Real-time Demonstration of MPEG-4 based Video Telephony over Wireless Systems using WiNe2[1]

Thomas STOCKHAMMER(1), Hrvoje Jenkac(2), Peter STRASSER (3), Günther LIEBL(4), Daniel PFEIFER(5), Joachim HAGENAUER(6), Tobias HINZ(7), and Thomas WIEGAND(8)

(1) Institute for Communications Engineering (LNT), Munich University of Technology (TUM), D-80290 Munich, Germany, E-mail: stockhammer@ei.tum.de
(2) see (1), E-Mail: jenkac@ei.tum.de
(3) see (1), E-Mail: peter.strasser@gmx.de
(4) see (1), E-Mail: liebl@ei.tum.de
(5) see (1), E-Mail: pfeifer@lnt.ei.tum.de
(6) see (1), E-Mail: hagenauer@ei.tum.de
(7) Image Processing Department, Heinrich-Hertz-Institute (HHI) D-10587 Berlin, Germany, E-Mail: hinz@hhi.de
(8) see (7), E-Mail: wiegand@hhi.de

Abstract

In this work we present a flexible and fast, but yet accurate simulation environment for wireless systems, which satisfies real-time constraints imposed by the respective applications and protocols of interest. We take into account system and demonstration aspects and concentrate on real-time video-conferencing services. The presented wireless network simulation and demonstration environment (WiNe2) includes several audio-visual applications, the corresponding transport protocols, a network-level simulator, and appropriate server and client software and hardware. We will present details in this work and show the suitability of this environment to test, evaluate, and assess video-conferencing applications over GSM GPRS and EGPRS. Video coding is based on MPEG-4, audio coding uses GSM AMR. Appropriate parameters to allow rate adaptation and error resilience are integrated in the multimedia encoding and decoding algorithms. Screenshots and subjective observations are discussed in the paper, while the entire demonstration platform for this specific application will be presented online at the conference.

1 Introduction

Applications like video telephony, video conferencing, multimedia streaming, or multimedia messaging for mobile terminals will be important features in emerging 2.5G, 3G, and future mobile systems and may be a key factor to their success. The video-capable displays and enhanced processing power, as well as additional storage and memory capacity of new mobile devices, pave the road for these new applications. In addition, most current and future cellular networks, like GSM-GPRS, UMTS, or CDMA-2000, contain a variety of packet-oriented transmission modes: The latter allow transporting practically any type of IP-based traffic to and from mobile terminals, thus providing users with a simple and flexible transport interface. However, compared to wireline networks, radio links exhibit some specific properties, which often cause problems in combination with standard network protocols: The residual bit error rate is often non-negligible, and the maximum attainable bit rate per user is significantly lower and time-variant. Furthermore, since multiple users with different applications and channel conditions share the costly resources power and bandwidth, sophisticated resource allocation strategies are required. Therefore, it is important to investigate the impact of cellular links on the end-to-end quality of real-time multimedia applications.

This constitutes the need for a flexible and fast, but yet accurate simulation environment, which satisfies the real-time constraints imposed by the respective applications and protocols of interest. Concepts to realize these needs have previously been presented for the General Packet Radio Service (GPRS) in [1] and for Enhanced GPRS (EGPRS) in [2]. While [1] and [2] focus on the real-time wireless network level simulator part, in [3] the work in [1] has been extended by taking into account the system and demonstration aspects for several IP-based applications with concentration to real-time services over GPRS.

In this work we will first review the wireless network simulation and demonstration environment (WiNe2). We will present concepts and realization details and show the suitability of this environment to test, evaluate, and assess wireless real-time multimedia applications with a focus on MPEG-4 based video-telephony. Next, we extend the work of [3] by including several new components in the modular WiNe2 concept: Among others, the GSM system is upgraded from GPRS to EGPRS, simple audio coding will be replaced by GSM AMR, and Robust Header Compression (RoHC) is integrated in WiNe2. Whereas simple screenshots are used to explain the WiNe2 concepts, the entire demonstration platform will be presented at the conference.

2 The WiNe2 Concept

Multimedia transmission over mobile systems requires huge design, implementation, and realization flexibility on

[1] *WiNe2:* Wireless Network Simulator based on ns-2; Phonetic spelling: [winetu].

each system layer. An optimized design of future mobile systems, transport protocols, or multimedia applications, as well as the combination of those is without any doubt a challenging task: On the one hand, it is important to develop optimized system components, such as highly efficient multimedia coding techniques, optimized transport protocols or highly efficient wireless transmission algorithms. On the other hand, it becomes increasingly important to study the interworking and combination of different techniques on various system layers, as well as to specify and exploit the information flow in networks and especially across layers (usually referred to as cross-layer design). However, this requires a thorough and detailed understanding of existing solutions, especially when it comes to realization or interoperability aspects. Even for already standardized or deployed systems like GSM GPRS or UMTS, a suitable radio link configuration in combination with an appropriate selection of transport and multimedia coding methods for different applications is not straightforward. Finally, objective performance measures of multimedia applications, like mean-square error, average delay, or average consumed bit rate are not sufficient to evaluate the perceived quality seen by a user in different situations.

These rationales, among others, lead to the conception, development, and realization of WiNe2: The integration, evaluation, assessment, and demonstration of real-time multimedia transmission over wireless networks. A Linux-based software solution has been chosen which allows re-using many *open source and freely available components*. This includes, for example, the network simulator ns2 [4] with the real-time extension nse, IP-based interfaces for applications and freely available multimedia conferencing and streaming tools such as vic [5], rat [6], or the Darwin Streaming server [7]. Common APIs and IP-based network interfaces allow extending WiNe2 with almost any application, transport protocol, or mobile system. Thus, it can be used as a development platform, as well as an educational environment for students to carry out demonstration and simulation projects.

The huge amount of different mobile channel parameters and characteristics in combination with mobility, multi-user aspects, data traffic patterns, etc. makes a complete simulation of mobile systems completely infeasible. Therefore, we restrict ourselves to certain *case studies* reflecting typical or worst-case situations in terms of transmission conditions, user topology, and applications. We assume that we investigate the situation in a single cell of a wireless network, where a certain number of mobile terminals are attached to one base station serving the area. The respective users are dynamically establishing and releasing different services. The resulting data flow on the downlink has to be transported on a common resource, a packet-switched transport channel, where the actual type of link sharing is fixed and determined by the characteristics of the cellular network. The central scheduling unit is assumed to be located in the base station, where it is responsible for distributing a certain number of incoming IP-streams from different users, where each user might use multiple applications in parallel. The effects of different parameter settings such as user topology, link quality, traffic characteristics, etc. on the mobile system and the transport protocols are monitored and the consequences of parameter alterations on the application quality are assessed in real-time.

In order to satisfy the real-time constraints, the simulation model is divided into two parts: an offline link-level simulator that collects statistics about the loss characteristics at the physical layer for a huge set of different parameters, and a real-time network-level simulator that allows injecting IP-traffic into a virtual wireless network. The IP-traffic might origin from one or several live users or applications, or from statistical IP traffic generators.

The primary objective of the network-level simulator is to model the various operations that are performed on incoming IP packets at the data link layer of a cellular network. All higher protocol layers, as well as the backbone network of the provider, are assumed error-free and over-provisioned. This constitutes a reasonable assumption for practical systems, and may be changed at any time by adding further protocol stacks to the network-level simulator that reflect the behavior of the access network, if desired. A real-time implementation of the physical layer with appropriate channel modeling in software is completely impossible. However, radio packets with a block check sequence indicating an error are in general declared lost to avoid error propagation through the network. The concept of a so-called *packet erasure channel* has already been introduced to characterize the loss process below the data link layer of cellular networks [8]. The statistics of the loss process are defined by the configuration of a specific mobile system and the long-term attributes of a mobile channel, the so-called operation point (OP). Among others, an OP includes the logical channel structure, channel coding and modulation schemes at the transmitter, demodulation and detection methods at the receiver, and the actual characteristics of the underlying physical transmission channel. The payload size in bits of each link layer packet, as well as the transmission time interval, is usually given by the system, the bearer service, and the coding scheme of interest. The generation of the statistics for each OP is done via an off-line simulator, which usually consists of a software radio simulator for the time-variant wireless channel and a physical layer module. The generated erasure statistics can either be stored in a trace-file, or be used to develop a stochastic description of the loss process, which is, for example, based on higher order Markov models [8].

3 The WiNe2 Realization

The concept chosen to realize the simulation and demonstration platform is based on two end terminals both connected to a PC on which the entire wireless network is simulated. All traffic between these terminals now has to cross the simulator, whose sole task is to delay or drop the injected IP packets according to the characteristics of the respective cellular system. Figure 1 shows the laboratory setup, which allows to demonstrate conversational applications between two terminals, such as video telephony, as well as client-server applications, such as video streaming. The Master-PC on which the WiNe2 simulator is executed has two network interface cards (NICs) with different IP-address spaces to distinguish two separate (private) networks: the client network and the server network. Thus, client and server are only capable to communicate with each other via the Master-PC.

For the network simulation, we have chosen the ns-tool from UC Berkeley [4], as it already contains various common network elements, like agents, queues, or transmission links. Furthermore, it supports almost any type of IP-based traffic, e.g. protocols like TCP or UDP. Even more interesting for us is the fact that since release 2.1b5, the distribution also includes a real-time enhancement, called **nse**: Via the packet capture (PCAP) network interface, it is possible to access the NICs of a regular PC. Thus, incoming IP packets from live users can be read from the interface and injected into a virtual simulation scenario. Correspondingly, any packets that leave the virtual network can be handed over to a different NIC for transmission. Therefore, a PC running *nse* can be viewed as a router that additionally modifies the traffic streams in both directions, which is exactly what we need.

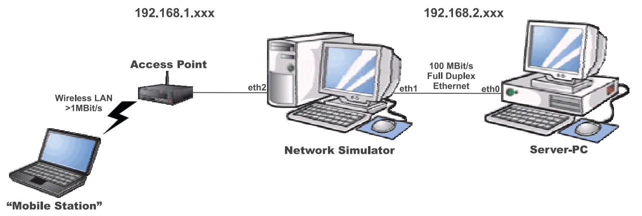

Figure 1 Setup: Multimedia Server, WiNe2 Simulator, Mobile Multimedia Client.

The current realization of the WiNe2 simulator is shown in Figure 2: It supports one live and several virtual users, each possibly with one or more IP-based applications running in both directions. The network-level part of the simulation environment consists of a base station element containing a control unit and several mobile users attached to it. While the mobile users have distinct SNDCP/LLC, RLC, and MAC agents for each application, as well as separate nodes, the base station only has distinct SNDCP/LLC and RLC agents for each flow, while the MAC agent and the node only exist once. This is due to the fact that resource allocation at the base station is done jointly for all incoming flows. On top of each protocol stack resides either a TCP/UDP agent corresponding to virtual sources/sinks, or a tap agent that exchanges live traffic with the network interfaces. Finally, depending on the respective virtual traffic type, a suitable application agent has been placed on top of the TCP/UDP agents. The specific tasks and the implementation of the different agents will be presented in the following.

Tap agents map live network data captured via the PCAP interface onto simulated packets and vice-versa. We have extended the functionality of the tap agents to be able to separate incoming live IP traffic to different transmission links based on, for example, port numbers. The primary function of the *SNDCP/LLC* agent is the adaptation of IP packets from higher layers to the mobile network, e.g. by performing header compression. The *RLC agent* performs the segmentation and reassembly procedures depending on the selected bearer service. All segments resulting from a single SNDCP/LLC-PDU are placed into the data buffer. Furthermore, the RLC agent can operate in either unacknowledged or acknowledged mode, depending on whether the application requires a reliable link layer or not. The *MAC agent* is directly attached to a node in the virtual environment and is responsible for allocating transmission resources to the different users or application flows. The scheduling process is executed periodically to react appropriately to changing channel or flow characteristics, which are reflected by the respective buffer fill level. Since we are mainly interested in the downlink performance of a wireless system, where different medium-rate multimedia streams have to compete for resources, we have integrated three different scheduling strategies [10]: Round-Robin, Ratio-Aware, and a modified version of Class Based Queuing (CBQ).

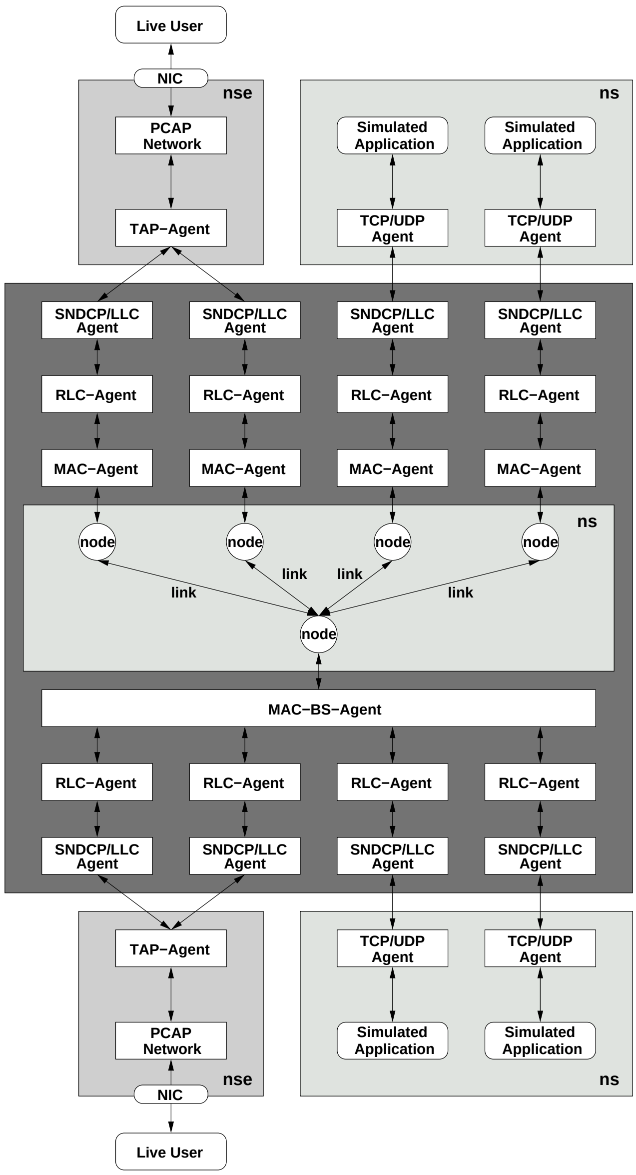

Figure 2 The WiNe2 simulator realization.

Nodes and *Links* are objects which are already available in the original ns2 release. While the first act as a packet-based entrance point to the network simulator, the latter are characterized by a certain bit rate, delay, and loss model. We have extended the link object definition to integrate our packet erasure models derived from offline link layer simulations. The above three attributes (bit rate, delay, and loss model) can be changed dynamically to adapt to variations in the system simulation. Finally, a *control entity* is introduced in the network simulator, which handles the variation of the OPs and passes this information over to the respective agents and loss generators to trigger the appropriate loss models. The operation point can be varied either by drawing a random value from a fixed distribution, especially in case of virtual users, or by polling an external server in case of the live user. The latter is usually connected by WLAN to the network simulator, which allows for mobility, but does not influence the simulated performance (since the achievable WLAN throughput is at least 5-10 times higher than in GPRS or EGPRS).

4 IP-Based Video Telephony over GSM EGPRS

4.1 Audio-Visual Applications

A typical video-conferencing system has to support real-time encoding of video and speech on mobile devices, and must have minimal end-to-end delay for appropriate communication. Thus, it relies on highly efficient video and audio encoding methods, which are in general very sensitive to (bit) errors or lost data parts. Further aspects are bit rate consumption, which usually reflects the costs for mobile users, synchronization between audio and video signals, error-resilient encoding and decoding of audio and video applications, and adaptation to time-varying network conditions. In our current version, we have integrated the GSM AMR codec [11] together with the RTP payload format specification [12] into the rat-tool [6]. In addition, a real-time MPEG-4 encoder and decoder [13] with basic error-resilience tools has been integrated in the vic-tool [5] in combination with an appropriate RTP format [14]. Up to this point, different parameters of the media codecs can be controlled dynamically via external configuration. The encoding parameters include mode selection of the AMR codec, and for the MPEG-4 encoder, among others, frame rate, intra frame distance, random intra-macroblock refresh ratio [15], and maximum packet length adjustment by use of resynchronization markers. Furthermore, we can choose between two different bit rate modes: In CBR mode the bit rate can be specified and is kept constant by a rate control unit. In VBR mode the quantization parameter is kept constant, which results in almost constant quality, but variable bit rate. Obviously, in case of a real system these parameters should be adjusted automatically using control and feedback information, e.g. via RTCP messages, to adapt to varying channel and network conditions. This is subject of current work. In case of errors or lost data, both video and speech decoder have to apply appropriate error concealment. A session tool controls synchronous presentation at the receiver even in case of delay jitter. The main evaluation criterion is the observed subjective quality of different applications or situations. For example, in some situations video with fluent motion is more important than details in the frame, whereas in other situations a high-resolution low frame-rate slide show might be preferable. The human observer can change these parameters by monitoring the costs in terms of used bit rate and received quality.

4.2 GSM Packet Radio Services

The General Packet Radio Service (GPRS) has been designed to operate in most of the current 2G spectrum allocations. It uses an IP-based core network, while the air interface is, for example, based on traditional GSM: Thus, it uses a TDMA-based, packet-switched radio technology with 200 kHz channel spacing and frame structure identical to GSM. Similarly, it involves GMSK modulation and supports transmission rates ranging from 9.05 kbps to 21.4 kbps per channel. GPRS allows the combination of up to all eight time slots per user to support a maximum data rate of 171.2 kbps. However, GPRS has been primarily designed to support best-effort data services, while its enhanced version (EGPRS) will provide integrated services. For sake of simplicity, we will discuss only GPRS in more detail in the following. The additional features of EGPRS are of little relevance when explaining the concepts and will be summarized at the end of this subsection.

GPRS consists of a radio access network and a core network (for details on the network part we refer to [16], [17]). Next, we will present a very brief overview of the protocol stack and the packetization in line with the general setup of our simulator according to the previous subsection. The subnetwork dependent convergence protocol (SNDCP) adapts the upper layer protocols to the functionality of the underlying GRPS layers. It performs segmentation and reassembly of long user data packets and provides means for header compression and data encryption on the mobile link to ensure privacy of user communication.

The data link layer encompasses three sublayers: Logical Link Control (LLC) is used to establish a logical link between MS and SGSN. Optional backward error correction is provided in form of a Go-back-N retransmission protocol. The radio link control (RLC) layer performs segmentation of the LLC packet data units (PDU) into short blocks of fixed length according to one of the channel coding schemes described in the following: In GPRS, four different coding schemes (CS) have been defined, ranging from strong error protection with rate 1/2 convolutional coding (CS-1) to uncoded transmission (CS-4) in case of very good link conditions. However, each of these schemes results in a total of 456 coded bits, since the structure of the underlying interleaver and radio bursts has not been changed compared to traditional GSM. A procedure called link adaptation can be applied to dynamically switch between the coding schemes after every RLC block. This allows adapting the level of error protection to the channel characteristics. Furthermore, before channel encoding a block check sequence is appended, which in combination with sequence numbering allows the detection of erroneous or lost segments. Finally, RLC provides optional retransmission to achieve a reliable data transfer when needed.

The medium access control (MAC) sublayer performs multiplexing of user data and signaling information. The multiplexing on the MAC layer is based on various scheduling strategies to allocate resources to the different users or flows. In our current setup, we consider a single carrier frequency, i.e. there is only one TDMA frame with eight timeslots available. The scheduling process is periodically executed every 2.5ms (which equals the time to transfer one 456 bit radio block, regardless of the used coding scheme). The scheduler in the base station performs joint resource allocation of all incoming flows. As already mentioned, Round-Robin, Ratio-Aware, and a modified version of CBQ [9] can be selected. Furthermore, each of them can be extended to include the actual quality of the link in the scheduling process [10]. Finally, the physical layer consists of two sublayers: The physical link layer (PLL) provides means of forward error correction (FEC), whereas the radio frequency layer (RFL) equals the one specified for GSM.

In EGPRS, the conventional GMSK modulation can be replaced by 8-PSK in case of favorable link conditions, thus increasing the gross bit rate on the air interface by almost a factor of 3. Both channel spacing and frame structure stay the same. Nine additional coding schemes (MCS) have been defined, where MCS-1 to MCS-4 are still used for GMSK (i.e. result in a total of 456 coded bits), but are rate-compatible (in contrast to CS-1 to CS-4) and thus allow for more sophisticated retransmission strategies. MCS-5 to MCS-9 are used for 8-PSK (i.e. result in a total of 1392 coded bits, which are mapped on the different 8-PSK symbols), and are again rate-compatible.

4.3 Wireless Videophone Realization

The proposed demonstration environment can be used in many different ways. In the following we present the wireless network simulator in combination with a video phone application. Figure 3 shows a screenshot of the wireless network simulator and the video received over the uplink at the server. All eight time-slots are available for GPRS and the average carrier-to-interference ratio (C/I) is adapted via feedback of the received WLAN signal strength. Four different users are active within the cell: While the live user runs a video conferencing application, three additional users generate virtual traffic: a conversational speech user, a web user requesting data interactively, and a background user downloading e-mails to his mobile device. Different traffic classes, coding schemes, transmission modes, and long-term C/I distributions have been assigned to each of them. This can be observed from the middle window (*LNT GPRS simulation*), which lists the four users in different colors. The live user (red) in our case has been assigned streaming class, which allows the use of a reliable link layer. The given traffic class assignment prioritizes the virtual conversational user over the live user for CBQ. In addition, this window shows the current coding scheme, link-layer packet error rate, and buffer fill level in units of RLC packets. The small window below (*LNT*) shows the current resource allocation strategy (CBQ in our example). The window to the left (*Long Term Profile Visualization*) contains the C/I used for the loss model of the live user over the last 30 seconds. In addition, the applied coding scheme is depicted in different colors. Obviously, for good channels, a higher channel code rate is selected, whereas for worse conditions the level of error protection is increased at the expense of reduced data transmission rate (this can be easily observed when the live user moves away from the WLAN access point). The upper window (*Slot Assignment*) shows the actual resource allocation (slot assignment) over time, where different users are indicated with different colors: Vertically, the eight GSM time slots are bundled, and horizontally, the allocation over time is plotted. Whereas the upper window contains the last five seconds, the lower part contains the slot assignment history over the last 30 seconds. The window below (*Packet Delay*) shows the packet delay history for each application. The video control window (*vic*) depicts the selectable video coding parameters, and, finally, the video window (*Mobile Multimedia Phone*) contains the received video from the mobile client (see Figure 4).

From the screenshot, as well as from continuous observations, several interesting conclusions can be drawn about the performance of the live user and the system in general for this typical setup. If the coding scheme is selected to maximize the throughput for both acknowledged and unacknowledged mode, the subjective performance is significantly better in case of acknowledged mode despite of the delay due to retransmitted packets: The introduced delay jitter is less annoying than the visual effects on the video in case of errors. However, different coding scheme selection strategies or the introduction of improved error resilience in the video decoder [18] might help to improve the performance of video in unacknowledged mode. In general, the usage of the unacknowledged mode should be considered very carefully even for applications with delay restrictions. Concerning the different scheduling algorithms, it has been observed that especially in heavy-loaded situations CBQ significantly outperforms the simpler strategies as depicted in the packet delay window: Each traffic class is delayed according to its importance. Whereas the conversational application experiences hardly any delay even for high loads, the interactive and background class are delayed significantly. However, after a certain time the available resources are reassigned to these classes and the buffers are emptied.

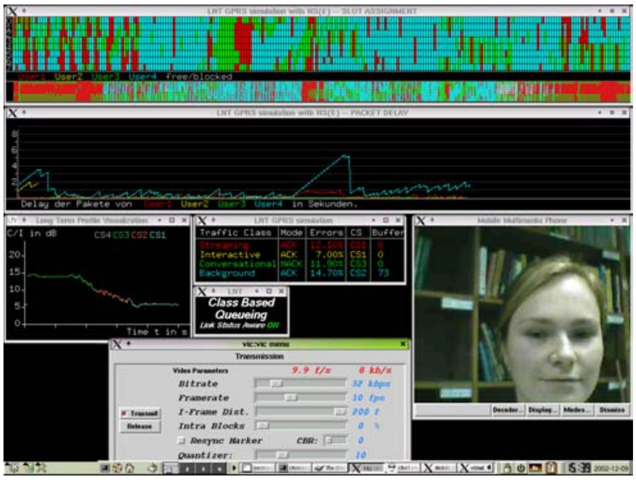

Figure 3 Screenshot of WiNe2 and received video in real-time with 4 users sharing 8 GPRS slots. Slot assignment, packet delay, user statistics, live-user C/I history, scheduling strategy, video coding parameters, and received video.

Concerning the video application, two interesting aspects have been observed: Although a constant bit rate video might be preferable, if a fixed resource allocation is applied, the situation changes in case of dynamic link-sharing. Then variable bit rate applications offer significant benefits: If assigned to the appropriate traffic class, the video application requests more bit rate in case of increased movement or more details in the video sequence, but releases these resources whenever the content is less complex. In the slot assignment window, this behavior is easily visible: The video traffic pattern exhibits bursts, and the background traffic is thus transmitted whenever there is no or little video data. This strategy is not only beneficial in terms of constant quality, but also results in reduced overall system delay. The available resources are thus distributed very efficiently among different users and applications. Finally, it is worth to mention that test users have different opinions of appropriate quality: Whereas in the beginning usually fluent motion is preferred regardless of the quality of each frame, after some time lower frame rates with high quality are often preferred.

Figure 4 The wireless client laptop with integrated camera, microphone, loudspeakers, and real-time video conferencing software based on MPEG-4.

5 Conclusions and Future Work

We have presented a flexible and fast, but accurate simulation environment for wireless systems. This environment satisfies real-time constraints imposed by the respective applications and protocols of interest. We have taken into account both system and demonstration aspects and concentrated ourselves on real-time video-conferencing services. The proposed wireless network simulation and demonstration environment (WiNe2) includes several audio-visual applications, corresponding transport protocols, a network-level simulator, and appropriate server and client software and hardware. We have shown the suitability of this environment to test, evaluate, and assess video-conferencing applications based on MPEG-4 video and GSM AMR audio coding transmitted over GPRS and EGPRS.

Several extensions are already being worked on or are planned for future releases: On the one hand, emerging mobile systems with appropriate link-level simulations and modeling such as UMTS (largely finalized) or HSDPA (planning phase) will be integrated. This requires reconsideration of scheduling and resource allocation strategies. In addition, new applications such as wireless streaming (finalized) or MBMS (planned) can be integrated into the existing testbed. The integration of H.264, MPEG Audio, or any other emerging media codec in our multimedia coding bench is already on the way, especially in combination with suitable extensions for wireless channels according to [1]. The entire protocol stack will be improved in future releases to adapt to varying channel conditions in an appropriate way. Thus, suitable combinations of applications, protocols, and link-layer features will be evaluated.

Acknowledgements

Part of the work has been funded by the German BMBF (Bundesministerium für Bildung und Forschung) and Siemens Corporate Technology within the INVINET project. Parts of the algorithms in the system have been integrated by Markus Kaindl and the students David Liebermann, Florian Reif, and Claudius Blank.

References

[1] G. Liebl, T. Stockhammer, P. Strasser, and D. Liebermann, “A Real-Time Simulation Environment for IP-Traffic over Cellular Links,” in *Proc. 5th World Multi-Conference SCI 2002*, Orlando, FL, USA, July 2002.

[2] M.-J. Lin, H. Luo, and L. F. Chang, “A Linux-based EGPRS Real-time Test Bed Software for Wireless QoS and Differentiated Service Studies,” in *Proc. ICC 2002*, New York, NY, USA, April 2002.

[3] T. Stockhammer, G. Liebl, H. Jenkac, P. Strasser, D. Pfeifer, and J. Hagenauer, “WiNe2 – Wireless Network Demonstration Platform for IP-based Real-Time Multimedia Transmission,” in *Proc. Packet Video Workshop 2003*, Nantes, France, April 2003.

[4] K. Fall and K. Varadhan, The NS Manual, The VINT Project, UC Berkeley, January 2002.

[5] S. McCanne, “vic and RTPv2,” Audio-Video Transport Working Group, IETF 31. December 6, 1994. San Jose, CA, download at http://www-nrg.ee.lbl.gov/vic/

[6] V. Hardman, A. Sasse, M. Handley, and A. Watson, “Reliable Audio for Use over the Internet,” in *Proc. INET'95*, June 1995, Honolulu, Hawaii, download at http://www-mice.cs.ucl.ac.uk/multimedia/software/rat

[7] http://developer.apple.com/darwin/projects/streaming/.

[8] G. Liebl, T. Stockhammer, and F. Burkert, “Modelling and simulation of wireless packet erasure channels,” in *Proc. 10th Virginia Tech/MPRG Symp. on Wireless Personal Communications*, Blacksburg, VA, USA, June 2000.

[9] S. Floyd, and V. Jacobson, “Link-sharing and Resource Management Models for Packet Networks”, IEEE/ACM Trans. Networking vol. 3, no. 4, pp. 365–386, Aug. 1995.

[10] G. Liebl, P. Strasser, and T. Stockhammer, "Dynamic Multiplexing of IP-Streams onto Shared Cellular Links," in *Proc. 11th Virginia TECH/MRPG Symp. on Wireless Personal Communication*, Blacksburg, VA, USA, May 2002.

[11] 3GPP TS 26.090, "Adaptive Multi-Rate (AMR) speech transcoding", version 4.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[12] J. Sjoberg, M. Westerlund, A. Lakaniemi, and Q. Xie, “Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs”, RFC3267, June 2002.

[13] ISO/IEC JTC1, “Generic Coding of Audiovisual Objects – Part 2: Visual (MPEG-4 Visual)”, ISO/IEC 14496-2, Vers. 1: Jan. 1999, Vers. 2: Jan. 2000; Vers. 3: Jan. 2001.

[14] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, “RTP Payload Format for MPEG-4 Audio/Visual Streams”, RFC3016, Nov. 2000.

[15] Q. F. Zhu and L. Kerofsky, “Joint Source Coding, Transport Processing, and Error Concealment for H.323-Based Packet Video,” in *Proc. SPIE VCIP*, vol. 3653, pp. 52–62, Jan. 1999.

[16] G. Brasche et. al., “Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service,” IEEE Commun. Magazine, August 1997.

[17] 3GPP TS 23.060 v.3.1.0, Digital Cellular Telecommunication System (Phase 2+); General Packet Radio Service (GPRS); Service Description; Stage 2, October 1999.

[18] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, “H.264/AVC in Wireless Environments,” IEEE Transactions on Circuits and Systems, vol. 13, no. 7, July 2003.