A step towards standardization of wireless sensor networks: a layered protocol architecture perspective

Sajjad A. Madani, Stefan Mahlknecht, Johann Glaser
Institute of Computer Technology, Vienna University of Technology, Austria
{madani, mahlknecht, glaser}@ict.tuwien.ac.at

Abstract

Unlike the Internet, Wireless Sensor Networks (WSNs) have yet to result in wide deployment in the real world. The unique properties of wireless communication including mobility, rapidly changing and unpredictable link quality, limited resources in terms of computation and energy, opportunistic exploitation, environmental obstructions, and new design paradigms motivates to divert from traditional layered architectures. At the same time, "plug and play" like features of the layered architecture which resulted in wide range deployment of the systems are required. In this paper we focus on the layered protocol architecture for WSNs, which provides the benefits of traditional layered architectures (interoperability) as well as focuses on cross layer design to leverage from the benefits offered by the unique wireless communication properties.

1. Introduction

In referenced architectures like OSI, the communication between non-adjacent layers is not allowed [1]. Sensor networks with constrained resources and its longevity requirements, Cross Layer Design (CLD) becomes necessary. CLD according to [1] is "*Protocol design by the violation of reference layered communication architecture is cross-layer design with respect to the particular layered architecture.*" According to [1], the violation of referenced design may include redefinition of boundaries, creation of new interfaces between adjacent and non-adjacent layers, tuning of parameters on different layers by changing network parameters from another layer, and interdependency between layers of protocol design. We define cross layer optimization as adapting certain parameters of one of the layers on the basis of feedback from another layer, for instance to improve energy efficiency or end to end delay.

The authors in [2] discuss the importance of good architectural design and have emphasized that only performance enhancements at the cost of good architectural design can never result in a system which can be globally deployed like the Von Neumann, the OSI, and the Shannon's communication architectures. The main point the authors want to stress is that, "*the trade-off between performance and architecture needs to be fundamentally considered*".

We introduce a layered protocol architecture which takes care of the issues discussed in [2], as well as provide benefits from the unique wireless communication characteristics by a cross layer approach. The proposed architecture (Figure 1.) is composed of traditional layers, including application, transport, network, link, and physical layer. The application layer (AL) can communicate via well defined interfaces with the routing (RL). The direct connection between AL and RL is required where no Transport Layer (TL) is used (As for most cases in sensor networks, end to end communication is not important and mostly relies on hop by hop paradigm). The interfaces between RL and Mac Layer (ML), and ML and Physical Layer (PL) are introduced. Inspired from [3], we introduce a Cross LAyer Management Plane (CLAMP) to provide cross layer benefits but in an optional way so that the concept of modularity of layered architectures is maintained. As a wireless sensor node has limited energy and it is not practical to replace the energy supply unit because of cost or geographic reasons, an Energy Management Plane (EMP) is introduced to provide services to different layers. In most cases, security is considered as a stand-alone component of system architecture which usually is a flawed approach to network security [31]. We present a Security Management Plane (SMP) so that security can easily be integrated into every component as discussed in [31].

This paper is organized as follows. Section II provides an overview of the related work. The proposed solution with interfaces between different layers and management planes are discussed in section III. Section IV discusses implementation issues while section V gives overview of performance analysis. We conclude our discussion in Section VI.

2. Related Work

In [4], the authors have presented a unifying link abstraction for WSNs. They consider Sensor-net Protocol (SP) as a "narrow waist", just like Internet protocol for the Internet. SP is an abstract layer present between the network layer and the link layer enabling different routing and MAC schemes to co-exist. They have introduced the concept of a neighbor table in which data related to the neighbors is kept so that different protocols running on the same node do not keep independent routing tables and get access to the routing and link layer parameters in this shared table. Our approach is different from it in many ways. Firstly, we follow the basic architectural style as of the OSI for reasons mentioned in [2]. We do not define a message pool, neighbor table, an additional abstract layer, and additional vertical plans (other than we have defined) because of resource constraints on sensor nodes.

The authors in [5] discuss architectures for heterogeneous WSNs. They have classified the applications, routing and

MAC schemes into different categories and have introduced Protocol Stack Tree (PST), which is a combination of different existing protocols and each path on the tree is able to satisfy different application requirements. The authors talk about cross layer entities but in a general way.

The ZigBee [6] stack architecture is based on the OSI reference model but considers only application, network, link and physical layer. The physical layer and the medium access control sub-layer are defined by the IEEE 802.15.4 [7] standard while the ZigBee Alliance defines the layers above. Why ZigBee cannot provide a viable solution is discussed in [4] as "*ZigBee proposes a classic layered architecture, but each layer assumes a specific instance of the surrounding layers: e.g., the routing layer assumes the IEEE 802.15.4 link and physical layers. An architecture build on static technologies is destined for obsolescence*".

In [8] the authors discuss a network stack architecture. They have introduced an architecture composed of Application Layer, Data Fusion Layer (DFL), Data Service Layer (DSL), Medium Access Layer and Radio Layer. DFL is introduced to fuse data based on application requirements or based on a fact that the sensed data may be correlated which would reduce end to end latency as the message will not have to go up the stack to the application layer at relay nodes or specified nodes. The DSL essentially serves the purpose of a routing layer. They have also introduced an Information Exchange Layer as a shared database that serves the purpose of cross layer optimization. No emphasis is given to security concerns and it is stated that the DSL can handle it. It also lacks an energy management plane despite energy efficiency and system lifetime is one of the main challenges faced by the research community in this specific area.

3. Proposed Solution

The main focus of the proposed solution is two-fold: firstly it should be similar to the traditional layered architectures because they have already proven to be successful and secondly, it should deal with the unique characteristics offered by the wireless communication paradigm (e.g. mobility, rapidly changing and unpredictable link quality, limited resources in terms of computation and energy) by a CLD approach.

The proposed architecture comprises traditional layers and management planes as shown in Figure 1. We include a PL, ML, RL, an optional TL and an AL similar to the OSI model. The CLAMP, EMP, and SMP are introduced and connect to the full set of layers for unlimited interaction to gain the full optimization potential. Network diagnosis and Management (NM) (e.g. resetting nodes, remote firmware deployment, address assignment, querying availability of nodes) is connected to management planes and also to horizontal layers via CLAMP.

For each of the layers and planes we propose defined interfaces which are either mandatory, optional or user defined. Mandatory interfaces have to be implemented whereas optional interfaces may be implemented, but both have to follow the specification strictly. User defined interfaces can be specified and implemented to the user's requirements. Every layer and plane can be implemented in several different ways (i.e. different MAC protocols) and then used interchangeably in conjunction with the other layers. The defined interfaces allow replacing one such entity without having to touch the others. For every pair of two such entities the interfaces of both sides are drawn in interface graphs. We will discuss these layers and planes one by one in the proceeding sections.

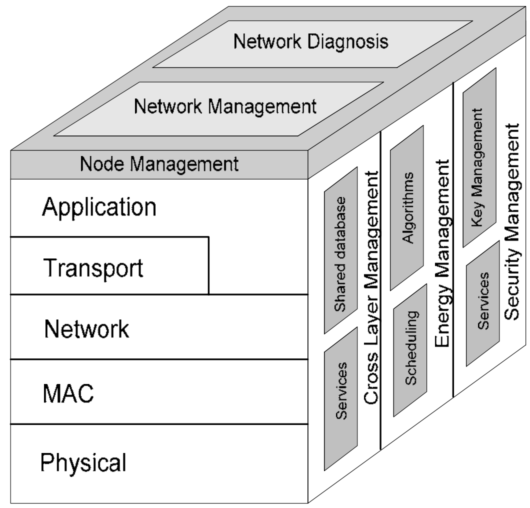
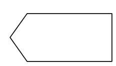
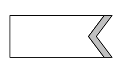
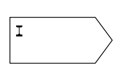
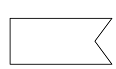


Figure1. Proposed protocol architecture for WSN

3.1. Symbol Definitions

Functional Interfaces of each module may represented as a graphical symbol as shown in Table 1.

Table 1: Description of symbols used in interface diagrams

Symbol	Description
	Function call from within a module to invoke a functional interface in an adjacent module. The user can name it anything whatever he/she wants to.
	A functional interface invoked by a callee of an adjacent layer. The name of this functional interface cannot be changed. The shaded region at the back depicts that some data is associated with its invocation. E.g. "*receive*" in AL is asynchronously invoked by lower layers when data arrives at the node.
	An interrupt driven functional call, used to interrupt the next upper layer on the reception of data so that the received data can be processed first.
	A functional interface called by the callee to query or notify some thing to the adjacent layer. For instance, querying regarding channel status or notifying to listen.

3.2. Application Layer

A few applications previously introduced include monitoring and health care systems [9-14]. In [5], the author has classified WSN applications on the basis of information delivery (query driven, event driven, and continuous), delay (real time, non-real time, and delay tolerant), infrastructure type (homogeneous and heterogeneous), and deployment (deterministic and non deterministic).

Having discussed diverse network application requirements (combination of the above different classifications), it can be dealt with in two ways as discussed earlier: either go for an application specific architecture to attain performance (e.g. energy efficiency or end to end delay) gains at the cost of good architectural design or rely on a more generic solution at the cost of performance. As our focus is to draw the line somewhere in-between, we provide simple interfaces between AL and TL (Figure 2.).

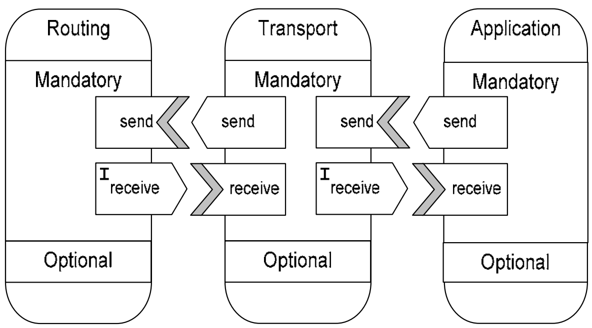


Figure2. Interfaces between AL & TL and TL & RL.

3.3. Transport Layer

In [15], the author mentioned that the transport layer is required when the system has to talk to the internet or any other communication network. Most of the communication within sensor networks is done hop by hop (no notion of end to end delivery in many cases), and normally there are dedicated nodes per sensor networks, which are connected to an external network. We provide interfacing between AL and RL, so that if in a particular case, the TL is not implemented, the architecture is still flexible enough to accommodate this. The interfaces provided between TL and AL as well as between TL and RL are the same as between AL and RL (Fig. 2).

3.4. Routing Layer

The authors of [16] present a survey on energy efficient routing protocols by classifying them into different categories known as data-centric, hierarchical and quality of service routing, each of them suitable for a specific application or a group of application scenarios. Mobility, localization, and data fusion/aggregation services are also required to decrease energy utilization in WSNs. Keeping in view energy, size, and memory constraints, we provide a simple set of interfaces between RL and its adjacent layers. The rest of the components (data fusion and aggregation, localization, mobility management, forwarding, determining minimum path cost) are to be implemented within the RL as sub-modules. The interfaces between RL and AL are shown in Figure 2, and the interfaces between RL and ML are shown in Figure 3.

3.5. MAC Layer

The attributes of MAC schemes for WSNs include energy efficiency, scalability and adaptability to changes [17]. A wide range of MAC schemes [18-21] has been introduced previously, each of them suiting a specific group of application requirements. The basic set of interfaces between ML and RL required is the same set of interfaces discussed for RL and AL. For ML-PL interfacing some additional interfaces are required (Figure 3).

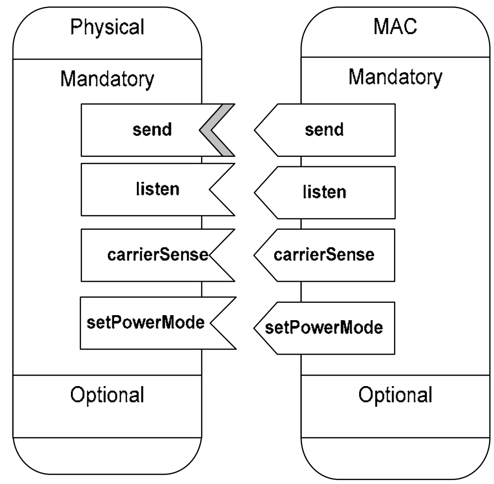


Figure3. Interfaces between MAC and PHY Layer

3.6. Physical Layer

The role of the physical layer in WSNs is not well defined yet [1]. This is because of the new modalities in wireless systems. As an example, in some radios, [22], the CRC check is implemented in hardware. Similarly, a wakeup radio [21] may require additional processing at the physical layer to figure out if the packet is intended for this specific node or not (This can be used to drop packets not intended for it and hence save processing energy at upper layers). These issues can be dealt with either by introduction of additional bits in frame headers or can be achieved with the help of user defined interfaces available at the user's disposal.

3.7. Cross LAyerManagement Plane (CLAMP)

The interfaces provided by CLAMP are *publish*, *update*, *subscribe*, and *query*. Initially the CLAMP database is empty. Every module can *publish* its parameters and thus is their owner. Later on it can *update* the value of a particular parameter. All other modules can *query* the current value of a parameter. Additionally they can *subscribe* to parameters. Once there is a change in any of the values of the known parameters, the CLAMP notifies all the subscribed modules with the help of the *notify* call. If any layer subscribes to a parameter which has not (yet) been published, the services module keeps track of the client and notifies it as soon as that particular parameter is published.

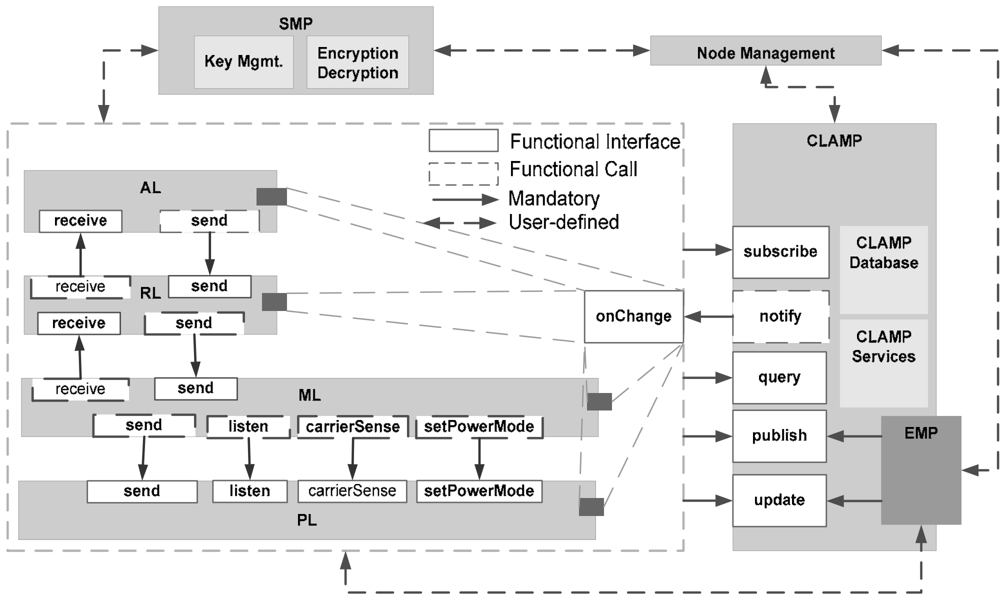


Figure4. Unified View of Interfaces

The network parameters that are provided by the CLAMP to different layers can be useful in many ways. For instance, if the AL sets the parameter “realTime” to 1, RL can route it on the basis of minimum hop routing. The “packetLenght”, “outputPower” and “BER” are directly related [23]. The “modulation” can be changed according to the remaining capacity of the battery [23]. The “dataRate” can be increased or decreased depending upon the link quality and SNR. The remaining capacity may allow the node to back-off so that the battery undergoes the recovery effect [24]. When and how to use these parameters is a challenging issue [1] and we are still in the process of finalizing, and fine tuning these parameters.

3.8. Energy Management Plane

Sensing, communication, and processing are the three main energy consuming components in a wireless sensor node [25]. As the sensor nodes are battery operated or powered by an energy scavenging technique, the restricted amount of energy in a sensor node becomes the main issue in the deployment of a sensor network. As technology advancement in the chemistry of batteries is slow compared to silicon chip technology [26], the Energy Management Plane (EMP) may provide a viable solution for efficient energy utilization. The goal of the EMP is to maximize the network life time. When taking into consideration batteries, the actual capacities are different from rated capacities because of non linear battery effects, different algorithms [24], [27], and [28] can be implemented in the EMP to find out the remaining capacity of the battery, which can be utilized by different layers to do energy aware computing.

The EMP may also take the responsibility for scheduling of different events to save energy. Such events include periodic listening, sensing of different types of sensors, updating timers, or analyzing incoming messages. Because it consumes time and energy to either change the state of the radio from sleep/idle to transmit state or any other hardware such as turning on the power supply of the sensors or waking up the CPU, it is very critical to implement algorithms which synchronize different activities.

Such energy management concepts are usually implemented implicitly in the sensor node firmware, e.g. by placing function calls in a specific order. Changing these concepts at development time is mostly tedious, whereas dynamic reconfiguration is nearly impossible. The EMP enables implementing these concepts in an explicit manner.

With this explicit approach it is easily possible to investigate various energy management concepts during development time. The defined interfaces ensure seamless interchangeability. Such an energy management concept can even dynamically change its behavior depending on the remaining energy.

The space and time complexity of these algorithms needs in-depths consideration, as we need to find out the relationship between energy utilized by these algorithms (processing energy) on individual nodes and the analysis of prolonging the network life time before implementing them.

3.9. Security Management Plane

Because of limited resources, security requirements in WSNs are more challenging than in conventional networks. Security for WSNs entails key establishment and trust setup, secrecy and authentication, privacy, secure routing, intrusion detection, and secure data aggregation [29]. We provide a security management plane (SMP) similar to the “*security service provider*” in ZigBee Security Architecture [6] where every layer is connected to it with standardized interfaces. The SMP includes key a management algorithm and provides security services to individual layers like helping RL in secure routing, encryption and decryption at ML, and/or authentication at TL. These functionalities are provided by a “services” component of SMP.

4. Implementation

We have implemented the concept in the PAWiS framework [30]. A WSN node is split up into different modules for instance, RL, AL, ML, PL, CLAMP, EMP, SMP, ADC, Timers, and CPU etc. Each of these modules is implemented as a C++ class. The interfaces are implemented as message passing calls which we call "Functional Interfaces". These are used for communication between different modules. All modules are implemented as Finite State Machines (FSM). Every invocation is managed by the discrete event simulation environment utilizing a future event list and a lot of overhead to deliver messages between the various modules.

The functional interfaces of a particular layer or plane are invoked by other modules with the PAWiS Framework method `invoke(`*module name*`,` *interface name*`,` *input parameters*`,` *output parameters*`)`. "Module name" is the name of the module whose interface is being called. "Interface name" is the name of the interface which is invoked. The third and fourth parameters are pointers to objects which conduct input and output parameters. For instance, assume that the ML wants the PL to start listening. This can be done by invoking a functional interface `invoke("Phy", "listen", &inParams, &outParams)`. In this way the ML would provide the amount of time for which PL should listen in the `inParam`, while as soon as some data is available at the PL, it is provided to ML in `outParam`.

When implementing a sensor node on real hardware, for the firmware different techniques have to be utilized for interfaces and parameters to reduce complexity and overhead. All interfaces should be implemented as regular function calls. CLAMP parameters which are only queried (i.e. no modules need immediate notification of changes) should be implemented as global variables. For CLAMP parameters with subscribed modules another approach is necessary. We propose callback function calls here. In OS based embedded systems, the architecture can be implemented as callback functions, events and commands in a similar way as implemented in the simulation framework.

5. Simulation and Results

The PAWiS simulation framework [30] built on the top of OMNeT++ [32] was used for simulation. The AL implements a simple temperature sensing application (normal distrubtion of temperature changes). Every two seconds the value is measured and if there is a 20% change, a data packet is generated which is sent to the sink node. AL also publishes parameter "delay" which defines delay tolerance of AL. 0 % delay tolerance means least possible delay while 100% means "don't care" condition. Energy Aware Distance Vector Routing (EADV) [33] is implemented on the RL. On ML, simple a CSMA based MAC scheme is implemented where a node first senses the channel with random delay. If the channel is free, it sends the data after a random delay based on uniform distribution. If the channel is busy, the node backoff's and senses the channel again after a random delay. It tries 10 times to send the packet and if it fails to send the packet, the packet is dropped. It is assumed that all nodes are synchronized with each other. Energy consumption to transmit and receive a packet is a function of packet lenght, transmit power and data rate. 24mA current consumption is considered to recieve a packet while 19mA is considered to transmit a packet at 0dBm. Energy consumption values are taken from CC2400 datasheet [34].

The topology used for simlutaion consists of 12 nodes (Figure 5). Nodes are placed such that each node is connected to two other nodes. Node 0 is considered to be sink node with unlimited amount of energy. Node 9, 10, and 11 are initialized with 25 % of energy as compared to other nodes. This creates two distinct paths, one with enough energy and one with low delay. For example, consider node 8 initiates a transfer, so the path 8 → 9 → 10 → 11 → Sink would be the high cost path with low delay (based on EADV) while path 8 → 7 → 6 → 5→ 4 → 3 → 2 → 1 → Sink would be a low cost path with high delay if energy is considered as a metric for routing. It would be vice versa if hop count is used for the cost metric. During normal operation the routing mechanism should allow delay and therefore utilize paths with enough energy. On the other hand, an alarm has to be transmitted as fast as possible, so the routing layer should use the lowest delay route.

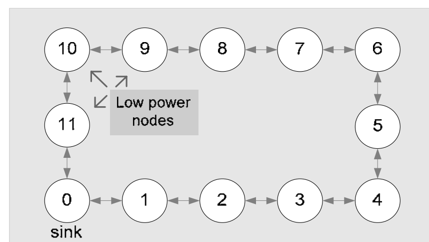


Figure5. Topology used for simulation

Figure 6. shows the end to end delay of packets from node 8 towards the sink node. The AL of all nodes produced a synchronized alarm by updating the "delay" parameter in the CLAMP database. This is a simplification for the sake of demonstration. In real scenarios this can be achieved by generating an alarm broadcast packet by the "alert" node which informs the other nodes accordingly. The RL which has already subscribed to the aforementioned parameter, changes its routing strategy from minimum energy routing to minimum hop routing. Figure 6 clearly indicates a decrease in the average end to end delay during the alarm period (ranging from time intervals 25 to 50). The update of the delay parameter by the AL to 100 % stops the alarm period and the RL again changes its routing strategy from minimum hop to minimum energy.

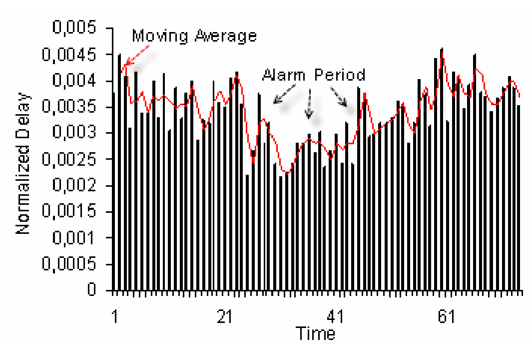


Figure6. End to End Delay from Node 8 to sink node

One may argue that the results obtained from simulation can be achieved with different architectures without the Cross LAyer Management Plane, e.g. by message passing between AL and RL. Another application is a major change of the packet sizes during run time which may affect output power and bit error rate [34]. In [36] the location information of nodes is used for energy savings by switching a group of neighboring nodes to a sleep state. This all can be achieved by message passing between the network layer but needs additional interfaces which have to be implemented for both interdependent layers. Therefor a change in implementation at one layer would require a lot of changes at other layers. In our approach we propose a standardized way to achieve cetain performance enhancements. This allows simple plug and play of different implementations at different layers with less changes required and thus enables true rapid prototyping. It also allows optimizing performance aware and energy aware parameters.

6. Conclusion

In this paper we proposed a WSN architecture with well defined interfaces between layers as well as blocks for cross layer management, energy management and security. The basic idea is to define an architecture as near as possible to traditional architectures by taking into consideration WSN specific constraints and opportunities. We kept the architecture simple but extendable in order to have light weight implementations in the real world. We have simulated the architecture and shown that the concept is feasible and enables powerful yet affordable functionality. We are currently working on the definition of well defined messages, the refinement of CLAMP parameters and their usage, and evaluation of the SMP and EMP. This architecture will be implemented in the real world as a part of the PAWiS project [30] funded in part by the Austrian Research Program FIT-IT and Infineon as industry partner.

7. References

[1] Vineet Srivastava and Mehul Motani, "Cross-Layer Design: A Survey and the Road Ahead", IEEE communication magazine, December 2005

[2] Vikas Kawadia and P. R. Kumar, "A Cautionary Perspective on Cross-Layer Design", IEEE wireless communication, 2005

[3] Vlado Handziski, Andreas K opke, Holger Karl, Adam Wolisz, " A Common Wireless Sensor Network Architecture", in Proc. 1. GI/ITG Fachgespräch ``Sensornetze'' July 2003

[4] Polastre, J., Hui, J., Levis, P., Zhao, J., Culler, D., Shenker, S., and Stoica, I. 2005. A unifying link abstraction for wireless sensor networks. 3rd international Conference on Embedded Networked Sensor Systems, USA, November 02 - 04, 2005.

[5] Durresi, A., "Architectures for heterogeneous wireless sensor networks," IEEE Personal, Indoor and Mobile Radio Communications, 2005.

[6] www.zigbee.org

[7] www.ieee802.org/15/pub/TG4.html

[8] Rajnish Kumar , Santashil PalChaudhuri , David Johnson, Umakishore Ramachandran, " Network Stack Architecture for Future Sensors" available as Rice Technical Report 04-447 at http://www.cs.rice.edu/~santa/research/stack/stack.pdf

[9] S. Mahlkench and S. A. Madani, "On architecture of low power wireless sensor networks for container tracking and monitoring applications", INDIN07, Vienna, Austria, 2007 [Accepted]

[10] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. ACM SIGCOMM Workshop on Data Communications in the Caribbean, April 2001

[11] Ning Xu, " A survey on Sensor Network Application", Computer Science Department, University of Southern California

[12] Seong-Moo Yoo, " Sensor Network for Automobile Routing", a report submitted by University Transportation system of Albama.

[14] Mani B. Srivastava, Richard R. Muntz, and Miodrag Potkonjak. Smart kindergarten: sensorbased wireless networks for smart developmental problem-solving enviroments. In Mobile Computing and Networking, pages 132.138, 2001

[15] Akyildiz, I.F.; Weilian Su; Sankarasubramaniam, Y.; Cayirci, E., "A survey on sensor networks," Communications Magazine, IEEE , vol.40, no.8pp. 102- 114, Aug 2002

[16] Kemal Akkaya and Muhammad Yousaf, " A survey on Routing protocols for wireless sensor networks" retrieved on November 03, 2006 from www.cs.umbc.edu/~kemal1/mypapers

[17] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, Volume: 12, Issue: 3, Pages:493 - 506, June 2004.

[18] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of InfrastructureWireless Sensor Networks", ISCC'04, pages 244-251, Egypt, June 2004

[19] S. Mahlknecht, M. B¿ck, "CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks," Proceedings WFCS, pp. 73 - 80. , 2004

[20] Wie Ye, John Heidemann, and Deborah Estrin, "An energy efficient mac protocol for wireless sensor networks.", In proceedings of IEEE Infocom, pp. 1567-1576, New York, NY, June 2002

[21] Xue Yang, Nitin H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving Balance between Energy Saving and End-to-end Delay," rtas, p. 19, 10th RTAS'04, 2004

[22] www.chipcon.com/files/220_Data_Sheet_1_2.pdf

[23] Vijay T. Raisinghani, Sridhar Iyer, "Cross-layer design optimizations in wireless protocol stacks", Computer Communications 27, 2004.

[24] Rakhmatov, D.; Vrudhula, S.; Wallach, D.A., "A model for battery lifetime analysis for organizing applications on a pocket computer," IEEE Transactions on VLSI , vol.11, no.6pp. 1019- 1030, Dec. 2003.

[25] Du, X.; Lin, F., "Efficient energy management protocol for target tracking sensor networks," 9th IFIP/IEEE International Symposium on Integrated Network Management, 2005.

[26] Gao, Q., Blow, K. Holding, D. Marshall, Ian, "Analysis of Energy Conservation in Sensor Networks", Wireless Networks, Volume 11, Number 6, pp. 787-794(8) ,November 2005

[27] M. Handy and D. Timmermann, "Simulation of Mobile Wireless Networks with Accurate Modelling of Non-linear Battery Effects", In proceedings of Applied simulation and modeling (ASM), Marbella, Spain, 2003

[28] C.F. Chiasserini, R.R. Rao, "Energy efficient battery management," Proc. of Infoconi 2000, Tel Aviv, Israel, March 2000.

[29] Perrig, A., Stankovic, J., and Wagner, D. 2004. Security in wireless sensor networks. Commun. ACM 47, 6 Jun. 2004

[30] www.ict.tuwien.ac.at/pawis

[31] Perrig, A., Stankovic, J., and Wagner, D. 2004. Security in wireless sensor networks. Commun. ACM 47, 6 (Jun. 2004), pp. 53-57

[32] www.omnetpp.org

[33] S. Mahlknecht, S. Madani, and M. Rotzer. Energy Aware Distance Vector Routing Scheme for Data Centric Low Power Wireless Sensor Networks. In *4th IEEE INDIN06*, 16-18 Aug. 2006.

[34] www.chipcon.com

[35] Vijay T. Raisinghani, Sridhar Iyer, "Cross-layer design optimizations in wireless protocol stacks", Computer Communications 27, 2004, pp. 720-724 .

[36] Carreras, I.; Chlamtac, I.; Woesner, H.; Zhang, H., "Nomadic sensor networks", Proceeedings of the Second European Workshop on Wireless Sensor Networks, Jan.-2 Feb. 2005 Pages: 166- 175.