Andreas Reinhardt, Ralf Steinmetz:
Exploiting Platform Heterogeneity in Wireless Sensor Networks for Cooperative Data Processing. In: Proceedings of the 8th GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", Hamburg, Germany, August 2009.

Exploiting Platform Heterogeneity in Wireless Sensor Networks for Cooperative Data Processing

(Extended Abstract)

Andreas Reinhardt, Ralf Steinmetz
Multimedia Communications Lab, Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany
Email: {andreas.reinhardt, ralf.steinmetz}@kom.tu-darmstadt.de

***Abstract*—In wireless sensor networks, nodes are often fitted with low-power components to allow for a long node lifetime when operated on batteries. However, these available resources can be insufficient to perform sophisticated data processing on a local scale, necessitating the transmission of all sensor readings to an external sink. These transmissions are expensive both in terms of delay and energy, and thus undesirable. To alleviate the situation, we propose the use of a heterogeneous sensor network with higher-capacity processing nodes that allow to perform more complex data processing operations within the sensor network. Estimates of the energy consumptions confirm that employing a heterogeneous sensor network can preserve energy and thus lead to an extended lifetime of the network.**

I. Introduction

Nodes in wireless sensor networks (*WSNs*) are generally designed with energy considerations in mind to allow for long lifetimes when operated on batteries [1]. These energy savings however often come at the cost of low-power microcontroller units (*MCUs*) with reduced computational capabilities and low clock frequencies. With only a few kilobytes of RAM and program Flash storage, the possible complexity of applications is additionally limited. These tight resource limits of sensor node platforms (*motes*) disallow some operations to be performed within the sensor network, and commonly require the transmission of data to external nodes which perform the resource-intensive processing tasks. Commonly, the situation is resolved by transmitting all collected data (often only slightly processed, if at all) to an external sink node which performs the processing tasks.

This data forwarding process is however expensive in terms of energy, as especially in the case of multi-hop transmissions in large networks, delay and energy demand increase linearly with the number of hops. A common way to alleviate the number of transmissions in the multi-hop case is data aggregation ([2], [3]), where packets that share the same route are merged on their way to the destination. Aggregation can thus lead to an overall reduction of the number of packets sent, although further data-specific processing is generally not performed.

As a high volume of traffic might still be present in the network, we propose to move the data processing into the network by deploying dedicated processing nodes. These processor nodes provide greater MCU power (and ideally, larger memory sizes) than the deployed nodes to allow for tasks with greater complexity to be performed within the network. By forming a sensor network, which is heterogeneous in terms of computational power, demanding processing operations can be performed within the network, and thus the amount and size of packet transmissions to a base station significantly reduced.

To show the feasibility of this approach, we exemplarily discuss three application scenarios that would significantly benefit from processing the data inside the network. Concisely, we evaluate the demands of *data compression*, *cryptography*, and *high data-rate sample processing*. To provide computational resources for in-network processing, we exemplarily assume TelosB [4] and SunSPOT [5] devices, as they are present in our TWINS.KOM testbed [6]. However, other combinations of motes are possible as well.

After presenting the related work in Sec. II, we present our vision of collaborative data processing in Sec. III, and show a theoretical energy analysis in Sec. IV. We conclude this paper in Sec. V, where we summarize our results and present the next steps.

II. Related Work

Existing *hybrid* sensor network architectures target to reduce the number of hops a packet requires to reach its destination by supplementing a WSN by additional connections over a secondary, often wired, medium.

Sharma and Mazumdar have investigated the use of *limited infrastructure*, i.e. networks with a number of wired connections between sensor nodes, in [7]. Their approach establishes a small-world graph utilizing wired links between a subset of nodes to reduce the overall energy demand as well as the different energy consumption rates of participating nodes. The additional efforts required for the wiring however make it suited for long-term deployments of sensor networks only.

Hu et al. have built a hybrid network from Mica2 motes and Stargate devies for detecting cane toads in northern Australia [8]. Similar to our proposed system, a two-tiered sensor network structure with low-power motes and higher-power processing nodes is given. However, the Stargate's comparably high energy consumption of 4 watts leads to a quick depletion of its battery and thus renders the solution unsuited for long-term autonomous operation.

Wagenknecht et al. also propose to deploy nodes with higher computational capabilities within a WSN to act as clusterheads for *sensor subnetworks*, i.e. partitions of the sensor network [9]. They use embedded systems with a 233 MHz clock frequency and 128 megabytes of RAM as the backbone to interconnect the sensor subnetworks through a wireless mesh network. Although deploying additional gateways allows for shorter multi-hop routes, the energy savings are possibly counterbalanced by the greater energy requirements of the gateways, which are not analyzed in detail in the paper.

A different approach to shift computational tasks into the network is the use of mobile agents. In such networks, data is not forwarded to an external sink, but instead, the processing application (the *mobile agent*), including its state variables, are sent to the node and executed locally [10]. As all process context data are contained within the agent, it can be supplied with input data at one node, while the processing can be performed at a different, more powerful, system. We thus consider it a well-suited supplement to migrate tasks between nodes.

Further dimensions of heterogeneity have been analyzed, such as heterogeneous link qualities ([11], [12]), or energy level heterogeneity [11]. The authors however focus on means to alleviate heterogeneity rather than exploit it.

III. Collaborative Data Processing

Often, motes are too weak to perform computationally intensive tasks locally, or do at least not provide sufficient energy budgets to perform the demanding operations numerous times during their battery lifetime. Especially in the presence of low-power 8-bit microcontroller units (*MCUs*), operations to process 32-bit data require many more instructions than when executed on a native 32-bit platform.

To overcome this limitation of many existing sensor networks, we propose deploying a heterogeneous set of nodes with two distinct levels of computational capability. Like other WSNs, low-power sensor nodes perform sensing and basic processing tasks, and an external sink node acts as data collector. However, additional dedicated nodes are present within the network to perform data processing operations, thereby alleviating the energy-consuming multi-hop data transport to the sink, while overcoming resource limits of low-power platforms. We indicate a sample network topology with nine low-power sensor nodes, two processor nodes, and a single external sink in Fig. 1.

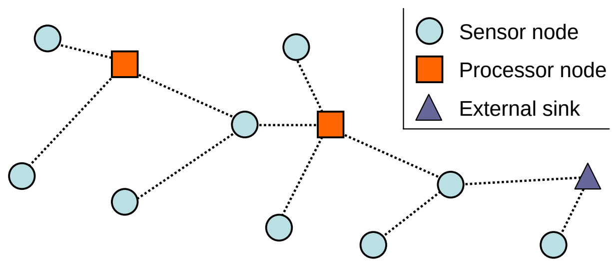

Fig. 1. Exemplary heterogeneous sensor network topology

It is essential to distinguish our system from related work where nodes with secondary network interfaces, peripheral ports, and power consumptions of several watts are used. Instead, we propose the use of low-power embedded systems with support for duty-cycled operation, such as SunSPOTs or Intel's Imote2 nodes. Both resemble native 32-bit architectures with greater computational capabilities, but also greater power consumptions than the low-power motes. We briefly compare both platforms to two common mote platforms in Table I.

TABLE I
Comparison of Mote Platforms

	Mica2[13]	TelosB[4]	SunSPOT[5]	Imote2[14]
RAM size	4 kB	10 kB	512 kB	32 MB
System clock	7.37 MHz	8 MHz	180 MHz	104 MHz*
Sleep current	15 µA	1 µA	31 µA	820 µA
Active current	8 mA	1.8 mA	80 mA	66 mA
Word size	8 bits	16 bits	32 bits	32 bits

* The Imote2 can dynamically adjust its clock frequency between 4 and 416 MHz

In the following, we present how three common application scenarios for WSNs can be supported by our proposed heterogeneous WSN architecture.

A. Data Compression

As both route length and packet size of multi-hop radio transmissions in WSNs have a dependency on the overall energy consumption of the transfer, data compression (also referred to as *source coding*) is a viable approach to compact data (e.g. [15]) prior to sending it. However, the limited amount of memory present on motes is often insufficient to store complex models or code tables, and thus leads to degraded compression gains.

As we have determined in [16], compressing data on a per-packet basis often leads to no improvements over the uncompressed data size at all, while knowledge about the temporal history of data can achieve significant size reductions. Supplementing the source coding by means of in-network processing, such as data aggregation (cf. [2]), can even lead to further savings, but also has higher computational demands on nodes that perform the processing operation. This makes the presented data compression scenario well suited for the proposed heterogeneous networks.

Applied to the scenario depicted in Fig. 1, the processing nodes should integrate within the tree rooted at the external sink node. When configured to aggregate packets on their way to the sink, and compress the results, their presence can lead to energy savings resulting from an overall smaller number of transmissions.

B. Cryptography

High computational demand is also present in area of cryptography, where many algorithms rely on heavy use of the modulo operation. Especially, when the used key length exceeds a platform's native word size, additional operations to emulate the corresponding operations are required, which come at a significantly increased time and thus energy consumption. Emulating these instructions is however often necessary to ensure sufficiently large key lengths.

Measurements on real hardware, performed by Gura et al. in [17], have shown that both ECC and RSA-1024 require more than 4.5 seconds to execute on an 8-bit microcontroller clocked at 14.7 MHz. When more powerful processing nodes are integrated with the sensor network, their greater computational capabilities allow them to perform strong cryptography within reasonable time limits.

Especially, the 32-bit word size and the significantly increased RAM size of the processor nodes reduce the need for instruction emulations and expensive data buffering on external memory, and can thus reduce the required execution time. When the processor nodes act as in-network terminals to provide secured links to the sink node, low-power sensor nodes can employ the AES-128 support of their CC2420 radio transceiver [18] to establish encrypted connections to the processor nodes with a low hop count.

C. High Data-Rate Sample Processing

When sensors that generate high-volume data (such as image or audio sensors) are present within the network, their samples cannot be processed by the sensor node at all times, but are instead forwarded to the sink for further processing. The lack of hardware multipliers in many embedded systems also limits the use of algorithms with many multiplication and addition operations, such as the Fast Fourier Transform (*FFT*). Transferring all data to the sink however leads to a significant volume of traffic in the network.

If instead, a heterogeneous set of nodes is present in the network, resource-demanding tasks can be performed in less time when configuring the processor nodes to specialize on these tasks and request the sensor nodes to transmit their data there. Due to their higher clock frequency and the larger RAM size, the processor platforms inherently consume more energy in all operation modes. However, their reduced processing time improves both transmission delays and power demand, and thus counterbalances the higher energy consumption.

IV. Theoretical Energy Analysis

When considering the current consumption values quoted in Table I, it is obvious that both platforms suited as processors (SunSPOT and Imote2) have a significantly greater energy demand in both active and deep sleep modes than the two sensor node platforms (Mica2 and TelosB). However, the clock frequencies differ by one order of magnitude, hence many more operations can be performed on a processor node within the same amount of time. For the sake of simplicity, we assume an identical number of instructions required to perform the same task on all platforms, although sophisticated features and special extensions to the instruction set present in the processing nodes may lead to deviations.

TABLE II
EXECUTION TIME AND POWER CONSUMPTION OF THE DEMO METHOD

	Mica2	TelosB	SunSPOT	Imote2
Execution time	13.6 ms	12.5 ms	0.55 ms	0.96 ms
Energy per call	327 µJ	67.6 µJ	82.7 µJ	98.4 µJ
Average power	3.3 mW	0.68 mW	0.85 mW	1.37 mW

A. Execution Duration

To visualize the impact of the clock speed, we have assumed a demo method of 100,000 instructions and evaluated the time and energy required to execute it. The corresponding results for a single call are shown in Table II. Additionally, the table contains the results from our analysis of the overall power consumption when calling the method 10 times per second and immediately putting the MCU into sleep mode when the method has finished.

Although both processor node platforms require between 22 and 45 percent more energy to perform the operation, their benefit of a 32-bit architecture and the corresponding reduced emulation demand for complex algorithms is expected to counterbalance the additional energy requirements. Additionally, the average power consumption of the SunSPOT is only 25% higher than the TelosB's when duty-cycling the node, and put into perspective when considering the achievable savings in terms of the overall network traffic.

B. Node Lifetimes

Having determined a comparable energy demand to perform the same algorithms on the more powerful processing platforms, it has become clear that a WSN can benefit from the use of heterogeneous nodes. However, to ensure a long network lifetime, processors should not deplete their batteries faster than the remaining nodes in the network. When continuously operating SunSPOT nodes with a battery capacity of 750 mAh, their lifetime is limited to around nine hours. In contrast, when assuming a duty cycle of only 10% (i.e. spending 90% of the time in sleep mode), lifetime increases to 93 hours, and when activity phases are limited to 2%, the overall node lifetime extends to 16.5 days. It is thus mandatory to find algorithms which achieve a tradeoff between energy and delay constraints, considering the costs of local computation, in-network processing, or the transfer to the external sink in their decision process.

V. Conclusion and Outlook

In this paper, we have presented the benefits of heterogeneous sensor networks, comprising nodes with different computational capabilities. By adding nodes with higher computational performance to a WSN, complex tasks can be performed within the network instead of transferring all data to an external sink node. Although the faster processor nodes exhibit an increased energy consumption, we have theoretically shown that energy savings can be achieved by deploying processor nodes, as their greater energy consumption is counterbalanced by reduced execution times and less traffic in the network.

A. Future Work

In successive work, we target to investigate deployment strategies for the processor nodes and conduct practical experiments with heterogeneous sensor networks, based on our TWiNS.KOM testbed, which integrates TelosB and SunSPOT devices [6]. We also intend to evaluate the applicability of the developed algorithms on networks that are heterogeneous in terms of energy.

Acknowledgements

We would like to thank Parag S. Mogre for the fruitful discussions and his constructive contributions to this paper.

References

[1] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, 2000.

[2] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *Proceedings of the International Workshop on Distributed Event-Based Systems (DEBS)*, 2002.

[3] T. Arici, B. Gedik, Y. Altunbasak, and L. Liu, "PINCO: A Pipelined In-Network COmpression Scheme for Data Collection in Wireless Sensor Networks," in *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN)*, 2003.

[4] *TelosB Datasheet*, Crossbow Technology, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[5] Sun Microsystems Inc., "Project SunSPOT - Sun Small Programmable Object Technology," Online: http://www.sunspotworld.com, 2008.

[6] A. Reinhardt, M. Kropff, M. Hollick, and R. Steinmetz, "Designing a Sensor Network Testbed for Smart Heterogeneous Applications," in *Proceedings of the Third IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, Oct 2008.

[7] G. Sharma and R. Mazumdar, "Hybrid Sensor Networks: A Small World," in *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[8] W. Hu, V. N. Tran, N. Bulusu, C.-T. Chou, S. Jha, and A. Taylor, "The Design and Evaluation of a Hybrid Sensor Network for Cane-toad Monitoring," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005.

[9] G. Wagenknecht, M. Anwander, T. Braun, T. Staub, J. Matheka, and S. Morgenthaler, "MARWIS: A Management Architecture for Heterogeneous Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Wired/Wireless Internet Communications (WWIC)*, 2008.

[10] X. Wang, A. Jiang, and S. Wang, *Advances in Intelligent Computing*. Springer Berlin/Heidelberg, 2005, vol. 3645/2005, ch. Mobile Agent Based Wireless Sensor Network for Intelligent Maintenance.

[11] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting Heterogeneity in Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[12] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks," in *Proceedings of the 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA)*, 2004.

[13] *MICA2 Datasheet*, Crossbow Technology, http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[14] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "IMOTE2: Serious Computation at the Edge," in *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC)*, 2008.

[15] C. M. Sadler and M. Martonosi, "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[16] A. Reinhardt, M. Hollick, and R. Steinmetz, "Stream-oriented Lossless Packet Compression in Wireless Sensor Networks," in *Accepted for Publication in: Proceedings of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2009.

[17] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPUs," in *Cryptographic Hardware and Embedded Systems (CHES)*. Springer Berlin/Heidelberg, 2004.

[18] Texas Instruments Inc., "CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B)," 2007. [Online]. Available: http://www.ti.com/lit/gpn/cc2420