

Power Aware Communication in Wireless Pervasive Smart Camera Networks

Thomas Winkler, Bernhard Rinner

Pervasive Computing Group
Institute of Networked and Embedded Systems
Klagenfurt University
Lakeside Park B02b, 9020 Klagenfurt, Austria
{thomas.winkler, bernhard.rinner}@uni-klu.ac.at

Abstract—Visual sensor networks bring together the research domains of smart cameras and wireless sensor networks. Focusing on power consumption and long-term deployments, typical sensor nodes come with limited computing power and low communication performance. A smart camera that performs on-board video analysis and exchanges data with adjacent nodes has higher computation and communication requirements. Therefore, a main challenge in visual sensor networks is to find a tradeoff between performance and power consumption. This work presents a camera prototype equipped with a low- and a high-performance radio allowing to trade communication performance for power consumption by selecting the appropriate device. Radio power consumption is evaluated for different modes of operation and a power model for clusters of dual-radio cameras is presented.

I. INTRODUCTION AND MOTIVATION

Smart cameras are embedded systems doing on-board image analysis and being capable of delivering high level event descriptions [1]. Over the last few years, a lot of research was performed on wireless sensor networks (WSNs) for e.g. environmental monitoring using countless, cheap devices. The emerging field of Visual Sensor Networks (VSNs) aims to combine concepts from smart cameras and WSNs opening a wide field of novel applications. These include smart homes, elderly care or entertainment. VSNs bear the potential to change our conception of cameras as boxes that capture images into a more general notion of cameras as spatially distributed, pervasive sensors that generate data and events.

One of the challenges on the way to realize this vision is power awareness. Traditional WSNs are designed to collect and process scalar values. The amount of data to be processed by cameras is orders of magnitudes higher. Similar observations apply for networking: WSNs are designed for low-volume, low-power data transmission. Consequently, they are not capable of delivering live images e.g. in an elderly-care scenario where unusual events are reported and the operator needs to evaluate the situation before taking further actions.

In this work, we present an approach of using a dual-radio network for VSNs that allows to find a trade-off between power consumption and efficient and fast transmission of larger amounts of data. The contribution of this work includes the definition of a dual-radio networking architecture, detailed

power consumption measurements of our prototype platform, a power consumption model for dual-radio clusters and a radio selection procedure for our camera middleware framework [2].

The remainder of this paper is organized as follows: After discussing related work in section II, section III outlines hardware and software of our prototype system, our radio architecture and our testbed. In section IV we present radio power consumption measurements followed by considerations for a power aware radio selection procedure as proposed in section V. Section VI concludes the paper and outlines future work.

II. RELATED WORK

Meerkates [3] and Panoptes [4] are early representatives of systems falling into the class of VSNs. They are based on Stargate motes, webcams and PCMCIA WiFi adapters. Both systems are running Linux and the authors discuss the power consumption of individual system components, and highlight the relatively high power consumption for wireless networking.

Using a dual-radio wireless network to conserve power has been investigated by a number of researchers. With CoolSpots, Pering et al. [5] describe a system where devices with WiFi and Bluetooth communication channels are used for web-surfing and file-transfers. The appropriate network link is dynamically selected depending on the current network load. CoolSpots is limited to two communication partners. Reported energy savings go up to 75% depending on the scenario.

In [6] Stathopoulos et al. use a dual-radio platform to implement a system that selectively enables high-bandwidth radios to form end-to-end communication paths. The low-bandwidth network is used to control the high-bandwidth network. The work focuses on development of topology control and routing mechanisms for dual-radio networks.

Lymberopoulos et al. [7] examine the energy efficiency of a platform equipped with a Chipcon CC2420 802.15.4 and a 802.11b radio. They note that the startup times and startup power consumption is considerably lower for the 802.15.4 radio. They conclude that using 802.11 only amortizes for larger amounts of data. The authors also discuss the importance of using appropriate computing platforms for different

radios because low-performance systems are not capable of efficiently handling high-performance radios.

Jung and Savvides [8] not only consider sensor nodes with multiple radios but also multiple processors and sensors. They propose an approach based on a reconfigurable interconnect between the individual system components. They define a Semi-Markov decisions model for switching between high- and low-performance system components thereby significantly increasing system lifetime.

III. SYSTEM ARCHITECTURE

In this section we briefly describe the hard- and software of our camera prototype system followed by an overview of our dual-radio networking architecture.

A. Camera Prototype Architecture

Figure 1 shows our pervasive smart camera platform. The platform is based on the BeagleBoard¹ equipped with an OMAP 3530 processor from Texas Instruments. The processor is based on an ARM Cortex-A8 clocked at 480 MHz and an additional TMS320C64x+ digital signal processor running at 430 MHz. The system provides 128 MB RAM and 256 MB NAND flash. Peripherals can be attached via USB, I2C, SPI, DVI as well as stereo in/out. In our setup, USB is used to connect a Logitech QuickCam S5500 (color, VGA), an RA-Link RA-2571 802.11b/g WiFi adapter as well as a SunSPOT [9] mote providing 802.15.4 wireless connectivity. For development and debugging purposes, the nodes additionally are equipped with USB to Ethernet adapters. As operating

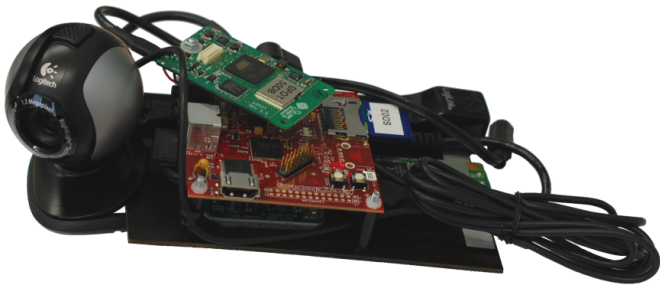


Fig. 1. A Pervasive Smart Camera prototype using an embedded processing board, a webcam, an 802.11 radio and a SunSPOT for 802.15.4 connectivity.

system a Debian GNU/Linux distribution compiled for the ARM platform together with an OMAP specific kernel is used. Without any optimizations, the system currently requires 500 MB of the 8 GB SD card it is stored on.

To support application development, a custom middleware framework has been designed [2] that allows to compose applications from individual blocks which have well defined interfaces for input and output. Besides building applications from locally running blocks, applications can also subscribe to output generated by blocks running on a remote node. This feature is required if multiple cameras work on a common problem such as multi-camera object tracking or when there

is insufficient computing power on a node and the workload needs to be distributed.

B. Dual Radio Network Architecture

In previous work [10], we proposed a dual radio network for inter-camera communication using a low-performance 802.15.4 network for system management and coordination and an 802.11g radio network for bulk transmission. The 802.11g network is intended to be turned off when not needed to conserve power. We now enhance our approach by defining clusters of cameras based on network topology with the aim of simplified network management.

Cluster Definition: A cluster consists of N cameras equipped with high and low performance radios. Additionally, a cluster is characterized as follows:

- 1) The high performance radio is operated in ad-hoc mode and all cluster members are in single-hop radio communication distance to all other cluster members when using the high performance radio channel.
- 2) Using the low-performance radio, all cluster members are reachable in no more than three hops. This should compensate the fact that 802.11 can have a transmission range up to three times larger than 802.15.4 [7].
- 3) One node per cluster is assumed to act as an uplink node for communication with information consumers or other clusters. This node can have additional networking capabilities like Ethernet or another radio.

Figure 2 presents a PSC network testbed as deployed at our institute. The network consists of five cameras with adjacent or overlapping fields of view. The cameras of the testbed form a cluster that currently is statically defined. Each camera node is equipped with a high bandwidth radio to e.g. deliver video streams to a consumer. The low performance multi-hop network is used for inter-node coordination and low-volume data transmission.

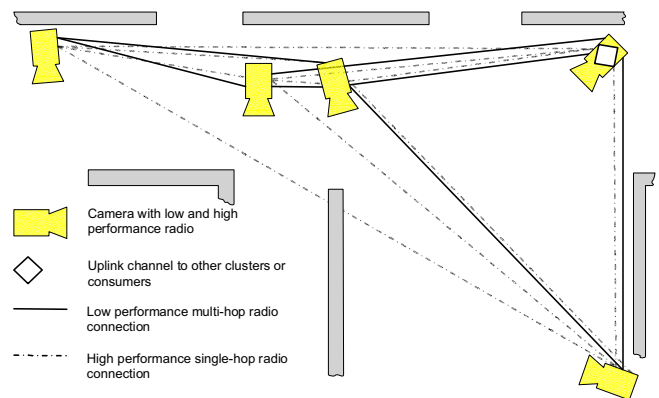


Fig. 2. The networking architecture of the PSC testbed at our institute. The setup consists of five camera nodes equipped with high and low performance radios. For the uplink, Fast Ethernet is used.

IV. DUAL RADIO EVALUATION

As a foundation for a radio selection procedure, this section presents an evaluation of the power consumption of the

¹BeagleBoard Website: <http://www.beagleboard.org> (June 2009)

individual components of our prototype platform.

A. Radio and Platform Power Consumption

For power consumption evaluations, a Measurement Computing USB-1408FS data acquisition tool was used. For the WiFi radio the voltage drop over a 1 Ohm (1%) resistor in the 5V supply line of the RA-2571 USB dongle was measured. To evaluate the power consumption of the CC2420 radio, a modified SunSPOT was used with a 1 Ohm (1%) resistor placed in the 3V supply line of the radio chip.

Figure 3 presents the power consumption of the 802.11g radio for different modes of operation. In idle mode when the radio is waiting for transmissions, average power consumption is 1W. When receiving data, average power consumption is 1.152W and for transmitting data it goes up to 1.391W. The radio can be put into a suspend mode where power consumption goes down to about 0.4W. Resuming from suspend mode requires less than 5ms. Also shown in figure 3 is the full deactivation of the radio by unloading the kernel modules.

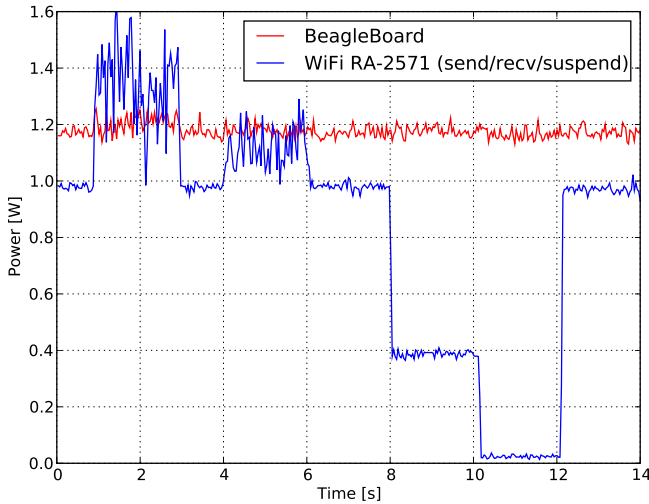


Fig. 3. From second 1 to 3 the WiFi radio is sending data and from second 4 to 6 data is received. At second 8, the radio goes from idle to suspend mode and at second 10, the drivers are unloaded to fully deactivate the radio. Additionally to the WiFi radio, the BeagleBoard power consumption is shown. Accurate power consumption values are given in table I.

Figure 4 shows the power consumption measurements for the CC2420 radio of the SunSPOT. When broadcasting data, average power consumption is 56.52mW, when sending data in unicast mode, power consumption was measured at 83.37mW and in receive mode power consumption is 85.73mW. Having a higher receive than transmit power consumption complies with the CC2420 specification.

All presented measurements are summarized in table I. For completeness we also measured the power consumption of the remaining system components. The BeagleBoard with Linux running on the ARM core and an inactive DSP consumes 1.2W while idle and 1.5W when under load. The USB camera has an average power consumption of 0.6W. The USB hub that connects the individual components consumes 0.34W. We expect that a fully integrated and optimized system could

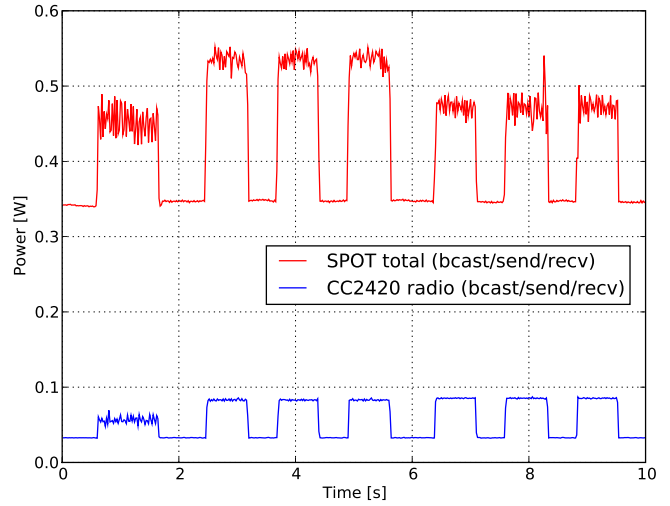


Fig. 4. This figure shows the power consumption of the CC2420 radio and the entire SunSPOT (including radio) for radio broadcast (second 0 to 2), sending three times 5kB of data (second 2 to 6) and finally receiving three times 5kB of data (second 6 to 10). Power consumption values are summarized in table I.

considerably reduce overall power consumption. However, as radio power consumption is the main focus of this work, other components are not considered in the following discussions.

	Description	Abbreviation	Value
802.11g (RA-2571)	Idle	$P_{W_{idle}}$	1012.5 mW
	Receive	$P_{W_{RX}}$	1152.4 mW
	Transmit	$P_{W_{TX}}$	1391.8 mW
	Suspend	$P_{W_{susp}}$	399.3 mW
	Startup (full)	$t_{W_{start}}$	1100 ms
	Startup (from Susp.)	$t_{W_{susp}}$	<5 ms
802.15.4 (CC2420)	Idle	$P_{C_{idle}}$	32.98 mW
	Broadcast	$P_{C_{BC}}$	56.52 mW
	Receive	$P_{C_{RX}}$	85.73 mW
	Transmit	$P_{C_{TX}}$	83.37 mW
	Startup	$t_{C_{start}}$	<1 ms
BeagleBoard	Idle	$P_{B_{idle}}$	1205.6 mW
	Full Load	$P_{B_{load}}$	1500 mW
SunSPOT	Radio Idle	$P_{S_{idle}}$	349.43 mW
	Transmit	$P_{S_{TX}}$	536.94 mW
Webcam	Receive	$P_{S_{RX}}$	475.02 mW
	640x480, 15 fps	P_{cam}	600 mW
USB Hub		P_{hub}	340 mW

TABLE I
SUMMARY OF POWER CONSUMPTIONS OF INDIVIDUAL SYSTEM COMPONENTS. ADDITIONALLY, RADIO STARTUP TIMES ARE GIVEN.

B. Radio Startup Times

Full 802.11 radio startup involves loading the kernel modules and configuring the network. On our prototype platform, this process has been measured to take 1.1 s. One of the main advantages of the 802.15.4 radio is its low startup time which is less than 1 ms.

C. Radio Transmission Times

Table II presents measurements of single-hop data transmission times for the 802.11 and 802.15.4 radios. In case of 802.11, data was sent using iperf² via TCP/IP and on the

²Iperf Website: <http://sourceforge.net/projects/iperf> (June 2009)

SunSPOTs a custom program based on a RadioStream connection was used. The SunSPOTs were programmed using SDK version “blue”. For both radios, measurements are presented for single hop communication. The application level payload of a SunSPOT radio package was measured to be 97 Bytes. For 802.11 we measured a maximum datarate of 16 Mbit/s (2048 kB/s) and over 802.15.4 55.2 kbit/s (6.9 kB/s) could be achieved. This results in a energy consumption of 151.65 nJ/bit for 802.11 and 2979,97 nJ/bit for 802.15.4 which clearly shows that 802.11 is more energy efficient than 802.15.4 for data transmission. The main reasons why using only 802.11 is problematic, are its high startup and idle energy consumptions.

	1 Byte	80 Bytes	100 Bytes	1 kB	2 kB	5 kB	10 kB
802.15.4	8.5	14.4	21.8	148.5	293.6	717.4	1433.8
802.11g	0.004	0.041	0.049	0.472	0.981	2.482	4.879

all values given in [ms]

TABLE II

SINGLE HOP TRANSMISSION TIMES FOR 802.11G AND 802.15.4 ON THE PSC PROTOTYPE FOR DIFFERENT PAYLOAD SIZES.

V. COMMUNICATION CHANNEL SELECTION MODEL

Communication between cameras is based on events. Event content can range from raw data images over intermediate results like extracted features up to high level events such as local decisions to be fused into a global result. The amount of data and the frequency at which events are transmitted depends on the actual event type. For the channel selection model, the following parameters have been selected:

Event Size d_e : The amount of data to be transmitted per event. This could be an uncompressed QVGA color (YUV) image with $d_e = 320 * 240 * 2 \text{ bytes}$ or a high-level decision represented by just a few bytes.

Event Frequency f_e : The number of occurrences of the event per second (e.g. $f_e = 10 \text{ Hz}$ for a video stream with 10 frames per second).

Event Distance in Hops h_e : The number of hops between the event source and the event sink. The hop count is determined at runtime and is only relevant for the 802.15.4 network.

Each application is annotated with its event size d_e and event frequency f_e . Applications can support multiple quality of service (QoS) levels (e.g. video streams at different resolutions and frame rates). Therefore, an ordered list of (d_e, f_e) pairs, each representing a QoS level, can be defined.

A. Low Performance Radio Energy Consumption

The low power 802.15.4 radio is used as an event and paging channel and is assumed to be always enabled. Overall power consumption consists of the power $P_{C_{idle}}$ consumed while the radio is idle and the power consumption when data is transmitted between two nodes (equation 1). Note that this is a simplified model that does not take into account effects such as packet loss or retransmissions.

$$P_{C_{data}} = P_{C_{TX}} + P_{C_{RX}} [W]. \quad (1)$$

For a cluster consisting of N nodes, the baseline energy consumed by the idle low performance radios over an observation

period t_O is given by equation 2. The energy consumption for data transmissions over h_e hops depends on the amount of data d_e in bytes to be transmitted as shown in equation 3 where t_{act} denotes the activity time of the radios. It is assumed that $d_e \leq BW_{C_{avail}}$ where $BW_{C_{avail}}$ denotes the available bandwidth in bytes/second between the communication endpoints.

$$E_{C_{base}} = N * t_O * P_{C_{idle}} [J]. \quad (2)$$

$$E_{C_{data}} = t_{C_{act}} * P_{C_{data}} [J]. \quad t_{C_{act}} = \frac{d_e}{BW_{C_{avail}}} * h_e [s]. \quad (3)$$

The energy consumed by the idle nodes during a multi-hop data transmission is given by equation 4. Note that this model assumes only one active transmission in the cluster at a given time. The total energy consumption of the low performance cluster during data transmission is given in equation 5.

$$E_{C_{idle}} = t_{C_{act}} * P_{C_{idle}} * (N - 2) [J]. \quad (4)$$

$$E_{C_{trans}} = E_{C_{idle}} + E_{C_{data}} [J]. \quad (5)$$

Up to now the fact has been ignored that data transmission in a camera network is not a singular event but typically occurs at a defined rate (e.g. the frequency at which images are processed). This fact is modeled by the event frequency f_e at which the event data d_e is produced resulting in an updated radio activity time t_{act2} shown in equation 6. It now has to be ensured that $f_e * d_e * h_e \leq BW_{C_{avail}}$.

$$t_{C_{act2}} = t_O * \frac{f_e * d_e}{BW_{C_{avail}}} * h_e [s]. \quad t_{C_{idle}} = t_O - t_{C_{act2}} [s]. \quad (6)$$

Total energy consumption of the low performance cluster during transmission of an event consists of the baseline energy consumption 7 for the time where no transmission is going on plus the energy consumption during periodic transmission 8.

$$E_{C_{total}} = t_{C_{idle}} * P_{C_{idle}} * N \quad (7)$$

$$+ t_{C_{act2}} * (P_{C_{data}} + P_{C_{idle}} * (N - 2)) [J]. \quad (8)$$

B. High Performance Radio Energy Consumption

Contrary to the low performance radio, the high performance WiFi radio is off by default and only turned on when required. Consequently, the startup energy consumption as given in equation 9 is of interest. The power consumption during startup was measured to be identical to the radio idle power consumption $P_{W_{idle}}$.

$$E_{W_{start}} = t_{W_{start}} * P_{W_{idle}} [J]. \quad (9)$$

The energy consumption for data transmission between two nodes (eq. 10) is modeled similar to that of the low performance radio with the difference that only single hop communication is used. The total energy consumption given in equation 11 is composed of the startup energy consumption of the two communication partners and the energy consumption for the actual data transmission. As for the low performance radio, effects like packet loss are not considered in this model.

$$E_{W_{data}} = \frac{d_e}{BW_{W_{avail}}} * (P_{W_{TX}} + P_{W_{RX}}) [J]. \quad (10)$$

$$E_{W_{trans}} = 2 * E_{start} + E_{data} [J]. \quad (11)$$

When describing the data transmission as periodic events occurring at a frequency f_e , equations 12 specifies the radio activity and idle times. Finally, equation 13 gives the total energy consumption for data transmission over the high performance radio consisting of the startup energy consumption and the energy consumption during radio activity and idle periods.

$$t_{W_{act}} = t_O * \frac{f_e * d_e}{BW_{W_{avail}}} \quad t_{W_{idle}} = (t_O - t_{W_{act}}) [s]. \quad (12)$$

$$E_{W_{total}} = 2 * E_{W_{start}} + t_{W_{act}} * (P_{W_{TX}} + P_{W_{RX}}) + t_{W_{idle}} * (2 * P_{W_{idle}}) [J]. \quad (13)$$

Note that our current model neither covers pipelining of data transmitted over more than two hops nor overhearing of adjacent transmissions.

C. Radio Energy Consumption Comparison

Based on the radio energy consumption model discussed in the previous subsection and the measurements presented in section IV, we are now comparing the energy consumption of the two radios of our prototype platform.

Figure 5 shows the energy consumption of 802.11 single-hop transmission. Data is sent in blocks of growing size. For every data size, not only the amount of energy for transmission but also the 802.11 startup energy for one or both communication partners, as shown in equation 11, is included.

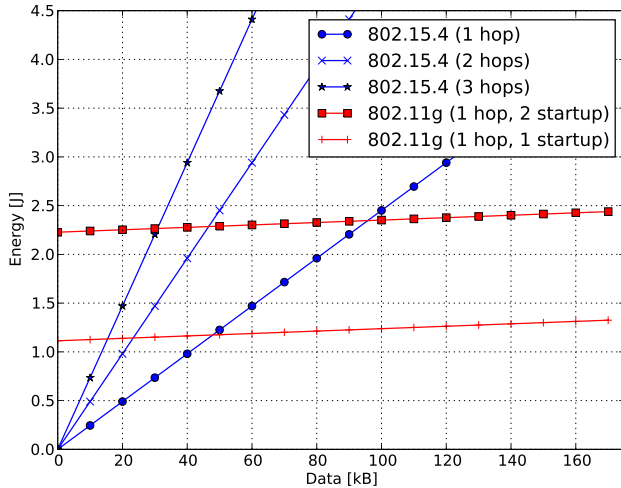


Fig. 5. Energy consumption for single hop 802.11 transmission (incl. startup energy consumption) and up to three hops for 802.15.4. For 802.11, separate plots for starting both or only one communication endpoint are given.

In case of 802.15.4 startup energy consumption is not considered because in our system 802.15.4 is enabled by default. To achieve similar communication distances as 802.11, up to three hops have been considered for 802.15.4. As shown in figure 11, the breakeven points where 802.11 gets more energy efficient when both communication endpoints have to be started up are 96.4 kB (1 hop 802.15.4 communication), 46.94 kB (2 hops) and 31.02 kB (3 hops). If one of the

two 802.11 communication partners are already enabled, the breakeven points drop to 48.2 kB (1 hop), 23.47 kB (2 hops) and 15.5 kB (3 hops).

Figure 6 shows the energy consumption for transmission of an event with $f_e = 1$ and d_e increasing up to the evaluated maximum data rate of the radios. The shown 802.15.4 energy consumption also includes the idle power consumption of nodes not involved in transmission and is given for our testbed cluster size $N = 5$. The first plot for 802.11 shows the energy consumption for single hop transmission between two communication partners (as given in eq. 13 but *without* including the startup energy). After the data transmission is completed, the radios remain in idle mode consuming $P_{W_{idle}}$ for the rest of the communication period ($t_{W_{idle}}$). The second plot shows 802.11 energy consumption if the WiFi radios are switched to suspend after transmission completes. For small amounts of data, $t_{W_{idle}}$ dominates. With increasing data sizes, the activity time $t_{W_{act}}$ grows and both plots are converging towards a power consumption of 2.5 W per second. The figure shows that with our radios, in case of small amounts of data, transmission via 802.15.4 is to be preferred over 802.11. This also holds true when including the idle power consumption of all the 802.15.4 radios of the cluster as done in figure 6.

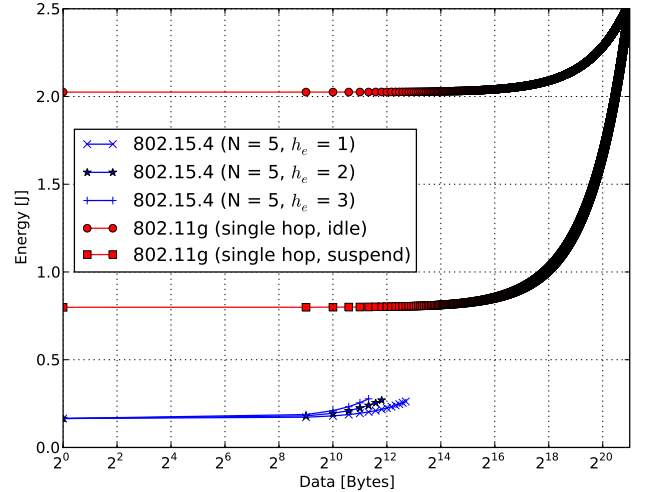


Fig. 6. Energy consumption for a cluster ($N = 5$) of 802.15.4 radios and one data transmission with d_e going up to the maximum 6.9 kB, $h_e = 1 \dots 3$ and $f_e = 1$. 802.11 energy consumption (without $E_{W_{start}}$) is shown for data sizes d_e up to 2048 kB ($f_e = 1$). After transmission is complete, the 802.11 radio either stays idle or is suspended for the rest of the transmission period.

Based on our observations, the following guidelines can be derived for dual radio networking in our smart camera testbed:

- 1) As 802.15.4 bandwidth is limited, it primarily will be used for control data and high level events based on features or decisions. Image delivery is only an option if they are at low resolutions (e.g. regions of interest), heavily compressed or transmission time is not critical.
- 2) Due to the high startup energy consumption of 802.11, 802.15.4 is preferred for transmissions of singular events up to the breakeven points shown in figure 5. Above that, the startup costs for 802.11 are compensated by its higher energy efficiency during transmission.

- 3) As transition times from 802.11 suspend mode to full operation are rather small, whenever possible, the system should suspend the radio when idle. To ensure appropriate service quality, this should be managed by the camera middleware as it, contrary to the OS, is aware of the event schedule of the running applications.

D. Radio Selection Procedure

Our software framework [2] allows to easily start applications at runtime to dynamically adapt the system configuration. As part of this process the system should make a decision, according to a predefined model, which communication channel to use. This decision is based on the requirements of the application and the available communication resources. As applications can support multiple quality of service levels, it is possible on application startup to specify a minimal required QoS level Q_{min} . To estimate the available bandwidth, the middleware on each node has to keep a record of the bandwidth consumption of the running applications. For our system we propose the following channel selection procedure:

- 1) Determine the number of hops h_e and the available bandwidth $BW_{C_{avail}}$ between the communication partners on the low performance network.
- 2) If the amount of data to be sent at Q_{min} can be transmitted via the low bandwidth radio ($d_e * f_e * h_e \leq BW_{C_{avail}}$) proceed to step 3, else continue with step 7.
- 3) Check if the high performance network is already enabled at the source and the destination node. If yes, proceed to step 7. Note that the high performance channel offers less latency and is more efficient per transmitted bit. So it is the preferred choice when both endpoints already have it enabled.
- 4) Compute energy consumption $E_{C_{total}}$ for data transmission via the low performance network according to equation 8 for all QoS levels feasible for $BW_{C_{avail}}$ that are $\geq Q_{min}$.
- 5) Compute the WiFi energy consumption $E_{W_{total}}$ for all QoS levels $\geq Q_{min}$ according to equation 13. Beforehand, check if any of the two high performance radios is already active such that $E_{W_{start}}$ can be computed correctly.
- 6) If $E_{C_{total}} < E_{W_{total}}$ use the low performance network for data transmission. The setup of the communication channel is now complete and all following steps can be ignored.
- 7) Via the low performance paging channel check if the high performance network interface is already active at source and/or destination and enable it where required.
- 8) Check the available bandwidth of the high performance communication channel and select a QoS level equal or greater than Q_{min} . If Q_{min} can not be met, return an error. Otherwise, the channel setup is complete.

VI. CONCLUSION AND FUTURE WORK

In this work we presented a visual sensor network platform equipped with a low- and a high-bandwidth radio. After defining a clustering schema based on radio topology, we presented detailed power consumption measurements of our prototype system with a focus on the radios and their different operating modes. A main part of this work is the definition of an energy consumption model for the proposed camera cluster architecture. In this model, the 802.15.4 radios are used for transmission of control and low-volume data while 802.11 is used for bulk transmissions. We evaluated the breakeven points where 802.11 becomes more energy efficient than 802.15.4 and discussed the high energy consumption of 802.11 during startup and idle periods. Having a mechanism that allows to

quickly disable and enable WiFi, would make 802.11 much more attractive for use in VSNs. In this context, we plan to evaluate power consumption and startup times of other WiFi radios. We conclude our evaluations with guidelines to choose an appropriate radio and propose an algorithm for automatic radio selection on application startup.

A. Future Work

In ongoing work, we are integrating the proposed radio selection mechanism into our middleware framework to facilitate extended evaluations on our testbed. In the future, we plan to address the following open issues: (1) Additionally to inter-cluster communication over a backbone network, we also want to investigate logical clustering. Cameras in different network clusters might still have an overlapping field of view and should be grouped into a logical cluster. (2) Duty-cycling the low-performance radio could help to further reduce power consumption compared to the “always on” approach of our current model. (3) Right now, only the radios are considered for power saving. A more holistic approach could also take into account e.g. the processor and image sensor. They could be suspended if there is no activity in the field of view of the camera and woken up by adjacent cameras as required.

VII. ACKNOWLEDGEMENTS

We thank Robert Alkire from Sun Microsystems for providing the modified SunSPOT we used for our measurements.

REFERENCES

- [1] B. Rinner and W. Wolf, “Special Issue on Distributed Smart Cameras,” *Proceedings of the IEEE*, vol. 96, no. 10, Oct. 2008.
- [2] W. Schriebl, T. Winkler, A. Starzacher, and B. Rinner, “A Pervasive Smart Camera Network Architecture applied for Multi-Camera Object Classification,” in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2009, (to appear).
- [3] C. Margi, V. Petkov, K. Obraczka, and R. Manduchi, “Characterizing energy consumption in a visual sensor network testbed,” in *Proceedings of the International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM)*, 1–3 March 2006, p. 8pp.
- [4] W.-C. Feng, E. Kaiser, W. C. Feng, and M. L. Baillif, “Panoptes: Scalable Low-Power Video Sensor Networking Technologies,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 2, pp. 151–167, 2005.
- [5] T. Perring, Y. Agarwal, R. Gupta, and R. Want, “CoolSpots: Reducing the Power Consumption of wireless mobile Devices with multiple Radio Interfaces,” in *Proceedings of the International Conference on Mobile Systems, Applications and Services*, 2006, pp. 220–232.
- [6] T. Stathopoulos, M. Lukac, D. McIntire, J. Heidemann, D. Estrin, and W. Kaiser, “End-to-End Routing for Dual-Radio Sensor Networks,” in *Proceedings of the Int. Conference on Computer Communications*, 2007.
- [7] D. Lymberopoulos, N. B. Priyantha, M. Goraczko, and F. Zhao, “Towards Energy Efficient Design of Multi-radio Platforms for Wireless Sensor Networks,” in *Proceedings of the International Conference on Information Processing in Sensor Networks*, 2008, pp. 257–268.
- [8] D. Jung and A. Savvides, “An Energy Efficiency Evaluation for Sensor Nodes with Multiple Processors, Radios and Sensors,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 13–18 April 2008, pp. 439–447.
- [9] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, “Java on the Bare Metal of Wireless Sensor Devices,” in *Proceedings of the International Conference on Virtual Execution Environments*, 2006.
- [10] T. Winkler and B. Rinner, “Pervasive Smart Camera Networks exploiting heterogeneous wireless Channels,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2009, pp. 296 – 299.