Distributed Multimedia System Design: A Holistic Perspective

Radu Marculescu
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213-3890
radum@cmu.edu

Massoud Pedram
Department of EE-Systems
University of Southern California
Los Angeles, CA 90089-2562
pedram@ceng.usc.edu

Joerg Henkel
C&C Research Labs
NEC USA
Princeton, NJ 08540
henkel@nec-labs.com

Abstract

Multimedia systems play a central part in many human activities. Due to the significant advances in the VLSI technology, there is an increasing demand for portable multimedia appliances capable of handling advanced algorithms required in all forms of communication. Over the years, we have witnessed a steady move from stand-alone (or desktop) multimedia to deeply distributed multimedia systems. Whereas desktop-based systems are mainly optimized based on the performance constraints, power consumption is the key design constraint for multimedia devices that draw their energy from batteries. The overall goal of successful design is then to find the best mapping of the target multimedia application onto the architectural resources, while satisfying an imposed set of design constraints (e.g. minimum power dissipation, maximum performance) and specified QoS metrics (e.g. end-to-end latency, jitter, loss rate) which directly impact the media quality. This paper addresses a few fundamental issues that make the design process particularly challenging and offers a holistic perspective towards a coherent design methodology.

1. Introduction

The continuous increase in the demand for portable appliances capable of handling advanced multimedia algorithms required in all forms of communication (text, graphics, audio, video) requires a fundamental change in the way we think and design such systems. Indeed, as we move deeper into the DSM technology, the system-level design issues become more important and a re-evaluation of the real choices at designer's hand becomes mandatory.

From an implementation standpoint, there is an increasing demand for complex systems that can be integrated on the same chip. Yet, the pressure of tighter time-to-market deadlines continues to make the design process more difficult. Without significant changes in the design methodology, the designers will be able to exploit less and less from the potential the new technologies have to offer [1]. Consequently, emerging design platforms consisting of hardware and software resources that can be shared across multiple multimedia applications, are currently considered as being a promising solution [2]. Such generic design platforms consist of fixed processing resources (e.g. ASICs) and programmable resources (e.g. general-purpose or DSP processors) that can co-operate and run the target application (e.g. MPEG-2 audio/video decoder, e-mail, web browsing, etc.).

Over the years, we have witnessed a constant move from stand-alone (or desktop) multimedia to deeply distributed multimedia systems. Most notably, the transition from desktop multimedia to portable multimedia based on heterogeneous design platforms brings concurrency and communication as prime candidates for system-level analysis and optimization. As such, the design issues change significantly: Whereas desktop-based systems are mainly optimized based on performance constraints, power consumption becomes the key design constraint for multimedia devices that draw their energy from batteries. Consequently, it is crucial to maximize the operating time between two recharge cycles and thus efficiently use the available amount of energy. Examples are mobile personal multimedia systems like MP3 players, Personal Digital Assistants (PDAs) with built-in cameras and gaming features, cell phones, but also non-personal devices like sensor networks.

Other design concerns are generated by *i*) the large number of multimedia systems that need to provide services relying on the energy provided by a battery of limited weight and size, *ii*) the limitation on computational capability of multimedia systems because of heat dissipation issues, and *iii*) the dependability of multimedia systems operating at high temperatures because of excessive power dissipation. Last but not least, the designing and manufacturing costs are increasingly important since many of the multimedia devices have to be affordable in order to fulfill their prospective area of deployment (as an example, think of a sensor network where hundreds or even thousands of computation nodes are needed; plus, the entire sensor network may have a limited live span of only a few days). The design time also needs to be kept very low in order to keep pace with market trends. Cell phones, for instance, experiences two major product cycles a year compared to only one some years ago.

Consequently, the researchers in embedded systems, real-time and networking communities have to work synergistically to design efficient distributed multimedia systems. To this end, we believe that the design of complex networked multimedia systems should be, at the same time, node- and network-centric with emphasis on low-power. This paper discusses the main issues in distributed multimedia systems design and points out interesting research venues in the light of emerging trends in multimedia.

2. Basic issues in multimedia systems design

Multimedia systems represent a very special class complex computing systems [3]. As such, their design process should start by taking into consideration their unique characteristics which are dominated by the huge amount of data that needs to be processed and transmitted in a continuous manner, and the timing constraints that need to be satisfied in order to have an informational message meaningful to the end-user. Another important characteristic is the Quality of Service (QoS) which embraces all the non-functional properties of a system (e.g. power consumption, latency, jitter, cost, etc.). In multimedia systems, QoS requirements vary considerably from one media type to another. For example, due to the large amount of data that needs to be processed, the video streams require consistently high throughput, but can tolerate reasonable levels of jitter and packet errors. In contrast, the audio applications manipulate a much smaller volume of data (therefore do not require such a high bandwidth), but place tighter constraints on jitter and error rates.

Simply speaking, designing a multimedia system consists of mapping the target application, onto a given implementation architecture, while satisfying a prescribed set of design constraints (e.g. power, performance, cost, etc.). From a practical standpoint, finding the "best match" between application and architecture implies

several actions (e.g. providing enough buffering space, choosing the appropriate scheduling technique, etc.) which have a direct impact on the end-to-end latency, power consumption, achievable throughput rate, etc. We also note that, such metrics are closer to the average behavior rather than the worst-case performance of these systems. Indeed, the computational requirements of multimedia systems show such a large statistical variation that designing them based on the worst-case behavior (typically, orders of magnitude larger than the actual execution time [4]) would result in completely inefficient systems.

2.1 Modeling issues

As in most practical cases, the design of multimedia systems starts with the modeling step of the multimedia *application*. In its most abstract form, a multimedia application can be reduced to a set of different media streams (audio, video, etc. coming from the same or different sources of information) that satisfy a particular temporal relationship. For instance, in order to enforce lip-synchronization, the audio and video streams needs to be synchronized at precise time instances [5]. With respect to this temporal relationship, multimedia applications are characterized by 'soft' rather than hard real-time constraints and then they may tolerate a small percentage of missed deadlines. In other words, the behavior of multimedia applications is not necessarily characterized by a single hard real-time constraint, as is the case for safety critical applications, but by a probability (or distribution of probabilities) which captures some variability in the performance metrics.

To model the application of interest, we need to think about representing streams of information. A natural choice is to use process graphs where each node corresponds to a *process* in the multimedia application, while each edge represents a communication *channel* (link) which allows data to be exchanged (usually asynchronously) between different communicating processes. This communication process happens through dedicated buffers that behave like finite-length queues.

As for the abstraction itself, a multimedia stream consists of the *Source* (e.g. encoder), the *Sink* (decoder), and the *Channel* (lossy or lossless) as shown in Fig.1(a). The functionality of each component is obvious: the *Source* generates a continuous sequence of packets which are relayed by the *Channel* to the *Sink*, which later displays it at a certain rate. As shown in Fig.1(a), the real channel can be modelled as an automaton which simply transmits packets from the *transmitter* (*Tx*) to the *receiver* (*Rx*) buffers. The packets may be sent over the channel with error, or may be simply lost during transmission. If the packets are transmitted successfully, then they are stored into the *Rx-buffer*.

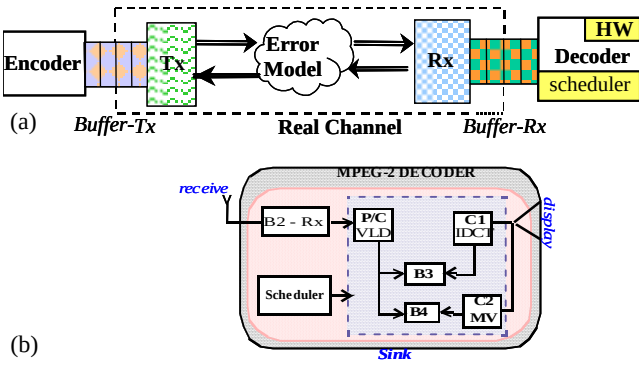

Fig.1: A generic multimedia stream and a decoder example

Whether or not the *Channel* in Fig.1(a) models a wired or a wireless scenario, simply depends on whether or not one targets an on-chip or off-chip implementation. In any case, the *Producer-Consumer* paradigm can be successfully applied to model both situations. For instance, for the generic MPEG-2 video decoder in Fig.1(b), applying the *Producer-Consumer* paradigm locally, implies explicit modeling of the data exchange between the *Producer* (*VLD*) and *Consumer* processes (*IDCT/MV*) which happens through the buffers *B3* and *B4*. The average length of these buffers is very important as it reflects their utilization over time. On the other hand, applying *Producer-Consumer* from a network perspective, implies explicit modeling of processes that have both computation and communication states. This is important since, in order to identify the best trade-off between power and performance, one must take into consideration the entire environment (i.e. source, sink, and communication channel) for which the system is being designed. By doing so, one can decide, at the highest level of abstraction, the best rate for the source, how much retransmission can be afforded, etc. [6].

Once the model of the multimedia application is built, we need to model the *architecture* as well. System-level architecture modeling shares many ideas with application modeling that was just discussed. Having the application and the architecture models, the next step is to map the application onto architecture and then evaluate the model using either simulation or some analytical approach. Mapping for instance the simple VLD-IDCT/MV processes in Fig.1(b) onto a platform with a single CPU, would imply another process, namely the *scheduler*. This process determines the sequence in which the various concurrent processes of the application run on the different architectural components, particularly if the resource is shared.

Once the entire model is built, finding out the steady-state behavior of the complete system is a very useful measure from the design standpoint since it is related to the overall systems performance. Indeed, once the steady-state probability distribution is determined, different performance measures such as throughput, response time, power consumption, etc. can be easily derived.

2.2 Analysis

Generally speaking, the steady-state behavior of a multimedia system can be estimated using explicit simulation or analytical methods. Due to its conceptual simplicity, *simulation* is the method of choice in most practical situations. The only problem with simulating multimedia is the huge volume of data that is typically needed to simulate to gather relevant statistics for the average-case behavior. Considering that a few minutes of compressed MPEG-2 video can easily require a few Gbytes of input data to simulate, the advantage of having available analytical tools that can quickly derive power/performance estimates becomes evident.

Analytical approaches, on the other hand, rely on theoretical assumptions (for instance, exponentially distributed arrival times) that are needed in order to make the analysis tractable in terms of memory and run-time requirements. The objective of any analysis technique is the computation of the stationary probability distribution for a distributed system consisting of several processes that operate and interact concurrently [7]. Depending on the semantics, several approaches have been proposed for the analysis of multimedia applications [8-11]. Although significant progress has been made, the development of well founded mathematical approaches often lags behind more practical, less rigorous approaches like simulation or prototyping. One of the fundamental issues that contributes to this discrepancy is the difficulty of modelling and incorporating real time into formal analysis. Indeed, although timed extensions for most modern formalisms have been proposed (e.g. Petri Nets, process algebras) [5], they suffer from excessive complexity and their application to solving real examples remains problematic at best.

3. Micro-architectural design: The node-centric perspective

The transition to network-centric multimedia systems has a significant impact on the micro-architecture as well. From the micro-architectural point of view, there are various possible design scenarios. One is the use of a GPP (General Purpose Processor) enhanced with a multimedia-specific instruction set a la Intel's MMX technology. This option has proven to be very powerful in desktop multimedia applications. However, because of the large base processor core and the extensive list of multimedia-specific instructions, this option represents an over-design for power-conscious, light-weight multimedia systems that may be very specialized and then only a small fraction of the whole MMX instruction set may actually be utilized.

On the other hand, there is the option of going for an ASIC design that can be tailored to the multimedia application without any unnecessary overhead. In addition, an ASIC design will result in an unsurpassed performance-per-power ratio. However, due to the high costs associated with semiconductor processes at 130nm and below, standard cell designs seem to become economically uninteresting unless large quantities (depending on the design scenario more than a million units might be required to amortize development and mask NRE costs) are expected. In addition, ASIC designs have prohibitive long design times and their re-usability across various multimedia platforms is quite limited.

Application Specific Instruction-set Processors (ASIPs), as the core component of multimedia systems, represent a very efficient option with respect to performance-per-power ratio, design costs/time, manufacturing costs, flexibility (programmability), re-usability etc. In recent years, the so-called extensible processor platforms, as the state-of-the-art in ASIP design, have evolved and are offered nowadays by multiple commercial vendors [12-17]. In what follows, we address a few issues specific to these processors.

3.1 Extensible processors for multimedia

The customization of an extensible processors for multimedia applications encompasses typically the following three levels of instruction extension, inclusion/exclusion of predefined blocks and parameterization.

a) *Instruction Extension*: The designer has the choice to freely define highly customized multimedia instructions by describing their functionality in a high-level language. This description is then used by the platform's design flow as an input and subsequent synthesis steps will generate accordingly custom multimedia instructions that co-exist with the base instruction set of the extensible processor core. Practically, some restrictions may apply: for example, the complexity of an instruction (in terms of number of cycles for execution) may be limited in order to integrate the resulting data path into the existing pipeline architecture of the base core. Also, the designer of the custom instructions may be responsible for a proper scheduling when instructions require multi-cycling. Other restrictions may constrain the total number of extensible instructions that can be defined and integrated per processor, etc.

b) *Inclusion/Exclusion of Predefined Blocks*: Predefined blocks as part of the extensible processor platform may be chosen to be included or excluded by the designer. Examples are special function registers, MAC operation blocks, caches, etc.

c) *Parameterization*: Finally, the designer may have the choice to parameterize the extensible processor for a specific multimedia application. Examples include setting the size of instruction/data caches in order to accommodate for the characteristics of the multimedia application, choosing the endianness (little or big endian), choosing the number of general purpose registers, etc.

Fig.2 depicts a typical design flow of an extensible processor platform with the goal to customizing the extensible processor for use in a specific multimedia application which may be available in a C/C++-like specification. Profiling by means of an ISS (Instruction Set Simulator) resembling the target processor unveils the bottlenecks through cycle-accurate simulation i.e. it shows which parts of the application represent the most time consuming ones (or, if the energy consumption is the constraint, which ones are the most energy consuming). The results of the profiling are the basis for the designer to identify possible instruction extensions, inclusion of pre-defined blocks and parameterization. This step is followed by defining a set of extensible instructions with a custom language. It includes scheduling within the processors instruction pipeline, instruction word format, etc.

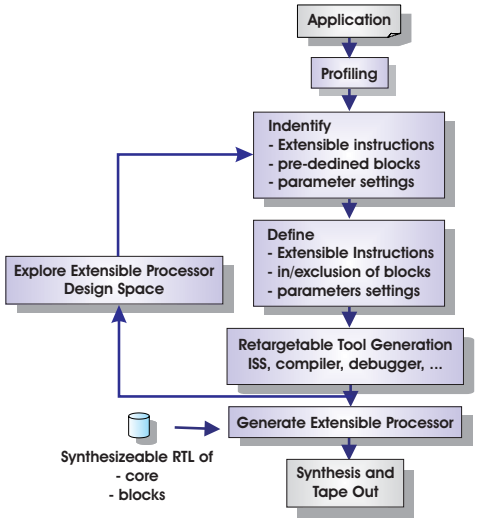

Fig.2: Design flow of extensible processors

The subsequent step is to verify that the various customization levels of the extensible multimedia processor core meet the given constraints (performance, power, etc.) associated with the multimedia application. This can be accomplished by retargetable tool generation. Simply said, retargetable tool generation is a technique that allows to "retarget" compilation/ simulation/analysis tools to the customized micro-architecture.

Once the ISA of an extensible processor is enhanced by a new set of customized instructions, retargetable techniques allow then to automatically generate a compiler that is aware of the new instructions i.e. it can generate code and optimize using the recently defined extensible instructions. Accordingly, an ISS is able to cycle-accurate simulate an executable that contains the new instructions, etc. ISS, compiler assembler are then used in the next step to verify how the multimedia-application-imposed constraints can be met. This step can be iterated until the final tape-out or the setup for rapid prototyping is fixed.

As an example of using this methodology, a complete voice recognition system has been implemented using a base processor core enhanced with less than 10 low-complexity custom instructions. Such a system benefits from speed-up factors between 5x-10x (compared to the version without custom instructions) at a total gate count less than 200k. The total design time (without tape-out but including and extensive design space exploration) took around one month.

In summary, designing a multimedia system using extensible processors as building blocks represents an efficient means to compromise between the major design constraints inherent to complex multimedia systems.

3.2. The communication architecture

Future multimedia systems will become increasingly complex as the demand for functionality increases steadily. In fact, as predicted by ITRS, toward the end of the decade the complexity of SOCs will reach more than 1 billion transistors comprised on a single die. This allows for hundreds of heterogeneous processors to be integrated on one chip. Each of them will be highly customized for a specific part of the multimedia system. At this point, communication becomes a major concern as traditional bus-based architectures fail because of their limited bandwidth in conjunction with their inability to scale.

The NOC approach based on regular architectures was recently proposed as a possible solution to mitigate the complex on-chip communication problems [18]. Such a chip consists of regular tiles, where each tile can be a general-purpose processor, a DSP, a memory subsystem, etc. A router is embedded within each tile with the objective of connecting it to its neighboring tiles. Thus, instead of routing design specific global on-chip wires, the inter-tile communication can be achieved by routing packets.

A NOC has the following attractive characteristics: a) packets are transmitted instead of words. Since the destination address of a packet is encoded as part of the packet header, address lines like in buses become superfluous; b) As opposed to a bus-based system, transactions can potentially be performed in parallel. This is especially important in multimedia systems where access to common memories via a bus often represents the prominent performance bottleneck; c) Routers/switches in the network provide decoupling of wires and thus they alleviate cross-talk and clock skew issues that would otherwise arise when large system buses are used; d) Due to the regularity of typical NOCs (e.g. as a 2D mesh network), the routing of wires is not an issue any more.

Without loss of generality, we assume that a multimedia SoC with a NoC on it is a system of heterogeneous computation nodes. As an example, assume a video surveillance system that has to perform such diverse tasks as motion detection, filtering, rendering, object matching, etc. each of which can be performed by one dedicated application-specific computation node (e.g. an extensible processor). In fact, this heterogeneity in the computation nodes can be very well exploited when designing the NoC. Note, that this is a fundamental difference compared to large-scale networks where each node in the network has to be treated in the same fashion. Whereas in large-scale networks typically no correlation between nodes can be assumed, in multimedia NoCs there is indeed plenty of correlation. Revisiting the video surveillance example from above, it is obvious that the data flow passes from the node performing motion detection to the one performing filtering, so on so forth. Along this path, the network should provide the highest bandwidth, whereas other computational nodes (for example, reading and interpreting user input) require less bandwidth, as well as lesser frequent communication.

As for the design of a specific node, the buffer size of the in/out ports is a key customization parameter from an architectural point of view. In this case, besides the characteristics of the physical resources that compose the multimedia system, the traffic generated by different multimedia applications plays a major part in the overall system performance. Indeed, contrary to other common applications, the bursty nature of the multimedia traffic makes self-similarity a critical design factor for multimedia applications [19]. This has a considerable impact on the queueing performance of the communication architecture since self-similar (or long-range dependent) processes have properties which are completely different from the traditional Markovian processes. Indeed, as opposed to Markovian processes, the self-similar processes typically obey some power-law decay of the autocorrelation function. This produces scenarios which are drastically different from those experienced with traditional short-range dependent models such as Markovian processes. This is the subtle point where the long-range dependence analysis surpasses classical Markovian analysis and proves its practical value.

3.3 NOC optimization

For multimedia systems running on batteries, targeting low-energy consumption is extremely important. At the same time, satisfying the performance constraints that ensure the desired timing behavior is another essential characteristic. Given the target application described as a set of concurrent tasks, its communication profile, a pre-selected architecture and set of available IPs, a few problems to solve are *i*) which tile each IP should be mapped to, *ii*) what routing algorithm is suitable for directing the information among tiles, such that the metrics of interest are optimized and *iii*) how to schedule the computation tasks and the communication transactions onto the target architecture. All these design steps have a significant impact on the overall energy and performance metrics of the system. Indeed, a recently proposed algorithm for energy-aware mapping of the IPs onto regular NoC architectures shows that more than 50% energy savings are possible, for a complex video/audio application, compared to an ad-hoc implementation [20].

Going even at a finer level of granularity for NOC optimization, deciding the packet size is also of paramount importance since it determines how much of the potentially available network bandwidth can be exploited [21][22]. A multimedia system may favor large packet sizes since, for example, entire video frames should be transmitted by means of a small total number of packets. On the other hand, large packets might prohibitively long block a network link causing a degradation in the allowable network throughput. Other customization potentials like HW/SW partitioning of the switch's arbitration algorithm are not discussed here.

The last step above includes deciding on the assignment of tasks and communication transactions onto different computation and communication resources, respectively, and fixing the order of their execution on these resources. Again, for a complex multimedia application, more than 40% energy savings have been observed, on average, compared to the schedules generated by a standard earliest-deadline-first scheduler [23].

Finally, as a general note to efficiently design multimedia NoCs the designer should provide as many local memories as possible instead of few large and globally accessed ones (i.e. memories that are attached to the local computation node via a private link) because of the high memory bandwidth requirements in multimedia systems. If access to few large global memories would be provided through the NoC, the NoC would have to be designed prohibitively conservative to satisfy the worst case node-to-memory bandwidth requirement. Though research of NoCs is still in its infancy, we believe that NoCs bear a large potential to drive the evolvement of complex future multimedia systems.

4. Networking multimedia systems: The network-centric perspective

With advances in process fabrication and design technologies and escalating demand for ubiquitous communication, we witness an explosive growth in wireless multimedia applications, e.g.

streaming audio and video. This trend, in turn, poses two challenges: *i*) establishing and maintaining a stable channel as a communication medium for real-time operation of a network of multimedia systems and *ii*) energy-aware operation so as to increase the service lifetime of a network of multimedia systems while delivering acceptable levels of QoS. Furthermore, it is desirable to provide mechanisms for graceful degradation in QoS such that a dynamic power manager (DPM) can incrementally trade off QoS for higher energy efficiency.

In a network of battery-powered multimedia systems, there are two sources of energy consumption: *computation energy* for processing the video stream and *communication energy* for transmitting or receiving the data. The computation energy is usually a strong function of the CPU clock frequency of the multimedia system, which may be varied by using methods such as dynamic voltage and frequency scaling (DVFS) [24]. The communication energy, on the other hand, strongly affects the bit-error-rate (BER), and thereby, the received video quality.

There are detailed studies of the trade-off between energy consumption and BER in the communication field [25]. This previous work can be roughly divided into two categories. The first category of techniques, which focus on the pass-band transceiver, exploits the fact that different modulation schemes result in different BER vs. received signal-to-noise ratio (SNR) characteristics. The key trade-off is thus between the modulation and/or power levels and the BER. The second category of techniques, which focus on the base-band transceiver, studies the interaction between code performance and encoder/decoder design complexity. The key trade-off is between the complexity of the encoding/decoding algorithms and the BER.

Combining these two techniques, a low energy wireless communication system can be envisioned [26], where the modulation level and transmit power of the transmitter and the complexity of the channel decoder of the receiver are dynamically changed to match the characteristics of the communication channel thereby minimizing the energy consumption of the transceivers. Experimental results show an average of 12% reduction in the overall energy consumption of the transceivers without any appreciable performance penalty.

In [27], the authors propose an energy-optimized image transmission system for indoor wireless applications that exploits the variations in the image data and the wireless multi-path channel by using dynamic algorithm transformations and joint source-channel coding. A detailed energy model for the client-server system is proposed and a global optimization problem is solved by using the feasible direction methods. This results in an average of 60% energy saving for different channel conditions.

4.1. Energy-aware video streaming

Although there are many examples of networks of multimedia terminals, we consider the MPEG-4 video streaming in both client-server (infrastructure mode) and distributed (ad hoc mode) network configurations. Generally speaking, the achievable video quality in streaming video systems is determined by three factors: encoding capability of the sending multimedia host (video server), decoding capability of the receiving multimedia host (video client), and the wireless channel error rate. It is well known that channel bandwidth fluctuation due to various factors result in the severe degradation in the video quality. This is due to the streaming nature of this real-time operation and the extra time, which is required for retransmissions if errors occur in the data packets.

The encoding (decoding) aptitude of the video server (client) is defined as the amount of data that can be processed by a deadline. This aptitude is proportional to the inverse of the video frame rate. When the server (or/and the client) changes its operating frequency and voltage to extend its lifetime, the encoding (decoding) aptitude is also affected, so is the quality of the streaming video.

In [28] a low energy MPEG-4 FGS streaming [29] policy using a client-feedback method is presented, where the client decoding aptitude in each timeslot is communicated to the server, and the server subsequently determines the additional amount of data in the form of enhancement layers on top of the MPEG-4 base layer. Therefore, the server adjusts its data rate based on the feedback value from the client. On the client side, a dynamic voltage and frequency scaling technique is used to adjust the decoding aptitude of the client while meeting a constraint on the minimum achieved video quality. As a measure of energy efficiency of the video streamer, the notion of a normalized decoding load is introduced. It is shown that a video streaming system that maintains this normalized load at unity produces the optimum video quality with no energy waste. Based on the actual current measurements on an XScale-based testbed, the authors report an average of 15% communication energy reduction in the client by making the MPEG-4 FGS streamer energy-aware.

4.2. Energy-aware routing protocols

One of the main design constraints in mobile ad hoc network (MANET) of multimedia systems is that they are energy constrained. Hence, network routing algorithms must be developed to consider energy consumption of the multimedia hosts in the network as a primary objective. In MANETs, every multimedia host has to perform the functions of a router. So if some hosts die early due to lack of energy, thereby causing the network to becomes fragmented, then it may not be possible for other hosts in the network to communicate with each other. It is therefore critical to develop energy-aware routing protocols for MANETs whose aim is to maximize the network lifetime (which in this context may be defined as the duration of time after which a fixed percentage of multimedia hosts in the network "die" as a result of energy exhaustion.)

Many researchers have addressed the problem of energy-efficient data transfer in the context of multi-hop wireless networks. Existing protocols may be classified into two distinct categories. One category of protocols is based on *minimum-power routing* algorithms, which focus on minimizing the power requirements over end-to-end paths. A typical protocol in this category selects a routing path from a source to some destination so as to minimize the total energy consumption for transmitting a fixed number of packets over that path. Each link cost is set to the energy required for transmitting one packet of data across that link and Dijkstra's shortest path algorithm is used to find the path with the minimum total energy consumption. These protocols traditionally ignore the power dissipated on the receiver side in a node, and therefore, tend to result in routing paths with a large number of short hops. A key disadvantage of these protocols is that they repeatedly select the least-power cost routes between source-destination pairs. As a result, nodes along these least-power cost routes tend to "die" soon by rapidly exhausting their battery energy. This is doubly harmful since the nodes that die early are precisely the ones that are most needed to maintain the network connectivity (and hence increase the useful service life of the network.) An example include the Minimum Power Routing in [30].

A second category of protocols is based on routing algorithms that attempt to *increase the network lifetime* by distributing the forwarding load over multiple different paths. This distribution is performed by either intelligently reducing the set of nodes needed to perform the forwarding duties, thereby, allowing a subset of nodes to sleep over different periods of time, or by using heuristics that consider the residual battery power at different nodes and route

around nodes that have a low level of remaining battery energy. In this way, they balance the traffic load inside the MANET so as to increase the battery lifetime of the nodes and the overall useful life of an ad hoc network. These protocols indeed constitute the state-of-the-art in power-aware network routing protocols. Examples include the Battery-Cost Lifetime-Aware Routing of [31] and the Lifetime Prediction Routing [32].

Although these power-aware network routing protocols and algorithms tend to create additional control traffic, simulations show that they improve the network lifetime by more than 20%, on average.

5. Emerging trends in multimedia

Ambient multimedia is at the very heart of a more human-centric (rather than computer-centric) world that the vision for Ambient Intelligence systems brought to life in the recent years. Simply stated, ambient multimedia represents the vision of pushing the idea of distributed multimedia systems to the extreme by completely embedding (or hiding) multimedia systems into surroundings. Practically speaking, such systems are at the very basis of building truly smart spaces (future homes, offices, shopping areas, airports, etc.) that can not only involve complex interactions among humans, different smart fabrics, networks of sensors, etc. but also change completely the life styles in our society. From a research perspective, implementing multimedia applications based on resource-constrained (i.e. memory, battery lifetime, computational power, etc.) systems brings many interesting issues into the picture. For instance, from dealing with mobility, fault-tolerance, power management, security, all the way to designing interfaces, processors and customized circuits, just to name a few [33].

Recent years have witnessed an increased interest towards making Ambient Intelligence a reality of our lives. Although it appears that the advances in the electronic industry will allow someday to build routinely such systems, the ambient multimedia remains perhaps the biggest challenge to overcome in such an endeavor. Indeed, a few characteristics make these systems quite unique. To start with, they should be completely embedded into the environment, able to operate with limited resources and failing parts, and, at the same time, really inexpensive. More importantly, since the human user gets the driver seat through a system of complex interactions based on sensing and actuation, the ability to consider users behavior when building the overall performance model becomes a must. Since users tend to behave non-deterministically, there is room for stochastic modeling based on capturing the uncertainty in users behavior [34]. Under such circumstances, what needs to be figured out is a completely new design methodology that would enable the evolution from distributed multimedia (e.g. couple of bulky video cameras that can barely follow a moving speaker) to ambient (or pervasive) multimedia (e.g. many tiny cameras inconspicuously embedded into the surroundings along with support from smart interfaces, flexible middleware, etc.).

6. Conclusion

In this paper, we have discussed several issues related to successful design of distributed multimedia applications and envisioned a holistic perspective towards a coherent design methodology. While low-power is already a strong design constraint, we believe that most of the challenges in the future will come from designing ambient multimedia systems.

References

[1] S. Edwards, L. Lavagno, E. A. Lee, A. Sangiovanni-Vincentelli, ‘Design of embedded systems: formal models, validation, and synthesis,’ Proc.IEEE, Vol.85, no.3, March, 1997.

[2] G. Martin et al, ‘Surviving the SOC Revolution: A Guide to Platform Based Design,’ Kluwer Academic Publishers, 1999.

[3] S. V. Ranghavan, S. K. Tripathi, ‘Networked Multimedia Systems,’ Prentice Hall, 1998.

[4] A. Kalavade, P. Moghe, ‘A tool for performance estimation of networked Embedded End-Systems,’ Proc. DAC, June 1998.

[5] G. Blair, L. Blair, H. Bowman, A. Chetwynd, ‘Formal Specification of Distributed Multimeida Systems,’ University College London Press, London, 1998.

[6] R. Marculescu, A. Nandi, L. Lavagno, A. Sangiovanni-Vincentelli, ‘System-Level Power/Performance Analysis of Portable Multimedia Systems Communicating over Wireless Channels,’ in Proc. ICCAD, Nov. 2001.

[7] B. Plateau, J. M. Fourneau, ‘A Methodology for Solving Markov Models of Parallel Systems,’ Journal of Parallel and Distributed Camp., Vol. 12, 1991.

[8] J-Y. Brunel et al., “Communication Refinement in Video Systems On Chip,” in Proc. CODES, Rome, 1999.

[9] P. van der Wolf, et al, ‘An MPEG-2 Decoder Case Study as a Driver for a System Level Design Methodology,’ in Proc. CODES, Rome, 1999.

[10] H. Bowman, J. Bryans, and J. Derrick, ‘Analysis of a Multimedia Stream using Stochastic Process Algebra,’ the Computer Journal, Vol. 44, No.4, British Computer Society, 2001.

[11] A. Nandi, R. Marculescu, ‘System-Level Power/Performance Analysis for Embedded Systems Design,’ in Proc. of DAC, June 2001.

[12] Arctangent Processor. ARC International. (http://www.arc.com).

[13] AsipMeister. (http://www.eda-meister.org/asip-meister/).

[14] Jazz DSP. Improv Systems Inc. (http://www.improvsys.com).

[15] Lexra Processor. Lexra Inc. (http://www.lexra.com).

[16] Lisatek/CoWare Inc. (http://www.coware.com).

[17] Xtensa Processor. Tensilica Inc. (http://www.tensilica.com).

[18] A. Jantsch, H. Tenhunen, eds, ‘Networks on Chip,’ Kluwer Academic Publishers, 2003.

[19] G. Varatkar, R. Marculescu, ‘Traffic Analysis for On-chip Networks Design of Multimedia Applications,’ in Proc. DAC, June 2002.

[20] J. Hu, R. Marculescu, ‘Energy-aware mapping for tile-based NoC architectures under performance constraints,’ Proc. ASP-DAC, Jan. 2003.

[21] T. T. Ye, L. Benini, G. De Micheli, ‘Packetized on-chip interconnect communication analysis for MPSoC,’ Proc. DATE, March 2003.

[22] J. Liang, S. Swaminathan, R. Tesssier, ‘aSoC: a scalable, single-chip communications architecture,’ Proc. PACT, Oct. 2000.

[23] J. Hu, R. Marculescu, ‘Energy-Aware Communication and Task Scheduling for Network-on-Chip Architectures under Real-Time Constraints,’ in Proc. DATE, Feb. 2004.

[24] T. Pering, T. Burd, R. Broderson, ‘The simulation and evaluation of dynamic voltage scaling algorithms,’ in Proc. ISLPED, 1998.

[25] J. Proakis, Digital Communications, McGraw-Hill, 3rd Edition, 1995.

[26] A. Iranli, H. Fatemi, M. Pedram, "A Game Theoretic Approach to Dynamic Energy Minimization in Wireless Transcievers," Proc. ICCAD, Nov. 2003.

[27] S. Appadwedula, et al, ‘Total System Energy Minimization for Image Transmission,’ Journal of VLSI Signal Processing Systems, Feb. 2001.

[28] K. Choi, K. Kim, M. Pedram, "Energy-aware MPEG-4 FGS Streaming," in Proc. DAC, 2003.

[29] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," IEEE Trans. on Circuits and Systems for Video Technology, Vol.11, No. 3, Mar. 2001.

[30] S. Singh, M. Woo C.S. Raghavendra, "Power-Aware Routing in Mobile Ad hoc Networks," in Proc. Mobicom, 1998.

[31] C.K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad hoc Networks," IEEE Communication Magazine, June 2001.

[32] M. Maleki, K. Dantu, M. Pedram, "Lifetime Prediction Routing in Mobile Ad Hoc Networks," Proc. IEEE Wireless Communications and Networking Conf., Mar. 2003.

[33] D. Marculescu, N.H. Zamora, P. Stanley-Marbell, R. Marculescu, "Fault-Tolerant Techniques for Ambient Intelligent Distributed Systems," in Proc. ICCAD, San Jose, CA, Nov. 2003

[34] G. Doherty, M.Massink, G. Faconti, “Reasoning about Interactive Systems with Stochastic Models,” in Proc. DSV-IS 2001, Glasgow, LNCS, Vol. 2220, 2001.