Energy Management in Wireless Networked Embedded Systems

G. Manimaran
Iowa State University, USA

INTRODUCTION

Real-time systems have undergone an evolution in the last several years in terms of their number and variety of applications, as well as in complexity. A natural result of these advances, coupled with those in sensor techniques and networking, have led to the rise of a new class of applications that fall into the distributed real-time embedded systems category (Loyall, Schantz, Corman, Paunicka, & Fernandez, 2005; Report, 2006). Recent technological advancements in device scaling have been instrumental in enabling the mass production of such devices at reduced costs. As a result, applications with a number of internetworked embedded systems have become prominent. At the same time, there has been a need to move from stand-alone real-time unit into a network of units that collaborate to achieve a real-time functionality. Extensive research has been carried out to achieve real-time guarantees over a set of nodes distributed over wired networks (Siva Ram Murthy & Manimaran, 2001). However, there exist a number of real-time applications in domains, such as industrial processing, military, robotics and tracking, that require the nodes to communicate over the wireless medium where the application dynamics prevent the existence of a wired communication infrastructure. These applications present challenges beyond those of traditional embedded or networked systems, since they involve many heterogeneous nodes and links, shared and constrained resources, and are deployed in dynamic environments where resource contention is dynamic and communication channel is noisy (Report, 2006, Loyall et al., 2005). Hence, resource management in embedded real-time networks requires efficient algorithms and strategies that achieve competing requirements, such as time sensitive energy-efficient reliable message delivery. In what follows, we discuss some applications in this category, and discuss their requirements and the research challenges.

Safety-critical mobile applications running on resource-constrained embedded systems will play an increasingly important role in domains such as automotive systems, space, robotics, and avionics. The core controlling module in such mission critical applications is an embedded system consisting of a number of autonomous components. These components form a wireless (ad hoc) network for cooperatively communicating with each other to achieve the desired functionality. In these applications, a failure or violation of deadlines can be disastrous, leading to loss of life, money, or equipment. Hence, there arises a need to coordinate and operate within stringent timing constraints, overcoming the limitations of the wireless network. For example, robots used in urban search and rescue missions cooperate together and with humans in overlapping workspaces. For this working environment to remain safe and secure, not only must internal computations of robots meet their deadlines, but timely coordination of robots behavior is also required (Report, 2006). Other such medium-scale distributed real-time embedded applications include target tracking systems that perform surveillance, detection, and tracking of time critical targets (Loyall et al., 2005), or a mobile robotics application where a team of autonomous robots cooperate in achieving a common goal such as using sensor feeds to locate trapped humans in a building on fire. Other more passive applications include the use of networked embedded systems to monitor critical infrastructure such as electric grids (Leon, Vittal, & Manimaran, 2007). These applications need to meet certain real-time constraints in response to transient events, such as fast-moving targets, where the time to detect and respond to events is shortened significantly. In surveillance systems, for example, communication delays within sensing and actuating loops directly affect the quality of tracking. While providing real-time guarantees is the primary requirement in these applications, mechanisms need to exist to meet other crucial system needs such as energy consumption and accuracy (Rusu, Melhem & Mosse, 2003). In most cases, there are tradeoffs involved in balancing these competing requirements.

BACKGROUND

The typical architecture in a distributed real-time embedded system consists of several processor-controlled nodes interconnected through one or more interconnection networks. The system software running on each node enables the execution of one or more concurrent tasks that are activated by the arrival of triggering events generated by the external environment, a timer, or arrival of a message from another task. A response to an event generally involves several tasks to be executed on different nodes, and several messages to be exchanged in the network. The tasks on the same node may share data and resources using synchronization mechanisms present in shared memory systems, and also interact with tasks on other nodes by exchanging messages using the services provided by the communication subsystem. For the proper functioning of the whole system, each individual task, as well as all the messages exchanged, need to be completed before specified deadlines.

The workload in the majority of the distributed embedded real-time applications is similar to those found in traditional real-time systems comprising of periodic and aperiodic tasks. Periodic tasks form the base load invoked at regular intervals, while aperiodic tasks include the transient load generated in response to alarm or an external environment stimuli. However, one can expect stronger cooperation between the internetworked units in more dynamic and complex systems inducing richer communication patterns than simple periodic messages. For distributed real-time embedded system, the primary requirement is that there is an end-to-end timing requirement that needs to be met. This implies that there exists a set of messages with complex precedence constraints that need to be exchanged between the networked nodes before some deadline. Hence, one needs to characterize the different message communications and computations that are possible, and perform a preruntime analysis to guarantee, a priori, that all the task deadlines will be met. Moreover, in a distributed real-time system, the ability to meet task deadlines largely depends on the underlying task allocation, and hence, we need a preruntime task allocation algorithm that takes into consideration the real-time constraints. Intertask communication significantly influences the response time of these distributed applications and hence, the design needs to account for the effect of delays imposed by the communication network and precedence constraints imposed by the communicating tasks during task allocation. Since the inherent nature of many of the discussed applications precludes the use of wired networks, wireless networks are commonly used in such applications.

The wireless medium is inherently unreliable due to characteristics such as fading and interference. Hence, to guarantee that tasks should meet timing constraints, it becomes necessary to develop techniques that characterize the unreliability in the network channel, and take them into account while making transmission scheduling decisions. Energy management is another crucial aspect for internetworked embedded devices. These devices contain not only radio and computer components, but also complete system functionalities, such as networking functions across all levels of the protocol stack. Energy savings and allocation among these modules will affect the life time of these battery-powered devices. Energy management also needs to be considered, together with other constraints in size, real-time requirements, functionalities, and network connectivity.

In summary, the combination of temporal requirements, limited resources and power, networked system architectures, time-varying wireless channel, and high reliability requirements presents unique challenges

(Loyall et al., 2005; Report 2006).The end goal of most of the research in this area is to devise efficient resource management algorithms for energy-constrained and highly dynamic wireless networks in order to support end-to-end system requirements that are comparable to their wireline counterparts.

MAIN FOCUS OF THE CHAPTER

Energy management is one of the key issues in the design and operation of networked embedded systems, which involves energy management at the system level considering both computing and communication subsystems. For embedded computing, there are well known techniques, such as dynamic voltage scaling (DVS) (Aydin, Melhem, Mosse, & Alvarez, 2004; Shin, Kim, & Lee, 2005) and dynamic power adaptation (DPM), that have been exploited by intertask and intratask scheduling algorithms. For wireless communication, techniques such as dynamic modulation scaling (DMS) (Raghunathan, Schurgers, Park, & Srivastava, 2002), dynamic code scaling (DCS), power adaptation (Raghunathan, Pereira, Srivastava, & Gupta, 2005), and adaptive duty cycling have been employed for minimizing energy consumption. These techniques essentially provide energy-time tradeoff, that is, the lesser the time taken for execution of tasks or transmission of messages, the higher the energy consumed.

Our research contributions are in the design of a comprehensive energy management framework, with associated off-line and online scheduling algorithms, for networked embedded systems. In system-level energy management (Kumar, Sudha & Manimaran, 2007; Unsal & Koren, 2003), the fundamental question to be answered is how much of the available slack be allocated to each of task execution and message transmission. In general, the computation energy (for a CPU cycle) consumed is much lesser than the communication energy (for a bit of data transmission) for currently available technologies. Therefore, allocating as much slack as possible for communication energy optimization sounds appealing on the surface. However, our analysis shows that there is a diminishing return when the transmission time is increased beyond a certain threshold, with coding taken into account (Kumar et al., 2007). Therefore, the slack should be allocated in a balanced manner between computing and communication subsystems, considering current energy levels of tasks and messages and the channel condition. In our research, we considered DVS and DMS for energy

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/energy-management-wireless-networked-embedded/13756

Related Content

Universal Access to Health Services Through the Digital Terrestrial Television
Aldo Franco Dragoni (2013). *Journal of Information Technology Research (pp. 51-87).*
www.irma-international.org/article/universal-access-health-services-through/80254

ICT as an Example of Industrial Policy in EU
Morten Falchand Anders Henten (2008). *Information Communication Technologies: Concepts, Methodologies, Tools, and Applications (pp. 882-888).*
www.irma-international.org/chapter/ict-example-industrial-policy/22708

How to Transform the Information Infrastructure of Enterprise into Sustainable, Global-Oriented and to Monitor and Predict the Sustainability of Civilization: The Organizational and Social Aspects
Andrew Targowski (2010). *Information Resources Management Journal (pp. 1-10).*
www.irma-international.org/article/transform-information-infrastructure-enterprise-into/43717

Potential Challenges of ICT Implementations in Sri Lanka
Kennedy D. Gunawardana (2008). *Information Communication Technologies: Concepts, Methodologies, Tools, and Applications (pp. 2032-2051).*
www.irma-international.org/chapter/potential-challenges-ict-implementations-sri/22798

Roles of Organizational Climate, Social Bonds, and Perceptions of Security Threats on IS Security Policy Compliance Intentions
Princely Ifinedo (2018). *Information Resources Management Journal (pp. 53-82).*
www.irma-international.org/article/roles-of-organizational-climate-social-bonds-and-perceptions-of-security-threats-on-is-security-policy-compliance-intentions/193612