CLAMP: **C**ross **LA**yer **M**anagement **P**lane for low power wireless sensor networks

Sajjad Ahmad Madani, Stefan Mahlknecht, Johann Glaser
Institute of Computer Technology, Faculty of Electrical Engineering and IT.
Vienna University of Technology
{madani, mahlknecht, glaser}@ict.tuwien.ac.at

Abstract

Traditional layered architectures used for wireless networks pose drawbacks in terms of performance and efficiency. The constrained resources of wireless sensor nodes such as memory, computational power, and energy motivate to modify traditional layered architectures. In this paper we present a cross layer management plane for low power wireless sensor networks which enables sensor nodes to exploit cross layer information for efficient resource utilization. A set of well known parameters to be used to benefit from the synergy across layers is presented. The feasibility of the proposed scheme and the advantages drawn from using cross layer information is shown by simulation**.**

Key words— Wireless Sensor Networks, Cross Layer Design, Layered Architectures

1. Introduction

The advances in electronic circuitry and micro electro-mechanical systems have made it possible to build smart environments powered by sensor networks [1, 2]. Sensor networks comprise a large number of inch sized sensor nodes. As the sensor nodes are battery operated or scavenge energy from the environment, the restricted amount of energy becomes the main issue in deployment of sensor networks. Sensor nodes are also characterized by constrained memory resources and computation power [33]. Besides limited energy, memory, and computational power, other differences between sensor networks and ad-hoc networks include the number of participating nodes, density of nodes, unattended operation, potentially harsh conditions, redundancy of data packets, and frequent failures [38-40]. Such constraints and differences in sensor networks motivate to divert from traditional layered architectures.

Unlike traditional Ad hoc wireless networks, where Quality of Service (QoS), reliability and transmission delay were the main issues [3], sensor networks need to last months to years. The traditional layered networking approach has loopholes in terms of performance and efficiency of the system for wireless networks [3]. We propose a Cross LAyer Management Plane (CLAMP) to be integrated as an additional vertical plane to the traditional layered networking approach. CLAMP is sub module of our recently proposed protocol architecture [21], (see Fig 1.) which is similar to the traditional layered approach providing cross layer optimization benefits [4] in an optional way so that the concept of modularity of layered architectures is maintained. The idea is simple in terms of space and time complexity to be implemented in resource constrained sensor nodes. The main point is to provide a list of explicit parameters which are accessed with publish-subscribe-update paradigm. The explicit list should contain well known parameters just like we have port numbers [30] for well known services [31] in TCP/IP protocol suite so that network software for different modules can be used interchangeably e.g. different routing protocols which use cross layer information can be plugged and played with out customizing other concerned layers. In this way, the problems (implementation, debugging, and standardization) discussed in [24] can be addressed.

Preliminary results (section V) show that implementation of CLAMP in a traditional layered architecture can prove to be very beneficial to extend the network life time.

The rest of the paper is organized as follows: Section 2 gives an overview of related work. Section 3 presents the proposed solution. Section 4 discusses implementation details and preliminary results are discussed in section 5. Finally, Section 6 concludes the discussion with future directions.

2. Related Work

In [5], the authors propose a cross layer optimization frame with an optimization agent providing top down and bottom up feedback to different layers of the protocol stack to benefits from the current network conditions. In principal our approach is similar to the one discussed in [5], but we define a set of well-defined parameters in advance which can affect network performance and energy utilization at run time. Knowing the set of parameters in advance, module implementations on different layers can be exchanged without any modifications in the other modules. We also keep the usage of these parameters to be optional so that if a particular module on some layer does not want to use them or a particular parameter is not available to certain module (which it intends to use), the architecture should be flexible enough to accommodate this.

In [6] the presents the benefits of cross layer feedback and related survey but do not propose a specific architecture for cross layer design.

The authors in [7] present cross layer feedback architecture for wireless networks. They introduce tuning layers (to provide interfaces to data structures stored on different layers) and optimization subsystems (algorithms for cross layer optimizations) to gain cross layer benefits. As the architecture is proposed for wireless comparatively high power mobile devices; the processing overhead of tuning layers and optimizing subsystems may not be well suited for low power wireless sensor networks.

The authors in [8] propose an Information Exchange Service (IES) to introduce cross-layer optimization. The approach is very similar to our approach but like [5] it does not provide an explicit list of parameters to be maintained by the cross layer management plane which restricts the usage of “plug and play” features. When replacing a single layer the adjacent layers have to be adjusted too. The overall protocol stack architecture is also different from the traditional approach with the introduction of data fusion and data service layers.

Many papers [25] can be found related to cross layer design and optimization. The problem with all this research is that the researchers have focused on one, two, or three parameters (or interaction between two layers) and showed the benefits of using cross layer interaction between these layers. More important is the *presence of a standard way to achieve cross layer benefits* which would reduce the time to design protocol stack for network of embedded devices.

3. Proposed Architecture

The CLAMP is sub-module of our recently proposed layered protocol architecture [21] for low power wireless sensor networks shown in Figure 1. The protocol architecture is composed of traditional layers, the energy management plane (EMP), the security management plane (SMP), the node management and the CLAMP. The architecture is additionally shown with partial software to hardware mapping to clearly understand the major energy consumption components For example, the radio module is mapped to the physical layer and the energy consumed in transmitting and receiving data can be attributed to the physical layer. Similarly, sensors are mapped on to the application layer and energy consumption profile of the application layer would depend on the data sampling technique (e.g. cyclic, or probabilistic).

The CLAMP communicates with all the layers of the protocol stack, with the energy and security management plane, and with the node management plane allowing every individual component of the system to fully exploit the benefits from information available across all layers.

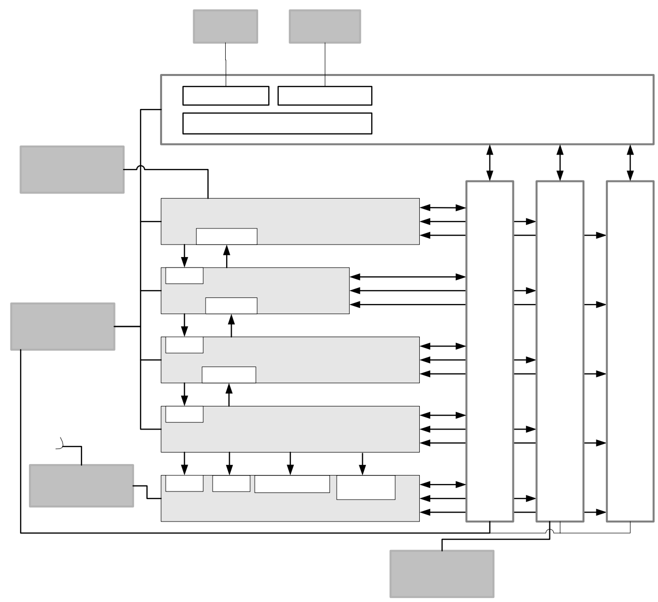


Figure 1: Protocol architecture for wireless sensor networks with partial hardware mapping

3.1. CLAMP Architecture

The CLAMP architecture (Fig. 2) comprises three components: a database, services and a set of interfaces. The database is basically a collection of well-defined performance aware and energy aware network parameters and is implemented as a simple list.

The interfaces are offered to all other modules of the sensor node to publish, query and subscribe to parameters. The service component keeps record of the modules which have subscribed to certain parameters and notifies them. The main emphasis is given to the fact that the architecture should be accommodated in least possible amount of memory and its processing overheads should not counter-act the benefits which cross layer design can offer to wireless sensor networks.

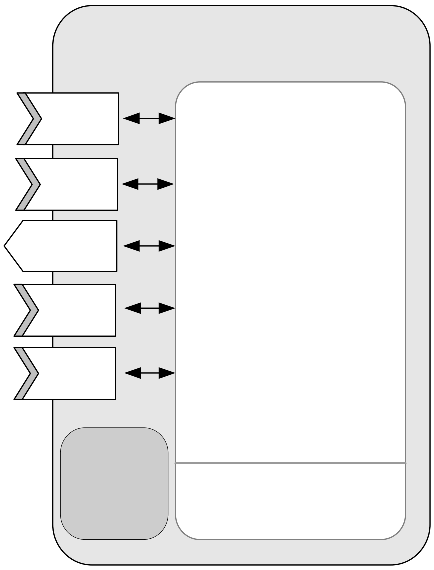


Figure 2: CLAMP Architecture

3.2. CLAMP Database

We define a list of parameters kept by the CLAMP database with their meaning, the owner, the type, and the valid range. The list is non exhaustive and we are still in process of refining these parameters. But what we argue is that it should contain a list of well-defined parameters instead of implementing a service discovery (parameter discovery) routing. The latter would result in processing overheads which is avoided by our approach and facilitates the plug and play feature to a great extent. The list of network parameters and their potential use is given in Table 1.

TABLE 1: CLAMP parameters; profile and potential use

Delay:

Owner: Application Layer (AL)
Description: Delay tolerance defined by the Application Layer. 0 % would mean real time application with strict end to end delay requirements. 100% means that there are absolutely no delay requirements.
Potential Use: The AL defines the delay tolerance and the Routing Layer (RL) may act accordingly, keeping in view e.g. the remaining battery capacity or any other potential parameter. In case of minimum delay tolerance, the RL may decide to use minimum hop routes or less congested routes instead of minimum cost routes to send the information within time restrictions at the cost of more energy consumption.

packetLoss:

Owner: Application Layer (AL)
Description: Packet Loss Tolerance defined by AL.
Potential Use: The AL may set it to notify other layers, regarding its packet loss tolerance and the RL or Mac Layer (ML) can avoid acknowledgment messages, or retransmissions to save some energy.
Additional Considerations: A low value of *packetLoss* may require a Transport layer for end to end reliability and also initiates the need of acknowledged services.

Address:

Owner: Routing Layer (RL)
Description: Logical address set by the RL.
Potential Use: The own address is written to this field and can be used by different layers, e.g. if a wakeup radio is used, and it is woken up by a wake-up signal including the address of the node, so it can decide what to do. The address may change from time to time depending upon RL requirements and hence included here (for example in position based routing address).

Location:

Owner: RL
Description: The location of the node. The location is defined by the longitude and latitude and elevation for some specific location e.g. <03'37"55, 56'13"23, 837m>
Potential Use: The reference or global location information is provided by RL and can be used by different layers, e.g. AL may decide that it has already enough information regarding the required phenomenon in a specific location and it does not need this information from a group of nodes for some time (defining area dominating set [9]). So this group of nodes can go to sleep to save energy. Location information is the basic requirement in geographic routing protocols [10, 35, and 36].

noOfNeighbors :

Owner: RL
Description: Number of neighbors of a node.
Potential Use: This information may be utilized by the ML for synchronization purposes or adaptive division of time slots for accessing the medium.

linkQuality:

Owner: MAC Layer (ML)
Description: Link quality provided by the ML.
Potential Use: For high link quality PL can increase the data rate to exploit the opportunity or it may decrease the transmit power to save energy.

BER:

Owner: ML
Description: bit error ratio, calculated by $-10 \times \log_{10}(BER)$, so 40 means $BER = 10^{-4}$, 70 means $BER = 10^{-7}$
Potential Use: Depending upon *BER*, the PL can increase the output power or it can be compared with packet loss tolerance of the AL and decide what to do.
Additional Consideration: A calculation model of the *BER* is not provided here. A receiver could estimate the *BER* (and/or *SNR*) from a received packet and its count of bit errors. In WSNs this is a major computation effort and therefore usually not provided.

packetLenght:

Owner: ML
Description: Length of the transmit packet defined by ML.
Potential Use: The packet length can effect output power and bit error rate [11]. Short packet sizes results in inefficient energy usage because of large overheads while long packet sizes may experience higher number of errors, so energy efficiency can be maximized by optimal packet size [12]. If packet reception probabilities are used instead of transmission radius for communicating with neighbors, it can be calculated based on bit error ratio and packet length [12].
Additional Consideration: Here we talk about the Packet Length which is actually transmitted over the physical medium as this length is the one which affects different network variables.

Modulation:

Owner: Physical Layer (PL)
Description: Digital modulation technique utilized by the main transceiver.
Potential Use: The modulation at PL can be changed depending upon the remaining capacity of the battery [11]. The number of packets in the system (in buffer or queue or being in transmission) can affect the constellation size of the modulation scheme [13].

SNR: Signal to Noise Ratio (*SNR*) defined by PL

Owner: PL
Description: Signal to noise ratio of the received packet expressed in dB.
Potential Use: If the *SNR* is low and AL has provision of delay tolerance, and the battery capacity is also low, than it can be decided to back off for some time and complete communication later on or otherwise output power can be increased.
Additional Consideration: Usually a transceiver offers an *RSSI* value but one can't measure the noise level. Therefore the *SNR* value will not be provided by most implementations.

dataRate: Data rate defined by PL

Owner: PL
Description: Data rate in kilobytes per second.
Potential Use: The lifetime of the network can be extended by using varying data rate at each node in the routing path. Reducing transmission rates at critical nodes (energy constrained) also results in extended network life time [14]. If data rate is increased, the probability of encountering errors also increases, so a higher output power is required to have an acceptable *SNR* and thus *BER* at the receiver. [15].Based on the data rate requirements, the modulation scheme can be selected [15].

outputPower: Transmit power defined by PL

Owner: PL
Description: transmit power of the radio given in dBm.
Potential Use: The modulation scheme, with certain *BER* threshold values and *SNR* can be used to calculate the transmit power [15]. The optimal transmit power increases with increase in the data rate (vulnerable time is decreased but thermal noise is also increased) [16]. A carefully chosen data rate can have high impact on transmit power and network life time [16].

remainingBatteryCapacity: defined by EMP

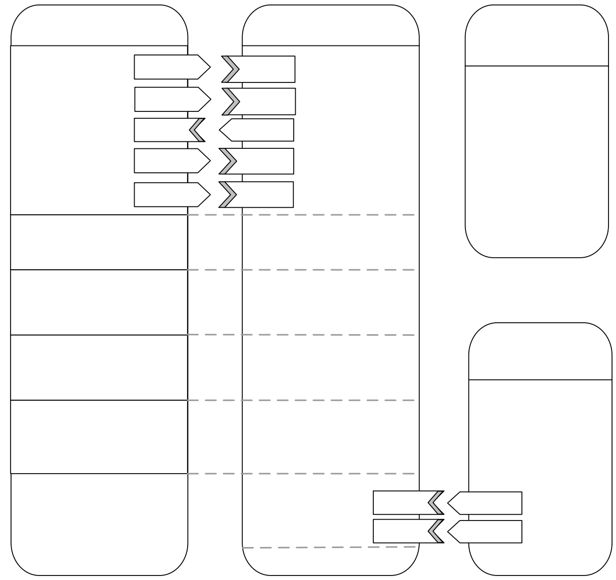
Owner: Energy Management Plane (EMP)
Description: Remaining batter capacity in mWs, i.e. a value of 100 means there are 100mWs. Full scale value means that remaining capacity is more than the value can represent.
Potential Use: If the remaining capacity is at some threshold, than the node can back-off for some time, to allow the battery to recover and than take part in communication. It can also be helpful in case of energy aware routing protocols like EADV [17] or

Mac protocols like CSMA-MPS [20] where the wakeup frequency could be adjusted based on the amount of available energy.

Additional Consideration: Calculating this value is difficult because the voltage characteristic (commonly used to calculate the capacity) of accumulators doesn't offer good estimations. From this value a decision should be done whether we continue to send packets or we send a “I go to sleep” packet. The resolution for such a value should fit for large batteries (10Ah) down to capacitors (1F). Therefore a value giving the relative capacity of the battery (0-100%) would be better suited.

3.3. CLAMP Interfaces and Services

Basic information about the CLAMP interfaces is already provided in [21] but we discuss them here again for the understanding of the reader. The interfaces provided by the CLAMP module comprises *publish*, *update*, *subscribe*, and *query*. When the application is initiated, the CLAMP database does not hold any value related to any parameter. Each module can publish its parameters with the help of *publish* interface. Each module can also update the value of already published parameter with the help of *update* interface. Any module can subscribe to any of the CLAMP parameters with *subscribe* interface. Once a module subscribes to a particular parameter, it is notified by the CLAMP module via CLAMP *services*. Each module provides *onChange* interface to be notified by the CLAMP module by an internal functional call *notify*. If a particular module does not required to be notified too frequently or on every change in the value of the parameter, it can query that particular parameter with the *query* interface provided by CLAMP module. Each module can subscribe to a particular parameter even if it is not published yet. The *services* module stores this information with the client ID and the potential subscriber’s ID. As soon as the parameter is published by the owner, the services module checks for pending requests if any and notifies the concerned module. The similar blackboard concept is recently implemented in the mobility framework [29] for OMNeT ++ [34] for simulation purposes.



The architecture is currently implemented in the PAWiS framework for modeling and simulation of wireless sensor networks as part of the PAWiS project [18, 33]. This uses C++ classes to model the node modules and uses so called functional interfaces to model interfaces (i.e. remote procedure calls) between modules. These functional interfaces are utilized to implement the communications between the network layers as well as to the CLAMP and other planes. 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 descriptions in the preceding sections assume the usage of the PAWiS framework rather than a firmware implementation of a particular sensor node.

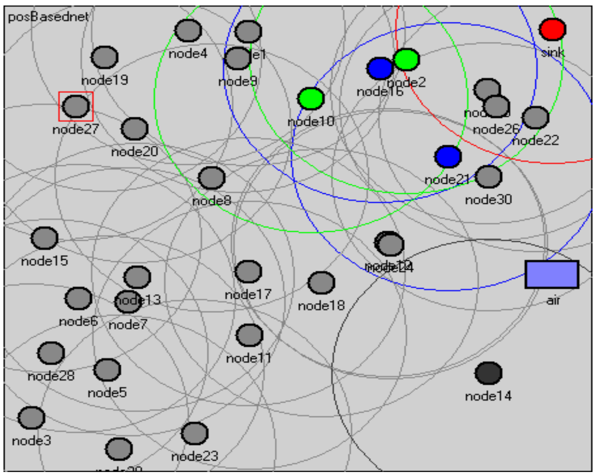
In the firmware different techniques have to be utilized for interfaces and parameters to reduce complexity and overhead. All interfaces should be implemented as regular function calls. CLAMP parameters which are only queried without modules which need immediate notification of changes should be implemented as global variables. For CLAMP parameters with subscribed modules another approach is necessary e.g. call back function calls. In sensor nodes with real time operating system like tinyOS [22] or Contiki OS [37], the architecture can be implemented in more or less the same way by utilizing method invocations and function calls. Detailed information regarding implementation can be found in [21].

5. Preliminary Results

Currently we have simulated the architecture in the PAWiS framework [32]. The current simulation results show that the concept is computationally feasible and can be beneficial for low power wireless sensor networks. We simulated the sensor nodes distributed randomly based on uniform distributed in an area of 200 x 200 m^2 as shown in Figure 4.

Figure 4: Sensor nodes distributed over an area

As an example, we implemented a non-linear remaining battery calculating algorithm [19] in the energy management plane, which updates the approximate remaining battery capacity within the CLAMP database. Based on [19], the authors in [43]



have presented an algorithm for the computation of life time of the battery which is used for the simulation purpose here. The high level analytical model discussed in [19] is given by equation 1.

$$\alpha = \sum_{k=1}^{n} 2I_{k-1} A(L, t_k, t_{k-1}, \beta) \quad \text{where} \qquad (1)$$

$$A(L, t_k, t_{k-1}, \beta) =$$

$$\sqrt{L - t_k - 1}\left[1 + 2\sum_{m=1}^{10}\left(e^{-\frac{\beta^2 m^2}{L - t_{k-1}}} - \frac{\pi e^{-\frac{\beta^2 m^2}{L - t_{k-1}}}}{\pi - 1 + \sqrt{1 + \pi \frac{L - t_{k-1}}{\beta^2 m^2}}}\right)\right] -$$

$$\sqrt{L - t_k}\left[1 + 2\sum_{m=1}^{10}\left(e^{-\frac{\beta^2 m^2}{L - t_k}} - \frac{\pi e^{-\frac{\beta^2 m^2}{L - t_k}}}{\pi - 1 + \sqrt{1 + \pi \frac{L - t_k}{\beta^2 m^2}}}\right)\right]$$

In function A, the series should ideally run from 1 to infinity to give perfect results, but the authors in [19] has proven that running the loop 10 times gives acceptable results. L in this case is the life time of the battery, α is the amount of charge lost by time L, β is the coefficient which covers non linear battery behaviour, t_k is the start time of k^{th} discharge interval, and t_{k-1} is the start time of $(k-1)^{th}$ discharge interval. This remaining battery capacity can now be used by the AL to back off for some time so that the battery can undergo the recovery effect (encircled area in figure 5). Figure 5 shows the graph of the remaining battery capacity against time for three different nodes. Node 3 has higher battery level than node 1 and node 2 because in the simulated topology, node 3 only generated the self traffic and didn't relay any data packets. Figure 5 depicts the normalized battery value of only 3 nodes. We showed only three nodes because the algorithm is computationally complex and does not scale for higher number of nodes (time to simulate increases exponentially because it requires complete historical load profile of the battery).

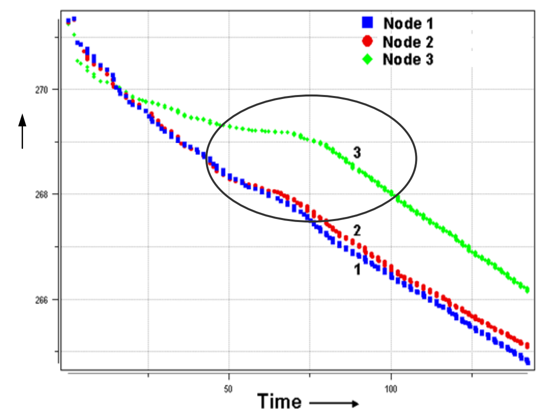


Figure 5: Remaining battery capacity information provided to CLAMP by energy management plane.

For the rest of the simulation we adopted the energy consumption model presented in [42] and given by equation 2 and 3.

$$E_{TX} = E_{elec} \times k + E_{ampl} \times k \times R^2 \qquad (2)$$

$$E_{RX} = E_{elect} \times k \qquad (3)$$

where E_{TX} energy consumed during transmission, E_{RX} is energy consumed during reception, k is packet length, R is the transmission radius, E_{elec} is energy consumed by electronic circuitry and E_{ampl} is energy consumed by amplifier. The values for these parameters were adapted from [42].

This information can be very useful in many aspects if a particular node knows about the battery level of itself and the battery level of its neighbors for routing decisions in energy aware routing protocols.

The remaining battery capacity provided to the routing layer resulted in extended network life time. The batteries were initialized with unrealistic limited amount of energy without loss of generality to reduce the simulation run time. Figure 6 clearly shows that the routing [17] based on remaining battery capacity has extended network life time as compared to the routing based on progress (minimizing the distance towards the fusion center in position based routing schemes) towards the sink node. Both the curves show a downward trend with increasing number of nodes. This is due to the increasing chances of collisions and hence retransmissions at a higher density of nodes. Network life time is considered to be the exhaustion of the first node in the networks.

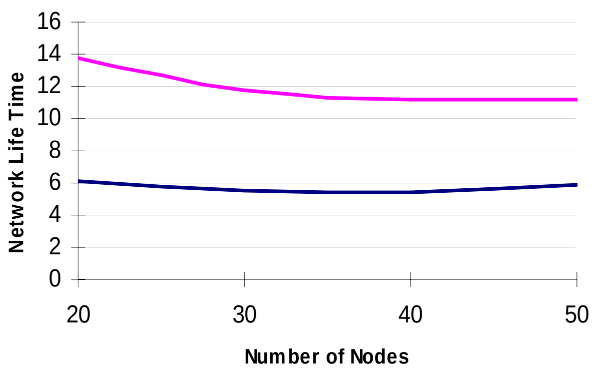


Figure6: Effect of remaining energy aware routing on network life time

Figure 7 shows the remaining energy of each node at the end of the simulation time. It is clear from figure 7 that those nodes running remaining-battery-energy aware routing protocol has higher energy values than those running progress aware routing protocol. The two nodes encircled in red show that almost the same amount of remaining energy, because these nodes only generates self traffic and does not take part in relaying data because of their topological location. The set of available parameters make it possible even to change the behavior of routing schemes at run time. For instance, in environmental application, with periodic sampling of data, the routing decisions can always be taken on the basis of remaining energy to extend the network life time. But incase of an unusual event, like reported temperature is beyond certain upper threshold, routing decisions can be based on delay aware metrics.

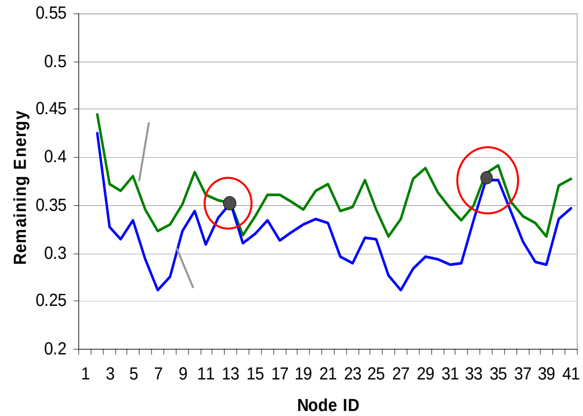


Figure 7: Remaining energy of individual nodes after simulation run time

6. Conclusion and outlook

In this paper we presented architecture for a cross layer management plane as a part of protocol architecture for low power wireless sensor networks. We simulated the architecture and have shown that the concept is computationally feasible and is beneficial. The simulation results show the advantage of using a few example parameters which resulted in extended network lifetime. Having CLAMP parameters can also be used for purposes other than extending network life time. For example, routing can be switched from energy aware routing to delay aware routing based on application requirements. We have not yet realized the proposed scheme on real wireless sensor node platform as we are in a process of fine tuning the parameters to be provided by the CLAMP database for cross layer optimization benefits. A distributed algorithm, which utilizes the cross layer information and provides input to different layers in a standardized way, requires further research [24]. When and how to use cross layer information is a challenging issue [28]. How to utilize synergy between different layers to let the development of adaptive and application independent schemes for different modules requires attention. The implementation of architecture on nodes with light weight operating system can be realized with little effort but would require detailed performance analysis and tradeoffs in terms of energy and memory consumption. The real challenge comes with implementing this architecture on the sensor node without operating system. We intend to implement this architecture on the real node (tinyMote [23] developed at our institute) as part of the PAWiS [18] project funded in part by the Austrian Research Program FIT-IT [26]

Routing based on remaining battery capacity

Routing based on maximizing progress

Acknowledgments

We would like to acknowledge the support of Higher Education Commission of Pakistan and Austrian Research Program FIT-IT.

References

[1] S. Mahlkench and S. A. Madani, “On architecture of low power wireless sensor networks for container tracking and monitoring applications”, INDIN07, Austria, 2007.

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

[3] Van Hoesel, L.; Nieberg, T.; Jian Wu; Havinga, P.J.M. “Prolonging the lifetime of wireless sensor networks by cross-layer interaction”, Wireless Communications, IEEE, Vol.11, Iss.6, Dec. 2004, Pages: 78- 86.

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

[5] Su, W. and Lim, T. L. 2006. Cross-Layer Design and Optimization for Wireless Sensor Networks. In Proceedings of the Seventh ACIS international Conference on Software Engineering, Artificial intelligence, Networking, and Parallel/Distributed Computing (Snpd'06) 2006.

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

[7] Raisinghani, V.T.; Iyer, S. “Cross-layer feedback architecture for mobile device protocol stacks”, Communications Magazine, IEEE, Vol.44, Iss.1, Jan. 2006

[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

[9] Carreras, I.; Chlamtac, I.; Woesner, H.; Zhang, H., “Nomadic sensor networks”, Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.

[10] Zeng, K., Ren, K., Lou, W., and Moran, P. J. Energy-aware geographic routing in lossy wireless sensor networks with environmental energy supply. In Proceedings of the 3rd international Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, August, 2006.

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

[12] Inwhee Joe, “Optimal packet length with energy efficiency for wireless sensor networks”, IEEE International Symposium on Circuits and Systems May 2005.

[13] Schurgers, C.; Aberthorne, O.; Srivastava, M.B., “Modulation scaling for energy aware communication systems”, Low Power Electronics and Design, International Symposium on, 2001., Vol., Iss., 2001,Pages:96-99.

[14] AbouGhazaleh, N.; Lanigan, P.; Gobriel, S.; Mosse, D.; Melhem, R. “Dynamic rate-selection for extending the lifetime of energy-constrained networks”, IEEE International Conference on Performance, Computing, and Communications, 2004 Pages: 553- 558.

[15] Wang, W., Peng, D., Wang, H., and Sharif, H. 2006. Study of an energy efficient multi rate scheme for wireless sensor network MAC protocol. In Proceedings of the 2nd ACM international Workshop on Quality of Service &Amp; Security For Wireless and Mobile Networks, 2006.

[16] Panichpapiboon, S.; Ferrari, G.; Tonguz, O.K. “Optimal Transmit Power in Wireless Sensor Networks”, Mobile Computing, IEEE Transactions on, Vol.5, Iss.10, Oct. 2006 Pages: 1432- 1447.

[17] Mahlknecht, S., Madani, S.A., Roetzer, M. “Energy Aware Distance Vector Routing Scheme for Data Centric Low Power Wireless Sensor Networks”, IEEE International Conference on Industrial Informatics, 2006, Aug. 2006.

[18] http://www.ict.tuwien.ac.at/pawis/.

[19] Rakhmatov, D. N. and Vrudhula, S. B. 2001. An analytical high-level battery model for use in energy management of portable electronic systems. In Proceedings of the 2001 IEEE/ACM international Conference on Computer-Aided Design , 2001.

[20] S. Mahlknecht and Rötzer, „CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks”, in the proceedings of IEEE WFCS, Vienna, Austria, 2004.

[21] Sajjad A. Madani, S. Mahlknecht, Johann Glaser, “A step towards standardization of Wireless Sensor Networks: a layered protocol architecture perspective”, International conference on sensor technologies and applications, SENSORCOMM 2007, Valencia, Spain [Accepted]

[22] www.tinyos.net

[23] http://www.ict.tuwien.ac.at/wireless/

[24] Weilian Su, Tat L. Lim, “ Cross-Layer Design and Optimization for Wireless Sensor Networks”, proceedings of the seventh ACIS internation conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNDP 06), 2006.

[25] Tommaso Melodia, Mehmet C. Vuran, and Dario Pompili, “State of the art in cross layer design for

wireless sensor networks", Wireless Syst./Network Architect. 2005, LNCS 3883, pp. 78–92, 2006.

[26] http://www.fit-it.at accessed on August 20, 2007.

[27] http://www.infineon.com

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

[29] W. Drytkiewicz and S. Sroka and V. Handziski and A. Koepke and H. Karl; A Mobility Framework for OMNeT++; 3rd International OMNeT++ Workshop; 2003.

[30] http://www.iana.org/assignments/port-numbers accessed on August 24, 2007.

[31] www.sockets.com/services.htm accessed on August 24, 2007.

[32] Daniel Weber and Johann Glaser and Stefan Mahlknecht; Discrete Event Simulation Framework for Power Aware Wireless Sensor Networks; Proceedings of the 5th International Conference on Industrial Informatics, INDIN 2007; 2007; 1; 335-340; Vienna, Austria.

[33] Stankovic, J. A. 2004. Research challenges for wireless sensor networks. SIGBED Rev. 1, 2 (Jul. 2004), 9-12.

[34] www.omnetpp.org accessed on August 20, 2007.

[35] Martin Mauve and J¨org Widmer and Hannes Hartenstein.; A survey on position-based routing in mobile ad-hoc networks; IEEE Network Magazine; 2001.

[36] I. Stojmenovic; Position-based routing in ad hoc networks; IEEE Commun. Magazine; 2002.

[37] Adam Dunkels and Björn Grönvall and Thiemo Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I), November 2004.

[38] C.S. Raghavendra, Krishna M. Sivalingam and Taieb Znati, "Wireless Sensor Networks", Kluwer Academic Publishers, 2004.

[39] Ulema, M., Nogueira, J. M., and Kozbe, B. 2006. Management of Wireless Ad Hoc Networks and Wireless Sensor Networks. J. Netw. Syst. Manage. 14, 3, Sep. 2006.

[40] http://www.hicss.hawaii.edu/Hicss38/ tutorial_description.htm accessed on October 19, 2006.

[41] 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.

[42] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. Hawaii International Conference on System Sciences Jan. 2000.