
D5.6 – Phase II Experimental validation - Emulations

Deliverable Type *	: PU
Nature of Deliverable **	: Demo
Version***	: Released
Created	: 01 July 2010
Contributing Workpackages	: WP5
Editor	: Bart Dhoedt (IBBT)
Contributors/Author(s)	: Rodolfo Torrea Duran (IMEC), IBBT (Pieter Simoens, Farhan Azmat Ali, Bert Vankeirsbilck)
File Name	: [MobiThin_D5.6_WP5_30June2010_IBBT_V1.1]

- ***Deliverable type:**
 - *PU = Public,*
 - *RE = Restricted to a group of the specified Consortium,*
 - *PP = Restricted to other program participants (including Commission Services),*
 - *CO= Confidential, only for members of the MobiThin Consortium (including the Commission Services)*

- **** Nature of Deliverable:**
 - *P= Prototype,*
 - *R= Report,*
 - *S= Specification,*
 - *T= Tool,*
 - *O = Other.*

- *****Version:**
 - *Preliminary,*
 - *Draft 1, Draft 2,...,*
 - *Released*

Abstract:

This deliverable describes emulation based validation carried out in the second phase of the MobiThin project. This work mainly addresses the validation of approaches taken for optimising the wireless link in terms of power efficiency. Two emulation frameworks were intensively used, i.e. NS-2 network emulations focusing on optimisations by increasing the time spent in sleep time, and an emulation environment allowing to model the hardware specifics of the transceiver more accurately and to verify the implementability of a cross-layer algorithm.

This deliverable comes in parallel with D5.7, describing the validation work carried out based on lab-trials.

“The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 216946”

Keyword List: Mobile thin clients, experiments, wireless link optimization, session migration, user management

The **MOBITHIN Project Consortium** groups the following Organizations:

Interdisciplinary Institute for BroadBand Technology vzw	IBBT vzw	BE
Deutsche Telekom Laboratories	DTAG	DE
Prologue Software	Prologue	FR
Interuniversitair Micro-Electronica Centrum vzw	IMEC vzw	BE
NEC Technologies (UK) Ltd	NTUK	UK
Institut Télécom – Télécom Sud Paris	IT	FR
JCP-Consult SAS	JCP	FR

EDITORIAL Change Management

Version	Date	Editor/Author	Comments
Preliminary		B. Lécroart	Template created
	May 19, 2010	R. Torrea	First draft
	June 21, 2010	R. Torrea	Updated revision
Final version	June 30, 2010	Bart Dhoedt	Released Version
V1.1	July 1 st , 2010	Bart Dhoedt	Figures references

Contents

1.	<i>Executive summary</i> _____	5
2.	<i>Introduction</i> _____	6
2.1	Project Context _____	6
2.2	Relation with D5.5 and D5.7 _____	6
3.	<i>Scope of the deliverable</i> _____	7
4.	<i>Experimental reports on each demo</i> _____	7
4.1	Link Optimization NS2 Emulation _____	7
4.1.1	Software/hardware architecture _____	7
4.1.2	Description of experiments performed _____	8
4.1.3	Results and interpretation _____	10
4.1.4	Conclusions – Requirements met _____	21
4.2	Link Optimization Hardware Emulation (XMSF) _____	21
4.2.1	Software/hardware architecture _____	21
4.2.2	Description of experiments performed _____	22
4.2.3	Results and interpretation _____	22
4.2.4	Conclusions – Requirements met _____	27
5.	<i>Overall conclusions</i> _____	28
5.1	Results Summary _____	28
5.2	Requirement coverage _____	28
5.3	Final remarks _____	30
6.	<i>References</i> _____	30

Figures Index

Figure 1 - High-level MobiThin project structure.	6
Figure 2 - Physical set-up for the NS2 simulator.	8
Figure 3 - Overall XL approach.	9
Figure 4 - Immediate (a) and deferred (b) server responses to display update requests.	11
Figure 5 - Cross-layer approach.	11
Figure 6 - Energy consumption of the wireless energy platform.	13
Figure 7 - Bandwidth measured in downstream direction.	15
Figure 8 - Downstream data per display update in downstream direction.	15
Figure 9 - Number of display updates at the end the experiment.	16
Figure 10 - Bandwidth measurement in upstream direction.	16
Figure 11 - Power-goodput curves for different pathloss values for UL traffic.	17
Figure 12 - Video application during a 50 s simulation.	18
Figure 13 - Average power consumption between XL and SoA solutions for the UL.	18
Figure 14 - Power-goodput curves for different pathloss values for DL traffic.	19
Figure 15 - Video application during a 50 s simulation.	20
Figure 16 - Average power consumption between XL and SoA solutions for the DL.	20
Figure 17 - Timing and state information per component of the SDR platform.	22
Figure 18 - Power breakdown of the SDR platform using SoA.	23
Figure 19 - Power breakdown of the SDR platform using XL.	23
Figure 20 - Power breakdown of the SDR platform using SoA.	24
Figure 21 - Power breakdown of the SDR platform using XL.	24
Figure 22 - Power breakdown of the SDR platform using SoA.	25
Figure 23 - Power breakdown of the SDR platform using XL.	25

Tables index

Table 1. Mapping of D5.5 PoC's to reporting in D5.6/7.	5
Table 2: Average idle time.	13
Table 3: Relative gain in total power consumption.	14
Table 4: Power consumption and energy gains.	27

1. EXECUTIVE SUMMARY

This report, together with its accompanying deliverable 5.7, summarizes the validation results obtained in project phase II of MobiThin. The results addressed are obtained through emulations, i.e. for validating the work performed, the system under study has been emulated on other hardware. The results presented relate to evaluation work carried out using NS-2 as emulation framework, and a hardware emulation framework (XMSF) incorporating more hardware details on the transceivers used (allowing a more accurate assessment of energy consumption).

Three mechanisms have been investigated using NS-2 as underlying emulation platform. First, a cross-layer approach was investigated, putting the radio in sleep mode whenever possible, based on application (i.e. thin client protocol) information. Using this approach, energy gains up to 49% were observed for experiments on WiFi networks (the same techniques can be applied on other types of networks). A second track consists of using UDP as transport layer protocol for downstream traffic in a push protocol mode. It is shown how bandwidth reductions up to 75% in the upstream and 29% in the upstream can be achieved. Furthermore, the work opens perspective to an improved cross-layer algorithm as the push-protocol allows a better prediction.. A third topic investigates cross-layer approaches for LTE-networks, adopting a cross-layer optimization of MAC and PHY layers, that based on bandwidth settings and channel information (path loss) adapts the transmission configuration to achieve a higher energy efficiency. Results indicate considerable energy gains when compared to current state-of-the-art solutions (maximum goodput or lazy scheduling). For a video scenario (2 Mbit/s traffic) gains amount to 51% and 23% for uplink and downlink traffic respectively.

To model more accurately the wireless interface, a cross layer MAC simulation framework (XMSF) was developed and its implementation on a software defined radio platform was evaluated. This framework makes abstraction of the higher level protocols, and allows to model energy consumption under constant traffic load. Emulations were performed for varying bandwidths and a certain channel quality, indicating that the energy consumption can be decreased with respect to state of the art with higher data rates. For a video scenario (2 Mbit/s traffic) gains of 35% were achieved assuming a MAC-PHY cross layer optimization with a constant data rate. Also a comparison between the gains obtained with XMSF and NS2 for the same scenarios is presented. Results show a high correlation between both NS2 and XMSF emulators.

The emulations reported here have been constructed in accordance with the proof-of-concept design document, The results obtained allow to conclude that the approach taken wireless link optimization indeed offers important benefits in terms of power efficiency, without jeopardizing the interactivity as perceived by the end-user of the system.

2. INTRODUCTION

Common text for D5.6 and D5.7

2.1 PROJECT CONTEXT

The MobiThin project is organized in two phases, allowing an incremental approach to realize its final goals. The deliverables D5.6 and D5.7 conclude the validation work of the second project phase, thereby essentially providing final evidence and demonstration of the projects approach and concept.

Work package 5 of the MobiThin project is to design, develop and evaluate proof-of-concept demonstrators in order to validate the approach taken in the project (e.g. on the architectural or algorithmic level). After the design (D5.5, M21) of these PoC's, several PoC's were indeed realized, based on various components (either developed in the first or second project phase) realized in WP3 and WP4 (Figure 1). The findings of the validation work are reported in D5.6 and D5.7 and, in addition to show-casing and dissemination, results will mainly be used further integration, moving the solution closer to commercial valorization. While D5.6 focuses on emulation demonstrators (major part of the hardware is emulated in software), D5.7 reports on lab-trials (primarily using the intended hardware for the project).

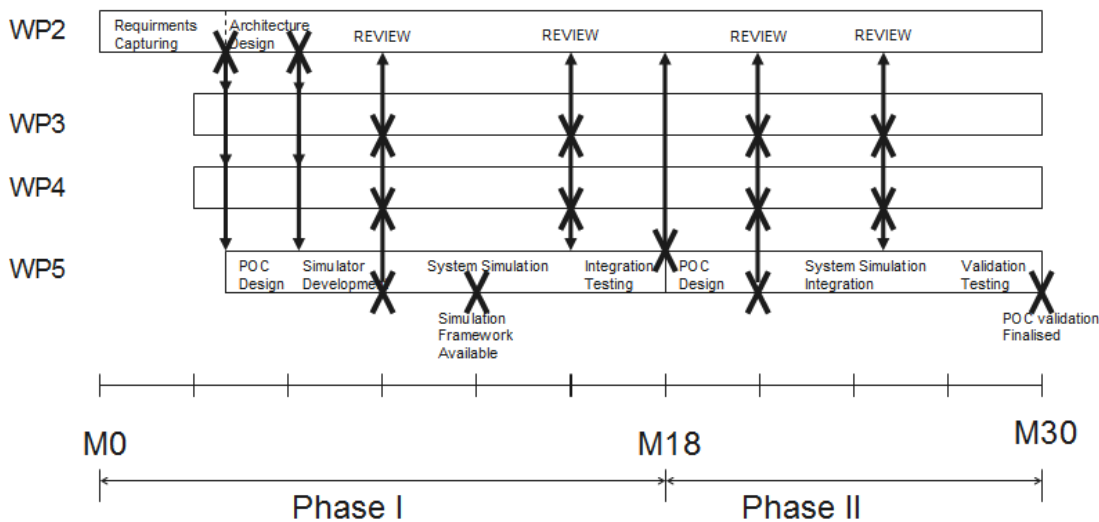


Figure 1 - High-level MobiThin project structure.

2.2 RELATION WITH D5.5 AND D5.7

Common text for D5.6 and D5.7

The following table identifies the demonstrations performed during the second phase of the Mobithin project.

PoC number	Title	Report (section)
1	Link Optimization NS2 Emulation	D5.6 (4.1)
2	Link Optimization Hardware Emulation (XMSF)	D5.6 (4.2)
3	Integrated Demonstration	D5.7 (4.1)
4	Multimedia Thin Client based on SIP	D5.7 (4.2)
5	Effects of Constraints on Image Transmission	D5.7 (4.3)
6	Remote devices / Peripherals	D5.7 (4.4)
7 (*)	Local Mobile Device Peripherals	D5.7 (4.5)

Table 1. Mapping of D5.5 PoC's to reporting in D5.6/7.

(*)Note : This demonstration was not identified in D5.1 POC document. This demonstration has been added in order to increase the usability of Thin Client on mobile device. During the first review in March 2009, the reviewers emphasized the importance of the usability of the Thin Client for the end users. The usability point was

also noticed by industrial members of the consortium who points the difficulty to use Thin Client over limited screen device such as mobile phone. In order to improve usability of thin client systems and to extend the thin client market, a demonstration based on the connection of external user-interface peripheral (Keyboard, screen, mouse) was added. The demonstration has been named “Local Mobile Device Peripherals”.

3. SCOPE OF THE DELIVERABLE

As outlined above, this deliverable focuses on reporting experimental validation results on emulation oriented proof-of-concepts. The emulated PoC’s are:

1. Link Optimization Network Simulator 2 (NS2) Emulation
2. Link Optimization Hardware Emulation (XMSF)

The reports of both PoCs (detailed in section 4 of this document) have the same base structure:

- (i) a recap on the hardware/software architecture,
- (ii) presentation of the experimental set-up,
- (iii) presentation of obtained results and interpretation,
- (iv) conclusions (where appropriate, feedback to the requirements outlined in D2.1 is given).

4. EXPERIMENTAL REPORTS ON EACH DEMO

4.1 LINK OPTIMIZATION NS2 EMULATION

4.1.1 Software/hardware architecture

Since the details of this emulation were explained in D5.5, we will present a brief recap of the simulator set-up in this document only. The demonstrator consists of 3 desktop PCs as indicated in Figure 2, implementing the VNC protocol through an emulated wireless network. The PC on the right of Figure 2 runs the VNC server while the PC on the left runs the VNC client. They are both connected through virtual tap interfaces to the middle PC running the NS2 IMEC simulator which emulates the MAC and PHY layers of the terminal and the access point as well as the wireless channel. This simulator also provides the radio-link level energy models for both state-of-the-art and cross-layer solutions developed in the context of WP3.1. Further enhancements include an overall cross-layer integration between the wireless link and application layers.

The goal of this emulator is to characterize a thin client system over WLAN, and 802.11e, and 3GPP Long Term Evolution (LTE) networks in terms of energy consumption and quality of service (QoS). As a second goal, we validated the MAC-PHY and the Overall Cross-Layer (XL) energy gains with respect to a state-of-the-art radio link control.

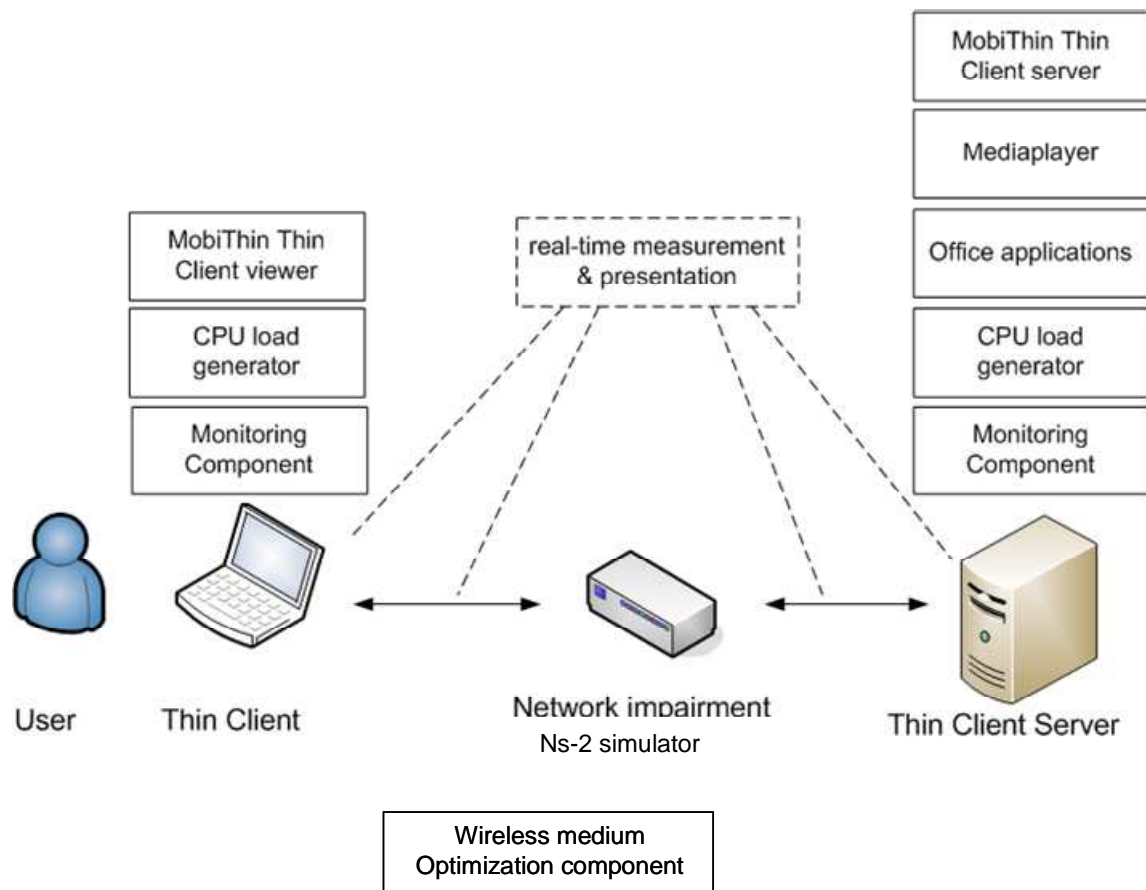


Figure 2 - Physical set-up for the NS2 simulator.

4.1.2 Description of experiments performed

The tests, including the MAC-PHY XL solution, were reported in D5.3 for a WLAN 802.11 scenario. There, we showed that the MAC-PHY XL solution outperforms state-of-the-art especially in high-traffic and congested network scenarios in terms of energy consumption (24% with high traffic 74% gain with high congestion).

In this document, we report the tests including an overall XL approach. This approach extends the functionality of the MAC-PHY XL approach from [1] in order to benefit from sleeping periods at the wireless transceiver when the application layers allow this. Furthermore we extended the MAC-PHY XL approach for an LTE network.

4.1.2.1 Overall XL

The basic idea of the overall XL approach is depicted in Figure 3. It consists of having a closer interaction between application layers and the wireless link in order to allow larger passive sleeping periods.

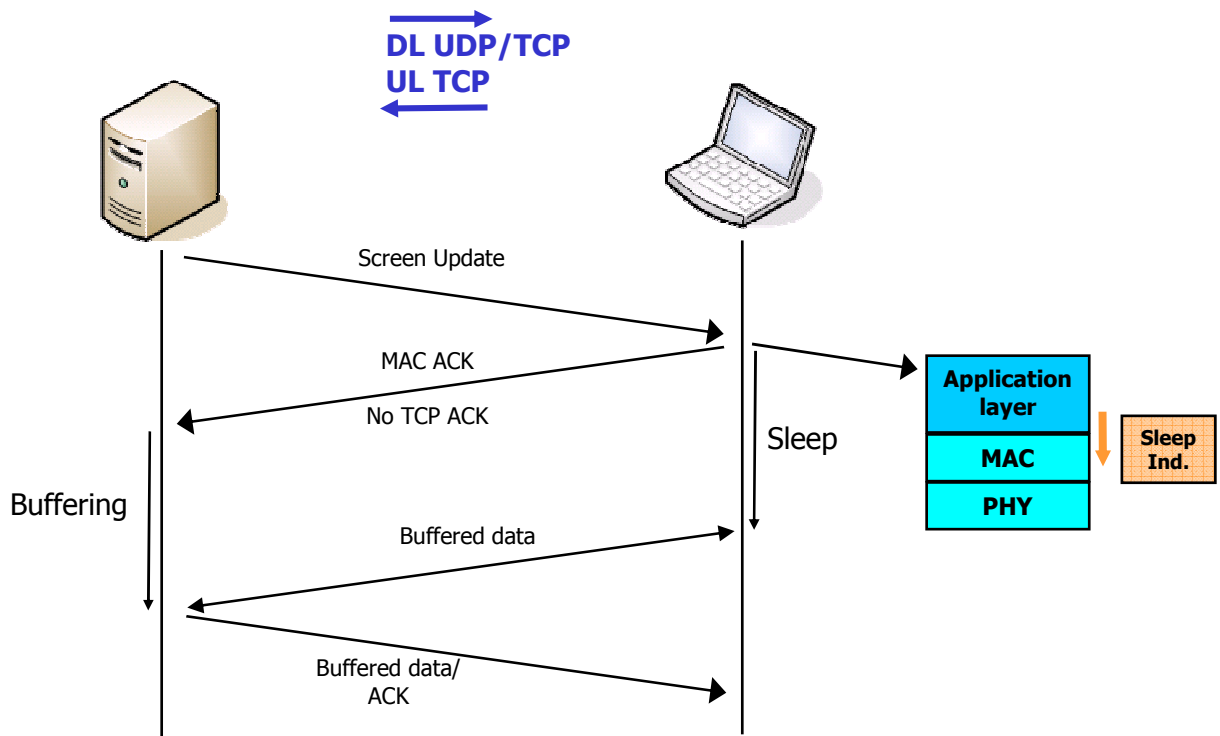


Figure 3 - Overall XL approach

During the “active” period, the terminal is constantly receiving screen updates from the server. After the last packet has arrived, no more updates will be sent by the server for a certain period. In the downstream, VNC is currently using TCP to transmit display updates. This has the advantage of reliable data delivery, but results in overhead in both downstream and upstream direction that can be reduced by using UDP in the downstream. Therefore, the impact of using UDP in the downstream direction was investigated as well.

After the last of the screen updates, the application layer from the terminal sends an indication to the wireless link to go into sleep mode for a certain fixed and well defined period. This mode turns off most of the mobile platform components for energy saving. During this period, the server and the client are buffering data that will be transmitted after the sleeping period.

4.1.2.2 MAC-PHY Cross-Layer for LTE

The MAC-PHY XL approach for a WLAN 802.11a transceiver was explained in detail in D5.2, and it is based on the approach from [1]. Here we extend this approach into the context of a 3GPP Long Term Evolution (LTE) scenario.

Similarly to the WLAN case, MAC and PHY layers are jointly controlled in order to save energy. For this purpose, we consider one external monitor, namely the channel path loss, and one requirement, the data rate. The first one can be computed from the received signal of known pilot symbols and the second one from the number of time-frequency resources or physical resource blocks (PRBs) assigned to the user. These parameters are used to select a transmission configuration that is energy-optimized for the mobile terminal while providing the desired data rate. The transmission configuration consists in a combination of modulation scheme, code-rate, and power amplifier (PA) settings that are configurable at the terminal. Specifically for LTE, 15 possible combinations of modulation schemes and code-rates are supported [2].

Contrary to WLAN, in LTE the base station or evolved Node B (eNB) takes the decision of the transmission configuration based on parameters such as channel information and user equipment (UE) requirements to define the final configuration. Therefore a purely-terminal XL approach can only partially benefit from the achievable energy gains. A similar approach at the eNB could fully exploit the XL solution.

Baseband Energy Scalability

In our approach, we aimed to reduce the total energy consumption at the UE based on its scalability in transmission parameters, detailed platform power and performance models (packet error rate at MAC level) from the baseband (BB) chain, the analog front-end (FE) and the digital platform.

Based on component power values from the IMEC software-defined radio (SDR) platform described in [3], we can build a platform-level model that specifies the power consumption of the whole platform for different radio

power states (transmit, receive, idle, sleep) that are differentiated by the activation of certain platform components. However, the energy consumption of the whole platform depends not only on the active components in a given radio power state, but also on the time they remain in that state. The fraction of time a certain component is active is called duty cycle. The duty cycling factors are derived from the selected sub-frame pattern defined by LTE [2]. The final energy values result from the power consumption and the duty cycling of each component.

Power Amplifier Scalability

Among the platform components, the PA is by far the largest power contributor of the transmitter front-end (FE) scalability, while the other components generally have a fixed consumption for a given bandwidth. In our implementation, the PA is modeled in terms of consumed power, output power, and distortion level, for any possible configuration. It has a large role in uplink scenarios, while in downlink scenarios most components have a fixed and lower power consumption. In DL the gains come mainly from optimized modulation and code-rate.

Globally-Optimized Solution

Taking into consideration all the previous models, we arrive to a global energy optimization problem. This optimization involves searching into all the configuration combinations for those that offer the lowest power consumption for a given path loss and data rate. In this way we can build up a set of optimal configuration points that the UE can use while maintaining the data rate requirements.

Practically, this optimization is done in two steps. All the optimal combinations are pre-computed at design-time. Then these values are stored into a database that can be accessed by the terminal at run-time. Then, depending on the required goodput and the measured path loss, the UE informs of the transmission configuration selected to the eNB through the channel quality indicator (CQI), so that in the next transmission opportunity, the eNB can use the reported energy-optimized configuration.

4.1.3 Results and interpretation

4.1.3.1 Overall XL - TCP

The share of the total energy consumption that is required for sending and receiving thin client protocol data is limited. The wireless platform of thin clients is mostly in the idle state due to the bursty character of thin client protocol traffic [4]. Consequently, for the thin client case, more important energy savings can be expected from approaches that put the wireless platform in the energy-conserving sleep mode during intervals without communication, rather than from algorithms that optimize encoding and transmission parameters. In a thin client approach, we can assume that all applications are offloaded to an application server in the network. Only a single network connection is required to communicate user events and display updates over a thin client protocol. We present a cross-layer algorithm that analyzes the thin client protocol information and puts the medium access control layer (MAC) in sleep mode when no traffic from the server is expected.

4.1.3.1.1 Thin client protocol operation

In general thin client protocols can operate either in push or pull mode. In push protocols, such as Microsoft Remote Desktop Protocol, the server autonomously determines the rate of display updates that is sent to the client. In pull protocols, such as Virtual Network Computing (VNC), the client explicitly requests a new display update from the server. Although push protocols can guarantee a fixed transmission rate of display updates under all circumstances, they might overwhelm the client. In pull protocols, the rate is automatically adjusted to the network status and to the client's processing capability, but the frequent requests lead to an increasing upstream bandwidth consumption. Here, we will focus on VNC, which is a pull protocol.

VNC was chosen because of several reasons. First, VNC is (besides FreeNX) one of the few available open source thin client architectures. Furthermore it is widely used. Inside the MobiThin scope, VNC was therefore withheld for implementation of thin client protocol optimization prototypes. By implementing all optimizations in the same architecture, compatibility remains ensured and a possible future integration is facilitated.

Figure 4 depicts the two ways how a VNC server can respond when it receives a request from the client. If the display has changed since the previous update was sent, the request will be answered immediately. If the display is unmodified since the previous update, a *deferred update* mechanism is activated. In this case, a timer is started when the display is updated for the first time after receiving the request. This timer goes off after T_{def} , and then a display update is sent. This update includes the display changes that occurred during the timer interval.

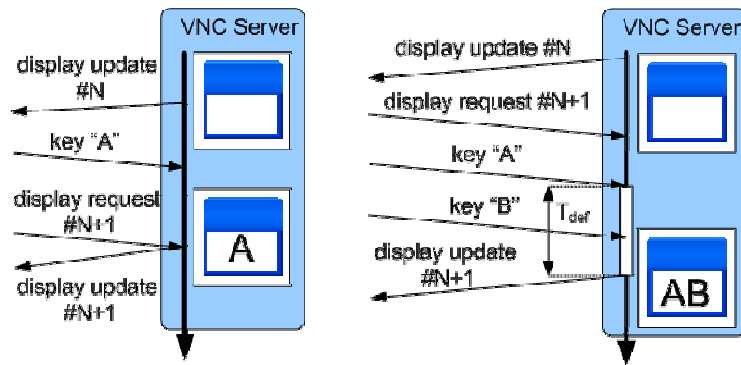


Figure 4 - Immediate (a) and deferred (b) server responses to display update requests.

4.1.3.1.2 Cross-layer algorithm

From the viewpoint of the client, the update following a display request must not be expected earlier than one network roundtrip time (RTT), as can be seen on Figure 5. During this interval, the wireless platform on the client can be put in sleep mode without risking to miss any incoming data. The platform needs to wake up only when an update is expected, or when user events must be transmitted. In the following discussion, we will assume that only with an interval of T_{ue} , user input is sent upstream. This can be achieved by buffering user events at the application layer during T_{ue} .

The size of the interval to put the MAC layer in sleep mode can only be calculated based on application level information and hence a cross-layer mechanism is required, as presented in Figure 5. Every time a user event or a request is transmitted, the application layer determines how long the MAC layer can be put in sleep mode, based on the forecasted reception of the next update. This forecast algorithm, presented in Algorithm 1, is activated each time the client transmits or receives data.

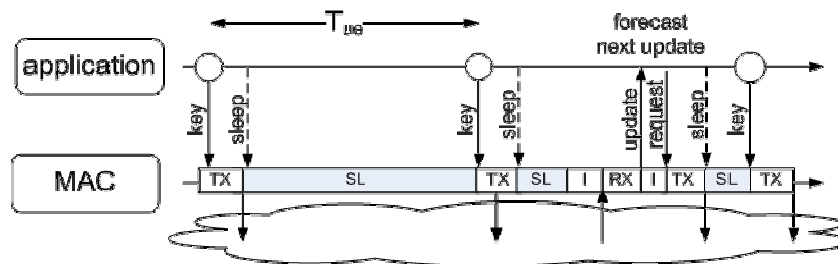


Figure 5 - Cross-layer approach

The application forecasts when the next update will be received. The wireless platform is put in sleep mode during intervals without communication.

Algorithm 1: Algorithm to forecast the next update, invoked each time data is sent or received.

```

Require:  $RTT$  and  $T_{def}$  are initialized
 $CT \leftarrow \text{getCurrentTime}()$ 
if  $\text{data.type} == \text{request}$  then
  if  $\text{displayedUserEvents} < \text{seqUserEvent}$  then
     $\text{timeNextUpdate} \leftarrow CT + RTT$ 
     $\text{nextUpdateScheduled} = \text{true}$ 
  else
     $\text{deferredUpdate} = \text{true}$ 
     $\text{nextUpdateScheduled} = \text{false}$ 
  end if
else if  $\text{data.type} == \text{input}$  then
   $\text{seqUserEvent} ++$ 
  if  $!\text{nextUpdateScheduled}$  and  $\text{deferredUpdate}$  then
     $\text{timeNextUpdate} \leftarrow CT + RTT + T_{def}$ 
     $\text{nextUpdateScheduled} = \text{true}$ 
  end if
else {update received}
   $\text{nextUpdateScheduled} = \text{false}$ 
   $\text{displayedUserEvents} = \text{data.update.displayedEvents}$ 
end if

```

When a request is transmitted, the algorithm checks if there are any transmitted user events that are not yet displayed. If this is the case, the request will be immediately answered by the server, and the update can be expected in 1 RTT. Otherwise, the deferred update mechanism of Figure 3(b) will be activated when the next user event is received by the server. The update is forecasted to be received $RTT + T_{def}$ after transmitting the next user event.

4.1.3.1.3 Simulation results and discussion

We have implemented the cross-layer algorithm in TightVNC 1.3.10 for Unix and validated on a test set-up composed of a client and server machine that are interconnected by an impairment node and a 802.11 channel simulator implemented in ns-2. The parameter T_{ue} was varied between 20 and 70 ms, the network roundtrip time (RTT) between 30 and 100 ms and the wireless channel path loss between 60 and 90 dB. Each experiment was repeated 10 times to reduce the influence of random variations. The ns-2 simulator was equipped with energy models to measure the power consumption of the wireless platform. In the experiments, keystrokes were generated during 2 minutes, with a period of T_{ue} , and transmitted to OpenOffice Writer, a text editor.

Figure 6 provides insight in the energy gains that can be achieved when activating the cross-layer algorithm (XL). The total energy consumption is shown for both the state of the art (SoA) and the cross-layer algorithm presented in previous section. The SoA approach refers to maximum good-put link adaptation algorithm that uses the fastest possible transmission scheme for a certain path loss (described in D5.3). For XL, also the energy consumption in each of the 4 states of the wireless platform is shown. These 4 states include transmit, receive, idle, and sleep. During transmit (TX) and receive (RX) states, as their name says, the terminal is either transmitting or receiving information. During the idle state, the terminal is only detecting incoming frames by activating the front-end (FE) but not the digital base band (BB) to shift quickly to another state. It can be optimized through a sleep state that turns off most of the platform components when we are sure that no data is received.

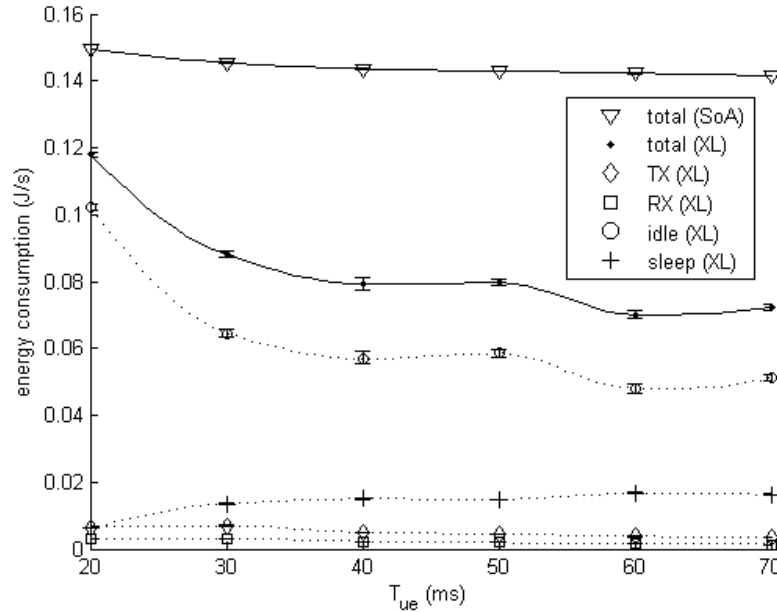


Figure 6 - Energy consumption of the wireless energy platform with RTT = 30 ms and 60 dB pathloss. Higher values of T_{ue} result in longer periods without communication to put the wireless platform in sleep mode.

Increasing values of T_{ue} result in less data to be sent, because user data is generated less frequently. This also means that the display will change less frequently. Hence, the amount of display updates decreases accordingly, less data is received by the client and less energy is consumed by the wireless platform in the receive state. Nevertheless, in the case of SoA, higher values of T_{ue} are not reflected in a significant decrease of the total power consumption. This confirms the observation that an important fraction of the wireless platform energy consumption is due to the time spent in the idle mode, as explained in section 4.1.3.1. By contrast, the XL algorithm is able to exploit these longer intervals without communication, and power savings are achieved of 21 % for T_{ue} = 20 ms and up to 49 % for T_{ue} = 70 ms.

More results, for other values of the path loss and the network RTT, are presented in Table 1. For T_{ue} = 40 ms, the relative gain in total power consumption varies between 35 % and 49 %.

Table 1: Average idle time per second for both state-of-the-art (SoA) and the cross-layer algorithm (XL), for T_{ue} = 40 ms. The values are in ms.

		pathloss [dB]							
		60		70		80		90	
RTT [ms]	30	SoA	XI.	SoA	XI.	SoA	XI.	SoA	XI.
	50	988	414	986	432	987	430	983	483
	70	987	404	987	418	987	415	983	423
	100	988	533	987	536	987	537	984	547
	100	990	378	990	394	989	389	987	421

For SoA, it can be seen that the platform is more than 98 % of the time in idle state, due to the low rate of thin client protocol traffic. XL succeeds in reducing this idle time to approximately 40 %. We have chosen to implement the algorithm in a conservative way, in order to preserve the user responsiveness. If the time error between the predicted and the actual moment of receiving the next update was higher than 10 ms, the XL algorithm was disabled until two more display updates are received. If the platform is in sleep mode while a display update is being transmitted, this will be observed as packet loss at the TCP layer and as increased latency at the higher layer. Consequently, the TCP algorithm will apply the congestion control algorithm and reduce its sending rate, resulting in decreased user responsiveness for the next updates as well. By disabling the XL algorithm for the next two updates, we allow TCP to recover from the path loss and increase its sending rate to the same level as before.

Because the results on the gain in total energy consumption are hardware dependent, we have added Table 2, presenting the average idle time per second of the wireless platform.

Table 2: Relative gain in total power consumption of XL vs SoA for $T_{ue} = 40$ ms.

		pathloss [dB]			
		60	70	80	90
RTT [ms]	30	44.88	43.79	43.72	38.90
	50	45.40	44.81	44.94	43.86
	70	36.25	36.65	36.49	35.39
	100	48.42	47.83	48.04	45.22

4.1.3.1.4 Conclusion

When running an office application over a thin client protocol, the wireless platform is about 98 % of the time in the idle mode. In this chapter, a cross-layer architecture was presented that uses application layer information to instruct the MAC layer to go in sleep mode when no data needs to be transmitted or received. More specifically, the reception time of display updates from the server is predicted. The results show that power savings up to 49 % can be achieved.

4.1.3.2 Overall XL – UDP

Originally, VNC works using a pull protocol, with the client specifically requesting a new server update. As explained in D3.4, this leads to unnecessary upstream bandwidth consumption. Therefore, VNC has been modified to operate as push protocol, with the server pushing display updates to the client. The server controls the rate of display updates and this rate is a trade-off between Quality of Service (QoS) and energy consumption of thin client. Whereas higher display update rate will improve user perceived responsiveness, it will also result in more traffic to be received in downstream direction and hence more energy consumption.

A second modification to the original VNC operation is the use of UDP in downstream direction, instead of TCP on the original VNC. The rationale behind this is the fact that we will be pushing display updates to client at regular interval. Hence, lost display updates should not be retransmitted (as would be the case with TCP). Secondly, the switch to UDP eliminates the TCP acknowledgements sent in upstream. Third, because the client can easily predict the arrival of the next update, the WNIC of the device can be put in sleep mode for longer intervals, as compared to pull-case reported in section 4.1.3.1. Consequently, important energy gains are expected. The work reported in this deliverable focuses on bandwidth gains that can be achieved by using VNC as a push protocol over UDP.

4.1.3.2.1 Bandwidth measurements

Bandwidth measurements are performed for VNC over UDP and compared to the standard VNC solution running over TCP. Data is captured at both server and client side through Wireshark for downstream and upstream traffic respectively. Measurements are performed for VNC over UDP for different display update rates e.g. 30, 60, 90, 120 and 150 ms, and the results are compared with VNC running over TCP to observe the bandwidth gains.

Measurements are performed for a scenario in which the user opens Firefox browser and reads a BBC news article, checks the Mobithin website and replies to an email using Gmail.

4.1.3.2.1.1 Downstream

In Figure 7, the downstream bandwidth consumption is presented for both TCP and UDP. The total downstream bandwidth is composed of TCP acknowledgements and VNC traffic. For the latter, both the VNC/UDP/IP headers and the actual payload (display updates) are shown. It is observed that by using UDP as a transport protocol and by decreasing display update rate, a bandwidth reduction can be achieved up to 29 % depending on the refresh rate at server side. This bandwidth reduction will also decrease the energy consumption of thin client as wireless card consumes lot of energy while sending and receiving data. For display update interval of 30 ms, the downstream bandwidth consumption exceeds the TCP-pull bandwidth consumption. It appears that, for applied use case, the average interval between subsequent updates is about 50 ms for TCP-pull. In other cases, with different values of network roundtrip time or used application, this might differ.

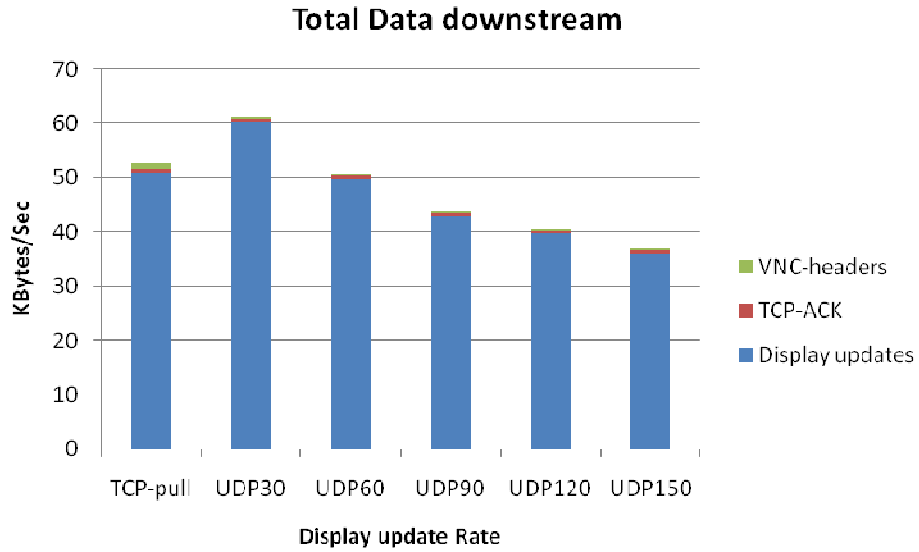


Figure 7 - Bandwidth measured in downstream direction. Most bandwidth is consumed by the display updates. “VNC-headers” is the bandwidth used for IP/UDP/VNC headers to encapsulate this data. TCP-ACK are the acknowledgements for the received user events.

It is observed that the total downstream bandwidth only increases slightly with the display update rate. At first sight, one would expect a linear decrease, i.e. doubling the interval between subsequent display updates would half the downstream bandwidth. Because the total downstream bandwidth is a function of both the size of an individual update and the number of display updates that is actually sent, we have investigated the effect on both parameters of changing the display update rate. If we decrease the display update rate, more changes will occur at the server side, more areas of the screen need to be updated and the display updates are larger in size. This effect is clearly seen in Figure 8. This graph shows the average size of a display update, which increases for higher values of the interval between subsequent display updates.

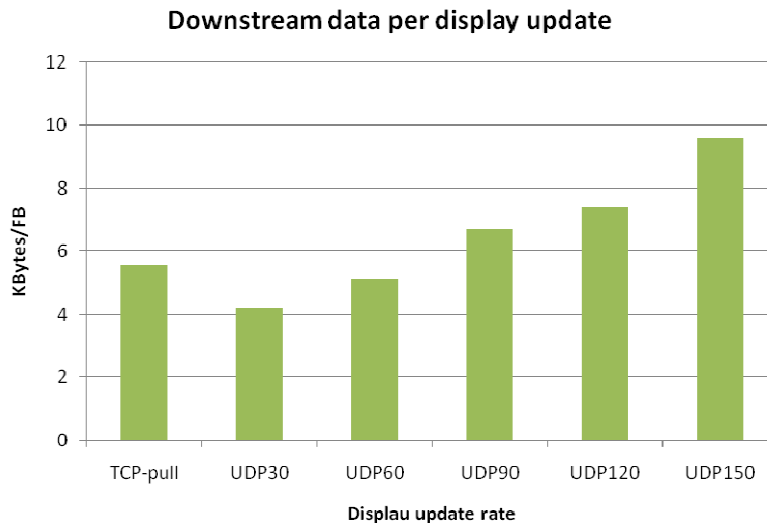


Figure 8 - Downstream data per display update in downstream direction

The configured display update rate only determines the period at which the server checks whether the client display update needs to be updated or not. If no display changes occurred, nothing will be sent. For high display update rates, it can be expected that sometimes the display has not changed. This is confirmed by Figure 8, indicating the number of display updates that is actually sent.

If we compare Figure 8 and Figure 9, we can conclude that with a decrease in the display update rate, the number of display updates decreases and downstream data per display update increases as there will be more data to be

sent in downstream direction. On Figure 8, it is observed that at refresh rate of 30 ms, a large number of times there is no change on server side. So, no update is sent.

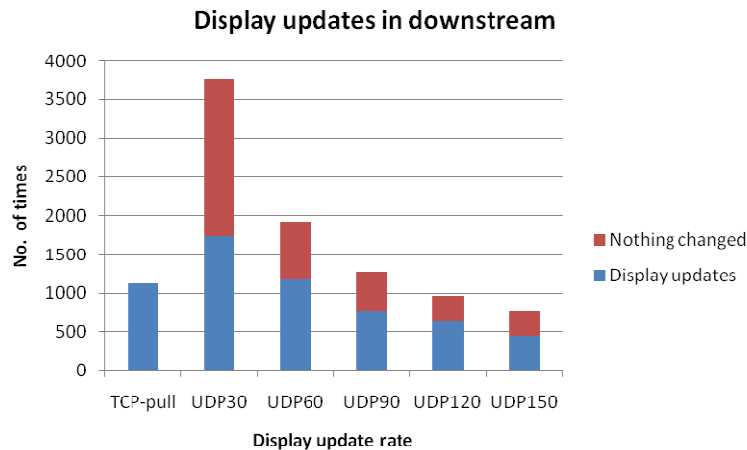


Figure 9 - Number of display updates at the end the experiment.

4.1.3.2.1.2 Upstream

Because loss of user events cannot be tolerated, it was decided to keep TCP as transport protocol in the upstream direction. However, using UDP protocol in downstream direction has major impact on bandwidth reduction in upstream direction as well. By using UDP in the downstream direction, no upstream TCP ACKs need to be sent. Second, no more display update requests need to be sent. As indicated in Figure 10, these two factors represent a significant fraction of the upstream traffic in the TCP-pull case. We can gain up to overall 75% bandwidth reduction by eliminating 58% TCP acknowledgements and display update requests help to reduce more bandwidth up to 17%. Display update rate has no effect on the upstream direction as this rate is related to updates coming from server side. User events are generated at the same rate for both TCP-pull and UDP-push version of VNC.

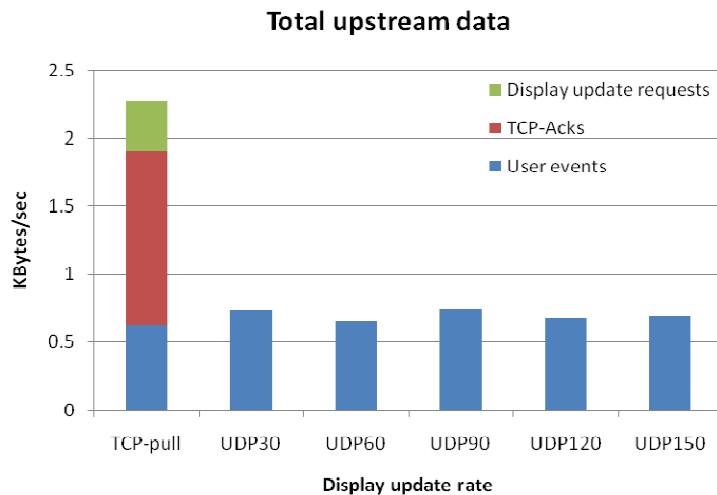


Figure 10 - Bandwidth measurement in upstream direction

4.1.3.2.2 Conclusion

We can conclude that by using VNC over UDP, the bandwidth reduction can be achieved in both upstream and downstream direction. Major gains can be achieved up to 75% in upstream direction and 29 % in downstream direction. Reduction in bandwidth will also reduce the energy consumption of wireless card of thin clients as it consumes lot of energy while sending and receiving data. We can gain up to 70% of energy consumption by running this UDP-push VNC on Overall Cross layer algorithm.

4.1.3.3 MAC-PHY Cross-Layer for LTE

For validation we consider the following scenario: LTE uplink (UL) and downlink (DL) transmission with a 20 MHz bandwidth, using the software defined radio platform (SDR) power consumption figures [3]. For the UL scenario we selected the time division duplex (TDD) sub-frame sequence from the standard with the highest UL/DL sub-frame ratio and for the DL the sequence with the lowest ratio. We then derive for different values of path loss (60-105 dB with steps of 5 dB), a trade-off curve between consumed power and goodput based on the design flow as described in Section 4.1.2.2 (Figure 11). For these curves all the possible configurations are simulated at design-time. The optimal ones are kept and stored in a look-up table, which the terminal can access at run-time. Our configuration space consists of a set of modulation schemes (QPSK, 16-QAM, and 64-QAM), code-rates (1/3 1/2 2/3 3/4 7/8 1) and PA settings combination for the single-antenna case (SISO) all supported by the LTE standard.

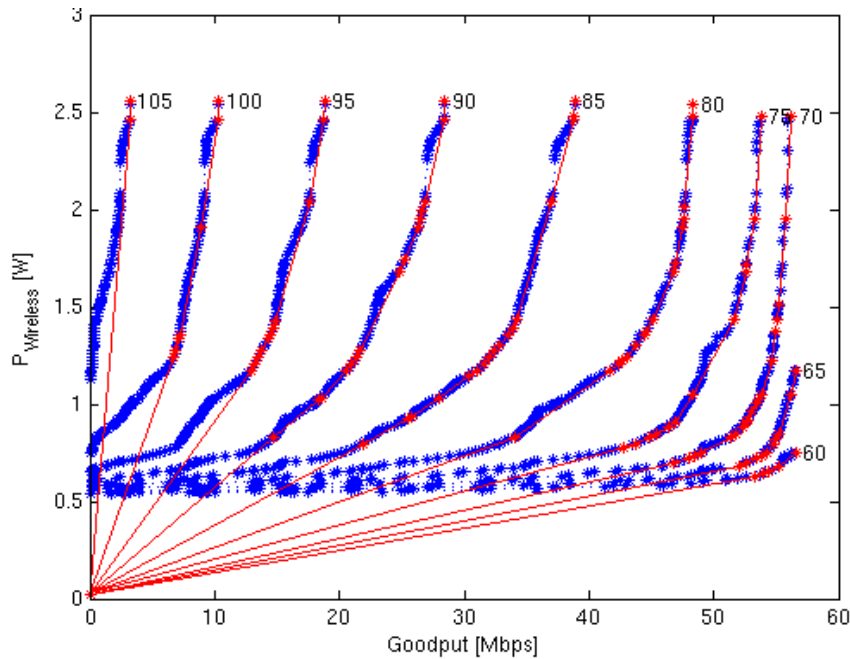


Figure 11 - Power-goodput curves for different pathloss values for UL traffic

On each curve, the red markers touching the lines denote optimal configuration points. The linear interpolation represents the duty-cycling mode. This means that when the required goodput is lower than the goodput offered by a certain configuration, the power consumption will be linearly scaled through the red line. When the path loss is reduced, the system can exploit higher-order constellations and, by doing so, reduce its duty-cycling, hence the average active power.

To validate our approach, we simulated a UL transmission of an average of 2 Mbps, with an average path loss of 80 dB (100m distance from the eNB) under a time-varying wireless scenario (Figure 12). We compare the total energy of the MAC-PHY XL solution to two reference state-of-the-art (SoA) policies: maximum goodput and lazy scheduling. The first one selects the modulation, code-rate, and PA settings that achieve the highest goodput for a given path loss. The second one selects the lowest transmission speed fitting the required data rate. We assume that the eNB is always using the modulation and code rate that optimizes the UE energy consumption, if this is not the case, the energy gains from the XL approach will be lower.

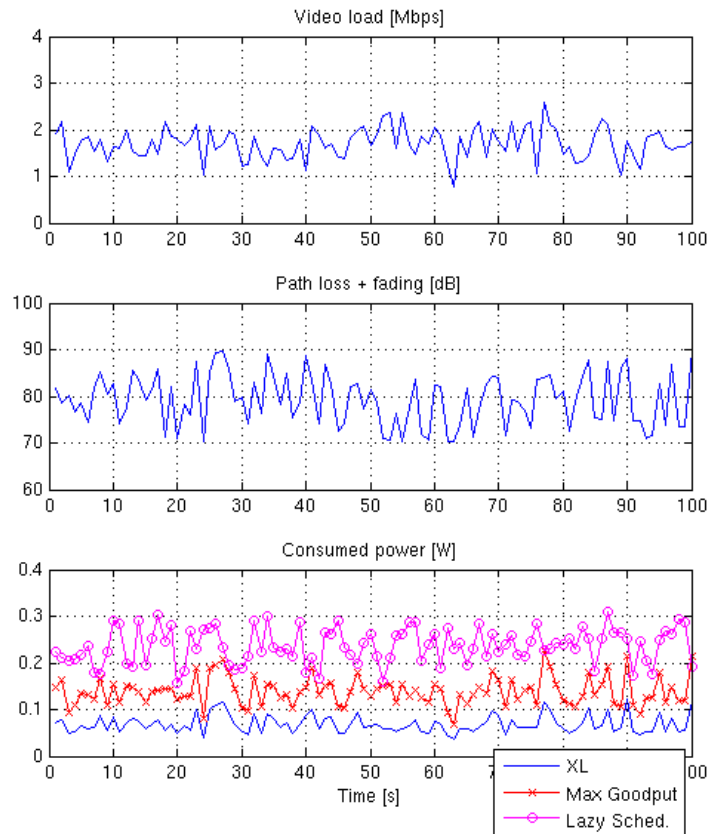


Figure 12 - Video application during a 50 s simulation.

The first two sub-plots on Figure 13 represent the data rate requirements and the instantaneous path loss. Each of these variables shows large fluctuations typical of a wireless multimedia scenario. The third subplot shows the corresponding required power for the reference cases compared to the XL solution. The gain achieved leads to an average power reduction of 50.7% compared to maximum goodput and 70.89% compared to lazy scheduling with an average power of 69 mW. The average power breakdown per platform component is compared for the three different policies in Figure 13. The components described in the figure include the power amplifier (PA), the transmit front end (TxFE), the receiver front end (RxFE), the local oscillator (LO), the digital base band (BB), and the sleep or leakage energy from all the whole platform.

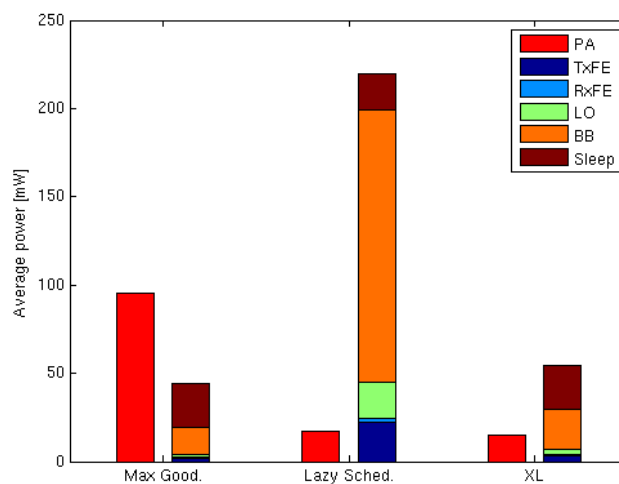


Figure 13 - Average power consumption between XL and SoA solutions for the UL

Compared to the XL solution, the maximum goodput strategy is penalized by the large PA power consumption. On the other hand, lazy scheduling is penalized by the large duty cycling of the other components. This duality is

central to the success of cross-layer approaches: without an integrated optimal decision, the system wastes energy either in the PA or in excessive duty-cycling.

Comparing our approach with other state-of-the-art SDR platform solutions like WCDMA [5][6] used in UMTS standard we can obtain a power reduction from 240mW to 69mW with an energy efficiency of 30 nJ/bit instead of 120 nJ/bit for the same data rate [7].

For the DL case, the gains tend to be smaller since the PA has limited contribution (mainly acknowledgements) in this context and the energy savings achieved from the baseband can be marginal. The power-goodput curve for the DL is shown in Figure 14.

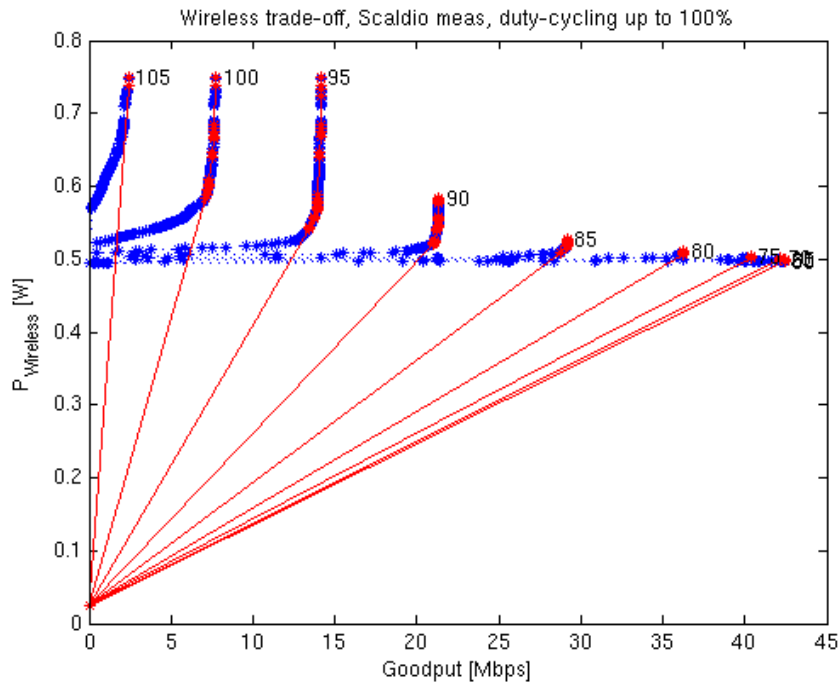


Figure 14 - Power-goodput curves for different pathloss values for DL traffic.

In this case, the scalability comes mainly from the BB processing rather than from the PA, reducing in this way the potential gains.

For a similar comparison, we simulated a DL transmission with an average of 2 Mbps and an average path loss of 80 dB under a time-varying wireless scenario (Figure 15). Again, we compare the total energy of the MAC-PHY XL solution to two reference state-of-the-art (SoA) policies.

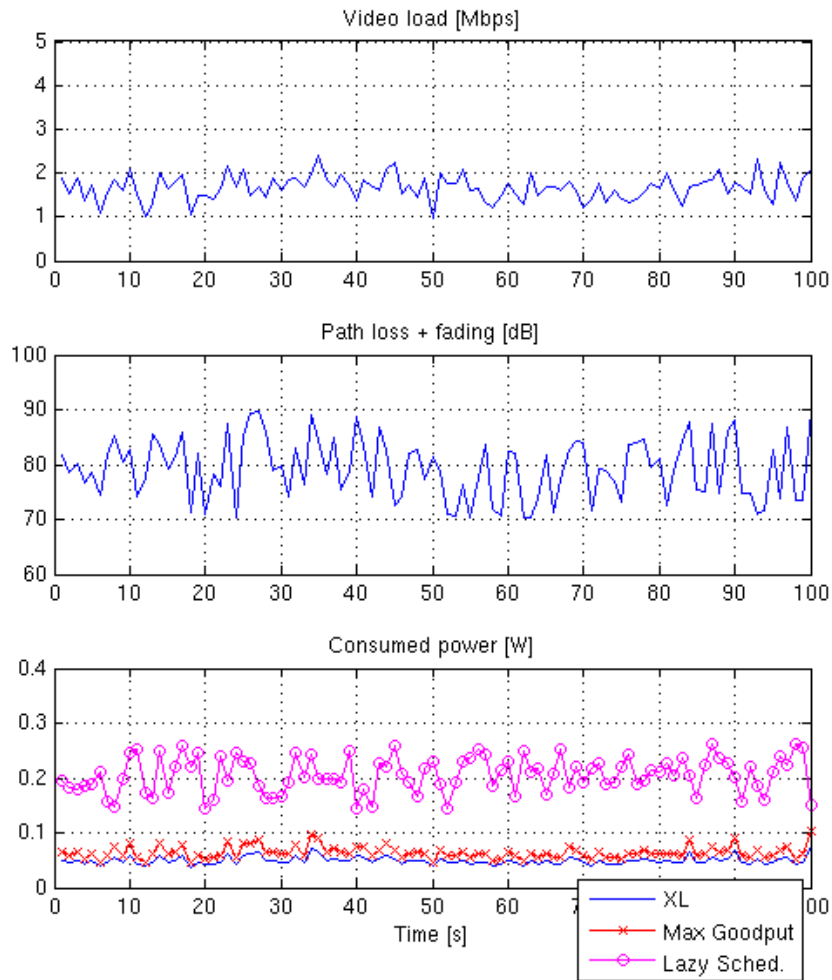


Figure 15 - Video application during a 50 s simulation

As in the previous example, the first two sub-plots on Figure 15 represent the data rate requirements and the instantaneous path loss. The third subplot shows the corresponding required power for the reference cases compared to the XL solution. The gain achieved leads to an average power reduction of 23% compared to maximum goodput and 76% compared to lazy scheduling with an average power of 49 mW. The average power breakdown per platform component is compared for the three different policies in Figure 16.

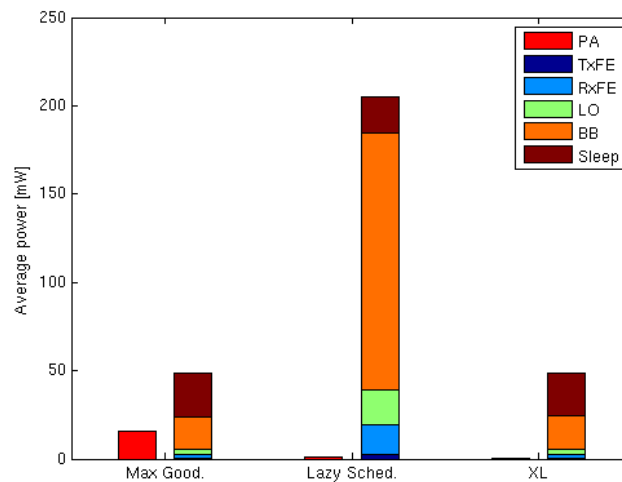


Figure 16 - Average power consumption between XL and SoA solutions for the DL

The potential gains of XL with respect to maximum good-put are reduced because the PA consumption penalty comes only from the transmission of acknowledgements. On the other hand, the gains with respect to lazy scheduling remain high since in the DL it is the duty cycling of the rest of the components that dominates the

power consumption. Clearly, as in the WLAN case, an Overall XL approach can be taken which benefits from the information of upper layers.

4.1.4 Conclusions – Requirements met

The following table summarizes the applicable system requirements established in D2.1.

No	Requirement	Comment
4	The user SHOULD be as much as possible in the dark about using the MobiThin framework.	The adaptivity of the overall XL approach is transparent to the user.
5	The MobiThin framework SHALL support different user movements (user and terminal mobility).	OK
10	The Thin client device using MobiThin adaptive protocol SHOULD be energy efficient and be aware of the trade-off to quality and energy efficiency.	Overall XL approach targets energy efficiency for a required data rate without degrading QoS
11	The power consumed by a Thin client device using MobiThin adaptive protocol SHALL be lower than when using traditional thin client protocol for the most demanding application	At MAC and PHY, XL consumes less energy than SoA solution. This effect is more pronounced for high-traffic applications. Energy gains are highly increased with close interaction between application layers and the wireless link.
28	The thin client device SHALL support at least one wireless communication interface (e.g. WiFi, UMTS, WiMAX, LTE, Bluetooth...)	Currently, XL supports WiFi interface, as well as LTE.
32	The targeted mobile thin client device SHALL support an adaptive cross layer optimisation feature for bandwidth usage reduction, reduced power consumption, optimal user experience whatever the network conditions.	MAC-PHY XL adaptive algorithm optimizes power consumption for a required application data rate under a given channel path loss. With an overall XL algorithm benefits from sleeping periods allowed by the application layer.
42	MobiThin framework SHALL cope with packet loss in the network and adapt the error resilience mechanisms to the various types of applications	MAC-PHY XL algorithm copes with the packet loss by selecting a terminal configuration that provides a certain goodput (error-free throughput) with the lowest power consumption. By including overall XL we can achieve higher energy gains without compromising QoS.

4.2 LINK OPTIMIZATION HARDWARE EMULATION (XMSF)

Supporting the Link Optimization NS2 Emulation, we show the gains achieved by the MAC-PHY approach under a more hardware-oriented simulator (only lower layers MAC and PHY) for an 802.11a transceiver. This simulator is called Cross Layer MAC Simulation Framework (XMSF). Its modeling is based on the IMEC SDR platform [3]. The main difference with the NS2 simulator is that the XMSF is more accurate at hardware level, while NS2 works at packet level.

The main objective of the XMSF emulator is to verify the implementability of the XL approach in an SDR platform by a detailed hardware modeling. This platform-oriented emulator is therefore equipped with both XL and SoA algorithms. In this context, XMSF can only show the operation and potential gains of XL during transmission, i.e., in the UL case, and as we will see, higher gains can be achieved between XL and SoA for larger data rates.

4.2.1 Software/hardware architecture

Since the details of this emulation were given in D5.5, we will present a brief recap only. The XMSF is a set of tools that allows simulating the control software for the IMEC SDR platform. The control software runs on the control processor (ARM) and is written in ANSI C. It implements the data path that forms the 802.11a physical layer and the some time critical functions of the 802.11 MAC layer. Besides that, the MAC-PHY XL functionality has been mapped on it.

With this emulator, several emulated terminals (transmitting or receiving traffic) can be connected together through a wireless channel model. In this way, the PHY and MAC functionality can be validated at network level using a realistic scenario by displaying information such as:

- The channel state between all the simulated terminals, the packet error rate (PER),
- The energy consumption of the major components
- The state of all these major components: on/off, receiving/transmitting, modulation, coding rate.
- The state of the PHY layer control software: Rx/Tx, carrier sensing, transient modes.
- The state of the MAC layer control software: Network Allocation Vector (NAV), SIFS timings, back off, acknowledgements, header information.
- The state of the MAC-PHY XL and state-of-the-art functionality: configuration selection

For each terminal, this information is displayed in a window like in Figure 17.

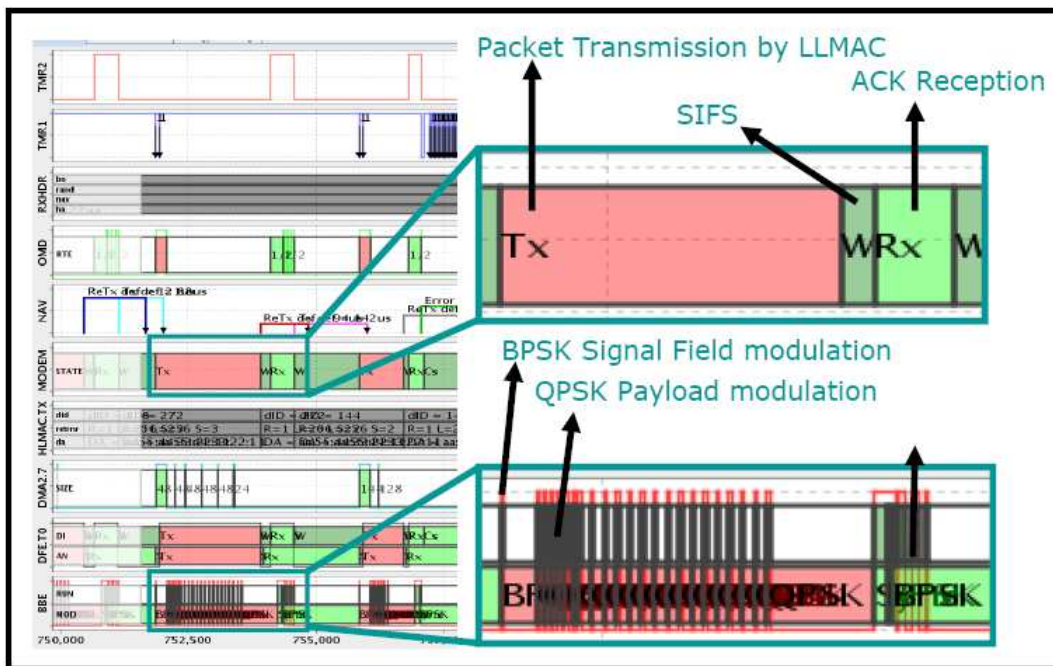


Figure 17 - Timing and state information per component of the SDR platform

Each of the rows represents a different component of the platform, timers, header information. The colors represent the radio state (red for transmitting, green for receiving, light green for idle). This type of demo shows a very accurate description of the SDR platform and represents a valuable tool in the design flow.

4.2.2 Description of experiments performed

Based on the experiments defined on D5.5, we assessed the energy consumption of a thin client with a state-of-the-art solution and with a MAC-PHY XL solution. For this we simulated 2 receiving and 2 transmitting terminals in a WLAN network and compared their average power consumption.

The experiments consisted of varying the data rate transmission for low mobility and fixed path loss. Three different data rates were considered, 2 Mbps, 1 Mbps, and 200 kbps, common to a low-traffic scenario. Compared to NS2 where we use real traffic from application layers, here it is assumed that the data rate is constant and we focus on the modeling and timings of each of the SDR platform components rather than the packet-level simulation. The traffic simulated through XMSF includes the transmission of data packets and the reception of acknowledgements, so no DL implementation of XL is simulated. During each experiment the state of all platform components is simulated with a high timing accuracy and its energy consumption is computed using XL and using SoA approaches.

4.2.3 Results and interpretation

As explained before, the first experiment consists of a 2 Mbps scenario. Below we show the results of a transmitting terminal first using a SoA approach and then XL. The components shown include: CSDI : power consumption for the carrier sensing of the digital platform

- RxAN: receive power in the analogue part
- RxDI: receive power in the digital part
- TxAN: transmit power in analogue part
- TxDI: transmit power in digital part
- WTDI: power consumption in waiting or idle mode (assuming knowledge of the data rate)
- BBE: base band engine
- DFE.T0: digital front end for single-input-single-output (SISO), i.e. one single antenna.

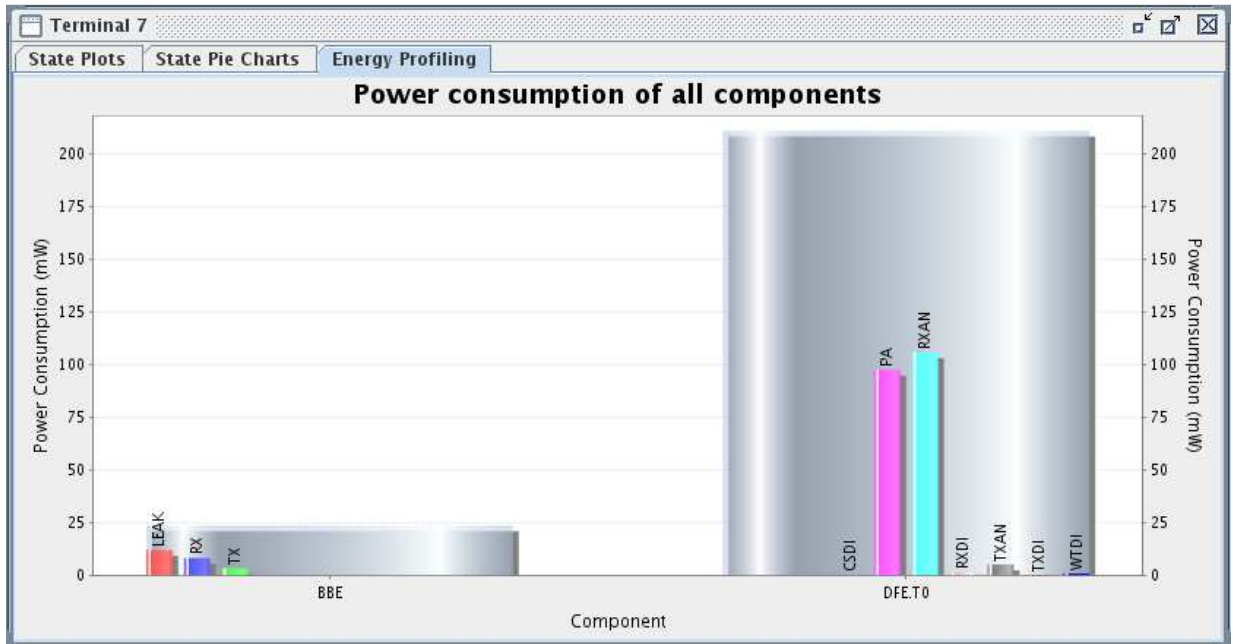


Figure 18 - Power breakdown of the SDR platform using SoA

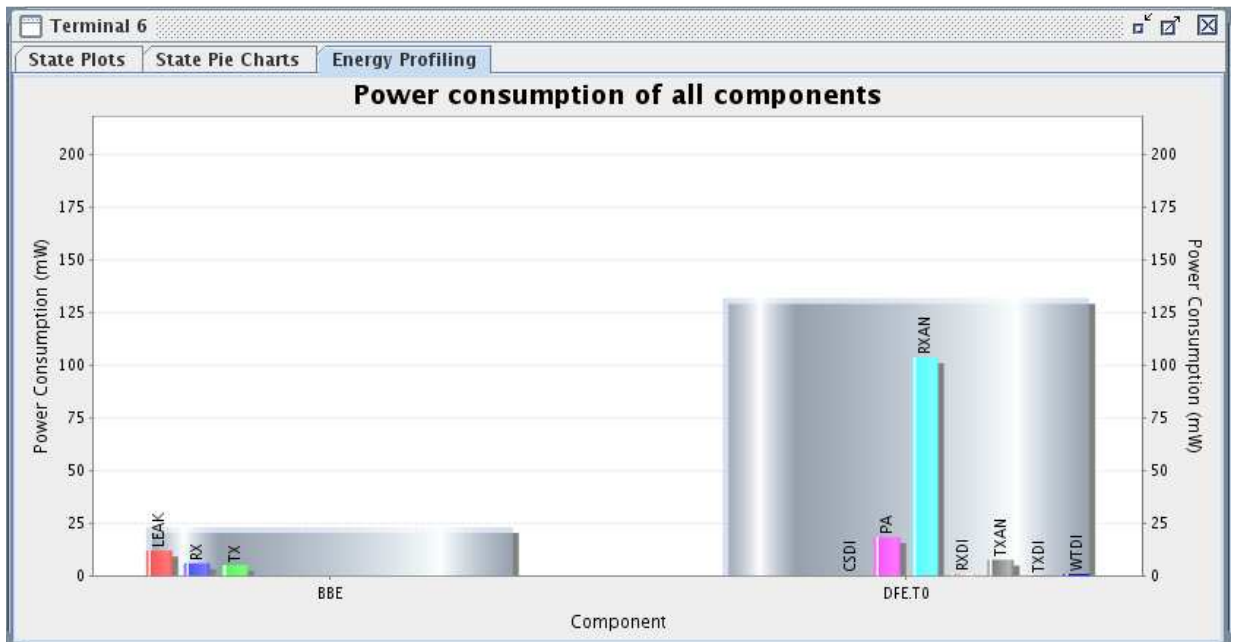


Figure 19 - Power breakdown of the SDR platform using XL

As can be seen in Figure 18 and Figure 19, the average power consumption of the front end (DFE) is dominant over the baseband (BBE), being the Analog FE receiver (RxAN) and the PA the largest contributors. The scalability of the PA exploited by the XL approach contributes to the energy gains achieved, i.e. from an average power consumption of 215 to 132 mW reduction.

A 1 Mbps scenario is shown in Figure 20 and Figure 21 first using a SoA approach and then XL.

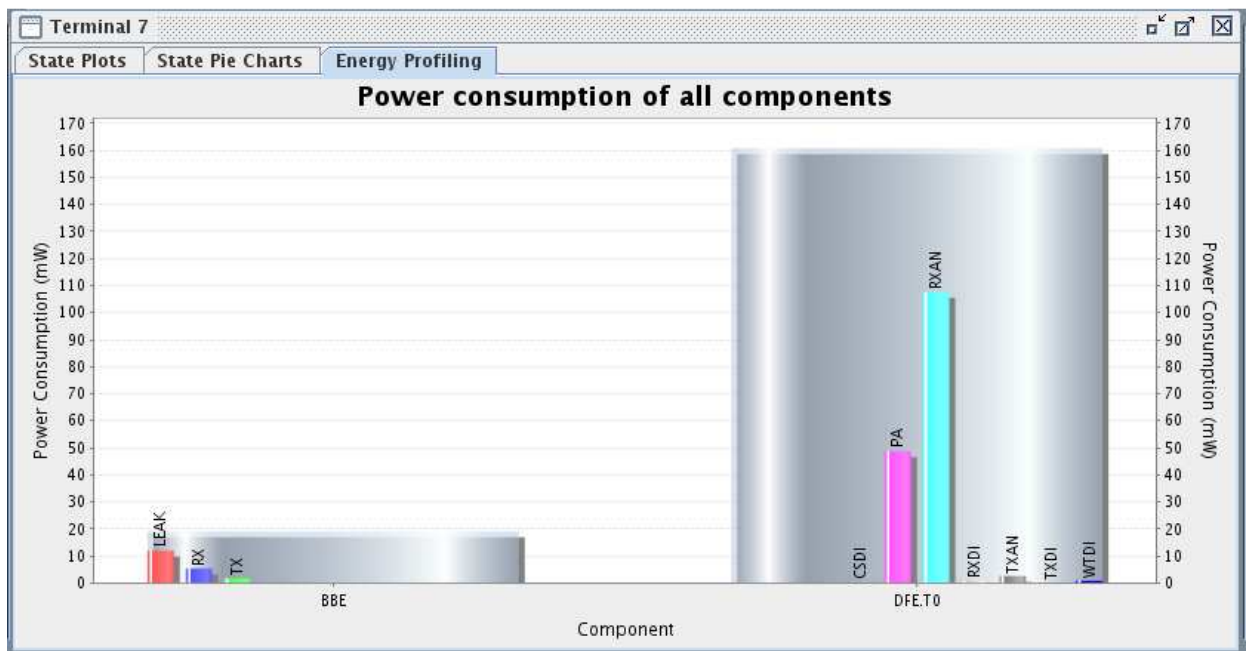


Figure 20 - Power breakdown of the SDR platform using SoA

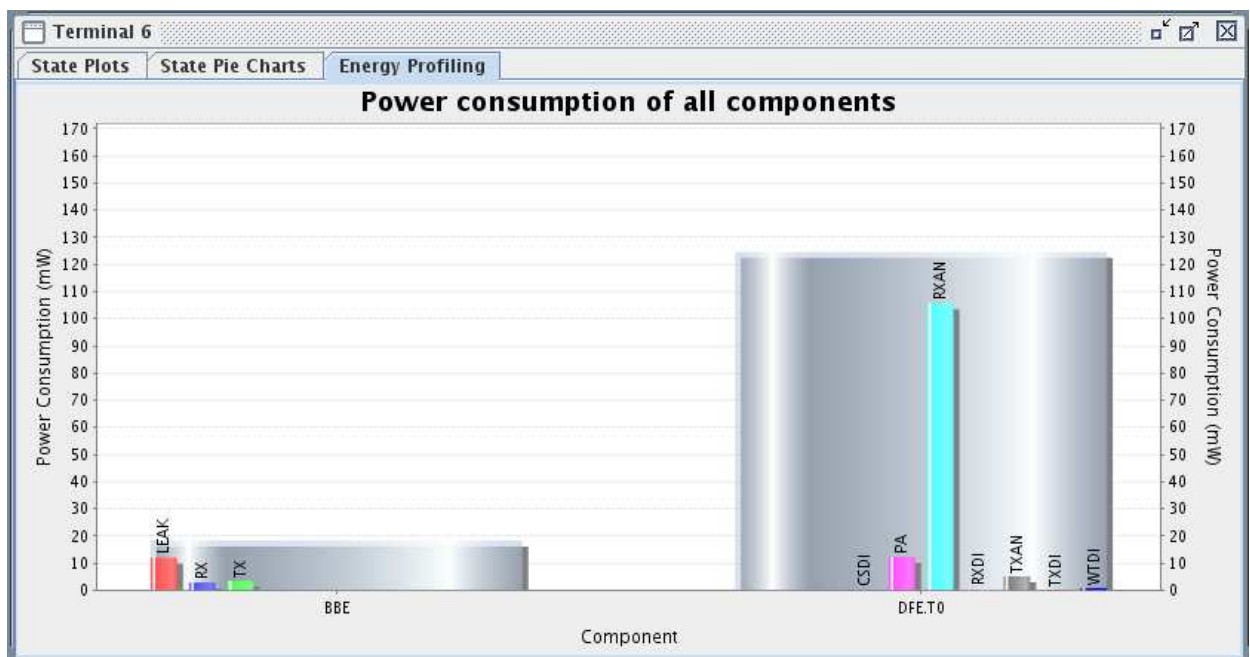


Figure 21 - Power breakdown of the SDR platform using XL

A 200 kbps scenario is shown in Figure 22 and Figure 23 first using a SoA approach and then XL.

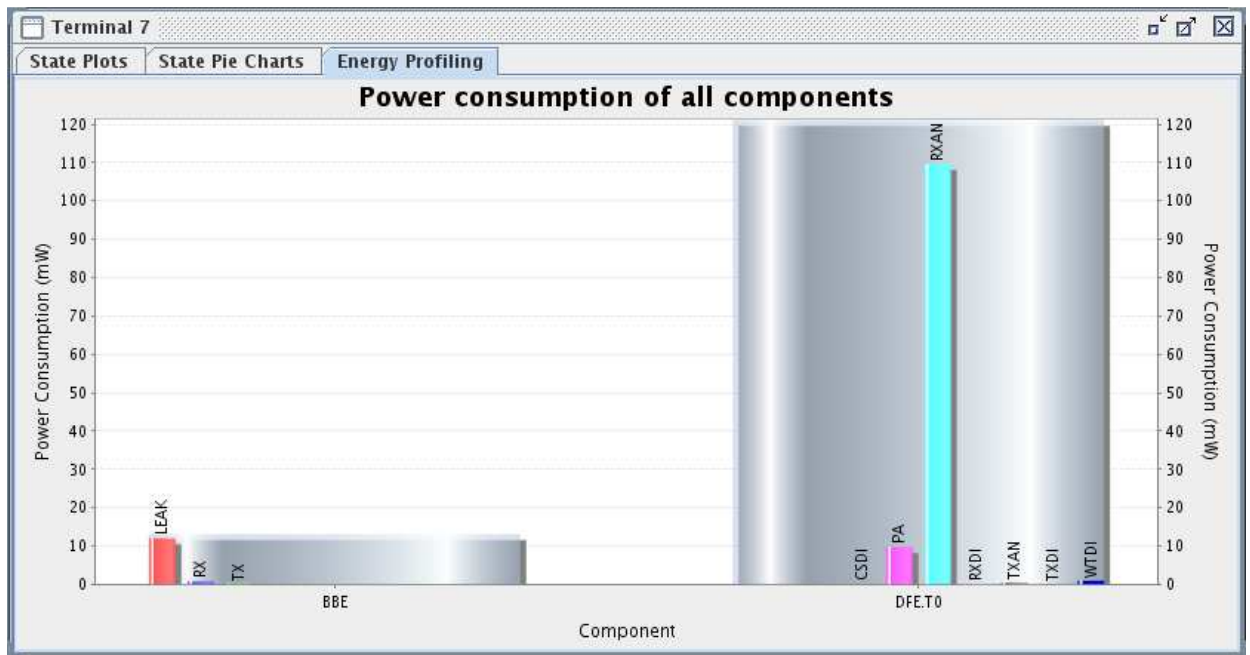


Figure 22 - Power breakdown of the SDR platform using SoA

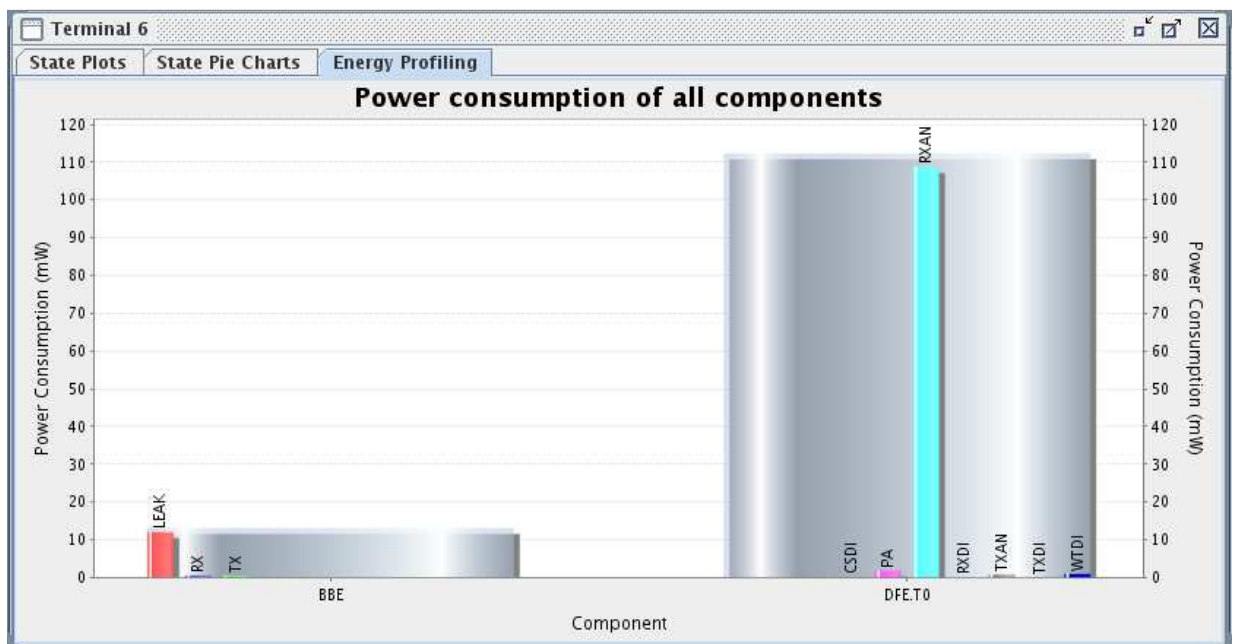


Figure 23 - Power breakdown of the SDR platform using XL

Table 3 shows the results of the power consumption and gains for the three cases described before. As expected this gains are increased with higher UL traffic, due to the PA scalability. As the data rate is decreased, the BBE contribution becomes more dominant and it is from the adaptation of knobs like the modulation and code rate and the duty cycling (ON/OFF duration of the platform) that some energy gains can be achieved. Also a comparison between the gains obtained with XMSF and NS2 for the same scenario is presented. As can be seen there is a high correlation between the results obtained with both emulators.

As a logical consequence, these results point out that even higher energy gains can be achieved through a tighter communication with upper layers as was demonstrated with the Overall XL implemented in NS2.

Table 3: Power consumption and energy gains for both state-of-the-art (SoA) and the cross-layer algorithm (XL)

Data Rate	Power consumption SoA (mW)	Power Consumption XL (mW)	Energy gains shown with XMSF	Energy gains shown with NS2
2 Mbps	239	156	35%	35%
1 Mbps	178	140	21%	22 %
200 kbps	131	122	7%	6%

4.2.4 Conclusions – Requirements met

The following table summarizes the applicable system requirements established in D2.1.

No	Requirement	Comment
4	The user SHOULD be as much as possible in the dark about using the MobiThin framework.	The adaptivity of the overall XL approach is transparent to the user.
5	The MobiThin framework SHALL support different user movements (user and terminal mobility).	OK
10	The Thin client device using MobiThin adaptive protocol SHOULD be energy efficient and be aware of the trade-off to quality and energy efficiency.	MAC-PHY XL approach targets energy efficiency for a required data rate without degrading QoS
11	The power consumed by a Thin client device using MobiThin adaptive protocol SHALL be lower than when using traditional thin client protocol for the most demanding application	At MAC and PHY, XL consumes less energy than SoA solution, more pronounced for high-traffic and UL applications.
28	The thin client device SHALL support at least one wireless communication interface (e.g. WiFi, UMTS, WiMAX, LTE, Bluetooth...)	The XMSF simulator supports currently and 802.11a transceiver
32	The targeted mobile thin client device SHALL support an adaptive cross layer optimisation feature for bandwidth usage reduction, reduced power consumption, optimal user experience whatever the network conditions.	MAC-PHY XL adaptive algorithm optimizes power consumption for a required application data rate under a given channel path loss. This was shown with a detailed platform modeling from the XMSF.
42	MobiThin framework SHALL cope with packet loss in the network and adapt the error resilience mechanisms to the various types of applications	MAC-PHY XL algorithm copes with the packet loss by selecting a terminal configuration that provides a certain goodput (error-free throughput) with the lowest power consumption.

5. OVERALL CONCLUSIONS

Text in common for D5.6 and D5.7

5.1 RESULTS SUMMARY

Evaluation reports were presented on well-selected PoC's covering essential parts of the MobiThin system, as well as evaluating integrated subsystems. The PoC's, defined in D5.5 were selected on this basis (i.e. optimising coverage of the MobiThin system), and all have been realized according to the design envisaged in D5.5. An additional PoC, addressing direct access of local peripherals by the mobile terminal, has been designed, realized and evaluated. Each of the proof-of concepts reported were designed, observing the architectural requirements (D2.2) and the specified interfaces.

As major results from the PoC related work, we specifically mention:

- Prototyping of all major components of the MobiThin architecture, and inclusion of key-components in an integrated demonstrator.
- Important protocol optimization strategies were designed and evaluated, more specifically
 - optimization of energy consumption, through various cross-layer approaches (MAC-PHY as well as intelligent buffering of upstream traffic)
 - a new MPEG compliant image transmission has been developed,
 - localization of video content, in order to adapt the transmission strategy to the specific content to deliver,
- The Service Management Framework was developed and evaluated in an E2E setting to assess functional correctness. In addition energy saving and session migration strategies were designed and thoroughly evaluated on a emulated environment.
- Compliance with IMS by including SIP-based demonstrators was achieved
- The I/O facilities for the mobile thin client were enhanced considerably:
 - remote peripherals were demonstrated (focusing on USB devices – enabling remotely running applications to access advanced I/O facilities of mobile terminals)
 - the issue of accessing advanced local peripherals (i.e. I/O devices connected to the mobile terminal) were successfully demonstrated.
 - an application based on speech input was demonstrated. On the mobile terminal, an innovative demonstration of peripherals has been performed, allowing the mobile terminal to access local peripherals thereby optimizing I/O and hence user experience.

5.2 REQUIREMENT COVERAGE

Each of the PoC satisfies an important number of requirements, already specified in D2.1. Nevertheless, not all requirements have been demonstrated in the PoC's of D5.6/7.

Following major reasons why some requirements are not covered by the demonstrations can be identified:

1. The requirement is on the MobiThin environment rather than on the MobiThin system itself (e.g. underlying network infrastructure). As such, these requirements are not to be met by the MobiThin system, but rather by this environment. (This was also the case for Phase I demonstrations.)
2. Although implemented as components, some components were not included in the PoCs. This is the case for
 - a. the monitoring subsystem
 - b. the application delivery subsystem
 - c. the business support system

For the latter (i.e. the business support system), the approach was followed to provide standard interfaces to business support systems (many competing products are available), rather than to tightly integrate one such system. Therefore, the business support functions are not demonstrated, but it can be readily expected that offering standard interfaces will allow proper operation of business support functions. The monitoring subsystem was not included as a whole (some monitoring functions were used in specific demonstrations) in the integrated demonstration, mainly due to timing constraints (here also, the use of

standard interfaces (in this case SNMP) reduces the risk for integration issues). Including the application delivery subsystem in the integrated demonstration is currently underway.

In the table below, a quantitative overview is given to show how many requirements fall within each of the categories. A limited number of requirements (6) can not be attributed to one of the categories listed above, and will be further detailed below.

Type of Requirement	SHALL	SHOULD	MAY	Total
<i>Covered in Phase I demos</i>	28	21	11	60
<i>Covered in Phase II demos</i>	34	20	8	62
Covered in Phase I or Phase II demos	39	28	12	79
Requirement on environment	1	3	1	5
Monitoring subsystem not included in demo	21	7	2	30
Business subsystem not included in demo	9	2	0	11
Application Delivery Service not included in demo	5	3	0	8
NON-requirement (54)	1	0	0	1
Total	76	43	15	134

Requirement coverage: quantitative overview.

Regarding the requirements on the MobiThin environment, these mainly address the targeted device (note that many of the requirements on these devices have been included in the demo “Mobile thin client Integration”) and the network environment.

21	SHOULD	The targeted mobile device SHOULD have a built-in or external smartcard reader
26	MAY	The mobile device MAY provide for other interfaces such a 1D/2D bar code reader, a RJ45 interface
55	SHOULD	The network layer SHOULD enable terminal mobility
66	SHALL	Data storage servers SHALL allow users to access their data from outside of a thin client session.
67	SHOULD	Data SHOULD be transferred to TC Server at very high speed to avoid any additional latency

Requirements on MobiThin environment.

Requirements not demonstrated, which are not included in the categories described above, are given in the table below. All these requirements, except for requirement 59, were identified as “under discussion for demonstration in Phase II” in D5.3/4

1	SHALL	The MobiThin framework SHALL support residential and business user.	After discussion, found not relevant to include in project demo’s.
41	SHOULD	The dejitter strategy SHOULD be adapted based on the intrinsic properties of the traffics	The project has focused on real time traffic (no video/audio playback, for which dedicated solutions exist). Dejittering is hence not applicable.
59	MAY	TC server MAY run Windows applications	Not included in the demo, no problems expected due to virtualisation approach taken at the TCS level.

76	MAY	SMF MAY take into account anonymous users	After discussion, found not relevant to include in project demo's (a dedicated account can be crated for test purposes, reducing the need for anonymous users).
80	MAY	SMF MAY additionally support one or more of the following authentication methods (non exhaustive list): - SSO - Access Control Authentication - Network Authentication ...	As MobiThin is not focusing on AAA issues, this MAY requirement was not taken on board for the componente work.
128	MAY	Session migration service MAY be suspended	Not implemented, but straightforward to do so (no architectural problem, only a matter of updating the session migration algorithm).

Miscellaneous requirements not covered by the PoC's.

5.3 FINAL REMARKS

All Proof-of-Concepts originally envisaged have been realized, showing not only proper operation of components and subsystems in isolation, but also in an integrated end-to-end system. For the subsystems currently not integrated, the architectural approach and the use of standardized interfaces reduces any risk for further integration. As such, the realization on the PoC will allow further integration of the total system, bringing the solution closer to a commercial product.

In addition, we can state that all main requirements have been addressed, as is also concluded in the system review, reported in D2.6. The system also allows for advanced I/O facilities (including audio/video input through remote peripherals and audio/video output on high-end devices connected to the mobile terminal). Compliance with IMS, a very important aspect for deployment in telco environment. has been studied in D2.6 and some of the PoC were indeed designed to demonstrated this compliance.

6. REFERENCES

- [1] A. Dejonghe, B. Bougard, S. Pollin, L. Van der Perre, F. Catthoor, Green Reconfigurable Radio Systems: Creating and Managing Flexibility to Overcome Battery and Spectrum Scarcity. IEEE Signal Processing Magazine, special issue on Resource-Constrained Signal Processing, Communications, and Networking. May 2007.
- [2] 3GPP TS 36.213, "Physical Layer Procedures", V8.8.0, September 2009.
- [3] B. Bougard, et al. A Coarse-Grained Array based Baseband Processor for 100Mbps+ Software Defined Radio, Proceedings of the conference on Design, Automation and Test in Europe 2008, pp. 716-721
- [4] P. Simoens et al. "Characterization of power consumption in thin clients due to protocol data transmission over IEEE 802.11". In 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless. IEEE 2009, Seoul 2009.
- [5] J. Glossner et al. The Sandbridge SB3011 platform. EURASIP journal on embedded systems, Volume 2007, issue 1, January 2007.
- [6] J. Glossner et al. The Sandblaster SBX 2.0 Architecture, SDR forum, 2007.
- [7] V. Deruder et al. A 200Mbps+ 2.14nJ/b digital baseband multi processor system-on-chip for SDRs, 2009 Symposium on VLSI Circuits Digest of Technical Papers.