Policy Construction and Validation for Energy Minimization in Cross Layered Systems: A Formal Method Approach [*]

Minyoung Kim, Nikil Dutt, Nalini Venkatasubramanian
School of Information & Computer Sciences
University of California, Irvine, CA 92697-3425
{minyounk, dutt, nalini}@ics.uci.edu

Abstract

The highly dynamic nature and stringent timing constraints of distributed, real-time, and embedded (DRE) systems lead to complex cross-layer interactions and valid designs must satisfy a multitude of constraints. In this paper, we focus our attention on design validation considering multidimensional interoperability in the context of cross-layer approaches for power optimization under timing constraints in distributed mobile systems. Specifically, we (i) formally specify each layer of abstraction in consort with timing and energy properties, and (ii) evaluate an optimized policy for design validation as well as provide a time-energy critical path for further optimization that will cost-effectively address the Quality of Service (QoS)/performance tradeoffs. By providing a design flow that includes both timing verification and cross layer optimization, we can achieve (i) timing guarantees for design verification, (ii) better optimizations for resource management, and (iii) adaptive parameter settings for resource sharing and energy minimization. We present preliminary results on an MPEG application.

1 Introduction

With advances in technology and the trend towards convergent mobile computing, there is a growing demand for distributed, real-time, and embedded systems such as high quality mobile multimedia communications. A key challenge in developing applications for these systems is to satisfy multidimensional constraints (e.g., timing, energy, reliability, and cost) as well as to guarantee correctness with degrees of confidence. For instance, resource-sensitive delivery of multimedia content with high quality attributes requires tradeoff analysis across various layers (e.g., enabling hardware, OS, middleware, and application) of system implementation and functionality. In this context, several studies [1, 2] have focused on exploiting cross-layer optimizations to achieve energy gains for mobile devices primarily running multimedia applications while provisioning QoS. To be effective, such optimizations must take into account other competing criteria such as delivery of defined functionality within a limited energy budget satisfying given timing constraints.

Additionally, since real-time systems differ from untimed systems in that their behavioral correctness relies not only on the results of their computations but also on the time when the results are produced, timing verification is critical to ensure the correctness of system design. However, due to the system's complexity and interactions among components, verifying that a system satisfies its timing constraints is extremely difficult. Conventional approaches to validate the constructed policy are simulation or an actual implementation of the policy in the system under consideration. However, these techniques are time consuming and often not completely reliable in the sense that it cannot cover all potential corner cases and possible errors. Under these circumstances, formal approaches become more attractive, offering systematic analysis based on well-defined models since it can guarantee full performance corner-case coverage and ensure bounds for critical performance parameters. Therefore, as an alternative approach, formal techniques can be utilized to assure the correctness of a given system with certain desired timing properties. A good survey on formal verification of timed systems can be found in [5]. In this extensive review, the author discusses commonly accepted models, specification languages, and verification frameworks.

Note that previous studies have independently dealt with either cross-layer optimization or timing verification. Our important observation here is that both issues should be addressed simultaneously, especially in the context of energy-constrained mobile communications. The key contribution of this paper is to propose an integrated formal reasoning framework for conceptualizing and verifying system timing requirements across all layers of a distributed real-time system composed of heterogeneous resource-constrained devices; this framework can deliver cumulative resource (specifically energy) gains across different abstraction levels.

Research Contributions

The focus of this paper is on design validation considering multidimensional interoperability in the context of cross-layer approaches for power optimization under timing constraints in distributed mobile systems. Specifically, we (i) consider power and timing issues simultaneously, (ii) formally specify each layer of abstraction, (iii) evaluate an optimized policy for design validation as well as identify time-energy critical paths for further optimization, and (iv) develop this particular work as a component of a design space exploration framework that provides designers with feasible solutions satisfying multidimensional constraints with degrees of confidence.

---

[*] This work was partially supported by NSF award ACI-0204028 and ONR MURI grant N00014-02-1-0715.

2 Proposed Approach: Formal System Modeling and Analysis

Through insights gained from our previous studies [1, 12], we have identified a critical need for a methodology that captures the dependency and interaction across subsystems to maximize resource gain while satisfying global timing constraints; existing cross layer optimizations only consider the integration of the individual techniques for each layer without concrete timing analysis. To verify if time critical systems comply with their time bounds, we discuss the use of formal methods, in the verification and analysis of multimedia systems. Therefore, the focus of this paper is on conceptualizing and formally verifying the system properties related with timing constraints for cumulative resource gain. Formal analysis provides corner-case coverage and bounds for critical performance parameters as well as supporting extensive trade-off analysis. In addition, sensitivity analysis to assign the constraints with varying degrees of accuracy and tolerance to delay can be performed for more aggressive adaptation. The objectives are to develop analysis techniques to ensure correctness of a design with given timing and resource constraints, and to provide a framework for design space exploration.

To achieve our goals, we need to take the following steps: (i) analyze and derive relevant timing properties at different cross-computational levels. (ii) formally specify and evaluate the cross layer tasks that incorporate properties and perform cross-layer optimization considering timing issues; map timing primitives across layers in order to support cross layer specification and analysis. (iii) understand the implications of system dynamicity and corresponding adaptations; alter specifications to accommodate dynamic changes and to incorporate adaptation policies through back annotation for aggressive resource management (e.g., exploit latency margin).

By carefully modeling how timing delays at each layer affects the timing of entire systems, we will be able to achieve timing guarantees for design verification and better optimizations for resource management (e.g., energy) while preserving application QoS. Furthermore, design space exploration can be done using extensive tradeoff analysis on energy reduction and tolerance level to timing violation with a highly abstracted system model.

2.1 Real Time Maude as Formalism

We need ways of formally expressing application requirements, dynamicity of environment, and correctness constraints. What to specify and how to identify the interesting properties will affect the accuracy and efficiency of the analysis. The extraction of relevant aspects from the system descriptions and the mapping to the formal model is dependent on the level of abstraction in the extraction step and the granularity in the mapping step. As for the policy construction, the FORGE framework [1] can exploit a good policy considering cross layer optimization. However, our main concern in this paper is policy analysis based on a formal reasoning framework.

Our formal modeling approach utilizes Real-time Maude [3], which is a language and tool for the high-level formal specification, simulation, and analysis of real-time and hybrid systems. Its theoretical background is rewriting logic [13]. Real-time Maude emphasizes ease and generality of specification, including support for real-time object-based systems that can be distributed, and where the number of objects and messages can change dynamically. Furthermore, it supports a wide spectrum of formal methods, including timed search and time-bounded linear temporal logic model checking from a specified initial state. Real-time Maude has been used to model and analyze timing and correctness properties in substantial case studies such as the AER/NCA protocol suite, the OGDC density control algorithm for wireless sensor networks, and the CASH scheduling algorithm [3, 14].

Therefore, we take advantage of Real-time Maude to enable the development of executable specifications with timing constraints and the subsequent analysis of systems with timed execution, time sampling strategies, time-bounded model checking. However, it should be noted that any other formal verification language and tools that have real time analysis features could be utilized.

2.2 Specification

In Real-time Maude, the system state (configuration) is represented as a multi-set of objects and messages. System behavior can be represented by rewrite rules of the form

$$\boxed{(c)rl{:}configuration \rightarrow configuration'\ (if\ condition\ )}$$

where *rl* and *crl* are terms indicating an unconditional rule and a conditional rule, respectively. Rules describe local and potentially concurrent state transitions that can either be instantaneous or timed. To model the elapse of time in a system for a timed transition, tick rules are used as follows.

$$\boxed{\begin{array}{l} crl\ [tick]{:}\{C{:}Configuration\} \\ \quad \rightarrow \{delta(C{:}Configuration,\ T{:}Time) \\ \qquad in\ time\ \ if\ T\ <=\ mte(C) \end{array}}$$

This tick rule advances time non-deterministically by any time *T* less than or equal to *mte* (maximum time elapse) with the effect of altering the configuration as stated in the delta function.

2.3 Analysis

Once we have a Real-time Maude specification, (timed) rewriting and timed (time-bounded) search are performed to generate execution traces for analyzing the system. Rewriting simulates one behavior of the system from a specified initial state while timed search analyzes all possible behaviors of the system relative to a chosen time sampling strategy in a breadth-first manner for states which match a given search pattern satisfying a given condition. Details are in reference [3].

From the information obtained by (timed) rewriting and timed (time-bounded) search, model checking can statically validate the properties expressed in rewriting temporal logics as shown below.

$$\boxed{\begin{array}{l} mc\ initState\ |{=}\ t\ formula\ in\ time\ t \\ \qquad\qquad\qquad (or\ with\ no\ time\ limit)\ . \end{array}}$$

Corner case behaviors in a system might not be discovered during extensive simulations, as they only investigate one behavior. In addition to being able to execute a single behavior of a system, Real-Time Maude's other analysis capabilities can be used to perform a more through analysis on the given schemes in search for design errors/performance estimations. For example, to check the range of battery life time, we can use the *find earliest* or *find latest* command.

```
(find earliest{initial configuration})
=> * { < CPU : HW | consumedEnergy :
(F : Float) > REST : Configuration}
such that (F : Float) >= residual energy of a battery
                              that you want to check .
```

Time-bounded model-checking can be used to ask questions such as whether a system can survive up to a certain point with certain amount of residual energy as follows.

```
(mc{initial configuration}) |= t []
energyIsLessThan(the amount of residual energy)
in time <= time you want to check .
```

The *find* commands can be used to determine the time bound for model-checking queries in the following form.

```
(mc{initial configuration}) |= t []
~ predicateDescribingStateLookedFor
in time <= time value returned by find earliest .
```

3 Preliminary Experiments

This section illustrates the use of Real-Time Maude analysis on several power management schemes as a very simple but representative example. We model an MPEG video streaming client in a layered manner as depicted in Figure 1. The application layer is abstracted as workload variation. When the processor runs at full speed, the actual execution times of the tasks are assumed to be uniformly distributed between best case execution time (BCET) and worst case execution time (WCET). The middleware layer can be simply modeled by a network delay model. For the simplicity of mathematical analysis, we assume the arrival of incoming processing requests is an exponential process with an average arrival rate. For the OS layer, we modeled four different power management schemes: *Always-on, Greedy, Cluster, DVS scheme*. The *Always-on* policy does not care about energy reduction. In the *Greedy* scheme, the power manager (PM) shuts down the device whenever its idle period is enough to compensate the overhead while in the C*luster* scheme, the PM tries to aggregate idle periods to maximize energy efficiency. We also modeled the *DVS* (Dynamic Voltage Scaling) scheme. Note that since we model each layer of abstraction in a modular manner, a system model can be extended to another complex model by adopting different types of task distribution, network delay, OS scheduling, and enabling hardware technology. For this experiment, we use the workload variation of MPEG in [10] and the hardware implementation in [4]. Figure 2 presents our simulation results. Simulation in Real-time Maude is done with the timed fair rewrite command which executes one behavior in the system, starting from a given initial state. By selecting different initial states, we can easily generate more execution traces. We believe a formal approach that guarantees timing/QoS provision can lead to better solutions for the issues of how to realize a good power manager that considers complex system dynamics, and how to provide the interface between technology knobs and their controls.

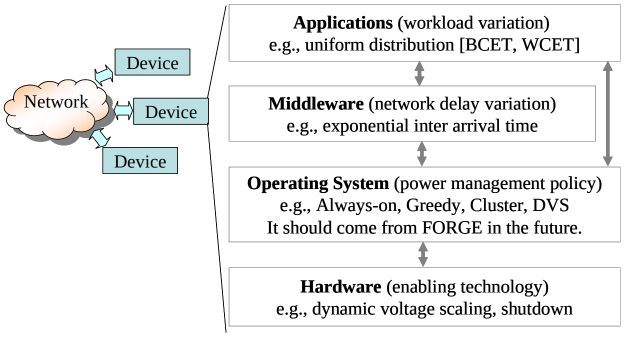
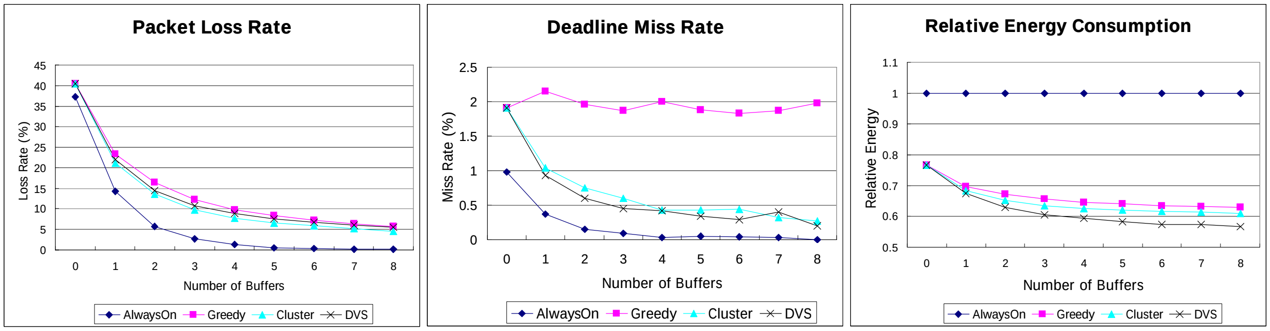


Figure 1. Experimental set-points

Figure 2 presents the estimated QoS, timing, and energy consumption from the policies under evaluation. As for the QoS measure, we simply use packet loss rate due to limited buffer capacity since it is natural that higher packet loss results in lower user satisfaction. In case of timing issue, the deadline miss rate is used. As seen in Figure 2, the *Always-on* strategy leads to the highest quality and timing guarantee at the cost of more energy than any other power management schemes. Figure 2 also illustrates that even though we have an unlimited amount of buffers, the QoS/energy gain will not increase linearly. Figure 2(a) shows that the *Cluster* scheme performs better than other policies from the perspective of packet loss rate since (i) the *Greedy* scheme makes a shutdown decision without any knowledge about the future demands that can result unnecessary wakeup overhead (delay), and (ii) the *DVS* scheme lengthens the execution time of a task that may lead to unnecessary buffer overflows. On the other hand, Figure 2(b)-(c) show that the *DVS* scheme outperforms others since supply voltage reduction results significant energy gain and the time required to change operating voltage/frequency of a device is much shorter than the wakeup delay from the low-power state.

4 Related Works

In this section, we survey previous works from the perspectives of (a) specification, (b) optimization, and (c) validation, and distinguish/highlight our research contributions. As for (a) specification, the authors of [6, 7, 8] attempt to use formal specification models, languages and tools to facilitate the design of multimedia systems. In the area of (b) optimization, several approaches have been proposed for application specific energy minimization in predictive [9, 10] and stochastic [6, 11] schemes. As for (c) validation, formal validation of a given policy based on probabilistic model checking has been introduced in [11]. However, to the best of our knowledge, there has been no work done on the integration of formal validation and cross-layer optimization. It should be also



(a) **(b)** **(c)**

Figure 2. Estimated QoS, timing, and energy perspective on varying power management strategy and buffer size: (a) packet loss rate, (b) deadline miss rate, and (c) energy consumption as a QoS, timing, and validation metric, respectively.

pointed out that our formal system modeling and analysis improves the quality of the constructed policy by providing time-energy critical paths for further optimization as well as avoiding time consuming error-prone simulation. Lastly, since we develop this particular work as a component of a design space exploration framework, we provide system designers with feasible solutions satisfying multidimensional constraints with some degree of confidence.

5 Summary and On-going Work

In this work, we focus on energy-awareness from a global perspective and timeliness delivery based on formal analysis from the observation that resource management and guaranteeing timing properties are critical issues in DRE systems. By providing a design flow that includes both timing verification and cross layer optimization, we can achieve (i) timing guarantees for design verification, (ii) better optimizations for resource management, and (iii) adaptive parameter settings for resource sharing and energy minimization. Therefore, we believe that a comprehensive design methodology for guaranteeing end-to-end requirements of real-time systems based on formal reasoning frameworks can potentially serve cumulative resource (specifically energy) gains in cross-layer optimization.

Preserving timing constraints with dynamic adaptations opens a wide scope of future research subjects. As an immediate following step, we will develop design space exploration mechanisms for extensive tradeoff analysis on dynamic resource management policies and tolerance to timing violation using abstract system models. Our future work will aim to deal with uncertainty in timing. We also seek a technique for specification of hard real-time process interaction and the inter-process timing relationships. From the viewpoint of system constraint refinement, we will pursue research on specification techniques that have the capability of modeling, analyzing, and studying transformations that convert the end-to-end requirements into a set of intermediate rate constraints on the tasks in a compositional manner.

References

[1] http://forge.ics.uci.edu

[2] http://rsim.cs.uiuc.edu/grace

[3] http://heim.ifi.uio.no/~peterol/RealTimeMaude

[4] http://www.intel.com/design/pca/prodbref/252780.htm

[5] F. Wang. Formal verification of timed systems: A survey and perspective. *IEEE*, 92(8):1283 - 1305, August 2004.

[6] Q. Qiu, Q. Wu, M. Pedram. Dynamic power management of complex systems using generalized stochastic petri nets. In *DAC '00: Proceedings of the 37th conference on design automation*, pages 352-356, 2000.

[7] S. Campos, B. Ribeiro-Neto, A. Macedo, L. Bertini. Formal verification and analysis of multimedia systems. In *MULTIMEDIA '99: Proceedings of ACM international conference on Multimedia (Part 1)*, pages 419-430, 1999.

[8] S. M. Koriem. R-nets for the performance evaluation of hard real-time systems. *J. Syst. Softw.*, 46(1):41-58, 1999.

[9] C. Tayor, et al. Energy/latency/image-quality tradeoffs in enabling mobile multimedia communication. *Software Radio: Technologies and Services*, 2001, pp.55-66.

[10] C. Im, H. Kim, and S. Ha. Dynamic voltage scheduling with buffers in low-power multimedia applications. *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 4, Nov. 2004, pp. 686-705.

[11] G. Norman, D. Parker, M. Kwiatkowska, S. Shukla, and R. Gupta. Using probabilistic model checking for dynamic power management. *Formal Aspects of Computing*, 17(2), pages 160-176, 2005.

[12] S. Mohapatra, R. Cornea, H. Oh, K. Lee, M. Kim, N. Dutt, R. Gupta, A. Nicolau, S. Shukla, N. Venkatasubramanian. A cross layer approach for power-performance optimization in distributed mobile systems. In *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium - Workshop 10*, page 218.1, 2005.

[13] J. Meseguer. Software specification and verification in rewriting logic. Technical report, NATO Advanced Study Institute, Marktoberdorf, Germany, 2003.

[14] P. Olveczky, M. Keaton, J. Meseguer, C. Talcott, S. Zabele. Specification and analysis of the aer/nca active network protocol suite in real-time maude. In *FASE '01: Proceedings of the 4th Conference on Fundamental Approaches to Software Engineering*, pages 333-348, 2001.