Low-Power Image Transmission in Wireless Sensor Networks using ScatterWeb Technology

Enrico Köppe, Achim Liers, Hartmut Ritter, Jochen Schiller
Institute of Computer Science
Freie Universität Berlin, 14195 Berlin, Germany
[koeppe, schiller, liers, hritter]@inf.fu-berlin.de

Abstract

ScatterWeb, a distributed, heterogeneous platform for the ad-hoc deployment of sensor networks, offers hardware and open, fully documented software for the deployment of embedded sensor networks. Already low power by design, the sensor nodes offer additional energy conservation mechanisms and support energy efficient routing, such as, e.g., solar aware routing. Due to advances in CMOS technology, ScatterWeb nodes can now also capture and transmit images at low cost. Depending on the transmission strategy either lower latency or lower power consumption can be achieved. However, both strategies allow for the capturing and transmission of several thousand images using a single AA battery. Together with additional power saving, auto-configuration, and remote reprogramming techniques, these mechanisms enable ScatterWeb nodes to survive many years in real-life scenarios without any in-place maintenance.

1. Introduction

Wireless sensor networks are an information gathering paradigm based on the collective efforts of many small wireless sensor nodes. The sensor nodes, which are intended to be physically small and inexpensive, are equipped with one or more sensors, a short-range radio transceiver, a small micro-controller, and a power supply in the form of a battery or mechanisms that try to scavenge energy from the environment. Sensor network deployments are envisioned to be done in large scales, where each network consists of hundreds or even thousands of sensor nodes. In such a deployment, human configuration of each sensor node is usually not feasible and therefore self-configuration of the sensor nodes is important. Energy efficiency is also critical, especially in situations where it is not possible or at least very expensive to replace sensor node batteries.

Most sensor network applications aim at monitoring or detection of phenomena. Examples include office building environment control, wild-life habitat monitoring, and forest fire detection [10], [15]. For such applications, the sensor networks cannot operate in complete isolation; there must be a way for a monitoring entity to gain access to the data produced by the sensor network. By connecting the sensor network to an existing network infrastructure such as the global Internet, a local-area network, or a private intranet, remote access to the sensor network can be achieved.

Many research projects focus on routing in wireless sensor networks as this has to take into account the very limited resources of the nodes. While it is quite often assumed that all nodes in a sensor network are battery-driven [1], [8], [16], [17] nodes can also be powered by other energy sources such as vibration, temperature differences, or solar power. Nodes powered by such a source can receive and transmit packets without consuming battery energy. Therefore, routing packets via such nodes is appealing [14]. It is, however, complicated by the fact that the environmental energy source is quite often not permanent [9].

Although there has already been done a lot of research in the area of sensor networks, most of it relies on simulations. We believe in a sound combination of simulations and real hard- and software. This is the only way to really investigate into the important topics of energy aware application design, energy aware protocols and routing. In our lab, we have developed sensor boards which are low power by design and can even be powered by, e.g., solar cells. Furthermore, we designed several gateways to bridge sensor networks with common wired or wireless networks. The newest component in ScatterWeb is a low cost camera module that can transmit images with very little power consumption.

The key contributions of this paper are the extension of ScatterWeb, an open and flexible sensor network platform, to integrate camera modules and the comparison of different transmission strategies with respect to their power consumption.

The paper is organized as follows. Section 2 gives a short overview of the ScatterWeb platform and its hard- and software. Section 3 describes basic power conservation mechanisms implemented in ScatterWeb's sensor nodes, while section 4 discusses the integration of camera modules and the transmission of image data.

2. ScatterWeb

ScatterWeb [13] offers a distributed, heterogeneous platform for the ad-hoc deployment of sensor networks. The platform comprises both hardware and open, fully documented software. Figure 1 shows a typical example for a sensor network based on ScatterWeb. Many sensor nodes, called ESB (Embedded Sensor Board, section 2.1), are distributed in the environment for monitoring parameters such as temperature, vibration, light etc. The sensor nodes may forward their data via other sensor nodes towards more powerful data sinks. ScatterWeb uses so-called embedded web servers (EWS, section 2.2) as gateways between the sensor world and classical wired or wireless networks (Ethernet, Bluetooth, GPRS …). ESBs as well as EWSs can be deployed in an ad-hoc manner and after that the nodes perform automatic topology discovery and auto-configuration. The backbone of the sensor network, set-up between the EWSs, can be wireless or wired. Section 2.2. motivates the usage of simple web servers for this purpose.

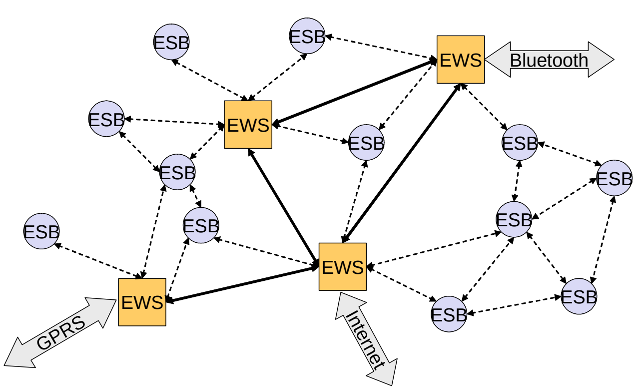

Figure 1. ScatterWeb network example

2.1. Embedded Sensor Board (ESB)

As the name already implies, sensor networks need sensors, typically hundreds or thousands of sensors, to collect data from their environment. All sensor nodes may form an ad-hoc network with some nodes acting as data sources, some as relays, and some as data sinks collecting data. Nodes may act in all three roles at the same time. Typical communication scenarios of sensor networks based on the ESBs are:

- ESBs communicate via the serial port or USB with a standard PC/PDA for application development.
- ESBs communicate with GSM/GPRS modules to connect to wide-area mobile phone networks. This enables remote configuration of ESBs even via short messages (SMS) as well as reception of sensor data on mobile phones world-wide.
- ESBs communicate via their radio interface with other ESBs or an embedded web server (EWS) to form a truly embedded, highly flexible sensor network solution.

A full featured ESB as used for prototyping, research, and teaching (see Figure 2), comprises besides a controller and transceiver the following components: Luminosity sensor, noise detection, vibration sensor, IR movement detection, precise timing, microphone/speaker, IR sender/receiver.

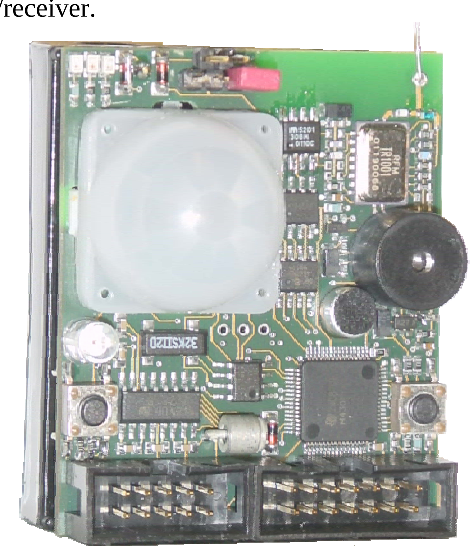

Figure 2. Full featured embedded sensor board

Sensors should follow the principle "embed and forget", i.e., they must have an extreme low power design. Typical average values for a full featured ESB are:

- ESB up and running with all sensors: 12 mA
- ESB transmitting data: less than 8 mA (avg.)
- ESB deep-sleep (clock only): 8 µA

Example scenarios with standard AA batteries (life-time varies depending on the sensors used):

- 5 years life-time with a duty-cycle of 1% (sensing & transmitting)
- 17 years life-time sending 25 byte every 20s

Surely, no normal battery will live that long and, thus, these values are rather theoretical. Due to its modular design it is relatively simple to derive other modules from the ESB as shown in Figure 2. Figure 3 shows an almost radio-only module – it still contains a temperature and vibration sensor and is used in monitoring intelligent buildings.

Other nodes designed within the ScatterWeb project comprise combined sensor/actor nodes for soccer playing robots, nodes for temperature measurement in the Baltic Sea, or nodes for the protection of jewelry.

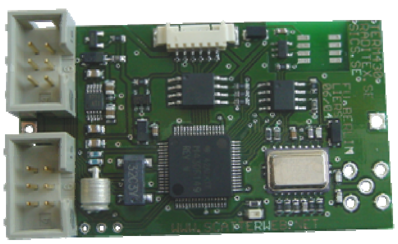

Figure 3. Low-power temperature sensor

2.2. Gateways and programming

A very convenient access to a ScatterWeb is the ScatterFlasher (see Figure 4). A simple USB stick connects a PC to the sensornet and enables over-the-air flashing of all sensors, debugging, collecting sensor data etc. The stick installs as USB device on the PC/PDA. This means, that no more cables are needed for the reconfiguration of the sensor network. No matter how many nodes have to be reprogrammed, the new software is distributed throughout the network; the nodes check the new software for correctness and then flash themselves. The firmware used for flashing is protected and, thus, no node is rendered useless even if the new software is faulty. In case of faulty software (e.g., endless loops, crashing software) the hardware can flush the defect software and waits for a new download.

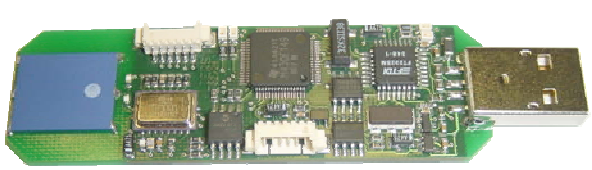

Figure 4. ScatterFlasher for over-the-air programming

The software generated by the compiler is fragmented into smaller packets for downloading over the air. This is done to minimize packet loss due to bit errors. If a node misses a packet due to interference, retransmission is requested locally first. With a very high probability neighbors received the missing packet(s) and can perform the retransmission without involving the gateway. We use simple web servers (EWS, embedded web server) as gateways to the networks outside the sensornet. The main reason for this is the simplicity of access to the sensor network. No matter which network is used for accessing the sensor network, sensor data is provided as simple web page. This allows for a plethora of devices accessing the sensed data as long as a web browser is running on the device. A very typical scenario is reading the temperature values of rooms in a building with a PDA connected by Bluetooth with the EWS collecting data from all sensor nodes. However, the web servers are not only used for reading values but also for setting up an (ad-hoc) infrastructure for the sensor network. Typically, an EWS is connected to power and not battery driven. Therefore, it can use more power for radio transmission and, thus, cover longer distances. The EWSs provide also the gateway for reprogramming the sensor nodes in addition to the direct access via the ScatterFlasher. Reprogramming can be done (after authentication) from arbitrary hosts in the Internet by accessing the web server and downloading new software. Figure 5 shows an EWS that bridges Ethernet and a sensor network. Power-over-Ethernet provides the energy necessary for the embedded ARM processor and the transceiver circuitry. Besides the standard Internet protocols the module performs auto-configuration with the help of DHCP and offers complete web pages and applets for accessing and configuring the sensor network.

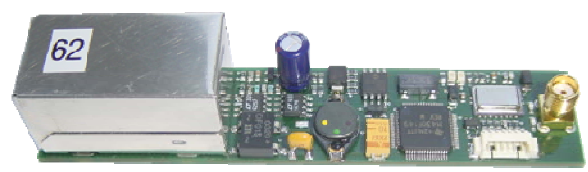

Figure 5. Embedded Web Server (EWS) connecting the sensor network with Ethernet

All components are easy to access via an open and fully documented C interface [13]. The platform provides software for the following features:

- Configuration and analysis of all sensor data.
- Control of peripherals: timer, transceiver, serial port, EEPROM, IR-send/receive, LEDs, beeper, switches.
- Sending and receiving of data packets via the radio interface, transmission of sensor data, communication with other systems using the same radio interface (ESBs, EWSs).

- Sensing and receiving of short messages (SMS) via a GSM module/mobile phone. Just send the ESB an SMS triggering temperature monitoring every Tuesday morning from 7:00 a.m. until 11:43 a.m., if and only if there is movement in the room but no light - it's possible with a single SMS (or also via GPRS).
- Sending (e.g. to control home entertainment devices) and receiving (e.g. from conventional remote controls) of RC5 packets via infrared.
- Periodic polling of sensor data and (depending on settings) automatic transmission via the radio interface, the serial port (terminal or mobile phone/GPRS) or storage into the ring buffer of the local EEPROM.
- Simple and easy configuration and control of the ESB via terminal commands over the serial interface or a mobile phone.
- The standard TCP/IP protocol suite is available for ScatterWeb [5] including a simple web server directly running on the sensor node.

3. Energy conservation

Wireless sensor networks always have to take into account the very limited energy resources of the nodes compared to their wired counterparts. While many projects assume that all nodes in a sensor network are battery-driven [1], [8], [16], [17], nodes can also hoard energy by scavenging their environment [6], [12]. Typical environmental energy sources are solar power, vibrations, temperature differences, or mechanical forces (opening doors, flipping light switches etc.). While energy collected from the environment is considered free, the nodes still have to manage the scarce amount of energy stored, e.g., in capacitors, in similar ways it has to do for battery power.

First of all, ScatterWeb's sensor nodes are low power by design. We use TI's MSP430 as micro controller that requires less than 2 µA in deep-sleep mode. The sensors can interrupt and wake-up the system without any polling required. The controller offers several low power modes by activating/deactivating certain function blocks, e.g., the A/D converters.

For many sensor node simulations it is assumed that sending is always much more expensive in terms of power consumption than receiving. This is not necessarily the case as Figure 6 and Figure 7 show. One version of the sensor board uses very simple 868 MHz transceivers (RFM TR1001, [11]). Figure 6 shows a typical sending cycle. The circuitry wakes up from a low-power mode and wakes up the transceiver. After some time the transceiver circuits need for stabilizing, sending of data starts (first sync patterns then data) consuming less than 8 mA on average. Figure 7 shows that receiving data is similar expensive with 7 mA. Therefore, using a transceiver like we do changes many simulation scenarios that rely on the fact that sending is always much more expensive. Clearly, sending power consumption very much depends on the output power requested at the antenna. With the simple transceiver as shown we can reach approx. 300m in open space and 30m inside buildings. The newer nodes using more advanced transceivers (e.g., CC1021 from Chipcon [3]) offer roughly ten times the range (e.g., transmission through 15 floors in a building is possible).

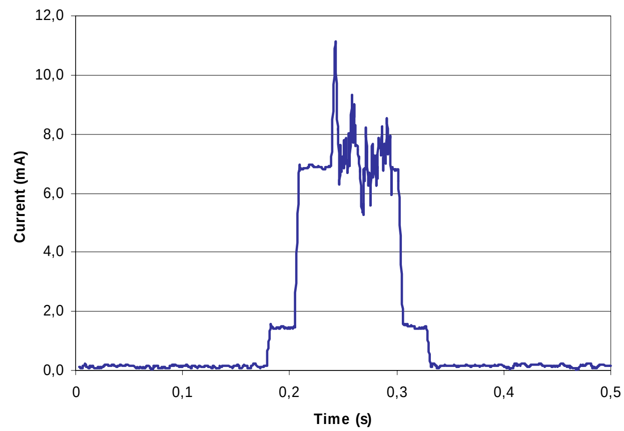

Figure 6. Power consumption: sending

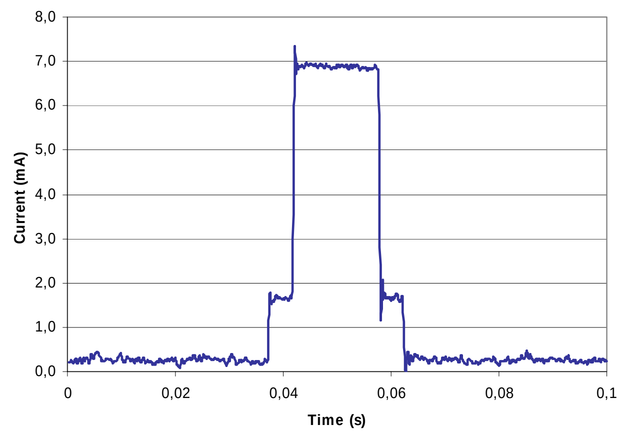

Figure 7. Power consumption: receiving

A typical cycle for a sensor is the following: deep sleep for more than 90% of the time, waking up and start sensing the environment, letting the values stabilize, switch off the sensors and start transmitting the sensed values. Figure 8 shows such a typical cycle. 200ms are typically more than enough to sense and transmit data plus receiving a short acknowledgement.

No matter how low-power the design is, energy is always limited on sensor nodes, particularly if they run on scarce environmental energy sources. It is crucial for reliable operation that nodes automatically switch to low-power modes before running out of energy or even perform a safe shut-down if all energy is consumed. Furthermore, power-up must be performed in a way that does not leave the node in a useless state.

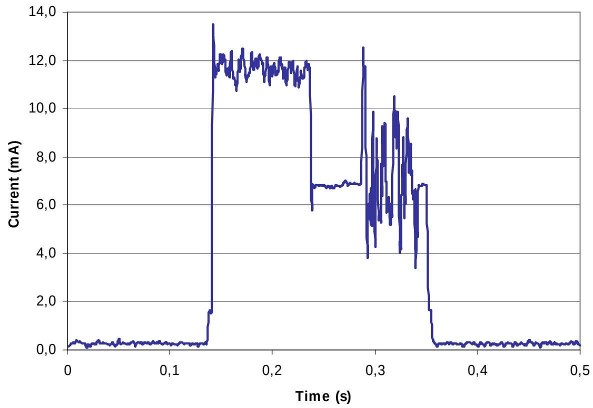

Figure 8. Power consumption: sensing/sending

ScatterWeb's sensor nodes come with an elaborated power sensing mechanism that allows operation with batteries or fluctuating environmental power sources. As soon as enough energy is available to run the controller, the nodes perform periodic power sensing and start the transmission of data if and only if the energy stored in a capacitor is sufficient for the complete transmission plus the reception of an acknowledgement. Also the period of transmitting sensor values depends on the available energy.

Figure 9 shows the (simulated) behavior of a power source with a first increasing then again decreasing voltage. The (measured) behavior of an ESB is as follows (input current shown). As soon as the voltage reaches a certain level, the circuitry performs a clean reset to enter a well-known state. After that, the sensor node waits until the available voltage is high enough to start periodic sensing of the stored energy. This is done by periodically drawing a certain current and measuring the behavior of the energy source (short peaks in the lower line). If the stored energy seems to be large enough, the node starts sending (high peaks). If the voltage drops again the node enters the periodic checking mode again. If the voltage further drops below a certain threshold the node switches itself off (final peaks in the lower line).

We use gold-cap capacitors for storing energy and solar cells, piezo crystals, or thermo-elements for energy scavenging. With the 1 F capacitors we use we can, e.g., perform over 420 sensing and sending cycles without any additional power supply. Thus, the stored energy in such a capacitor is more than enough to keep the sensor alive over night for typical monitoring applications.

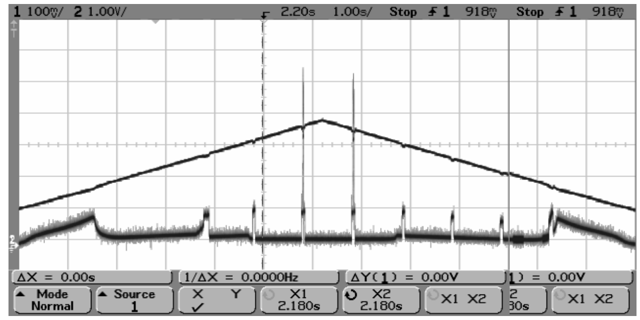

Figure 9. Adaptive power management

4. Low-power image transmission

Up to some years ago, no one would have given a thought to low-cost, low-power image transmission. However, due to the tremendous success of mobile phones with built-in cameras and the advances in CMOS technology the price for low-power VGA-resolution cameras is continuously dropping. Thus, it seems quite natural to include image capturing and transmission capabilities in low-cost sensor networks as monitoring an environment quite often also requires a visual impression. One hurdle was the integration of camera modules plus the image compression software needed to produce compressed images for storage and transmission. For our sensor nodes including a camera we use the C328-7640 VGA camera modules from COMedia [4]. The board as shown in Figure 10 comprises a VGA camera chip, a JPEG compression engine, and a serial interface. Via the serial interface, users can send out a snapshot command in order to capture a full resolution still picture. The picture is then compressed by the JPEG compression engine and transferred back via the serial interface. The whole board operates at 3.3 V and measures 20x28 mm. The maximum resolution is 640x480 pixels; however, the module can be configured to downsample the image to 80x64, 160x128, or 320x240 pixels. The transfer rate over the serial interface is 115.2 kbit/s. The built-in color conversion hardware provides 2/4/8 bit gray scale or 12/16 bit color images. An additional interesting feature of the module is the capability to stream a 160x128/8 bit preview video with 0.75-6 frames per second.

We attached the camera module to our ESB boards via a simple serial interface running at 115.2 kbit/s. Figure 11 shows sample pictures at different resolutions together with the resulting image size taken with an ESB430 module facing a mirror. Up to now we have only built prototypes and we are currently designing new sensor nodes integrating the camera module.

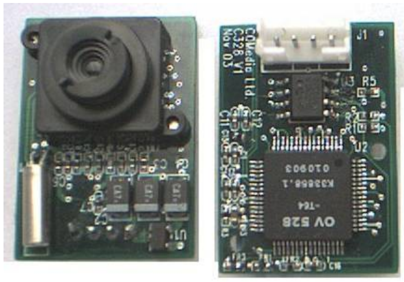

Figure 10. Camera module from COMedia [4]

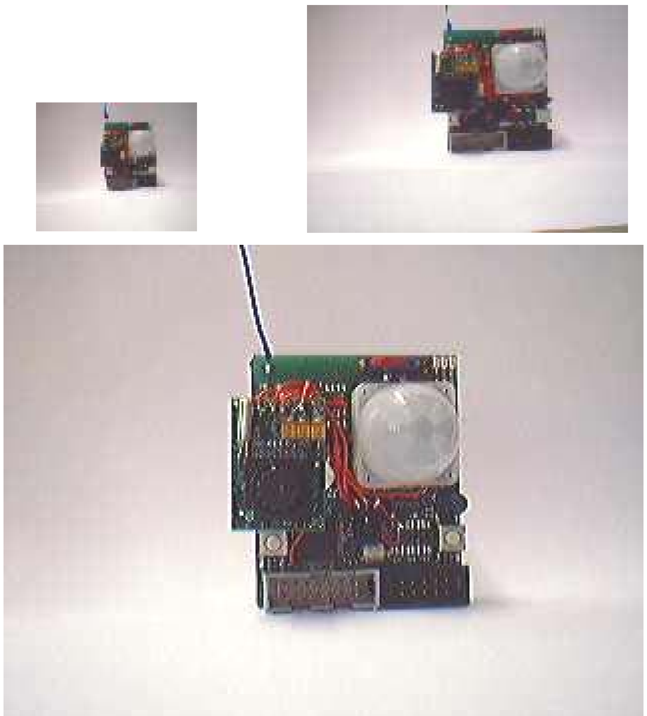

Figure 11. Sample images taken from the prototype ESB/Camera module facing a mirror (80x64 pixel, 1072 byte; 160x128 pixel, 2180 byte; 320x240 pixel, 6304 byte)

An active camera module consumes 50 mA at 3.3 V. Our software allows for the activation and deactivation of the camera module. The power consumption of the camera module drops to 100 µA in power down mode. Using our ESB or ECR sensor nodes we can even further save energy by switching off the complete camera module by simply switching off the power supply. However, this requires a short initialization phase (1 s at 52 mA) every time the camera module is switched on to set certain parameters (image size, color depth etc.). Depending on the frequency pictures are taken either the power down mode at 100 µA or switching the module completely off might be better. We think that in typical sensor network scenarios the second alternative is more common: a sensor detects movement, switches on the camera module, takes a picture, and transmits it via other sensors to a data sink. If the movement persists the sensor may continue taking pictures.

As we connect the camera module over a serial interface with 115.2 kbit/s, downloading compressed pictures from the camera to the sensor board takes only about 2 s even at the maximum resolution. JPEG compression on board of the camera module results in picture sizes of 20-30 kbyte for 640x480/16 bit. Before transmission the sensor node packetizes the image data. Each packet contains up to 64 byte image data and 8 additional bytes for the header. Figure 12 shows the packet format chosen for transmission. Destination and source fields represent the receiver's and sender's node ID, respectively. The type field can be used to signal the packet type (last packet in sequence, expedited forwarding etc.). The sequence number simply counts packets for reassembly. The length field indicates the length of image data in the packet that is currently limited to 64 byte to allow for better time-multiplexing of transmitting sensor nodes (max. 64 kbyte).

2	2	1	1	2	1-64
dest	source	type	seq	length	data

Figure 12. Packet format used for image transmission

The data packets are relatively short and the sensor nodes avoid sending a permanent data stream. Applying a CSMA/CA scheme in the sensor network gives other nodes still a chance to transmit sensor data. Additionally, data transmission in the 868 MHz ISM band has to obey certain regulations (similar to the 915 MHz ISM band in the US that only requires another pin-compatible transceiver module). The CEPT/ETSI regulations for short range devices (SRD) using 868-868.6 MHz (Class 1f) allows for a max. output power of 25 mW and a duty cycle of less than 1% ([7]; for the US 915 MHz ISM band see FCC Code of Federal Regulations, Title 47, Section 15.249). Table 1 shows measured sample durations for image transmissions at 19.2 and 115.2 kbit/s. A typical 640x480 pixel image results in 20-30 kbyte of compressed data and, thus, requires 320-480 data packets.

Table 1. Image size vs. transmission duration

Image size [pixel]	Image size [byte]	Packets	Time [s] at 19.2 kbit/s	Time [s] at 115.2 kbit/s
80x64	1072	17	0.637	0.106
160x128	2180	35	1.312	0.218
320x240	6304	99	3.712	0.618
640x480	21161	331	12.412	2.068

Following the advisory [2] to facilitating sharing between systems in the same frequency band, a transmitter should not exceed 3.6 s "on" time with a minimum of 1.8 s "off" time. We can easily follow this advice by inserting the necessary gaps. This increases the transmission time of the 640x480 pixel image from 12.4 to 19.8 s (4 data bursts plus 3 gaps).

There are two basic strategies for image transmissions implemented on our sensor nodes that either offer lower latency or lower power consumption.

4.1. On-the-fly image transmission

The sensor nodes are able to read the data packet-wise from the camera module and send the data directly without buffering. Sending and reading can be performed simultaneously. This solution does not require additional memory and enables a receiver to immediately starting the rendering process of the picture. Thus the latency is quite low. However, this solution exhibits two drawbacks. As we send packet per packet directly after reading from the camera we do not perform any retransmission in case of packet loss. If an application requires permanent monitoring this does not pose a real problem as image data is sent continuously. Figure 13 shows the second problem: power consumption. As we typically transmit data only at 19.2 kbit/s to achive robust transmissions, the transmission speed of the camera is 6 times higher (115.2 kbit/s). However, as we continuously read out data from the camera the module must stay active during the complete transmission cycle.

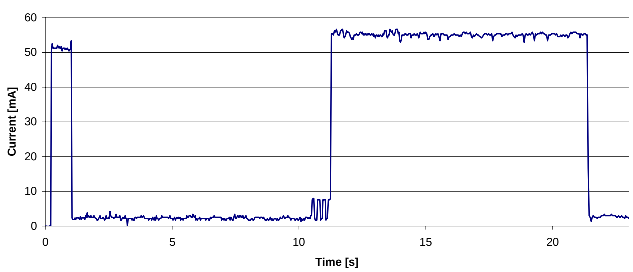

Figure 13. Power consumption of the camera module with on-the-fly transmission (plus initialization)

Figure 13 shows in the beginning the initialization cycle needed after start-up of the camera (setting of camera parameters). An active ESB together with a powered-down camera module consumes about 3 mA (100 µA for the camera). After 11 s we transfer commands to the camera module in order to switch it on. This lets the power consumption jump to 56 mA. The current stays that high during the whole reading and transmitting cycle. In this example we transmit a 640x480 picture which requires about 10 s. Thus, the whole cycle of image reading and transmitting consumes almost 0.16 mAh. Assuming a typical rechargeable AA battery with a capacity of 2000 mAh and a usable capacity of 80% an ESB using the on-the-fly method can transmit about 10000 pictures.

4.2. Two-phase image transmission

The second solution implemented on our sensor nodes transfers the complete image from the camera module to the built-in 64 kbyte EEPROM first and then starts transmitting the image over the air (it is possible to use larger EEPROMs if required). The disadvantages of this solution are the need for a buffer and the higher delay before the first image data reaches a receiver. However, this solution allows for retransmissions of lost image data and saves a lot of energy. Figure 14 shows a typical cycle of image transmission. First of all, the sensor node commands the camera to start the image transfer into the EEPROM. This transfer takes about 2.5 s and consumes about 56 mA. After that, the sensor node puts the camera module in low-power mode or switches the module off. Then the transmission takes place at 7 mA and takes about 9.6 s. This results in an over all power consumption of 0.058 mAh per transmitted picture. Assuming the same battery as in the first scenario a sensor can now transmit about 27500 pictures. Therefore, the second solution is about 2.7 times more efficient than the first solution. As sending 20-30 kbyte images requires about 12-17 s between 3 and 5 pictures per minute are possible to capture and to transmit (19-24 s, thus 2-3 pictures, with gaps as advised in [2]).

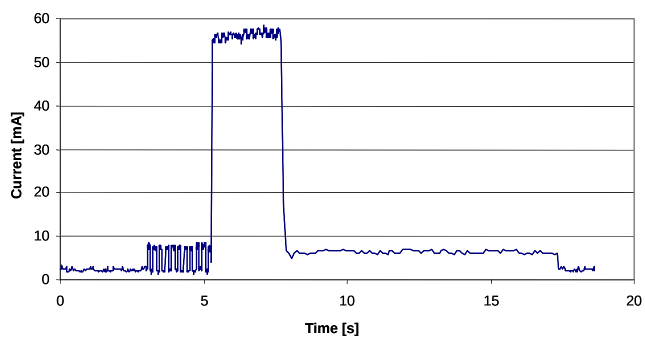

Figure 14. Power consumption of the camera module with two-phase transmission

5. Conclusion and outlook

This paper introduced ScatterWeb, an open and flexible platform for implementing sensor networks. Major goals of ScatterWeb are the design of low-power sensor nodes, the simple management of large-scale sensor networks, and the seamless integration into other wired or wireless networks by using Internet protocols and by providing several gateways. Furthermore, ScatterWeb serves as platform for the implementation of energy-aware routing algorithms [14]. This allows for the derivation of real-world parameters for simulation models. Due to advances in technology, even the integration of low-cost camera modules in sensor networks is possible. We demonstrated not only the feasibility but showed also the trade-off between on-the-fly and two-phase transmission of image data. With the help of ScatterWeb's low power design, sensor nodes are able to capture and transmit more than 25000 pictures using a single AA battery.

Currently, we compare the usage of new, faster transmitters allowing for higher transmission speeds with slower, but more robust transmission. Additionally, we extend the features of the sensor nodes depending on project requirements: special modules for energy scavenging, new RF transceivers, voice output and speech recognition etc. Furthermore, we extend the existing software by applications for localization, for integration into building management systems, or for graphical programming of sensor networks.

6. References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: a survey", *Computer Networks*, 38:393-422, 2002.

[2] CEPT/ERC recommendation 70-03 relating to the use of short range devices, CEPT/ERC/REC 70-03 E, 1999.

[3] Chipcon AS, www.chipcon.com, 07-2004.

[4] COMedia Ltd., Hong Kong, China, www.comedia.com.hk, 07-2004.

[5] A. Dunkels, J. Alonso, T. Voigt, H. Ritter and J. Schiller, „Connecting Wireless Sensornets with TCP/IP Networks", Wired/Wireless Internet Communications, Second International Conference, WWIC 2004, Frankfurt/Oder, Germany, 2004.

[6] EnOcean GmbH, www.enocean.de, 07-2004.

[7] ETSI, Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW, European Telecommunications Standards Institute, ETSI EN 300 220, 2000.

[8] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", International Conference on System Sciences, Hawaii, USA, 2000.

[9] A. Kansal, D. Potter, M. Srivastava, "Performance Aware Tasking for Environmentally Powered Sensor Networks", Joint International Conference on Measurement and Modeling of Computer Systems, ACM/IFIP Sigmetrics – Performance 2004, New York, USA, 2004.

[10] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless sensor networks for habitat monitoring", First ACM Workshop on Wireless Sensor Networks and Applications (WSNA 2002), Atlanta, USA, 2002.

[11] RF Monolithics, Inc., www.rfm.com, 07-2004.

[12] S. Roundy, P. Wright and J. Rabaey, Energy Scavenging for Wireless Sensor Networks, Kluwer Academic Publishers, Boston, ISBN 1-4020-7663-0, 2003.

[13] The ScatterWeb project, www.scatterweb.net, 07-2004.

[14] Voigt, T.; Ritter, H.; Schiller, J.: "Solar-aware Routing in Wireless Sensor Networks",
Personal Wireless Communications, PWC 2003, Venice, Italy, 2003

[15] A. Willig, R. Shah, J. Rabaey and A. Wolisz, "Altruists in the PicoRadio Sensor Network", Factory Communication Systems, Västerås, Sweden, 2002.

[16] F. Ye, G. Zhong, J. Cheng, S. Lu and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks", International Conference on Distributed Computing Systems, Rhode Island, USA, 2003.

[17] Y. Zhao, R. Govidan and D. Estrin, "Residual Energy Scan for Monitoring Sensor Networks", Wireless Communications and Networking Conference, Orlando, USA, 2002.