

IST-4-027310 MEMBRANE

Deliverable D5.2.3

MEMBRANE Prototype Evaluation

Contractual Date of Delivery to the CEC:	June 30 th , 2008
Actual Date of Delivery to the CEC:	September 30 th , 2008
Author(s):	A. Valkanas ¹ , K. Avgeropoulos ¹ , Ch. Antonopoulos ¹ , K. Stamatis ¹ , A. Maltsev ² , R. Maslennikov ² , A. Khoryaev ² , A. Alexiou ³ , K. Leung ⁴
Participant(s):	ICOM ¹ , INTEL ² , LUCENT ³ , IMPERIAL ⁴
Workpackage:	WP5.2
Est. person months:	9
Security:	Public
Nature:	R
Version:	1.0
Total number of pages:	51

Abstract:

This deliverable concludes the work carried out within WP5.2 of MEMBRANE which main aim was to develop and implement several novel algorithms and protocols into a hardware platform. This implementation serves also to prove the viability of the solutions investigated in the project and proposed for wireless meshed backhaul networks. The performance evaluation results obtained and implementation complexity issues faced during the development of the prototypes are presented. From various configurations of the demonstrators that correspond to relevant MEMBRANE scenarios, the performances and a proof-of-concept of the techniques implemented are drawn. The results obtained are compared with simulations achieved in WP5.1 in order to validate the prototypes setups and outcomes. Finally, jointly with the complexity analysis, some initial recommendations for the development and implementation of modules for the MEMBRANE meshed network nodes can be deduced.

Keyword list: Prototype, Platform, Demonstrator, Implementation, Complexity, Performances

EXECUTIVE SUMMARY

The purpose of the MEMBRANE project has been to investigate new concepts and technologies that serve to optimize wireless broadband multi-hop backhaul networks. Using a cross-layer approach spanning throughout the lower layers of the telecom system and in particular for the physical to the transport layer serious enhancements in performances can be achieved and optimal solutions can be reached. Methods investigated concern the exploitation of multi-hop meshed deployment, intelligent antenna techniques based on advanced multi-element processing algorithms and opportunistic jointly optimized routing, scheduling and power control protocols. The amalgam of these technologies boosts the efficiency of the MEMBRANE target system to its highest limits. The evaluation and validation of the MEMBRANE system has been driven from the development of a twofold performance evaluation platform. . Beside the development of a high layer system simulation platform, the functionalities offered by the MEMBRANE are validated in commercial hardware platforms. The MEMBRANE demonstration activity aims at proving the performances, economical, and implementation viability of selected techniques of the project.

From both the natural complexity of the design of an advanced wireless backhaul network and the cross-layering optimization approach, the techniques of the MEMBRANE affect several components of the end-to-end transmission chain. To cover the maximum of the methods used to jointly overcome the wireless propagation effect and maintain high capacity provision, our initial prototyping plan has been adapted. Basically, the MEMBRANE prototyping activity has been upgraded with the parallel development of two demonstration prototypes. This upgrade has not only been quantitative but also qualitative since provision of proof-of-concept involved extra developments in high layers and affected a wide range of components blocks. The outcome has been a better understanding of the capabilities of the MEMBRANE network.

Hence, the demonstration through prototyping of MEMBRANE concerns two parts. At first, the advantages of advanced PHY-based techniques that compensate the drawbacks from the wireless environment must be demonstrated. On the other hand, maintenance of the extremely high requirements in terms of “local” and end-to-end throughput is another goal of the prototype. To this end, two algorithms have been selected and implemented in two distinct commercial custom-made demonstration platforms. The two algorithms and the respective hardware platform used for their evaluation are briefly described in section. In order to provide a valid proof-of-concept of the prototyping activity, an evaluation framework has been established. This corresponds to the definition, on one hand, of specific configuration of interest that will raise the advantages of the implemented algorithms. On the other hand, the metrics and measurements needed to fully understand the implementation issues and performances of the techniques prototypes are identified and presented.

This document mainly contains the results of a measurement campaign on the two prototypes after the conclusion of the implementation of the chosen algorithms and protocols. The performances obtained are used to validate the operational mode(s) of the techniques and identify practical issues faced during their implementation. Hence, an analysis is carried out and presented for both prototypes in order to deduce the implementation complexity of the algorithms. In particular, the impact of the necessary simplifications carried out during the integration of these techniques is assessed. Also, the developments of the demonstrators are validated through cross-comparison with results achieved through simulations.

Finally, this deliverable concludes by providing to the reader several recommendations for the implementation of the two selected MEMBRANE techniques into nodes of the wireless backhaul network. The design, development, integration and implementation of the two hardware prototypes have been successful. Hence, the MEMBRANE prototyping activity can be subject to further exploitation and extension either within other European research projects or by similar internal industrial activities.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	2
LIST OF FIGURES	4
LIST OF TABLES	4
1 INTRODUCTION	5
2 PROTOTYPE IMPLEMENTATIONS	6
2.1 Novel Algorithms & Protocols	6
2.1.1 The Data Splitting Algorithm	6
2.1.2 The MEMBRANE Distributed Scheduler	7
2.2 Implementations	9
2.2.1 The PHY Demonstrator	9
2.2.2 The MAC Demonstrator	11
2.3 Configurations	12
2.3.1 Centralized Demonstrator Scenarios with DSA and PHY Demonstrator Setup	13
2.3.2 De-centralized Demonstrator Scenarios with MDS and MAC Demonstrator Setup	15
2.4 Performance Measurements	16
2.4.1 PHY Layer Performance Metrics	17
2.4.2 MAC Layer Performance Metrics	18
3 PERFORMANCE EVALUATION AND COMPLEXITY ANALYSIS	21
3.1 Performance Evaluation	21
3.1.1 PHY Demonstrator Performances	21
3.1.2 MAC Demonstrator Performances	25
3.1.3 Comparison with Simulations and Performance Conclusions	31
3.1.3.1 PHY demonstrator	31
3.1.3.2 MAC demonstrator	33
3.2 Implementation Complexity	34
3.2.1 PHY Demonstrator Implementation	34
3.2.2 MAC Demonstrator Implementation	36
3.2.2.1 Implementation Components	36
3.2.2.2 Device Driver	39
3.2.2.2.1 Handling a packet received from upper layers	40
3.2.2.2.2 Handling a packet received from the PHY (i.e. network card interface)	41
3.2.2.2.3 Handling a packet in a data phase	42
3.2.2.2.4 Control packet creation	43
3.2.2.2.5 Driver initialization	44
4 RECOMMENDATIONS FOR THE DESIGN OF MEMBRANE NODES	45
4.1 MEMBRANE System using DSA	45
4.2 MEMBRANE System using MDS	46
5 CONCLUSIONS	48
TERMS AND ACRONYMS	49
REFERENCES	51

LIST OF FIGURES

Figure 1: DSA Transmission Modes for PHY Demonstrator	7
Figure 2: MDS Algorithm Frame Structure	8
Figure 3: Block diagram of MEMBRANE PHY demonstrator	9
Figure 4: Photo of the MEMBRANE PHY Demonstrator	10
Figure 5: Signal processing blocks of HLLC	11
Figure 6: MEMBRANE MAC Demonstration Hardware Configuration (Phase 1 and 2)	12
Figure 7: GUI parameters of PHY Demonstrator	14
Figure 8: Three-node topology	16
Figure 9: The effect of control packet size and number of stations on average mini-slot duration	18
Figure 10: The effect of control packet size and number of stations on control phase duration	19
Figure 11: The effect of control packet size and number of stations on Control/Data phase ratio	19
Figure 12: High SNR and Blocked Direct Path	22
Figure 13: Low SNR and Blocked Direct Path	23
Figure 14: High SNR and 3 dB loss in Direct Path	24
Figure 15: Low SNR and 3 dB Loss in Direct Path	25
Figure 16: Ping Application Performance Evaluation	26
Figure 17: Sing flow Scenario Performance Evaluation	27
Figure 18: Sing flow Scenario Fair Channel Distribution Evaluation	27
Figure 19: Two Concurrent Crossed flow Scenario Performance Evaluation	28
Figure 20: Two Concurrent Crossed Scenario Fair Channel Distribution Evaluation	29
Figure 21: Ping application considering 3 station setup	29
Figure 22: UDP application performance for 2 and 3 station setup	30
Figure 23: Station addition overhead effect estimation considering TCP applications	31
Figure 24: BER results for HLLC and reference floating point model in AWGN channel	32
Figure 25: BER curves obtained with the HLLC	33
Figure 26: NS-2 Throughput Results of MDS	34
Figure 27: Architecture of hardware link level emulator	35
Figure 28: Modified network stack incorporating MDS algorithm & respective packet framework	36
Figure 29: Primary connection forming requirement by IEEE 802.16e based platform	37
Figure 30: MEMBRANE IEEE 802.16e connection conceptual diagram	38
Figure 31: Communication links between stations	38
Figure 32: Correlation of communication links to PMP phases	38
Figure 33: Procedure followed when a packet is received from upper layers to be transmitted	40
Figure 34: Procedures followed when a packet is received from network interface card	41
Figure 35: Procedure to be followed when data phase is triggered	42
Figure 36: Procedure to be followed when control mini-slot is triggered	43
Figure 37: Additional checks to be made during network driver initialization	44

LIST OF TABLES

Table 1: PHY Demonstrator Related Parameters	13
Table 2: PHY Demonstrator Scenarios	15
Table 3: MAC Demonstrator Packet Header Format	15
Table 4: MAC Demonstrator Control Payload Format	15
Table 5: MAC demonstrator transmission and loss parameters	16
Table 6: Evaluation Setup for the MDS	18
Table 7: Characteristics of FPGA Implementation of HLLC	35

1 INTRODUCTION

The MEMBRANE project investigates novel techniques that aim at improving the performances and capabilities of wireless multihop backhaul networks. Through complex multi-antenna algorithms and advanced resource allocation protocols, the MEMBRANE system must be able to compensate the wireless propagation medium deteriorating effects to the transmitted signals and maintain high throughput capabilities. Within the project, several techniques have been investigated to this end. However, in order to properly evaluate them and validate their economical viability in future backhaul networks, a prototyping activity has been carried out. Modules capable to support advanced investigated features have been implemented in commercial hardware platforms configured to satisfy the high demands of wireless backhaul networks based on MEMBRANE technology. In order to cover most of the solutions proposed within MEMBRANE, we carried out prototyping of two distinct algorithms that aim to demonstrate two different aspects of the wireless backhaul network. On one hand, the effects generated by the propagation conditions are counteracted with the advanced techniques proposed. On the other hand, the highly demanding capacity required for backhaul networking is satisfied by judicious usage of novel protocol procedures. These two distinct features and algorithms have been implemented into two separate hardware demonstrators. Hence, the proof-of-concept of the viability of these algorithms is demonstrated while their complexity in implementation is also deduced.

This deliverable concludes the prototyping activity and contains the performance results and complexity analysis of the implementation carried out in WP5.2 of MEMBRANE. This document is presenting, at first, the performances achieved thanks to the implementation of the algorithms selected for prototyping. The implementation path chosen for the development and integration of the techniques into the hardware platforms is partially depicted. The algorithms and the hardware boards selected for each implementation are briefly described followed with a description of some implementation choices and the reasoning behind them. Complexity issues faced during implementation are also presented justifying certain choices and simplifications needed for prototyping of the algorithms. Thanks to both performance results and complexity analysis, it is possible to deduce some recommendations for wireless meshed backhaul networks.

This document is subdivided into three sections. In the first part, a brief overview of the two algorithms selected for implementation is drawn followed also by a short description of the two platforms chosen for the MEMBRANE prototyping activity. The same section includes the description of the demonstration scenarios – demonstrator setup and configurations – needed to obtain performance results of interest. A description of the different metrics used and the parameters needed for the configuration of the prototypes in order to produce reliable performance results concludes the section. The second section of the deliverable contains the performance results obtained from the two prototypes and a complexity analysis of the implementation of the novel models. The performance outcomes of the prototype boards are compared with results obtained through simulations for validation purposes. Finally, this document concludes with the provision of some general recommendations for the implementation of the selected algorithms into future wireless backhaul networks such as MEMBRANE.

2 PROTOTYPE IMPLEMENTATIONS

2.1 Novel Algorithms & Protocols

The work carried out within WP5.2 of MEMBRANE targets to validate the techniques and algorithms investigated within the framework of the project in order to demonstrate the feasibility of future implementation of these technologies in novel backhaul networks. The development of certain algorithms over hardware platforms provides a concrete proof that these concepts investigated in MEMBRANE can be exploited by next generation wireless telecommunication networks. However, due to the large research framework, not all features can be prototyped and demonstrated. Hence, only a selection of advanced algorithms and protocols are implemented and used to validate the concepts studied in the project. Since the research area of MEMBRANE spans along several layers of a typical communication system a judicious selection of the implemented techniques is carried out in order to have a better insight of the theoretical and practical advantages of the MEMBRANE network. Two features have been selected for demonstration. First, the compensation of the wireless medium effects through a careful exploitation of the mesh topology of the network is demonstrated by the implementation of the Data Splitting Algorithm (DSA) investigated extensively in WP4.1. Secondly, the enhancements offered with opportunistic scheduling and routing protocols able to satisfy and support the heavily backhauling throughput demands of the network are shown with the Distributed Scheduler proposed in WP4.2.

2.1.1 The Data Splitting Algorithm

The Data Splitting Algorithm (DSA) developed within the scope of MEMBRANE project was selected as an algorithm for implementation at the MEMBRANE PHY demonstrator.

The DSA algorithm implements a two-hop transmission protocol in the communication system with end node (or base station or source node), several intermediate nodes (or relay stations) and access node (or subscriber station or destination node). In the first hop the transmission is done from the source node to the relay nodes and during the second hop the transmission of the data is performed from the relay nodes to the destination node. Also the direct transmission between the source and the destination nodes is supported. Each node may be equipped with multiple antennas and the DSA algorithm supports simultaneous transmission of the several parallel data streams. Different combinations of the parallel spatial data streams can be performed in the system with the DSA and the DSA selects the optimal routing of these spatial streams in order to maximize the throughput of the considered communication system.

For each station the DSA calculates weight vectors for TX and RX antennas performing the closed loop MIMO optimization employing the optimal Singular Value Decomposition (SVD) techniques. At the same time the non-interfering transmission between different spatial streams (even coming through different relay stations) is guaranteed. The non-interfering transmission is achieved by appropriately selecting orthogonal subspaces when applying SVD optimization for the spatial streams sent between different stations. More details on the Data Splitting Algorithm may be found in [MEM D4.1.2].

The performance of the DSA with different number of the relaying stations and antenna elements for each source, relay and destination stations have been considered in the [MEM D4.1.2] by using results of the simulations and demonstrating the performance gains which may be achieved with the DSA. At the same time the prototype implementation of the DSA was limited to only one configuration which had to include all the specific features of the DSA system and be able to demonstrate all the advantages provided by this algorithm. The configuration with four MEMBRANE nodes has been selected for implementation: one destination node, two relay nodes and one source node. Each node was equipped with two TX and two RX antennas.

The operation of the DSA algorithm for the system with the considered configuration is the following:

- The data may be transmitted using two spatial streams in different combinations. Each combination corresponds to one transmission mode.
- Transmission modes may be: 2x2-MIMO direct path transmission, 2x2-MIMO only one relay transmission, 2x2-MIMO two relay transmission, etc.
- DSA calculates antenna weight vectors and throughputs for all possible transmission modes.
- Finally, the transmission mode with largest throughput is selected for operation.

All possible DSA transmission modes for two spatial data streams for the PHY demonstrator are shown in Figure 1.

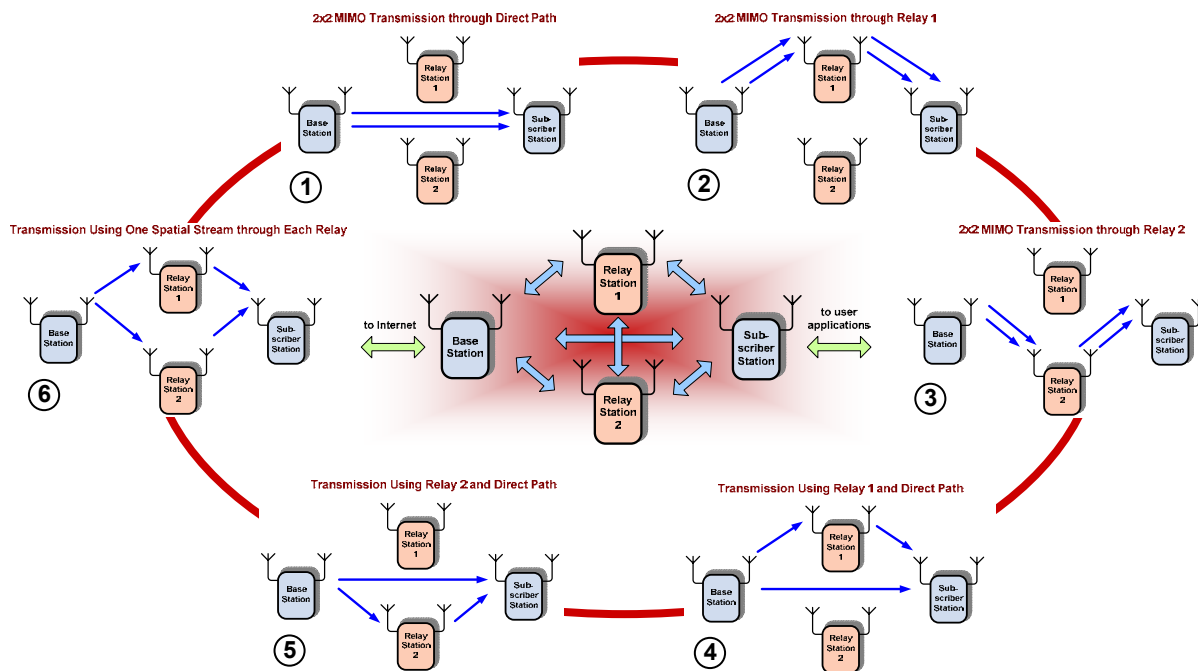


Figure 1: DSA Transmission Modes for PHY Demonstrator

The detailed description of the performance analysis results obtained for the PHY demonstrator evaluation is provided further in this document.

2.1.2 The MEMBRANE Distributed Scheduler

Throughout research carried out in WP4.2 on “routing, scheduling and power control for wireless backhaul network optimization”, a framework for scheduling has been established from which the novel MEMBRANE Distributed Scheduling (MDS) algorithm has been deduced. The distributed scheduling algorithm has been described in [MEM D4.2.1] and more analytically in [MEM D4.2.2]. The algorithm is basically composed of four phases: utility exchange, initial decision, initial decision exchange and final decision. For simplification, the one-round version of the algorithm has been selected for implementation.

a) The Utility Exchange phase

During this phase, each node is computing the appropriate utility function for each of its incoming and outgoing links from U_{out}^i and U_{in}^i which are the utility functions of an outgoing and incoming link i at the node. The utility functions usually reflect the “quality” and “benefit” of transmission or reception

over this link measured during the previous time slots. For the demonstrator, two different metrics are used as utility functions: Signal-to-Interference Ratio (SIR) and Proportional Fair (PF).

b) The Initial Decision phase

Taking into account all links i of each node, if $\max_i U_{out}^i > \max_i U_{in}^i$, then the node makes an initial decision to transmit on the link that maximizes the utility function in the next resource allocation slot. Otherwise, it chooses to receive and expect reception from the link with the maximum utility function.

c) The Initial Decision Exchange phase

The nodes having concluded on an initial decision of “transmission” during next slot, must broadcast their initial “transmit” decision to all their one-hop neighbour nodes via the dedicated control channel. The control message must indicate the IDs of the intended transmitter (Origin) and receiver (Destination) based on the initial decision. This assumes that all nodes are aware of their one-hop neighbourhood and have connections established with the nodes involved in the scheduling process.

d) The Final Decision phase

It is possible that a link conflict occurs when two or more nodes have an initial decision to transmit at the same time. In this situation, each node that receives a control message from its neighbour in the previous phase, checks if any link conflict exists for any of its associated links. If a link conflict is identified, then the node changes its status from “transmit” to “receive”. This corresponds to a “backoff”. Otherwise, the node can choose one of its links indexed by $\arg \max_i U_{out}^i$ to transmit in the next available slot.

The implementation of the algorithm has used, as mentioned before, two different utility functions. The overall frame structure of transmission between the nodes of the MEMBRANE hardware prototype is composed of control and data sub-frames. The control messages issued from the algorithm and the phases described above are exchanged between nodes through portions of the control frames, i.e. mini-slots describe in Figure 2.

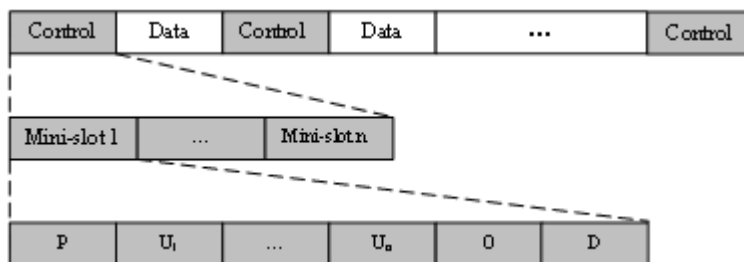


Figure 2: MDS Algorithm Frame Structure

In the figure, (P) correspond to a Pilot phase used basically for measurement of the quality of the link if needed, (U_1) to (U_n) are utility function values of the incoming links measured at the specific node, (O) is the origin of the decided transmission and (D) is the initial decision destination node identifier. It can also be seen that a set of n mini-slots is required to support a one-hop sub-network of n nodes.

2.2 Implementations

In order to provide a more complete description of the capabilities offered within the MEMBRANE network, the implementation of the two selected algorithm has been carried out into two distinct hardware prototypes. This methodology offers flexibility over the evaluation of the complexity, development, implementation of the algorithms leading in diversified results that can prove the independence of the algorithms to the hardware components and platform. The issues raised throughout the integration process are better understood and multi-facet solutions can be proposed.

The two demonstrator of the project have been identified as PHY and MAC demonstrators due to the layers where the major parts of the algorithms are implemented. Hence, the PHY demonstrator aims to demonstrate the advantages of the DSA algorithms which mainly affect the PHY layer of communication of the network. On the other hand, the distributed scheduling mainly involves operations carried out in the MAC layer and above. Hence, the MAC demonstrator targets to provide proof-of-concept of distributed scheduling advantages.

2.2.1 The PHY Demonstrator

As it was discussed in the previous sections, the selected scenario for the PHY demonstrator included implementation of the Data Splitting Algorithm (DSA) for the system configuration with four nodes. The four elements of the considered system are the source node, two relay nodes and the destination node. Every node is equipped with two transmit and two receive antennas.

The block diagram describing the architecture of the developed PHY demonstrator is shown in Figure 3.

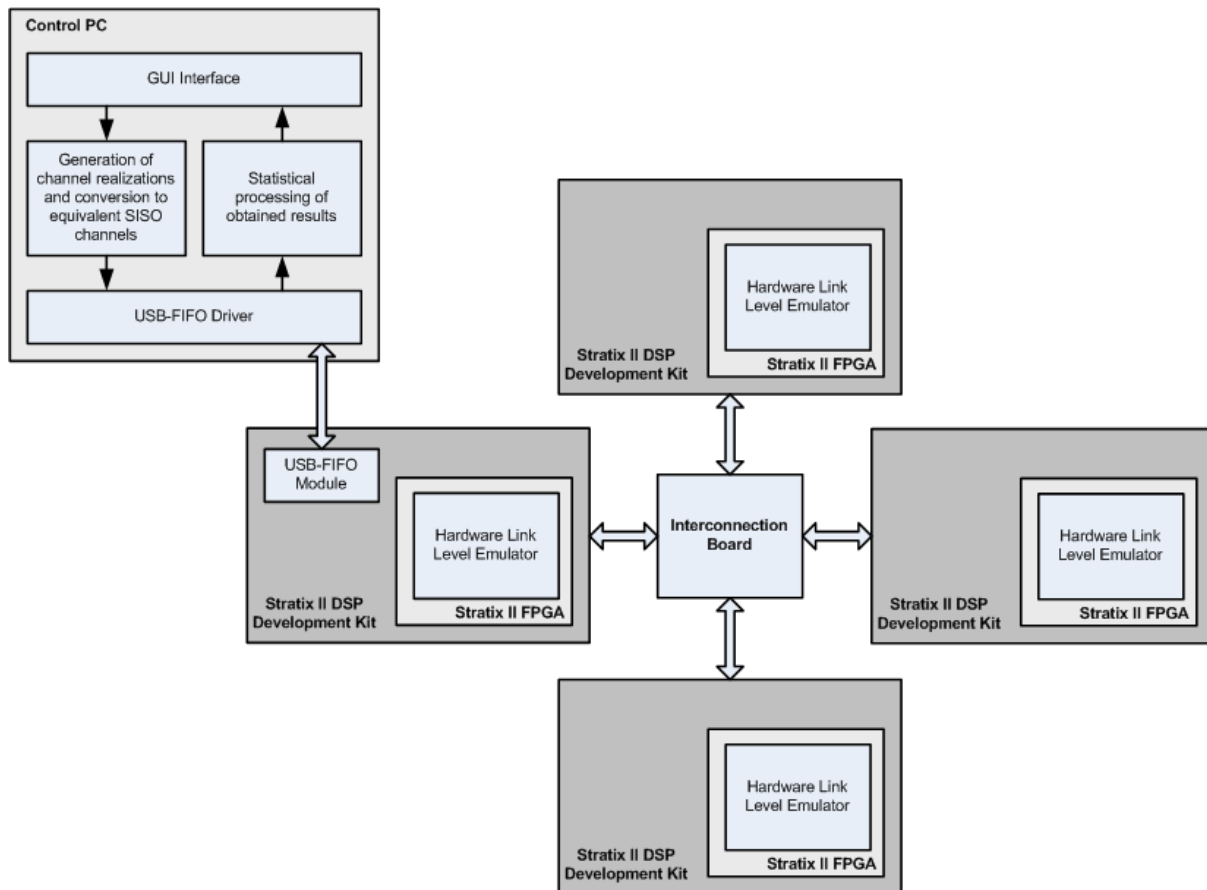


Figure 3: Block diagram of MEMBRANE PHY demonstrator

The photo of the operation of the PHY demonstrator is shown in Figure 4.

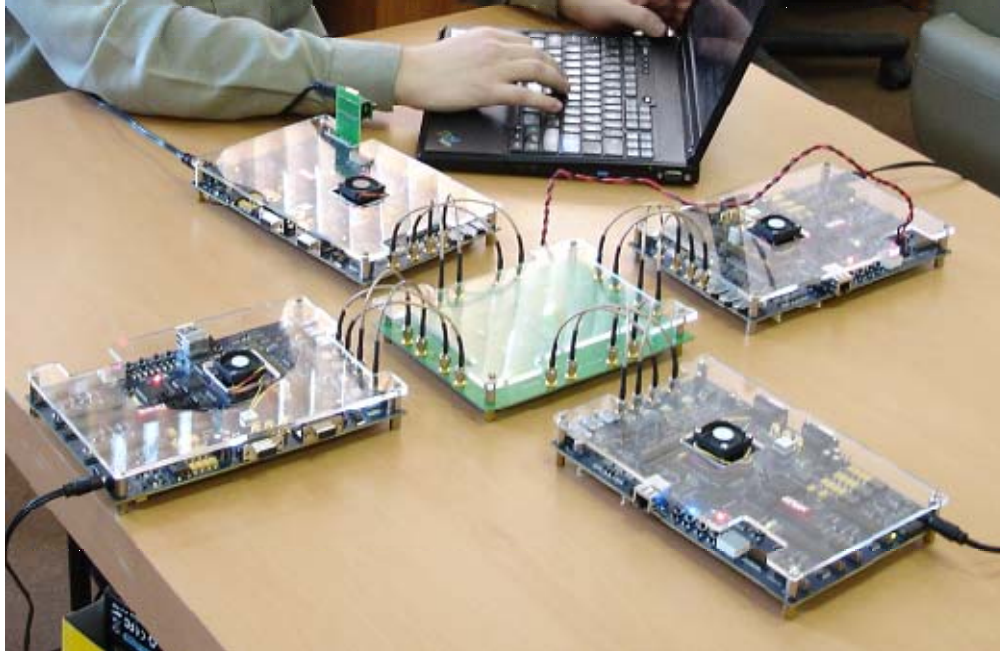


Figure 4: Photo of the MEMBRANE PHY Demonstrator

The DSA algorithm, from the implementation point of view, requires support of the linear MIMO beamforming at both TX and RX ends of each node. Taking this into account, the following described channel emulation scheme was designed for the PHY demonstrator implementation. The MIMO channel of each 2x2 MIMO links was recalculated into two equivalent SISO links and the emulation of these effective SISO links was completed. The calculation of the MIMO beamforming coefficients and the equivalent SISO links of those beamforming weight vectors are required only at the beginning of each new channel realization. Hence, they were developed in MATLAB in the PC controlling the demonstrator. The High-speed Hardware Link Level Emulator (HLLE) for the SISO OFDMA links has been developed and implemented in the FPGA. The HLLE constitutes a hardware simulation pipeline and includes transmit, channel emulation and receive parts in the same IP module. The block diagram of the HLLE is shown in Figure 5.

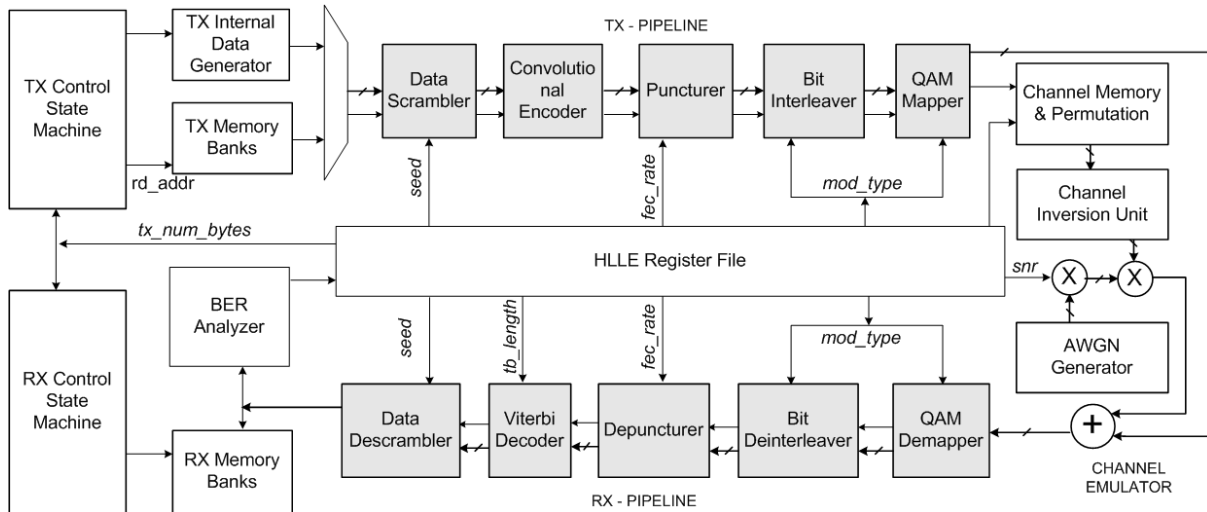


Figure 5: Signal processing blocks of HLLC

Thus, emulation of the whole communication link was done into the single FPGA device using the HLLC. Nevertheless, the configuration of the demonstrator with four FPGA boards representing four nodes was kept. The emulation of the link between two nodes was done at either TX or RX node (FPGA board) of the considered link balancing the computational cost of different FPGA boards (nodes). The dedicated custom interconnection board was designed to allow data exchange between the FPGA boards (nodes).

The high-level operation of the PHY demonstrator is performed in the following way. The user sets the required simulation parameters for the PHY demonstrator using the Graphical User Interface (GUI) running at the control PC. Then, the required iterations over the channel realizations and different Signal-to-Noise Ratios (SNRs) are started. For each iteration, the MIMO channel realization are generated, the sets of the weight vectors for different modes of the DSA algorithms are calculated and the equivalent SISO channels for the required modes are obtained. Then, the equivalent SISO channels characteristics together with the SNR values Modulation and Coding Sets (MCSs) indices are downloaded to the master board (source node) of the prototype using the USB-FIFO interface. Each node of the demonstrator includes one realization of the HLLC at the FPGA board and is able to perform emulation of the OFDMA SISO channel with the real-time speed of about 20 Mbps. The simulation task obtained by the master/main board (MB) is conveyed to either its own HLLC or the HLLCs on the other secondary/slave boards (SBs). The dedicated transceivers are implemented within each node to support the information exchange between nodes using the interconnection. Upon completion of the simulations for the given task the results are sent back the MB and communicated to the control PC. The PC software processes the raw simulation data (BER and PER characteristics) obtained from the PHY demonstrator, processes them to obtain the required high-level parameters and presents to the user with the help of the GUI interface.

2.2.2 The MAC Demonstrator

The MAC demonstrator of MEMBRANE targets to validate the behaviour of advanced protocols of the projects that aim to optimize the resource allocation schemes of the wireless mulithop backhaul network. The MEMBRANE Distributed Scheduler (MDS) has been developed, implemented and evaluated in two phases.

The aim of the first phase was to implement the MDS algorithm under a simplified environment that could support the full functionalities and specifications of the algorithm while at the same time avoid most of the typical hardware constraints, difficulties and irregularities. To achieve this goal, off-the-shelf components were used that allow the implementation of the MDS algorithm fully on the MAC

layer and keep it independent of the under-laying transmission medium or the rest of the network stack lying above it. Hence, the wireless transmission medium emulation has been provided by a standard 5-port switch. All stations of this configuration are attached to the switch that is in charge to offer receiving and transmit capabilities to all stations at the physical layer level under the assumption that each nodes are in the same transmission range. Additionally, using a standard laptop running Linux Operating System for each node, the independence and interoperability of the MAC components developed to provide the MDS features is established.

In the second phase of the MAC Demonstrator development, the central wireless medium entity is replaced by a hardware platform to support transmissions compatible to the IEEE 802.16e standard [802.16e-2005]. The platform contains 4 custom hardware boards with embedded parallel DSPs and network processors that can support the PHY and MAC functionalities required to support the MDS algorithm. The platform is completed with 2 additional boards: a commercial FPGA board in charge to manage and maintain synchronized the data and control transmissions from the 4 hardware entities used to form the system nodes of the prototype and a custom made passive hardware board that provides the connection interface of all the boards of the platform. Besides the 3 laptops used in the first phase, an additional PC is used for management and control over the entire system. This PC is connected to the 4 hardware boards through their Ethernet ports in order to perform all sort of operations spanning from code download and debugging to performance metric sink. A custom made Graphical User Interface (GUI) is running on the management PC and provides both control and diagnostics over the entire MAC demonstrator platform.

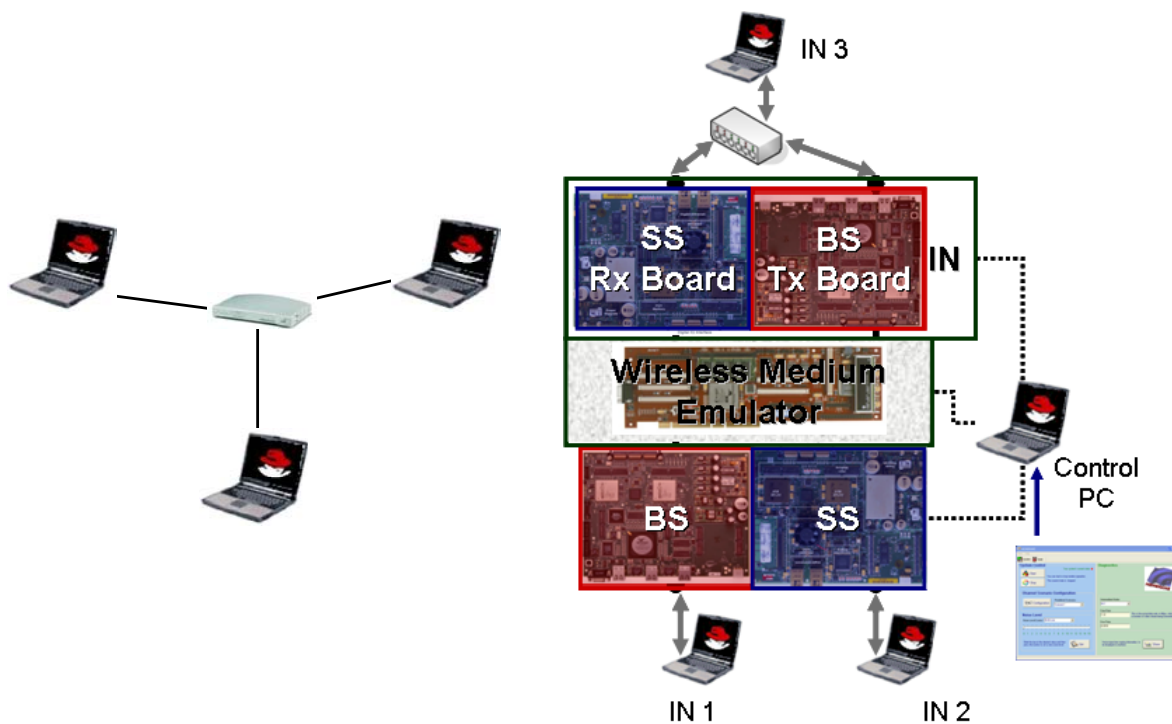


Figure 6: MEMBRANE MAC Demonstration Hardware Configuration (Phase 1 and 2)

2.3 Configurations

In order to provide reliable proof-of-concept of the gains achieved with the selected algorithms for implementation, specific demonstration scenarios must be followed. From the outcomes of these scenarios it is possible to deduce the reliability of the implementation process and also the importance of the algorithms. The configurations described in the sections are used not only to setup the two demonstrators but also to deduce relevant results that can lead to a better understanding of the

algorithms. From the specific configuration it also becomes possible to issue some recommendations for the effective nodes of the MEMBRANE system.

2.3.1 Centralized Demonstrator Scenarios with DSA and PHY Demonstrator Setup

The analysis of the target deployment scenarios for MEMBRANE network has been performed in [MEM D2.1]. Two most relevant scenarios for MEMBRANE network have been identified based on the operation areas which are urban and rural scenarios.

The [MEM D2.1] document includes description of the main parameters for selected urban and rural scenarios. All parameters are divided to four major groups:

- topology parameters (i.e. type of channel model, propagation conditions, pathloss and shadowing models, frequency planning, bandwidth and population density);
- antenna and device characterization parameters (number of sectors per node, number of antennas per sector, antenna type/gain/spacing/arrangement, transmit power and receiver sensitivity);
- network and system parameters (cell size, cell type, number of nodes, backhaul end-to-end performance requirements);
- service parameters and traffic models.

All the recommendations for scenarios characteristics given in [MEM D2.1] have been taken into account at the development of the evaluation scenarios for the MEMBRANE PHY demonstrator. The most relevant parameters for PHY Demonstrator that were used for system evaluation are listed in Table 1 below:

Table 1: PHY Demonstrator Related Parameters

Parameter		Value		
Node		EN (End node)	IN (Intermediate)	AN (Access node)
Channel information	Type of channel model	LOS	LOS	LOS/NLOS
	Propagation conditions	Outdoor	Outdoor	Outdoor
	Path Loss model	Scenario U1	Scenario U1	Suburban Macro
	Shadowing model	Lognormal 3 dB std dev	Lognormal 3 dB std dev	Lognormal 3 dB std dev for LOS; 8 dB std dev for NLOS
	Frequency planning	2.5GHz		
	Bandwidth	10 MHz		
Antenna and Device characterization parameters	Transmit power	20-40 dBm per sector	20-40 dBm per sector	10-20 dBm per antenna element
	Receiver sensitivity	NF 4 dB, IL 2dB	NF 4dB, IL 2dB	NF 6 dB, IL 2dB
	Number of sectors per node	1	1	1
	Number of antennas per sector	2	2	2
	Antenna type	Omnidirectional	Omnidirectional	Omnidirectional
	Antenna gain	10 dBi	10 dBi	10dBi
Antenna arrangement	Linear	Linear	Linear	

As it was outlined before the PHY demonstrator implements four node network configuration with one end node (base station), two intermediate nodes (relay stations) and one access station (subscriber station). Each node in the network is equipped with two transmit and receive antennas and is able to support up to two spatial streams with other stations in the closed-loop MIMO mode.

So the parameters provided in Table 1 have been adopted for the PHY demonstrator evaluation scenarios with taking into account the main goal of the MEMBRANE prototype implementation which is the performance evaluation of the DSA algorithm.

The parameters used to control the PHY demonstrator performance are divided into the two main categories – parameters which can be controlled through MATLAB scripts and parameters which can be changed using designed GUI. The parameters realized in the control scripts are default parameters and are hidden from user. These default parameters are used to generate spatial channel models for all links i.e. between EN – IN, EN – AN and IN – AN. The channel models are generated in accordance with selected scenario urban/suburban and using specified in Table 1 parameters of antenna and device characterization. The parameters that can be tuned /adjusted before each simulation cycle were placed at GUI. Figure 7 shows PHY Demonstrator parameters that can be modified using GUI.

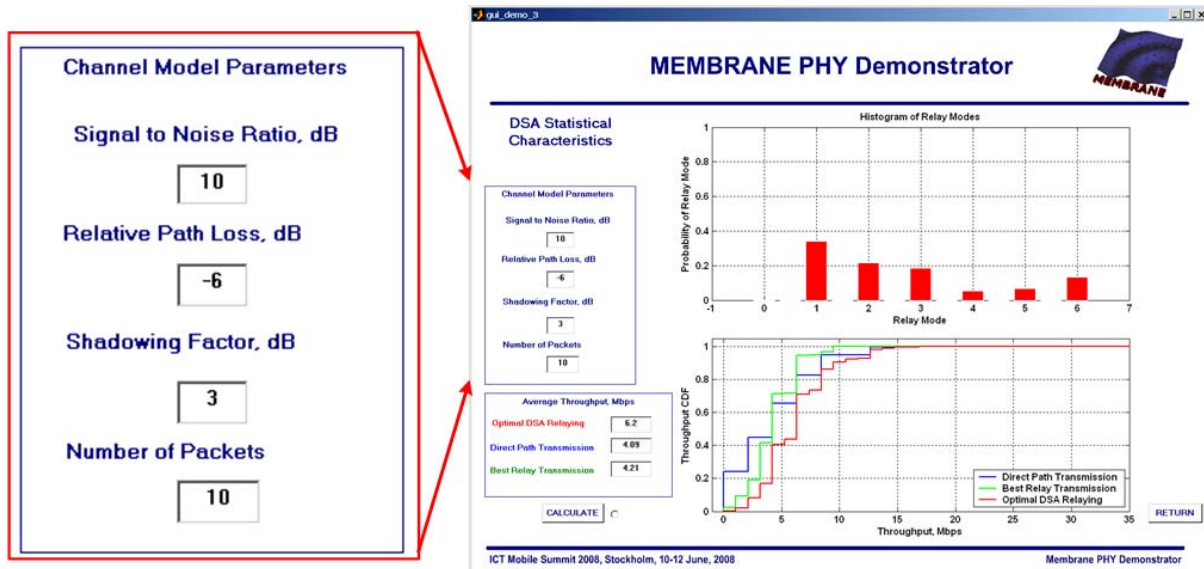


Figure 7: GUI parameters of PHY Demonstrator

As it can be seen from Figure 7 the four main parameters can be tuned before each run of PHY Demonstrator simulation. There are:

1. Signal to Noise Ratio (SNR) - this parameter specifies average SNR for each link;
2. Relative path loss – specifies relative ratio of path losses between direct path and all other links;
3. Shadowing Factor – specifies value of lognormal shadowing standard deviation for all links;
4. Number of packets – specifies the number of packet to be transmitted/processed by PHY Demonstrator before it makes a decision about modulation and coding scheme that can be used at particular link. Note that PHY demonstrator uses real data packets instead of PHY abstraction methodology. The usage of this parameter allows to measure and control accuracy of the BER/PER measurements on each link.

Finally four illustrative scenarios have been identified for the PHY demonstrator performance evaluation. These scenarios are listed in Table 2 and were modeled using Winner B1 channel model.

Table 2: PHY Demonstrator Scenarios

Scenario name	Value of relative path-loss	SNR value
1) High SNR and Blocked Direct Path	-18	25
2) Low SNR and Blocked Direct Path	-18	5
3) High SNR and 3dB loss in Direct Path	-3	25
4) Low SNR and 3dB loss in Direct Path	-3	5

2.3.2 De-centralized Demonstrator Scenarios with MDS and MAC Demonstrator Setup

The MDS algorithm de-centralized operation relies on the exchange of control data information among the MEMBRANE nodes. This exchange ensures that all nodes acquire knowledge of the network state at the time of the exchange so that they all make the desired, and most of all consistent, scheduling decision. The demonstrator is thus focused on the overhead caused by this control data exchange. The absolute value of this control information depends on the format of the control packets in the demonstrator, which is presented below:

Table 3: MAC Demonstrator Packet Header Format

Field	Length (bytes)	Description
Destination address	4	This field has always the value of 0xFFFFFFFF (broadcast) in control packets.
Source address	4	Address of the sender.
Packet type	1	Packet type (control or data)

Table 4: MAC Demonstrator Control Payload Format

Field	Length (bytes)	Description
Intended destination	4	When the node's initial decision is to send, this field contains the intended destination of the transmission. When the node opts not to send, then this field is zero.
Utility vector length	4	Length (in entries) of the utility vector.
Utility data		Utility data contain formation on the link between the specified neighbour and the sender of this control packet. The value is calculated according to the scheduling metric being employed (SIR based, proportional fair, etc). Since there is one such entry per neighbour, the field's total length in bytes is <i>Utility vector length</i> x <i>Utility data size</i> .
- Neighbour address	4	
- Utility value	4	

Note that we selected a 4-byte address space for our demonstrator. A real system might use a longer or shorter space that must guarantee uniqueness in all cases. A possible option would be to use the well known IEEE 802 address (6 bytes).

The scenarios investigated in the demonstration involve two and three nodes. The two-node scenario aims to verify basic operation of the protocol with a minimal topology. The three-node scenario investigates a case where three nodes compete for bandwidth in the same broadcast medium (i.e. the nodes are connected in a mesh, where everyone can transmit directly to (and interfere with) all other nodes. This topology is shown below:

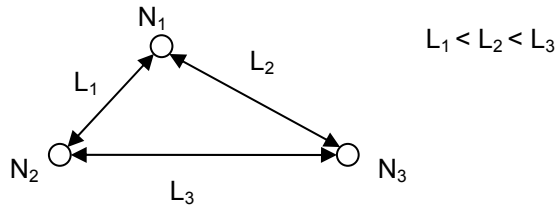


Figure 8: Three-node topology

As shown, node N_3 is placed further from the other two and therefore its links with the rest of the network face greater loss. In the case of channel-aware scheduling utilities, such as capacity (or SIR) based scheduling or proportional fair scheduling, the algorithm incorporates this situation into its decisions.

Finally, the transmission characteristics and loss vector considered is shown in the table below:

Table 5: MAC demonstrator transmission and loss parameters

Transmission power	20 dBm
Modulation and Coding Scheme	64 QAM with 3/4 convolutional code (used for artificial error generation)
Bandwidth	10 MHz
Noise level	-174 dBm/Hz
L_1	100 dB
L_2	103.17 dB
L_3	107.75 dB

2.4 Performance Measurements

The metrics used for the validation of the algorithms that have been selected for implementation are described in this section. They are mainly categorized in two groups according to the layer where the measurements take place. However, this distinction does not imply that these metrics are exclusively defined and used by components of the layer. The range of exploitation of the specific metrics can extend to higher or lower layers. In this section, we mainly focused to provide consistent parameters required for the understanding of the algorithms. The PHY layer metrics are mainly used to estimate the quality of the physical transmission between different nodes of the demonstrators. On the other hand, the MAC layer performance metrics identify the quality of the links established among nodes of the system.

2.4.1 PHY Layer Performance Metrics

The developed PHY Demonstrator is able to measure different PHY Layer performance metrics. It allows to measure BER/PER curves versus SNR for any channel model and different modulation and coding schemes (MCS). Using the developed demonstrator it is possible to measure the end-to-end BER/PER performance for multi-hop transmissions and estimate total network throughput and throughput of each hop. It also can be used to estimate the system level performance metrics for different propagation scenarios, network topologies and PHY layer MIMO algorithms. The hardware acceleration allow processing of real data packets and direct measurement of BER and PER that can be used for system level performance evaluation instead of PHY abstraction methodologies.

The main PHY layer performance metrics measured in current project were selected to demonstrate performance gains that can be achieved by DSA algorithm in application to multi-hop backhaul network using multiple antennas. As soon as this algorithm involves possibility of multi-hop transmission and selection of intermediate node the following metrics were used as the main PHY performance metrics:

1. Distribution of DSA transmission modes. The developed PHY demonstrator plots histograms of different transmission modes. Using this statistics it is possible to characterize the best transmission mode for any particular propagation scenario and give a recommendation for system deployment.
2. Average system throughput. The PHY demonstrator estimates the throughput of each possible transmission mode of DSA algorithm. This metric show effectiveness and gains achieved by DSA algorithm as it always selects the transmission mode with maximum throughput.
3. Throughput Cumulative Distribution Function (CDF). Using throughput CDF curve it is possible to analyze outage probabilities and average spectral efficiency for any network topology and propagation scenario.

Note that the on-line measurements of all listed above PHY layer metrics was implemented in PHY Demonstrator and incorporated to Matlab based GUI.

In the MAC Demonstrator, the entire scheduling process remains a task completely carried out in the data link control layer of the system. However, inputs from the physical layer are required for the estimation of the quality of the transmission links involved in the MDS algorithm and in order to deduce the corresponding utility functions. In particular, a typical PHY layer metric that has been selected and used in MDS as utility function is the Signal-to-Interference Ratio (SIR). This metric can also be exploited in the Proportional Fair (PF) configuration of the MDS algorithm. In the prototype, SIR values are obtained from measurements during both the control and data phases of the meshed transmission shown in Figure 2. The signal strength of each link is measured from the Pilot phases broadcasted by each node during its allocated mini-slot in the control sub-frame. Thanks to the Pilot parts of the mini-slots, each node is capable to measure the signal transmit power of their neighbours. On the other hand, the expected interference of the incoming links is deduced using the data sub-frames. The power of the transmitted data is measured from all neighbouring nodes and considered as the interference issued from the nodes allowed transmitting data during the specific frame.

The utility function based on the Signal-to-Interference Ratio is given below.

$$U_k[n] = \text{SIR}_k[n] = \frac{P_k[n-1]}{I_k[n-1] + N_k[n-1]} \quad (1)$$

In the equation, P_k , I_k and N_k are the signal power, interference level and noise variances of node k , respectively. These values are measured during time instance $n-1$ in order to define the utility function (and the SIR value) of time slot n .

2.4.2 MAC Layer Performance Metrics

Concerning the MDS algorithm implementation this section focuses on the control plane performance/capabilities analysis of the algorithm.

The objective is to evaluate the delay and throughput impact of the imposed control mini-slots varying predefined parameters under optimum communication conditions (wired network, no artificially induced errors). These parameters will be the number of communicating nodes, mini-slot and packet size. Thus the evaluation process is summarized in the table below.

Table 6: Evaluation Setup for the MDS

Evaluation Setup	
Number of participating stations	1-3
Control packet size	50, 70, 90, 1500 bytes
Each Experiment Duration	~90 seconds
Each Experiment Repetition	3

The first and of outmost importance is that, assuming error free control packet transmission and guaranteed synchronization, the implementation proved quite robust exhibiting zero errors, un-synchronizations, and any out of sequence events.

Apart from the errors, valuable insights are provided from measuring the effect of the considered parameters to the average duration of a mini-slot as presented in the following figure.

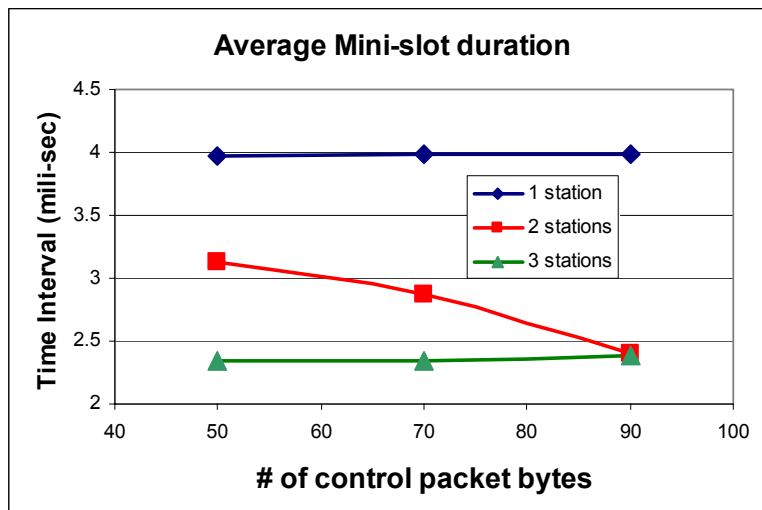


Figure 9: The effect of control packet size and number of stations on average mini-slot duration

The first observation concerns a steady average mini-slot decrease with respect to the number of stations involved in the algorithm. This is because of the method measuring the intervals which is based on packet transmission times (provided by network analyzing applications) rather on packet reception times. Thus, the last mini-slot duration proved more time consuming since pre-data phase process occurred compared to the rest where pre-mini slot process is required. Thus and for example in the “3 stations” experiments, two out of three mini-slots are less time consuming than the third one leading to a reduced average compared to the “1 station” experiments where all mini-slots are the more time demanding ones. Of course “2 stations” experiments are somewhere in the middle.

The second observation regards the clear immunity appearing with respect to the length of the control packet, except the “2 stations” cases. This actually depicts an observation quite clear in all experiments, due to the fact that the delay overhead due to the extra bytes was actually negligible

compared to the delay overhead imposed by the required processing of the algorithm and the operation of the driver. Thus it can be concluded that on one hand the algorithm control capabilities are not significantly affected by the added byte overhead since in controlling and distributed algorithms processing overhead and interrupt response time may well be much more significant. However, as seen the “2 stations” experiments presented a slightly different behaviour, which however since it involved a marginal difference of 0.5 milliseconds can be attributed to coincidental reasons. However, analyzing only the mini-slot duration does provide a myopic view of the parameters’ effect on the whole system behaviour. This effect, which could well be more important, is presented in the following figure.

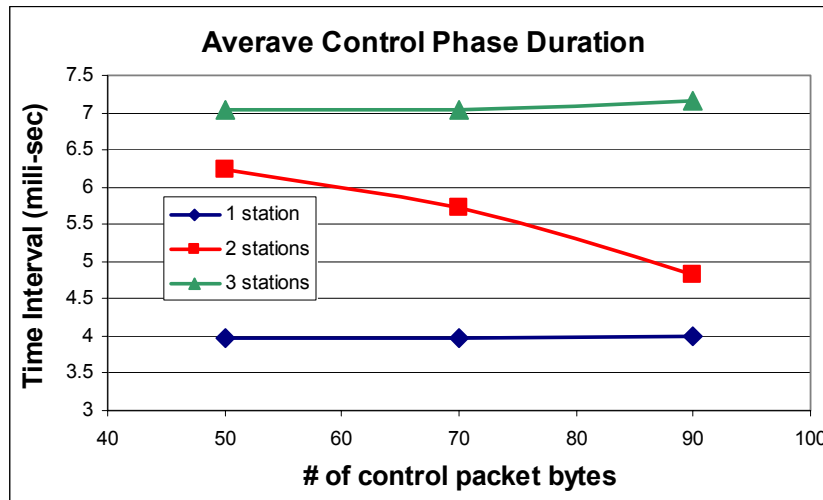


Figure 10: The effect of control packet size and number of stations on control phase duration

These are actually calculated graphs originating from the previous figure by multiplying the “2 stations” cases by 2 and the “3 stations” cases by 3. Thus, it is depicted that although the mini-slots apart from the first one are less time consuming the more stations participating in the algorithm the more delay overhead must be expected. In fact all the stations (besides the first one) add an average 1.5 millisecond delay in each control phase.

The final graph presents the ratio of the control phase to the data phase assuming a steady data phase of a 50 byte packet.

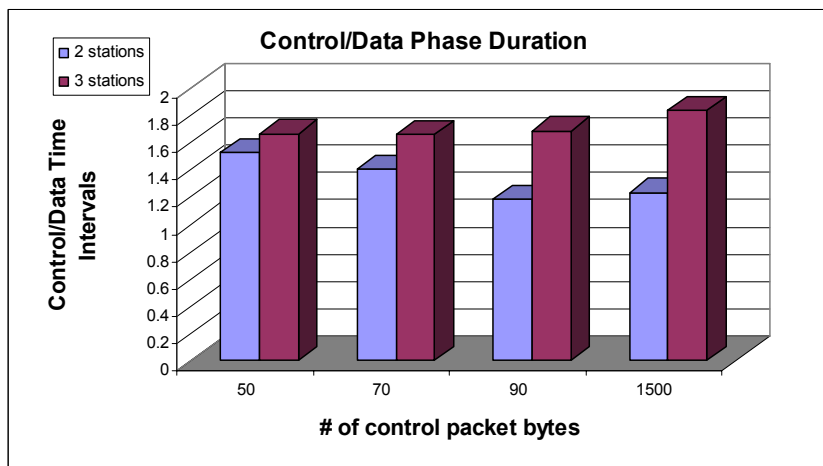


Figure 11: The effect of control packet size and number of stations on Control/Data phase ratio

This figure attempts to provide a measure of the time available for user data transmission compared to control packet transmission and additionally to estimate the anticipated effect of adding an additional station in the algorithm. Once again it is shown that the main source of delay is processing and interrupts servicing rather than the size of the packet. In the case of 2 stations this ratio is in the area of 1.4 to 1.18. Considering there are two control phase in each algorithm cycle, if control and data phase required processing were equal one would expect this ration to be close to 2. Thus, it is indicated that one mini-slot's overhead is significantly lower than the data phase overhead. A more important observation can be drawn, however, from the 3 stations cases. As seen the addition of an extra station increases the ratio only by 0.2 for 50 byte control packet and 0.6 for the excess case of 1500 byte packet. Thus adding an extra station does not severely burdens the network operation and performance. Reminding that no synchronization errors were reported in any case it indicates a rather robust operation and algorithm with deterministic behaviour, which an importance attributes considering the control capabilities of the implementation.

3 PERFORMANCE EVALUATION AND COMPLEXITY ANALYSIS

In this section, the performances and the implementation complexity of the MEMBRANE demonstrators are described. Using the configurations of the prototypes to obtain measurements of the metrics presented in the previous section, the capabilities and performances of the prototypes can be drawn. The performance results obtained for this section are used to demonstrate the selected algorithms and protocols constitute a viable solution for future wireless backhaul networks based on MEMBRANE technologies. The performance results gathered are compared with the ones achieved through simulations. On one hand, the implementation of the algorithms in the prototypes is partially validated through comparison with the simulation curves. In particular, deviations from the expected operability behaviour are corrected. On the other hand, the additional losses and delays losses between the theoretical and the practical versions of the algorithm are assessed. The cost in performances from the simplifications of the algorithms and the implementation hardware constraints is hence evaluated. Additionally, a complexity analysis of the implementation modules and process for both prototypes is carried out to demonstrate the implementation advantages, the hardware simplifications required and the constraints overcome during the integration of the algorithms into the hardware platforms. Finally, from the outcomes of the performances outcomes and the complexity evaluation of the prototypes, recommendations for the implementations of these techniques on backhaul nodes can be drawn.

3.1 Performance Evaluation

In this sub-section, performance results of the PHY layer and MAC layer demonstrators are given. The features implemented and the capabilities offered are reflected to the outcomes of the predefined measurements carried out in the two prototypes. The results deduced are also compared with the theoretical expected results of the equivalent simulators. The validity of the implementation of the algorithms and the appearance of diverse imperfections due to hardware constraints are explained and conclusions are drawn.

3.1.1 PHY Demonstrator Performances

In this section, the performance results of the PHY Demonstrator are given for the setups and scenarios described in section 2. The presented results were obtained emulating Wi-MAX based OFDMA PHY layer assuming 512 point FFT and assuming 5 MHz channel bandwidth. The above rooftop (ART) to above roof top Winner B spatial channel model with strong LOS component was selected for system evaluation. The simulation results, obtained using PHY Demonstrator, are shown in the following figures.

Figure 12 shows the DSA performance characteristics for propagation scenario when the direct path between EN and AN is blocked by some obstacle. As a result the direct path has 18 dB higher pathloss in comparison to pathlosses between EN – IN and IN – AN which on average have SNR equal to 25 dB. Since the direct path has a weak channel the DSA algorithms selects the transmission modes that avoids transmission exploiting direct link between EN and AN. As it can be seen for this particular scenario the optimal DSA relaying has a throughput equal to about 17 Mbit/s that is higher in two times comparing to direct path transmission (mode 0 in relay mode histogram). Note that in this particular scenario the relaying mode number 6 that involves simultaneous transmission of two spatial data streams to two IN outperforms the relaying mode that simultaneously sends two spatial streams through one IN (modes 2 and 3). This fact can be explained by the LOS propagation scenario between EN and IN. So in such propagation scenarios several INs are required to achieve spatial gain and increase system throughput.

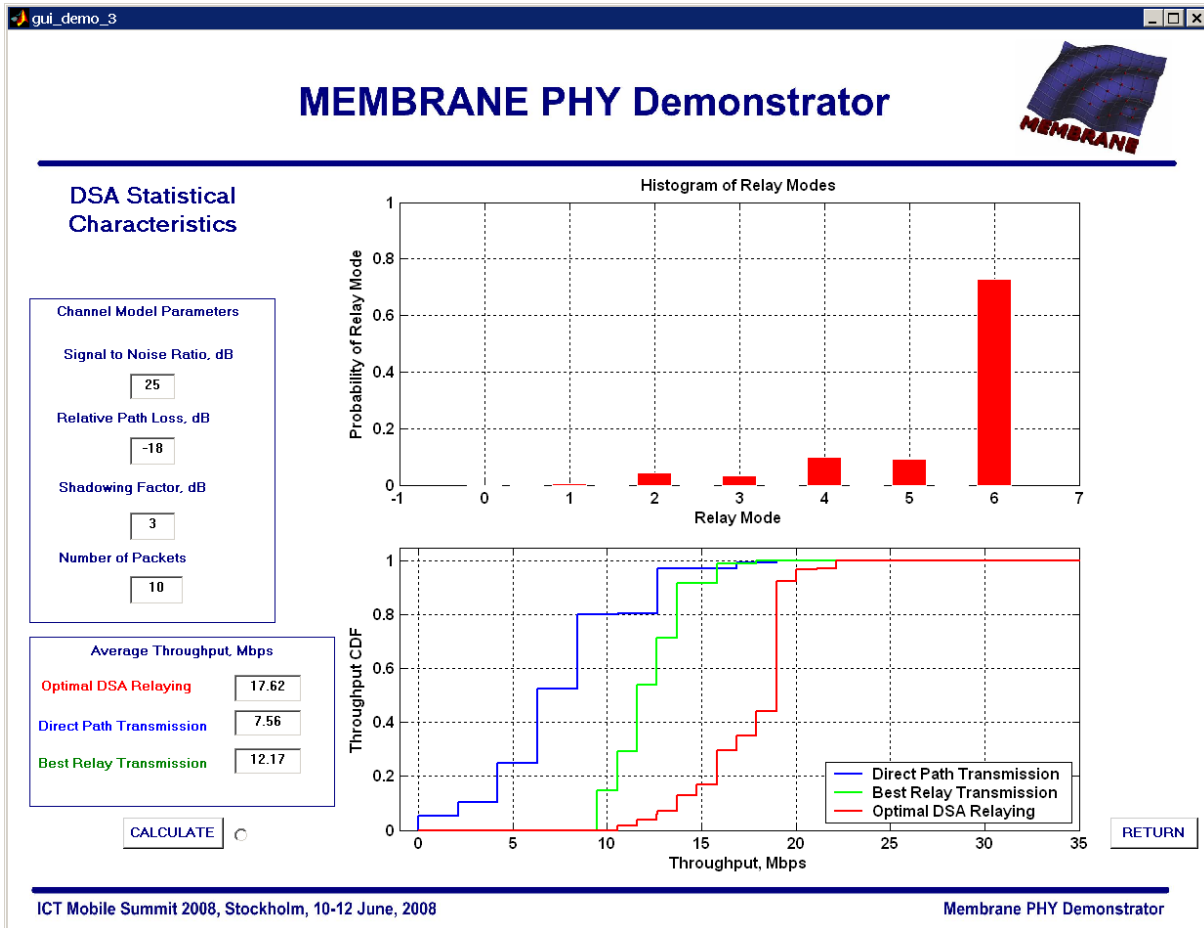


Figure 12: High SNR and Blocked Direct Path

Figure 13 shows the DSA performance characteristics for propagation scenario when the direct path between EN and AN is blocked by some obstacle resulting in 18 dB higher pathloss comparing to pathlosses of other links. In addition it is assumed that EN – IN and IN – AN links have low SNR (equal to 5 dB). Since the direct path has very poor channel gain the DSA algorithms never selects transmission modes that exploit direct link between EN and AN. As it can be seen for this particular low SNR scenario the optimal DSA relaying has low throughput equal to about 3.1 Mbit/s that outperforms best relay transmission mode only on 10 %. Note that the transmission through direct path is not possible at all. As it can be seen analyzing histogram and throughput CDF curve shown in Figure 13 the relaying modes number 2, 3, and 6 the have almost equal transmission probabilities and similar throughput. Note that in such bad propagation conditions the optimal DSA relaying algorithm still allows data transmission at very low data rate significantly reducing outage probabilities.

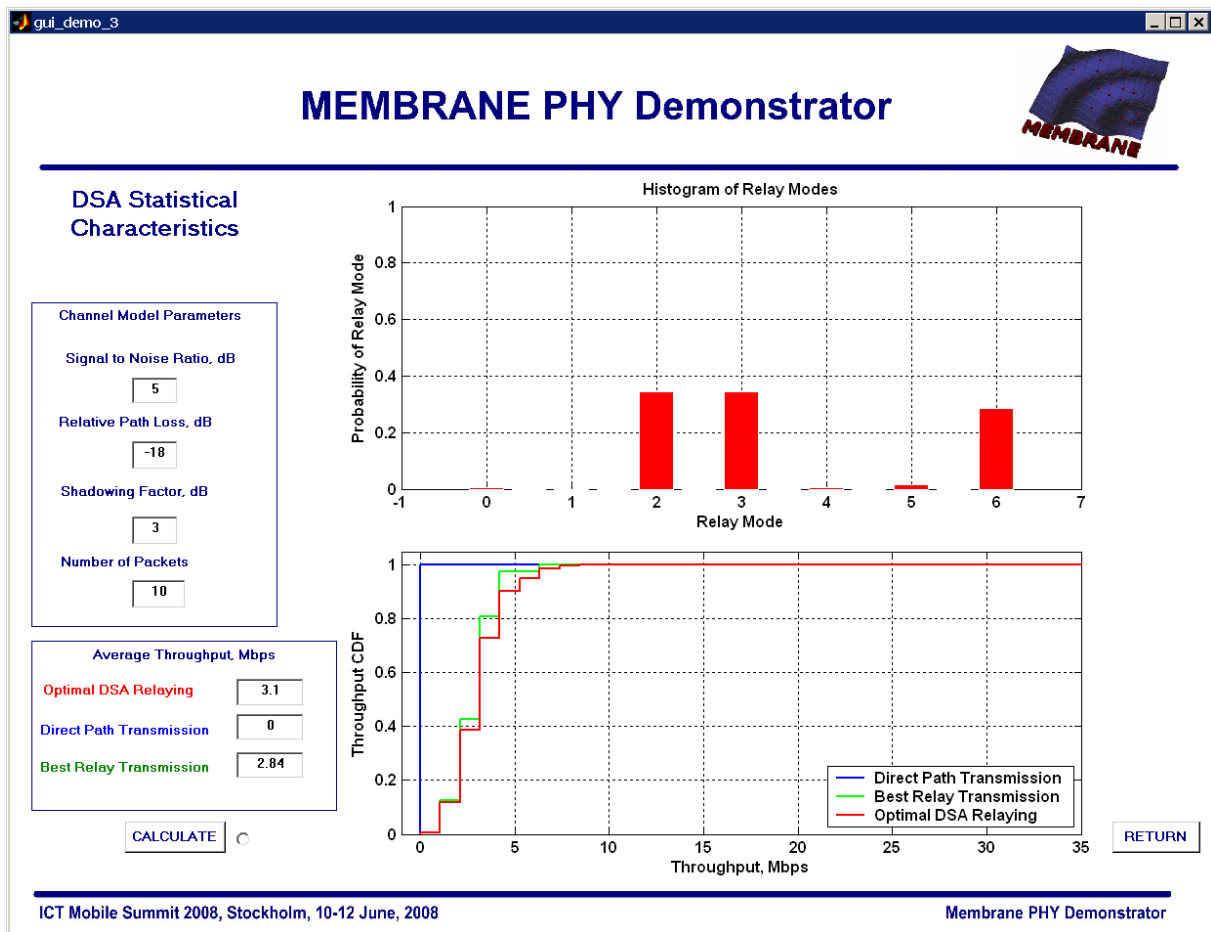


Figure 13: Low SNR and Blocked Direct Path

Figure 14 shows the DSA performance using propagation scenario when the direct path between EN and AN has only 3 dB higher pathloss comparing to pathloss values of other links. In addition it is assumed that average SNR at EN – IN and IN – AN is high (equal to 25 dB). Since the direct path has very good channel quality the DSA algorithms selects the transmission modes that use direct link between EN and AN (i.e. relaying modes 1, 4 and 5 have highest probabilities). It can be explained by the fact that direct path between EN and AN does not introduce any transmission overhead that results in capacity penalty. As it can be seen in this scenario the DSA demonstrates average throughput equal to 26.87 Mbit/s and outperforms best relay transmission in two times. However its gain relatively to direct path transmission is not so high. It shows only about 16% throughput improvement. The reason for this is that at high SNR the direct path transmission has high spectral-efficiency. Note that relaying modes 4 and 5 are selected by DSA algorithm more often than direct path transmission. This can be explained by the LOS propagation between EN and AN resulting in very low gain of second spatial stream. Adapting to this conditions the DSA algorithms decides to split two spatial streams through different branches EN – AN and EN-IN-AN introducing more spatial gain.

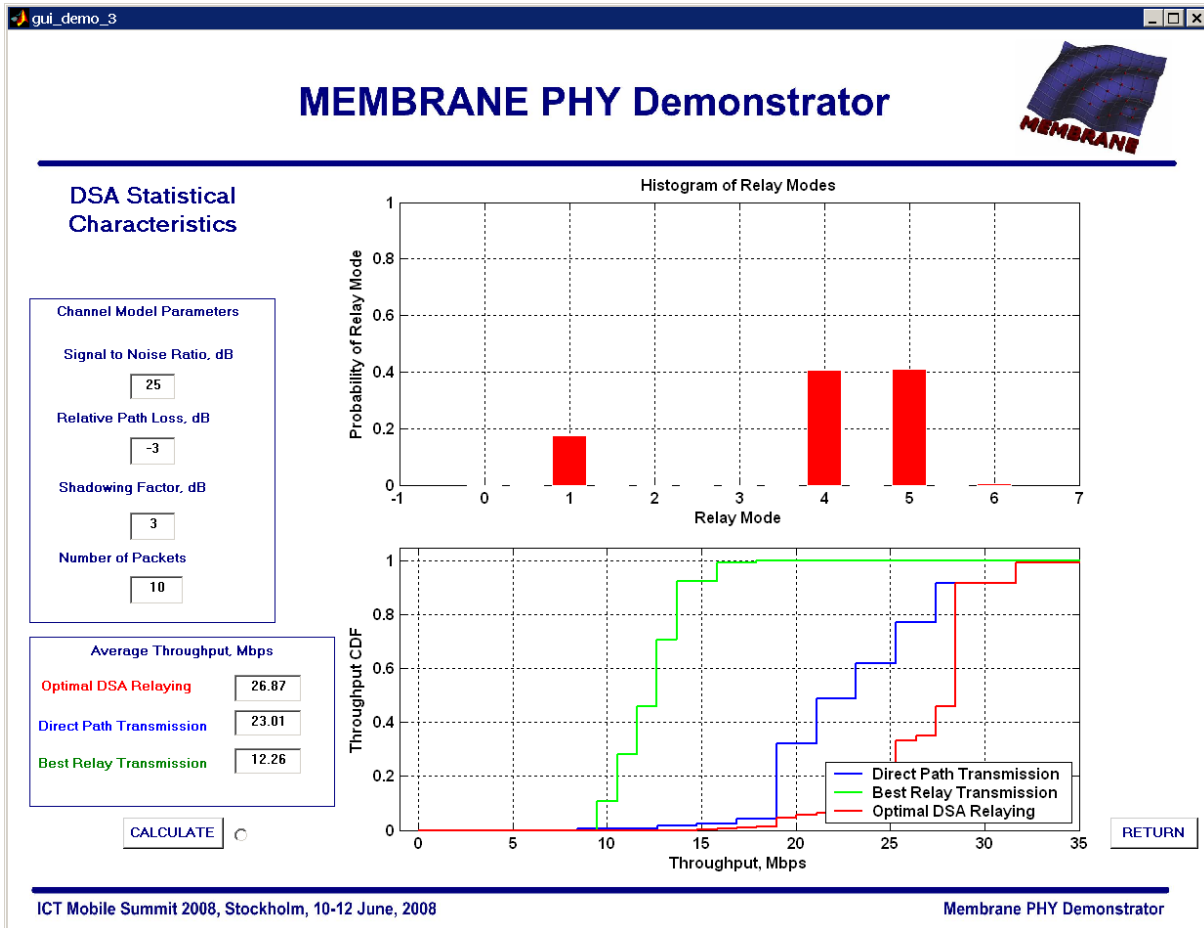


Figure 14: High SNR and 3 dB loss in Direct Path

The last considered scenario is shown in Figure 15. In accordance with this scenario all links have low SNR (equal to 5 dB) and the average power of direct path is 3 dB less than average power of other transmission links. In this scenario the most often selected relaying mode is the transmission through direct path which is used in about 50 % cases. The other relaying modes have about the same relay mode selection probability that varies in the range 8 – 18 %. Note that in this scenario all modes have low throughput. The average throughput of DSA algorithm is higher of average direct path throughput on about 27 %. However the application of DSA algorithms significantly reduces the throughput outage probability of direct path transmission which in 20% cases have zero throughputs.

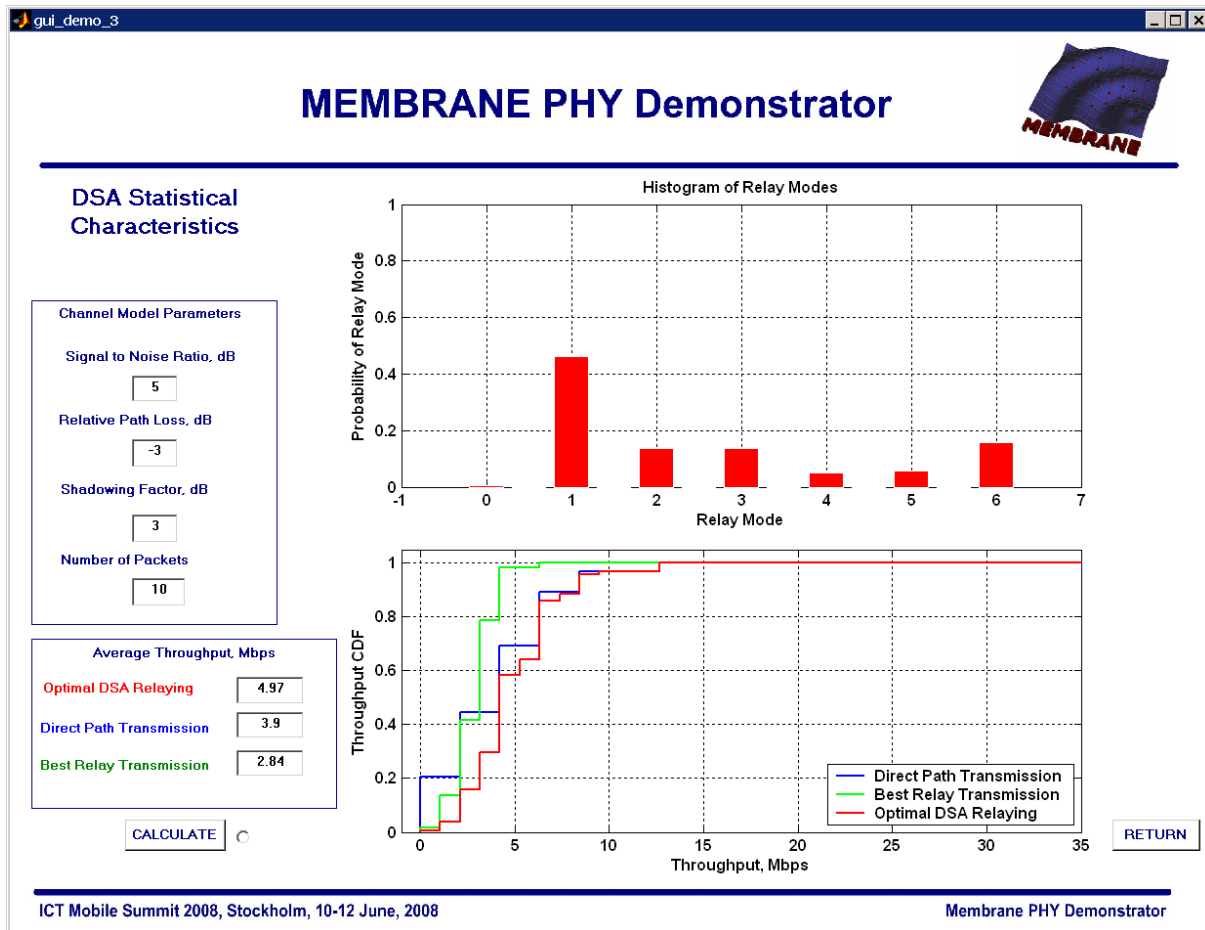


Figure 15: Low SNR and 3 dB Loss in Direct Path

3.1.2 MAC Demonstrator Performances

This section presents measurements concerning the performance of the MDS implementation focusing on comparative evaluation of critical parameters' effect. The evaluation process involved three typical application used in related processes, three different packet lengths (for TCP protocol utilization), two different data flow scenarios and three different bandwidth distribution approaches while two stations were participating in the network operation. The parameters and the range of values are as follows

- Applications: Ping, UDP packet communication, TCP packet communication
- Packet lengths for TCP cases: 600, 1100 and 1400 bytes
- Two different data flow scenarios:
 - One data flow: both directions are measured separately for 2 stations.
 - Two data flows: Crossed for 2 station setup, while for the 3 station setup (ping and UDP application) we considered that they concentrate to a centre station.
- Bandwidth distribution approach: Round Robin, SIR based, Proportional Fair

The metrics the evaluation is based on is delay and delay deviation for ping application and measured throughput for the rest after 90 second experiments repeated at least once to increase measurements validity.

Considering the ping application the following figures presents in an aggregated manner all delay related results while it must be pointed out that in any case ping application produced no errors which is anticipated since ping is a rather relaxed application.

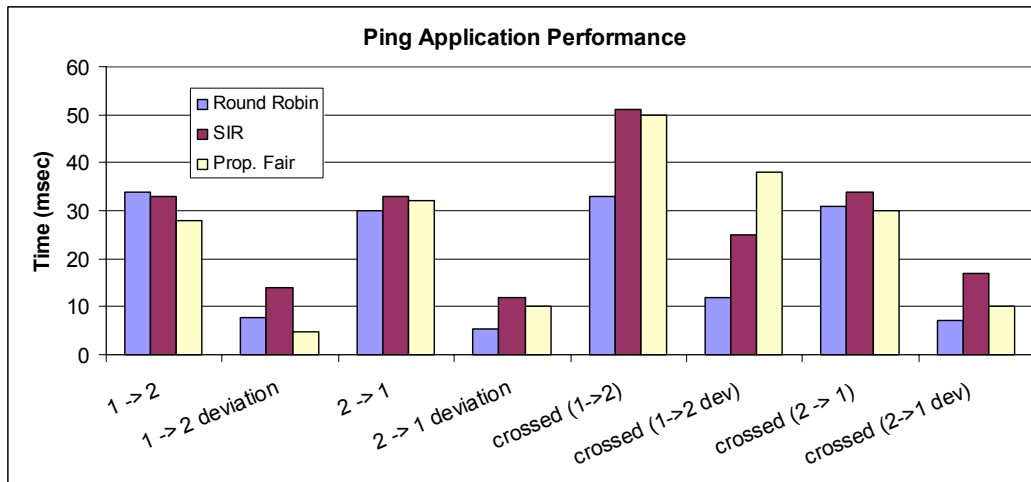


Figure 16: Ping Application Performance Evaluation

As depicted by the first four groups of columns when there is only one flow the performance is the same independently to the direction or the bandwidth shearing approach. This can be extracted comparing groups 1 and 3 which present round trip delays and 2 and 4 which presents deviations around the mean delays. In both cases, differences are marginal and not leading any other conclusion besides analogous performance. This observation is actually logical, and is indication of the algorithm's correct operation, since there is only one data flow with rate below saturation point and therefore all available bandwidth should be allocated to that data flow.

Very interesting results are presented by the next four groups as well representing the 2 crossed data flow scenario. Firstly, once again for the RR (Round Robin) approach the bandwidth is fairly allocated among the two flows both from delay and delay deviation perspective. But when SIR or PF approach is applied a clear favouritism is presented towards the 2 → 1 direction. From the delay measurements this is expressed with a ~20 msec overhead for the 1 → 2 direction representing a 40% increase. This difference is quite evident from the delay deviation as well. Direction 1→2 presents higher delay deviation by 32% for SIR approach and a 70% for PF respectively.

Now, moving on to the UDP and TCP applications which tended to saturate the network, so as to comparatively evaluated networks measurements concerning throughput.

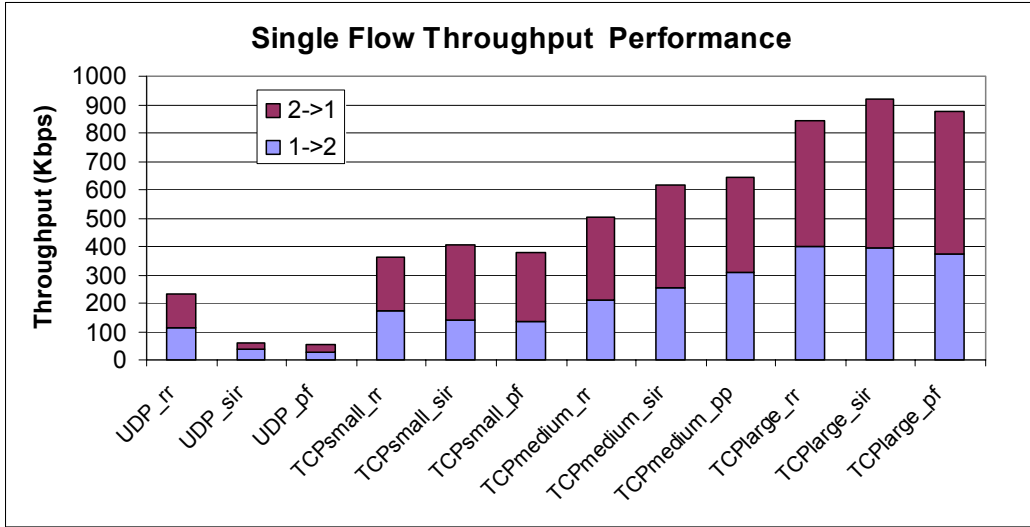


Figure 17: Sing flow Scenario Performance Evaluation

Starting with the single flow experiments and focusing on throughput achieved measurements are presented as soon in the previous figures where each column includes both directions' measurements. The first observation concerns the clearly higher performance achieved by the TCP application. Considering the RR approach where UDP performed its best, TCP due to its connection oriented approach and congestion control algorithms exhibited a higher achieved throughput ranging from 36% up to 72%. Furthermore, it is depicted that RR clear outperforms the two other approaches.

Concentrating on TCP application is evident that in, independently to the approach followed, the larger the data packet, the higher the anticipated achieved throughput. This is reasonable considering that most of the delay overhead is attributed to processing thus when a station "wins" the medium channel it should make sure to send as much data as possible.

Finally, apart from 600 bytes data packets where all approaches perform poorly, RR seems to have a slight but noticeable disadvantage compared to SIR and PF. More specifically in medium 1000 bytes packets RR is outperformed from SIR by 18% and from PF by 4 % respectively and for large 1400 bytes SIR exceeds RR by 8% while PF exhibits an almost equal performance.

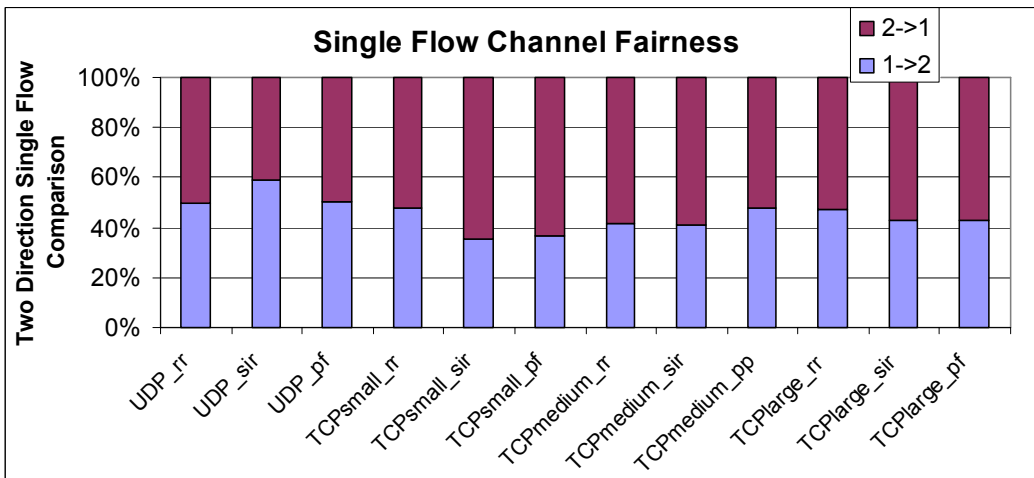


Figure 18: Sing flow Scenario Fair Channel Distribution Evaluation

From a channel distribution fairness perspective in all cases the algorithm tend to distribute the channel quite evenly among the two directions which is a positive attribute since in any case only one flow existed and therefore all bandwidth is allocated to that data flow. In cases were deviation is observed it tends to favour the 2 → 1 direction as seen in the ping application but I all cases the deviation is rather minor.

Moving on to the crossed data flow scenario which is of high importance since in this scenario two different data flows compete for the same resource (i.e. channel bandwidth) and it's interesting to investigate how that resource is distributed under different approaches.

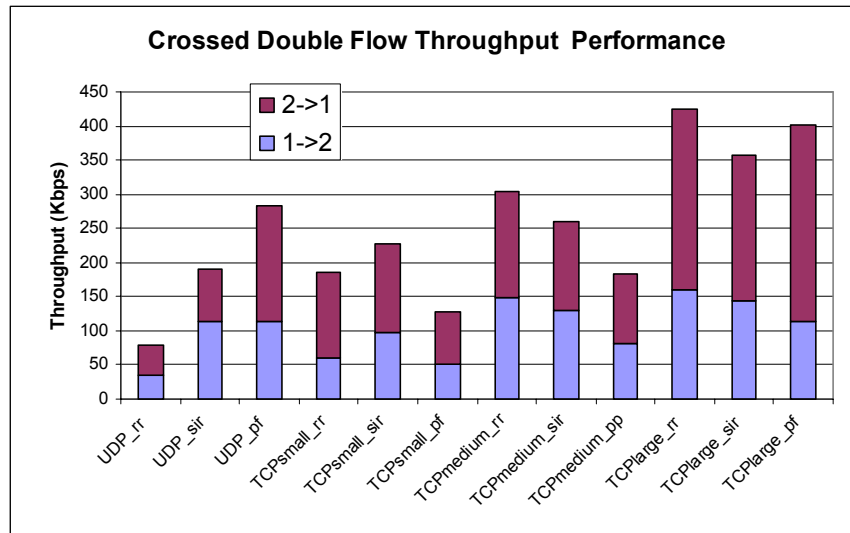


Figure 19: Two Concurrent Crossed flow Scenario Performance Evaluation

In the previous figure both the aggregated throughput of the concurrent data flows is presented as well as each flow's individual throughput.

Concerning the UDP application a rather fair channel distribution is depicted among both directions once again. On one hand this behaviour can be attributed to the fact that channel link qualities are the same for both directions. On the other hand, SIR and PF relying on channel quality metric, which is the equal for both directions, enforce the observed behaviour. Another important observation is that SIR more than doubles the aggregated achieved throughput while PF exhibits a 33% higher aggregated throughput compared to SIR respectively.

Focusing one TCP application it is depicted that the previously mentioned more or less equal channel distribution is still apparent due to the controlling effect of TCP protocol. Furthermore, and enforcing observations made earlier, it is clear that the larger packets lead to higher measured throughput under the approach that each node must take full advantage of any data packet transmitting opportunity since the transmitting delay is only a small proportion of the actual delay detected. On the other hand no secure conclusions can be drawn comparing the three distributing approaches concerning the measured aggregated throughput since both links are of equal quality. After all, this aspect of the algorithm has been well studied and evaluated from the theoretical and simulation point of view. Overall, however, it can be notes that RR produced the steadier and high performance followed by SIR and finally PF which, as show, exhibiting significantly lower throughput for small and medium packet and only for high packet came between the other two approaches.

Concerning fair distribution of transmission channel the following figure presents the proportion of the aggregated throughput each flow represents. Firstly it is shown that UDP manages to provide fair

distribution of bandwidth. In all cases exhibit deviation, no more than 10% that is quite acceptable considering that the measurements concern a real system well random and unforeseeable events occur. Secondly, in the majority of TCP cases the controlling effect of the transport layer protocol managed to provide a rather fair distribution of channel bandwidth since apart from 2 cases the maximum deviation from the optimum 50/50 was by 10% leading to 60/40 bandwidth distribution again in favour of 2 → 1 direction. The highest deviation from 50/50 channel distribution is observed for 1400 byte packet under PF approach with a distribution equal to 70/30. The above observations show a robust system able to distribute the channel fairly among two identical data flows.

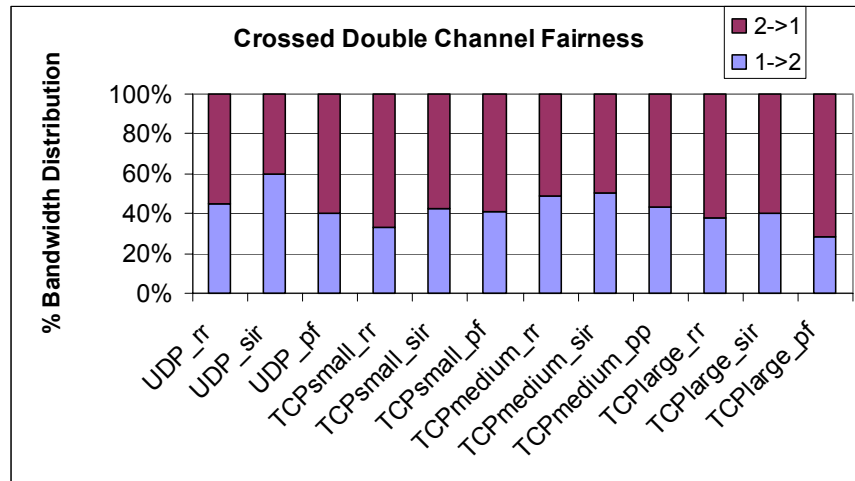


Figure 20: Two Concurrent Crossed Scenario Fair Channel Distribution Evaluation

Now moving on to 3 station setup the main focus lays on trying to evaluate the overhead imposed by the addition of the extra station considering real traffic. The first set of measurements presented in Figure 21 concern the round trip times of the standard ping application. The results are analogous to the 2 station setup and respective experiments.

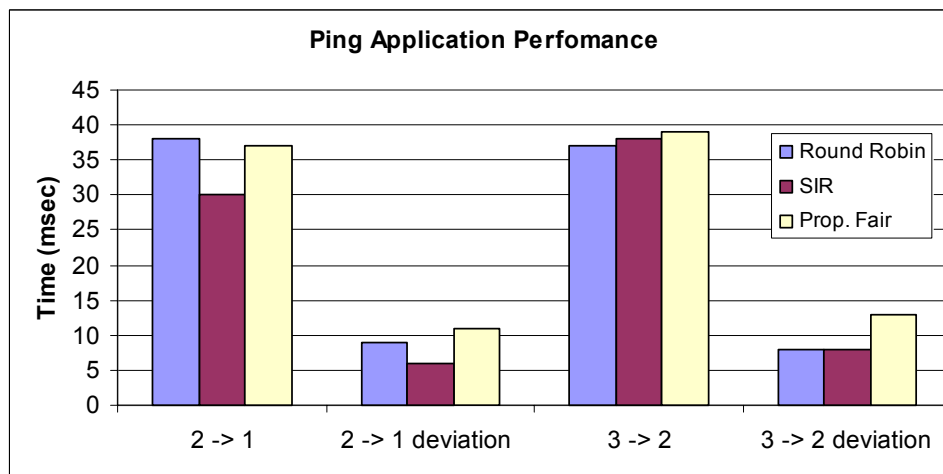


Figure 21: Ping application considering 3 station setup

Indeed only a marginal increase in the order of 5msec can be identified, attributed to the increased number of mini-slots. On the other hand the measured deviation again is in the same range as in the 2 station setup enforcing the indication in the control related measurements, pointing out that the algorithm can accept more stations participating without significant delay overhead for the data transfer.

Useful insights are also provided by the UDP measurements presented in Figure 22 combined with UDP measurements already presented concerning 2 station setup.

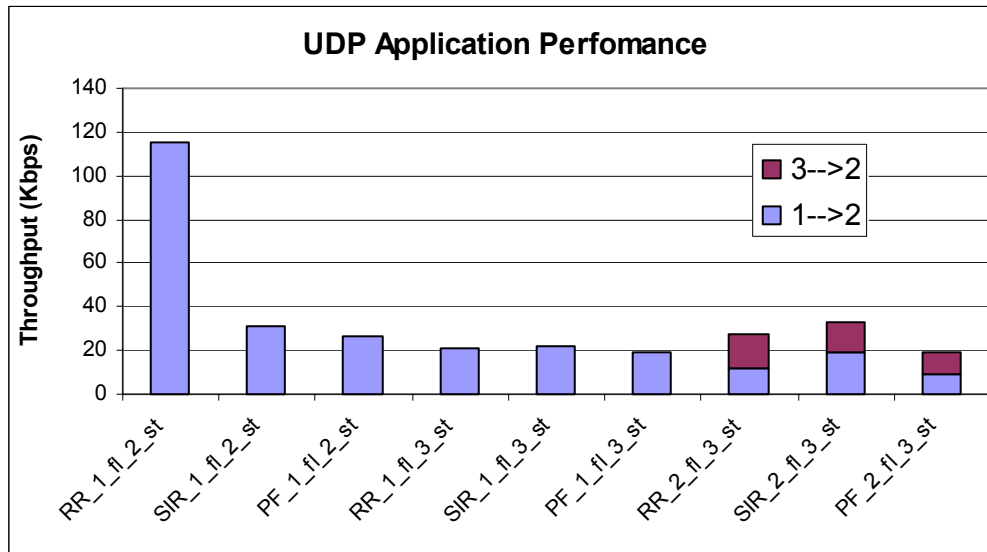


Figure 22: UDP application performance for 2 and 3 station setup

Before going into the presentation some guidelines about decoding the column naming is in order. The first infix refers to the distribution approach used; the second indicates the number of concurrent data flows followed by the fl (flow) indication. Then the fourth concerns the number of stations participating in the experiment followed by the indication st (station).

Apart the first case were, for round robin approach, a single data flow and 2 station setup, the network performed substantially better than all other cases the following observation are of high significance. Increasing the number of participating stations while retaining the same number of flows leads to a measurable but rather minor decrease in measured throughput. But, the interesting observation concerns the reaction of the network when increasing the number of flows along the number of participating stations. For such a case as the three right hand side columns depict, compared to the three middle columns, the measured throughput is actually slightly increased indicating a more efficient utilization of bandwidth without, at the same time, negatively affecting the fair distribution of the bandwidth. This is quite advantageous for the system under testing since it exhibits a robust and deterministic behavior.

Finally the overhead from a station addition is evaluated considering TCP application and a single flow so as in all cases the network tries to allocate all available bandwidth to that flow and there are no uncertainties from the bandwidth distribution approach. This estimation is attempted through measurements presented in Figure 23 where once again the controlling effect of TCP protocol substantially improves network performance compared to UDP traffic in all cases. Furthermore, measurements presented previously are enforced, indicating significant performance improvement moving from small to medium and large data packet sizes. Last but not least a significant throughput increase for the 3 station setup is observed for medium data packets ranging from 16% up to 36% followed from a significantly lower increase for large packet in the range of 3% to 9%. This difference can be attributed to link quality allocated on the traffic link combined with the fact that due to smaller packet size, more packets must be transmitted to send the same amount of data. Thus, a higher quality link is more likely to exhibit higher performance when more data packet transmission events must be handled.

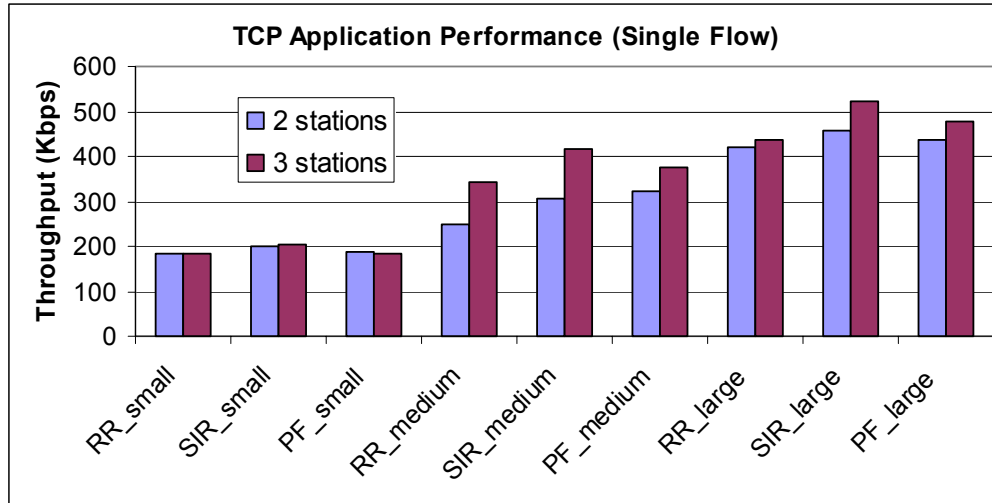


Figure 23: Station addition overhead effect estimation considering TCP applications

3.1.3 Comparison with Simulations and Performance Conclusions

3.1.3.1 PHY demonstrator

The verification of the PHY demonstrator performance was carried out relatively to the simulation results obtained with the simulation pipelines. As it was explained above the PHY demonstrator operation includes “off-line” calculation of the equivalent SISO channels for the given MIMO channel characteristics and after that “on-line” execution of the hardware emulation of the SISO OFDMA communication links. So the “off-line” procedures are reused between the software simulator and hardware implementation of the PHY demonstrator and the goal was to verify the performance of the hardware link level emulator (HLL) only.

To accomplish the verification of the HLL performance a number of tests have been carried out to test the Bit Error Rate (BER) accuracy of the HLL versus floating point software simulations.

For example, Figure 24 shows simulations for Additive White Gaussian Noise (AWGN) channel model using different Modulation and Coding Schemes (MCSs) – QPSK $\frac{1}{2}$, 16-QAM $\frac{3}{4}$ and 64-QAM $\frac{2}{3}$. The BER results for each SNR point were simulated by averaging over 10000 frames with each frame having 1 KB length.

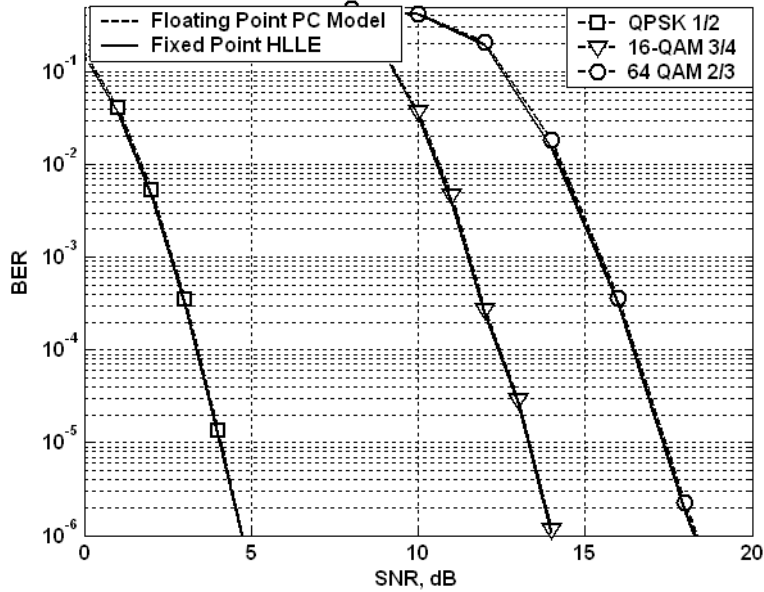


Figure 24: BER results for HLLE and reference floating point model in AWGN channel

It can be seen that for all simulated points the HLLE results match very well the results obtained with the reference floating model. The deviation between two simulation methods is less than 0.1 dB. The typical accuracy of the PHY abstraction models is known to be about 1 dB. So, the application of the HLLE may significantly improve the accuracy of the system level simulation results.

Considering the BER (or equivalent packet error rate or block error rate) performance characteristics the developed PHY demonstrator has even some advantages over traditional software simulations. Typically the BER characteristics are measured with the software simulations up to the values of about 10^{-6} . The measurement of the lower values of the BER is usually prohibitive because of the exponential increase in simulation time required to obtain the next level of accuracy in the logarithmic BER scale.

The high simulation speed of the HLLE (about 200-250 times relatively to the software simulations [MalKhorLom]) allows estimation of the BER characteristics for significantly lower values like 10^{-9} and below. One potential application of the HLLE may be the investigation of the performance error correction codes at the very low bit error rates. For example, Figure 25 shows the BER curves obtained by the HLLE for QPSK1/2 and 16QAM1/2 modulation and coding sets with the HLLE for BER values of to 10^{-9} values. The high speed of the HLLE allows investigating such performance effects of the error correction codes including error floor at the very low BER values. The HLLE provides investigation of these effects for a variety of modulation and coding sets and channel models.

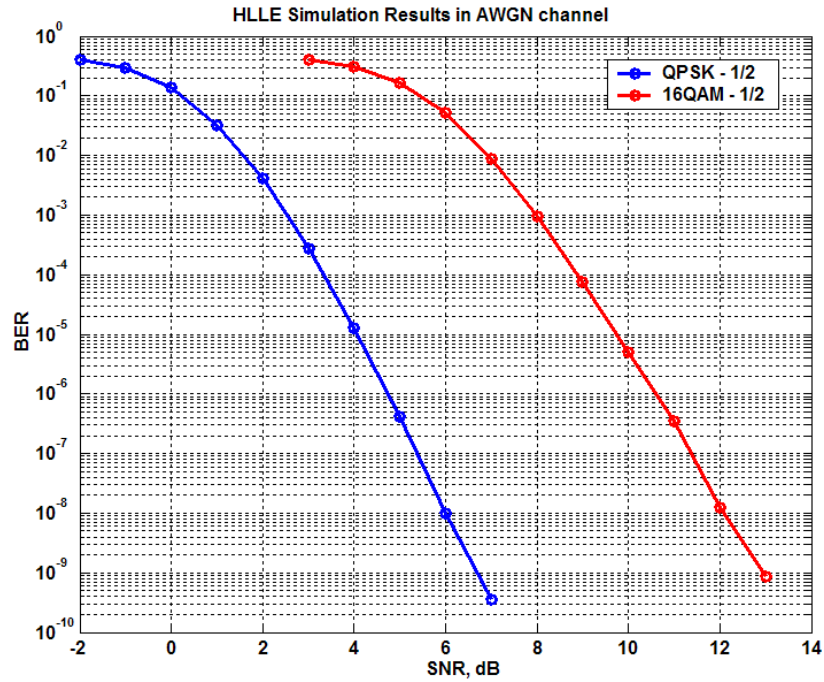


Figure 25: BER curves obtained with the HLLC

The other potential application of the HLLC may be its exploitation for the hardware acceleration of the system level simulations. Usually, the system level simulations include simulation of many parallel interfering communication links and the PHY abstraction methodology is commonly used to reduce the computational complexity of the simulations. The alternative approach to the application of the PHY abstraction may be direct simulations with the HLLC. The application of the HLLC for the system level simulations has been discussed in [MalKhorLom].

3.1.3.2 MAC demonstrator

In order to verify the operation of the MAC prototype, we compared its behavior with the simulations observed in our NS-2 MDS simulation experiments. The simulator results are summarized in the figure below (the topology and parameters are given in the scenarios section above):

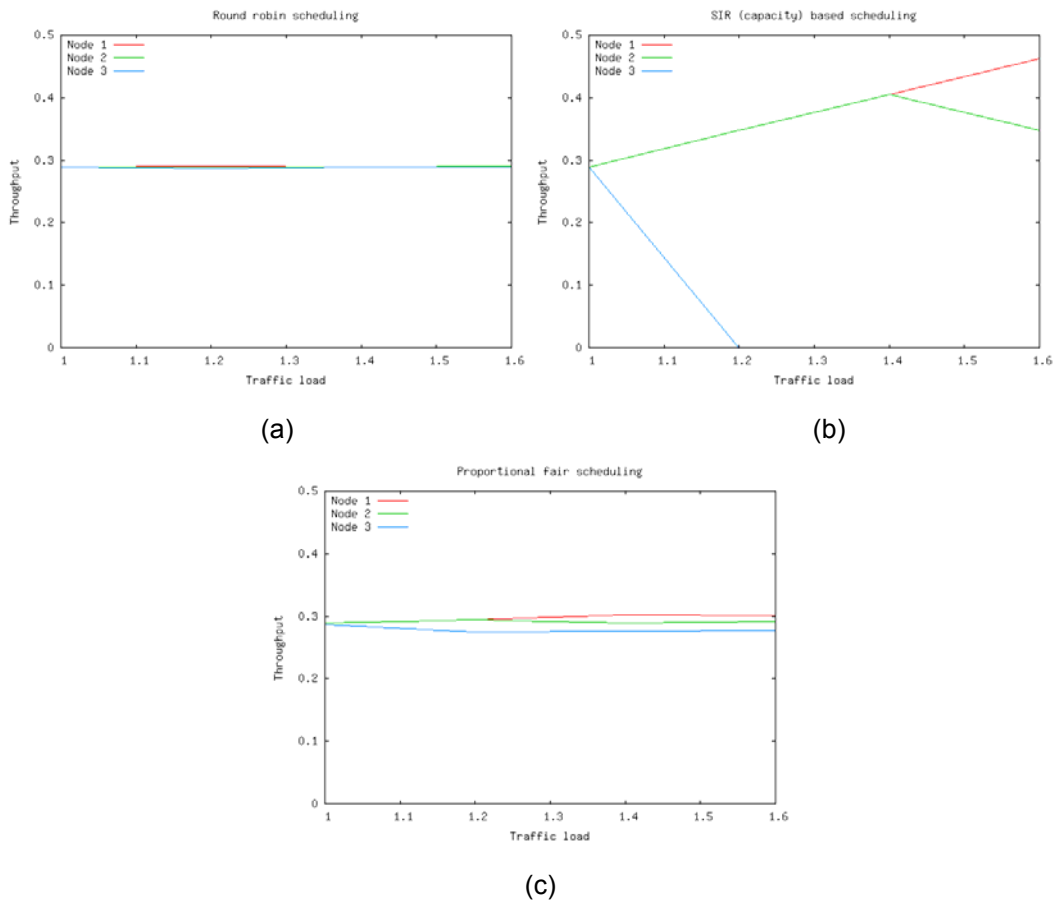


Figure 26: NS-2 Throughput Results of MDS

As shown in the concluding results of section 3.1.2 the prototype behavior follows the one predicted. Round robin scheduling yields absolutely fair allocation, capacity based scheduling effectively gives priority to the best links while proportional fair scheduling constitutes a balance between best link scheduling and fairness.

3.2 Implementation Complexity

3.2.1 PHY Demonstrator Implementation

The implementation of the PHY demonstrator included three major tasks:

1. Design of the HLLE FPGA module and embedded software for the NIOS processor
2. Implementation of the interconnection board and USB-FIFO module and assembly of the hardware platform for the PHY demonstrator;
3. Development of the software for the control PC to implement the functions of data exchange with FPGA boards, “off-line” processing functions and graphical user interface (GUI).

The most complex task for the design of the PHY demonstrator was development of the HLLE and its integration in the FPGA System-on-Chip (SoC) infrastructure.

The HLLE is designed as a System-on-Chip (SoC) component that can be easily integrated to FPGA system to allow fast evaluation of wireless communication system characteristics. The high level

diagram of developed Hardware Link Level Emulator (HLL) is shown in Figure 27. The HLL is composed of the three major signal processing blocks: transmit pipeline, channel emulator and receive pipeline. Additional blocks are the control and interface blocks that include Bit Error Rate (BER) analyzer, interface to the system bus and register file.

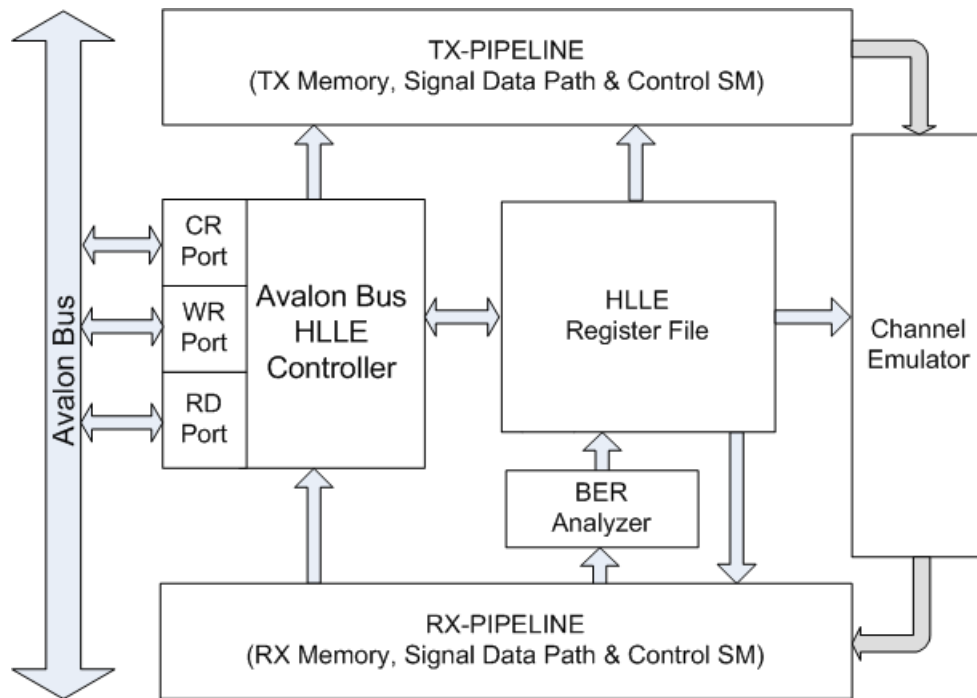


Figure 27: Architecture of hardware link level emulator

The HLL was designed for the potential reuse in other projects and for other applications. So its implementation characteristics and implementation complexity was analyzed in details. Using the FPGA synthesis tools the characteristics of the HLL implementations were obtained for Altera Stratix, Stratix II and Stratix III FPGA families. Table 7 shows the results of the logical synthesis obtained for these devices.

It can be seen from Table 7 that the HLL requires from 15K to 17K of equivalent logic elements depending on the FPGA implementation technology.

Table 7: Characteristics of FPGA Implementation of HLL

Synthesis results	Stratix I	Stratix II	Stratix III
Max clock frequency, MHz	60	140	132
Max throughput, Mbps	60	140	132
Number of equivalent logic elements (LEs) occupied by HLL	15300	17000	17260
Memory size required for HLL, Kbits	214	214	401
Number of DSP blocks used by HLL	19	19	19

The implementation of the embedded software for the NIOS processor, design of the USB-FIFO module and interconnection board and development of the software for the control PC were standard

engineering tasks which were required to implement the HLLC concept. The detailed description of these designs may be found in [MEM D5.2.2].

3.2.2 MAC Demonstrator Implementation

3.2.2.1 Implementation Components

This section presents the software components required to implement the first phase of the MEMBRANE demonstrator composed by a number of stations connected on a common switch emulating the wireless transmission medium. Figure 28 is an extension of Figure 6 (Phase I) that includes the modified network stack running on standard Linux OS laptops, as well as, the general framework required by the MDS algorithm.

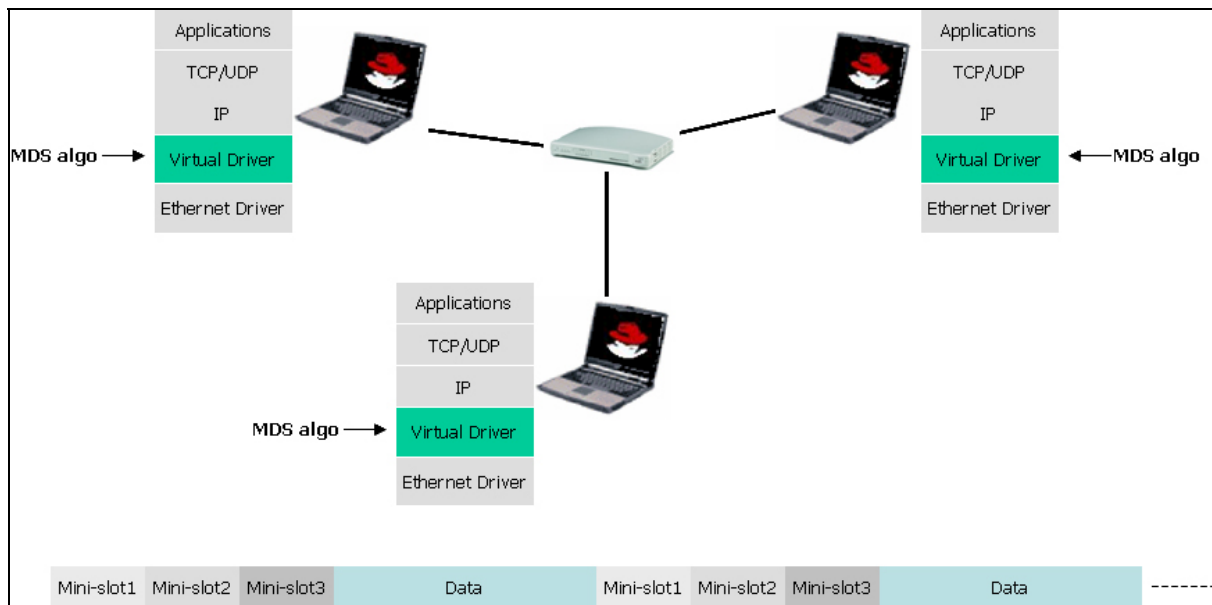


Figure 28: Modified network stack incorporating MDS algorithm & respective packet framework

As depicted, the main component incorporating all required MDS functionality is the novel virtual driver layer. This component can be a stand alone module, mounded at run time over the actual Ethernet driver. Another approach can be to directly manipulate the Ethernet driver which also can be mounded and un-mounded during run time. In either case, the main feature is that the node can capture a packet received by the PHY or just before it is passed to the NIC for transmission and freely manipulates it. Furthermore, it is possible to create specific custom packets and schedule them for transmission. Through these capabilities, we are able to create the required control packets implementing the Mini-slots depicted in Figure 28. On the other hand, we can manipulate outgoing and incoming packets by adding and stripping respectively MEMBRANE custom headers to/from the payload and add/remove MEMBRANE specific packet types. Three different types of packets are defined as follows:

- Type 0xAA04: MEMBRANE control packet representing a single mini-slot
- Type 0xAA05: MEMBRANE data packet indication. When a packet arrives for transmission, its 2 Ethernet type bytes are replaced by these values so that the receiving node can handle it accordingly.

- Type 0xAA06: MEMBRANE idle data slot indication. Due to the nature of MDS distributed algorithm, if a station does not have a packet send, it still transmits an idle packet for synchronization reasons as it is going to be presented later on.

As can be easily imagined, packet arrivals from upper layer and packet transmission to the medium through the presented sequence have totally independent frequencies. Therefore, some buffering is required. This is achieved by employing a MEMBRANE specific data packet queue through that all packet traverse before actually been transmitted to the medium.

Since the MAC demonstrator implements a distributed network and algorithm, no central control station exists to maintain synchronization. Therefore, the time synchronized MDS algorithm presented in [MEM D4.2.1] has been modified to an event based one. The events triggering each new phase of the algorithm are the packet transmissions and receptions in combination with MAC header examination. Details will be provided in the next section were the actual implementation of the various MDS phases are going to be analyzed. The common knowledge that all stations must share in order to implement the event based MDS algorithm is a table containing all stations' MAC addresses and an index informing each stations the position of its own address in the table.

Finally, the approach followed to implement the different link quality scenarios is based on specifically defined error vectors for each link.

In the context of the second phase, the main objective was to port the MDS algorithm on a hardware platform following IEEE 802.16e-like specifications [802.16e-2005]. The main configuration is the support of three nodes, each one able to communicate with both others as indicated in Figure 29.

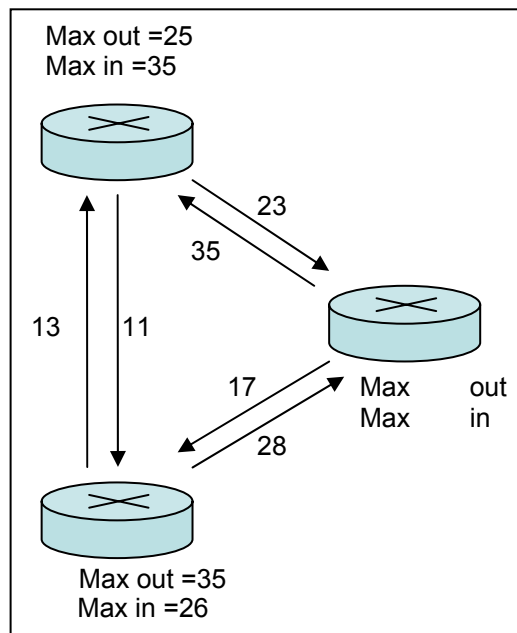


Figure 29: Primary connection forming requirement by IEEE 802.16e based platform.

Considering that, one of stations would be a BS and another one a SS, the third node should incorporate significant capabilities since it should be able to receive from a BS, send to a SS in the context of PMP operational mode. To facilitate such constraint, the novel type of MEMBRANE node namely “Intermediate Node (IN)” is comprised by a typical SS station interconnected through an Ethernet wire-line connection with a typical BS station able to both receive from a BS though the SS part and transmit to the SS through the BS part (see Figure 30). However, since the SS is associated to the BS part of the IN it cannot simultaneously be associated to the BS station. Therefore and in order

to avoid major modifications of the standard specifications, a novel approach is introduced which allows data concerning BS-SS communication to be relayed through the IN station with some small delay overhead. Figure 30 presents how the required communication links are provided by the MEMBRANE IEEE 802.16e [802.16e-2005] based platform though the passive wireless medium emulator card.

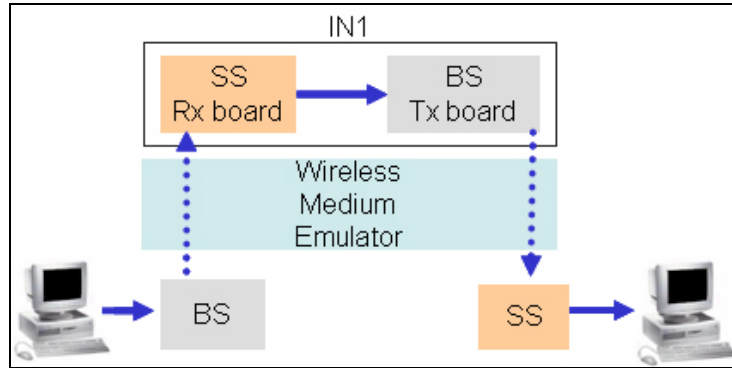


Figure 30: MEMBRANE IEEE 802.16e connection conceptual diagram

Besides the introduction of the intermediate node type and stations interconnection, a critical component of the MAC demonstrator implementation is the correlation of each MEMBRANE communication phase to the downlink and uplink phases of the involved stations especially considering that BS1-SS2 (see Figure 31) communication must be traversed through the IN1 entity. Figure 31 depicts all possible communication in ascending order while Figure 32 allocates each communication in the context of the PMP operational mode.

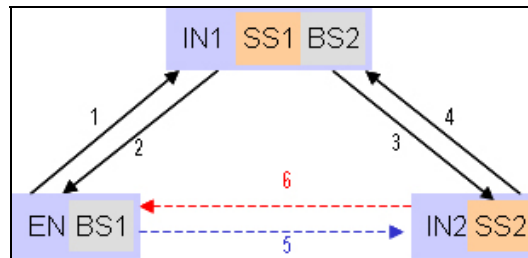


Figure 31: Communication links between stations

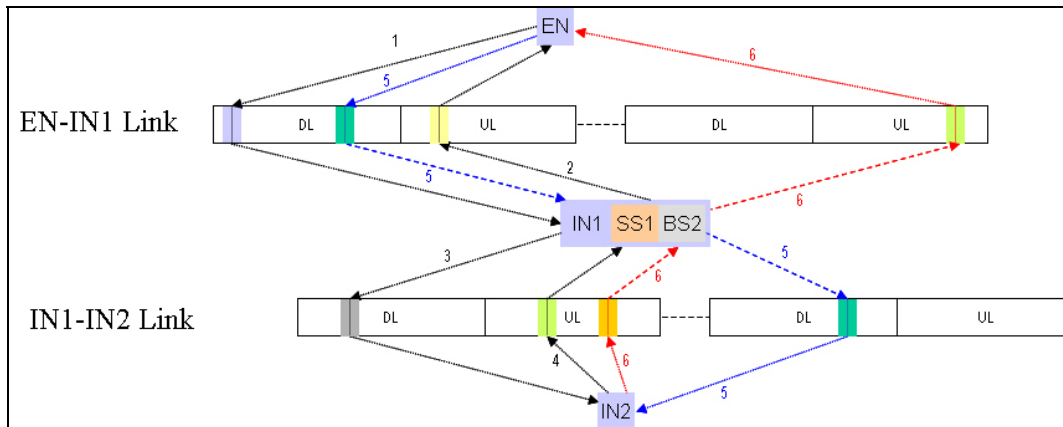


Figure 32: Correlation of communication links to PMP phases

As depicted EN-IN1 and IN1-IN2 occurs without modification in the standard PMP frame structure. However EN-IN2 constitutes special cases since they must relayed through the intermediate IN1. As a result communication 5 may be one frame apart from communication 6. Other than that, all communications are scheduled according to standard PMP mode.

3.2.2.2 Device Driver

Probably the most important advantage of Linux based OSs is that all their internal codes are open to modifications without any restrictions. However, the Linux kernel as a whole remains an extremely complex construction practically restricted to a small number of programmers that can examine, understand and efficiently modify or extend. Therefore, in order to easily, efficiently and with minimum risk, modify particular software components, programmers require an entry point. This “gateway” is provided by the device drivers. They are software components that make a particular hardware respond to a well defined internal programming interface. They hide completely the details of how the device works, which is a critical feature since the same device can be handled differently by different drivers based on the programmer’s skills or user particular demands. Another important feature is that they can be built, plugged and un-plugged to the kernel at run-time. Such software component controls the network card of any computer system. Thus, our work in the first MAC demonstrator implementation phase was to extend a popular network device driver, so that the network communication is made in the context of an even based version of the MDS algorithm introduced by MEMBRANE project. This section’s objective is to provide details concerning these extensions.

Following the above brief description of the device driver’s role in general, this section will present the additional implemented functionality due to the MDS algorithm. This functionality can be broke down to five main parts as follows:

- Handling a packet received from upper layers
- Handling a packet received from the PHY (i.e. network card interface)
- Handling a packet in a data phase
- Control packet creation
- Driver initialization

For each of these parts, a detailed flow chart is provided with a short description of the main points. It must be pointed out that references to “first” or “last” stations correspond to the entry that each MAC address occupied in the common MAC address matrix all stations share. Furthermore, triggering events are implemented by appropriate timer functions to control the triggering delay, which could be valuable to the system robustness as well as the network workload imposed based on respective requirements.

Finally, presented flowcharts focus only on the additional and newly introduced functionality due to the MDS algorithm and not on the already provided features of the considered network driver. Additionally, it may be useful to point out that a MAC address requires 6 bytes, that the Ethernet packet type field 2 bytes while the membrane header 3 bytes.

3.2.2.2.1 Handling a packet received from upper layers

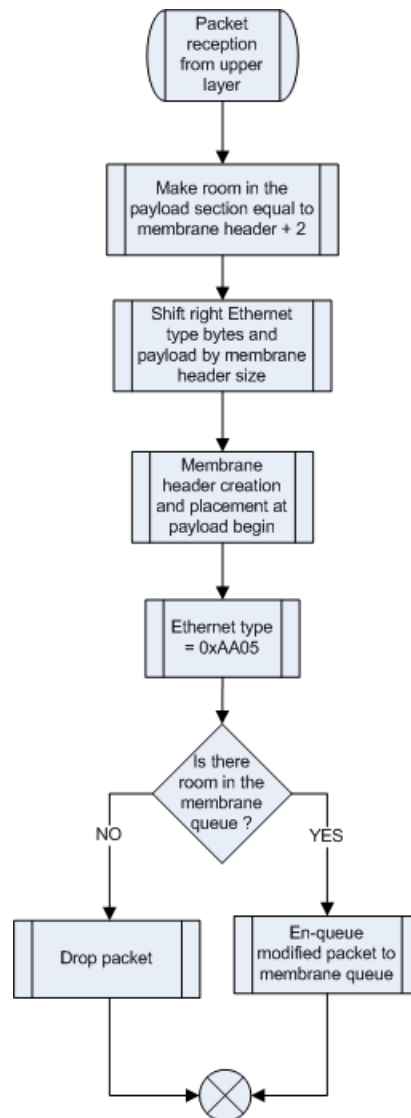


Figure 33: Procedure followed when a packet is received from upper layers to be transmitted

As presented in Figure 33, when a packet is received from upper layers, the main actions include the creation of the membrane header, creating enough space for the header, stamping the packet with the MEMBRANE custom DATA type without losing the original type which is shifted inside the payload and finally, if adequate space is available in the MEMBRANE queue, store it for future transmission when possible considering the queue follows a FIFO approach.

3.2.2.2.2 Handling a packet received from the PHY (i.e. network card interface)

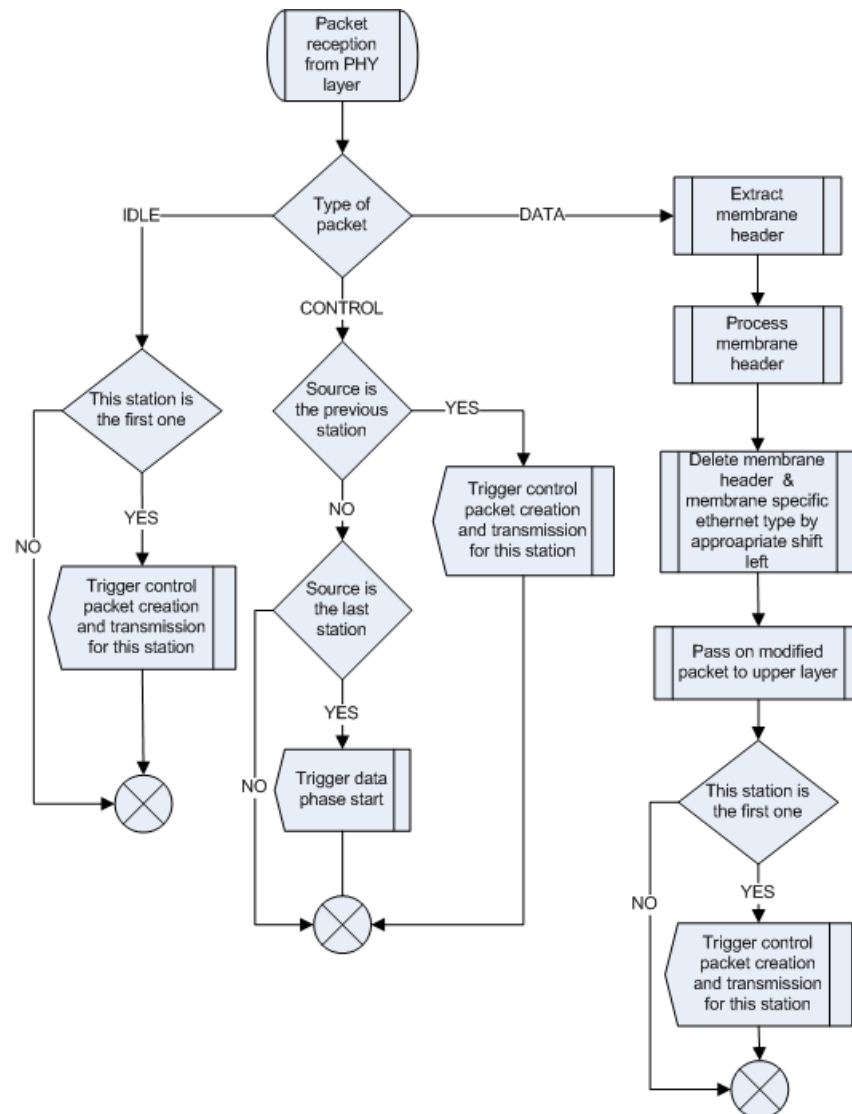


Figure 34: Procedures followed when a packet is received from network interface card

As it is clearly shown, this is a critical part of the MDS algorithm implementation since the robustness of the packet sequence depends considerably on it. According to the MEMBRANE stamping approach three types can be received.

If a CONTROL packet is received, the station checks if the source is the previous entry in the MAC address table. If that is true, the station schedules the transmission of its own mini-slot control packet and returns. If the decision is false, the only condition the station should check if whether the source is the last entry in the matrix table. Such an event triggers the commencing of the data phase.

If a DATA packet is received, the first action of the protocol is to strip the packet of MEMBRANE specific headers and call the appropriate processing functions. Afterwards, the original structure of the packet (when created at the source application layer) is restored and is passed on to the upper layers. As a last check, if the receiving station is the first one on MAC address table, it is its responsibility to commence the next sequence of mini-slots by triggering the control phase. Finally, if the received packet is an empty packet transmitted for synchronization reasons, the only check to be made is

whether the receiving station is the first one and therefore should commence the next control sequence.

3.2.2.2.3 Handling a packet in a data phase

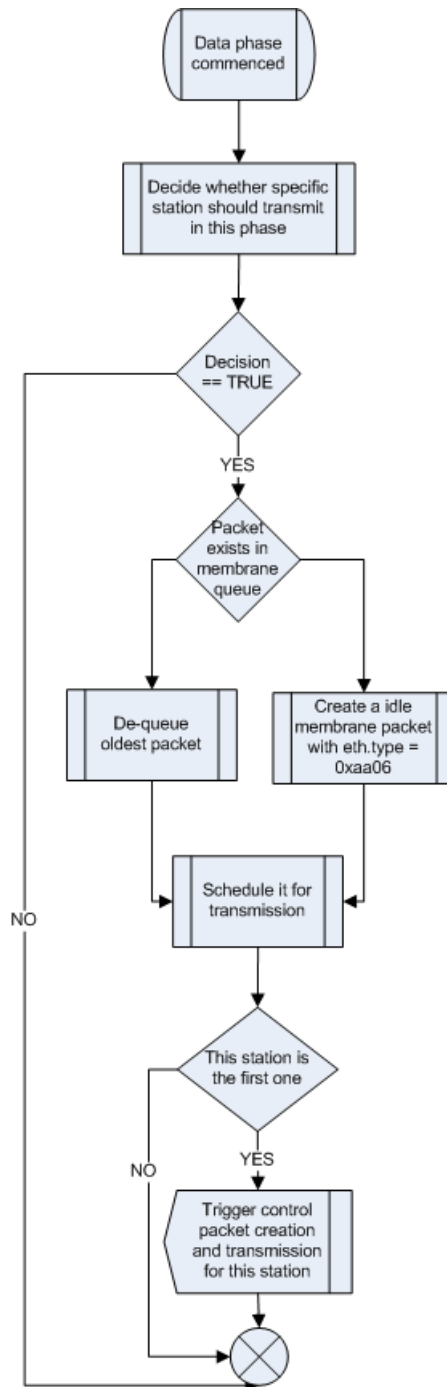


Figure 35: Procedure to be followed when data phase is triggered

The data phase commences and is triggered when the last of involved stations transmits its control packet. The first responsibility of each station when the data phase begins is to determine whether it is the one that is allowed to transmit during this data phase. The MDS distributed algorithm guarantees

that only one station will decide to transmit avoiding collisions. The station that “wins” the data phase checks whether there is a packet in the queue waiting to be transmitted. If that is the case, the oldest one is de-queued and scheduled for transmission. On the other hand, if the queue is empty, a specific IDLE packet is structured and scheduled for transmission. As a final action, the transmitting node should be checked whether it is the first entry in the MAC address table and if that is the case the next control phase is triggered.

3.2.2.2.4 Control packet creation

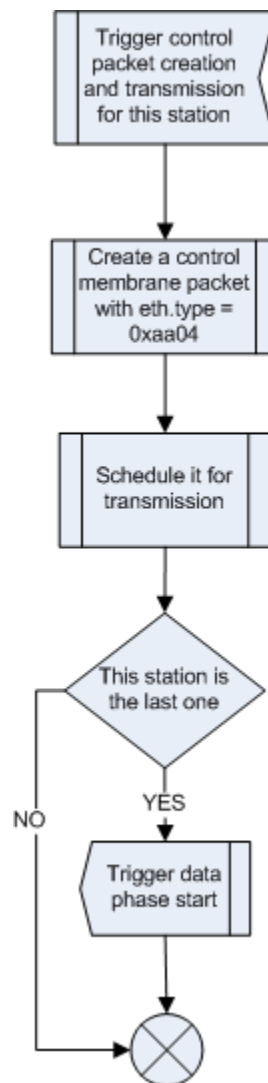


Figure 36: Procedure to be followed when control mini-slot is triggered

When the control phase is triggered (mini-slot according to the MEMBRANE terminology), the only responsibility of the station is to schedule the respective packet with the appropriate stamping for transmission and check whether it is the last station in the table since that would be the triggering event for the data phase of the particular station.

3.2.2.2.5 Driver initialization

Concerning the additional functionality during the initialization of the driver the only check to be made is whether the initializing station is the first occupant of the MAC address table. If that is true, the node should commence the MEMBRANE algorithm by triggering the first control phase as depicted in Figure 37.

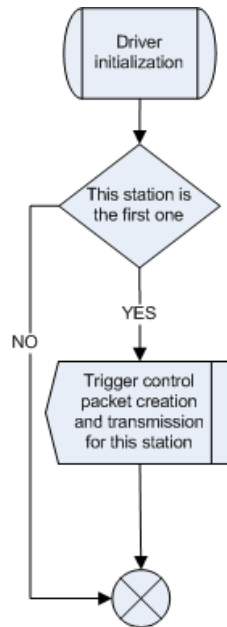


Figure 37: Additional checks to be made during network driver initialization

Summarizing, through a well specified additional software components, a typical Ethernet driver is modified to operate according to an event based version of the MDS algorithm. This approach imposes minimum delay overhead due to the network stack while offering adequate bandwidth in order to accurately validate MDS functionality, possible network performance enhancements as well as the algorithms limitations. Due to the event based implementation of the MDS algorithm, which is TDD in nature, there are cases where synchronization can be lost. To prevent the network getting in trapped situations a generic approach is followed. A general watchdog clock is running in the station holding the first entry of the MAC address table monitoring the activity of the MDS algorithm through monitoring the packet being exchanged. If the network is perceived idle for more than a specific interval, the network is assumed to be in a trapped situation and to exit from it, the control phase of the first station is triggered to re-initialize the packet sequence. This is an important parameter of the implementation since it controls the trade-off between immediate response and overreaction leading to an unstable system.

4 RECOMMENDATIONS FOR THE DESIGN OF MEMBRANE NODES

From the performance and complexity analysis of the two MEMBRANE prototypes described in the previous section, a certain number of recommendations on future design of MEMBRANE nodes can be issued. In this section, we assume that the MEMBRANE system is capable to operate in two distinct modes supporting either the DSA algorithm or the MDS protocols. The recommendations for the wireless backhaul nodes are deduced focusing, in particular, on the implementation aspects related to the exploitation of the investigated algorithms in a MEMBRANE system.

4.1 MEMBRANE System using DSA

The design and evaluation of the MEMBRANE PHY demonstrator performed by the MEMBRANE team allow to provide recommendations regarding the design of the MEMBRANE nodes using Data Splitting Algorithm (DSA) in the real broadband backhaul network.

Currently design of the broadband wireless network is based on the well-established technologies like IEEE 802.16-2004 Fixed WiMAX or IEEE 802.11n Wi-Fi adjusted for the operation over long distances. Both the mesh and point-to-multipoint network topologies are used by the networks based on these technologies.

It can be seen from the description provided above that application of the DSA algorithm requires no modifications relatively to the traditional Wi-Fi or WiMAX OFDM and OFDMA systems except for the implementation of the dedicated algorithm for antenna weight vectors calculation and their application to different transmitted data streams.

The most recent WiMAX and Wi-Fi standards (IEEE 802.11n and IEEE 802.16e standards respectively) include support of multiple antenna techniques such as MIMO and SDMA modes of transmission, which allow spatial multiplexing of data streams from one or multiple users. Most of the the MIMO detection schemes which are used in 802.11n and 802.16e transceivers are based on Zero-Forcing (ZF) or Minimum Mean Square Error (MMSE) linear algorithms, because of their efficiency and relatively low computational complexity. Also many available commercial transceivers now include support of the Singular Value Decomposition (SVD) calculation for the implementation of the closed loop MIMO systems. The efficient hardware implementations for the open loop and closed loop MIMO systems are known now. Some work on this subject of the efficient realization of the MIMO algorithms using QR decomposition and CORDIC algorithm has been also performed within the MEMBRANE project [MalPestMasl].

The implementation of the DSA algorithm requires calculation of the antenna weight vectors and their application to the transmit and receive antenna systems. The application of the antenna weight vectors is a standard procedure of the MIMO transceiver. The calculation of the antenna weight vectors done by the DSA uses SVD but the SVD is not applied directly to the channel matrices but to the orthogonal subspaces defined for different users also with the help of the SVD. So the SVD hardware used for the closed loop MIMO processing may be also reused for the DSA algorithm calculations.

So the proposed DSA investigated with the MEMBRANE PHY demonstrator may be seen as a very effective algorithm for the practical implementation in the real wireless backhaul systems. The implementation of the DSA does not require development of any dedicated signal processing blocks but instead only requires the modification of the signal processing functions available in the modern wireless communication transceivers.

4.2 MEMBRANE System using MDS

The one-round MEMBRANE Distributed Scheduler (MDS) implemented in the prototype has been evaluated. The capability of the algorithm to distribute the scheduling decision and processing load among nodes of the network, its performances and its response behaviour under different conditions has been assessed. The MDS protocol has demonstrate a relatively robust operational mode not easily influenced by synchronization errors and easily overcoming deadlock situations. A first recommendation for MEMBRANE system concerning the MDS algorithm is that it is preferred to be used in relatively subset of the network where no End-Node (EN) is present. Near an EN the flow tends to have a single direction. In that case, the algorithm fails to provide fairness to the nodes and becomes obsolete. Hence, it is preferable to exploit MDS in a meshed network composed only of intermediate nodes (IN).

From the measurements and the performance results obtained by the prototype, we have deduced that the effect of adding more nodes in the algorithm process does not seriously affect the network operational mode or robustness. In fact, the load of additional nodes can be mitigated by a judicious selection of the length/duration of the data part. As a recommendation for MEMBRANE nodes, besides the implementation of different options of large data sequences, thoughtful processing that avoids interruption procedure should be considered. Additionally, relative experiment showed that the size of the control packets exchanged only marginally affect the operational mode of MDS. Thus, if and when additional control information is required, it can be conveyed with relatively minor effect on the network operation.

From the higher layer and application point of view, we have noticed that the MDS algorithm behaves better when the higher layer protocol is connection oriented and uses some sort of congestion control mechanism such as TCP. In UDP mode, the algorithm does not provide any fairness to the nodes. Hence, it is recommended to exploit the advantages of distributed scheduling by joining the MDS scheduling decisions with routing and congestion control rules. Furthermore, it is clearly depicted that the algorithm's performance benefits substantially from large packets as opposed to constrained. Thus, it is recommended to use the maximum size of data packets that the network supports.

In a competing flow environment, the MDS algorithm seems to provide additional throughput in SIR and PF mode. The RR version of the algorithm however seems steadier in capacity independently of the length of the data packet used. Therefore, combining QoS differentiation schemes among data flows and available or configurable link qualities between specific stations, different approaches are advocated.

In contrast, when no competing stations exist the MDS algorithm, independently to the selected approach, manages to provide the available bandwidth where it is required.

From the complexity analysis and overhead evaluation, it is shown that the approach of implementing the algorithm close to the hardware (i.e. between the PHY and MAC layers as device driver or kernel module) as opposed to an application layer approach seems to be the most efficient choice. It is recommended to install the scheduling procedure in the centre of the control entities of the nodes and to assign a relatively high priority to this process. By judicious triggering mechanisms and memory management, the processing time can be reduced which is the main reason of extra delays produced by the MDS algorithm.

The distributed scheduler selected for implementation is the one-round version of the algorithm where each node is aware only of information concerning its immediate neighbour. As it is mentioned in deliverable [MEM D4.2.2], the efficiency provided by the two-round scheduler is much higher since each node participating in the algorithm obtains the link utilities functions of its one-hop and two-hop neighbours. We have seen that the length of the control packet seems not to affect the duration of the slots. Hence, judicious usage of the mini-slots to include the second hop utilities information might not

affect the performances of the scheduler. In particular, it is recommended to insert the information about the second ring of neighbours in a compress manner to minimize the length of the mini-slots. Also, the form of the information contained in mini-slots must be organized in such manner to minimize possible interruptions in the scheduling process of the algorithm.

Finally, from memory point of view depending in the scalability requirements different recommendations can be made. The footprint of the code is actually relatively small, taking advantage of elements already existing in the kernel. Therefore, if the number of stations remain relatively small, no excess additional memory is required in contrast to large networks where the maintained vectors and matrixes may require additional memory. However, and furthermore, additional memory can always be proven valuable when matrix/vectors manipulation is expected.

5 CONCLUSIONS

In order to assess the viability of the techniques and research ideas produced within the MEMBRANE project, a prototyping activity has been carried out. The objective of WP5.2 is to demonstrate the implementation feasibility of a subset of the algorithms and protocols investigated within MEMBRANE. In fact, the final aims of the workpackage are to demonstrate, on one hand, that the techniques proposed for the enhancement of wireless backhaul networks can effectively reach the performances promised and, on the other hand, that the algorithms can be relatively easily be implemented on available custom hardware with performances unaffected from the integration simplifications. To this end, two demonstrator paths have been followed to provide a proof-of-concept of the algorithms implemented. The objectives have been to demonstrate the compensation of the wireless environment effects and the provision of large throughput to the backhaul network under mesh topology. Hence, two algorithms have been implemented on two distinct hardware prototype platforms. The performances and the implementation complexity of these implementations are described in this document.

In this document, a brief overview of the algorithms selected for implementation is given followed by a short description of the platforms used. The Data Splitting Algorithm (DSA) has been selected to demonstrate the compensation of the wireless medium effects while the one-round MEMBRANE Distributed Scheduler (MDS) shows efficiency in capacity provision to the backhaul meshed network. From the performance results of the DSA, it can be seen, as expected, that the algorithm performs its best when the relaying link have high SNR values and when the direct transmission link is deteriorated. Then, the algorithm take advantage of its knowledge of the wireless environment to redirect the transmission path and increase significantly the overall capacity of the system. Similarly, the performance results of the MDS show that more fairness can be provided to a meshed topology where routing and congestion control mechanisms are combined to finalize the scheduling decisions. In addition, it is preferable to opt for various configurations with different and large data package sizes depending on the transport layer protocol. The MDS also is preferred for a non-centralized – i.e. meshed – uniform topology where every node has similar bandwidth requests. From a complexity point of view, the implementation of both algorithms has been relatively simple and it seems to require very little modifications to transmission and reception paths of currently available communication systems. The implementations of the algorithms are affecting only specific modules of the communication system. Also the implementation complexity of the novel modules and their hardware requirements such as memory, are very little making their integration into existing system relatively easy. Hence, both these techniques are very good candidates for future wireless backhaul networks based on MEMBRANE technology.

Finally, the prototyping activity of MEMBRANE has been concluded with success. The two prototypes have been developed with particular difficulties and the two algorithms have been easily integrated into the demonstrators. The development of these prototypes has offered us the opportunity to effectively evaluate the difficulties to be challenged for the production of the nodes of the MEMBRANE system. We have been able to assess the implementation difficulties of the algorithms and the advantage they can provide when applied over existing telecom systems. These prototypes are currently working independently and are capable to provide services and offer solutions to different topologies and propagation conditions of a wireless backhaul network. An integration of both these prototype into one, that is capable to gain from different environments and take the maximum of both algorithms can be considered as a future step for the MEMBRANE prototyping activity. Future work on the MEMBRANE prototypes can also be the investigation of enhancements to the prototypes with the integration of add-ons capabilities to the selected algorithms.

TERMS AND ACRONYMS

AN	Access Node
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BS	Base Station
CDF	Cumulative Distribution Function
DSA	Data Splitting Algorithm
DSP	Digital Signal Processing
EN	End Node
FIFO	First In First Out
FPGA	Field Programmable Gate Array
HLLE	High-speed Hardware Link Level Emulator
ID	Identity
IN	Intermediate Node
IP	Internet Protocol
GUI	Graphic User Interface
LOS	Line-Of-Sight
MAC	Media Access Control
MB	Master Board
MCS	Modulation and Coding Scheme
MDS	MEMBRANE Distributed Scheduler
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Square Error
NLOS	Non Line-Of-Sight
OFDMA	Orthogonal Frequency Division Multiple Access
OS	Operating System
PER	Packet Error Rate
PF	Proportional Fair
PHY	Physical
PMP	Point-to-MultiPoint
QoS	Quality of Service
RR	Round Robin
RX	Receiver

SB	Secondary/Slave Board
SIR	Signal-to-Interference Ratio
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
SS	Subscriber Station
SVD	Singular Value Decomposition
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TX	Transmitter
UDP	User Datagram Protocol
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
ZF	Zero Forcing

REFERENCES

- [MEM DoW] IST-MEMBRANE, Description of Work, "Multi-Element Multihop Backhaul Reconfigurable Antenna Network", IST-4-027310.
- [MEM D2.1] IST-MEMBRANE, Deliverable 2.1, "Most relevant community/market needs and target deployment scenarios for wireless backhaul multi-element multihop networks".
- [MEM D4.1.1] IST-MEMBRANE, Deliverable 4.1.1, "Reconfigurable IA/MIMO transceiver algorithms".
- [MEM D4.1.2] IST-MEMBRANE, Deliverable 4.1.2, "*IA/MIMO-enabled Techniques for Reconfigurable Transmission and Routing*", July 2007.
- [MEM D4.2.1] IST-MEMBRANE, Deliverable 4.2.1, "*Joint Routing, Scheduling and Power Control Algorithms*", December 2006.
- [MEM D4.2.2] IST-MEMBRANE, Deliverable 4.2.2, "*Performance studies of the new routing, scheduling and power control algorithms by simulation and/or mathematical analysis*", December 2007.
- [802.16e-2005] IEEE 802.16e-2005, "Part 16: Air Interface for Fixed Broadband Wireless Access Systems – Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile operation in Licensed Bands and Corrigendum 1", IEEE Standards Department, New York, IEEE Std 802.16e-2005, Feb. 2006. (Available online at: <http://www.ieee.org>)
- [MEM D5.2.2] IST-MEMBRANE, Deliverable 5.2.2, "Report on the HW Demonstrator (Prototype)"
- [MalKhorLom] A. Maltsev, A. Khoryaev, A. Lomayev, R. Maslennikov, M. Shilov, V. Pestretsov, A. Sevastyanov "Hardware Link Level Emulator for System Level Simulations of WiMAX-like Systems" proceedings of the ICT Mobile Summit 2008, June 2008, Stockholm, Sweden.
- [MalPestMasl] A. Maltsev, V. Pestretsov, R. Maslennikov, A. Khoryaev, "Triangular Systolic Array with Reduced Latency for QR-decomposition of Complex Matrices," 2006 IEEE International Symposium on Circuits and Systems (ISCAS 2006), 21-24 May 2006, Island of Kos, Greece.