A Multi-Criteria Message Forwarding Architecture for Wireless Sensor Networks[*]

Manoussos Athanassoulis, Ioannis Alagiannis and Stathes Hadjiefthymiades

University of Athens,
Department of Informatics and Telecommunications,
Pervasive Computing Research Group,
Panepistimiopolis, Ilissia, 15784 Athens, Greece
`{m.athanassoulis, i.alagiannis, shadj}@di.uoa.gr`

Abstract. Wireless Sensor Networks (WSN) comprise a fast-developing research area with a vast spectrum of applications. A WSN design is influenced by many factors such as transmission errors, network topology and power consumption. Consequently, developing a WSN application introduces several implementation challenges. In this paper, we describe a multi-criteria architecture in order to achieve energy-aware and consistent message forwarding over a WSN. Using the proposed architecture a directed acyclic graph (DAG) is formed throughout the WSN. Such DAG is used for multi-source data aggregation to a single sink. Intermediate nodes evaluate their energy reserve and induced error and decide whether message retransmission is needed. A sink is necessary in order to collect process and probably forward these data to a more sophisticated system for further processing. The discussed architecture is developed using TinyOS, an event-driven lightweight operating system designed for sensor network nodes, and nesC, a highly modular and declarative extension of C.

1 Introduction

The recent advances in highly integrated digital electronics and wireless communication technology have led to the development of low cost, large-scale and low power sensor networks. Such networks are composed by a large number of micro-sensor nodes, which are equipped with communication and minimal computation capabilities. Sensor nodes are able to monitor a wide variety of physical parameters such as temperature, humidity, light, radiation, noise, etc., and report them using ad hoc network protocols and algorithms. The capabilities of sensor networks have significant impact on numerous application areas with varying requirements and characteristics in our life such as military control and communications; environment forecast systems, forest fire detection, medical treatment, as well as, traffic control and security. In the future, sensors collecting data will become really ubiquitous i.e., be found everywhere; in machines, buildings, even on our clothes.

The constraints of sensor nodes make the problem of designing and management of a WSN very challenging. Firstly, sensors have limited resources such as battery life-

---

[*] This work is supported by the University of Athens, Research Grants Committee (project number: 70/4/7819, “Kapodistrias”)

time (varying from hours to several years depending on the application), computational power, data storage and communication bandwidth. Hence, it is important for a WSN architecture to take into consideration the network's topology, power consumption, data rate and fault tolerance in order to avoid significant energy consumption and improve bandwidth utilization.

In this paper we propose a multi-criteria message forwarding scheme over the WSN. Whenever a new sensor reading arrives at a certain WSN node, a decision is taken whether to forward this further on or not. The decision criteria are the current energy reserve, data consistency as well as time constraints. The assumed topology of the WSN is a directed acyclic graph. The nodes comprising the considered WSN are divided to three different types[†]:

- Sensing nodes
- Communication (relay) nodes
- Sink node (single-sink topology)

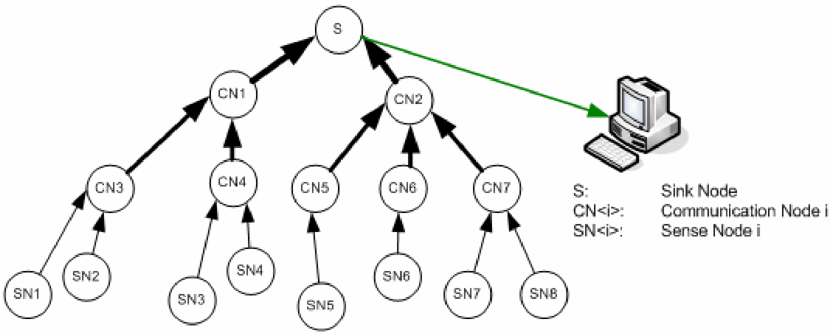

Fig. 1. WSN deployment

Sensing nodes perform the measuring task and provide their readings to a pre-assigned communication node (we assume a non-volatile network topology) according to the discussed criteria. Communication nodes receive data either from sensing nodes or from other communication nodes. Such data are forwarded, using the above criteria, towards the sink node or straight to the sink node depending on their exact network position and using already known paths (established by means of the underlying routing scheme). Finally, the sink is a single node which collects the sensing task results and connects the WSN to the outside world enabling advanced data processing.

The forwarding decision for a received message (and of transmitting the measurement in case of a sensing node) is based upon the energy state of current node, the induced error (by not forwarding it) and the elapsed time from the next obligatory reading transmission. Such criteria are quantified using a Utility function, described in detail in Section 3, *System Architecture*. The proposed architecture does not apply a retransmission policy upon message loss. Instead, data are approximated using an arithmetic method (e.g., Lagrange, Least Squares) based on recently received measurements. The same arithmetic method is used to approximate data if forwarding does not take place according to the thresholding of the utility function.

[†] We assume the mobility of nodes is relatively limited.

The main goal of this architecture is to ensure long lifetime of a WSN application. This is achieved by reducing unnecessary transmissions over the network while preserving the induced error in acceptable levels. The presented architecture can cover a wide variety of application requirements and can be further optimized through data aggregation, subject to the peculiarities of the observed physical parameters and WSN spatial distribution.

The rest of the paper is organized as follows. Section 2 refers to existing protocols and algorithms, for energy aware routing. In Section 3 we present our multi-criteria message forwarding architecture and describe the adopted utility function. Section 4 is dedicated to the presentation of implementation details about development and energy awareness issues. Finally, our ideas for future work are summarized in Section 5.

2 Prior and Related Work

In the recent years a lot of research has been conducted on WSNs. Numerous articles have been published describing new algorithms, routing protocols and architectures aiming at maximizing sensor networks' lifetime, having as main issue energy awareness. To minimize energy consumption, already proposed routing techniques ([2], [3]) for WSNs, employ routing tactics such as data aggregation, in-network processing, clustering, different node role assignment and data-centric methods. There are several ways of categorizing these protocols and algorithms. For example, they can be discriminated depending on the network structure to Flat Networks Routing (FNR), Hierarchical Networks Routing and Location-based Routing [2]. FNR is also referred as Data-centric routing [3]. Several representative protocols are mentioned below.

Directed Diffusion: Intanagonwiwat et al. [11] proposed a data-centric (i.e. all communication is for named-data) and application-aware paradigm aiming at avoiding unnecessary operations of network layer routing in order to save energy by selecting empirically good paths and by caching and processing data within the network.

COUGAR: Another data-centric protocol is proposed by Yao and Gehrke [12]. COUGAR proposes an architecture which considers the network as a huge distributed database system. The architecture provides in-network computation which ensures energy efficiency in situations when the number of sensors generating and sending data to the leader (which is a node selected to perform aggregation and transmit data to a sink) is extensive.

Energy Aware Routing: The protocol proposed by Shah and Rabaey [13] although similar to Directed Diffusion it differs in the sense that it uses occasionally sub-optimal paths to provide substantial gains. These (minimum-energy) paths are chosen by means of a probability function. This protocol can achieve longer network lifetime as energy is dissipated more equally among all nodes.

TEEN and APTEEN: These two hierarchical routing protocols are proposed by Manjeshwar and Agarwal [16]. TEEN (Threshold-sensitive Energy Efficient sensor Network protocol) and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) are suitable for time-critical applications. The key factor for both protocols is the value of the measured attribute. If this value is out of the range of interest no transmission is performed. Moreover, if there is no change in the measured value or the change is insignificant no transmission is needed and, consequently, no transmission occurs. The additional feature of APTEEN is the capability of changing the periodicity and the parameters of TEEN according to user and application needs.

Apart from routing protocols, PowerTOSSIM [4], a WSN simulation tool has recently been launched. PowerTOSSIM is a scalable simulation environment for WSN that provides an accurate, per-node estimate of power consumption. PowerTOSSIM is an extension of TOSSIM ([5], [6], [7]), an event-driven simulation for TinyOS [10] applications. PowerTOSSIM estimates the number of CPU cycles executed by each node and includes a detailed model of hardware energy consumption based on the Mica2 sensor node platform.

3 System Architecture

The considered system architecture relies on three roles of sensor nodes:

- Sensing nodes (or sources) that sense certain physical parameters and transmit the relevant information towards other nodes in the infrastructure.
- Communication (or relay) nodes that, wirelessly, receive readings from sensing nodes (or other communication nodes) and relay them upstream towards the final recipient of such information. Communication nodes come into play whenever direct network connectivity is not feasible (due to limited resources such as power in the radio interface) and bridge the, otherwise inaccessible, nodes.
- Sink nodes that are the final recipients of the sensed information. Sink nodes are typically connected to conventional computing equipment for complex processing of the accumulated readings. Alternatively, sink nodes may be attached to another, more elaborate network topology (e.g., a WLAN or a fixed network) for further forwarding.

As already discussed, the aforementioned nodes form a directed acyclic graph, a rooted tree structure. The root of the tree is the sink node (exactly one node), all other nodes may assume the role of sensing nodes (at least one node is required), or communication nodes. In Fig. 2, the sink node is N6, nodes (leafs) N10, N9, N8, N5, N1 and N2 are sensing nodes. All remaining nodes could be sensing nodes, but surely serve as communication nodes for the forwarding of messages upstream (i.e., towards N6). The discussed topology has been formulated by means of a WSN routing protocol ([1]), not discussed in this paper.

Each intermediate node in the WSN topology (i.e., nodes N3, N4, N7) reserves memory space proportional to the number of distinct data flows (DF) that it serves. A DF is an association of a certain leaf node with the root of the topology. For example, node N4 serves three distinct DF, namely the associations N1↔N6, N2↔N6 and N5↔N6. The reserved memory space in each communication node enables the formulation of buffers (per DF) that hold a certain number of sensor readings. Leaf nodes reserve similar space only for their DF.

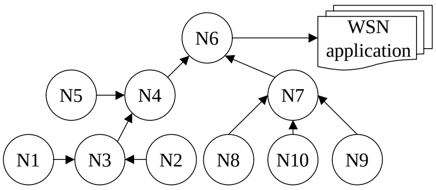

Fig. 2. WSN topology

Sensing and communication nodes try to optimise the energy consumption within the WSN. Specifically, whenever the need for the upstream transmission of a new measurement message arises, an embedded control mechanism (Transmission Control Mechanism, TCM) determines the utility for the said transmission. The TCM takes certain criteria into account and may decide not to propagate the considered message upstream. The peer TCM (i.e., the TCM found in the next node upstream) should be able to conceive this situation and react accordingly. Below, we elaborate on the criteria considered by the TCM for assessing the utility of message transmission. Before elaborating further on the TCM criteria, we should note that all the TCMs in the WSN operate in a synchronised manner (i.e., transmit messages within a specific time period) and tolerate small deviations from the network-wide clocking.

Since the discussed message-forwarding scheme relies on the conditional transmission of information within the WSN, WSN nodes should be able to determine whether downstream nodes are alive. In this respect, a Heart-Beat (HB) message is introduced. HB messages contain sensor readings and are transmitted regularly and unconditionally from the sensing (leaf) nodes. The HB messages are transmitted using reliable transport services to ensure their delivery within the architecture. HB messages should be relayed unconditionally from all communication nodes. Whenever, a HB message is lost (i.e., is not received within the predetermined clocking period), the upstream node deems that some WSN node has failed. Similar, “I-am-alive” messages may exist on the routing protocol level, but are not exploited since they cannot facilitate the message forwarding scheme (as they do not convey application related traffic i.e., sensor readings). The TCMs are fully cognisant of the HB message status, i.e., know the elapsed time from the previous HB and the estimated time until the next HB.

Each TCM implements an extrapolation scheme on the received sensor readings. The monitored physical parameter is assumed to vary smoothly over time (e.g., as a polynomial function of time). Whenever a new measurement is presented to the TCM, the latter entity determines whether the peer TCM (in the upstream path) can reproduce the new value without, explicitly, receiving it. To achieve this objective, the a-priori agreed extrapolation scheme (common throughout the WSN) is engaged. The local TCM calculates an extrapolated value (EV) for the sensed physical variable using previous measurements (stored in the DF memory space discussed before). The EV is compared against the actual, new measurement and the relevant error is calculated. The estimated error level will contribute to the determination of the message transmission utility. If the message is not transmitted upstream, then the peer TCM will perform the same extrapolation calculation and consider the (locally estimated) EV as the new received measurement. The mandatory forwarding of HB messages avoids an unconstrained error increase spatially and temporally. This scheme is applied for all the DFs handled by the considered node.

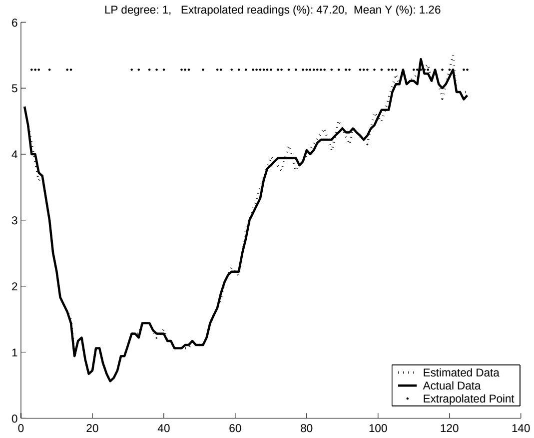

Fig. 3: Lagrange extrapolation (degree *n=1*)

The extrapolation scheme used in the initial prototype of this architecture is Lagrange Polynomial (LP). This design option is driven by the need for a low complexity, accurate, extrapolation formula. A high complexity scheme is incompatible with the resource available in a sensor node (i.e., RAM ~512 bytes, limited CPU capacity, low energy reserve). The computation of the EV is based upon the *n-1* more recently received readings, resulting to an *n* degree extrapolation polynomial. The LP (of degree *n*) is provided below:

$$P(t)=\sum_{j=0}^{n} P_j(t)\cdot f_j,\ where\ P_j(t)=\prod_{\substack{k=0\\ k\neq j}}^{n}\frac{t-t_k}{t_j-t_k} \qquad (1)$$

f_i are the sensor readings, $P_j(t)$ are Lagrange coefficients, $P(t)$ is the extrapolated value and t is time. In our implementation of the LP, the time difference between two consecutive readings t_i and t_{i+1} ($t_{i+1} - t_i = \delta t$) is fixed and the extrapolated value, $P(t)$, is the value corresponding to the next sensor reading.

We have simulated the performance of the discussed extrapolation scheme for outdoors temperature readings (coming from a meteorological system) and different degrees of the LP. In Fig.3 and Fig.4 we plot the actual sensor data along with the extrapolated information. We monitor an error metric *Y, calculated as follow*s.

$$Y=\left(\left|d(i)-est(i)\right|/\left|d(i)\right|\right) \qquad (2)$$

d(i) denotes actual reading (entry *i*) and *est(i)* the extrapolated values. Whenever *Y* exceeds a predefined threshold, *T*, (in our case T = 5%) extrapolation is not performed and the plotted value matches the actual data. *Q* denotes the percentage of extrapolated readings. The optimum parameterization of the extrapolation scheme (selection of LP degree and *T*) should yield a high *Q* and a low *Y* value.

In our simulations, we observed that a high LP degree (high extrapolation accuracy) achieves lower *Y* and *Q* values, in contrast to a low LP degree. Hence, the adopted LP degree and the error threshold *E* are application specific and should be selected in an ad-hoc manner.

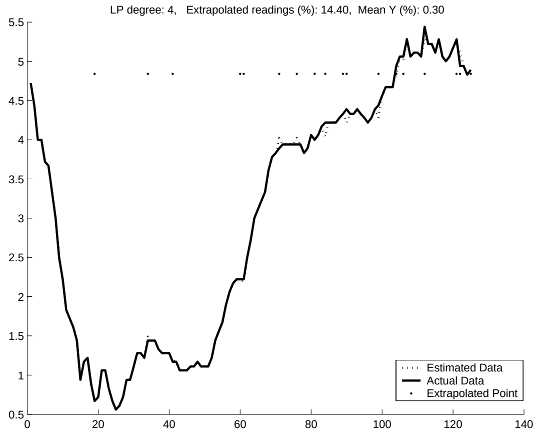

Fig. 4: Lagrange extrapolation (degree *n=4*)

Another parameter that affects the decisions of the TCM is the current energy reserve of the considered node. Each node is equipped with some energy source (e.g., a solar cell) that re-charges the onboard battery, thus, elongating the node's effective lifetime. The TCM knows the exact energy reserve and takes it into account when assessing the utility of an additional message transmission.

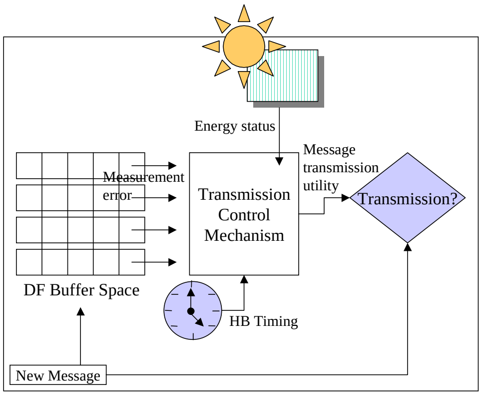

Fig. 5. Transmission Control Mechanism

3.1 Discussion on the Utility function design

In this section we elaborate on the scheme that TCM adopts for the assessment of the utility of a message transmission upstream. Let m_i denote a message (i.e. actual readings – entry *i*), *M* the set of all messages and *H* the set of Heart-Beat messages. Moreover, let U_k denote the utility of the sensor node k with respect to the transmission of a new (not HB) message upstream. U_k is a function of time, the current node energy reserve and the received measurement for a certain DF. U_k is calculated as follows:

$$U_k = w_1 \cdot U^k_{energy} + w_2 \cdot U^k_{error} + (1 - \sum_{i=1,2} w_i) \cdot U^k_{time},\ \sum_{i=1,2} w_i \le 1,\ \ m_i \in \left(M - H\right) \qquad (3)$$

$$U_k = 1,\ \ m_i \in H \qquad (4)$$

The weights w_i are application specific and non-negative. The three utility components, for a given sensor node k, are calculated as follows:

$$U^k_{energy} = 1 - e^{\left[-10\frac{E}{E_{\max}}\right]} \qquad (5)$$

$$U^k_{error} = \begin{cases} err^2 / \left(err_{threshold}\right)^2 & , \ \ err \le err_{threshold} \\ 1 & , \ \ err \ge err_{threshold} \end{cases} \qquad (6)$$

$$U^k_{time} = \begin{cases} 2 \cdot \Delta t / \Delta \mathrm{T} & , \ \ 0 \le \Delta t \le \Delta \mathrm{T}/2 \\ -2 \cdot \Delta t / \Delta \mathrm{T} + 2 & , \ \ \Delta \mathrm{T}/2 \le \Delta t \le \Delta \mathrm{T} \end{cases} \qquad (7)$$

where E denotes the current energy reserve of the considered node, E_{max} is the maximum energy quantity that can be accumulated in the node, err denotes the error induced in the measurement sequence by the extrapolation scheme that is globally adopted throughout the WSN topology, $err_{threshold}$ is the maximum tolerable deviation that can be induced in the collected readings, ΔT is the HB interval and Δt is the time that elapsed from the previous HB message transmission.

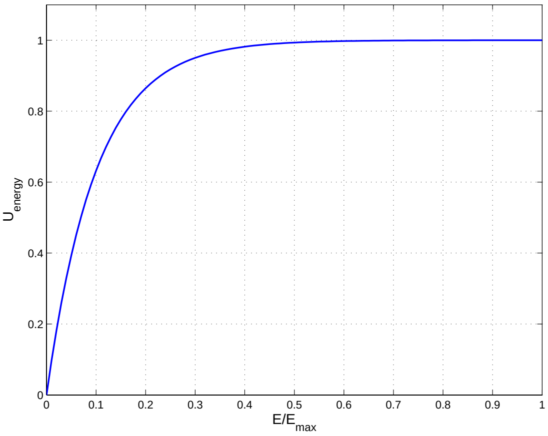

Fig. 6. Utility component (Energy)

The three utility components provide a full synopsis of the current status of the WSN, i.e., the energy component reflects the status of the node, the error component reflects the variance within a DF, and the time component reflects the clocking status of the entire topology.

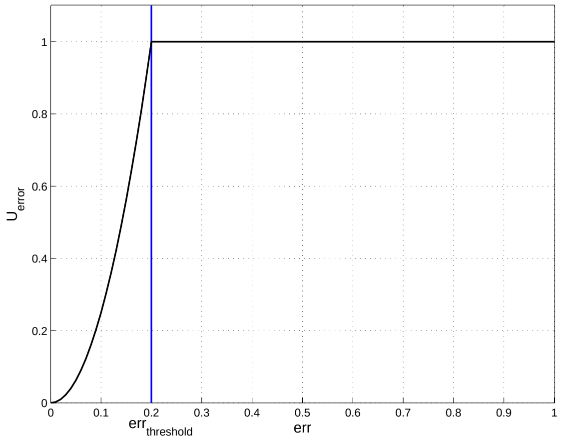

Fig. 7. Utility component (Error)

Whenever the utility for a given sensor node k drops below an application specific threshold *g*, the sensor node halts upstream message re-transmission. Hence, the control condition for intelligent, energy aware message forwarding is:

$$U_k \geq g > 0 \tag{8}$$

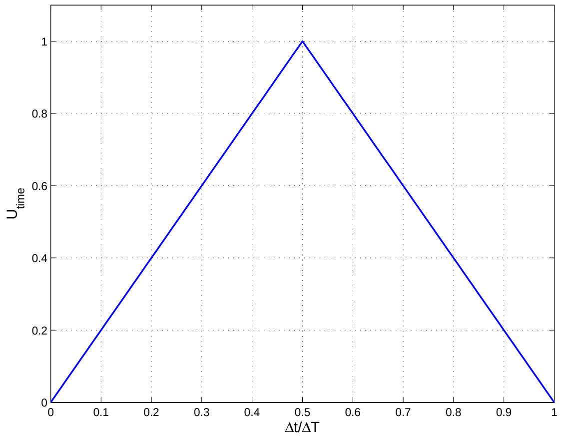

Fig. 8. Utility Component (Time)

4 Implementation Details

In this section we discuss some issues pertaining to the implementation of the proposed architecture. Specifically, we describe the underlying operating system, the adopted simulation platform and provide details for the energy state model of a WSN node.

4.1 System Software

TinyOS, an event-driven operating system specifically designed for sensor networks, has been used to develop several parts of the proposed multi-criteria message forwarding scheme. TinyOS has become a popular environment for experimenting with and developing sensor network applications. This is due to its modular nature, support for several common sensor node platforms and ability to operate having very limited resources (e.g., 8Kbytes of program memory, 512 bytes of RAM). A TinyOS program is a graph of components (independent entities). It is a component-based runtime environment which has been developed using the nesC language. NesC ([8], [9], [17]) is an extension of C that provides support for the TinyOS component and concurrency model and all the low-level features necessary for accessing hardware.

TinyOS supports a simulation environment, called TOSSIM (TinyOS SIMulator), which has been used to test some parts of the proposed architecture. TOSSIM is a discrete event simulator which executes the code that is intended for the WSN node but on PC hardware. Following the successful debugging/simulation (energy aware) of the intelligent message forwarding scheme in TOSSIM we have deployed the derived nesC components in our (Berkeley) Motes testbed which covers all the roles indicated in figures 1 and 2 and discussed throughout the text.

4.2 Energy Model

Communication between nodes, as already stated, is much more energy consuming than executing CPU instructions. A transmission of one byte consumes the same energy as, approximately, 11000 cycles of computation. The energy reserve of each node is reduced according to the computational complexity of the executed code and the number and duration of radio transmissions. Furthermore, the energy reserve of a node could increase using an energy source, for example, a solar cell. In this paper we have simulated such a scheme using TOSSIM. We may securely assume that a solar energy source provides a stochastic energy intake modeled as a Rayleigh distributed random variable with mean value equal to 9.6 W/cm^2. Such value is reported in the engineering literature as the maximum energy yield of a solar cell of centimeter dimensions which could fit on a sensor board.

5 Conclusions and Future Work

In this paper we have presented a multi-criteria message forwarding architecture. The goal of the proposed architecture is to reduce energy consumption by avoiding unnecessary message transmissions. Energy awareness in WSNs is an emerging research area and the protocols presented in the relevant literature are focused on determining low-cost paths within the existing network. On the other hand, we try to avoid in-network transmissions if the induced error is acceptable. A combination of both techniques would lead to better results ensuring the prolongation of a WSN application’s lifetime. Two protocols that could be combined with the proposed architecture are Energy Aware Routing and TEEN.

Another issue is the exact model of power estimation. PowerTOSSIM includes a detailed model of sensor energy consumption. It could be incorporated to our scheme in order to get a more realistic estimate of node-level energy consumption.

Finally, we believe that it is very important to evaluate the responsiveness of our architecture to increased node mobility. Node mobility is a prerequisite for some WSN applications, thus, resulting to even more demanding energy awareness and routing protocols. Moreover, we plan to implement intelligent data aggregation schemes to be embedded in the communication nodes. Such schemes may significantly reduce the upstream communication requirements by merging DF at a certain level within a WSN hierarchy. The applicability of the aggregation model is closely related to the nature of the monitored physical variables, the spatial WSN node distribution and temporal correlation of upstream messages.

References

1. I. F. Akyildiz et al., “Wireless sensor networks: a survey”, *Computer Networks, Vol. 38, pp. 393-422, March 2002.*
2. N. Al-Karaki and A.E. Kamal, “Routing Techniques in Wireless Sensor Networks: a survey”, *IEEE Wireless Commnications, December 2004*
3. K. Akkaya and M. Younis, “A survey on Routing Protocols for Wireless Sensor Networks”, *Elsevier Journal of Ad Hoc Networks*
4. V. Shnayder et al., “Simulating the Power Consumption of Large Scale Sensor Network Applications”, *proceeding of ACM Conference on Embedded Network Sensor Systems (SenSys), November 2004*
5. P. Levis et al., “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, *ACM Conference on Embedded Network Sensor Systems (SenSys), November 2003*
6. P. Levis and N. Lee “TOSSIM: A Simulator for TinyOS Networks”, *TinyOS Documentation*
7. A. Sridharan et al., “Integrating Environment Simulators with Network Simulators”
8. D. Gay et al., “nesC 1.1 Language Reference Manual”
9. D. Gay et al., “The nesC Language: A Holistic Approach to Networked Embedded Systems”, *ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI), San Diego, California, June, 2003*
10. www.tinyos.net
11. C. Intanagonwiwat et al., "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *Proceedings of ACM MobiCom '00, Boston, MA, 2000.*
12. Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks”, *in ACM SIGMOD Record, September 2002*.
13. R. C. Shah and J. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks”, *IEEE Wireless Communications and Networking Conference (WCNC), March 2002, Orlando, FL.*
14. F. Zhao and L. Guibas “Wireless Sensor Networks: An Information Processing Approach”, *Elsevier, Morgan Kaufmann Publishers*
15. J. Hill, “System Architecture for Wireless Sensor Networks”, *PhD Thesis, Spring 2003*
16. A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
17. nescc.sourceforge.net