IST-4-027310 MEMBRANE

Deliverable D5.2.1

Demonstrator Platform Description & Demonstration Scenarios

Contractual Date of Delivery to the CEC:	December 31st, 2006
Actual Date of Delivery to the CEC:	December 29th, 2006
Author(s):	A. Valkanas[1], S. Perissakis[1], A. Maltsev[2], R. Maslennikov[2], A. Khoryaev[2], A. Alexiou[3], K. Yu[3], G. Paltenghi[4], G. Panza[4]
Participant(s):	ICOM[1], INTEL[2], LUCENT[3], CEFRIEL[4]
Workpackage:	WP5.2
Est. person months:	10
Security:	Public
Nature:	R
Version:	1.0
Total number of pages:	62

Abstract:

D5.2.1 will deal first with the analysis of the scenarios defined by WP2 in order to define the demonstration scenarios, applications and testing procedures for the prototype. An overall diagram of the prototype setup will be provided along with a description of its elements. Moreover, the implementation strategy for the selected functionality will be described. The main parameters of the setup enabling the evaluation of the prototype will be given

Keyword list: Prototype, Platform, Demonstrator, Demonstration Scenarios, Implementation

EXECUTIVE SUMMARY

The basic objective of the MEMBRANE project is to evaluate design trade-offs in the development and implementation of multi-element multi-hop wireless networks. Hence, within the MEMBRANE project a hardware demonstrator is developed that targets to provide a proof-of-concept testbed for the evaluation of gains achievable by some of the proposed novel multi-antenna link algorithms, routing and scheduling protocols investigated.

In this document, the framework describing the work to be carried out to develop the hardware prototype of MEMBRANE is defined. This deliverable makes an attempt to establish a common path that will lead to the demonstrator of MEMBRANE. The main characteristics of the hardware demonstrator are also identified.

The aim of the deliverable is two-fold. First, the demonstration scenarios are described. The proposed scenarios aim to demonstrate the efficiency of the techniques and methods addressed in MEMBRANE. A second purpose of this document is to describe the platform selected for demonstration. The capabilities of the hardware elements chosen and how they are set to demonstrate the MEMBRANE system are presented.

In the first part of the document, the demonstration scenarios are described. From work completed in MEMBRANE, two large groups of possible propagation scenarios are defined for the wireless backhaul network, namely the rural and the urban scenarios. Several sub-scenarios exist for each of these large groups. For the prototype, only two of the most relevant and interesting sub-scenarios are selected for demonstration purposes. One sub-scenario is taken out of each group of propagation conditions. For each of these scenarios, two demonstration methods are identified. They represent two different approaches to determine the wireless multi-hop backhaul IP network optimisation of the MEMBRANE proposal.

The algorithms and protocols that have already been investigated within MEMBRANE are presented. These are the ones that are most eligible to implementation and are capable to provide the demonstration of the concepts tackled in MEMBRANE. From the adopted algorithms, some basic system requirements of the hardware platform are produces.

In the second part of the deliverable, the platform of the MEMBRANE demonstrator is presented. The prototype platform is composed of a set of boards which represent a small subset of the wireless backhaul network. Each board corresponds to a different node of the MEMBRANE system satisfying different roles and functionalities. Particular attention is also brought to the connection between the nodes. A block diagram of each board of the platform is deduced. The mode of operations of the boards and the platform, in general, is also tackled.

The final part of the deliverable is dedicated to establish a common framework of cooperation and collaboration among the partners involved in the project to the development of the hardware prototype. The tasks needed and their duration in order to develop and implement the prototype are identified and described. The methodologies to follow in order to test and validate the algorithms and blocks developed within this activity of the project are presented.

TABLE OF CONTENTS

EXECUTIVE SUMMARY 2
LIST OF FIGURES 4
LIST OF TABLES 5
1 INTRODUCTION 6
2 DEMONSTRATION SCENARIOS 7
2.1 Scenarios Description 7
2.1.1 Propagation scenarios 7
2.1.2 Demonstration scenarios 8
2.2 Description of the Proof-of-Concept Procedures 9
2.2.1 Algorithms and Protocols 9
2.2.2 System Requirements for the Prototype 14
3 PROTOTYPE DESCRIPTION 18
3.1 Prototype Elements 18
3.1.1 Prototype Nodes Description 18
3.1.2 Connectivity 19
3.1.3 Applications 21
3.2 Prototype Diagram 22
3.3 Prototype Platform Description 23
3.4 Operation Mode 24
4 IMPLEMENTATION STRATEGY 25
4.1 Collaboration Framework among Partners 25
4.1.1 Work Planning 25
4.1.2 Exchanging Procedure among Partners 28
4.2 Design, Development and Integration Phases 28
4.3 Testing and Validation Procedures 30
4.3.1 Test-bed prototyping 30
4.3.2 Testing methodology 31
4.3.3 Test specifications and validation 32
5 CONCLUSIONS 35
TERMS AND ACRONYMS 36
REFERENCES 38
ANNEX I – Traffic models 41
1. Statistic Traffic modelling in IP networks 41

LIST OF FIGURES

Figure 1: Block diagram of the MEMBRANE prototype 18
Figure 2: Block diagram of channel emulation for MEMBRANE prototype at TX side 20
Figure 3: Block diagram of channel emulation for MEMBRANE prototype at RX side 20
Figure 4: Block diagram of MEMBRANE Prototype 22
Figure 5: DSP development kit, Stratix II Edition from Altera 23
Figure 6: Pert Diagram of figure 8 of the “Description of Work” of MEMBRANE 26
Figure 7: Gantt Chart 28
Figure 8: General testing methodology schema (simulation modelling case) 31
Figure 9: Voice activity detector behaviour. 46
Figure 10: Talk-spurt and pause durations. 48
Figure 11: Cumulative distribution functions. 49
Figure 12: Bit Rate for a One-Layer H,261 codec. 52
Figure 13: Shape of one layer codec probability density function 53
Figure 14: PDF for the Two Layer Codec 54
Figure 15: Document size distribution 56
Figure 16: Comparison of Unix file sizes with WWW documents sizes 56
Figure 17: Average number of requests for file sizes 57
Figure 18: Requests for document by document size 57
Figure 19: Zipf's law applied to WWW documents 58
Figure 20: Trace of Fractional Gaussian Noise 61
Figure 21: Self-Similarity behaviour of the aggregated traffic 62

LIST OF TABLES

Table 1: WiMAX application classes 21
Table 2: Measured speech related temporal parameters. 48
Table 3: Video source characteristics. 53
Table 4: Transition probability matrix for the one layer 54

1 INTRODUCTION

The purpose of MEMBRANE is to design an efficient and adaptive wireless backhaul network. The development and the deployment of a wireless backhaul network capable to support the end users demands in high speed access and the service provider demands of an economically attractive solution requires several technological breakthroughs. The needs of such efficient network concern the provision of quality of service, the capability of reconfiguration of the network, the resolution of heterogeneity aspects, the provision of ubiquitous services or deployment and the openness to novel standards. The above requirements can be met with the development of novel multi-antenna link signalling, routing, scheduling and power control algorithms that target the optimisation of wireless backhaul networks. In order to evaluate and provide a proof-of-concept of the algorithms and techniques developed within MEMBRANE, a system simulation software platform and a hardware prototype are developed.

This document is the first deliverable of workpackage WP5.2 of MEMBRANE, where a hardware demonstrator is developed implementing some of the state-of-the-art techniques investigated within MEMBRANE.

The deliverable is mainly composed of three parts. The first part concerns the description of the demonstration scenario that is selected to provide the proof-of-concept of the MEMBRANE system. In this part the most suitable scenarios selected for the demonstration are described. Then the techniques and procedures that will demonstrate the comparative advantage of MEMBRANE are presented.

The second part of the document is dedicated to the description of the platform of the MEMBRANE demonstrator. This includes the description of all the nodes forming the platform, their connectivity, a block diagram of the platform and the operational mode.

The last section of the deliverable is dedicated to the description of the collaboration framework between partners. This part is important in the sense that it establishes the interaction between partners coming from different countries, with different methodologies, technical knowledge and interests.

2 DEMONSTRATION SCENARIOS

As described in Deliverable 2.1 and 4.1.1, various transmission scenarios have been defined. The defined target scenarios are categorized into two large groups, namely the rural scenarios and the urban scenarios. Within each group, several subgroups are further defined according to different antenna heights at both ends. Across WP4, different algorithms and protocols are investigated from the physical layer up to the transport layer, in order to improve the link and system performance. In the physical layer, algorithms using multiple antennas are studied to exploit the diversity gain and/or multiplexing gain. Furthermore, transmission schemes for multi-node scenarios are also investigated. In WP4.1, the routing and scheduling algorithms are studied, in WP4.2, joint routing and power control algorithms are examined while the transport-layer protocols are investigated in WP4.3.

In this section, we briefly mention the target scenarios defined previously and describe the two most interesting propagation and demonstration scenarios that will be implemented in WP5.2 in more details. Moreover, we propose some selected algorithms and protocols to be demonstrated in the prototype. We also discuss the minimum system requirements that need to be supported by the demonstrator.

2.1 Scenarios Description

There are two different sorts of scenarios needed for the implementation of a prototype capable to demonstrate the optimization algorithms of the wireless multihop network investigated in MEMBRANE. These two sets of scenarios establish the proof-of-concept testbed for evaluation of the algorithms considered within this project. First, the selected propagation rural and urban scenarios described in D2.1 and D4.1.1 determine the propagation environments of interest by the demonstrator. These target scenarios serve as link between the work conducted in WP2 and WP4, and the simulators and demonstrator to be developed in WP5. More specifically, the demonstrator is dimensioned following specific propagation conditions already established in WP4, and hence the specifications of the algorithms and protocols deduced in WP4 can be directly applied to the prototype. Secondly, the demonstration scenarios followed in order to establish the achievable gains of the algorithms studied in MEMBRANE are described in the second part of this subsection.

2.1.1 Propagation scenarios

In MEMBRANE project, two large groups of scenarios have been defined for the wireless backhaul networks, namely the rural scenarios and the urban scenarios. For the rural scenarios, there are two sub-scenarios, the planned infra-structure (R1) where the antennas mounted high at all nodes, and the unplanned infra-structure (R2) where the antennas at the access node are at a moderate height. For the urban scenario, three sub-scenarios are defined, namely rooftop – rooftop (U1), below-rooftop – below-rooftop (U2) and rooftop – below-rooftop (U3). More details on these scenarios can be found in [D4.1.1] and [D2.1].

In order to demonstrate the effectiveness of the proposed algorithms and protocols, the two most interesting sub-scenarios are selected, one from each large group. For the Rural scenarios, sub-scenario R1 is selected. For the Urban scenarios, the below-rooftop – below-rooftop U2 sub-scenario is used in the demonstration. These two scenarios represent two typical scenarios, the LOS scenario with relatively small angular spread and long transmission distance, and the LOS/NLOS scenarios with relatively large angular spread and short transmission distance. This needs to be taken into account when determining the system setup and the selected transmission schemes that are going to be implemented in the prototype. Details of these two sub-scenarios are described below.

2.1.1.1 R1 Scenario

A large number of remote small population centres are connected to the MEMBRANE mesh backhaul network. Each centre is surrounded by large agricultural areas. Population centres are quite distant from each other. Land is assumed to be relatively flat. Various vegetation patterns can be met. Essentially, the Access Nodes (ANs), End Nodes (ENs) and, Intermediate Nodes (INs) are all placed on high towers. In practice, the tower heights are selected such that the 1st Fresnel zone is clear and LOS conditions are met for the required link distance. Furthermore, the actual delay spread and angle spread due to propagation environment can be effectively reduced due to the utilization of directional antennas with high gain. This is more likely when a wide area access technology is utilized, such as UMTS and WiMAX.

For the demonstrator, the specifications and parameters described in [D4.1.1] and the references within the deliverable will be used to model the rural propagation conditions R1. The parameters for LOS are selected from the R1 propagation scenario. For this particular propagation scenario, the main purpose of the prototype is to demonstrate the capabilities of the MEMBRANE system under a relatively “solid” propagation environment such as the R1 scenario. Hence, the challenging task under the R1 scenario is concentrate to scheduling and routing algorithms developed in MEMBRANE.

2.1.1.2 U2 Scenario

The urban scenario covers urban environments, ranging from business centres to residential neighbourhoods. Traditionally, the business centres have been covered in 2G and 2.5G systems by microcells (or even picocells to improve indoor coverage). In the new technological settings, 3G nodes have to coexist with wireless LAN (802.11x), PAN (802.15x) and MAN (802.16x) access points. The residential areas have been covered in 2G and 2.5G systems by macro and microcells. There is also a growing interest to deploy *domestic nodes* or *femtonodes* to improve the coverage in the 3G system. In the mean time, wireless (WiFi, 802.11x) access points are also present in residential environments, hence we have to cope with the simultaneous presence of both kinds of nodes.

The U2 scenario treats the case where the nodes at both ends are deployed below the surrounding buildings, at heights in the order of 3-10 m. Such moderate antenna height often happens at the access nodes as well as some intermediate nodes (to reduce the deployment cost). This type of scenario covers both LOS and NLOS outdoor propagation conditions. Nodes are assumed to be deployed in a Manhattan grid fashion. For any given node, the streets can be classified as the main street where the node is located, perpendicular streets and parallel streets.

Similarly, to the previous propagation scenario, the channel models implemented in the prototype are specified and parameterized according to directions and values obtained from [D4.1.1] and reference within. For simplicity reasons again, the LOS outdoor propagation conditions will be modelled in the hardware prototype. Since U2 scenarios, represent a more challenging environment, the demonstrator will mainly show the advantages of the selected smart antenna techniques of MEMBRANE.

2.1.2 Demonstration scenarios

In the previous paragraphs, we described the two different scenarios from the environmental point of view in the sense that we basically described the propagation condition of interest for the demonstrator. In the following part, the scenarios are re-classified in terms of the methods and procedures followed to establish the proof-of-concept of the MEMBRANE system through the demonstrator.

We consider two possible demonstration scenarios that reveal the enhancements of the MEMBRANE techniques investigated throughout the project. Each of these scenarios is oriented to one specific layer of the prototype and targets to demonstrate the advantages of the MEMBRANE techniques for the specific layer. Hence, we identify a PHY scenario and a MAC scenario. It is important to notice that for both scenarios the algorithms developed for both layers are involved. However, our interest in the behaviour of our system for each scenario is more focused in a particular algorithm and in a particular behaviour of a specific layer of the demonstrator.

2.1.2.1 The PHY layer scenario

In this scenario, we investigate the reaction of the MEMBRANE demonstrator with respect to a sudden modification of the propagation conditions in a particular link. The demonstrator of MEMBRANE is composed of three basic nodes, i.e. one End Node (EN) and two Intermediate Nodes (IN). In this particular example, we first establish a stable transmission status for the demonstrator. This means that the transmission parameters of the system such as the modulation or the coding rate remain invariants while there are also no variations of several propagation metrics such as the Signal-to-Noise Ratio (SNR). We then, force to the demonstator a specific variation in one of the transmission links by for example modifying the interference level applied to this specific propagation link. This transmission link can either be the one between the two Ins, or one of the two EN-IN links. The target of the scenario is to demonstrate the compensation of this modification by the MEMBRANE system through the reconfiguration of the algorithms applied to the physical links. The proof-of-concept of the reconfigurable IA/MIMO transceiver algorithms selected in WP4.1 will be demonstrated. The metrics used in this scenario are physical metrics such as the Bit Error Rate (BER).

2.1.2.2 The MAC layer scenario

In this scenario, the response of the MEMBRANE system to a change of traffic demands is investigated. At first, the demonstrator is configured to operate with a relatively “light” traffic load and with relatively low constraint in QoS demands. In other words, the requests to the EN from the two INs are Best Effort traffic demands representing demands issued from a relatively small set of end users. Then, we will emulate a sudden change of traffic demands in one particular IN node. The traffic is then modified to requests for real-time applications (video) with a significant increase of number of end-users. The purpose of the scenario is to demonstrate the satisfaction of these traffic requests and this sudden burst of demand through the exploitation of advanced routing and power control techniques investigated in WP4.1 and WP4.2. In this scenario, the behaviour of the protocols implemented in the MAC layer are investigated, and hence packet losses are measured.

2.2 Description of the Proof-of-Concept Procedures

Within WP4, different functionalities have been studied across various layers to improve the link as well as system performance. In this subsection, selected algorithms and protocols are described. The algorithms selected in D4.1.1 as baseline are subject to be implemented in the prototype. The minimum system requirements, such as number and type of antennas elements, are also provided.

2.2.1 Algorithms and Protocols

2.2.1.1 Selected smart antenna technology

In this section, we briefly describe the spatial transmission schemes that are subject to be implemented in the prototype. In general, these transmission schemes can be categorized into two major groups, with and without data splitting. Here data splitting refers to the transmission schemes that split the whole data stream into different sub-steams and transmit via different IN (different routes). Finally these data will be put together at the EN.

- ***<u>Alamouti code:</u>***

The Alamouti code [Ala 98] is a special case of orthogonal space-time block codes (OSTBC) with 2 transmit antennas. It spreads the symbols in time and space in a block-by-block fashion. Moreover, it is designed based on an orthogonal structure and is especially attractive because of its low complexity.

Using linear processing, joint maximum-likelihood decoding of all the transmitted symbols decouples into symbol-by-symbol decoding. In each space-time transmission block, the transmitter sends

$$S_{\text{Alamouti}} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix}$$

where rows and columns represent transmit antennas and symbol intervals, respectively. Note that this transmission scheme assumes that the channel is constant for two consecutive transmit intervals. No Channel State Information (CSI) is required at the transmitter.

Therefore for a system with 2 receive antennas, the input and out put relationship can be written as

$$\mathbf{y}_1 = \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

$$\mathbf{y}_2 = \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \begin{bmatrix} -s_2^* \\ s_1 \end{bmatrix} + \begin{bmatrix} n_3 \\ n_4 \end{bmatrix}$$

Hence

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2^* \end{bmatrix} = \sqrt{\frac{E_s}{2}} \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \\ h_{1,2}^* & -h_{1,1}^* \\ h_{2,2}^* & -h_{2,1}^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3^* \\ n_4^* \end{bmatrix} = \sqrt{\frac{E_s}{2}} \mathbf{H}_{eff} \mathbf{s} + \mathbf{n}$$

Note that the effective channel matrix $\mathbf{H}_{eff}$ is always orthogonal, we have

$$\mathbf{r} = \mathbf{H}_{eff}^H \mathbf{y} = \sqrt{\frac{E_s}{2}} \|\mathbf{H}\|_F^2 \mathbf{I}_2 \mathbf{s} + \tilde{\mathbf{n}}$$

This means the simple Alamouti code achieves full diversity and full rate (rate 1) transmission, therefore the rate is preserved compared to the single antenna case. However, for the number of transmit antennas larger than 2, no OSTBC can achieve full rate transmission and therefore other than Alamouti code, no other OSTBC will be implemented in the prototyping.

- ***Fixed Beamforming:***

The fixed beamforming method predefines a set of antenna weight vectors, which can be generated using Butler matrix or antenna array manifold. Based on the beam selection information, the transmitter sends the information to the receiver using one selected beam that can provide best performance. This transmission scheme can be coupled with SDMA, i.e. the transmitter can transmits to several users using different beams as long as these users are well separated. Note that there exist overlap between two beams, therefore it may not be beneficial to use the neighbouring beams simultaneously.

- ***Dominant Eigenmode Transmission***

The Alamouti code does not require any CSI at the transmitter. When the CSI is known at the transmit side, it can be used to improve the system performance. One way of doing this is the so-called dominant eigenmode transmission. The main idea is to multiply the transmitted symbol with a transmit weight vector at the transmitter and multiply with a receive weight vector at the receiver, i.e.

$$r = \sqrt{\frac{E_s}{M_T}}\mathbf{gHw}s + \mathbf{n}$$

where $\mathbf{g}$ is the receive weight vector, $\mathbf{w}$ is the transmit weight vector. $\mathbf{H}$ is the $M_R \mathrm{x} M_T$ channel matrix, s is the symbol being transmitted, r is the output at the receiver and $\mathbf{n}$ is the zero mean complex Gaussian noise vector. Let the singular value decomposition (SVD) be

$$\mathbf{H} = \mathbf{UDV}^H.$$

It can be shown that the receive SNR is maximized when $\mathbf{w}/\sqrt{M_T}$ and $\mathbf{g}$ are the right and left singular vectors associated with the largest singular value $\sigma_{\max}$ of $\mathbf{H}$. By doing this, the effective input-output relationship becomes

$$r = \sqrt{E_s}\sigma_{\max} s + n.$$

The dominant eigenmode transmission provides same diversity order as the OSTBC, but with larger array gain. More details can be found in [Pau03].

- ***<u>Spatial Multiplexing (SM)</u>***

Spatial multiplexing is a technique that exploits the spatial multiplexing gain. It transmits different symbol streams at different transmit antennas, and works especially well in the rich scattering environment while multiple data streams can be supported. When CSI is available at both the transmitter and the receiver, by multiplying $\mathbf{V}$ at the transmit side and $\mathbf{U}^H$ at the receive side, we get

$$\mathbf{r} = \sqrt{\frac{E_s}{M_T}}\mathbf{DWs} + \tilde{\mathbf{n}}$$

where $\mathbf{s}$ is the transmitted symbol vector, D is a diagonal matrix with singular value of $\mathbf{H}$ on its main diagonal, and $\mathbf{r}$ is the received vector. $\tilde{\mathbf{n}}$ is the noise vector with same statistical property of $\mathbf{n}$. $\mathbf{W}$ is the diagonal waterfilling matrix that optimally allocates the power for multiple parallel SISO channels.

Since the MIMO channel has been decomposed into a set of parallel SISO channels, the techniques used for SISO channel detection can then be used at the receiver.

When the CSI is not available at the transmit side, we can still transmit different data streams at different transmit antennas. In this case, the power is equally allocated for each transmit antennas, i.e. $\mathbf{W}$ is an identity matrix and no precoding matrix $\mathbf{V}$ is used. At the receive side, different techniques, such as zero-forcing (ZF), minimum mean square error (MMSE), VBLAST-ZF, VBLAST-MMSE, can be used to estimate the transmitted symbols. The expression for the linear ZF and MMSE filters are given as

$$\mathbf{G}_{ZF} = \sqrt{\frac{M_T}{E_s}}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$$

$$\mathbf{G}_{MMSE} = \sqrt{\frac{M_T}{E_s}}(\mathbf{H}^H\mathbf{H} + \frac{M_T}{\rho}\mathbf{I}_{M_T})^{-1}\mathbf{H}^H$$

The V-BLAST architecture belongs to the non-linear space-time receivers. The main idea is to successively decode the symbol streams by peeling off layers. The symbol with best SNR is decoded first by using ZF or MMSE filtering, and its contribution is subtracted from the received signal. Then the symbol with the best SNR in the remaining signal is decoded and subtracted. The procedure ends once the whole symbol vector has been decoded. More details can be found in [Wol98] and [Pau03].

- ***Hybrid Scheme***

The OSTBC provides high diversity gain but the data rate is always equal (for 2 transmit antennas) or less than 1. While at the same time the SM provides higher data rate without diversity gain. For 4 transmit antennas, a hybrid scheme that makes a balance between diversity gain and data rates is proposed in [Zhu03]. This hybrid scheme is a combination of Alamouti code and SM. The main idea is to use Alamouti code for two pairs of transmit antennas at the same time. This means for two transmission intervals, 4 symbols will be transmitted, 2 for each pair of transmit antennas. Therefore the diversity gain is the same as Alamouti code and the data rate is 2. Since the target, at the moment, is to develop a demonstrator with 2 antennas at both sides, this algorithm serves only as a reference to a possible extension of the prototype to 4 antennas at both ends.

Let us assume there are M_R receive antennas, $\mathbf{h}_k(n)$ denotes the received vector from *k*th transmit antennas at time *n*. Assume the channel is constant from time *n* to time *n*+1, the received signal vector $\mathbf{r}$ can be expressed as

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_n \\ \mathbf{r}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1(n) & \mathbf{h}_2(n) & \mathbf{h}_3(n) & \mathbf{h}_4(n) \\ \mathbf{h}_2^*(n) & -\mathbf{h}_1^*(n) & \mathbf{h}_4^*(n) & -\mathbf{h}_3^*(n) \end{bmatrix} \mathbf{s} + \begin{bmatrix} \mathbf{n}_n \\ \mathbf{n}_{n+1} \end{bmatrix}$$

where $\mathbf{s} = [s_1 \quad s_2 \quad s_3 \quad s_4]^T$ is the transmitted symbol vector. At the receive side, a simple ZF or MMSE filter can be deployed to detect the symbols. For instance, after ZF filtering, the estimated signal vector can be written as

$$\hat{\mathbf{s}} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{r}$$

Similar results can be obtained for the MMSE receiver, see [Zhu03] for more details.

- ***Data splitting***

All the above described transmission schemes are designed for point-to-point transmission, and therefore can be used to transmit the whole data stream from one node to another node (AN-IN, IN-EN). In the situation when multiple INs are available, there exist two main approaches to transmit the data stream from the AN to the EN via INs. One approach is to use the Alamouti, SM, or hybrid scheme mentioned before to transmit the whole data stream at AN. All the INs will receive the transmitted signal. Then we can either select the best IN to relay the data to the EN, or several INs can relay the same data to the EN. This approach, however, will generate large interference to the other nodes in the backhaul network.

Another approach is to split the whole data stream into several sub-streams, and use some selected INs to relay these sub-streams to the EN respectively. We call this data splitting. There are several benefit from data splitting. First of all, by doing data splitting, the data are usually transmitted to some specific directions. This may greatly reduce the interference generated to the other existing nodes in the network. Secondly, for the MEMBRANE scenario, the AN may have the knowledge about the whole network topology. By selecting the best routes associated with different INs, the total end-to-end throughput may be increased. Thirdly, there exist tradeoff between the overall throughput and the delay in the backhaul network. Note that different applications may require different throughput and have delay tolerance. By using data splitting, we are able to use different routes (INs) to transmit data associated with different application at the same time, and therefore improve the overall network performance.

One simple technique to do data splitting is to transmit using fixed beamforming with SDMA. The beams can be generated various methods, e.g. array manifold, Butler matrix. Some taping techniques can be used to lower the interference between the neighbouring beams. The generated beams can then be used to transmit to different INs according to the higher layer routing and scheduling algorithms. To further lower the interference, we may not use the neighbouring beams, i.e. use only one every other beam to do SDMA. Other more advance algorithms include Block Diagonalization (BD), Successive Minimum Mean-Square-Error precoding (SMMSE) and other non-linear techniques. These algorithms are currently under investigation in WP4, and can be considered for implementation under reservation of time and effort availability towards the end of the demonstrator development.

2.2.1.2 Pilot assisted channel estimation

To implement the algorithms described in Section 2.2.1.1, the channel needs to be estimated at the receiver. For certain spatial transmission schemes that require CSI/CQI at the transmitter, the estimated CSI/CQI needs to be fed back. Note however, that for TDD systems, like the one developed in the MEMBRANE prototype, the feedback may not be necessary due to the channel reciprocity. Therefore, no CSI/CQI feedback is implemented in the prototype. The channel estimation is usually conducted by sending the pilot signals from the transmitter to the receiver. In general, two types of pilot signals are used in the channel estimation procedure, namely the common pilot and the dedicated pilot. Below, we briefly discuss these two types of pilot.

- Common pilots

The common pilots are usually shared and re-used among all receivers within a cell/sector. Therefore, the overall pilot overhead per user may be reduced. Common pilots are mainly used in the case where the unweighted channel transfer functions are to be estimated. When multiple transmit antennas are deployed, different common pilots can be used for each transmit antenna but they are still shared among all receivers within a cell/sector. Furthermore, common pilots can also be used to determine the gains when beamforming is used and/or to further assist scheduling process. In addition to conventional common pilots mentioned above, when fixed beamforming is used, specific common pilots can be sent for each beam and they can be shared among receivers that are covered by that beam.

- Dedicated pilots

When the user-specific signal processing is performed at the transmit side, the effective CSI is often required at the receiver. Here, the effective CSI takes into account both the fading channel transfer function and the spatial signal processing applied. In such cases, the transmitter needs to send user-specific dedicated pilots with same signal processing as the data symbols being transmitted. Note that dedicated pilots have to be orthogonal among users. One major drawback of dedicated pilots is the larger overhead since different pilots need to be sent to specific users.

The demonstrator targets to follow the specifications of the Mobile WiMAX standard as closely as possible. Hence, the specifications and locations of the pilot signals, that are planed to be used in the prototype, would not be different from the ones defined in the standard. Exceptions will be considered only to satisfy specific demands of the novel algorithms implemented.

2.2.1.3 Any other high layer protocols

2.2.1.3.1 TCP over wireless

TCP was designed as an end-to-end transport protocol for use on wired networks. It proved to be efficient and reliable in traditional networks consisting of wired links and stationary hosts which have a minimum of delays. Congestion situations due to traffic and errors (not breaks in the connection) are handled well using TCP possibly combined with other algorithms.

However, in wireless networks, links are inherently prone to errors and delays therefore packet losses frequently occur. This is problematic for Internet connections utilising TCP since this protocol interprets the packet losses as congestion and consequently reduces the transmission rate. This behaviour of TCP is not desirable since the packet losses are not caused by congestion, but link breakages, noise or spurious wireless signals. The wireless network difficulties which are producing a reduction in throughput are compounded by the behaviour of TCP which reduces further the throughput.

Therefore, problems originating at a wireless hop (e.g. a base station) such as signal breakages and long delays, result in inefficient TCP handling, which culminated in several localised solutions such as the split connection approach, the fast-retransmit approach and link level retransmissions.

In a wireless network where just the host has a radio interface, a follow-on solution called the snoop protocol improves TCP performance. The crux of Snoop is to implement a network layer solution at the base station (BS) or End Node (EN) which caches and retransmits where necessary any missed packets lost or dropped due to signal related problems.

The novel solution of TCP over wireless that is going to be conceived within the framework of the MEMBRANE project can take advantage of the analysis results about the previous proposals on the matter (e.g. Snoop protocol), but it will better adapt to the specific target multi-hop wireless scenario.

Largely, a dynamic tuning of the congestion window according to the transmission bandwidth actually available across the concerned communication path should be achieved in a robust and consistent manner. For the purpose, different approaches can be followed. Even the exploitation of information about the wireless channel status (i.e. CSI) and related to the MAC-PHY layers of the issued source terminal. An explicit congestion notification mechanism is also foreseen in order to signal congestion conditions at the IP interfaces of the concerned path.

Of course, it is fundamental to test the newly devised algorithm for validation and assessment purposes, from both a functional and a performance point of view.

Two types of prototype are going to be employed to achieve such a goal: a simulation modelling and a practical implementation in the test-bed.

Different features and functionality will be provided with each prototype. The former is likely to be quite detailed and complete of all the supported mechanisms, while the latter is more a proof-of-concept in the field, therefore just a fair implementation of the designed protocol.

However, in this deliverable of WP5 the focus is on the test-bed. Therefore, paragraph 2.2.2.1 will describe the system requirements for the testing and assessing objectives, concerning TCP over wireless for this kind of prototype.

2.2.2 System Requirements for the Prototype

- **Number of Antenna Elements:** At the moment, two antenna elements are planned to be used for each node of the prototype. All advantages of using multiple antenna algorithms (described in 2.2.1) for MEMBRANE network will be demonstrated by using two antenna elements. The number of the antenna elements is mainly limited by the complexity of the channel emulation scheme which will be implemented in the prototype. The use of two antennas for each node will allow using of the off-the-shelf prototyping platform (Altera DSP Development Kit, Stratix II Professional Edition). The implementation of nodes with larger number of antenna elements will require us to build a custom prototyping hardware platform which is very resource consuming. It is important to notice, that a solution currently considered to overcome this problem is the exploitation of a multichannel emulator such as the PROPSIM C8 Wideband Channel Emulator of Electrobit.

- **Inter-element distance:** Since the antenna emulation will be done in baseband any antenna inter-element distance can be emulated. The first candidates to consider are inter-element distances of 0.5 and 4 wavelengths.
- **Type of antenna elements:** Any type of antenna elements can be modelled by including antenna characteristics into channel impulse responses used for channel emulation.
- **Carrier frequency and System Bandwidth:** 5 GHz and 10 MHz bandwidth.
- **Number of sub carriers:** 1024 subcarriers for 10 MHz channel
- **Frame structure:** The frame structure will be in accordance with IEEE 802.16e-2005 and draft IEEE 802.16j specifications. Only TDD transmission format will be supported.
- **Transmit power:** The impact of different transmit powers will be emulated with the proposed channel emulation scheme.
- **Channel models:** the prototype will be able emulate any of the channel models considered in the MEMBRANE deliverables 4.1.1 and 5.1.1. However, basically, demonstration will focus on the propagation scenarios described in 2.1.1.
- **Required BER/FER:** The BER and FER requirements will depend on application chosen and also some other parameters used (e.g. automated repeat request (ARQ) scheme). Currently the BER equal to 10^{-6} is considered as a main target for the error rate.
- **Targeted system throughput:** The max system throughput is defined by the channel bandwidth, modulation order, code rate, spatial multiplexing order, uplink / downlink ratio and some other factors like guard interval size. Currently it is supposed to use 64-QAM modulation and ¾ code rate and two spatial channels. This would result in about 30-40 Mbps max achievable throughput for 10 MHz channel bandwidth.
- **Traffic model:** The required traffic patterns will be generated at the PC connected to End node and Access node of the prototype in accordance with specification provided in the MEMBRANE deliverable 2.1 for rural and urban target scenarios.

2.2.2.1 TCP over wireless

An optimized TCP over multi-hop wireless network is a key target for MEMBRANE project. As such, the newly conceived transport protocol must be properly tested and assessed within the related WPs. Being the test-bed a proof-of-concept, not all the new features are required to be implemented, but in the simulation modelling process, which represents the first kind of prototype that is going to be developed into the framework of the project (refer to WP4.3).

The idea is to demonstrate the effectiveness of the enhancements over typical MEMBRANE scenarios, as issued and described in the previous sections of this deliverable.

The test-bed prototype will be implemented on a common platform (i.e. either WINDOWS- or Linux-based), and will be a piece of software substituting the former TCP protocol at transport layer of the ISO OSI model. The software will be executed in the PCs connected to the EN and INs boards composing the prototype.

Of course, the prototype is going to include both the basic functionality of TCP (e.g. congestion and transmission window management, timer setting and checking, packet retransmissions, etc.) and a possibly reduced set of advances.

The critical issue of detecting actual congestion into the network, discerning it from delay and loss due to the presence of multi-wireless hops, will be certainly considered.

The matter is to properly set the congestion window, dynamically tuning it according to the actually available bandwidth across the concerned transmission path. Different approaches can be followed to achieve such a goal.

The first is by evaluating the time a packet (i.e. a set of consecutive packets for a stable and consistent behaviour) of a given size waits before being transmitted at the source side. A second method is to regulate the congestion window in accordance with the averaged delay experienced by packets (i.e. the RTT considering the corresponding ACK arrivals).

In the test-bed the former idea will be implemented, while the latter might be included as needed (in case the achieved performance is susceptible of further improvements). A well-known technique to detect congestion at IP layer, already standardized by IETF, is the Explicit Congestion Notification (ECN): a mechanism that was formerly employed in Frame Relay (data-link layer technology). With it, in case of congestion or even contingent congestion at a given IP router interface, a packet or (a set of packets) is marked (e.g. a bit in ToS field is set to 1) at that interface. Such functionality is really helpful; therefore, it will be implemented in the PCs connected to each relay node (IN) of the MEMBRANE test-bed and exploited by the end-points (in principle, just by the source TCP peer).

In theory, TCP performance can be increased leveraging on CSI (Channel State Information) and other information coming from the lower layers (i.e. MAC and PHY), such as queue status, BER, PER, etc..

Properly managing such information is not a trivial task. However, if previous studies carried-out for example by means of the simulation modelling process show that a not negligible improvement can be achieved in this way, even such functionality will be supported in the test-bed.

2.2.2.2 Traffic model

As already stated, different methods to generate traffic patterns can be deployed in the built prototypes (i.e. the test-bed and the simulation modelling). Statistical/analytical models are typically more appropriate for a simulation prototype, while real user operation and pre-recorded traces are more suitable for a test-bed.

However, analytical models constitute a helpful reference when employing software or hardware traffic generators, which have a good flexibility in terms of available statistical distributions and related parameters in order to properly generate traffic with given and well defined characteristics, i.e. for the session birth and death process, average and peak rates, burstiness, as well as length of packets for the artificially synthesized traffic patterns.

As a general consideration, it is reasonable to leverage on SW/HW generators when the traffic to be injected into the network has a simple statistical modelling or should be produced in an aggregate manner, for which the employment of a score of real users is practically unfeasible. A common example of the latter is the background traffic, whose important goal is the creation of realistic network conditions in which the end-users typically access the Internet.

In the other cases, the employment of human personnel is the best choice.

Alternatively, pre-recorded traces represent a valid substitution of both the aforementioned methods, depending on the specific context and needs.

Therefore, the following techniques are going to be used in the test-bed for the generation of each type of traffic according to the target MEMBRANE applications.

- Background traffic: HW or SW traffic generators for asynchronous data traffic (TCP-based) and pre-recorded traces for synchronous multimedia traffic (UDP-based).
- Data traffic (FTP, WWW browsing, Telnet, e-mailing, etc.): real users (possibly by means of the help of scripts that automatically generate traffic/requests).
- Voice (e.g. G.711, G.723, G.729): real users
- Audio/video (streaming or interactive, MPEG2/4 or H.26x coded): real users or pre-recorded traces. The first for interactive multimedia (e.g. audio/video calls, video conferencing, e-learning), while the latter more likely for streaming applications (e.g. video on demand).

A wide variety of tools and applications can be deployed in the test-bed. In terms of both HW and SW data generators and multimedia applications, which are well known, spread and often even free, either for WINDOWS or Linux based platforms. Furthermore, recorded traces of TCP data aggregates and video are largely available in the Internet.

The characteristics and modelling of these traffics subject to be uses in the testbed prototype are provided in the annex.

3 PROTOTYPE DESCRIPTION

In this section, the demonstrator of the MEMBRANE project is described in details. In the first part of the section, the different elements are analysed including a description of the characteristics of the nodes composing the prototype, the proposed connectivity between the nodes of the demonstrator and the applications used and/or needed for demonstration purposes. From these descriptions, a diagram of the prototype is also produced. The entire prototype platform is described in the third part of this section. Finally, the operation modes of the platform are produced.

3.1 Prototype Elements

3.1.1 Prototype Nodes Description

The prototype will support simple MEMBRANE network topology consisting of two Intermediate Nodes (IN) and End Node (EN) with arbitrary connections between them. From the two INs, one is assumed to provide connection to one or more Access Nodes (AN) while the second one is a simple wireless router of the backhaul network. The EN is supposed to have a wired connection to the backbone network. It is expected to be mounted on a relatively high tower, a fact that is reflected in the emulated channels on its two links, to the two INs. The EN is the source of downlink traffic and the sink for uplink traffic. The IN providing working also to provide access to the ANs is supposed to be serving a number of individual users and in the prototype setup it will be the source and sink of data traffic. On the other hand, the IN without AN connections is assumed to be part of the wireless backhaul network and does not generate or consume any traffic. All nodes will be connected to the PCs through the Ethernet for control and monitoring purposes. The block diagram of the proposed system is shown in Figure 1.

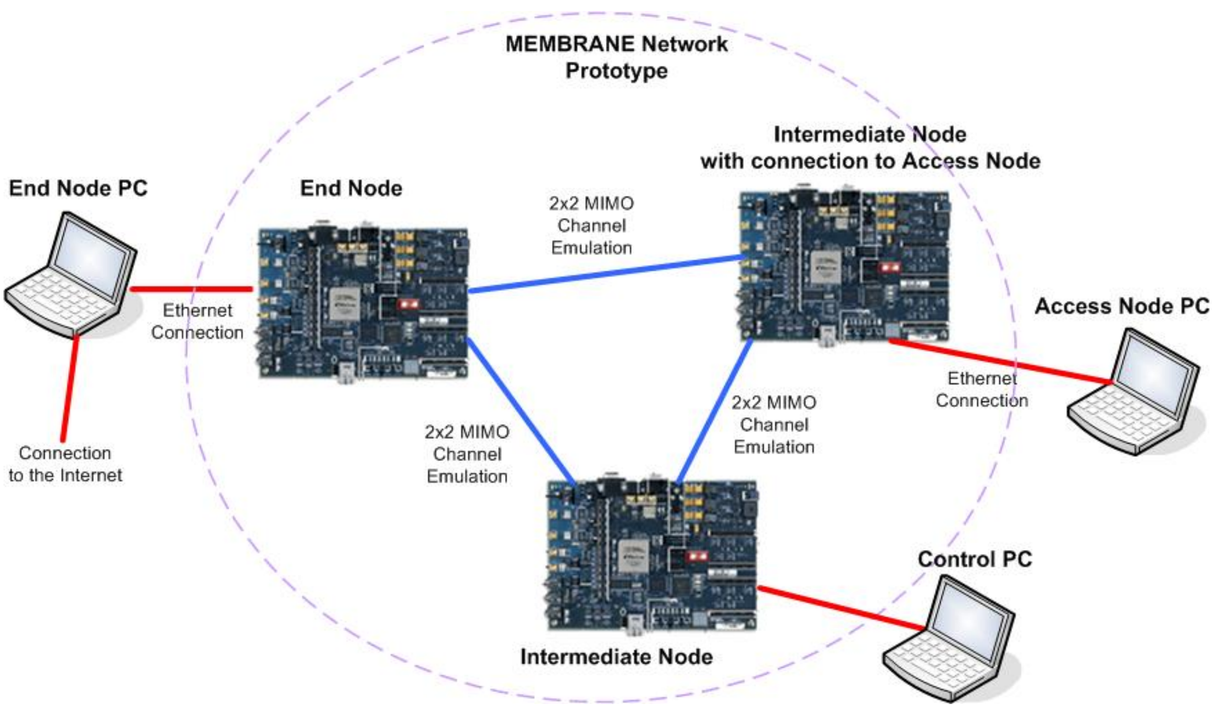


Figure 1: Block diagram of the MEMBRANE prototype

The transmission format of the system will be mainly in accordance with IEEE 802.16e standard and 802.16j specification which is being developed now. The EN node will have an enhanced functionality of the 802.16e base station. The centralized scheduling will be done at the PC connected to the EN and the schedules will be distributed by EN in the service messages. The schedules will specify the frequency-time regions to be used by the EN to transmit data to the INs and by each IN to the other IN and the EN.

The PHY layer of the each node will be able to support transmission and reception in both DL and UL mode using 802.16-2005 OFDMA specification with 2 TX and 2 RX antennas. The classes and parameters of the supported permutations (PUSC, FUSC, AMC) will be defined later based on the results of simulation. The acquisition algorithms will include detection and channel estimation schemes. The classes of the multiple antenna algorithms to be supported also described in 2.2.1, are:

- beam-switching;
- maximum ratio combining with interference cancellation;
- open-loop MIMO with space-time block codes (matrices A and B);
- spatial multiplexing in UL and DL.

The maximum channel bandwidth of the prototype will be 10 MHz which will result in about 70 Mbps maximum throughput when using two spatial channels with 64-QAM modulation.

From the network perspective all nodes will function as the Layer 3 routers. The joint routing and scheduling will be done at the EN PC and the routes/schedules will be distributed to all nodes. Also the embedded TCP/IP stack will function at every node to allow communication with PCs to exchange control information.

3.1.1.1 Use of pre-existing blocks for PHY implementation

The complete development of the 802.16e PHY implementation is a large and resource-consuming task which can not probably be done within MEMBRANE project. INTEL's MEMBRANE team may adopt pre-existing hardware PHY blocks of 802.16e-OFDMA SISO system to be used for the MEMBRANE prototyping but can provide them in the netlist form only. These modules cannot be made publicly available as they were not designed during the MEMBRANE project. The blocks that will be developed during the project (e.g. the MIMO processing blocks) will be made publicly available. All the software of the MEMBRANE prototype will be publicly available.

3.1.2 Connectivity

The channel emulation for the MEMBRANE prototype will be done by embedding it into the digital signal processing inside the FPGA following the signal processing at the TX side. The TX and RX antenna patterns for all nodes will be taken into account at the generation of channel transfer functions between the nodes. No up/down conversion at the RF frequency and over-the-air transmission will be done.

The 2x2 TX-RX antenna configuration will require 2x2 matrix connection between every two nodes. To minimize the number of the physical connections between FPGA boards implementing different nodes (one board for one node is assumed) it is planned to use totally 6 cables, i.e. each two nodes will be connected using 2 cables. One cable will be used to send all information in one direction between two nodes. The four (2x2) transmission channels between each two nodes will be multiplexed into one cable as follows. The number of physical channels will be reduced by a factor of 2 by combining signals coming from different TX antennas to one RX antenna after the convolution with the corresponding channel transfer functions (see Figure 2, "channel emulation block"). To emulate the functionality of two RX antennas by using one cable frequency multiplexing is exploited. Signals coming to different RX antennas are further transmitted through different frequency bands to one DAC, one physical cable and one ADC of the next node. It should also be noted that the proposed architecture makes I/Q splitting and combining in digital domain also. The block diagram of the proposed scheme is shown in Figure 2 and Figure 3.

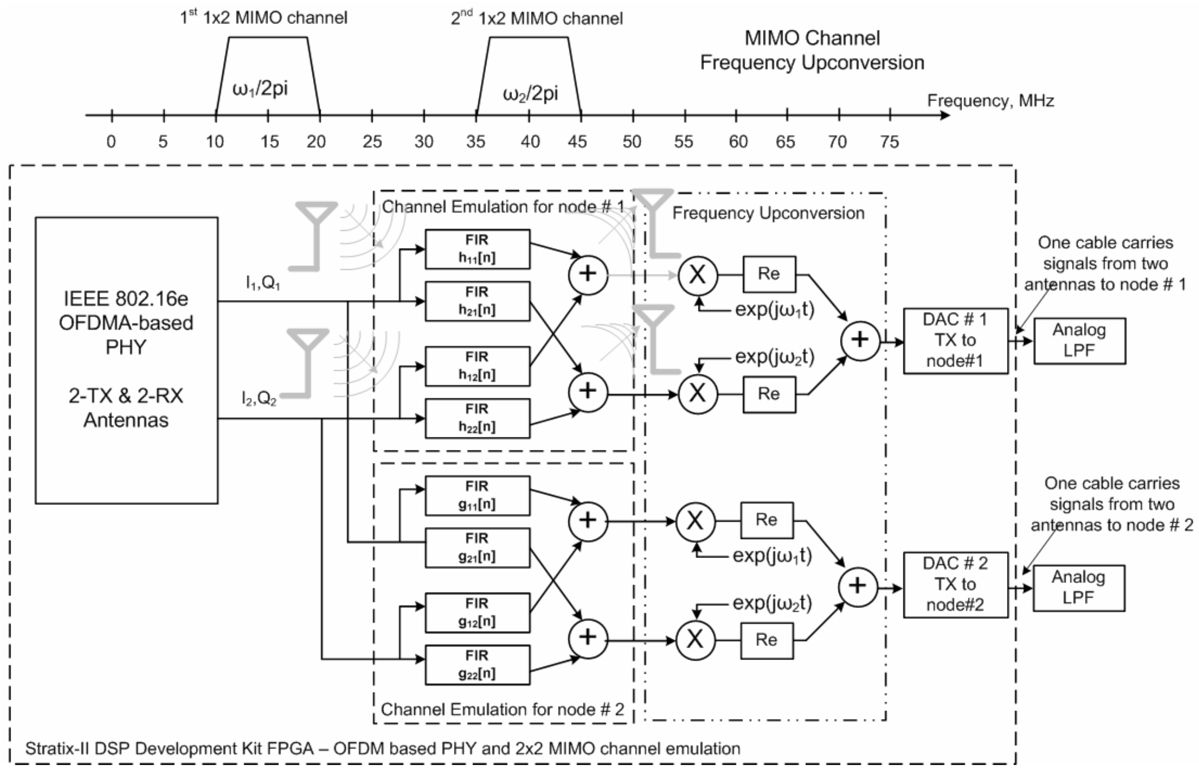


Figure 2: Block diagram of channel emulation for MEMBRANE prototype at TX side

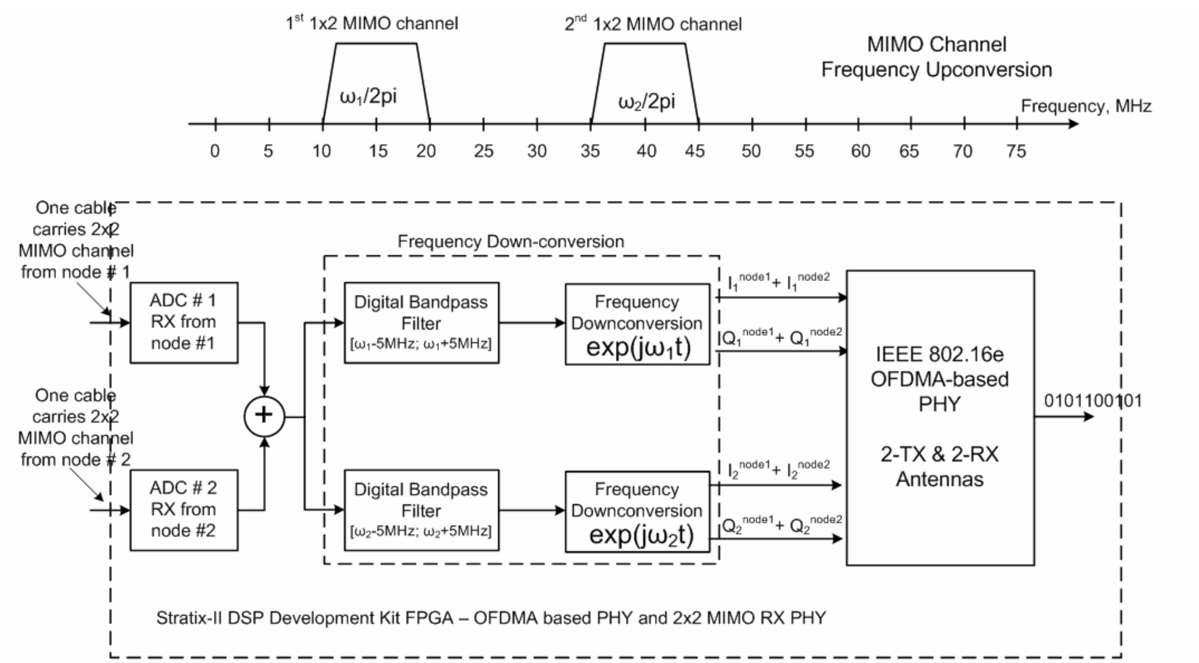


Figure 3: Block diagram of channel emulation for MEMBRANE prototype at RX side

So the proposed scheme will use only one cable to emulate 2x2 MIMO channel propagation. It should also be noted that this scheme will allow us to use off-the-shelf DSP Development Kit from Altera with 2 ADCs and 2 DACs as prototyping platform.

The nodes of the prototype will be connected to the PCs using the Fast Ethernet (100 Mbps) connections.

Also each node will have RS-232 serial port to be connected to the PC for debug purposes.

3.1.3 Applications

WiMAX applications can be categorized in 5 classes [WiMAX06], depending on the requirements they place on the basic performance metrics: bandwidth (data throughput), latency, and jitter. The following table summarizes the definitions of the 5 classes and the respective requirements.

Table 1: WiMAX application classes

Class	Application	Bandwidth Guideline		Latency Guideline		Jitter Guideline	
1	Multiplayer Interactive Gaming	Low	50 Kbps	Low	<25msec	N/A	
2	VoIP & Video conference	Low	32-64 Kbps	Low	<160msec	Low	<50msec
3	Streaming Media	Low to High	5 Kbps to 2 Mbps	N/A		Low	<100msec
4	Web Browsing & Instant Messaging	Moderate	10 Kbps to 2 Mbps	N/A		N/A	
5	Media Content Downloads	High	> 2 Mbps	N/A		N/A	

The above categorization is a good reference for selecting the most representative applications to stress all characteristics of the multihop wireless backhaul network developed in MEMBRANE. In particular, the following two applications appear to give a good coverage of bandwidth, latency, and jitter:

- Streaming media (stresses bandwidth, and to a small extent jitter)
- VoIP (stresses latency and jitter)

Streaming media transmission can be the playback of a video clip, of appropriate frame size and frame rate, stored on one PC (e.g. the one connected to the EN) and played back on the other (connected to the IN with AN connections). A candidate utility for video streaming is **VIC**, a real-time multimedia application for video conferencing. VIC was designed with a flexible and extensible architecture to support heterogeneous environments and configurations, and allows users to participate in point-to-point and multi-point video conferences. Stressing the latency and jitter performance with VoIP will be based on an appropriate utility program such as **RAT**, a network audio tool that allows users to participate in audio conferences over the Internet.

In addition to the above, we will consider the possibility of generating synthetic traffic at each endpoint, with controlled characteristics such as mean data rate, packet arrival rate, etc, and use this to measure the impact of different traffic patterns on the overall network performance.

Finally, a user interface (GUI) will be developed, in order to:

- calculate and distribute scheduling and routing information for the MEMBRANE nodes
- generate and collect network traffic corresponding to different application scenarios
- measure required MEMBRANE network statistics
- display the MEMBRANE performance characteristics in a graphical format

The exact characteristics of the required network traffic models, network parameters and measured network statistics should be defined after performing network simulations.

3.2 Prototype Diagram

The full diagram of the entire prototype including the blocks of the novel algorithms finally selected for implementation could be available in the final deliverable of the workpackage. For the present time, a general block diagram of the modules composing each node of the demonstrator is described in the following figure.

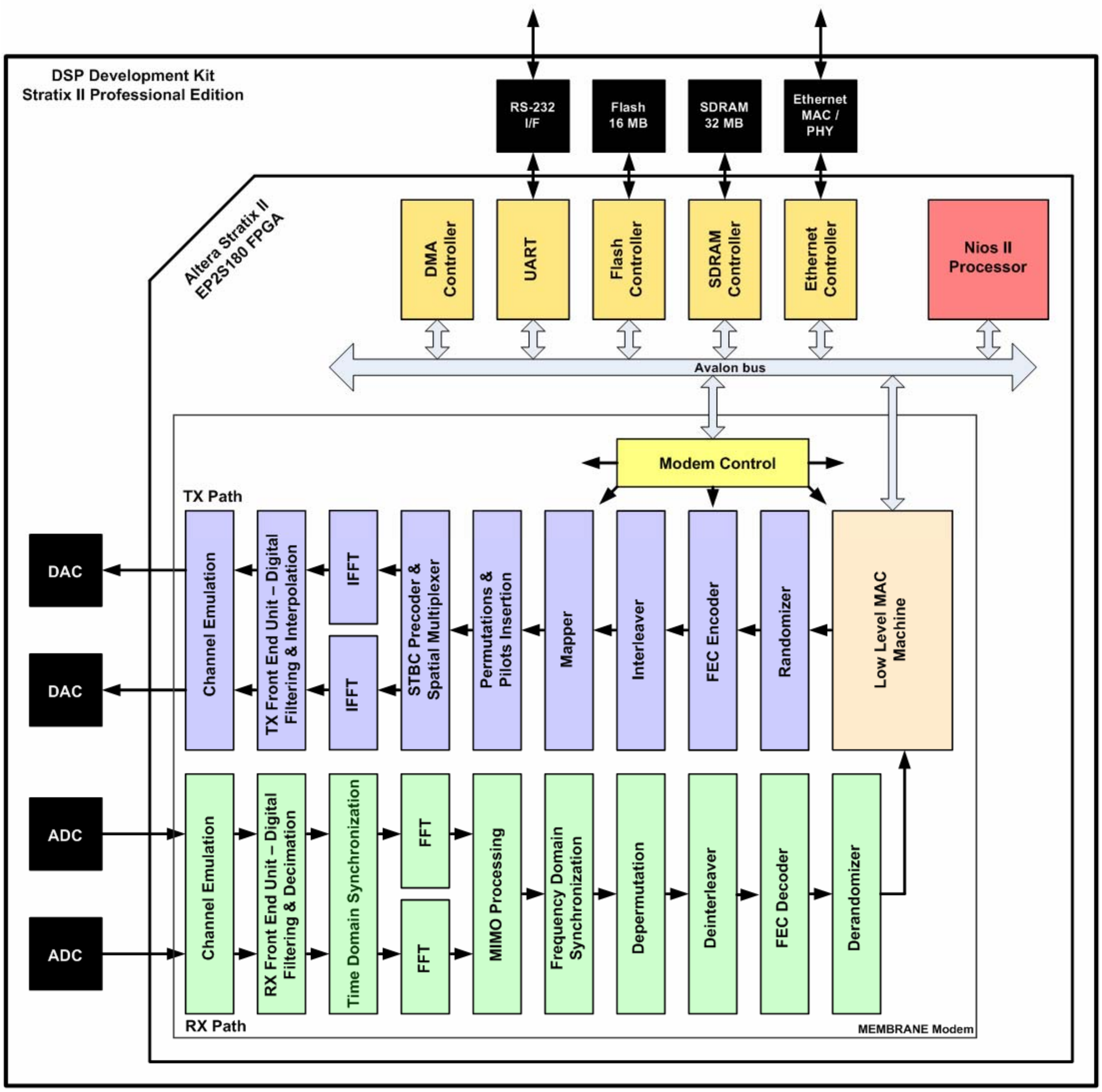


Figure 4: Block diagram of MEMBRANE Prototype

The block diagram of the PHY layer of each node in the MEMBRANE Prototype is shown in Figure 4. It is assumed that the DSP Development Kit Stratix II Professional Edition from Altera is used to prototype one node. The MEMBRANE modem and System-on-Chip (SoC) infrastructure are implemented inside the FPGA. The modem is made of hardware blocks needed to make 802.16e-OFDMA transceiver and low level MAC. The SoC infrastructure consists of Nios II Processor and controllers interconnected together and with the modem using Avalon bus. From this architecture, the blocks modified compared to the 802.16e standard are mainly the “MIMO processing”, the “STBC precoder & spatial multiplexer” and the “low level MAC machine”.

3.3 Prototype Platform Description

It is highly desirable to use commercially available platform for the MEMBRANE prototyping because designing of some custom development board may take considerable part of the MEMBRANE’s time and budget resources and is not a main purpose of the project.

The current platform proposed for the implementation of all the MEMBRANE nodes is the DSP development kit, Stratix II Professional Edition from Altera Corporation. One development board will be required for every node of the demonstrator. So in total, three boards are required. The photo of the board is shown in Figure 5.

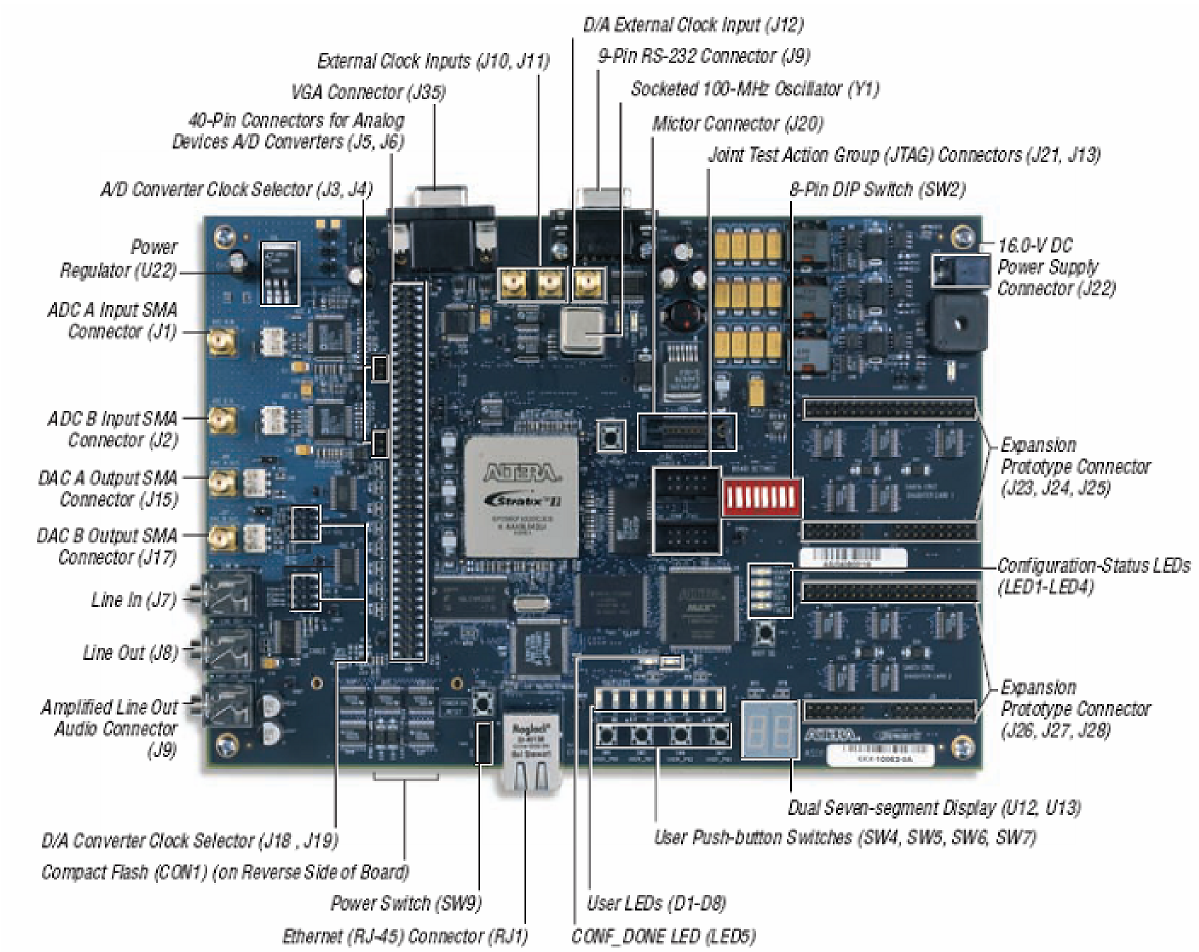


Figure 5: DSP development kit, Stratix II Edition from Altera

The main features of the DSP development kit are:

- The Altera Stratix II EP2S180 FPGA. Its capacity is equal to 180K of equivalent logic elements (LE). This FPGA will be sufficient to implement required prototype functionality.
- The board includes two 12-bit 125-MHz A/D converters and two 14-bit 165-MHz D/A converters.

- Soft high-performance 32-bit NIOS II processors can be configured inside of the Stratix II FPGA for execution of the networking software and control functions. Several processors may be instantiated.
- 10/100 Ethernet MAC/PHY chip is included in the kit.
- External SDRAM and Flash memories are available on the board.

INTEL's MEMBRANE team has experience of working with this board and is able to provide some ready-to-use RTL blocks (memory controllers, Ethernet controller) and software for Nios II processor (Ethernet driver, embedded TCP/IP stack and others) which can help to quickly solve the background tasks and concentrate on the development of the core MEMBRANE blocks.

3.4 Operation Mode

Since the channel emulation is included in each ALTERA board, the prototype of the MEMBRANE system is composed only of three boards. In order to compensate any problems of synchronization between the two boards, the EN card will be elected as the master and the two IN boards will act as slaves.

The two IN boards have two operation modes: a sleep mode where they listen with no transmission and an operation mode where they switch between transmission and reception. The transition from one mode to the other is triggered by the EN. The EN is the only board that has only one operation mode which is the transmission/reception mode. The two IN boards switch from the sleep state after the activation of the EN board. There is also a control signal transmitted from the EN to either one of the IN cards that orders these cards to switch to their respective sleep mode.

Values of selected demonstration parameters will be acquired in regular time intervals that are defined according to the demonstration needs and the characteristics of the parameters. The acquisition of these values will be the task of specifically designed applications running in the PCs individually connected to each of the nodes of the demonstrator through an Ethernet connection.

4 IMPLEMENTATION STRATEGY

The purpose of this section is to provide some initial guidelines concerning the work conducted from all the partners involved in the specific workpackage. This subsection is important in the sense that the partners assigned to develop the MEMBRANE prototype have different procedures and background. This section targets to align the effort of all the partners of WP5.2. Therefore, at first, an initial collaboration framework among the partners is described including the exchange procedure among partners. Then, the different phases towards the development of the prototype are described. Finally, the testing procedures to validate the prototype are drawn.

4.1 Collaboration Framework among Partners

In this section, the effort planned within the workpackage and the development of the prototype, is subdivide into several subtasks and some milestones have been set. This breakdown of the effort needed for the completion of the MEMBRANE demonstrator provides the general progression path of the activity. This session can be used as a framework to align the partners involved in WP5.2. Also, this session presents the exchange procedures of codes and knowledge between partners.

4.1.1 Work Planning

According to the PERT diagram of figure 8 of the Description of Work of the project [MEM DoW], WP5 takes inputs from WP2 and WP4. From WP2, the scenarios targeted for the MEMBRANE system are needed in WP5, and especially in WP5.2, in order to identify those scenarios that are more suitable for demonstration purposes. The selected scenarios are deduced in section 2 of this deliverable. They are more suitable to provide the proof-of-concept of the demonstrator developed within the MEMBRANE framework. The algorithms implemented in the prototype are taken from WP4 activities. In particular, the smart antenna link, routing, scheduling and power control algorithms that should be implemented in WP5.2 are described in WP4.1 and WP4.2.

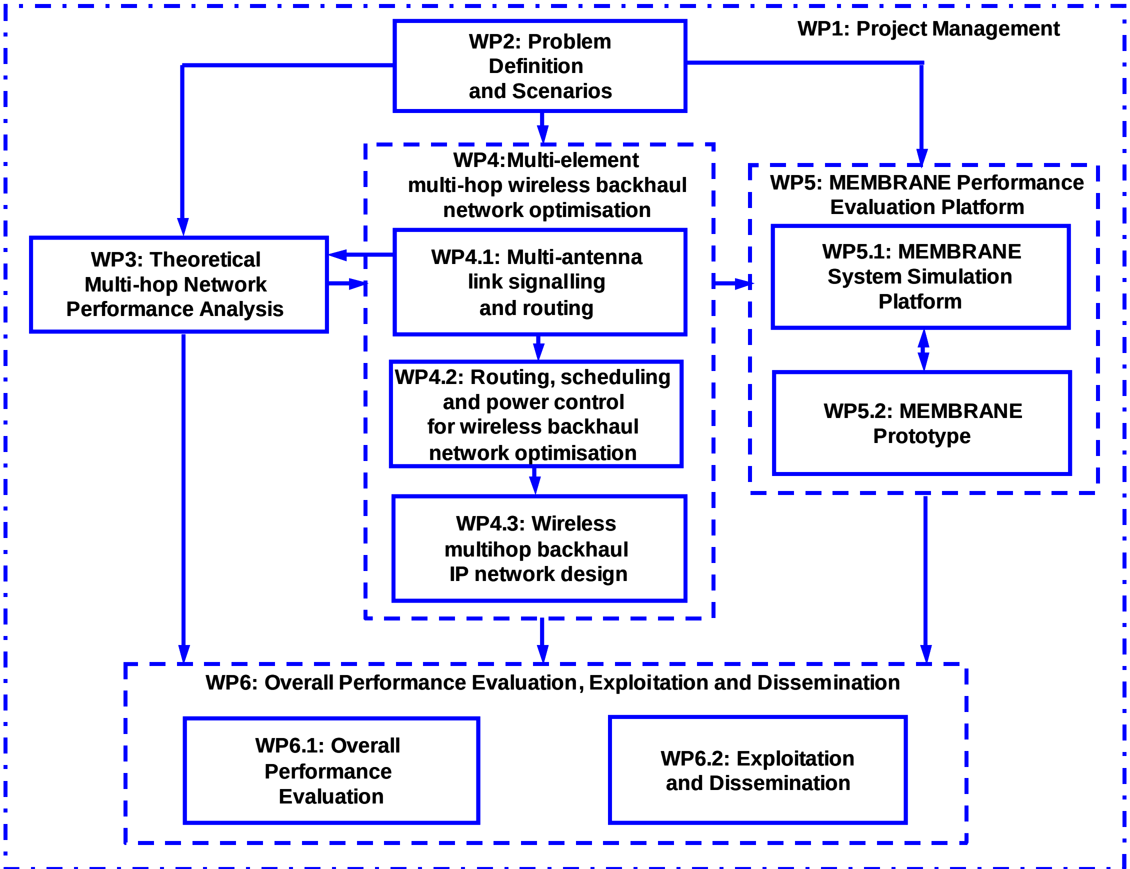


Figure 6: Pert Diagram of figure 8 of the "Description of Work" of MEMBRANE

The following 6 subtasks constitute the core of WP5.2:

1. **WP5.2a: Prototype Scenario Description (M7-M10, June-October 2006)**

   During this 4 months subtask, the prototype scenario suitable for implementation to the prototype is identified and described. An initial distinction between the expected outcomes from WP5.1 and WP5.2 is established. A collaboration framework between the work conducted in WP5.1 and WP5.2 can be deduced. Inputs for this subtask are deliverable D2.1 and D4.1.1. The target scenarios selected for the demonstrator are described in section 2.1.1

2. **WP5.2b: Platform Identification and Description (M10-M12, October-December 2006)**

   During this 3 months subtask, the most suitable platform capable to demonstrate the functionalities studied in MEMBRANE is clearly identified. The platform is described in details and a justification of the selection of the platform is deduced. Information of the demonstrator platform and the elements composing it are given in section 3 of this deliverable. The outcome of this subtask and previous one, is this deliverable.

3. **WP5.2c: Block Diagrams and HW/SW partitioning (M13-M15, January-March 2007)**

   During this 3 months subtask, the block diagram of the final prototype chain is produced based on the selection of novel algorithms from WP4. Each module of the prototype is identified and its functionalities are described analytically. The Hardware and Software partitioning of the prototype is deduced. Inputs to this subtask are deliverables D5.1.1 and D4.2.1 but mainly the work conducted in WP4.1 and WP4.2.

4. **WP5.2d: Development of Modules (M16-M24, April-December 2007)**

   The modules and blocks identified in the previous subtasks are developed and tested individually. Particular attention and effort is dedicated to the blocks implementing the algorithms of the MEMBRANE system.

5. **WP5.2e: Integration Phase (M23-M28, November 2007-April 2008)**

   During this activity, the blocks developed in the previous subtasks are integrated to the selected prototype platform.

6. **WP5.2f: Validation and GUI Development (M29-30, Mai-June 2008)**

   This final subtask targets to evaluate the prototype. A GUI suitable to demonstrate the prototype capabilities is developed. Also, during this phase, final rectifications can take place.

There are three major milestones to be reached in WP5.2:

1. **WP5.2A: Collaboration Framework of System Simulators and Prototype Platform (M10, End of October 2006)**

   This milestone is related to subtask WP5.2a where the scenario of the prototype is described. In WP5.2a, the demonstrable scenario of WP5.1 is required to produce a suitable demonstration scenario. The milestone implies that the demonstration targets of WP5.1 and WP5.2 are well-defined. A framework of distinction and collaboration of the two WP is deduced.

2. **WP5.2B: Demonstrator Features and Prototype Platform Selection (M10, End of October 2006)**

   Based on inputs of WP4, subtask WP5.2a and the previous milestone, the basic requirements of the prototype are defined which leads to the selection of the appropriate Prototype platform.

3. **WP5.2C: Prototype Algorithms Selection from WP4 (M12, End of December 2006)**

   The milestones points out the date where initial algorithms suitable for implementation in the prototype are selected. The algorithms are chosen based on the work carried out in WP4 and in particular in WP4.1 and WP4.2. At this milestone, algorithms for the prototype must be available. A refinement of these algorithms and a complexity evaluation is possible during subtask WP5.2c where the end-to-end block diagram of the prototype is performed.

Collaboration with other workpackages is planned as follows:

- **With WP5.1:** During subtask WP5.2a to reach milestone WP5.2A
- **With WP4:** For the algorithm selection in WP5.2C and during subtask WP5.2c.

The timetable of these subtasks is shown in the following graph. The subtasks and milestones referred in this document are marked in red.

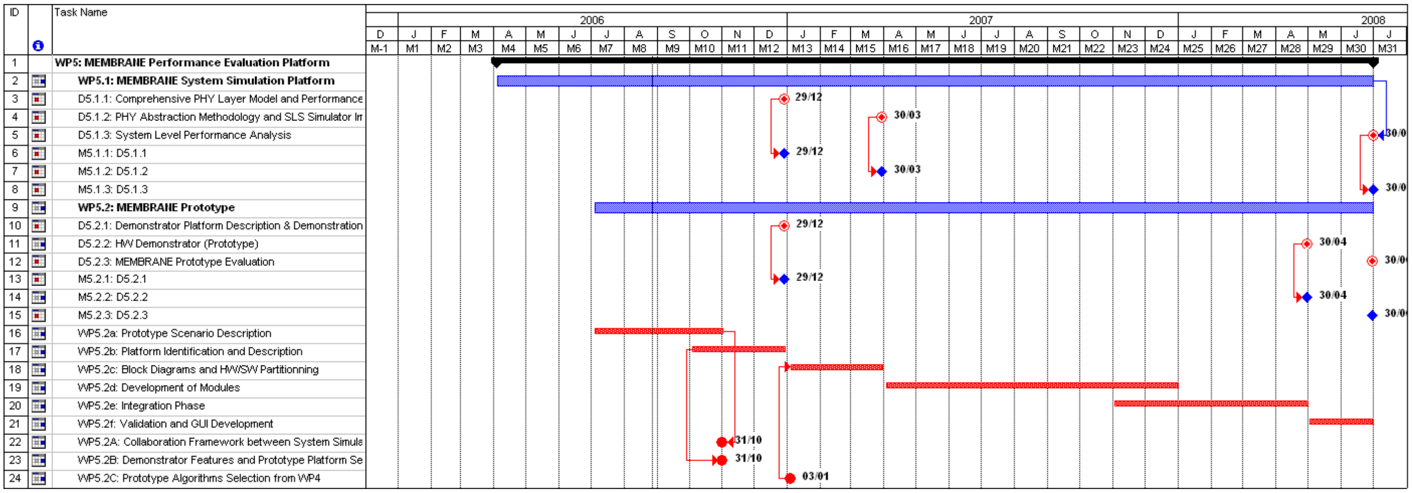


Figure 7: Gantt Chart

It is obvious that the first two subtasks WP5.2a and WP5.2b are currently completed. The outcome of these tasks is this deliverable, where the prototype scenarios and the platform are described.

The milestones WP5.2A and WP5.2B have been reached in M10. Therefore, the INTEL platform and boards are described in this document. It has also been taken care of that the Link Level Simulator (LLS) results of the deliverable WP5.1.1 "Comprehensive PHY layer model and performance analysis of the PHY link for different scenarios" due also in M12 (December 2006) are aligned with the prototype under consideration. Finally, the prototype algorithms selected have been already presented in D4.1.1 "Reconfigurable IA/MIMO transceiver algorithms" delivered in M6 (June 2006).

However, there is always room for further improvement and finalization of the decisions, specifications and constraints selected so far. It is possible that some further additions and/or modifications are considered in order to take into account new requirements, ideas or algorithms developed within the MEMBRANE project.

4.1.2 Exchanging Procedure among Partners

The blocks requested and exchanged between partners are of two kinds: First the algorithms with the specifications and characteristics of MEMBRANE, and the blocks developed for the prototype.

For the first type of codes, the functions are given in either MATLAB format or ANSI C code. The partners will take care to assure that the codes are readable and well-commented. No code will be delivered without an accompanying document describing the functionalities of the code, the inputs and outputs of the functions provided. If possible, testing functions should be provided. In any cases, test procedures and eventually test vectors should be included in the package delivered.

For the prototype blocks, similar constraints are required. In particular, for proprietary code delivered in netlist form only, a document describing the block and a testing environment for validation of the block are mandatory.

4.2 Design, Development and Integration Phases

The methodology described here reflects typical design methodology for both HW and SW systems, as well as mixed HW-SW systems. In general, the following five main phases can be identified:

1. System specification
2. System-level design

3. Units development
4. Unit tests
5. System integration

Following integration, the entire system will be verified, a procedure that is a subject of the next section. In the following, the above steps are defined and explained.

1. Specification

In this phase, the entire system, in terms of functionality and performance, is defined. All functions of the system, operating procedures and external interfaces are captured in a system specifications document that will serve as the guideline for the following steps. This step will be based on input from the 802.16 standard and the novel PHY and MAC techniques specified in WP4.

2. System-level design

During this phase, the entire system is divided into several subsystems. Starting with a coarse breakdown of the system into major subsystems, a decision will have to be made as to whether each subsystem will be implemented with the hardware or software resources of the prototype platform. In particular, since the system will consist of at least a PHY layer and a simple MAC layer (as per the 802.16 standard) as well as a demo application, a first breakdown is already defined, with hardware for the PHY layer and software for the application being likely (but not necessary) implementation targets.

Following the first-level breakdown, the design is progressively refined, until units are identified, where further breakdown is deemed impractical or not beneficial. Such units are logic blocks for the hardware components and tasks for the software ones. The interfaces between the components are defined in this step and individual units are assigned to developers for the development step.

3. Units development

In this step, all individual units are developed by the individual engineers, according to the specification of the first step and the analysis and elaboration carried out in the second step. Suitable HDL (Verilog or VHDL) will be used for the hardware components, while C or other high-level language is the target language for the software components. Assembly language is an option in case performance-critical sections are required to run at maximum efficiency, although this is only considered as a last resort due to the additional design effort required.

4. Unit tests

As each unit is developed, it is tested in isolation for conformance to the required functionality and performance, as well as to its interface specification. An appropriate number of test vectors is used to stress all operating modes of the component. For the more complex components, a combination of directed and random tests is used, in order to identify and verify corner cases that are difficult to foresee and stress manually. All sets of test vectors are archived and organized in a way that they can be re-executed in batch mode, in anticipation of change requests that may result from the integration phase.

5. System integration

After testing of individual units, they are gradually integrated into larger subsystems and eventually the entire system is assembled by the responsible partner.

Due to the large number of units involved and the complexity of the overall system, a phased approach is recommended for the construction of the final system. Early instances of the system, or *milestones*, with limited functionality, but with end-to-end communication possible, will be integrated and verified first. Once verification of an earlier milestone is complete, integration of additional functionality may follow. The same is true for the overall system as well as for its major subsystems. For example, for the 802.16-based PHY subsystem, the following milestones may be distinguished:

- SISO Tx-Rx chain
- MIMO chain with basic features (e.g. diversity, simple spatial multiplexing)
- Fully functional chain.

The exact milestone features and overall integration plan should be specified in a relevant document after the system-level design (step 2) is completed and once the development of individual units is under way. Integration of the top-level subsystems (application, MAC, and PHY) will also be the subject of this integration plan.

4.3 Testing and Validation Procedures

A proof-of-concept prototype will be established to serve as a means to evaluate selected concepts addressed by the MEMBRANE project.

On one hand, specific features enabling the optimised backhaul network will be implemented in a commercial hardware platform and a demonstrator set-up will be established, able to evaluate both the performance and the economic viability of the MEMBRANE network. As a means to accomplish this task, a commercial hardware platform based on high performance FPGAs, has been selected, as described in Section 3.3, which is able to support diversity schemes and enhanced capacity, roughly as the real designed system should do.

On the other hand, analytical and simulation modelling allow a detailed analysis of the designed algorithms, mechanisms and overall system in general. As a matter of fact, such modelling techniques are complementary to the test-bed prototype. This means that a combination of different approaches should be applied for an effective system testing and evaluation.

By theoretical and empirical studies, it is well known that at least two of the above mentioned techniques are needed. It is likely and sometimes more straightforward to use analytical modelling, especially for algorithms with heavy mathematical basis. However, test-bed prototype and simulation modelling are to be necessarily employed in such complex systems as conceived in the MEMBRANE project.

In order to correctly and properly assess the solution proposed by the project consortium, first of all it is fundamental to test and validate the tools you are going to use for the purpose.

In the following sections, a brief description of a test-bed prototype, simulation modelling and validation process in general is provided. Subsequently, the specific testing and validation procedures will be presented.

4.3.1 Test-bed prototyping

A test-bed is a platform environment used to exploit, verify and validate a network aspect or a network architecture model also with regards to a specific set of services. It is very important to properly define and build the test-bed platform, because on its correctness, consistency and reliability, depend the results and conclusions about the issues to be addressed.

Several elements are needed to completely specify a testbed environment, such as involved network devices and their interconnections, supported protocol suites and technologies, operating system, type and capabilities of the end-hosts and running applications.

It is also fundamental to select the relevant measurements to be gathered and analyzed, as well as the network points and conditions to collect significant values. The measurement process has to be defined and executed with a certain level of detail that is a trade-off between the desired accuracy and the resulting complexity. The measurement type can be both intrusive and non-intrusive, depending on the case and can be collected either with a specific probe device (custom hardware and software created for that purpose) or at application level. The possibility to use configurable traffic generators allows a greater flexibility in the realization of the measurement process.

In MEMBRANE project, the realization of a testbed within the pilot is a fundamental goal. Particular care will be taken in defining the overall testbed platform, the implemented novel algorithms and mechanisms, the measurement process, the utilized probe devices, the set of applied traffic generators and the specific traffic models for the concerned applications, as well as the environment characteristics.

4.3.2 Testing methodology

An effective testing methodology, as that shown in Figure 8 (for clearness, related just to a simulation-based testing), should be used to analyse in an appropriate manner complex network issues. This is necessary if we want to obtain from the simulation and test-bed prototypes meaningful results that may be exploited in the architectural and configuration design of the novel MEMBRANE proposal.

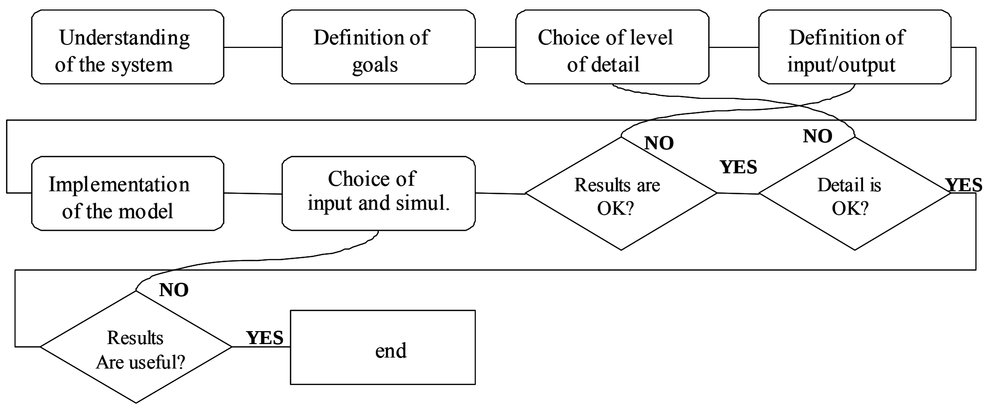


Figure 8: General testing methodology schema (simulation modelling case).

Given a more or less accurate specification of the system to test, the first, extremely important step is to formalize the most critical and important issues to be considered. In other words, to get inputs from system specification and requirements, even in its very early stages, isolating the functionalities and parameters more related to the issues to be addressed. By this, the definition of the goals and objectives of the testing process arise, hence which measurements and results to collect.

In the following steps, for the simulation modelling it is necessary to choose the proper detail level of implementing the real system, while for the test-bed it is more a matter of selecting the subset of functionalities to be included. Furthermore, it is required to define the input traffic and the concerned output, run the tests and gather the results. At this point it is fundamental to verify if the results are correct and whether the level of details chosen before for the simulation modelling, or the supported functionality for the test-bed, are sufficient to achieve the defined objectives. If this is not the case, it is required to go back in steps, namely in “Choice of level of details” or “Definition of input/output”; otherwise you can go to the last step where a control of the significance and usability of the results has to be performed and acting as depicted in the above schema.

4.3.2.1 Verification and validation of the prototypes

As already stated, an important step of the simulation modelling process and test-bed building is the verification and validation of the implemented prototypes. This is not a trivial task, because it entails the testing of pieces of software, which can be very hard if the set of the concerned variables and the execution paths is large. This is also true for a hardware component with several different functionality and setting parameters.

After a proper and deep code inspection (from both a semantic and syntax point of view) and correct compiling of it, for what concerns the software components, as well as a proper configuration of the hardware modules, the common way to proceed is to establish a very basic testing scenario and stimulate it with inputs for which we already know, more or less precisely, the corresponding outputs, at least in a qualitative manner; however, also quantitative considerations must be taken on the collected results in order to properly conclude about the consistency of the simulation model and test prototype implementation. A comparison between the results collected with both the employed prototypes (i.e. simulation and test-bed) in roughly the same scenario must be used to test and validate them. Therefore, the simulation scenarios of WP5.1 and the prototype scenarios of WP5.2 will be similar.

This operation should be carried out for a large enough number of target scenarios and this can lead to non-negligible time and effort. Nevertheless, this is necessary in order to ensure the validity of the functional and performance assessment achievements of the concerned system. For the MEMBRANE case, we consider only two target scenarios as defined in section 2.1.1.

4.3.2.2 Computer simulations

In projects, such as MEMBRANE, where it is necessary to study realistic network architectures, computer simulations are essential. Therefore, computer simulations should be used from the beginning to the end of the project. In the first stage, simulations should be used to simply test network architectures since its very early phase, in order to recalibrate the system as needed. During last phases, when network architecture is defined, simulations could be used for configuration issues and to verify or justify results obtained on the test-bed system, and also helping in finding and solving software and hardware bugs. In the context of this Deliverable, the latter is the main relevance of simulation-based testing. The combined use of simulation modelling and test-bed allows an effective testing and validation of the prototypes themselves. The described computer simulation activity is mainly the task of WP5.1, however, this activity will be supported by the work conducted in the demonstrator workpackage in particular for simulations considering the small set of nodes defining the prototype. The comparison of the results will be part of the overall performance evaluation of WP6.1.

4.3.3 Test specifications and validation

In this section, the specification of the tests to be performed and the related validation criteria for the novel functionalities, mechanisms, algorithms and architectural solutions designed into the framework of the MEMBRANE project, as well as for the simulation model and test-bed prototype implementations will be discussed.

Basic guidelines are going to be provided for an effective assessment of the single component and overall system, for both the novelties and the correct realization of the prototypes on the whole, which also requires a proper validation of the standard functionality implementation.

The paragraph is organized according to the ISO OSI reference model, dealing with each layer separately. Since the prototype will only implement functionalities of the PHY and MAC layers, we limit the discussion to those two layers. For every layer, the related tests and validation criteria are specified, avoiding explanations of trivial concerns and testing procedures.

The tests and validation criteria are distinguished for the simulation and the test-bed as needed. Furthermore, some functionality might be analyzed just for one prototype, because implicitly and correctly supported by the other. For example, if you think of addressing, for the simulation prototype, the issue can be automatically managed by the adopted simulation tool, while a checking of a proper address assignment in the prototype is required.

Of course, the implementation of the prototypes is to be validated when all tests are passed successfully and concerning the system behaviour (for the single components and in general), a consistency between the results collected with both of them must be verified.

4.3.3.1 Data Link Layer

Basic signalling and data exchange

Some examples of features concerned in this point are: packet transmission (with or without ARQ), queuing management, flow-control and sequencing.

Multiplexing and scheduling techniques

This functionality can be put in strict relationship with the opportunistic routing and scheduling algorithms, both conceptually and in particular in this context, from a testing and validation point of view.

The goal is to assess the newly devised mechanisms for effective packet scheduling, also at data-link layer, concerning the peculiarity of a wireless multi-hop network, where spatial multiplexing (i.e. multi-antennas) and advanced resource management (e.g. enhanced PF scheduling) are going to be developed into the framework of the MEMBRANE project. More precisely, for the scheduling algorithm proposed in WP4.2, a time-slotted and synchronized system is required. The frame-structure of data and a separate channel for control information are expected to be supported. Also, the test-bed is desired to perform the labelling algorithm of fresh nodes to the network, as this has been proposed and described in WP4.2. Initial simulation results will be based on omni-directional antennas and mutually independent channels between different users; directional antennas and beam-forming will be also included in the next phase.

Some of the parameters that are expected to be collected by the test-bed include the utility functions of each link, the initial decisions of each transmitting node, the routing decisions and queue length at each node, data throughputs for comparison, the smoothed MAC layer throughput which is part of PF scheduling metric, etc. Moreover, a realistic delay of the feedback information has to be taken into account.

Finally, the preliminary thought is to compare the proposed distributed scheduling algorithm with other two approaches, namely the tree-structure algorithm and the centralized optimal algorithm. Thus, it is desirable that the testing methodology will be able to deal with the mapping algorithm for the tree-structure algorithm. That is to find out the level, the parent and the children for a new node added into the network. Regarding the centralized algorithm, a central control node that has full knowledge of the network has to be enabled.

For the test-bed, the instrument(s) to be employed for testing purposes strongly deal with the lower layers of the concerned network devices: therefore, vendor-specific HW and SW components might be required.

The fairly complete set of application and network scenarios that are going to be results from a compromise between different and several needs and constraints: exhaustive analysis, device and measurement tool availability, radio interface setup, overall complexity, etc…

Effective sharing of communication resources among multiple users (e.g. contention resolution)

In this case, the issue of resource sharing in a common medium environment (i.e. multiple access techniques) must be tackled.

A pool of different working modes are available for each technologies and at least the ones that are going to be used during the assessment phase of the MEMBRANE proposals must be considered in detail. For example, a TDM or OFDM access without collisions.

The approximations introduced in the prototypes are going to be more dramatic for the simulation modelling; for this reason, it is necessary to carefully look at the collected testing data, both from a functional and a performance point of view, in order to well understand their impact. Indeed, the radio channel might be modelled in a too simple manner in such prototype, also for the difficulty in taking into account of complex physical phenomena.

Different radio channel conditions should be created and analyzed (i.e. tested and validated), in accordance with the issued network scenarios (e.g. AWGN channel without or with slow fading).

For the simulation prototype, proper analytical models should be implemented; while for the test-bed, wired connections emulating a wireless hop are going to be deployed. The couples of configuration settings for the two realized prototypes, able to create the same medium conditions, will be roughly determined.

4.3.3.2 Physical Layer

Modulation and channel coding

The modulation and coding schemes supported by IEEE 802.16e-2005 standard (QPSK, 16-QAM, 64-QAM, convolutional and turbo coding) will be implemented and tested in the prototype.

First, a proper selection of the techniques that are going to be employed during the test-bed is required, For example, BPSK and QAM are going to be employed for a proof-of-concept. Then, by imposing well defined channel conditions (i.e. transfer function, SNR), a testing input patterns of bits will be transmitted for each possible combination of interested modulation and coding channel techniques. By checking the corresponding output patterns, and comparing the achieved bit-error rate (BER) with the one predicted from theory, the correct implementation of the realized prototypes with respect to this issue can be verified.

Again, as already discussed, different testing components and methods are to be employed for the simulation modelling or the test-bed. For the former, debugging is really suitable; while for the latter, both SW and/or HW tools, and monitoring facilities (when available) by the specific HW platform used are going to be exploited.

The results collected with both the prototypes should be matched. However, the simulation models might not be detailed enough. When a bit level granularity is not available with it, the validation process must be performed in a statistical manner, by looking at the bit error distributions, mean values, variances, etc.

Intelligent Antenna models

This is one of the main features of the MEMBRANE project. Reconfigurable IA is expected to boost throughput gains, reduce interference substantially and obtain an effective power control, thus improving overall end-to-end performance.

Such goals are addressed by profitably employing all the advances designed and developed by the project consortium, which have been already discussed from a testing and validation point of view in various paragraphs of this section. The main issue here is to test and evaluate the wireless multi-hop system on the whole, in terms of correct implementation, functioning and achieved performance, by combining several mechanisms and algorithms, whose testing and validation criteria have been already specified separately.

5 CONCLUSIONS

The objective of this deliverable is to describe the scenarios and the platform selected for the demonstration of the state-of-the-art techniques and concepts developed and investigated within the MEMBRANE project. This document is the first deliverable of workpackage WP5.2, the task of which is the development of the MEMBRANE demonstrator. Therefore, this deliverable does not only represent the outcome of the work conducted so far in the workpackage. It can also serve to guide and align the work of the partners involved in WP5.2 towards the common goal, i.e. the development of a prototype of a multielement multihop backhaul system capable to demonstrate part of the advanced techniques investigated in MEMBRANE.

To this end, the scenarios have been defined based on inputs from D4.1.1. The propagation conditions considered for the demonstrator were produced from D4.1.1 and D2.1. In addition, two different demonstration scenarios have been produced in function to the layer where the major modification takes place. Algorithms and protocols for the prototype have been presented, and initial system requirements have been produced.

The platform selected for the MEMBRANE demonstrator has been described in details including the elements composing the platform, the connectivity between the boards, the applications running that evaluate the gains achieved by the proposed algorithms. A diagram of the prototype has also been included in this deliverable.

Finally, the collaboration framework among partners is explained by the subdivision of the effort of the workpackage into several subtasks. The code exchange procedure among partners is included. Testing and validation procedures to unite the results and work conducted have been introduced.

TERMS AND ACRONYMS

2G	2^{nd} Generation
3G	3^{rd} Generation
ADC	Analog-to-Digital Converter
AMC	Adaptive Modulation and Coding
AN	Access Node
ARQ	Automated Repeat request
AWGN	Additive White Gaussian Noise
BD	Block Diagonalization
BER	Bit Error Rate
CQI	Channel Quality Indicator
CSI	Channel State Information
DAC	Digita-to-Analog Converter
DL	DownLink
DSP	Digital Signal Processing
ECN	Explicit Congestion Notification
EN	End Node
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
FUSC	Full Usage of SubChannels
GUI	Graphic User Interface
HW	Hardware
IETF	Internet Engineering Task Force
IN	Intermediate Node
IP	Internet Protocol
MAC	Media Access Control
MAN	Metropolitan Area Network
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Square Error
LAN	Local Area Network
LE	Logic Element
LOS	Line-Of-Sight
NLOS	Non Line-Of-Sight

OFDMA	Orthogonal Frequency Division Multiple Access
OSI	Open Systems Interconnection
OSTBC	Orthogonal Space Time Block Codes
PAN	Personal Area Network
PUSC	Partial Usage of SubChannels
PER	Packet Error Rate
QAM	Quadrature Amplitude Modulation
RX	Receiver
SDMA	Space Division Multiple Access
SISO	Single-Input Single-Output
SM	Spatial Multiplexing
SMMSE	Successive Minimum Mean-Square-Error
SNR	Signal-to-Noise Ratio
SoC	System-on-Chip
SVD	Singular Value Decomposition
SW	Software
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TX	Transmitter
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UL	UpLink
VBLAST	Vertical Bell Labs Architecture Space Time
VoIP	Voice over IP
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
ZF	Zero Forcing

REFERENCES

[AJSmith] A. J. Smith. ”Analysis of long term file reference patterns for application to _file migration algorithms”.. IEEE Transactions on Software Engineering, 7(4):403{410, July 1981.

[Ala98] S.M. Alamouti, “A Simple Transmit Diversity Technique for Wireless Communications”, IEEE Journal of Selected Areas in Communications, Vol. 16, pp. 1451-1458, October 1998.

[ArlWil] M. F. Arlitt and C. L. Williamson. “Web server workload characterization: The search for invariants”. May 1996. Preprint. To appear in Proc. 1996 ACM SIGMETRICS.

[BRADY] BRADY (P.T.). “A statistical analysis of on-off patterns in 16 conversations”. BSTJ, pp. 73-91, January 1968.

[CrovBest] M. Crovella and A. Bestavros. “Self-similarity in World Wide Web trace: Evidence and possible causes”, May 1996. Preprint. To appear in Proc. 1996 ACM SIGMETRICS.

[CCITTT XII-20] CCITT Contribution COM XII-20. “On-off characteristics of conversational speech (CSELT)”. Study Period 1989-1992.

[CCITT D-42] CCITT Contribution Delayed D.42 (WP XII/1). “Collecting procedure for on-off characteristics of conversational speech in telecommunication (NTT)”. Study Period 1989-1992.

[CCITT COM-64] CCITT Contribution Delayed COM-64 (WP XII/1). “Generation of artificial voice with pauses (NTT)”. Study Period 1989-1992.

[CroBes] Crovella M. and Bestavros A.. 1996. Self-similarity in World Wide Web traffic: evidence and possible causes. *Proceedings of SIGMETRICS '96*.

[CunBesCro] C. R. Cunha, A. Bestavros and M. E. Crovella. “Characteristics of WWW Client-based Traces”.

[DanJanMit] Danzig P., Jamin Caceres S. R., Mitzel D., and Estrin D.. 1992. An empirical workload model for driving wide-area TCP/IP network simulations. *Internetworking: Research and Experience* 3:1–26..

[FloJac] Floyd, S. and Jacobson V.. 1992. On traffic phase effects in packet-switched gateways. *Internetworking: Research and Experience* 3:115–156.

[HKim] H. Kim. ”A Non-Feedback Congestion Control Framework for High-Speed Data Networks”. PhD thesis, University of Pennsylvania, 1995.

[ITU-T RecP59] ITU-T Recommendation P.59. “Artificial Conversational Speech”. March 1993.

[KliMukSon] S.M. Klivansky, A. Mukherjee and C. Song. “On Long-Range Dependence in NSFNET Traffic”. Technical Report GIT-CC-94-61. December 19, 1994.

[KliMukSong] S.M. Klivansky, A. Mukherjee and C. Song. “Factors Contributing to Self-similarity over NSFNet”. Technical Report, 1994.

[LEE UN] LEE (H.H.), UN (C.K.). “A study of on-off characteristics of conversational speech”. IEEE Trans. On Comm., Volume COM-34, No. 6, pp. 630-637, June 1986.

[LomMorPal] A, Lombardi, G. Morabito, S. Palazzo and G. Schembra. "A Markov-based Model of MPEG-2 Audio/Video Traffic". Globecom'99.

[LelTaqWil] Leland W., M. Taqqu, W. Willinger, and D. Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking* 2:1–15.

[LelTaqWilWil] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, "Self-Similarity in High-Speed Packet Traffic Analysis and modeling of Ethernet Traffic Measurements". Statistical Science, 1994.

[MEM DoW] IST-MEMBRANE, Description of Work, "Multi-Element Multihop Backhaul Reconfigurable Antenna NEtwork", IST-4-027310

[MEM D4.1.1] IST-MEMBRANE, Deliverable 4.1.1, "Reconfigurable IA/MIMO transceiver algorithms"

[MEM D2.1] IST-MEMBRANE, Deliverable 2.1, "Most relevant community/market needs and target deployment scenarios for wireless backhaul multi-element multihop networks"

[MSalyam] M. Satyanarayanan. "A study of file sizes and functional lifetimes". In Proceedings of the Eighth ACM Symposium on Operating System Principles, December 1981.

[MurTea] J- Murphy and J. Teahan. "Video Source Modelling for ATM networks". IEEE 1994.

[ParKimCro] K. Park, G. Kim and M Crovella."On the relationship between file sizes, transport protocols and self-similar network traffic". Proc. Fourth International Conference on Network Protocols. Oct. 1996.

[ParkKimCro] K. Park, G. Kim, and M. Crovella. "On the effect of traffic self-similarity on network performance". Technical report, Boston University Computer Science Department, 1996.

[Pau03] A. Paulraj, R. Nabar, and D. Gore, "Introduction to Space-Time Wireless Communications," Cambridge University Press, 2003.

[PaxsonFloyd] V. Paxson and S. Floyd. "Wide-Area Traffic: The Failure of Poisson Modeling". [PaxFlo] Paxson V., and Floyd S.. 1995. Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Net-working* 3:226–244.

[PaxFas] Paxson V., Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic. Computer Communications Review, V. 27 N. 5, October 1997, pp. 5-18

[Paxson] Paxson V. 1994b. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking* 2:316–336.

[Sam06] D. Samardzija, H. Huang, R. Valenzuela, and T. Sizer, "Initial Report on an Indoor Network MIMO Experiment", submitted to IEEE International Conference on Communications, 2007.

[TaqqWilSher] M. Taqqu, W. Willinger and R. Sherman. "Proof of a Fundamental Result in Self-Similar Traffic modeling".

[WiMAX06] WiMAX Forum. "Mobile WiMAX – Part 1: A Technical Overview and Performance Evaluation", 2006.

[WilTaqShe] Willinger W., Taqqu M., Sherman R. and Wilson D.. 1995. Self-similarity through high-variability: statistical analy-sis of Ethernet LAN traffic at the source level. *Proceedings of SIGCOMM '95*, pp. 100 113.

[WilinTaqqSher] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. “Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level”. In Proc. ACM SIGCOMM '95, pages 100{113, 1995.

[Wol98] P.W. Wolniansky, G.J. Foschini, G.D. Golden and R.A. Valenzuela, “V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel”, URSI International Symposium on Signals, Systems, and Electronics, September 1998.

[Zhu03] X. Zhuang, F.W. Vook, S. Rouquette-Leveil and K. Gosse, “Transmit Diversity and Spatial Multiplexing in Four-Transmit-Antenna OFDM”, IEEE International Conference on Communications, vol.4, pp. 2316-2320, May 2003.

ANNEX I – Traffic models

A fundamental aspect about the testing and assessment of the MEMBRANE proposal is the proper selection of the traffic patterns to be used for the network modelled by the built prototypes. This refers both to the type of traffic (e.g. multimedia or data only, real-time or not) and the way such traffic is generated (e.g. by humans with a software application, pre-recorded traces, analytical models).

In the following paragraphs, first statistical (i.e. analytical) models for the traffic types issued by the MEMBRANE scenarios are introduced and discussed. Subsequently, in section. 2.2.2.2 the traffic patterns (i.e. applications) and the way they are likely to be generated for the test-bed are specified.

The different generation modes have advantages and disadvantages with each other. An analytical model must be properly studied, designed, developed and configured. This entails additional effort, but on the other hand, it can be easily replicated and controlled in its behaviour targeting well defined and desired traffic/application scenarios. While traffic created by reading pre-recorded traces is very straightforward to deploy, but it is limited to the type of traces that are available and used. Lastly, employing real users could appear a good choice, because of its direct relationship with the life, but it has the non-negligible drawbacks of humans retrieval, test replication and possible introduction of misleading bias.

1. Statistic Traffic modelling in IP networks

Modelling the behaviour of an IP network is an immensely challenging undertaking because of the network's great heterogeneity and rapid change. The heterogeneity ranges from the individual links that carry the network's traffic, to the protocols that interoperate over the links, to the “mix” of different applications used at a site and the levels of congestion (load) seen on different links. We discuss two key strategies for developing meaningful statistic traffic modelling, in the face of these difficulties: searching for in-variants and judiciously exploring the model parameter space.

1.1 The search for invariants

The first observation we make is that, when faced with a world in which seemingly everything changes beneath us, any “invariant” we can discover then becomes a rare piece of bedrock on which we can then attempt to build. By the term invariant we mean some facet of IP networks behaviour which has been empirically shown to hold in a very wide range of environments.

Thinking about IP networks properties in terms of invariants has received considerable informal attention, but has been hardly addressed systematically. We therefore undertake here to catalogue what are promising candidates:

1.1.1 Longer-term correlations

Longer-term correlations in the packet arrivals seen in aggregated IP networks traffic are well described in terms of “self-similar” (fractal) processes. To those versed in traditional network theory, this invariant might appear highly counter-intuitive. The standard modelling frame-work, often termed Poisson or Markovian modelling, predicts that longer-term correlations should rapidly die out, and consequently that traffic observed on large time scales should appear quite smooth. Nevertheless, a wide body of empirical data argues strongly that these correlations remain non-negligible over a large range of time scales (see [LelTaqWil], [PaxFlo] and [CroBes]).

“Longer-term” here means, roughly, time scales from hundreds of milliseconds to tens of minutes. On shorter time scales, effects due to the network transport protocols—which impart a great deal of structure on the timing of consecutive packets—are believed to dominate traffic correlations, although this property has not been definitively established. On longer time scales, non-stationary effects such as diurnal traffic load pat-terns become significant.

In principle, self-similar traffic correlations can lead to drastic reductions in the effectiveness of deploying “buffers” in IP routers in order to absorb transient increases in traffic load. However, we must note that the network research community remains divided on the practical impact of self-similarity.

That self-similarity is still finding its final place in network modelling means that a diligent researcher conducting IP networks simulations must not a priori assume that its effects can be ignored, but must instead consider how to incorporate self-similarity into any traffic model used.

Unfortunately, accurate synthesis of self-similar traffic remains an open problem. A number of algorithms exist for synthesizing exact or approximate sample paths for different forms of self-similar processes. These, however, solve only one part of the problem, namely how to generate a specific instance of a set of longer-term traffic correlations. The next step—how to go from the pure correlation structure, expressed in terms of a time series of packet arrivals per unit time, to the details of exactly when within each unit of time each individual packet arrives—has not been solved. Even once addressed, we still face the difficulties of packet-level statistic modelling vs. source-level statistic modelling, we note that in [WilTaqShe] one promising approach for unifying link-level self-similarity with specific source behaviour is discussed, based on sources that exhibit ON/OFF patterns with durations drawn from distributions with heavy tails.

1.2 Carefully exploring the parameters space

Another fundamental coping strategy is to judiciously explore the parameters space relevant to the traffic models to be considered. Because the IP networks are such a heterogeneous world, the results of a modelling based on a single set of parameters are useful for only one thing, namely determining whether the used parameters set exhibits a show-stopping problem. As one Internet researcher has put it, “If you used a single set of parameters for the considered models, and produce a single set of numbers (e.g., throughput, delay, loss), and think that that single set of numbers shows that your algorithm is a good one, then you haven't a clue.”

Instead, one must analyze the results of modelling process for a wide range of parameters.

Selecting the parameters and determining the range of values through which to step is a challenging problem. The basic approach is to hold all parameters (protocol specifics, how routers manage their queues and schedule packets for forwarding, network topologies and link properties, traffic mixes, congestion levels) fixed except for one element, to gauge the sensitivity of the scenario to

the single changed variable. One rule of thumb is to consider orders of magnitude in parameter ranges (since many IP networks properties are observed to span several orders of magnitude).

In addition, because IP networks include non-linear feedback mechanisms, and often subtle coupling between their different elements, sometimes even a slight change in a parameter can completely change numerical results (see [FloJac]).

In its simplest form, this approach serves only to identify elements to which a statistic modelling is sensitive. Finding that the modelling results with the chosen parameter set do not change as the considered parameter is varied, does not provide a definitive result, since it could be that with other values for the fixed parameters, the results would indeed change. However, careful examination of why we observe the changes, we do may in turn lead to insights into fundamental couplings between different parameters and the network's behaviour. These insights in turn can give rise to new invariants, namely properties that, while not invariant over IP network traffic in general, are invariant over an interesting subset of IP network traffic.

1.3 Statistic Traffic modelling

As in the real internet world, in MEMBRANE project we deal with the presence of different traffic sources: monomedia streams (audio/voice), multimedia streams (audio and video) produced by different coders (MPEG-2/4 and H.261/263/264) and various kinds of asynchronous data traffic produced for instance by FTP, E-mail, telnet and WWW clients.

In the next subparagraphs, a possible model for each identified traffic source is described in detail, also in terms of significant configuration parameters. To be noted that the proposed models have been selected with particular care to their feasibility in the field. For this reason they can result by a rational simplification of more complex ones available in literature (in most cases, too complex models can't be easily handled in practice).

1.3.1 Poisson (Classical) Model

First studies on data traffic indicated that the data traffic sources in communication networks were often bursty in nature, i.e. relatively short sequence of source activities are followed by long idle periods. During 70's and early 80's, those and some other studies suggested the following assumptions as reasonable, if somewhat simplified, for external data sources:

1. the interarrival times of messages generated by an external data source are exponentially distributed, i.e., each external data source behaves as a Poisson process. Let G(i), i=1,2,…,N, be a random variable denoting interarrival times of messages generated from the ith data source
2. the length of messages generated by an external data source is exponentially distributed. Let H(i), i=1,2,…,N, be a random variable denoting length of messages from the ith data source
3. processes described by random variables G(i) and H(i) are stationary and independent.

As a consequence of the assumption 1, the aggregate traffic from several data sources would get smoother and smoother with an increase of a number of sources.

The assumption 2, about exponential distribution of lengths of messages can be relaxed to general distributions and still closed-form solutions for different statistical parameters (mean, variance and other moments of the distribution) could be obtained using queuing theory methods.

Assumption 3, above is not realistic, because at least interarrival times and message lengths for message streams entering a communication node are clearly statistically dependent. For example, if we consider two successive messages arriving into a switch, the second message cannot get into the switch before the first message has arrived completely.

1.3.2 Packet Train Model

The packet train model assume that a group of packets travel together, and it should be obvious that a protocol design based on the assumption of a train arrival would be quite different from one based on independent arrivals. In the car model, each car has to decide at each intersection (or exit) whether to take exit or not. Even if all packets are going to one destination, they each make independent decision, which may result in unnecessary overhead. The overhead is apparent on computer networks in which all intermediate nodes (routers, gateways, or bridges) must make this decision for all packets. In a train model, on the other hand, the locomotive (the first packet of the train) may make the routing decision, and all other packets of a train may follow it.

It must be pointed out that the packet train model is a source model. It applies only when we look at the packets coming or going to a single node.

Unlike the Poisson processes, trains are not additive. The sum of a number of trains is not a train.

In order to allow analytical modelling with a simplified form of train model, usage of a two-state Markov model is suggested. The source can be either in generation (train) state or idle (inter-train) state. The transitions between these states are memory less (Markovian). The duration of the two states is exponentially distributed, with intertrain arrival times possibly of the order of several seconds and intercar times inside the trains of the order of a few milliseconds.

This statistic model, as presented in its basic version, is no more applied in practice, but a very popular source models family is derived from it: the ON-OFF traffic models. This family covers an important role in nowadays traffic modelling; for this reason a detailed description will be provided further in this paragraph.

1.3.3. Long-Memory (Self-Similar) Model

Some recent studies show that packet traffic in modern networks is strongly auto-correlated and there exists a long-range dependency, i.e. persistence in their correlation structures does not die even for large lags.

For a stochastic process X = (Xt: t=0,1,2,…) to be a second order (weak or in covariance or in wide-sense) stationary, it is sufficient to have the existence of a stationary mean, a stationary and finite variance and a stationary auto-covariance function.

It is possible to think of a packet traffic process X consisting of a set {Xt}, where Xt is the number of packets that arrive in the t-th time unit.

Let X(m) = (Xj (m) : j =1,2,3…) for each m=1,2,3…, be the new second order stationary process, obtained by averaging the original process X over non-overlapping blocks of size m. By analysing the statistic properties of process X(m), the nature of process X can be easily determined.

Process X can be a short-memory process (short-range dependence) or a long-memory process (long-range dependence).

It can be shown that for long-range dependent processes auto-correlation of processes X(m) does not asymptotically depend on m (note that considering only stationary processes, the term "auto-covariance function" can be used the term "auto-correlation function" instead). This property is called asymptotic second-order self-similarity. So, long-range dependency implies asymptotic second-order self-similarity.

The long-range dependence process X is said to be exactly second-order self-similar (or fractal), if the auto-correlation of processes X(m) does not depend on m at all.

From an intuitive point, possibly the most enlightening property is that the averaged process X(m) takes a non-degenerated correlation structure for large m. An implication is that the averaged process will not appear as white noise. Instead, the (typical) aggregated traffic will have bursty subperiods and less bursty subperiods for small, as well large time-scales.

1.4 The ON-OFF models with general distributions

In this paragraph a general description of ON-OFF model with arbitrary distributions is provided.

An ON-OFF model with generic distributions is constituted by an alternation of two periods: an active or ON period, in which the source continuously transmits packets and a silent or OFF period, in which no packets are transmitted. Besides, the duration of any periods (silent or active) is independent on the duration of any others. In other words the model is completely uncorrelated at burst level but, the process is correlated at packet level (as explained after on).

Let Pa and Ps the probability mass functions of the active and silent periods respectively, the average values Na and Ns (in number of packets) for the two periods can be easily calculated by using the expectation function E[x] in terns of the MBL (Maximum Burst Length) the maximum duration of the active period and the MSP (Maximum Silent Period) the maximum duration of the silent period,

The normalized average rate M is expressed by the following:

$M = Na/(Na+Ns)$

The rate M is not obviously sufficient to specify the behaviour of a ON-OFF traffic model, it is also necessary consider its burstiness characteristics.

Burstiness is more a concept than a parameter. It expresses the degree of variability of the traffic rate generated by the source. In general, packet network can support a wide range of different traffic sources. Each one with own bursty characteristics, that strongly impact on the network performance evaluation. For this task some definition of burstiness has to be introduced. Several choices are possible, the most appropriate depending on the specific network context. In this description a measure of the inter-arrival time variations of source packets is adopted, since the sequence of the inter-arrival times are closely related with the evolution of the instantaneous rate of the involved process.

It can be shown that the burstiness B, of an ON-OFF source obeys a very simple structure. This was basically obtained by previously constructing an exact Markovian representation of the sequence of inter-arrival times. The expression is the following:

$B = M(1-M)b$

$b = (1+c2)Ns - (1-M)/M$

where, c2 is the squared coefficient of variation of the silent period distribution. It is particularly noticeable in this expression that the variance of the active period distribution does not affect the burstiness of the process (intuitively, the greater the average duration of the silent period and its variation, the greater the burstiness, at the same average rate M).

With regards to autocorrelation, the definition is unique. It is particular noticeable the fact that ON-OFF models with general distributions capture correlation at packet level, though they are uncorrelated at burst level. The Markovian representation of the inter-arrival times has also been used as before. If k denotes the lag, it can be shown that the autocorrelation function of these models can be expressed in the following exact way:

$\Phi(k) = [t(k)-1/Na] / (1+ c2-1/Na), \forall k \geq 1$

Where t(k) is an auxiliary function recursively evaluated from the component of the probability mass function of the active period:

$t(k)= \sum l \ \ Pa(l)\ t(k-l) \ \ (l=1,..,k);\ t(0)=1$

The following observations about the autocorrelation function can be done:

1. as mentioned before the ON-OFF models with general distributions are uncorrelated at burst level, but they may capture correlation at packet level, even high correlation
2. the autocorrelation function strongly depends on the distribution of the active period, in a very particular way: autocorrelation until lag k only depends on the probability mass function until sample k
3. the specific silent period distribution appears, neither in burstiness nor in autocorrelation functions. Just the variance has to be considered.

For these reasons and its simplicity the ON-OFF model family have found a large diffusion in various network research activities.

1.5 Conversational speech model

To more efficiently use bandwidth, voice-over-IP networks employ functionality referred to as silence suppression or voice activity detection. A voice activity detector (VAD) is a component of a voice terminal that suppresses the packetization of voice signals between individual speech utterances, such as during the silent periods in a voice conversation. VADs can often adapt to varying levels of noise vs. voice. Similarly to adaptive jitter buffers and echo cancellers, VADs can converge on appropriate thresholds to optimize their performance for a given voice conversation. Since human conversations are essentially half-duplex in the long term, the use of a VAD can realize approximately 50 percent reduction in bandwidth requirements over an aggregation of channels. Figure 9 depicts the behaviour of a VAD and its parameters.

While a VAD performance does not affect clarity directly, if it is not operating correctly it can certainly decrease the intelligibility of voice signals and overall conversation quality. Excessive front end clipping (FEC), for example, can make it difficult to understand what is said. Excessive hold-over time (HOT) can reduce network efficiency, and too little hold-over time can cause speech utterances to ”feel” choppy and unconnected when cutting in, even in short speech pauses.

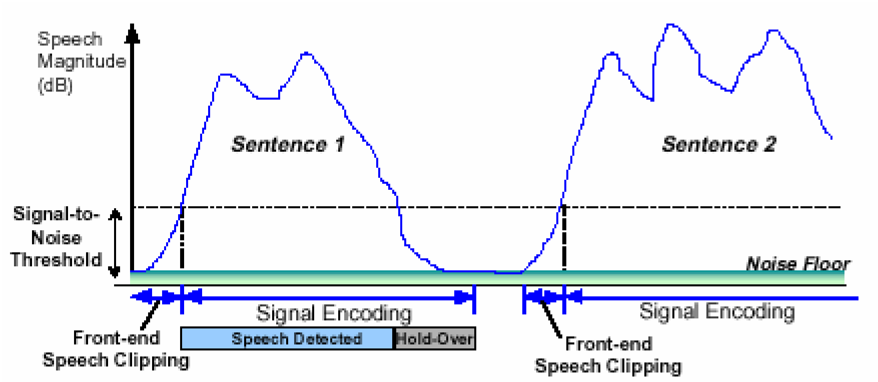


Figure 9: Voice activity detector behaviour.

Nevertheless, VADs are going to be widely deployed in modern voice over packet networks, to better exploit the available bandwidth.

At this point clearly appear, as the most suitable model for a conversational speech (voice source) is an ON-OFF model, in which the OFF state represents the silent period of the speech. ON-OFF model family has been characterized in detail in the previous paragraphs, now remains to properly set the relevant parameters: period duration, in terms of average value and statistical distribution and source transmission rate in the active period.

In particular, the reported results (extracted from [ITU-T RecP59]) are related to period duration characterization, rather than to the source transmission rate, because the latest does not depend on the voice nature but essentially on the used coding technique. Several coders could be deployed in the MEMBRANE test-bed, for example: G.711 (the former PCM at 64 Kbit/s), G.729 (8 Kbit/s) and G..723 (6.3 or 5.3 Kbit/s).

1.5.1 Characteristics of human conversational speech

The durations of talk-spurt and pause vary according to the measurement conditions. The following specifies two values for each parameter in conversational speech. One is based on measurement of speech without hangover time, while the other is from that with hangover time.

1.5.1.1 Characteristics measured without hangover time

1) Talk-spurt characteristics

The probability mass function (pmf) of talk-spurt duration can be modelled by two weighted geometric pmfs:

$$f_1(k) = C_1(1 - U_1)\, U_1^{k-1} + C_2(1 - U_2)\, U_2^{k-1}, k = 1, 2, 3, \ldots$$

where

$$C_1 = 0.60278 \quad U_1 = 0.92446$$
$$C_2 = 0.39817 \quad U_2 = 0.98916$$

Every increment of the variable k is equal to 5 ms. The cumulative distribution function of talk-spurt durations is shown in diagram a) of Figure 10 The average talk-spurt duration is 227 ms.

2) Pause characteristics

The pmf of pause duration is also modelled by two weighted geometric pmfs:

$$f_p(k) = D_1(1 - W_1)\, W_1^{k-1} + D_2(1 - W_2)\, W_2^{k-1}, k = 1, 2, 3, \ldots$$

where

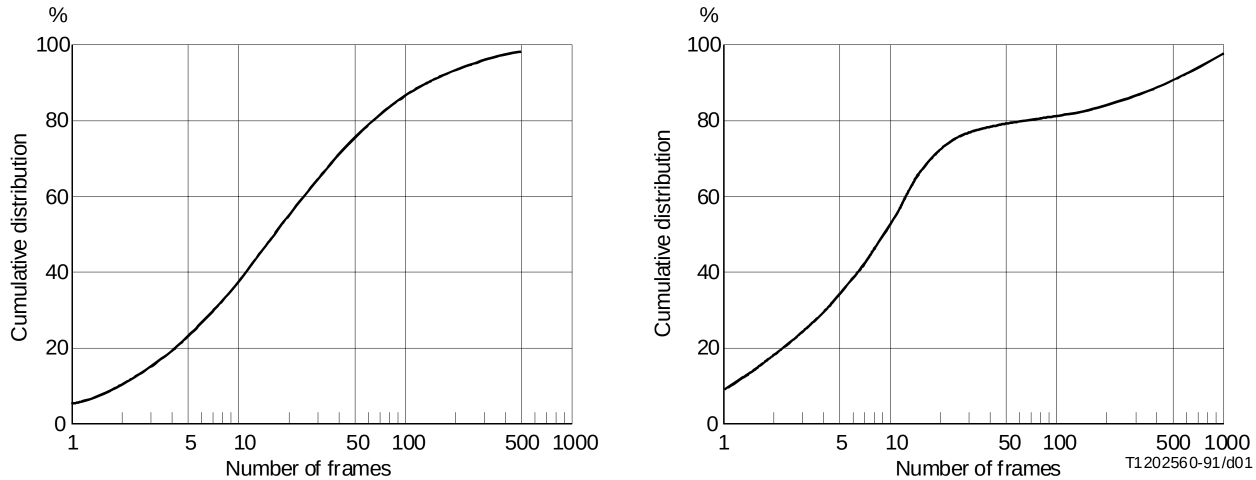
$$D_1 = 0.76693 \quad W_1 = 0.89700$$
$$D_2 = 0.23307 \quad W_2 = 0.99791$$

The cumulative distribution function of pause duration is shown in diagram b) of Figure 10

3) Activity factor

The average pause duration of 596 ms, combined with the 227 ms average talk-spurt duration, yields a long-term speech activity factor of 27.6 per cent.

It is to be noted that this value is measured by a meter without hangover. Otherwise, if the meter is with hangover a higher activity factor is to be expected (see Table 2)



a) Talk-spurt duration in 5 ms frames

b) Pause duration in 5 ms frames

FIGURE 1/P.59

Cumulative distribution of talk-spurt and pause durations (without hangover time)

Figure 10: Talk-spurt and pause durations.

1.5.1.2 Characteristics measured with hangover time

Table 1 lists the values of feature parameters in human conversational speech. These values were obtained by averaging the values reported in [LEE UN]-[CCITT COM-64].

Temporal parameters in conversational speech (average for English, Italian, and Japanese)

Table 2: Measured speech related temporal parameters.

Parameter	Duration (s)	Rate (%)
Talk-spurt	1.004	38.53
Pause	1.587	61.47
Double talk	0.228	6.59
Mutual silence	0.508	22.48

The cumulative distribution function of talk-spurt duration is approximated by an exponential function and that of pause durations is approximated by a constant-plus-exponential. That is, for talk-spurt:

$$Pts(t) = 1 - \exp(-Ats \quad t)$$

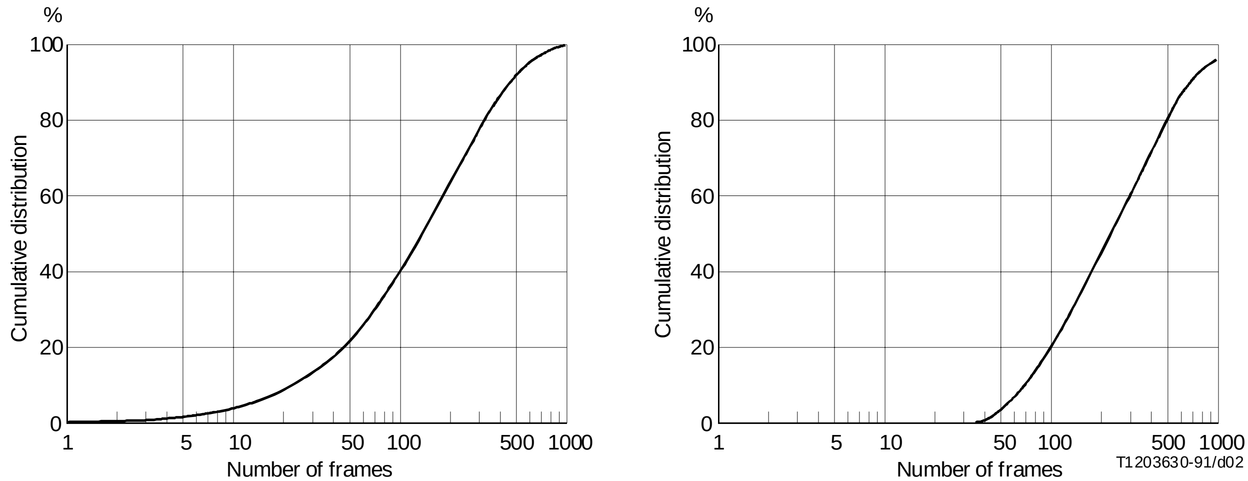
Ats = 1/Ttsm, Ttsm: average talk-spurt duration,

and for pause:

$$Pps(t) = \begin{cases} 0 & \text{for } 0 \le t \le 0.2 \\ 1 - \exp[-Aps(t - 0.2)] & \text{for } t > 0.2 \end{cases}$$

Aps = 1/(Tpsm – 0.2) Tpsm: average pause duration.

Both characteristics are shown in Figure 11.



a) Talk-spurt duration in 5 ms frames

b) Pause duration in 5 ms frames

FIGURE 2/P.59

Cumulative distribution of talk-spurt and pause durations (with hangover time)

Figure 11: Cumulative distribution functions.

1.6 MPEG-2 audio/video traffic modelling

The MPEG-2 system standard defines methods for multiplexing one or more elementary audio and video streams. At the time of encoding, audio samples and video frames are separately encoded. As far as an audio source is concerned, if it is a voice source, the MPEG-2 encoder can use a voice activity detector to reveal speech and silence periods. During a silent period it will not emit. The input of the MPEG-2 video encoder is a sequence of frames. The encoder output is a deterministic periodic sequence of Group of Pictures (GoPs) realized with three types of pictures: I, B and P frames. In the analysis it is assumed the GoP: I, B, B, P, B, B. After encoding, each media stream is packetized and time stamps are added to form the packetized elementary streams (PESs) PESs for audio and video are then multiplexed into the MPEG-2 multiplexer, to form a single output stream, called MPEG-2 ISO/IEC 13818 stream.

1.6.1 Modelling MPEG-2 multiplexer input traffic

When the performance of a network loaded by multimedia traffic is to be evaluated, the multimedia sources cannot be modelled as the superposition of independent monomedia emission process: correlation between processes has to be taken into account. In a MPEG-2 source consisting of VBR coded sources, variations occur in the statistical parameters characterizing each of the component traffic streams because of the activity changes of the other component traffic streams. Because these activity changes in both voice and video are due to the same user behaviour, when a multimedia source is to be modelled, they have to be considered as correlated events. Source activity changes have been observed and modelled in network research, both for video and voice traffic through Markov-based processes constituted by N sub-processes, one for each activity state of the source and the transitions between the above sub-processes to model activity changes in the source.

To this purpose, in order to obtain a model of the multimedia source entering the MPEG-2 multiplexer, it is necessary not only to model the monomedia voice and video sources, highlighting the activity changes in each sources, but also take into account the inter-media correlation, that is, considering the activity changes of the component monomedia sources to be correlated.

In the proposed modelling, it is specifically used the most general Markov-based model in the discrete-time domain, for the emission process, the switched batch Bernoulli process (SBBP). This is an arrival process, V(n), representing the number of packets emitted in the slot n, modulated by an underlying Markov chain with a certain state space. An SBBP is characterized by a parameter set {Q, B}, where Q is the transition probability matrix of the state of the Markov chain while B is the matrix describing the emission process in each state, that is, its generic element, brs, represents the probability that the process emits packets when the state of the Markovian chain at the slot n, S(n), is s (see [LomMorPal] for more details).

With regards to the voice stream, source changes mainly occur when the average length of the pauses between two successive voice talk-spurts detected by a VAD changes. Therefore a voice source can be modelled by means of an SBBP, whose Markovian chain is constituted by a number of macrostates each modelling one of the different activities of the voice source. Each macrostate contains the states OFF and ON of the relating activity. A transition between two of these macrostates models an activity change in the source. Say ϕ the set of activity states and ψ={OFF,ON}, the state S(n) belongs to the set, Cartesian product of the two component subset $\phi\times\psi$, state space of the Markovian chain of the SBBP modelling the voice source.

A voice source is characterized by:

T(a,OFF) and T(a,ON) $\forall a\in\phi$, the average duration of the states OFF and ON, respectively, during the activity macrostate a;

Ta $\forall a\in\phi$, the average duration of the activity macrostate a;

qa,b $\forall a,b\in\phi$, the probability that the activity of the voice source moves from a to b, provided that the activity leaves a;

P(a,ON) ,the emission packet rate during the ON periods.

Let Δ the slot duration, equal to the time needed for the voice source to emit one voice sample, the parameter set of the SBBP can be calculated as function of the previously defined parameters themselves (see again [LomMorPal] for more details and explanations).

With regards to the video source modelling, as an MPEG-2 video source emits frames according to a fixed sequence given by the GoP structure, an activity change in this kind of source determines a variation in the probability function (pdf) of the emission process of each frame. For this reason, the main features characterizing MPEG video traffic are: the gamma shape of the emission pdf for all kinds of frames and the dependence of the mean value and the variance of the emission pdf of B and P frames on the value of the I frame of the same GoP. Therefore the I frame value in each GoP determines the activity in the whole GoP. Previous studies have successfully proposed an SBBP model for MPEG video traffic with a Markovian chain constituted by a number of macrostates, each characterizing an activity state grouping a number of states, one for each frame in a GoP. A transition between two of these macrostates models an activity change in the source.

Let ϕ be the set of activity macrostates and ψ the set of states within each activity macrostate. If a GoP of 6 frames is considered, ψ={I,B,B,P,B,B}.The states space of the Markovian chain of the SBBP modelling the voice source is determined by the Cartesian product $\phi\times\psi$.

A video source is characterized by:

μI(a), μB(a), μP(a), σI2(a), σB2(a) and σP2(a) $\forall a\in\phi$, the mean values and variances of the gamma functions pdfs of the I, B and P frames, respectively, when the activity macrostates is a;

Ta $\forall a\in\phi$, the average duration of the activity macrostate a;

qa,b $\forall a,b\in\phi$, the probability that the activity of the video source moves from a to b, provided that the activity leaves the macrostate a;

Let Δ the slot duration, equal to the time duration of one frame, the parameter set of the SBBP can be calculated as function of the previously defined parameters themselves (see [PaxFlo]).

1.6.2 Multimedia MPEG-2 source model

In order to model a multimedia MPEG-2 source from the voice and video models defined above, the intermedia correlation has to be captured; on the basis of what has been said so far this resides in the inter-correlation between the monomedia SBBP processes with each others by expressing the SBBP models of the voice and the video sources in the same time scale, that is, with the same slot duration Δ and by defining, for each monomedia process, the probabilities of transition between two activity macrostates as a function of the activity macrostate of the other monomedia sources.

Since in the majority of multimedia applications there is one monomedia source (the master) which is completely independent of the others (the slaves) and these are dependent only on the first one, the dependence of the activity macrostate will be in the slave source only, as a function of the master activity. If M(n) is the emission process modelling the master and the slave sources obtained by superpositioning and correlating the previously proposed SBBPs, on a new common time scale with a slot of Δ, the transition probability matrix of the slave source (dependent on the activity macrostate of the master source) can be calculated through the results of the previous analysis (illustrated in detail in [LomMorPal])

A complete mathematical analysis and statistical characterization of process M(n) can be again found in [LomMorPal]; hereafter the numerical results related to a case study relevant for the issues of MEMBRANE project, are reported (such results can be also directly and profitably exploited during the project studies and trials).

Let us consider a distance learning session: a lecturer is filmed and both voice and video are encoded with the same MPEG-2 encoder. For the voice encoding a CBR encoder with a VAD is used; video is encoded with a GoP of six frames, like the one taken as an example in the proposed modelling of this paragraph. It is possible to note two different kinds of behaviour on the part of the lecturer: one when he speaks very slowly in order to comment what he is writing on the blackboard and another when the lecturer speaks to the students and is facing them.

These two kinds of behaviour determine two different statistics in the emission process of both the voice and video sources. Therefore, two different activity macrostates in modelling each monomedia traffic stream can be used.

Let L1 and H1 be the two activity macrostates modelling the voice emission process; L1 presents a low emission activity, in order to model slow speaking periods. While H1 presents a high emission activity. Likewise, let L2 and H2 be the two activity macrostates modelling the video emission process; macrostate L2 is characterized by a high intra- and inter-frame correlation, that is, by a low emission activity, in order to model source emission when the blackboard takes up most of the image and the lecturer's movements are slow; macrostate H2 is characterized by certain statistics, the video is taken as the master source.

The durations measured for the macrostates L2 and H2 for the video source are 36.4 sec and 61.6 sec respectively, while the durations measured for the macrostate L1 and H1 for the voice source are 10.5 sec and 9.6 sec respectively, when the video source is in the macrostate L2, 9.1 sec and 16.4 sec respectively, when the video source is in the macrostate H2.

In [LomMorPal] are reported all the average and variance values for the different kinds of frame in the two macrostates L2 and H2 . For the sake of simplicity a single statistical distributions can be used for each activity macrostate instead, such as a normal distribution with average and variance values dependent on the specific case; also the statistical distribution of the macrostate durations can be simple exponentials, of specified mean values.

The ON and OFF periods of the voice source in the macrostates L1 and H1 can be effectively modelled by exponential distributions with measured average values of 0.277 sec and 1.04 sec respectively, in the macrostate L1, 027 sec and 0.327 sec respectively, in the macrostate H1. The emission rate in the ON periods depends on the specific case.

Last but not least, with an acceptable approximation, the duration of the activity macrostates L1 and H1 for the slave source can be assumed exponentially distributed, with mean values as previously specified (in dependence of the activity macrostate of the master source L2 and H2).

1.7 H.261/H.263 audio/video traffic modelling

Video, like voice, is naturally bursty, but can be manipulated to give a constant output bit rate. To gain the maximum efficiency from a packet-switched network only the real amount of information should be sent. This will change as the information content of the picture changes and so a variable bit rate output from the codec is expected. To gain maximum compression a single one layer codec could be used. Sudden movements in the picture or scene changes will produce large information transfers and so will produce high bit rates. Small changes from frame to frame will result in low bit rates.

In IP networks it is sometimes hard to characterize the sources. Performance evaluation of the network becomes difficult and subsequent guarantees of quality of service are hard to achieve. However a QoS for a constant bit rate source might be easier to guarantee than that of a variable source. This has led to the development of two layered codecs. One layer has a constant bit rate, called the base layer, and delivers a minimum QoS to the user. The second layer contains the enhancement to the base layer and is bursty in nature. This improves the quality of the picture to the user. The relevant quality of pictures of the two schemes is subject to the coding involved in the codecs, the bandwidth available and the probability of packet loss.

The empirical data on which the models presented in this paragraph are based, were generated using algorithms implemented as in [AJSmith]. These are one and two layer implementations of the H.261/H.263 codec standard. This video compression standard can handle a variety of service quality from low quality video telephony (64 kb/s) to video conferencing (384 kb/s) and higher quality video (2 Mb/s) using a bit rate defined by p*64 kb/s where p=1...31.

Some of the main elements of H.261/H.263 include inter-frame and intra-frame prediction, motion compensation, discrete cosine transformation (DCT), quantization of DCT coefficients and run length coding of the DCT coefficients. Interframe prediction takes advantage of the fact that consecutive frames will be quite similar allowing just the difference between frames to be transmitted rather than the entire frame data. In contrast intraframe prediction occurs when consecutive frames are substantially different, for example at the start of a video sequence or a sudden motion/scene change.

The video sequences considered for the numerical results are of just a single person talking with some zooming motion. The spatial resolution of the sequence is CIF (384, 288) and the temporal resolution 25 Hz. In order to capture the data set for the encoded video sequences, on a per frame basis, the outputs of the codecs are sampled every 0.04 seconds. This produced approximately 7500 samples to be considered. It can be seen from Figure 12 below that the bit rate of this one layer codec varies considerably over the time interval.

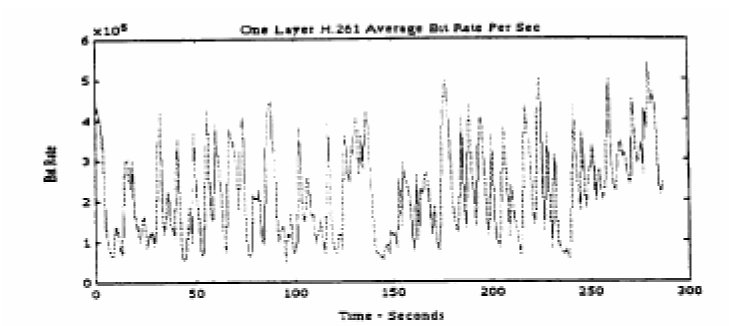


Figure 12: Bit Rate for a One-Layer H,261 codec.

Even though the H.261/H.263 standard is complied with, there is scope for different codecs. The two codecs discussed in particular in this subsection are the one layer 256 kb/s source and the two layer 384 kb/s source: they can be considered as a reference for an easy extension to different codec layer rates. The main characteristics of the sources are:

Table 3: Video source characteristics.

	One Layer	Two Layers	
		Base	Enhanc.
Min Bit Rate (kb/s)	222	369	39
Standard Deviation	Mean	0	High

1.7.1 Modelling the video source

The one and two layered codecs are considered separately. For the one layer model a discrete state Markov chain is designed. This follows from works which verified that, models based on multi-state Markov chains are sufficiently accurate for traffic studies of this type. The frame is taken as the basic unit of time for the chain. The objective is to capture the burstiness of the source.

The proposed Markov chain model is produced by carrying out the following analysis on the empirical data. The first issue is to examine the probability density function of the size of frames produced. A comparable measure is the number of frames versus the frame size normalized to bits per second as is shown in Figure 13 below.

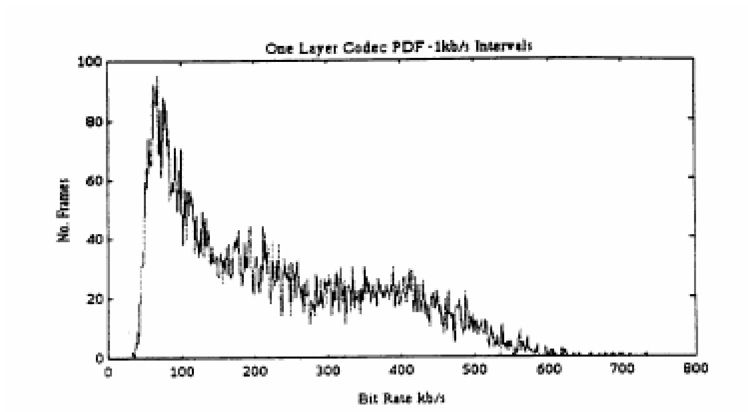


Figure 13: Shape of one layer codec probability density function

A few criteria can be used to select the states of the Markov chain. It is intuitive that at least five states are needed to capture the shape of the pdf but the number should not be too large as the model would become complicated. It is felt that all states should be close to being equiprobable so that for short runs they can all be reached.

If the number of states is small, then a birth/death process can be easily achieved but, if the number of states is not large enough, the assumption of a birth/death process itself fails. A possible number of states, chosen in this analysis, is nine (see [MurTea]), which is a compromise between keeping a small number of states and also trying to have a birth/death process.

The birth/death process greatly simplifies the modelling of the Markov chain.

Careful selection of the nine states results in a certain transition matrix, which presents small entries at a distance more than one from diagonal. The assumption made induce to neglect these values. However the entries in the matrix must sum to unity and so a method of proportioning the excess to the other entries has to be found. This can be done by starting at either end of the Markov chain and conditioning on only the allowed transitions to the nearest neighbours. By starting at either end there would be two slightly different matrixes. A possible resulting matrix is shown in Table 4.

Table 4: Transition probability matrix for the one layer

State	State0	State1	State2	State3	State4	State5	State6	State7	State8
State0	0.81	0.19							
State1	0.18	0.59	0.23						
State2		0.19	0.60	0.21					
State3			0.18	0.57	0.25				
State4				0.22	0.57	0.21			
State5					0.22	0.60	0.18		
State6					0.17	0.75	0.08		
State7							0.17	0.81	0.02
State8								0.55	0.45

To ensure that the states are almost equiprobable, states have been chosen so that the total frame count in each state do not exceed 15 percent of the total frame count.

Within each state there can be an upper bound and lower bound on the bit rate of the source.

The indicated transition matrix can be effectively exploited in a general case, in such a way. A normal distribution of the source emission rate in each state can be assumed and considering an equal emission rate difference, in absolute value, between the average rate of adjacent states, it is possible to obtain the mean rate of all the states by solving simple linear equations, where the average source emission rate and other parameters, such as source variance or source peak rate, are set.

The two layer codec is modelled differently. The pdf of the two layers is shown in Figure 14.

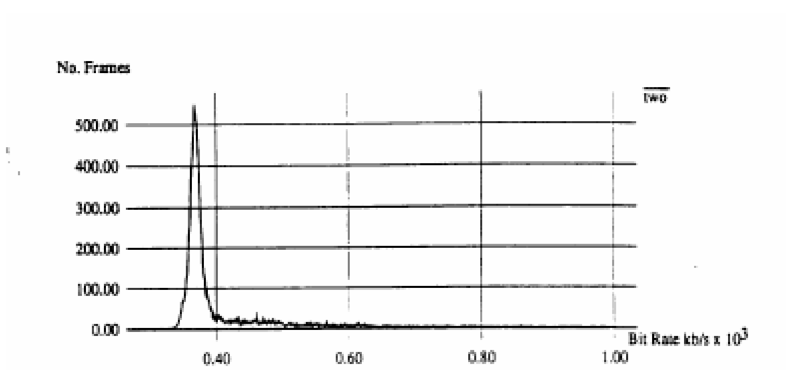


Figure 14: PDF for the Two Layer Codec

The base layer is almost a constant bit rate and so it is modelled by just one state. Within that state the bit rate is uniformly distributed, in the specific case, from 350 kb/s to 388 kb/s giving the correct mean rate (of 369 kb/s in the analyzed data). The enhancement layer could be modelled by a Markovian chain, similarly to the one layer codec. However a simpler choice (see [MurTea] for more details) consists in modelling the two layers by means of a single hyper-exponential distribution. The hyper-exponential function has to produce a certain mean bit rate (39 kb/s in the case). To fit the distribution to the data also a proper standard deviation has to be chosen (124 kb/s for the reported data).

1.8 Modelling Telnet connections

Network connections such as telnet and more in general TCP traffic can be modelled using the M/G/ queue.

In the M/G/ queue customers (new connections) arrive according to a poissonian process. The required service time is obtained from a heavy tailed distribution with infinite variance. In fact, empirical distributions of telnet packet inter-arrival times have shown that they are heavily tailed. Modelling telnet packet inter-arrival times with exponential distribution underestimates the burstiness of the traffic for a single connection (as well as that of multiplexed traffic).

The M/G/ model implies that multiplexing constant-rate connections that have poissonian connection arrivals and a heavy tailed distribution for connection lifetimes would result in self-similar traffic [PaxsonFloyd].

The auto correlation function r(k) for the arrival process is as follows:

$r(k) = cov[X(t)X(t+k)] = \int 1 - F \; dx$

where is the poissonian arrival rate, F the distribution of the service time (heavy tailed in this case) and X(t) the number of customers (simultaneously active connections) at time t (the integral spans from k to .

Therefore applications such as FTP, Telnet and WWW can be modelled in such a way that the sessions arrive in a Poisson manner and the duration or size of each session has heavy tailed distribution. This results in an asymptotically self-similar traffic i.e. the auto-correlation r(k) $k^{-(2-2H)}$ L(k) as k (where H is a constant, called "Hurst Parameter" and L(k) is a function that assumes only finite values).

First, the assumption is made that the power spectrum of the time series corresponds to a Fractional Gaussian Noise Model. The second step is to create a series of complex numbers (zi) corresponding to the FGN power spectrum. Thirdly inverse discrete Fourier transform technique is applied to obtain the time series equivalent.

The difficulty behind this approach is to accurately compute zi which corresponds the FGN power spectrum. In terms of speed [ITU-T RecP59], a sample path of 32,768 points took about 11 seconds on a SPARC station IPX and 262,144 sample points took less than 2 minutes.

This method compared to other schemes such as the Random Mid Point Displacement method was twice as fast. Also in [KliMukSong], tests were carried for various values of the Hurst parameter (H) to see if the samples produced, matched what is expected for FGN. Using the Whittles estimating technique, the data generated using the fast Fourier transform method was consistent with FGN for the desired value of H. In comparison to the random mid point displacement method (RMD), RMD suffers from biases at certain values of the Hurst parameter which is not observed in the FFT technique.

The point to note is that, there is speed vs. accuracy tradeoff when synthesizing self-similar traffic. Aggregation of many ON-OFF sources is more suitable in a parallel computing environment and the generated traffic is close to the real time Ethernet traffic trace. The FFT and RMD technique is fast but results only in an approximate self-similar traffic.

1.9 WWW-client characterization

A primary goal is to obtain a clear picture of the size distribution of the documents available in general on the web, of the requests made by MEMBRANE potential users, and of the relationship between document size and popularity.

Previous studies of file sizes in general purpose Unix systems has shown that small sizes are much more common than large sizes. However, in view of the potential use of the Web for multimedia objects, documents on the Web could have a significantly different distribution from those in general purpose file systems.

On the whole, the most striking aspect of studies of the WWW is the dominance of the power law (or Pareto) distribution. The shape of the power-law distribution is a hyperbola; with parameter α its probability mass function is:

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}$$

and its cumulative distribution function is given by:

$$P[X \leq x] = 1 - (k/x)^{\alpha} \quad \alpha,k \geq 0,\ x \geq k$$

Power-low distributions have a number of properties that are qualitatively different from distributions that are more commonly encountered, such as exponential, normal or Poisson. If $\alpha \leq 2$, then the distribution has infinite variance; if $\alpha \leq 1$ then the distribution has infinite mean. So depending on α, an arbitrarily large portion of the probability mass may be present in the tail of the distribution, hence the name heavy-tailed. In practical terms a random variable that follows a heavy-tailed distribution can give rise to extremely large values with non-negligible probability.

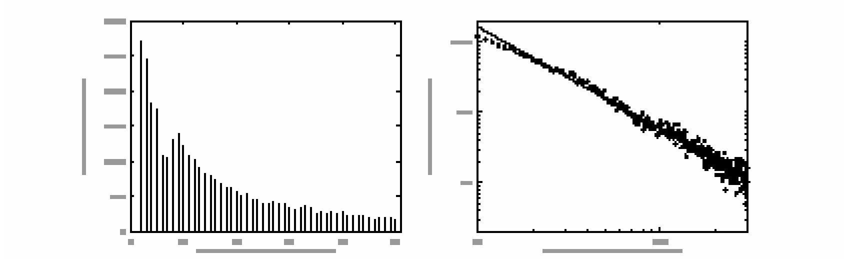


Figure 15: Document size distribution

Figure 15 shows the size distribution of objects referred to collected log files [CunBesCro]. Although this is not a random sample of documents available on the Web, anyway it provides a reasonable estimate of the actual size distribution of Web documents. On the left of the figure is a histogram of file sizes up to 6400 bytes; on the right is a histogram on a log-log scale of file sizes of 1280 bytes or more.

The figure shows the pronounced hyperbolic distribution of file sizes.

A comparison of WWW document sizes with the file size distribution that might be found in a typical Unix file system is instructive.

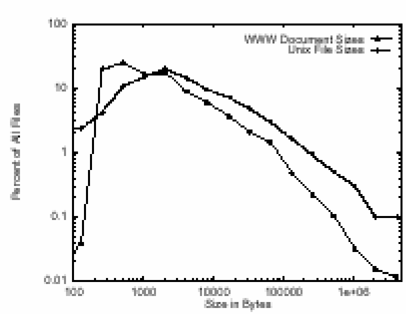


Figure 16: Comparison of Unix file sizes with WWW documents sizes

In Figure 16the distribution of document sizes found in the Web and that of data file are compared. It is clear as in the Web there is a stronger preference for small files than in Unix file systems. The Web strongly favours documents in the 256 to 512 byte range, while Unix files are more commonly in the 1 KB to 4 KB range. More importantly, the tail of the distribution of WWW files is not nearly as heavy as the tail of the distribution of Unix files.

Thus, despite the inclusion of multimedia in the Web, It is possible to conclude that Web documents are currently more biased toward small files than are typical Unix file systems.

Related to the question of document size distribution are also questions involving user preferences: the relationship between size and popularity, and the popularity of individual documents.

To explore the influence of user choice on the distribution of documents actually transferred through the network, it is significant to measure the relationship between the number of times a document is accessed and the size of the document. Figure 17shows a plot of the average number of times documents of a given size were referenced (also collected in [CunBesCro]). The data shows that there is an inverse correlation between file size and file popularity. The line depicted is a least square fit to a log-log transform of the data (y−x-0.33). This fit is statistically significant at a 99.9% level, but only explains a small part of the total variation.

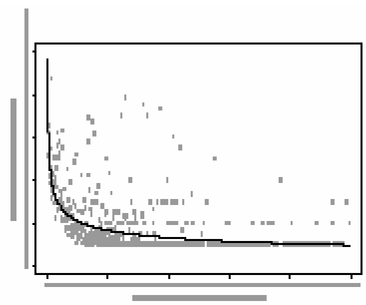


Figure 17: Average number of requests for file sizes

Thus, in addition to the tendency for WWW documents as created to be small, users additionally prefer small documents. The combined effect of these two trends is shown in Figure 18 (on the left linear scale of file sizes up to 6400 bytes, on the right log-log scale for file sizes from 1280 bytes). It depicts the distribution of document requests made by users, by request size: the actual document traffic generated by user requests also follows a hyperbolic distribution (y−x-1.66).

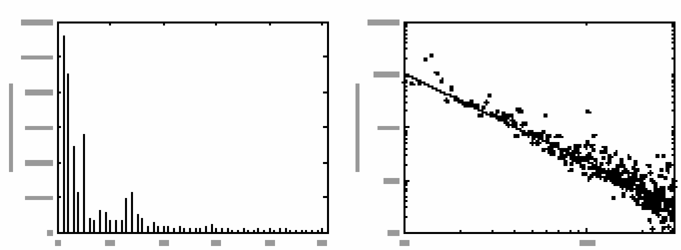


Figure 18: Requests for document by document size

It is worthwhile to note that at the present time (dated to some years ago), the distribution of transfer sizes is more strongly determined by document sizes than by user preferences, although both contribute. The final instance of hyperbolic distributions in the collected data occurs as an instance of Zipf's law. Zipf's law was originally applied to the relationship between a word's popularity in terms of rank and its frequency of use. It states that if one ranks the popularity of words used in a given text (denoted by ρ) by their frequency of use (denoted by P) then:

$P - 1/\rho$

Note that this distribution is parameter-less, i.e., ρ is raised to exactly -1, so that the nth most popular document is exactly twice as popular as the 2nth most popular document. Zipf's law has subsequently been applied to other examples of popularity in the social sciences.

The collected data shows that Zipf's law applies quite strongly to documents on the WWW. This is demonstrated in Figure 8 for all 46,830 documents referenced in the logs (It shows a log-log plot of references to each document as a function of the document's rank in overall popularity). The tightness of the fit to a straight line is remarkable, as is the slope of the line: -0.986. Thus the exponent relating popularity to rank for WWW documents is very nearly -1, as predicted by Zipf's law.

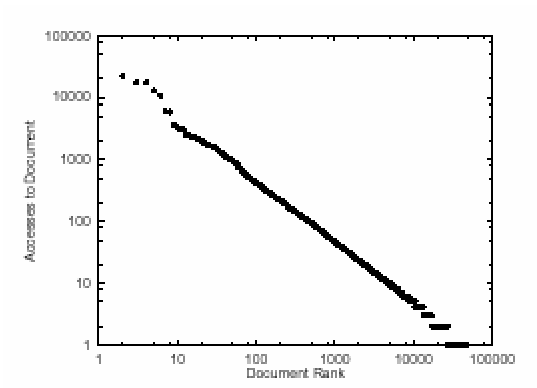


Figure 19: Zipf's law applied to WWW documents

From the gathered data [WilTaqShe] can be also deduced that user preference for small documents is not due to a preference for HTML files over images; in fact, the reverse is true: image files are by far the most commonly requested object, and they tend to be significantly larger than HTML files.

1.10 On the relationship between file sizes, transport protocols and self-similar network traffic

As mentioned before, recent measurements of local-area and wide-area traffic [WilinTaqqSher] have shown that network traffic exhibits variability at a wide range of scales. Such scale-invariant variability is in strong contrast to traditional models of network traffic, which show variability at short time scales but are essentially smooth at large time scales. This effect is described statistically as long-range dependence, and time series showing this effect are said to be self-similar. Since self-similarity is believed to have a significant impact on network performance, understanding the causes and effects of traffic self- similarity is an important problem.

In [ParKimCro], it is studied and analyzed a mechanism that induces self-similarity in network traffic. It is shown that self-similar traffic can arise from a simple, high-level property of the overall system: the heavy-tailed distribution of file sizes being transferred over the network. When the distribution of file sizes is heavy-tailed (meaning that the distribution behaves like a power law thus generating very large file transfers with non negligible probability) then the superposition of many file transfers in a client/server network environment induces self-similar traffic and this causal mechanism is robust with respect to changes in network configuration. Properties of the transport/network layer in the protocol stack, however, it is shown to play an important role with respect to preserving and modulating this causal relationship. The mechanism proposed in [Paxson] is motivated by the ON/OFF nature of the traffic. The ON/OFF model shows that self-similarity can arise in an idealized context with unbounded resources and independent traffic sources as a result of aggregating a large number of ON/OFF traffic streams whose ON or OFF periods are heavy-tailed. The success of the simple ON/OFF model in capturing the characteristics of measured traffic traces, such as files and messages transfer, is surprising given that it ignores interaction among different traffic sources contending for network resources which in real networks can be as complicated as the feedback congestion control algorithm of TCP.

To establish the link between file (or message) sizes and traffic self-similarity, it is necessary to show that the degree of self-similarity varies as a direct result of variation in file (or message) size distribution.

In [ParKimCro], it is shown that in a “realistic" client/server network environment (i.e., one with bounded resources leading to the coupling of multiple traffic sources contending for shared resources) the degree to which file (or message) sizes are heavy-tailed directly determines the degree of traffic self-similarity. Specifically, measuring self-similarity via the Hurst parameter H and file (or message) size distribution by its power-law exponent α, it is verified that there is a nearly linear relationship between H and α over a wide range of network conditions when subject to the influence of the protocol stack. This mechanism gives a particularly simple explanation of why self-similar network traffic may be observed in many diverse contexts. This relationship is robust in the sense that it is present over a wide range of conditions including changes in bottleneck bandwidth and buffer capacity, interference from cross-traffic possessing dissimilar traffic characteristics, and changes in the distribution of conversation interarrival times.

For example, if self-similar traffic is mixed with cross-traffic that is only short-range dependent, hence smooth at large time-scales, then self-similarity remains persistent in the aggregated traffic.

In addition, if the interarrival time distribution is very heavy-tailed, it can amplify traffic self-similarity when file (or message) sizes are only moderately heavy-tailed. However, if the size distribution is very heavy-tailed, then the interarrival time distribution has virtually no effect on traffic self-similarity.

It is also demonstrated in [ParKimCro] that, in practice, the presence of self-similarity depends on whether reliable and flow-controlled communication is employed at the transport layer. For example, in the absence of reliability and flow control mechanisms such as when a UDP-based transport protocol is used, much of the self-similarity of downstream traffic is destroyed as compared to the case of upstream traffic. The resulting traffic, while still bursty at short ranges, shows significantly less long-range correlation structure. In contrast, when TCP (Reno, Tahoe, or Vegas) is employed, the long-range dependence structure induced by heavy-tailed file (or message) size distributions is preserved and transferred to the link-layer, manifesting itself as scale-invariant burstiness. In essence, this is a combined effect of the input traffic being flow controlled and conserved through retransmission-based reliable transport.

Also, it is worthwhile to highlight the effect of self-similarity on network performance. It has been discovered that as self-similarity is increased in an UDP-based non-flow-controlled environment, performance declines drastically as measured by packet loss rate and mean queue length. However, if reliable communication via TCP is used, packet loss, retransmission rate, and file transmission time decline gracefully, i.e., roughly linearly as a function of H. The exception is mean queue length, which shows the same superlinear increase as in the unreliable non-flow-controlled communication case. This graceful decline in TCP performance under self-similar loads comes at a cost: a disproportionately increased consumption of buffer space. The sensitive dependence of mean queue length on self-similarity agrees with previous work showing that queue length distribution decays more slowly for long-range dependent traffic than for short-range dependent sources. The graceful performance decline exhibited by reliable communication is a result of shaping a large file and/or email transfer into an on average, “thin" packet train (stretching-in-time effect). Stretching-in-time is effected through the joint action of retransmission (or conservation) of lost packets, and the conservative nature of linear increase/exponential-decrease feedback congestion control [HKim]. This also suggests, in part, why the ON/OFF model has been so successful despite its lack of coupling among traffic sources (a principal effect of interaction among traffic sources in an internet-worked environment seems to lie in the generation of lengthy packet trains). Thus, the translation of large file (or messages) (transfers into well behaved elongated packet trains may be related to system-wide traffic self-similarity via the multiplexing of a large number of traffic streams (just 30 are sufficient to yield high Hurst parameter).

Finally, it is worth noting that the important issue whether file and/or message size distributions in practice are in fact typically heavy-tailed, and whether file and/or messages size access patterns can be modelled as randomly sampling from such distributions, seems to not have been completely addressed. Previous measurement-based studies of file systems have recognized that file size distributions possess long tails, but they have not explicitly examined the tails for power-law behaviour ([MSalyam], [AJSmith]).

However, evidence of heavy tails in the distribution of file sizes has been noted in some specific contexts. In [CrovBest], it is shown that the size distribution of files found in the World Wide Web appears to be heavy-tailed with α approximately equal to 1. This result is also in general agreement with measurements reported in other studies. A general study of Unix file systems has found size distributions that appear to approximate power-law distribution. Additional evidence of power-law behaviour is present in some data on transmission lengths of network transfers. The authors in [ArlWil] show that the sizes of reads and writes to an NFS server appear to show power-law behaviour. In [PaxsonFloyd], it was found that the upper tail of the distribution of data bytes in FTP bursts was well fit to a Pareto distribution with $0.9<\alpha<1.1$.

1.10.1 Synthesizing Self-Similar traffic

Many methods have been proposed to generate self-similar traffic which approximates the real time data trace. For example, in [TaqqWilSher] self-similar traffic is synthesized by superimposing various individual ON-OFF sources. The packet train produced by each individual ON-OFF source was synthesized by an individual processor in a parallel computing environment.

To produce a synthetic trace of length 100.000, it is required a massively parallel computer with more than 16.000 processors and the time required was the order of a few minutes.

Fast, efficient and accurate schemes are needed to synthesize this type of traffic. In [PaxFas] a technique to synthesize approximate self-similar traffic using the fast Fourier transform technique is presented. It is as follows.

First the assumption is made that the power spectrum of the time series corresponds to a Fractional Gaussian Noise Model. The second step is to create a series of complex numbers (zi) corresponding to the FGN power spectrum. Thirdly inverse discrete Fourier transform technique is applied to obtain the equivalent time series.

The difficulty behind this approach is to accurately compute zi which corresponds to the FGN power spectrum. In terms of speed [PaxFas], a sample path of 32,768 points took about 11 seconds on a SPARC station IPX and 262,144 sample points took less than 2 minutes.

This method compared to other schemes such as the Random Mid Point Displacement method is twice as fast. Also in [PaxFas], tests were carried for various values of the Hurst parameter (H) to see if the samples produced, matched what is expected for FGN. Using the Whittles estimating technique, the data generated using the fast Fourier transform method was consistent with FGN for the desired value of H. In comparison to the random mid point displacement method (RMD), RMD suffers from biases at certain values of the Hurst parameter which is not observed in the FFT technique.

The point to note is that, there is speed vs. accuracy trade-off when synthesizing self-similar traffic. Aggregation of many ON-OFF sources is more suitable in a parallel computing environment and the generated traffic is close to the real time traffic trace. The FFT and RMD techniques are fast but results only in an approximate self-similar traffic.

1.11 Modelling aggregated Ethernet traffic

In this paragraph a graphical demonstration (see [TaqqWilSher] for a mathematical proof) is given as to how by aggregating simple renewal ON-OFF processes results in self-similar behaviour. The aggregation of individual ON-OFF sources also allows for the explanation of observed self-similarity in aggregated Ethernet traffic.

In this particular case the generic traffic source is either transmitting packets at a constant rate R during the ON period or is idle in the OFF period. The time spent during the ON state (ton) and during the OFF state (toff) is i.i.d and in general to be considered a heavy tail distribution e.g. the Pareto distribution with finite mean and infinite variance (1< <2).

When a large number of these sources are aggregated it results in traffic having fractal (self-similar) characteristics and the model is called the Fractional Gaussian Noise (FGN) Model (a trace of FGN traffic is shown in Figure 20)

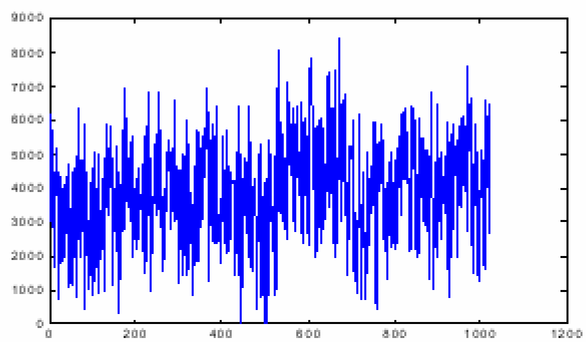


Figure 20: Trace of Fractional Gaussian Noise

The results in [TaqqWilSher] also provide evidence that for a large number of sources, the self similarity property observed in wide area Ethernet traffic doesn't depend on the underlying access schemes used.

The Hurst Parameter H represents the degree of self-similarity in the observed traffic. When the value of H is between 0.5 and 1 the traffic is said to be self-similar (values of H closer to 1 indicate a high degree of self-similarity). The relation between H and is given by H = (3-)/2.

When trying to fit the synthetic generated data to the actual WAN traffic certain parameters play crucial roles. The first one is which describes the intensity of self-similarity, once this is estimated H can be calculated using the above equation. The other important parameter is the number of sources (M), since the value of M is considered to be large it can be chosen of the order of some hundreds or even thousands. Other parameters to be considered are the rate at which packets are generated during the ON period and the lower cut-off of the Pareto distribution.

As shown in [TaqqWilSher], self-similar traffic generated by superimposing a number of ON-OFF sources, easily passes the "visual test" of actual Ethernet traffic. An intuitive feel for self-similarity can be obtained by observing the following traces at various time-scales (the zoom level used was a factor of 2).

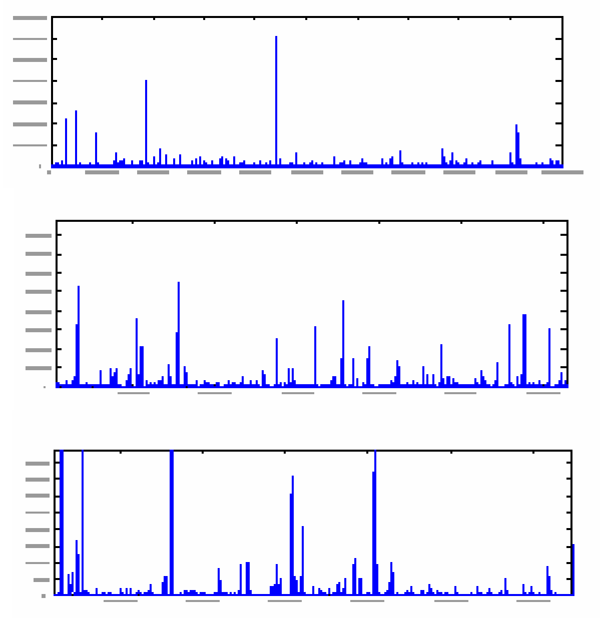


Figure 21: Self-Similarity behaviour of the aggregated traffic