A SYSTEM FOR QOS-ENABLED MPEG-4 VIDEO TRANSMISSION OVER BLUETOOTH FOR MOBILE APPLICATIONS

Corina Scheiter, Rainer Steffen, Markus Zeller, Rudi Knorr
Fraunhofer Institute for Communication Systems (ESK), Munich
Email: Corina.Scheiter@esk.fhg.de

Benno Stabernack, Kai-Immo Wels
Fraunhofer Institut für Nachrichtentechnik
Heinrich-Hertz-Institut, Berlin
Email: stabernack@hhi.de

ABSTRACT

Streaming media distribution is a rapidly growing application. The increasing number of broadband internet subscribers, the growing market for smart in-home networks and the upcoming broadband mobile phone networks encourage this trend. In addition, the number of multimedia enabled mobile devices such as PDAs, smart embedded devices and notebooks is rapidly growing. MPEG-4 is one of the favorite video coding schemes, and Bluetooth is a preferred wireless technology for small and smart devices because of its small size, low power consumption and cost efficiency. In this paper we present a complete Quality of Service (QoS) enabled system that integrates a MPEG-4 Codec and a Bluetooth device for mobile video applications.

1. INTRODUCTION

To meet the requirements of future mobile multimedia applications, we need to achieve a complete solution to get a multimedia-enabled embedded system. This solution needs to be small, flexible, inexpensive and has to be suitable for various applications.

The suggested solution consists of a MPEG-4 encoder module, a Bluetooth device and an ARM processor that hosts the software for the media server, the MPEG-4 decoder, the middleware and applications (see figure 1). The overall design is modular, so that the Bluetooth component can easily be replaced by another transmission technology (e.g. IEEE 802.11) without changing the other parts.

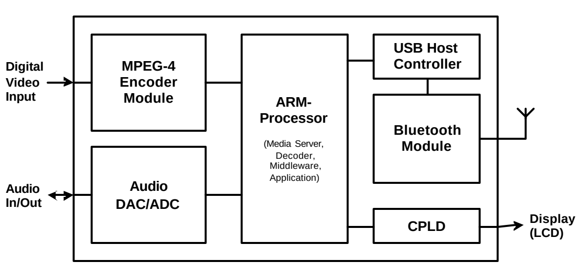

Figure 1: Block diagram of the described system

The remainder of the paper is organized as follows. In section 2 we briefly discuss the requirements and conditions for MPEG-4 Video transmissions over Bluetooth. In section 3 we describe the architecture for the MPEG-4 Encoder Module and the basic functionality of the media streaming process that is used to realize the overall system. Section 4 describes the framework for transmitting IP video streams over Bluetooth, the QoS enhancements and the functionality and interaction of the middleware. Finally, in section 5 we present a conclusion.

2. REQUIREMENTS AND CONDITIONS

Bluetooth is a low cost, small size communication technology that enables ad-hoc interconnection of different end-user and network devices. It allows a short range communication of up to 100m with a gross data rate of 1 Mbit/s. The transmission of Video over a Bluetooth network requires the integration of video applications, codecs and the Bluetooth network technologies. The QoS requirements of MPEG-4 must be matched to the restrictions of Bluetooth so that a reasonable transmission quality is possible without overloading the network.

The requirements of MPEG-4 video transmission depend on the desired video quality. The Media-Server can deliver streams with different resolutions, frame rates and data rates. It is up to the application and the user to choose the appropriate parameters. The use of Bluetooth restricts the range of these parameters. Bluetooth just has a limited bandwidth that must be shared within a piconet. The available data rate mainly depends on the number of Bluetooth devices, the used packet types and the signal to noise ratio (SNR) of the links. Figure 2 shows the maximum data rates that could be achieved between two Bluetooth devices with different bit error probabilities. The different packet types are called DM1, DM3, DM5 and DH1, DH3, DH5, whereas DMx packets contain a 2/3 forward error correction scheme (FEC) and DHx packets have no FEC.

In addition to the technical conditions given by Bluetooth and the video codec, there are user requirements which must be considered. One of them is the simplicity of the service utilization. For user convenience, the configuration of the system should run automatically and in an ad-hoc manner. The user could be unaware of the network address, the location of the service, the available bandwidth and the QoS-Parameters. He can discover and access all services in the same way, without knowing the network specific details.

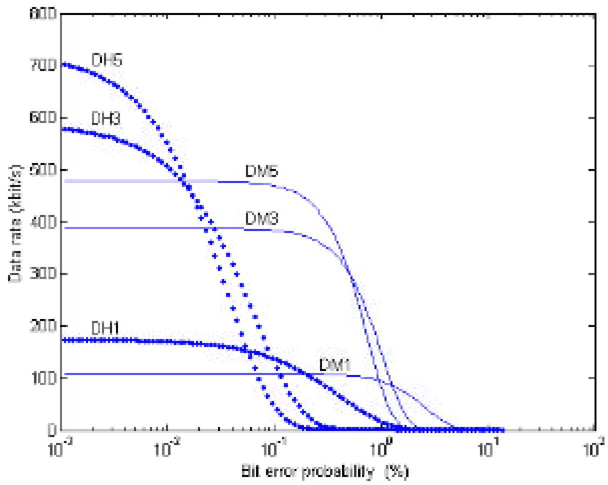

Figure 2: Data rates of Bluetooth at different BER

3. MPEG-4 CODEC

The MPEG-4 standard [1] has been created by the Moving Picture Experts Group (MPEG) to address a wide range of multimedia data compression applications. Compared to previously defined coding standards like MPEG-1, MPEG-2 or H.263, it offers a larger amount of tool sets, each of them addressing a specific application field. MPEG-4 defines compression schemes for audio and video streams for a wide range of channel data rates. Therefore it has also been adopted for the next generation of visual communication systems. The variety of implementations for an MPEG-4 system ranges from pure software solutions to application specific processor systems [2], [3]. Pure software solutions require a high processing power and are therefore mainly restricted to run on PCs. Application specific processors allow a hardware efficient implementation and therefore are the choice for compact and power sensitive solutions as required e.g. for mobile applications.

In the following, a system shall be considered which is composed of a standard processor extended by dedicated hardware parts. The main advantage of such a system concept is the high performance and low power consumption.

3.1 Hardware-Software Partitioning

Based on intensive profiling runs using an optimized MPEG-4 video codec software for the targeted ARM processor core, the hardware-software partitioning was performed. The whole software has been implemented from scratch using C without any processor specific optimization which leads to a well known software base. The software supports MPEG-4 simple profile @ level 0 as well as levels 1, 2, 3 which differ slightly from each other in the tools used.

Based on the specifications of the target application, the highest supported bit rate of 128kBit/s for QCIF resolutions with frame rates of up to 15 frames/sec. was chosen to perform the profiling runs. Based on this analysis, the deblocking and deringing filter, the color conversion and image scaling have been chosen as dedicated hardware blocks in the decoder part. On the encoding side the motion estimation was identified as a candidate for a dedicated hardware implementation due to the average needed processing power of approximately 47 percent of the whole encoding process. In addition, hardware modules for input scaling and color conversion are used.

Besides the analysis of the needed processing power and the hw/sw partitioning, a detailed memory access analysis was performed to evaluate the access characteristic of the codec and the needed memory sizes for the different supported MPEG-4 coding levels. For all simulations the ARM simulation environment Armulator has been used with ARM922 as the target processor platform. In figure 3 the overall architecture of the complete MPEG-4 Simple Profile video codec is shown. Based on the HW/SW partitioning, parts of the video codec algorithm are mapped onto a programmable processor; others are mapped to dedicated hardware modules.

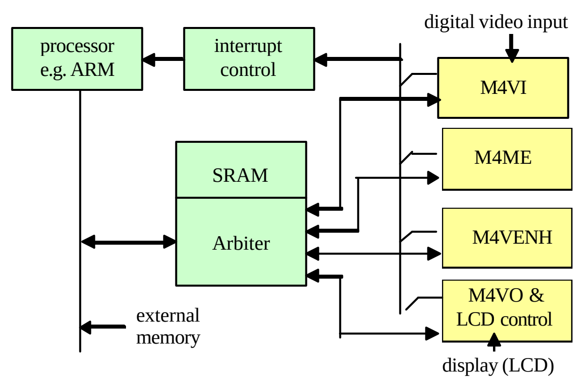

Figure 3: Block Diagram MPEG-4 Video Codec

The video input module (M4VI) comprises a camera interface and in the general case also scalers. The video output unit provides picture scaling and color conversion, an on-screen display (OSD) unit and an LCD interface. A motion estimation module (M4ME) can be used to avoid the computational intensive calculation of motion vectors by the processor. A similar option holds for the post filter task. For the implementation of this task, a video enhancer module (M4VENH) is provided. The data flow is controlled by interrupts, which are processed by an interrupt control unit. A common SRAM is used for the exchange of data between the processor and the different hardware modules.

Using the software simulation model which was implemented for profiling and simulation purposes, the complete software codec was derived. To achieve real-time performance on the targeted processor platform, most parts of the codec had to be optimized. The codec was optimized in terms of needed computational power, needed memory space and number of memory accesses / memory bandwidth. All software optimizations could be realized using C-code instead of hard-to-maintain assembly language constructs. For real-time optimizations, well known optimizations have been applied like loop unrolling, using shift operations instead of multiplication where applicable and avoidance of division due to the fact that the ARM instruction set does not provide integer divisions. Using the above-mentioned techniques, the overall performance of decoder and encoder has been doubled in terms of processing speed.

To minimize the needed memory space of the codec, special care was taken to reduce the memory size for the prediction buffers in the encoder. Based on the assumption that only motion vectors with a limited range are used, the prediction buffer has been reduced to four macro block lines. The implementation of special software modules for border padding, which is needed for the processing of MPEG-4 unrestricted motion vectors, allowed us to store only the needed frame buffers without a padding border.

To reduce the needed memory space for variable length decoding tables, in most cases a multi-step approach for VLD processing was chosen. Instead of storing the complete address range of a Hufman table, the table was split into several significant smaller tables, which represent the complete table. The drawback of this approach is a higher computational complexity.

To reduce the memory bandwidth, the code was optimized using the achieved profiling runs performed with the ARM Armulator simulation environment. The results showed that the use of data elements stored in structures which reside in the main memory, instead of using local variables, leads to a much higher memory bandwidth than needed. The whole software has been investigated to avoid multi memory accesses to the same data element, instead of using a single local variable.

For encoding and decoding, the needed memory size is about 24 Kbytes of data memory and 30 Kbytes of program memory not including the needed frame memory, which depends on the used picture resolution.

To interface the software modules with the corresponding hardware components, the software has been implemented to support a macro block line oriented processing; that means that the synchronization of the several processing tasks, e.g. motion estimation or video output processing, is performed on macro block line level.

A more detailed description of system components is provided in the MPEG-4 codec and corresponding hw/sw module data sheets [4].

3.2 Video Streaming

The streaming framework for the network transport of MPEG4-data provides support for both real-time encoding and decoding of MPEG-4 media to fulfil demands of live transmission.

The incorporated signalling scheme is based on RTSP, which uses TCP/IP as the transport protocol. To fulfil the requirements of media streaming, a request-response scheme in terms of server-client communication is implemented. The communication on the signalling channel is established using a small set of commands. A typical streaming session can be divided into three phases. These are: negotiation, streaming and shutdown.

The media transmission is achieved using RTP, which is based on the UDP datagram protocol. For each media channel, one RTP channel is used. To avoid unnecessary data packet overhead, the data is packed without using the MPEG-4 SL packetization scheme. Synchronization is performed using RTP timestamps, which are already defined in the RTP specification. The framework which is currently implemented comprises a server for data retrieval as well as for real time media encoding. The players are available for different platforms such as Win2k and Linux.

4. MPEG-4 TRANSMISSION OVER BLUETOOTH

4.1 Bluetooth QoS Enhancements

There are different ways to transmit the RTP packets from the codec as IP packets over a Bluetooth network. The most promising solution is to use the Bluetooth Network Encapsulation Protocol (BNEP) as shown in figure 4. The BNEP interface behaves like a conventional network interface (e.g. Ethernet), so that the higher layers (IP, TCP, UDP) could easily be attached. The IP packets are supplemented by a BNEP Header (at least 1 Byte) and handed down to the L2CAP layer. This method enables a flexible transmission of IP data with low overhead and low delay.

For a stable real-time video transmission, it is important to provide a guaranteed QoS support. The specified and implemented QoS mechanisms of Bluetooth are not always sufficient, especially when there are more video streams to be transmitted at the same time so that the applications compete for the bandwidth. Our solution contains a resource reservation add-on above the HCI-Interface. The higher layers can request a specific data rate by an additional resource manager that controls the access to the network (admission control). The master can assign data rates to different Bluetooth devices by changing the poll interval of the connection. Because of the mobility and the dynamic behavior of wireless devices, the resource manager continuously monitors the actual state of the network (reservations, traffic load, bit error rate etc). In case of too poor link quality, the resource manager informs the higher layers about the changes. The higher layers could then adjust the MPEG-4 Codec settings or decrease the data rate of other applications.

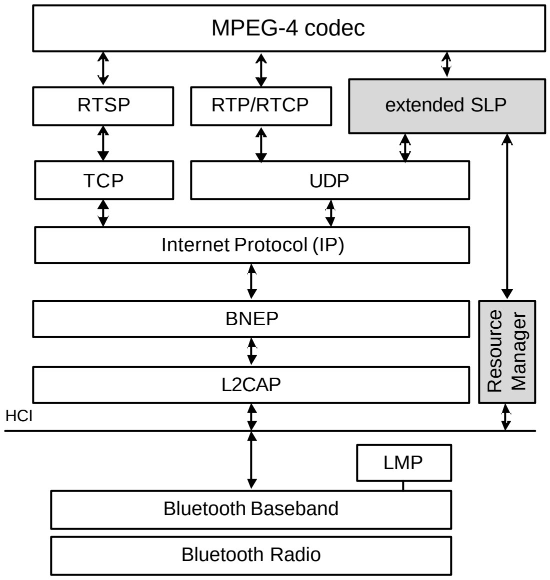

Figure 4: Protocol stack for MPEG-4 over Bluetooth

4.2 Middleware with QoS

Middleware technologies like the service location protocol from the IETF (SLP) generally form a homogeneous distributed system out of different underlying network technologies. From the middleware point of view, there are only service providers and service users, which communicate via a fixed command set for registering, browsing and searching services without knowing the exact location (IP address and port number) of the service.

For an overall QoS-enabled video system, it is necessary to combine the QoS requirements of the video codec and the possibilities of the Bluetooth network. As the middleware is located between the network layers and the MPEG-4 codec, it is the suitable component for the coordination of the QoS handling. Only in this way can the user negotiate and use his services, being sure that the desired transmission quality can be delivered: after finding a service (e.g. a camera) and its QoS requirements over SLP, the user chooses the transmission parameters. The middleware maps these parameters to the Bluetooth-specific QoS parameters and initiates the verification of the connection parameters and the Bluetooth resource reservation. If it succeeds, it starts the player application for the user and data transmission. If the reservation fails, the middleware asks the user to renegotiate the service parameters.

Therefore, the middleware communicates with user applications, services and the Bluetooth resource manager. Its tasks are:

- controlling the global access to the system (admission control)
- enabling the negotiation of QoS parameters between service provider and user
- initiating reservation of negotiated parameters at the Bluetooth network.

To do this, the middleware uses extended mechanisms and interfaces. The service description contains the QoS requirements of a service. This information is exchanged with the user for registering, searching and using services. Via the communication interface with the Bluetooth resource manager, the middleware receives information about the actual network state and can reserve transmission parameters. These Bluetooth and middleware extensions enable the user to reserve a specific quality for his transmission and enable the system to manage existing resources efficiently.

5. CONCLUSION

Multimedia transmission over wireless networks is an emerging area for new networking scenarios such as mobile video on demand, video conferencing or security systems: wireless linking of surveillance cameras and mobile devices offers the security staff the possibility of walking around the building and viewing video sequences e.g. from the nearest or any desired camera. The devices are automatically connected to cameras and the user has simple and uniform access to all services. The services deliver video streams with dedicated transmission requirements; the user has quality demands which should be fulfilled but, on the other hand, should not overload the network.

Our solution for such scenarios consists of an optimized video codec for mobile application and QoS extensions for Bluetooth and SLP. All system components work together to handle QoS for the data transport. The advantages of such a system are the transmission of MPEG-4 Video streams with adequate quality and low bit rates over a Bluetooth network, the possibility of reserving a specific transmission quality, efficient resource management and load control, as well as simple and user friendly system access and ad hoc networking.

6. REFERENCES

[1] ISO/IEC 14496-2, "Coding of audio-visual objects", part 2: visual, Dec. 1999.

[2] H.-J. Stolberg et.al., "The M-PIRE MPEG-4 Codec DSP and its macroblock engine", Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), May 2000.

[3] D. Petrescu, "Efficient implementation of video post-processing algorithms on the BOPS parallel architecture", Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), May 2001.

[4] B. Stabernack et.al., “A processor based system on a chip design for mobile multi media applications”, IEEE Int. Symp. on Consumer Electronics (ISCE 02), TU-Ilmenau, Sep. 2002.