Poster:
Approximation: A New Paradigm also for Wireless Sensing

Thiemo Voigt, Magnus Själander, Frederik Hermans,
Alexandra Jimborean, Erik Hagersten, Per Gunningberg, Stefanos Kaxiras
Department of Information Technology
Uppsala University

Abstract

While for sensor networks energy-efficiency has always been one of the major design goals, energy-efficiency has, for the past decade, also become increasingly important in other disciplines. An emerging class of applications do not require a perfect result. Rather, an *approximate* result is sufficient. *Approximate computing* enables more energy-efficient design of computer systems, reducing, for example, the energy dissipation in data centers. In this poster abstract we argue for embracing the concept of approximation computing in the sensor networking community.

1 Introduction

Energy-efficiency has always been a key concern in wireless sensor networks. This has lead to a plethora of energy-efficient protocols in particular at the MAC and routing layer. Recently, energy-efficiency has also become a major design issue in other disciplines such as computer architecture. Besides mobile computing, one reason is the increasing power consumption of data centers. Data centers consume roughly 3% of the electricity in the US today with an annual growth of 15%. While sensor networks can be used to reduce the cooling cost and power consumption, a promising, orthogonal approach to energy-efficient design that includes also other disciplines is *approximate computing*.

Approximate computing is motivated by the key insight that many applications do not require exact results but that an approximate result is sufficient. Examples include media processing such as video, audio and images as well as data mining. For the latter, the input data contains a lot of noise and hence it makes it hard to distinguish the output of the perfect search result from an approximate result [15]. Other reasons why certain applications do not require perfect results are that [15, 20] *(i)* the human brain might not be able to distinguish between the best and a non-perfect result, *(ii)* the perfect result does not exist as in, e.g., a search, and that *(iii)* input data is noisy or contains sufficient redundancy so algorithms can be lossy but accurate.

Also for many sensing applications, approximate input is sufficient. For a fire monitoring application, the exact temperate value is not needed to make the decision whether a fire has broken out. Often unreliable sensor inputs can be fused to get a reliable understanding of the task at hand.

We argue for embracing the term and concept of approximation also in the sensor networking community. First, we have been implicitly working with approximations already for a long time. Furthermore, with the Internet of Things (IoT), sensor networks are no longer isolated islands but IoT applications span the whole chain from the sensing device to the data center (and back). Towards this end, building on the same concept of approximate computing will help to foster collaboration and optimize systems end-to-end.

2 Approximate Computing

Approximate computing has attracted interest from a wide variety of research communities.

Circuits and Architecture: Reducing the supply voltage causes a quadratic reduction in power and attempts have therefore been made to overscale the voltage without adjusting the frequency [13]. Such scaling affects the critical paths of a circuit and often leads to errors in the most significant bits, which cause unreasonable high errors. Different circuit techniques have been attempted to more gracefully scale the errors with the voltage through various error correcting techniques [10]. A different approach is to design circuits that approximate a behavior and in so doing are able to reduce circuit complexity. Probabilistic pruning identifies circuit paths with low activity and deletes those that do not cause a too large error on the final output [2]. Substitute and simplify is another technique that identifies close to equivalent paths and substitutes one path with the other [23]. A more controversial proposal is to employ neural networks that can be trained and completely replace the execution of an algorithm on a conventional core [14].

Memory: Memory in many systems can be a large contributor to the total power and lends itself easily for approximate computing. The data can be classified with different criticality and then stored in different memory locations with different power and error resilience characteristics. For dynamic random access memories (DRAM) the refresh rate can be altered for different chips. Reducing the refresh rate significantly reduces the power while it increases the probability of errors [21]. For phase change memories (PCM) the write speed and wear out can be traded against error resilience [4].

EWSN'16, February 15–17, 2016, Graz, Austria.

Programming Languages: Several programming languages have been proposed that enable the programmer to classify data that can be treated approximately [3,21]. Rely is a language that also enables the programmer to specify the required accuracy that is then assured by the compiler according to a provided hardware specification [19]. A more automatic approach tries to statically infer which operations that are critical and non-critical and use profiling and runtime monitoring to further classify the critical operations into likely and not likely critical [8].

Algorithms: Approximation can also be achieved through pure software means by, e.g., loop perforation [22] or approximation of functions as in the green framework [6]. A benefit of these approaches is that they can be deployed on conventional hardware.

3 Approximation in Sensor Networking

While the sensor networking community has rarely used the term approximation explicitly, there are many examples of approximation in sensor networking and wireless sensing.

A large number of approaches have been developed with the goal of avoiding to send all collected sensor readings to the sink for processing [5]. Approaches include distributed regression [7] and Chu et al.'s replicated dynamic probabilistic models [9]. The latter indeed use the term "approximate data collection". Along similar lines, Köpke et al. [1] propose to "transmit more important messages more reliably than less important messages, in order to achieve an optimal balance between energy expenditure and reliability". Their motivation is a fire alarm application where the alarm messages should be transmitted more reliably than the periodically sent heartbeat messages. Also, in convergecast applications, messages carrying aggregated values of more sensor nodes should be transmitted with higher reliability than messages with non-aggregated sensor values. Others have proposed similar ideas [17,25].

As discussed in the previous section, Esmaeilzadeh et al. [13] have presented an architecture that leverages approximate computing and low-voltage operations to save energy. In the WSN domain, Kulau et al. [24] have shown that dynamic voltage scaling is also feasible and can extend a sensor node's lifetime with more than 35%.

Approximate computing also has the potential to significantly reduce the energy that sensor networks spend on radio communication. Today's networks mostly rely on retransmissions to ensure that each received packet is a bit-perfect copy of the sent packet. However, if a sensor node can reason about the number and location of errors in a received packet, it can avoid costly retransmissions by operating on approximate data instead. This implies that there is a trade-off between data quality and energy consumption since operating on approximate data is less energy-consuming than performing retransmissions. Hermans et al. [12] have shown that transmission errors in outdoor networks follow specific patterns. Jamieson et al. [18] and Hauer et al. [16] developed techniques to estimate which bytes in a transmission are erroneous, so that only those bytes need to be retransmitted. Similarly, packet combining builds on the idea that two corrupt packets may be used to infer the corresponding correct packet [11]. These approaches pave the way to embracing approximate computing in sensor networks, since they enable receivers to reason about the amount and the location of errors in a received packet.

4 Conclusions

We have argued that the concept of approximation is relevant also for the sensor networking community, in particular when considering Internet of Things applications that span the whole way from the devices to data centers.

5 References

[1] A. Köpke et al. Using energy where it counts: Protecting important messages in the link layer. In *Proc. of EWSN*, 2005.

[2] A. Lingamneni et al. Energy parsimonious circuit design through probabilistic pruning. In *Proc. of DATE*, 2011.

[3] A. Sampson et al. EnerJ: Approximate data types for safe and general low-power computation. In *Proc. of PLDI*, 2011.

[4] A. Sampson et al. Approximate storage in solid-state memories. In *Proc. of MICRO*, 2013.

[5] B. Krishnamachari et al. The impact of data aggregation in wireless sensor networks. In *Proc. of ICDCSW*, 2002.

[6] W. Baek and T. M. Chilimbi. Green: A framework for supporting energy-conscious programming using controlled approximation. In *Proc. of PLDI*, 2010.

[7] C. Guestrin et al. Distributed regression: an efficient framework for modeling sensor network data. In *Proc. of IPSN*, 2004.

[8] J. Cong and K. Gururaj. Assuring application-level correctness against soft errors. In *Proc. of ICCAD*, 2011.

[9] D. Chu et al. Approximate data collection in sensor networks using probabilistic models. In *Proc. of ICDE*, 2006.

[10] D. Mohapatra et al. Design of voltage-scalable meta-functions for approximate computing. In *Proc. of DATE*, 2011.

[11] H. Dubois-Ferrière et al. Packet combining in sensor networks. In *Proc. of SenSys*, 2005.

[12] F. Hermans et al. All is not lost: Understanding and exploiting packet corruption in outdoor sensor networks. In *Proc. of EWSN*, 2014.

[13] H. Esmaeilzadeh et al. Architecture support for disciplined approximate programming. In *Proc. of ASPLOS*, 2012.

[14] H. Esmaeilzadeh et al. Neural acceleration for general-purpose approximate programs. In *Proc. of MICRO*, 2012.

[15] J. Han and M. Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In *Proc. of ETS*, 2013.

[16] J.-H. Hauer et al. Mitigating the Effects of RF Interference through RSSI-Based Error Recovery. In *Proc. of EWSN*. 2010.

[17] J. P. Benson et al. Reliability control for aggregation in wireless sensor networks. In *Proc. of LCN*, 2007.

[18] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proc. of SIGCOMM*, 2007.

[19] M. Carbin et al. Verifying quantitative reliability for programs that execute on unreliable hardware. In *Proc. of OOPSLA*, 2013.

[20] R. Venkatesan et al. Macaco: Modeling and analysis of circuits for approximate computing. In *Proc. of ICCAD*, 2011.

[21] S. Liu et al. Flikker: Saving DRAM refresh-power through critical data partitioning. In *Proc. of ASPLOS*, 2011.

[22] S. Sidiroglou-Douskos et al. Managing performance vs. accuracy trade-offs with loop perforation. In *Proc. of FSE*, 2011.

[23] S. Venkataramani et al. Substitute-and-simplify: a unified design paradigm for approximate and quality configurable circuits. In *Proc. of DATE*, 2013.

[24] U. Kulau et al. A node's life: Increasing wsn lifetime by dynamic voltage scaling. In *Proc. of DCOSS*, 2013.

[25] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE TII*, 2007.