Camera ready version of paper submitted for

MTT-S Wireless 98

5-9 October 1998, Amsterdam, The Netherlands

Authors: Ulrich Vornefeld, Markus Radimirsch, Andreas Krämling, Martin Bornemann

Title: **Experiences with the Implementation of a 34 Mbit/s Wireless ATM Air Interface**

Address: Aachen University of Technology (RWTH)
Communication Networks (COMNETS)
Kopernikusstr. 16
52074 Aachen, Germany
Tel.: +49-241-80-7928
Fax.: +49-241-8888-242
E-Mail: {ulvo|akr}@comnets.rwth-aachen.de,
{Markus.Radimirsch|Martin.Bornemann}@fr.bosch.de

Topics: Implementation of protocols, SAMBA project, Wireless ATM demonstrator

Experiences with the Implementation of a 34 Mbit/s Wireless ATM Air Interface

Ulrich Vornefeld[*], Markus Radimirsch[o], Andreas Krämling[*], Martin Bornemann[o]

[*] Communication Networks, Aachen University of Technology, e-mail: ulvo@comnets.rwth-aachen.de

[o] Robert Bosch GmbH, Corporate Research and Development, e-mail: Markus.Radimirsch@fr.bosch.de

ABSTRACT

This paper focuses on implementation issues concerning the demonstrator platform of the ACTS project SAMBA (System for Advanced Mobile Broadband Applications). Examples of the hardware architecture, the software design and implementation approaches are described for a protocol processing which has to cope with bit rates of up 68 Mbit/s (34 Mbit/s simultaneously on up- and downlink).

INTRODUCTION

Based on the research carried out in the R2067 MBS (Mobile Broadband System) [1] project the follow-up AC204 SAMBA (System for Advanced Mobile Broadband Applications) project [2] focuses on building a trial platform and continues the system studies for future implementations. One of the key issues of the MBS concept is to give mobile users access to the services of the B-ISDN (Broadband Integrated Services Digital Network) while providing the same QoS (Quality of Service) as the fixed network does. Therefore a promising approach is to extend the Asynchronous Transfer Mode (ATM) to the air interface.

After a brief description of the SAMBA trial platform, the paper focuses on the hardware architecture of the protocol processing platform (Control Unit, CU) and gives a short overview of the applied Data Link Control (DLC) protocols developed for Wireless ATM (W-ATM). The chosen approaches, methods and algorithms used for the real time implementation of these protocols with respect to the physical layer (modem) and the CU are described. Due to the user requirements the protocol processing has to cope with bit rates of up to 68 Mbit/s (34 Mbit/s full duplex on up- and downlink).

THE SAMBA TRIAL PLATFORM

The architecture of the Trial Platform is depicted in Fig. 1. It is designed as a digital cellular radio network which can be connected to the fixed ATM network via standardised interfaces. The platform comprises a Base Station Controller (BSC) equipped with two Base Station Transceivers (BST) which cover one radio cell each. An ATM switch connected to the BSC and to an ATM Mobility Server (AMS) provides the mobility management and the interface to the fixed ATM network. For the trial platform, two man-portable mobile terminals are built which can also be mounted in vehicles with velocities up to 50 km/h. The mobile terminal consists of a Mobile Terminal Adapter (MTA) which is connected to standard ATM user terminal. For the demonstrations two user applications are chosen: a wireless video camera and a medical application. Although the physical layer is out of the scope of this document, its basic characteristics are given in Tab. 1. Further details on the can be found in [3]. A comprehensive description of the SAMBA architecture is given in [2]

ARCHITECTURE OF THE CONTROL UNIT

Providing flexible stochastic multiplexing at such high data rates makes protocol execution a very demanding task with respect to processing power and real time requirements. Due to the tight time schedule of the SAMBA project, a comercially available processing platform had to be chosen. The investigations of available products led to the selection of powerPC based CPU boards, controlled by the multiple node real time operating system pSOS+M and interconnected via VME64 bus.

Base Station Controller

The hardware architecture of the trial platform is depicted in **Fehler! Verweisquelle konnte nicht gefunden werden.**. The BSC and BSTs of the trial platform are based on VME64-Bus which provides a standardised bus technology for the interconnection of various boards. In addition, many user I/O pins are available at the backplane which can be used for own purposes. This avoids to carry signals via flat ribbon cable on the front side. BSC and BST are connected with TAXI interface chips which can be used either with optical fiber or with twisted pair copper cable.

The BSC operates three purchased Motorola PowerPC CPU boards. One board for each radio cell and a third board is used for the execution of the mobility management protocols, online monitoring and data sampling for performance evaluation. The 4x4 ATM switch and the standard ATM STM1 line interface were bought from Cellware, Berlin, providing a proprietary interface for carrying ATM cells via free pins on the VME64 backplane. This Interface is called Universal Cellware Interface (UCIF) and is realised as a 8 bit parallel interface with a maximum capacity of approximately 155 Mbit/s. The control functions of the Cellware boards are released and tested using pSOS+(m) on a PowerPC platform as used in the trial platform. Thus, the VME64 backplane, the PowerPC boards with pSOS+(m) and the Cellware ATM boards fit well together and form an ideal environment for the trial platform.

The DLC (data link control) protocol which is being performed by the PowerPC board is rather sophisticated but has nonetheless to be performed in real-time. In order to minimise memory accesses and other time consuming tasks on the PowerPC board and to carry the TAXI interface which is not available on the CPU board, an additional board called ATM cell memory (ACM) has been designed. Its functions are

Reception of ATM cells from the ATM network, storing the ATM cell payload, extracting the ATM header, passing the header to the DLC software on the PowerPC board

Reception of ATM headers together with MAC headers from the PowerPC board, assembling the MAC (Medium Access Control) PDU (Protocol Data Unit) (formerly stored payload), passing it on to the TAXI interface in order to transmit it via the air interface.

Reception of MAC PDUs from the air interface, storing the ATM payload, extraction of the MAC and ATM header, passing them on to the PowerPC board

Reception of ATM headers from the PowerPC board, assembling the ATM cell (formerly stored ATM payload), transmission towards the network.

Basically the same hardware is used in the MTA except that everything is contained in one rack. Therefore, the main focus of this paper is the BSC.

ACM board

The ATM cell memory board is a powerPC based unit which provides fast ATM cell memory, a TAXI interface and an UCIF interface. The board is designed as a standard PMC Card (IEEE P1386.1). The bit serial TAXI interface connects the CU to the BBPU via multimode optical fibres. A peak data rate of about 16.5 Mbytes/s over a distance of up to 2 km can be achieved. The UCIF carries the ATM cells between the fast cell memory and the line interface. The payload of the arriving ATM cells from the air interface or the fixed network is stored on the ACM board. This concept avoids extensive cell copying, as for protocol processing it is sufficient to consider the cell headers and some additional protocol control information. The interworking of the interfaces with the powerPC and the fast cell memory is controlled by a programmable FPGA which provides the required flexibility for the development and integration process. Fig. 4 shows the block diagram and a photograph of the ACM board which is one of the key components of the SAMBA trial platform. To give access to the functions of the ACM board an API (Application Programming Interface) called HAL (Hardware Abstraction Layer) was developed, cf. Fig. 5. The HAL makes the services of the physical layer available to the DLC software and, thus, enables a hierarchically structured design of the software. This supports the distributed development of software components by different partners. Additionally, this structured approach allows the DLC software to access the required operating system functions by using HAL functions. Due to the tight time schedule of the project, software development had to be started in parallel. The structured approach together with an early definition of the HAL interface functions made it easier to fulfil this requirement.

SOFTWARE IMPLEMENTATION APPROACH OF DLC AND ATM LAYER

The protocols and algorithms applied in the DLC layer have been developed and evaluated using an object oriented system simulation tool written in C++. It is based on the CNCL (Communication Networks Class Library) [4], which provides an event driven simulation environment. The protocols, algorithms and utility functions have been implemented with respect to a flexible and versatile usage. The structure has been based on the OSI reference model to enable the fast and easy replacement of whole layers. The run-time efficient execution of the protocols has been a second order optimisation criterion so far. For the implementation of the DLC protocols on the powerPC system the well structured approach of the simulator could only be used as a skeleton since the considered bit rates make fast and efficient protocol execution the main objective which prohibits the use of a strict OSI structure for the software.

Based on experiences gained so far, an object oriented approach for the design and implementation of communication protocols is the most promising solution. Hence this approach was also used for the implementation in C++. One drawback of C++ is the excessive dynamic memory usage which is rather time consuming. Therefore, the object management of the software has been based on a self-developed memory management concept which mostly avoids the dynamic allocation of system memory. The protocol execution consists of time critical data transmission routines

and service routines with lower priority. The obvious approach to use two pSOS+M tasks with different priorities has been evaluated and discarded since the extensive task switching seemed to be too time consuming. Instead only one main tasks is used which is connected to a software interrupt service routine (ASR, Asynchronous Signal Routine). The ASR method avoids task switching and enables the interruption of the service routines by more urgent data transmission functions. The interrupts which trigger the ASR are based on BBPU signals and generated by the system hardware, e.g. ACM board.

DLC AND ATM LAYER

One of the most challenging aspects of W-ATM is the extension of the stochastic multiplexing principle of ATM to the air interface. Therefore the protocol stack is extended by a DLC layer. The approach is to build a distributed ATM-multiplexer around the air interface which includes the base station, the wireless terminals and, most important, the radio channel. For the implementation of this distributed multiplexer, a medium access (MAC) protocol is needed that is able to perform stochastic multiplexing on the radio channel and that coordinates the access of the base station and the terminals to the shared radio resource. The virtual ATM connections operated by the terminals are controlled within the LLC layer which also applies an adaptive ARQ (Automatic Repeat Request) protocol to enhance the reliability of the communication link [5,6].

The Medium Access Control Layer (MAC)

The MAC layer executes the FDD (Frequency Division Duplexing) based protocol version of the Dynamic Slot Assignment (DSA++) protocol family [5] with a fixed period length of 80 MAC-PDUs (Protocol Data Unit) [2]. Each of the MAC-PDUs contains 2 ATM cells and an additional 32bit CI field (Control Information). A MAC connection (Data Channel, DCH) is established for each wireless terminal. A challenging aspect in the implementation of this MAC scheme is the impact of processing delays. Simulation studies carried out with respect to real time implementation [7] have shown that an appropriate arrangement of signalling and data slots allows to cope with these delays, cf. Fig. 8. The structure reflects the processing delay τ_{eval} which is needed to evaluate MAC-PDUs received via the air interface. During τ_{gen} , received uplink signalling messages are decoded to be considered for the next downlink signalling message.

The flexible arrangement of signalling slots reduces the impact of the delays introduced by protocol processing. Fig. 7 shows the usage of the control information field. The DCH Identifier (DCH Id) is a temporary valid short address for the mobile terminal which is assigned during the establishment of the MAC connection. BSC Id and BST Id are used to identify the corresponding base station controller and base station transceiver. The type field describes the content of the cell fields (MAC command PDUs, LLC command PDUs, ATM cells or empty). The urgent bit indicates a message with higher priority. This is used for a preferred processing of the downlink signalling message. The signalling type field determines the usage of the signalling information field. On the downlink, only LLC-acknowledgements are transmitted. On the uplink, LLC-acknowledgements compete with the transmission of capacity request messages (dynamic parameters). These dynamic parameters are different for each service class, cf. Fig. 7. *NIL Dynamic Parameters* is sent if no more capacity is requested. The dynamic parameters for the CBR service class are mainly used for synchronisation since the deterministic arrival times of CBR cells are exploited. However, the optimised handling of CBR connections is not implemented in the demonstrator platform. Capacity requests for CBR connections also use the *VBR Dynamic Parameters*. The LLC acknowledgement consists of a type field which determines the type of the acknowledgement (Receive Ready, Selective Reject or Reject). An ARQ Id used as a short address together with a sequence number which indicates the expected PDU or the missing PDU.

The Logical Link Control Layer (LLC)

The LLC layer contains the ARQ-instances which perform the error recovery and connection management functions. The functions within the LLC layers are based on the processing of single ATM cells and additional protocol control information (sequence number, ARQ identifier and a poll bit). The VCI (Virtual channel identifier) and the VPI (virtual path identifier) are mapped by the LLC layer to an ARQ-Identifier (ARQ Id). The ARQ-Id is used as a short address and reduces the signalling overhead especially for the transmissions of acknowledgements. The LLC layer contains an ARQ-manager which performs the multiplexing of the active virtual channels to the appropriate DCH. The DCH is an object related to one ARQ-instance . It contains the ARQ-sender, the ARQ-receiver and ARQ-evaluation objects which determine the performance figures such as cell loss, delay and throughput for scientific evaluation of the system. The DLC-manager contains the interface to the functions of the MAC and ATM layer, cf. Fig. 6.

The ATM layer

The ATM layer relies on the services of the DLC layer and has to communicate with the mobility management protocols. Since these functions are distributed to three processor boards in the base station, the ATM layer organises the communication between the nodes and provides a common interface for the higher layers to the protocol functions.

Therefore service primitives and control procedures have been developed to enable the common access to the distributed functions within the ATM layer using shared memory communication via the VME bus.

DEVELOPMENT APPROACH AND USED TOOLS

Due to the tight time schedule of the SAMBA project, software development had to be performed in parallel to the hardware development. At the beginning only the powerPC boards and the operating system were available for the software development. Therefore software emulations of key hardware components, mainly of the ACM board and the BBPU, have been developed after the specification of the HAL interface. These emulations, which run on workstations, are based on the CNCL [4] and use IP sockets for the communication with other system parts. Since many people were involved in the implementation and only three target system boards were available, a two-pass development scheme was applied. The first implementation step and the first function tests took place on workstations, considering the special features of the target hardware. After cross compiling the code for the powerPC system in a second step, functional tests of the protocols took place using the self-developed system emulation and graphical protocol debugging tools. After the correct implementation of the protocols had been verified, a first integration step led to a back to back link of two CPU boards via the TAXI interface. A screenshot of the self-developed LLC-protocol debugging tool is given in Fig. 9. It shows states of sending and receiving windows for each virtual connection. As an example, a selective repeat ARQ protocol with a maximum window size of 4 is given. In the diagram, PDU(2) was rejected and PDU(3) has been sent but not yet acknowledged.

ACKNOWLEDGEMENT

The authors would like to thank all project partners for their contributions partially funded by the European Commission ACTS program.

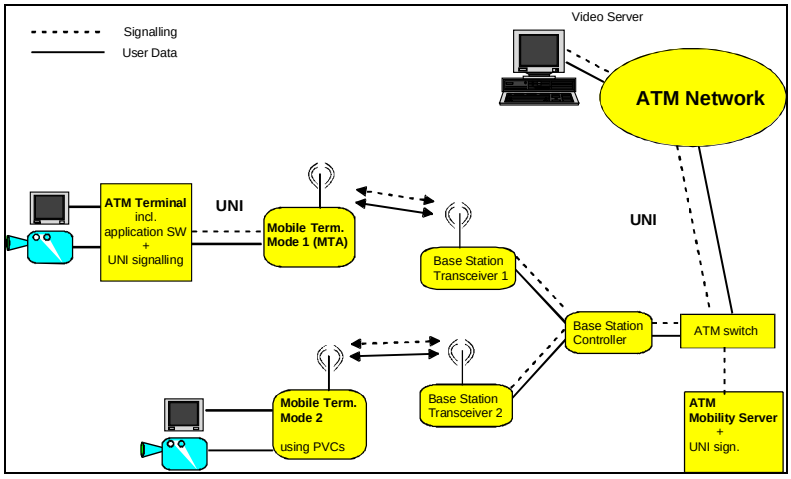


Figure 1: SAMBA trial platform architecture

Item	Parameter/Characteristics
Carrier frequencies	Uplink: 39.58 & 39.74; downlink: 42.58 & 42.74 GHz
Multiple access / duplexing	TDMA / FDD
Modulation	Offset Quarternary Phase Shift Keying (QPSK)
Symbol rate	32 MBd
Reception	Space diversity with max. ratio combining
Equalisation	DFSE-Viterbi, 8 symbols (250 ns)
Forward Error Correction	(130, 110) Reed-Solomon code, 8 bit symbols
Antennas	Three types of dielectric lens antennas for street & vehicle, wide cell & portable terminal

Table 1: Physical layer characteristics

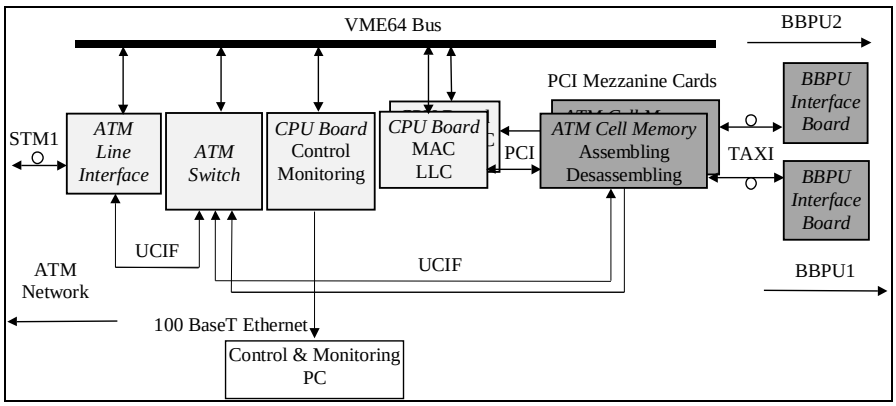


Figure 2: Components of the Base Station Controller (BSC)

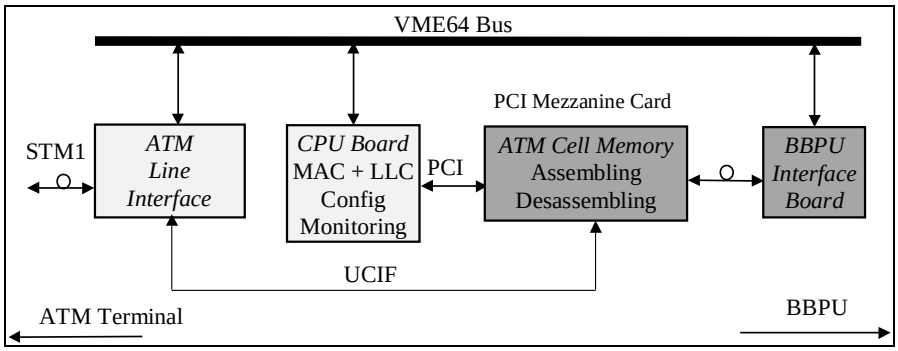
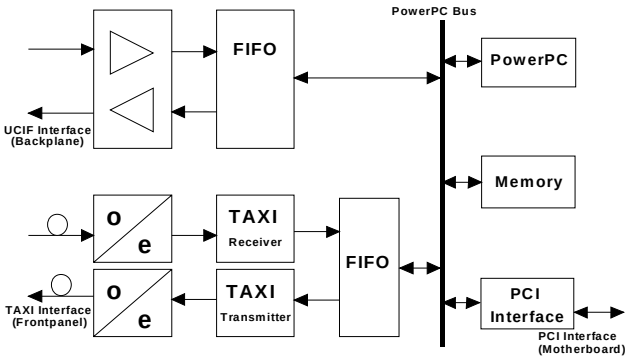


Figure 3: Components of the Mobile Terminal Adapter (MTA)



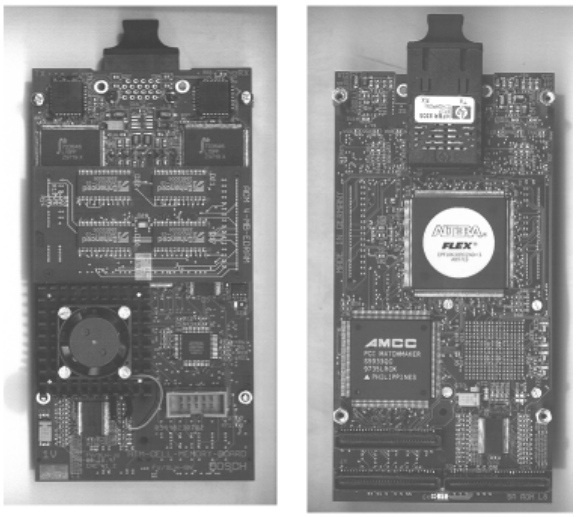


Figure 4: ACM board block diagram and its photograph

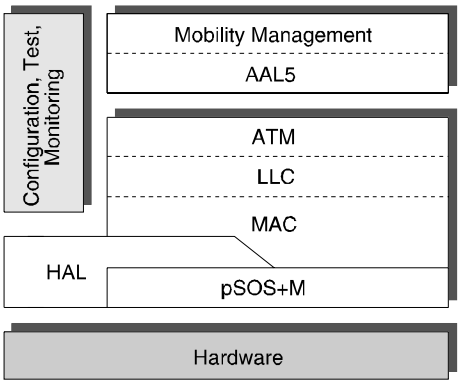


Figure 5: Layered software structure

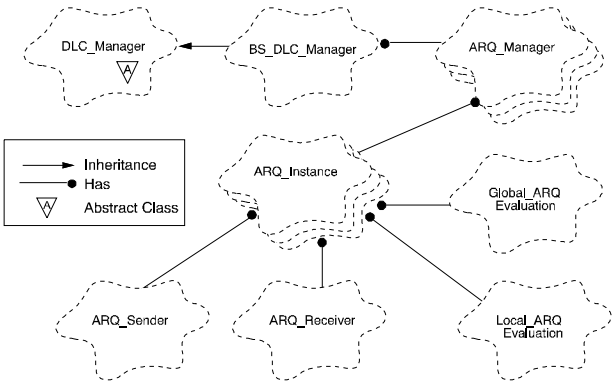


Figure 6: C++ object structure of BS LLC layer

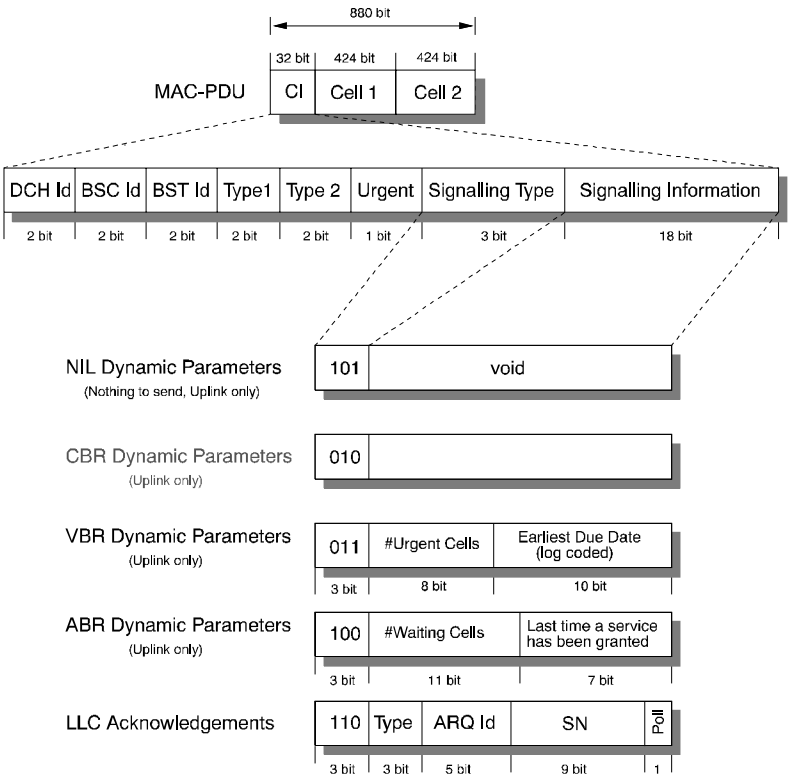


Figure 7: Usage of Control Information (CI) field

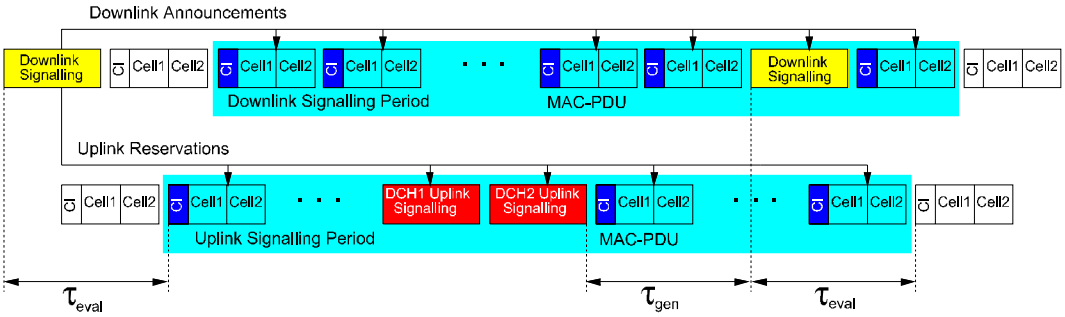


Figure 8:Structure of the DSA++ signalling period with processing delays

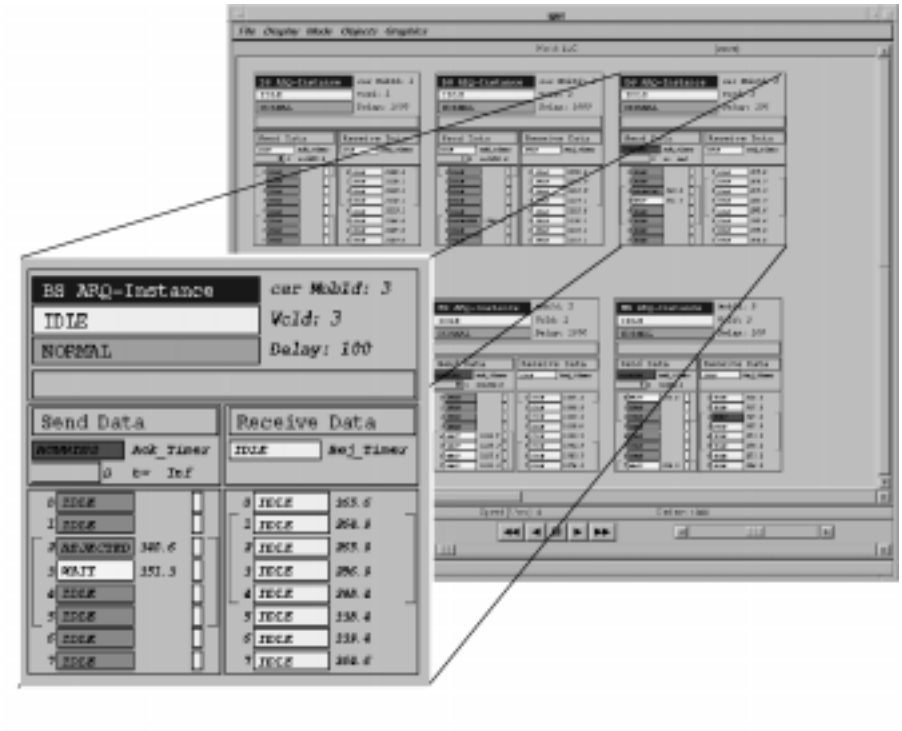


Figure 9: Screenshot of the LLC protocol debugging tool

REFERENCES:

[1] L. Fernandes: *Developing a system concept and technologies for mobile broadband communications*, IEEE Personal Communications Magazine, Vol. 2, No 1, Feb 1995, pp 54-59

[2] M. Dinis, V. Lagarto, M. Prögler, J. Zubrzycki: *SAMBA: a Step to Bring MBS to the People*, Proceedings of the ACTS Mobile Communications Summit, Oct. 1997, Aalborg, Denmark

[3] T. Fujino, M. Miyake, K. Murakami, A. Shibuya, F. Ishizu, H. Kubo, T. Shikama, B. Penther: *Baseband Signal Processing Technologies for 64 Mbit/s Mobile Radio Transmission at 40 GHz*, Proceedings of the ACTS Mobile Communications Summit, Oct. 1997, Aalborg, Denmark

[4] M. Steppler, M. Junius, D. Büter, et al.: *CNCL - Communication Networks Class Library*, ACM Sigcomm '93, Francisco, CA, USA

[5] D. Petras, A. Krämling, A. Hettich, *Design Principles for a MAC Protocol of an ATM Air Interface*, Proceedings of the ACTS Mobile Communications Summit, Nov. 1996, Granada, Spain

[6] D. Petras, U. Vornefeld: *Joint Performance of DSA++ MAC Protocol and SR/D-ARQ Protocol for wirelss ATM under realistic traffic and channel models*, Proceedings of 1st International Workshop on Wireless Mobile ATM Implementation, April 1998, Hanzhou, China

[3] Andreas Krämling, Günter Seidel, Markus Radimirsch, Wolfgang Detlefsen: *Performance Evaluation of MAC schemes for wireless ATM systems with centralised control considering processing delays*, Proceedings of The Second European Personal Mobile Communications Conference, Bonn, Germany, Oct 1997, pp . 173-180