



TAMPERE UNIVERSITY OF TECHNOLOGY

ALEXANDER PYATTAEV

SYSTEM LEVEL PERFORMANCE EVALUATION OF CLIENT  
COOPERATION IN WIRELESS CELLULAR NETWORKS

Master's thesis

Examiners: Professor Yevgeni Koucheryavy  
Researcher Sergey Andreev  
Examiner and topic approved by the Faculty  
Council of the Faculty of Computing and Elec-  
trical Engineering on 9 November 2011.

## Preface

This thesis concludes a long-going research on client relay by Network and Protocols Group led in TUT under supervision of Professors Prof J. Harju and Yevgeni Koucheryavy, which was started in 2009. During the last two years, a lot of experience was gained, and a lot of usable results have been obtained. This thesis is, probably, the most complete text on the research done, and covers most of the technical aspects that have been omitted in previous research publications before due to their space limitations.

I wish to thank my colleagues from TUT for the contributions to our collaborative work on the client relay. Without Sergey Andreev's inspiration and deep insight into IEEE 802.16 and LTE protocols I would have never started such a project. Dmitri Moltchanov has given countless advices on the statistical processing and traffic modeling. Recent contributions by Olga Galinina also allowed our group to understand the deep relationships behind client relay, and also inspired me to push on with the analytical approach in my next research project on generalized relaying models. I also wish to thank my friend Mikhail Gerasimenko for his help with annoying issues such as debugging, testing and verification.

I dedicate this thesis to randomness, for it drives this somewhat crazy world.

20.10.2011

ALEXANDER PYATTAEV

## Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

PYATTAEV, ALEXANDER: "SYSTEM LEVEL PERFORMANCE EVALUATION OF CLIENT COOPERATION IN WIRELESS CELLULAR NETWORKS"

Master of Science Thesis, 48 pages, 15 appended pages

November, 2011 Major subject: Communications Engineering Examiners: Professor Yevgeni Koucheryavy, Researcher Sergey Andreev.

Growing demand for bandwidth dictates the use of smaller wireless cells, which results in increased inter-cell interference. In most contemporary cellular systems, the clients at the cell edge typically have the worst chance of successful uplink transmission due to interference from the neighboring cells using the same frequency. Cooperative communications are believed to be a promising technique to enhance the performance of cell-edge users by allowing them to exploit other users as relay nodes and thus improve their throughput by reducing the number of retransmissions.

This thesis presents in-depth system-level evaluation of client relay technique in state-of-the-art wireless cellular networks (IEEE 802.16, LTE release 10). Several important scenarios are considered, including opportunistic client relay behavior and various network layouts. It is demonstrated that client cooperation may considerably improve system performance in terms of cell-edge user performance for the cost of some increase in energy consumption of cell-center user.

Keywords: client relay, wireless networks, cooperative, simulation, performance evaluation.

# Contents

<b>Abbreviations and symbols</b>	<b>v</b>
<b>List of tables</b>	<b>ix</b>
<b>List of figures</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Structure of the thesis . . . . .	3
<b>2 Theoretical background</b>	<b>4</b>
2.1 Overview of the research area . . . . .	4
2.2 Research goals . . . . .	6
<b>3 System model derivation</b>	<b>7</b>
3.1 Baseline system model . . . . .	7
3.1.1 Client relay algorithm . . . . .	10
3.2 Extended model with physical layer abstraction . . . . .	11
3.3 Cellular network model . . . . .	13
<b>4 Analytical modeling</b>	<b>15</b>
4.1 Analysis of the baseline system model . . . . .	15
4.1.1 Medium access delay distribution . . . . .	16
4.1.2 Mean performance parameters for arbitrary load . . . . .	17
4.2 Analysis of the extended system model . . . . .	20
4.2.1 Client-relay aware scheduling . . . . .	22
<b>5 Simulation methodology and environment</b>	<b>26</b>
5.1 Simulation platform . . . . .	26
5.2 Simulator internal structure . . . . .	27
5.3 Accuracy control . . . . .	28
5.3.1 Path loss model and fading . . . . .	29
5.3.2 Power control and interference loopback . . . . .	30
5.3.3 Signal detection and SNR mapping . . . . .	31
5.4 Protocol abstraction . . . . .	32
5.4.1 IEEE 802.16 mapped to extended system model . . . . .	32
<b>6 Performance evaluation results</b>	<b>35</b>
6.1 Baseline system model . . . . .	35
6.1.1 Performance of different schedulers . . . . .	35
6.2 Extended system model . . . . .	36
6.3 Opportunistic cooperation performance . . . . .	36

---

6.4 BS-assisted cooperation performance . . . . .	39
<b>Conclusion</b>	<b>43</b>
<b>References</b>	<b>46</b>
<b>Appendix 1: configuring simulation environment, baseline system model</b>	<b>49</b>
Setup for analytical verification . . . . .	49
The simulation scenario for analytical verification . . . . .	52
<b>Appendix 2: configuring simulation environment, IEEE 802.16</b>	<b>56</b>
SLS setup for IEEE 802.16 OFDMA PHY . . . . .	56
Scenario setup for IEEE 802.16 network . . . . .	59

---

## Abbreviations and symbols

- ACK (Acknowledgment) – a confirmation of packet or ARQ block delivery
- AF (Amplify and Forward) – a relaying scheme where relay does not perform error-correction before forwarding the data, acting as a digital repeater.
- AMC (Adaptive Modulation and Coding) – a scheme allowing the transmission system to switch modulation and coding schemes based on situation.
- ARQ (automatic repeat request) – a mechanism that allows to request retransmission of a lost packet
- BS (Base Station) – a network node that manages the medium access for the clients associated with it.
- CDF – cumulative distribution function.
- Client – a network node that is using the network to get service.
- Cooperator – a client that relays data for other clients.
- DF (Decode and Forward) – a relaying scheme where relay has to decode the FEC code and correct errors before forwarding the data to the destination.
- DL (Downlink) – direction from base station towards client terminals
- FEC (forward error correction) – error correcting coding performed before transmission
- Fragment – the minimal ARQ retransmission unit
- Frame – a MAC signaling unit that denotes repeated access cycles (this term is never used to denote layer 2 packet.)
- LLS – link level simulation
- LQI (link quality indication) – a scalar mapping link quality to some value range.
- LTE (Long-Term Evolution) – a next generation network by 3GPP
- MAC (Medium Access Control) – a protocol/system governing access to shared medium
- MIMO (Multiple Input Multiple Output) – a transmission system with multiple transmitters and receivers.
- MISO (Multiple Input Single Output), Multiray reception – a reception mode when the receiver analyzes several sources of the same signal and combines them.

- 
- MPR (Multi-Packet Reception) – a capability of the receiver to decode multiple packets at the same time.
  - MRC (Maximal ratio combining) – a way of combining multiple copies of received signal that provides best SNR.
  - MU-MIMO (Multi-user MIMO) – same as MIMO, but the transmitters, receivers or both may be located on logically separate nodes in the network.
  - NACK (Negative Acknowledgment) - a message that shows that a packet or ARQ block contained an error.
  - NC (Non-Cooperative) – a classic networking scheme with no cooperation
  - Node – a network entity capable of sending and receiving traffic.
  - OFDM (Orthogonal frequency-division multiplexing) – a frequency multiplexing method based on orthogonal subcarriers and overlapping frequency channels.
  - OFDM-MIMO – a MIMO system designed for OFDM signals. Usually implies presence of the pilot signals for channel equalization.
  - PDF – probability distribution function
  - PDU (Protocol Data Unit) – a useful payload of a protocol
  - RB (Resource Block) – a set of slots allocated for some purpose, such as user's data transmission.
  - RX – reception or receiver
  - SLS – system level simulation
  - SNR – signal to noise ratio
  - SINR – signal to interference and noise ratio
  - Slot – a unit of channel allocation, corresponding to fixed integer number of sub-channels allocated for fixed integer number of symbols. It can span the entire frequency band, in which case classic definition of TDM slot applies.
  - Subcarrier – one of the orthogonal carriers in OFDM signal.
  - Subchannel – a logical frequency channel within OFDM signal.
  - Subframe – a fraction of a Frame, reserved for uplink or downlink traffic
  - TX – transmission or transmitter

- UL (Uplink) – a direction from client terminals towards base station
- WIMAX (Worldwide Interoperability for Microwave Access) – wireless 4G network based on IEEE 802.16.



Table 1: Symbols and Notation

Variable	Description
$N_x$	The number of nodes in some node group x
$S_x$	The set of sources in node group x
$D_x$	The set of destinations for node group x
$i, j, k$	Array indices
$t$	Time or a state of random process
$n_x$	The number of entities x
$k_x$	Waiting positions in x
$\lambda$	Arrival rate
$\mu$	Service rate
$\rho$	$\frac{\lambda}{\mu}$
$a, b, c$	Temporary variables
$s_x$	State with index x
$L_x$	Delay x
$I$	Event
$E()$	Expectation or mean value
$A, B, C$	Nodes in the network
$p, Pr(), q$	Probability of an event
$\theta$	A network planning parameter that defines the split between near and far nodes in a cell

## List of Tables

1	Symbols and Notation . . . . .	viii
2	Analytical model notations . . . . .	18
3	Delivery probabilities for source groups( $q$ -function) . . . . .	35
4	Evaluation parameters - opportunistic cooperation . . . . .	38
5	Evaluation parameters - assisted cooperation . . . . .	40

---

## List of Figures

1	Cellular network planning with frequency reuse 3/1/1 . . . . .	2
2	Cooperative relaying without negotiation . . . . .	5
3	Simple "triangle" topology . . . . .	7
4	Example timeline of relay operation . . . . .	10
5	Packet delay verification . . . . .	20
6	The difference between OFDMA and OFDM resource allocations . . . . .	21
7	The mapping of sector into a line for analysis . . . . .	23
8	Simulation tool structural scheme . . . . .	27
9	Simulation tool internal flow diagram . . . . .	28
10	Network planning used for simulations . . . . .	31
11	Delay distribution for node 1 . . . . .	36
12	Scheduler performance under different conditions . . . . .	37
13	Average uplink throughput (Mbit/s), OFDM . . . . .	39
14	CDF for per-node transmission success probabilities, OFDM . . . . .	40
15	CDF for per-node transceiver utilization, OFDM . . . . .	41
16	Average uplink throughput (Mbit/s), OFDMA . . . . .	41
17	CDF for per-node transmission success probabilities, OFDMA . . . . .	42
18	CDF for per-node transceiver utilization, OFDMA . . . . .	42

# 1 Introduction

Almost any wireless cellular network eventually faces the capacity problem. In essence, bigger cells mean that the resources are spread across vast area and multiple clients, and therefore less can be offered to any given one of them. Until recently, it was possible to increase the network capacity by increasing the spectral efficiency of the physical layer. Currently, however, contemporary wireless access technologies are about to completely exhaust the physical layer capabilities. With modern SDR methods it is remarkably easy to implement an OFDM-based digital transmission system with up to 95% spectral efficiency, and leading cellular technologies, such as IEEE 802.16m [1] and LTE-Advanced [2], are approaching this margin. Therefore, in pursuit of increased capacity, operators tend to decrease the cell sizes and increase the reuse factor. Although the desired capacity increase can be achieved this way, there are some implications.

In cellular networks multiple independent cells use the same radio resources. Whereas these cells are not necessarily adjacent, it is crucial to control the inter-cell interference, which otherwise would degrade communication performance. The nature of wireless channel makes interference mitigation extremely challenging, since it calls for decreased transmission power and contradicts reliability. The users at the cell edge, following the power control procedure, are forced to increase the transmission power to keep reasonable *signal to SINR*. Conventional radio network planning tries to take this issue into account, and if the cells are large enough the problem is solved more or less satisfactorily as shown in Figure 1. White cells use the same frequency, whereas gray cells use another one. Since the path loss of link  $M_1B_2$  is on average higher than that of  $M_1B_1$  or  $M_2B_2$ , the scheme succeeds in keeping the path loss between the users of different cells above the path loss to the own *base station (BS)*.

Until recently, the reuse factor values have been usually around 1/9 [3], and this scheme worked just fine. But as we decrease the cell sizes and increase reuse factor, the path loss  $L_{M_1B_2}$  approaches  $L_{M_1B_1}$  and  $L_{M_2B_2}$ , and it is no longer the path loss that defines the reachability of a given user, but rather the interference level.

Now obviously, cell-edge users cause the most interference - they have highest transmission power, they are the closest to the neighboring cell, and they take the most time to send the same amount of data due to low modulation index. Unfortunately, there is not much that can be done about any of those things in a conventional cellular network. To some extent, this issue may be mitigated by scheduling the users across all cells to transmit at carefully selected times [4] for the cost of extra coordination between the BSs. Unfortunately, such scheduling decision requires solving NP-hard problem in real time, resulting in prohibitive system complexity.

There are, however, more realistic ways to improve the situation. One of the worst situations is when a given cell-edge user is forced to retransmit due to fading, as it will cause more and more interference, consequently decreasing its own success chances. Therefore,

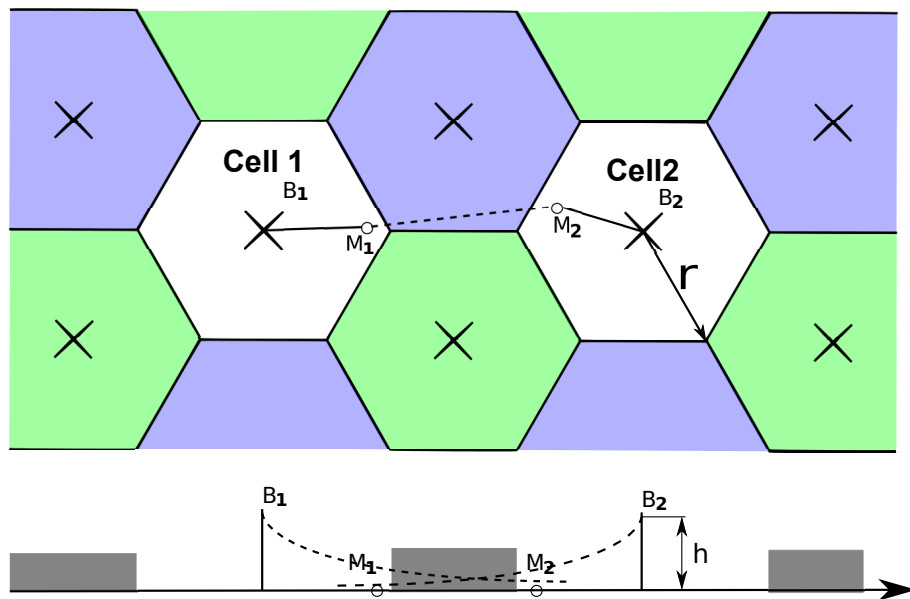


Figure 1: Cellular network planning with frequency reuse 3/1/1

it would be beneficial to somehow assist such cell-edge users in order to limit the number of attempts they have to perform to reach the BS successfully. Hopefully, this would provide substantial benefits to the overall interference level, and therefore allow for more capacity in the network in general.

The cellular network logic dictates, that all the traffic has to go from the subscriber's user equipment to the BS before it can be forwarded anywhere else [3]. But as we introduce the concepts of packets and retransmissions, the question arises if it is actually the optimal approach. Recently there has been a lot of research on mesh networks, as well as other alternatives to conventional cellular systems. Some of the proposed protocols turned out to be very useful, especially in sensor networks[5]. Nevertheless, for high-throughput networks, mesh protocols do not usually perform so well.

An attractive alternative solution to the indicated problem with cell-edge users might be achieved by reducing the transmission power or increasing the success chances for the cell-edge users, and consequently controlling the interference. Of course, in such a case measures should be taken to compensate for the loss in data rate and link reliability. For example, static relays may be deployed across the cells to assist mobile users. While being effective standard solutions [6], they are costly to deploy and require much effort to tune properly.

In contrast, the client terminals themselves may be allowed to relay packets for each other. Such schemes will be referred to as "client relay". Various client relay concepts have been studied extensively and in some cases turned out to be too complex to implement. The main problem with most relaying concepts is the way one organizes the signaling. Signaling is, in most cases, the biggest problem, as a handshake is required for each packet to be relayed. Such tiny signaling packets are prone to get lost, and therefore

compromise the viability of the whole protocol.

This research concentrates on so-called opportunistic cooperation, where no explicit signaling is involved. This makes such relay protocol highly robust and a lot easier to implement and analyze. In previous work on the topic such issues as purely opportunistic relaying [7] as well as heuristic relay selection schemes in different scenarios have been addressed [8, 9]. This thesis concludes the performed research, and acts as a complete reference of the results obtained so far.

## **1.1 Structure of the thesis**

The thesis is organized as follows. The theoretical background section 2 provides insight into the alternative client relay approaches and justifies our choices in terms of protocol design.

After that in section 3 we formally define the system model and show that a network of interest (for example IEEE 802.16 or LTE) can be easily mapped onto it. The analytical approaches and applicable models are considered in section 4.

A large section 5 is devoted to the simulation study and the tool that was designed specifically for the task. The most important results are provided in Section 6. The appendices contain extra information on the simulation setup. Appendix 1 contains the setup used for the baseline system model, and Appendix 2 contains the setup for IEEE 802.16 network.

## 2 Theoretical background

Based on the discussion in the introduction section, let us have a look at how other researchers have approached the problem of wireless relaying, including, but not limited to client relay.

### 2.1 Overview of the research area

First of all, let us quickly consider "true" multi-hop networks, such as ZigBee[10] or IEEE 802.11 [11]. Obviously, those provide the maximal flexibility compared to infrastructure-based cellular systems, but at the same time they usually can not compete in spectral efficiency and cost per data rate. The key problem with multi-hop is that every wireless transmission has to be carefully scheduled in such a way that it does not collide with others. And for networks without fixed points of authority, this task is remarkable hard to solve. Therefore, although the spectral efficiency of the shorter links is higher than that of longer cellular links, the signaling overhead easily outweighs the possible throughput gains. Also, due to multiple wireless links, mesh networks can never have such small latencies as cellular systems. Providing *Quality of service* (QOS) also becomes challenging, as one can not be sure that all intermediate links will be available.

Although the concept of wireless multi-hop transmission works well in sensor networks which do not require high throughput or QOS support and simply aim for increased spectral efficiency, it is not typically scalable enough to work in the area of cellular systems. This does not mean, however, that attempts have not been made. Some studies propose hybrid protocols that incorporate simple mesh routing inside preallocated time slots within uplink subframes in IEEE 802.16 [12, 13, 14]. For example, one may allocate a small resource block in LTE uplink for peer-to-peer communication between clients that have poor channel or are simply outside of the coverage area [12]. Careful resource allocation for relays may replace peer-to-peer negotiation, as it was shown in [13]. Some attempts have been made to build a wireless network with relaying as a built-in feature within WINNER project [15]. The client cooperation scheme for such network is discussed in [14]. Unfortunately, as of now there are no plans for actual deployment of WINNER networks, so the applicability of this research is questionable.

There are also more conservative research initiatives concentrating on traffic relaying with static relays [16, 17]. Many studies are devoted to modeling physical channels with relays as such [18, 19].

As it was noted in the introduction, one of the most challenging problems for wireless relays is signaling between the involved parties. What is generally missed out is the fact that in wireless networks we can not usually rely on small data bursts to be delivered reliably. Coding theory clearly states that the longer is the transmitted fragment (with fixed code rate), the better are chances of getting it delivered correctly. That is why any sort of single-bit feedback, as in [20] or mesh route announcement has good chances of

taking a lot more than a couple of bits to be transmitted reliably. We note, however, that such small signaling messages can be embedded into the data packets with almost no overhead. For example, we can expect the BS to broadcast *acknowledgments*(ACK's ) for each packet it has successfully decoded, and negative acknowledgments (NACK's) for each packet it was expecting to be sent but could not decode correctly.

We can note, that it is always a gamble if a relay is available or not, and if it is available, it is again a gamble if the entire handshake will work fine and no signaling packets will be lost. Therefore, most of the schemes proposed in alternative works on wireless relaying assume error-free signal channels, single-bit feedback, among other things of questionable implementability. This gives good theoretical results, but does not really give any usable solutions in the end. Therefore some other solution is needed, that does not deny the presence of the original transmission link even when relays are cooperating. A simple sketch in Figure 2 illustrates the idea behind the proposed view on client relay.

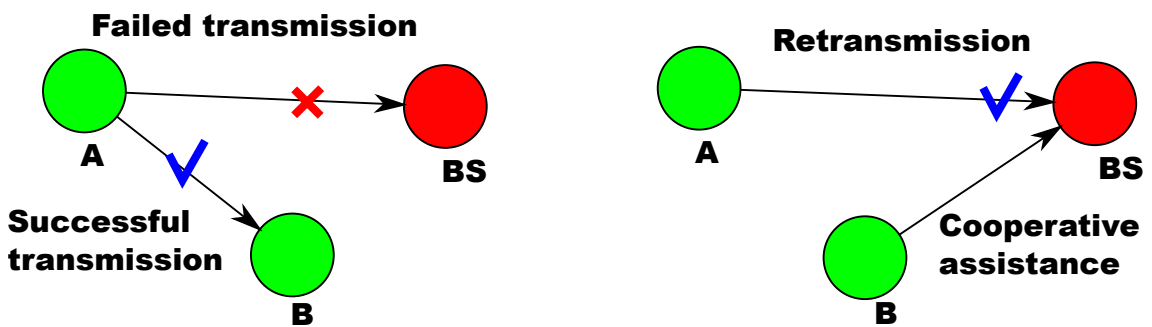


Figure 2: Cooperative relaying without negotiation

As it usually happens, one does not have to storm a complex problem head-on, there are several ways around it. One way to go is to use physical layer relays. The idea is that a repeater needs no signaling to work. In reality, such repeater has to be carefully placed in tuned to operate, which is costly and might not be feasible in the end. As part of the fixed infrastructure, a repeater may be assisted by the BS such as in IEEE 802.16j [6]. This gives good results and is a well-standardized solution. The only problem is that it still needs a lot of work to set up.

Another option is to relay the packets completely at random, even if not explicitly asked or allowed to do so, or in other words opportunistically. The idea of opportunistic relay caught our attention in works of R.Rong and A.Ephremides[21]. They have proposed a simple analytical model for such relay protocol, and have shown that even with slightest gain in delivery probability a considerable increase in system capacity can be expected. In the research mentioned above, a *multi-packet reception* MPR environment is considered. While MPR receivers are indeed possible with MIMO, we can not just assume that such mode is available in deployed devices. Instead, we can restrict the packets in MPR scheme to be identical. In this case we no longer require to have orthogonal channels, what we need is an advanced multipath compensation scheme, which happens



to exist in almost any OFDM based system - the guard interval.

Detailed look at OFDM-MIMO shows, that it is usually possible to separate several transmissions even if they are not carrying the same information. In case the transmissions are duplicates, there is a possibility for diversity gain in the receiver. This work does not concentrate on the details of OFDM-MIMO, as most of the required information is available in [22], as well as other works by the same authors. Similarly, we omit the details of HARQ, power capture and some other advanced physical layer phenomena due to complexity of modeling.

A new relaying protocol that does not involve any signaling for relay selection is evaluated within this thesis work. In opportunistic cooperation every single node chooses independently whether it should listen, forward, or skip the next transmission. This procedure is referred to as relaying policy. In this research we are only interested in uplink transmissions, since for the downlink we can usually afford more transmission power without compromising the interference levels in neighboring cells. Opportunistic relay was considered by other researchers as well, for example in [23]. The core difference of the proposed approach is the fact that the cooperator never has to request resources for relaying, since it transmits at the same time and frequency as the original source would have while retransmitting.

## 2.2 Research goals

The performed research on client relay had several goals, all of which have been reached:

- Develop a client relay protocol that would not have the problems typical to other similar solutions, such as excessive signaling overhead or extreme complexity, which would be robust and flexible.
- Prove that client relay can be implemented in a realistic network without major modifications of the network protocol or physical layer.
- Construct models to perform analytical and simulation study of client relay.
- Evaluate the performance of the client relay in terms of delay, throughput, energy efficiency and other typical performance parameters.
- Study the effects of scheduling, power control and other specific mechanisms, find client relay's limitations and drawbacks.

The goals presented here also correspond to the timeline of the performed research work. In the following sections all of them will be addressed.

### 3 System model derivation

There are two system models, baseline model preserves analytical tractability while the extended system model allows for a more detailed evaluation.

#### 3.1 Baseline system model

Let us consider a wireless client relay system which has, in simplest case, three entities. Those are 2 client nodes,  $A$  termed the originator and  $R$  termed the relay, and a single sink node  $B$  termed the BS (see Figure 3). Both of the clients have their own traffic to be delivered to the  $B$ . Additionally, the relay may eavesdrop on the packets from the originator and store them for the subsequent retransmission. The packets are stored in their original form in a special relaying buffer. The BS receives data packets from both the originator and the relay. The BS has no own traffic. This setup follows the "triangle" model used in [21]. Similar scenario was encountered in other research, and appears to be the minimal possible setup.

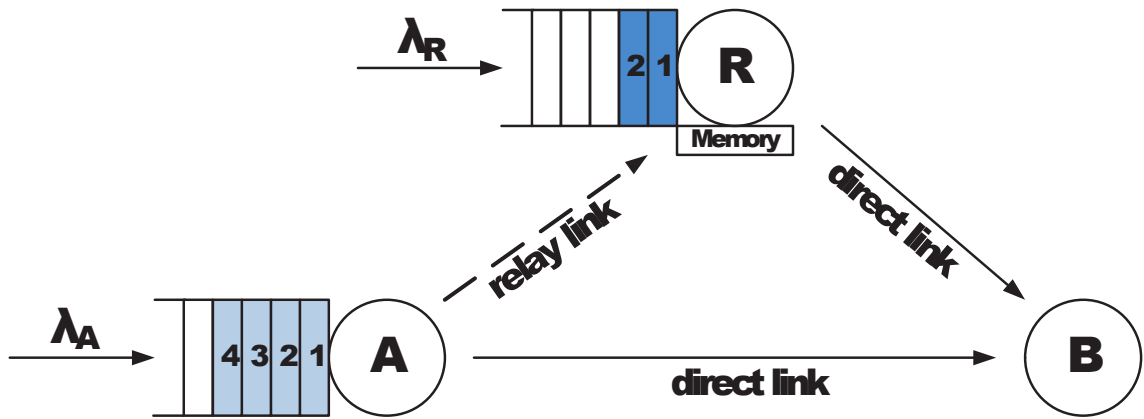


Figure 3: Simple "triangle" topology

Let us formalize some of the ideas that we have discussed in the previous section. The formalization is valid for a single isolated cell, but can be easily extended to a multi-cell scenario.

#### 1. System-level parameters

##### (a) Synchronous time

The system time is dictated by the BS, and every client is capable of synchronizing to it. The time is discrete and there is a known quantum of time.

##### (b) Link-level topology

Unlike classic cellular star topology, we consider the complete graph of links between all the nodes. We also consider virtual links that correspond to multiple sources ( $A$  and  $R$  in the example) transmitting simultaneously.

(c) Data packets

All data packets sent over the network are unicast, and are destined towards BS. Downlink traffic exists, but is of no interest and is assumed to be over a separate uncorrelated channel. In most wireless systems, actual data packets are further fragmented into smaller MAC layer PDU's, which in turn are further divided into FEC blocks, subject to the ARQ procedure. To simplify the discussion, we will concentrate on the lowest-level, that is ARQ blocks, since the performance of the above layers can be easily derived from that. Unless mentioned otherwise, by packet we understand a smallest unit of data that is still subject to ARQ procedures.

(d) Scheduling

The system has a centralized scheduler that never allows collisions.

2. Transmission medium

(a) Multiray reception

All nodes in the network are capable of multiray reception, that is if several sources are transmitting the same signal, all of them are useful. At the same time, any other signal is considered to be interference.

(b) Time division access

Only one transmission is legitimate at any given point of time, and it occupies all the available channel bandwidth. This corresponds to TDD OFDM system.

(c) Non-ideal channel

Any uplink packet is subject to random loss, the probability of such loss depends on the set of transmitting nodes and destination only, so the channel is memoryless. This probability is assumed to be known.

3. Signaling

(a) Reliable feedback

There are just 2 outcomes of the packet transmission - success or failure. The error-detection is perfect, false-positive or false-negative result is not possible. The feedback is sent back to the clients with predictable delay, and always before their next transmission opportunity.

(b) Transparent scheduling

Every client node knows exactly which node is scheduled to transmit next, and it can also determine which packet exactly will be (re)transmitted.

(c) Retransmissions

Every node will attempt retransmissions of the lost packets until a specific limit is reached, then the packet is discarded. The retransmissions will retain

the relative ordering of the original packets. Every node in the network can reliably predict if there will be a retransmission and which packet will be retransmitted.

#### 4. Clients

##### (a) Finite buffers

It is assumed that every client node has a fixed, finite-sized buffer for data packets to be sent. Without loss of generality, we assume that those do not require further fragmentation. Every client node also has a special buffer for eavesdropped packets. The size of this buffer is enough to accommodate as many packets as needed.

##### (b) Traffic patterns

Every client node has its own arrival flow, which is in most cases not relevant, but it is assumed that at least Poisson arrivals and saturation mode are possible. Newly arrived packets are served according to the FIFO discipline. Depending on the scheduler, several packets from single source may be scheduled in a row.

##### (c) Client node operation

At any given time, a client node may transmit, stay idle or listen for the transmissions from the other nodes. Due to duplexing issues, the nodes can never transmit and receive at the same time. Client nodes will always give priority to their own traffic over eavesdropping. The client nodes store the eavesdropped packets in a special buffer specific to every relaying session. It is enough to have just one storage cell for each session as only one packet can be in transmission at any given time.

##### (d) Opportunistic relay

Every potential relay decides independently whether it should eavesdrop or not. We may assume, for example, that it eavesdrops on every subsequent packet opportunistically with fixed probability  $p_{rx}$ . A node is not expected to eavesdrop if its channel conditions are worse than those reported by the BS to the packet originator. Also the actual transmission will only occur if the above channel quality condition still holds. A node may decide not to transmit for other reasons, those decisions are independent for each retransmitted entity. For example, a node may transmit in relay sessions with probability  $p_{tx}$ .

To clarify the above system model, let us now consider an example of system operation in Figure 4. Assume that there are only 2 client nodes transmitting packets. According to the system model the scheduling information is assumed to be available over a separate channel and consumes no resources. The cooperators decide opportunistically if it will eavesdrop on the transmissions from the originator in the following slot, and may

then help if the eavesdropping was successful. Cooperative transmissions are carried out by  $R$  in the same slot which was granted to  $A$  for its retransmission. In Figure 4, we observe unsuccessful transmissions by node  $A$ , while node  $R$  performs eavesdropping, and finally a successful cooperative transmission. We also notice that eavesdropping is not always successful.

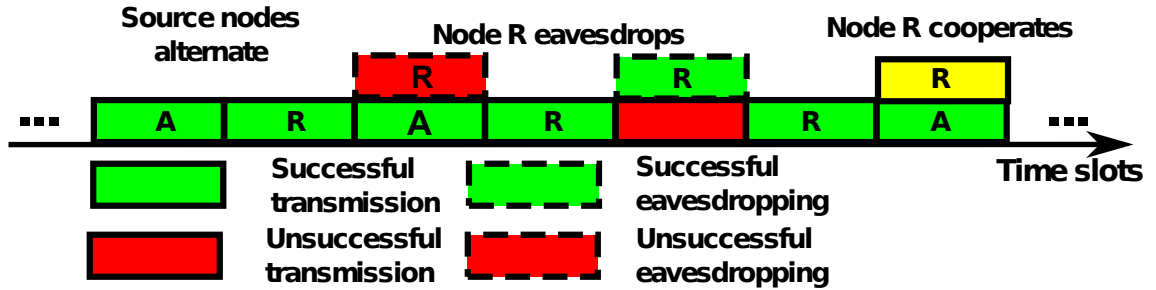


Figure 4: Example timeline of relay operation

There are also some small details that distinguish this work from many similar ones. In contrast to [21] and many other works, where relay nodes acknowledge the receipt of the packet from the originator and keep transmitting the relay packet as if it was its own, in this work the cooperation is only possible if the originator attempts retransmission of the packet. When it is scheduled to retransmit, the relay may transmit the eavesdropped packet at the same time. This, given appropriate physical layer, greatly increases the probability of successful delivery. Generally, for BS it appears as an extra beam for each relay involved, forming so called "virtual MIMO".

Also, the originator may be unaware of the cooperative help from the relay. No service information is transmitted between the originator and the relay by contrast to [21], where the relay was to acknowledge packets received from the originator. This principle allows implementation of relaying without rewriting the base MAC protocol and allocating extra channels for signaling.

### 3.1.1 Client relay algorithm

The considered client relay protocol is summarized by Algorithm 1.

In the study two physical channel models are considered, both with their own purposes. Let us first discuss the simplest one.

- Every possible combination of sources maps to a known delivery probability for a given destination, that is not affected by packet size or any other factor.
- All nodes are aware of the delivery probabilities.
- Delivery probabilities never change, loss process follows the Bernoulli scheme.

---

**Algorithm 1** Client relay operation
 

---

```

repeat
  Generate new arrivals into the client buffers.
  Schedule a node to transmit a packet.
  for all client nodes in a cell do
    if cooperation criteria met then
      if node has the some of the packets that are being transmitted then
        Node joins the transmission.
      else
        if node decides to listen according to local policy then
          Node starts eavesdropping.
        end if
      end if
    end if
  end for
  Transmit the packet and receive feedback.
  if packet received at the base station then
    Discard the packet from the originator queue and relay buffers.
  end if
  if packet received at the relay node then
    Store the packet in relay buffer.
  end if
until network shutdown

```

---

Although this is not much of a channel model, it still captures the most important behavior - random packet losses and possible gains when multiple nodes are transmitting. The key advantage of such model is that it is memoryless, it can be mapped to a Markov chain, and therefore it is perfect for analytical study.

### 3.2 Extended model with physical layer abstraction

In system level simulations we try to simulate a real continuous-time channel, but this proves to be a tricky task. Let us summarize the key system-level properties that should be captured.

- Path loss, TX power, sensitivity along with other components of the link budget define the signal strength as observed by receiver given that only one transmitter is active.
- Multiple transmissions are combined at the receiver according to MRC scheme.
- Each link is subject to slow and fast fading, which exhibit the expected correlation properties.
- The interference in the system defines the noise floor, that is the generated interference is looped back into the target cell.

- The SNR at the receiver is mapped into frame loss probability, which is used to synthesize loss events according to Bernoulli scheme.

The above points summarize the way channel is modeled in most system-level simulations. In the client relay study the procedure is very similar, apart from the fact that every node now is always a potential receiver or transmitter, while in most wireless network models only transmissions between nodes and BS are possible. This creates some unique conditions that make SLS design very challenging. In the Section 5, we will discuss the way SLS for client relay was implemented.

For the link layer abstraction the SNR mapping is used, a technique that allows to skip the tedious bit-by-bit modeling of packet delivery process. Such link level model proves to be one of the best choices, as it captures most of the effects critical for cell planning without compromising simplicity. While being harmful for the accuracy, this technique allows to simplify model up to a point when hundreds of packets and multiple nodes can be considered.

Most important downside of SNR mapping model is the correlation between consecutive states. Such correlation makes it nearly impossible to use it for analytical study, and also complicate the analysis for simulations due to transient period at start.

The list below summarizes the key difference of extended system model compared to baseline.

- Realistic fading channel

The channel is a stateful system, the fading processes on each link are time-correlated. The link capacity is determined by the available power, path loss and fading. Path loss depends on the distance between nodes and antenna height

- Different antenna properties

The BS antenna can be directional, covering 120 degree sector, or omnidirectional. TE antenna is always omnidirectional. TE antenna has smaller height than BS antenna, which results in different path loss on P2P links compared to terminal-BS links of the same length.

- Non-immediate duplexing

The transceivers can not switch instantaneously from transmission to listening and vice versa, so for the relay to be able to listen it has to be completely free of transmissions before, after, and during the intended listening interval. In practice the turn-around time is less than one OFDM symbol.

- Parallel frequency channels (OFDMA mode)

The system has several parallel frequency subchannels following the OFDMA principle. Each of them is independently slotted and allocated to the users by the BS. Each user is capable of receiving all of the frequency channels at once.

- Mobile users

The client terminals are moving around the cell, but can not leave it. The number of users is fixed. The motion is affecting both path loss and fading processes.

- Fragmentation procedure

The data packets to be transferred are fragmented into ARQ fragments, which are the minimal retransmission units. If one of those fragments reaches the retransmission limit, it is discarded along with the original data packet. Such operation is by far more realistic than the previous assumption, as fragment sizes are chosen to align to the slot boundary.

- Power control

The BS is capable of controlling the user terminal's transmission power within allowed limits based on the signal quality, resulting in closed-loop power control. The terminal may also employ open-loop power control if necessary.

- Downlink traffic

The downlink now affects the model, since the downlink framing affects the delay between transmission and the feedback. The ARQ feedback is now available only after the corresponding system message is scheduled in the downlink, and therefore can be more than one slot. We still assume that the downlink transmissions are error-free.

### 3.3 Cellular network model

In the simplest case we can assume an isolated cell with no interference. Such model, while being idealistic, is simple and allows analytical study. Only this model is used for the baseline system model. It does not, however, work for small cells as in such case the capacity is not limited by path loss or frequency resource, but rather the interference level.

A more complete model would include a proper planning scenario. In studies of cellular networks, usually a large cluster of 19 hexagonal cells is considered [24]. The problem is that such scenario requires massive computations, and does not eliminate the effect of border cells that have less neighbors. A different model is used, namely  $N$  strongest interferers [24]. The idea is that if we can emulate, for example,  $N = 6$  interferers from neighboring cells, we can create feasible model. Such task is simplified by the fact that we are considering uplink direction only, therefore we can just measure interference at the neighboring BS to calculate what should have been the interference at our own BS. The extended cellular network model is used in conjunction with extended system model.

In what follows, we will mostly consider the OFDM-based IEEE 802.16 systems, since for them a clear evaluation methodology is defined [24]. Also, such systems may operate in *time division duplex* (TDD) mode, which ensures that the relay can listen while



other users are transmitting. In *frequency division duplex* (FDD) systems such operation may not always be possible, as the TE will have to switch to other frequency for the DL reception, and during that period some other users that require relay may be transmitting.

Since in both IEEE 802.16 and 3GPP LTE the BS broadcasts a special message (UL\_MAP) that tells every user when it is scheduled to transmit, it is reasonable to consider a collision-free environment only. The presence of such map also means that every user in a cell knows exactly when all the other users are scheduled to transmit.

Following [19], this research concentrates on the decode and forward relay over amplify and forward strategy, as it provides the best overall spectral efficiency. It also allows to utilize the powerful FEC features of the underlying technology, which gives several extra DB's in sensitivity. It also allows to verify that the packet was received correctly before forwarding it. All those extra side-effects have been proven very useful in [5]. As a side-effect, the transceivers have to spend extra energy decoding error-correction code and encoding the packet again.

## 4 Analytical modeling

A wide spectrum of analysis and evaluation methods has been used during the research. In the following subsections, the most notable of them are summarized. To preserve continuity, for analytical methods I also present the most interesting results, as they are not directly usable for the full-scale system-level evaluation.

Before we continue to the discussion of the evaluation methodology, let us specify exactly what we are looking for.

- Most importantly, we are interested in the system throughput.
- Average delay is an important QoS parameter, and since we are optimizing retransmission procedures, it is of the highest importance.
- Derived statistics, such as average retransmission count per packet, may be of interest to show the relations within the network.
- System energy efficiency and individual energy efficiencies (in bits per transmission) in the relay and no-relay cases. Here we should differentiate between idle power, transmit power and receive power of a source node. Unfortunately, this requires the knowledge of the exact hardware specs, and is therefore replaced with a similar statistic - the bits per active time of transceiver.

Many other parameters may be of interest, such as the effect of client distribution on the relaying effectiveness, but those can not be directly found without the exact knowledge of the cell topology, which compromises generality.

### 4.1 Analysis of the baseline system model

There are several processes within client relay that are analytically tractable. Essentially, there are solutions for the following problems:

- Mean throughput per node
- Mean delay per node
- Mean energy efficiency per node
- The distribution for the medium access delay for a selected node.

Let us consider those.

### 4.1.1 Medium access delay distribution

Medium access delay is a very tricky parameter. In a queuing system, the service process might take different time for different requests. In case of a wireless network, the medium access delay is defined as the service time distribution for a packet that is about to be transmitted. The service is complete when the acknowledgment is received. This means that the smallest possible service time is one slot, as the feedback is available by the end of the slot. The medium access delay is not a very useful parameter for the end-to-end optimization, but it is very useful as a demonstration of the deep effect of client relay. Essentially, any impact of client relay on the network performance can be traced back to this distribution.

In order to obtain the distribution of the number of attempts that have to be taken in order to transmit a packet, one may observe, that within simplified system model the retransmission process is a Markov process, which branches every time a new node joins relaying. Obviously, it has consuming states that correspond to packet being delivered or dropped.

First of all, since the channel properties are fixed, for each set of sources  $S$  and destinations  $D$ , and every possible transmission setting  $P$  (such as power or modulation scheme), there are known delivery probabilities  $q(S, x, P) = \text{const}(t)$ , where  $x \in D$ .

Since the amount of nodes that may potentially join the transmission  $N$  is finite, and there is a finite limit on retransmission count  $n_r$ , there is a finite set of states in which the system may be found before each transmission. It is now clear that for each packet of each source one may construct a finite set of states defining the delivery process. Since the retransmission process is memoryless in nature, Markov chain can be used to represent the delivery process. The chain has following states:

1. There are exactly two absorbing states, which represent the successful delivery and packet drop. We will denote them as  $s_{OK}$  and  $s_{DROP}$ .
2. Each attempt  $i$  for each set of sources  $S$  is also a state, denoted as  $s_{i,S}$ . Any node from  $S$  except for the original source may stop transmitting at any given frame. In general case the relay may choose either not to eavesdrop on the originator's packets or not to transmit them subject to some relaying policy.

System is initialized with state  $s_{0,(A)}$ . The transition probability matrix  $\Lambda$  may be formed according to simple algorithm:

1.  $\forall j \leq i P(s_{i,S1} \rightarrow s_{j,S2}) = 0$  – The amount of attempts spent can not decrease.
2.  $P(s_{n_r,S} \rightarrow s_{DROP}) = 1$  – The packet is always dropped when retransmission limit is reached.
3. The next formula defines the rest of the transition probabilities.  $P(s_{i,S} \rightarrow s_*) = q(S, *, P)$  – the delivery probability is a function of the channel and nothing else, and is predefined according to system model.

One may obtain the distribution of the number of attempts necessary to deliver a packet  $D$  or the loss probability by simply calculating the power of the matrix  $\Lambda$  we formed. The probability of successful delivery in  $j$  steps is found as  $D_j = \Lambda^j[1, OK]$ , where  $j \leq n_r$ . The packet drop probability is found as  $p_l = \Lambda_r^n[1, DROP]$ . In order to obtain the *probability mass function* (pmf) of the delivery attempt count  $d_j$ , one should take the derivative of  $D_j$ .

One extra step allows to use the obtained result to get the medium access delay. In order to make that step possible, it is assumed that all nodes are in saturation conditions, which implies that the schedule is fixed. Therefore, in case of fair scheduling, the time between consequent transmission grants for any given user would be constant and proportional to the number of clients. In order to take into account the existence of other nodes in the access frame, we add  $n - 1$  zeros between each value, assuming that the node retains its position in schedule. Such assumption typically holds in case of round-robin scheduling. Here it is necessary to note that the resulting function will not be valid as a delay distribution. Since we did not consider the dropped packets, we need to normalize it with coefficient  $k = \frac{1}{1-p_l}$ .

From the medium access delay distribution it is possible to obtain the average delay, which is  $L = E[D]$ . It is possible to find out the average throughput of the resulting link as  $T = \frac{1-p_l}{L}$ . Since the approach allows us to take into account relaying, but does not force us to do so, we may also use it for non-cooperative case by adjusting transition probabilities.

So, we have defined another possible way to find out delay and throughput we have been looking for. The applicability is, however, very limited due to following factors:

1. It is difficult to extend the result to non-saturation conditions.
2. It is not possible to use non-memoryless channel (since the previous user's packet will affect the selected user's transmission).

So, the presented approach is not usable for full-scale modeling of real networks. Further development of this approach have proven to be very complicated and did not scale, but another approach allowed to relax the saturation condition requirement.

#### 4.1.2 Mean performance parameters for arbitrary load

In non-saturation condition, the success or failure of one node affects the service rates for all other nodes. Careful observation shows, that we can not assume that delivery delay distributions for the nodes are dependent through scheduling mechanism. However, such dependency does not affect delivery success, as described above. Therefore, the only thing that can not be applied anymore is the assumption that the node retains its position in the queue.

To compute mean performance parameters we sometimes approximate the real system. To analyze the delays, we relax the dependencies between queues and compensate

Table 2: Analytical model notations

Variable	Description
$\lambda_X$	Mean arrival rate of packets to node $X$
$p_{AB}$	Probability of successful reception at $B$ when $A$ transmits
$p_{RB}$	Probability of successful reception at $B$ when $R$ transmits
$p_{AR}$	Probability of successful reception at $R$ when $A$ transmits
$p_{CB}$	Probability of successful reception from $A$ and $R$ at $B$ when nodes cooperate
$N_X$	Maximum number of transmission attempts for node $X$
$\tau_{AR}$	Mean service time of a packet from node $A$
$\tau_{RA}$	Mean service time of a packet from node $R$
$\rho_{AR}$	Queue load coefficient of node $A$
$\rho_{RA}$	Queue load coefficient of node $R$
$q_X$	Mean queue length of node $X$
$\delta_X$	Mean packet delay of node $X$
$\eta_X$	Mean throughput of node $X$
$p_{rx}$	Probability of opportunistic eavesdropping
$p_{tx}$	Probability of cooperative transmission attempt
$\epsilon_X$	Mean energy expenditure of node $X$
$\phi_X$	Mean energy efficiency of node $X$
$P_{RX}$	Average power in the packet reception state
$P_{TX}$	Average power in the packet transmission state
$P_I$	Average power in the idle state

for the introduced error. This makes it impossible to find pmf's, but on the other hand the model works in non-saturation conditions. Below I present results of the research that was started by me and later extended by our research group in [25, 9].

For simplicity, we consider a simplified system with two source nodes (the originator  $A$  and the relay  $R$ ). For the purposes of the analytical study, we assume that we know most of the system's observable parameters according to Table 2. We take into account almost all parameters one would expect to see in a wireless network model. The resulting approximation for the average delay of the originator  $A$  is:

$$\delta_A \cong \frac{\rho_{AR}}{\lambda_A} + \frac{\lambda_A}{2(1-\rho_{AR})} \left[ X \frac{2-\tilde{p}_{CB}}{\tilde{p}_{CB}^2} - Y \frac{2-p_A}{p_A^2} \right] - \frac{\lambda_A X}{2\tilde{p}_{CB}(1-\rho_{AR})} \left[ \frac{2N_A}{\tilde{p}_{CB}} + \frac{2-\tilde{p}_{CB}}{\tilde{p}_{CB}^2} \right] (1-\tilde{p}_{CB})^{N_A} + \frac{\lambda_A Y}{2p_A(1-\rho_{AR})} \left[ \frac{2N_A}{p_A} + \frac{2-p_A}{p_A^2} \right] (1-p_A)^{N_A},$$

where  $X = \frac{p_{AR}p_{rx}(1-\tilde{p}_{AB})\tilde{p}_{CB}}{1-\tilde{p}_{CB}-(1-\tilde{p}_{AB})(1-p_{AR}p_{rx})}$  and  $Y = X - \tilde{p}_{AB}$ . Similarly, for the relay  $R$  we obtain:

$$\delta_R \cong \frac{\rho_{RA}}{\lambda_R} + \frac{\lambda_R(2-\tilde{p}_{RB})}{2(1-\rho_{RA})\tilde{p}_{RB}^2} - \frac{\lambda_R(1-\tilde{p}_{RB})^{N_R}}{2(1-\rho_{RA})} \left[ \frac{2N_R}{\tilde{p}_{RB}} + \frac{2-\tilde{p}_{RB}}{\tilde{p}_{RB}^2} \right].$$

The delay is one of the key parameters for client relay, as it shows directly if our system is retransmitting less or not. Unfortunately, as the queues of the nodes are mutually dependent, it is very difficult to obtain the exact delay, however the simulations show that

the approximation is very close.

For the throughput, as it was indicated before, there is no need for approximation, the exact value of the throughput  $\eta_A =$

$$\begin{cases} \lambda_A \left(1 - (1 - \tilde{p}_{AB})^{N_A}\right), & \text{no saturation} \\ (1 - \lambda_R \tau_{R0})(\tau_{A0})^{-1} \left(1 - (1 - \tilde{p}_{AB})^{N_A}\right), & \text{saturation for } A \\ (2\tau_{A0})^{-1} \left(1 - (1 - \tilde{p}_{AB})^{N_A}\right), & \text{saturation for } A, R. \end{cases}$$

The expression for  $\eta_R$  may be established accordingly. The resulting functions are segmented due to the conditional nature of saturation condition. Here the saturation conditions imply the following relations:

- For A:  $(\lambda_A \tau_{A0} + \lambda_R \tau_{R0} > 1)$  and  $(\lambda_R \tau_{R0} < 0.5)$ .
- For R:  $(\lambda_A \tau_{A0} + \lambda_R \tau_{R0} > 1)$  and  $(\lambda_A \tau_{A0} < 0.5)$ .
- For A and R:  $(\lambda_A \tau_{A0} > 0.5)$  and  $(\lambda_R \tau_{R0} > 0.5)$ .

The exact mean energy expenditure of the source nodes is proportional to their activity in listening and transmitting with weighting coefficients  $P_{RX}$  and  $P_{TX}$ .

$$\begin{aligned} \varepsilon_A &= P_{TX} \eta_A \tau_{A0} + P_I (1 - \eta_A \tau_{A0}) \text{ and} \\ \varepsilon_R &= P_{TX} \left( \eta_R \tau_{R0} + \eta_A P_{tx} \frac{1 - \tilde{p}_{AB} \tau_{A0}}{\tilde{p}_{CB} - \tilde{p}_{AB}} \right) + P_{RX} p_{rx} \left( \eta_A \tau_{A0} - \eta_A \frac{1 - \tilde{p}_{AB} \tau_{A0}}{\tilde{p}_{CB} - \tilde{p}_{AB}} \right) + \\ & P_I (1 - \eta_R \tau_{R0} - \eta_A \tau_{A0} p_{rx}) - P_I \left[ (p_{tx} - p_{rx}) \eta_A \frac{1 - \tilde{p}_{AB} \tau_{A0}}{\tilde{p}_{CB} - \tilde{p}_{AB}} \right], \end{aligned}$$

whereas the exact mean energy efficiency is  $\varphi_A = \frac{\eta_A}{\varepsilon_A}$  and  $\varphi_R = \frac{\eta_R}{\varepsilon_R}$  respectively.

Our analytical approach is based on calculating the mean service time. In particular, the service time of a packet from node A conditioning on the fact that  $\lambda_R = 0$  is:

$$\tau_{A0} = \frac{p_{AR} p_{rx} (1 - p_{AB})}{1 - p_{CB} - (1 - p_{AB})(1 - p_{AR} p_{rx})} \frac{1 - (1 - p_{CB})^{N_A}}{p_{CB}} - \left( \frac{p_{AR} p_{rx} (1 - p_{AB}) p_{CB}}{1 - p_{CB} - (1 - p_{AB})(1 - p_{AR} p_{rx})} - p_{AB} \right) \frac{1 - [1 - (1 - p_{AB})(1 - p_{AR} p_{rx})]^{N_A}}{[1 - (1 - p_{AB})(1 - p_{AR} p_{rx})]^2}.$$

Analogously,  $\tau_{R0} = \frac{1 - (1 - p_{RB})^{N_R}}{p_{RB}}$ . Also we take advantage of the following auxiliary probabilities:

$$\begin{aligned} \tilde{p}_{AB} &= p_{AB} \frac{\rho_{A0}}{\rho_{AR}}, \quad \tilde{p}_{RB} = p_{RB} \frac{\rho_{R0}}{\rho_{RA}}, \\ \tilde{p}_{CB} &= [p_{CB} p_{tx} + p_{AB} (1 - p_{tx})] \frac{\rho_{A0}}{\rho_{AR}}, \text{ and} \\ p_A &= \tilde{p}_{AB} + p_{AR} p_{rx} - \tilde{p}_{AB} p_{AR} p_{rx}. \end{aligned}$$

Clearly, the queue load coefficients of node A conditioning on the fact that  $\lambda_A = 0$  is  $\rho_{A0} = \lambda_A \tau_{A0}$ . The expression for  $\rho_{R0}$  is symmetric. We set  $\rho_{A0} > \rho_{R0}$  as an example. The key idea of our approach is to approximate the unconditional queue load coefficients as  $\rho_{AR} \cong \frac{\rho_{A0}}{1 - \rho_{A0}}$  and  $\rho_{RA} \cong \frac{\rho_{A0}}{1 - \rho_{A0}} - \rho_{A0} + \rho_{R0}$ .

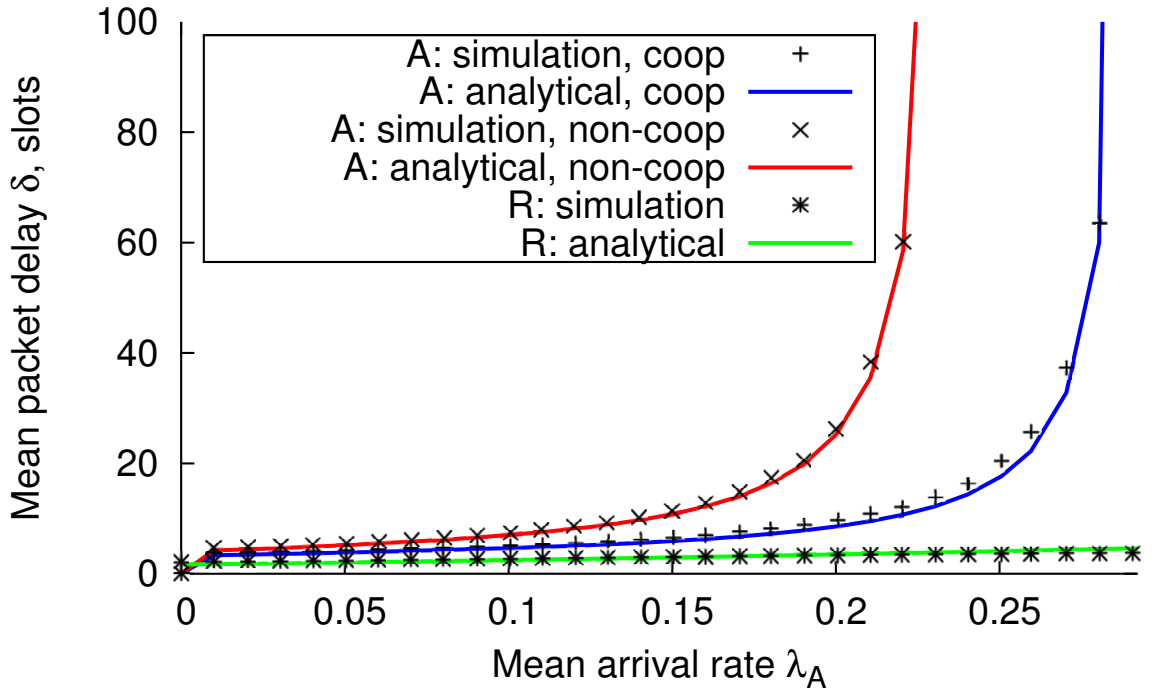


Figure 5: Packet delay verification

The obtained results can be easily cross-referenced with simulations. Let us vary  $\lambda_A$  across the stability region and set e.g.  $\lambda_R = 0.15$ . The opportunistic cooperation with probability  $p_{rx} = 0.5$  yields the dependencies as shown in Figure 5. The result for node  $R$  does not change and is shown only once. We conclude, that our simulation accords extremely well with the analytical results. The precision of the simulator is a tunable parameter, and in this case it is set to 0.1% of the measured value.

Convergence is controlled by means of method of replications, its implementation is detailed in [8] and further on in section 5.

## 4.2 Analysis of the extended system model

The extended system model is intended to fit 4G networks. In this section I show how exactly it can be done, and the application of the analytical model for baseline system model. First of all, we notice that all the signaling that we have assumed to be present is in fact available in the service messages both in LTE and in IEEE 802.16. This includes acknowledgments, transmission timetables and power control.

Next-generation wireless access systems operate with fixed frames and OFDMA scheme. Each frame contains multiple time-frequency resource blocks (in LTE terminology) or slots (in 802.16 terminology). For consistency, we will address those as slots. According to extended system model the TE can not listen and transmit at the same time. Combined with the fact that multiple users may transmit simultaneously, it makes eavesdropping more complicated, since now the possible relay has to decide whether it

will help, say, users A and B that are retransmitting, or it should listen for transmissions of user C that is scheduled at the same time. Figure 6 clarifies the situation. Different colors denote different users in the schedule.

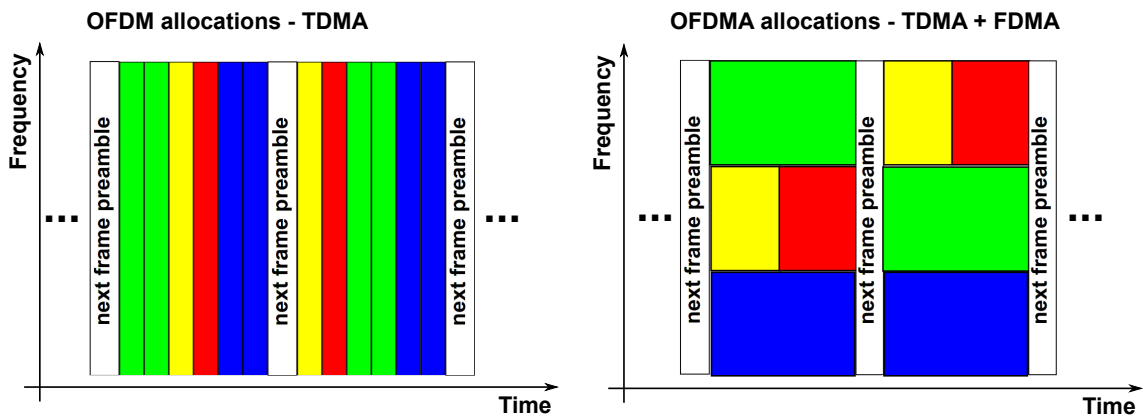


Figure 6: The difference between OFDMA and OFDM resource allocations

Rephrasing from IEEE 802.16 standard [1], the rule for normal data allocations is as follows: for the user data slots, the allocation shall start at the lowest numbered unallocated subchannel on the first unallocated OFDMA symbol defined by the Allocation Start Time field of the UL-MAP message that is not allocated with service channels. The allocations shall represent the number of slots provided for the allocation. Each allocation IE shall start immediately following the previous allocation and shall advance in the time axis. If the end of the UL zone has been reached, the allocation shall continue at the next subchannel at first OFDMA symbol allocated to that zone that is not allocated with service channels. Therefore, each transmission has a "starting" and "ending" slot number. This slot numbering scheme greatly simplifies the allocator at BS, but at the same time it makes it more difficult to employ techniques like client relay. The nodes can not eavesdrop in OFDMA mode, since they are all transmitting at the same time or at least overlap somewhere. Therefore, blind opportunistic relaying does not fit well, in most cases the relay will have uplink packets scheduled at the same time as the node for which it could have listened.

There are several possible solutions to the indicated problem. First solution is to schedule users in such a way that they get the whole channel for shorter time, instead of just one channel for longer time. Unfortunately, this contradicts the idea of OFDMA, since with a 20 MHz channel even the smallest time allocation will have a size of over 6 kilobytes. For VOIP traffic it is just too much, and therefore the utilization will be quite low. Also, the duplexing delay would not allow to switch between listening and transmission immediately, and therefore possible cooperation slots should be spaced from eavesdropping slots. This proved to be very challenging task as well.

Alternative solution comes from the shape of the reception reliability curves and TDD operation mode. Basically, users at the center of the cell never really require relaying,



since their transmitters can compensate fading with extra power. Also, due to their position away from neighboring cells, they do not cause interference increase.

The cell edge users, however, do require relaying, and the closer they get to the cell edge, the more they need cooperation for stable communication. Increasing power for them is not possible, and would not have worked anyway due to interference increase. Based on this, we can split the users into two groups - *near* users that never use relays, but relay packets for others, and *far* users that use relays but never relay themselves. In such case it is possible to schedule near users in odd frames, and far users in even frames. This is where TDD mode is beneficial - since the number of far users is not necessarily equal to number of near users, it allows us to redistribute resources between frames to maintain allocation fairness among the two groups. Fairness here means that each group gets resources in proportion to the user count.

Now when the even frame starts, each user from near group decides if it should listen or transmit opportunistically. Therefore, there are always some listeners and also some cooperators available. Also when the node decides to listen, it would capture all the transmissions that require relaying, and therefore would maximize the relaying gain.

The obvious choice of the border between near and far users based on their capability of path loss compensation is, according to our research, not necessarily the best one. For example, every node at the cell edge should be able to reach reasonable number of relays, otherwise the whole arrangement is pointless. In the following subsection, we show how the BS may decide more intelligently on where to put the border  $\theta$  between near and far users by predicting the gain obtained from relaying.

#### 4.2.1 Client-relay aware scheduling

Applying the above results to the concept of near and far nodes, we can use the simplified relay model to predict the value of the parameter  $\theta$ . First of all, we map the real 120 degree sector into a straight line, adjusting the node density as shown in Figure 7

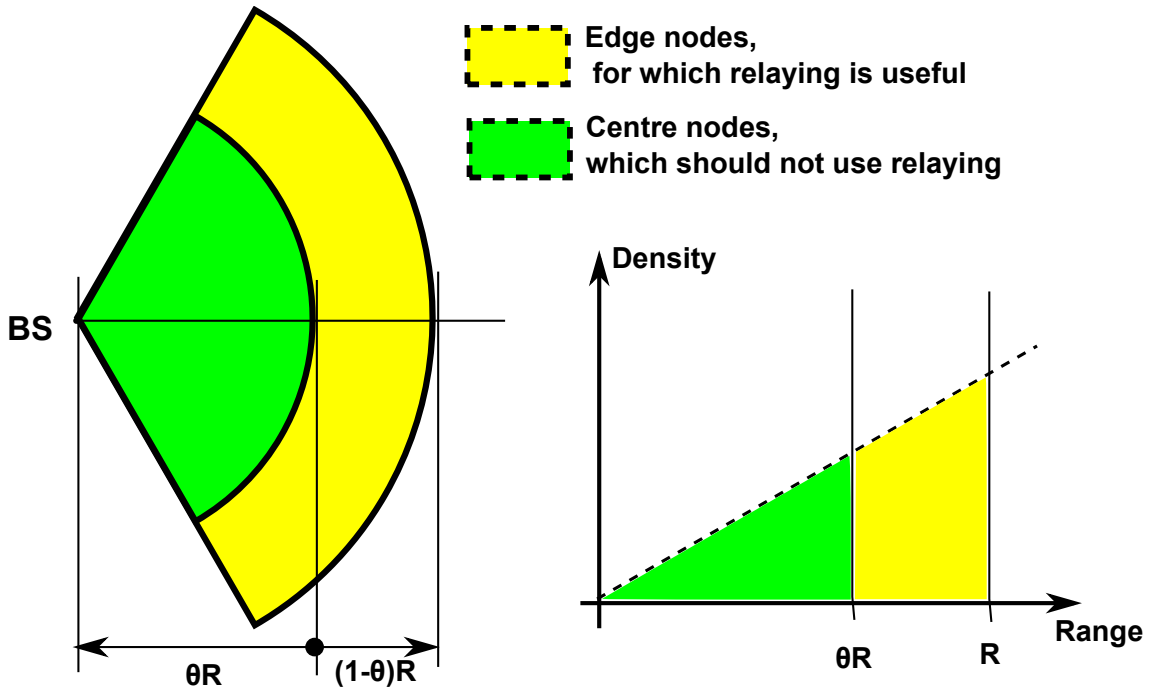


Figure 7: The mapping of sector into a line for analysis

Assume that packet delivery probabilities  $p_{AB}$  and  $p_{RB}$  are certain functions of one variable  $x$ , where  $x$  stands for a distance between the source and the base station, while  $p_{RB}$  is a function of  $x_A - x_R$  where  $x_A - x_R$  is a distance between the relay and the originator. Those functions can be easily obtained based on the cell planning information.

Also let the cooperative delivery probability  $p_{CB} = p_{CB}(x_A, x_R)$  be a known function of node positions  $x_A$  and  $x_R$ . As before,  $R$  denotes the cell radius. Probability density function of node position is  $f(x) = 2x/R^2$ , where  $x$  - the distance between the node and the BS. Here we assume that the cell sector is narrow enough to be approximated with a straight line.

We set the fixed threshold  $\theta$  dividing the set of all nodes  $U$  into two:  $U = S_A \cup S_R$ , where  $S_A$ ,  $S_R$  are the sets of relay and originator nodes respectively. Fix a node  $A$  from the set  $S_A$ . It can be characterized by the distance  $x_A$  and probability  $p_{AB}$ . For the fixed  $A$  we find the set of possible relays taking into account the number of sources between the BS and the point  $x = \theta$ .

In order to average the characteristics of relay nodes we integrate the region  $0 < x < \theta$ . Then we get the following averaged variables:

$$\bar{x}_R = \int_0^{\theta} xg(x)dx, \quad (1)$$

where  $g(x)$  - probability density function of relay nodes for the node  $A$  (if any node from  $S_R$  is ready to become a relay for  $A$  then  $g(x) = 2x/\theta^2$ ). In that case

$$\bar{x}_R = \frac{2\theta}{3}, \quad (2)$$

The average probability of successful transmission to BS:

$$\bar{p}_{RB} = \int_0^\theta p_{RB}(x)g(x)dx = \frac{2}{\theta^2} \int_0^\theta p_{RB}(x)x dx. \quad (3)$$

The average probability of successful eavesdropping:

$$\bar{p}_{AR}(x_A) = \int_0^\theta p_{AR}(x_A - x)g(x)dx = \frac{2}{\theta^2} \int_0^\theta p_{AR}(x_A - x)x dx. \quad (4)$$

The average probability of successful transmission:

$$\bar{p}_{CB}(x_A) = \int_0^\theta p_{CB}(x_A, x)g(x)dx = \frac{2}{\theta^2} \int_0^\theta p_{CB}(x_A, x)x dx. \quad (5)$$

Given four main probabilities described above we can use the three-node relay model and get the gain of the cooperative mode based on any desired parameter as follows:

$$\delta\eta(x_A) = \eta^* - \eta, \quad (6)$$

where  $\eta^*$  is the optimized parameter in cooperative mode and  $\eta$  in non-cooperative mode.

To find the aggregated gain we sum individual gain for all possible originators:

$$S(\theta) = \frac{2}{R^2} \int_0^R \delta\eta(x_A)x_A dx_A. \quad (7)$$

Maximizing  $S(\theta)$  by the parameter  $\theta$  we can get an optimum for given node distribution and delivery probabilities introduced above. We could maximize, for example, throughput, delay or energy efficiency like that. The most important drawback of such approach is that it does not take into account different coding rates that are possible, and

therefore can not be applied immediately to the BS scheduler as long as adaptive modulation is used. Another drawback is the inability of such model to cope with inter-cell interference. These issues could also be addressed, but it turns out to be much easier to use simulations to tabulate the values of  $S(\theta)$ .

In the following section I consider the simulation environments and the methods employed for simulation study. We will see how the analytics could be applied to verify the consistency of simulations and also to guide the research in a proper direction, thus saving time and effort.

## 5 Simulation methodology and environment

Several aspects of the *system level simulation* (SLS) are of a great interest in order to understand how the results have been obtained, This section focuses on topics related to accuracy, sources for models used and motivations behind. The main problem about most cellular network simulation tools is that they enforce star topology inside the cell. Unlike contemporary cellular system, client relay network requires complete connectivity graph, which does not fit into available simulation environments, even with complete replacement of PHY layer. This was the main reason why the new SLS was designed specifically for client relay studies. In the following subsection it is shown how different well-known concepts can be applied to the design of new SLS.

### 5.1 Simulation platform

System-level simulations of communications networks are very resource-hungry, mostly due to complex statistical relations that have to be uncovered. For example, it is not enough to observe just one or two packets in the network to conclude if it is stable or not, the actual observation should last a lot longer than any feedback process in the network. If the buffer sizes are large, this can mean hundreds of thousands of packets. Wireless networks add one extra problem - the channel model is typically an  $N^2$  complexity algorithm, and it has to be run for each and every symbol transmitted over the radio interface.

Currently, some SLS tools are implemented as a step-advance models with Matlab. Unfortunately, complex logic of the network in question makes them rather slow, practically useless for long runs. Implementation with C/C++ would be generally much faster than Matlab, but requires a lot of coding for handling of memory, configuration parameters, input and output. All this makes C programs hard to maintain for research properties, where quick working solutions are usually preferred over thoroughly optimized production quality programs. At the same time, compiled C code results in, probably, the most resource efficient implementation for most algorithms.

Generally, scheduling and relay selection algorithms operate with such terms as sets, maps and dynamic priority queues. Such high-level data structures do not fit well into the paradigm of C/C++. Therefore, it was decided to split the model into two parts: the link-level model implemented with C and the control level implemented with Python. Obviously, the bulk of calculations was to be performed at the link-level, and the complex control logic was to be implemented with Python. The post-processing and pre-processing tasks have been left for Matlab. The exact structure and tools used can be seen in Figure 8

The Python interpreter has even more flexibility than Matlab and runs a lot faster. A special software called PyPy, which is essentially a compiler for Python, allows to run Python programs with speeds close to those approachable with compiled C++. At the same time it retains the flexibility of the scripting language. Also it provides the unique

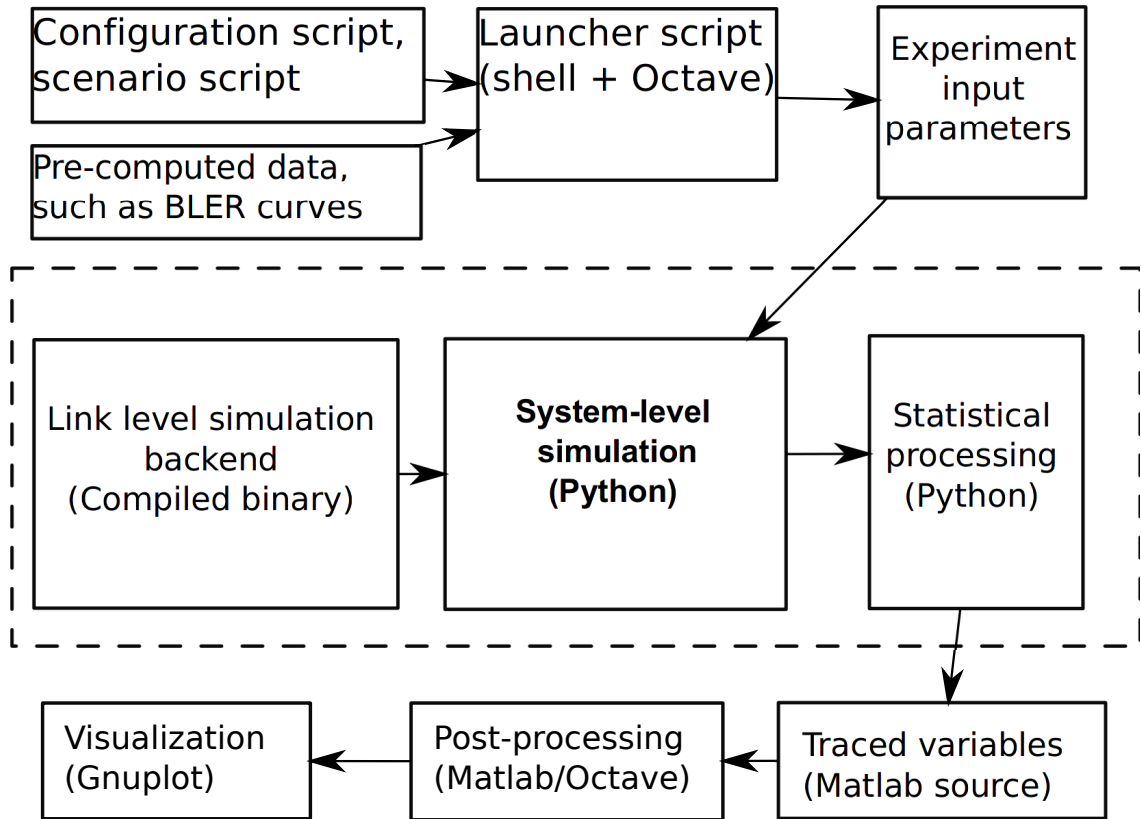


Figure 8: Simulation tool structural scheme

opportunities in configuration, as most of the model scenario can be constructed with object constructors and for loops, rather than cryptic configuration files or unreadable shell-like TCL scripts.

As for the C part, the SWIG wrapper generator allows to maintain the linking between the Python top-level and binary backend in a clean, efficient and cross-platform manner. It also allows for a nice trick - the wrappers can be also generated for Octave, which allows to verify the channel model parts against their analytical models in Matlab-like environment.

## 5.2 Simulator internal structure

The core of the simulator is flow-oriented, that is it has no explicit notion of state, all events are globally scheduled over common time line and can happen simultaneously if necessary, that is all transmitters scheduled to transmit at some instant will appear in the channel at the same time and will be processed accordingly. The complexity of the model calls for a very high-level description. A simplified block diagram in Figure 9 shows the internal structure of the simulation core. The chart does not match the object model exactly, but captures the dependencies between simulation entities.

The simulation is configured in an object-oriented way very similar to OPNET [26],

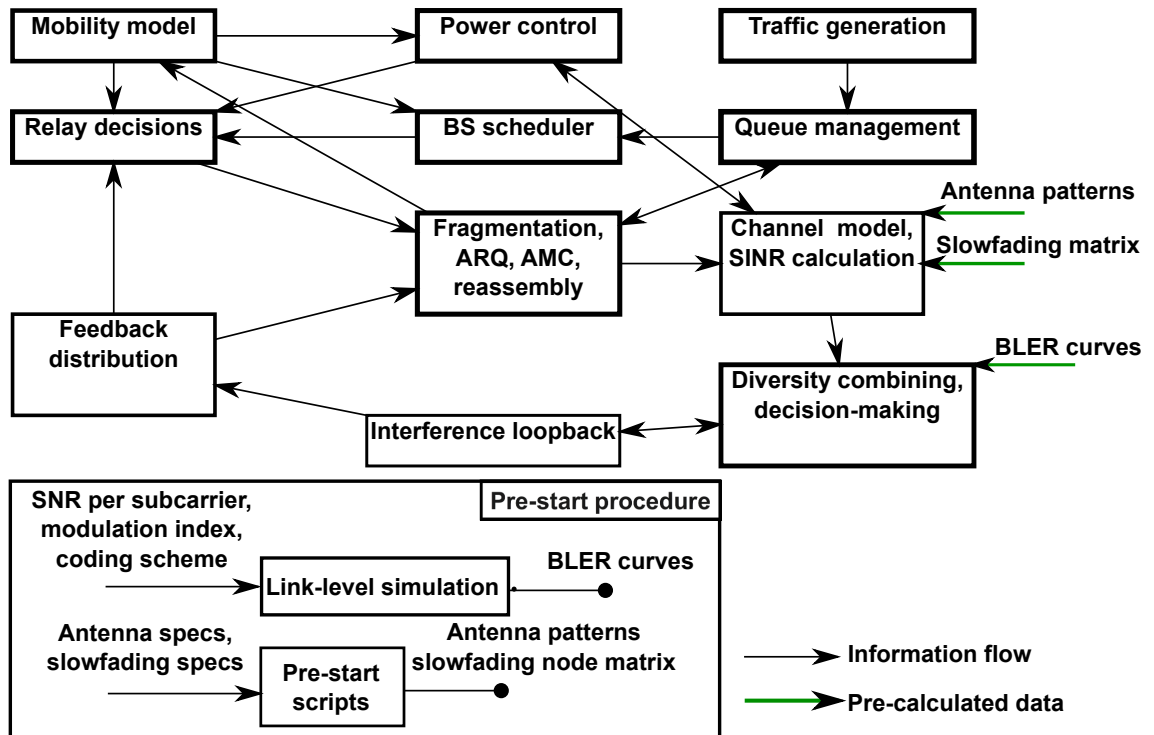


Figure 9: Simulation tool internal flow diagram

but with a scripting language instead of C++, which allows for easier and more flexible configuration, as the user may override the default behavior if needed directly in the configuration file. The configuration files used for the system-level evaluation are presented in the appendices. Let us now briefly touch the most notable parts of the simulator in more detail.

### 5.3 Accuracy control

Another major problem in most modern network simulators such as NS2 [27] or OPNET [26], is the estimation of the result reliability. Basically, as networks are modeled as random processes, there is always a mean value that we are looking for and some deviation around it. Method of replications [28] allows to compute how accurate the current estimate is and to act accordingly, for example increasing the simulation length.

Replication analysis can be summarized in a simple algorithm 2.

---

**Algorithm 2** Modeling a network with replications
 

---

```

 $a = 0$ 
repeat
  Reset all states in the network
  Simulate the network for some small time
   $a = a + 1$ 
  if Current error < Target error then
    Break loop
  end if
  if  $a >$  maximum replication count then
    Terminate, system is non-stationary
  end if

until Exit condition triggered
Save all statistics
Terminate, simulation completed

```

---

The replication analysis is a very consistent method, as it allows to estimate the mean values and variance of the measured variables. However, there may be correlation between consequent batches of data, which could affect the results. An alternative is the regenerative approach, which differs in the positions of the replication cut points. The regenerative simulation detects the moments when all queues are empty (system is in the initial state) and cuts the replications at those moments. This ensures that the batches are not correlated in any way, but is not suitable for queuing systems that are not stable, as they never return to their initial state, and therefore replications never happen. Therefore, while being a better option for simulation of the baseline system model, such method is incapable of coping with extended system model scenarios, that tend to have one or two edge-nodes that always have packets in their queues, which results in ridiculously long simulations.

### 5.3.1 Path loss model and fading

The path loss model is borrowed from [24] for the uplink communication with the BS (which, in turn, originates from [29]). A similar empirical path loss model in [30] is used for eavesdropping by mobile terminals.

Fast fading is described by a thoroughly validated Rayleigh fading process, which is modeled according to Young's method [31]. Each transmission ray is associated with a fast fading process, and those are not correlated for consequent transmissions. This greatly decreases the amount of state information without sacrificing the main purpose of fast fading simulation - random unpredictable loss of data.

Slow fading process simulation, however, is non-trivial due to extra mobile-to-mobile links. Those invalidate the classic approach with linear interpolation, between fixed slow-fading nodes, which expects all the transmissions to be directed to the BS. Therefore we



extend the model proposed in [24]. It has been shown by [32], that slow fading process  $L_{sf}$  has high correlation with the speed of a node, which should be taken into account. In our model, following [32], we assume that slow fading between every two nodes depends on the node's speeds, and not on their current position. Therefore a static node would receive some random  $L_{sf}$  which would not change over time. By contrast, the mobile nodes should have  $L_{sf}$  updated periodically, such that the speed correlation properties are preserved.

Since there are multiple directions in which the transmissions may happen, every node has several slow fading processes (4 to be exact) associated with it, each one responsible for a certain general direction.

As a result, we deliberately disregard the static terrain features that may be present inside the cell, since accounting for them would require full ray-tracing, and therefore would be extremely slow for SLS. On the other hand, we account for speed-dependent correlation coefficient  $\alpha = \varepsilon(D)^{\frac{vT}{D}}$ . Here  $D$  is the distance over which the correlation between slow fading samples is exactly  $\varepsilon(D)$ , whereas  $v$  is the speed of a node, and  $T$  is the time interval between the packets. Slow-fading strength is typically specified for  $\varepsilon = 0.5$  [24]. In our simulation tool, every time a node changes its position,  $L_{sf}$  is recalculated by low-pass filtering of the Gaussian noise process. The cutoff is set at  $\omega_c = 1 - \alpha$ , which gives exactly the desired result and also agrees with the measurements from [32]. Obviously, when two mobile nodes communicate, both of them have their slow fading processes updated, which automatically takes faster channel changes into account.

### 5.3.2 Power control and interference loopback

A simplified version of IEEE 802.16m power control is employed by the mobile nodes (with baseline power control according to [1] and extra modifications from [33]). The exact formula is  $P_{transm} = SNR_{target} + P_{noise} + L$ , where  $SNR_{target}$  is the target SNR value,  $L$  is the reported uplink path loss, and  $P_{noise}$  is the noise level estimated within a cell. The maximum transmission power is limited according to the frequency band specifications. This is essentially closed-loop power control, as the BS is assumed to calculate  $L$  accurately.

Interference loopback mechanism is very similar in essence to power control, and can be described as follows. The measurement nodes monitor all subcarriers for transmissions and record total power. Then this power is averaged over a long period of time (50 frames) and over all measurement nodes to avoid fluctuations, and then used as a thermal noise floor in all receivers. Such scheme gives the same results as  $N$  strongest interferers scheme, but is much better for client relay, as there would be needed much more interferers to model it accurately, while with measurements it is not important whether the clients are relaying or not. Following the original idea of  $N$  strongest interferers, 6 special measurement nodes are positioned at the nearest base stations that operate at the same frequency. One may find the frequency reuse schemes that have been used for simula-

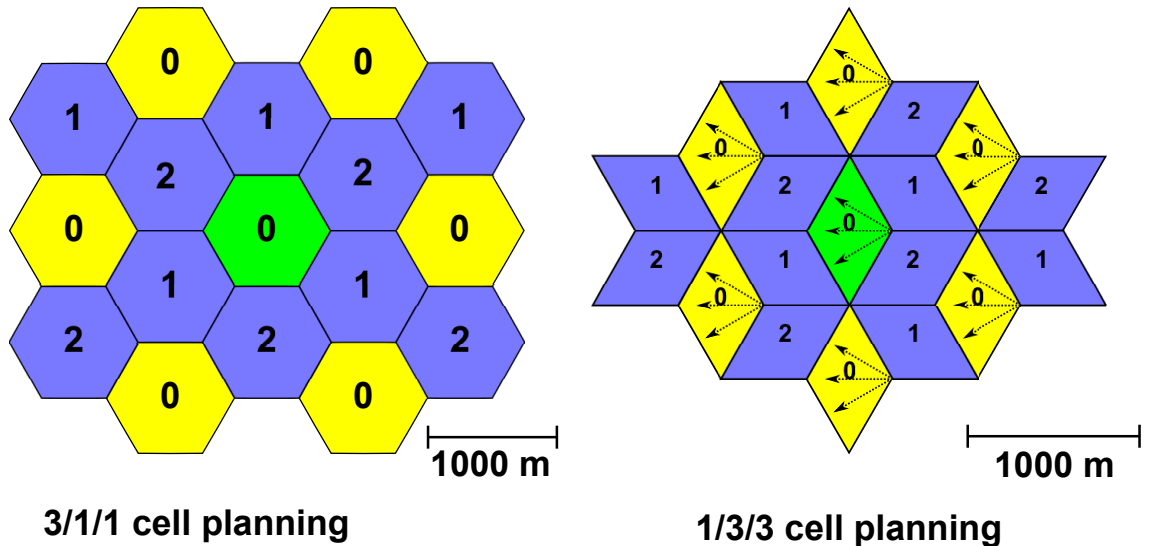


Figure 10: Network planning used for simulations

tions in the Figure 10. In Figure 10 red cells correspond to the cells which are potential interferers, and are also affected by interference. Blue cells use different frequency and do not affect the selected cell, which is colored green.

### 5.3.3 Signal detection and SNR mapping

As link level simulations are very computationally-expensive, it is not practical to run those inside channel abstraction module of SLS. Therefore, the SLS requires a special mapping function that would allow to map channel conditions such as path loss, noise level, delay spread and such (which are rather easy to obtain) to the delivery success or failure events. Since such mapping represents a random process, it is usually possible to capture it with a simple technique called effective SNR mapping [24].

The idea behind effective SNR is quite simple. One may use a model of AWGN channel to compute the *block error rate*(BLER) for a given modulation, SNR and code rate combination. Here block is a forward error correction code block, not the actual packet. By collecting such mappings for a reasonable range of SNR's and all possible modulation and coding schemes, a so called BLER curve family is constructed. Now simple table lookup with interpolation allows to find the BLER value for given channel conditions quickly.

In a compound signal such as an OFDM signal, each subcarrier has its own SNR, but taking into account 1024 different values is still too expensive for SLS, therefore SNR is averaged over all subcarriers of interest. After that, the BLER curve is used to obtain the loss rate for the whole symbol. When the BLER is available, a Bernoulli scheme can be used to generate the success or loss events for a given packet.

## 5.4 Protocol abstraction

Protocol abstraction is the core part of any SLS, and therefore requires special attention. For the baseline system model, we can directly utilize the relaying algorithm 1 from section 3. Due to the way the system is described, scheduler is reduced to a simple selection algorithm that just picks the next transmitter instead of coming up with actual schedule. As there is only one cell actually modeled, there is no need to have any time units smaller than 1 slot. Within each slot, the same set of events would be triggered all the time:

1. Slot start
  - (a) Run traffic generators for all nodes
  - (b) Run scheduler to decide which node transmits next
  - (c) For the scheduled node choose if it will be new transmission or not
  - (d) Run relay selection for each node and get the lists of cooperators and listeners
  - (e) Add the information on transmitting and listening nodes to the channel model
2. Slot duration - skip time, as nothing can happen
3. Slot end
  - (a) Calculate the transmission outcomes for each node in listener list
  - (b) Register the packet delivery or loss with the source

The presented above list of actions is obviously quite far from what is happening in a real network, but it proves to be a rather close approximation. Complex systems like 802.16, however, have some features that could not be disregarded and do not fit into such simple model. Those are considered in more detail further on.

### 5.4.1 IEEE 802.16 mapped to extended system model

A most critical feature of any SLS is the ability to capture properties, typical to the evaluated system's protocol stack. The complexity of the SLS in that sense depends on the complexity of the protocol. Therefore, to capture all the properties of IEEE 802.16 of LTE-advanced network accurately, one needs to implement the complete protocol stack for both TE and BS, and also for the E-node B in case of LTE. In most cases such detail is not needed, however. What is needed is to capture the most important relations while abstracting most of the smaller details, not relevant for current scenario. This can be achieved by fixing some tunable parameters that would normally be dynamic to some fixed values. Let us now look at the protocol abstraction that was performed for the client relay evaluation.

- Frame structure

The TDD mode is used for simulation, as it is in fact more flexible. In general, the frame consists of several defined sections, in order: the physical preamble, downlink subframe and uplink subframe. Preamble is technically a part of downlink subframe and carries no information. One can always shrink the downlink subframe to zero and assume downlink is available separately in FDD mode. To allow the transceivers to switch between reception and transmission, a special duplexing delays are inserted between subframes.

- Downlink subframe content

The downlink subframe begins with the *uplink map message* (UL\_MAP), that specifies which nodes are scheduled to transmit during the next uplink and when. Also any transmission feedback is available at this point. The clients can use the preamble for ranging, so they are assumed to know the distance to BS, which is used for association instead of SNR. This allows to run simulation without the explicit modeling of handover procedure. Apart from mentioned service information, downlink is assumed to be filled with data messages which are not of interest.

- Uplink subframe content

The uplink subframe begins with the reservation and initial ranging slots. Those are allocated for network entry and bandwidth requests, and therefore carry no payload. The bandwidth request procedure is assumed to be successful at all times (in reality it is based on random multiple access, but the collision probability is rather low).

The resource slots need to be numbered efficiently. The UL\_MAP message carries the total number of symbols in the UL subframe. The number of frequency subchannels is known from the modulation and coding scheme, which is fixed for a given BS.

The service channels (HARQ, fast-feedback power control etc) are allocated as resource blocks in absolute addressing. Those create sort of a blacklist of slots that can not be used by any data allocations. They can be anywhere as long as they are aligned with the slot boundaries. In the SLS they are assumed to be negligibly small or not used and ignored silently.

Every slot that was not blacklisted is subject to allocation in a manner that suits particular scheduler. In fact, the entire frame can be allocated to a single user if needed.

- Packet fragmentation and packing procedure

The MAC layer tries to balance the frame loss probability with the MAC overhead. It is easiest to mimic ARQ operation with rearrangement. In the SLS, details of

---

MAC overhead are aggregated to a single average in order to preserve simplicity. The packet processing procedure looks as follows

- An application-layer packet is sent to the system, arrival time is recorded.
- The size of the outgoing queue is used for bandwidth requests.
- As soon as burst size is available, the packets in TX queue are fragmented into ARQ blocks.
- For simplicity, an ARQ block always ends on the packet boundary, but never exceeds some predefined maximum size. In reality, the application data is first converted into a sequential stream of bytes that is partitioned into ARQ blocks, therefore they do not have to be aligned. This, however, should not cause any major disturbance in the simulation results.
- All ARQ blocks are transmitted and retransmitted independently. In reality ARQ retransmissions are up to the client, but in our case it important to keep their pattern deterministic. It is assumed that the ARQ fragments are retransmitted in order in which they have been formed. This allows the relays to choose which packet to send on the air when cooperating

There are many more aspects of protocol mapping that have been considered while making the SLS, but their effect is not major. For example, real WIMAX systems have multiple traffic priorities that affect scheduling, but they do not affect the client relay performance directly. Some of the details of the protocol abstraction can be found from the Appendix 6.4, which contains the SLS settings for IEEE 802.16 based network. The following section contains the most representative results along with their analysis and comments.

## 6 Performance evaluation results

### 6.1 Baseline system model

Let us first consider the simple system model to prove that client relay is a logically feasible concept. In order to do so, we need a proper test case. Since we do not intend to consider user placement strategies, a channel property matrix was generated randomly, using the assumption that more transmitters have better chances of delivering the packet successfully. For simplicity we consider 3 client nodes, which are obliged to participate in relaying whenever possible. The maximum amount of retransmissions is 3. The arrivals are providing saturation conditions. Actual channel properties used for example are presented in table 3.

The rest of the simulation parameters are set to the values that correspond to baseline system model. An example of such setup is provided in Appendix 6.4. There are 3 source nodes and 1 BS node in the model. Saturation conditions are assumed on all nodes.

Using either the analytics described in 4.1.1 or simulation it is possible to obtain medium access delay distribution and packet loss probability, as shown in figure 11. In order to compare the scenarios with different packet losses the pmf of access delay is not normalized.

It is possible to calculate or measure directly with simulator the average throughput for selected node, it is approximately 0.171 with  $l_p \approx 0.016$  in cooperative mode and 0.100 with  $l_p \approx 0.240$  in non-cooperative mode. That means almost twice the throughput of non-cooperative mode. Such gain appears mostly due to initially low delivery probability. Essentially, the relaying gain is the bigger the worse initial channel is.

#### 6.1.1 Performance of different schedulers

As a basic practical result, we can estimate the performance of different scheduling algorithms in relay network. As before, we consider synthetic scenario, generated with similar algorithm as before. We study network of 4 nodes, and use proportional fair scheduler, with fairness parameter  $f$ . Actual proportional fair parameters are as follows:  $\alpha = 1 - f$ ,  $\beta = f$ . We change  $f$  in range  $[0..1]$  and also plot a point for round-robin scheduler. We measure cell throughput and plot it against real fairness of the network, as calculated over average throughputs of the nodes. It should be noted that we normalize

Table 3: Delivery probabilities for source groups( $q$ -function)

Sources	1	2	3	1,2	1,3	2,3	1,2,3
Destinations							
1	1.0	0.4	0.3	1.0	1.0	0.2	1.0
2	0.7	1.0	0.7	1.0	0.2	1.0	1.0
3	0.6	0.5	1.0	0.2	1.0	1.0	1.0
BS	0.3	0.4	0.8	0.6	0.9	0.85	0.95

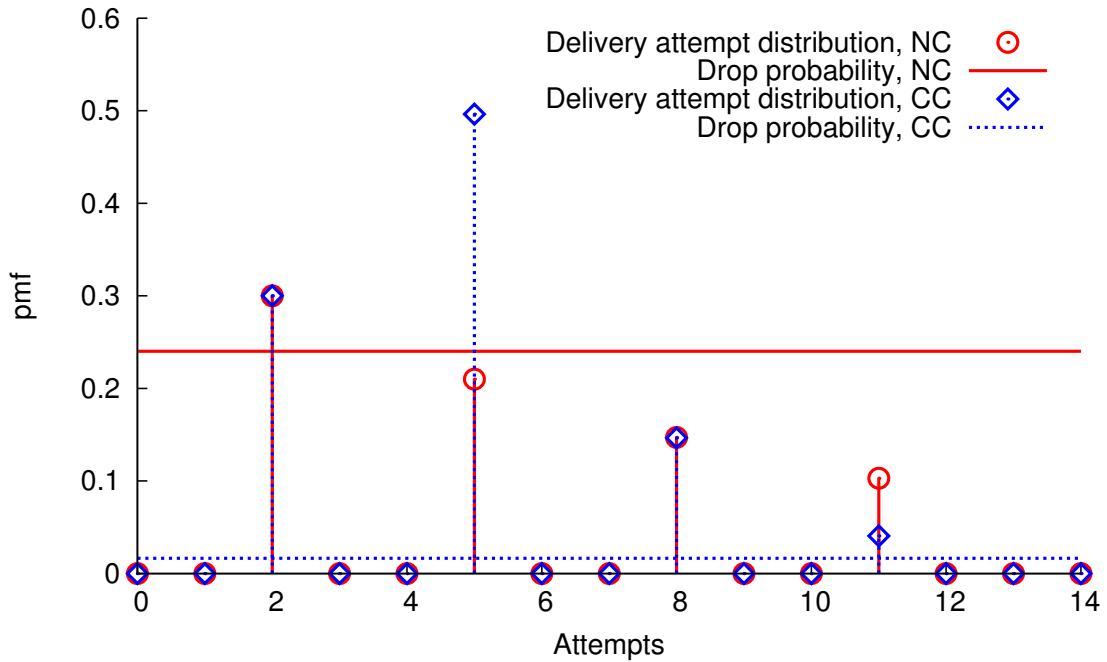


Figure 11: Delay distribution for node 1

Jain's fairness index [34] to fit range [0..1] using formula  $f = \frac{(\sum thr_i)^2}{\sum thr_i^2} - 1 / (N - 1)$ , where  $thr_i$  denote individual throughputs.

As the result, we have obtained the plot presented in figure 12.

It can be noted, that a more primitive round-robin scheduler appears to provide both reasonable throughput and fairness in client relay network, while a proportional fair scheduler, being more complicated to implement, does not really improve the situation when operating in reasonable fairness range. Since the proportional fair scheduler utilizes both network queue state and channel data, it is appears to be somewhat redundant in network with relays. While disregarding the nodes with bad channel, it effectively diminishes possible gains of client relay. At the same time if set to preserve fairness they do not give any real gain over round-robin scheduler.

## 6.2 Extended system model

A complete system model with realistic channel allows us to approach some important questions - how would client relay work in a real network? Would it, for example, work with sectorized antennas any better then with omni-directional ones? Let us consider some scenarios step by step.

## 6.3 Opportunistic cooperation performance

In this section we evaluate the performance of a single wireless cell that operates with an OFDM PHY with only one frequency channel. In order to evaluate the performance

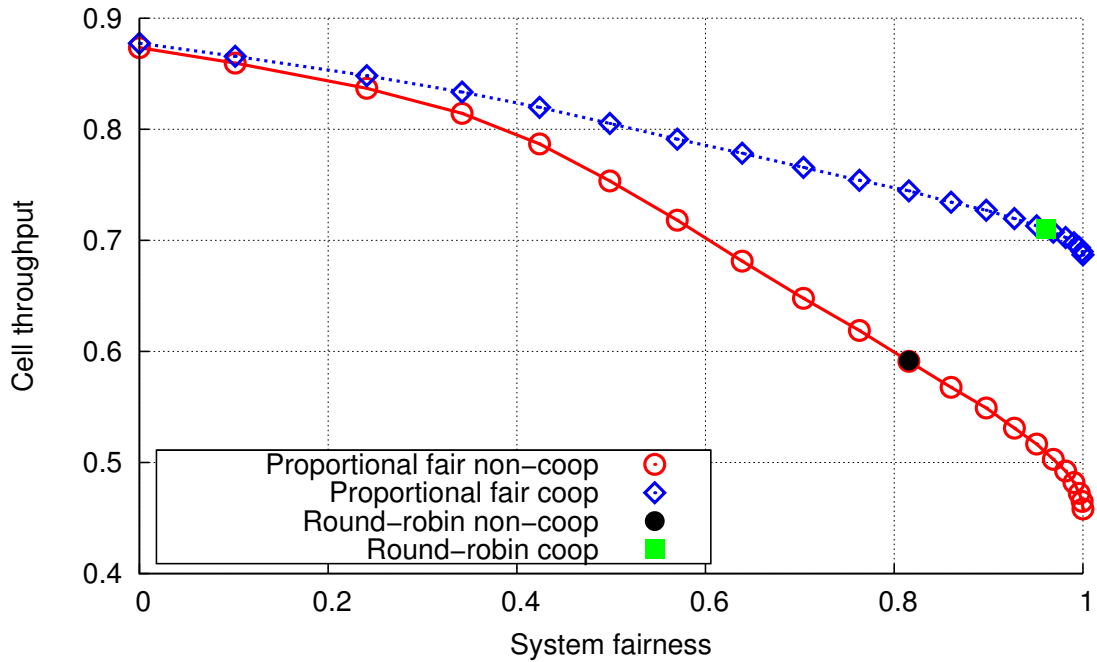


Figure 12: Scheduler performance under different conditions

of a wireless cell properly, many parameters have to be defined. Table 4 summarizes the most important parameters we used for the evaluation of the considered client relay system. Most of them originate from [24] and [29].

For simplicity, it is assumed that all slots have same fixed size, modulation and code rate. This basically means that the *adaptive modulation and coding* (AMC) is disabled. The reason behind such simplification is the conflict between AMC and client relay - AMC reduces the data rate, thus increasing transmission success chances and decreasing the usefulness of relays. Another contradiction comes from the fact that relay energy expense does not depend on the modulation, therefore each relay transmission results in energy efficiency decrease. In order to maintain fairness the BS scheduler would have to allocate more slots to the edge users, therefore the overall cell throughput would decrease. The simulation shows that enabling relay for the AMC case in fact has almost no effect on throughput or delay.

In Figure 13, we observe the dependence of the cell performance on the cooperation probability. As we enable cooperation, the success rate increases. However, at some point, as  $p_{rx}$  values increase, the extra interference generated by the relays starts affecting the transmissions, and the throughput gradually drops. It is possible to locate a point that corresponds to the best balance between the generated interference and the relaying gain. We fix the point where the success rate is maximal, that is, at around  $p_{rx} \cong 0.6$  in our example. The position of this point depends on traffic patterns and intensity, node density, channel quality and other parameters. In general, the curve above may have one or two local maximums depending on the balance between interference affecting



Table 4: Evaluation parameters - opportunistic cooperation

RF part	
Channel bandwidth	20 MHz
Number of data subcarriers	1536 (2048 FFT)
PHY type	OFDM, TDMA
Maximum transmission power	23 dBm
Minimum transmission power	-20 dBm
Modulation scheme	QAM64
FEC encoding	3/4 CC
RX and TX noise figure	-5 dB
Power control target SNR	30 dB
Time slot duration	1 symbol
Cell planning	
Target cell radius $r$	300 m
Reuse scheme	3/1
BS antenna height	35 m (15 m above roof)
MS antenna height	1.5 m
Carrier frequency	2.0 GHz
Users in a cell (mean)	20
Shadowing standard deviation	8 dB
Shadowing correlation	0.5 at 50 m
Scheduling policy	Fair Round-Robin[8]
Node parameters	
Mobility model	Random walk
Mean state time	10 s
Speed range	0.5..1.0 m/s
Node traffic pattern	Saturation
Fragment retry limit	4
Fragment size	860 bytes (1 symbol)

transmissions and relaying gain. In any case, the first peak is preferred due to better transceiver utilization.

In some setups the curve will actually never go up at all, this corresponds to setups in which client relay does not work, for example a very large cell where no relay can be reliably reached. Another possible shape is a logarithm-like curve with clear saturation, which can be observed when interference loopback is either off or does not affect the model due to network planning.

Adaptive run-time estimation of the optimal cooperation probability have proven to be a very challenging task. At the very least, it requires up-to-date information from BS to be broadcast to the relays and accurate estimation of relay position. It also does not cope with non-uniform relay distribution in the cell, so if a massive cluster of nodes is concentrated in a small area, BS has to be aware of that and configure cooperation for those nodes separately.

Now let us consider some important system-level performance parameters for  $p_{rx} \cong$

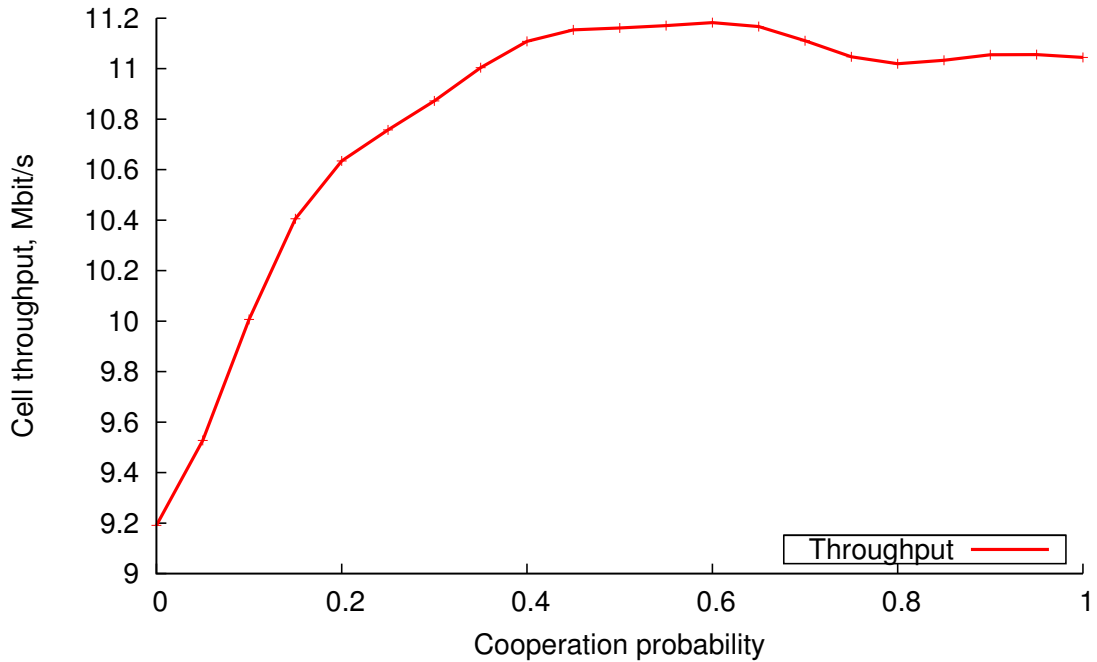


Figure 13: Average uplink throughput (Mbit/s), OFDM

0.6 are provided and compared against the non-cooperative case. In Figures 14 and 15, we confirm that the gain in transmission success rate concerns mostly the nodes at the cell edge, as expected. Also it is important to notice that the transceiver utilization in the cooperative system drops almost by half. Here the transceiver utilization is defined as the ratio between active transmission time that resulted in the own packet delivery over the total amount of time when transceiver was active (retransmitting, listening or relaying). So, in essence, it is correlated with the energy efficiency, but does not depend on the actual device parameters. The data from [35] was used previously, but due to fast progress in receiver architecture the actuality of such information becomes questionable. On the other hand a separate study of the average energy consumption for OFDM transceivers might make up a small project of its own and therefore was not feasible.

The utilization is quite low mostly due to the fact that the packets transmitted by the nodes with relatively good channel conditions are still eavesdropped, which results in unproductive listening. A better heuristic for the relay selection might allow to tackle this problem, but as we will see later on, it is not the only possible approach.

Let us now consider the more advanced case of OFDMA transmission mode with BS assistance for client relay.

## 6.4 BS-assisted cooperation performance

As it was discussed above in sections 4 and 5, BS assistance allows the system to operate in a much more progressive OFDMA mode. This, however, does not change the

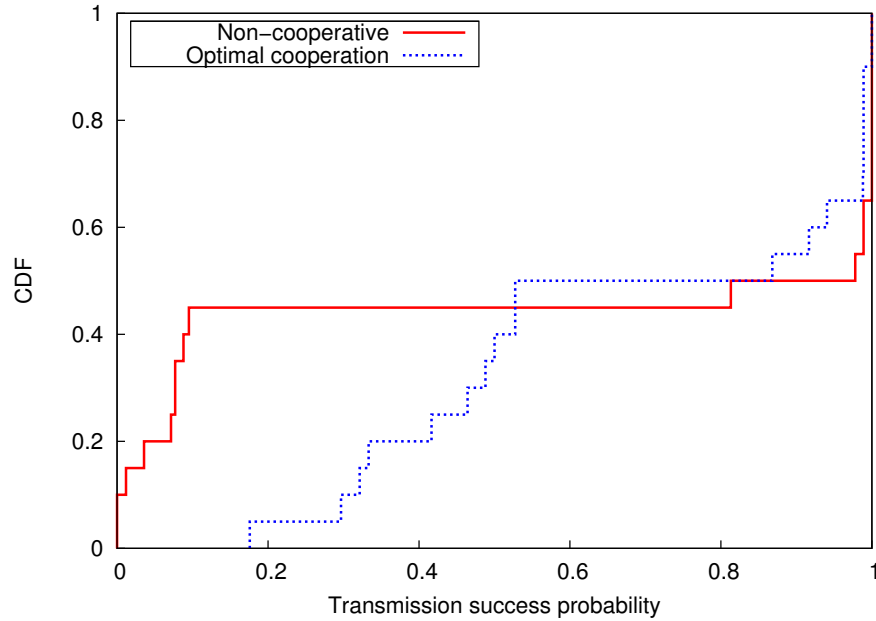


Figure 14: CDF for per-node transmission success probabilities, OFDM

theoretical system capacity, and therefore such system can be directly compared with the above case as long as the basic RF parameters are kept. The Table 5 summarizes the changes in the setup compared to previous scenario.

Table 5: Evaluation parameters - assisted cooperation

Cell planning	
Cell sectors	3 (120 degrees)
Frequency reuse	1/3/3
Scheduling policy	Relay-aware round robin
Near-far separation $\theta$	0.8
PHY type	OFDMA, TDD

As before, in Figure16 we provide the plot for cell throughput. As we can see, it is consistent with the previous result. Of course, due to better utilization and directed antennas, we observe higher overall throughput values. Due to better cell performance, the relaying gain is not that big as in previous case.

Based on the information in Figure17 and Figure18, we conclude that the success rate increase is not so huge as before, but the price in utilization is still high. On the other hand, compared to the 3/1/1 OFDM scenario the overall cell performance is much better.

It should be also noted, that BS-assisted cooperation guarantees that the relays are useful when they transmit. By dividing the nodes based on their channel conditions and not the distance, BS could have gained even more refined control over the network's performance. As we may observe, in the OFDMA mode the system behaves in almost same way as in OFDM, therefore we have achieved our goal of integrating the client relay into OFDMA-based protocol. The proposed scheduling mechanism does not harm

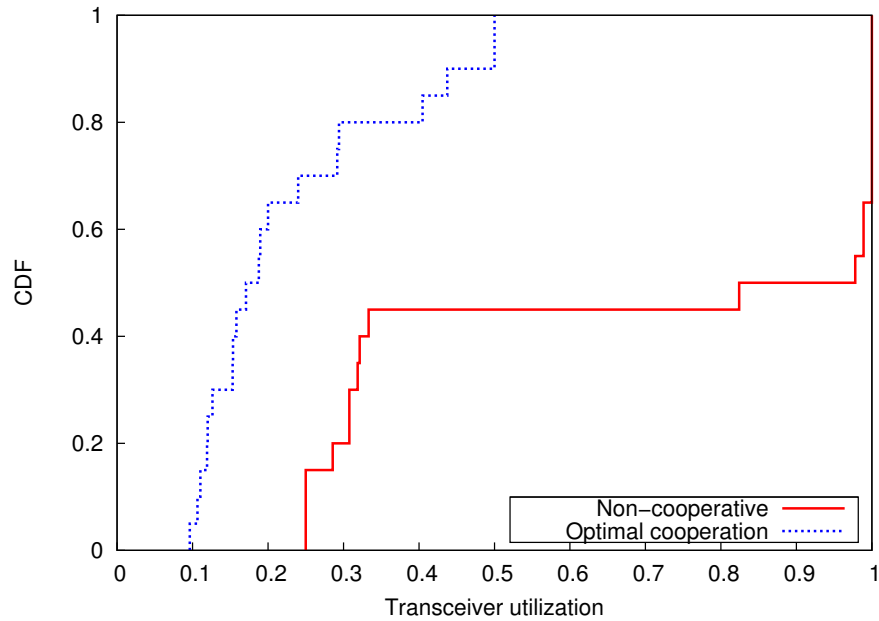


Figure 15: CDF for per-node transceiver utilization, OFDM

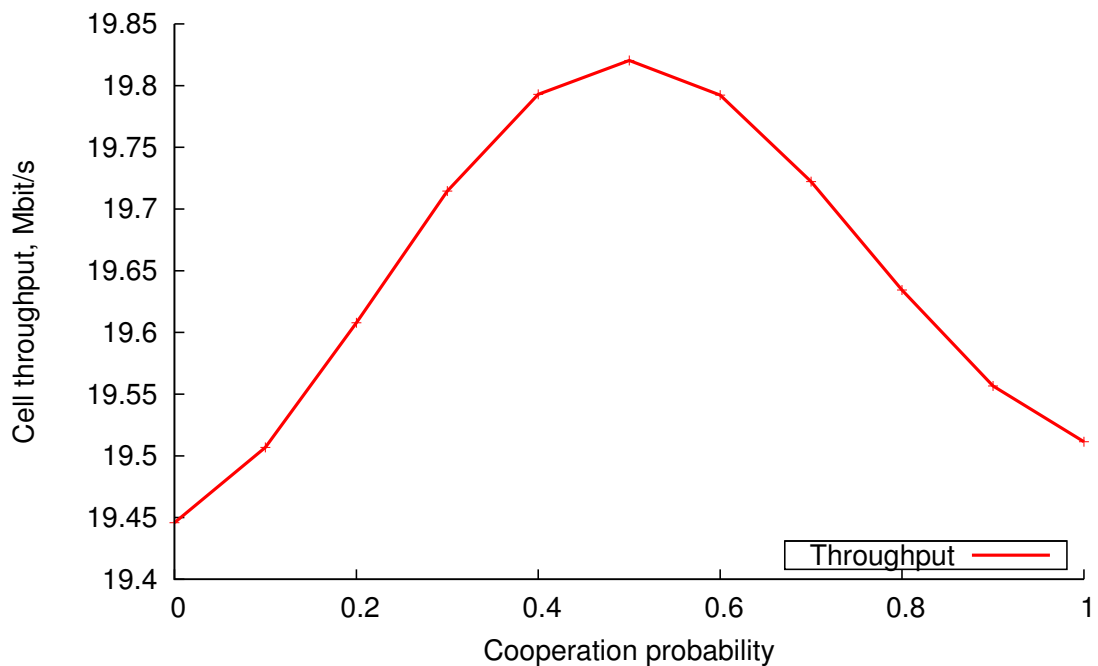


Figure 16: Average uplink throughput (Mbit/s), OFDMA

fairness, and can be employed also in OFDM-based networks in the same way as in OFDMA networks.

Another positive side-effect of the increased transmission success rate is the decrease in medium access delay, which could prove to be useful for time-critical traffic. In general, every retransmission in IEEE 802.16 MAC protocol means at least 5 ms of waiting time,

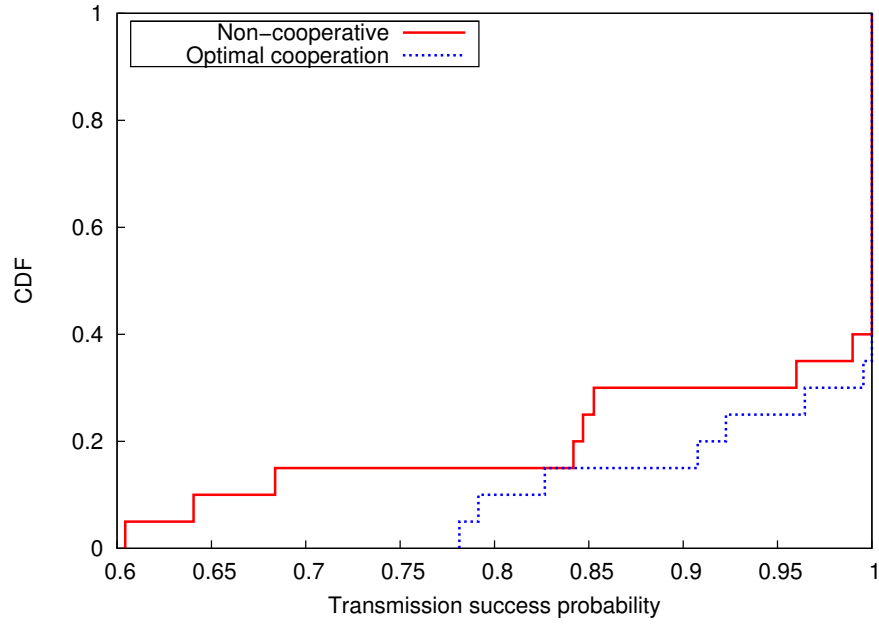


Figure 17: CDF for per-node transmission success probabilities, OFDMA

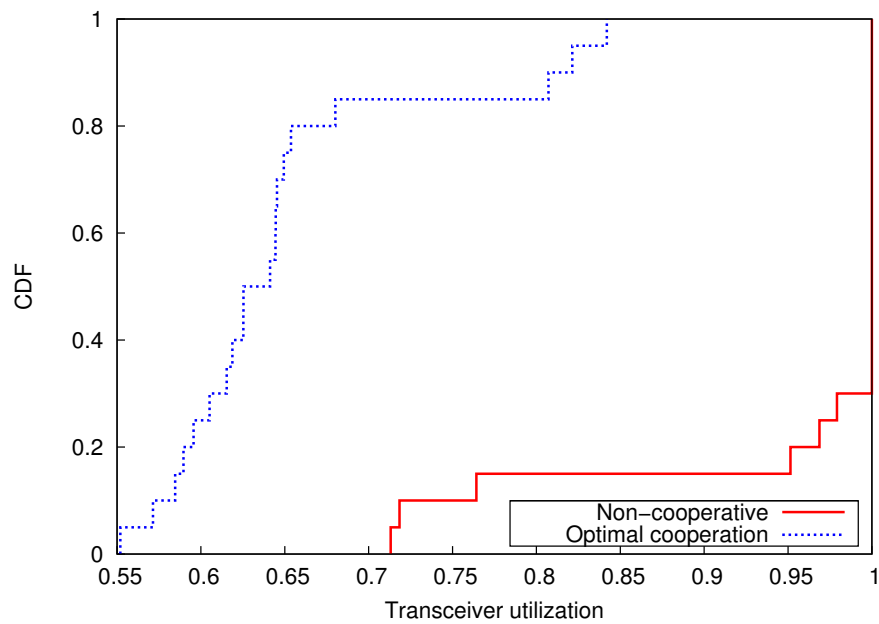


Figure 18: CDF for per-node transceiver utilization, OFDMA

and with 4 retransmissions per fragment that gives up to 20 ms delay on MAC PDU. After that the PDU has to be retransmitted, so the total delay may up to 25 ms and more. Client relay effectively guarantees that the packet will be delivered after second attempt, therefore limiting the delay at 10 ms.

## Conclusion

In conclusion, let us review the most important results of the performed studies.

- It was shown that client relay can give very good performance in form of an effective protocol that does not require signaling to operate.

Such protocol was shown to be feasible without any major modifications to existing systems. The fact that the proposed client relay technique can be implemented within the IEEE 802.16 signaling without significant modification of the protocol proves that it can be done for any reasonable system with scheduling (in other words not based on random multiple access) and OFDM PHY.

- Analytical model for the client relay protocol was derived.

The proposed client relay protocol was found to be very promising for analytic study, while most of the client relay protocols can not be modeled analytically at all. Following the previous publications [8, 9] and the material in this thesis, one may easily obtain most of the performance parameters analytically. This includes average throughput, total delay, energy efficiency, medium access delay distribution and fairness. The analytical models are applicable to both saturated and unsaturated networks. Unlike most similar studies that concentrate on just three nodes, obtained analytical models have been shown to be applicable to cases with any number of nodes.

- Throughput and reliability gains have been observed in system level evaluation.

With smaller cells, gains in throughput and delay performance for the cell-edge users and for the entire cell can be observed. The established gain of 10% cell throughput in saturation is a valuable contribution to the cell throughput, as it is achieved mostly by improving the performance of cell-edge users.

- Client relay was shown to result in improved fairness without sacrificing cell throughput.

The actual improvements, of course, depend on the cell setup and protocol setup, but generally client relay network performs almost as fair as proportional fair scheduling with considerably higher system throughput.

- Client relay energy efficiency impact was estimated.

Typically, enabling relaying causes about 40% decrease in the amount of time the transceivers have to be active. Most of this increase comes from eavesdropping, which in some cases can be almost as expensive as transmission [36]. Obviously, not every client would agree to cooperate on such terms, as it decreases battery

life. As a side-effect of client relay the BS no longer requires to treat the edge-users different when scheduling, as the relay mechanism will compensate for the imbalance in throughput.

- A new SLS concept was developed with specific targeting for relay study on system level.

The new simulation environment has evolved from a rather simple program which was only capable of running baseline system model [7] to a complete SLS that can simulate an IEEE 802.16 network. The new SLS is useful on its own as a basis for further research of cellular networks, even those not involving relaying.

- Certain applicability limits for client relay have been found.

Client relay proves to be a good option in cases when common relays are not suitable for some reason, for example due to cost concerns. In case when installing a static relay is possible, the usefulness of client relay is quite low, since it is only really useful when the base success probability is rather small, around 0.9-0.8, which is not the typical case for cellular systems. Such conditions are usually met only for a small area at the cell edges, as it was shown in section 4. In fact, not in every planning scheme such conditions would exist at all. With reuse factors 1/9 and below enabling client relay does not change the throughput at all, since all the terminals can successfully deliver the packets at most with second attempt.

It can be also noted that with larger cells the client relay would not work as well, mostly because the relay antenna is so close to the ground. The effective relay reception radius is limited around 150 meters for 2 GHz carrier, and if the cell becomes large, the successful eavesdropping probability is nearly zero. This means that most of the listening will be a waste of energy. Even if the relay would capture the packet, the path loss between relay and next interfering BS is not much larger than between originator and the interfering BS, therefore the relay would cause extra interference into the neighboring cell, thus reducing the system capacity.

- A conflict between proportional fair scheduling and adaptive modulation schemes was located.

It appears that client relay contradicts the idea of AMC and fair scheduling. Essentially, if AMC compensates for channel degradation, client relay has no packet loss to begin with. As relay is really useful if it would take more than two attempts to deliver a packet, the AMC makes it very much redundant. Proportional fair schedulers, on the other hand, allocate less resources to the more successful nodes trying to reallocate resource to the cell-edge nodes. This mechanism becomes redundant when such compensation is already done by the relay mechanism.

In spite of all limitations, possible trade-offs between throughput, delay, and energy efficiency make the discussed client relay approach a very promising concept. In some

---

situations it may be necessary, for example, to boost the reliability or coverage of a given system, and in such case client relay comes in handy. It should also be noted that there is a considerable potential for improvement. Since the relay is spatially separated from the originator, they have uncorrelated channels, and MIMO system may be investigated instead of MISO. This is expected to provide substantial gains in the effective SNR [37], allowing the users to transmit simultaneously. This would allow the relay to perform the retransmissions while the originator is already sending the next packet. For such system zero-signaling might not be the best approach anymore, and a simple single-bit feedback as in [20] could be used.

The next step in this approach is using the relay as the only means of reaching BS, instead of transmitting with maximum power initially. The originator may then concentrate on feeding the packets to the network and rely on the cooperative support to get them delivered to the BS. Clearly, care should be taken to make the algorithm adaptive, such that if no relays are available then the node would attempt to use fallback mechanisms.

In the end, many interesting relations have been uncovered, along with new exciting challenges. For instance, optimal relaying strategy is still to be found, along with appropriate handling of HARQ and proper rules for adaptive modulation and coding. And, of course, the end-users have to be convinced to cooperate, which may be an unbreakable obstacle for any kind of client relay. This means, that the network itself should provide considerable motivation for the users to cooperate, at the same time punishing any attempts of exploitation. As a final result of the performed evaluation, I believe that client relay is not a technology that would be helpful tomorrow, but rather will be an integral part of the networks beyond 4'th generation, when the capacity and reliability would cost much more than battery life.



## References

- [1] IEEE. IEEE 802.16: Air Interface for Broadband Wireless Access Systems, 2009.
- [2] *LTE Release 10 & beyond (LTE-Advanced)*.
- [3] Theodore S. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall, 2002.
- [4] G. Song and Y. Li. Asymptotic throughput analysis for channel-aware scheduling. *IEEE Transactions on Communications*, 54:1514 – 1514, 2006.
- [5] Ernesto Zimmermann, Patrick Herhold, and Gerhard Fettweis. On the performance of cooperative relaying protocols in wireless networks. In *European Transactions on Telecommunications Volume 16 Issue 1*, 2004.
- [6] IEEE. IEEE 802.16j : Multihop Relay Specification, 2009.
- [7] Alexander Pyattaev, Sergey Andreev, Alexey Vinel, and Boris Sokolov. Client relay simulation model for centralized wireless networks. In *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*, 2010.
- [8] Alexander Pyattaev, Sergey Andreev, Yevgeni Koucheryavy, and Dmitri Moltchanov. Some modeling approaches for client relay networks. In *In proceedings of IEEE CAMAD workshop*, 2010.
- [9] A. Pyattaev, S. Andreev, O.Galinina, and Y. Koucheryavy. System-level evaluation of opportunistic client cooperation in wireless cellular networks. In *Proceedings of the ICCCN conference*, 2011.
- [10] Zigbee specification. <http://www.zigbee.org/>, Jan 2008.
- [11] IEEE Std 802.11s-2011 Amendment 10: Mesh Networking. 10 2011.
- [12] George Calcev and Jeff Bonta. OFDMA Cellular Networks with Opportunistic Two-Hop Relays. *EURASIP Journal on Wireless Communications and Networking*, 2009.
- [13] Klaus Doppler, Simone Redana, Michal Wodczak, Peter Rost, and Risto Wichman. Dynamic resource assignment and cooperative relaying in cellular networks: Concept and performance assessment. In *EURASIP Journal on Wireless Communications and Networking, Volume 2009:14*, 2009.
- [14] Klaus Doppler, Michal Wodczak, and Peter Rost. On the Integration of Cooperative Relaying into the WINNER System Concept. In *IST2007 - 16th IST Mobile & Wireless Communications Summit*, 2007.

- 
- [15] Carl Wijting, Klaus Doppler, Kari Kalliojärvi, Niklas Johansson, Johan Nystrom, Magnus Olsson, Afif Osseiran, Martin Döttling, Jijun Luo, Tommy Svensson, Mikael Sternad, Gunter Auer, Thierry Lestable, and Stephan Pfletschinger. WINNER II System Concept: Advanced Radio Technologies for Future Wireless Systems. In *In proceedings of IST2007 - 16th IST Mobile & Wireless Communications Summit, 2007*.
- [16] Shahid Mumtaz, Lee Tham Tu, Rasool Sadeghi, and Atilio Gameiro. Relay Implementation in WiMAX System Level Simulator. In *IEEE Transactions on Magnetics*, 2009.
- [17] Kyungmi Park, Hyun S. Ryu, Chung G. Kang, Daeyoung Chang, Seungho Song, Jongguk Ahn, and Jongate Ihm. The Performance of Relay-Enhanced Cellular OFDMA-TDD Network for Mobile Broadband Wireless Services. *EURASIP Journal on Wireless Communications and Networking*, 2009.
- [18] Thomas M. Cover and Young-Han Kim. Capacity of a class of deterministic relay channels. In *IEEE Transactions on Information Theory*, vol. 54, no. 3, 2007.
- [19] Patrick Herhold, Ernesto Zimmermann, and Gerhard Fettweis. A simple cooperative extension to wireless relaying. In *European Transactions on Telecommunications Volume 16 Issue 1*, 2004.
- [20] Ramy Tannious and Aria Nosratinia. Spectrally-efficient relay selection with limited feedback. *IEEE Journal On Selected Areas In Communications*, 26, 2008.
- [21] Beiyu Rong and Anthony Ephremides. On opportunistic cooperation for improving the stability region with multipacket reception. In *Proceedings of the 3rd Euro-NF Conference on Network Control and Optimization*, pages 45–59, 2009.
- [22] A. van Zelst and T. C. W. Schenk. Implementation of a MIMO OFDM based wireless LAN system. *IEEE Transactions on Signal Processing*, 52, 2004.
- [23] Fan Jiang, Hui Tian, and Ping Zhang. A user cooperation simulating strategy based on cooperative game theory in cooperative relay networks. *EURASIP Journal on Wireless Communications and Networking*, 2009.
- [24] Roshni Srinivasan, Jeff Zhuang, Louay Jalloul, Robert Novak, and Jeongho Park. IEEE 802.16m Evaluation Methodology Document (EMD).
- [25] S. Andreev, O. Galinina, and A. Vinel. Performance evaluation of a three node client relay system. *International Journal of Wireless Networks and Broadband Technologies*, 1, 2011.
- [26] OPNET Technologies, Inc. <http://www.opnet.com/>.

- 
- [27] ISI. <http://www.isi.edu/nsnam/ns/>.
- [28] Harry Perros. *Computer Simulation Techniques: The definitive introduction!* Computer Science Department NC State University Raleigh, NC, 2009.
- [29] ITU. Recommendation ITU-R M.1225.
- [30] J. Turkka and M. Renfors. Path loss measurements for a non-line-of-sight mobile-to-mobile environment. In *In proceedings of ITST 2008, 8th International Conference on Telecommunications*, 2008.
- [31] D. J. Young and N. C. Beaulieu. The generation of correlated rayleigh random variates by inverse discrete fourier transform. *IEEE Transactions on Communications*, 48:1114–1127, 2000.
- [32] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronic Letters*, 27:2145–2146, November 1991.
- [33] Arne Simonsson and Anders Furuskär. Uplink power control in lte – overview and performance. In *IEEE VTC conference*, 2008.
- [34] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *Electronic at arXiv*, 9 1998.
- [35] Koen De Turck, Sergey Andreev, Stijn De Vuyst, Dieter Fiems, Sabine Wittevrongel, and Herwig Bruneel. Performance of the iee 802.16e sleep mode mechanism in the presence of bidirectional traffic. In *IEEE ICC Conference*, 2009.
- [36] Sergey Andreev, Yevgeni Koucheryavy, Nageen Himayan, Pavel Gonchukov, and Andrey Turlikov. Active-mode power optimization in ofdma-based wireless networks. In *Proceedings of IEEE GLOBECOM*, pages 799 – 803, 2010.
- [37] David Gesbert, Marios Kountouris, Robert W. Heath Jr., Chan-Byoung Chae, and Thomas Sälzer. From Single User to Multiuser Communications: Shifting the MIMO Paradigm.

## Appendix 1: configuring simulation environment, baseline system model

The OFDM PHY is the closest to the baseline system model. Only one client can transmit at any given time, and the frame structure provides all necessary timetables and feedback to satisfy the assumptions. Therefore, such PHY can be used for verification of the SLS, since corresponding results can be also obtained analytically with reasonable amount of effort. Swapping the channel model to fixed probabilities and setting frame size to 1 slot is all that has to be done to make analytics work.

This appendix presents the configuration parameters for the SLS that have been used to represent the baseline system model. It should be noted that SLS works in TDD mode, that is it actually emulated downlink timings although they do not directly affect the simulation for consistency with real systems. Therefore the downlink duration was set to zero to match the abstraction of baseline model

### Setup for analytical verification

The listing below shows SLS parameters for verification. See also the scenario file after it for complete picture.

```

                                std_const.py
#Frame duration in time units. For real model it would be
  in seconds.
FRAME = 1.0
#total available bandwidth in Hz (affects capacity)
BANDWIDTH=20e6
#FFT size
N_FFT = 2048
#Guard bands are needed for capacity calculation
GUARD_BANDS = (160, 159)
#Used subcarriers
N_USED = N_FFT - sum(GUARD_BANDS)

#Noise model is not used as we use static delivery
  probabilities
NOISE_FLOOR = 0.0

#OFDM cyclic prefix ratio
CYCLIC_PREFIX = 0.0

```

*#The length of a single physical slot – the smallest time unit measurable by terminal. See als tick.*

PS=FRAME

*#Duplexing time (receive to transmit)*

RTG\_PS = 0

*#Duplexing time (transmit to receive)*

TTG\_PS = 0

*#the duration of a single symbol (without guard prefix)*

USEFUL\_SYMBOL\_DURATION = 1.0

*#and with guard prefix*

SYMBOL\_DURATION = USEFUL\_SYMBOL\_DURATION  
\*(1.0+CYCLIC\_PREFIX)

*#and same in PS*

SYMBOL\_DURATION\_PS = int(round(SYMBOL\_DURATION / PS))

*#Number of phy slots in one frame*

FRAME\_PS=int(round(FRAME/PS))

*#number of usable PS's*

USEFUL\_FRAME\_PS = FRAME\_PS – RTG\_PS – TTG\_PS

*#number of OFDM syms per frame*

SYMBOLS\_PER\_FRAME = 1

*#ratio of uplink to downlink symbols(only uplink)*

UL\_DL\_RATIO = 1.0

(UL\_SYMBOLS, DL\_SYMBOLS) = (1,0)

*#Frame starts with uplink subframe*

(UL\_START, UL\_START\_PS) = (0,0)

*#Downlink timings*

*#Disable preamble*

PREAMBLE\_PS = 0

DL\_START\_PS = PREAMBLE\_PS

---

```
#####resource allocations for OFDM
#subchannel bandwidth
SUBCHANNEL_CARRIERS = N_USED
#No pilots
SUBCHANNEL_PILOTS = 0
#slot size (symbols, subchannels)
SLOT_SIZE = (1, 1)

#UL dimensions
(UL_SLOTS_TIME, UL_SLOTS_FREQ, UL_SLOTS_TOTAL) = (1,1,1)

#real slot size as seen by PHY
SLOT_SIZE_PHY = (1,1)

#Slot data capacity (that is how many data syms are there)
SLOT_CAPACITY = 1000
#####SLS parameters

#how many bytes does MAC header take for each packet
MAC_OVERHEAD = 0
#The max size of ARQ block in bytes
MAX_ARQ_BLOCK = 1
MAX_ARQ_RETRIES = 4

#The most important parameter – the system clock tick in
seconds.
#System clock should be capable of specifying any
scheduled event
#as a multiple of TICK. Accuracy of traffic generators
depends on
#the granularity of TICK variable. All times inside
simulation are
#integer multiples of TICK, so trafgen code relies on it
being small enough
TICK = PS/100

#Power expense levels
CONSUMPTION_TX = 1.0
CONSUMPTION_RX = 0.7
CONSUMPTION_ID = 0.1
```

## The simulation scenario for analytical verification

The scenario file creates the objects in the network, sets up traffic patterns and other relations. The main execute function is supplied with extra options specified over command line.

scenario.py

```

#=====
# Model block import
#this is important! do not modify this section!!!
#=====
#Node models
import node
from mobile_node import *
import schedule
#Traffic generation
import trafgen
#Node placement etc
from utils import *
from scenario_utils import *
#Debug
from debug import *
#Global state variables
from gl_vars import gl
#Channel model abstraction
import channel_model
#Statistics
from statistics import *
#The SLS config from the file above
import std_const

#System libs
import argparse

# The main simulation script that is run to set up the
  environment in a
#way similar to NS

def execute(sim, opt):
    #=====Define model global variables

```

```

info("Scenario_options:"+opt)
#Create static channel based on CSV file
gl.chan =
    channel_model.TStaticChannel("conf/channel.csv")
sched = gl.sched
#=====
#Below you can define the model scenario
parser = argparse.ArgumentParser(
    description='Scenario_options')
parser.add_argument('--load_a', default=1.0, type=float,
    help='Node_A_load')
parser.add_argument('--load_r', default=0.0, type=float,
    help='Node_A_load')
parser.add_argument('--relay_prob', default=0.0, type=float,
    help='Node_R_relaying_probability')
args=parser.parse_args(opt.split())
params = node.node_params()
#remove relay session limit
params.max_relay_sessions = 100
#Disallow full-duplex operation
params.opt_duplex = True
params.relay_probability = 0.0
params.listen_probability = 0.0

#Create nodes
id=0
n = node.TClientNode(id, params)
sim.user_list.append(n)
trafgen.TPoissonGen(n, lambd =
    args.load_a/std_const.FRAME, packet_size = 1)

id+=1
params.relay_probability = 1.0
params.listen_probability = args.relay_prob
n = node.TClientNode(id, params)
sim.user_list.append(n)
trafgen.TPoissonGen(n, lambd =
    args.load_r/std_const.FRAME, packet_size = 1)

cell_id=0

```



```

id +=1
n=node.TBSNode(id , cell_id ,params)
#attach a scheduler to the BS
schedule.TRoundRobinSchedule(n)
sim.bs_list.append(n)

#set up statistics collection
gl.stats.fragments_delivered = TAccTraceVar()
gl.stats.fragments_retransmitted = TAccTraceVar()
gl.stats.fragments_transmitted = TAccTraceVar()
#Arrange for a histogram collection for packet delay
gl.stats.packet_delay.hist = THistogram(0,
    100*std_const.FRAME_PS, std_const.FRAME_PS)
#Same for ARQ fragments (this can be used for
    analytical verification)
gl.stats.fragment_delay.hist = THistogram(0,
    100*std_const.FRAME_PS, std_const.FRAME_PS)

#Set the replication cycle length to 5000 frames
replication_every( std_const.FRAME_PS * 5000)
#stop the simulation after 50000 frames
sched.enterabs( std_const.FRAME_PS * 50000, 1000,
    sim.stop)

```

The above scenario would create a network of 3 nodes, one of which is a relay with opportunistic listening. The model would run 10 replication cycles of 5000 frames each, and report mean and variance for delivered, transmitted and retransmitted ARQ fragments, medium access delay and packet delivery delay, and also delay histograms. As the model is run several times with different options, this results in large amount of output data. The output is therefore structured in a Matlab-usable code that is used instead of classic trace file. As a result the model output can be directly used for analysis without any further processing.

The multiple point iteration mentioned above is performed by a special Octave script based on the configuration file. The parameters specified here are fed directly to the scenario program which creates the simulation scenario according to those.

scenario.py

```

t.path='load_a';
t.fmt='float';
t.start=0;
t.stop=1.0;
t.step=0.05;

```

```
vals { i++ } = t ;
```

```
t . path = ' relay_prob ' ;
```

```
t . fmt = ' float ' ;
```

```
t . start = 1.0 ;
```

```
t . stop = 0.0 ;
```

```
t . step = 1.0 ;
```

```
vals { i++ } = t ;
```

## Appendix 2: configuring simulation environment, IEEE 802.16

IEEE 802.16 networks use OFDMA as their PHY. Therefore, multiple users can be transmitting at the same time. The SLS takes care of that as it reconstructs the entire frame structure of 802.16 with reasonable accuracy. This appendix presents the configuration parameters for the SLS that have been used for the OFDMA studies.

### SLS setup for IEEE 802.16 OFDMA PHY

The listing below shows SLS parameters for verification. See also the scenario file after it for complete picture.

```

                                std_const.py

#Duration of a single frame in seconds
 #(as in 802.16 ch. "8.4.5.2 Frame duration codes")
#This also works in OFDM mode
FRAME = 0.005

#central carrier
CARRIER=2.1e9

#total available bandwidth in Hz
BANDWIDTH=20e6

#OFDMA sampling factor is a function of bandwidth in 802.16
SAMPLING_FACTOR=__samp_factor(BANDWIDTH)

#OFDM sampling freq
F_SAMPLING = int(SAMPLING_FACTOR*BANDWIDTH/8000.0)*8000

#FFT size
N_FFT = 2048

#generator function: (2^(7+x) - 20*2^x)
GUARD_BANDS = (160, 159+1)

#Used subcarriers
N_USED = N_FFT - sum(GUARD_BANDS)

SUBCARRIER_SPACING = F_SAMPLING/N_FFT

```

---

```

#OFDM cyclic prefix ratio
CYCLIC_PREFIX = 1.0/8.0

#The length of a single physical slot in seconds
PS=4.0/F_SAMPLING
#Duplexing time (receive to transmit)
RTG_PS = 168
#Duplexing time (transmit to receive)
TTG_PS = 296

#the duration of a single symbol (without guard prefix)
USEFUL_SYMBOL_DURATION = 1.0/SUBCARRIER_SPACING
#and with guard prefix
SYMBOL_DURATION = USEFUL_SYMBOL_DURATION
                    *(1.0+CYCLIC_PREFIX)
#and same in PS
SYMBOL_DURATION_PS = int(round(SYMBOL_DURATION / PS))
#Number of phy slots in one frame
FRAME_PS=int(round(FRAME/PS))
#number of usable PS's
USEFUL_FRAME_PS = FRAME_PS - RTG_PS - TTG_PS
#number of OFDM syms per frame
SYMBOLS_PER_FRAME = int(USEFUL_FRAME_PS /
                        SYMBOL_DURATION_PS)
#Downlink timings just for fun
PREAMBLE_PS = SYMBOL_DURATION_PS
DL_START_PS = PREAMBLE_PS
#ratio of uplink to downlink symbols
UL_DL_RATIO = 0.3

def get_ul_dl_syms(ul_dl_ratio):
    #Number of symbols in UL and DL
    ul_syms = int((SYMBOLS_PER_FRAME-1)* ul_dl_ratio)
    dl_syms = SYMBOLS_PER_FRAME - ul_syms
    return (ul_syms , dl_syms)

(UL_SYMBOLS, DL_SYMBOLS)= get_ul_dl_syms(UL_DL_RATIO)

#####Timings within frame

```

---

```

def get_ul_start(dl_syms):
    #the beginning of UL interval
    ul_start = (dl_syms * SYMBOL_DURATION + TTG_PS*PS)
    ul_start_ps = dl_syms * SYMBOL_DURATION_PS + TTG_PS
    return (ul_start , ul_start_ps)

(UL_START, UL_START_PS) = get_ul_start(DL_SYMBOLS)

#=====resource allocations for OFDMA
#OFDMA mode subchannel bandwidth. AMC-mode.
SUBCHANNEL_CARRIERS = 9
#There is only one pilot in AMC mode, that is in the
middle of subchannel
SUBCHANNEL_PILOTS = 1
#slot size (symbols, subchannels)
SLOT_SIZE = (1, 6)

SLOTS_FREQ = int(N_USED /
    (SLOT_SIZE[1]*SUBCHANNEL_CARRIERS))
def get_ul_dimensions(ul_syms):
    #dimensions of UL subframe
    slots_time = int( ul_syms / SLOT_SIZE[0])
    slots_total = slots_time*SLOTS_FREQ
    return (slots_time ,SLOTS_FREQ,  slots_total)

(UL_SLOTS_TIME,  UL_SLOTS_FREQ,  UL_SLOTS_TOTAL) =
    get_ul_dimensions(UL_SYMBOLS)

#real slot size as seen by PHY
SLOT_SIZE_PHY = (SLOT_SIZE[0] * SYMBOL_DURATION_PS,
    SLOT_SIZE[1]*SUBCHANNEL_CARRIERS)
#Slot data capacity (that is how many data syms are there)
SLOT_CAPACITY =
    SLOT_SIZE[0]*SLOT_SIZE[1]*(SUBCHANNEL_CARRIERS -
    SUBCHANNEL_PILOTS)

SUBCHANNEL_NOISE_FLOOR = 10*log10(4*
    1.38e-23*290*SUBCARRIER_SPACING*SLOT_SIZE_PHY[1])
NOISE_FLOOR = SUBCHANNEL_NOISE_FLOOR+10*log10(SLOTS_FREQ)
#=====SLS parameters

```

---

```

#the size of the simulation landscape in meters (measured
  from center point)
GRID_SIZE=3000

#how many bytes does MAC header take for each packet
MAC_OVERHEAD = 10
#The max size of ARQ block in bytes
MAX_ARQ_BLOCK = 2**10
MAX_ARQ_RETRIES = 4
#Retransmissions always ahead of data?
RETRANSMIT_PRIORITY = False
#Target SNR for power control. If this can not be met,
  code rate is changed.
TARGET_SNR = 35.0
#Target BER for code selection
TARGET_BER = 1e-6

#The most important parameter – the system clock tick in
  seconds.
#System clock should be capable of specifying any
  scheduled event
#as a multiple of TICK. For WiMAX PS is a good tick value.
TICK = PS

#Power levels
RX_SENSITIVITY = -130.0
TX_POWER = (-40, 23.0)

```

## Scenario setup for IEEE 802.16 network

scenario.py

```

#=====
# Model block import
#this is important! do not modify this section!!!
#=====
#Node models
import node
from mobile_node import *
import schedule

```

---

```

#Traffic generation
import trafgen
#Node placement etc
from utils import *
from scenario_utils import *
#Debug
from debug import *
#Global state variables
from gl_vars import gl
#Channel model abstraction
import channel_model
#Statistics
from statistics import *
#The SLS config from the file above
import std_const

#System libs
import argparse
#Everything that has to do with PHY abstraction
import ofdm

# The main simulation script that is run to set up the
environment in a
#way similar to NS
def execute(sim, opt):
    #####Define model global variables
    ch_param = ofdm.chan_params_t()
    pathloss_ms_bs =
        ofdm.load_func("pathloss/pathloss_itu.py_result")
    ch_param.pathloss_ms_bs= pathloss_ms_bs
    ch_param.pathloss_ms_ms =
        ofdm.load_func("pathloss/pathloss_renfors.py_result")
    #That is 4*k*T*B, where B is bandwidth of the channel.
    ch_param.noise_floor = std_const.NOISE_FLOOR
    #The power level in subchannel after which we do not
care if there even was something
    ch_param.sensitivity = std_const.NOISE_FLOOR -10.0

    ch_param.max_subchannels = std_const.SLOTS_FREQ
    #Set up fading function (Rayleigh)

```

```

fading_func =
    ofdm.load_func("rayleigh/rayleigh.m_result")
ch_param.fastfading_func = fading_func
#The PHY simulation step size – this is the sampling
    frequency in essence
ch_param.step = int(std_const.SYMBOL_DURATION_PS/4)
#Create OFDMA channel object
gl.chan = channel_model.TOFDMAChannel(ch_param)
sched = gl.sched
#=====
#Below you can define the model scenario
parser = argparse.ArgumentParser(
    description='Scenario_options')
parser.add_argument('--theta', default=0.9, type=float,
    help='Node_load')
parser.add_argument('--nodes_cnt', default=2, type=int,
    help='Node_count')
parser.add_argument('--relay_prob', default=1.0, type=float,
    help='Node_relaying_probability')
parser.add_argument('--radius', default=300.0, type=float,
    help='Cell_radius')
parser.add_argument('--assist_thr', default=20.0, type=float,
    help='assistance_threshold')
parser.add_argument('--assist_range', default=300.0, type=float,
    help='assistance_max_distance')
args=parser.parse_args(opt.split())

R=args.radius
ms_ant_pattern =
    ofdm.load_func("ant_patterns/omni.m_result")
bs_ant_pattern =
    ofdm.load_func("ant_patterns/emd.m_result")

#Basic node params
params = node.node_params()
params.max_tx_qlen = 50
params.max_relay_sessions = 10000
params.speed = 0.0
params.h=1.5
params.show_label=True

```



```

params.assist_threshold = args.assist_thr
params.assist_range = args.assist_range
params.relay_probability = 1.0
params.listen_probability = args.relay_prob
params.power_control_decay = False#args.relay_prob>0
params.opt_duplex = True
params.near_far_enable = True

#Place some static nodes inside the cell
for id in range(args.nodes_cnt):
    zz, zzz, a = hexgrid(0, 0, 0, R)
    params.x, params.y = pos_sector(0, 0, 10.0, R, a)
    n = node.TClientNode(id, params)
    n.antenna = mk_ant(ms_ant_pattern, ofdm.NODE_MS,
        0.0)
    sim.user_list.append(n)
    #Attach saturation traffic generator to them
    trafgen.TSaturationGen(n)

id+=1
params.x=0
params.y=0
params.h=15
cell_id=0
n=node.TBSNode(id, cell_id, params)
zz, zzz, a = hexgrid(0, 0, 0, R)
n.antenna = mk_ant(bs_ant_pattern, ofdm.NODE_BS, a)
n.antenna.node_type = ofdm.NODE_BS

#Create relay-aware scheduler
scparams=schedule.schedule_params()
scparams.near_far_threshold = R*args.theta
schedule.TRelaySchedule(n, scparams)
sim.bs_list.append(n)

#listener nodes. Those have same frequency as original
BS
pos = ((-1, -1), (-1, 0), (0, -1), (0, 1), (1, -1),
    (1, 0))
for p in pos:

```

```

    id+=1
    params.x, params.y, a = hexgrid(p[0], p[1], 0, R)
    n = node.TMeasNode(id, cell_id, params)
    n.antenna = mk_ant(bs_ant_pattern, ofdm.NODE_BS, a)
    sim.special_list.append(n)

gl.stats.data_bytes_delivered = TAccTraceVar()
gl.stats.fragments_delivered = TAccTraceVar()
#add replications
replication_every(std_const.FRAME_PS * 500)
#Add interference loopback
if_loopback_every(std_const.FRAME_PS * 5)
#stop the simulation
sched.enterabs(std_const.FRAME_PS * 2500, 1000,
    sim.stop)

```

The above scenario would create a network of **nodes\_cnt** nodes, each of which is a relay opportunistic listening. The OFDMA channel model would be used, along with real IEEE 802.16 framing. The simulation includes also 6 measurement nodes for interference loopback function. As before, replications would be used to control accuracy.