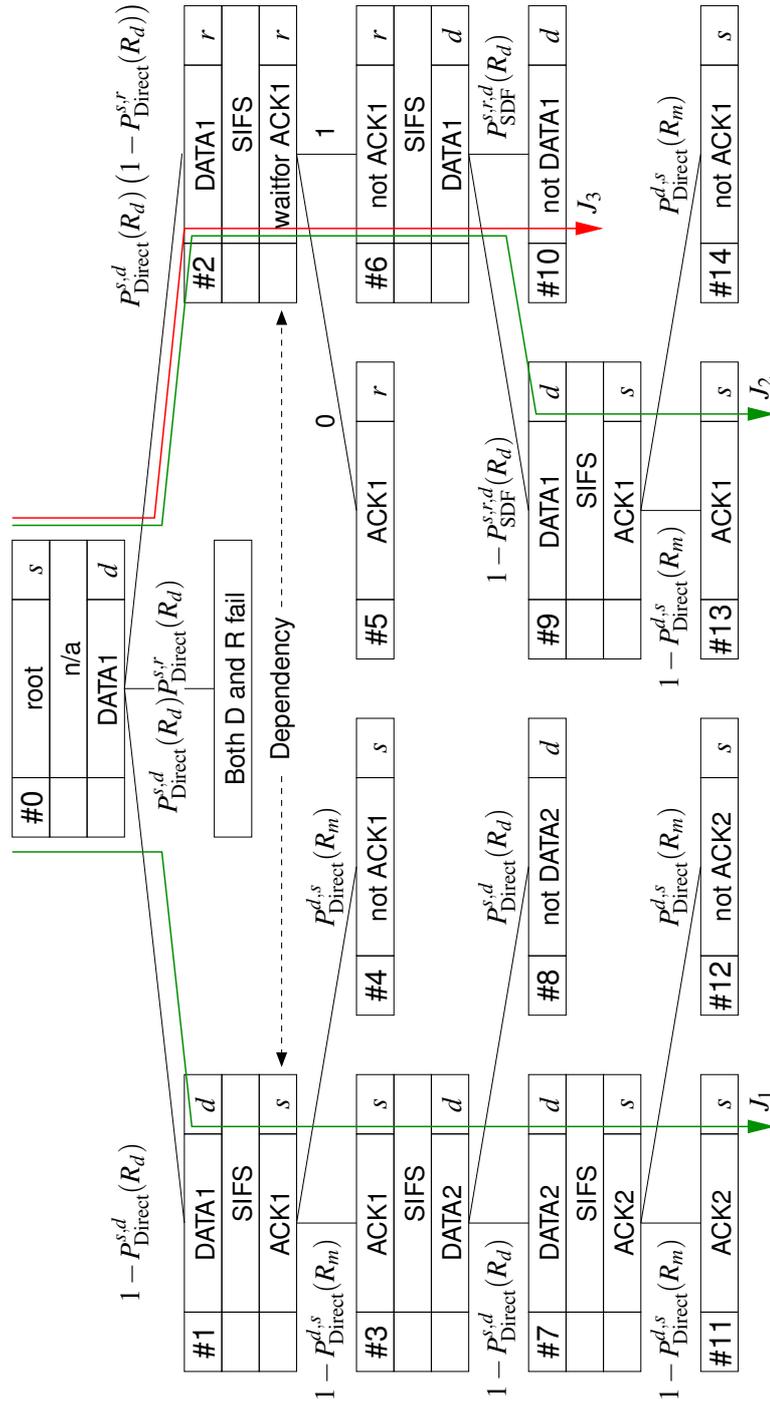


Figure 6.9.: This abstract program tree corresponds to the simple two-phase cooperative protocol shown in Figure 6.3. The branches are annotated with the success/outage probability terms that are generated by the compiler during tree traversal. A path  $J_i$  from root to leaf can either result in a successful data transmission (e.g., DATA1 is successfully conveyed to the destination on  $J_1$  and  $J_2$ ) or in an outage event (e.g.,  $J_3$ ).

summing up the probabilities of all *failing* paths or by summing up the probabilities of all *successful* paths and computing the complement.

Figure 6.9 illustrates this approach for the simple two-phase cooperative protocol shown in Figure 6.3. It shows the corresponding abstract program tree annotated with the success or outage probabilities of the particular branches by the compiler. For path  $J_1$  no transmission errors occur and all frames arrive successfully at the intended receivers, corresponding to the primary pattern shown in Figure 6.3(a). Every branch is labeled with the probability of being taken, where  $P_{A_i}^{C_i}(R_i)$  is the outage probability associated with the transmission; it is either a closed-form expression or must be solved numerically (e.g., M-QAM with  $M > 4$  and realistic codes). Equation 6.4 shows the probability of path  $J_1$ , which due to the nature of  $J_1$  is a success probability.

$$P_{J_1} = \left(1 - P_{\text{Direct}}^{s,d}(R_d)\right)^2 \left(1 - P_{\text{Direct}}^{d,s}(R_m)\right)^2 \quad (6.4)$$

The branch at the root node describes two parallel actions and is, thus, different from all branches that follow, which really describe mutually exclusive alternatives as motivated in Figure 6.8. This is because a single frame may arrive independently at two different nodes, destination and relay. To satisfy the fundamental property that the sum of a node's branches corresponds to a sure event, the relay branch needs to be conditioned further. Note that one needs a relay transmission only if DATA1 did not arrive at the destination correctly. Thus, the probability of the relay branch actually implies the outage probability  $P_{\text{Direct}}^{s,d}(R_d)$ . Then, only the outage event for the transmission to both nodes remains, which is  $P_{\text{Direct}}^{s,d}(R_d)P_{\text{Direct}}^{s,r}(R_d)$ . In summary, the sample space at the root is correctly modeled as

$$P_{O_{\text{root}}} = \left(1 - P_{\text{Direct}}^{s,d}(R_d)\right) P_{X_0} \quad (6.5)$$

$$+ P_{\text{Direct}}^{s,d}(R_d) \left(1 - P_{\text{Direct}}^{s,r}(R_d)\right) P_{Y_0} \quad (6.6)$$

$$+ P_{\text{Direct}}^{s,d}(R_d) P_{\text{Direct}}^{s,r}(R_d). \quad (6.7)$$

The compiler can identify the dependency between destination  $d$  and relay  $r$  (indicating that further conditioning is required) since first both nodes simultaneously process the same received data frame and second the reaction of  $d$  (namely, to send an ACK) is anticipated at  $r$  as well. Under the assumption of (6.6), this dependency implies a zero probability for the ACK being overheard, since the additional condition  $P_{\text{Direct}}^{s,d}(R_d)$  implies that an ACK is not transmitted at all.

Finally, I illustrate how the compiler chooses the correct outage probability equation depending on whether combining is used or not. When DATA1 is transmitted by the relay (see node #6), the compiler knows that, this time, the outage probability of SDF is required instead of direct transmission for two reasons: (1) DATA1 is flagged in the specification as combinable and (2) the above dependency indicates that a previous DATA1 transmission was unsuccessful. The channel information is given through the root node ( $s, d$ ), node #2

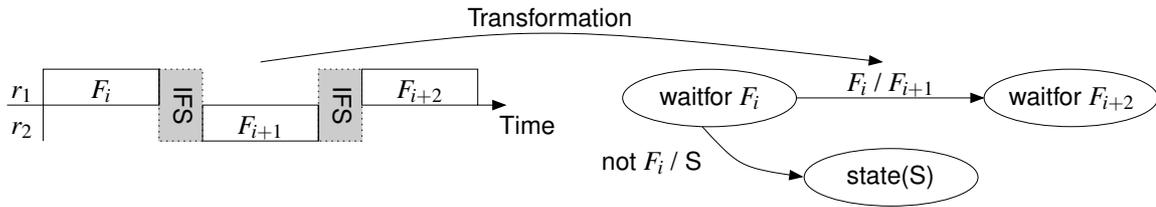


Figure 6.10.: Transformation of a linked pair of frame compounds involving roles  $r_1$  and  $r_2$  into states of a MAC automaton for role  $r_2$ .

$(d, r)$ , and node #6  $(r, d)$ .

### 6.3.3. Simulator and SDR backends

Theoretic performance analysis as discussed in the previous section yields only bounds for performance, but cannot give exact results to what is achievable by an *implemented* protocol. Integrating user cooperation diversity into, e.g., IEEE 802.11 requires additional relaying protocols at the MAC layer, additional control channels, as well as coding and combining extensions at the physical layer [87]. The performance of such functions, their overhead, control latency, integration side-effects and, finally, the performance of the complete system has to be studied with realistic scenario assumptions. For this, simulations and prototypes are needed. I now illustrate how a backend for a simulator can be built that generates an implementation of the corresponding MAC automaton and receiver model from the specification.

#### MAC automaton

The compiler generates a single MAC automaton that involves all roles. It begins code generation with an initial automaton that always provides the four basic states *idle*, *contend*, *quiet*, and *busy*. These states are necessary for the implementation of contention, e.g., according to IEEE 802.11 DCF (*contend* state) with virtual carrier sensing through the NAV (*quiet* state) and a final IFS before entering another contention period (*busy* state). The compiler introduces additional states by traversing the abstract program tree and applying transformations according to Figure 6.10. The figure shows how states are generated for the automaton to handle a frame  $F_i$  at its receiver. I discuss the general case where  $F_i$  and  $F_{i+2}$  do not correspond to the first and last frame of a pattern, respectively. Algorithm 6.4 shows a specification for the linked frame compounds. The dotted parts of Algorithm 6.4 embed the two frame compounds into a complete pattern, and are not required to perform the transformation step shown in Figure 6.10.

At some point in time, the compiler encounters a node whose action is given by the **send** statement that causes frame  $F_i$  to be sent to a user in role  $r_2$ . Consequently, the automaton must provide a state in which  $r_2$  accepts the frame  $F_i$ . This state is called *waitfor  $F_i$* , and two

---

**Algorithm 6.4** A specification that yields the linked frame compounds shown in Figure 6.10.

---

```

on ... at  $r_1$  send  $F_i$  to  $r_2$  {
  on  $F_i$  at  $r_2$  send  $F_{i+1}$  to  $r_1$  after  $T_{IFS}$  {
    on  $F_{i+1}$  at  $r_1$  ...
  } else  $S$ ;
} else ...;

```

---

events must be handled there, leading to two transitions in the automaton. Either the frame arrives, in which case another handler must be present in the body of the **send** statement that specifies the reaction. Here, it is to reply with a frame  $F_{i+1}$ , which establishes the link between two frame compounds. If  $F_{i+1}$  triggers another frame  $F_{i+2}$ , the automaton enters a waiting state in which  $r_2$  can accept it. Alternatively,  $F_i$  does not arrive at the MAC layer because it was incorrectly decoded or never sent. In this case, the **else** part of the handler defines the alternative action by a statement  $S$ . The resulting state of the automaton depends on  $S$ . For example, the **reset** statement creates a transition to the *idle* state, whereas the **retry** statement would cause the user to contend again for retransmission, thus, creating a transition to the *contend* state. Furthermore, statements such as **retry** or **reset** do not only create transitions, the compiler also inserts predefined basic operations into the implementation. For example, a user does not retry ad infinitum. Instead, its MAC implementation uses retry counters and limits to give up and signal an error to the upper layer. The backend provides these basic operations in the form of method templates. During transformation the compiler inserts calls to these methods into the generated code. Porting the backend to other targets requires to rewrite those templates for all statements that MPDL provides.

### Decider

The decider is a module used only in simulators to model the decoding and combining process at the receiver. It determines whether a frame was correctly received or not. It takes this decision based on, e.g., signal-to-noise ratio, modulation, and FEC. Figure 6.11 shows a flow diagram of an MRC-capable decider, a crucial part of each SDF cooperation simulator. Upon reception of a frame, the decider verifies its correctness by comparing the instantaneous SNR of the frame to the associated threshold. In practical systems, the threshold comparison at the decider reflects the CRC. Correct frames are passed on to the MAC layer, whereas incorrect ones are buffered. However, only those frames are buffered that have been declared as combinable in the specification. This is to avoid the unnecessary buffering of frames for which cooperation should not be applied. The compiler includes a type field in every frame according to the specification, and uses it for comparison in the generated code. The decider always associates buffered frames with a time-out that, when triggered, causes the decider to remove the frame from the buffer since there is no use in

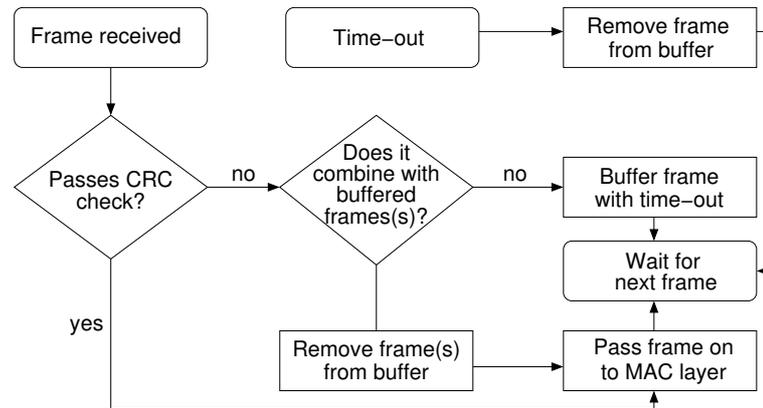


Figure 6.11.: Flow diagram of a Maximal Ratio Combining decoder, whose implementation can be generated automatically from the specification of the pattern.

keeping frames for which no more redundancy will arrive. In practical systems, the CRC decides whether combination with a buffered frame was successful. So the basic operations, which the compiler must generate code for, are verifying frames (CRC), buffering frames, unbuffering frames, and combining frames. These operations are not specific to the target and must be implemented in any simulator framework, even when a different combining algorithm is to be used.

## 6.4. Conclusions

Formalizing MAC patterns allows to automate the most error-prone parts of MAC protocol development. Also, it is the first design approach taking the specifics of *cooperative* communication into account. To this end, I showed how to construct a variety of compiler backends for analysis and code generation. Engineers can use the proposed compiler-assisted method to, in a first step, automate the theoretical performance analysis of the specified protocols. Based on the powerful but non-trivial method of outage analysis, the analysis backend is a large help for engineers to find the most efficient protocol designs. In a second step, these designs can be studied by simulation using program code automatically generated from the simulator backend. Finally, engineers can employ an SDR backend to generate code for an SDR, thus obtaining either a prototype or running system, all from the same specification.

MPDL and my compiler-based tool chain already provide intuitive and fast design, analysis, and implementation of *cooperative* MAC protocols. Nonetheless, based on the given examples, MPDL compiler backends for various simulators and SDR platforms can be constructed. The challenge in building compiler backends is the definition of computations that transform the specification into target code. With the help of generating tools such as Eli, this process is greatly simplified.

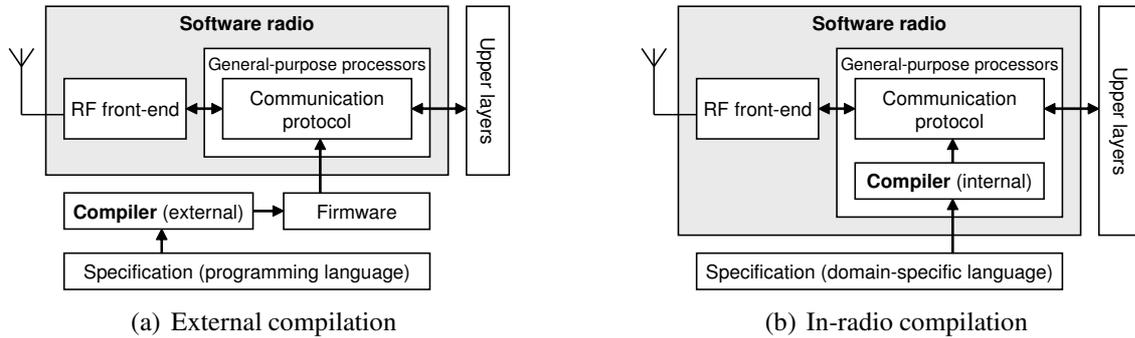


Figure 6.12.: A software radio consists of an analog radio front-end and one or more general-purpose processors that execute the communication protocol (firmware). Conventionally, the firmware is compiled and linked externally, and then downloaded onto the software radio. With this approach, the firmware can be updated by compiling specifications of communication protocols on the software radio itself, not requiring additional development hardware.

The proposed compiler for protocol generation could also be used directly on SDRs to provide a powerful prototyping box. Currently, engineers prototype new network protocols according to Figure 6.12(a). The firmware that executes on the SDR is developed and compiled on an external computer, and then downloaded onto the software radio's internal storage device (e.g., flash memory). Due to the flexibility of SDRs, even the proposed compiler can be integrated into the SDR along with the communication protocol. Then, specifications of communication protocols can be compiled on the software radio itself (which I call *in-radio compilation*), and then replace an existing communication protocol. Figure 6.12(b) illustrates this idea. By integrating the compiler into the software radio, only the specification needs to be supplied externally. This way, the step of compiling the firmware externally and downloading it is replaced by downloading a specification as input for the software radio's compiler, which then generates executable code and updates the firmware accordingly. Such approach is possible since a compiler for a domain-specific language is itself software that can be executed on the general-purpose processors that constitute a software radio. Therefore, the software radio can generate code for itself from a specification in a suitable domain-specific language. Since domain-specific languages are tailored to specialized problem domains, specifications written in domain-specific languages are not only shorter than equivalent specifications in programming languages (e.g., C or Assembler), they are even less error-prone as they force engineers to focus on the pattern itself, not on the complex and repeating implementation details (such as timers or state transitions).



## 7. Conclusions and future work

Cooperative relaying has become a mature technology for enhancing the capacity of next generation wireless systems. Valentin [85] demonstrated that, in the basic three-node configuration, the theoretical gains known from theory even appear in practice, albeit realistic assumptions such as limited CSI feedback, erroneous control frames for signaling, and limited network connectivity degrade these gains. Prototyping efforts by Hunter et al. [37] further back up this insight. Although all this work was important to develop a basic understanding of cooperative relaying, it was not yet sufficient to argue that cooperative relaying remains feasible if applied at larger scale in wireless multi-hop networks. Important questions remained that I answered in this thesis.

**Improving spatial reuse** Cooperative relaying does not only improve a single transmission in a wireless network, it even benefits the *entire* network's capacity. My spatial analysis showed that cooperative relaying techniques that involve up to two additional nodes consume, for practically interesting scenarios, even less space than direct transmission or non-cooperative relaying at the same capacity and robustness. The exponential diversity gain is large enough to reduce the spatial consumption by two orders of magnitude, yet it achieves the same outage capacity as direct transmission or non-cooperative relaying. This improvement leaves enough room to even increase the capacity on a single link with less spatial consumption. This is useful for soft real-time traffic where correct reception in the first attempt is more valuable than retransmissions.

**Reducing costs for unicasting** Multi-hop networks can reduce the multiplexing loss of cooperative relaying for unicasts. To reduce the multiplexing loss, my two-for-one MAC extension only requires information about the second hop from the routing layer. A single cooperative retransmission then provides the diversity gains at two nodes on the path instead of only one when restricted to one-hop neighborhood information. As my approach does not require to change the routing protocol, modifications at layer 1 and 2 of the protocol stack suffice.

**Reducing costs for broadcasting** For broadcasts, retransmissions need to occur anyway in the broadcast set. Therefore, nodes already receive multiple transmissions. By simply combining corresponding transmissions, nodes can benefit from the diversity gains without redundant transmissions. Although this improves the broadcast latency, the processing complexity increases – an issue that can be solved by ASICs.

The probabilistic model with cooperative diversity can be integrated into existing broadcasting or MCDS heuristics. To highlight some example extensions: When computing the metric of a candidate node for adoption into the backbone, lookahead heuristics are frequently used [28]. Which nodes such a lookahead would inform depends on the model and that can be easily replaced with the probabilistic model just as with standard neighboring arguments. Correspondingly in pruning heuristics [94], a node can only be removed if its contribution to the probability mass of all other nodes is not essential and the required threshold  $\varepsilon$  is still reached. Also, the distributed versions of building broadcast sets (e.g., performing a leader election among all suitable candidates for each iteration [19]) immediately work with my promotion rules.

**Compiler-assisted MAC development** The formal description of wireless communication protocols by patterns allows to design a simple specification language to automatically generate the corresponding communication protocol. This enables the design of a compiler for the specification language based on the pattern formalism. Such approach can even be used to generate code for the wireless communication device on the device itself (*in-radio compilation*).

There is a trade-off associated with this compiler-assisted approach. The development of a new specification language and a suitable compiler initially increases development time. Developers must first learn how to use the language, and, second, they must develop a backend for their target platform. However, this cost will pay off since the compiler-assisted approach abstracts standard approaches and allows to concentrate on the design of the protocol. By using a specification language such as MPDL, evaluation not only becomes faster and more convenient, it also becomes more comprehensible by the community. When new patterns for cooperative relaying are invented and specified in a commonly used specification language, performance results can be easily transferred to other environments, e.g., from a simulator to an SDR. Engineers are not forced to use a particular environment but can simply generate a pattern's code for the environment that they are fond to use. Put briefly, compiler-assisted development is a solid basis stimulate the development of promising communication approaches.

During my work towards large wireless multi-hop networks employing cooperative diversity, I revealed the following issues that require further investigation.

**Expected interference** The analytical model of interference still offers room for improvement. Receivers of concurrent transmissions also span guard zones in which other transmissions are inhibited. Since my derivation ignores these guard zones, I overestimate the overall interference of the random network. This makes my solution practicable only for deriving *lower* bounds on the ad hoc network capacity.

---

**Relay selection** Two-for-one cooperation requires a relay to assist the receiving nodes in both the first and second hop of a multi-hop transmission. With the two-for-one cooperation protocol, each node that overhears a transmission must find both receiving nodes in its neighborhood table to be a potential two-for-one relay. If more than one potential relay exists, I did not discuss how to choose among them. Future work should identify promising relay selection schemes for two-for-one cooperation which, ideally, select a relay that maximizes the data rate while minimizing the spatial consumption.

**Spatial reuse in practice** I showed theoretically that cooperative relaying achieves better spatial reuse than non-cooperative transmissions. This result should also be observable in practice. This validation requires large testbeds for measuring the spatial consumption of cooperative relaying.

**Asymmetry of PCDB sets** The PCDB heuristic creates broadcast sets that are no longer CDSs. By aggregating energy, the newly informed nodes depend on the nodes that previously transmitted. While this is unproblematic if the same nodes initiate the broadcast with which the heuristic started, the same broadcast set may no longer achieve a high delivery ratio for a different node. This is due to the asymmetry of cooperative links. Imagine that source and relay are close to each other but both are far apart from the destination. In that case, the distance between source and destination as well as relay and destination is approximately equal. If source and relay can only reach the destination jointly, the reverse direction does not work as the destination does not have a partner to create the cooperative diversity gain. Consequently, any node requires its own broadcast set with PCDB. It is an open question how to construct a CDS exploiting both probabilistic model and cooperative diversity to have only a single broadcast set regardless which node initiates the broadcast.

**PCDB in practice** The delivery ratios of PCDB in the simulation results are close to 100 % since the heuristic uses a fading model that closely resembles that in the simulator. It must be answered how well PCDB performs in practice where the fading model does not always hold. For practical use, a distributed variant of PCDB is also needed that only relies on local neighborhood information.



# A. Mathematical derivations

## A.1. Area of two circle's intersection

The common area of two equally sized circles is given by twice the area of the circular segment spanned by the intersection points. The circular segment's area is given as

$$A_{CS} = \frac{g^2}{2}(\theta - \sin \theta) \quad (\text{A.1})$$

where  $\theta$  denotes the segment's angle (refer to Figure A.1). The angle  $\theta$  can be determined using the inverse tangent

$$\theta = 2 \tan^{-1} \frac{y}{x} \quad (\text{A.2})$$

where  $x$  and  $y$  are the coordinates of the upper intersection point given by [22]

$$x = \frac{s}{2} \text{ and } y = \frac{1}{2} \sqrt{4g^2 - s^2}, \quad (\text{A.3})$$

allowing us to express  $\theta$  in terms of  $s$  and  $g$  as follows.

$$\theta = 2 \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \quad (\text{A.4})$$

Inserting (A.4) into (A.1) yields an unwieldy term which can fortunately be simplified by exploiting  $\sin 2z = (2 \tan z)/(1 + \tan^2 z)$  in which case our choice of (A.2) allows us to simplify

$$\sin \left( 2 \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} \right) = \frac{s}{2g^2} \sqrt{4g^2 - s^2}. \quad (\text{A.5})$$

Noting that the intersection of the two circles is  $2A_{CS}$  we finally arrive at

$$A_{IS} = 2g^2 \tan^{-1} \sqrt{\frac{4g^2}{s^2} - 1} - \frac{s}{2} \sqrt{4g^2 - s^2}. \quad (\text{A.6})$$

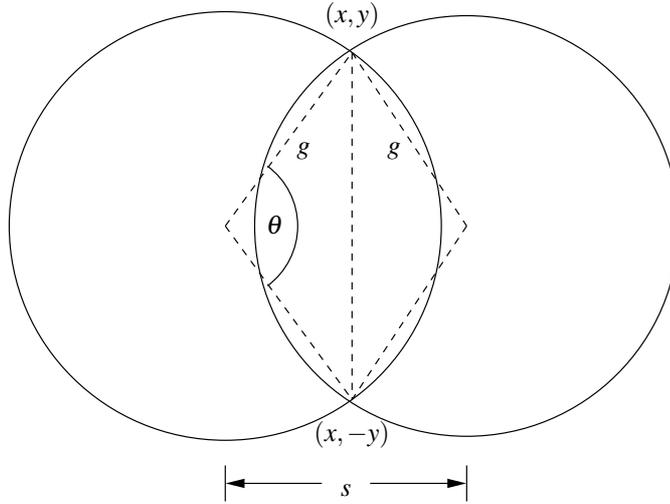


Figure A.1.: Intersection of two circles with equal radius  $g$  whose center points are a distance of  $s$  apart.

## A.2. Area of three circle's intersection

Three overlapping circles each with equal radius  $g$  whose center points form an equilateral triangle with side length  $s$  span an equilateral circular triangle as shown in A.2. To derive a formula for its area, we first look at the generic circular triangle which can be viewed as an ordinary triangle plus three circular segments. The area of the ordinary triangle, with  $c_i$  being the side lengths of the triangle, is given by Heron's formula as

$$\frac{1}{4} \sqrt{(c_1 + c_2 + c_3)(c_2 + c_3 - c_1)(c_1 + c_3 - c_2)(c_1 + c_2 - c_3)}. \quad (\text{A.7})$$

The area of a circular segment is given by (A.1) and, using simple geometry and trigonometry, can be expressed in terms of the chord length and radius. Summing it up for all three circular segments leads to

$$\sum_{i=1}^3 \left( g_i^2 \sin^{-1} \frac{c_i}{2g_i} - \frac{c_i}{4} \sqrt{4g_i^2 - c_i^2} \right). \quad (\text{A.8})$$

For an *equilateral* circular triangle all chord lengths are equal so its area  $A_{\text{ECT}}$  is given by adding (A.7) and (A.8) with  $c_1 = c_2 = c_3$ .

$$A_{\text{ECT}} = \frac{\sqrt{3}}{4} c^2 + 3 \left( g^2 \sin^{-1} \frac{c}{2g} - \frac{c}{4} \sqrt{4g^2 - c^2} \right) \quad (\text{A.9})$$

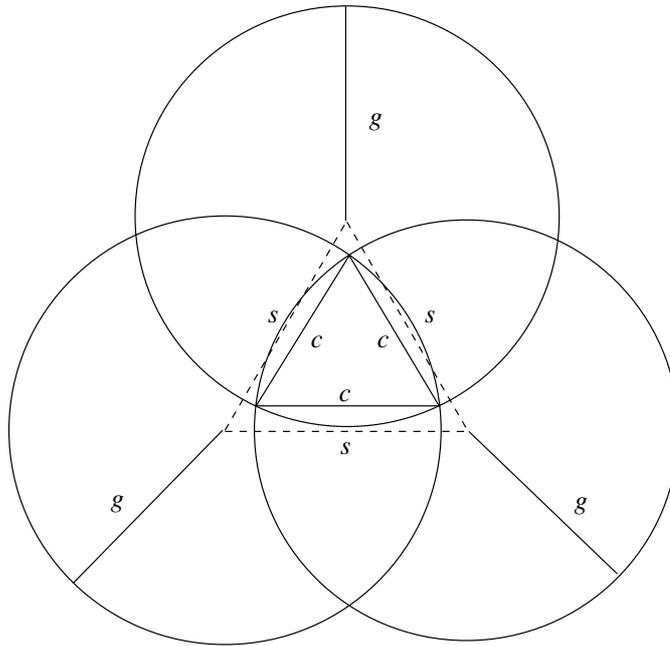


Figure A.2.: Intersection of three circles with equal radius  $g$  whose center points are pairwise a distance of  $s$  apart.

For the equilateral circular triangle, the chord length has been found to be [22]

$$c^2 = 3g^2 - \frac{s^2}{2} - s\sqrt{3g^2 - \frac{3s^2}{4}}. \quad (\text{A.10})$$



## B. Occurrence probabilities of diamonds in random networks

Valentin et al. [86] studied the probability with which WFDs and SFDs occur in random networks to derive occurrence-conditioned outage capacities. For this thesis, I briefly summarize the results from [86] to show that both WFDs and SFDs occur in random wireless networks. This justifies the design of particular protocols that are tailored to these topologies like the routing-informed MAC protocol I proposed in Section 4.2.

We develop a model to derive the occurrence probability for the triangle and diamond configurations. For this we define the following thresholds:

1. Detection threshold ( $\text{th}_S$ ) to *sense* a transmission
2. Decoding threshold ( $\text{th}_M$ ) to receive low-rate *management* frames required to establish paths
3. Decoding threshold ( $\text{th}_D$ ) to receive high-rate *data* frames

The  $\text{th}_D$  threshold varies depending on the modulation type and code rate used. It obviously holds that  $\text{th}_S < \text{th}_M < \text{th}_D$ . Since we want to estimate an upper bound on the occurrence, we choose  $\text{th}_M = \text{th}_D$  for the remainder of this chapter, with values  $\text{th}_S = 4.5$  dB and  $\text{th}_M = 6$  dB according to a typical IEEE 802.11a/g WLAN transceiver specification [4].

### B.1. Modeling cooperative triangles and diamonds

We consider cooperative triangle and diamond topologies along two-hop paths  $a \rightarrow c \rightarrow d$  only. To determine their occurrence probabilities, we normalize all occurrences by the number of so-called *base configurations* that constitute *any* triangle and diamond. Nodes form a base configuration if (1) at least management frames can be exchanged between source and relay  $c$  on the path ( $\gamma_{a,c} \geq \text{th}_M$ ) as well as between relay  $c$  and destination ( $\gamma_{c,d} \geq \text{th}_M$ ); (2) the relay  $b$  not on the path must receive data frames without being helped itself ( $\gamma_{a,b} \geq \text{th}_D$ ) to satisfy the SDF constraint.

Note that every triangle and diamond contains a base configuration, but not every base configuration can be extended to a triangle or diamond. For triangles, it holds that  $b = c$  and that  $\gamma_{a,d} \geq \text{th}_S$ , i.e., the destination must be able to at least sense the source.  $\gamma_{a,d} + \gamma_{c,d} \geq \text{th}_D$  must also hold so combining two data frames results in a correct frame (refer to 2.2.1).

In contrast, for all diamonds it must hold that  $b \neq c$  and  $\gamma_{b,d} \geq \text{th}_S$ : The receiver at the following hop  $d$  must be able to at least sense the relay  $b$ .

With the WFD the direct link is too weak to be useful ( $\gamma_{a,d} < \text{th}_S$ ), but relay  $c$  may sense transmissions from relay  $b$  ( $\gamma_{b,c} \geq \text{th}_S$ ). The destination  $d$  can combine both relay transmissions ( $\gamma_{c,d} + \gamma_{b,d} \geq \text{th}_D$ ). Additionally, relay  $c$  can combine transmissions from  $a$  and relay  $b$  ( $\gamma_{a,c} + \gamma_{b,c} \geq \text{th}_D$ ). The SFD achieves additional diversity with respect to WFD thanks to direct sensing from  $a$  to  $d$  ( $\gamma_{a,d} \geq \text{th}_S$ ). The destination  $d$  can combine this transmission with the ones from both relays at  $d$  ( $\gamma_{c,d} + \gamma_{b,d} + \gamma_{a,d} \geq \text{th}_D$ ). SFD does not require relay  $c$  to receive the transmission from  $a$  with high probability as it allows relay  $c$  to combine  $a$ 's and  $b$ 's transmissions ( $\gamma_{a,c} + \gamma_{b,c} \geq \text{th}_D$ ).

## B.2. Counting in unobstructed and Manhattan grid scenarios

We systematically check all possible combinations of nodes to find all diamonds and triangles by iterating over nodes. In every iteration, the simulator picks one as source  $a$ . Iterating over all neighbors of  $a$  that can receive at least management frames, the simulator picks one as relay  $c$ . For all nodes  $d$  that complement  $a \rightarrow c$  to a two-hop path and  $a \neq c \neq d$ , the simulator then picks a node  $b$  that is neighbor of  $a$  and complements the two-hop path to a base configuration, for which  $b \neq d$ . The simulator counts all base configurations found for later normalization. Now, it may be the case that  $b = c$  where the simulator checks for a triangle and, if found, counts it. For  $b \neq c$  the simulator must check whether a WFD or SFD applies and count the corresponding one.

Any two-hop path can be extended to at most one triangle, since there exists exactly one choice for some node  $b$  such that  $b = c$  (namely  $c$ ). In contrast, diamonds can appear more often, since they are characterized by  $b \neq c$ . They are upper bounded by the number of neighbors of  $a$  minus one, since at most any neighbor of  $a$  except for  $c$  can complement the base configuration to a diamond. Therefore, we expect the fraction of diamonds to be higher than the fraction of triangles, with respect to base configurations, if nodes have on average at least two neighbors.

We study unobstructed and Manhattan scenarios (corresponding to urban areas) where we model the effect of buildings such that they obstruct a signal (i.e., their attenuation is infinite). If a building lies on the LOS between two nodes  $a$  and  $b$ , it always holds that  $\gamma_{a,b} = 0 < \text{th}_S$ , i.e., all fading paths are absorbed by the building (*ideal absorber*). The simulator can easily check this by determining whether the line segment representing the shortest path intersects with any line segment corresponding to the building's walls. In our simulations, we use a Manhattan grid as a layout for  $78 \text{ m} \times 78 \text{ m}$  square buildings and  $20 \text{ m}$  street width, on a playground of  $1000 \text{ m} \times 1000 \text{ m}$  size.

Figure B.1 shows the fraction of base configurations that can be extended to a CTR, WFD, or SFD. We study the effect of average number of neighbors on the occurrence

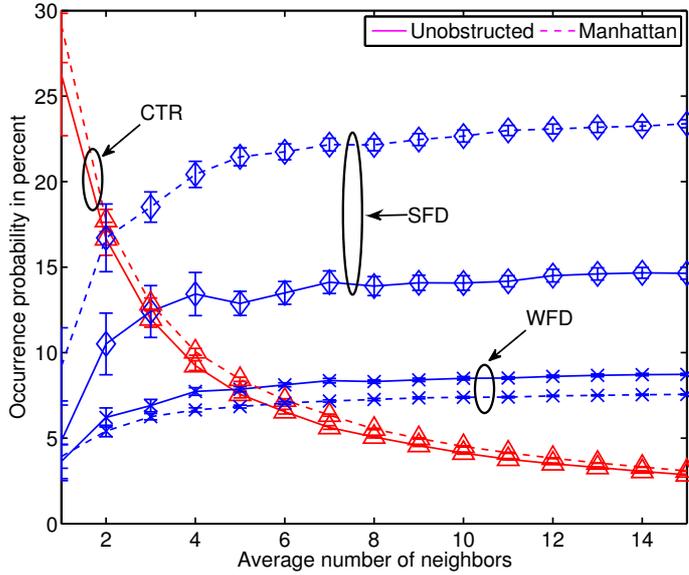


Figure B.1.: Occurrence probability of CTR, WFD, and SFD vs. average number of neighbors at low transmission power (-21 dBm)

of the configurations at low transmission power (and hence low SNR) where cooperation diversity benefits most. We observe and conclude the following:

1. CTR dominates if nodes have on average up to two neighbors. In this case, diamonds rarely occur since (on average) there is no additional node that can become a relay and complement the two-hop path to a WFD or SFD. The occurrence of the CTR degrades with increasing number of neighbors since for a single two-hop path the number of neighbors equals the number of potential diamonds, but only one triangle can be formed for that two-hop path.
2. The SFD dominates the WFD since, being able to combine up to three transmissions, it offers the highest probability to meet the given thresholds. Its occurrence increases for small number of neighbors, and with more than five neighbors only increases marginally.
3. It is more likely to find SFDs in a Manhattan scenario than in an unobstructed scenario where far less nodes are required to reach the same average number of neighbors due to blocked links. This and the fact that nodes in a Manhattan scenario can only reside in narrow streets leads to a significant increase in node density and, hence, smaller node distances.



## C. NP-hardness of Minimum Latency Cooperative Broadcast

Lichte et al. [56] first showed that the following elementary version of the cooperative broadcast problem is already NP-complete. We assume time-slotted transmissions and that a node requires exactly one time slot to retransmit a broadcast message. A node receives a message correctly if the SNRs of all collision-free receptions add up to 1. We assume further that all nodes are in one collision domain, i.e., when one node is transmitting all other nodes will receive some energy of this transmission. Thus, in any case two transmissions which overlap in time will interfere at all nodes and no useful energy can be extracted. It follows that a broadcast schedule in this scenario needs to be strictly serialized. Since transmissions are time-slotted and since each message requires exactly one slot, asking for a minimum latency broadcast is equivalent to asking for a broadcast with a minimum number of serial retransmissions. This setting corresponds to the following decision problem:

**Definition 5.** *Minimum Slotted Cooperative Broadcast*

*Instance:* A finite set  $V$  of nodes, a start node  $s \in V$ , the SNR received by  $w$  if  $v$  transmits given by non-negative random variables  $X_{vw}$  with  $\mathbb{P}\{X_{vw} > 0\} > 0$  and  $X_{vw} \sim X_{wv}$ , a constant  $0 \leq \epsilon \leq 1$ , and an integer constant  $L > 0$ .

*Question:* Existence of a broadcast schedule  $S = (v_1, \dots, v_k)$  such that:

1. Node  $s$  will initialize the broadcast:

$$v_1 = s$$

2. Active nodes receive the message before they transmit:

$$\text{for all } 2 \leq l \leq k \text{ holds } \mathbb{P} \left\{ \sum_{i=1}^{l-1} X_{v_i v_l} < 1 \right\} < \epsilon$$

3. Passive nodes will eventually receive the message:

$$\text{for all } v \in V \setminus \{v_1, \dots, v_k\} \text{ holds } \mathbb{P} \left\{ \sum_{i=1}^k X_{v_i v} < 1 \right\} < \epsilon$$

4. The last broadcast transmission is before or at  $L$ :

$$k \leq L$$

**Theorem 1.** *Minimum Slotted Cooperative Broadcast problem is in NP if restricted such that for any node  $v \in V$  the expression  $\mathbb{P}\{\sum_{u \in V} X_{uv} < 1\}$  has polynomial bounded operational complexity<sup>1</sup> of  $O(p(m))$ , where  $m$  is the input length and  $p(\cdot)$  is a polynomial which is independent of the problem instance.*

*Proof.* Consider a problem instance  $(\varepsilon, V, s, L, \{c(X_{vw}) : v, w \in V\})$  where  $c(X_{vw})$  is a representation of the random variable  $X_{vw}$ . Let  $m$  be the number of symbols required to represent the problem instance. The proof needs to show the existence of a polynomial  $q(\cdot)$  which is independent of the problem instance and which satisfies that if one guesses a solution  $(v_1, \dots, v_k)$  one can verify it in polynomial time  $O(q(m))$ .

The precision of  $\varepsilon$  is limited by  $m$  (the number of symbols used to encode the whole problem instance). Thus, for checking  $\mathbb{P}\{\sum_{u \in U} X_{uv} < 1\} < \varepsilon$  it is sufficient to evaluate the expression to an output precision of  $m$ . Given an output precision  $m$ , by assumption  $\mathbb{P}\{\sum_{u \in U} X_{uv} < 1\}$  can be computed with operational complexity  $O(p(m))$  with  $p(\cdot)$  being a polynomial independent of the problem instance.

Using schoolbook methods, additions/subtractions require  $O(m)$ , and multiplications/divisions require  $O(m^2)$  single-digit operations. Using Newton's method, a  $k$  root extraction bounded to precision  $m$  can be performed with  $O(m^2)$  single-digit operations [10].<sup>2</sup> Thus, the time complexity to compute  $\mathbb{P}\{\sum_{u \in U} X_{uv} < 1\} < \varepsilon$  is upper bounded by  $O(m^2 p(m))$ .

To check condition 2 and 3, the expression  $\mathbb{P}\{\sum_{u \in U} X_{uv} < 1\} < \varepsilon$  has to be evaluated at most  $|V|$  times for each condition. Thus, since  $|V|$  is upper bounded by the input length  $m$ , conditions 2 and 3 can be checked in time  $O(m^3 p(m))$ . Moreover, the time for checking condition 1 and 4 is trivially bounded by  $O(m)$ . In summary, a solution can be verified in polynomial time  $O(m^3 p(m))$ .  $\square$

**Theorem 2.** *Minimum Slotted Cooperative Broadcast is NP-hard and it remains NP-hard if restricted on random variable classes  $\mathcal{R}$  satisfying:*

- For all  $\varepsilon > 0$  there exists a non-negative random variable  $X$  belonging to class  $\mathcal{R}$  such that  $\mathbb{P}\{X < 1\} < \varepsilon$ .
- For all  $\varepsilon > 0$  there exists a non-negative random variable  $X$  belonging to class  $\mathcal{R}$  such that  $\mathbb{E}[X] < \varepsilon$ .

---

<sup>1</sup>Operational complexity counts the number of additions, subtractions, multiplications, divisions, and  $k$  root extractions performed to a precision bounded by the precision  $m$  of the output. Note that this implicitly means that the function can be computed with these arithmetic functions.

<sup>2</sup>Faster algorithms exist but we just need a coarse upper bound.

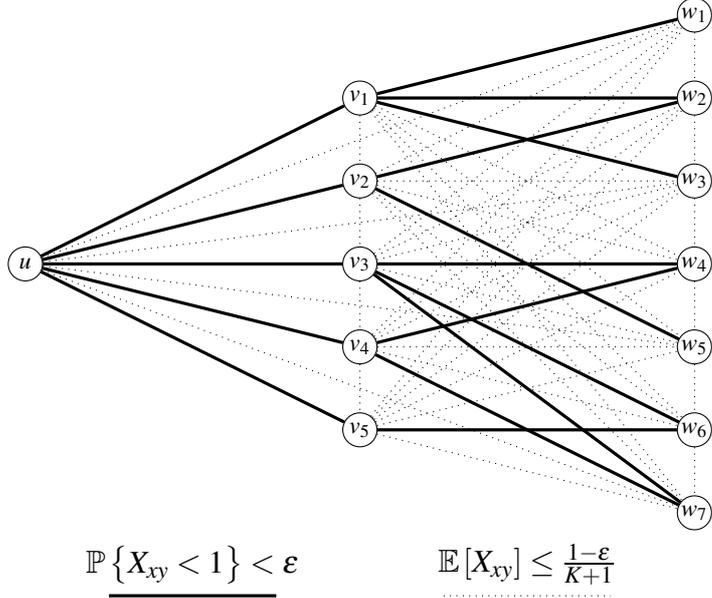


Figure C.1.: Polynomial time reduction example for  $C_1 = \{s_1, s_2, s_3\}$ ,  $C_2 = \{s_2, s_5\}$ ,  $C_3 = \{s_4, s_6, s_7\}$ ,  $C_4 = \{s_4, s_7\}$ , and  $C_5 = \{s_6\}$

*Proof.* It is sufficient to show NP-hardness when we restrict the problem instances to any random variable class  $\mathcal{R}$  satisfying the listed conditions. NP-hardness of the unrestricted problem then follows immediately by restriction.

We now construct a polynomial time reduction of the NP-hard problem Set Covering [45] to the Minimum Slotted Cooperative Broadcast problem.

In Set Covering, for a collection  $\mathcal{C} = \{C_1, \dots, C_k\}$  of subsets of a finite set  $U = \{s_1, \dots, s_n\}$  and a positive integer  $K \leq |\mathcal{C}|$ , we ask for the existence of a subset  $\mathcal{C}' \subseteq \mathcal{C}$  with  $|\mathcal{C}'| \leq K$  such that every element of  $U$  belongs to at least one member of  $\mathcal{C}'$ .

We construct the following Minimum Slotted Cooperative Broadcast instance. Refer to Figure C.1 for an illustrating example. Define a network node  $u$ , a network node  $v_i$  for each subset  $C_i \in \mathcal{C}$ , and a network node  $w_i$  for each element of  $s_i \in U$ . For each pair of distinct nodes  $x, y \in \{u, v_1, \dots, v_k, w_1, \dots, w_n\}$ , select non-negative random variables  $X_{xy}$  and  $X_{yx}$  from  $\mathcal{R}$  such that  $X_{xy} \sim X_{yx}$  and such that the following holds: If  $\{x, y\} = \{u, v_i\}$  or  $\{x, y\} = \{v_i, w_j\} \wedge s_j \in C_i$  it follows that  $\mathbb{P}\{X_{xy} < 1\} = \mathbb{P}\{X_{yx} < 1\} < \epsilon$ . Otherwise, it follows that  $\mathbb{E}[X_{xy}] = \mathbb{E}[X_{yx}] \leq \frac{1-\epsilon}{K+1}$ .

The question now is: Given start node  $u$  and time  $L = K + 1$ , does a broadcast schedule exists that meets the conditions of Minimum Slotted Cooperative Broadcast?

The reduction is performed in time  $O(|\mathcal{C}| \cdot |U|)$ , and is thus a polynomial-time one. It remains to show that a solution exists for Set Covering if and only if there exists one for the Minimum Slotted Cooperative Broadcast instance.

$\Rightarrow$ : Assume there exists a solution  $\mathcal{C}' = \{C_{j_1}, \dots, C_{j_m}\}$  with  $m \leq K$  for Set Covering.

Define  $x_1 = u$  and  $x_i = v_{j_{i-1}}$  for  $2 \leq i \leq m+1$ . The schedule  $(x_1, \dots, x_{m+1})$  with a solution for the constructed Minimum Slotted Cooperative Broadcast instance: In fact, condition 1 and 4 hold since the schedule starts with node  $u$  and since it has length  $m+1$  which is less than or equal to  $K+1 = L$ . Moreover, condition 2 holds as well. For  $X_{xy} \geq 0$  and  $2 \leq l \leq m+1$  we have:

$$\mathbb{P} \left\{ \sum_{i=1}^{l-1} X_{x_i x_l} < 1 \right\} < \mathbb{P} \{X_{x_1 x_l} < 1\} = \mathbb{P} \{X_{uv_{j_{l-1}}} < 1\} < \varepsilon$$

Finally, condition 3 holds as follows: Let  $x \in \{u, v_1, \dots, v_k, w_1, \dots, w_n\} \setminus \{x_1, \dots, x_{m+1}\}$ . If  $x = v_j \in \{v_1, \dots, v_k\}$  then

$$\mathbb{P} \left\{ \sum_{i=1}^{m+1} X_{x_i x} < 1 \right\} < \mathbb{P} \{X_{uv_j} < 1\} < \varepsilon$$

If  $x = w_j \in \{w_1, \dots, w_n\}$  then there exists a  $C_{j_l} \in \mathcal{C}'$  such that  $s_j \in C_{j_l}$ . Thus:

$$\mathbb{P} \left\{ \sum_{i=1}^{m+1} X_{x_i x} < 1 \right\} = \mathbb{P} \left\{ \sum_{i=1}^{m+1} X_{v_{j_{i-1}} w_j} < 1 \right\} < \mathbb{P} \{X_{v_{j_l} w_j} < 1\} < \varepsilon$$

It follows that there exists a solution for the Minimum Slotted Cooperative Broadcast instance, too.

$\Leftarrow$ : Assume there exists a solution  $S = (x_1, \dots, x_l)$  with  $l \leq L$  for the Minimum Slotted Cooperative Broadcast instance. Let  $v_{j_1}, \dots, v_{j_m}$  be the nodes in  $\{v_1, \dots, v_k\}$  which are contained in  $S$ . Define  $\mathcal{C}' = \{C_{j_1}, \dots, C_{j_m}\}$ . This is a solution for the Set Covering instance of the problem. In fact, with  $x_1 = u$  it follows that  $m \leq l-1$ , thus  $m \leq L-1 = K$ . Moreover, for each  $s_j$  there exists a  $C_{j_l} \in \mathcal{C}'$  such that  $s_j \in C_{j_l}$ . Assume for the sake of contradiction that there exists an  $s_j$  such that  $s_j \notin C_{j_l}$  for all  $1 \leq l \leq m$ . With  $X_{uv} \geq 0$  and with the Simple Markov Inequality  $\mathbb{P} \{Z \geq z\} \leq \frac{\mathbb{E}[Z]}{z}$  it follows for each subset  $V \subseteq \{x_1, \dots, x_l\}$ :

$$\begin{aligned} \mathbb{P} \left\{ \sum_{x \in V} X_{x w_j} < 1 \right\} &\geq \mathbb{P} \left\{ \sum_{i=1}^l X_{x_i w_j} < 1 \right\} \geq 1 - \mathbb{E} \left[ \sum_{i=1}^l X_{x_i w_j} \right] \\ &= 1 - \sum_{i=1}^l \mathbb{E} [X_{x_i w_j}] \geq 1 - l \cdot \frac{1 - \varepsilon}{K+1} \geq \varepsilon \end{aligned}$$

Thus, the schedule  $S$  is not a solution for the Minimum Slotted Cooperative Broadcast Problem, a contradiction.  $\square$

The Minimum Slotted Cooperative Broadcast problem can be extended in several ways. Examples include transmissions not restricted to slots, parallel non-interfering transmissions, different reception thresholds, and asymmetric communication channels. Assuming

that time is scaled such that one message transmission requires one time unit, a technically generalized problem formulation considering these extensions can be defined as follows:

**Definition 6. Minimum Latency Cooperative Broadcast**

Instance: A finite set  $V$  of nodes, a start node  $s \in V$ , a set of allowed transmission times  $T$ , the SNR received by  $w$  if  $v$  transmits given by non-negative random variables  $X_{vw}$ , for each  $v \in V$  the minimum accumulated power  $\gamma(v)$  needed by  $v$  to receive a message correctly, a constant  $0 \leq \varepsilon \leq 1$ , and a constant  $L$ .

Question: Existence of a subset  $S \subseteq V$  and transmission start times  $\tau(v) \in T$  for each  $v \in S$ , such that:

1. Node  $s$  will initialize the broadcast:

$$s \in S \text{ and } \tau(s) = 0$$

2. Active nodes receive the message before they transmit:

$$\text{for all } v \in S \setminus \{s\} \text{ holds } \mathbb{P} \left\{ \sum_{w \in M(v)} X_{wv} < \gamma(v) \right\} < \varepsilon$$

3. Passive nodes will eventually receive the message:

$$\text{for all } v \in V \setminus S \text{ holds } \mathbb{P} \left\{ \sum_{w \in N(v)} X_{wv} < \gamma(v) \right\} < \varepsilon$$

4. The last broadcast transmission is before or at time  $L$ :

$$\max \{ \tau(v) : v \in S \} \leq L$$

while  $N(v)$  is the set of nodes from which  $v$  receives the message without interference:

$$\begin{aligned} N(v) = & \{w \in S \setminus \{v\} : \mathbb{P}\{X_{wv} > 0\} > 0 \text{ and} \\ & \text{for all } u \in S \setminus \{w\} \text{ with } \mathbb{P}\{X_{uv} > 0\} > 0 \text{ holds} \\ & [\tau(u), \tau(u) + 1) \cap [\tau(w), \tau(w) + 1) = \emptyset\} \end{aligned}$$

and  $M(v)$  is the set of nodes which transmit before  $v$  and from which  $v$  receives the message without interference:

$$M(v) = \{w \in N(v) : \tau(w) + 1 \leq \tau(v)\}$$

Minimum Slotted Cooperative Broadcast is in fact a special case of Minimum Latency Cooperative Broadcast with the following restrictions:

1. Node transmissions are restricted to slots:  $\tau(v) \in \mathbb{N}$  for all  $v \in S$
2. Parallel message transmissions always interfere:  $\mathbb{P}\{X_{vw} > 0\} > 0$  for all  $v \neq w$
3. A node's reception threshold is restricted to 1:  $\gamma(v) = 1$  for all  $v$
4. Communication channels are symmetric:  $X_{vw} \sim X_{wv}$  for all  $v \neq w$

Relaxing any of the restrictions will make the problem of finding cooperative broadcast with minimum latency more complicated and thus such extensions are NP-hard as well. Moreover, a solution to the problem will not become significantly harder to be verified, i.e., it should remain a problem in NP. We record this fact in the following two corollaries:

**Corollary 1.** *Relaxing any of the restrictions 1–4, including any combinations thereof, of Minimum Slotted Cooperative Broadcast keeps it a problem in NP if restricted as defined in Theorem 1 and if in case of relaxing restriction 2 the expression  $\mathbb{P}\{X_{vw} > 0\} > 0$  can be verified in polynomial time.*

*Proof.* We have to show that a solution for the most generalized form, the Minimum Latency Cooperative Broadcast problem, can be verified in polynomial time.

Consider a problem instance  $(\varepsilon, V, s, L, \{c(X_{vw}) : v, w \in V\})$ , where  $c(X_{vw})$  is a representation of the random variable  $X_{vw}$ . Let  $m$  be the number of symbols required to represent the problem instance. We have to show the existence of a polynomial  $q(\cdot)$  independent of the problem instance such that if we guess a solution  $(S, \tau(\cdot))$  we can verify it in polynomial time  $O(q(m))$ .

By assumption  $\mathbb{P}\{X_{vw} > 0\} > 0$  can be verified in time  $O(p_1(m))$  for some polynomial  $p_1(\cdot)$  independent of the problem instance. Thus, finding a node  $v$ 's non-interfering receptions  $N(v)$  can be done in time  $O(|V|^2 \cdot p_1(m))$  by considering all node pairs and checking if their transmissions collide at  $v$ . Moreover, the correct receptions  $M(v)$  before a node  $v$  transmits is then done in time  $O(|V|)$ . Thus, with  $|V| \leq m$  the time required for computing  $N(v)$  and  $M(v)$  for all nodes is bounded by  $O(m^2 \cdot p_1(m))$ .

As we have shown in the proof of Theorem 1, the time complexity for computing  $\mathbb{P}\{\sum_{u \in U} X_{uv} < 1\} < \varepsilon$  is upper bounded by  $O(m^2 \cdot p_2(m))$  for a polynomial  $p_2(\cdot)$  independent of the problem instance. Thus, having computed  $N(v)$  and  $M(v)$ , checking condition 2 and 3 can be done in time  $O(|V| \cdot m^2 \cdot p_2(m))$  for one node and thus in time  $O(|V|^2 \cdot m^2 \cdot p_2(m))$  for all nodes. Condition 1 and 4 can trivially be checked in time  $O(m)$  by considering each node's transmission start time. In summary, with  $|V| \leq m$  a solution can be verified in time  $O(m^2 \cdot p_1(m) + m^4 \cdot p_2(m))$ .  $\square$

**Corollary 2.** *Relaxing any of the restrictions 1–4, including any combinations thereof, of Minimum Slotted Cooperative Broadcast keeps it an NP-complete problem which stays NP-complete if restricted on the class of random variables specified in Theorem 2.*

*Proof.* We have to consider each of the restrictions 1–4 and show that independent whether other restrictions are relaxed or not, the problem is NP-hard to solve if that particular restriction is relaxed.

*Relaxing restriction 1:* The problem obviously remains NP-hard to solve. A solution  $(S, \tau(\cdot))$  can be transformed into a slotted one by starting transmission of  $v \in S$  at  $\lfloor \tau(v) \rfloor$  instead of  $\tau(v)$ . Since transmissions last one time unit, the overlapping transmissions will be the same. Moreover, the length of the slotted version has to be the same as the length for the unslotted one. Otherwise, this would contradict the optimality of the unslotted one.

*Relaxing restriction 2:* The problem remains NP-hard to solve. Otherwise, a solution  $(S, \tau(\cdot))$  for a problem instance where all  $X_{vw}$  satisfy  $\mathbb{P}\{X_{vw} > 0\} > 0$  could be computed in polynomial time. In fact, since  $\mathbb{P}\{X_{vw} > 0\} > 0$  holds for all random variables, the set  $N(v)$  simplifies to:

$$N(v) = \{w \in S \setminus \{v\} : \text{for all } u \in S \setminus \{w\} \text{ holds} \\ [\tau(u), \tau(u) + 1) \cap [\tau(w), \tau(w) + 1) = \emptyset\}$$

Thus, the expressions  $\sum_{w \in M(v)} X_{vw}$  and  $\sum_{w \in N(v)} X_{vw}$  add only transmissions which do not overlap in time. Ordering the nodes in  $S$  as  $(v_1, \dots, v_k)$  with  $\tau(v_i) < \tau(v_{i+1})$  gives thus a serial broadcast transmission schedule.

*Relaxing restriction 3 and 4:* Obviously restricting all  $\gamma(v) = 1$  or restricting all  $X_{vw}$  to  $X_{vw} \sim X_{wv}$  will result in a problem instance whose solution under a polynomial time algorithm solving the generalized problem will be an optimal solution for the restricted problem instance as well.  $\square$

## C.1. NP-hardness for the exponential distribution

In this section we show that the problem remains NP-hard even if restricted to the exponential distribution.

**Corollary 3.** *The Minimum Slotted Cooperative Broadcast problem and problems where any of the restrictions 1–4, including any combinations thereof, are relaxed is in NP if:*

- *restricted on the class of stochastically independent exponentially distributed random variables and*
- *for all nodes  $v$  in the node set  $V$  in the set of random variables  $\{X_{vw} : w \in V\}$  there are  $K_v$  random variables with the same rate and the remaining  $N_v = |V| - K_v$  random variables with distinct rates.*

*Proof.* Consider any node  $v \in V$ . Consider any ordering  $(v_1, \dots, v_n)$  of  $V \setminus \{v\}$ . Define  $X_i = X_{v_i v}$ . With Corollary 1 we have to show that  $\mathbb{P}\{X_i > 0\} > 0$  can be verified in polynomial time, which is however obvious by just checking if the input representation of  $X_{vw}$  encodes a rate of infinity for that random variable, and that  $\mathbb{P}\{\sum_{i=1}^n X_i < 1\}$  has a polynomially

bounded operational complexity. Let  $K = K_v$  and  $N = N_v$ . Let  $\lambda_e$  be the rate of the  $K$  random variables with the same rate and let  $\lambda_1, \dots, \lambda_n$  be the distinct rates of the remaining  $N$  random variables. With the PDF given in [47] the CDF at 1 can easily be determined as:

$$\begin{aligned} \mathbb{P} \left\{ \sum_{i=1}^n X_i \leq 1 \right\} &= \int_0^1 \sum_{n=1}^N E_n \lambda_n e^{-\lambda_n x} + \sum_{k=1}^K A_k \frac{x^{k-1} \lambda_e^k e^{-\lambda_e x}}{\Gamma(k)} dx \\ &= \sum_{n=1}^N E_n (1 - e^{-\lambda_n}) + \sum_{k=1}^K A_k \left( 1 - \frac{\Gamma(k, \lambda_e)}{\Gamma(k)} \right) \end{aligned}$$

where the values  $E_n$  and  $A_k$  are expressions whose definition can be found in [47].

As it can easily be verified in [47] the values  $E_n$  and  $A_k$  are computed with  $O(n)$  and  $O(n^2)$  subtractions, multiplications, and divisions, respectively. With  $n \leq m$  follows an operational complexity of  $O(m^2)$ .

According to [10] exponentiation has an operational complexity of  $O(\log m)$  and the series approximation of the incomplete Gamma function has an operational complexity of  $O(\sqrt{m} \cdot (\log m)^2)$ . With  $N, K \leq n$  it follows that the total computation time is upper bounded by  $O(m^3)$ .  $\square$

**Remark 1.** *The corollary also covers the special cases where all random variables in  $\{X_{vw} : w \in V\}$  have the same rate or where all have distinct ones. What is not covered is the general case that there may be arbitrary many subsets of random variables having the same rate. For this case the closed form expression of the CDF is known from [3] (see (3) in that publication). The problem is that the operational complexity of this formula is not necessarily polynomially bounded. Consider an  $n$  such that  $\sqrt{n} \in \mathbb{N}$  and consider rates  $\lambda_1, \dots, \lambda_n$  with  $\lambda_1 = \dots = \lambda_{\sqrt{n}+1}$  and the remaining rates all being different. In that particular case,  $l$  in the sum expression in (3) of [3] approaches  $\sqrt{n} + 1$ . For any  $k$  the size of the set  $\Omega_{kl}$  defined in [3] is:<sup>3</sup>*

$$\begin{aligned} |\Omega_{kl}| &= \frac{(\sqrt{n} + \sqrt{n} - 1)!}{(\sqrt{n} - 1)! (\sqrt{n})!} \geq \frac{(2\sqrt{n} - 1)!}{2(\sqrt{n})!} = \frac{1}{2\sqrt{n}} \cdot \frac{(2\sqrt{n})!}{2(\sqrt{n})!} \\ &= \frac{1}{4\sqrt{n}} \cdot \prod_{i=1}^{\sqrt{n}} (\sqrt{n} + i) \geq \frac{1}{4\sqrt{n}} \cdot (\sqrt{n})^{\sqrt{n}} = \frac{1}{4} \cdot (\sqrt{n})^{(\sqrt{n}-1)} \end{aligned}$$

Thus, the size of  $\Omega_{kl}$  and with that the number of multiplications in the expression  $\Psi_{kl}$  defined in [3] is lower bounded in the order of  $\Omega(\sqrt{n}^{(\sqrt{n}-1)})$  which is not a polynomial bound. It follows, with the formulas given by [3] some problem instances cannot be verified in polynomial time. Whether there exists a better computation method with polynomial

<sup>3</sup>In [3] the set is defined as  $\Omega_2(0)$ . It stores all vectors  $(i_1, \dots, i_k)$  with integer entries  $0 \leq i_j \leq k$  that satisfy  $\sum_{j=1}^k i_j = l$ . Since  $\Omega_2(0)$  depends on  $k$  and  $l$  we denote this set  $\Omega_{kl}$  for clarity. The size of this set is  $\binom{k}{l} = \frac{(k+l-1)!}{(k-1)! l!}$ .

*bounded operational complexity is not known to us.*

**Corollary 4.** *The Minimum Slotted Cooperative Broadcast problem and problems where any of the restrictions 1–4, including any combinations thereof, are relaxed remains NP-hard if restricted on the class of exponentially distributed random variables.*

*Proof.* We have to show that for given  $\varepsilon > 0$  there exist exponentially distributed random variables which satisfy the conditions specified in Theorem 2. Consider exponentially distributed random variables  $X$  and  $Y$  with rate  $\lambda_X < \ln(\varepsilon - 1)$  and  $\lambda_Y > \frac{1}{\varepsilon}$ , respectively. It follows:  $\mathbb{P}\{X < 1\} = 1 - e^{-\lambda_X} < 1 - e^{-\ln(\varepsilon-1)} = \varepsilon$  and  $\mathbb{E}[Y] = \frac{1}{\lambda_Y} < \varepsilon$ .  $\square$



# Bibliography

- [1] H. Adam, C. Bettstetter, and S. M. Senouci. Multi-hop-aware cooperative relaying. In *Proc. 69th IEEE Veh. Technology Conf. (VTC-Spring)*, pages 1–5, April 2009.
- [2] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers – Principles, Techniques, & Tools*. Pearson Education, Inc., 2nd edition, 2007.
- [3] S. V. Amari and R. B. Misra. Closed-form expressions for distribution of sum of exponential random variables. *IEEE Trans. Reliab.*, 46(4):519–522, 1997.
- [4] *AR5414 Technical Specifications*. Atheros, 6th edition, 2007.
- [5] A. S. Avestimehr and D. N. C. Tse. Outage capacity of the fading relay channel in the low-SNR regime. *IEEE Trans. Inf. Theory*, 53:1401–1415, 2007.
- [6] E. Beres and R. Adve. Cooperation and routing in multi-hop networks. In *Proc. IEEE Int. Conf. Commun. (ICC)*, 2007.
- [7] A. Bin Sediq and H. Yanikomeroglu. Diversity combining of signals with different modulation levels in cooperative relay networks. In *Proc. IEEE 68th Veh. Technology Conf. (VTC-Fall)*, pages 1–5, September 2008.
- [8] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. In *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Comput. Commun.*, pages 133–144, 2005. ISBN 1-59593-009-4.
- [9] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman. A simple cooperative diversity method based on network path selection. *IEEE J. Sel. Areas Commun.*, 24(3):659–672, 2006.
- [10] J. M. Borwein and P. B. Borwein. *Pi and the AGM: A Study in the Analytic Number Theory and Computational Complexity*. Wiley-Interscience, 1987.
- [11] J. Boyer, D. D. Falconer, and H. Yanikomeroglu. Diversity order bounds for wireless relay networks. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, pages 1800–1804, March 2007.
- [12] D. G. Brennan. Linear diversity combining techniques. *Proc. IEEE*, 91(2):331–356, February 2003.

- [13] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. J. Hwang. Symbolic model checking:  $10^{20}$  states and beyond. In *Proc. 5th Annual IEEE Symp. Logic Comput. Sci.*, pages 428–439, June 1990.
- [14] C. T. Chou, A. Misra, and J. Qadir. Low-latency broadcast in multirate wireless mesh networks. Technical Report UNSW-CSE-TR0514, University of New South Wales, Australia, 2005.
- [15] C. T. Chou, A. Misra, and J. Qadir. Low-latency broadcast in multirate wireless mesh networks. *IEEE J. Sel. Areas Commun.*, 24(11):2081–2091, November 2006.
- [16] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.
- [17] J. Corbet, A. Rubini, and G. Kroah-Hartman. *Linux Device Drivers*. O’Reilly, 3rd edition, 2005.
- [18] T. M. Cover and A. A. El Gamal. Capacity theorems for the relay channel. *IEEE Trans. Inf. Theory*, IT-25(5), 1979.
- [19] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proc. IEEE Int. Conf. Commun. (ICC)*, volume 1, pages 376–380, June 1997.
- [20] O. Dousse, M. Franceschetti, and P. Thiran. On the throughput scaling of wireless relay networks. *IEEE Trans. Inf. Theory*, 52(6):2756–2761, 2006.
- [21] S. Favaro and S. Walker. On the distribution of sums of independent exponential random variables via Wilks’ integral representation. *Acta Applicandae Mathematicae*, 109:1035–1042, 2010.
- [22] M. P. Fewell. Area of common overlap of three circles. Technical Report DSTO-TN-0722, Defence Science and Technology Organisation, 2006.
- [23] M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Trans. Inf. Theory*, 53(3): 1009–1018, March 2007.
- [24] D. Gesbert, M. Shafi, Da shan Shiu, P.J. Smith, and A. Naguib. From theory to practice: An overview of MIMO space-time coded wireless systems. *IEEE J. Sel. Areas Commun.*, 21(3):281–302, April 2003.
- [25] M. G. Gouda. Protocol verification made simple: A tutorial. *Comput. Netw. ISDN Syst.*, 25(9):969–980, 1993.
- [26] M. G. Gouda. *Elements of Network Protocol Design*. John Wiley & Sons, Inc., 1998.

- 
- [27] R. W. Gray, S. P. Levi, V. P. Heuring, A. M. Sloane, and W. M. Waite. Eli: A complete, flexible compiler construction system. *Commun. ACM*, 35(2):121–130, 1992.
- [28] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, 1998.
- [29] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. Inf. Theory*, 46(2):388–404, March 2000.
- [30] M. Haenggi. On distances in uniformly random networks. *IEEE Trans. Inf. Theory*, 51(10):3584–3586, October 2005.
- [31] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE J. Sel. Areas Commun.*, September 2009. Invited Paper.
- [32] A. Hasan and J. G. Andrews. The guard zone in wireless ad hoc networks. *IEEE Trans. Wireless Commun.*, 6(3):897–906, March 2007.
- [33] S. Haykin and M. Moher. *Modern Wireless Communications*. Prentice Hall, 2004.
- [34] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proc. 7th Annual ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, pages 236–251, New York, NY, USA, 2001. ACM. ISBN 1-58113-422-3.
- [35] G. J. Holzmann. Protocol design: Redefining the state of the art. *IEEE Softw.*, 9(1): 17–22, January 1992.
- [36] Y.-W. Hong and A. Scaglione. Energy-efficient broadcasting with cooperative transmissions in wireless sensor networks. *IEEE Trans. Wireless Commun.*, 5(10):2844–2855, Oct. 2006.
- [37] C. Hunter, P. Murphy, and A. Sabharwal. Real-time testbed implementation of a distributed cooperative MAC and PHY. In *Proc. 44th Annual Conf. Information Sciences and Systems (CISS)*, pages 1–6, 2010.
- [38] T. E. Hunter, S. Sanayei, and A. Nosratinia. Outage analysis of coded cooperation. *IEEE Trans. Inf. Theory*, 52(2):375–391, February 2006.
- [39] IEEE. *IEEE Std 802.11a-1999 – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – High-speed Physical Layer in the 5 GHz Band*. IEEE, 1999.

- [40] F. Ingelrest and D. Simplot-Ryl. Localized broadcast incremental power protocol for wireless ad hoc networks. pages 28–33, June 27–30, 2005.
- [41] M. R. Islam and W. Hamouda. An efficient MAC protocol for cooperative diversity in mobile ad hoc networks. *Wirel. Commun. Mob. Comput.*, 8(6):771–782, 2008.
- [42] P. Jacquet. Shannon capacity in poisson wireless network model. *Problems of Information Transmission*, 45:193–203, 2009.
- [43] V. Jungnickel, L. Thiele, M. Schellmann, T. Wirth, A. Forck, W. Zirwas, T. Haustein, and E. Schulz. Implementation concepts for distributed cooperative transmission. In *Proc. 42nd Asilomar Conf. Signals, Systems, and Computers*, pages 1035–1039, 2008.
- [44] V. Jungnickel, L. Thiele, T. Wirth, T. Haustein, S. Schiffermüller, A. Forck, S. Wahls, S. Jaeckel, S. Schubert, C. Juchems, F. Luhn, R. Zavrtak, H. Droste, G. Kadel, W. Kreher, J. Mueller, W. Stoermer, and G. Wannemacher. Coordinated multipoint trials in the downlink. In *Proc. 5th IEEE Broadband Wireless Access Workshop (BWAWS), co-located with IEEE GLOBECOM*, November 2009.
- [45] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
- [46] U. Kastens. *LIDO – Computations in Trees*. University of Paderborn, 1997. available online <http://eli-project.sourceforge.net/>.
- [47] H. V. Khuong and H.-Y. Kong. General expression for PDF of a sum of independent exponential random variables. *IEEE Commun. Lett.*, 10(3):159–161, March 2006.
- [48] J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.
- [49] T. Korakis, Z. Tao, S. R. Singh, P. Liu, and S. S. Panwar. Implementation of a cooperative MAC protocol: performance and challenges in a real environment. *EURASIP Journal on Wireless Communications and Networking*, 2009:1–19, 2009.
- [50] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Trans. Inf. Theory*, 50(12):3062–3080, 2004.
- [51] H. S. Lichte and S. Valentin. Implementing MAC protocols for cooperative relaying: A compiler-assisted approach. In *Proc. 1st Int. Conf. Simulation Tools and Techniques for Commun., Networks and Syst.*, March 2008. best paper award.

- [52] H. S. Lichte, S. Valentin, H. Karl, I. Aad, L. Loyola, and J. Widmer. Design and evaluation of a routing-informed cooperative MAC protocol for ad hoc networks. In *Proc. 27th IEEE Conf. Comput. Commun. (INFOCOM)*, pages 1858–1866, April 2008.
- [53] H. S. Lichte, S. Valentin, and H. Karl. Automated development of cooperative MAC protocols: A compiler-assisted approach. *Mobile Networks and Applicat.*, 15(6): 769–785, 2009.
- [54] H. S. Lichte, S. Valentin, H. Karl, I. Aad, and J. Widmer. Analyzing space/capacity tradeoffs of cooperative wireless networks using a probabilistic model of interference. In *Proc. 12th Int. Symp. Modeling, Anal. and Simulation of Wireless and Mobile Syst. (MSWiM)*, October 2009.
- [55] H. S. Lichte, S. Valentin, H. von Malm, H. Karl, A. Bin Sediq, and I. Aad. Rate-per-link adaptation in cooperative wireless networks with multi-rate combining. In *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2009.
- [56] H. S. Lichte, H. Frey, and H. Karl. Fading-resistant low-latency broadcasts in wireless multihop networks: The probabilistic cooperation diversity approach. In *Proc. 11th ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, September 2010.
- [57] H. S. Lichte, S. Valentin, and H. Karl. Closed-form expected interference in wireless networks using a geometric path loss model. *IEEE Commun. Lett.*, 14(2):1–3, February 2010.
- [58] Z. Lin, E. Erkip, and M. Ghosh. Rate adaptation for cooperative systems. In *Proc. IEEE Global Telecommun. Conf.*, pages 1–5, December 2006.
- [59] B. Liu, C. Chou, A. Misra, and S. Jha. Rate-diversity and resource-aware broadcast and multicast in multi-rate wireless mesh networks. *Mobile Networks and Applications*, 13:38–53, 2008.
- [60] N. Marchenko, C. Bettstetter, and E. Yanmaz. On radio resource allocation in proactive cooperative relaying. In *Proc. IEEE Int. Conf. Communications Workshops*, pages 1–5, June 2009.
- [61] N. Marchenko, E. Yanmaz, H. Adam, and C. Bettstetter. Selecting a spatially efficient cooperative relay. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–7, December 2009.
- [62] I. Maric and R. D. Yates. Cooperative multicast for maximum network lifetime. *IEEE J. Sel. Areas Commun.*, 23(1):127–135, January 2005.

- [63] R. Mathar and J. Mattfeldt. On the distribution of cumulated interference power in rayleigh fading channels. *Wirel. Netw.*, 1(1):31–36, 1995.
- [64] D. Munoz-Rodriguez, O. Uribe-Arambula, C. Vargas, and H. Maturino. Interference bounds in power controlled systems. *IEEE Commun. Lett.*, 4(12):398–401, December 2000.
- [65] T. Nechiporenko, K. T. Phan, C. Tellambura, and H.H. Nguyen. Performance analysis of adaptive M-QAM for Rayleigh fading cooperative systems. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 3393–3399, May 2008.
- [66] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *IEEE Commun. Surveys Tuts.*, 3(3): 2–12, 2000.
- [67] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. 5th Annual ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, pages 151–162, 1999. ISBN 1-58113-142-9.
- [68] A. Nosratinia, T. E. Hunter, and A. Hedayat. Cooperative communication in wireless networks. *IEEE Commun. Mag.*, 42(10):74–80, October 2004.
- [69] B. O’Hara and A. Petrick. *IEEE 802.11 Handbook: A Designers Companion*. IEEE Press, 1999.
- [70] A. Özgür, O. Leveque, and D. N. C. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Trans. Inf. Theory*, 53(10):3549–3572, October 2007.
- [71] C. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications*, pages 3–12, 1999.
- [72] P. C. Pinto and M. Z. Win. Communication in a poisson field of interferers—Part I: Interference distribution and error probability. *IEEE Trans. Wireless Commun.*, 9(7): 2176–2186, 2010.
- [73] P. C. Pinto and M. Z. Win. Communication in a poisson field of interferers—Part II: Channel capacity and interference spectrum. *IEEE Trans. Wireless Commun.*, 9(7): 2187–2195, 2010.
- [74] J. G. Proakis. *Digital Communications*. 4th edition, 2000.
- [75] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno. Coordinated multipoint transmission/reception techniques for LTE-Advanced. *IEEE Wireless Commun. Mag.*, 17(3):26–34, June 2010.

- [76] A. Sendonaris, E. Erkip, and B. Aazhang. Increasing uplink capacity via user cooperation diversity. In *Proc. IEEE Int. Symp. Information Theory*, page 156, August 1998.
- [77] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, July, October 1948.
- [78] M. K. Simon and M.-S. Alouini. *Digital Communication over Fading Channels*. John Wiley & Sons, Inc., 2nd edition, 2005.
- [79] P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. pages 1763–1772, 2001.
- [80] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: A core-extraction distributed ad hoc routing algorithm. *IEEE J. Sel. Areas Commun.*, 17(8):1454–1465, August 1999.
- [81] B. Sklar. Rayleigh fading channels in mobile digital communication systems—Part I: Characterization. *IEEE Commun. Mag.*, 7:90–100, 1997.
- [82] W. Stallings. *Wireless Communications and Networks*. Prentice Hall, 2nd edition, 2004.
- [83] I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez, and E. Villanueva-Pena. Physical layer impact on the design and performance of routing and broadcasting protocols in ad hoc and sensor networks. *Comput. Commun.*, 28(10):1138–1151, 2005.
- [84] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.
- [85] S. Valentin. *Cooperative Relaying and its Application – From Analysis to Prototypes*. PhD thesis, Faculty for Electrical Engineering, Computer Science and Mathematics, University of Paderborn, October 2009.
- [86] S. Valentin, H. S. Lichte, H. Karl, I. Aad, L. Loyola, and J. Widmer. Opportunistic relaying vs. selective cooperation: analyzing the occurrence-conditioned outage capacity. In *Proc. 11th Int. Symp. Modeling, Anal. and Simulation of Wireless and Mobile Syst. (MSWiM)*, pages 193–202, October 2008. ISBN 978-1-60558-235-1.
- [87] S. Valentin, H. S. Lichte, D. Warneke, T. Biermann, R. Funke, and H. Karl. Mobile cooperative WLANs – MAC and transceiver design, prototyping, and field measurements. In *Proc. 68th IEEE Veh. Technology Conf. (VTC-Fall)*, pages 1–5, September 2008.

- [88] S. Valentin, D. H. Woldegebreal, T. Volkhausen, and H. Karl. Combining for cooperative WLANs - a reality check based on prototype measurements. In *Proc. IEEE Workshop on Cooperative Mobile Networks (CoCoNET2)*, pages 1–5, June 2009.
- [89] E. C. van der Meulen. Three-terminal communication channels. *Adv. Appl. Prob.*, 3: 120–154, 1971.
- [90] J. Venkataraman, M. Haenggi, and O. Collins. Shot noise models for outage and throughput analyses in wireless ad hoc networks. In *Proc. IEEE Military Communications Conf. MILCOM 2006*, pages 1–7, 2006.
- [91] T. Wang, X. Du, W. Cheng, Z. Yang, and W. Liu. A fast broadcast tree construction in multi-rate wireless mesh networks. In *Proc. IEEE Int. Conf. Commun.*, pages 1722–1727, June 2007.
- [92] T. Wang, B. Li, Z. Yang, and W. Cheng. A minimized latency broadcast in multi-rate wireless mesh networks: Distributed formulation and rate first algorithm. In *Proc. IEEE Int. Conf. on Multimedia Computing and Syst.*, 2007.
- [93] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. 19th IEEE Conf. Comput. Commun. (INFOCOM)*, volume 2, pages 585–594, March 26–30, 2000.
- [94] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. 4th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999.
- [95] J. Wu, M. Cardei, F. Dai, and S. Yang. Extended dominating set and its applications in ad hoc networks using cooperative communication. *IEEE Trans. Parallel Distrib. Syst.*, 17(8):851–864, August 2006.
- [96] J. Yackoski, L. Zhang, C.-C. Shen, L. Cimini, and B. Gui. Networking with cooperative communications: Holistic design and realistic evaluation. *IEEE Commun. Mag.*, 47(8):113–119, August 2009.
- [97] Z. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical SNR-guided rate adaptation. In *Proc. 27th IEEE Conf. Comput. Commun. (INFOCOM)*, pages 2083–2091, April 2008.
- [98] L. Zheng and D. N. C. Tse. Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels. *IEEE Trans. Inf. Theory*, 49(5):1073–1096, 2003.