Integrating Building Automation Systems and Wireless Sensor Networks

Fredrik Österlind, Erik Pramsten, Daniel Roberthson, Joakim Eriksson, Niclas Finne, Thiemo Voigt
Swedish Institute of Computer Science
{fros,epr,dro,joakime,nfi,thiemo}@sics.se

Abstract

Building automation systems (BAS) are used to control and improve indoor building climate at reduced costs. By integrating BAS with wireless sensor networks, the need for cabling can be removed, and both installation and operational costs significantly reduced. Furthermore, temporary BAS installations are made possible. By implementing and evaluating the open BAS standard BACnet on resource-constrained sensor nodes we show that integrating existing standard BAS protocols with wireless sensor networks is feasible.

1. Introduction

Building Automation Systems (BAS) are used to both improve the indoor climate in buildings and to reduce the operational costs. Originally, BAS mostly consisted of heating, ventilation and air-conditioning (HVAC) systems. To further increase management and reduce costs, both lighting, safety, security, and transportation supervision have been integrated into BAS. Traditionally, BAS have been used in large buildings such as schools, hospitals, and offices. For such buildings, the construction costs constitute only one seventh of the overall operational costs [5]. Hence, BAS provide a great savings potential by reducing the operational costs.

Wireless sensor networks (WSN) consist of small sensor nodes that sense the environment, perform computations, and communicate with other nodes using the onboard radio module. Typically, sensor nodes transport the measured data to a base station using multi-hop communication. The size of typical sensor nodes is close to a matchbox. Sensor nodes are typically powered by batteries. Since it is in general not possible, or too labor-intensive, to replace the node batteries, reducing power consumption is important in wireless senor networks. Since wireless communication is the major energy consumer [11], a particular focus has been on power-efficient communication protocols for wireless sensor networks.

Integrating WSN and BAS has a number of advantages. The main advantage is that the installation costs of WSN are lower than that of traditional BAS, since wiring is avoided. New buildings can therefore be equipped with a BAS based on wireless sensor network technology. It is also possible to extend an existing BAS in order to increase the sensor coverage. The decreased installation costs make it possible to increase the number of sensors and hence the spatial resolution. The increased spatial resolution allows for more fine-grained measurements and control. A further advantage is that wireless technology enables temporary measurements: a network can be set up to perform measurements during a limited time in order to measure, optimize and evaluate the effect of the optimization. We call such an effort ad hoc benchmarking. Furthermore, wireless sensors can also be installed more easily in unapproachable places such as at high heights.

In this paper, we show that it is possible to implement a standard protocol for building automation on typical resource-constrained sensor nodes found in wireless sensor networks. We report on our implementation of BACnet [1] on ESB nodes [12] that feature only 2 KB RAM and 60 KB flash ROM. Even with such severe constraints we show it is possible to implement an important subset of the BACnet services such as the read and write property services as well as the change of value (COV) service that fits very well into the resource-constrained nature of wireless sensor networks. We also discuss the limitations and future improvements of our current implementation.

Previous work [6] has used simple sensors that transmit temperature readings at predefined intervals to mains-powered receivers whereas we actually implement a building automation protocol on the sensor nodes which is a more generic solutions enabling all kinds of services.

2. Background

This section briefly introduces building automation systems, wireless sensor networks and the sensor node operating system Contiki [3].

Building Automation Systems Traditional BAS use wired technologies where sensors and actuators are connected to controllers. The first modern BAS deployed point-to-point centralized control that was later replaced by a centralized bus. Modern BAS use distributed bus control, also called field buses, and all system nodes have a built in control unit which enables them to act without centralized control.

1-4244-0826-1/07/$20.00 © 2007 IEEE

Wireless Sensor Networks A wireless sensor network (WSN) consists of tiny resource-limited devices called sensor nodes that feature also sensors, e.g. a temperature, light or humidity sensor. Further, nodes have an on-board low-power radio and a micro-controller. Sensor nodes are usually battery driven and transport measured sensor data in a multi-hop fashion to a base station.

Research challenges in WSNs include system software, communication protocols, power management and self-configuring networks. System software is challenging due to the limited memory on the nodes. If nodes are battery driven it is important to minimize the energy consumption in order to prolong the network lifetime. Typically this is performed by advanced sleep schedules that cause nodes to turn off their radios in a synchronized fashion.

The Contiki Operating Systems The Contiki [3] operating system is designed for resource-limited devices such as sensor nodes. Contiki uses an event-based kernel which reduces the overall system memory requirements by sharing a common stack between all processes. Contiki is implemented in C and ported to several platforms such as the TI MSP430 and the Atmel AVR. Contiki supports dynamic loading and unloading of programs at run-time and regular Internet networking through the uIP TCP/IP stack [2].

3. Integrating Wireless Sensor Networks with BACnet

When applying wireless sensor networks on building automation systems it is preferable to use existing BAS standards. By using existing standards, compatibility with already deployed solutions is ensured and future BAS protocol updates can be incorporated more easily.

After analyzing a number of existing BAS protocols including LonWorks, OPC, KNX and Modbus, we chose the BACnet protocol, “A Data Communication Protocol for Building Automation and Control Networks” [1]. The main reasons are that BACnet is an open, widely used standard with support for many underlying network technologies including TCP/IP. The server implementation is not very complex making it possible to implement BACnet on resource-constrained devices. Moreover, an open reference implementation is available through the BACnet homepage.

BACnet uses an object oriented approach to describe the functionality of nodes, called devices, in a network. A BACnet object describes one certain device functionality, e.g. information about a physically attached input (analog or binary input). Every object encapsulates a collection of data elements called properties used to describe the object. The properties may be accessed and altered by other objects.

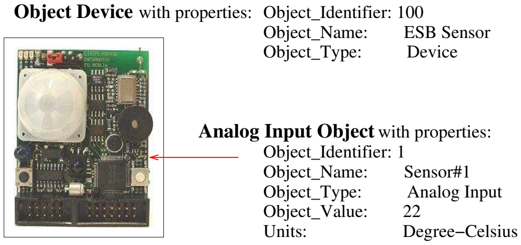

Figure 1. BACnet device object

The BACnet standard specifies a number of objects. These include objects for managing analog and binary in- and outputs, collecting trends, scheduling of operational hours and describing control loops. Every device has to implement one special object called the *device object* that describes the device. An example device object is shown in Figure 1.

In BACnet devices use *services* to pass information between each other. The communication model used is a traditional client-server model where each BACnet node represents a server, see Figure 2. Using services, BACnet devices can fetch information about other devices and objects, command devices to perform certain actions and inform other devices that an event has occurred. BACnet services are divided into five different groups: File Access Services, Object Access Services, Alarm and Event Services, Remote Device Management Services and Virtual Terminal Services.

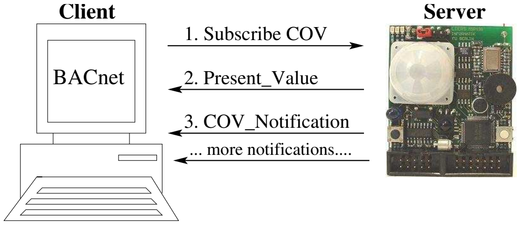

Figure 2. BACnet uses a client-server communication model

Object Access Services are a collection of services to access and modify object properties, and to create and delete objects. Properties in any BACnet object can be accessed through these services. This group contains the most commonly used services: the read and write property services. There also exist more powerful variants of these services, for example to read or write multiple properties at the same time or to read properties given certain conditions. Some object types can also be created and deleted with object access services, often used for calendars and schedules.

Alarm and Event Services are used to handle the communication regarding events. These events can be value changes for certain properties and changes in object state that meet predefined criteria, those considered serious are reported as alarms. One very interesting service for wireless sensor networks is the change of value (COV) service. Using COV, a device subscribes to receive updates of a property when it changes with a predefined amount since the last update, see Figure 2. This is for example useful for receiving temperature updates only when the temperature actually has changed.

4. Implementation and Evaluation

We have implemented BACnet on the Embedded Sensor Board (ESB) from FU Berlin [12]. The 2 KB RAM offered by the ESB nodes are comparatively smaller than the RAM size on other sensor nodes. For example, the widely used Telos node [10] has 10 KB of RAM memory.

BACnet is implemented as an application running on top of the Contiki operating system. TCP/IP communication is provided by the uIP stack [2]. The use of TCP/IP in WSN is controversial but there are efforts to make TCP/IP in WSN viable and there is at least one company selling WSN products based on TCP/IP (see `http://www.archrock.com`). Our implementation follows closely the reference implementation available on the BACnet homepage. This implementation did not include the COV service. Since we deem this service as extremely valuable for wireless sensor networks, we have implemented it.

The key contribution of this work is to demonstrate the feasibility of applying a well-known standard protocol for building automation on wireless sensor networks. In this section, we demonstrate that the memory footprint of our BACnet implementation is small enough to fit in memory-constrained sensor nodes and also discuss possible improvements of the current implementation. For more results, see [8].

4.1 Change of Value for Wireless Sensor Networks

Since the Change of Value (COV) service is very interesting we deploy it in a simple temperature collecting network. In the experimental setups the deployment was active in a home living room for 96 hours. We configured the change of value service to notify the client whenever the temperature had changed more than 1 degree Celsius since the last notification. The environment was relatively stable in terms of temperature fluctuations, but we believe it reflects typical BAS environments well.

During the experiment we received from one sensor node 9 COV notifications. If we had not implemented the COV service directly on the sensor nodes but instead on a network gateway or other entity the temperature values must have been polled or periodically transmitted from the sensor nodes. In comparison, assuming a temperature resolution of 10 minutes would be sufficient, the sensor node would have sent 576 radio messages instead of 9 in the living room. Since radio communication is the most energy consuming application on sensor nodes, using the COV service can significantly improve the life-time of a battery-driven wireless building automation system.

4.2 Memory consumption

In wireless sensor networks the available memory often limits the functionality of the system. We measure the static memory consumption of a number of different BACnet configurations on our sensor nodes.

The results confirm that our base system, consisting of the Contiki operating system with the uIP TCP/IP stack, is fairly small. The base system requires only around 1 KB RAM. The memory consumption increases significantly when adding the smallest possible BACnet configuration; the read property service with only one object. Table 1 shows that memory consumption increases with the complexity of the services and, in the case of COV, the number of subscribers. For example, each additional COV subscriber requires 16 bytes ROM and 82 bytes RAM. The results in do not include any multi-hop functionality since routing protocols are not part of the uIP stack.

4.3 BACnet and Multi-hop communication

Since multi-hop communication is essential for scalable sensor network solutions we measure the memory consumption of BACnet and a simple multi-hop protocol. The multi-hop protocol we use is an Ad hoc On-demand Distance Vector (AODV) protocol [9] which is part of the Contiki system. To decrease memory consumption, the protocol is limited to 8 routing entries. The rest of the system is left unchanged; the TCP/IP packets with BACnet data are just tunneled and forwarded to its final destination just above the MAC layer.

The extra memory required by the AODV routing protocol limits possible BACnet configurations. For example, we are not able to combine the COV service and the multi-hop functionality when using our not fully optimized BACnet implementation on the resource-constrained ESB nodes.

Stack usage analysis To analyze the actual memory requirements of a running system we use the MSPsim MSP430 emulator [4] connected to the Contiki OS sensor network simulator COOJA [7]. In these measurements we use a configuration with both the read property and write property services enabled. The write property service enables a sensor node to control external devices. We measure the stack usage during different events in the system to compute the total memory usage. The results from the simulation confirm our earlier discussion about the need to optimize the BACnet implementation for the intended low-memory platforms. The object-oriented design of BACnet and its reference implementation requires a lot of memory to handle incoming network packets. The reason for this is that each packet is structured into a set of headers and each header is handled in its own function. And each function call has several 32-bit arguments which are placed on the stack. Since these calls get deeper for each handled header and more stack is required for every such function call, the stack usage peaks during the packet handling.

System configuration			Memory requirements	
Read property (#AI)	Write property (#BO)	COV (#Subscribers)	Flash/ROM (Bytes)	RAM (Bytes)
Only base Contiki & uIP system			25772	1054
Yes (1)	No	No	39934	1472
Yes (1)	Yes (1)	No	44516	1524
Yes (1)	No	Yes (1)	43924	1554
Yes (1)	No	Yes (2)	43940	1636
Yes (1)	Yes (1)	Yes (1)	48506	1606
Yes (2)	Yes (1)	Yes (1)	48510	1606

Table 1. Memory consumption for different BACnet configurations

The results from the stack analysis show that handling a new incoming packet requires around 400 bytes of RAM. With the AODV protocol the lowest amount of available memory during the simulation is only 50 bytes, i.e. dangerously close to stack overflow which may cause the system to malfunction. Figure 3 shows an overview of the stack memory usage during an incoming radio packet. The duration of the graph is 8 milliseconds.

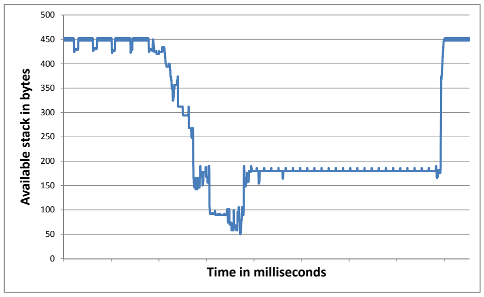

Figure 3. The available memory decreases significantly during packet handling (bytes)

5. Conclusions

Large overhead costs including installation and maintenance in building automation system can be reduced by using wireless sensor networks. We have implemented and evaluated the open BACnet standard on top of a wireless sensor network. The results show that standard solutions, not originally designed for wireless sensor networks, such as BACnet and TCP/IP networking can successfully be applied in this domain. In particular, we demonstrate that it is possible to run BACnet over TCP/IP on ESB nodes, one of the more resource-constrained sensor node platforms.

Acknowledgement and Source Code Availability

This work has been funded by the Swedish Energy Authority. The source code is available on request. The source code for Contiki and uIP can be found at `http://www.sics.se/nes`.

References

[1] ASHRAE. *BACnet - A Data Communication Protocol for Building Automation and Control Networks*, ansi/ashrae standard 135-2004 edition, 2004.

[2] A. Dunkels. Full TCP/IP for 8-bit architectures. In *Proceedings of The First International Conference on Mobile Systems, Applications, and Services*, May 2003.

[3] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Workshop on Embedded Networked Sensors*, Tampa, Florida, USA, Nov. 2004.

[4] J. Eriksson, A. Dunkels, N. Finne, F. Österlind, and T. Voigt. Mspsim – an extensible simulator for msp430-equipped sensor boards. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, Delft, The Netherlands, Jan. 2007.

[5] W. Kastner, G. Neugschwandtner, S. Soucek, and H. Newmann. Communication systems for building automation and control. *Proceedings of the IEEE*, 93(6):1178–1203, June 2005.

[6] M. Kintner-Meyer and R. Conant. Opportunities of wireless sensors and controls for building operation. In *ACEEE Summer Study on Energy Efficiency in Buildings*, pages 3–139–3–152. American Council for an Energy-Efficient Economy, Washington, DC., 2004.

[7] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *International Workshop on Practical Issues in Building Sensor Network Applications*, Tampa, Florida, USA, Nov. 2006.

[8] F. Österlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt. Integrating building automation systems and wireless sensor networks. Technical Report T2007:04, Swedish Institute of Computer Science.

[9] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, Internet Engineering Task Force, 2003.

[10] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proc. IPSN/SPOTS'05*, Los Angeles, CA, USA, Apr. 2005.

[11] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, Mar. 2002.

[12] J. H. Schiller, A. Liers, H. Ritter, R. Winter, and T. Voigt. Scatterweb - low power sensor nodes and energy aware routing. In *HICSS*, 2005.