Protocol Implementation for a 5 GHz OFDM-Testbed

H.Wijaya[2,3], N.Esseling[2], O.Klein[2], A.Vidal[2], W.Zirwas[1]

[1] = Siemens AG, [2] = ComNets, RWTH Aachen, [3] = AixCom GmbH

Ph.: + 49 241 80 27 954, Fax: + 49 241 80 22 242, Kopernikusstr.16 – 52056 Aachen - Germany

e-mail: hari@comnets.rwth-aachen.de

ABSTRACT

This paper provides a comprehensive overview of the development of performance critical communication systems such as HiperLAN/2(H/2), the corresponding timing concepts for H/2 Mobile Terminal (MT) and Access Point (AP) protocols and the protocol implementation in a hard real time system. Moreover, the paper introduces some concepts to optimize the processing power sharing among real time and non real time tasks.

1. INTRODUCTION

The project COVERAGE (Cellular OFDM system with Extension Points for increased transmission RAnGE), supported by the German Ministry of Education and Research (BMBF), was initiated by the Siemens AG. The main objective of this project is to observe and to develop Multi-Hop networks [6][7]. For this purpose, an OFDM (Orthogonal Frequency Division Multiplexing) testbed that operates in the 5 GHz frequency band, is being developed.

The Quality of Service supporting H/2 protocol based on the ETSI Broadband Radio Access Network (BRAN) standard [16][17] has been chosen to run on this testbed. The protocols for both AP and MT will be formally specified in SDL (Specification and Description Language) [13][14]. Timing issues are the most critical part in the H/2 protocol development. Therefore, this paper provides implementation concepts to run the H/2 SDL protocol in the Real Time (RT) Linux kernel space as real time module being part of the real time operating system. Using RT Linux, the H/2 protocol has lower delay and lower interrupt latency. Thus it is of substantial interest to figure out the best way how to share the computing power between the time critical tasks and the non time critical tasks which should not be blocked at all.

2. TESTBED ARCHITECTURE

The generic architecture of the testbed is shown in figure 1. A similar hardware/software co-design for broadband systems has already been proven in [18]. The OFDM part of the testbed depicted as the lower block contains the modem as well as the transceiver and receiver. The upper block includes a hardware and a software component. The hardware can be used to implement the most time critical parts of the protocol. The software comprises the remaining parts of the protocol stack as well as the Modem Interface Software (MIS).

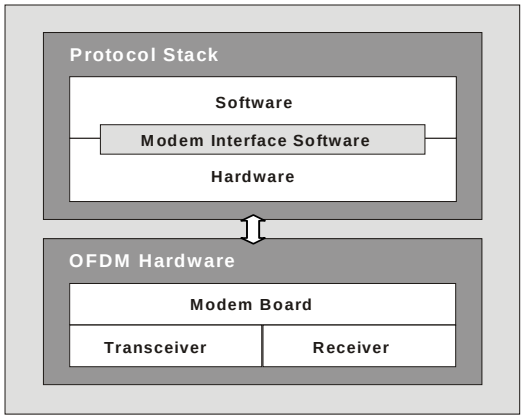

Figure 1: Generic Architecture of the Testbed.

The MIS is responsible for the communication flow between the different parts of the protocol stack located in soft- and hardware.

Figure 2 shows the final target architecture of the testbed running the H/2 protocol including the functional entities in the H/2-Protocol Engine (PE).

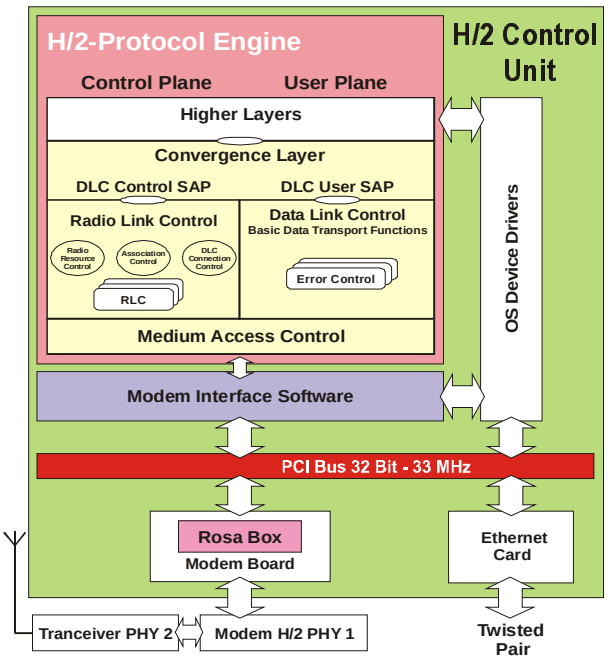

Figure 2: Testbed Architecture

The H/2-PE is running in the Control Unit which is a high end Intel based PC Hardware with PCI bus. Due to hard timing constraints, it is necessary to put some functionality of the H/2 Medium Access Control (MAC) Layer into the Radio Operation Service Accelerator (ROSA)-Box on the Modem Board to be on the safe side. The H/2-PE is a program running the H/2 protocols as depicted on the PC hardware. The software part of the MAC-Layer communicates with its hardware counter part via Interrupts (IRQ). The communication flow controlled by the MIS is realized via the PCI bus. The hardware architecture of AP and MT are identical.

3. TIMING REQUIREMENTS

3.1 Access Point

In figure 3 the timing within the AP is shown. Since the AP is responsible for the scheduling of the capacity for all its MTs in both transmission directions (uplink (UL) and downlink (DL)) the process is more time consuming than at the MT.
The MAC-Frame is generated during the foregoing MAC-Frame. The beginning of the generation process is initiated by an IRQ sent by the ROSA-Box to the MIS. The software part of the Data Link Control (DLC) [4][5] will begin with the generation of the Broadcast Channel (BCH), Frame Channel (FCH) and DL phase. In the following MAC frames, the ROSA-Box will then trigger the IRQ to the DLC only after finishing the Access feedback Channel (ACH) phase. After the Random Channel (RCH) phase ends, the ROSA-Box triggers another IRQ to notify the DLC that it may begin to evaluate the UL and RCH phases.

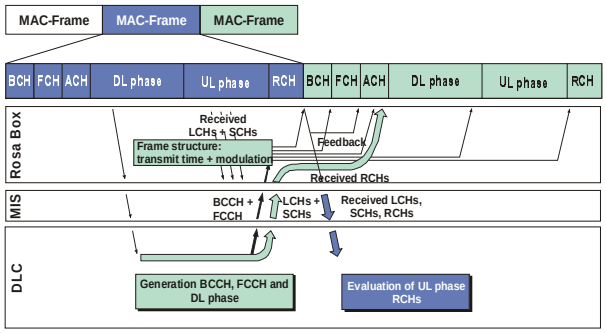

Figure 3: Timing of the Base Station

It has to be guaranteed that the generation of the MAC-Frame is finished in time. Furthermore the time required for signal processing must be considered. The data which is transferred to the ROSA-Box can be divided into two groups: information for the broadcast phase of the MAC-Frame (BCH, FCH) and the data for the DL phase. The DL data is provided as sequences of Short transport Channels (SCH) and Long transport Channels (LCH). These sequences are ordered according to the sequence they are sent out.
As shown in figure 3 the ROSA-Box obtains the modulation/coding scheme and the transmission/receive time from the BCH/FCH information. This reduces the processing time as well as the amount of data which has to be transferred to the ROSA-Box.
The data received on the uplink (LCHs, SCHs, RCHs) is processed when processing time is available. This will usually be after the generation of the MAC-Frame. Besides the received data, the protocol also requires information about the result of the Cyclic Redundancy Check (CRC) and the link quality which is transferred attached to the data.
The acknowledgements transmitted within the ACH are related to the RCH of the previous MAC-Frame. Since the time to generate the ACH is limited to the duration of the BCH and FCH, it is generated by the ROSA Box.
In the AP all Protocol Data Units (PDU) of all MAC_IDs are received. The receiving time and Physical (PHY)-modi [1] have already been announced inside the FCH.

3.2 Mobile Terminal

In the following the timing inside the MT and the dependencies from the MT's point of view are explained.
Figure 4 shows the timing of the MT. The BCH contains information about the position and the length of the FCH. Thus, it is evaluated inside the ROSA-Box to be able to receive the FCH and transferred to the DLC, where it is probed. The Information Element (IE)-Blocks of the FCH contain all required information for the DL phase which is transmitted immediately after the ACH.
The time which is available for the MT to generate the data to be transmitted during the UL phase is limited by the duration of the DL phase. Therefore, the LCHs and SCHs received in the DL phase are not evaluated by the DLC immediately. They will be requested after the UL data has been provided. In the meantime they are stored by the MIS.

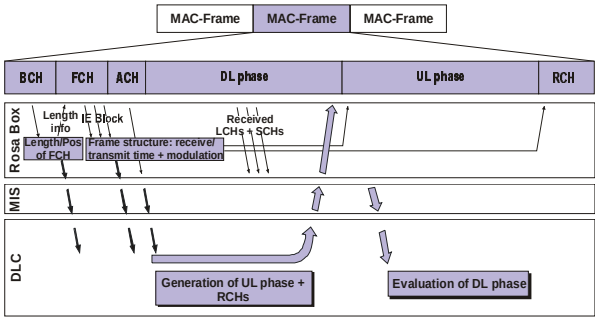

Figure 4: Timing Mobile Station

The DLC is triggered by the ROSA-Box to generate the UL phase. The trigger is initiated by the ROSA-Box by an IRQ after receiving BCH/FCH/ACH data. Initially it is proposed to transfer the BCH/FCH/ACH data and the receiving conditions in one block to save latency time. Therefore a minimum time is required to arrange the uplink data.
The production of the UL data is completed by sending the information via the MIS to the ROSA Box. The data is provided as a sequence of SCHs and LCHs and the RCH if one is required. The order of SCHs and

LCHs depends on the MAC_IDs to be served and the connections to be served.
No information is attached to the LCH/SCH sequences, as the information on sending time and PHY-mode vs. MAC_ID is already known to the ROSA Box.
When the DLC generates the UL data the ROSA-Box receives the DL data autonomously. The data is transferred to the DLC when the DLC has provided the UL data. The data is transferred as a sequence of LCHs/SCHs together with the relevant receiving conditions and link quality. For each announced PDU of the served MAC_IDs at least the receiving information is available.

4. PROTOCOL SPECIFICATION

The H/2 protocol [1][2][3][4][5] has formally been specified using SDL. Figure 5 depicts the SDL specification of the H/2 AP and Figure 6 depicts the SDL specification of the H/2 MT respectively.

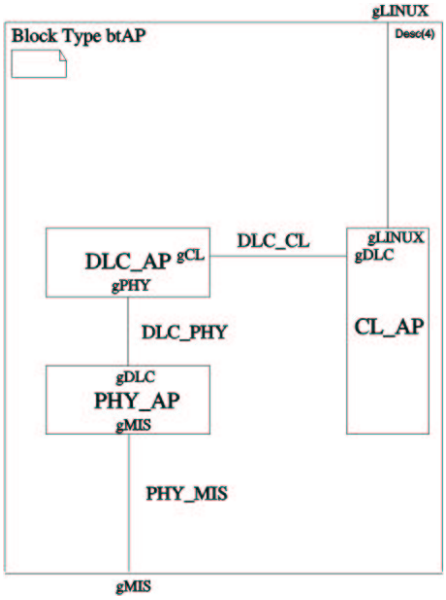

Figure 5: H/2 AP in SDL

As shown, both AP and MT have the same block structure which contains PHY, DLC and Convergence Layer (CL) [2][3] blocks only with some different functionalities.

For instance, the DLC block of the AP is responsible for controlling resource assignment/grant in a cell where the AP resides and the DLC of the MT is responsible for requesting resources. Moreover, the PHY blocks of both AP and MT are responsible for data mapping and data controlling between the DLC block and the MIS whereas the CL blocks of both AP and MT are responsible for data mapping between the DLC block and a Linux module.

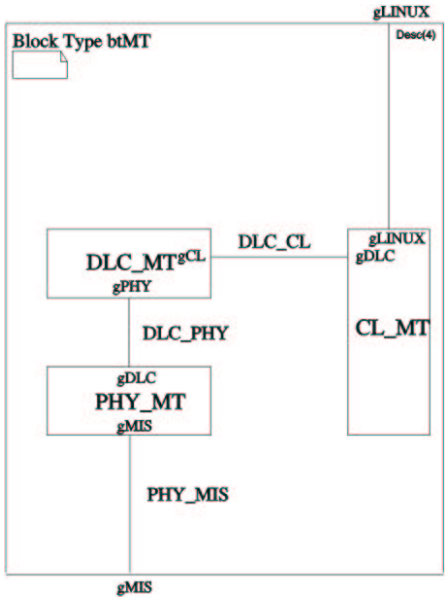

Figure 6: H/2 MT in SDL

5. HIPERLAN/2 IS A REAL TIME SYSTEM

Since H/2 requires a highly strict time requirement due to its extremely short MAC frame which lasts only 2 ms, it has been considered to implement its protocol into a real time system.
A real time system can be defined as a system that performs its function and responds to external, asynchronous events within a specified amount of time [8][15].
These systems are characterized by the fact that several consequences will result if logical as well as timing correctness properties of the system are not satisfied. There exist *hard* and *soft* real time systems. Soft real time systems are those in which timing requirements are statistically defined and in which a miss of a deadline can be occasionally allowed. However in a hard real time system, the deadlines must be guaranteed.
Since all H/2 tasks can be noted as time critical tasks, it is basically important to consider the computing power management so that each time critical task will meet its timeliness requirement, whereas non time critical tasks such as Fast-Ethernet and Internet Protocol (IP) modules will be executed so efficient that the average response time of such tasks can be minimized.
The most time critical tasks of H/2 is the generation of the BCH and the DL data at the AP.

6. H/2 PROTOCOL IMPLEMENTATION FOR RTLINUX

SDL provides a very good and efficient way to formally specify communication protocols but the execution time of standard SDL specifications is very slow which makes such specifications not well suited for the implementation in a real time system.

Moreover, due to its stand-alone data structure, it is even more difficult to develop a hardware interface which uses both SDL and hardware data structures.

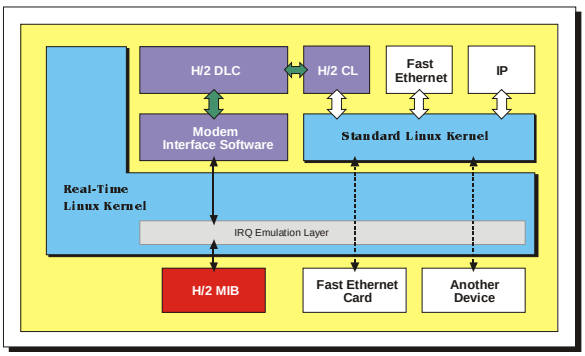

Figure 7: H/2 protocol in real time system

Beyond it, SDL executables introduce another extremely high delay since such executables run in the user space. Therefore, in order to fulfill the highly strict time requirements of a H/2 system, some modifications should be made so that the H/2 SDL specification can be implemented in a real time system.

Figure 7 depicts the implementation of a H/2 protocol into a real time environment. In this case, the H/2 protocol is no longer an executable running in the user space but it is an operating system module running in the kernel space, which runs in parallel to the standard Linux kernel.

The reason for compiling the H/2 protocol as a Real Time Linux module is that standard Linux supports only Round-Robin scheduling [9][10][11] which guarantees fairness among all tasks while Real Time Linux supports priority scheduling [12] which guarantees lower delay for high priority tasks. As shown, the MIS and the H/2 DLC protocol are Real Time Linux modules whereas the H/2 CL is a Linux module. The CL acts as a *bridge* for data communication between the H/2 DLC protocol and other Linux modules such as IP and Fast Ethernet.

In order to make SDL protocols running in the kernel space as Real Time Linux module, some modifications in the SDL kernel should be undertaken.

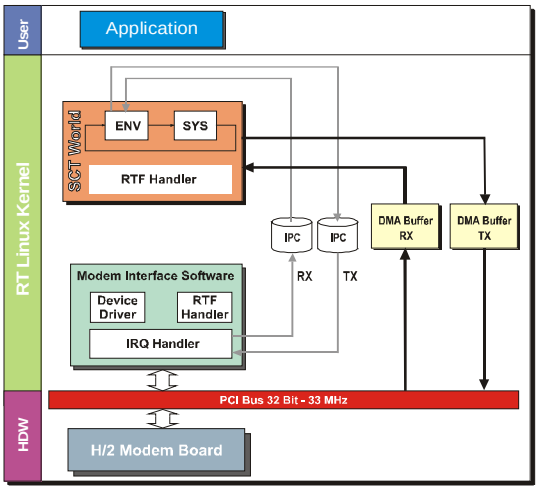

Figure 8: SDL Kernel–soft Real Time approach

Two different approaches namely *soft Real Time* and *hard Real Time* approaches will be introduced in the following.

Figure 8 depicts the implementation of the SDL kernel using the soft Real Time approach. In this approach, the SDL kernel is programmed to be active after a specific predefined interval and to be non active after its corresponding tasks have been completed. In this case, every time when the MIS intercepts IRQs from the H/2 modem board, it will then write a corresponding message into the Receive (RX) Inter Process Communication (IPC) queue so that when the SDL kernel is active, it will then instruct the environment (ENV) to poll the RX IPC queue to check whether there has been an IRQ or not.

If there is an specific IRQ, the ENV will notify the System (SYS) which here is equal to the SDL protocol to carry on some corresponding tasks. After finishing all tasks, the SYS will notify the ENV to send a message through the Transmit (TX) IPC queue to the MIS that some tasks have been completed and it will put itself into the inactive state.

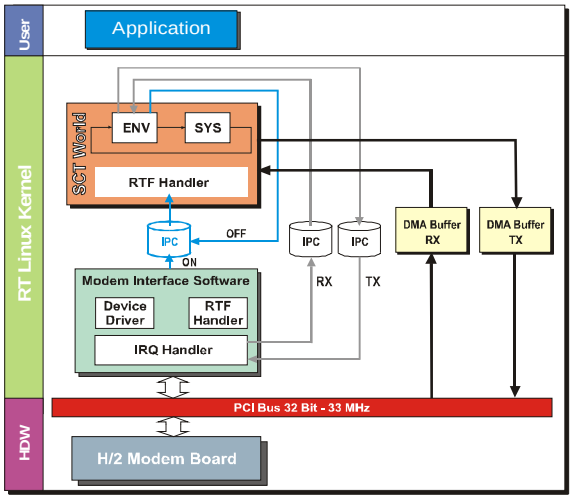

Figure 9: SDL Kernel–hard Real Time approach

Figure 9 depicts the implementation of the SDL kernel using the hard Real Time approach where the SDL kernel will be active only when the MIS sends an "ON" signal and a message through the RX IPC queue after receiving an IRQ from the H/2 modem board. The SDL kernel will then carry on executing the same tasks as in the soft Real Time approach when it is in active state. Only when the SYS has finished carrying on its tasks, it will then send an "OFF" signal to the SDL kernel so that the SDL kernel can put itself into the inactive state.

Therefore the latter approach is more suitable than the first approach in context of timing requirement. The only disadvantage is that to perform debugging is more difficult than in the first approach. This can be well described in a situation when the H/2 modem board does not send any IRQ due to lost of synchronisation (e.g an MT is too far away from an AP), the SDL kernel will be always inactive so that some Radio Resource Management activities such as Handover or Dynamic Frequency Selection (DFS) cannot be

performed anyway due to inactiveness of the SDL kernel which will lead to some erroneous situations.
To prevent this a status IRQ is invented to switch the DLC into the active state and inform it about the actual HW status.

7. SUMMARY AND CONCLUSIONS

In this paper the implementation of the H/2 protocol in a 5GHz OFDM-Testbed is introduced. The Testbed offers the possibility to implement some parts of the protocol in hardware and others in software. It is reflected that the protocol of the Access Point is more time critical than the one of the Mobile Terminal. The time critical software components were located in the more suitable hardware concerning the latency of reaction. To meet the timing requirements beyond this a Real Time Environment with Real Time Linux as Operating System is chosen. Therefore some modifications in the SDL specification have to be undertaken. It is shown that the implementation of the SDL kernel using the hard Real Time approach has the best timing performance.
The system currently is in the implementation phase.

ACKNOWLEDGEMENTS

This work was supported by the German research project COVERAGE (Cellular OFDM systems with extension points for increased transmission RAnGE) funded by the German Ministry for Education and Research (BMBF) and the SIEMENS AG Munich/Bocholt.
The authors would like to thank the members of the project for the valuable discussions.

REFERENCES

[1] ETSI TS 101 475, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer”, Februar 2001
[2] ETSI TS 101 493-1 V1.1.1, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet Based Convergence Layer; Part 1: Common Part”, April 2000
[3] ETSI TS 101 493-2 V1.1.1, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet Based Convergence Layer; Part 2: Ethernet Service Specific Convergence Sublayer (SSCS)”, April 2000
[4] ETSI TS 101 761-1 V1.1.1, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions”, April 2000
[5] ETSI TS 101 761-2 V1.1.1, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC) sublayer”, April 2000
[6] W. Zirwas, T. Giebel, N. Esseling, E. Schulz, J. Eichinger, “Broadband MultiHop Networks for Public Hot Spot Scenarios”, Wireless World Research Forum 3rd Meeting Stockholm, September 2001
[7] N. Esseling, “Extending the Range of HiperLAN/2 Cells in Infrastructure Mode using Forward Mobile Terminals”, European Personal Mobile Communication Conference 2001 Vienna, February 2001
[8] J. A. Stankovic, K. Ramamritham, “Hard Real-Time Systems”, IEEE Computer Society Press Order Number 819
[9] D. P. Bovet, M. Cesati, “Understanding the Linux Kernel”, O’Reilly, ISBN 0-596-00002-2, January 2001
[10] A. Rubini, “Linux Device Drivers”, O’Reilly, ISBN 1-56592-292-1, February 1998
[11] M. Beck, H. Böhme, M. Dziadzka, U. Kunitz, R. Magnus, D. Verworner, “Linux Kernel Internals”, Addison-Wesley, ISBN 0-201-33143-8, 1999
[12] M. Barabanov, “A Linux-based Real Time Operating System”, master thesis, http://www.rtlinux.org , June 1997
[13] ITU-T, “Recommendation Z-100, Specification and Description Language SDL”, November 1999
[14] A. Mitschele-Thiel, “Systems Engineering with SDL, Developing Performance-Critical Communication Systems”, John Wiley & Sons, Ltd, ISBN 0-471-49875-0, 2001
[15] B. Fuhrt, D. Grostick, et al. “Real-time UNIX systems: design and application guide”, Kluwer Academic Publishers Group, Norwell, MA, USA, 1991.
[16] B.Walke, N.Esseling, J.Habetha, A.Hettich, A.Kadelka, S.Mangold, J.Peetz, U.Vornefeld, “IP over Wireless Mobile ATM – Guaranteed Wireless QoS by HiperLAN/2” in Proceedings of the IEEE, Vol.89,pp.21-40, January 2001
[17] ETSI TR 101 173, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; System Overview”, Februar 2000
[18] U.Vornefeld, N.Esseling, “SAMBA: The Proof of a Concept” in Proc. Ot the 4th European Personal Mobile Communications Conference (EPMCC’01), Session 26.5, Vienna, Austria, February 2001